Decoupling Trust and Wireless Channel Induced Effects on Collaborative Sensing Attacks

Yifeng Cai[1], Liu Cui[2], Konstantinos Pelechrinis[2], Prashant Krishnamurthy[2], Martin B.H. Weiss[2], Yijun Mo[1]
[1]Huazhong University of Science and Technology, Wuhan, Hubei, China, 430074
[2]University of Pittsburgh, Pittsburgh, PA, USA, 15260
Emails: cyf@hust.edu.cn, lic49@pitt.edu, kpele@pitt.edu, prashant@mail.sis.pitt.edu,
mweiss@sis.pitt.edu, moyj@mail.hust.edu.cn

***Abstract*—One of the most crucial functionalities of cognitive radio networks is spectrum sensing. Completing this task in an accurate manner requires opportunistic spectrum access. Traditionally, sensing has been performed through energy detection by each individual secondary user. In order to increase accuracy, individual measurements are aggregated using different fusion functions. However, even though collaborative spectrum sensing can increase accuracy under benign settings, it is prone to falsification attacks, where malicious secondary users report fake sensings. Previous studies have designed trust (reputation) based systems to contain the effect of the adversaries, ignoring to a large extent the wireless channel irregularities when performing the computation. In this paper, we decouple the reasons behind a false sensing report and propose the Decoupling Trust and Capability Spectrum Sensing System (DTCS^3). DTCS^3 is a collaborative spectrum sensing system that takes into account both a secondary sensor node's trust and its capability to sense the channel. Through thorough evaluations that consider a large variety of attack strategies, we show that by accounting for wireless induced effects while calculating the reporting trust of a secondary user, we can significantly improve the performance of a collaborative spectrum sensing system as compared to existing schemes in the literature. In particular, the true positive/negative rates can be improved by as much as 36%, while DTCS^3 is able to *track* and *respond* to dynamic changes in the adversaries behavior.**

***Index Terms*—Cooperative Spectrum Sensing; Trust; Sensing Capability; Security**

I. Introduction

The concept of dynamic spectrum access and cognitive radio networks is not new and it was firstly introduced by Mitola [1]. In brief, the main idea is enabling opportunistic access to the available resources; licensed spectrum can be made available to unlicensed users (also called *secondary* users), when the licensed entities (also called *primary* users) are absent. Even though cognitive radio networks have yet to take off, there are significantly increased efforts towards commercializing them[1], especially due to the scarcity of the wireless spectrum and the rapidly increasing demand for wireless connectivity.

For systems using opportunistic sharing an accurate system to detect the presence or absence of a primary user needs to be in place[2]. If the spectrum holes are stochastic in nature, *spectrum sensing* is the dominant approach to acquire the context information needed to realize a cognitive radio network [4]. As we will discuss in more details later, sensing is mainly based on energy detection. Every secondary user observes the activity on the spectrum and decides whether there is a licensed user or not. In order to improve accuracy, collaboration among the secondary sensor nodes has been proposed. This essentially, reduces the uncertainty and consequently increases the confidence that the overall decision is the correct one. For instance, while a few sensors might suffer from severe fading and cannot detect the primary signals, overall the cooperation of the secondary users will provide a reliable decision.

This cooperation though, allows enough space for manipulation from malicious entities. Depending on the actual fusion algorithm applied on the individual secondary sensor readings, one or more colluding adversaries, can lead the system to a wrong decision. This can either cause disruptions in the service of the primary users (i.e., interference), or degrade the performance of the secondary users (i.e., missed spectrum access opportunities). Existing literature, utilizes reputation-based approaches to *filter* "untrusted" reports from the secondary sensors. While, there are different approaches for computing the reputation/trust, which we will discuss later in this paper, the majority of them treat every erroneous individual report as originating from a non-trusted entity.

However, let us consider the case of a completely trustworthy secondary user, say Jack, who suffers from severe fading conditions. Fading might force him to provide erroneous reports. Existing algorithms will consider this as an evidence of malicious behavior and Jack's reputation will be reduced rapidly, even though he is completely trustworthy. Furthermore, while "bad news travels fast", that is, the reputation of a user can degrade fast (only after a few erroneous reports), "good news travels slowly" (i.e., it takes time to build up one's reputation back) [5]. Hence, even when wireless environment conditions for Jack improve (e.g., due to the temporal or spatial variations of the wireless channel), his reportings will not be taken into account and this can pose a significant performance degradation. It should be evident that if the underlying reputation system could disengage the reasons behind a *false* report, similar to the above phenomena could be avoided.

In this paper, we design a module that operates on top of a collaborative spectrum sensing system and probabilistically decouples the reasons behind an erroneous report. In particular,

[1]For instance, IEEE 802.22 and the White Spaces Coalition [2].
[2]See [3] for a longer discussion of different modes of spectrum sharing.

we add the secondary user's *capability factor*. Hence, for every user we calculate two metrics, which both scale the user's sensor readings in the fusion function: (i) its *capability* to provide a correct report and (ii) its *reporting trust*[3]. To reiterate, the capability factor is related with the effect of wireless channel induced factors on the reportings and for its calculation we utilize a wireless fading channel model. On the other hand, the reporting trust is essentially an indicator of whether the user is malicious or not. We design the Decoupling Trust and Capability Spectrum Sensing System (DTCS^3), which operates in two phases. First it filters nodes according to both their trust and capability values. Consequently, DTCS^3 utilizes the sensing readings of the remaining nodes to decide whether there is a primary user on the channel or not.

The main contribution of our study is the **decomposition of the reasons behind the erroneous reportings under falsification attack settings**. We evaluate our scheme via extensive simulations and we find that under a large variety of attacks, our system is able to contain the effect of adversaries by significantly reducing the miss detection rate, which in turn decreases the interference imposed to the primary users. In addition, DTCS^3 reduces the false alarm rate, thus, improving the performance of the secondary user network. While this work focuses on cooperative spectrum sensing, its **scope is much broader**. It sets a novel paradigm on decoupling wireless induced effects from other system/user attributed that can affect network operations. This decoupling can be very beneficial for many network functionalities with degraded performance (e.g., non-delivering routing).

The rest of the paper is organized as follows. Section II discusses related to our work studies, while Section III presents the cooperative sensing model and the different attack strategies we consider. Section IV presents DTCS^3, our proposed scheme. Finally, Section V presents our simulation results, while Section VI concludes our study.

II. Related Work

In this section we will review some representative studies, which are directly related to our work.

A. Non-Cooperative Detection

Non-cooperative detection is also referred to as *local spectrum sensing*. Sensing is accomplished by a single node without cooperating with other secondary users. Energy based detection is the most widespread way to determine the availability of the channel. In energy detection, the node monitors the received energy over certain time period. Comparing the observed value with a predefined energy threshold, the secondary user decides the availability of the spectrum. Urkowitz [6] analyzes the characteristic of the simple energy detector under non-fading channel. In particular, when the primary user is absent, the detection output exhibits a central Chi-square distribution. When the primary user is present, the detected energy is non-central Chi-square distributed. Digham *et al.* [7] study the energy detection performance over fading channels. They provide the theoretical probabilities of miss detection and false alarm under different channel models and benign settings.

Matched filter detection [8], [9], cyclostationary feature detection [10], [11], [12] and eigenvalue-based detection [13] comprise alternative techniques to energy detection. These schemes can provide better performance as compared to energy sensing under specific settings and assumptions. For instance, while matched filter detection outperforms the energy detector, it requires a priori knowledge of the primary users signal "shape". However, energy detection is still the most commonly used technique due to its low computational complexity and its practicality.

