Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 1450418, 16 pages
https://doi.org/10.1155/2018/1450418

Research Article

Sensing Device Management for History-Based Spectrum Sharing in Cognitive Radio Networks

Mohammed Hawa[1], **Fahed Jubair**[2], **Raed Al-Zubi**[1], **and Ramzi Saifan**[2]

[1] *Electrical Engineering Department, The University of Jordan, Amman 11942, Jordan*
[2] *Computer Engineering Department, The University of Jordan, Amman 11942, Jordan*

Correspondence should be addressed to Mohammed Hawa; hawa@ju.edu.jo

Received 23 February 2018; Revised 31 August 2018; Accepted 8 October 2018; Published 1 November 2018

Academic Editor: Juan F. Valenzuela-Valdés

A novel approach to managing a fully distributed cognitive radio network (CRN) is presented. This approach builds on the concept of history-based spectrum access, in which cognitive base stations (BSs) independently estimate the system load using history records and adaptively swap their occupied spectrum bands to ensure allocation fairness and high overall throughput. In addition, cognitive BSs monitor primary user (PU) behavior in order to avoid interfering with active PUs. In this work, we address two issues that afflict history-based access: the first is the high cost of the sensing devices needed at each cognitive BS to be able to independently draw conclusions about the status of the CRN and the second is the unreliability inherent in practical sensing hardware (such as energy detectors). Simulation results show that the proposed technique manages to solve the two abovementioned issues without any noticeable drop in performance and without sacrificing the distributed nature of the protocol.

1. Introduction

Cognitive radio achieved prominence in recent years because it promises to solve the radio spectrum underutilization issue. In *cognitive radio networks* (CRNs) two classes of users exist: The first is *primary users* (PUs) who are licensed to use the spectrum but many times leave the spectrum underutilized, such as the case of terrestrial TV broadcasters in rural and suburban areas [1]. The second class of users is *secondary users* (SUs), who are cognitive users that can sense the spectrum bands left unused by PUs and can opportunistically utilize this spectrum for their transmission needs [2, 3].

A large body of work has been dedicated to researching the various aspects of CRN operations, such as spectrum sensing to identify empty bands [4–11], channel allocation among SUs [1, 12, 13], establishment and maintenance of a *common control channel* (CCC) to allow persistent communications between different SUs [3, 14], and spectrum handoff and mobility. A summary of relevant research is presented in the next section.

A cognitive secondary user or a cognitive base station (BS) typically operates in two stages. In the first stage a BS must perform spectrum sensing, in which the cognitive BS listens to the received signal on one or more frequency bands that the BS wants to transmit on. The purpose of this sensing is to determine whether a PU is currently transmitting on this frequency or not. If a PU exists, the cognitive BS avoids this band and attempts to sense another. If a PU is not detected, the cognitive BS goes into the next phase (called channel allocation), which decides whether the BS should access the band for itself or leave it for other neighboring cognitive BSs, for fair usage of the spectrum.

The conventional wisdom for running these two processes in CRNs was to divide the time into frames (i.e., time slots) and then divide each frame into two periods [3]: In the first period, called the *quiet period* (QP), all cognitive BSs refrain from transmitting and instead sense the bands to listen to any active PUs (who keep transmitting as they are unaware of the CRN existence). The longer the QP, the more accurate the sensing results, because more samples are collected for the received signal by the sensing devices available at the cognitive BSs [6].

The second period of the frame is called the *transmission opportunity* (TxOp) (see Figure 1(a)), in which the cognitive

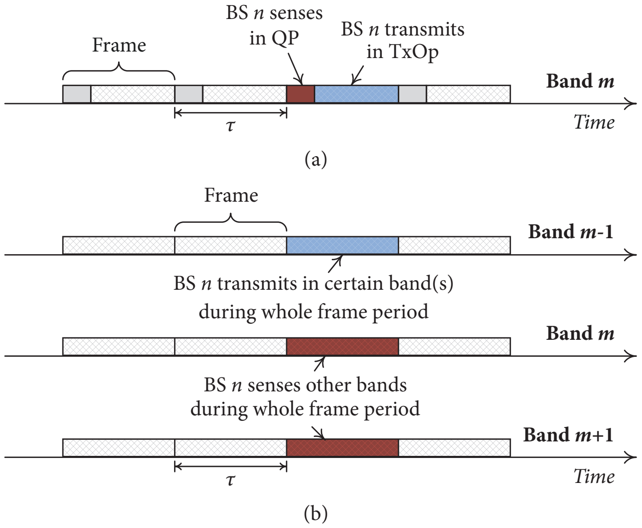


Figure 1: Illustration of the (a) conventional method of sensing and accessing spectrum (using a QP followed by a TxOp) and (b) the proposed procedure for simultaneous sensing/transmission over multiple spectrum bands.

BSs start accessing the spectrum holes left by PUs. In a fully distributed CRN, cognitive BSs do not coordinate with each other when accessing the spectrum during TxOps. This might lead to contention across spectrum bands. The traditional technique for resolving such conflicts (when they arise) is to use contention resolution techniques, such as CSMA with exponential backoff [11, 15], which unfortunately exhibits poor performance, in terms of both throughput and delay, when increasing the number of BSs [16].

A departure from this conventional CRN access technique, in which QPs are followed by TxOps, is the work presented in [17], called HOP-M, where a different methodology for sensing and accessing the spectrum was introduced. This new approach, designed for distributed CRNs, requires cognitive BSs to continually swap access to spectrum bands amongst themselves (at the end of what is called *maintenance periods*), rather than aggressively contending for such bands at the beginning of every TxOp. This significantly reduces contention between competing cognitive BSs, as each BS regularly and voluntarily gives up some of its acquired spectrum bands to the benefit of others, while contention only happens at the time of the swap.

The above method also mitigates the need for QPs at the beginning of every time slot, since the cognitive BSs can keep transmitting until the time of the band swap. If a PU starts transmitting in the band, it is easily detected by the cognitive BS because the ensuing interference will prevent proper *acknowledgements* (ACKs) from arriving at the BS, which causes the cognitive BS to stop transmitting very quickly to honor the PU's lawful right to the spectrum. However, other cognitive BSs do not attempt to interfere with the BS holding the band until the time of the swap. The elimination of QPs within each frame increases the channel utilization and greatly improves CRN performance, without severely affecting PU operations. Our employment of the above novel idea is explained thoroughly in Section 3.

Though this new HOP-M swapping approach has excellent performance, it also has its own limitations. The main one is that it requires each cognitive BS to have a large number of sensing devices at its disposal to be able to sense all spectrum bands in the CRN, specifically the bands the BS is not occupying at the moment. This allows the cognitive BS to draw a clear picture about the system around it without the need to communicate with other cognitive BSs or any dedicated central authority. The need for this many sensing devices represents a large cost overhead that needs to be addressed.

In addition, the work in [17] assumes highly advanced and sensitive sensors (such as cyclostationarity based detectors, feature detectors, or eigenvalue-based sensors) that can perform almost perfectly when sensing the spectrum [1, 6], even under noisy conditions. This increases the cost of the system even further and is not preferred compared to the more cost-conscious sensing devices. Many CRNs might want to employ the simpler *energy detectors* (EDs), which are cheaper, albeit not very accurate in their sensing abilities [1, 3].

In this work, we attempt to address the above two limitations of the HOP-M technique. Hence, our contribution is twofold: First, we modify several aspects of the HOP-M algorithm to allow us to operate with a far smaller number of sensing devices per cognitive BS, yet without any noticeable drop in performance. The improvements we introduce pertain to how this limited set of sensing devices can be used to obtain almost a full picture of the system to help run our proposed technique (named *History-oriented Procedure for spectrum Scheduling and Sensing*, or HoPSS for short). This contribution can significantly reduce hardware costs of the sensing devices required for the CRN and, hence, make the idea of swapping spectrum bands among cognitive BSs more practical.

The second contribution of this paper involves new extensions to the original protocol so that HoPSS can operate properly even under unreliable sensing conditions, which are typical for less sensitive (but cheaper) ED sensors. We show that the performance penalty for using more-affordable (but less-reliable) sensors is minimized for the HoPSS algorithm, compared to the HOP-M technique. In short, the new HoPSS technique can achieve excellent performance in terms of throughput, stability, and quick reaction to PU behavior, while using cheap ED sensors.

The rest of this paper is organized as follows: Section 2 offers some background research related to this work. Section 3 describes the problem statement and details the proposed spectrum sensing/allocation procedure, which we call HoPSS. After that, Section 4 explains the simulated network setup and the metrics used to evaluate the performance of our technique. Finally, a discussion of the simulation results is presented in Section 5, while Section 6 concludes with final comments.

2. Background

Designing an efficient, fair, and robust channel allocation mechanism is a major challenge for CRNs. This is why there is a wide body of research addressing this issue. Existing mechanisms for channel allocation can be generally classified into centralized, distributed, and hybrid techniques.

Centralized policies for channel allocation can exhibit better performance compared to distributed policies, which are typically based on random selection executed by each cognitive BS without communicating with the rest of the network. Unfortunately, uncoordinated access to spectrum bands in distributed CRNs can result in excessive collisions, thus increasing delay and lowering throughput [15, 16]. Centralized mechanisms can provide a much higher throughput and better levels of fairness (compared to distributed ones), due to the help of a central coordinator.

Centralized techniques for spectrum access typically involve running optimization algorithms to reach the desired channel allocation strategy. Different approaches have been proposed to solve such an optimization problem, such as particle swarm optimization [18, 19], ant colony optimization [20], genetic and evolutionary algorithms [12, 21, 22], fuzzy logic [23], and neural networks [24]. However, centralized optimization techniques suffer from some disadvantages, such as a single point-of-failure, extra implementation complexity, sensitivity to node or link failure, and the need to preestablish a proper CCC between the cognitive BSs and the controller.

Distributed CRNs, on the other hand, have also garnered widespread interest due to their advantages. First, there are no bottlenecks and no single point-of-failure. In addition, there is no need to setup and maintain a CCC within the unpredictable spectrum of the CRN. This greatly simplifies the design and implementation of the system [3, 14]. Also, distributed systems have inherent scalability and robustness that cannot be matched in a centralized network [1, 25].

A popular class of distributed CRNs are the ones based on game-theoretic models [26–30]. In such models, each cognitive BS acts as a player in a game and hence follows a strategy proper to that game to decide independently on channel assignment. The cognitive BS can change its strategy based on observations of spectrum usage. The goal of the game is typically to reach a *Nash equilibrium*, and one usually defines a utility function to maximize certain performance metrics at equilibrium, such as throughput or fairness.

3. HoPSS Spectrum Management

3.1. System Model. We consider a set $\mathbb{M}$ of spectrum bands numbered from 1 to $M = |\mathbb{M}|$, where $|\mathbb{M}|$ is the cardinality of set $\mathbb{M}$. A set $\mathbb{N}$ of cognitive BSs in a CRN attempts to access such spectrum bands in a fully distributed manner. These BSs are numbered 1 to $N = |\mathbb{N}|$. Each BS $n \in \mathbb{N}$ has a *spectrum demand* $\widehat{S}_n$, which represents the number of spectrum bands this BS would like to occupy successfully in order to serve the various SUs attached to this BS.

