Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2011, Article ID 242304, 10 pages
doi:10.1155/2011/242304

Research Article

Modeling the Lion Attack in Cognitive Radio Networks

Juan Hernandez-Serrano,[1] Olga León,[1] and Miguel Soriano[2]

[1] *Department of Telematics Engineering, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain*
[2] *Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), 08860 Barcelona, Spain*

Correspondence should be addressed to Olga León, olga@entel.upc.edu

Received 1 June 2010; Accepted 23 July 2010

Academic Editor: Christos Verikoukis

Cognitive radio is a promising technology aiming to improve the utilization of the radio electromagnetic spectrum. A cognitive radio is a smart device which runs radio applications software to perform signal processing. The use of this software enables the device to sense and understand its environment and actively change its mode of operation based on its observations. Unfortunately, this solution entails new security challenges. In this paper, we present a cross-layer attack to TCP connections in cognitive radio networks, analyze its impact on TCP throughput via analytical model and simulation, and propose potential countermeasures to mitigate it.

1. Introduction

Traditionally, spectrum has been allocated by regulatory agencies such as the Federal Communications Commission (FCC) in a static and inefficient manner. All frequencies below 3 GHz are assigned to specific services which operate under license leading to a lack of spectrum for new wireless applications. However, recent studies show that most of the time, the allocated spectrum is vastly underutilized.

In this context, Cognitive Radio Networks (CRNs) emerge as a possible solution to solve the lack of spectrum, allowing to take profit of unused frequency bands and to improve the overall availability of the data. CRNs are composed of smart devices which can sense and identify "white spaces"—or vacant areas—in the spectrum. Based on current measurements and on that learnt in the past, such devices can intelligently adjust their transmission parameters, giving the opportunity to secondary users to make use of the spectrum left unused by licensed services or primary users. However, as primary transmissions must not be interfered, a CRN must continuously sense the medium [1] in order to detect the presence of a primary user or incumbent in the current band in use. In this case, the CRN must rapidly switch to another channel (perform a frequency handoff), leading to the temporal interruption of the CRN connections until a new channel is available. The interval of time needed until connections are resumed, that is, the handoff duration, will obviously vary depending on the number of available channels and the detection time, but typically can take values around 2 seconds [2].

The particular attributes of CRNs such as cooperative spectrum sensing, incumbent- and self-coexistence mechanisms, and so forth, raise new security implications [3, 4]. Mainly the literature has focused on three specific attacks: the Primary User Emulation (PUE) attack, the Objective Function Attack (OFA), and the specific attacks to cooperative sensing mechanisms.

The PUE attack, first coined in [1], is based on the fact that CRN devices or secondary users are only allowed to operate in licensed bands on a noninterference basis. An attacker could pretend to be an incumbent by transmitting a signal with similar characteristics to a primary signal, thus, preventing secondary users from using vacant bands.

OFAs [3] are targeted to disrupt the learning algorithm of Cognitive Radios (CR) devices. Within a CRN, incumbents control several radio parameters in order to enhance network performance. The parameters choice is often done by means of an artificial intelligence algorithm that makes slight modifications of several input factors to find their optimal values that maximize an objective or goal function. An attacker can alter the performance of the learning to its own profit by intentionally degrading (e.g., by jamming) the

channel when some input factors are greater than a certain threshold. As a naïve example, the attacker can jam the channel whenever the security of the protocol is set, and hence the learning algorithm will conclude that it is better to work without any security.

Cooperative sensing in CRNs [5, 6] allows taking a decision about the presence of a primary user in a given channel, based on the reports provided by a set of CRs. Each secondary user senses the spectrum individually and shares its results with the rest of the nodes in order to improve detection probability. As a consequence, malicious and selfish behaviors can arise, such as a malicious node which deliberately report false measurements leading to false positives or negatives or a selfish node, which do not cooperate in order to save energy, for instance.

In this paper, we first detail the Lion attack, a cross-layer attack specific to CRNs performed at the physical link layer and targeted to the Transport Control Protocol (TCP), and we introduce some potential countermeasures. Furthermore, we derive an analytical model for such attack, and we evaluate its impact both by means of simulations and with the provided analytical model. The attack, originally outlined in [7], consists of performing a PUE in order to force the CRN network to switch from one band to another (frequency handoff) with the aim of degrading the throughput of TCP connections within the CRN. This attack can turn into a permanent Denial of Service (DoS) if the attacker can predict or know the new transmissions parameters to be used by the sender after the handoff. If each time the sender switches to a new frequency and the attacker performs a PUE, the sender will not be able to send any data successfully.

The paper is structured as follows. Section 2 provides a detailed description of the attack and a set of countermeasures to mitigate its effects. Next, in Section 3, we present an analytical model of such attack. Section 4 analyzes the effect of the attack on TCP throughput via simulation and validates the analytical model presented in the previous section. Finally, in Section 5, we present the conclusions of the work.

2. The Lion Attack

2.1. Target and Motivation. The Lion attack is a cross-layer PUE-based attack targeted to the transport layer, aiming at degrading the throughput of TCP connections within a CRN. PUE attacks allow the attacker to easily force frequency handoffs which, as explained below, could have a harmful impact over the TCP throughput. The Lion attack uses PUE attacks to effectively reduce the throughput. Moreover, if the attacker knows or can guess some of the connection parameters, He or she can even perform a DoS just by emulating a primary transmission at specific instants of time which can be easily predicted (see Section 2.2). Because of this, the Lion attack is more cost effective in reducing TCP throughput that performing simple PUE attacks or just jamming.

Although frequency handoffs could also be forced by means of jamming, there are fundamental differences which may incentivize an attacker to perform specifically a PUE and not simply jam the channel.

First, a CRN is required to perform a frequency handoff upon detection of a primary transmission even if the next channel in use has worse transmission conditions. With jamming, the victim CRN may just perform the handoff if the overall transmission conditions are below a certain threshold and another better frequency channel is available. Moreover, the cost of a PUE is reduced to just transmit a signal similar to a real primary signal (television or wireless microphones) or replay a real one.

Second, with the same effort or resources, the scope of a PUE attack can be much larger. Although the fake primary transmission may be detected in a small-scoped area of the CRN, it will force a frequency handoff, thus affecting the whole CRN. By means of jamming, a small area with just a degraded communication channel should not be enough to force the CRN to perform such handoff.

