

SpecWatch: Adversarial Spectrum Usage Monitoring in CRNs with Unknown Statistics

Ming Li*

Dejun Yang*

Jian Lin*

Ming Li†

Jian Tang‡

*Department of Electrical Engineering and Computer Science, Colorado School of Mines, Golden, Colorado 80401, USA

†Department of Electrical and Computer Engineering, University of Arizona, Tucson, Arizona 85721, USA

‡Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, New York 13244, USA
{mili, djyang, jilin}@mines.edu, lim@email.arizona.edu, jtang02@syr.edu

Abstract—In cognitive radio networks (CRNs), dynamic spectrum access has been proposed to improve the spectrum utilization, but it also generates spectrum misuse problems. One common solution to these problems is to deploy monitors to detect misbehaviors on certain channel. However, in multi-channel CRNs, it is very costly to deploy monitors on every channel. With a limited number of monitors, we have to decide which channels to monitor. In addition, we need to determine how long to monitor each channel and in which order to monitor, because switching channels incurs costs. Moreover, the information about the misuse behavior is not available a priori. To answer those questions, we model the spectrum usage monitoring problem as an adversarial multi-armed bandit problem with switching costs and design two effective online algorithms, SpecWatch and SpecWatch⁺. In SpecWatch, we select strategies based on the monitoring history and repeat the same strategy for certain timeslots to reduce switching costs. We prove its expected weak regret, i.e., the performance difference between the solution of SpecWatch and optimal (fixed) solution, is $O(T^{2/3})$, where T is the time horizon. Whereas, in SpecWatch⁺, we select strategies more strategically to improve the performance. We show its actual weak regret is $O(T^{2/3})$ with probability $1-\delta$, for any $\delta \in (0, 1)$. Both algorithms are evaluated through extensive simulations.

I. INTRODUCTION

With the proliferation of wireless devices and applications, demand for access to spectrum has been growing dramatically and is likely to continue to grow in the foreseeable future [1]. However, there is a paradoxical phenomenon that usable radio frequencies are exhausted while much of the licensed spectrum lies idle at any given time and location [2]. To improve the radio spectrum utilization efficiency, dynamic spectrum access (DSA) in cognitive radio networks (CRNs) has been proposed as a promising approach. Among various DSA strategies, opportunistic spectrum access (OSA) based on the hierarchical access model has received much attention recently [3–6]. This underlay approach achieves spectrum sharing by allowing secondary (unlicensed) users (SUs) to dynamically search and access the spectrum vacancy while limiting the interference perceived by primary (licensed) users (PUs) [7].

OSA helps to improve the spectrum utilization but also results in spectrum misuse or abuse problems due to the

flexibility of spectrum opportunity. For example, an SU may intentionally disobey the interference constraints set by the PU; or some greedy SUs may transmit more aggressively in time and frequency to dominate the spectrum sharing. Through such spectrum access misbehavior, the malicious users (MUs), i.e., the misbehaving SUs, not only harm the spectrum access operations of normal users, but also impede the CRNs to function correctly since there is no incentive to pay for spectrum access [8]. Thus, spectrum usage monitoring is necessary and imperative.

To address the spectrum misuse problem, different trusted infrastructures have been proposed to detect spectrum misuse and punish MUs [8–10]. In addition, various detection techniques have been designed, including enforcing silence slots [11], publicizing back-off sequences [12, 13], exploiting spatial pattern of signal strength [14], measuring detector value [15]. There is also a crowdsourcing-based framework named SpecGuard [16] which explores dynamic power control at SUs to contain the spectrum permit in physical layer signals. However, all these works assume that all channels can be monitored at the same time. In this paper, we consider spectrum usage monitoring in multi-channel CRNs with limited monitoring resource. In particular, we deploy one monitor with multiple radios where each radio is in charge of one channel. The main problem is the selection of channels since monitoring all channels simultaneously is very energy-consuming and impractical. It is challenging because the information of MUs is unknown a priori. In addition, switching costs caused by changing channels must be considered. To solve this problem, we design effective online algorithms with guaranteed performance by formulating it as an adversarial (non-stochastic) multi-armed bandit (MAB) problem [17] with switching costs.

The MAB problem first introduced by Robbins [18] has been extensively studied in the literature. The classical MAB problem models the trade-offs faced by a gambler who aims to maximize his rewards over many turns by exploring different arms of slot machines and to exploit arms which have provided him more rewards than others. The gambler has no knowledge about reward of each arm a priori and only gains knowledge of the arms he has pulled. An MAB algorithm should specify a strategy by which the gambler chooses an arm at each turn.

This research was supported in part by NSF grants CNS-1444059, CNS-1619728 and CNS-1443966. The information reported here does not reflect the position or the policy of the federal government.

