Communication and Block Game in Cognitive Radio Networks

Haosen Pu
Tsinghua University
Beijing, P.R. China
phs199205@gmail.com

Zhaoquan Gu
Tsinghua University
Beijing, P.R. China
demin456@gmail.com

Qiang-Sheng Hua*
Huazhong University of Science and Technology
Wuhan, P.R. China
qshua@hust.edu.cn

Hai Jin
Huazhong University of Science and Technology
Wuhan, P.R. China
hjin@hust.edu.cn

ABSTRACT

In this paper, we initiate the Communication and Block Game between two unlicensed users and an adversary in Cognitive Radio Networks (CRNs). In each time slot, the two unlicensed users can successfully communicate on the common available channel if it is not blocked by the adversary. In the communication and block game, the two unlicensed users aim to maximize their communication load, denoted as the number of time slots of their successful communications, while the adversary aims to minimize it. We propose efficient algorithms for both users and the adversary and we prove the proposed algorithms will lead a Nash Equilibrium, i.e. the users can achieve the maximum communication load against any adversary's blocking strategy, while the adversary can minimize the users' communication load against any users' channel accessing strategy. We also present efficient algorithms for both users and adversary for the multiple channels scenario where the users and the adversary are equipped with multiple radios. These algorithms also guarantee high communication load for the users, while the adversary can also block a considerable number of users' communications. Our simulations validate the theoretical analyses.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

Keywords

Cognitive Radio Networks; Rendezvous; Game Theory

1. INTRODUCTION

Due to the increasing numbers of wireless devices and large amount of wireless service, the unlicensed spectrum has been overutilized while the utilization of licensed spectrum is pretty low [1]. Cognitive Radio Networks (CRNs) are thus proposed to alleviate the spectrum scarcity problem where the unlicensed users are equipped with cognitive radios to exploit and access the portion of the licensed spectrum that is not occupied by any nearby licensed users. Unless otherwise specified, 'users' mentioned in the paper refers to the unlicensed users.

In constructing a CRN, *rendezvous* is the fundamental process to construct a communication link on some licensed channel [12,20]. Technically speaking, the licensed spectrum is assumed to be divided into n non-overlapping channels, and time is assumed to be divided into slots of equal length. In each time slot, the user can access one *available* channel that is not occupied by the licensed users, and two neighboring users achieve rendezvous if they choose the same channel in the same time slot. The state-of-the-art results guarantee rendezvous in $O(n^2)$ time slots [8].

When two users rendezvous on some channel, they can establish a communication link and exchange information through it. However, if an adversary exists in the network who can listen and block the channel in each time slot [7,11,14,15,17], the communication link is disrupted if the adversary blocks the rendezvous channel. Therefore, the users should choose the other available channels for further communication. Correspondingly, when the users are able to construct communication links on different available channels, the adversary also desires to block the communication between two users in as many time slots as possible.

In this paper, we study the communication and block game between two users and an adversary, where the users can communicate through a common available channel if it is not blocked by the adversary. Generally, we assume an adversary joins the network after two users find out the common available channels via the first rendezvous. The adversary is not aware of the users' available channels, but it can listen or block one channel in each time slot. The users are able to establish communication links on different common available channels in different time slots, but the link could not work if the channel is blocked by the adver-

*Corresponding Author

MSWiM'15, November 2–6, 2015, Cancun, Mexico.
© 2015 ACM. ISBN 978-1-4503-3762-5/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2811587.2811599.

sary meantime. Therefore, the users in the problem aim to maximize the communication load, which is defined to be the number of successful communication time slots against the adversary; and the adversary aims to block the users' communication in as many time slots as possible, i.e. to minimize the users' communication load. In addition, we also extend the problem to the multiple channels scenarios, where the users and the adversary are equipped with multiple radios to sense and access (or to block) multiple channels in one time slot.

In tackling the problem, we face many challenges. First, even though the users are aware of the set of common available channels, they cannot communicate through a predefined channel accessing sequence based on the set, because once the adversary finds out the sequence they can never communicate. Second, when the adversary successfully blocks one time slot's communication between the users, it is a hard choice to block the same channel or the other channels in the next time slot, since the users would also make the same intellectual decision. Third, the users and the adversary cannot be aware of the other's strategy, and thus the designed algorithms for the users (or the adversary) should work efficiently for any adversary's (or the users') strategies. Fourth, when the users or the adversary can access or can block multiple channels, the communication and block game is much harder, especially for the adversary who has to block all possible communication channels.