From the previous arguments, an attacker has enough reasons to use Lion as the best cost effective attack in order to degrade or even starve TCP throughput over CRNs.

2.2. Attack Insights. As it is well known [8], the TCP protocol is especially sensitive to high variations of delay and bandwidth, and therefore the interruption of the transmission due to the frequency handoff can lead to a very poor performance. As the transport layer is not aware of the interruption, the TCP sender keeps sending data which is queued for transmission at lower layers. Thus, outstanding TCP segments can be delayed or even lost if the queue overflows during the process of spectrum handoff, triggering the TCP congestion control mechanisms.

TCP keeps a retransmission timer for each outstanding segment whose value is set based on Round-Trip Time (RTT) measurements performed along the connection. If the retransmission timer for a given segment expires; that is, a Retransmission Time Out (RTO) takes place and no acknowledgment has been received, it is considered to be lost, so the segment is retransmitted and the congestion window is reduced to one segment, thereby reducing its throughput [9]. The expiration of the retransmission timer can be due to the lost of a segment but also to a sudden increase in the RTT, for example, if there is a route change or, in the case of CRNs, when a spectrum handoff takes place.

Moreover, as the retransmission timer backs off (doubles its value) with each unsuccessful retransmission attempt, the TCP sender may remain inactive even after the frequency handoff has finished, since it is not allowed to transmit any data until a retransmission timer expires. Figure 1 depicts the effect of the attack, considering an initial RTO of 200 ms. A PUE is performed and after t_D s; the CRN detects the presence of a (fake) primary user and performs a frequency handoff with a duration of 1.5 s. During the handoff, as the channel is not available, the data sent by the TCP sender is not acknowledged, leading to the expiration of the retransmission timer. The first retransmission attempt is performed 200 ms after the original transmission and, since the handoff has not finished, is unsuccessful. As a consequence, the TCP sender backs off doubling its

retransmission timer and tries to retransmit the segment after $2 \cdot \text{RTO} = 400$ ms. All retransmissions matching a handoff interval will fail, triggering the backoff mechanism. In this example, the forth retransmission finally succeeds, but the TCP sender has remained inactive for at least $15 \cdot \text{RTO} = 3$ s.

The smart version of the Lion attack is based on the knowledge of the value of the retransmission timer of a TCP connection. In typical CRNs such as WRAN 802.22 [2], the RTT value for in-network communications is around some hundreds of microseconds. Although the value of the retransmission timer is variable and depends on the RTT estimations, most implementations use a minimum value for the RTO of 100 ms or 200 ms, much higher than the estimation of the RTT performed. This fact will lead the TCP sender to make use of a fixed value for the RTO, which will be doubled for each unsuccessful attempt. The attacker can take advantage of this information to force handoffs which coincide with the retransmission instants, therefore completely starving the TCP source, as shown in Figure 2.

2.3. Mitigating the Lion Attack. As explained in Section 2.2, a Lion attack forces the CRN to perform a frequency handoff, incurring a substantial delay until transmission is resumed and degrading TCP throughput. With the purpose of counteracting this attack, the CRN should be able: (1) to detect its operation and to identify/locate the attacker and (2) to provide with information about the disconnection to the transport layer so as to minimize the impact of the attack on the protocol.

Many cross-layer solutions have been proposed in the literature [10–12] to deal with typical TCP problems in wireless links, such as losses, drastic changes in routes, or temporal lost of connectivity. These approaches make TCP aware of what is happening at the physical link layers and modify its behavior to react according to network conditions, thus improving its performance. Among them, it is worth mentioning Freeze-TCP [10], a TCP variant designed to improve TCP performance in mobile environments where temporal disconnections occur frequently. In Freeze-TCP, the receiver is responsible for monitoring the signal strength to predict disconnections and advertising a zero window to the sender before the disconnection takes place. Upon the reception of a zero-window size, the sender enters the ZWP (Zero-Window Probe) mode, in which it "freezes" its transmission parameters (congestion window, retransmission timers), and it cannot transmit any data. By means of this mechanism, it is possible to avoid potential losses and prevent the congestion window from dropping because no retransmission timers expire during the handoff. When the connection is resumed, the receiver advertises a nonzero window which allows the sender to continue its transmission. A modified version of Freeze-TCP could be used in CRNs, in which the TCP sender is responsible for freezing itself its own parameters without the need of being warned by the receiver, as it is the case in Freeze-TCP. Since within a CRN all members share information about the channel, the sender itself could predict the disconnection due to an incoming frequency handoff [7].

Notice that although the attacker knows the CRN is freezing TCP connections during the handoffs, it cannot take advantage of this information in order to improve the attack. The fact is that freezing TCP parameters limits the attacker to only degrade the TCP throughput, since there are no transmissions during the handoff time. However, if the attacker continues forcing frequency handoffs, it can produce a permanent DoS attack. In order to avoid it, the CRN must prevent the attacker from rapidly detecting the next spectrum band to be used. Assuming the attacker is also a CR device, it can predict the next frequency in two ways: (1) through sensing and (2) by obtaining this information from the CRN common control channel. Notice that the common control channel provides the attacker with a priori knowledge of the next operation channel, while sensing requires a given amount of time until the attacker discovers the new channel. Consequently, securing the control channel should be incorporated by default in any CRN technology [4]. The 802.22 workgroup is dealing with such risk and the current draft [13] defines a security sublayer to provide features such as authentication, authorization, message integrity, and data encryption for data and control channels.

All the previously presented countermeasures can partially mitigate the effects of the Lion attack but cannot stop it at all since it cannot effectively deal with DoS or channel degradation due to jamming. Therefore, a parallel system for finding the attack source such as *Intrusion detection systems* (IDS) is necessary. IDSs in CRNs should monitor other devices for intentional deviation from protocol, that is, misbehavior, detecting which nodes are suspicious or malicious. Several IDS approaches [14–16] could be somehow applied but their particularization to CRNs is still challenging. However, dealing with an IDS for CRNs is out of the scope of this paper.

