Stochastic model for Cognitive Radio Networks under jamming attacks and honeypot-based prevention *

Suman Bhunia[1], Xing Su[2], Shamik Sengupta[1], and Felisa Vázquez-Abad[2]

[1] University of Nevada, Reno, USA, {sbhunia, ssengupta}@unr.edu
[2] City University of New York, NY, USA {xsu, FVazquez-Abad}@gc.cuny.edu

Abstract. Limited and dynamically available resources and "no right to protection from interference" in the open access dynamic spectrum access model bring forth a serious challenge of sustenance among the secondary networks and make them more susceptible to various spectrum etiquette attacks. Among these, the most common are jamming-based denial of service (DoS) attacks, which result in packet loss. The concept of a honeypot node or honeynode has been explored for wireless networks and has shown to be effective in attracting attacks, thus deterring the jammers from productive nodes. Yet a single dedicated honeynode, on account of its permanent idleness, is wasteful of an entire node as resource. In this paper, we seek to resolve this dilemma by dynamically selecting the honeynode for each transmission period, and we explore various methods of doing so. To begin with, we develop the first comprehensive queuing model for CRNs, which pose unique modeling challenges due to their periodic sensing and transmission cycles. We then build a simulation of CRNs under attack from jammers, introduce a series of strategies for honeynode assignment to combat these attacks, and assess the performance of each strategy. We find that the predictions of our mathematical model track closely with the results of our simulation experiments.

Keywords: Cognitive Radio, Honeypot, Stochastic Model, Queuing theory, queue with vacation

1 Introduction

The conventional static spectrum allocation policy has resulted in suboptimal use of spectrum resource leading to over-utilization in some bands and under-utilization in others [1]. Such "artificial spectrum starvation" has led to the recent spectrum policy reform by the U.S. Federal Communication Commission (FCC). This reform, known as dynamic spectrum access (DSA), is expected to be achieved via the recently proposed concept of the cognitive radio (CR) [1]. The IEEE 802.22 is an emerging standard for CR-based wireless regional area networks (WRANs) [2]. The IEEE 802.22 standard aims at using DSA to allow the unused, licensed TV frequency spectrum to be used by unlicensed users on a non-interfering basis. To protect the primary incumbent services, IEEE 802.22

* This research was supported by NSF CAREER grant CNS #1346600.

devices (e.g., base station and consumer premise equipment) are required to perform periodic spectrum sensing and evacuate promptly upon the return of the licensed users (as depicted in Fig. 1).

Even though the primary user protection mechanisms have been proactively specified, neither the secondary-secondary interaction mechanisms nor the protection of secondary devices/networks have been specifically defined or addressed in IEEE 802.22 standard. The "open" philosophy of the cognitive radio paradigm makes such networks susceptible to attacks by smart malicious user(s) that could even render the legitimate cognitive radios spectrum-less. Due to software reconfigurability, cognitive radios can even be manipulated to disrupt other secondary cognitive radio networks or legacy wireless networks with even greater impact than traditional hardware radios. There are several ways in which operations of cognitive radio networks can be jeopardized. Cognitive disruption through malicious channel fragmentation/aggregation/bonding, primary user emulation attack, multi-channel jamming or spectrum stealing through induced attack are just some of the examples that can severely cripple the cognitive radio networks. A number of defenses against such attacks have been attempted, among which the most effective is the so-called honeypot[3]. The concept of "Honeypot" in Cybercrime is defined as "a security resource whoś value lies in being probed, attacked or compromised". In Cybercrime defense, honeypots are being used as a camouflaging security tool with little or no actual production value to lure the attacker giving them a false sense of satisfaction thus bypassing (reducing) the attack impact and giving the defender a chance to retrieve valuable information about the attacker and attack activities. This node is called honeynode. Misra et al.[4] and Yek et al.[5] have proposed a honeypot-based attack detection and prevention mechanism for WLANs.

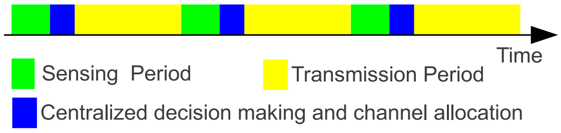


Fig. 1: Time domain representation of Cognitive Cycle

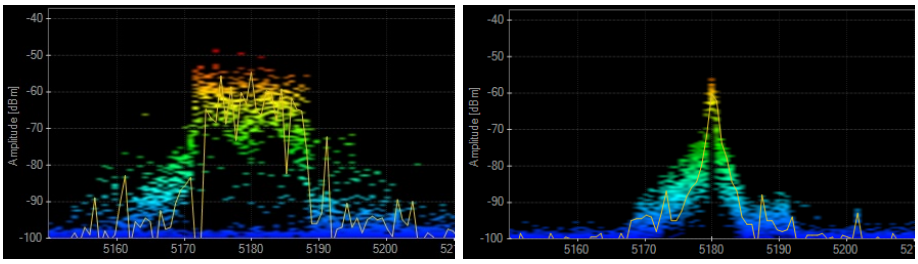
Several practical challenges however need to be co-opted and addressed to achieve such goal in uncoordinated DSA networks. Honeypot is not a "free-ride". While using a SU as honeynode potentially make the CRN robust but it also sacrifices the effective system throughput/QoS to some extent. How would the honeynode be chosen then? Who will be responsible ("honeynode" selection) for auxiliary communications and monitoring in honeynodes? To answer the above questions, we must however first understand the complexity of the CRN traffic behavior under DSA scenario (periodic sensing and transmission with channel switching due to primary arrival resulting in queuing with vacation). Consider a scenario wherein a user is conducting a number of simultaneous transmissions - say, videoconferencing, downloading a file using FTP, downloading large files from numerous seeders via torrent, periodically downloading data from a server with a widget, continuously synchronizing with a cloud server, and more. All these applications generate packets randomly and independently of other applications. The complex nature of data traffic makes it difficult to analyze the Quality of Service (QoS). CRNs, meanwhile, exhibit unique behavior patterns that remains yet to be investigated by any mathematical model. For example, the periodic sensing by SUs forces interruption of transmission, affecting end-to-end QoS by imposing delay and jitter on packet processing. The dynamic nature of spectrum access also introduces a particular complexity to CRN behavior. Thus, a major goal of this work is to model CRN service using stochastic analysis and use our model to estimate baseline performance indicators. Only then we can assess the performance of experimental honeypot selection algorithms.