B. Cooperative Detection

Compared with local spectrum sensing, cooperative detection is able to provide more reliable result [14]. Ghasemi *et al.* [15] show that under fading channels, the detection performance can be significantly improved by employing collaborative sensing even when individual users utilize simple energy detectors for their decisions. Taricco [16] considers a cooperative sensing approach in which the local decisions are linearly combined to provide the final, global decision. He provides a way to obtain the optimal coefficients for the linear combination of the local sensor readings under the assumption of constant SNR at the secondary users. Furthermore, a cluster-based cooperative spectrum sensing method is proposed by Sun *et al.* [17]. They propose to divide the secondary users into different clusters. The final decision is then based on the most favorable user's output in each cluster. Finally, Huang *et al.* [18] assuming different average SNR for each node, incorporate fading into the collaborative spectrum sensing under benign settings.

C. Trust Based Cooperative Detection

While collaborative sensing provides multiplexing gains and improves accuracy, it raises security concerns. In particular, on the one hand malicious users can decrease the spectrum utilization from secondary users by always reporting the presence of a primary user. On the other hand, they can interfere with the primary users' system by reporting the former's absence leading to secondary users' interfering transmissions. Of course, the exact fusion algorithm employed dictates the level of vulnerability. Traditionally, reputation-based schemes have been proposed for alleviating the effects of similar falsification attacks.

Qin *et al.* [19] propose a trust value calculation algorithm based on the Beta reputation model [20]. They assign different trust values to every secondary user based on the historical local sensing results for each node. They also use a sliding window scheme, where the latest observation has the highest contribution to the calculation of the reputation value. Chen *et al.* [21] propose the use of a Weighted Sequential Probability Ratio Test (WSPRT) for the trust based cooperative sensing. Their reputation based scheme is proved to be robust against

[3]The terms trust, reputation and reliability will be used interchangeably in the rest of the paper.

the Byzantine failure problem. In their trust value assignment algorithm, if a node frequently generates inconsistent results with the final decision, its reputation value will eventually reduce to zero. The proposed scheme exhibits good performance but requires the knowledge of the a priori probabilities that a primary user is active. Wang *et al.* [22] introduce the notion of consistency on trust values. While the trust value indicates the average "performance" of a node with regards to its reliability, the consistency value captures its stability. If both trust and consistency values of a user are below predefined thresholds, i.e., his trust is consistently low, its locals sensing report will not be considered in the final decision.

Sensing can also be disassociated from the radios as proposed by Weiss *et al.* [23]. From a security perspective, this approach has merits in that the sensors in such a setting might be more effectively secured, so the reports they send might be deemed more trustworthy. While the detailed evaluation of this approach is beyond the scope of this paper, it remains a potentially interesting alternative for acquiring stochastic spectrum holes.

We would like to emphasize on the fact that the notion of consistency in [22] is different from the capability value we define in the following section. While one can think of this consistency value as capturing the intricacies of the wireless medium (e.g., frequency changes), this is only in approximation and indirectly. In our work, we clearly distinguish the reasons behind an erroneous report. We take a holistic approach, designing a vigorous cooperative spectrum sensing scheme, towards a new paradigm of security assured systems that can deliver high performance as well.

III. System Model

In this section we will introduce the collaborative spectrum sensing model and the falsification attacks we consider in this work.

A. Collaborative Spectrum Sensing

As per the cognitive radio network paradigm, no secondary user is allowed to contend with the primary user while the latter is transmitting. Secondary users are only able to access the medium when the primary user does not use the channel. To ensure this, all secondary users need to know whether the spectrum is occupied by the primary user. To reiterate, collaborative spectrum sensing is based on the cooperation between the secondary users to determine the channel status (busy or idle). As explained, this significantly increases the network performance (under benign settings).

In collaborative spectrum sensing, each secondary user performs local spectrum sensing independently, and uploads its sensing result to a *Data Fusion Center (DFC)*. The DFC can be either one of the secondary users or a dedicated external unit. The sensing reports can either be binary (0-1 local decisions) or the actual local energy measurements (e.g., energy values expressed in dBm). The former case corresponds to the *hard decision* schemes, while the latter belongs to the *soft decision* schemes. Upon the reception of all the reports from the secondary users for a given time slot t, the DFC makes the final, binary decision of whether the channel is occupied or not. Once the decision is made, DFC informs the secondary users of the decision.

Data Fusion Rule is the algorithm that DFC uses for combining all the local sensing reports from secondary users to make the final sensing result. There are different kinds of fusion rules that can be used depending on the type of decision scheme used (hard versus soft):

1) Hard Decision: As aforementioned, when hard decision is utilized, the inputs to the DFC are one bit local decisions ('0' or '1') from all secondary users. The most widely used hard decision-based fusion rules are:

- *"OR" Rule:* the final output of the DFC will be '1' (medium is busy) if one or more of the reported local sensing results is '1'. Otherwise the final decision will be '0' (idle medium).
- *"AND" Rule:* the final output of the DFC will be '1' only when all the reported local sensing results are '1'. Otherwise the final decision will be '0'.
- *"Majority" Rule:* the final output of the DFC will be '1' if there are more '1's than '0's among all the reported local sensing results. Otherwise the final decision will be '0'.

Besides the simple rules above, more complicated algorithms, such as the Neyman-Pearson Detection [24], the Sequential Probability Ratio Test [24], and the Dempster-Shafer Theory of Evidence [25] are also applicable for hard decisions. Since hard decision only requires 1-bit from each secondary user, it keeps the communication overhead minimal and simplifies the computation. Hence, both faster decisions and lower energy consumption are viable.

2) Soft Decision: Soft decision-based schemes are more sophisticated compared with the hard decision ones. A DFC that uses soft decision requires the original energy readings from all secondary users. Using this more detailed input information, the DFC is able to provide more accurate decisions. However, this happens at the cost of higher complexity and a slightly increased communication overhead. There exists a significant volume of literarture on soft decision-based schemes[4]. However, hard decision systems have gained more attention, mainly due to their simplicity. Hence, in this work, we focus on hard decision as well.

B. Spectrum Sensing Data Falsification (SSDF)

The higher sensing accuracy possible with cooperative spectrum sensing comes at the cost of higher vulnerability in secondary users' misbehaviors. In particular, *Spectrum Sensing Data Falsification (SSDF)* attacks can diminish any performance gains obtained from the collaboration among the non-malicious secondary users. A malicious user launches an SSDF attack by sending false sensing report to the DFC. The behaviors of an SSDF attacker can be classified into the following categories:

[4]A few references for the interested reader can be found in [26], [27], [28] and [29].

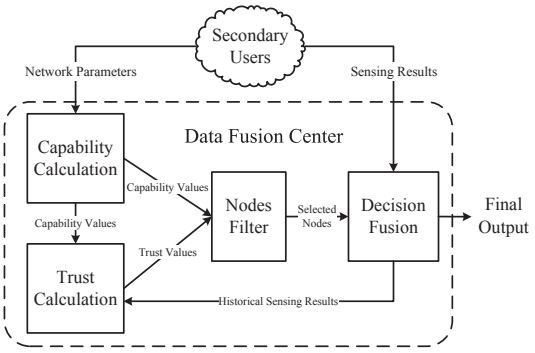


Fig. 1. The architecture of DTCS^3 based data fusion center.