As illustrated in Figure 1, time is divided into fixed-length frames called *time slots*, each is equal to τ seconds. During each time slot $t \in [1, T]$, a cognitive BS typically *occupies* a subset of the $\mathbb{M}$ frequency bands. We denote the *set* of spectrum bands successfully occupied by BS n during a particular time slot t (i.e., without suffering from contention due to other BSs and without interference from an active PU in that band) using the symbol $\mathbb{S}_n(t)$, where the number of such *successful bands* is $s_n(t) = |\mathbb{S}_n(t)|$. On the other hand, the set of bands occupied by BS n but suffering from contention from other BSs or interference from a PU (during time slot t) is represented by the set $\mathbb{C}_n(t)$, where the number of such *collision bands* is $c_n(t) = |\mathbb{C}_n(t)|$. The whole set of occupied bands (by BS n) during time slot t, whether successful or under collision, is denoted by $\mathbb{O}_n(t) = \mathbb{S}_n(t) \cup \mathbb{C}_n(t)$. It is mentioned shortly that BS n can determine the status of any of the occupied bands $m \in \mathbb{O}_n(t)$ via the correct reception of an acknowledgement (ACK) or lack thereof. Hence, using a dedicated sensing device (such as an ED) to sense these occupied bands will not be necessary.

To reduce the overall system cost, each BS n is assigned a limited set of dedicated sensing devices (detectors), named $\mathbb{D}_n$, where the number of such devices per BS is $D_n = |\mathbb{D}_n| \leq M$. For example, each of these sensing devices can be an energy detector (ED) that can be assigned to sense one of the M available spectrum bands. BS n will utilize such dedicated devices during any time slot t to sense a maximum of D_n bands that are not currently being occupied by the BS (i.e., out-of-band sensing). This set of sensed bands is denoted by $\mathbb{V}_n(t) \subset \mathbb{M}$. BS n maintains the condition $\mathbb{V}_n(t) \cap \mathbb{O}_n(t) = \emptyset$.

We note that the number of bands being sensed via out-of-band sensing is $v_n(t) = |\mathbb{V}_n(t)| \leq D_n$ because in theory all M bands can be occupied by BS n at a certain time slot, which means out-of-band sensors can be deactivated in this case. However, in reality the BS spectrum demand $\widehat{S}_n$ and, hence, the number of occupied bands by the BS $o_n(t) = |\mathbb{O}_n(t)|$ during any time slot t are both much smaller than the number of available spectrum bands M, which means that, typically, the number of bands $v_n(t)$ being sensed by out-of-band sensing devices is equal to the number of available detectors D_n.

Not only to this extent, but since the number of available detectors D_n is much smaller than M and due to the fact that BS n does not attempt to acquire all bands in the system, there will be bands that are neither occupied nor sensed by the BS. These are denoted by the set $\mathbb{U}_n(t)$, which means that their status is *unknown* to BS n during time slot t. Surely it is easy to see that, for any time slot t, we have $\mathbb{O}_n(t) \cup \mathbb{V}_n(t) \cup \mathbb{U}_n(t) = \mathbb{M}$. The HOP-M technique presented in [17] assumes that $D_n = M$, which means that $\mathbb{U}_n(t) = \emptyset$ for all times, thus simplifying the process of identifying the system status as a whole. Our proposed HoPSS technique, on the other hand, does not have this luxury and has to be clever in utilizing the limited set of D_n detectors to glean a good picture of the system status without needing extra help from other BSs or a central coordinator.

3.2. Sensing and History. The action of BS n in sensing the $\mathbb{V}_n(t)$ spectrum bands during time slot t via its available sensing devices is very similar to how traditional CRNs perform sensing in the QP. However, instead of sensing during a QP, we assume that sensing is performed for the whole duration τ of the time slot. This increases the number of samples available to the sensing device compared to the shorter QP, which improves detection performance (as seen shortly).

We also assume in HoPSS that simple energy detectors (EDs) are used as sensing devices. Such detectors sample the signal $y(t)$ received at the antenna of the BS sensing device when tuned to a particular band. The BS needs to determine which of the following two possibilities (hypotheses) is correct:

$$\begin{aligned}\mathscr{H}_1&: y(t) = x(t) + w(t)\\ \mathscr{H}_0&: y(t) = w(t),\end{aligned} \tag{1}$$

where $w(t)$ is an additive white Gaussian noise (AWGN) signal and $x(t)$ is the signal transmitted by a nearby PU or another BS. A typical energy detector (ED) would obtain K samples of the received signal $y(t)$ and then use these samples to calculate a test statistic to estimate the received signal energy as follows:

$$\Theta(K) = \frac{1}{K}\sum_{k=1}^{K} |y(k)|^2. \tag{2}$$

This statistic $\Theta(K)$ is then compared to a threshold θ (see below) to decide on the spectrum status. For $\Theta(K) > \theta$, the band is considered busy; otherwise it is considered empty. However, there is a possibility that the signal $x(t)$ is weakened because of shadowing or fading in the wireless channel. In addition, there is a possibility that $w(t)$ spikes temporarily due to an extra burst of noise. In such cases, the ED can make a wrong determination of the status of the band. Typically, the performance of the sensor device is expressed in terms of two probabilities: *probability of false alarm*, denoted by P_{fa}, and *probability of misdetection*, represented by P_{md} [5].

The probability of misdetection is when a PU (or another BS) signal is truly present but cannot be seen due to shadowing or deep fades. This probability is given by

$$P_{md} = P\left(\Theta(K) \le \theta \mid \mathscr{H}_1\right). \tag{3}$$

On the other hand, the probability of false alarm is the probability that the detector incorrectly infers that the PU (or another BS) is present when it is actually not, probably due to excessive noise, and is defined by

$$P_{fa} = P\left(\Theta(K) > \theta \mid \mathscr{H}_0\right). \tag{4}$$

To understand the factors effecting the above performance metrics, we present a quick derivation of these probabilities under the following common assumptions (adopted by several researches [5, 9]). First, we assume that the samples of the received PU (or BS) signal $x(k)$ are independent and normally distributed with mean 0 and variance η_x^2. This variance represents the average power of the received signal. In addition, the samples of the additive white Gaussian noise $w(k)$ are independent and identically distributed (iid) with zero mean and variance η_w^2. The received signal and white noise are considered statistically independent.

Since the test statistic $\Theta(K)$ in (2) is the sum of the squares of a finite number K of independent normally distributed random variables, by definition we get a chi-squared distribution for $\Theta(K)$ as follows:

$$\Theta(K) \sim \begin{cases}\chi_K^2(\gamma), & \mathscr{H}_1\\ \chi_K^2, & \mathscr{H}_0,\end{cases} \tag{5}$$

where χ_K^2 and $\chi_K^2(\gamma)$ represent the central and noncentral chi-squared distributions, respectively, with K degrees of freedom (and noncentrality parameter of γ in the latter case). Here, $\gamma = \eta_x^2/\eta_w^2$ is the signal-to-noise ratio (SNR) of the received signal $y(t)$ as perceived by the energy detector.

When the threshold θ used for comparison is a single fixed-value threshold, the false alarm probability for the energy detector can be evaluated by finding the area under the tail of the $\mathscr{H}_0$ distribution above the threshold as follows [31]:

$$P_{fa} = \int_{\theta}^{\infty} \chi_K^2 dz = \frac{\Gamma(K,\theta/2)}{\Gamma(K)}, \tag{6}$$

where $\Gamma(.)$ and $\Gamma(.,.)$ denote the complete and incomplete gamma functions, respectively [7]. On the other hand, the probability of misdetection can be obtained from the $\mathscr{H}_1$ distribution as follows [31]:

$$P_{md} = \int_{-\infty}^{\theta} \chi_K^2(\gamma)\, dz = 1 - Q_M\left(\sqrt{2\gamma}, \sqrt{\theta}\right), \tag{7}$$

where $Q_M(.,.)$ is the generalized Marcum Q-function. Simpler equations for the above probabilities can be found when the number of samples K obtained by the sensor increases to a large enough value (which is the case in our proposed HoPSS technique). Under this condition, we can use the central limit theorem to asymptotically approximate the chi-squared distribution using the Gaussian distribution as [31]

$$\Theta(K) \sim \begin{cases}\mathscr{N}\left(\eta_w^2(1+\gamma), \dfrac{2\eta_w^4(1+2\gamma)}{K}\right), & \mathscr{H}_1\\ \mathscr{N}\left(\eta_w^2, 2\eta_w^4/K\right), & \mathscr{H}_0,\end{cases} \tag{8}$$

where $\mathscr{N}(\lambda, \eta^2)$ indicates the normal distribution with mean λ and variance η^2.

Therefore, using the Gaussian complementary function $Q(x) = 1/\sqrt{2\pi}\int_x^{\infty} \exp(-t^2/2)dt$, we can calculate the area under the tail of the normal distribution in (8). This gives the false alarm probability for the energy detector using a threshold θ by [9, 32]

$$P_{fa} = Q\left(\frac{\theta - \eta_w^2}{\eta_w^2\sqrt{2/K}}\right). \tag{9}$$

On the other hand, the probability of misdetection becomes

$$P_{md} = 1 - Q\left(\frac{\theta - \eta_w^2(1+\gamma)}{\eta_w^2\sqrt{2(1+2\gamma)/K}}\right). \tag{10}$$

Note that P_{fa} does not depend on the power of the PU signal; thus it remains the same if a fading channel (Rayleigh, Rician, Nakagami, etc.) is considered. On the other hand,

P_{md} can be easily affected by the received power from the PU, which in turn depends on the fading channel response [10, 11]. It is also clear that increasing the number of samples K obtained by the energy detector can reduce P_{fa}. In addition, increasing the SNR value γ of the PU at the receiver can reduce P_{md}.

Since the spectrum sensing performance depends on the chosen threshold θ, we need to determine this threshold to minimize sensing error [32]. In a Neyman–Pearson detector, the selected threshold θ is based solely on the distribution of the test statistic under $\mathscr{H}_0$ [3]. Hence, from (9) the detection threshold for ED is

$$\theta = \eta_w^2 \left(1 + Q^{-1}\left(P_{fa}\right)\sqrt{\frac{2}{K}}\right). \tag{11}$$

A major drawback of the above ED scheme is its poor performance under low SNR conditions, or when the noise level at the sensor is unknown a priori, a condition known as *noise uncertainty* [6, 32].

Several ideas were proposed to improve ED performance, including adaptive ED, in which the detection threshold θ is calculated from (11) based on an estimated noise power η_w^2. The sensor updates this estimate every calibration period [10] and keeps updating the threshold as the noise power changes in the environment. Another proposed method to enhance ED performance is the use of two detection thresholds [32], wherein if the test statistic $\Theta(K)$ is below a lower threshold θ_1, hypothesis $\mathscr{H}_0$ is declared. However when $\Theta(K)$ is between the lower threshold θ_1 and an upper threshold θ_2, the sensing device repeats its sensing for another round to improve the reliability of the result.

In addition to the above modifications to the ED, several other more advanced sensing devices were developed that can improve sensing performance by reducing P_{fa} and P_{md}, albeit with the disadvantage of higher hardware cost. These include matched filter detectors, waveform detectors, cyclostationary detectors, and eigenvalue-based detectors [6, 32].

