CR-Honeynet: A learning & decoy based Sustenance Mechanism Against Jamming Attack in CRN

Suman Bhunia, Shamik Sengupta
University of Nevada, Reno, USA
sumanbhunia@gmail.com, ssengupta@unr.edu

Felisa Vázquez-Abad
Hunter College of the City University New York, USA
felisav@hunter.cuny.edu

***Abstract*—Cognitive Radio Network (CRN) enables secondary users to borrow unused spectrum from the proprietary users in a dynamic and opportunistic manner. However, dynamic and open access nature of available spectrum brings a serious challenge of sustenance amongst CRNs which makes them vulnerable to various spectrum etiquette attacks. Jamming-based denial of service (DoS) attack poses serious threats to legitimate communications and packet delivery. A rational attacker targets certain transmission characteristics to find the highest impacting communication of CRN and causes maximum disruption. In this paper, inspired by the honeypot concept in cybercrime, we propose a honeynet based defense mechanism, which aims to deter the attacker from jamming legitimate communications. The honeynet passively learns the attacker's strategy from the past history of attacks and actively adapts preemptive decoy mechanisms to prevent attacks on legitimate communications. Simulation results show that the with help of honeynet mechanism, CRN successfully avoids jamming attacks and thereby improves system performance in terms of packet delivery ratio.**

Keywords*—*Cognitive Radio, Jamming, Honeynet, Stochastic Learning

I. INTRODUCTION

Dynamic Spectrum Access (DSA) based Cognitive Radio (CR) [1] aims to provide a solution for spectrum scarcity by allowing Secondary Users (SUs) to use idle licensed spectrum on a non-interfering basis. In contrast to conventional wireless technologies, a CRN can reconfigure itself by controlling its operating frequency, channel bandwidth, modulation techniques, transmission power, etc. [1]. Because of the licensed primary user (PU) priority, SUs must periodically sense the channel of communication for the presence of the PU. If the current channel is blocked by presence of PU of that channel, SU must switch to another free channel. Even though the PU protection mechanisms have been proactively studied, neither the secondary-secondary interaction mechanisms nor the protection of secondary users from malicious disruption has been specifically defined or addressed [1], [2], [3].

The "open" philosophy of the cognitive radio paradigm makes CRN susceptible to Jamming based Denial of Service (DoS) attacks by smart malicious users. An attacker can scan through channels, identify legitimate SU communications and then transmit a jamming signal on the same channel or fragment of the channel causing disruptive interference to the SU, which in effect can completely block the legitimate SU's transmission [1], [2], [3]. However, note that, from an intelligent and rational attacker's perspective, jamming a communication randomly will not yield optimal result; rather an attacker can be most disruptive if it targets the communication that impacts the CRN most severely upon interruption [3], [4], [5], [6]. The attacker succeeds in determining highest impacting communication by observing certain transmission characteristics as for example, highest transmission power, highest data rate, modulation scheme, packet inter arrival time, quality of route with end-to-end acknowledgments, etc [5], [7]. Again, an attacker may also use combination of transmission characteristics instead of a single characteristics to find out the highest impacting communication. To defend against such attackers, a CRN must learn about the strategy (the targeted transmission characteristic(s)) that the attacker uses to figure out the highest impacting communication. The attacker's strategy of finding highest impacting communication can be used as a trap by the defending CRN to detract the attacker from attacking legitimate communications.

In this paper we propose *CR-Honeynet*, a honeynet based defense mechanism where the CRN passively learns the *strategy of the attacker* using stochastic learning, and then place an active decoy namely *honeynode* to entice the attacker for jamming the honeynode transmission. Thus, the attacker gets a false impression of attacking the highest impacting communication whereas legitimate SU communications avoid attacks and reduce attack impact on the CRN. One or multiple SUs act as honeynode in each transmission period. The SU acting as honeynode refrains from transmitting its own data packets and transmits garbage data with specific transmission characteristics. Such transmission characteristics lure the attacker to jam honeynode's transmission. The transmission characteristics that the attacker aims is learned from the history of attacks. As an example, if an attacker targets highest transmission power then the honeynode transmits with highest possible power while all other SUs keep their transmission power lower than honeynode's power.

The evolving nature of the attacker as well as dynamic and stochastic nature of the wireless medium pose several challenges to the learning mechanism. Suspicious of being trapped, an attacker may intentionally change its strategy of finding highest impacting communication. Also, due to erroneous and stochastic nature of wireless medium, an attacker may err in sensing CRN's highest impacting communication. Such error may results in attack on different SU communication instead of the communication with desired/targeted characteristics. Such circumstances must be taken into account for effective luring. In this paper, we use statistical monitoring threshold to decide whether the changes in recent attack pattern is due to error in attacker's sensing or whether the attacker has changed its attacking strategy. Our proposed stochastic learning mechanism correctly detects attacker's strategy with a probability of 0.958 within 15 iterations and identifies change in attacker's strategy dynamically with 95% confidence

This research was supported by NSF CAREER grant CNS #1346600

interval within 5 iterations. The simulation results show that CR-Honeynet learns attacker's strategy correctly and adapt with attacker's strategy change dynamically which in effect enhances CRN's performance in terms of packet delivery ratio.

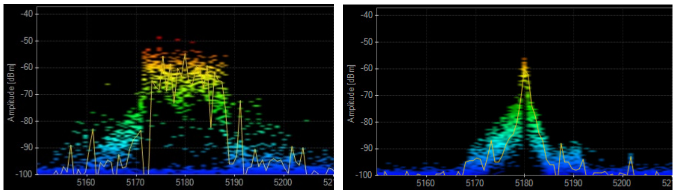
The rest of the paper proceeds as follows: in section II, we discuss the motivation for our work and background studies. section III presents our proposed model. In section IV we describe our simulator and then analyzes CR-Honeynet's performance. Finally section V concludes the paper.

