This paper was presented as part of the main technical program at IEEE INFOCOM 2011

ETCH: Efficient Channel Hopping for Communication Rendezvous in Dynamic Spectrum Access Networks

Yifan Zhang, Qun Li, Gexin Yu
The College of William and Mary, VA, USA
{yzhang,liqun}@cs.wm.edu, gyu@wm.edu

Baosheng Wang*
National University of Defense Technology, China
bswang@nudt.edu.cn

***Abstract*—In a dynamic spectrum access (DSA) network, communication rendezvous is the first step for two secondary users to be able to communicate with each other. In this step, the pair of secondary users meet on the same channel, over which they negotiate on the communication parameters, to establish the communication link. This paper presents ETCH, Efficient Channel Hopping based MAC-layer protocols for communication rendezvous in DSA networks. We propose two protocols, SYNC-ETCH and ASYNC-ETCH. Both protocols achieve better time-to-rendezvous and throughput compared to previous work.**

I. Introduction

Dynamic spectrum access (DSA) is a promising technique that solves the spectrum scarcity problem and increases network capacity. In DSA networks, unlicensed users (i.e., secondary users) are granted the right of accessing licensed spectrum while the licensed users (i.e., primary users) are not using them. In other words, DSA opens the door towards much larger spectrums for secondary users, but the secondary users must stop using these spectrums when they sense that the spectrum's primary users are present.

As in normal multi-channel communication networks, communication rendezvous is the first step for a pair of DSA network nodes (i.e., secondary users[1]) to be able to communicate with each other. Specifically, a pair of DSA network nodes wishing to communicate should first agree on certain control information, which data channel to use in particular, before they are able to exchange the communication data. The spectrum over which the nodes negotiate to reach the agreement is called a *control channel*. Communication rendezvous for the pair of nodes is to establish a control channel between them. The common control channel approach, where a well-known channel is designated as control channel for all nodes, suffers from the problem of control channel congestion and is vulnerable to jamming attacks [1]. Moreover, this approach cannot be applied in DSA networks because of the dynamics of the channel availability. The channel hopping approach, in contrast, increases control channel capacity and is immune to jamming attacks by utilizing multiple control channels. In this approach, all idle network nodes hop on a set of sequences of *rendezvous channels* (i.e., channels that are assigned for the purpose of control information exchange). When two nodes that want to communicate with each other hop to the same rendezvous channel, this channel will serve as a control channel between the pair of nodes. The time that it takes for a pair of nodes to establish the control channel is called "*time-to-rendezvous*" or *TTR* for short.

To establish control channels in a multi-channel network through channel hopping (abbreviation CH), each CH sequence should be able to rendezvous with all other CH sequences periodically. Apart from this, due to the unique property of DSA networks that the channel availability is dynamic, any pair of nodes should be able to utilize all rendezvous channels as their control channel. Otherwise, a pair of nodes would not be able to communicate if a primary user occupies the channels in which they rendezvous, even though there may still exist some other available channels to exchange the control information. This new requirement excludes the possibility of using some existing multi-channel communication protocols, such as CHMA [2] and SSCH [3], to establish control channels in DSA networks.

QCH [4] is a recently proposed control channel establishment protocol specifically designed for DSA networks. It utilizes the overlap property of quorums in a quorum system to develop CH sequences such that any two CH sequences are able to rendezvous periodically. Meanwhile, to accommodate the dynamics of the channel availability in DSA networks, QCH divides a period of CH sequence into several frames, where the number of the frames equals to the number of rendezvous channels. Theoretical analyses and simulation results show that QCH outperforms other CH-based multi-channel communication protocol in terms of TTR and traffic throughput. Nevertheless, there is plenty of space to improve on QCH given the following two observations.

The first and the most notable one is that in synchronous QCH all nodes always compete for just one rendezvous channel as control channel, which would lead to a high probability of traffic collision and low traffic throughput. We propose SYNC-ETCH, a synchronous ETCH protocol, which efficiently utilizes the frequency diversity in establishing control channels for DSA network nodes.

*This work was done while the author was visiting the College of William and Mary.

[1]In this paper, DSA network nodes always refer to secondary users of the DSA network.

Furthermore, the asynchronous QCH only guarantees two of the rendezvous channels to be used as control channels. The secondary users would not be able to communicate if the two control channels are not available, or undergo a high level of traffic collision. We propose ASYNC-ETCH, an asynchronous ETCH protocol, which solves the problems by using all rendezvous channels as control channels.

The main contributions of this paper are summarized as follows:

- We formulate the problem of designing channel hopping based communication rendezvous protocols by considering all relevant metrics and requirements. We provide an in-depth and systematical analysis about the principles in designing this type of protocols. This is valuable for future research in this field.
- We propose an optimal synchronous protocol for communication rendezvous in DSA networks. The optimality of this protocol lies in that its average time-to-rendezvous is shortest under the premise that all the rendezvous channels should be utilized in every hopping slot. This approach achieves good time-to-rendezvous while greatly increasing the capacity of the DSA network at the communication setup stage.
- We propose a novel asynchronous protocol that enables two DSA network nodes to rendezvous without the existence of global clock synchronization mechanisms. Our protocol achieves better time-to-rendezvous and traffic throughput than the existing schemes.

The rest of this paper is organized as follows. We summarize the related work in section II. In section III, we describe the problem formulation. The SYNC-ETCH protocol and the ASYNC-ETCH protocol are detailed in section IV and section V respectively. We compare ETCH with existing solutions in section VI and evaluate the performance of ETCH in section VII. Finally, we conclude this paper in section VIII.

II. Related Work

DSA [5] network research is mainly focused on spectrum sensing ([6], [7], [8], [9], [10], [11]), spectrum management ([11], [12]), spectrum mobility and spectrum sharing.

Spectrum sharing techniques in DSA networks can be categorized into two classes based on the network architecture. Techniques in the first class assume there is a centralized entity that is responsible for the spectrum allocation for all the secondary users in the network. DSAP [13] is a typical solution that belongs to this category. The second class of spectrum sharing techniques perform the sharing in a distributed manner. These techniques can be further divided into two groups based on the assumption about the existence of a common control channel. Techniques the first group [14] assumes there is common control that is available to all secondary users, while the second group of techniques do not use this assumption. Our work together with QCH [4] and SeqR [15] fall in the second group. We will discuss QCH and SeqR at length later. HD-MAC [1] is another spectrum sharing scheme that performs spectrum sharing without assuming the existence of a common control channel. Different from ETCH, HD-MAC does not use the channel hopping technique to establish a control channel between a pair of secondary users. In this scheme, secondary users self-organize into groups based on similarity of available channels. In each of the groups, a group control channel, elected by group members, is used to carry control information of the group nodes. A weakness of HD-MAC is that it relies on all-channel broadcast to spread spectrum availability information and control channel votes. Both sender and receiver of a broadcast message need to rotate on all their available channels to send or receive the message, which will take a long time in establishing the group control channel especially when the number of channels is high.