In this paper, we propose both algorithms for the users and the adversary that would be a Nash Equilibrium for them. Moreover, we also propose efficient algorithms for the multiple channels scenario that achieve more communication load for users and behave more stable for the adversary. The contributions of the paper are summarized as follows:

- We design an efficient algorithm for the users against any adversary's strategy such that in a long run of T time slots, two users can achieve communication load no less than $\frac{1}{4}T$ when there is only one common available channel, and can achieve communication load no less than $(1-\frac{1}{M})T$ when there are M common available channels;

- We introduce an efficient algorithm for the adversary against any users' strategy. We show that the algorithm works best against the proposed algorithm for the users such that the users' communication load matches the above mentioned lower bound ($\frac{1}{4}T$ and $(1-\frac{1}{M})T$ when there is only one common available channel and there are M ($M > 1$) common available channels, respectively);

- We show that the proposed algorithms for the users and the adversary would be a Nash Equilibrium when $M \geq 2$, i.e. the users can achieve the maximum communication load against any adversary's blocking strategy, while the adversary can minimize the users' communication load against any users' channel accessing strategy;

- We also present efficient algorithms for both users and adversary for the multiple channels scenario, which guarantees high communication load for the users, while the adversary can also block a considerable number of users' communications.

The rest of the paper is organized as follows. We introduce some related works in the next section and the preliminaries are provided in Section 3. In Section 4, we introduce the algorithm for the users which work efficiently against any adversary's blocking strategy, and we present the algorithm for the adversary in Section 5. Then we show the proposed algorithms are a Nash Equilibrium in Section 6. Moreover, we present the proposed algorithms under the multiple channels scenario in Section 7. In Section 8, we conduct simulations to evaluate our proposed algorithms and the paper is concluded in Section 9.

2. RELATED WORKS

Though many elegant algorithms for the rendezvous problem in constructing a CRN [4,8] have been proposed, to the best of our knowledge, no existing works have considered the problem of maximizing the communication load after rendezvous happens and an adversary may exist in the network. Some works aim to maximize the rendezvous diversity such that the users may achieve rendezvous on all available channels [3], while some works study how an adversary or a jammer could influence the rendezvous process [11,15], but none of them considered the communication and block game in this paper.

In order to simplify the rendezvous process, some rendezvous algorithms employ a central controller or a dedicated common control channel(CCC) through which the users can make agreement on the schedule of the channels. But these methods suffer from several issues: vulnerable to attack, expensive for establishment as well as low flexibility. Therefore, many distributed rendezvous algorithms have been proposed, where the users generate a hopping sequence of the available channels and access the channels by the sequence.

There are two types of such channel hopping based rendezvous algorithms. If the users construct the same hopping sequences based on all channels, we call it global sequence (GS) based algorithms [8,13,18], otherwise, the algorithms constructing hopping sequences on the basis of the users' local information are referred to as local sequence (LS) based algorithms [4,5,9].

There are three state-of-the-art GS based algorithms: The Channel Rendezvous Sequence (CRSEQ) [18] algorithm constructs the hopping sequence based on triangle number and modular operation; the Jump-Stay(JS) algorithm [13] designs both jump frames and stay frames to guarantee rendezvous; and the Disjoint Relaxed Different Set (DRDS) algorithm constructs the hopping sequence by showing its equivalence to the carefully designed disjoint relaxed different set. These algorithms could guarantee rendezvous in $O(n^2)$ time slots if there are n channels in total.

Different from GS based algorithms, LS based algorithms construct the hopping sequences based on the users' available channels and their distinct identifiers (IDs). Alternate Hop-and-Wait (AHW) [5], Modified Local Sequence (MLS) [9] and Conversion Based Hopping (CBH) [10] convert the users' IDs to facilitate the design of different hopping sequences of different users. Another efficient algorithm without using the users' ID is proposed in [4], which constructs the hopping sequences based on graph coloring and the design of Catalan strings. However, all these algorithms only focus on the first time they rendezvous, none of them study the communication process against an adversary after rendezvous.

If an adversary exists in the network, some works have analyzed its impact on rendezvous. In [15], an adversary who can sense and block a channel in each time slot exists when two users try to achieve rendezvous using the JS algorithm. By letting the adversary estimate the hopping sequence in the first jumping stage, Channel Detecting Jamming Attacks (CDJAs) proposed in [15] reduces the successful rendezvous rate of users from 100% to 20% . And in [7], Multi-Radio Channel Detecting Jamming Attack (MRCDJA) is proposed to generalize CDJAs into multiple channel cases(the adversary can access multiple channels simultaneously), which can figure out the hopping sequence in $O(\frac{M}{n})$ expected time which is n times better than that in [15].

In [11,14,17], algorithms of users based on quorum-system are proposed to maximize the probability two users successfully rendezvous. In [11], Frequency Quorum Rendezvous (FQR) is presented based on quorum system to establish a common key for future communication under jamming attack, which works about 40% better than random hopping(RH) and Pseudo-random Frequency Hopping (PFH) [19]. In [14], the authors point out that FQR algorithms are still vulnerable against some smart adversary. An efficient adversary is presented in [14], which decreases the success rate of FQR to 35%, moreover they design Role-based Frequency Rendezvous (RFR) scheme based on different roles of users to achieve steady result of more than 90%. In [17], a game between users and a jammer is proposed and several pure strategies are introduced. The authors also show that the rendezvous performance also depends on whether the receiver and adversary are synchronized, and whether they have a common guess for the sender's strategy.