II. Background Studies

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 they can fend off attackers. Thus, SUs are left vulnerable against malicious jamming attacks [1]. Jamming can be broadly categorized into two types [8], [9]. In *physical layer jamming*, the attacker jams the channel of communication by sending strong noise or jamming signals. The *data-link / MAC layer jamming* targets several vulnerabilities present in the MAC layer protocol. Jamming essentially means disrupting communication of legitimate users.

To illustrate the effect of jamming, we ran an experiment in our lab. Two computers were configured to communicate over a WLAN (IEEE 802.11-a) channel 36 (centered at 5,180 MHz). When communicating in full throttle it achieved end-to-end throughput of 11 Mbps. We observed the Power Spectral Density (PSD) over the channel using the Wi-spy spectrum analyzer [10]. The PSD for normal communication is shown in Fig. 1a. The plot clearly shows that the transmission is using a 20 MHz channel as well as some energy leakage to the neighboring channels. Then we started transmitting a very narrow band jamming signal of 2MHz from a GNU Radio [11] enabled USRP board [12]. At the presence of the jamming signal, the genuine transmission was blocked completely as can be viewed in Fig. 1b where only the jamming signal is visible. The attacker is exploiting the vulnerability of IEEE 802.11 MAC that enforces a node to sense 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. Irrespective of the jamming technique, a target node suffers significant amount of data or packet loss and sometimes completely loses the channel. CRN being a next-generation intelligent network should incorporate a mechanism to mitigate, avoid or prevent these attacks.

Due to the noise in wireless medium, detection of jamming is crucial in combating with an attacker. A good survey of different detection mechanisms for jamming based DoS attack has been presented in [5]. It is difficult to correctly detect jamming based on a single system parameter. Several system parameters such as received-signal-strength, packet-send-ratio, packet-delivery-ratio, carrier-sensing-time etc. are used for modeling jamming detection system. Consistency check among system parameters are used for more efficient detection. Authors of [13] have classified spectrum usage anomaly detection data fusion algorithms. Through different fusion algorithms, anomalies in spectrum usage can be detected successfully with higher efficiency. A cross layer detection mechanism of anomalous spectrum attack has been proposed in [14] where, the network maps the jammed geographical region using spectrum sensing reports sent from each SUs that are equipped with localization module.



(a) Normal communication

(b) Jamming Signal

Fig. 1: PSD for data communication and jamming signal

Already proposed defense mechanisms against jamming based DoS attack can be broadly categorized into *Channel Surfing*, *Spatial Retreat*, *Mapping Jammed Region*, *Spread Spectrum*, etc. In *Channel Surfing* technique, the node which is under attack migrates its channel of communication upon detection of jamming [9]. Authors of [15] proposed proactive frequency hopping where the nodes change its channel of operation irrespective of attacks to avoid jamming. The authors considered fixed number of channel of the attacker that is known to a SU, which in reality is difficult to achieve. In *Spatial Retreat* [16] mobile nodes relocate themselves physically to avoid jamming. The constraint of this approach is that the nodes are required to be highly mobile which is not realistic for static nodes. In *Mapping Jammed Region* [17] approach, the multi-hop and intensely populated CRN avoids routing through the links that have been affected by jamming. This mechanism fails if there is only one path and that link is attacked. In *Spread Spectrum* [18] technique, low bandwidth data stream uses higher bandwidth channel to pass the information irrespective of jamming. Although this mechanism increase reliability of communication it provide very poor data rate. A single channel honeypot based channel surfing to mitigate jamming-based DoS attacks has been proposed in [8]. The network dedicates a node as honeypot to monitor attacks and upon detection of attack, the network switches its channel of operation which results in long time communication disruption. Majority of the previous works have assumed that the attacker is naive and does not evolve. Thus, none of these works have focused on learning the strategy of attacker where the attacker is also dynamic and changes its strategy of choosing the target communication characteristics.

In our previous work [19], we introduced the concept of honeynet in CRN and presented the benefits of dedicating one SU as honeynode in a multichannel CRN provided the honeynode is successful of enticing the attacker. We presented a stochastic model for honeynode selection which proved that in the case of uniform traffic, selecting a SU with lowest queue size is optimal in terms of overall system performance. Extending on previous work, in this paper we present CR-Honeynet, where the CRN learns the strategy of the attacker and then dedicates one SU as honeynode. Honeynode acts as the optimal target for the attacker so that the attacker gets false satisfaction of attacking highest impacting SU communication, thus reducing attack impact on other legitimate communications.

III. Proposed Model

A. Model for attackers

In this paper we consider three types of attacking strategies, as follows:

I: Attacker targeting a particular channel.

II: Targets specific SU transmission characteristic(s).

III: Randomly targeting channel with active SU transmission.

Attacking strategy of type I and III causes less harm on a CRN as it does not search for the best communication that causes the highest impact in CRN. However, an intelligent and rational attacker of type II can choose any transmission characteristics to determine the best communication for attack that causes highest impact on the CRN. From the CRN's point of view, it is difficult to generate such characteristic space. Such targeted characteristic space of an attacker can be learned by two methods: *manually by domain experts* or through *automatic learning* from data obtained for long time. For the first step, we are dealing with the first method and wish to extend our model to perform the second option and learn an attacker's possible strategical view points by automatic learning. We present a generalized model considering the d possible transmission characteristics or a combination of transmission characteristics.

B. Honeynet Defense mechanism

To protect PU transmissions, SUs are required to perform periodic spectrum sensing and evacuate promptly upon the return of the PU. SUs scan the wireless environment for free channels in the *sensing period* (T_s). During *transmission period* (T_t), a SU transmits packets through its own channel dedicated by a centralized controller. We assume that a SU cannot switch its channel during the transmission period as it is unaware of the condition of the other channels and can switch only on the next transmission period. Upon being attacked, all data packets transmitted by the SU are lost. In CR-Honeynet, the central controller assigns the role of honeynode to a SU at the beginning of each transmission period. Fig. 2 illustrates this channel allocation based on time domain (Sensing period not shown). Due to the error of the attacker or strategy change of the attacker, some attacks are trapped by honeynode transmission, and others disrupt legitimate SU communications. We define a parameter, *attractiveness of honeynode* (ξ) as the probability that the honeynode transmission is attacked, conditional on observing a jamming attack.

When acting as a honeynode, a SU doesn't transmit its packets, instead, it queues all its incoming packets and transmits garbage data packets. Honeynode allocation results in more delay as well as packet drop due to finite buffer sizes for the chosen SU, both of which are undesirable. If the attractiveness of the honeynet (ξ) is low, then the CRN will suffer the delay caused by honeynode allocation as well as the packet drop with a probability of $(1-\xi)$ due to the attacks on legitimate SUs other than the one chosen as honeynode. The threshold, lowest attractiveness of the honeynet (ξ^*) is the value where the net gain is zero; below ξ^* the CRN is better off facing the loss from the attacks than dedicating one SU intending to lure the attacker.

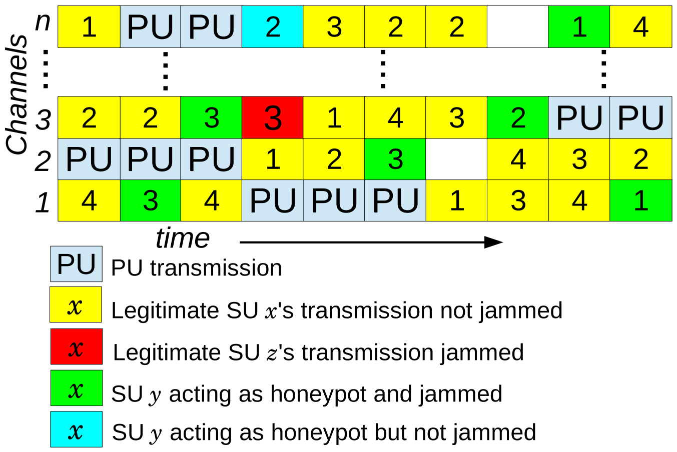

Channels
n 1 PU PU 2 3 2 2 1 4
3 2 2 3 3 1 4 3 2 PU PU
2 PU PU PU 1 2 3 4 3 2
1 4 3 4 PU PU PU 1 3 4 1
time
PU PU transmission
x Legitimate SU x's transmission not jammed
x Legitimate SU z's transmission jammed
x SU y acting as honeypot and jammed
x SU y acting as honeypot but not jammed

Fig. 2: A snapshot of CR-Honeynet channel allocation

In accordance to an attacker's strategies, we define the following honeynode strategies:

- If the attack strategy is believed to be of type-I then the vulnerable channel will be assigned to the honeynode.
- If the attack strategy is believed to be of type-II then the actual target property should be learned and used as a lure for the honeynode.
- If the attack strategy is believed to be of type-III then we use a *special honeynode strategy* that delays all but the honeynode's transmissions in order to reduce the number of vulnerable channels to 1.

If there are $\mathcal{C}$ available channels, then an attacker must sense for activity on each of them. Let's assume switching a channel incurs a delay of κ ($\kappa = 7.6ms$ has been measured for Atheros WiFi [15]). The attacker needs at most $\mathcal{C}\kappa$ time units to scan all available channels for activity. Under the special strategy we must therefore delay all SUs at least $\mathcal{C}\kappa$ units of time beyond the sensing period, during which only the honeynode will transmit. When using the special strategy there is an added loss or cost of luring as all the other SUs are delayed in their transmissions, albeit much less than the delay caused to the chosen honeynode. So this strategy should be avoided by CR-Honeynet if possible. In contrast, type-III is the only strategy that increases attractiveness of honeynet (ξ) to 1.

C. Stochastic Model

We assume that at time slot n the attacker's strategy S_n follows a random switching process, with consecutive switching times $T_k \in \mathbb{N}$. The model need not be a *Hidden Markov Model*, but we assume that the holding times $h_k = T_{k+1} - T_k$ are long enough for learning. We will specify the exact assumption later on.

The *base model* for an attacker with a type-II strategy is stated now. Because of measurement errors, the attacker may not always be successful in identifying the correct communication to attack. Let p_1 denote the probability of attacking the communication with the target characteristics. We will assume that the number of available channels is larger than d, and use d of the SUs as learning probes, each with a different target property. Counting only the time slots when one of the probes is attacked, the total number of attacks to each of the probes within n such time slots is modeled as a multinomial random variable with probabilities:

$$p_1(\theta) = \frac{\theta}{\theta + d - 1}; \quad p_k(\theta) = \frac{1}{\theta + d - 1}, \text{ for } i \in \{2, 3, \dots, d\}, \tag{1}$$

where $\theta > 1$.

The above model corresponds to the situation where probe $k = 1$ is targeted and hit with probability $p_1 < 1$. Under error measurement, any other probe will be attacked with equal probability $p_i, i \neq 1$. The number $\theta = p_1/p_i$ provides the ratio between p_1 and the rest. For the base model, using the fact that all other probabilities are equal, $\theta = (d-1)p_1/(1-p_1)$.

Define the function:

$$\phi(\theta, n) = \sum_{y \in \mathcal{P}(n)} \frac{n!}{y_1!\, y_2!, \dots y_d!} p_1(\theta)^{y_1} \left(\frac{p_1(\theta)}{\theta} \right)^{\sum_{i=2}^{d} y_i} \tag{2}$$

where the summation is over the set of all possible observations of a sample of size n of the multinomial with parameters (eq. 1) where the first component dominates the others, that is:

$$\mathcal{P}(n) = \left\{ y \in \mathbb{N}^d : \sum_{k=1}^{d} y_i = n, \text{ and } y_1 \geq y_i; i \geq 2 \right\}.$$

It is straightforward to show that this is the exact probability of correct selection in a sample of size n from the base model when the maximum likelihood estimator is used. Specifically, let $Y_i(n)$ count the number of attacks to probe i under the base model, so that: $(Y_1(n), Y_2(n), \dots Y_3(n)) \sim M(p(\theta), n)$, then the MLE for the parameter p_i is simply $\hat{p}_i(n) = Y_i(n)/n$ and $\phi(\theta, n) = \mathbb{P}(Y_1(n) = \max(Y_1(n), \dots, Y_d(n))$.

Let $\alpha \in (0, 1)$ be a confidence level for statistical significance. Then under the base model we can calculate the sample size required to ensure a probability of correct selection of at least $1 - \alpha$:

$$N^*(\theta, d) = \min\left(n : \phi(\theta, n) \geq 1 - \alpha\right). \tag{3}$$

Bechhofer *et al.*[20] have tabulated the function $\phi(\theta, n)$ for $d = 2, 3, 4$ using various values of θ and n. For example, if $d = 4$, then a sample size of $n = 25$ ensures a correct selection with level $\alpha = 0.200579$ when $\theta = 2$, and with level $\alpha = 0.038559$ when $\theta = 3$.

Suppose that honeynet correctly identifies a lure, but $p_1 < \xi^*$. Clearly, the best it can do here is to use its (correct) guess for the honeynode, but this will provide at most a probability p_1 that the honeynode will be attacked. Because p_1 is below the threshold, it will not be worth using honeynode in this case and we use special honeynet strategy similar to type-III. Thus, such values of ξ^* provide a threshold value $\theta^* = \frac{(d-1)\,\xi^*}{(1-\xi^*)}$ below which it is not worth using honeynode.

Definition: We call a *naive* attacker one of type II where the probability of error in measurement is lower than $1 - \xi^*$, and we assume that $\mathbb{P}(h_k < N^*(\theta^*, d)) = 0$.

The above definition says that this type of attack is fairly accurate (usually $p_1 \gtrapprox .85$) and also that the strategy is kept long enough to learn the target probe. Specifically, because $\theta \geq \theta^*$ for a naive attacker, then $N^*(\theta, d) \leq N^*(\theta^*, d)$ if we use the MLE to identify the target with $\arg\max(Y_i(n))$ for $n \approx N^*(\theta, d)$.

D. Learning Attacker's strategy

When the learning mechanism starts, given a confidence level α, the number $n = N^*(\theta^*, d)$ is calculated as a first estimate for an adequate sample size to detect type II attackers. When $d < \mathcal{C}$ it is possible that error in measurements results in false attacks to communications that have not been allocated any lures. Thus, we will focus only on time slots when attacks happen to lures. According to our model, this "sampled" process corresponds to the base model for attackers of type-II. Given n, define $\tau(n)$ as the total number of time slots required to see n attacks to the lures.

During the learning phase, the d different lures for type-II attacks are assigned to d different communications among the available ones with uniform probability and no honeynode is yet allocated. Let $Y_i(0) = 0; i = 1, \dots, d$ and define for each $i = 1, \dots, d$ and the counting processes:

$$Y_i(k) = Y_i(k-1) + \mathbf{1}_{\{i\text{-th lure is attacked at time } k\}}$$

for $k = 1, 2 \dots$, where the notation $\mathbf{1}_{\{A\}}$ stands for the indicator function of event A (or Dirac delta). In parallel, define $C(1) = c$, if c is the first channel to suffer an attack, and let

$$C(k+1) = C(k)\mathbf{1}_{\{\text{channel } c \text{ is attacked at time } k+1\}}.$$

Because we have allocated the lures randomly among SU communications, it follows that

$$\mathbb{P}(C(k) = 1 \,|\, \text{type II or III}) \leq \max\left\{ \left(\frac{1}{d}\right)^k, \left(\frac{1}{\mathcal{C}}\right)^k \right\}$$

Define n_0 as the smallest power that makes this probability smaller than our given confidence level α, that is, when $d < \mathcal{C}$

$$n_0 = \lceil \log(1/\alpha) - \log(d) \rceil.$$

The number of tests to check for type I is thus typically very small. For example, if $\alpha = 0.001, d = 2$ then $n_0 = 7$, for $\alpha = 0.005$ and $d = 6$, $n_0 = 4$.

If $C(n_0) = c$, we declare having learned that the attack is of type I and we identify c as the target channel. From this point onwards, we place the honeynet in this channel and keep monitoring. Because attacks of type I are not subject to error in identifying the channel, as soon as $C(k) = 0$ we declare a regime change and re-set the learning phase.

Otherwise, if $C(n_0) = 0$ then we keep assigning lures to channels for as many time slots are required to observe

$n = N^*(\theta^*, d)$ attacks to lures. Gelfand *et al.*[21] provides a comparison between various estimators and confidence intervals for $\hat{p}_1$. In particular, his findings support the fact that under attacks of type II the approximate confidence interval based on the CLT is adequate, even for small to moderate sample sizes. Following this approximation, if

$$\hat{p}_1 - 1.96\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n}} \geq \xi^* \tag{4}$$

then we declare having learned that the strategy is of type II and we identify the lure. From this time onwards, we use the honeynode with that lure and start the monitoring phase. Notice that by construction, naive attacks are ensured to be correctly identified with probability at least $1 - \alpha$.

If (eq. 4) does not hold, then we do not have significant evidence that our candidate lure will be sufficiently effective. From this point onwards (whether the attacker is of type II but with large measurement errors, or of type III) we use the special honeynode allocation by delaying all other SUs. It is important to note that while the regular honeynode entails a delay for the chosen SU, the special strategy delays all of the rest of the SUs, albeit by a much smaller amount of time.

E. Regime change detection: monitoring phase

Once the learning period is over, the corresponding honeynet strategy is used and honeynet keeps monitoring the attack counts, keeping track of running window averages. This is the monitoring phase where the honeynet is sensing for a possible change in attacker's strategy, as follows.

If the honeynet is under the assumption of a type I attack, then it keeps track of $C(k), k \geq n_0$ until the first time slot where $C(k) = 0$. Then it restarts the learning phase.

If the honeynet is under the assumption of a type II naive attacker then it uses sliding window averages to test for regime changes. During the monitoring phase the honeynet uses the detected lure for the honeynode allocation. Honeynet's first monitoring test uses a standard control chart for frequencies, and the second proposed method uses a regression for the slope of the frequency of attacks. Let $\hat{p}_1(k); k \geq n$ be as in (eq. 5) re-calculated with increasing observations beyond the initial horizon n and call

$$L(k) = \hat{p}_1(k) - 3\sqrt{\frac{\hat{p}_1(k)(1-\hat{p}_1(k))}{w}}.$$

Given a window of size w time slots, let $\tilde{\xi}_w(k)$ be the estimate of p_1 (and also of ξ) for time slot $k > n$ using the observations $(Y_{(d)}(k-w), \ldots, Y_{(d)}(k))$. As soon as $\tilde{\xi}_w(k) < L(k)$, the honeynet declares a change of regime and restarts the learning phase (resetting all counters).

The regression test works very similarly. (To complete, regression with the window and test for $H_0 : \beta < 0$, where β is the slope or method of residuals).

Finally, if the honeynet is operating under the assumption of a type III attack or a type II attack for small ξ, then honeynode's current strategy is the special honeynet strategy that delays all but the honeynode. The honeynet keeps a new counter $H(k) = H(k-1) \times \mathbf{1}_{\{\text{honeynode is attacked}\}}$, initialized at the value 1. As soon as the attack goes to another channel ($H(k) = 0$), the honeynet declares that the attacker is not aiming at random, but it must be targeting now either a specific channel or a specific property of the transmissions. Then the honeynet restarts the learning phase.

IV. Simulation and Results

A. Simulator

We coded a *tick based simulator* [22] using *Python* for simulating the CR-Honeynet. In the simulation we have considered 20 SUs and 1 attacker which can effectively attack one SU communication. The CR-Honeynet dedicates 1 SU as honeynode in each slot. The attacker follows algorithm 1 and the honeynet follows algorithm 2. All SUs generate packets in accordance with *Poisson* process and queue them while in sensing period or when that SU is acting as a honeynode. During transmission period, SUs that are not acting as honeynode transmit packets from the queue. Packet transmission time (S_n) follows uniform distribution of 0.1 - 1.7 ms. A sensing Period (T_s) of 50 ms and a transmission Period (T_t) of 950 ms has been considered for cognitive cycle. We consider attacker has target transmission characteristics (d) space as 4. From the CRN's point of view, attractiveness threshold (ξ^*) is considered as 0.6. Type-I learning horizon (n_0) and Type II learning horizon ($N^*(\theta^*, d)$) are calculated as 5 and 15 respectively. We have run the simulation for 5,000,000 ms *simulation time* with 100,000 ms as *warm-up time* [1].

Algorithm 1: Algorithm for attacker

```
if strategy = attack particular channel then
    scan channel $c \in \mathcal{C}$ in the initial stage of $T_t$
    if SU is active on c then
        attack on channel $c$
    end
else if strategy = attack transmission characteristics $x$
then
    Scan all $c_i \in \mathcal{C}$ at initial stage of $T_t$
    attack the channel which have highest $x$
else if strategy = attack randomly then
    Scan all $c_i \in \mathcal{C}$ at initial stage of $T_t$
    attack randomly a channel $c$ where SU is active
end
```

B. Learning attacker's strategy

We plot $\phi(\theta^*, n)$ (eq. 2) with respect to learning period (N^*) in Fig. 3. We can clearly see that with an increase in N, confidence level also increases. We have defined earlier, confidence level for statistical significance ($\alpha = 1 - \phi$). From (eq. 3) we can get the optimal N^*. For our simulation we have considered $\xi^* = 0.6$. We see that $N^* = 15$ ensures correct learning with level $\alpha = 0.015$ for $d = 4$. From the figure we can conclude that for a certain desired confidence of learning ($\phi(\theta^*, n)$), an increase in the number of lured characteristics (d), results in a decrease in required slots for learning (N^*). In another way, the more transmission characteristics or combination of transmission characteristics

[1] To obtain reliable steady state results for system staring with empty queue, a simulator run for *warm-up period* [22] without recording data. Once the warm period is over simulator starts gathering data

```
Algorithm 2: Algorithm for Honeynet
1  Calculate $n_0, N^*$ based upon $d$ and $\xi^*$
2  Reset all counters such as $y, n$ etc.
3  Run initial learning phase for $n_0$ slots
4  if all attack happens on channel $c \in \mathcal{C}$ then
5      Put honeynet on $c$ in every slot until attack
       observed on other channel.
6      Go to step 2
7  else
8      Continue counting for $n = N^*(\theta^*, d)$ slots
9      if $\hat{p}_1 - 1.96\sqrt{\hat{p}_1(1-\hat{p}_1)/n} \geq \xi^*$ then
10         $lure = \mathrm{argmax}(y)$
11         put honeynode with $lure$ on every slot until
           $\xi_w(k) < L(k)$
12         Go to step 2
13     else
14         Use special honeynet strategy until honeynode
           is not attacked. Go to step 2.
15     end
16 end
```

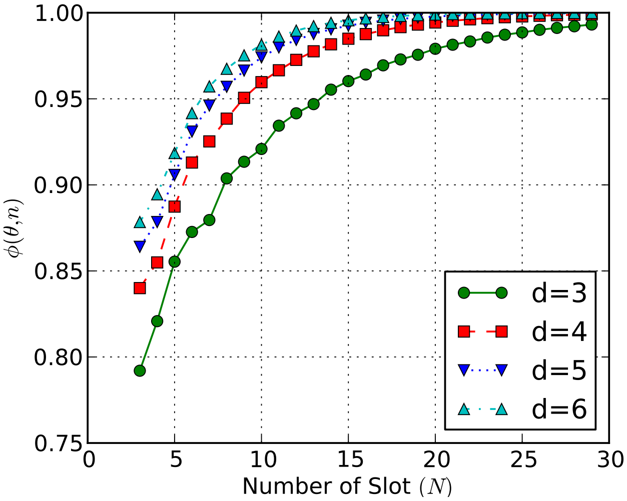


Fig. 3: Confidence level of Learning

an attacker can target, CR-honeynet takes lesser time to learn with the same confidence.

Fig. 4 provides an illustration of a learning phase. In this scenario, the attacker with type II strategy is aiming for lure 2 (the lure is characteristics of transmission) to attack. Lure 2 is actually attacked with a probability 0.8. The actual attacks on the various lures are shown on the upper subplot. The middle subplot depicts $\hat{P}$ for the different lures calculated using (eq. 5). The third subplot provides $\hat{p}_1 - 1.96\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n}}$ which can be taken as a measure of learning. With $d = 4$ and $\xi^* = 0.6$, the probability of correct selection after $N^* = 15$ samples is 0.985.

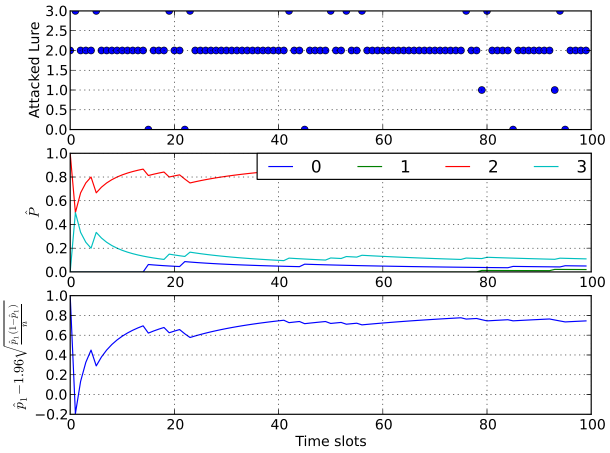


Fig. 4: Depiction of Learning while $d = 4$

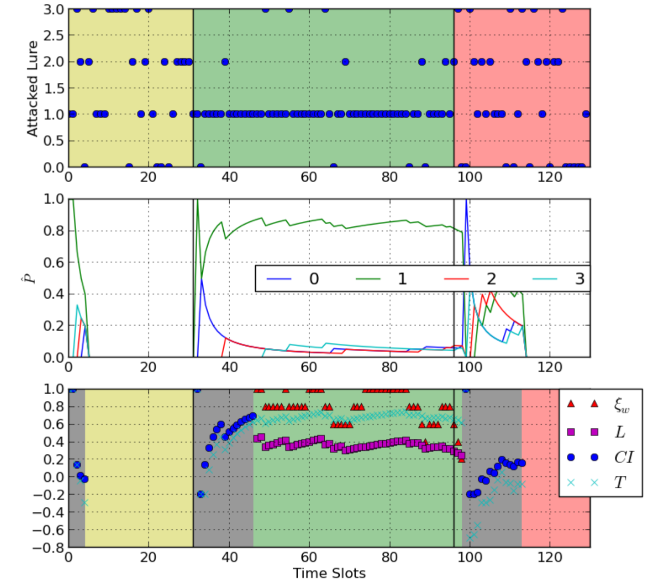


Fig. 5: Honeynet Learning phase corresponding to attacker's strategy change from I to II and then to III

C. Dynamic evolution with change in attacker's strategy

Fig. 5 and 6 provide the results for different experiments, each of which corresponds to a different sequence of attack processes $\{S_n; n = 1, 2, \ldots\}$. The upper subplots provide the attacker's strategy. *Yellow, green* and *red* colors indicate type-I, type-II and type-III attacking strategies respectively. Blue dots indicates the attacker's aimed transmission characteristics to find highest impacting communication. Here we have used 4 types of lure, i.e. transmission characteristics ($d = 4$).

Middle subplots give CR-Honeynet's observation of $\hat{p}$ (eq. 5) for different lures. It uses the MLE estimators for the two highest probabilities:

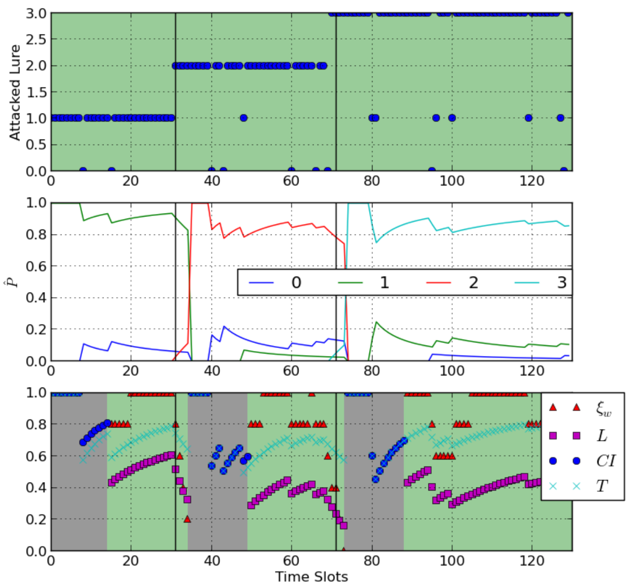


Fig. 6: Honeynet Learning phase corresponding to attacker of type II and attacker is changing its target lure

$$\hat{p}_1 = \frac{Y_{(d)}(\tau(n))}{n}; \quad \hat{p}_2 = \frac{Y_{(d-1)}(\tau(n))}{n}, \tag{5}$$

where the notation $(x_{(1)}, \ldots, x_{(d)})$ is the usual notation for the ordered statistics.

In lower subplots of these 3 figures, we present phases of Honeynet. Background colors *Grey, yellow, green* and *red* indicate the learning phase, type-I, type-II and type-III defense strategies respectively. Then we plot the estimation of $CI = \hat{p}_1 - 1.96\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n}}$ in the learning phase. We can see that with increase in slots, CI is increasing. When the learning phase is over and honeynet decides which strategy to take, it change its phase. When it detects a regime change, it enters to the learning phase again. When it is in type-II honeynet strategy, the honeynet monitors $L(k)$ and $\tilde{\xi}_w$. Honeynet enters learning phase when $\tilde{\xi}_w \leq L(k)$. An approximate test of level 0.05 which decides whether the attack is of type II or III is to test if $T > 0$, for the statistics:

$$T = (\hat{p}_1 - \hat{p}_2) - 1.96\sqrt{\frac{\hat{p}_1(1-\hat{p}_1) + \hat{p}_2(1-\hat{p}_2) - \hat{p}_1\hat{p}_2}{n}}.$$

If $T \leq 0$ then we infer, the attacks are "sufficiently random" between at least two main contenders.

Fig. 5 shows how the honeynet learns the change of strategy of attacker dynamically. We can see that, for type-I attack, honeynet learns in 5 iterations. To distinguish between type II and III, honeynet takes 15 slots. When the attacker deviates from type I, honeynet learns it on the next iteration. However, when the attacker is in type II and changes its strategy, honeynet takes 2 iterations to detect the change in strategy of attack.

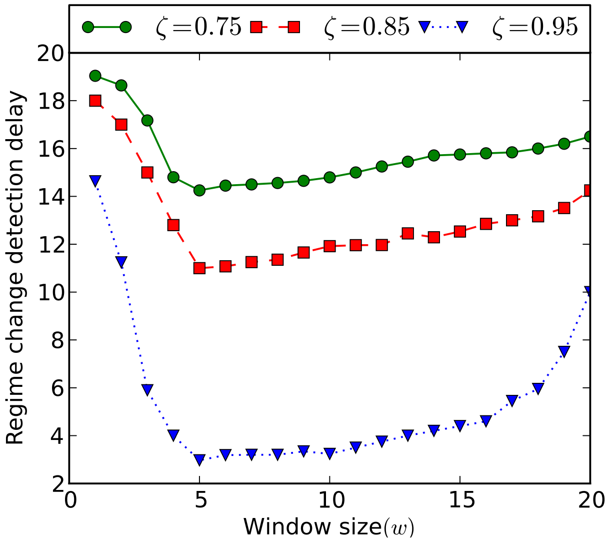


Fig. 7: Regime change detection delay for type II attacker

Fig. 6 depicts a scenario where the attack strategy is of type II. It changes its targeted SU transmission characteristics dynamically. For the first phase, attacker aims characteristics 1 and then 2 and then 3. We can see that for imperfect scanning, the attack may actually happen on a different lure. Honeynet identifies the correct strategy and particular type of attack in 15 iterations. We can see that for this particular simulation, honeynet detect attack strategy change after 3 iteration in the first case and after 2 iterations in the second one.

When honeynode is placed with the wrong lure, legitimate communications are disrupted. We see honeynet detects the regime change very quickly, which decrease the loss. Mainly, the loss is during the learning phase when CR-Honeynet does not deploy honeynode. To see how long does it take for the honeynet to detect the regime change while in type II lure strategy, we present a comparison to select optimal window size in fig. 7. An attacker of type-II strategy is attacking a particular lure with probability ζ. We simulated for 3 different values of ζ. For every value of ζ and w, the simulation is run for 100,000 slots to ensure accurate results. In this simulation, the attacker is changing its targeted transmission characteristics randomly with mean interval of 100 steps. We can clearly see that, window size $w = 5$ provide optimal result i.e. it can detect regime change very quickly and efficiently.

D. *Overall system performance*

We now code an *Event Driven Simulator* to compare the system performance between using honeynet and not using honeynet for an infinite buffer CRN. For simplicity, we have considered 20 SUs and kept $\xi = 0.8$. We vary average packet inter-arrival time (λ) to examine system performance with varying load. We observe that for all values of λ, with CR-Honeynet the average packet dropping probability is 0.01, while without honeynet results packet dropping probability of 0.05 . Fig. 8 provides the comparison of average queuing delay for a SU. From the figure, we can conclude that, using honeynet for lower λ is highly beneficial as packet drop is minimized. Better packet delivery ratio is achieved at the cost