- *"Always Yes" attack:* the attacker always reports that a primary user exists.
- *"Always No" attack:* the attacker always reports that there is no primary user.
- *"Always False" attack:* the attacker always reports the opposite of its local spectrum sensing result.
- *Random/Intermittent attack:* the attacker acts intermittently. He alternates between active and inactive periods. During his active periods he can launch any type of the above three attacks. The fraction of the active time of a random attacker, is referred to as *attack ratio*.

Under the SSDF attack both false alarm and miss detection rates may increase on the DFC side. False alarm occurs when the DFC decides that the channel is occupied, while in fact there is no primary user active. This prevents all the secondary users from accessing the medium, resulting in the waste of spectrum resources and poor performance of the secondary user network. When miss detection occurs, the DFC announces the channel to be available when in reality a primary user is already using it. In this case, interference to the licensed user will be introduced if some of the secondary users try to access the spectrum.

In this paper, we focus on the most widely considered hard decision scheme. However, note here that, the general framework can be applied to soft decision schemes with the appropriate modifications. Furthermore, we examine the "Majority" data fusion rule, since "OR" and "AND" rules are not very practical due to their high vulnerability; even a single attacker can significantly degrade the performance. Finally, we consider all four attack types aforementioned.

IV. Proposed Decoupling Scheme

In this section, a novel collaborative spectrum sensing mechanism, Decoupling Trust and Capability Spectrum Sensing System (DTCS^3) is presented. It explicitly decouples the trust and the sensing capability of a node, since the wireless channel conditions can significantly impact the sensing results. As we will show DTCS^3 is robust against SSDF attacks.

DTCS^3 in a nutshell: The overall picture of our framework is shown in Fig.1, where the DFC is divided into four modules. The Capability Calculation unit obtains network parameters (e.g., a secondary users' SNR as we will see in what follows) and computes the capability values of secondary users. The calculated capability value, together with the historical sensing results, form the necessary input to the Trust Calculation unit. The Nodes Filter module considers both capability and trust values to select which secondary users will be used for the final decision fusion. The Decision Fusion utilizes a certain fusion rule to obtain the final sensing decision based only on the local sensing results of the selected nodes. The Decision Fusion provides not only the final output, but also feedback of the historical sensing results to the trust calculation unit. In what follows, we will describe the algorithms of every component of our framework in detail.

A. Capability Calculation

The capability value for a secondary user represents its ability to correctly provide the local sensing result. We define a pair of capability values for the secondary user i at time slot t, denoted by $c_i^0(t)$ and $c_i^1(t)$, for the idle channel scenario and the busy channel scenario, respectively. The definitions of the capability values are

$$c_i^0(t) = P(o_i(t) = 0|H_0), \quad (1)$$
$$c_i^1(t) = P(o_i(t) = 1|H_1), \quad (2)$$

where H_0 and the H_1 are the hypotheses of idle channel and busy channel, respectively, and $o_i(t)$ is the binary local observation value for node i at time slot t.

Suppose that the secondary users apply energy detection to perform the spectrum sensing. A simple energy detector diagram is shown in Fig.2, where T is the observing time period. The input signal of the detector is firstly processed by the noise pre-filter. If the central frequency is f_0 and the bandwidth is W, after the pre-filter, only the signal within the band $[f_0 - \frac{W}{2}, f_0 + \frac{W}{2}]$ will be kept. The output signal of the noise pre-filter for node i is denoted by $y_i(t)$. The expression of $y_i(t)$ equals

$$y_i(t) = h \cdot s_i(t) + n_i(t), \quad (3)$$

where $h = 0/1$ under hypothesis H_0/H_1, $s_i(t)$ is the received signal from the primary user and $n_i(t)$ is the additive white Gaussian noise (AWGN). $y_i(t)$ is squared and integrated over the time period T. Then the output signal is normalized by $\frac{N_i(t)}{2}$, where $N_i(t)$ is the one-sided power spectral density of the noise for node i at time slot t, and the final output from the multiplier is denoted by $V_i(t)$. In hard decision, $V_i(t)$ is compared with a given threshold, say ν, to obtain the binary local spectrum sensing result ($o_i(t) = 0$ or $o_i(t) = 1$). To reiterate, Urkowitz [6] has shown that $V_i(t)$ have central chi-square distribution with $2a$ degree under H_0 and noncentral chi-square distribution with $2a$ degree and a non-centrality parameter of $2a\gamma_i(t)$ under H_1, where $a = TW$ and $\gamma_i(t)$ is the signal-to-noise ratio for node i at time slot t. Furthermore,

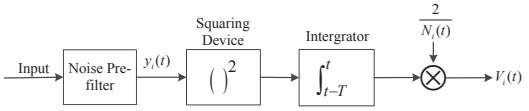


Fig. 2. Block diagram of the energy detector used by the secondary user

the probability density function of $V_i(t)$ can be written as [7]:

$$f_{V_i(t)}(x) = \begin{cases} \frac{x^{a-1}e^{-\frac{x}{2}}}{2^a\Gamma(a)}, & H_0 \\ \frac{1}{2}(\frac{x}{2\gamma_i(t)})^{\frac{a-1}{2}} e^{-\frac{2\gamma_i(t)+x}{2}} I_{a-1}(\sqrt{2\gamma_i(t)x}), & H_1 \end{cases} \tag{4}$$

where $\Gamma(.)$ is the gamma function [30] and $I_z(.)$ is the z^{th}-order modified Bessel function of the first kind [31].

Under H_0, the probability that node i outputs a correct result at time slot t equals:

$$P(o_i(t)=0|H_0) = P(V_i(t) < \nu|H_0) = 1 - \frac{\Gamma(a, \frac{\nu}{2})}{\Gamma(a)} = c_i^0(t), \tag{5}$$

where $\Gamma(\cdot,\cdot)$ is the incomplete gamma function [30]. Under H_1, the probability that node i outputs a correct result at time slot t equals:

$$\begin{aligned} P(o_i(t)=1|H_1) =& P(V_i(t) > \nu|H_1) \\ =& Q_a(\sqrt{2\gamma_i(t)}, \sqrt{2\nu}) \\ =& c_i^1(t), \end{aligned} \tag{6}$$

where $Q_a(\cdot,\cdot)$ is the generalized Marcum Q-function [32].

In a real network, the SNR is measured at the receiver (i.e., the secondary users) and will be collected by the DFC along with its local sensing result for calculating the above probabilities and hence, the corresponding instantaneous capability values. Note here that, in this paper, we only consider the impact of channel irregularities on the decision of the nodes' local sensing functionality. On the contrary, the communication channel between the DFC and any secondary user is assumed to be ideal. In other words, a transmission from a secondary node to the DFC will always be successful. This is a rather realistic assumption; the amount of the control information trasmitted from a secondary user to the DFC is small and hence, less prone to interference.

As it might be evident, since the SNR is reported by the secondary users, a (malicious and/or selfish) secondary user can report fake SNR values to disrupt the above calculations. However, there is no motivation for a malicious users to adopt such a strategy. If a malicious user reports SNR which is higher than the real value, it will be assigned lower trust value due to its malicious attack behavior. If it uploads lower SNR, its capability value will be degraded and its sensing result will not be used by the DFC based on our node filter strategy either. In our evaluations in Section V we further examine this issue in more detail.

