DISCERN: Cooperative Whitespace Scanning in Practical Environments

Tarun Bansal, Bo Chen and Prasun Sinha
Department of Computer Science and Engineering
The Ohio State University, Columbus, OH 43210
{bansal, chebo, prasun}@cse.ohio-state.edu

***Abstract*—Cognitive radio devices opportunistically operate on whitespace channels, provided those channels are not in use by the primary users. This opportunistic reusing of channels requires secondary users to perform fast and efficient sensing to determine the unused channels. Although individual secondary clients may be unwilling to frequently sense all the channels, their density could be exploited for tasking the individual devices to collaboratively extract useful information on spectrum usage. It is critical to determine how the sensing tasks should be assigned to different secondary users. This is particularly challenging in practical networks due to the variability in the sensing accuracy of different users that may arise because of multipath effects on the signal and varying distances from the primary transmitters. Further, presence of multiple Primary Users on the same channel makes it challenging to select the best users for sensing. Finally, to reduce the sensing overhead, it is beneficial to limit the number of channel sensing tasks that can be performed within a given time period. We propose a novel metric that captures the sensing accuracy of a given sensing assignment. Using our metric, we design an algorithm DISCERN for computing the sensing assignment that maximizes the sensing accuracy. Our algorithm is the first to take into account the limitations in practical networks. Our work is motivated by experimental measurements. Trace-driven simulations show that DISCERN increases the sensing accuracy by at least 30%. Theoretical analysis shows that the sensing assignment computed by DISCERN performs within 63% of the exponential-time optimal solution.**

I. INTRODUCTION

In recent years the limited use of licensed spectrum has been observed and deemed as an opportunity [1] to exploit for operating secondary networks [2]. Dynamic solutions for re-using the licensed channels require efficient and frequent channel sensing. It has been shown that the channel sensing capabilities can be enhanced by using multiple cooperative sensors [3]. *The objective of this paper is to design a solution to enable highly accurate sensing under strict budget constraints in terms of time allocated for sensing while taking into account various practical considerations.*

A naive option is to simply select all the Secondary Users (SUs) to sense all the channels. Requiring every user to sense would result in considerable loss of throughput at the Secondary Base Stations (SBS) and the SUs. However, selecting a subset of SUs for sensing is challenging due to the following reasons:

- **Varying channel detection accuracy:** In practical environments, different SUs may face different levels of shadowing, multipath fading and path losses, and thus have different accuracy of detecting primary channel activity. Further, these losses are channel-dependent due to the difference in relative positioning of Primary Users (PUs) with respect to the SUs and are difficult to estimate using statistical models [3]. This makes it challenging to estimate the accuracy of individual SUs [3] for sensing different channels.
- **Unknown locations of PUs:** An SBS may have a range of as much as 35 km [4], [1] with hundreds of associated clients [1]. Due to this large coverage area of SBS and small transmission range of some licensed users such as femtocells, mobile communication centers (deployed in disaster recovery scenarios or by the military), mobile cell sites, and microphones, it is possible that multiple PUs are present in the range of a single SBS on the same channel. The lack of prior knowledge of the precise locations of the PUs in the region makes it challenging to decide which users should be sensing which channels.
- **Limited resources allocated for sensing:** Sensing a channel requires all the neighboring SUs to be quiet [5], resulting in low throughput. Further, for every channel sensed, the SUs have to report the sensing result back to the SU base station resulting in loss of throughput at both the SU and the SU base station. Thus, it is beneficial for the SU base station to estimate the channel states with minimum sensing overhead.

The accuracy of sensing a particular channel depends on the set of SUs that are assigned the task of sensing the channel. As an illustration, consider the example network shown in Figure 1. Here, if the SBS has to select a set of 3 SUs for sensing, then some choices for selecting the sensing nodes are as follows: (i) Selecting nodes with maximum individual accuracy: say, $\{n_3, n_4, n_5\}$; (ii) Selecting nodes with maximum distance among nodes: say, $\{n_1, n_3, n_6\}$; (iii) Selecting nodes with high RSSI value of the primary user signal [6]: say, $\{n_3, n_4, n_5\}$; and (iv) Selecting nodes with minimum correlation among them [7]: say, $\{n_1, n_3, n_6\}$. However, it is important to select secondary users so that both the PUs can be detected with high accuracy. As n_1 and n_2 are behind obstacles with respect to Primary User 1, their individual sensing accuracy is lower while the sensing accuracies of n_3, n_4 or n_5 in detecting Primary User 2 are higher. So, it may be enough to select one SU for scanning Primary User 2 while scanning Primary User 1 may require two secondary users. Thus, an allocation that can provide high enough accurate detection of activity of both the primary users will be $\{n_1, n_2, n_3\}$. When the SBS has to compute the sensing assignment for multiple

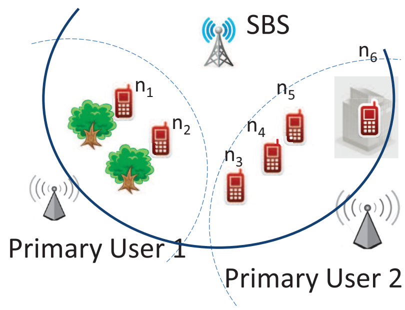


Fig. 1. Network consisting of an SBS, associated SUs and two PUs, all operating on the same channel. Here n_6 is located indoors, and thus has lower accuracy than n_3, n_4 or n_5. Although n_3 and n_6 are in the range of the same PU, still the correlation between them would be low since the readings of n_3 will be different from those of n_6 due to different sensing accuracies.

channels, the problem becomes significantly more challenging as it also needs to decide how many nodes should sense each of the channels and ensure that the budget constraints are not exceeded. Thus, with the presence of obstacles in the environment and the presence of multiple PUs, it becomes challenging to select the best SUs for sensing. However, none of the previous works in the area of cooperative sensing take into account all the three challenges when computing the sensing assignment. In this paper, we propose an algorithm called DISCERN, that given a set of channels, determines the best sensing assignment. *To the best of our knowledge, DISCERN is the first algorithm that takes into account the presence of multiple PUs as well as differences in accuracy of different SUs when computing the sensing assignment.* This paper makes the following contributions:

- We define a novel concept called *knowledge factor* which is an estimation of the probability that two users are sensing the same set of primary users.
- Using knowledge factor, we define a new metric called Ω-metric that characterizes the accuracy of a given sensing assignment.
- Based on the metric we design a greedy algorithm for computing the sensing assignment. Using theoretical analysis, we show that the sensing assignment computed by DISCERN performs within 63% of the exponential-time optimal solution.
- Using extensive trace-driven simulations, we show that compared to three other algorithms DISCERN increases the channel sensing accuracy by at least 30%.