The rest of the paper proceeds as follows: In Section 2, we discuss the motivation for our work, i.e. DoS attacks and honeypot limitations. Section 3 presents a mathematical model to estimate CRN performance using queue with fixed server vacation. We describe our simulation results and analysis in Section 4. Section 5 concludes the paper.

2 Motivation

2.1 Jamming Attack

In traditional wireless networks, the user of a particular channel has proprietary access to that channel and thus has the right to penalize any trespassers. The threat of penalty can discourage potential attackers. However, if a channel is being accessed by a CRN, the SUs are only borrowing the channel, and they have no grounds from which to fend off attackers. While PUs are able to discourage attackers, SUs are left vulnerable against malicious jamming / disruptive attacks [6]. Jamming can be broadly categorized into two types [4], [7]. In *physical layer jamming*, the attacker jams the channel of communication by sending strong noise or jamming signals. The *datalink / MAC layer jamming* targets several vulnerabilities present in the MAC layer protocol.



(a) Normal communication

(b) Jamming Signal

Fig. 2: PSD for data communication and jamming signal

To see the effect of jamming, we run an experiment in our lab. Two computers are configured to communicate over a WLAN (IEEE 802.11-a, channel 36, central frequency: 5.18 GHz) with 54 mbps data rate. As the units are communicating in full throttle, we observe the Power Spectral Density (PSD) over the channel using the Wi-spy spectrum analyzer [8]. The PSD for normal communication (without jamming) can be seen in Fig. 2a, which shows that the transmission is using a 20 MHz channel and leaking energy to neighboring channels. Then we begin transmitting a very narrow band jamming signal of 2MHz from a third machine, also on channel 36. At the presence of the jamming signal, the actual transmission of the WLAN was stopped totally as can be observed in Fig. 2b: only the jamming signal is getting through. As the bandwidth is very narrow, this jamming costs very little power. However it is taking advantage of the vulnerability of IEEE 802.11 MAC (sensing the channel before transmission). When the legitimate transmitter senses that there is some energy on the channel, it refrains from transmission. In effect, the attacker successfully jams the channel with very little cost, denying service to the SU. Irrespective of the jamming technique, a target SU suffers a significant amount of data or packet loss and sometimes completely loses the channel. The CRN is a next-generation intelligent network, and it should incorporate a mechanism to mitigate, avoid or prevent these attacks.

2.2 Use of Honeypot in avoiding attacks

In CRN, honeypot mechanism can be deployed in SUs which act as normal data transmission. In order to attract the attacker to this channel, the honeynode tries to obtain the attacker's fingerprint. The authors of [3] and [5] have implemented honeypots to detect attacks on server and access-points in regular WLAN. They introduced fake access-point as honeypot nodes to guess the fingerprint of the attack. In [9] a learning mechanism is proposed to mitigate spoofing DoS attacks. Misra *et al* [4] implemented a channel surfing approach to mitigate jamming-based DoS attacks. Upon detecting an attack, the victim switches its transmission to a different channel.

When a honeypot mechanism is used consistently, the attacks will sometimes be trapped by honeypot, and sometimes can attack legitimate SUs. We define one parameter, that we call the *attractiveness of honeypot* (ξ) as the probability that the honeynode is the one to be attacked, conditional on observing a jamming attack [3]. Honeypot ensures less data loss at the cost of end-to-end delay. Some application can tolerate data loss but not delay and others the opposite. The goal is to build a mathematical model that can estimate system performance before we actually apply honeypot. If honeypot degrades performance very badly and the system can tolerate some data loss then we can opt for not applying. The end-to-end delay in CR is mainly affected by queuing delay as processing, transmission and propagation delays are negligible compared to queuing delay. Our theoretical model focus on determining queuing delay. Then we concentrate on honynode selection mechanism to achieve better over-all system performance.

[3] Understandably each honeypot strategy has a different ξ, and ξ also depends on the attacker's strategy. Calculating ξ on different strategy is beyond the scope of this paper.