B. Trust Calculation

The reporting trust value of a secondary user, is the probability that it will provide to the DCF its actual sensing reading (regardless if it is the correct one or not). As alluded to above, the majority of the existing trust calculation algorithms utilize only the sensor's historical performance, *punishing* nodes whose reports do not match the final outcome of the DCF, with low trust values. Nevertheless, to reiterate, this is not fair since a completely trustworthy node with a poor sensing capability will be erroneously assigned a low trust as well. In our scheme, we calculate the trust value considering both the historical performance and the node's capability.

In order to calculate the trust value for a node, say node i, we still need historical performance information (evidence set). We denote this set at time slot t as $\Omega_i(t)$, and each sample[5] point is a tuple that contains the DFC's output, which is considered as the ground truth, and the nodes' sensing report. Let us assume that there are in total k samples in the evidence set and the sample interval is τ. Then at time t, the data within the sample set correspond to the last evidence collection interval window $[t-k\tau, t-\tau]$ and we have:

$$\Omega_i(t) = \left\{ \begin{array}{c} < O(1), o_i(1) >, \\ ... \\ < O(k), o_i(k) > \end{array} \right\} \tag{7}$$

where $O(j)$ and $o_i(j)$ are the j^{th} DFC's and node i's output respectively during the last training interval window. Note here that for notation simplicity we are using the sample index (j), rather than the actual time ($t - j\tau$). We further map every evidence tuple i to a binary g_i such that:

$$g_i(j) = \begin{cases} 1, & \text{if } o_i(j) = O(j); \\ 0, & \text{if } o_i(j) \neq O(j). \end{cases} \tag{8}$$

For a given j, $g_i(j)$ is a Bernoulli trial and we define its probability of success as $r_i \cdot w_i(j)$, where r_i is the node's trust value (to be estimated) and $w_i(j)$ is the (weighted) capability value (formally defined later using the raw capability values $c_i^x(j)$). Hence, the probability density function of $g_i(j)$ can be expressed as:

$$f(g_i(j) = x) = (r_i \cdot w_i(j))^x (1 - r_i \cdot w_i(j))^{1-x}, \tag{9}$$

where $w_i(j)$ is defined as:

$$w_i(j) = \begin{cases} c_i^1(j), & \text{if } o_i(j) = 1, \\ c_i^0(j), & \text{if } o_i(j) = 0, \end{cases} \tag{10}$$

where $c_i^0(j)$ and $c_i^1(j)$ are the raw capability values of node i (Section IV-A).

To obtain the trust value r_i, we apply Maximum Likelihood Estimation (MLE) [33]. In particular, assuming a constant trust value through our evidence collection window, we obtain the trust value of node i as the solution of the following optimization problem:

$$\text{maximize} \quad \frac{1}{k} \log(\prod_{j=1}^{k} f(g_i(j)|r_i)) \tag{11}$$

$$\text{subject to} \quad l_i \leq r_i \leq 1 \tag{12}$$

[5]We will use the terms evidence, sample and observation interchangeably in the rest of the paper.

where l_i is the lower bound for node i's trust value based on the sample set $\Omega_i(t)$. By defining:

$$L_1 = \frac{1}{k}\sum_{j=1}^{k} g_i(j) \tag{13}$$

$$L_2 = \prod_{j=1}^{k}(1 - w_i(j)), \tag{14}$$

l_i will then be:

$$l_i = \begin{cases} L_1, & \text{if } L_1 > 0; \\ \dfrac{L_2}{2}, & \text{if } L_1 = 0. \end{cases} \tag{15}$$

The percentage of the positive samples in $\Omega_i(t)$ (i.e., L_1) is the minimum trust that we can have on node i. However, in brief (more details are given in the Appendix), when this percentage is 0, the minimum trust on i is given by assuming that all the samples are negative due to wireless induced effects (i.e., L_2). Finally, the trust values are updated every k samples; once a full observation set is collected we re-run MLE to obtain a new trust estimation.

Estimation Error of the Maximum Likelihood Estimation: As with every statistical inference technique, MLE is associated with an estimation error. However, it has been proven that the maximum likelihood estimator is *unbiased*, i.e., converges to the true value with large sample sizes and its variance achieves the Cramér-Rao lower bound [34]. This means that there exists no other unbiased estimator that has (asymptotically) lower mean squared error. In particular, the variance of our maximum likelihood estimator is given by:

$$\begin{aligned} var(r_i^{ML}) &= (I(r_i))^{-1} \\ &= \left(\frac{\partial^2(\frac{1}{k}\log(\prod_{j=1}^{k} f(g_i(j)|r_i)))}{\partial r_i \partial r_i'} \right)^{-1}, \end{aligned} \tag{16}$$

where r_i^{ML} denotes the maximum likelihood estimator, and $I(r_i)$ is the Fisher information [34] of r_i. Furthermore, as k increases, the maximum likelihood estimator is normally distributed [34], that is

$$r_i^{ML} \overset{a}{\sim} N(r_i, (I(r_i))^{-1}), \tag{17}$$

where $N(x, y)$ denotes a normal distribution with mean value x and variance y.

However, in reality, it is impossible to estimate the trust value via infinite samples for various reasons. For instance, if we were to use an arbitrarily large sample size, there would be an infinitely large delay associated with the decision. Nevertheless, even if this was not a major concern, and for instance one could use sample sizes of the order of $k \approx 100$, some of the older observations might be deemed stale[6]. Hence, it is desirable to use small sample sizes. In our evaluations (see Section V), we examine this issue further and we show that, even though we use finite samples to perform the estimation, our estimation can still recover the actual trust values with small error.

[6]Note that the central limit theorem will still be applicable with a good approximation.

C. Nodes Filter

When both capability and trust values are computed, we can choose the nodes whose local sensing results would be used in the final decision fusion. We use a two-stage algorithm for filtering the nodes.

Initially, we remove the nodes that have low trust values. First, we normalize all the trust values of the secondary users using the following transformation:

$$r_i'(t) = \begin{cases} 1, \text{ if } \max_i(r_i(t)) - \min_i(r_i(t)) \le \varepsilon; \\ \frac{r_i(t) - \min_i(r_i(t))}{\max_i(r_i(t)) - \min_i(r_i(t))}, \quad \text{otherwise.} \end{cases} \tag{18}$$

where $r_i(t)$ is the trust value calculated for node i at time slot t, ε is a small positive number close to zero and $\max_i(r_i(t))/\min_i(r_i(t))$ are the maximum/minimum values among all secondary users' trust values at time slot t. The normalization process makes the *raw* trust values of the nodes more distinct, thus, helping DTCS^3 to separate trustworthy and non-trustworthy nodes with higher precision. Consequently, any node whose normalized trust value is less than λ_r ($0 \le \lambda_r \le 1$) will be filtered out.

From the remaining nodes we remove those with low capability values. Similarly, we first normalize the capability values using a similar transformation:

$$c_i^{0'}(t) = \begin{cases} 1, \text{ if } \max_i(c_i^0(t)) - \min_i(c_i^0(t)) \le \varepsilon; \\ \frac{c_i^0(t) - \min_i(c_i^0(t))}{\max_i(c_i^0(t)) - \min_i(c_i^0(t))}, \quad \text{otherwise.} \end{cases} \tag{19}$$

$$c_i^{1'}(t) = \begin{cases} 1, \text{ if } \max_i(c_i^1(t)) - \min_i(c_i^1(t)) \le \varepsilon; \\ \frac{c_i^1(t) - \min_i(c_i^1(t))}{\max_i(c_i^1(t)) - \min_i(c_i^1(t))}, \quad \text{otherwise.} \end{cases} \tag{20}$$

Since each secondary users is associated with a tuple of capability values, we only retain the nodes for whom both normalized capability values are above the threshold λ_c ($0 \le \lambda_c \le 1$).