3. Analytical Model

As explained in Section 2, a Lion attack can degrade the throughput of a TCP connection, leading in some situations to the starvation of the TCP source. In this section, we derive an analytical expression both for the average inactivity time of a TCP source and the reduction of the throughput due to the attack. It is important to remark that presented model is just an approximation, that is, neglecting many marginal contributions. Its accuracy is nevertheless proved by comparing the results with simulated ones in Section 4.

3.1. Mathematical Background. Let S_k as in expression (1) be the sum of $k \in \mathbb{N}$ independent and identically distributed (i.i.d.) random variables X_i, $i \in [1, k] \subseteq \mathbb{N}$, with probability density function (pdf) as in (2) and cumulative distributed function (cdf) as in(3)

$$S_k = X_1 + X_2 + \cdots + X_k = \sum_{i=1}^{k} X_i, \tag{1}$$

$$f_{S_k}(t) = \left(f_{X_1} * f_{X_2} * \cdots * f_{X_k}\right)(t), \tag{2}$$

$$F_{S_k}(t) = \int f_{S_k}(t)dt. \tag{3}$$

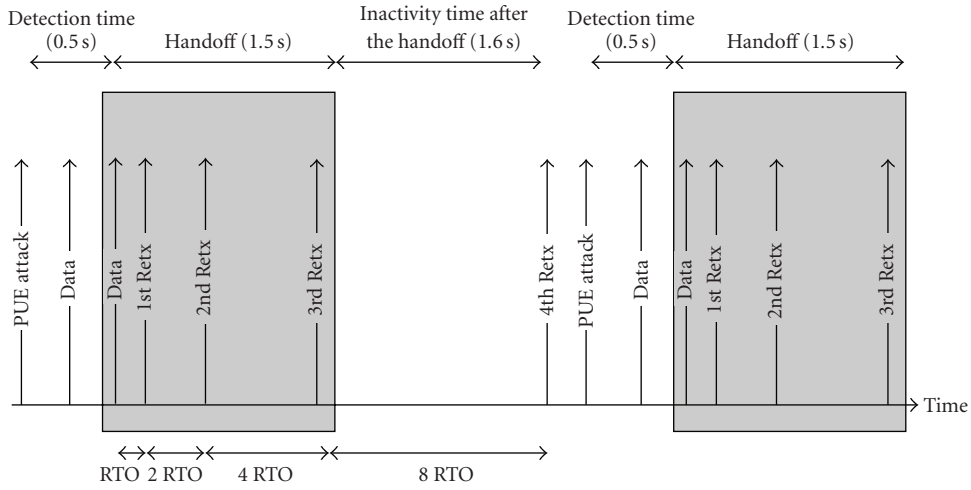

Figure 1: Lion attack.

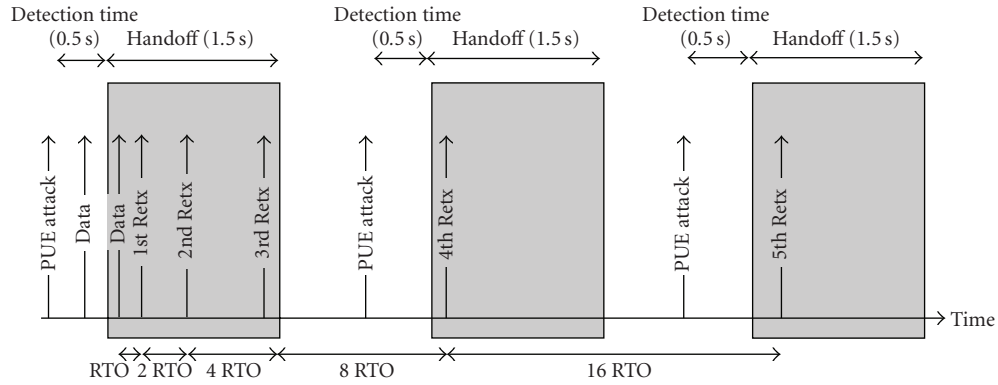

Figure 2: Smart Lion attack.

Lemma 1. *Given S_k as in* (1)*, the probability of only and no more than $k \in \mathbb{N}$ events occurring within the interval $(t, t+\tau]$, $t \geq 0$, $\tau > 0 \in \mathbb{R}$ is*

$$\Pr(k \text{ events in } (t, t+\tau]) = F_{S_k}(\tau) - F_{S_{k+1}}(\tau). \tag{4}$$

Proof. Let us denote by $A = \{S_{k+1} : S_{k+1} \geq \tau\}$, $B = \{S_k : S_k \leq \tau\}$ and $C = \{S_k : S_k > \tau\}$. The probability of only and no more than $k \in \mathbb{N}$ events occurring within the interval $(t, t+\tau]$ can be expressed as the probability of $A \cap B$.

As $A = (A \cap B) \cup (A \cap C)$, being

$$\begin{aligned} \Pr(A) &= \Pr(S_{k+1} \geq \tau) = 1 - F_{S_{k+1}}(\tau), \\ \Pr(A \cap C) &= \Pr(C) = \Pr(S_k > \tau) = 1 - F_{S_k}(\tau), \end{aligned} \tag{5}$$

then

$$\Pr(A \cap B) = \Pr(A) - \Pr(A \cap C) = F_{S_k}(\tau) - F_{S_{k+1}}(\tau). \tag{6}$$

□

3.2. Assumptions. In order to develop the model, the following assumptions have been adopted.

(i) A malicious user performs several attacks, each one leading to a frequency handoff.

(ii) The duration of a handoff, which we denote by t_H is fixed.

(iii) The time needed in order to start a handoff after the CRN detects the presence of a primary user (channel detection time) is fixed with value t_D.