III. Problem Formulation

A. Problem Setting

In a DSA network, there are N licensed channels labeled as $C_0, C_1, ..., C_{N-1}$ that can be used for control information exchange. In other words, there are N rendezvous channels in the DSA network. Any pair of nodes wishing to communicate with each other should first establish a control channel between them before they are able to communicate. We assume that there is no centralized entity that globally controls the allocation of communication channels, so the control channel establishment between a pair of nodes should be done in a distributed manner.

In a CH-based solution, idle nodes[2] **periodically** hop on (i.e., switch their working channel according to) a *CH sequence*, which is a sequence of rendezvous channels. The time that a node stays on a channel is defined as a *hopping slot* (or *slot* for short). Similar to representations used in [4], we represent a CH sequence S as a sequence of hopping slots which are defined as $(i, S[i])$ pairs:

$$S = \{(0, S[0]), (1, S[1]), ..., (i, S[i]), ..., (p-1, S[p-1])\},$$

where i ($0 \leq i \leq p-1$) is the index of a hopping slot, and $S[i] \in \{C_0, \cdots, C_{N-1}\}$ ($0 \leq i \leq p-1$) is the label of the rendezvous channel assigned to slot-i of the sequence S. The time it takes for a node to hop on a CH sequence once is called a *hopping period*. Suppose the length of a hopping slot is t, the length of a hopping period is $t \times p$, where p is the number of hopping slots in the CH sequence. When two nodes that are about to communicate hop to the same channel, the channel is established as the control channel between them. If more than two nodes meet on the same rendezvous channel in the same slot, they use existing collision avoidance mechanisms (e.g. RTS/CTS) or retransmission to establish pairwise control channels between them.

The CH-based solution should take account of the following requirements in its design.

- **Overlap requirement**. This requirement mandates that any two of the CH sequences must overlap in order to ensure any pair of nodes are able to communicate with

[2]Here idle nodes refer to nodes waiting to initiate a communication with other nodes and nodes waiting others to connect to them.

each other. Given two CH sequences S_0 and S_1, they *overlap* if there exists a slot $(i, S_0[i]) \in S_0$ and a slot $(i, S_1[i]) \in S_1$ that $S_0[i] = S_1[i]$. The i-th slot is called an *overlapping slot* between S_0 and S_1, and the rendezvous channel $S_0[i] \in \{C_0, \cdots, C_{N-1}\}$ is called an *overlapping channel* of S_0 and S_1. If a rendezvous channel serves as an overlapping channel between a pair of CH sequences in slot-i, we say that *the rendezvous channel is utilized (as a control channel) in slot-i.*

- **Full utilization of rendezvous channels**. This requirement requires that any pair of nodes be able to utilize all the rendezvous channels as control channels. Otherwise, a pair of nodes would not be able to exchange control information if the primary users of the overlapping channels of the CH sequences they are following appear, even though there still exist available rendezvous channels.
- **Even use of rendezvous channels**. This requirement requires that all the rendezvous channels should have approximately the same probability to appear in each CH sequences. If a CH sequence heavily relies on a certain channel (i.e., the channel is assigned to most of the slots of the CH sequence), nodes that hop on this CH sequence will lose contact with most of other nodes when the heavily relied channel is occupied by the primary user.

The overlap requirement is the fundamental requirement that must be satisfied in order for a CH based solution to be applicable to establishing control channels in DSA networks.

B. Metrics

The following three metrics are used to theoretically evaluate the performance a communication rendezvous protocol.

- **Average rendezvous channel load.** This metric measures the average fraction of nodes that meet in the same rendezvous channel among all the nodes. Given a DSA network with M nodes that is using a communication rendezvous protocol with average rendezvous channel load α $(0 < \alpha \leq 1)$, there are on average $M\alpha$ nodes rendezvous in the same channel. A small average rendezvous channel load helps to alleviate traffic collisions and increase communication bandwidth.
- **Average time-to-rendezvous.** This is the number of hopping slots that two nodes need to wait on average before they can rendezvous. A small average time-to-rendezvous (TTR) makes nodes rendezvous and establish a communication link quickly.
- **Rendezvous channel utilization ratio.** This is the ratio between the number of rendezvous channels that can be utilized as control channels in a hopping slot and the total number of rendezvous channels. It measures, in a given hopping slot, the extent that a communication rendezvous protocol utilizes the frequency diversity in establishing control channels. A larger rendezvous channel utilization ratio contributes to increasing the network capacity at the communication setup stage. This metric does not apply to the asynchronous protocols in which hopping slot boundaries are not necessarily aligned.

Apart from the previous three theoretical metrics, two practical metrics, traffic throughput and actual time-to-rendezvous, are used to measure the actual performance of a communication rendezvous protocol. We will show that ETCH outperforms existing solutions through mathematical analysis and simulation in section VI and section VII respectively.

C. Assumptions

We have the following assumptions regarding DSA networks and a node's hardware.

- All rendezvous channels in a DSA network are known to all the nodes in the network. Information about rendezvous channels of a DSA network can be announced by regulation authorities such that all secondary users wishing to join the network will have this information.
- Each node is equipped with a single transceiver, which means at a certain time point a node can only engage in a channel for communication. This assumption is in accordance with the ability of most commodity wireless devices.
- We assume each node is able to switch its working channels with negligible overhead. This assumption is valid because most wireless hardware manufacturers claim that the channel switching delay is of the order of 80-90μs [16]. This delay is negligible compared to the length of a slot in a hopping sequence which is in the magnitude of 10ms.

IV. SYNC-ETCH

SYNC-ETCH is developed under the assumption of global clock synchronization. Specifically, SYNC-ETCH assumes that there exist some mechanisms to synchronize network nodes in a way that they can periodically start a new execution of their CH sequences at the same global time. The protocol consists of three parts: rendezvous scheduling, rendezvous channel assignment, and CH sequence execution.