D. Decision Fusion

As previously discussed there are several fusion rules that can be used. Our design choice for DTCS^3 is to use the majority rule. As aforementioned, both AND and OR rules are vulnerable even under the presence of a single malicious node and this can lead to significant system degradation even when any trust/reputation system is in place. Formally, using the notation introduced, the majority rule is expressed as:

$$O(t) = \begin{cases} 1, & \text{if } \sum_i o_i(t) > \sum_i (1 - o_i(t)); \\ 0, & \text{if } \sum_i o_i(t) \le \sum_i (1 - o_i(t)). \end{cases} \tag{21}$$

Note that, our selections in this paper for all the components do not restrict the compatibility of the DTCS^3 framework, and any other algorithm for the same purpose can be used to replace the corresponding algorithm mentioned above while the whole system is still DTCS^3 based.

V. Evaluations

A. Simulation Setup

In our simulations we consider 40 secondary users, randomly located in the vicinity of the primary user. In particular, the secondary users are constrained within a $2km \times 2km$ square which is 20 km away from the primary transmitter. The network topology is shown in Fig.3.

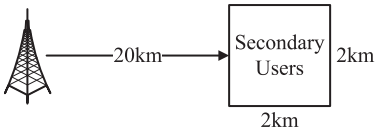


Fig. 3. Simulation network deployment.

We use the Okumura-Hata model [35] to calculate the path loss. The transmitter and receiver antenna heights are set to 100m and 1.6m, respectively. The central frequency of the primary user is 700MHz and the bandwidth equals 6MHz. The primary user's transmission power is 100kW. The multipath fading is assumed to be Rayleigh distributed, while the shadow fading follows a log-normal distribution with a standard deviation of 5.5dBm. We use the minimum signal level for TV receivers (-83dBm) [36] as the primary user detection threshold. The environmental noise is assumed to be white Gaussian. The average noise power level is set to the value that makes the false alarm rate of a secondary user equal 0.1, which is the maximum requirement of IEEE 802.22 [37]. Each experiment lasts for 500 simulated minutes while the sensing period is 10 seconds. The primary user is active with 50% possibility and its occupation time is exponentially distributed with a mean value of 1 minute. The training period is set to 3 minutes, which allows for $k = 18$. All the results are the average values based on 5 discrete runs of each scenario.

To simulate the temporal variations of the wireless signals, we sample the above distributions every time slot. Fig.4 depicts the SNR range of every node during a representative simulation run.

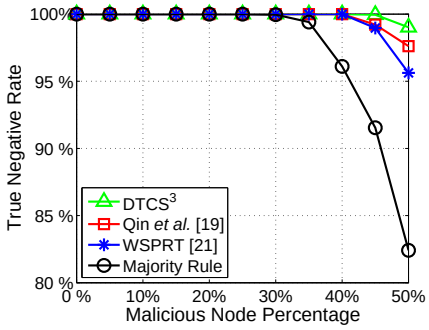
Our main evaluation metrics are the true positive (i.e., complementary of miss detection) and true negative rates (i.e., complementary of false alarm) as we vary the ratio of malicious secondary users from 0% (benign settings) to 50%.

We compare DTCS^3 with three existing schemes: (1) the trust aware hybrid spectrum sensing scheme by Qin *et al.* [19]; (2) the Weighted Sequential Probability Ratio Test (WSPRT) system proposed by Chen *et al.* [21] and (3) a plain Majority rule based on collaborative sensing system without nodes filter ("Majority Rule" for short). For all the systems, we obtain their optimal parameters through an extensive set of simulations, thus, assuring that our results correspond to the optimal performance of each system under consideration.

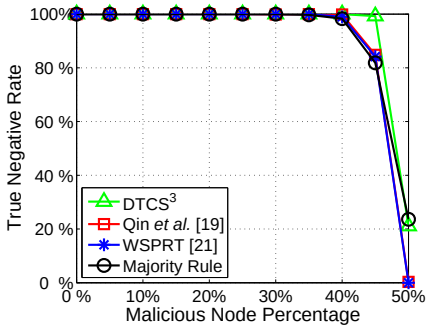
B. Efficacy of DTCS^3

In this subsection, we focus on the efficacy of our system under a variety of SSDF attack strategies. For all the attack strategies, we try both constant and random attacks while the attack ratio of the random attack is set to 75%[7].

"Always Yes" Attack: The purpose of this attack is to deceive the DFC into believing that the channel is occupied. Hence, if there is actually no primary user, the spectrum resources are wasted. In other words, under "Always Yes" attack, the false alarms increase as compared to the benign case. However, note here that, there is no effect on the missed detections. When a primary user is active, the malicious node reports its presence. Hence, we examine only the true negative rate under this scenario.



(a) Random



(b) Constant

Fig. 5. True negative rate for different systems under "Always Yes" attack

Fig.5 shows that, as one might have expected, the attack has higher impact on the correct sensing rate when the malicious node ratio increases. However, when the malicious node ratio increases beyond 50% all the schemes perform poorly especially for the constant attack mode. This is due to the fact that the DFC's output is considered as the ground truth. Therefore, when initially the number of malicious nodes is larger than that of the benign DFC is forced to provide an erroneous decision and punish the honest nodes. This initial error carries over at the consequent estimations. All the schemes perform better in the random attack case since all the random attackers launch their malicious behavior independently and they may not be active simultaneously. Hence, the *effective* percentage of attackers is lower.

"Always No" Attack: The purpose of this kind of attack is to deceive the DFC into believing that the channel is idle.

[7]Results for other attacks are similar and omitted.

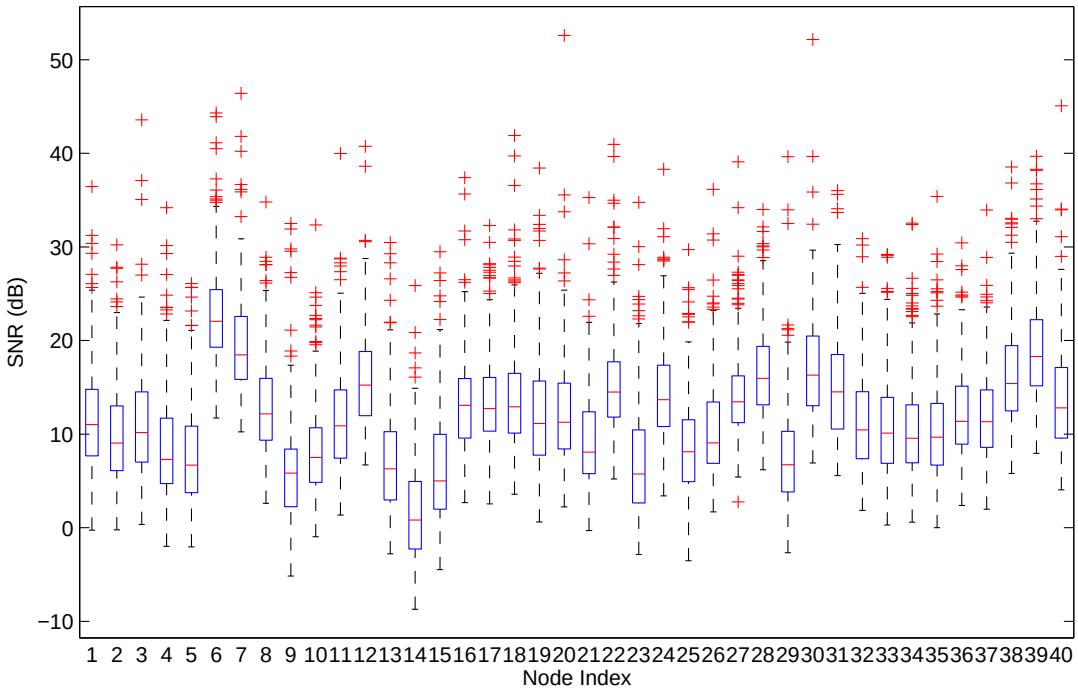


Fig. 4. The box plot of SNR for all the nodes.