The rest of the paper is organized as follows. Section II discusses related work in the area of cooperative sensing. Section III presents the system model and the problem statement. The next section describes the computation of the proposed metric and the DISCERN algorithm that determines the sensing assignment using the metric. Section V presents the results of our experiments and simulations. Finally, the paper is concluded in Section VI along with pointers to future work.

II. Related Work

In order to jump start the re-use of licensed spectrum, FCC has ruled that the primary users (PUs) update their usage information in a location specific database which can then be looked at by cognitive radio devices willing to re-use the channels [8]. Although this is a simple approach for the short term, it has several limitations going forward. First, it requires the primary users to register the time intervals when they plan to use it in advance, which makes it difficult to make changes in the usage patterns under short notice. Second, in case a PU is not transmitting, there is no incentive for it to de-register a previously registered transmission interval with the database authority. Therefore, channels that are marked as *busy* in the database may actually be *free* resulting in spectrum wastage. And third, it requires knowledge of geographical location of the secondary users and connectivity with the database, which may not be available in all scenarios. As pointed out by FCC [8], [9] itself, the use of a database is only a good solution until we are able to overcome the limited physical sensing capability of cognitive radios.

Using experiments, Sahai *et al.* [3] have shown that by combining the readings from multiple cooperative sensors, it is possible to reduce the overall probability of miss. However, it is not clear how multiple SUs should be selected for sensing such that the accuracy of their combined readings is maximized. The problem of selecting a subset of SUs for sensing is challenging because of multiple factors: (i) Multiple PUs such as femtocells, mobile cellular sites, and microphones can be present in the area that are operating on the same channel; (ii) Statistical models do not hold well in practical environments due to obstacles and multipath effects leading to different SUs having different sensing accuracies that are difficult to estimate [3]; and, (iii) Different SUs have different sensing accuracy. However, all the previous works related to the problem of selecting the best SUs for cooperative sensing either implicitly assume that there is only a single PU in the network [10], [11], [12], [6], [13], [14], [15] or they neglect presence of obstacles by assuming that statistical models of signal transmission hold [14] or they simplistically assume that every SU has identical accuracy [10], [11] or they assume that the probability of detection and probability of false positive of SUs is known beforehand [15].

Authors in [6], [13] use the RSSI values to select the best nodes for sensing. A common drawback with RSSI based selection algorithms is that all the SUs selected by them for sensing might be associated with only one PU. This may leave other PUs on the same channel as unmonitored (as seen in example network in Section I). Authors in [16], [12], [7] have proposed using correlation to select the best nodes for sensing. However, they do not consider the effect of probability of false positives on the accuracy of readings. Further, if some nodes have low probabilities of detection, then their algorithms may select many such nodes for sensing since their correlation with other nodes in the system will typically be low (as seen in example network in Section I where correlation between n_3 and n_6 was low). On the other hand, DISCERN computes the probability that two SUs are in range of the same PU using a novel concept called *knowledge factor* instead of correlation, which helps it to avoid selecting SUs with low probability of detection for sensing.

For networks with multiple PUs, [17] improves the sensing accuracy by forming distance-based clusters. However, their algorithm does not take into account the presence of obstacles in the network, and may select SUs with low sensing accuracy.

Apart from using cooperation, some authors have proposed using wide-band sensing to reduce the time spent in sensing. However, performing wide-band sensing requires sophisticated hardware [5] which may not be always available. Further, some authors propose transmitting the complete raw signal to the SU Base Station instead of taking decisions distributively at different SUs. This requires a wide-band control channel which may not be always available [5]. A similar problem of cooperative sensing has been explored in the area of Wireless Sensor Networks [18], [19] as well. However, there the challenge is to cover the whole geographical area, rather than covering all the *event sources*. Further, it is not clear how their algorithms can be extended to multiple channels.

III. System Model and Problem Statement

We consider a secondary network (Figure 1) consisting of secondary user base stations (SBSs) and secondary user clients (SUs), both of which are equipped with cognitive radio based transceivers. Unused primary channels can be used by the secondary network. *In this work we focus on a single SBS and its associated SUs.*

For a secondary user n_i and channel c_k, π_{ik} represents the set of primary users operating in channel c_k for which the received signal strength at n_i is above a certain minimum threshold. To simplify the notation, we avoid the channel subscript and use π_i when the discussion is only limited to a single channel. The presence of obstacles like trees, walls etc. and multi-path effects may introduce sensing errors at the SUs. These errors can be of two types: (i) *miss*: when an SU fails to detect the presence of a PU; and, (ii) *false positive*: when an SU incorrectly determines the channel to be busy. Table I lists the commonly used symbols in this paper. We assume that probability of detection of all SUs (P_i^d) is at least as much as their probability of false positive (P_i^f).

Time is divided into *rounds* of fixed durations. At the beginning of each round, the SBS assigns a subset of channels to each SU for scanning. The assignment is represented by a set of tuples $S = \{(n_i, c_k) : \text{SU } n_i \text{ scans channel } c_k\}$. The SUs scan the assigned channels during the predetermined *quiet period* [5] in that round and send the binary results (OFF/ON) back to the SBS. Based on these results, the SBS determines the channel that should be used for communication (if needed).

In a practical system, it is possible that an SU is simultaneously in the range of multiple PUs on the same channel. In that case, for the sake of exposition, we substitute the set of all PUs in π_i by a single *virtual PU*. This virtual PU is ON *iff* any PU in π_i is ON. We assume that our system is *quasi-static* at all times implying that mobility speed of SUs is slow enough such that the system is able to adapt to mobile SUs. Mobility based simulations are presented in Section V.