A newly joined node executes SYNC-ETCH to establish a control channel with another node as follows. First, the node needs to construct a set of CH sequences. To efficiently exploit the frequency diversity in establishing control channels, SYNC-ETCH generates the CH sequences fully utilizing all the rendezvous channels in each of the hopping slots while guaranteeing the satisfaction of the overlap requirement.

Theorem 1: In a DSA network with N rendezvous channels, for any CH based synchronous communication rendezvous protocol where all the rendezvous channels are utilized in each of the hopping slots, the average TTR is no less than $\frac{2N-1}{2}$.

Proof: Since each hopping slot of a CH sequence is assigned to one rendezvous channel, it is obvious that to let all the N rendezvous channels be utilized (i.e. all the N rendezvous channels are used as overlapping channels of at least two CH sequences) in every hopping slot, there must be at least $2N$ CH sequences. In order to make each of the $2N$ CH sequences overlap with all other $2N-1$ sequences, there must be at least $2N-1$ hopping slots in each CH sequence. Hence the TTR is no less than $\frac{2N-1}{2}$ on average. ■

To achieve the optimal average TTR, SYNC-ETCH constructs a set of $2N$ CH sequences, each of which has $2N-1$ hopping slots. Each of these $2N$ CH sequences overlaps with all other $2N-1$ CH sequences in its $2N-1$ slots such that different overlappings happen in different slots.

SYNC-ETCH constructs these $2N$ CH sequences in two steps. In the first step, SYNC-ETCH generates $2N-1$ rendezvous schedules among a set of $2N$ **empty** CH sequences such that each CH sequence is paired with a different CH sequence in each of the rendezvous schedules. In the second step, the $2N-1$ rendezvous schedules, each of which corresponds to a hopping slot, are used as the basis of assigning rendezvous channels to the empty CH sequences. These two steps are referred as SYNC-ETCH's rendezvous scheduling mechanism and rendezvous channel assignment mechanism respectively.

At the completion of the CH sequences construction, the node synchronizes to the existing nodes, and starts the channel hopping process according to the CH sequence execution scheme of SYNC-ETCH. Once two nodes wishing to communicate hop to the same channel (i.e. they rendezvous), they exchange control information over the channel, and this channel is said to be established as control channel between them for this communication.

A. Rendezvous Scheduling

Recall that the goal of rendezvous scheduling is to construct $2N-1$ rendezvous schedules among a set of $2N$ empty CH sequences such that each CH sequence is paired with a different CH sequence in each of the rendezvous schedules. We now formalize the problem of ETCH's rendezvous scheduling as follows: given a set of $2N$ **empty** CH sequences, $U = \{S_0, S_1 \cdots, S_{2N-1}\}$, $D_p = \{d_0, d_1, \cdots, d_{N-1}\}$ is called a *rendezvous schedule* of U if $\bigcup D_p = d_0 \cup d_1 \cup \cdots \cup d_{N-1} = U$, where $d_i = \{S_s, S_t\}$ $(0 \le i \le N-1)$ is a set of two CH sequences that are scheduled to rendezvous in the slot p. Two rendezvous schedules of U, D_p and D_q, are *different* if $\forall d \in D_p$ and $\forall d' \in D_q$, $d \ne d'$. The rendezvous scheduling problem needs an algorithm to generate $2N-1$ different rendezvous schedules of U, each of which corresponds to a hopping slot. SYNC-ETCH uses Algorithm 1 to construct these $2N-1$ different rendezvous schedules.

In Algorithm 1, rendezvous schedule D_{sl} $(0 \le sl \le 2N-2)$ of slot-sl is constructed as follows. Within the CH sequence set $T = \{S_0, \cdots, S_{2N-2}\}$, S_a and S_b are scheduled to rendezvous in slot-sl (i.e. $\{S_a, S_b\} \in D_{sl}$) if $a + b \equiv sl(\mathrm{mod}(2N-1))$ and $a \ne b$. For CH sequence $S_a \in T$ that satisfies $2a \equiv sl(\mathrm{mod}(2N-1))$, it is scheduled to rendezvous with CH sequence S_{2N-1} in slot-sl (i.e. $\{S_a, S_{2N-1}\} \in D_{sl}$).

Fig. 1 shows an example of rendezvous scheduling in a DSA network with 3 rendezvous channels. To fully utilize all the 3 rendezvous channels, 6 CH sequences, each of which has 5 slots, will be constructed. Part (a) of Fig. 1 shows D_0 to D_4, the 5 rendezvous schedules returned by the Algorithm 1, each of which corresponds to a hopping slot. In each rendezvous schedule, the special CH sequence pair (i.e. the pair to which CH sequence S_{2N-1} belongs) is marked by blue. To better illustrate how the 6 CH sequences rendezvous, part (b) of Fig. 1 puts the rendezvous schedules in the format of CH sequences. In this part, given a hopping slot, CH sequences with the same type of fill rendezvous in that slot.

Algorithm 1 Rendezvous Scheduling

Input: A set of $2N$ empty CH sequences, $U = \{S_0, \cdots, S_{2N-1}\}$, each of which has $2N-1$ slots
Output: $2N-1$ different rendezvous schedules of U: $D_0, D_1, \cdots, D_{2N-2}$

1 Initialize $D_0, D_1, \cdots, D_{2N-2}$ to be empty;
2 **for** $sl \leftarrow 0$ to $2N-2$
3 $\quad T \leftarrow U \backslash \{S_{2N-1}\}$;
4 $\quad$ **for** $i \leftarrow 0$ to $N-1$
5 $\quad\quad a \leftarrow$ the smallest subscript in T;
6 $\quad\quad$ **if** $a \le sl$
7 $\quad\quad\quad b \leftarrow sl - a$;
8 $\quad\quad$ **else**
9 $\quad\quad\quad b \leftarrow 2N - 1 + sl - a$;
10 $\quad\quad$ **if** $a = b$
11 $\quad\quad\quad b \leftarrow 2N-1$;
12 $\quad\quad d_i = \{S_a, S_b\}$;
13 $\quad\quad D_{sl} = D_{sl} \cup \{d_i\}$;
14 $\quad\quad T = T \backslash \{S_a, S_b\}$;
15 **return** $D_0, D_1, \cdots, D_{2N-2}$;

We now prove the correctness of Algorithm 1 as follows.

Theorem 2: Algorithm 1 constructs $2N-1$ rendezvous schedules of U, and all these $2N-1$ rendezvous schedules are different.