(iv) The time since the end of a frequency handoff until the attacker performs the next attack is modeled by a random variable. Accordingly, we define X_i as a set of i.i.d. random variables (see Figure 3) and $X_i' = X_i + t_D + t_H$ as i.i.d. random variables that represent the time since the end of a handoff until the end of the next one. As a result, we can define S_k' as a random variable being the sum of $k \in \mathbb{N} X_i'$ as in (7) with pdf and cdf as in (8), being S_k the sum of $k \in \mathbb{N} X_i$ as in (1)

$$S_k' = \sum_{i=1}^{k} X_i', \tag{7}$$

$$\begin{aligned} f_{S_k'} &= f_{X_1'} * f_{X_2'} * \cdots * f_{X_k'} \\ &= f_{X_1} * f_{X_2} * \cdots * f_{X_k} * \delta(t - k(t_D + t_H)) \\ &= f_{S_k}(t - k(t_D + t_H)), \\ F_{S_k'}(t) &= F_{S_k}(t - k \cdot (t_H + t_D)). \end{aligned} \tag{8}$$

(v) The round trip time is always smaller than the minimum RTO of the TCP source $\mathrm{RTO}_{\min}$. As explained in Section 2, this can be assumed in CRNs such as 802.22 networks. With each unsuccessful attempt the RTO value is doubled until a maximum value $\mathrm{RTO}_{\max}$ that it is the RTO by a power of 2. As a result, the value of RTO for the ith retransmission can be expressed as in (9) and set of possible retransmission instants t_i defined as in (10)

$$\mathrm{RTO}_i = \begin{cases} 2^{i-1} \cdot \mathrm{RTO}_{\min} & \text{if } i \le i_{\max}, \\ \mathrm{RTO}_{\max} & \text{if } i > i_{\max}, \end{cases} \tag{9}$$

$$i_{\max} = \log_2 \mathrm{RTO}_{\max} + 1,$$

$$\mathrm{RTO}_{\max} = 2^{i_{\max}-1} \cdot \mathrm{RTO}_{\min},$$

$$t_i = \begin{cases} \mathrm{RTO}_{\min} & \text{if } i = 1, \\ t_{i-1} + \mathrm{RTO}_i & \text{if } i > 1 \end{cases}$$

$$= \begin{cases} (2^i - 1) \cdot \mathrm{RTO}_{\min} & \text{if } i \le i_{\max}, \\ (i - i_{\max} + 2) \cdot \mathrm{RTO}_{\max} - \mathrm{RTO}_{\min} & \text{if } i > i_{\max}. \end{cases} \tag{10}$$

(vi) As shown in Figure 3, we assume that it always takes place at least one handoff (handoff 0). Considering that the first segment loss takes place at the beginning of the handoff 0, the retransmissions attempts at $t_i < t_H$ will fall within this handoff and therefore will always fail, implying $\Pr(t = t_i) = 0$. For the sake of clarity, we define a new time axis $t' = t - t_H$, and thus we redefine the retransmission instants as $t'_l = t_i - t_H$ being $t'_1 = t_s - t_H$ with s the index of the first t_i satisfying the condition $t_i > t_H$. As a result l is defined as $i - s + 1$ for $i \ge s$.

3.3. Probability of k Handoffs in Interval $(t', t' + \tau]$. The probability $p_k(\tau)$ that k handoffs occur in the interval $(t', t' + \tau]$ is the probability of k events of the random variable X'_i in interval $(t', t' + \tau + t_H]$ (see Figure 3). Therefore, from Lemma 1, $p_k(\tau)$ can be expressed as in

$$p_k(\tau) = \begin{cases} 1 - F_{S'_1}(\tau + t_H) & \text{if } k = 0, \\ F_{S'_k}(\tau + t_H) - F_{S'_{k+1}}(\tau + t_H) & \text{if } k > 0. \end{cases} \tag{11}$$

3.4. Probability that a Given Instant t' Coincides with the kth Frequency Handoff. Let $h_k(t')$ be the probability function that a given instant t' coincides with the kth frequency handoff given that k handoffs have occurred. An expression for $h_k(t')$ can be easily obtained from Figure 3 as in

$$\begin{aligned} h_k(t')|_{k>0} &= \Pr\left(S'_k - t_H \le t' \le S'_k\right) \\ &= \Pr\left(t' \le S'_k \le t' + t_H\right) \\ &= F_{S'_k}(t' + t_H) - F_{S'_k}(t'). \end{aligned} \tag{12}$$

3.5. Probability that the Inactivity Time Is a Given Value. Let T be the inactivity time of a TCP source, that is, the time from the beginning of a frequency handoff until the TCP source successfully transmits a segment. Consequently, T is the sum of all the RTOs (explained in Section 3) expired before a retransmission succeeds. Therefore, we can define T as a discrete random variable with a set of possible values t_i defined as in (10).

The probability that $T = t_i$ is equal to the probability that the instant of time $t = t_i$ does not fall within a handoff interval, given that the previous instants $t = t_j$ with $j = 1 \cdots i - 1$ do fall within a handoff interval. For example, the inactivity time will be $T = 15 \cdot \mathrm{RTO}_{\min}$ whenever retransmissions performed at instants $t_1 = \mathrm{RTO}_{\min}$, $t_2 = 3 \cdot \mathrm{RTO}_{\min}$ and $t_3 = 7 \cdot \mathrm{RTO}_{\min}$ fail, because the connection is not available due to a frequency handoff, but the next attempt at $t_4 = 15 \cdot \mathrm{RTO}_{\min}$ succeeds.

Then, the probability $\Pr(T = t_i)$ can be computed as in (13), with $k_{\max}$ the maximum number of handoffs which can take place during the interval $[0, t'_l]$ as in (14) and $k_{\min} = l' - 1$ the minimum number of handoffs that must take place during the interval $[0, t'_l]$ in order to have an inactivity time of t_i, that is, the number of retransmission attempts that fail at $t'_{l-1}, t'_{l-2}, \ldots, t'_1$ before the next one succeeds at instant t'_l

$$\Pr\left(T = t_i = t'_l + t_H\right) = \begin{cases} 1 - F_{S'_1}\left(t'_l\right) & \text{if } l = 1, \\ \displaystyle\sum_{k=k_{\min}}^{k_{\max}(t'_l)} p_k\left(t'_l\right) * \zeta(1, 1, l, k) & \text{if } l > 1, \end{cases} \tag{13}$$

$$k_{\max}(t') = \left\lceil \frac{t' - t_D}{t_H + t_D} \right\rceil, \tag{14}$$

$$\zeta(l, j, l_{\max}, k)$$

$$= \begin{cases} \displaystyle\sum_{m=j}^{m_{\max}} \left(h_m\left(t'_l\right) \cdot \zeta(l+1, m+1, l_{\max}, k)\right) & \text{if } l < l_{\max}, \\ F_{S'_k}\left(t'_l\right) & \text{if } l = l_{\max},\ j \le k, \\ 1 & \text{if } l = l_{\max},\ j > k, \end{cases} \tag{15}$$

$$m_{\max} = \begin{cases} k - (l_{\max} - l - 1) & \text{if } k - (l_{\max} - l - 1) < k_{\max}\left(t'_l\right), \\ k_{\max}\left(t'_l\right), & \text{otherwise}, \end{cases} \tag{16}$$

where k is the total number of handoffs to be performed during the period $(t_H, t'_l + t_H)$; $j - 1$ the number of handoffs already occurred until instant t'_{l-1}; $l_{\max} - l - 1$, the number of handoffs which must occur after t'_l and coincide, each one of them, with the following periods $t'_{l+1}, t'_{l+2}, \ldots$ reach $t'_{l_{\max}}$; and $m_{\max} - j + 1$ the maximum number of handoffs that can take place until instant t'_l.