Energy detection (ED) based schemes, however, are still preferred in cognitive radio spectrum sensing due to their computational efficiency [3]. Employing good detector hardware design and increasing the sensing interval so that more samples are gathered by the sensing device mean that for most (general) conditions P_{fa} and P_{md} are small enough probabilities [5]. This prevents underutilization of transmission possibilities due to false alarms and also prevents excessive interference on PUs and other BSs due to signal misdetection. It is worth mentioning that the proposed HoPSS technique works well with both perfect sensors and unreliable sensors.

We do not fail to mention that BSs in both the original HOP-M and the proposed HoPSS techniques do not stop transmission when occupying a spectrum band, as there is no QP defined within the frame (see Figure 1). Hence, a sensing device cannot determine whether the received signal in a certain band is due to a PU or another BS. This limitation will be resolved later within our proposed algorithm. However, the sensing device of BS n can render a decision $h_n^m(t)$ on the observed status of each spectrum band $m \in \mathbb{V}_n(t)$ as being either *empty*, denoted by $h_n^m(t) = \mathbb{e}$, which corresponds to hypothesis $\mathscr{H}_0$, or *busy* (due to another PU or another BS using the band during this time slot). The latter case is denoted by the status $h_n^m(t) = \mathbb{b}$ and corresponds to hypothesis $\mathscr{H}_1$.

Now we move our attention to the set of bands $\mathbb{O}_n(t)$ currently occupied by BS n. Instead of introducing a QP for such bands in which cognitive BSs suspend their transmission, each BS in our proposed technique transmits for the duration of the frame and pauses only at the end to wait for an ACK. If an ACK arrives, this means that no interference (from any source whether a PU or another BS) occurred during that transmission. In such case, BS n records that the band status was a *successful* acquisition of the band, denoted by $h_n^m(t) = \mathbb{s}$.

The other possibility is when an ACK does not arrive at BS n after a transmission, most probably due to an interference from an active PU or another BS who is contending for this band at the same time. In this scenario, BS n records that the band status was acquisition under *collision*, or $h_n^m(t) = \mathbb{c}$. Of course, this case can also happen if a burst of noise garbles the transmission of the BS or the corresponding ACK. In such case, BS n might think that the band was under contention, while truly it was under noisy conditions. We will denote the probability of this occurring by P_n. Due to the reliability of modern digital hardware against noise (especially when using adaptive modulation and forward error correction techniques), we can safely assume that this probability is quite small [33].

The reliance on the presence or absence of an ACK to indicate the presence of PUs is a reliable indicator. This is because when collisions persist for a few consecutive time slots, it is mostly likely due to an active PU, not a random burst of noise. Consequently, we design the HoPSS protocol to discourage the BSs from leaving immediately a band under nonpersistent collisions since a PU might not be active in that band.

Finally we mention that the status of every band within the set of bands $\mathbb{U}_n(t)$ (which cannot be sensed directly or indirectly by BS n) is recorded as $h_n^m(t) = \mathbb{u}$, indicating an *unknown* status.

To reiterate, the status of a particular spectrum band as seen by BS n during time slot t can be in one of five possibilities: $h_n^m(t) = \mathbb{e}, \mathbb{b}, \mathbb{s}, \mathbb{c}$, or $\mathbb{u}$. In HoPSS, this state is recorded in a moving history table maintained by BS n. This moving history is similar to how one would manage a moving average record. The record length of the history table is H_n time slots, where H_n can be selected to strike a balance between performance and memory requirements at the BS.

3.3. HoPSS Operations. The proposed HoPSS algorithm is an extension of the HOP-M technique and hence borrows some of its operations. However, these operations are modified to allow HoPSS to operate using a small number of sensing devices $D_n \leq M$ and under unreliable sensing conditions.

We assume that each cognitive BS n attempts to occupy a certain number of spectrum bands to satisfy its demand $\widehat{S}_n$. However, every number W_n of time slots (called the *maintenance window* for BS n), the BS leaves voluntarily one of its occupied bands for the benefit of other cognitive BSs. Other BSs contend for the band only after it is released, not while a BS (or PU for that matter) is using that band. Hence, contention among BSs is possible only at the instant of acquiring a new spectrum band and ends shortly thereafter. This significantly minimizes wasted spectrum due to contention.

HoPSS shares with HOP-M three decisions BS n has to undertake (albeit with several modifications) specifically (a) when to leave a successfully occupied $\mathbb{s}$ band, (b) when to leave an occupied band under collision $\mathbb{c}$, and (c) when to occupy (join) a new empty $\mathbb{e}$ band. Notice that BSs in HoPSS do not attempt to disturb busy $\mathbb{b}$ bands or occupy unknown $\mathbb{u}$ bands.

The reader is reminded that the older HOP-M technique does not have unknown $\mathbb{u}$ bands since it has a total of $D_n = M$ sensing devices available to sense all $\mathbb{M}$ spectrum bands. On the other hand, HoPSS has a smaller number of sensors $D_n \leq M$. Hence, we introduce in HoPSS two new operational procedures, namely, (i) when to let a sensing device $d_n \in \mathbb{D}_n$ stop sensing a particular band m and (ii) to which new band m' this sensing device should move after it has stopped sensing band m. These two decisions have to be executed carefully to allow a small number of sensing devices D_n to perform equally well as a large number of sensing devices, whether we have reliable (perfect) sensing hardware or cheaper (unreliable) energy detectors.

To assist the proposed HoPSS algorithm in its decisions, three counters are maintained by BS n using the help of the history table mentioned in the above subsection. The first of these counters is the number of time slots $\epsilon_n^m(t)$ in which band m was observed by BS n to be empty during its recorded history up to time slot t, which is calculated as follows:

$$\epsilon_n^m(t) = \sum_{t'=t-H_n+1}^{t} \left[h_n^m(t') = \mathbb{e}\right], \quad (12)$$

where the notation $[\psi]$ means 1 if the condition ψ is true and 0 otherwise. Notice that the band-empty count $\epsilon_n^m(t)$ is an integer that assumes the range $[0, H_n]$, where the high value of $\epsilon_n^m(t) = H_n$ is obtained if band m is consistently sensed to be empty by BS n; otherwise $\epsilon_n^m(t)$ assumes a smaller value.

The second counter we maintain is the number of history time slots $\sigma_n^m(t)$ in which band m was observed to be successfully occupied by BS n (up to time slot t), which is

$$\sigma_n^m(t) = \sum_{t'=t-H_n+1}^{t} \left[h_n^m(t') = \mathbb{s}\right]. \quad (13)$$

And, finally, the third counter needed is the number of history time slots $\mu_n^m(t)$ in which band m status was unknown to BS n. This is

$$\mu_n^m(t) = \sum_{t'=t-H_n+1}^{t} \left[h_n^m(t') = \mathbb{u}\right]. \quad (14)$$

The five procedures implemented by the proposed HoPSS technique and the way they make use of the above counters are explained in detail in the following subsections.

3.4. Leaving a Successful Band. To ensure fairness within the CRN, we require that each BS n abandons some of its acquired bands from time to time for the benefit of others. BS n only abandons one successfully occupied band at a time, and it can do that only at the end of its maintenance window. The width of such maintenance window starts at an initial value of $W_n = w$ time slots (or $W_n = w \times \tau$ seconds), but the BS can change the maintenance window length W_n later to adapt to system conditions.

To minimize the possibility of BSs continuously abandoning their bands and then acquiring new ones, which represents an unnecessary burden on the system, we require that BS n leaves its successfully occupied bands only when the load on the system is observed to be high, which is manifested by the unavailability of empty bands. To that end, BS n calculates a load estimate $\Lambda_n(t)$ at the end of its maintenance window as follows:

$$\Lambda_n(t) = \frac{M \times \sum_{m=1}^{M} \epsilon_n^m(t)}{M \times H_n - \sum_{m=1}^{M} \mu_n^m(t)}. \quad (15)$$

Obtaining a value $\Lambda_n(t) \geq 1$ indicates to BS n that its history records show that, out of all the bands it has sensed, on average one or more empty spectrum bands exist. In such case, BS n *does not* give up any of its successfully occupied bands because other cognitive BSs have one or more empty bands available for them. On the other hand, when $\Lambda_n(t) < 1$, then history indicates that the system is overloaded because the sensed bands are mostly busy. Hence, BS n regularly releases one of its spectrum bands to help other BSs.

To ensure fairness among BSs and to also minimize frequency switching overhead for the altruistic BS, a probabilistic coin flip is performed by BS n to decide to leave a band at the end of the current maintenance window or not, where the *leave probability* $L_n(t)$ is calculated as follows:

$$L_n(t) = \min\left(\frac{s_n(t)}{\widehat{S}_n}, 1\right). \quad (16)$$

Equation (16) is designed to let BSs that have satisfied their spectrum demand (i.e., $s_n(t) \approx \widehat{S}_n$) be more willing to give up one of their bands compared to those that did not satisfy their demand yet (i.e., $s_n(t) \ll \widehat{S}_n$). This ensures fairness in a fully distributed system.

When the coin flip forces BS n to leave a band by the end of the current maintenance window, the abandoned band is selected as the band with the minimum successful utilization count $\sigma_n^m(t)$. This helps in reducing frequency switching overhead seen by the BS. In addition, when BS n concedes one of its successful bands, it then elongates its maintenance window to

$$W_n = \left\lceil 1 + a_d \times \frac{s_n(t)}{\widehat{S}_n} \right\rceil \times w \ \ (\text{slots}). \quad (17)$$

This gives time for other BSs in the network to reciprocate by giving some of their successfully occupied bands before BS n has to give up another band. From (17) we can see that when the system is overloaded (i.e., when $s_n(t)/\widehat{S}_n \ll 1$) the maintenance window will be shortened to allow BSs that really need a band to obtain one. The control parameter

$a_d > 0$ and the initial maintenance window length w both allow us to obtain a balance between system throughput and stability, by minimizing swapping of spectrum bands, while at the same time forcing quick stabilization of the system after a disturbance (say after a PU is activated), where some BSs affected by the disturbance will need new spectrum bands [17].

Finally, we mention that if BS n decides *not* to give up any of its bands by the end of the current maintenance window (which can happen with probability $1 - L_n(t)$), the BS resets the next maintenance window length to the original value $W_n = w \times \tau$ seconds.

3.5. Leaving a Collision Band. When a band is released by the end of a maintenance window (or was originally empty), multiple cognitive BSs may attempt to occupy it at the same time, causing a collision. In addition, when a PU gets activated, the interference from the PU also shows up as a collision for the cognitive BS. To resolve these situations, BS n counts the number of time slots in which BS n sensed band m under collision, denoted by $i_n^m(t)$. BS n does not concede the band before a *minimum contention time* $\tilde{I}_n$ has passed. In other words, BS n does not give up the band if the collision count is $i_n^m(t) < \tilde{I}_n$. A benefit of not leaving a collision band immediately is that bursts of random noise can seem like a collision to a BS. Waiting for a few consecutive time slots can tell the BS if this was transient noise or a recently activated PU.