In our work, we consider the number of successful rendezvous time slots in a long run, and we call it communication load. In previous works, all adversaries are aware of the rendezvous algorithms of the users, and the users adopt the same algorithm or use the common key after they have achieved rendezvous. However, in our work, the adversary only has the knowledge of n channels, and two users may play different strategies as soon as they rendezvous.

3. PRELIMINARIES

3.1 System Model

We divide the licensed spectrum into n channels that don't overlap with each other and denote them as $U = \{1, 2, \cdots, n\}$. Due to the appearance of the licensed users, the unlicensed users cannot access all of n channels, and we denote the channels not occupied by the licensed users as *available*. Therefore, both users can find out the set of available channels after taking a short sensing stage.

Suppose t_m is the minimum time that is sufficient for the users to establish a link and exchange messages when they access the same channel. We divide time into slots of length $2t_m$. Similar to some previous works [15], we assume two users in this paper take different roles: one is sender (denote as user S) who would send messages through a channel (or the constructed link) in each time slot, while the other is receiver (denote as user R), who would listen through a channel in each time slot, and send messages only upon having received the sender's messages. When the sender is sending messages through certain channel and the receiver is listening at the same one simultaneously, communication can be established and the process is referred as rendezvous [3,8,15].

We assume that two users are able to know their common available channels once they achieve first rendezvous on some channel. In previous works, two users would stay at the common channel and keep communication through it. However in our work, we assume an adversary (denote as A) could also join the network at any time after the users' first rendezvous. In each time slot, the adversary can choose one of all n channels to sense and block. If the sender is sending messages on that channel simultaneously, the adversary could learn that and block the messages from sending to the receiver. Under the attack of an adversary, the traditional strategy of staying at a certain channel for communication is vulnerable. Therefore in this paper, the task of two users is to keep communication and increase communication load (the number of time slots that the users can communicate successfully) in a long run, while the adversary aims to prevent users from communicating.

3.2 Problem Definition

Denote the available channels of the sender and the receiver as C_s, C_r respectively, and $C_g = C_s \bigcap C_r$ as common channels of them. Assuming the sender, receiver and adversary would access channels $f(C_s, t), h(C_r, t), g(t)$ in time slot t respectively, where t is the time slots elapsed when the adversary joins the network.

We say that two users communicate successfully or they can rendezvous in certain time slot t when they choose the same channel j but the adversary doesn't block the channel. We use $I(t)$ to indicate whether two users communicate successfully in time slot t, then $I(t) = 1$ when $f(C_s, t) = h(C_r, t) \neq g(t)$, otherwise $I(t) = 0$. We define the *communication load* as follows:

Communication load: Given the strategies of the users and the adversary $f(C_s, t), h(C_r, t), g(t)$, communication load $CL(f, h, g)$ in a long run T is defined as:

$$CL(f, h, g) = \sum_{t=1}^{T} I(t)$$

Moreover, we define *communication load ratio* $\lambda(f, h, g) = \frac{CL(f,h,g)}{T}$ as the fraction of time slots that the users communicate successfully in T time slots. We define the communication and block game from both the users' and the adversary's aspects.

Problem 1. ***(Communication problem for the users)****: Design strategy f, h for the sender and the receiver to maximize* $\min_{\forall g} \lambda(f, h, g)$.

Problem 2. ***(Block Problem for the adversary)****: Denote converge time CT as the time slots the adversary costs to acquire information of the users to block communication regularly in a long run. The problem is to design strategy g for the adversary to minimize* $\max_{\forall f,h} \lambda(f, h, g)$ *in a reasonable converge time CT.*

3.3 Multiple Channels Scenario

Due to the rapid development of wireless technology, the users as well as the adversary may have the ability to access more than one channel in each time slot. In our work, we assume both the sender and the receiver can access $m \geq 1$

Table 1: Notations

Notation	Definition
CL	communication load
CT	convergence time
WL	the worst communication load against any adversary
$C_s(C_r)$	available channel for the sender (the receiver)
C_g	The common channels set of the users
M	$M = \|C_g\|$
$\mathcal{S}_u(\mathcal{S}_a)$	the strategy sets for the users (the adversary)
$s_u(s_a)$	the strategies that the users (the adversary) adopted
$\mathcal{U}_u(\mathcal{U}_a)$	the utility of the users (the adversary)
$m(k)$	the number of channels the users (the adversary) can sense in each time slot
U_0	the strategy of the users that they stay on a certain channel to keep communication once they first achieve rendezvous
U_1	the strategy of the users that they change and keep communication on another channel when the users are blocked
U_2	the strategy of the users adopting Alg. 1
A_0	the strategy of the adversary that it continues to block the channel on which it firstly blocks the communication successfully
A_1	the strategy of the adversary adopting Alg. 2

channels in each time slot, and the adversary can sense and block $k \geq 1$ channels.