3 Mathematical Model of SU

3.1 Queuing Characteristics

In earlier telecommunication networks, voice packets were generated at fixed rates or at fixed burst sizes [10], [11]. For this kind of system the inter-arrival time is fixed and the value depends on the codec (voice digitization technique) used [10], [11] . Voice activity detection and Silence suppression techniques introduces randomness in packet arrival time. With the increase in usage of multimedia applications on smart-phones the nature of the traffic flow is very complex to model. Because of the independence between sources, a memoryless inter-arrival time may be a good model. This observation is supported by statistical analysis. The studies carried out in various experiments [12], [13], [14], [15] have concluded that when many different applications are merged, the packet arrival process tends to follow Poisson process. Fig. 3 provides an depiction of how packets from different applications flow to a queue. We use λ_i to denote the rate of the Poisson process of packet arrivals at SU labeled i, and $\{N_i(t); t \geq 0\}$ to denote the corresponding arrival process. When a single queue is analyzed, we drop the subindex i.

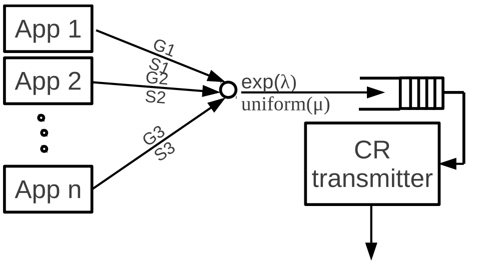


Fig. 3: Depiction of how packets are flowed to Queue

Data to be transmitted arrives at the different servers (corresponding to different SUs in a CRN) in packets. Each SU is modeled as a FCFS (first come-first served) queue with one server. Packets arrive according to a marked Poisson process with rate λ and "marks" specifying the packet size. In our model, the marks $\{Y_1, Y_2, \ldots\}$ are independent and identically distributed uniform random variables. The aggregate data arrival rate is thus $\lambda\, \mathbb{E}(Y)$. Each SU can transmit at a fixed rate r. Therefore, the service time of a packet size Y_n is, $S_n = Y_n/r$ and it has uniform distribution $U(\ell_1, \ell_2)$, with mean $\mathbb{E}(S) = (\ell_1 + \ell_2)/2$ and potential service rate, $\mu = 1/\mathbb{E}(S)$

During the transmission periods of length T_t, the model corresponds to a $M/G/1$ queue where the service time of consecutive packets are independent and identically distributed. During the sensing period of length T_s (and the transmission periods when it is chosen as a honeynode) our server stops servicing the queue, which nonetheless continues to accumulate arriving packets. Because the sensing period has deterministic size and frequency, the server model does not conform to the usual server with vacations ([16], [17]) where the server has the option to take vacations only at the end of busy periods. Instead, the sensing period acts as a "priority" customer whose inter-arrival rates and service times are deterministic. Under the assumption that the buffer is infinite, the effect of an attack during a transmission period when the SU is not a honeynode is that all packets transmitted in that slot are lost.

The two performance criteria of interest are the (stationary) average waiting time in queue per packet W_q, and the average packet drop rate PDR. In the case of infinite buffer PDR_i is also the long term probability that i'th SU is attacked, that we call θ_i.

3.2 Queuing Model with Vacations

In this section we assume that one SU is chosen to be "sacrificed" as a honenode at every transmission period. If a SU is chosen as a honeynode, then all the new arriving packets join the queue and wait until the next transmission period where the SU is not chosen as a honeynode. It is conceivable that intelligent policies include the possibility of regular transmissions without honeynode, and transmission periods with multiple honeynodes, but as a first step, we impose here that a SU (only one) be chosen at every transmission period. Let's consider there are N SUs in the CRN. Because the CRN works by assigning channel to SUs, we assume here w.l.o.g. that the there are more free channels than number of nodes or SUs in the network. Under the round robin policy, SUs are chosen as honeynode in a circular fashion, so one out of every N transmission periods becomes an "extended vacation" for the server. A random policy assigns the honeynode to i'th SU with probability p_i, independently of past assignments and of the state of the CRN. A random policy that targets fairness could be used by attempting to balance the utilization factors. Other random policies may be justified when they are used to increase the attractiveness of the honeypot, as we will discuss at the end of the paper. In a state dependent policy the centralized controller picks the SU according to the current values of the queues. These policies target a decrease in mean queuing delay.

We now calculate the *effective utilization factor* for the queue under the random policy. The amount of service time that must be postponed at the start of a sensing period is either 0 (when the server is idle at time of sensing) or it has the value of the random variable $\tilde{S}$ representing the fraction of service time that must be postponed. In steady state, if $\rho = \mathbb{P}(\text{the server is busy})$ then the server is not idle with probability ρ. When in steady state, the start of a sensing period may fall in in any subinterval of the current packet's service time with equal probability. Thus $\tilde{S} \sim U(\ell_1, S)$, where $S \sim U(\ell_1, \ell_2)$[4] represents the (random) amount of current service when the server is not idle.

Result: Assume that each queue is stable and call X the fraction of service that must be postponed at the start of a sensing period in steady state. Then:

$$\mathbb{E}(X) = \rho \left(\frac{\mathbb{E}(S) + \ell_1}{2} \right). \tag{1}$$

[4] The symbol "$\sim$" means "has distribution".