Proof: In order to prove Algorithm 1 constructs $2N-1$ rendezvous schedules, we need to prove given an integer sl $(0 \le sl \le 2N-2)$, D_{sl} is a rendezvous schedule of U. To prove this, we need to prove

(1) there is only a number $x \in [0, 2N-2]$ such that $2x \equiv sl \pmod{(2N-1)}$, and

(2) $\forall\ a, b, c, d \in [0, 2N-2]$ that satisfy $a + b \equiv sl \pmod{(2N-1)}$ and $c + d \equiv sl \pmod{(2N-1)}$, if $a \ne c$ then $b \ne d$.

By proving (1) we can guarantee that the CH sequence S_{2N-1} only exists in only a CH sequence pair d_i $(0 \le i \le N-1)$ within rendezvous schedule D_{sl}. From (1), (2) and the strategy that we always choose the first CH sequence of d_i $(0 \le i \le N-1)$ from a set of CH sequences that have never been chosen (i.e. set T in Algorithm 1)(line 5), we can ensure that $\bigcup D_{sl} = d_0 \cup d_1 \cup \cdots \cup d_{N-1} = U$ (i.e. D_{sl} is a rendezvous schedule of U).

We prove both (1) and (2) by contradiction. For (1), suppose there are two different number m and n that satisfy $0 \le m < n \le 2N-2$, $2m \equiv sl \pmod{(2N-1)}$ and $2n \equiv sl \pmod{(2N-1)}$, then we can have $2m = sl$ and $2n = 2N - 1 + sl$. A contradiction is found that sl is an even number because $2m = sl$, while sl is also an odd number

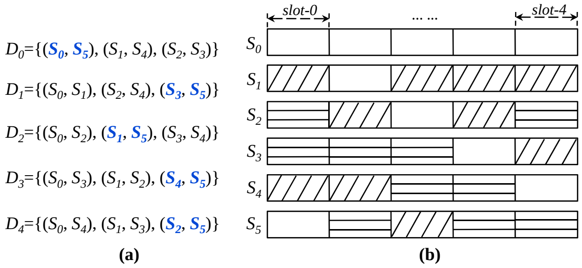


Fig. 1. Rendezvous schedules of a DSA network with 3 rendezvous channels.

	slot-0		... ...		slot-4
S_0	C_0	C_1	C_2	C_0	C_2
S_1	C_1	C_1	C_0	C_2	C_0
S_2	C_2	C_0	C_2	C_2	C_1
S_3	C_2	C_2	C_1	C_0	C_0
S_4	C_1	C_0	C_1	C_1	C_2
S_5	C_0	C_2	C_0	C_1	C_1

Fig. 2. CH sequences of a DSA network with 3 rendezvous channels.

because $2n = 2N-1+sl$. For (2), without loss of generality, we suppose $a < c$. If $b = d$, then we have $a + b = sl$ and $c + d = 2N - 1 + sl$. By subtracting these two equations we get $c - a = 2N - 1$ which is impossible because $0 \leq a < c \leq 2N - 2$.

In order to prove $\forall p, q \in [0, 2N - 2]$ $(p \neq q)$, rendezvous schedule D_p and schedule D_q are different, we need to prove $\forall d_i \in D_p (0 \leq i \leq N - 1)$ and $\forall d_j \in D_q (0 \leq j \leq N - 1)$, $d_i \neq d_j$. We prove this by contradiction. Suppose there exist $d_i \in D_p$ and $d_j \in D_q$ such that $d_i = d_j$, which means d_i and d_j contain the same pair of CH sequences. Suppose these two sequences are S_u and S_v, where $0 \leq u, v \leq 2N - 1$. Then we have $u + v \equiv p \pmod{(2N - 1)}$ and $u + v \equiv q \pmod{(2N - 1)}$, where $p, q \in [0, 2N - 2]$ and $p \neq q$, which is impossible. ■

B. Rendezvous Channel Assignment

After scheduling rendezvous among the empty CH sequences, SYNC-ETCH assigns rendezvous channels to each of these sequences. The goal of the rendezvous channel assignment is two-fold. First, to fully exploit the frequency diversity of a DSA network in establishing control channels, **all** the rendezvous channels should be utilized in every hopping slot. Second, the assignment needs to satisfy the even use of rendezvous channels requirement presented in Section III-A. Specifically, all the rendezvous channels should have roughly equal probabilities to appear in each CH sequence.

SYNC-ETCH employs a greedy algorithm to achieve the goal of rendezvous channel assignment. We briefly describe the algorithm here because of the page limit. In this algorithm, rendezvous channels are assigned to the $2N$ CH sequences slot by slot. When assigning channels to slot-i $(0 \leq i \leq 2N - 1)$ of all the CH sequences, the algorithm checks the rendezvous schedule D_i and selects the CH sequences pair $d_j = \{S_a, S_b\} \in D_i$ $(0 \leq j \leq N - 1)$ such that the sum of the number of outstanding channels of S_a and S_b are **greatest** (outstanding channels for a CH sequence are those channels that have not been assigned to the sequence). Then the algorithm chooses a rendezvous channel to assign to the slot-i of both S_a and S_b. This rendezvous channel is first selected from the intersection of slot-i's outstanding channels (i.e. the rendezvous channels that have not been assigned to the i-th slot of any CH sequence) and S_a's outstanding channels (suppose that S_a has more outstanding channels than S_b). If the intersection is empty, the channel is selected as a slot-i's outstanding channel that appears **fewest** times in S_a.

Fig. 2 shows the result of rendezvous channel assignment in a DSA network with 3 rendezvous channels, C_0, C_1 and C_2. CH sequences S_0 to S_5 are the final CH sequences in this network.

C. CH Sequence Execution

At the completion of constructing CH sequences, a node obtains a set of CH sequences, which are the same as those that any other node constructs. Then the node synchronizes to the existing nodes using the global synchronization mechanism, and starts the channel hopping process. The process is simple: the node randomly selects a CH sequence to follow. After it finishes all the slots, it performs the random CH sequence selection again and starts hopping on the newly chosen CH sequence. The node repeats this process while it is idle. The reason for the node to re-select a CH sequence when it finishes one is to make sure any pair of nodes are able to rendezvous in different rendezvous channels. Since the selection of CH sequence is random, the requirement of full utilization of rendezvous channels is satisfied. When a rendezvous channel's primary user appears, the nodes on the rendezvous channel should yield using the channel and wait until the next slot begins to hop to the next channel in the CH sequences.