Therefore, secondary users will try to access the spectrum, causing interference to the licensed users. This type of attack is potentially more harmful, since it affects the transmission of the primary users and might violate any service level agreement. However, note here that, there is no effect on the false alarms produced by the system. Hence, we examine only the true positive rate under this attack.

As it is evident from Fig.6, "Always No" attack has more significant impact on the system's performance as compared with the "Always Yes". True positive rate is significantly reduced even for a moderate number of malicious sensors. In all the cases, DTCS^3 outperforms the rest of the schemes by as high as 36%. For the case of intermittent attack, just as above, the random attackers have less impact on the system performance compared with the constant ones.

Always False Attack: The goal of an always false attacker is to simultaneously decrease both the true positive and true negative rates of the DFC to the extent possible. A successful "Always False" attack will introduce interference while the primary user is present and lead to spectrum inefficiency when the primary user is absent. Hence, under this attack we will evaluate both true positive and true negative rates. Nevertheless, DTCS^3 is able to recover more erroneous detections as compared to the rest of the examined systems.

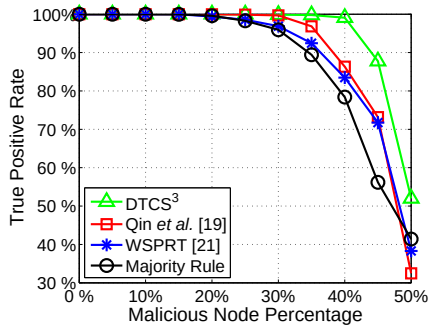
Taking a closer look at the true positive (Fig.7) and the true negative rates (Fig.8), we can see that both of them are better as compared to the single purpose attacks ("Always Yes" and "Always No"). Even when there are 45% malicious nodes, we can have rates higher than 95%; malicious nodes are very aggressive - they always report erroneous sensings - and thus all the proposed schemes rapidly reduce their trust values. Furthermore, since the sensing abilities of malicious nodes are not perfect, they sometimes provide correct results unintentionally, which leads to an acceptable performance even for the simple majority rule. This is even more pronounced in the random attack case since the attack intermittence combined with unintentional correct results can reduce the attack efficacy to a much larger extent (Figs. 7(a) and 8(a)).

C. Accuracy of Proposed Trust Estimation Algorithm

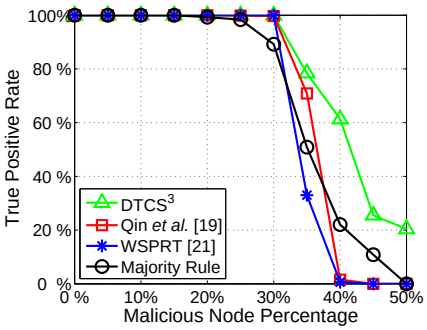
Now we turn our attention to the accuracy of the *core* module of DTCS^3, that is, the MLE-based trust estimation. To test this, we use a scenario where the malicious nodes randomly perform "Always False" attack. With the same system deployment we set nodes with ID 1 through 10 as malicious and their attack ratios are 10% through 100%, respectively. We want to evaluate whether DTCS^3 is able to capture the change in a node's attack ratio and assign suitable trust value accordingly.

Fig.9 depicts the capability and trust values inferred by our system average for each node at the end of the experiment. As we can observe, the (average) trust values estimated are very close to the complementary of the actual attack ratios.

We further calculate the cumulative distribution of the error

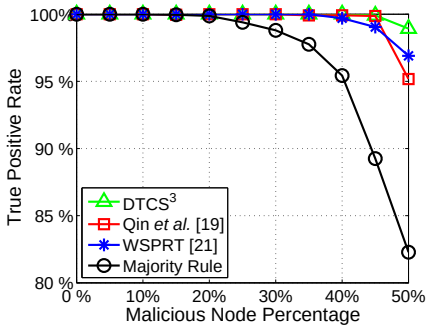


(a) Random

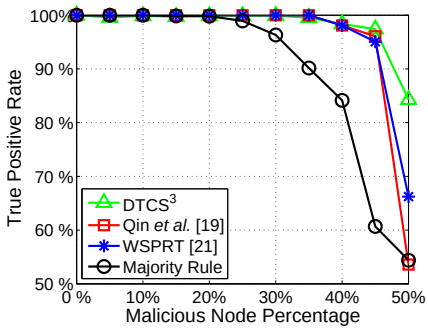


(b) Constant

Fig. 6. True positive rate for different systems under "Always No" attack



(a) Random



(b) Constant

Fig. 7. True positive rate for different systems under "Always False" attack

in the above trust estimations. The *trust error* is defined as

$$\text{trust error} = |\text{trust value} - (1 - \text{attack ratio})| \qquad (22)$$

Our results are shown in Fig.10, from which one can see that more than 90% of the trust estimations exhibit error less than 0.2.

We are also interested in the microscopic performance of DTCS^3. In particular, we seek to examine the ability of our scheme to converge to the true value of the SU's trust. For this, we track the inferred trust value for a specific malicious user and examine its convergence. Without loss of generality we monitor the trust values of node 5, whose attack ratio is 50%. The time series of the estimated trust values for this node is shown in Fig.11. In the same figure, we plot the average value of the estimated trust as computed by:

$$g(t) = \frac{1}{t} \sum_{\tau \in \{1,\ldots,t\}} trust(\tau). \qquad (23)$$

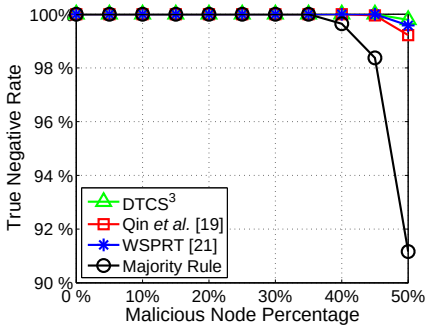
As we can see, after approximately 100 samples, the trust value converges to a value very close to the real one (i.e., 1-attack ratio).

Finally, we want to examine whether DTCS^3 can follow the dynamic behavior of a malicious node (i.e., one that has variable attack ratio with time). In our experiment, we set nine malicious nodes perform constant "always false" attack and one node perform random attack with variable attack ratio. We set the change period of attack ratio to 30 minutes. The simulation results are shown in Fig.12. As we can see, our scheme can *follow* the attack ratio change fairly accurate and fast.

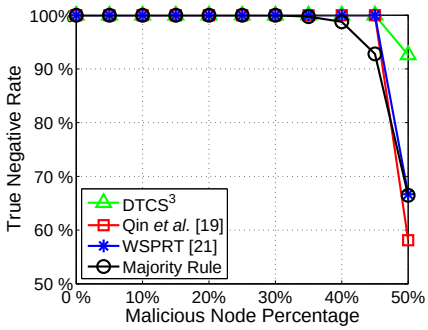
D. Fake SNR Attack

Since the SNR value is reported by the secondary user, a malicious node is able to upload fake SNR information to the DFC to affect the capability values' calculation. However, this is not of its benefit. As we discussed in Section IV-B, if the node sends an SNR value higher than the real one, its trust value will be degraded very fast due to its bad performance. On the other hand, if the node sends lower SNR values, this node will be filtered out from the final data combination due to the low capability value and hence, it will not be able to affect the decision.

To support our above claim, we run a series of simulations where attacking nodes report fake SNR values. Two types of fake SNR attacks are involved. In *fake high SNR* attack mode, a malicious user always reports an SNR which is 20dB higher than its real sensing value to the DFC. And the nodes performing *fake low SNR* attack will upload SNR values 20dB lower than the real values. Fig.13 through Fig.16 depict the results. From the figures, we can see malicious nodes don't benefit from reporting fake SNR and thus, they have no incentive to do so. Note here that, even though the fake SNR attack does degrade the system performance when malicious node ratio equals 50%, this would not be good stimulation for launching this kind of attack since in a real system malicious nodes are usually fewer than the benign ones.



(a) Random



(b) Constant

Fig. 8. True negative rate for different systems under "Always False" attack

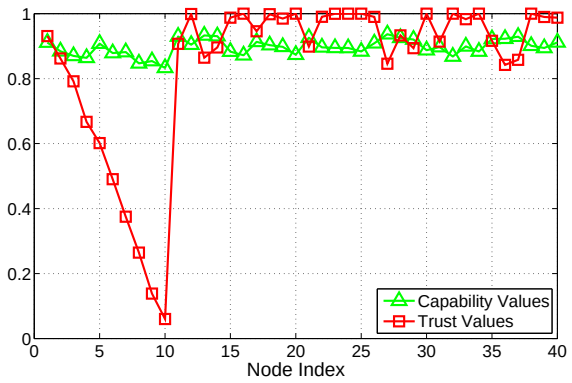


Fig. 9. Trust and Capability values for each node.

Fig. 10. Cumulative Trust Value Distribution.

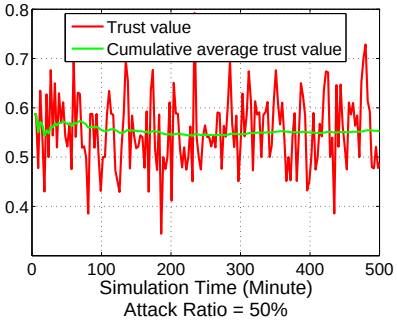


Fig. 11. Trust Value Convergence

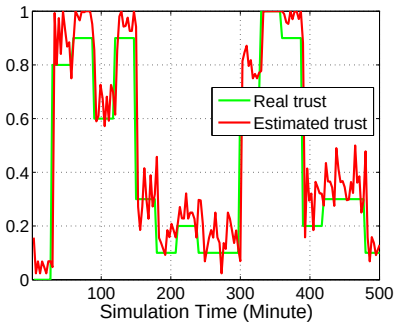


Fig. 12. Trust value variation as the attack ratio changes

E. Discussion and Scope of Our Work

The performance of the system can be significantly improved if there is feedback from the network on the actual ground truth, i.e., was or not the primary user active at the previous timeslot. Such feedback would drastically increase the accuracy of DTCS^3 calculations. However, since this might be hard in a real system we have not considered it.

Furthermore, while our simulations included only static scenarios, we would like to emphasize on the fact that DTCS^3, performs similarly under mobile nodes as well. The mobility of a secondary user will change its capability value (due to the spatio-temporal nature of the wireless medium) and our scheme is able to incorporate this in the calculations. Additionally, as we shown DTCS^3 is able to track the changes at user functional parameters.

Finally, we would like to emphasize on the fact that our work needs to be considered as a new paradigm of protocol design. Decoupling the reasons behind *failed* operations in a wireless network is crucial for achieving required levels of performance and security simultaneously. While we focused on the dynamic spectrum access problem in this work, our framework can be applied on other scenarios as well.

VI. Conclusion

Collaborative spectrum sensing has been proposed to improve accuracy and spectrum utilization in next generation, cognitive radio networks. However, different types of malicious nodes can disturb the underlying functionality causing

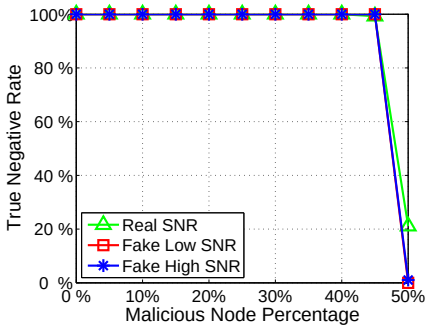


Fig. 13. True negative rate of different SNR status under "Always Yes" attack

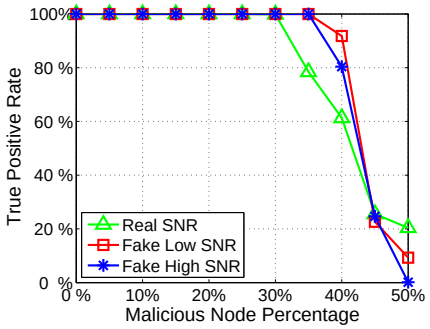


Fig. 14. True positive rate of different SNR status under "Always No" attack

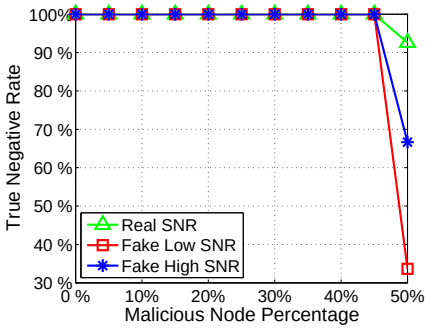


Fig. 15. True negative rate of different SNR status under "Always False" attack

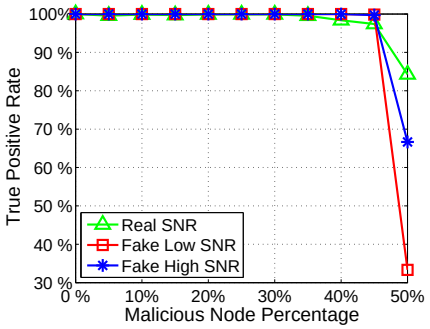


Fig. 16. True positive rate of different SNR status under "Always False" attack

degradation in the network performance. Thus, in order to enjoy the benefits possible with collaborative sensing, it is crucial to distinguish benign from malicious nodes. In addition, the wireless channel conditions can affect the sensor readings of a node and therefore need to be appropriately considered.

In this paper, we proposed Decoupling Trust and Capability Spectrum Sensing System (DTCS^3), a scheme that considers both the capability and the trustworthiness of a node to perform collaborative spectrum sensing. Our evaluations indicate that DTCS^3 exhibits a high degree of robustness under a large variety of attack scenarios. Our two-step node selection process guarantees that the selected sensors are both capable of completing this task and trustworthy. Our extensive simulations compare DTCS^3 with other existing schemes in the literature and show that our approach outperforms them under a diverse set of scenarios.

ACKNOWLEDGMENT

This work was supported in part by the U.S. National Science Foundation under Grant 1149422.

REFERENCES

[1] J. Mitola, "Cognitive radio for flexible mobile multimedia communications," in *Mobile Multimedia Communications, 1999. (MoMuC '99) 1999 IEEE International Workshop on*, 1999, pp. 3–10.