Let $a \geq 0$ be a non-negative constant. Then:

$$\mathbb{E}((a+X)^2) = a^2 + \rho\left(a\ell_1 + \frac{\ell_1^2}{3} + \frac{(3a+1)\ell_1}{3}\,\mathbb{E}(S) + \frac{\mathbb{E}(S^2)}{3}\right). \tag{2}$$

This result follows from straightforward calculation. Assuming that the queues are stable, the effective service rate for each of the SUs satisfies the equation:

$$\mu_i' = \mu\left(\frac{T_t - \rho_i\, 0.5(\mathbb{E}(S)+\ell_1)}{T_s + T_t}\right)(1-p_i), \quad \rho_i = \frac{\lambda_i}{\mu_i'},$$

which yields an implicit equation for μ':

$$\mu_i'(T_t + T_s) = \mu\left(\mu T_t - 0.5\frac{(\mathbb{E}(S)+\ell_1)\lambda_i}{\mu_i'}\right)(1-p_i) \tag{3}$$

Solving the quadratic equation (3) gives values that depend on λ_i. In particular, if all SUs have equal probability of being chosen (p_i), then the reduced service rate is the same as in the round robin policy.

Stationary policies. We now provide an analysis of the stationary queuing delay for the random (or round robin) policies. In this section we use our previous results that $\mathbb{E}(\tilde{S}) = 0.5\,\rho_i\,\mathbb{E}(S)$ for each of the nodes. The analysis is done for each queue, and the subscript i will be dropped from our notation.

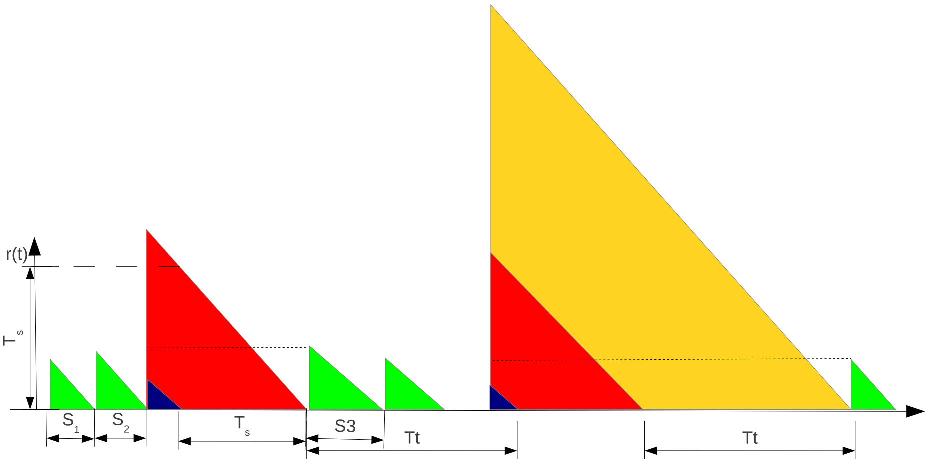


Fig. 4: Path of the residual service of customer currently in service, for the CRN model honeypot. Here, green indicates packet transmission, red is Channel sensing, yellow is Serving as honeypot and blue indicates, the node postpone current packet transmission as it can not be done within transmission period.

Theorem 1. *Suppose that i'th SU has incoming rate λ, and that it is chosen as a honeynode independently of the state of the queue, with long term frequency of*

p. Furthermore, assume that this queue is stable and ergodic and let X satisfy equations (1) and (2). Then the stationary delay in queue is:

$$W_q = \frac{R}{1 - \lambda \mathbb{E}(S)(1 + \varDelta)}, \tag{4}$$

where the stationary residual service time is:

$$R = \frac{\lambda\, \mathbb{E}(S^2)}{2} + \frac{\mathbb{E}(T_s + X)^2\, (1 - p) + \mathbb{E}(T_s + T_t + X)^2\, p}{2(T_s + T_t)} \tag{5}$$

and the correction factor for the vacations is:

$$\varDelta = \frac{T_s + \lambda \mathbb{E}(X) + pT_t}{(1 - p)(T_t - \mathbb{E}(X))}.$$

Proof. We use the residual service approach [17, 16] to calculate the stationary average delay in queue (assuming that it is well defined), as follows. Sensing periods of length T_s and honeynode periods of length T_t correspond to a "vacation" of the server, and are followed by transmission times of length T_t, during which consecutive packets with varying sizes enter service. Unlike the usual analysis of servers with vacations, here a vacation starts at deterministic times, and not necessarily at the end of busy periods. When not idle, the server can be in three different states: (a) a packet is being transmitted, (b) the server is on vacation, or (c) the current transmission is postponed and the server is waiting for the vacation.

Fig. 4 shows a typical path of the *residual* time until the completion of the current task (a service, a vacation, or the wait for the vacation), that we call $r(t)$. Under ergodicity, the stationary average residual service is the same as the long term average, given by:

$$R = \lim_{t\to\infty} \frac{1}{t} \int_0^t r(t)\, dt = \lim_{t\to\infty} \frac{1}{t} \left[\sum_{i=1}^{M(t)} \frac{S_i^2}{2} + \sum_{i=1}^{V(t)} \frac{L_i^2}{2} \right] \tag{6}$$

where $M(t)$ is the number of arrivals that have entered service up to time t, $V(t)$ is the number of sensing periods up to time t, and L_i is the length of the i-th vacation. It follows that L_i are iid random variables with composite distribution: with probability $1 - p$ the vacation length is $T_s + X$, and with probability p it is $T_s + T_t + X$. For $k \geq 1$, let

$$\tau_k = \min(t > \tau_{k-1} \,:\, Q(t) = 0); \quad \tau_0 \equiv 0$$

be the consecutive moments when the queue empties. The stability assumption implies that the queue empties infinitely often (that is, the state $Q = 0$ is positive recurrent), so that $\tau_k \to +\infty$ with probability one. At these times, $M(\tau_n) = N(\tau_n)$, and $N(t)/t \to \lambda$, because $N(\cdot)$ is a Poisson process. Because the limit R (assuming that it exists) is the same if we consider any divergent subsequence, we can take the limit along the subsequence $\{\tau_k; k \geq 0\}$. In our model $V(t)/t \to (T_s+T_t)^{-1}$. Under ergodicity, long term averages are stationary averages, and

$$\mathbb{E}(L_i^2) = \mathbb{E}(T_s + X)^2(1-p) + \mathbb{E}(T_s + T_t + X)^2\, p,$$

where X satisfies (1) and (2). Using these results in (6) gives expression (5).