V. ASYNC-ETCH

SYNC-ETCH is developed under the assumption that there exists a mechanism to make all nodes in a DSA network be able to periodically start hopping on CH sequences at the same global time. In contrast, ASYNC-ETCH can be used without assuming the existence of the global clock synchronization mechanism. Similar to SYNC-ETCH, nodes using ASYNC-ETCH also first construct a set of CH sequences when it joins the DSA network. The difference is that with ASYNC-ETCH, the node does not need to synchronize to the existing nodes. It starts the CH sequence hopping process immediately after the construction of the CH sequences is done.

Within a hopping period, a pair of nodes using ASYNC-ETCH that select the same CH sequence are guaranteed to rendezvous in 1 slot, and a pair of nodes that select two different CH sequences are guaranteed to rendezvous in N slots, no matter how the hopping processes of the pair of nodes are misaligned.

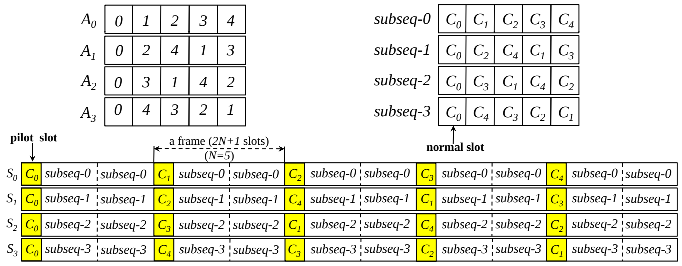


Fig. 3. CH sequences of a DSA network with 5 rendezvous channels.

The CH sequences are constructed in a way similar to SeqR [15]. However, we improve on [15] in the following aspects. First, our scheme derives multiple CH sequences rather than only one as in [15], which reduces the average number of nodes that select the same CH sequence. Second, our scheme makes nodes have more chance to rendezvous with each other within a hopping period, which further reduces the time-to-rendezvous of a pair of nodes in the average case. Moreover, we give rigorous proofs showing why the proposed scheme works without requiring the nodes being synchronized.

A. CH Sequences Construction and Execution

The algorithm that a newly joined node uses to construct the set of CH sequences is shown in Algorithm 2.

Given N rendezvous channels, where N is a prime number[3], Algorithm 2 returns $N-1$ CH sequences. The algorithm first constructs $N-1$ CH sub-sequences (line 5 to 7) that are derived by addition modulo the prime number N (line 1 to 4). Note that all integer sequences are derived with different addends. Then CH sequence S_i $(0 \leq i \leq N-2)$ is constructed from sub-sequence $subSeq_i$ (line 8 to 15) in a way that a pair of $subSeq_i$ are inserted into S_i following a *pilot slot* N times. The pilot slots of S_i, combined together, are exactly channels appearing in $subSeq_i$ in the same order. Slots in $subSeq_i$ are referred as *normal slots*. A pilot slot with its following two sub-sequences are called a *frame* in each CH sequence S_i. From Algorithm 2, it is easy to see that ASYNC-ETCH fulfills the requirement of even use of the rendezvous channels. Fig. 3 shows an example of the CH sequence construction process in a DSA network with 5 rendezvous channels.

After obtaining the set of CH sequences, the node starts the CH hopping process as specified in SYNC-ETCH: it randomly selects a new CH sequence to execute each time it finishes an old one. By doing this, it is ensured that any pair of nodes can meet in different rendezvous channels, which satisfies the requirement of full utilization of rendezvous channels, and we also eliminate the unfairness that nodes selecting the same CH sequence have less chance to rendezvous than nodes selecting different CH sequences.

B. Proof of Rendezvous

Here we show that ASYNC-ETCH satisfies the overlap requirement for CH sequences. Moreover, we analyze and prove the number of overlapping slots within a hopping period for two nodes using ASYNC-ETCH.

[3]Our scheme can be easily generalized for a non-prime number N. Discussion is omitted here to save space.

Algorithm 2 Asynchronous CH sequences Construction

Input: N rendezvous channels: $C_0, \cdots, C_{N-1}$ (N is prime)
Output: $N-1$ final CH sequences: $S_0, \cdots, S_{N-2}$

```
1  for i ← 0 to N − 2
2      A_i[0] = 0;
3      for j ← 1 to N − 1
4          A_i[j] = (A_i[0] + j(i + 1)) mod N;
5  for i ← 0 to N − 2
6      for j ← 0 to N − 1
7          subSeq_i[j] = C_{A_i[j]};
8  for i ← 0 to N − 2
9      k ← 0;
10     for j ← 0 to 2N^2 + N − 1
11         if j mod (2N + 1) = 0
12             S_i ← S_i ∪ (j, subSeq_i[j/(2N+1)]);   // pilot slot
13         else
14             S_i ← S_i ∪ (j, subSeq_i[k]);           // normal slot
15             k ← (k + 1) mod N;
16 return S_0, S_1, · · · , S_{N−2};
```

Our goal is to prove that for two nodes that select the same CH sequence constructed by Algorithm 2, they are guaranteed to rendezvous in at least 1 slot within a hopping period no matter how their hopping processes are misaligned (*Theorem* 3), and that for two nodes that select two different CH sequences, they are guaranteed to rendezvous at least N slots within a hopping period no matter how their hopping processes are misaligned (*Theorem* 4).

Before we prove the two theorems, We first borrow the definition of *rotation closure property* from [4] as follows.

Definition 1: Given a CH sequence S with p slots and a non-negative integer d, $\mathcal{R}(S,d) = \{(i, \mathcal{R}(S,d)[i]) \mid \mathcal{R}(S,d)[i] = S[(i+d) \bmod p]\}$ is called *a rotation of S with distance d*.

Definition 2: A CH sequence S with p slots is said to have the *rotation closure property with a degree of overlapping m* if $\forall d \in [0, p-1]$, $|S \cap \mathcal{R}(S,d)| \geq m$.

For instance, for a CH sequence with three hopping slots, $S = \{(0, C_0), (1, C_0), (2, C_1)\}$, the two possible rotations of it are $\mathcal{R}(S,1) = \{(0, C_0), (1, C_1), (2, C_0)\}$ and $\mathcal{R}(S,2) = \{(0, C_1), (1, C_0), (2, C_0)\}$, and S has the rotation closure property with a degree of overlapping 1.

Different from existing works, ASYNC-ETCH constructs multiple CH sequence rather than a single one. The concept that two CH sequences are different is defined as follows.

Definition 3: Two CH sequences, S_0 and S_1, each with p slots, are said be *different* if $\forall d \in [0, p-1]$, $S_1 \neq \mathcal{R}(S_0, d)$.