Supposing that $m \leq \min\{|C_r|, |C_s|\}$ and $k \leq n$, the sender and the receiver would choose a set of m available channels to attempt communication in each time slot, while the adversary can block and sense a set of k channels in all n channels. Denote the sets the sender, the receiver and the adversary choose as P_s, P_r, P_a respectively, then the users can communicate successfully if and only if $|P_r \bigcap P_s \setminus P_a| \geq 1$.

In this scenario, the task of the users is also to maximize the communication load against any adversary's strategy (similar to Problem 1), while the adversary aims to minimize the users' communication load for any strategies the users may adopt (similar to Problem 2).

3.4 Other Definitions and Facts

In this subsection, we introduce some definitions and facts that will be used in this paper.

Denote the strategy sets for the users and the adversary as $\mathcal{S}_u, \mathcal{S}_a$ respectively.We call certain strategy (s_u, s_a) a Nash equilibrium if and only if:

$$\mathcal{U}_u(s_u, s_a) \geq \mathcal{U}_u(s_u', s_a), \forall s_u' \subseteq \mathcal{S}_u$$
$$\mathcal{U}_a(s_u, s_a) \geq \mathcal{U}_a(s_u, s_a'), \forall s_a' \subseteq \mathcal{S}_a$$

where $s_u \in \mathcal{S}_u$, $s_a \in \mathcal{S}_a$, and $\mathcal{U}_u$,$\mathcal{U}_a$ is the utility of the users and the adversary respectively. In game theory, utility represents the satisfaction or the reward the player gains through the game [2]. In our work, if the sender, the receiver, and the adversary adopt strategy f, h, g respectively, we denote $\mathcal{U}_u = \lambda(f, h, g)$ and $\mathcal{U}_a = 1 - \lambda(f, h, g)$.

We also use some basic facts and inequalities, due to the page limit, we put them in the full version [16].

The notations used in this paper is listed in Table 1.

4. COMMUNICATION ALGORITHM FOR THE USERS

In previous works, two users would stay on a certain channel to keep communication once they first achieve rendezvous, and we denote the strategy for the users as U_0. However, it's vulnerable to the adversary. Suppose the adversary senses each of n channels in a round-robin pattern, once the adversary blocks the communication between two users successfully on certain channel, it keeps blocking the same channel in the following time slots, and we denote the strategy as A_0. Obviously, under the attack of the adversary, the users' strategy U_0 is inefficient since the adversary can find the channel after at most n time slots and the communication between the users would be blocked afterwards. In this section, we introduce an efficient algorithm for the sender and the receiver to increase communication load under the attack of any adversary's strategy.

4.1 Jumping Algorithm for The Users

Algorithm 1 Jumping Algorithm for the Users

```
if The user is sender then
    Generate random bits sequence $l$
end if
When achieve rendezvous the first time sender sends available channel sets $C_s$ and random bits $l$ to the receiver, upon receiving this, receiver sends $C_r$ back.
Both the sender and the receiver learn the tripe $(C_s, C_r, l)$. All the random choices made in the rest parts of this algorithm is based on $l$.
$C_g = C_s \bigcap C_r, M = |C_g|, S = C_g$
if $M = 1$ then
    Choose channel $i$ in $C_s \setminus S$ uniformly at random , $S = S \bigcup \{i\}$
end if
while The users keep communicating do
    Choose channel $j$ in $S$ uniformly at random and access that channel, if $j \notin C_r$, the receiver choose any channel to replace $j$ at that time slot.
end while
```

The idea of this algorithm is to randomly choose channels in the common channel set in each time slot. If there is only one common channel, the users choose another channel sacrificing some chance of communication to avoid being blocked easily by the adversary. In the algorithm, l is the sequence of random bits generated by the sender, then it is sent to the receiver along with set C_s when two users first achieve rendezvous. All random choices made by two users are then based on l, thus both users have the full knowledge of each other including the random choices. C_g is the set of common channels set between two users and denote M as the number of common channels. Both users choose channel j in set S in each time slot uniformly at random, where S equals to C_g when $M > 1$, or S contains the unique common channel as well as another random chosen channel in C_s.

4.2 Efficiency Against Adversary's Strategies

We show that our algorithm can achieve high communication load under *any* strategy adopted by the adversary who has the ability to sense and block one channel in each time slot.

Algorithm 2 Try-and-evaluate Algorithm for the Adversary

```
Initialize $E = (1, 1, \cdots 1)$
while The users keep communicating do
    Randomly choose a channel $j$. For each channel $i$, $Pr(j = i) = \frac{E_i}{\sum_{k=1}^{n} E_k}$, sense and block channel $j$
    if The adversary blocks the messages successfully then
        $\alpha = \min\{\frac{E_j}{\sum_{k \neq i, 1 \leq k \leq n} E_k}, 1\}$
        $l_1 = \max\{E_1, E_2, \cdots E_n\}, l_2 = \max\{E_j + 1, E_j * 2^{\alpha}\}, E_j = \max\{l_1, l_2\}$
    else
        $E_j = \frac{E_j}{2}$
    end if
end while
```

Theorem 1. *No matter what algorithm the adversary adopts, Alg. 1 can achieve communication load ratio $\lambda \geq \frac{1}{4}$ if $M = 1$; and $\lambda \geq 1 - \frac{1}{M}$ if $M \geq 2$.*

The proof of Theorem 1 can be found in the full version [16]. In addition, when $M > 1$, even if the adversary has the ability to **explore more information** (for example it knows C_r or C_s), our algorithm can still achieve $\lambda \geq 1 - \frac{1}{M}$ unless the adversary knows the sequence l of the random bits, since the algorithm works on the basis of l.