For the sake of clarity, let us suppose that we want to compute $\Pr(T = t_i = 6.2\,\mathrm{s})$ for a given connection with $\mathrm{RTO}_{\min} = 0.2\,\mathrm{s}$ and $t_H = 1.5\,\mathrm{s}$. The set of instants t_i to consider are $t_1 = 0.2\,\mathrm{s}, t_2 = 0.6\,\mathrm{s}, t_3 = 1.4\,\mathrm{s}, t_4 = 3\,\mathrm{s}, t_5 = 6.2\,\mathrm{s}$.

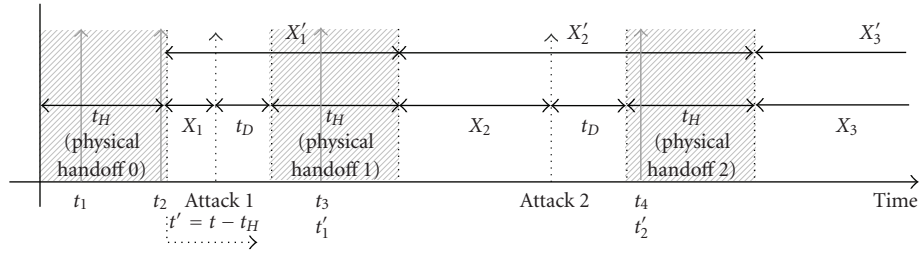

Figure 3: Analytical model for the Lion attack.

Assuming that the first handoff is performed at $t = 0$, the first retransmission attempt will take place at $t = 0.2$ s. Since it will match the first handoff it will fail, and the same will happen for the next retransmissions attempts at $t_2 = 0.6$ s and $t_3 = 1.4$ s (since $t_i < 1.5$ s). A new attempt will take place at $t = 3$ s when the first handoff has ended, but in order to have an inactivity time of $T = t_i = 6.2$ s, this retransmission should fail too. Otherwise, the inactivity time would be $T = t_4 = 3$ s.

Since the first instant satisfying $t_i > t_H$ is $t_i = t_4$, now we can define $t'_1 = t_4 = 3$ s and $t'_2 = t_5 = 6.2$ s, since $\Pr(T = t_i) = 0$ for the previous instants. Then,

$$\begin{aligned}\Pr(T = t_i = 6.2\,\text{s} = t'_2) &= \sum_{k=k_{\min}}^{k_{\max}} p_k\left(t'_l\right) * \zeta(1,1,l,k) = \sum_{k=1}^{3} p_k(t'_2) * \zeta(1,1,l,k) \\ &= p_1(t'_2) * \zeta(1,1,2,1) + p_2(t'_2) * \zeta(1,1,2,2) \\ &\quad + p_3(t'_2) * \zeta(1,1,2,3)\end{aligned} \tag{17}$$

with $k_{\min} = 1$, since at least one handoff must take place at $t'_1 = 3$ s and $k_{\max} = 3$, which can be easily obtained through (14).

If there is only one handoff during the interval (t_H, t'_i), it must coincide with $t'_1 = 3$ s, and therefore

$$\zeta(1,1,2,1) = \sum_{m=1}^{1} h_m(t'_1) \cdot \zeta(2,2,2,1) = h_1(t'_1). \tag{18}$$

If there are two handoffs during the interval (t_H, t'_i), one of them must coincide with $t'_1 = 3$ s, and the second must not coincide with $t'_2 = 6.2$ s; otherwise, the time of inactivity would be longer than $t'_2 = 6.2$ s. Then,

$$\zeta(1,1,2,2) = \sum_{m=1}^{1} h_m(t'_1) \cdot \zeta(2,2,2,2) = h_1(t'_1) * F_{S'_2}. \tag{19}$$

Finally, if there are three handoffs during the interval (t_H, t'_i), at least one of them must coincide with $t'_1 = 3$ s and the last one must not coincide with $t'_2 = 6.2$ s. Accordingly,

$$\zeta(1,1,2,3) = \sum_{m=1}^{1} h_m(t'_1) \cdot \zeta(2,2,2,3) = h_1(t'_1) * F_{S'_3}. \tag{20}$$

3.6. Calculation of the TCP Source Inactivity Time after a Handoff Occurs. Since T is a discrete random variable with a set of possible values t_i defined as in (10) with probabilities $\Pr(T = t_i)$ as in (13), the expected average time of TCP source inactivity $\overline{T}$ after receiving an attack can be obtained as in

$$\overline{T} = \sum_{i=1}^{\infty} t_i \cdot \Pr(T = t_i). \tag{21}$$

3.7. Obtaining the TCP Inactivity Percentage due to the Lion Attack. We can assume an inactivity percentage $U_{\text{inactivity}}$ as in (22), or, the other way round, the percentage U_{activity} as in (23) which shows the reduction of the throughput due to an attack with respect to the transmission time without the Lion attack.

$$U_{\text{inactivity}}\ (\%) = \frac{\overline{T}}{\overline{T} + \overline{A}} \times 100, \tag{22}$$

$$U_{\text{activity}}\ (\%) = \frac{\overline{A}}{\overline{T} + \overline{A}} \times 100, \tag{23}$$

$\overline{T}$, defined as in (21), is the average inactivity time of the TCP source due to the attack derived in the previous section.

The average activity time $\overline{A}$ is the mean time since the end of a frequency handoff until the next one starts and can be computed as in

$$\overline{A} = E[X_i + t_D] = E[X_i] + t_D. \tag{24}$$

4. Model Validation

With the purpose of validating the model proposed in Section 3, we have conducted a set of simulations with the ns-2 simulator [17]. The inactivity time of a TCP connection due to the Lion attack is computed and compared to the results provided by the model, which has been programmed in matlab [18].

The presented simulation results reflect the impact of the Lion attack on TCP throughput both when the victim source freezes TCP parameters and when it does not. Neither IDS countermeasures nor the use of unsecure control data are simulated. The rationale behind this is that with an IDS efficiently operating within the CRN, the attack has no impact since fake primary transmissions will be detected, and thus the CRN will not switch to another channel.

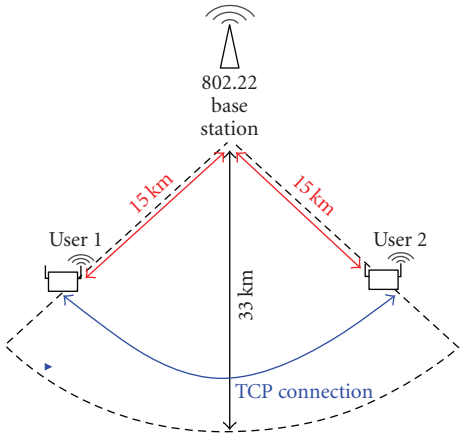

Figure 4: Simulation environment.

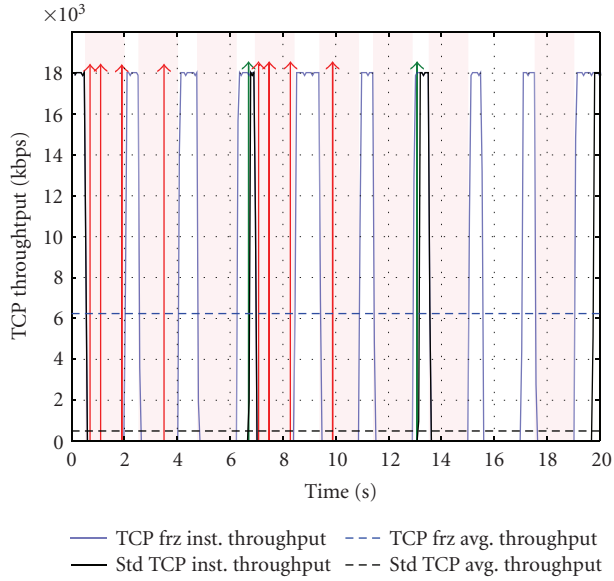

Figure 5: Effect of the lion attack on TCP throughput freezing and nonfreezing TCP transmission.

Furthermore, if the victim network uses an unsecure control channel, the attacker can easily get the next operational channel and perform a DoS. In this case, simulation results are of little value since we would get just a flat zero throughput.

4.1. Simulation Scenario. Figure 4 depicts the simulated environment, consisting of a TCP connection between two secondary users of an 802.22 CRN. As 802.22 specifications define spectral efficiencies ranging from 0.5 bit/(s/Hz) to 5 bit/(s/Hz), considering a mean of 3 bits/(s/Hz), we have assumed a network transmission capacity of 18 Mbps (6 MHz TV channel) [2]. Given that 802.22 standard defines a signal coverage of up to 33 Km for 4 W CPE EIRP, we have assumed an average distance between both secondaries and the base station of 15 Km and thus a propagation delay of 50 μs (speed of light). The process delay at the base station has been neglected and, in order to just reflect the effects of the handoffs on the throughput, also the bit error rate (BER).

The attacker must sense the medium in order to detect the next channel to be used by the CRN after the handoff. Assuming 45% of the TV channels in use, there are 36 free unlicensed channels for CRN operation (out of 67 TV channels available in the UHF and VHF bands). Primary transmissions should not be interfered, so at least there must be 2 empty channels between every pair of TV channels in use [2]. This fact reduces the amount of available channels for CRN operation to 12. Considering a channel sensing time of 46.95 ms [19] for detecting the occupation of a given channel, it will take to the attacker $(12/2) \cdot 46.95$ ms $= 305.175$ ms in average to discover the new CRN operation channel.

From the previous reasoning, we have modeled the time since the end of a handoff until the next attack begins, as an exponential random variable with mean $1/\lambda = 305.175$ ms. Although to get more realistic results other random distributions could be more suited, we have selected an exponential distribution for ease of computation. Notice that the sum of k of exponential random variables, that is, the base of the analytical model, can be easily computed as a gamma distribution.

After the CRN receives the attack by means of a PUE, it takes t_D s to detect the fake primary transmission and stop transmissions at PHY/MAC layer (*channel move time*). We have set a typical value for this parameter of $t_D = 500$ ms [2]. The handoff duration is also set to a typical value of $t_H = 1.5$ s [2].

The TCP sender is fed by an FTP source which generates TCP segments of 1040 bytes with two different implementations of TCP: standard TCP Reno and the proposed modification of TCP Reno (see Section 2.3). The only difference between them is that the later freezes congestion control parameters, that is, congestion window and threshold, as well as the retransmission timers whenever a handoff occurs (handoff beginning is provided by lower layers), resuming the transmission when the handoff ends (handoff end also provided by lower layers). On the contrary, standard TCP Reno is not aware of lower layers and thus continues transmitting during a handoff so, if the handoff lasts long enough, the retransmission timer expires for pending segments. This fact, as previously stated in Section 2.2, can imply long inactivity times. Taking into account that the RTT value for this scenario is much below 100 ms (see expression (26)), as afore mentioned a minimum retransmission time out of $\text{RTO}_{\text{min}} = 200$ ms has been adopted. Furthermore, a maximum value of $\text{RTO}_{\text{min}} = 12.8$ s (default TCP value in the simulator ns-2).

4.2. Simulation Results. Figures 5 and 6 represent the effects of the Lion attack on TCP throughput when using standard TCP Reno and TCP Reno with parameters freezing whenever a handoff occurs. In Figure 5, the attacker senses the media until it detects the new CRN operation channel and performs a new PUE. In Figure 6, handoffs are performed matching the retransmission attempts of the TCP sender.