It is obvious that the $N-1$ CH sequences constructed by Algorithm 2 are different.

We now analyze the first case that two nodes select the same CH sequence.

Lemma 1: For two nodes periodically hopping on a CH sequence that has the closure property with a degree of overlapping m, they can rendezvous in at least $\frac{m}{2}$ slots within a hopping period no matter how their hopping processes are misaligned.

Proof: This lemma has been proved in [4]. ■

Theorem 3: For two nodes that select the same CH sequence constructed by Algorithm 2, they can rendezvous in at least 1 slot within a hopping period no matter how their hopping processes are misaligned.

Proof: We need to prove that for any CH sequence S_i $(0 \leq i \leq N-2)$ returned by Algorithm 2, S_i has the rotation closure property with a degree of overlapping 2, which combined with *Lemma* 1 can lead to this theorem. Specifically, we need to prove $\forall d \in [1, p-1]$, $\exists a \neq b \in [0, p-1]$ such that $S_i[a] = \mathcal{R}(S_i, d)[a]$ and $S_i[b] = \mathcal{R}(S_i, d)[b]$, where $p = 2N^2 + N$ is the number of slots of S_i.

If $d \bmod (2N+1) = 0$ (i.e., slot-0 of both $\mathcal{R}(S_i, d)$ and S_i are both pilot slots), then all $subSeq_i$ in both S_i and $\mathcal{R}(S_i, d)$ are aligned, there are $2N^2$ different overlappings.

If $d \bmod (2N+1) \neq 0$ (i.e., slot-0 in $\mathcal{R}(S_i, d)$ is a normal slot while slot-0 in S_i is a pilot slot), then we find the 2 overlappings as follows.

First, $\forall m, n \in [0, N-1]$ $(m \neq n)$, we have $S_i[m(2N+1)] \neq S_i[n(2N+1)]$ (since slot-0 in S_i is a pilot slot) $\Rightarrow \bigcup S_i[p(2N+1)] = \{C_0, \cdots, C_{N-1}\}$, where $p = 0, \cdots, N-1$, and $\mathcal{R}(S_i, d)[m(2N+1)] = \mathcal{R}(S_i, d)[n(2N+1)] \in \{C_0, \cdots, C_{N-1}\}$ (since slot-0 in $\mathcal{R}(S_i, d)$ is a normal slot). Then there must exist a $p \in [0, N-1]$ such that $S_i[p(2N+1)] = \mathcal{R}(S_i, d)[p(2N+1)]$.

Second, for $k = 2N + 1 - d \bmod (2N+1)$, slot-$k$ in $\mathcal{R}(S_i, d)$ is a pilot slot while slot-k in S_i is a normal slot. Then similarly to the previous case, we can conclude that there exits an $p \in [0, N-1]$ such that $S_i[p(2N+1)+k] = \mathcal{R}(S_i, d)[p(2N+1)+k]$. ■

Then we analyze the second case that two nodes select two different CH sequence. Before giving our conclusion, we give the definition of integer sequences derived by the method of addition modulo a prime number with different addends, and prove the overlap property of the integer sequences.

Definition 4: Two integer sequences, $A = \{a_0, \cdots, a_{N-1}\}$ and $B = \{b_0, \cdots, b_{N-1}\}$ where N is a prime number, are said to be derived by the method of addition modulo the prime number N with different addends m and n if $a_i = (a_0 + im) \bmod N$, $b_i = (b_0 + in) \bmod N$ where $0 \leq a_0, b_0 \leq N-1$, $1 \leq i \neq j \leq N-1$ and $1 \leq m \neq n \leq N-1$.

Lemma 2: Given two integer sequences derived by the method of addition modulo a prime number with different addends, $A = \{a_0, \cdots, a_{N-1}\}$ and $B = \{b_0, \cdots, b_{N-1}\}$, there must exist an integer $t \in [0, \cdots, N-1]$ such that $a_t = b_t$.

Proof: Let's prove by contradiction. Suppose $\forall t \in [0, \cdots, N-1]$, $a_t \neq b_t$. Construct a integers sequence $C = \{c_0, \cdots, c_{N-1}\}$, where $c_i = a_i - b_i$ $(0 \leq t \leq N-1)$. It is easy to see that $\forall c_i, c_j \in C$ $(0 \leq i \neq j \leq N-1)$, $c_i \neq c_j$, otherwise we can get $a_0 - b_0 + i(m-n) \equiv a_0 - b_0 + j(m-n) \pmod{N} \Rightarrow m - n$ is multiple times of N, which is impossible. Since $a_t \neq b_t$ $\forall t \in [0, \cdots, N-1]$, C contains N different integers that are in the range of $[1, N-1]$, which is a contradiction. ■

TABLE I
COMPARISONS BETWEEN COMMUNICATION RENDEZVOUS PROTOCOLS

	Avg. Rend. channel load	Average TTR	Rend. channels utilization ratio
M-QCH	$\frac{2}{3}$	$\frac{3}{2}$	$\frac{1}{N}$
L-QCH	$\approx \frac{1}{\sqrt{2N-1}}$	$\frac{2N-1}{2}$	$\frac{1}{N}$
SYNC-ETCH	$\frac{1}{N}$	$\frac{2N-1}{2}$	1
A-QCH	$\frac{1}{2}$	$\geq \frac{9}{2}$	N/A
SeqR	$\frac{1}{N}$	$\frac{N^2+N}{2}$	N/A
ASYNC-ETCH	$\frac{1}{N}$	$\frac{2N^2+N}{N-1} \approx 2N$	N/A

Theorem 4: For two nodes that select two different CH sequence constructed by Algorithm 2, there must be at least N overlapping slots within a hopping period between the two CH sequences no matter how their hopping processes are misaligned.

Proof: The theorem is intuitive, but the proof is cumbersome. We have omitted the proof due to the page limit. ■

VI. COMPARISONS

In this section, we theoretically compare ETCH with QCH [4] and SeqR [15], which are two existing CH based solutions for communication rendezvous in DSA networks.

In QCH, three versions of communication rendezvous protocols are designed. M-QCH and L-QCH are two synchronous versions that assume clocks are synchronized between nodes, and A-QCH is the asynchronous version that is used without such an assumption. The design goal of M-QCH is to minimize time-to-rendezvous between two CH sequences, while L-QCH's goal is to minimize the number of nodes that rendezvous in the same channel. SeqR is a DSA network communication rendezvous protocol without assuming global clock synchronization. SeqR does not have a synchronous version. We divide the comparisons into two group. In the first group, we compare SYNC-ETCH with M-QCH and L-QCH, all of which assume the existence of global clock synchronization. In the second group, we compare three asynchronous protocols: ASYNC-ETCH, A-QCH and SeqR.