5. BLOCK ALGORITHM FOR THE ADVERSARY

If two users know the existence of the adversary, they may adopt different strategies to keep communication. Although the adversary's strategy A_0 (sense all channels and stay blocking the same channel when the first blocking happens) works quite well for some users' strategies (for example U_0 defined in Section 4), it is inefficient against most strategies of the users. For example, considering a trivial strategy of the users: the users' behavior is the same as U_0 until the communication is first blocked, then the users choose another common channel and keep communication (denote this strategy as U_1). It's obvious that the adversary's strategy A_0 can only block U_1 once, which turns out to be a bad result. In this section, we introduce an efficient algorithm for the adversary to find and block the common channels of users. The intuition of this algorithm is to restore a number for each channel which reflects the frequency the sender using that channel. Then the adversary blocks channels according to the evaluation of their frequencies, thus the adversary would block a channel that is accessed more often by the sender. We call this algorithm "try-and-evaluate" algorithm.

5.1 Try-and-Evaluate Algorithm for Adversary

In this algorithm, E is a vector of length n. For each $1 \leq j \leq n$, E_j is used to reflect the frequency that the sender sends message through the channel. This value in E would take a "half-reduce,slow-increase,fast-recovery" pattern. In each time slot, the adversary chooses to block a channel j, if the sender is not using channel j, E_j would be reduced by half; if the adversary successfully blocks a message through it, we divide it into 3 cases according to the value of E_j: in most cases E_j is slowly increased by 1; but E_j increases faster when it gets larger, and this method helps the adversary converge faster; and if E_j is smaller than some other E_i ($i \neq j$), we would recover it back to E_i and this is what we called "fast recovery" to compensate some underestimated channels in the process of "half-reduce".

5.2 Efficiency Against Users' Strategies

We show the efficiency of Alg. 2 from two aspects: 1) The converge time of Alg. 2 is ($O(n \log n)$) against the users' strategy U_0; 2) Alg. 2 can achieve the smallest communication load of the users in Theorem 1 against the users' strategy (Alg. 1).

Theorem 2. *The adversary running Alg. 2 would block the common channel with high probability after $t = O(n \log n)$ time slots.*

Theorem 3. *Alg. 2 achieves the best utility for the adversary in Theorem 1 if the users run Alg. 1.*

We omit the proof of Theorems 2 and 3 which are can be found in the full version [16]. Theorem 2 and Theorem 3 show the efficiency of Alg. 2 against both trivial users' strategy and our proposed strategy (Alg. 1). It is obvious that it also works well for the users' strategy U_1. Actually, it is an efficient algorithm for the adversary against almost all users' strategies, since it modifies the probability of blocking each channel dynamically according to the behavior of the users.

6. OPTIMALITY OF OUR ALGORITHMS

In this section, we do some further analysis for the algorithms for both the users and the adversary (Alg. 1-2), showing that our algorithms are optimal to some extant. For simplicity, we denote the set for all possible users' strategies of pattern $(p_1, p_2, \cdots, p_n)$ as $\mathcal{S}_u$, where the users access channel j with probability p_j in each time slot independently. Denote $WL(f, h)$ as the worst communication load in a long run T against any adversary's strategy if the sender and the receiver run strategy f, h respectively.

Theorem 4. *Our users' algorithm (Alg. 1) can achieve best WL.*

We omit the proof of Theorem 4 which can be found in the full version [16].

Similar to the notation of $\mathcal{S}_u$, we define the strategy set for the adversary as: $\mathcal{S}_a = \{s_a | s_a = (q_1, q_2 \cdots q_n)\}$, where q_j is the probability that the adversary blocks channel j in each time slot (note that, the choices are also independent for different time slots).

Theorem 5. *If $M \geq 2$, the users adopting strategy $s_u = \{p_1, p_2, \cdots, p_n\}$ and the adversary adopting strategy $s_a = \{q_1, q_2, \cdots, q_n\}$ would be a Nash Equilibrium if for any $j (1 \leq j \leq n)$, $p_j = q_j = \frac{1}{M}$ when $j \in C_g$ and $p_j = q_j = 0$ when $j \notin C_g$.*

The proof of Theorem 5 can be found in the full version [16].

Notice that, when $M > 2$, our algorithms for the users and the adversary actually do the same as the strategy in Theorem 5. Thus by adopting our algorithms, neither of the users and the adversary could gain more utility by changing their strategies. So the users can achieve the best utility as well as the adversary does simultaneously.