The rest of the argument is as follows. Because Poisson arrivals see time averages (PASTA), an arriving customer to a stationary queue will encounter N_q customers in queue, where N_q is a random variable that has the stationary distribution of the queue length. The average wait time is thus the sum of the expected service time of the N_q customers in queue, plus R, plus the contribution of the vacation periods during the waiting time. Call T the required service time for the customers in queue, then using Little's Law:

$$\mathbb{E}(T) = \mathbb{E}\left(\sum_{k=1}^{N_q} S_k\right) = \mathbb{E}(N_q \mathbb{E}(S)) = \lambda W_q \mathbb{E}(S).$$

Therefore, the stationary delay upon arrival at the queue will satisfy:

$$W_q = \lambda W_q \mathbb{E}(S) + R + v_s(\,T_s + \mathbb{E}(X)) + v_h\, T_t, \tag{7}$$

where v_s and v_h are the (expected) number of sensing and honeynode periods (respectively) that fall within the time required to transmit all the N_q customers in front of the new arrival. In the expression above we have used the fact that for every sensing period, the actual vacation time is not just T_s but we must add the lost time from the postponed service (if any). On average, the stationary contribution of this excess is $\mathbb{E}(X)$.

We now proceed to the calculation of v_s and v_h. In order to do so, we will use Wald's theorem. Given T, the actual number of (true) transmission periods required to provide the service for the N_q customers in queue is:

$$\nu_t = \min\left(n : \sum_{i=1}^{n}(T_t - X_i) \geq T\right), \tag{8}$$

where X_i is the fraction of postponed service at the i-th sensing period. This is a stopping time adapted to the filtration $\mathfrak{F}_n$ generated by $\{Z_i \equiv T_t - X_i, i \leq n\}$. In addition, Z_n is independent of $\mathfrak{F}_{n-1}$. For our model the random variables $\{Z_n\}$ are bounded, thus absolutely integrable. It is straightforward to verify that $\mathbb{E}(X_n \mathbf{1}_{\{\nu_t<n\}}) = \mathbb{P}(\nu_t < n)\, \mathbb{E}(X)$, and finally, $\mathbb{E}(\nu_t) < \infty$, which follows because $\nu_t \leq T/(T_t - \ell_2)$ w.p.1. Under these conditions, Wald's Theorem ensures that

$$\mathbb{E}\left(\sum_{i=1}^{\nu_t} Z_i\right) = \mathbb{E}(\nu_t)(T_t - \mathbb{E}(X)).$$

Rewrite (8) as: $\sum_{i=1}^{\nu_t} Z_i \leq T < \sum_{i=1}^{\nu_t+1} Z_i$ and take expectations to get:

$$\frac{\mathbb{E}(T)}{T_t - \mathbb{E}(X)} - 1 < \mathbb{E}(\nu_t) \leq \frac{\mathbb{E}(T)}{T_t - \mathbb{E}(X)}.$$

In stationary state, we use the approximation $\mathbb{E}(\nu_t) = \lambda W_q \mathbb{E}(S)/(T_t - \mathbb{E}(X))$. In order to calculate v_s and v_h we reason as follows: given the number of honeynode periods, the number of sensing periods is the number of true transmission periods required to exhaust the time T, plus v_h, that is:

$$v_s = \mathbb{E}(\nu_t) + v_h = \mathbb{E}(\nu_t) + p v_s, \implies v_s = \frac{\mathbb{E}(\nu_t)}{1-p}.$$

Replacing now these values in (7) and using $\mathbb{E}(T) = \lambda W_q \mathbb{E}(S)$, we obtain

$$W_q = \lambda W_q \, \mathbb{E}(S) \left(1 + \frac{T_s + \mathbb{E}(X)}{(1-p)(T_t - \mathbb{E}(X))} + p \frac{T_t}{(1-p)(T_t - \mathbb{E}(X))} \right) + R,$$

which yields (4), after some simple algebra.

3.3 State Dependent Policies

When a SU is chosen as a honeynode and it has an initial queue size of Q packets, the queue size at the beginning of the following transmission period is $Q + A$, where $A \sim \text{Poisson}(\lambda(T_s + T_t))$. For our model, we can solve recursions to approximate the probability that the queue empties within the following transmission period, or its expectation (details fall outside the scope of this paper and will be presented elsewhere). Strategies for honeynode allocation may include choosing the SU with largest probability of emptying. The particular case where all SUs have identical statistics (same arrival rates) is much simpler because it reduces to choosing the SU with minimal queue size, and no further calculations are necessary. However in the more realistic scenarios when λ_i are not only different but perhaps even slowly changing, it may prove essential to be able to calculate specific trade-offs between choosing honeynodes and avoiding attacks.

4 Simulation and Results

4.1 Simulation parameters and model

We coded a discrete event simulation written in Python in order to compare the three honeynode selection strategies mentioned in 3.2. All arrival rate (λ) are in millisecond domain and mean λ packets per millisecond. As a first step we consider in this paper equal arrival rates $\lambda_i = \lambda$ amongst the SUs. Under this assumption, the randomized and the queue dependent policies become a randomized policy with equal probabilities, and a minimum queue size policy, respectively. The data for our model is given in Table 1.

In all our simulations, we use the technique of antithetic random variables (ARN) for increased precision. For the infinite queue model where waiting and loss are monotone functions of the inter-arrival and service variables, ARN ensures variance reduction [18] (we used the inverse function method for generating

Table 1: Simulation Parameters

Parameter	Symbol	Value
Number of SU	N	20
Packet Service Time	S_n	$\sim U(0.1, 1.7)$ msec
Sensing Period	T_s	50 ms
Transmission Period	T_t	950 ms
Number of attacks / slot		1
Number of honeynodes /slot		20
Number of replication		30
Simulated time		5000000 msec
Warm-up time		100000 msec

random variables). For the finite buffer model, because some packets may be lost, it is no longer true that larger inter-arrivals (service times) always have a decreasing (increasing) effect on the delay. Although the theory does not ensure variance reduction for the finite buffer model, we verified by experimentation that this is the case.

Using a simulated time of $50,000$ time slots means that the number of packets served in each SU is a random variable. For each replication of the simulation, we discarded the "warm up" data corresponding to the first 100 time slots. Preliminary simulations were used to choose these numbers, testing for stationarity and a satisfactory precision. For each replication or run of $50,000$ slots we estimated the quantity $(1/N)\sum_i^n W_q(i)$ that we call the average wait time in the queues. We then used 30 independent replications to calculate 95% confidence intervals of the form:

$$\bar{W}_q \pm t_{29,0.975}\sqrt{\frac{\widehat{\mathbb{V}\mathrm{ar}(W_q)}}{30}},$$

where $\widehat{\mathbb{V}\mathrm{ar}(W_q)}$ is the sample variance from the 30 replications. In the plots that follow we do not report these intervals. In a typical simulation with $\lambda = 0.9$ and no honeynode the estimated average wait was 9.849 ± 0.097, which corresponds to a relative error of 1%.

4.2 Comparison of approximations

Using parameters in Table 1, μ' can be calculated as in (3). When $\lambda \in [0.1, 0.9]$ and no honeynodes are assigned ($p_i = 0$) we get the range of values $\mu' \in [1.05513, 1.05551]$. For random honeynode assignment ($p_i = 1/N$), the corresponding range is $\mu' \in [1.00283, 1.00320]$, ensuring stability for all queues. The analytical formulas available hold for the infinite buffer model and are as follows.

M/G/1 Queue. To obtain an expression for the stationary average delay or waiting time in the queue, a first crude approximation is to use the $M/G/1$ formulas with the effective rates λ and μ' [17].

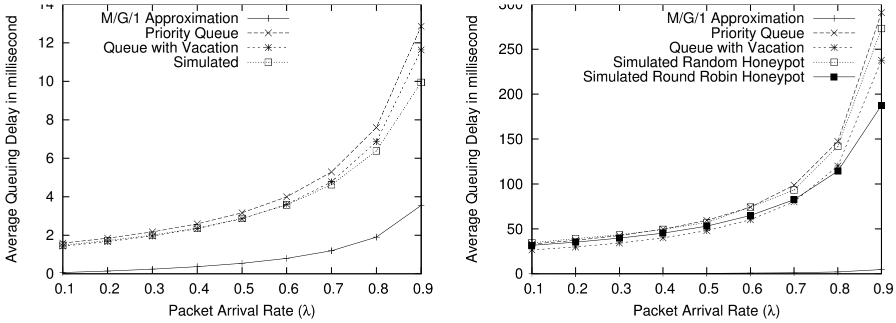
Priority model. A second approximation is based on a $M/G/1$ priority queue [17]. The sensing operation is to be served in higher priority, and packet transmission is the lower priority job. Formula (9) estimates the stationary average queuing delay for a packet with service priority i, when all customer classes arrive according to independent Poisson processes.

$$W_q^i = \frac{\lambda_1 \mathbb{E}[S_1^2] + \cdots + \lambda_n \mathbb{E}[S_n^2]}{2\prod_{j=i-1}^{i}(1 - \lambda_1 \mathbb{E}[S1] - \cdots - \lambda_j \mathbb{E}[Sj])} \tag{9}$$

This formula is only an approximation because the arrival rate of the "sensing" or high priority jobs is $\lambda_1 = (T_s + T_t)^{-1}$, and $S_1 = T_s$ is deterministic. Second high priority job is serving as honeynode where $\lambda_2 = p(T_s + T_t)^{-1}$ and $S_2 = T_t$, while $S_3 \sim U(0.1, 1.7)$ is the original packet service time distribution.

Vacation model. This corresponds to our formula (4). When no honeynodes are assigned, we use $p_i = 0$. For the random honeynode assignment we used $p_i = 1/20$.

Fig. 5a and Fig. 5b show the results. The discrepancies are not very visible for smaller values of λ but they become more apparent for heavier traffic regimes, where the vacation formula seems to agree best with the simulated system, under the round robin strategy.



(a) without Honeypot

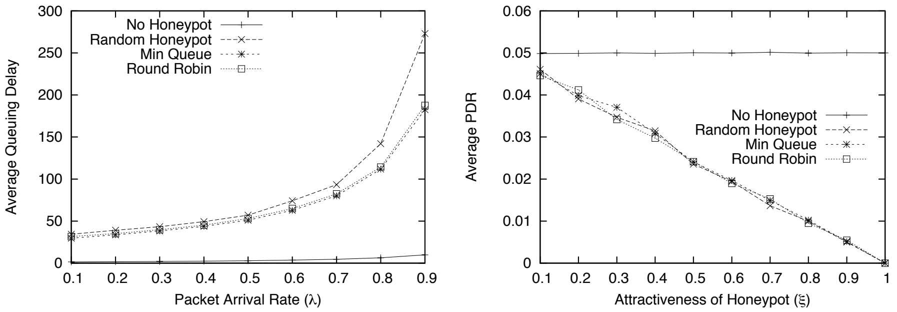
(b) With one Honeypot

Fig. 5: Average Queuing Delay for simple cognitive radio network

4.3 Comparison of honeynode assignment strategies

Fig. 6a shows the average wait per packet as λ increases with fixed $\xi = 0.8$ and infinite buffer sizes. In infinite buffer systems there is no packet drop for queue overflow, and therefore PDR is independent of λ. The only cause of packet drop is the jamming attack. Simulation results reflects that with $\xi = 0.8$, having no honeynode gives PDR of 0.05 and with one honeynode, PDR is 0.01 for all values of λ.

For the infinite buffer model, if θ_i is the probability that i'th SU is attacked and p_i is the long term fraction of periods where SU$_i$ is chosen as a honeypot (assuming stationarity), then PDR$_i = \theta_i((1 - p_i) + p_i(1 - \xi_i))$. When $\theta_i = p_i = 1/N$ and $N = 20$ we obtain the linear function $0.05(1 - 0.05\xi)$ as verified in Fig. 6b.



(a) Average Queuing Delay in millisecond

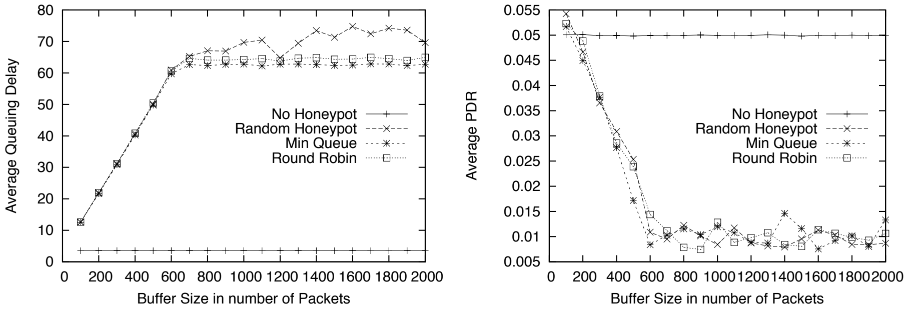
(b) Average PDR of all SUs

Fig. 6: Results for CRN with infinite buffer and $\lambda = 0.6$

The attractiveness (ξ) does not affect the queue size, which is only dependent on the strategy and the incoming rate λ. Simulation result shows average queuing delay for no honeypot, random honeypot, minimum queue and round-robin selection schemes are 3.54 ms, 72.55 ms, 62.1ms and 64.62 ms respectively for all values of ξ. These results clearly say that state dependent policy i.e. selecting honeynode based on minimum queue length is performing better compared with others strategy. Honeypot ensures less packet drop at the cost of increased queuing delay.

4.4 Finite Buffer Size

Fig. 7a and Fig. 7b show the results for the average wait and PDR respectively, as a function of the buffer size, when $\lambda = 0.6$ and $\xi = 0.8$ are fixed. On average $\lambda \times (2T_s + T_t) = 630$ packets would be queued when SU serves as honeynode. There would be queue overflow if buffer is smaller then CRN have significant PDR and for large buffer packet overflow happens and performance is acting as with infinite buffer.

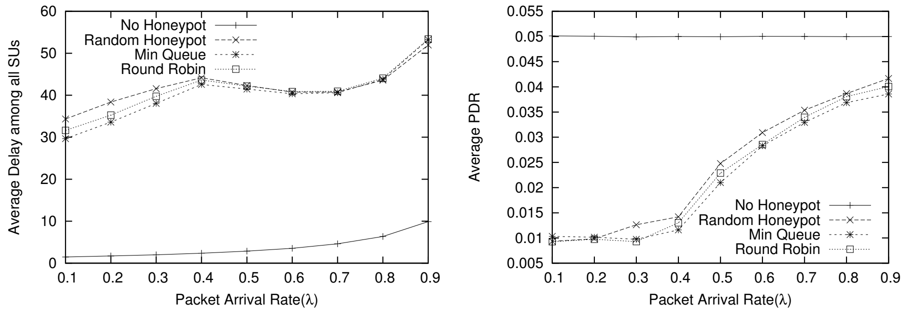


(a) Average Queuing Delay in msec

(b) Average PDR

Fig. 7: Results varying buffer size of SU with $\lambda = 0.6, \xi = 0.8$

To observe the effect of finite buffer, we run another set of simulation with buffer size as 400 packets for every SU. Fig. 8a and Fig. 8b shows the observed average queuing delay and PDR respectively. With low arrival rate λ (below a threshold) honeynode selection scheme based on minimum queue is performing better. This threshold value of λ should be Buffer Size$/(2T_S + T_t) = 0.381$ Because one SU can accumulate this amount of packets when serving as honeynode. When λ goes above the threshold, and when the SU is serving as honeynode, it accumulate many packets so that the queue overflows which cause significant amount of PDR. Below this threshold SU behaves as infinite buffer model. These two graphs shows a good trade-off between Delay and PDR. With limited buffer and from the two figures it is clear that the administrator have to come to a conclusion at particular value of λ and ξ and specified buffer size, whether to apply honeypot or not. At higher value of λ and loss tolerant traffic (such as real time video) not having honeypot as it can not tolerate delay.



(a) Average Queuing Delay in millisecond (b) Average PDR

Fig. 8: Results for CRN varying λ with $\xi = 0.8$ and Buffer of 400 packets

5 Conclusions and future work

In this paper we have presented a theoretical model to predict the performance of Cognitive Radio Network based on queuing model with fixed vacation. The model deals with the periodic sensing of cognitive cycle as fixed length vacation. Honeypot is well known to prevent jamming attack but assigning honeynode without considering queuing delay associated with it costs degradation of performance. Dynamic assignment of Honeynode is crucial from system performance perspective. We propose state dependent honeynode assignment scheme on every transmission cycle where the honeynode selection can be done by choosing the SU with lowest estimated queue size. This scheme performs well when all the SUs in network are having identical traffic load. In future research we will focus on the scenarios where SUs doesn't have identical load or have different QoS requirement.

References

1. S. Haykin, “Cognitive radio: brain-empowered wireless communications,” *IEEE Journal on Selected Areas in Communications,*, vol. 23, pp. 201 – 220, feb. 2005.
2. “IEEE draft standard for information technology -telecommunications and information exchange between systems - wireless regional area networks (WRAN) - specific requirements - part 22: Cognitive wireless ran MAC & PHY specifications: Policies and procedures for operation in the TV bands,” *IEEE P802.22/D2.0*, 2011.
3. Z. Liu, Y. Chen, W. Yu, and X. Fu, “Generic network forensic data acquisition from household and small business wireless routers,” in *IEEE International Symposium World of Wireless Mobile and Multimedia Networks (WoWMoM)*, pp. 1 –6, 2010.
4. S. Misra, S. K. Dhurandher, A. Rayankula, and D. Agrawal, “Using honeynodes for defense against jamming attacks in wireless infrastructure-based networks,” *Computers and Electrical Engineering*, vol. 36, no. 2, pp. 367 – 382, 2010.
5. S. Yek, “Implementing network defence using deception in a wireless honeypot,” in 2^{nd} *Australian Computer Network, Information and Forensics Conference*, 2004.
6. J. Burbank, “Security in cognitive radio networks: The required evolution in approaches to wireless network security,” in 3^{rd} *International Conference on Cognitive Radio Oriented Wireless Networks and Communications.*, pp. 1 –7, may 2008.
7. W. Xu, K. Ma, W. Trappe, and Y. Zhang, “Jamming sensor networks: attack and defense strategies,” *IEEE Network*, vol. 20, pp. 41 – 47, may-june 2006.
8. “Wi-spy spectrum analyzer.” http://www.metageek.net/products/wi-spy/.
9. S. Khattab, R. Melhem, D. Mosse, and T. Znati, “Honeypot back-propagation for mitigating spoofing distributed denial-of-service attacks,” in *20th International Parallel and Distributed Processing Symposium (IPDPS).*, april 2006.
10. L. Cai, X. Shen, J. Mark, L. Cai, and Y. Xiao, “Voice capacity analysis of wlan with unbalanced traffic,” *IEEE Transactions on Vehicular Technology*, may 2006.
11. L. Sun and E. Ifeachor, “Voice quality prediction models and their application in voip networks,” *IEEE Transactions on Multimedia*, vol. 8, pp. 809 –820, aug. 2006.
12. D. O. Vic Grout, Stuart Cunningham and R. Hebblewhite, “A note on the distribution of packet arrivals in high-speed data networks,” in *Proceedings of IADIS International Conference WWW/Internet*, 2004.
13. R. Jain and S. A. Routhier, “Packet trains-measurements and a new model for computer network traffic,” *IEEE Journal on Selected Areas in Communication*, vol. 6, pp. 986–995, September 1986.
14. M. Wilson, “A historical view of network traffic models.” http://www.cse.wustl.edu /j̃ain/cse567-06/ftp/traffic_models2. Accessed: 12/05/2012.
15. D. R. Dennis Guster and R. Sundheim, “Evaluating computer network packet interarrival distributions,” *Encyclopedia of Information Science & Technology*, 2009.
16. K. C. Madan, “An m/g/1 queue with optional deterministic server vacations,” *Metron - International Journal of Statistics*, vol. 0, no. 3-4, pp. 83–95, 1999.
17. S. M. Ross, *Introduction to Probability Models,* 10^{th}*Ed.* Academic Press, Dec. 2009.
18. S. M. Ross, *Simulation, 5th Edision.* Academic Press, October 2012.