On the other hand, if $i_n^m(t) \geq \tilde{I}_n$, then BS n concedes this collision band probabilistically with a leaving probability of

$$R_n^m(t) = \min\left(a_i \times \frac{i_n^m(t)}{\hat{I}_n} + (1 - a_i) \times \frac{s_n(t)}{\hat{S}_n}, 1 \right), \qquad (18)$$

where $\hat{I}_n$ is the *contention time limit* for BS n. Both of the control parameters $a_i \in [0, 1]$ and $\hat{I}_n$ allow a tradeoff to be struck between minimizing the time wasted due to collisions (by leaving the contended band quickly) thus improving throughput and at the same time ensuring, on the other hand, that BSs that did not meet their demand just yet will compete harder for the contended band. Once a contention is resolved, the winning BS considers the occupied band successful and retains exclusive use of it until the end of its maintenance window, at which time the BS has to decide to keep (or leave) one of its occupied bands (including the most recent acquisition).

3.6. Occupying an Empty Band. BS n can join any new band m at any time slot $t + 1$, so long as the band was sensed to be consistently empty during the previous time slot t plus during an additional $\tilde{E}_n$ time slots before that. The reason we require that a band be empty for $\tilde{E}_n + 1$ *consecutive* time slots before declaring it truly empty is to mitigate the problem that fading can cause, since deep fades can cause misdetection of a PU (or another BS), which fools BS n into thinking the band is empty, while it is actually not.

The decision to join a new empty band (once emptiness is confirmed) is based on the observed load on the system. When $\Lambda_n(t) \geq 1$ (which indicates low load), BS n contends for one of the bands sensed to be empty if and only if

$$\frac{o_n(t)}{\hat{S}_n} \times \left(1 - a_s \times \frac{\Lambda_n(t)}{M} \right) \leq 1, \qquad (19)$$

where $o_n(t) = s_n(t) + c_n(t)$ is the total number of bands occupied by BS n during time slot t. The band to be acquired is chosen to have the highest observed successful count $\sigma_n^m(t)$ out of all bands verified to be empty by the BS history table. If (19) is not true, BS n does not contend for any new bands. This equation attempts to force the BS to keep occupying bands until $o_n(t) = s_n(t) + c_n(t) \approx \hat{S}_n$. The $a_s \in [0, 1]$ weighting parameter represents an optional feature that permits BSs in an underloaded system to occupy more bands than their demand, rather than keeping those bands empty.

The band occupying behavior is slightly modified when $\Lambda_n(t) < 1$ indicating an overloaded system. In this case, BS n flips a coin to occupy an empty band using the join probability

$$J_n(t) = \max\left(1 - \frac{o_n(t)}{\hat{S}_n}, 0 \right). \qquad (20)$$

Equation (20) maintains fairness in the system and reduces contention between multiple cognitive BSs, because it requires that BSs that occupy a number of bands closer to their demand join new empty bands less frequently. If BS n decides to contend for an empty band, it picks the band with the highest success count $\sigma_n^m(t)$ or picks one at random from the set of bands with the same highest $\sigma_n^m(t)$.

3.7. Moving a Sensing Device. Sensing devices within BS n are a valuable asset since they are needed when the BS wants to occupy a new empty band or when it wants to estimate the overall system load. The proposed HoPSS technique is very careful in using such available sensing devices since their number is limited.

Firstly, to limit the number of required sensors, HoPSS cleverly utilizes the presence or absence of ACKs as an indirect method for sensing the set of occupied bands $\mathbb{O}_n(t)$. This mitigates the need for an extra $o_n(t)$ direct sensing devices, which would have been needed had ACKs not been employed.

Secondly, HoPSS keeps moving the D_n out-of-band sensing devices around to perform direct sensing of appropriate bands in an attempt to obtain a clear picture of the environment around the BS, including the ability to estimate the system load.

To effectively utilize such limited number of sensors, HoPSS is designed to veer the sensors towards empty spectrum bands. Empty bands provide a good indicator of the system load, since such bands quickly deplete as the system load increases. In addition, any band that the BS needs to join in a future time slot has to be an empty band, as busy bands carry the risk of having a PU (or another BS) active within them.

In HoPSS, we propose that each one of the active sensing devices $d_n \in \mathbb{D}_n$, which is currently sensing spectrum band $m \in \mathbb{V}_n(t)$, maintains a counter δ_n^m that starts at zero for that device as soon as it starts sensing band m. The counter is incremented by one for every time slot in which this band is actively sensed *and* its status is found to be busy ($\mathbb{b}$) by this sensing device. When the sensed band status changes from busy to empty in the next time slot (before the sensing device leaves the band), the counter δ_n^m is reset to zero.

The device d_n stops sensing band m and leaves to another band m' when the corresponding counter reaches a preset limit Δ_n (i.e., when we reach $\delta_n^m \geq \Delta_n$). In other words when a band is sensed to be busy for Δ_n consecutive time slots, the sensing device assumes the band is occupied by a PU or another BS and that PU or BS will stay there for some time (for example, for one full maintenance window).

Setting Δ_n to a large value prevents the sensing device from leaving a particular busy band even if it stays busy for a long time, thus wasting an opportunity to discover a new empty band. A small value of Δ_n, on the other hand, risks the sensor leaving an empty band quickly due to isolated events of false alarms. We chose a value of $\Delta_n \approx \widehat{I}_n$ to achieve a good balance between the above two factors. For example, if we set $\Delta_n = \widehat{I}_n = 3$ time slots, then three consecutive busy periods will rule out random bursts of noise (which typically do not generate three consecutive false alarms). In addition, if a collision occurs between two BSs on a sensed band, this collision will probably clear during about $\widehat{I}_n$ consecutive time slots, allowing the sensing device to observe if that band goes back to the empty state or not.

However, once the band is confirmed to be busy to a high degree of certainty (i.e., once $\delta_n^m \geq \Delta_n$), then there is no point of sensing the band further since the PU (or another BS) that occupied the band probably needs to use the band for some time, and hence we move that sensing device d_n to attempt to find a new empty band.

Needless to say, if an empty band being sensed by BS n is occupied (at a later point) by BS n itself (i.e., the band moves from the $\mathbb{V}_n(t)$ set to the $\mathbb{O}_n(t)$ set), the sensing device for BS n also decides to stop direct sensing of that band and moves somewhere else, as the band can be easily sensed indirectly using ACKs once it is occupied. However, if a band remains empty and is not occupied by BS n, the sensing device stays attached to that band indefinitely.

3.8. Picking a New Band to Sense. After making a decision to move a sensing device d_n from its current band m at time slot t, HoPSS has to decide which band m' from the unknown bands set $\mathbb{U}_n(t)$ to pick for direct sensing using this device starting in the next time slot $t + 1$. Our proposed technique for picking a new band for sensing follows a few criteria explained next (see the pseudocode in Algorithm 1).

First, the new band m' to be sensed should not be the same as band m that the sensor just left because band m was busy for Δ_n consecutive time slots. Second, the new band m' to be sensed should neither be a band that is currently occupied by BS n (i.e., $m' \notin \mathbb{O}_n(t)$) nor be a band that is currently being directly sensed by another sensing device (i.e., $m' \notin \mathbb{V}_n(t)$). In other words, the new band should be picked from the set $\mathbb{U}_n(t)$.

When a band is no longer sensed directly via a sensing device or indirectly via ACKs (i.e., when the band moves from the $\mathbb{V}_n(t)$ or $\mathbb{O}_n(t)$ set to the $\mathbb{U}_n(t)$ set), BS n starts a counter $\zeta_n^m(t)$ for that band and increments the counter every consecutive time slot that band status remains unknown. This timer represents the time period in which the band status is unknown to BS n.

A sensing device does not go back to an unknown band m until $\zeta_n^m(t)$ exceeds the minimum maintenance window size (i.e., until $\zeta_n^m(t) > w$). This gives some time for the PU or BS occupying that unknown band to leave it and also prevents the sensing device from jumping back and forth between busy and unknown spectrum bands. The above condition, however, is not a stringent condition, because if BS n cannot find any band $m' \in \mathbb{U}_n(t)$ that satisfies $\zeta_n^m(t) > w$, this condition is simply ignored.

Out of the $\mathbb{U}_n(t)$ spectrum bands that satisfy the above condition, the band with the maximum success history count $\sigma_n^m(t)$ is picked for sensing. If multiple bands have the same maximum success count $\sigma_n^m(t)$, one of such bands is picked at random. Randomness in distributing the different BS sensors over multiple bands helps minimize collisions when BSs eventually decide to acquire such bands.

Notice that the maximum success count $\sigma_n^m(t)$ is used to pick the band to be sensed. This is similar to the criterion for joining a new band explained earlier in Section 3.6. The choice makes sense because picking a band to sense is the first stage to acquiring that band, and if we pick bands to sense that have the potential to limit band switching overhead between BSs, we will eventually pick such bands for acquisition. This harmony of picking bands for sensing and then picking bands for acquisition is what provides us with remarkable performance for the proposed HoPSS technique, even with a small number of sensing devices. Algorithm 1 shows the procedure for picking a new band to sense.

4. Algorithm Performance

4.1. Simulation Setup. To test HoPSS performance, a distributed CRN is simulated with $N = 10$ cognitive BSs attempting access to $M = 100$ unlicensed spectrum bands. The simulation is run for a total of $T = 20,000$ time slots. The key control parameters used for both HoPSS and HOP-M protocols are summarized in Table 1. We assume the spectrum demand for all BSs is identical $\widehat{S}_n = \widehat{S}$, $\forall n \in \mathbb{N}$, and we vary this demand from $\widehat{S} = 5$ bands to $\widehat{S} = 15$ bands to control *system load*, which is $L = N \times \widehat{S}/M$. Hence, the system load varies between $L = 50\%$ and 150%.

It is worth noting that in our simulations of the HoPSS protocol we only utilize $D_n = 20$ sensing devices per BS. Compare this to the number of devices required by the HOP-M technique, which is $D_n = 100$ per BS. This illustrates the cost savings potential of HoPSS. We demonstrate later the comparable performance for the two techniques though HoPSS uses only one-fifth of the number of sensors used by HOP-M.

We also note in Table 1 the extra three HoPSS parameters that provide it with more resistance to the unreliability of practical sensing devices, namely, minimum contention time $\widetilde{I}_n$, empty verification period $\widetilde{E}_n$, and busy verification period

```
Algorithm: Pick new band $m'$ for sensing by BS $n$
Leave band $m$ in time slot $t$
Identify eligible bands set $\mathbb{Q} = \mathbb{U}_n(t)$
Remove band $m$ from eligible bands set $\mathbb{Q}$
Remove bands $\widehat{m}$ with $\zeta_n^{\widehat{m}}(t) \leq w$ from set $\mathbb{Q}$
if $\mathbb{Q} \neq \emptyset$ then /* possible bands */
    $\mathbf{\Omega}_t \longleftarrow \max_{\forall \widehat{m} \in \mathbb{Q}} \sigma_n^{\widehat{m}}(t)$ /* maximum success */
    Candidates $\mathscr{B} \longleftarrow \{\widehat{m} : \widehat{m} \in \mathbb{Q} \wedge \sigma_n^{\widehat{m}}(t) = \mathbf{\Omega}_t\}$
    Pick band $m'$ at random from candidates $\mathscr{B}$
else /* $\mathbb{Q} = \emptyset$ no eligible bands in first round */
    Identify eligible bands set $\mathbb{Q} = \mathbb{U}_n(t)$
    Remove band $m$ from eligible bands set $\mathbb{Q}$
    if $\mathbb{Q} \neq \emptyset$ then /* relaxing condition worked */
        $\mathbf{\Omega}_t \longleftarrow \max_{\forall \widehat{m} \in \mathbb{Q}} \sigma_n^{\widehat{m}}(t)$ /* maximum success */
        Candidates $\mathscr{B} \longleftarrow \{\widehat{m} : \widehat{m} \in \mathbb{Q} \wedge \sigma_n^{\widehat{m}}(t) = \mathbf{\Omega}_t\}$
        Pick band $m'$ at random from candidates $\mathscr{B}$
    else /* $\mathbb{Q} = \emptyset$ still no eligible bands */
        Pick band $m$ for sensing /* last resort */
    end
end
```

Algorithm 1: Pseudocode for picking a new band m' for sensing by BS n at slot t. Actual sensing starts in the next time slot $t + 1$.