7. ALGORITHMS FOR MULTIPLE CHANNELS SCENARIO

Due to the rapid development of wireless technology, both the users and the adversary could sense or block more than one channel in each time slot. Considering the multi-channel setting, we propose algorithms for the users as well as the adversary in this section, where the users can sense m channels in each time slot and the adversary can block k channels simultaneously. Moreover we briefly analyze the performance of the algorithms.

7.1 Algorithm for the Users

Algorithm 3 Algorithm for the Users in Multiple Channels Scenario

```
Run the first 7 lines of Alg. 1, the notation of $C_g, M, l$ is the same as Alg. 1
$t = |C_s|$
while The users keep communicating do
    $P = 4n^2 \log n, p = q = 0$
    $\delta = 0.4 + 0.5 * \frac{M}{M+30}$
    Choose $t-M$ channels in $C_s \backslash C_g$ uniformly at random, denote them as $c_{M+1} \cdots c_t$ , and $S = S \bigcup \{c_{M+1} \cdots c_t\}$
    Generate channel sets sequence $seq$ with $P * M$ channels such that each $seq_{ij} (1 \le i \le P, 1 \le j \le m)$ is chosen from $S$ uniformly at random independently
    for $i = 1$ to $P$ do
        In time slot $i$ of this round, Access channels $seq_{i1}$ to $seq_{im}$. For the receiver, for any $seq_{ij} \notin C_r (1 \le j \le M)$, choose any available channel $c \in C_r$ to replace it.
        for $j = 1$ to $m$ do
            if $seq_{ij} \in C_g$ then
                $p = p + 1$
                if Messages through channel $seq_{ij}$ is not blocked then
                    $q = q + 1$
                end if
            end if
        end for
    end for
    $X = q/p$
    if $X > \delta$ and $t > \max\{m, M\}$ then
        $t = t - 1$
    end if
end while
```

The algorithm for the users in multi-channel scenario is similar to that of single channel scenario, and we describe it in Alg. 3. In each time slot, the users choose m channels uniformly at random from t channels, where t is the number of available channels. We take $P = 4n^2 \log n$ time slots as a round, and we use p, q to calculate how frequently the communication through channels in C_g is not blocked in each round. If p/q is larger than some constant δ, we reduce t by 1 to achieve a better communication load.

7.2 Algorithm for the Adversary

We present our algorithm for the adversary in the multi-channel scenario in Alg. 4, where in each time slot the adversary uses one of k channels (Denote as c_1) to calculate the frequency that the sender accesses each channel. We also divide time slots into rounds where each round contains $P = 4n^2 \log n$ time slots. P_{pr}, P_n are vectors of length n to evaluate the probability that the sender uses each channel in the previous round and in this round respectively. D, F are length n vectors to calculate P_n. If P_n is close to P_{pr}, we regard that the users are in a stable state, and thus we block each channel j with probability $P_n(j)$ in the next round, otherwise we choose each of $k-1$ channels independently according to Alg. 2. Compared with the simple strategy that the adversary chooses k channels to block by adopting Alg. 2 separately, our proposed algorithm (Alg. 4) for the multi-channel scenario can reduce the fluctuation value in a long run.

Algorithm 4 Algorithm for the adversary in Multiple Channels Scenario

```
$P_{pr} = P_n = (0, 0, \cdots, 0)$
Construct vector $E$ the same as Alg. 2.
$\varepsilon = \frac{1}{100}, P = 4n^2 \log n$
while The users keep communicating do
    $D = F = (0, 0, \cdots, 0), bo = false$
    for $i = 1$ to $P$ do
        Choose $c_1$ uniformly from $n$ channels
        if $bo$ then
            choose $c_j$ ( $2 \le j \le k$ ) independently , and $Pr(c_j = r) = \frac{P_{pr}(r)}{\sum_{1 \le q \le n} P_{pr}(q)}$
        else
            choose $c_j$ ( $2 \le j \le k$ ) independently according to $E$ as Alg. 2.
        end if
        Sense and block channels $c_1, c_2 \cdots, c_k$.
        Update $E_{c_1}$ to $E_{c_k}$ according to Alg. 2.
        $D_{c_1} = D_{c_1} + 1$
        if The sender is using channel $c_1$ then
            $F_{c_1} = F_{c_1} + 1$
        end if
        $P_n(i) = F_i / D_i$, $1 \le i \le n$
        $bo = true$ if and only if for all $1 \le i \le n$, $|P_n(i) - P_{pr}(i)| \le \varepsilon$
        $P_{pr} = P_n$
    end for
end while
```

More analyses for Alg. 3 and Alg. 4 are presented in the full version [16].

8. SIMULATION RESULTS

In this section, we conduct extensive simulations to evaluate the performance of our algorithms for the users and the adversary under both single channel scenario and multi-channels scenario. We conduct the simulations for 100 separate times and take the average value as the simulation result. Denote Alg. 1 for the users as U_2, and Alg. 2 for the adversary as A_1.