A. Problem Statement

Let $\mathcal{M}$ be a set of channels and S be a sensing assignment that scans the channels in $\mathcal{M}$. After the scanning is over,

TABLE I. Symbols used

n_i	Node or secondary user client (SU) i
$\mathcal{N}$	Set of all SUs that are associated with the given SBS
c_k	Channel k
π_{ik}	Set of primary users that operate on c_k and in whose interference range n_i lies
x_{ik}	1 if n_i finds c_k to be occupied, otherwise 0
ρ	Limit on number of scans performed by SBS in a single round
S	Scanning assignment
P_{ik}^d	Probability of detection of n_i on channel c_k given by $P(x_{ik} = 1 \mid \pi_{ik} \text{ is ON})$
P_{ik}^f	False positive probability of n_i on channel c_k given by $P(x_{ik} = 1 \mid \pi_{ik} \text{ is OFF})$
P_{ij}	Probability that $\pi_i = \pi_j$ on a given channel

the SUs in S will send their binary scan result (x_{ik}) to the SBS which will aggregate them to estimate the state of all channels in $\mathcal{M}$. We need to compute the sensing assignment that maximizes the accuracy of the estimation of the state of $\mathcal{M}$ channels subject to constraint on the number of scans performed:

$$\max_{S} \ \Omega(S) = \frac{1}{|\mathcal{M}|} \sum_{c_k \in \mathcal{M}} \Omega(S, k) \quad (1)$$

$$\begin{aligned} where\ \Omega(S,k) = \ & \text{Probability of correctly estimating the} \\ & \text{state of the channel } c_k \text{ after} \\ & \text{aggregating readings from SUs in } S \\ \text{subject to} \quad & |S| \le \rho \end{aligned}$$

Here, metric $\Omega(S)$ captures the probability that after processing the scan results from SUs in S, the state of channels in $\mathcal{M}$ estimated by the SBS is correct. For a single channel (say c_k), the metric $\Omega(S, k)$ captures the probability that after processing the scan results from SUs in S, the state of channel c_k estimated by the SBS is correct. Solving (1) is NP-Hard (See Technical Report [20] for proof). In this paper we present a polynomial time greedy algorithm for the above problem.

IV. Discerning the Primary Users

In this section, we explain how DISCERN computes the best sensing assignment. To determine the best sensing assignment, we first define a metric that captures the sensing accuracy of a given subset of nodes (all of which are sensing the same channel) that satisfies the following requirements: (i) **Accounts for variability per PU**: It is possible that due to the large transmission range of SBSs [3], multiple independent PUs operating on the same channel are present in its range. This is especially true for microphones [3] and GSM channels [21] where the range of transmission of licensed users is not large. (ii) **Accounts for variability per SU**: The metric should take into account the individual sensing accuracy of different SUs as well as variations in the sensing accuracy of the same SU on different channels. (iii) **Normalized**: The value of the metric should preferably lie between a *min* and *max* value. The value should be *min* when the sensing result given by the subset of nodes is not more accurate than random guess and *max* when the sensing result given by the subset of nodes is expected to be always correct. Using such a metric, we can compute the sensing accuracy of a set of nodes for the given channel.

Finally, to determine the best sensing assignment, we can find a subset of nodes from $\mathcal{N}$ that maximizes the sensing accuracy while satisfying the restrictions on the maximum number of scans performed.

However, before we explain our own metric, we first study the suitability and shortcomings of some naive metrics that *may possibly* capture the scanning accuracy of a sensing assignment. For the sake of exposition, we assume that all the SUs in the assignment (say S) are sensing the same channel, say c_k:

- **Average** P^d: Metric is computed by taking average of P^d_{ik} over all SUs in S.
- **Area Covered**: The metric measures the percentage area of the region that is covered by SUs in S.
- **Average RSSI**[6], [13]: Metric is computed by taking the average RSSI of PU signals at all nodes in S.
- **Average Correlation**[16], [12], [7]: Computes sum of correlation among all pairs of SUs that are scanning c_k. Lesser the correlation, higher is the metric.

None of these metrics take into account the P^f of the SUs. Thus, they are unable to give higher priority to SUs with lower P^f. Further, as shown in Section I through the example network, *Average* P^d and *Average RSSI* do not work well when multiple PUs are present in the range of the SBS since they do not necessarily "cover" all the PUs. In the presence of obstacles and non-circular scanning ranges, *Area Covered* will give incorrect results. Similarly, *Average Correlation* may incorrectly select SUs with low detection accuracy as shown through the example network in Section I. *Thus, there is a need to define a new metric that meets the goals defined earlier.*

DISCERN Overview: When multiple PUs are present within range of the SBS, then for computing the best sensing assignment, it is imperative for the SBS to determine the SUs that are in range of the same PU. Further, it also needs to know the P^d and P^f of all SUs. In subsection IV-A, we explain how the SBS can estimate the probability that two SUs are in the range of the same PU. In the next subsection, we show how the metric $\Omega(S, k)$ is computed. In Subsection IV-C, we describe our algorithm DISCERN that uses the Ω-metric to compute the best sensing assignment. In Subsection IV-D, we show how the SBS can determine the SU specific parameters (such as P^d and P^f). Finally, in Subsection IV-E, we discuss the overhead of DISCERN.

A. Computing the probability that two SUs are in range of same PU

To obtain the best sensing assignment, it is important to determine which SUs are in the interference range of the same PU. Using this information, we can select only a few SUs per PU for sensing. In this section, we propose a method for computing the probability of two given SUs (say n_i and n_j) to be in the interference range of the same PU on a given channel (say c_k). For ease of notation, we will omit the channel subscript in the following discussion.

One way of determining if two SUs are in the range of the same PU is to compute the correlation [16] between them and if the correlation is above a certain threshold, it can be said that the two SUs are in the range of the same PU. However, this does not work well if one of the SUs is located behind obstacles as shown in the example network in Section I where n_3 and n_6 could have low correlation even though they are in the range of same PU. Therefore, such a correlation-based mechanism could incorrectly select both n_3 and n_6 for the sensing assignment. *Thus, we need another mechanism that works well even in the presence of obstacles.* To that end, we first define the *knowledge factor* K_{ij}, as the knowledge added by n_i to n_j about the state of the PU. We define K_{ij} as follows:

$$\begin{aligned} K_{ij} &= \min\left(1, \frac{P(x_i = 1 \cap x_j = 0)}{P(x_i = 1)P(x_j = 0)}\right) \\ &= \min\left(1, \frac{P(x_i = 1 \mid x_j = 0)}{P(x_i = 1)}\right) \end{aligned}$$

Next, we will intuitively describe how K_{ij} changes depending on if n_i and n_j are in the range of same PU. Lets consider the two cases:

- When n_i and n_j are in the range of same PU (i.e., $\pi_i = \pi_j$): First, assume that $P^d_i \leq P^d_j$. If x_j is 0, it indicates that π_j is likely OFF since SU n_j with high P^d did not detect PU activity. Now, since $\pi_i = \pi_j$, therefore, the probability that π_i is ON is also low. This implies that the probability that n_i would detect PU is also low. Thus, the value of $P(x_i = 1 \mid x_j = 0)$ would be low. Therefore, K_{ij} would also be low. Similarly, if $P^d_j \leq P^d_i$, then K_{ji} would be low.
- When n_i and n_j are in the range of different PUs: Then, $P(x_i = 1 \mid x_j = 0) \approx P(x_i = 1)$ and therefore, $K_{ij} \approx K_{ji} \approx 1$.