Table 1: Main simulation parameters.

Parameter	HoPSS	HOP-M
Out-of-band sensing devices D_n	20	100
Initial maintenance window w	30	30
History record length H_n	200	200
Constant a_d	4.0	4.0
Minimum contention time $\widetilde{I}_n$	2	N/A
Contention time limit $\widehat{I}_n$	3	3
Constant a_i	0.3	0.3
Empty verification period $\widetilde{E}_n$	2	N/A
Constant a_s	0.1	0.1
Busy verification period Δ_n	3	N/A

Δ_n. The minimum contention time $\widetilde{I}_n$ protects BSs from the effects of false alarms, where bursts of random noise can seem like a collision to a BS, forcing it to leave prematurely a successfully occupied band. On the other hand, the empty verification period $\widetilde{E}_n$ mitigates the problem of fading causing misdetection of a PU (or another BS), hence tempting a newcomer BS to join a band thinking it was empty, while it is actually not. Finally, the busy verification period Δ_n allows a sensing device d_n to verify that a band m is truly busy, ruling out the case of random bursts of noise masquerading as a PU transmission. This allows the sensing device to leave to another band m' comfortably knowing that band m was truly busy.

In our simulations, we consider two scenarios for both HoPSS and HOP-M. The first is when the BSs employ very complex and accurate hardware for sensing (which, of course, is expensive). In this case sensing accuracy is almost ideal, and we assume a false alarm probability $P_{fa} = 0.0$ and a misdetection probability $P_{md} = 0.0$. We also consider lost ACK probability to be $P_n = 0.0$. This is called the perfect sensing case. The other scenario we consider is the case when the sensing devices employed are the cheaper and less-reliable EDs, which deliver the inferior performance of $P_{fa} = 0.1$, $P_{md} = 0.1$, and $P_n = 0.01$.

We will hold a comparison between our proposed HoPSS technique and the following methods: (a) history-based HOP-M technique [17], which is assumed to have $D_n = 100$ dedicated sensing devices per BS available at its disposal, and (b) an ideal *centralized* system, which distributes spectrum bands among BSs according to their demand $\widehat{S}_n$. This is performed with the help of a centralized authority and one dedicated common control channel (CCC) for communications between the controller and the cognitive BSs. For this idealistic case, we will assume perfect sensing performance, which allows this setup to represent an upper bound on performance. Another system we compare to is (c) a *random* fully distributed allocation technique, which corresponds to a lower bound on performance. Here, cognitive BSs blindly select at the beginning of each time slot in a random fashion, as well as with equal probability, $\lceil \widehat{S}_n \rceil$ bands from the M existing bands. Due to the absence of prior sensing in this case, excessive collisions can occur between BSs themselves and between BSs and PUs.

One last system we simulate is (d) the *gradient ascent* learning algorithm described in [30]. This technique utilizes a distributed game theory approach to perform channel assignment. In such technique, BSs behave as players who go through a gradual learning process. Each BS n calculates a probability vector $\pi_n^m(t)$, describing the probability of BS n joining any band $m \in \mathbb{M}$ during time slot t. This vector represents BS n strategy for the game during that time slot.

The probability vectors from the various BSs are broadcast at the end of the time slot to other BSs, so they can adapt their future game strategies. Each BS believes that, by varying its own probabilities in a particular round, other BSs will reciprocate by altering their own probabilities (future game strategies) in response to its own strategy change. This is done by calculating the next iteration of probabilities $\pi_n^m(t+1)$. The calculations are determined by solving a data transmission optimization problem as explained in [30]. Because sensing is used here only to detect PUs and not to resolve contention, we will assume perfect sensing is available for this technique at all times. The parameters we use for gradient ascent are based on the recommendations in [30]; specifically we set belief factors $\alpha_{n,m}$ to be uniformly distributed between N and $2 \times N \times M - 1$ and the step size $\beta_n = 1/(2 \times N \times M - 1)$. The initial value for $\pi_n^m(1)$ is $1/M$, but then BS n adapts these probabilities based on the gradient ascent learning algorithm.

4.2. Performance Metrics. The following performance metrics are used to compare the performance of the proposed HoPSS technique to the HOP-M procedure, the purely random algorithm, the gradient ascent learning method, and the perfect centralized technique.

BS Successful Bands. The number of successfully occupied bands by BS n during time slot t, $s_n(t)$, is recorded. When unreliable sensing is utilized, $s_n(t)$ is the value observed by BS n, which is the same as the actual value, as confirmed by the arrival of an ACK. If a collision occurs with a PU or another BS, or when a burst of noise corrupts transmission and prevents an ACK from being received, such situation does not count as a successful transmission. We use the symbol S_n to represent the average of $s_n(t)$ for BS n over the whole simulation time, while denoting by $S^t = \sum_{n=1}^{N} s_n(t)/N$ the average of $s_n(t)$ for all BSs over one particular time slot t. Finally, the average of $s_n(t)$ for all BSs over the whole simulation time is designated by S.

BS Collision Bands. This is the number of spectrum bands acquired by BS n under collision during time slot t, $c_n(t)$. Collisions on a BS include all three possibilities: interference from a PU, contention from another BS, or a noise burst preventing an ACK from arriving. We will label the average of $c_n(t)$ over simulation time for one BS n as C_n, the average of $c_n(t)$ for all BSs over one time slot t as C^t, and the average of $c_n(t)$ for all BSs over the whole simulation time as C.

PU Interference Time. This is defined as the *number of consecutive time slots* that a cognitive BS takes to stop interfering with a PU after the PU gets activated. We denote this value by T_i.

Settling Time. This is the *number of consecutive time slots* needed for $s_n(t)$ of a disturbed BS n to return to its steady-state average value S_n after a disturbance to the CRN occurs. The settling time is denoted by T_s and represents a measure of the system convergence time after a disturbance, such as when a PU gets activated causing some BSs to lose some of the bands they are occupying to the benefit of the PU.

Spectrum Occupancy Rank Diagram. To show the amount of band swapping (frequency switching) overhead the cognitive BSs suffer from, we show the *spectrum occupancy rank diagram*, which displays the *occupancy frequency* $f_n(m)$ in which BS n successfully occupied band m throughout the simulation period. In other words, $f_n(m)$ is the ratio of successful acquisition time of band m divided by the total simulation time T. For better visuals, the x-axis in this diagram will show the rank of the spectrum band r_m (rather than its index m), where the most frequently occupied band is given rank $r_m = 1$, the second most occupied band is $r_m = 2$, and so on (see Figure 4).

A spectrum occupancy rank diagram that is compact in shape means that the BS limits itself to occupying a small set of spectrum bands, while a smeared-out curve means the BS keeps switching its operating frequency between a large set of bands [17].

5. Results and Discussion

To illustrate the various aspects of the HoPSS technique, we simulated a group of scenarios, the results of which are explained below.

5.1. Homogeneous Network. In the first experiment, all cognitive BSs have the same spectrum demand, $\widehat{S}_n = \widehat{S}$, $\forall n \in \mathbb{N}$, where $\widehat{S}$ is varied between 5 and 15 bands to attain a system load of $L = 50\% - 150\%$. For now, PUs are not activated within the CRN just yet. The behavior of the BSs when occupying bands under both low and high load conditions and under both perfect and unreliable sensing is investigated. We set HoPSS to use only $D_n = 20$ sensing devices, which is significantly smaller than what HOP-M uses ($D_n = 100$ to be specific).

The behavior of the HoPSS algorithm under both low load and high load scenarios (and perfect sensing) is illustrated in Figure 2, which shows the number of successfully acquired bands $s_7(t)$ for BS7, as an example, versus time. In the case of lower load ($L = 75\%$), HoPSS manages to provide the desirable stability for the BS, which quickly acquires its desired spectrum demand and stays in such bands without enduring band swapping overhead, since the BS notices that there are enough empty bands in the system for everyone to use. The ability of HoPSS to accurately estimate the load on the system without the need for a large number of sensing devices is quite impressive. This capability carries over as the load on the system increases to $L = 100\%$ and $L = 125\%$, in which the BSs are able to sense that increase in load and respond by moving into the band swapping mode to allow various BSs to obtain bands fairly (see Figures 2 and 6).

The behavior of HoPSS remains consistent even when unreliable sensors are used, which is the case shown in Figure 3. Of course, perfect stability is not possible in this case due to false alarms and instances of misdetection, which affect the number of empty bands observed by the BS, and hence can slightly change the load estimate obtained by HoPSS. This occasionally pushes HoPSS into the band swapping mode. However, moving into this mode does not mean a drop in performance since the *average* number of bands obtained by BS7, denoted by S_7, is almost identical to that of the perfect

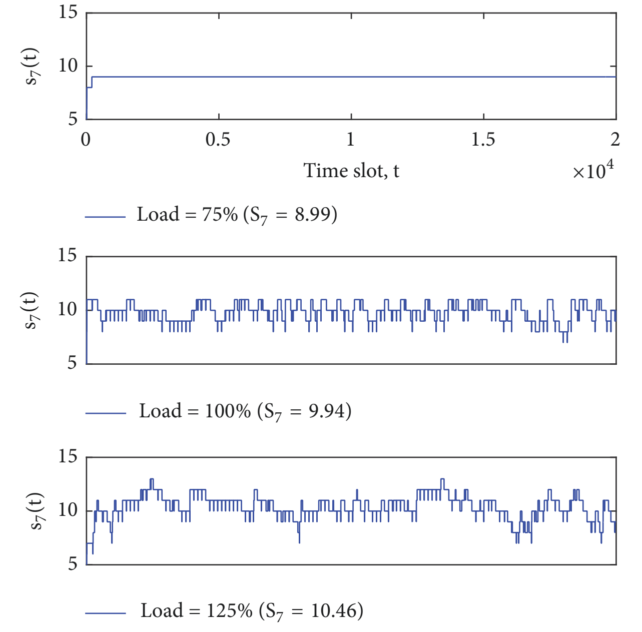


FIGURE 2: Number of bands successfully acquired by BS7 versus time under different system loads when sensing is perfect (P_{fa} = 0.0, P_{md} = 0.0, and P_n = 0.0).