8.1 Single Channel Scenario

To verify the efficiency of our algorithms for the users and the adversary, we first compare the performance of U_1 and U_0 against the adversary's strategy A_0. Moreover, the performance of A_1 is also compared with A_0 against U_0 and U_1. We depict the curves describing the relationship between time T and communication load ratio λ. In addition,

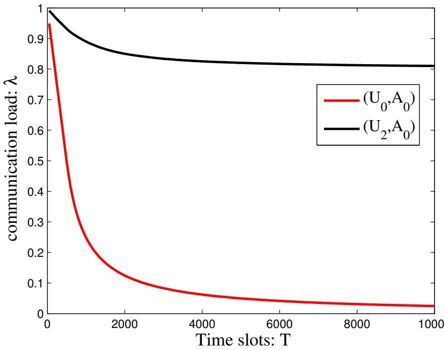
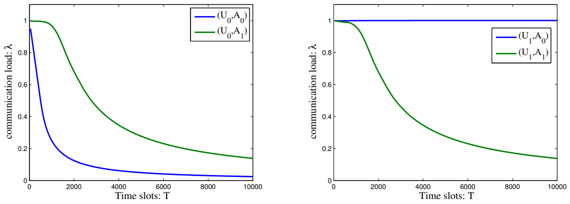


Figure 1: λ vs. T against A_0



(a) λ vs. T against U_0 (b) λ vs. T against U_1

Figure 2: λ for A_0 and A_1 against U_0 and U_1 when T ranging from 50 **to** 10000

we list the converge time CT of A_1 against U_0 for different n values as well as λ of A_1 against U_2 when $n = 1000$ and M varies.

In Fig. 1, λ of both U_0 and U_2 against A_0 are shown in the figure when $n = 1000$, $M = 5$ and T ranges from 0 to 10000. As depicted, U_0 almost cannot communicate after about 10000 time slots, this is because: once the adversary adopting A_0 finds the channel that the users communicate on, they can never communicate again. In contrast, U_2 achieves a better result and it can achieve the communication load of about $0.8T$ in a long run of T time slots, which also verifies the efficiency of Alg. 1.

Fig. 2 shows the performance of A_0 and A_1 against U_0 and U_1. We choose $n = 1000$, $M = 5$ and T ranges from 0 to 10000. From the curves, we know that A_0 blocks almost all communication between the users if they adopt strategy U_0 after a short converge time. However, when A_0 encounters U_1, the adversary can block few proportion of the communication, this is because A_0 always blocks the channel on which it blocks communication successfully for the first time, but the users could then turn to another channel for further communication after the channel is blocked. However A_1 works equally well for the users, though the converge time is larger against U_0 when compared with A_0, but the time is still acceptable.

Table 2 lists the converge time CT of A_1 against U_0 when n varies from 100 to 10000. We set $M = 5$ and define the converge time as the time slot cost until $E_j \geq \sum_{i \neq j} E_i$. It is shown that $CT/n\log n$ is almost monotone decreasing except for $n = 500$; and when $n \geq 500$, $CT/n\log n \leq 1$. The results corroborate our theoretical analysis that Alg. 2 would converge in $O(n\log n)$ times slots, and the shortest possible converge time would be $O(n)$ since it takes $O(n)$ time slots to find the channel that the users adopt at first.

Table 2: The Convergence Time of Alg. 2

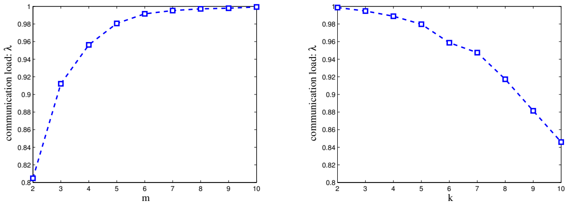
n	CT	$CT/n\log n$
100	214	1.07
500	1209	0.8959
1000	2863.1	0.954
2000	5568.37	0.843
5000	14964.8	0.809
10000	31473.6	0.788

Remarks: 1) The number of common channels $M = 5$; 2) CT is the converge time define as follows: denote j as the channel the users stay on, CT is the expected time cost until $E_j \geq \sum_{i \neq j} E_i$

Table 3: The Convergence Time of Alg. 2

M	λ	ε
1	0.2711	0.0211
2	0.50905	0.00905
3	0.67376	0.0071
4	0.755378	0.005378
5	0.804	0.004
8	0.879426	0.004426
10	0.902696	0.002696
100	0.990374	0.000374

Remarks: 1) The number of channels $n = 1000$; 2) λ is the proportion of successful communication time slots during enough long time $T = 500000$; 3) ε is the deviation of simulation results from our theoretical analysis



(a) λ vs. m when $k = 5$ (b) λ vs. k when $m = 5$

Figure 3: λ vs. m and k when $N = 1000, M = 10$

The communication load ratio λ of U_2 against A_1 is listed in Table 3, where the number of common channels M ranges from 1 to 100 and we set $n = 1000$. ε is the difference between the simulation results and our theoretical analysis. From Table 3, ε is less than 0.01 when $M \geq 2$, and it becomes smaller when M is larger, thus the results also corroborate our analysis in Section 5.

8.2 Multiple Channels Scenario

For the multiple channels scenario, we set $n = 1000$, $M = 10$, and try to find out how the communication load ratio λ changes when m and k vary respectively. Here m, k are the number of channels that the users and the adversary can sense in each time slot respectively.

Fig. 3(a) shows how λ changes when $k = 5$ and m ranges from 2 to 10, and Fig. 3(b) shows the change of λ when $m = 5$ and k varies from 2 to 10. As shown in the figures,

λ is monotone increasing when m increases and it is monotone decreasing when k increase. Compared with the result in single channel scenario, when $M = 10$, $\lambda = 0.9$; while as depicted in Fig. 3, when $m = k = 5$, $\lambda \approx 0.98$. Thus, the users can communicate successfully in more time slots for the multi-channel scenario. This is because the adversary has to block all m channels that the sender is using for communication. Moreover, λ is more sensitive if $m < k$, because the users can communicate successfully nearly all the time when $m \geq k$. Moreover, when $k - m$ gets larger, λ becomes more sensitive. This means that if the adversary can sense more channels in each time slot than that of the users, acquiring the ability to access more channels would be useful for the users (or the adversary) to increase (or decrease) the communication load.

Combining the simulation results for both single channel scenario and multi-channel scenario, our proposed algorithms for the users and the adversary can achieve good performance that corroborates our analyses.

9. CONCLUSION

In this paper, we introduce the Communication and Block Game between two users and an adversary in Cognitive Radio Networks (CRNs). In this game, the users aim to maximize their communication load in a long run of T time slots, while the adversary aims to minimize it. We design efficient algorithms for the users and the adversary. The algorithm for the users guarantee two users can achieve communication load no less than $\frac{1}{4}T$ when there is only one common available channel, and can achieve communication load no less than $(1 - \frac{1}{M})T$ when there are M $(M > 1)$ common available channels against *any* possible adversary. In addition, the algorithm for the adversary works best against the proposed algorithm for the users such that the users' communication load matches their lower bound. We further show that the introduced algorithms for the users and the adversary would become a Nash Equilibrium when $M \geq 2$, which means they can achieve their best utilities simultaneously. We also present algorithms for the users and the adversary in the multiple channels scenario (both equipped with multiple radios), which achieves better communication load for the users and a nontrivial utility for the adversary by blocking a considerable number of users' communications.

Since the proposed Nash Equilibrium considers only a subset of all possible strategies, in the future, we aim to generalize our results into all strategies. We will also try to propose more efficient algorithms and present more refined theoretical analyses in the multiple channels scenario.

10. ACKNOWLEDGMENTS

This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61103186, 61033001, 61361136003.

11. REFERENCES

[1] 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.

[2] R.J. Aumann(2008). "Game Theory" Introduction, The New Palgrave Dictionary of Economics, 2nd Edition.

[3] K. Bian, J.-M. Park. Maximizing Rendezvous Diversity in Rendezvous Protocols for Decentralized Cognitive Radio Networks. *IEEE Trans. on Mobile Computing*, 12(7):1294-1307, 2013.

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

[7] Y. Gao, Z. Gu, Q.-S. Hua and H. Jin. Multi-Radio Channel Detecting Jamming Attack Against Enhanced Jump-Stay Based Rendezvous in Cognitive Radio Networks. In *COCOON*, 2015.

[8] 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.

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

[10] Z. Gu, Q.-S. Hua, and W. Dai. Fully Distributed Algorithms for Blind Rendezvous in Cognitive Radio Networks. In *MOBIHOC*, 2014.

[11] E. Lee, S. Oh and M. Gerla. Frequency Quorum Rendezvousfor Fast and Resilient Key Establishment under Jamming Attack. In *Mobicom Poster*, 2010.

[12] 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.

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

[14] Y. Oh and D. Thuente. Limitations of Quorum-based Rendezvous and key establishment schemes against sophisticated jamming attacks. In *MILCOM*, 2012.

[15] Y. Oh and D. Thuente. Channel Detecting Jamming Attacks Against Jump-Stay Based Channel Hopping Rendezvous Algorithm for Cognitive Radio Networks. In *ICCCN*, 2013.

[16] H. Pu, Z. Gu, Q.-S. Hua, H. Jin. Communication and Block Game in Cognitive Radio Networks. http://grid.hust.edu.cn/qshua/mswim15full.pdf

[17] M. Rahamn and M. Krunz. Game-theoretic Quorum-based Frequency Hopping for Anti-jamming Rendezvous in DSA Networks. In *DySPAN*, 2014.

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

[19] M. Strasser, C. Popper and S. Capkun. Efficient uncoordianted fhss anti-jamming communication. In *MOBIHOC*, 2009.

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