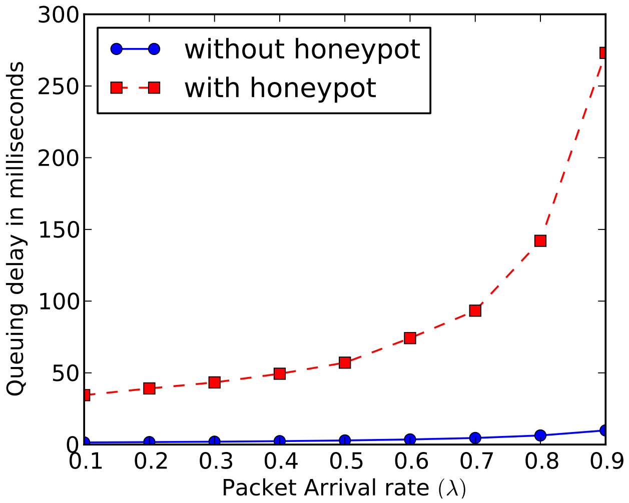


Fig. 8: Average queuing delay for $\mathcal{N} = 20, \xi = 0.8$

of higher packet delay. In our future work we shall try to get an estimation of ξ^* that can regulate CRN to use honeynode or not, depending λ and traffic type (elastic, non-elastic, real-time etc.)

V. Conclusions and Future Work

In this paper we propose CR-Honeynet, a CRN sustenance mechanism, which exploits the fact that an intelligent and rational attacker aims for certain transmission characteristics to gain highest impact out of jamming. The stochastic learning model presented in the paper shows that the honeynet can confidently learn the attacker's strategy and dynamically evolve with attacker's strategy change. The mechanism efficiently lures the attacker towards attacking the active decoy trap and thus bypassing attacks on legitimate SU communications. Currently, the mechanism has a drawback of not placing active decoy while it is passively learning attacker's strategy. In the future, we shall investigate more to improve the learning mechanism where the honeynet would be able to predict the attacker's strategy change and can place an active decoy to mitigate attack. Again this model can be further enhanced while considering the combination of transmission characteristics as attacker's strategy to find the highest impact communication.

References

[1] A. Fragkiadakis, E. Tragos, and I. Askoxylakis, "A survey on security threats and detection techniques in cognitive radio networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 428–445, 2013.

[2] S. Bhattacharjee, S. Sengupta, and M. Chatterjee, "Vulnerabilities in cognitive radio networks: A survey," *Computer Communications*, vol. 36, no. 13, pp. 1387–1398, 2013.

[3] T. X. Brown and A. Sethi, "Potential cognitive radio denial-of-service vulnerabilities and protection countermeasures: A multi-dimensional analysis and assessment," *Mobile Networks and Applications*, vol. 13, no. 5, pp. 516–532, 2008.

[4] S. Anand, S. Sengupta, K. Hong, K. Subbalakshmi, R. Chandramouli, and H. Cam, "Exploiting channel fragmentation and aggregation/ bonding to create security vulnerabilities," *IEEE Transactions on Vehicular Technology*, 2014.

[5] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 2, pp. 245–257, 2011.

[6] B. Wang, Y. Wu, K. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 4, pp. 877–889, 2011.

[7] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, no. 2, pp. 293–315, 2003.

[8] S. Misra, S. K. Dhurandher, A. Rayankula, and D. Agrawal, "Using honeynodes for defense against jamming attacks in wireless infrastructure-based networks," *Computers & electrical engineering*, vol. 36, no. 2, pp. 367–382, 2010.

[9] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing and spatial retreats: defenses against wireless denial of service," in *Proceedings of the 3rd ACM workshop on Wireless security*, pp. 80–89, ACM, 2004.

[10] "Wi-spy spectrum analyzer." http://www.metageek.net/products/wi-spy/.

[11] "GNU Radio." http://gnuradio.org/redmine/projects/gnuradio/wiki.

[12] "Usrp kit." https://www.ettus.com/product/details/UN200-KIT.

[13] V. Chatzigiannakis, G. Androulidakis, K. Pelechrinis, S. Papavassiliou, and V. Maglaris, "Data fusion algorithms for network anomaly detection: classification and evaluation," in *Networking and Services, 2007. ICNS. Third International Conference on*, pp. 50–50, IEEE, 2007.

[14] C. Sorrells, L. Qian, and H. Li, "Quickest detection of denial-of-service attacks in cognitive wireless networks," in *Homeland Security (HST), 2012 IEEE Conference on Technologies for*, pp. 580–584, IEEE, 2012.

[15] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, "Using channel hopping to increase 802.11 resilience to jamming attacks," in *26th IEEE International Conference on Computer Communications (INFOCOM)*, pp. 2526–2530, IEEE, 2007.

[16] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *Network, IEEE*, no. 3, pp. 41–47, 2006.

[17] C. Sorrells, P. Potier, L. Qian, and X. Li, "Anomalous spectrum usage attack detection in cognitive radio wireless networks," in *IEEE International Conference on Technologies for Homeland Security (HST), 2011*, pp. 384–389, IEEE, 2011.

[18] C. Popper, M. Strasser, and S. Capkun, "Anti-jamming broadcast communication using uncoordinated spread spectrum techniques," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 5, pp. 703–715, 2010.

[19] S. Bhunia, X. Su, S. Sengupta, and F. Vázquez-Abad, "Stochastic model for cognitive radio networks under jamming attacks and honeypot-based prevention," in *International Conference on Distributed Computing and Networks (ICDCN)*, pp. 438–452, Springer Berlin Heidelberg, 2014.

[20] R. E. Bechhofer, S. Elmaghraby, and N. Morse, "A single-sample multiple-decision procedure for selecting the multinomial event which has the highest probability," *The Annals of Mathematical Statistics*, pp. 102–119, 1959.

[21] A. Gelfand, J. Glaz, L. Kuo, and T.-M. Lee, "Inference for the maximum cell probability under multinomial sampling," *Naval Research Logistics (NRL)*, vol. 39, no. 1, pp. 97–114, 1992.

[22] S. Ross, *Simulation*. Elsevier Science, 2012.