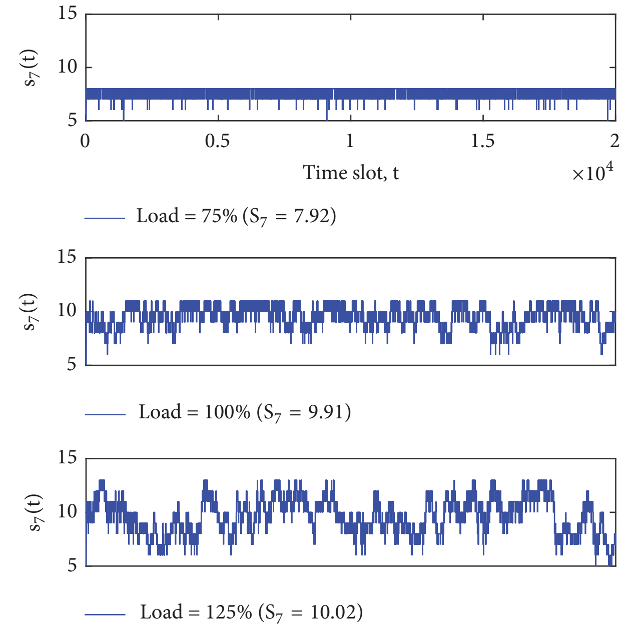


FIGURE 3: Number of bands successfully acquired by BS7 versus time under different system loads when sensing is unreliable (P_{fa} = 0.1, P_{md} = 0.1, and P_n = 0.01).

sensing case under system loads of L = 100% and L = 125%, which is quite remarkable. In the case of L = 75%, where each BS spectrum demand was $\widehat{S}$ = 7.5 bands, BS7 got more bands than it requested in both the perfect and unreliable sensing cases.

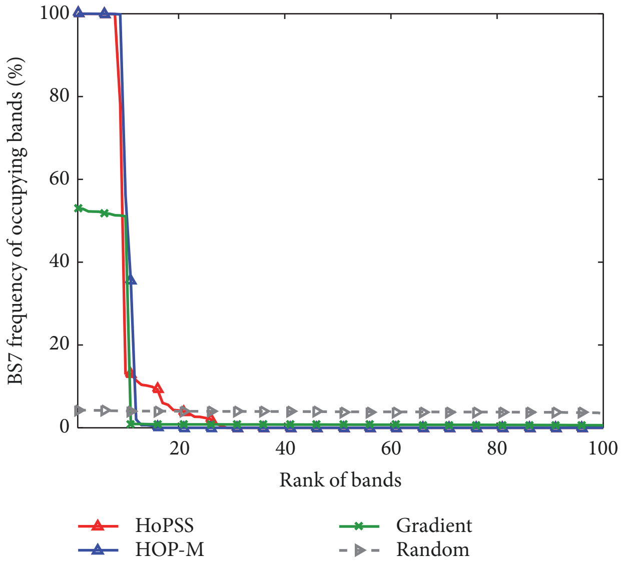


FIGURE 4: BS7 spectrum occupancy rank diagram at L = 100% when sensing is perfect.

To elaborate on the spectrum switching overhead incurred by HoPSS, we compare its behavior to other techniques in Figures 4 and 5, which show the spectrum occupancy rank diagram for BS7 under both perfect and unreliable sensing, respectively. It is clear from the figures that the performance of HoPSS is very close to that of HOP-M in both cases, as both techniques try to limit BSs from jumping back and forth between different bands. It is significant that HoPSS can achieve this reduction in frequency switching overhead while using a small number of sensing devices.

Now we turn our attention to test for fairness of HoPSS in distributing the frequency bands between competing cognitive BSs. Figures 6 and 7 show the average number of successfully acquired bands S_n by the different BSs under a fully loaded system (i.e., L = 100%). Figure 6 shows the case of perfect sensing, while Figure 7 shows the case of unreliable sensing. The performance of HoPSS is compared to the different spectrum sharing techniques we tested.

Interestingly, HoPSS managed to distribute the spectrum bands evenly between the different BSs, just like HOP-M did, even though it had one-fifth of the number of sensors available to HOP-M and even though it is a fully distributed protocol. Even more remarkable is the fact that HoPSS managed to overcome the limitations of the unreliable sensors (providing wrong readings of the spectrum) because HoPSS delivered an average of S = 9.82 bands per BS, while HOP-M performance dipped slightly due to the less-reliable sensing hardware, giving an average of S = 9.21 bands per BS. In this regard, the performance of HoPSS is closer to the ideal centralized system (compared to other tested methods) because it employs a clever methodology for accessing the spectrum.

This performance is consistent under all load conditions, as illustrated in Figures 8 and 9, which show the average number of successfully acquired bands per BS when we vary

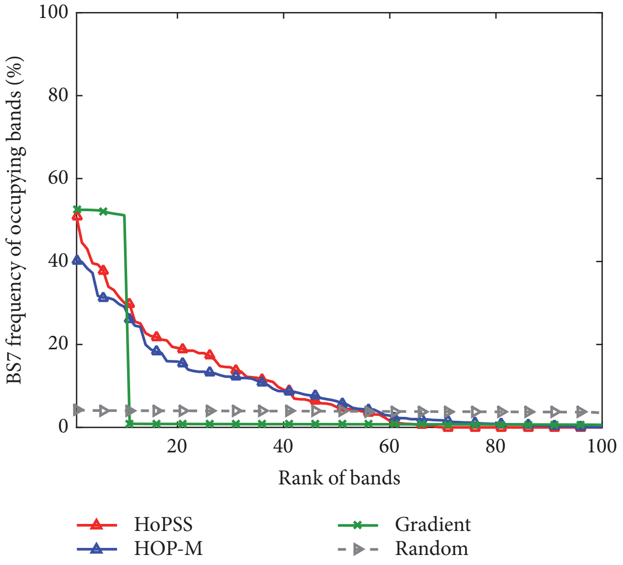


Figure 5: BS7 spectrum occupancy rank diagram at $L = 100\%$ when sensing is unreliable.

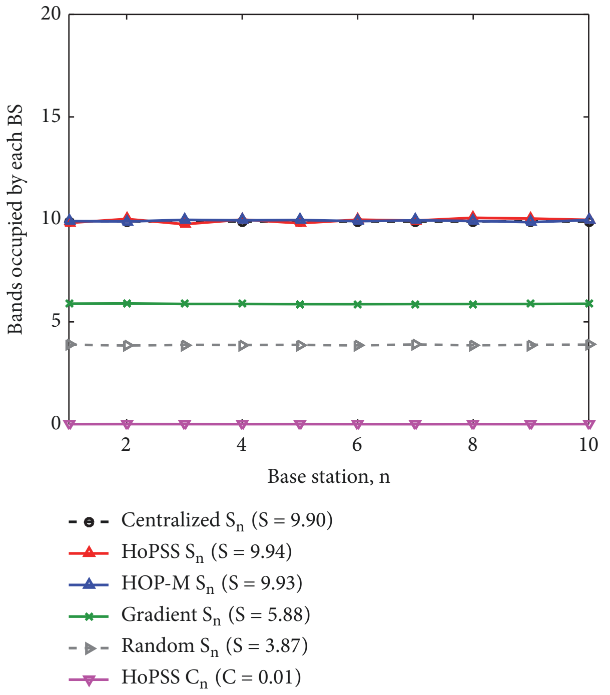


Figure 6: Successfully occupied bands by each BS when sensing is perfect and $L = 100\%$.

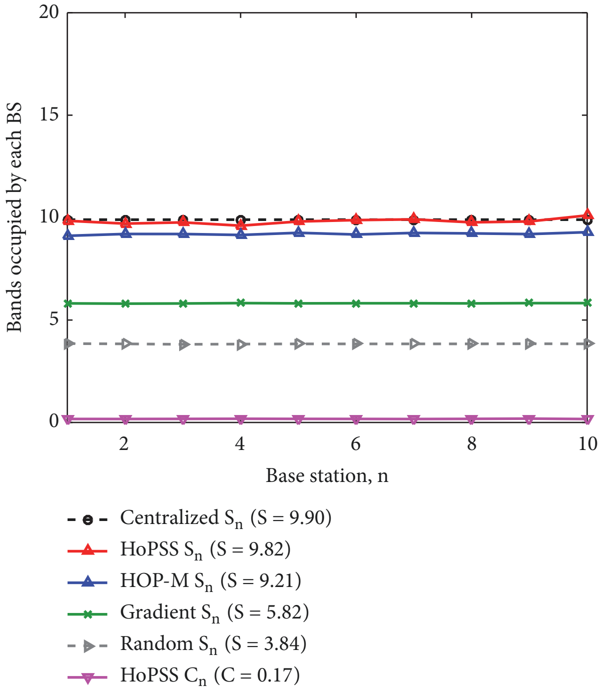


Figure 7: Successfully occupied bands by each BS when sensing is unreliable and $L = 100\%$.

the system load between 50% and 150%. Only when sensing is perfect does the HOP-M protocol manage to provide the same superior throughput provided by the proposed HoPSS technique, but HOP-M performance dips below that of HoPSS for the unreliable sensing case.

To show how resilient HoPSS is to the surrounding environment, we show in Figure 10 the average number of successfully acquired bands per BS (i.e., S) when varying the number of employed sensing devices for HoPSS and the load on the system. The figure clearly shows that the performance is almost insensitive to the number of sensors D_n utilized, irrespective of system load. Even if we use a very small number of sensing devices (as low as $D_n = 5$ sensors), the throughput drop is almost negligible (see $L = 75\%$ case), which opens the door for more savings in hardware cost. The little degradation in performance when the number of sensors is increased in HoPSS is an artifact of (15), in which the sum $\sum_{m=1}^{M} \mu_n^m(t)$ becomes smaller when the number of sensors increases, which slightly changes the system load estimate $\Lambda_n(t)$ that the HoPSS algorithm calculates, but not in any significant manner.

Finally, Figures 11 and 12 show the average number of successfully acquired bands per BS for both HoPSS and HOP-M when varying the false alarm and misdetection probability, respectively. In Figure 11 we vary P_{fa} between 0.0 and 0.25, while maintaining $P_{md} = 0.1$ and $P_n = 0.01$. In Figure 12, on the other hand, we vary P_{md} between 0.0 and 0.25 but maintain $P_{fa} = 0.1$ and $P_n = 0.01$. It is evident from the results that HoPSS is more resilient to degradation in sensor performance compared to HOP-M, allowing HoPSS to easily work with cheaper hardware.

5.2. PU Disturbance. The experiment conducted here is very similar to the previous experiment, with only one difference: the network has multiple PUs that get activated in the middle of the simulation time, and such PUs cannibalize all spectrum bands successfully occupied by BS1. This experiment allows us to study the behavior of the victim BS1 as well as other BSs when PUs get activated.

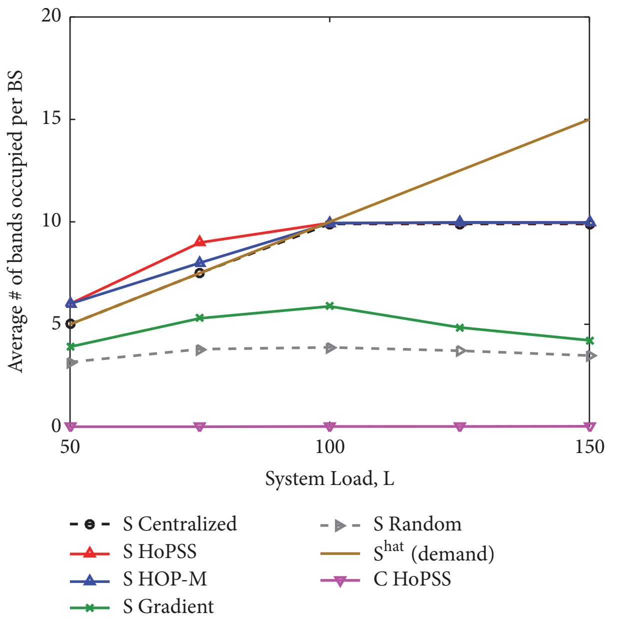


Figure 8: Average number of successfully acquired bands per BS for different system loads when sensing is perfect.