Therefore, when the two SUs are in range of the same PU, then at least one of K_{ij} or K_{ji} is small (regardless of the presence of obstacles) whereas when the two SUs are in range of different PUs, then both K_{ij} and K_{ji} are close to 1. Therefore, we can estimate the probability P^k_{ij} that indicates if n_i and n_j are in the range of the same PU or not on c_k, as follows:

$$P_{ij} = P_{ji} = P(\pi_i = \pi_j) = 1 - \min(K_{ij}, K_{ji}) \qquad (2)$$

If the SUs are present in the range of multiple PUs, then P_{ij} would indicate the probability that n_i and n_j are in the range of same set of PUs. In Section V, using experiments we compare the accuracy of P_{ij} and "correlation" in correctly estimating if two SUs are in the range of same PU. In practice, storing and comparing the readings of all pairs of SUs for all the previous rounds is both memory and computation intensive. Instead, we use Exponential Moving Average (EMA) to update the value of P_{ij} at the end of each round. If $P_{ij}(u, v)$ denotes the probability computed based on the received readings from round u to round v, and β is the coefficient of EMA, then,
$P_{ij}(1, r) = \beta P_{ij}(1, r-1) + (1-\beta)P_{ij}(r, r)$
where $P_{ij}(r, r)$ is computed using (2). To account for mobility, higher weight can be given to the recent readings.

B. Metric Computation

Next, we explain how the $\Omega(S, k)$ metric can be computed. The value of $\Omega(S, k)$ should be high if the scans done by the set of nodes in S is able to capture the state of all PUs on c_k with high accuracy. Thus, for a sensing assignment S with high $\Omega(S,k)$, it is expected that after processing the sensing results, our confidence about the ON/OFF status of all PUs in the network operating on channel c_k would be high. Since, we do not know the locations or the number of PUs operating on c_k, we start with the assumption that every SU (say n_j) is in interference range of some PU (denoted by π_{jk} or π_j after omitting channel subscript in favor of simpler notation). Therefore, given a scanning assignment, we first determine how confident we will be about the status of π_j that is interfering at n_j.

Confidence about state of π_j: It is possible to estimate the state of π_j using the sensing results from n_j as well as other nodes that scanned the channel c_k. Some other node (say n_i) can correctly estimate the state of π_j if: (i) It is also in range of π_j; (ii) It has high probability of detection; and, (iii) It has low probability of false positives. Since, P_{ij} is also the probability of n_i being in the range of π_j, therefore, we can estimate the probability that n_i will correctly estimate the state of π_j as: $P_{ij}(P_i^d - P_i^f)$. This can also be called as the accuracy of n_i.

Let S_k be the set of SUs that scan channel c_k as part of sensing assignment S. It is possible to estimate the state of π_j by aggregating the readings of all SUs in S_k. Therefore, the accuracy of estimating the state of π_j can also be computed by aggregating the accuracy of all SUs in S_k. Further, this aggregation should satisfy the following requirements: (i) If one of the SUs in S_k has an accuracy of 1, then the aggregated accuracy should also be 1, since it is possible to estimate the state of π_j from only the readings of this SU; and, (ii) Its value should preferably lie between 0 and 1: 1 when it is always possible to correctly estimate the state of π_j; and, (iii) With increase in the cardinality of S_k, aggregated accuracy should increase since more observations about the state of π_j are available. If $\Omega(S,k,j)$ is the aggregated accuracy of SUs in S_k in estimating the activity of π_j, then we can quantify this accuracy as:

$$\begin{aligned}\Omega(S,k,j) &= \begin{cases} 0 & \text{if } S_k = \phi, \\ 1 - \prod_{n_i \in S_k} (1 - \text{Accuracy of } n_i) & \text{if } S_k \neq \phi. \end{cases} \\ &= \begin{cases} 0 & \text{if } S_k = \phi, \\ 1 - \prod_{n_i \in S_k} (1 - P_{ij}(P_i^d - P_i^f)) & \text{if } S_k \neq \phi. \end{cases}\end{aligned} \tag{3}$$

Since, each of the $|\mathcal{N}|$ SUs can be in the interference range of a unique PU, therefore there can be as many as $|\mathcal{N}|$ PUs that are operating on c_k. $\Omega(S,k)$ is estimated by taking an average over these $|\mathcal{N}|$ SUs:

$$\Omega(S,k) = \frac{1}{|\mathcal{N}|} \sum_{n_j \in \mathcal{N}} \Omega(S,k,j) \tag{4}$$

To compute the metric across all the channels in the set $\mathcal{M}$, we take an average over all the channels as shown in (1).

C. Computing the sensing assignment

At the end of round r, DISCERN (Algorithm 1) computes the scanning assignment S for round $r+1$ by working in a greedy fashion such that at each step, it adds that node-channel pair to S which maximizes the value of $\Omega(S)$. This process is repeated until no node-channel pair exists that can be added to S. After the sensing assignment S is computed for round $r+1$, the SBS broadcasts it to all the SUs. In Section IV-D, we would explain how SBS computes the value of P_d and P_f for all SUs and P_{ij} for all pairs of SUs.