The performance of an algorithm is measured in regret, as will be elaborated in Section III. There are mainly two algorithm families based on different formulations of MAB. The upper confidence bounds (UCB) family of algorithms [19] works for stochastic MAB, whose regret can be as small as $O(\ln T)$ where T is the number of turns. However, these algorithms are established with the assumption that there exist fixed (though unknown) probability distributions of different arms to generate rewards, which is different from our model. The other algorithm family is the EXP3 family [17] for adversarial MAB. Auer [17] has studied MAB with no assumption on the rewards distribution and proposed algorithms with regret of $O(T^{1/2})$. However, switching costs are not considered in all above works. In our model, each time the monitor changes its monitoring channels, there are drastic costs in terms of delay, packet loss, and protocol overhead [20]. These costs must be taken into consideration when designing monitoring algorithms. Although there exists some work on stochastic multi-armed bandit problem with switching cost (MAB-SC) [20], little research has been done on adversarial MAB-SC. Dekel et al. [21] proved the lower bound of the regret for adversarial MAB-SC to be $\tilde{\Omega}(T^{2/3})$. In this paper, the upper bound of regret guaranteed by our algorithms matches this lower bound. Moreover, different from existing works, the strategy for each turn (or timeslot as in our model) is no longer a single arm because we consider a more general case where multiple channels can be monitored at the same time. Therefore, none of above algorithms can be directly applied to our problem.

In summary, we contribute in the following aspects:

- 1) We study the adversarial spectrum usage monitoring problem with unknown statistics in multi-channel CRNs, while considering the switching cost. We model this problem as an adversarial MAB-SC problem.
- 2) We first design an online spectrum usage monitoring algorithm, termed SpecWatch. We prove that the expected weak regret is $O(T^{2/3})$ which matches the lower bound in [21]. Therefore, SpecWatch is asymptotically optimal. Note that the expected value of normalized weak regret is guaranteed to be $O(1/\sqrt[3]{T})$, which converges to 0 as time horizon T approaches to ∞ .
- 3) To improve the performance of SpecWatch, we design another algorithm, SpecWatch⁺, which reduces the bias in selecting channels and explores all channels more efficiently. This algorithm guarantees the actual value of weak regret to be $O(T^{2/3})$, which is asymptotically optimal as well, with probability $1-\delta$, for any $\delta \in (0, 1)$.
- 4) Our algorithms guarantee the proved performance under any type of adversary settings. In addition, they can work with any spectrum misuse detection techniques in the current literature.

II. RELATED WORK

In this paper, we focus on the algorithm to determine the channels to choose, which is closer to the sniffer-channel (sniffers are also referred to as monitors) assignment problem [22] in wireless networks.

In [23], Yeo et al. were the first to develop a framework exploiting dedicated sniffers to monitor WiFi networks and identify malicious usages. Cheng et al. [24, 25] proposed an infrastructure and modeling techniques to monitor and analyze network behavior.

Spectrum usage monitoring problem has been formulated as optimization problems with different objectives. In [26], Shin and Bagchi modeled the channel assignment for monitoring wireless mesh networks as maximum-coverage problem with group budget constraints. They then extended it to the model where monitors may make errors due to poor reception [27]. Along the same line, Nguyen et al. [28] focus on the weighted version of the problem, where users to be covered have weights. To maximize the captured data of interest, Chen et al. [29] utilized support vector regression to guide monitors to intelligently select channels. Considering similar objectives, Shin et al. [30] designed a cost-effective distributed algorithm. With a different approach, Yan et al. [31] solved the problem by predicting secondary users' access patterns. However, we consider a different objective in this paper, which is to capture spectrum misuses. In addition, we assume no information about the malicious users.

The closest works to ours were presented in [32–36]. In [32], Arora and Szepesvari first modeled the spectrum usage monitoring problem as an multi-armed bandit problem (MAB) to monitor the maximum number of active users. They designed two algorithms to learn sequentially the user activities while making channel assignment decisions. Observing the above algorithms suffer from high computation cost, Zheng et al. [33] traded off between the rate of learning and the computation cost. They proposed a centralized online approximation algorithm and show that it incurs sub-linear regret bounds over time and a distributed algorithm with moderate message complexity. In [34], Le et al. considered switching costs for the first time and utilized Upper Confident Bound-based (UCB) policy [37] which enjoys a logarithmic regret bound in time that depends sublinearly on the number of arms, while its total switching cost grows in the order of $O(\log(\log T))$. Considering a different objective, Yi et al. [35] used UCB to capture as much as interested user data.

However, these works used the stochastic MAB model, where the rewards for playing each arm are generated independently from unknown but fixed distributions. Our model, in contrast, does not make such assumptions. The only work considered the similar problem model to ours is [36], where Xu et al. tried to capture packets of target SUs for CRN forensics. However, they did not provide any algorithm whose actual weak regret can be bounded with confidence value.