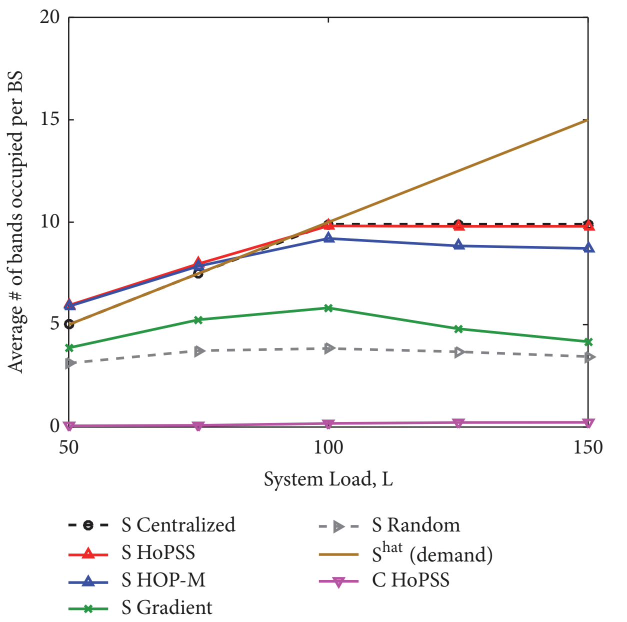


Figure 9: Average number of successfully acquired bands per BS for different system loads when sensing is unreliable.

Figure 13 shows that BS1 quickly vacates the bands occupied by PUs in the middle of the simulation, as the count of its successfully occupied bands $s_1(t)$ drops quickly to zero. This behavior is the one required from cognitive BSs in a CRN network. The interference time on PUs is limited to about $T_i = 8$ time slots when $L = 100\%$, which is very small (see also Figures 14 and 15). Once BS1 leaves its bands it starts looking for empty bands in the system, while its neighboring cognitive BSs slowly relinquish some of their bands to the benefit of BS1 so it can get its fair share of the spectrum.

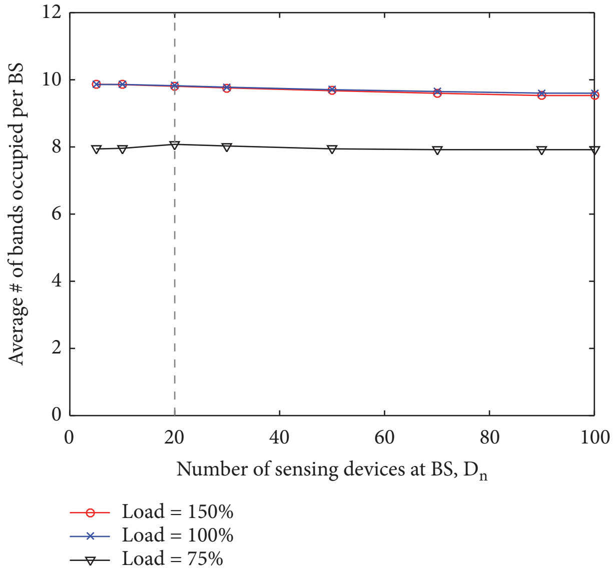


Figure 10: Average number of successfully acquired bands per BS when varying the number of sensing devices under unreliable sensing regime.

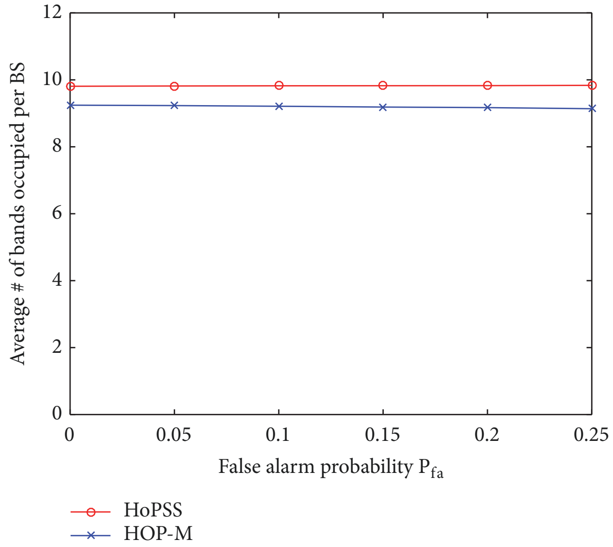


Figure 11: Average number of successfully acquired bands per BS when varying the false alarm probability P_{fa} for $L = 100\%$.

Both the BS settling time T_s and PU interference time T_i for HoPSS are compared to those for HOP-M in Figures 14 and 15 as we vary the system load. We show both cases when sensing is perfect and when sensing is unreliable, respectively. The most important observation is how small the PU interference time T_i is, irrespective of the load on the system, and this is true for both perfect and unreliable sensing.

The second observation is that HOP-M has better settling times compared to HoPSS, which is to be expected since BSs in HoPSS refuse to join new bands before waiting for at least

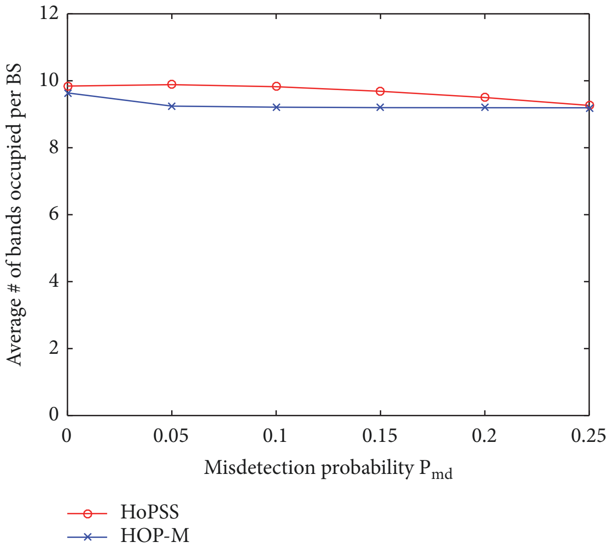


FIGURE 12: Average number of successfully acquired bands per BS when varying the misdetection probability P_{md} for $L = 100\%$.

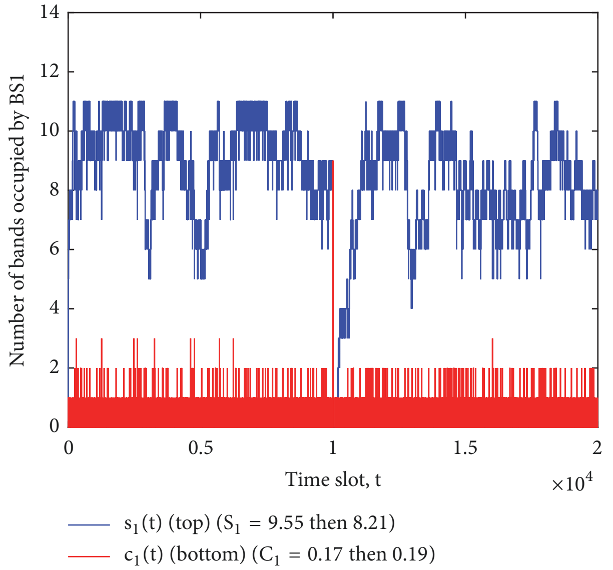


FIGURE 13: Number of spectrum bands occupied by BS1 versus time at $L = 100\%$ and under unreliable sensing regime.

$\tilde{E}_n + 1$ time slots to make sure the bands are actually empty, which is not the behavior of HOP-M. However, the difference in settling time is not a big disadvantage since it only manifests itself in overloaded systems ($L \geq 100\%$), which is not a typical operating scenario in real-life networks. For the more practical case of lower loads ($L < 100\%$), the settling time is negligible for both HOP-M and HoPSS.

To put things into perspective, consider a typical frame length (i.e., time slot) of 10 milliseconds [33]; a maximum of $T_s = 1600$ time slots means that BS1 can stabilize within only 16 seconds at such an overloaded system. Other BSs stabilize much faster. In addition, this settling time drops significantly for lower loads on the system.

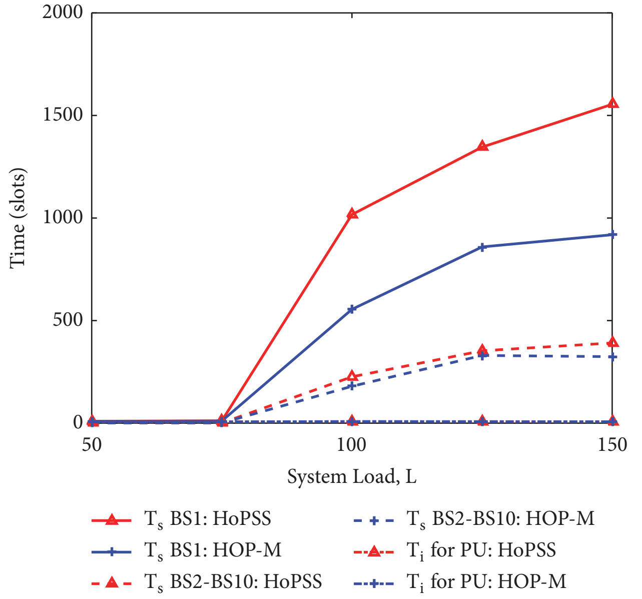


FIGURE 14: Settling time and PU interference time versus system load for perfect sensing.

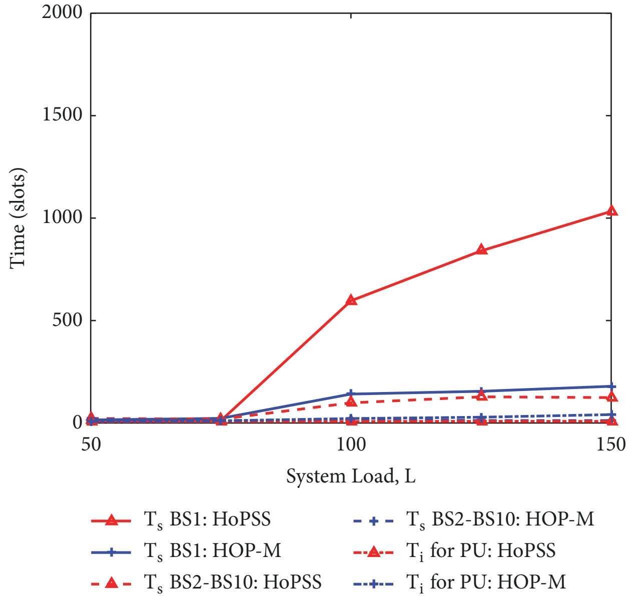


FIGURE 15: Settling time and PU interference time versus system load for unreliable sensing.