Algorithm 1: Algorithm DISCERN

Input: Values of P_i^d, P_i^f for all SUs; and P_{ij} for all pairs of SUs
Output: Scanning assignment S for round $r+1$
$S \leftarrow \phi$
while $|S| < \rho$ **do**
    $n_{max} \leftarrow NULL, c_{max} \leftarrow NULL$
    forall $\{(n_i, c_k) : n_i \in \mathcal{N} \wedge (n_i, c_k) \notin S\}$ **do**
        if $\Omega(S \cup \{n_i, c_k\}) > \Omega(S \cup \{n_{max}, c_{max}\})$ **then**
            $n_{max} \leftarrow n_i, c_{max} \leftarrow c_k$
    if $n_{max} = NULL$ **then**
        return S
    $S \leftarrow S \cup \{n_{max}, c_{max}\}$
return S

Next, we prove that the sensing assignment computed by DISCERN optimizes (1) within a factor of 0.63 when compared to exponential time optimal algorithm. The proof of the theorem is included in the technical report [20].

Theorem 4.1: If $S_{DISCERN}$ is the sensing assignment computed by DISCERN and $S_{optimal}$ is the sensing assignment computed by the exponential time algorithm that optimizes (1), then $\frac{\Omega(S_{DISCERN})}{\Omega(S_{optimal})} \geq 63\%$.

D. Determining SU-specific parameters

In the previous subsection, we assumed that the SBS is aware of the P^d and P^f of different SUs, however this information may not be available in practical networks. In this subsection, we show how the SBS can determine SU-specific parameters (P^d, P^f for all individual SUs and P_{ij} for all pairs of SUs). To do this, the SBS first needs to determine the state of the channel by aggregating the sensing readings of different SUs.

However, when multiple PUs are present in the range of one SBS, it becomes challenging to determine the state of the channel since the aggregation process has to take into account which SUs are in the range of the same PU. For the example network in Figure 1, if primary user 1 is ON while primary user 2 is OFF, only 2 SUs that are in the range of π_1 would report the channel to be busy while all other SUs would report the channel to be free. These sensing results would seem contradictory especially if all SUs have very high accuracy. In this subsection, we first explain how the SBS can determine the state of the individual PUs and then using the result, how it updates the value of the SU-specific parameters. The following discussion is also restricted to a single channel (c_k). In practical deployment, the SBS will repeat the following computation for every channel in $\mathcal{M}$.

1) Determining PU State and Channel State: To determine the state of a primary user π_j, the SBS aggregates the sensing results from all SUs that scanned c_k in the sensing assignment

S in the last round. Let S^{ON} and S^{OFF} be the set of those node-channel pairs that scanned channel c_k and who reported the channel to be *busy* and *not busy*, respectively:

$$S^{ON} = \{(n_i, c_k) : (n_i, c_k) \in S \wedge x_i = 1\}$$
$$S^{OFF} = \{(n_i, c_k) : (n_i, c_k) \in S \wedge x_i = 0\}$$

Using the previous notation, recall that $\Omega(S^{ON}, k, j)$ and $\Omega(S^{OFF}, k, j)$ denote the aggregated accuracy of the sensing assignments S^{ON} and S^{OFF} in estimating the state of π_j, respectively. Then, the SBS determines that π_j is ON *iff* aggregated accuracy of the sensing assignment S^{ON} is not less than the aggregated accuracy of the sensing assignment S^{OFF}:

$$\pi_j \text{ is ON } \iff \Omega(S^{ON}, k, j) \geq \Omega(S^{OFF}, k, j) \quad (5)$$

Finally, the SBS determines channel c_k to be *busy iff* there exists at least one PU operating on c_k whose state is ON. Recall that the computation of $\Omega(S^{ON}, k, j)$ and $\Omega(S^{OFF}, k, j)$ takes into consideration the probability of two SUs being in the range of same PU. Therefore, here also when determining the state of π_j, (5) gives higher weight to the readings of SUs that are estimated to be in the range of π_j. By doing this, *we are able to account for the presence of multiple PUs in the same coverage area.*

2) Updating SU specific parameters: We assume that average P^d and P^f of SUs over all the SUs is known beforehand. A new SU that joins the system is assigned the average P^d and P^f. However, with time, DISCERN will arrive at a precise estimation of both these probabilities for all SUs using exponential moving average (EMA) as follows: At the end of the round, once the state of all the PUs is determined, then, for each SU that participated in sensing in the last round, we use EMA to update its P^d and P^f (similar to Section IV-A). Specifically, if after aggregating the readings, π_i was found to be ON while n_i sensed the channel to be empty, then $P_i^d(r, r)$ is set to 0, otherwise it is set to 1. Similarly, P_i^f and P_{ij} are also updated using EMA (See Section IV-A).

E. Discussion

1) Overheads: DISCERN has 3 types of overheads: (i) Memory overhead: In our algorithm, the SBS maintains P^d, P^f for every SU associated with it, and P_{ij} for all pairs of SUs associated with it. Thus, the total information that needs to be maintained at SBS is $O(|\mathcal{N}|^2)$. (ii) Communication overhead: DISCERN does not increase the communication overhead of the system as it runs only at SBS. Any messages that are transmitted from SBS to SU (message requesting SU to scan a particular channel) or from SU to SBS (message reporting the scanning result) would nevertheless need to be transmitted in any other system that utilizes cooperative sensing. (iii) Computation overhead: DISCERN has a time complexity of $O(|\mathcal{N}|^2|\mathcal{M}|^2)$.

2) Selecting the ρ: The value of ρ determines the number of scans that can be performed by an SU in a single round. The value of ρ has to be carefully selected as a higher value may increase the overhead on SBS as well as reduce the throughput of the channel due to increase in the amount of results reported by SUs to SBS. In practice, the value of ρ can be adjusted by the SBS depending on the desired sensing accuracy. If the current value of ρ does not provide enough sensing accuracy, the SBS can increase the ρ until the desired accuracy is achieved.

V. Experiments and Simulations Results

In this section, we will first describe our experiment results to show the correctness of P_{ij}. Next, using C++ simulations, we evaluate the performance of DISCERN for a system with multiple channels, PUs and SUs.