[2] "The White Space Coalition." [Online]. Available: http://www.engadget.com/tag/white%20space %20coalition/

[3] M. B. Weiss and W. H. Lehr, "Market based approaches for dynamic spectrum assignment," Working Paper http://d-scholarship.pitt.edu/2824/, 2009.

[4] M. B. Weiss, M. Altamimi, and L. Cui, "Spatio-temporal spectrum modeling: Taxonomy and economic evaluation of context acquisition," *Telecommunications Policy*, vol. 36, no. 4, pp. 335–348, 2012.

[5] K. Pelechrinis, V. Zadorozhny, V. Kounev, V. Oleshchuk, A. Mohd, and Y. Lin, "Automatic evaluation of information provider reliability and expertise," *World Wide Web Journal*, 2013.

[6] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, vol. 55, no. 4, pp. 523–531, Apr. 1967.

[7] F. F. Digham, M.-S. Alouini, and M. K. Simon, "On the energy detection of unknown signals over fading channels," *Communications, IEEE Transactions on*, vol. 55, no. 1, pp. 21–24, Jan. 2007.

[8] D. Cabric, A. Tkachenko, and R. Brodersen, "Spectrum sensing measurements of pilot, energy, and collaborative detection," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, Oct. 2006, pp. 1–7.

[9] A. Sahai, N. Hoven, and R. Tandra, "Some fundamental limits on cognitive radio," in *in Forty-second Allerton Conference on Communication, Control, and Computing*, 2004.

[10] A. Fehske, J. Gaeddert, and J. Reed, "A new approach to signal classification using spectral correlation and neural networks," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, Nov. 2005, pp. 144–150.

[11] S. Haykin, D. Thomson, and J. Reed, "Spectrum sensing for cognitive radio," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 849–877, May 2009.

[12] Y. Hur, J. Park, W. Woo, J. Lee, K. Lim, C.-H. Lee, H. Kim, and J. Laskar, "Wlc05-1: A cognitive radio (cr) system employing a dual-stage spectrum sensing technique : A multi-resolution spectrum sensing (mrss) and a temporal signature detection (tsd) technique," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, Dec. 2006, pp. 1–5.

[13] Y. Zeng and Y. chang Liang, "Eigenvalue-based spectrum sensing algorithms for cognitive radio," *Communications, IEEE Transactions on*, vol. 57, no. 6, pp. 1784–1793, Jun 2009.

[14] E. Visotsky, S. Kuffner, and R. Peterson, "On collaborative detection of tv transmissions in support of dynamic spectrum sharing," in *New*

Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on, Nov. 2005, pp. 338–345.
[15] A. Ghasemi and E. Sousa, "Opportunistic spectrum access in fading channels through collaborative sensing," *Journal of Communications*, vol. 2, no. 2, 2007.
[16] G. Taricco, "Optimization of linear cooperative spectrum sensing for cognitive radio networks," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 1, pp. 77–86, Feb. 2011.
[17] C. Sun, W. Zhang, and K. Ben, "Cluster-based cooperative spectrum sensing in cognitive radio systems," in *Communications, 2007. ICC '07. IEEE International Conference on*, Jun. 2007, pp. 2511–2515.
[18] X. Huang, N. Han, G. Zheng, S. Sohn, and J. Kim, "Weighted-collaborative spectrum sensing in cognitive radio," in *Communications and Networking in China, 2007. CHINACOM '07. Second International Conference on*, Aug. 2007, pp. 110–114.
[19] T. Qin, H. Yu, C. Leung, Z. Shen, and C. Miao, "Towards a trust aware cognitive radio architecture," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, no. 2, pp. 86–95, Sep. 2009. [Online]. Available: http://doi.acm.org/10.1145/1621076.1621085
[20] B. E. Commerce, A. Jsang, and R. Ismail, "The beta reputation system," in *In Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.
[21] R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, Apr. 2008, pp. 1876–1884.
[22] W. Wang, H. Li, Y. Sun, and Z. Han, "Attack-proof collaborative spectrum sensing in cognitive radio networks," in *Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on*, Mar. 2009, pp. 130–134.
[23] M. B. Weiss, S. Delaere, and W. H. Lehr, "Sensing as a service: An exploration into the practical implementation of dsa," in *IEEE DySPAN*, 2010.
[24] P. K. Varshney, *Distributed Detection and Data Fusion*, 1st ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996.
[25] G. Shafer, *A Mathematical Theory of Evidence*. Princeton: Princeton University Press, 1976.
[26] J. Ma, G. Zhao, and Y. Li, "Soft combination and detection for cooperative spectrum sensing in cognitive radio networks," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 11, pp. 4502–4507, Nov. 2008.
[27] P. Kaligineedi, M. Khabbazian, and V. Bhargava, "Malicious user detection in a cognitive radio cooperative sensing system," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 8, pp. 2488–2497, Aug. 2010.
[28] A. Min, K. Shin, and X. Hu, "Secure cooperative sensing in ieee 802.22 wrans using shadow fading correlation," *Mobile Computing, IEEE transactions on*, vol. 10, no. 10, pp. 1434–1447, Oct. 2011.
[29] A. Min, K.-H. Kim, and K. Shin, "Robust cooperative sensing via state estimation in cognitive radio networks," in *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symposium on*, May 2011, pp. 185–196.
[30] I. S. Gradshteyn and I. M. Ryzhik, *Table of integrals, series and products*, Gradshteyn, I. S. & Ryzhik, I. M., Ed. Academic Press, 1994.
[31] G. Watson, *A treatise on the theory of Bessel functions*, 2nd ed. Cambridge University Press, 1995.
[32] A. Nuttall, "Some integrals involving the function (corresp.)," *Information Theory, IEEE Transactions on*, vol. 21, no. 1, pp. 95–96, Jan. 1975.
[33] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall London, 1993.
[34] W. Gould, J. Pitblado, and W. Sribney, *Maximum likelihood estimation with Stata*. Stata Press, 2005.
[35] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.
[36] D. Cabric, S. Mishra, and R. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, vol. 1, Nov. 2004, pp. 772–776.
[37] "IEEE 802.22 Working Group on Wireless Regional Area Networks," http://www.ieee802.org/22/.

APPENDIX

Calculation of l_i when $L_1 = 0$: When $L_1 = 0$, all the reports of the node under consideration (say i) do not align with those of DFC. In order to calculate the minimum trust on this node, we first need to compute the probability x that all the reports *failed* due to wireless effects (the trust on the node is assumed here to be equal to one). This probability is equal to $x = \prod_{j=0}^{k-1}(1 - w_i(j))$. If this probability is equal to zero (which means that the wireless environment is *perfect*), then all the transactions must have failed due to node and thus $r_i = 0$. As the probability x increases, our minimum trust on i also increases. This is because the sensing might have legitimately failed and thus, we cannot penalize the node without *knowing* why they failed with certainty. However, even if all reports were opposite due to wireless induced effects (i.e., $x = 1$), it may still not mean that the node is completely trustworthy. In fact, we do not know anything about him in this case and we should have $r_i = 0.5$. Hence, considering r_i to be a function of x, e.g., $f(x)$, we have f going through the points $(x_1, f(x_1)) = (0, 0)$, and $(x_2, f(x_2)) = (1, 0.5)$. Assuming for simplicity that f is linear we finally get: $l_i = \prod_{j=0}^{k-1}(1 - w_i(j)))/2$, when $L_1 = 0$.