We compare the two groups of communication rendezvous protocols on the three metrics introduced in section III-B: average rendezvous channel load, average TTR and rendezvous channels utilization ratio. Table I summarizes the comparison results, where N is the number of rendezvous channels of the DSA network.

In the synchronous protocols group, we pick parameters for L-QCH such that it produces the same number of CH sequences as SYNC-ETCH for the purpose of fair comparison. SYNC-ETCH outperforms M-QCH and L-QCH on the metrics of average rendezvous channel load and rendezvous channels utilization ratio, because in every hopping slot it efficiently utilizes all rendezvous channels in establishing control channels while in M-QCH and L-QCH only one rendezvous channel can be used as control channel. So theoretically SYNC-ETCH experiences less traffic collisions and achieves higher throughput than QCH. For the metric of average TTR, M-QCH achieves the best theoretical performance. However, it has a very large average load on each rendezvous channel ($\frac{2}{3}$ of all the network nodes use the same rendezvous channel), which will cause a high probability of traffic collisions and further make the time-to-rendezvous performance of M-QCH worse than its theoretical value in practice.

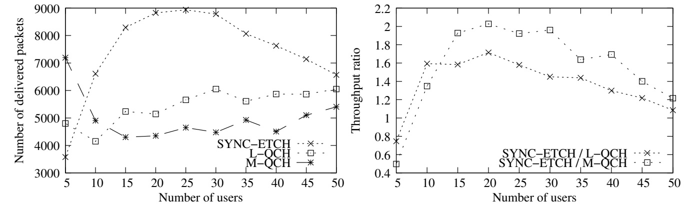
In the asynchronous protocols group, A-QCH has the worst performance in terms of average rendezvous channel load, because it only ensures two of the rendezvous channels can be used as control channels while both ASYNC-ETCH and SeqR utilize all the rendezvous channels in control channel establishment. Moreover, A-QCH cannot provide a bounded TTR. SeqR, which constructs only one CH sequence, can only guarantee one overlapping slot in a hopping period. So the average TTR for SeqR is half of the number of slots in the CH sequence. For ASYNC-ETCH's performance on the metric of average TTR, we make the following analysis: we proved in section V-B that for the cases that when two nodes select the same CH sequence and when they select two different CH sequences, they are respectively guaranteed to meet in at least 1 slot and at least N slot within a hopping period. Since ASYNC-ETCH generates $N-1$ different CH sequences and the CH sequence selection is random, on average there are $\frac{1}{N-1} + \frac{(N-2)N}{N-1} = N-1$ guaranteed overlapping slots in a hopping period. So the average TTR for ASYNC-ETCH is $\frac{2N^2+N}{N-1} \approx 2N$.

VII. Performance Evaluation

We evaluate ETCH's performance in the *ns-2* simulator by comparing with QCH and SeqR. Similar to Section VI, we divided the evaluation into two portions based on the assumption about the existence of global clock synchronization. In section VII-A, we compare the performance of SYNC-ETCH with those of M-QCH and L-QCH. In section VII-B, we compare ASYNC-ETCH with A-QCH and SeqR.

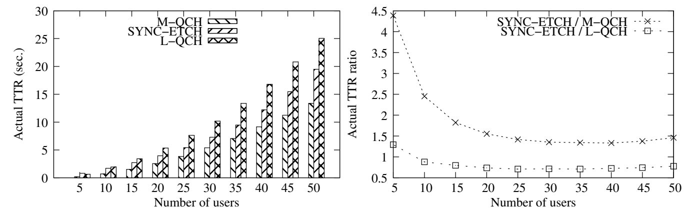
We modified the *ns-2* simulator to make it be able to perform multi-channel simulations based on the Hyacinth project [17]. In our simulations, there are a varying number of nodes in a $500m \times 500m$ area where each of the nodes is in all other nodes' communication ranges. The length of a hopping slot is set to 100 ms. We establish Constant Bit Rate (CBR) flows, where the packet size is set to 800 bytes and the packet rate is 125 packets/sec, from each node to all other nodes. These flows are started and stopped randomly during the simulation such that there is no more than one flow from the same node is activated simultaneously (this is because there is only one transceiver equipped with each node). Hyacinth's manual routing protocol is used in routing packets between the nodes. We disabled the RTS/CTS function in the simulator, and rely on the retransmission mechanism to deal with packet collisions.

In the simulations, the DSA network has 5 rendezvous channels each of which can possibly used by the primary user anytime. All the secondary users are supposed to be within the communication range of the primary user. The appearances of the primary user is simulated as follows. We first decide whether the primary user shows or not by flipping a coin. If the primary user appears, we randomly disable a rendezvous channel for a random period of time. Otherwise all the rendezvous channels are made to be available to the nodes also for a random period of time. We repeat this process during the whole simulation.



(a) Traffic throughput (b) Throughput ratio

Fig. 4. Throughput performances of the synchronous protocols.



(a) Actual time−to−rendezvous (TTR) (b) Actual TTR ratio

Fig. 5. TTR performances of the synchronous protocols.

A. Synchronous Communication Rendezvous Protocols

We performed two simulations to study the performances of the synchronous protocols on traffic throughput and actual time-to-rendezvous (TTR) respectively. We ran each simulation ten rounds with different secondary users in each round.