6. Conclusions

History-based spectrum access represents a promising approach to CRNs. In such technique, cognitive BSs independently estimate the system load and adaptively swap their occupied spectrum bands to ensure allocation fairness and high overall throughput. In this paper, we introduced several extensions to the history-based HOP-M protocol, which allow significant reduction in hardware cost required by HOP-M.

Our approach, called HoPSS, can use less than one-fifth of the sensing devices required by HOP-M, without noticeable penalty in performance, especially at moderate system loads. In addition, the HoPSS algorithm works seamlessly with unreliable sensing. Not only that, but also when unreliable sensing devices are employed, the throughput obtained by HoPSS exceeds that of HOP-M at a fraction of the cost. Finally, HoPSS retains the distributed nature of HOP-M and hence has all the advantages of not requiring a centralized controller or a common control channel (CCC).

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] M. Amjad, F. Akhtar, M. H. Rehmani, M. Reisslein, and T. Umer, "Full-Duplex Communication in Cognitive Radio Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2158–2191, 2017.

[2] R. Al-Zubi, K. A. Darabkh, M. Hawa, and I. F. Jafar, "RSP-WRAN: Resource sharing protocol for inter/intra WRAN communications," *Journal of High Speed Networks*, vol. 24, no. 1, pp. 31–47, 2018.

[3] A. Ali and W. Hamouda, "Advances on Spectrum Sensing for Cognitive Radio Networks: Theory and Applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1277–1304, 2017.

[4] L. Zhai and H. Wang, "Crowdsensing task assignment based on particle swarm optimization in cognitive radio networks," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 4687974, 9 pages, 2017.

[5] H. F. Al-Doseri and M. A. Mangoud, "Performance Analysis of Cooperative Spectrum Sensing under Guaranteed Throughput Constraints for Cognitive Radio Networks," *Journal of Computer Networks and Communications*, vol. 2016, Article ID 9706781, 9 pages, 2016.

[6] L. Yang, Z. Chen, and F. Yin, "Cyclo-energy detector for spectrum sensing in cognitive radio," *AEÜ - International Journal of Electronics and Communications*, vol. 66, no. 1, pp. 89–92, 2012.

[7] H. Lin, J. Hu, C. Huang, L. Xu, and B. Wu, "Secure cooperative spectrum sensing and allocation in distributed cognitive radio networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 10, pp. 674-591, 2015.

[8] W. Khalid and H. Yu, "Optimal sensing performance for cooperative and non-cooperative cognitive radio networks," *International Journal of Distributed Sensor Networks*, vol. 13, no. 11, 2017.

[9] A. Bagwari, J. Kanti, G. S. Tomar, and A. Samarah, "Reliable spectrum sensing scheme based on dual detector with double-threshold for IEEE 802.22 WRAN," *Journal of High Speed Networks*, vol. 21, no. 3, pp. 205–220, 2015.

[10] D. M. Martínez and Á. G. Andrade, "Adaptive energy detector for spectrum sensing in cognitive radio networks," *Computers and Electrical Engineering*, vol. 52, pp. 226–239, 2016.

[11] Feng Hu, Bing Chen, Xiangping Zhai, and Chunsheng Zhu, "Channel Selection Policy in Multi-SU and Multi-PU Cognitive Radio Networks with Energy Harvesting for Internet of Everything," *Mobile Information Systems*, vol. 2016, Article ID 6024928, 12 pages, 2016.

[12] J. Elhachmi and Z. Guennoun, "Cognitive radio spectrum allocation using genetic algorithm," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1–133, 2016.

[13] M. Hawa, A. AlAmmouri, A. Alhiary, and N. Alhamad, "Distributed opportunistic spectrum sharing in cognitive radio networks," *International Journal of Communication Systems*, vol. 30, no. 7, 2017.

[14] G. Prasad Joshi, S. Y. Nam, S. W. Kim, and B.-S. Kim, "Adaptive window size-based medium access control protocol for cognitive radio wireless sensor networks," *Journal of Sensors*, vol. 2016, Article ID 2049859, 9 pages, 2016.

[15] M. A. Hossain and N. I. Sarkar, "A distributed multichannel MAC protocol for rendezvous establishment in cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 70, pp. 44–60, 2018.

[16] N. Rastegardoost and B. Jabbari, "Blind channel selection strategies for distributed cognitive MAC," in *Proceedings of the 27th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC 2016*, pp. 1–6, Valencia, Spain, September 2016.

[17] M. Hawa, D. I. Abu-Al-Nadi, O. M. K. Alsmadi, and I. F. Jafar, "On Using Spectrum History to Manage Opportunistic Access in Cognitive Radio Networks," *IEEE Access*, vol. 4, pp. 5293–5308, 2016.

[18] N. Ul Hasan, W. Ejaz, N. Ejaz, H. S. Kim, A. Anpalagan, and M. Jo, "Network Selection and Channel Allocation for Spectrum Sharing in 5G Heterogeneous Networks," *IEEE Access*, vol. 4, pp. 980–992, 2016.

[19] L. Xu, J. Wang, Y.-P. Li, Q. Li, and X. Zhang, "Resource allocation algorithm based on hybrid particle swarm optimization for multiuser cognitive OFDM network," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7186–7194, 2015.

[20] N. Zhao, S. Li, and Z. Wu, "Cognitive radio engine design based on ant colony optimization," *Wireless Personal Communications*, vol. 65, no. 1, pp. 15–24, 2012.

[21] U. Mehboob, J. Qadir, S. Ali, and A. Vasilakos, "Genetic algorithms in wireless networking: techniques, applications, and issues," *Soft Computing*, vol. 20, no. 6, pp. 2467–2501, 2016.

[22] S. Latif, S. Akraam, and M. A. Saleem, "Channel assignment using differential evolution algorithm in cognitive radio networks," *International Journal of Advanced and Applied Sciences*, vol. 4, pp. 160–166, 2017.

[23] S. Latif, S. Akraam, and M. A. Saleem, "Joint optimization of interference and cost in cognitive radio heterogeneous network using fuzzy logic powered ants," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 1075252, 11 pages, 2017.

[24] D. López, E. Rivas, and O. Gualdron, "Primary user characterization for cognitive radio wireless networks using a neural system based on Deep Learning," *Artificial Intelligence Review*, pp. 1–27, 2017.

[25] N. Ul Hasan and W. Tu, "On energy-efficient performance-guaranteed channel assignment in cognitive radio-based wireless mesh networks," *International Journal of Communication Systems*, vol. 28, no. 17, pp. 2197–2213, 2015.

[26] Z. Huo, X. Li, S. Jin, and Z. Wang, "Nash Equilibrium of an Energy Saving Strategy with Dual Rate Transmission in Wireless Regional Area Network," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 9053862, 10pages, 2017.

[27] H. Li and Z. Han, "Collaborative Spectrum Sensing with a Stranger: Trust, or Not to Trust?" in *Proceedings of the Networking Conference (WCNC)*, pp. 1–6, Sydney, Australia, April 2010.

[28] Q. Minhas, H. Mahmood, and H. Malik, "Channel selection for simultaneous move game in cognitive radio ad hoc networks," *Wireless Networks*, vol. 22, no. 1, pp. 61–68, 2016.

[29] G. Li, Z. Shang, and K. Yang, "Detection and Leasing of Joint Space and Spectrum Opportunities by Multiple Secondary Network Operators in Cognitive Radio Systems," *Journal of Signal Processing Systems*, vol. 83, no. 2, pp. 293–308, 2016.

[30] X. Chen, Z. Zhao, D. Grace, and H. Zhang, "Reciprocal learning for cognitive medium access," in *Proceedings of the 2013 IEEE Wireless Communications and Networking Conference, WCNC 2013*, pp. 89–94, China, April 2013.

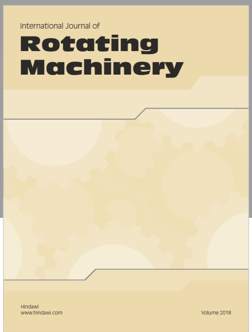
[31] S. L. Macdonald and D. C. Popescu, "Impact of primary user activity on the performance of energy-based spectrum sensing in Cognitive Radio systems," in *Proceedings of the 2013 IEEE Global Communications Conference, GLOBECOM 2013*, pp. 3224–3228, Atlanta, GA, USA, December 2013.

[32] C. Charan and R. Paney, "Eigenvalue based double threshold spectrum sensing under noise uncertainty for cognitive radio," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 15, pp. 5968–5975, 2016.

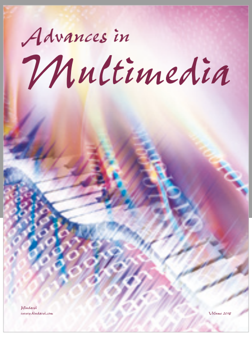
[33] IEEE, "IEEE standard for local and metropolitan area networks - specific requirements - Part 22. Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Policies and procedures for operation in the TV bands," IEEE 802.22-2011, 2011.

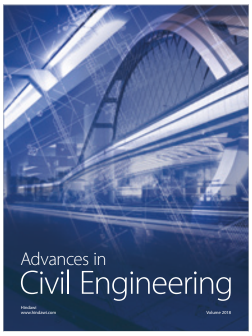
Journal of
Engineering

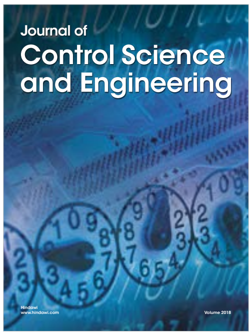
The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

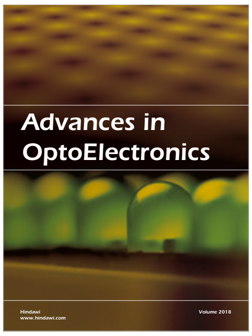

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

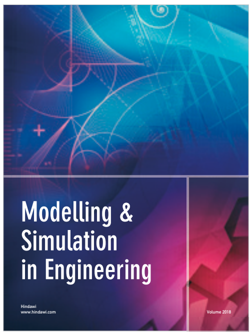
Journal of
Robotics

Journal of
Electrical and Computer Engineering

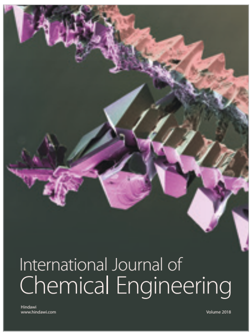

Advances in
OptoElectronics

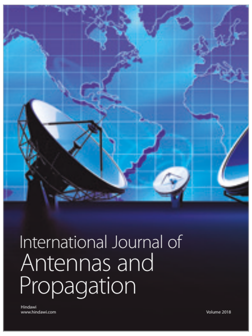
VLSI Design

International Journal of
Navigation and Observation

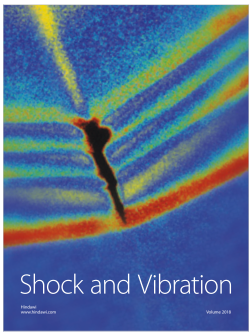

Modelling & Simulation in Engineering

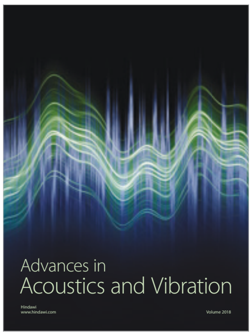
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration