Impact of Primary User Emulation Attacks on Dynamic Spectrum Access Networks

Z. Jin, *Student Member, IEEE,* S. Anand, *Member, IEEE* and K. P. Subbalakshmi, *Member, IEEE*

***Abstract*—Primary user emulation attack (PUEA) is a denial of service (DoS) attack unique to dynamic spectrum access (DSA) networks. While there have been studies in literature to detect and mitigate PUEA, the impact of PUEA on the performance of secondary networks has seldom been studied. In this paper, we analyze how PUEA affects call dropping and delay in secondary networks carrying both real-time traffic and non-real-time traffic. Numerical results indicate that PUEA can increase the number of dropped calls by up to two orders of magnitude and can increase the mean delay by up to a factor larger than two. We then evaluate the performance of secondary networks that deploy the protocols which we proposed previously to mitigate PUEA. Our protocols reduce the number of dropped calls by up to one order of magnitude. Our protocols are also shown to exhibit almost the same delay performance as that of a system with no PUEA, for low malicious traffic load. When malicious traffic load is high, our protocols provide an improvement on the delay performance by up to 54%.**

***Index Terms*—Dynamic spectrum access (DSA), primary user emulation attack (PUEA), Markov model, call dropping, delay**

I. Introduction

Cognitive radio enabled dynamic spectrum access (DSA) networks [1]-[3] allow unlicensed “secondary users” to access the spectrum bands unused by licensed “primary users” to improve spectrum utilization. The secondary users evacuate the spectrum bands upon the return of the primary users. This spectrum etiquette could be exploited by malicious users to mount a DSA specific attack called primary user emulation attack (PUEA) [4]. In such an attack, a set of malicious secondary users transmit signals whose characteristics resemble that of the primary transmitter, misleading the good secondary users to believe that the primary user is active and evacuate the spectrum unnecessarily.

There are several studies in literature that deal with detection and mitigation of PUEA [4]-[11]. Some of these include isolation of malicious users using directional antennas on the secondary users [4] or underlying sensors [5], while some mitigate PUEA by using hypothesis testing [7],[8]. Additional description and references on PUEA can be found in [12]-[17]. We proposed the first centralized protocol [9] to mitigate PUEA, in which secondary users convey their individual decisions to a centralized controller, which in turn, uses the individual decisions obtained from all the secondary users to come up with a decision for the entire network. We then developed the first distributed protocol to mitigate PUEA [10], in which secondary users exchange their individual spectrum decisions with their one-hop neighbors to thwart PUEA. The centralized protocol in [9] and the distributed protocol in [10] were found to reduce the probability of successful PUEA by up to four and three orders of magnitude, respectively.

While research has been performed on the mitigation of PUEA, most studies focus on reducing the error probabilities in the primary user sensing mechanism. The impact of PUEA on the quality-of-service (QoS) performance of the secondary network has not been studied in detail. In order to illustrate this, consider a secondary network in which secondaries use channels from the set $\mathcal{C} = \{1, 2, \cdots, C\}$, for communication. Let a particular secondary user use channel 1. If PUEA is successfully launched on channel 1, then this user has to switch to another channel that is neither used by the primary users nor by other secondary users. If such a channel is not available, then the call incident on the secondary user is dropped if it is a delay sensitive real-time application. If the call corresponds to non-real-time traffic, it can be buffered till a channel becomes available by virtue of a primary or another secondary call leaving the system. A dropped call results in unreliable communication. If the call is buffered then it causes additional delay, thereby resulting in degraded quality-of-service (QoS). To the best of our knowledge, the effect of PUEA on the performance of secondary networks (e.g., call dropping, delay), has not been studied.

In this paper, we study the impact of PUEA on secondary networks. Specifically we study the call dropping in secondary networks carrying real time traffic and the delay suffered by secondary networks carrying non-real-time traffic, due to PUEA. Note that in some scenarios, certain real-time traffic may be tolerant to some delay, while some calls corresponding to non-real-time traffic may be dropped if their waiting time in the buffer exceeds a specific threshold. In this paper, by “real-time traffic”, we mean delay-intolerant traffic which is dropped immediately when no available channels are found. By “non-real-time traffic”, we mean delay-tolerant traffic which can be buffered in the system until a channel becomes available. To perform our study, we consider two types of malicious behavior: (i) “obstructive” malicious users, who launch PUEA with the sole objective of evacuating secondary users but not to use the white spaces for themselves and (ii) “greedy” malicious users, who use the white spaces like other secondary users in addition to launching PUEA.

We model the channel occupancy in DSA networks under PUEA as a three dimensional continuous time Markov chain (3D-CTMC), and use the 3D-CTMC to analyze the call dropping in secondary networks with real-time traffic and the delay in secondary networks carrying non-real-time traffic, in

Manuscript received November 22, 2010. Revised September 14, 2011, January 21, 2012, March 29, 2012. This work, in part, was presented in IEEE GLOBECOM’2010 [21].

The authors are with the department of electrical and computer engineering, Stevens Institute of Technology. Email: {zjin, asanthan, ksubbala}@stevens.edu

the presence of PUEA. Numerical results indicate that PUEA can increase the number of dropped secondary calls by up to two orders of magnitude in networks carrying real-time traffic, and can increase the mean delay by up to a factor larger than two in networks carrying non-real-time traffic. We also evaluate the performance of secondary networks that deploy the centralized and distributed protocols we proposed in [9] and [10], respectively, to mitigate PUEA. We show that our protocols can improve the call dropping performance by up to one order of magnitude and can provide almost the same delay performance as that of a system with no PUEA, when malicious traffic load is low. When malicious traffic is high, our protocols improve the delay performance by up to 54%.

The rest of the paper is organized as follows. Section II presents the system model. Sections III and IV present the analysis of impact of PUEA on DSA networks with real-time traffic and non-real-time traffic, respectively. Results and conclusions are presented in Sections V and VI, respectively.

II. System Model

Consider a DSA network with multiple channels, which can be used by secondary users when primary users are not present. Malicious users can deceive secondary users into believing that a primary user is present (when it is not), by launching PUEA. We assume that the malicious users co-ordinate with each other to launch PUEA. When malicious users operate un-coordinately, each malicious user has to individually launch PUEA. We [9],[10] and other research work on PUEA [5] had shown that an individual malicious user can almost never launch PUEA singly without any help from other malicious users. Therefore, malicious users need to co-ordinate to launch PUEA. There has been work in the literature on PUEA [5],[18] and other research on security (e.g., jamming) [19], on how malicious users co-ordinate. We assume that the malicious users can use one of those mechanisms to co-ordinate. Secondary users that are successfully attacked by malicious users will have to switch to alternate channels to continue their transmission. If no alternate channels are available for switching, the call incidents on the secondary users are dropped if the network carries delay sensitive real-time traffic or are buffered to be transmitted later (following the departure of another secondary call or a primary call or the end of malicious activity) if the network carries reliability sensitive, non-real-time traffic. Only ongoing secondary calls that cannot continue on the same channel due to successful PUEA or primary activity are buffered (newly arriving secondary calls that do not find a channel for transmission are blocked). Therefore, we consider a finite buffer of size equal to the total number of available channels, C. The provision of such a buffer ensures that non-real-time secondary calls are completed (with additional delay) but not dropped.

It is of interest to determine the effect of PUEA on the call dropping performance of secondary networks carrying real-time traffic and on the additional delay suffered by the calls in secondary networks carrying non-real-time traffic. In order to perform the analysis, we consider two types of behavior for malicious users, namely, i) "obstructive" malicious users and ii) "greedy" malicious users. Obstructive malicious users do not use any spectrum for their own communications. Instead, their sole aim is to evacuate other secondary users out of the system by launching PUEA. This kind of behavior is common in tactical networks where the purpose of preventing the communication of other users is achieved by launching PUEA. Greedy malicious users launch PUEA to grab channels for their own communications. Both greedy and obstructive malicious users launch PUEA only when the system is full, i.e., when all the C channels are occupied by primary, secondary or malicious users. For obstructive malicious users, this is because, DSA networks allow secondary users to switch channels and hence, even if the PUEA is successful, the objective of disrupting secondary communication is failed if alternate channels are available for switching. Similarly, greedy malicious users do not require to launch PUEA if other channels are available for their communications.

We consider the case where all the primary users have identical call arrival rates. The case when different primary users have different arrival rates is very complex to analyze. In this case, a three dimensional Markov chain is insufficient. An accurate model would be a $C-$dimensional Markov chain, one corresponding to each of the primary users. The number of states in such a Markov chain could be exponential in C, thus making the analysis intractable and complex. However, if all the primary users are of the "same type", e.g., TV transmitters, then it is fair to assume that the arrival rate is equal for all the primary users. We consider primary and secondary calls to arrive according to a Poisson process with rates, λ_p and λ_s, respectively [20]. The primary and secondary users hold the channels for an exponentially distributed random time with means $\frac{1}{\mu_p}$ and $\frac{1}{\mu_s}$, respectively [20]. When all the C channels are occupied, malicious users launch PUEA according to a Poisson process with arrival rate, λ_m and hold the channel for a random time which is exponentially distributed with mean $\frac{1}{\mu_m}$ [21]. Malicious users can launch PUEA as soon as the system becomes full. However, the arrival epochs of the primary user and secondary users are independent random variables and hence the probability of a primary arriving at exactly the same instant the system is full, is zero. Hence, when malicious users launch PUEA exactly at the same instant the system becomes full, it makes the attacks easy to detect and mitigate [21]. So, malicious users launch PUEA according to a Poisson process in order to avoid being easily detected.

III. Performance Analysis: Real-Time Traffic

Real-time traffic is sensitive to delay and hence, DSA networks carrying real-time traffic drop ongoing secondary calls that do not find an alternate channel to switch to. In order to determine the fraction of dropped real-time secondary calls (or the call dropping probability), it is necessary to model the channel occupancy in the system. We model the channel occupancy as three-dimensional continuous time Markov chain (3D-CTMC). Each state in the 3D-CTMC is a 3-tuple, (s, m, p), representing the number of channels occupied by the secondary, malicious and primary users, respectively. The transition rates of the 3D-CTMC differ depending on the behavior of the malicious users (i.e., differ for obstructive and greedy malicious users). We present the analysis for obstructive malicious users in Section III-A and that for greedy malicious users in Section III-B.

A. "Obstructive" Malicious Users

"Obstructive" malicious users launch PUEA only when there are no available channels, and they do not attack nor use channels for their own communications when the system is not full. The transition rates of the 3D-CTMC are specified in Fig. 1 and explained as follows.

- At any state, (s,m,p), departures occur whenever an ongoing call is completed. Depending on whether the ongoing call is a secondary, malicious or primary call, transitions from state (s,m,p) occur to states $(s-1,m,p)$ (with rate $s\mu_s$), $(s,m-1,p)$ (with rate $m\mu_m$) and $(s,m,p-1)$ (with rate $p\mu_p$), respectively.
- If p channels are occupied by primary users, then a new primary call can arrive only in any one of the remaining $C-p$ channels. Therefore, if the system is not full, i.e., when $s+m+p<C$, transitions corresponding to primary arrival occur from state (s,m,p) to $(s,m,p+1)$ with rate $(C-p)\lambda_p$.
- Malicious users launch PUEA only on those channels occupied by secondary users and when the system is full, i.e., when the state is (s,m,p) with $s+m+p=C$ and $s>0$. Also, malicious users can only launch PUEA on those channels which are not yet occupied by primary users and those are not yet attacked by other malicious users. When an attack is successfully launched on a channel occupied by a secondary user, the corresponding secondary call is dropped. Thus transitions corresponding to malicious activity occur from state (s,m,p) to $(s-1,m+1,p)$ with rate $(C-p-m)\lambda_m$.
- When the system is full and has at least one secondary user, i.e., when the state is (s,m,p) with $s+m+p=C$ and $s>0$, if a malicious user is evacuated due to the return of primary user, it would instantaneously launch PUEA on one of the other channels being occupied by a secondary user, thus making the Markov chain transition from state (s,m,p) to $(s-1,m,p+1)$. However, if no channels are being occupied by good users, the malicious user has to leave the system. This would lead the Markov chain to transition from state $(0,m,p)$ to $(0,m-1,p+1)$.

Let Ψ denote the set of feasible states of the 3D-CTMC shown in Fig. 1.

$$\Psi = \{(s,m,p) \mid s \geq 0, m \geq 0, p \geq 0 \text{ and } s+m+p \leq C\}. \qquad (1)$$

Let $\psi(s,m,p)$ denote an indicator function of Ψ, given by

$$\psi(s,m,p) = \begin{cases} 1 & \text{if } (s,m,p) \in \Psi, \\ 0 & \text{otherwise.} \end{cases} \qquad (2)$$

At steady state, let $P(s,m,p)$ denote the steady state probability of the channel occupancy being given by the state (s,m,p). The local balance equations (LBE's) for the transitions in Figs. 1(a), 1(b) and 1(c) can then be written as in Eqns. (3)-(5). $P(s,m,p)$ can then be obtained, $\forall\ (s,m,p) \in \Psi$, by solving a system of linear equations specified by Eqns. (3)-(5) and the normalization condition.

The probability that a newly arriving secondary call is blocked (i.e., not admitted into the system), is the probability that all the C channels are occupied when the new secondary call arrives. Thus, the blocking probability, p_{block}, is given by

$$p_{\text{block}} = \sum_{\substack{(s,m,p)\in\Psi \\ s+m+p=C}} P(s,m,p). \qquad (6)$$

The call dropping probability, p_{drop}, i.e., the probability that an ongoing secondary call is dropped from the system before it is completed, is the probability that all the C channels are occupied and an arrival of a new primary or PUEA occurs. Thus, p_{drop}, is given by

$$p_{\text{drop}} = \sum_{\substack{(s,m,p)\in\Psi \\ s>0 \\ s+m+p=C}} \frac{(C-p-m)\lambda_m + (C-p)\lambda_p}{\lambda_s + (C-p-m)\lambda_m + (C-p)\lambda_p} P(s,m,p). \qquad (7)$$

B. "Greedy" Malicious Users

Both greedy and obstructive malicious users behave identically when the system is full. Therefore, when $s+m+p=C$, the transitions in the 3D-CTMC occur as shown in Figs. 1(a) and 1(b). However, when the system is not full, $s+m+p<C$, the transition rates of the 3D-CTMC differ from Fig. 1(c) (as shown in Fig. 2) since greedy malicious users also make requests for channels as secondary users do. The LBE for the state transition diagram in Fig. 2 is given by Eqn. (8)

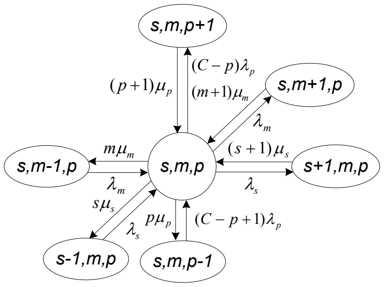


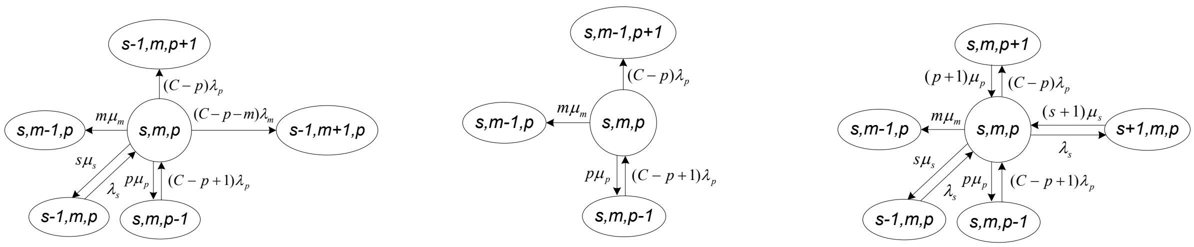
Fig. 2. State transition diagram of the 3D-CTMC for the DSA network carrying real-time traffic, with greedy malicious users, when $s+m+p<C$ (i.e., the system is not full). When $s+m+p=C$, the state transition diagrams are same as Figs. 1(a) and 1(b).

Since the 3D-CTMC with greedy malicious users has the same set of feasible states as that with obstructive malicious users, the steady state probability, $P(s,m,p)$, $\forall(s,m,p) \in \Psi$, can be solved from a system of linear equations specified by Eqns. (3), (4), (8) and the normalization condition. The blocking probability[1], p_{block}, and the dropping probability, p_{drop}, for a system under PUEA launched by greedy malicious users, can then be obtained by substituting this solution, $P(s,m,p)$, in Eqns. (6) and (7), respectively.

IV. Performance Analysis: Non-real-time Traffic

As mentioned in Section II, calls corresponding to reliability sensitive, non-real-time traffic that do not find an alternate channel when primary activity or PUEA occurs, are buffered

[1] Due to the figure limit, we will omit the numerical results of the blocking probability in Section V. Interested readers may refer to [21], a preliminary version of this paper, for more details on the blocking probability performance.



(a) For $s+m+p=C$ and $s>0$.

(b) For $m+p=C$ and $s=0$.

(c) For $s+m+p<C$.

Fig. 1. State transition diagrams of the 3D-CTMC for the DSA network carrying real-time traffic, with obstructive malicious users.

$$
\begin{aligned}
&[p\mu_p + s\mu_s + m\mu_m + (C-p)\lambda_p + (C-p-m)\lambda_m]\, P(s,m,p)\psi(s,m,p) \\
&= \lambda_s P(s-1,m,p)\psi(s-1,m,p) + (C-p+1)\lambda_p P(s,m,p-1)\psi(s,m,p-1)
\end{aligned} \tag{3}
$$

$$
[p\mu_p + m\mu_m + (C-p)\lambda_p]\, P(s,m,p)\psi(s,m,p) = (C-p+1)\lambda_p P(s,m,p-1)\psi(s,m,p-1) \tag{4}
$$

$$
\begin{aligned}
&[p\mu_p + s\mu_s + m\mu_m + (C-p)\lambda_p + \lambda_s]\, P(s,m,p)\psi(s,m,p) = \lambda_s P(s-1,m,p)\psi(s-1,m,p) + (C-p+1)\lambda_p \\
&P(s,m,p-1)\psi(s,m,p-1) + (s+1)\mu_s P(s+1,m,p)\psi(s+1,m,p) + (p+1)\mu_p P(s,m,p+1)\psi(s,m,p+1)
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
&\left[p\mu_p + s\mu_s + m\mu_m + (C-p)\lambda_p + \lambda_m + \lambda_s\right] P(s,m,p)\psi(s,m,p) = \lambda_s P(s-1,m,p)\psi(s-1,m,p) + (C-p+1)\lambda_p P(s,m,p-1)\psi(s,m,p-1) \\
&+(s+1)\mu_s P(s+1,m,p)\psi(s+1,m,p) + (m+1)\mu_m P(s,m+1,p)\psi(s,m+1,p) + (p+1)\mu_p P(s,m,p+1)\psi(s,m,p+1) + \lambda_m P(s,m-1,p)\psi(s,m-1,p)
\end{aligned} \tag{8}
$$

to be transmitted later following the departure of a primary or secondary call or the end of malicious activity on a channel. In order to compute the mean delay suffered by secondary calls corresponding to non-real-time traffic, the channel occupancy is once again modeled as a 3D-CTMC. However, in this case, the three tuple representing a state is (s,m,p) where s represents the total number of secondary calls in the system (including the active ones that have a channel for transmission as well as the ones that are buffered). A channel occupancy state, (s,m,p), therefore indicates p channels used by primary users and m channels occupied by malicious users. If $s+m+p \leq C$, then s represents the number of channels used by secondary users. A state, (s,m,p) such that $s+m+p > C$, represents a scenario when $(C-m-p)$ secondary calls are currently under transmission and $s+m+p-C$ secondary calls are buffered. The analysis for obstructive and greedy malicious users are presented in Sections IV-A and Section IV-B, respectively.

A. *"Obstructive" Malicious Users*

The transition rates of the 3D-CTMC when malicious users display obstructive behavior, is shown in Fig. 3. The set of feasible states of the 3D-CTMC, Ω, (Fig. 3), can be written as

$$
\Omega = \{(s,m,p) \mid 0 \leq s \leq C, m \geq 0, p \geq 0 \text{ and } m+p \leq C\} - \{(0,C,0)\}\,. \tag{9}
$$

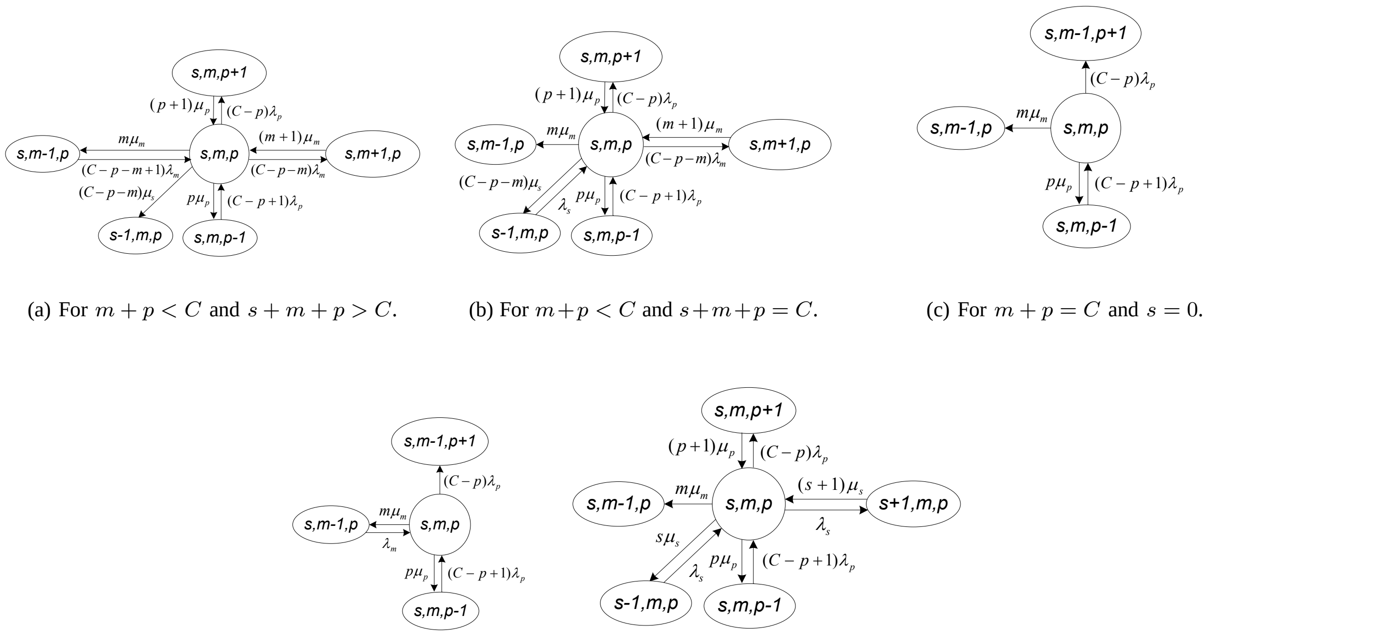
Let $\omega(s,m,p)$ denote an indicator function of Ω, given by

$$
\omega(s,m,p) = \begin{cases} 1 & \text{if } (s,m,p) \in \Omega, \\ 0 & \quad\text{otherwise.} \end{cases} \tag{10}
$$

The LBE's for the transitions shown in Figs. 3(a)-3(e) can then given by Eqns. (11)-(15). $P(s,m,p)$ can then be obtained, $\forall (s,m,p) \in \Omega$, by solving a system of linear equations specified by Eqns. (11)-(15) and the normalization condition.

The mean delay suffered by secondary calls is computed as follows. Consider the head-of-line (HOL) secondary call in the buffer when the system is in state (s,m,p) with $s+m+p > C$. This is the first secondary call that was buffered. This call has to wait for a time, $T_{\text{service}}(s,m,p)$, to obtain a channel, i.e., until at least one of the p primary calls or the m malicious calls or the $C-m-p$ secondary calls terminates. Since the call holding times of secondary, malicious and primary calls are exponentially distributed, $T_{\text{service}}(s,m,p) \sim \exp\left(\mu_{\text{service}}(s,m,p)\right)$, where $\mu_{\text{service}}(s,m,p)$ is given by Eqn. (16). Conditioned on (s,m,p), the rest of the buffer can be modeled as an $M/M/1$ queue (we later average over (s,m,p) in Eqn. (19)), with mean service time, $\mu_{\text{service}}(s,m,p)^{-1}$. The arrival rate into the buffer, $\lambda_{\text{buffer}}(s,m,p)$, is computed as explained below. Secondary calls arrive into the buffer only if $s+m+p \geq C$ and $s > 0$, and this occurrence is only due to the arrival of a primary call or due to successful PUEA launched by malicious users. Therefore, $\lambda_{\text{buffer}}(s,m,p)$ can be written as in Eqn. (17). The mean delay suffered by a secondary call that entered the buffer at state (s,m,p), $T_{\text{delay}}(s,m,p)$, is obtained as the mean sojourn time in the $M/M/1$ queue [22], which is given by

$$
T_{\text{delay}}(s,m,p) = \frac{1}{\mu_{\text{service}}(s,m,p) - \lambda_{\text{buffer}}(s,m,p)}. \tag{18}
$$



(a) For $m+p<C$ and $s+m+p>C$.

(b) For $m+p<C$ and $s+m+p=C$.

(c) For $m+p=C$ and $s=0$.

(d) For $m+p=C$ and $s>0$.

(e) For $s+m+p<C$.

Fig. 3. State transition diagrams of the 3D-CTMC for the DSA network carrying non-real-time traffic, with obstructive malicious users.

$$\begin{aligned}&[p\mu_p+(C-p-m)\mu_s+m\mu_m+(C-p)\lambda_p+(C-p-m)\lambda_m]P(s,m,p)\omega(s,m,p)\\&\quad=(C-p-m+1)\lambda_m P(s,m-1,p)\omega(s,m-1,p)+(m+1)\mu_m P(s,m+1,p)\omega(s,m+1,p)\\&\qquad+(C-p+1)\lambda_p P(s,m,p-1)\omega(s,m,p-1)+(p+1)\mu_p P(s,m,p+1)\omega(s,m,p+1)\end{aligned}\tag{11}$$

$$\begin{aligned}&[p\mu_p+(C-p-m)\mu_s+m\mu_m+(C-p)\lambda_p+(C-p-m)\lambda_m]P(s,m,p)\omega(s,m,p)\\&\quad=(C-p+1)\lambda_p P(s,m,p-1)\omega(s,m,p-1)+(m+1)\mu_m P(s,m+1,p)\omega(s,m+1,p)\\&\qquad+(p+1)\mu_p P(s,m,p+1)\omega(s,m,p+1)+\lambda_s P(s-1,m,p)\omega(s-1,m,p)\end{aligned}\tag{12}$$

$$[p\mu_p+m\mu_m+(C-p)\lambda_p]P(s,m,p)\omega(s,m,p)=(C-p+1)\lambda_p P(s,m,p-1)\omega(s,m,p-1),\tag{13}$$

$$\begin{aligned}&[p\mu_p+m\mu_m+(C-p)\lambda_p]P(s,m,p)\omega(s,m,p)\\&\quad=(C-p+1)\lambda_p P(s,m,p-1)\omega(s,m,p-1)+\lambda_m P(s,m-1,p)\omega(s,m-1,p)\end{aligned}\tag{14}$$

$$\begin{aligned}&[p\mu_p+s\mu_s+m\mu_m+(C-p)\lambda_p+\lambda_s]P(s,m,p)\omega(s,m,p)\\&\quad=\lambda_s P(s-1,m,p)\omega(s-1,m,p)+(C-p+1)\lambda_p P(s,m,p-1)\omega(s,m,p-1)\\&\qquad+(s+1)\mu_s P(s+1,m,p)\omega(s+1,m,p)+(p+1)\mu_p P(s,m,p+1)\omega(s,m,p+1)\end{aligned}\tag{15}$$

$$\mu_{\text{service}}(s,m,p)=(C-m-p)\mu_s+m\mu_m+p\mu_p.\tag{16}$$

$$\lambda_{\text{buffer}}(s,m,p)=\frac{(C-p+1)\lambda_p P(s,m,p-1)\omega(s,m,p-1)}{P(s,m,p-1)\omega(s,m,p-1)+P(s,m-1,p)\omega(s,m-1,p)}+\frac{(C-m-p+1)\lambda_m P(s,m-1,p)\omega(s,m-1,p)}{P(s,m,p-1)\omega(s,m,p-1)+P(s,m-1,p)\omega(s,m-1,p)}.\tag{17}$$

Averaging over all states (s,m,p) with $s+m+p>C$, the mean delay of ongoing secondary calls spent in the buffer, $T_{\rm delay}$, is obtained as

$$T_{\rm delay} = \frac{\displaystyle\sum_{\substack{(s,m,p)\in\Omega\\ s+m+p>C}} (s+m+p-C)T_{\rm delay}(s,m,p)P(s,m,p)}{\displaystyle\sum_{\substack{(s,m,p)\in\Omega\\ s+m+p>C}} (s+m+p-C)P(s,m,p)}. \quad (19)$$

B. "Greedy" Malicious Users

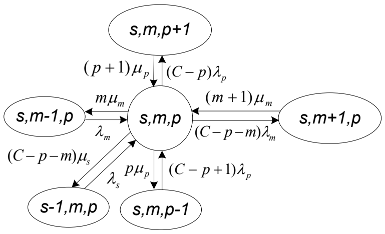


Fig. 4. State transition diagram of the 3D-CTMC for the DSA network carrying non-real-time traffic, with greedy malicious users, when $m+p<C$ and $s+m+p=C$. When $s+m+p<C$, the state transition diagram is same as Fig. 2. When $m+p<C$ and $s+m+p>C$, the state transition diagram is same as Fig. 3(a). When $m+p=C$, the state transition diagram is same as Fig. 3(d).

Greedy malicious users behave like secondary users when the system is not full, i.e., they also make requests for channels when channels are available. When the system is full, they launch PUEA as obstructive malicious users do. Thus, the system with greedy malicious users and that with obstructive malicious users have the same state transition diagrams when $m+p<C$ and $s+m+p>C$ and when $m+p=C$, as shown in Figs. 3(a) and 3(d), respectively. When $m+p<C$ and $s+m+p=C$, the state transition diagram for the system with greedy malicious users is different because greedy malicious users also make call attempts with arrival rate λ_m, as shown in Fig. 4. When $s+m+p<C$, the state transition diagram is same as that of the system with real-time traffic, as shown in Fig. 2. The set of feasible states of the 3D-CTMC for the system with greedy malicious users is given by

$$\Theta = \{(s,m,p) \mid 0 \le s \le C, m \ge 0, p \ge 0 \text{ and } m+p \le C\}. \quad (20)$$

Let $\theta(s,m,p)$ denote an indicator function of Θ, given by

$$\theta(s,m,p) = \begin{cases} 1 & \text{if } (s,m,p)\in\Theta, \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

The LBE for the state transition diagram in Fig. 4, is then given by Eqn. (22). Therefore, the steady state probability, $P(s,m,p)$, $\forall(s,m,p)\in\Theta$, can be solved from the system of linear equations specified by Eqns. (8), (11), (14), (22) and the normalization condition, by replacing all indicator functions with $\theta(\cdot)$ defined in Eqn. (21). The mean delay, $T_{\rm delay}$, can then be obtained from Eqn. (19) by replacing Ω with Θ.

The discussion thus far for the derivation of $p_{\rm drop}$ and $T_{\rm delay}$ in Eqns. (7) and (19) is for DSA networks where secondary users never miss the primary users, i.e., the probability of missing primary users, $p_{\rm miss}=0$, and malicious users are always successful in launching PUEA, i.e., the probability of successful PUEA, $p_{\rm PUEA}=1$. It is possible to reduce $p_{\rm PUEA}$ (at the cost of increasing $p_{\rm miss}$), by deploying the individual detection mechanism we proposed in [9] and [10], the centralized protocol we proposed in [9], and the distributed protocol we proposed in [10]. The expressions for the call dropping probability, $p_{\rm drop}$, and the mean delay, $T_{\rm delay}$, specified in Eqns. (7) and (19), respectively, can still be applied by replacing λ_p by $\lambda_p(1-p_{\rm miss})$ and λ_m by $\lambda_m p_{\rm PUEA}$, where $p_{\rm PUEA}$ and $p_{\rm miss}$ correspond to the probability of successful PUEA and the probability of missing the primary user, yielded by the respective mechanisms. The parameters for the system with no malicious users can be obtained by setting $\lambda_m=\mu_m=p_{\rm miss}=p_{\rm PUEA}=0$.

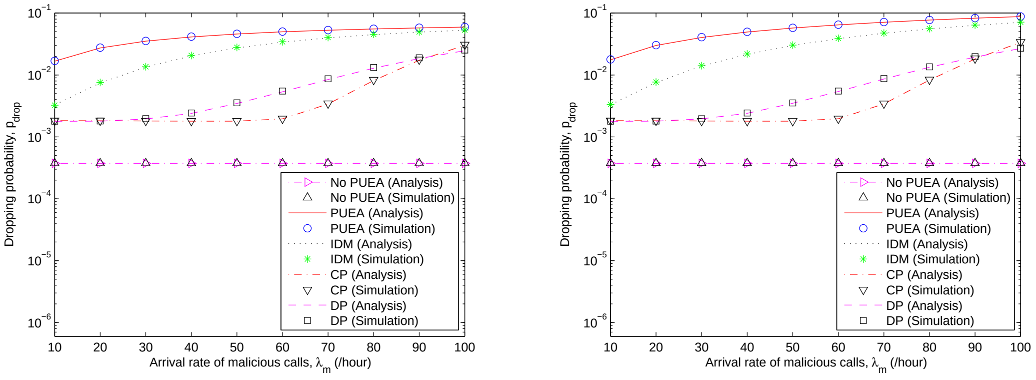
V. Numerical Results

We consider a DSA network with $C=5$ channels. The arrival rate of primary calls, λ_p, is 1/hour, and the arrival rate of secondary calls, λ_s, is 300/hour. We consider the mean of the holding time of primary calls $1/\mu_p=24$ seconds, and fix both the mean of the holding time of secondary calls and that of malicious calls at $1/\mu_s=1/\mu_m=36$ seconds [21]. We vary the number of malicious users in the system such that the arrival rate of malicious calls, λ_m, varies from 10 to 100 calls per hour. Our simulations were written in C++ and run on the UBUNTU Linux platform. We simulate a real system with primary, malicious and secondary traffic arrival and departure. The sketch of our simulations is described as follows.

1) Generating traffic for primary, malicious and secondary users, respectively.
2) Assigning channel for each call and implementing necessary spectrum handoff if secondary users are forced to leave their channels by primary users or malicious users successfully launching PUEA.
3) Constructing and maintaining a departure queue by adding newly arrived calls to it, according to calls' departure epochs. This departure queue represents the active calls on the channels. The calls are removed from the departure queue and considered finished if the current time passes their departure epoch.
    a) For real-time traffic, active secondary calls are dropped if the departure queue has the same number of calls as the number of channels and a new primary call or PUEA arrives. The number of dropped calls is counted, and the dropping probability is calculated accordingly.
    b) For non-real-time traffic, constructing and maintaining a buffering queue by adding buffered secondary calls to it, according to calls' arrival epochs in the buffer. Their departure epochs in the buffer are also recorded, and the delay they spend in the buffer before moving to the departure queue is calculated accordingly.

We compare the results obtained from the analysis with that by simulations. The results for the performance of secondary network with real-time traffic and non-real-time traffic are discussed in Sections V-A and V-B, respectively.

$$[p\mu_p + (C-p-m)\mu_s + m\mu_m + (C-p)\lambda_p + (C-p-m)\lambda_m]P(s,m,p)\theta(s,m,p) = (C-p+1)\lambda_p P(s,m,p-1)\theta(s,m,p-1)$$
$$+(m+1)\mu_m P(s,m+1,p)\theta(s,m+1,p) + (p+1)\mu_p P(s,m,p+1)\theta(s,m,p+1) + \lambda_m P(s,m-1,p)\theta(s,m-1,p) + \lambda_s P(s-1,m,p)\theta(s-1,m,p). \quad (22)$$



(a) Obstructive malicious users.

(b) Greedy malicious users.

Fig. 5. Call dropping in secondary networks carrying real-time traffic. The descriptions of the legends are provided in Section V-A.

A. Real-time Traffic

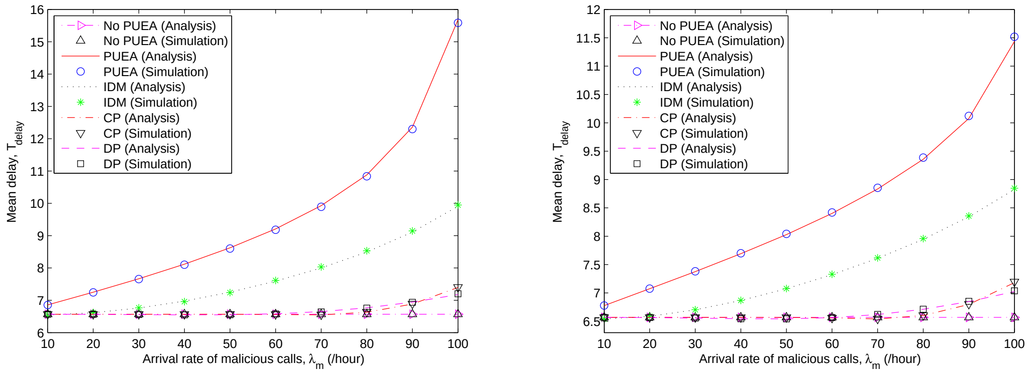
Fig. 5 presents the call dropping performance of the secondary network with real-time traffic, when deploying the protocols we proposed in [9] and [10] to mitigate PUEA. It is observed that the analytical results closely match the simulations. The legends in Figs. 5(a) and 5(b) are explained as follows. “No PUEA (Analysis)” and “No PUEA (Simulation)” represent the performances of secondary users when there is no PUEA (i.e., $p_{\mathrm{PUEA}} = p_{\mathrm{miss}} = 0$). “PUEA (Analysis)” and “PUEA (Simulation)” represent the results when the malicious users can always launch successful PUEA and secondary users use no protocol or other mechanisms to detect PUEA (i.e., $p_{\mathrm{PUEA}} = 1$ and $p_{\mathrm{miss}} = 0$). “IDM (Analysis)” and “IDM (Simulation)” represent the performance of secondary users when they make individual decisions on PUEA according to the individual detection mechanism described in [9] and [10]. “CP (Analysis)” and “CP (Simulation)” represent the results obtained when secondary users deploy the centralized protocol described in [9] to mitigate PUEA. “DP (Analysis)” and “DP (Simulation)” represent the results obtained when secondary users deploy the distributed protocol described in [10], to mitigate PUEA.

Fig. 5(a) depicts the call dropping performance of secondary network with real-time traffic, when PUEA is launched by obstructive malicious users. It is observed that PUEA launched by obstructive malicious users can increase the dropping probability significantly. For example, when $\lambda_m = 100$, the dropping probability is increased from about 0.0004 without PUEA to 0.06 with PUEA by obstructive malicious users, which is an increase by two orders of magnitude. This corresponds to an increase in the number of dropped calls in the system, N_{drop}, by two orders of magnitude, because $N_{\mathrm{drop}} = p_{\mathrm{drop}} N_{\mathrm{calls}}$, where N_{calls} is the number of admitted calls in the system. It can also be seen from Fig. 5(a) that the call dropping performance exhibits significant difference when different PUEA detection and mitigating mechanisms are used. While the individual detection mechanism reduces the number of dropped calls by about 80% for low traffic loads of malicious users, the performance is still poor for large traffic loads. The centralized protocol we proposed in [9] can reduce the number of dropped calls by about one order of magnitude for low malicious traffic loads. For large malicious traffic loads, the improvement obtained by deploying the centralized protocol is about 48%. The distributed protocol we proposed in [10] can provide a similar improvement on the call dropping performance for low malicious traffic loads. For large malicious traffic loads, the distributed protocol provides an improvement of about 58%.

Fig. 5(b) presents the call dropping performance of secondary network with real-time traffic, when PUEA is launched by greedy malicious users. It is observed that PUEA launched by greedy malicious users can also increase the number of dropped calls by up to two orders of magnitude. From Figs. 5(a) and 5(b), it is observed that greedy malicious users cause larger call dropping due to PUEA than obstructive malicious users. This is because, greedy malicious users also use the channels for themselves in addition to launching PUEA, which causes more secondary calls to be blocked and fewer secondary calls to be admitted into the system. Since fewer calls are admitted in the system, the ratio of the number of dropped calls to the number of admitted calls, i.e., the dropping probability, increases. Our protocols in [9] and [10] also result in significant improvements on the call dropping performance for the case when malicious users exhibit greedy behavior. In particular, the improvement is of the order of about 70% for $\lambda_m \geq 90$ and 90% for $\lambda_m \leq 20$.

B. Non-real-time traffic

Fig. 6 presents the mean delay suffered by a secondary network carrying non-real-time traffic. The legends in Fig. 6



(a) Obstructive malicious users.

(b) Greedy malicious users.

Fig. 6. Mean delay in secondary networks carrying non-real-time traffic. The descriptions of the legends are provided in Section V-A.

follow those in Fig. 5. Fig. 6(a) depicts the performance for the system with obstructive malicious users. It can be seen that PUEA launched by obstructive malicious users can increase the mean delay significantly. For example, when $\lambda_m = 100$, the mean delay is increased from about 6.6 seconds without PUEA to about 15.6 seconds with PUEA, i.e., a factor larger than two. It is also observed from Fig. 6(a) that both the centralized and distributed protocols can reduce the delay caused by PUEA significantly, by providing almost the same mean delay as that of a system with no PUEA when $\lambda_m \leq 70$. For large malicious traffic loads, the centralized protocol proposed in [9] and the distributed protocol proposed in [10] reduce the mean delay from 15.6 seconds to 7.4 seconds, i.e., by about 52%.

Fig. 6(b) depicts the delay performance of secondary network with greedy malicious users. It is observed that PUEA launched by greedy malicious users can increase the mean delay by up to 75% if no mitigating mechanisms are implemented. Note that, compared to Fig. 6(a), the mean delay caused by greedy malicious users is smaller than that caused by obstructive malicious users. This appears to be contrary to the behavior observed in Section V-A, where greedy malicious users resulted in a higher dropping probability. However, the reason for this behavior is as follows. Greedy malicious users use the channels themselves for their own communications and hence cause more secondary calls to be blocked and fewer to be admitted into the system, thus, causing fewer secondary calls to be buffered later on due to PUEA. Let the number of admitted secondary calls be N and $\hat{N}$ when malicious users operate in an obstructive and a greedy manner, respectively. As mentioned above, $\hat{N} < N$. Let d_i denote the delay that the i^{th} buffered call suffers. The mean delay in the system with obstructive malicious users, T_{obs}, is $T_{\text{obs}} = \frac{1}{N}\sum_{i=1}^{N} d_i$ and that in the system with greedy malicious users, T_{gr}, is $T_{\text{gr}} = \frac{1}{\hat{N}}\sum_{i=1}^{\hat{N}} d_i$. It is noted that the i^{th} buffered call has to wait till the previous $i-1$ calls leave the buffer. Therefore, $d_i \sim d_1$, i.e., $T_{\text{obs}} \sim (N+1)d_1$ and $T_{\text{gr}} \sim (\hat{N}+1)d_1$. Since $\hat{N} < N$, $T_{\text{gr}} < T_{\text{obs}}$, i.e., the mean delay in the system with greedy malicious users is less than that in the system with obstructive malicious users. It can also been seen from Fig. 6(b) that both the centralized and distributed protocols have similar performances, and both protocols show great resilience to PUEA. For example, when $\lambda_m = 100$, the centralized protocol and the distributed protocol reduce the mean delay from 11.5 seconds to about 7 seconds, i.e., a reduction of about 38%. When the malicious traffic load is not high (e.g., when $\lambda_m \leq 70$), both the centralized protocol proposed in [9] and the distributed protocol proposed in [10] provide almost the same delay performance as that of a system with no PUEA. *This demonstrates the effectiveness of our protocols in [9] and [10] in terms of improvements on the performance of secondary network.*

It is observed from Figs. 5 and 6, that the centralized protocol we proposed in [9] yields the same or better network performance than the distributed protocol we proposed in [10], for most values of malicious traffic loads. This is because, the distributed protocol uses only local information (the spectrum decisions of a secondary user and its one-hop neighbors) where as the centralized protocol utilizes global information (spectrum decisions of all the users in the system). However, the centralized protocol requires a centralized controller that can obtain spectrum sensing decisions from all the users in the system [9]. The distributed protocol only requires coordination between a user and its one-hop neighbors [10]. Our results in this paper provide a means to trade-off performance for complexity in the design of DSA networks.

VI. Conclusion

We presented the first study of the impact of PUEA on the performance of secondary DSA networks. A three dimensional continuous time Markov chain (3D-CTMC) model was proposed to analyze the call dropping and delay performance of secondary networks carrying real-time and non-real-time traffic, respectively. We studied two types of malicious behavior, namely, greedy and obstructive.

PUEA was observed to result in one and half time larger number of dropped real-time traffic calls and 75% additional delay for non-real-time traffic. The centralized and distributed protocols we proposed in [9] and [10] reduce the number of dropped real-time traffic calls by up to one order of magnitude. Our protocols also provide almost the same delay performance as that of a system with no PUEA, given that malicious traffic load is low. When malicious traffic is high, our protocols provide an improvement by up to 54% on the delay performance. The centralized protocol we proposed in [9] yields better network performance than the distributed protocol we proposed in [10] at the cost of higher complexity. Our analysis in this paper provides a means to trade-off between performance and complexity in the design of DSA networks under PUEA. Topics for extension include the analysis of the system with mixed real-time and non-real-time traffic, the attacks and defense scenarios for such systems and traffic with multiple classes and QoS requirements.

Acknowledgment

This research was partially funded by NSF # 0917008 and NSF # 0916180 and partially funded by 2009-92667-NJ-IJ. The authors would like to thank the anonymous reviewers for their valuable comments which helped clarify details that were not so evident in the original manuscript.

References

[1] J. Mitola and G. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Personal Commun.*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[2] S. Haykin, "Cognitive radio: Brain empowered wireless communications," *IEEE Jl. on Sel. Areas in Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[3] I. F. Akyildiz, W. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Elsevier Jl. on Comp. Networks*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.

[4] R. Chen and J. M. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," *Proc., IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, Sep. 2006.

[5] R. Chen, J. M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Jl. on Sel. Areas in Commun.*, vol. 26, no. 1, pp. 25–37, Jan. 2008.

[6] S. Anand, Z. Jin, and K. P. Subbalakshmi, "An analytical model for primary user emulation attacks in cognitive radio networks," *Proc., IEEE Symposia of New Frontiers in Dynamic Spectrum Access Networks (DySPAN'2008)*, Oct. 2008.

[7] Z. Jin, S. Anand, and K. P. Subbalakshmi, "Detecting primary user emulation attacks in dynamic spectrum access networks," *Proc., IEEE Intl. Conf. on Commun. (ICC'2009)*, Jun. 2009.

[8] ——, "Mitigating primary user emulation attacks in dynamic spectrum access networks using hypothesis testing," *ACM SIGMOBILE Mobile Computing and Commun. Review*, vol. 13, no. 2, pp. 74–85, Apr. 2009.

[9] ——, "Robust spectrum decision protocol against primary user emulation attacks in dynamic spectrum access networks," *Proc., IEEE Global Commun. Conf. (GLOBECOM'2010)*, Dec. 2010.

[10] ——, "NEAT: A NEighbor AssisTed spectrum decision protocol for resilience against primary user emulation attacks," *Technical Report*, Dec. 2009. [Online]. Available: http://personal.stevens.edu/∼ksubbala/

[11] R. W. Thomas, R. S. Komali, B. J. Borghetti, and P. Mahonen, "A Bayesian game analysis of emulation attacks in dynamic spectrum access networks," *Proc., IEEE DySPAN'2010)*, Apr. 2010.

[12] C. Zhao, W. Wang, L. Huang, and Y. Yao, "Anti-pue attack base on the transmitter fingerprint identification in cognitive radio," *Proc., Wireless Commun., Networking and Mobile Computing Conf. (WiCOM'2009)*, Sep. 2009.

[13] Z. Chen, T. Cooklev, C. Chen, and C. Pomalaza-Raez, "Modeling primary user emulation attacks and defenses in cognitive radio networks," *Proc., IEEE Intl. Perf. Computing and Commun. Conf. (IPCCC'2009)*, pp. 208–215, Dec. 2009.

[14] H. Li and Z. Han, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part I: Known channel statistics," *IEEE Trans. on Wireless Commun.*, vol. 9, no. 11, pp. 3566 – 3577, Nov. 2010.

[15] ——, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part II: Unknown channel statistics," *IEEE Trans. on Wireless Commun.*, vol. 10, no. 1, pp. 274 – 283, Jan. 2011.

[16] A. Asterjadhi and M. Zorzi, "JENNA: a jamming-evasive network coding neighbor-discovery algorithm for cognitive radio networks," *IEEE Wireless Commun.*, vol. 17, no. 4, pp. 24–32, Aug. 2010.

[17] S. Chen, K. Zeng, and P. Mohapatra, "Hearing is believing: Detecting mobile primary user emulation attack in white space," *Proc., IEEE Intl. Conf. on Comp. Commun. (INFOCOM'2011) Mini Conf.*, Apr. 2011.

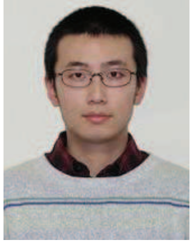
[18] T. Yang, H. Chen, and L. Xie, "Cooperative primary user emulation attack and defense in cognitive radio networks," *Proc., WiCOM'2011*, Sep. 2011.

[19] L. Dong, Z. Han, A. P. Petropulu, and H. V. Poor, "Cooperative jamming for wireless physical layer security," *Proc., IEEE Workshop on Statistical Signal Processing (SSP'2009)*, Sep. 2009.

[20] B. Wang, Z. Ji, K. J. R. Liu, and T. C. Clancy, "Primary-prioritized Markov approach for dynamic spectrum allocation," *IEEE Trans. on Wireless Commun.*, vol. 8, no. 4, pp. 1854–1865, Apr. 2009.

[21] Z. Jin, S. Anand, and K. P. Subbalakshmi, "Performance analysis of dynamic spectrum access networks under primary user emulation attacks," *Proc., IEEE GLOBECOM'2010*, Dec. 2010.

[22] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*, 3rd ed. Wiley, 1998.



Z. Jin (SM'08) is currently a Ph.D. student in the Department of Electrical and Computer Engineering at Stevens Institute of Technology, Hoboken, New Jersey. He received the B. E. degree in the Department of Information Engineering from Xi'an Jiaotong University, Xi'an, China, in 2006. His current research interests include cognitive radio network security and denial-of-service attack in dynamic spectrum access networks. He is a student member of IEEE, IEEE communications society and IEEE computer society. Further information can be found at http://personal.stevens.edu/∼zjin/.

S. Anand (M '05) received his B.E. degree from the College of Engineering Guindy, Chennai, India, in 1995 and his M.E. degree and Ph.D. degree from the Indian Institute of Science, Bangalore, India, in 1998 and 2003, respectively. During 2004-2006, he was with Samsung India Software Operations Pvt. Ltd. He is currently a research associate at the Department of ECE in Stevens Institute of Technology. His current areas of research include dynamics of social networks, spectrum management and security in dynamic spectrum access networks and covert timing channels. Dr. Anand received the Seshagiri Kaikini medal for the best Ph.D. dissertation in the electrical sciences division, Indian Institute of Science for the academic year 2003-2004. He has represented Samsung Electronics in 3GPP SA2 and IEEE 802.20 standardization meetings. For more information: http://personal.stevens.edu/∼asanthan.

K. P. (Suba) Subbalakshmi is an Associate Professor in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. Her research interests are in cognitive radio network security, wireless security, steganography and staganalysis as well as Internet forensics. Her research is supported by US NSF, US AFRL, US Army and other DoD agencies. Her research has led to the development of several digital forensic software tools that have been delivered to government agencies and industry. She is the Chair of the Security Special Interest Group, IEEE Multimedia Communications Technical Committee, COMSOC. She is the organizing chair of the Cognitive Networks track of the Symposium on Selected Areas of Communications, IEEE International Conference on Communications, 2009. She has chaired several conferences and serves on the editorial board of several journals. Suba is a Co-Founder as well as co-CTO of inStream Media, LLC, an interactive media company. For more information: http://personal.stevens.edu/kpsuba.