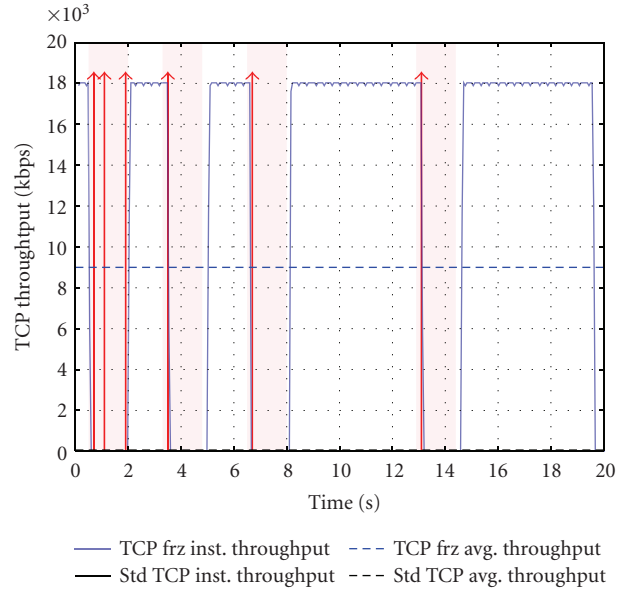

Figure 6: Effect of the smart lion attack on TCP throughput freezing and nonfreezing TCP transmission.

Figures 5 and 6 clearly show that TCP throughput is higher when freezing TCP parameters than without freezing, since the TCP source remains inactive just during the handoffs and makes the most of the available transmission time. However, standard TCP continues transmitting segments during the handoffs, leading to the expiration of the retransmission timers. This fact reduces TCP throughput because of two causes: (1) congestion window is reduced to 1 segment; and (2) every time a segment is retransmitted, the retransmission timer is doubled (until it reaches a maximum value). The latter increases the inactivity time, since the TCP sender is not allowed to transmit any data until the next retransmission timer expires. The former almost does not affect our CRN since the optimal window value for the connection is, as show in expression (26), just one segment.

$$\mathrm{RTT} = t_{tx} + 2t_{\mathrm{prop}} \approx 641\,\mu\mathrm{s}, \tag{25}$$

$$W_{\mathrm{opt}}\ (\mathrm{segments}) = \frac{\mathrm{RTT}}{t_{tx}} \approx 1.42. \tag{26}$$

As stated in (10), the time between consecutive retransmissions for a given segment is doubled with each unsuccessful attempt. Because of this, if a segment transmission fails at $t = 0$, the corresponding retransmission attempts will take place at $t = [0.2\,\mathrm{s}, 0.6\,\mathrm{s}, 1.4\,\mathrm{s}, 3\,\mathrm{s}, \ldots]$. In Figures 5 and 6, the retransmission attempts are represented as red arrows if the link is not available (and therefore the retransmission fails) and as green arrows otherwise. The handoff intervals are represented with a light red background. For example, Figure 5 shows that the first handoff takes place at $t = 0.5\,\mathrm{s}$ with a duration of 1.5 s. The first retransmission is performed 200 ms after the beginning of the handoff, at $t = 0.5\,\mathrm{s} + 0.2\,\mathrm{ms} = 0.7\,\mathrm{s}$, before the end of the handoff. The next take place at $t = 0.7\,\mathrm{s} + 0.4\,\mathrm{s} = 1.1\,\mathrm{s}$ and $t = 1.1\,\mathrm{s} + 0.8\,\mathrm{s} = 1.9\,\mathrm{ms}$,

Table 1: Average activity time, average inactivity time and percentage of inactivity.

λ (ms)	Non Freezing TCP			Freezing TCP		
	$\overline{A}$(s)	$\overline{T}$(s)	$U_{\mathrm{inactivity}}$ (%)	$\overline{A}$(s)	$\overline{T}$(s)	$U_{\mathrm{inactivity}}$ (%)
3.28	0.54	17.03	96.92	0.9	1.5	62.34
2	0.70	17.35	96.07	1.11	1.5	57.41
1	1.12	11.7	91.21	1.62	1.5	47.96

as well within the period of handoff. At $t = 0.5\,\mathrm{s} + 1.5\,\mathrm{s} = 2\,\mathrm{s}$ the first handoff ends, but the TCP sender remains inactive (waiting for the expiration of the retransmission timer) until time $t = 1.9\,\mathrm{s} + 1.600\,\mathrm{s} = 3.5\,\mathrm{s}$. By that time, the attacker has forced another handoff, and therefore the retransmissions fails again until time $t = 3.5\,\mathrm{s} + 3.2\,\mathrm{s} = 6.7\,\mathrm{s}$, which finally matches up with a period of communication, and therefore it succeeds. However, as it can be observed, the TCP connection (without freezing) has been inactive around 6.2 seconds.

On the other hand, Figure 6 shows an example of the smart Lion attack. In this case, the attacker can detect the new operational channel through local sensing and predicts the retransmission timer values, forcing the handoffs to coincide with the retransmissions attempts. The figure clearly shows that the throughput is null for standard TCP. However, freezing parameters makes the smart attack even less effective than the standard attack.

Table 1 reflects the percentage of inactivity $U_{\mathrm{inactivity}}$ of the TCP source when the attacker performs several attacks (see expression (23)), considering both TCP implementations. The time since the end of a handoff until the next begins follows an exponential distribution with mean ranging from 305 ms to 1 s. In addition, it provides the percentage of activity when the attacker performs a smart attack.

The results clearly show the degradation of TCP throughput when a Lion attack is received. Obviously the more frequent are the attacks, the bigger the negative impact on the TCP source. However, freezing TCP transmission parameters can deal with the standard attack, allowing the TCP sender to transmit whenever the channel is available. With regard to the smart attack, freezing TCP parameters during the handoff avoids unnecessary retransmissions, leading to a higher activity percentage of time. As the attacker forces handoffs only at potential instants of retransmissions (each time more infrequent), the TCP sender can transmit during a longer interval of time.

4.3. Analytical Model Results. The analytical model described in Section 3 has been programmed in matlab and run with the parameters given in Section 4.1. Table 2 provides the average inactivity time $\overline{T}$ and inactivity percentage $U_{\mathrm{inactivity}}$ obtained for the analytical model in comparison with the results obtained via simulation.