Fig. 4 shows the traffic throughput performances of the three synchronous protocols. Part (a) of this figure shows the actual throughput while part (b) illustrates the improvement ratio curves of SYNC-ETCH over L-QCH and M-QCH. SYNC-ETCH has a lower throughput than L-QCH and M-QCH when there are 5 secondary users in the network. This is because in CH sequences of L-QCH and M-QCH, rendezvous channels are randomly assigned to those non-frame-channel-slots, which may give a pair of nodes using L-QCH or M-QCH extra slots to rendezvous in other than the frame-channel-slot. And this is also because there are no or little collisions in this case. However, when the number of secondary users is equal or greater than 10, SYNC-ETCH achieves higher traffic throughput than L-QCH and M-QCH, especially when the nodes-channels ratio is in the range of 3 to 6 (i.e. when there are 15 to 30 nodes in the DSA network). In this case, traffic collision dominates the factors that influence the throughput performance. With both L-QCH and M-QCH, nodes are always compete for one rendezvous channel as control channel leaving all other rendezvous channels unused in a hopping frame, which causes a high probability of collisions when the nodes-channels ratio is bigger than 1. On the contrary, SYNC-ETCH schedules rendezvous among its CH sequences

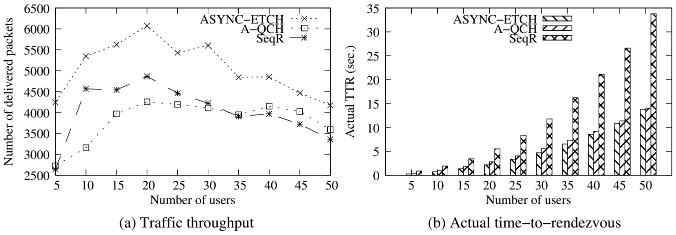


Fig. 6. Throughput and TTR of the asynchronous protocols.

such that all the rendezvous channels can be utilized in every hopping slot. This approach greatly reduces traffic collisions and hence increases throughput. Furthermore, it can be also noticed in Fig 4 that the throughput performance of the three synchronous protocols converges as the nodes-channels ratio approaches 10. This is because collisions dominate traffics in each rendezvous channel with all the synchronous protocols. In this case, it is suggested to assign more rendezvous channels to accommodate such a high number of secondary users.

Fig. 5 part (a) shows the TTR performances of the three synchronous protocols, and part (b) demonstrates the TTR ratios of SYNC-ETCH over L-QCH and M-QCH. The TTRs of the three protocols increase as the number of secondary users grows because of the increasing traffic collisions. Although M-QCH achieves the best TTR performance among the three as analyzed in section VI, it does not get the theoretical TTR performance boost over SYNC-ETCH. Theoretically speaking, M-QCH performs 3 times better than SYNC-ETCH in TTR (i.e. M-QCH's TTR is $\frac{1}{3}$ of SYNC-ETCH's TTR), because it has an average TTR of 1.5 while SYNC-ETCH's value is 4.5. However, SYNC-ETCH's actual TTR is only on average 1.5 times that of of M-QCH from the simulation result. This is because the nodes using M-QCH experience more severe traffic collisions that those using SYNC-ETCH, which deteriorates the theoretical TTR performance of M-QCH in practice.

From the above two simulations it can be seen that SYNC-ETCH achieves the best balance between traffic throughput and TTR among the three synchronous protocols.

B. Asynchronous Communication Rendezvous Protocols

In this subsection, we compare the throughput and the TTR performances between the three asynchronous protocols: ASYNC-ETCH, A-QCH and SeqR.

Fig. 6 shows the performances of the three asynchronous protocols. In Fig. 6 part (a), the traffic throughput performances are shown. ASYNC-ETCH performs constantly better than the other two protocols in this metric. This is because ASYNC-ETCH is able to utilize all the rendezvous channels as control channels while A-QCH uses only two of them. Meanwhile, ASYNC-ETCH improves on SeqR such that it achieves a shorter average TTR, which contributes to the throughput performance boost over SeqR. Fig. 6 part (b) shows the actual TTR performances of the three protocols. It is not surprised that ASYNC-ETCH performance better than SeqR, because ASYNC-ETCH's average TTR is shorter than that of SeqR (see Table I for details). For A-QCH, we construct CH sequences such that they have an average TTR of 4.5, which is the best that A-QCH is able to achieve. Even so, ASYNC-ETCH still performs better than A-QCH.

VIII. Conclusion

We have presented ETCH, efficient channel hopping based communication rendezvous protocols for DSA networks. ETCH protocols include SYNC-ETCH and ASYNC-ETCH. SYNC-ETCH, which assumes global clock synchronization, efficiently utilizes all the rendezvous channels in establishing control channels all the time. ASYNC-ETCH is able to make a pair of nodes rendezvous without being synchronized. Using a combination of theoretical analysis and simulations, we show that ETCH protocols perform better than the existing solutions for communication rendezvous in DSA networks.

Acknowledgment

The authors would like to thank all the reviewers for their helpful comments. This project was supported in part by US National Science Foundation grants CNS-0831904, DMS-0852452 and CAREER Award CNS-0747108.

References

[1] J. Zhao, H. Zheng, and G. H. Yang, "Distributed coordination in dynamic spectrum allocation networks," in *IEEE DySPAN*, December 2005.
[2] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "Channel-hopping multiple access," in *IEEE ICC 2000*, 2000.
[3] P. Bahl, R. Chandra, and J. Dunagan, "Ssch: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad hoc wireless networks," in *ACM Mobicom*, September 2004.
[4] K. Bian, J. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *ACM Mobicom*, June 2005.
[5] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *COMPUTER NETWORKS JOURNAL (ELSEVIER)*, 2006.
[6] A. Sahai, N. Hoven, and R. Tandra, "Some fundamental limits on cognitive radio," in *Allerton Conference on Communication, Control, and Computing*, 2004.
[7] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Asilomar Conference on Signals, Systems, and Computers*, 2004.
[8] A. Fehske, J. Gaeddert, and J. Reed, "A new approach to signal classification using spectral correlation and neural networks," in *IEEE DySPAN*, 2005.
[9] A. Ghasemi and E. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *IEEE DySPAN*, 2005.
[10] S. Shankar, "Spectrum agile radios: Utilization and sensing architectures," in *IEEE DySPAN*, 2005.
[11] B. Wild and K. Ramchandran, "Detecting primary receivers for cognitive radio applications," in *IEEE DySPAN*, 2005.
[12] H. Zheng and L.Cao, "Device-centric spectrum management," in *IEEE DySPAN*, 2005.
[13] V. Brik, E. Rozner, S. Banarjee, and P. Bahl, "Dsap: a protocol for coordinated spectrum access," in *IEEE DySPAN*, 2005.
[14] L. Ma, X. Han, and C. Shen, "Dynamic open spectrum sharing mac protocol for wireless ad hoc networks," in *IEEE DySPAN*, 2005.
[15] L. A. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *IEEE DySPAN*, October 2008.
[16] *Maxim 2.4GHz 802.11b Zero-IF Transceivers*, Maxim Integrated Products. [Online]. Available: http://pdfserv.maxim-ic.com/en/ds/MAX2820-MAX2821.pdf
[17] "Hyacinth: An ieee 802.11-based multi-channel wireless mesh network," *http://www.ecsl.cs.sunysb.edu/multichannel/*.