A. Correctness of P_{ij}

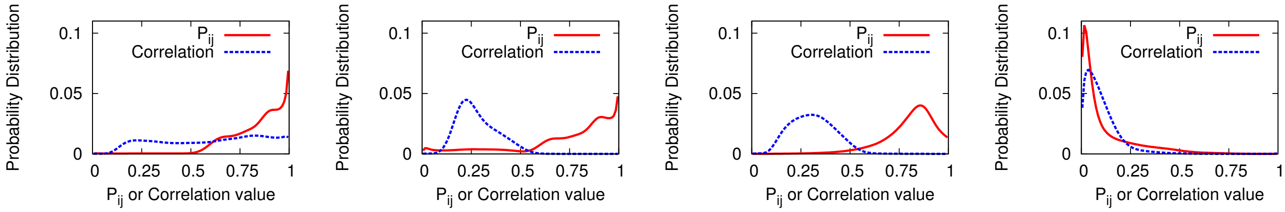
In this section, using experiments, we evaluate the ability of P_{ij} (DISCERN) in estimating if two SUs are in the range of same PU and compare it with correlation [7]. For this, we collected data on 40 channels (in the range 300MHz - 900 MHz) using two USRP radios [22] that were placed at different locations. Figure 2 shows the probability distribution of P_{ij} (DISCERN) and correlation [7] for the four cases: (i) When both SUs are adjacent to each other (both on roof of a 8-floor building); (ii) When one SU is in the basement while the other is on the roof of the 8-floor building; (iii) When both SUs are adjacent to each other (both in the basement of the 8-floor building); and, (iv) When one SU is on the roof of a 8-floor building while the other is in an open parking lot at a distance of 80 miles. The close location of the SUs in the first three settings ensures that both the SUs are in range of the same set of PUs while the last setting ensures that the two SUs are in range of different set of PUs[1].

For case (i), when both SUs are adjacent to each other, correlation between the readings of two nodes is typically high but it could still be as low as 0.2. P_{ij} on the other hand is always above 0.55. For cases (ii) and (iii), the value of correlation becomes very low while P_{ij} is still high. For case (iv), where the two SUs are in range of different PUs, typically P_{ij} is smaller than the value of correlation. In summary, Figure 2 indicates that P_{ij} is low for SUs that are in range of different PUs while it is typically high for SUs that are in range of the same PU even if SUs have different sensing accuracy. On the other hand, the value of correlation between the readings of two SUs can be low even if SUs are in the range of the same PU. Thus, P_{ij} is a better metric than correlation for identifying if two SUs are sensing the same PU or not. After combining the data from four different cases, we computed the correctness of both correlation and the P_{ij} metric in determining if two SUs are in range of the same PU or not. We observed that correlation with optimal threshold correctly classified the SUs in 69% cases while P_{ij} metric in 95% cases. Thus, P_{ij} improves the accuracy by over 25%.

B. Simulation Setup and Trace Collection

We assume that the *center* of the simulation field is the downtown of our city (Columbus, OH, USA). The SBS was located at the center and 300 SUs were randomly deployed around it in a circular field of 20 miles. We assumed that there are a total of 10 channels (in the range 300MHz - 900 MHz) that the SUs sense and can communicate on. Then, we set up 40 PUs on these 10 channels within a radial distance of 20 miles from the center. The locations of the PUs were

[1]Verified using the FCC database.



(a) Case (i) (b) Case (ii) (c) Case (iii) (d) Case (iv)

Fig. 2. Probability distribution of P_{ij} (DISCERN) and correlation for 4 different cases

established from the FCC database [23]. Each channel had an average of 4 PUs. Capturing the ON/OFF state of all the PUs using USRP radios [22] was challenging because of diverse geographical locations of the PUs. We solved this problem by first capturing the state of two different PUs (located 30 miles from each other) on each channel for multiple hours using USRP radios and then we divided the collected data into contiguous chunks where each chunk holds ON/OFF data collected from one of the two PUs over a continuous time period. Next, for every PU in the simulation operating on that channel, we emulated its ON/OFF status by using one randomly selected contiguous chunk of data. The power level of the PUs was also obtained from the FCC database [23]. The power level of the signal transmitted from PU will decrease with distance due to path loss and shadowing. The path loss was modeled using the following equation [24]:

$$P_{rx}(in\ dB) = P_{tx}(in\ dB) - 20\log_2\left(\frac{4\pi d}{\lambda}\right)$$

where P_{tx} is the transmission power level, d is the distance (in m.) between PU and the SU, λ is the wavelength (in m.) of the channel and P_{rx} is the received power level at the SU. The shadowing effect was modeled with number of obstacles varying between 0 and 5 [3]. Multipath fading was modeled using Jakes model. In addition to that, we had 4 more channels on which microphones operated. For this, on each of these four channels, we added an average of 4 PUs with each PU having a range of 2 miles. Each of these PUs was ON and OFF for an exponentially distributed time period with average of 5 minutes. To model the behavior of mobile microphones, 2 of these PUs moved randomly within a range of 0.25 mile from their center when ON with an average speed of 3MPH.

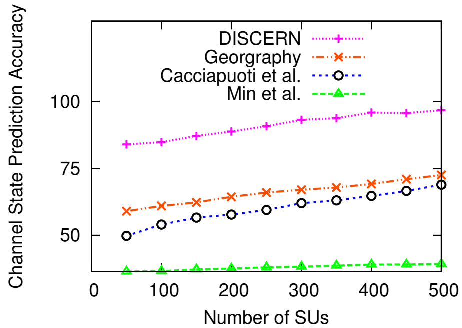
The mobility of the SUs was modeled from the traces collected in [25]. The default value of ρ was set to be 3 times the number of nodes in the network. Value of EMA coefficient (β) was set to 0.9. Upon scanning, if the SU observes the received power level to be above -90dB, only then it concludes that the channel is busy. The scanning result is transmitted by the SU to the SBS that uses the received results to update P_i^d, P_i^f and P_{ij} for all SUs (or pairs of SUs) as discussed in Section IV-D2. The duration of the round was set to 30 seconds. For this duration, we can safely assume that ON/OFF state of the PUs will not change often over a round [21]. In each simulation, we ran DISCERN for 15 rounds before collecting any data. This was done to ensure that the calculated values of P^d and P^f are stabilized before any measurement is done. For the same set, the simulations were performed 20 times in order to obtain accurate results.

1) Evaluating Ω-metric and DISCERN: For the purpose of comparison, we also implemented the following algorithms and compared their performance with DISCERN:

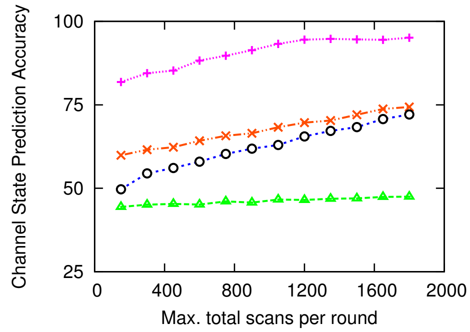
- **Geographical Select**: For each channel, at each step, this algorithm selects that SU for sensing which has the maximum distance from the already selected nodes. This can help in getting more *spatially diverse* readings. However, if obstacles are present between SUs and the PU, then the performance of this algorithm may suffer as discussed in Section I.
- **Min et al. [6]** selects nodes with the highest received signal strength (RSS) of the PU signal. This helps the algorithm in selecting nodes that are geographically close to the PU and are not located behind obstacles. However, as shown in Section I, this may lead to selection of SUs that are associated with only a subset of PUs.
- **Cacciapuoti et al. [7]** selects nodes that have minimum correlation with each other. Such an algorithm selects those SUs for scanning that are behind obstacles as their correlation with other SUs is low.

All the three baseline algorithms allocate equal number of sensing tasks to all the channels. At the end of each round, the SBS estimates the state of the $\mathcal{M}$ channels by aggregating the scan results. Next, we study the percentage accuracy of this channel state estimation for the four algorithms under different scenarios. For the sake of fairness, to estimate the channel state based on scan results, all the four algorithms used (5).

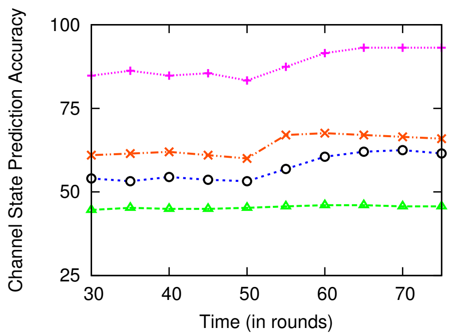
- **Variation with SU density**: As the number of SUs in the field increases, the number of nodes with high accuracy increases. This increases the estimation accuracy for all the four algorithms as shown in Figure 3(a). We observed that on average, compared to the baseline algorithms, DISCERN improves the accuracy by at least 30% (Geographical Select), 130% (Min et al. [6]) and 40% (Cacciapuoti et al. [7]). The improved sensing accuracy indicates that Ω-metric faithfully characterizes the sensing accuracy of a given assignment.
- **Variation with Max. scans allowed**: For every round, there is a limit of ρ on the maximum number of scans that can be performed. With increase in this limit, the estimation accuracy is expected to increase (Figure 3(b)).
- **Variation with time**: We also evaluated the performance of DISCERN with time (Figure 3(c)). For this,



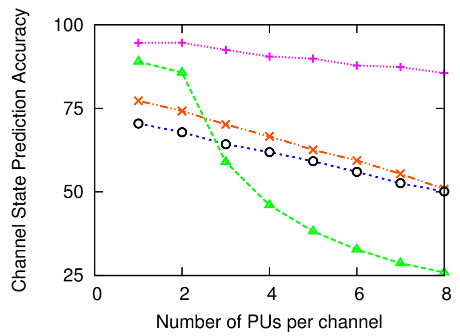
(a) With variation in number of SUs



(b) With variation in number of total scans allowed per round per SU



(c) With variation in time



(d) With variation in number of PUs per channel

Fig. 3. Accuracy of channel state prediction. Average values over multiple runs are plotted.

we initially deployed only 100 SUs in the system. Then after running the simulation for 50 rounds, we added 200 more SUs to the system. With more SUs in the system, for Geographical Select, the estimation accuracy increases immediately. However, for DISCERN, the increase is gradual as it assumes that initially all the nodes have average P_d and P_f. However, with time, it updates the P_d and P_f for the new SUs. This makes it possible for DISCERN to select SUs with better scanning accuracy, thereby increasing the accuracy considerably. Similarly, Cacciapuoti et al. [7] takes a few rounds to compute the correlation between SUs before it is able to select the SUs for sensing.

- **Variation in Number of PUs per channel**: We vary the number of PUs per channel. As the number of PUs increases, performance of Min et al. [6] degrades since it selects SUs with high RSSI even if all of them are in the range of same PU. The relative accuracies of the other two baseline algorithms do not change significantly since both these algorithms are independent of the presence of multiple PUs in the network.

It is possible that different SUs may scan different number of channels. To study that, we plotted the CDF of number of channels scanned by the SUs. Figure 4 shows that for all algorithms most of the SUs are scanning between 2 to 4 channels. This is expected as the scanning accuracy of an SU depends on its relative location with respect to the PU. Thus, for DISCERN and Min et al. [6], every SU gets to scan only those channels on which its accuracy is high.

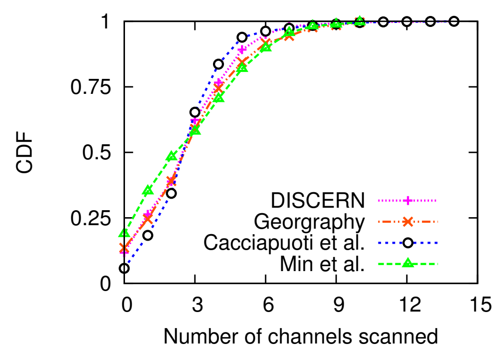


Fig. 4. CDF of number of channels scanned by each SU

Cacciapuoti et al. [7] may also select SUs with low accuracy which helps in balancing the sensing tasks while Geographical Select randomly picks SUs based on location, resulting in balancing of sensing tasks.

2) Correction of channel state determination: Next, we evaluate the correctness of our channel state determination algorithm (5) and compare it with Neyman Pearson (NP) Test [13], [12], [26]. NP test evaluates likelihood of two hypothesis say $\mathcal{H}_0$ (e.g., PU is OFF) and $\mathcal{H}_1$ (PU is ON) and chooses $\mathcal{H}_0$ over $\mathcal{H}_1$ if the ratio of likelihood of $\mathcal{H}_0$ over likelihood of $\mathcal{H}_1$ is above a certain threshold. For a fair evaluation, we assumed that NP test as well as DISCERN are both aware of the exact values of P^d and P^f for all SUs. Figure 5 shows the results. We show the results from two different variations of NP test: (i) With the optimal threshold (or ratio) that maximizes accuracy

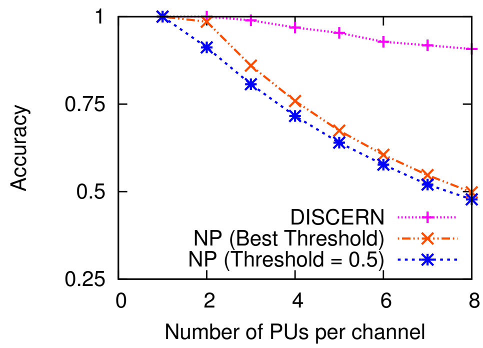


Fig. 5. Accuracy of Channel State Estimation with varying number of PUs operating on the same channel.

of channel state determination; and, (ii) With the threshold value fixed at 0.5. Observe that the optimal threshold is hard to determine in practical networks as P^d and P^f of SUs may not be known beforehand. NP Test works well only when there is only one PU in the network since it aggregates readings of all SUs even if they are in range of different PUs. On the other hand, DISCERN determines probability of SUs being in the interference range of the same PU and based on that, during aggregation, it gives higher weight to readings of SUs that are in the range of the same PU, resulting in more accurate channel state prediction.

VI. Conclusions and Future Work

Cooperative scanning can be used to increase the sensing accuracy of cognitive radios. However, due to the unknown locations of SUs, PUs and obstacles, it is not apparent which nodes should be selected for scanning which channels. In this paper, we presented a novel knowledge based mechanism that determines the probability that two SUs are in the range of the same PU. Using this knowledge based method, we defined a metric (Ω) that captures the accuracy of a given sensing assignment. The metric takes into account the differences in probability of detection and probability of false positive among nodes, presence of multiple PUs on the same channel as well as the limit on the amount of resources allocated for scanning. Using the metric, we propose an algorithm for computing the best sensing assignment. Experiment results show that our knowledge based mechanism improves the accuracy of determining if two SUs are in the range of the same PU by over 25%. Simulation results show that our algorithm improves the accuracy of channel state estimation by at least 30% when compared to other algorithms. In this paper, we restricted our focus to only one SBS and its associated clients. However, multiple SBSs with overlapping transmission range may be present in the network. In such scenarios, cooperation between SBSs can help in reducing the time spent in scanning by individual SUs.

VII. Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant CNS-1161490.

References

[1] K. Harrison, S. Mishra, and A. Sahai, "How Much White-Space Capacity is There?" in *Proc. of IEEE DySPAN*, 2010.

[2] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White Space Networking With Wi-fi Like Connectivity," in *Proc. of ACM SIGCOMM*, 2009.

[3] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative Sensing Among Cognitive Radios," in *Proc. of IEEE ICC*, 2006.

[4] M. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky, "Practical Metropolitan-Scale Positioning for GSM Phones," in *Proc. of UbiComp 2006*.

[5] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative Spectrum Sensing in Cognitive Radio Networks: A Survey," *Elsevier Physical Communication*, 2010.

[6] A. W. Min and K. G. Shin, "An Optimal Sensing Framework Based on Spatial RSS-profile in Cognitive Radio Networks," in *Proc. of IEEE SECON*, 2009.

[7] A. S. Cacciapuoti, I. F. Akyildiz, and L. Paura, "Correlation-Aware User Selection for Cooperative Spectrum Sensing in Cognitive Radio Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 297–306, 2012.

[8] *FCC Second Memorandum Opinion and Order, ET Docket No. 10-174*, Federal Communications Commission, Sep. 2010.

[9] *FCC Notice of Inquiry Spectrum Policy Task Force Report, ET Docket No. 10-198*, Federal Communications Commission, Nov. 2010.

[10] Z. Quan, S. Cui, and A. Sayed, "Optimal Linear Cooperation for Spectrum Sensing in Cognitive Radio Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 28–40, 2008.

[11] W. Zhang, R. K. Mallik, B. Letaief *et al.*, "Cooperative Spectrum Sensing Optimization in Cognitive Radio Networks," in *Proc. of IEEE ICC*, 2008.

[12] H. Li, "Learning the Spectrum via Collaborative Filtering in Cognitive Radio Networks," in *Proc. of IEEE DySPAN*, 2010.

[13] E. Peh and Y. C. Liang, "Optimization for Cooperative Sensing in Cognitive Radio Networks," in *Proc. of IEEE WCNC*, 2007.

[14] H. Kim and K. G. Shin, "In-band Spectrum Sensing in Cognitive Radio Networks: Energy Detection or Feature Detection?" in *Proc. of ACM MOBICOM*, 2008.

[15] S. Li, Z. Zheng, E. Ekici, and N. B. Shroff, "Maximizing System Throughput by Cooperative Sensing in Cognitive Radio Networks," in *Proc. of IEEE INFOCOM*, 2012.

[16] Y. Sun, H. Hu, F. Liu, H. Yi, and X. Wang, "Selection of Sensing Nodes in Cognitive Radio System Based on Correlation of Sensing Information," in *Proc. of IEEE WiCOM*, 2008.

[17] J. Liu, X. Zhang, R. Zheng, Q. Pan, and D. Yang, "Novel Cooperative Schemes on Spectrum Sensing in Multi-Primary-User Cognitive Radio Network," in *Proc. of VTC 2010*.

[18] J. Hwang, T. He, and Y. Kim, "Exploring In-Situ Sensing Irregularity in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 4, pp. 547–561, 2010.

[19] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[20] T. Bansal, B. Chen, and P. Sinha, "DISCERN: Cooperative Whitespace Scanning in Practical Environments," Tech. Rep., http://www.cse.ohio-state.edu/~bansal/DISCERN_13_tr.pdf.

[21] D. Chen, S. Yin, Q. Zhang, M. Liu, and S. Li, "Mining Spectrum Usage Data: A Large-Scale Spectrum Measurement Study," in *Proc. of ACM MOBICOM*, 2009.

[22] E. Research, "Ettus Research," http://www.ettus.com/.

[23] FCC, "License Search," http://wireless2.fcc.gov/UlsApp/UlsSearch/searchGeographic.jsp.

[24] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.

[25] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong, "CRAWDAD trace set ncsu/mobilitymodels/gps (v. 2009-07-23)," Downloaded from http://crawdad.cs.dartmouth.edu/ncsu/mobilitymodels/GPS, Jul. 2009.

[26] Q. Zou, S. Zheng, and A. H. Sayed, "Cooperative Sensing via Sequential Detection," *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6266–6283, 2010.