Note that the model derived is valid for any probability distribution, so it can be used to analyze different attack patterns. Accordingly, it can be used to study the impact on TCP connections of other phenomena, such as for example noise, by choosing the right distribution.

Table 2: Analytical model versus simulation.

λ (ms)	Simulation			Analytical model		
	$\overline{A}$(s)	$\overline{T}$(s)	$U_{\text{inactivity}}$ (%)	$\overline{A}$(s)	$\overline{T}$(s)	$U_{\text{inactivity}}$ (%)
3.28	0.54	17.03	96.92	0.305	16.03	98.13
2	0.70	17.35	96.07	0.5	16.61	97.08
1	1.12	11.7	91.21	1	11.6	92.06

5. Conclusions

Cognitive Radio Networks arise as a promising solution to share and take advantage of the scarcity of radio spectrum as well as to enhance the overall availability of transmitted data. These networks are composed of smart devices that "intelligently" select the best spectrum opportunities. Although CRNs make use of existing technologies, their particular characteristics pose new security challenges and can increase the complexity of other known attacks.

In this paper, we have detailed the Lion attack, originally outlined in [7] and its potential countermeasures. The Lion attack is a cross-layer attack to CRNs performed at the physical link layer and targeted to TCP that relies on emulating a licensed transmission in order to force a CRN to perform frequency handoffs. Connections within the CRN are interrupted during the handoffs, thus reducing TCP throughput. Proper election of when to force a handoff can even starve at all the TCP throughput. With the aim of mitigating this attack, we have first described some modifications to the TCP protocol in order to avoid the degradation of the throughput due to frequency handoffs. In this way, CRN devices will be able to freeze TCP connection parameters during frequency handoffs and adapt them to the new network conditions after the handoff. Second, we have also addressed the need for securing the control data in order to prevent the attacker from eavesdropping current and future actions of the CRN, and we have denoted the necessary use of *intrusion detection systems* (IDSs) specifically adapted to CRNs.

The main contribution of this paper is the evaluation of the impact of the Lion attack on TCP performance through an analytical model. The model provides an expression for the average time of inactivity of a TCP sender due to the attack and also the percentage of inactivity, parameters which measure the impact of the attack on TCP throughput. Moreover, the model has been validated through simulations considering two implementations of TCP: the standard TCP Reno and the modified version proposed to mitigate the effects of the attack. The results obtained show that freezing TCP parameters reduces the effect of the handoffs (caused by the attack) on the throughput of TCP. Moreover, the smart version of the attack prevents it from leading to a DoS.

Further work is needed in order to analyze how the attack can be mitigated by means of an IDSs. Its use may avoid unnecessary handoffs due to fake primary transmissions, but it will also lead to false negative and/or positives that could take the network to continue forcing unnecessary handoff for the former and to illegally perform for the later. Although we strongly believe that IDS can be effective in dealing with the attack; its impact on network performance should also be studied in depth.

Acknowledgment

This paper has been supported partially by the Spanish Research Council (CICYT) Project no. TEC2008-06663-C03-01 (P2PSEC), by the Spanish Ministry of Science and Education with CONSOLIDER CSD2007-00004 (ARES) and by Generalitat de Catalunya with Grant no. 2005 SGR 01015 to consolidated research groups.

References

[1] R. Chen and J.-M. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," in *Proceedings of the 1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks (SDR '06)*, pp. 110–119, September 2006.

[2] C. Cordeiro, K. Challapali, D. Birru, and N. S. Shankar, "IEEE 802.22: an introduction to the first wireless standard based on cognitive radios," *Journal of Communications*, vol. 1, no. 1, pp. 38–47, 2006.

[3] C. T. Clancy and N. Goergen, "Security in cognitive radio networks: threats and mitigation," in *Proceedings of the 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom '08)*, May 2008.

[4] O. León, J. Hernández-Serrano, and M. Soriano, "Securing cognitive radio networks," *International Journal of Communication Systems*, vol. 23, no. 5, pp. 633–652, 2010.

[5] C. Song and Q. Zhang, "Achieving cooperative spectrum sensing in wireless cognitive radio networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 2, pp. 14–25, 2009.

[6] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *Proceedings of IEEE International Conference on Communications (ICC '06)*, pp. 1658–1663, July 2006.

[7] O. León, J. Hernández-Serrano, and M. Soriano, "A new cross-layer attack to TCP in cognitive radio networks," in *Proceedings of the 2nd International Workshop on Cross Layer Design (IWCLD '09)*, pp. 1–5, 2009.

[8] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: motivation, architecture, algorithms, performance," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1246–1259, 2006.

[9] V. Jacobson, "Congestion avoidance and control," in *Proceedings of the Communications Architectures and Protocols Symposium (SIGCOMM '88)*, pp. 314–329, 1988.

[10] T. Goff, J. Moronski, D. Phatak, and V. Gupta, "Freeze-TCP: a true endto- end tcp enhancement mechanism for mobile environments," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 3, pp. 1537–1545, March 2000.

[11] A. Al Hanbali, E. Altman, and P. Nain, "A survey of tcp over ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 3, pp. 22–36, 2005.

[12] D. Le, X. Fu, and D. Hogrefe, "A cross-layer approach for improving TCP performance in mobile environments," *Wireless Personal Communications*, vol. 52, no. 3, pp. 669–692, 2010.

[13] "IEEE 802.22 Working Group on Wireless Regional Area Networks ," IEEE 802.22 draft v3.0, http://www.ieee802.org/22/.

[14] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 275–283, August 2000.

[15] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 48–60, 2004.

[16] V. Bhuse and A. Gupta, "Anomaly intrusion detection in wireless sensor networks," *Journal of High Speed Networks*, vol. 15, no. 1, pp. 33–51, 2006.

[17] X. PARC and UCB, USC/ISI, SAMAN, CONCER, ACIRI, and etc, "The network simulator - ns-2," http://www.isi.edu/nsnam/ns/.

[18] "Matlab—the language of technical computing," http://www.mathworks.com/.

[19] G. Chouinard, D. Cabric, and M. Gosh, "IEEE P802.22 Wireless RANs-Sensing Thresholds," May 2006, https://mentor.ieee.org/802.22/dcn/06/22-06-0051-04-0000-sensing-thresholds.xls.