III. SYSTEM MODEL AND PROBLEM STATEMENT

We consider a cognitive radio network which adopts a hierarchical access structure with primary users (PUs) and secondary users (SUs). We assume the spectrum is divided into a set $\mathcal{K} = \{1, 2, \dots, k, \dots, K\}$ of K channels. The total time period is discretized into a set $\mathcal{T} = \{1, 2, \dots, t, \dots, T\}$ of T timeslots. Ideally, SUs seek spectrum opportunities

among K channels in a non-intrusive manner. However, the malicious users (MUs) may perform unauthorized access or selfish access. We consider the scenario where there exists one monitor with l radios and a set $\mathcal{M} = \{1, 2, \dots, m, \dots, M\}$ of M MUs. Note that for the case of multiple monitors, if there is a central controller, it is equivalent to one monitor with the same number of radios; otherwise, each monitor can execute our algorithms independently.

Since the monitor is equipped with l radios, it can monitor up to l channels at the same time. Assume that one radio is tuned to monitor channel k , and there are M_k MUs on that channel, then the detection probability of that radio to successfully detect MUs' presence is $p_d(M_k)$, which is dependent on the monitor's hardware and the detection technique. Any technique in [11–15] can be adopted to detect spectrum misuses. In practice, the detection probability will also be dependent on the presence of PU and other SUs. However, since our algorithms do not require the knowledge of the detection probability, we simplify the notion to $p_d(M_k)$ where it seems that only M_k matters.

Let $\{0, 1, \dots, l\}^K$ denote the strategy space of the monitor. A strategy s is represented as $(a_{s1}, a_{s2}, \dots, a_{sK})$, where the value of the a_{sk} represents the number of radios assigned to monitor channel k . Therefore, $\sum_{k \in \mathcal{K}} a_{sk} = l$. For example, considering 4 channels and a monitor with 3 radios, strategy $(0, 1, 0, 2)$ indicates that one radio is tuned to monitor channel 2 and two radios are tuned to monitor channel 4. For notational simplicity, we will write $k \in s$ instead of $a_{ik} \geq 1$ to denote that channel k is chosen in strategy s . Since each radio is assigned one out of K channels to monitor, and we have l radios in total, the number of strategies is $S = K^l$. The whole strategy set is represented as $\mathcal{S} = \{1, 2, \dots, s, \dots, S\}$. Note that K and l are usually small. For example, the regulated 2.4 GHz band is divided into only 14 channels. The maximum number of radios on each monitor defined by both the IEEE 802.11af and 802.11n Standards is set to be 4 [38, 39].

At the beginning of timeslot $t \in \mathcal{T}$, the monitor selects only one strategy from the strategy set \mathcal{S} , and we denote the chosen strategy as X_t . We assume the switching cost $c(X_{t-1}, X_t) \in [0, 1]$, but our algorithm can be generalized to any range $[\underline{c}, \bar{c}]$, $\underline{c} < \bar{c}$ by scaling, where \underline{c} and \bar{c} are the minimum value and maximum value of the switching cost, respectively. For notational simplicity, let $c(X_0, X_1) = c(X_1)$. Clearly, $c(X_{t-1}, X_t) = 0$ if $X_{t-1} = X_t$.

Threat Model: At each timeslot $t \in \mathcal{T}$, each MU $m \in \mathcal{M}$ chooses one channel to attack (conduct misuses) according to its attack probability distribution $\mathbf{P}_t^m = \{P_{t,1}^m, \dots, P_{t,K}^m\}$ where $P_{t,k}^m$ denotes the probability of MU m attacking channel k in timeslot t . Since MUs may not attack in some timeslot, $\sum_{k \in \mathcal{K}} P_{t,k}^m \leq 1$ for any $m \in \mathcal{M}$ and $t \in \mathcal{T}$. We consider two types of adversary:

- 1) Oblivious Adversary (Stochastic): The MUs keep their attack patterns regardless of how the monitor work. For any MU m , the attack distribution \mathbf{P}_t^m remains the same throughout the time horizon. In this paper, we consider three different adversary settings (elaborated

in Section V): fixed adversary, uniform adversary, and normal adversary.

- 2) Adaptive Adversary (Adversarial): The MUs know every action of the monitor from the beginning to the current timeslot and adjust their strategies accordingly based on any learning algorithms, i.e., the attack distribution might change with time.

Now we define the reward for the monitor. The *actual strategy reward* of choosing strategy s in timeslot t is

$$g_{s,t} \stackrel{\text{def}}{=} \begin{cases} \sum_{k \in s} f_{k,t} & \text{if } s = X_t, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where the *actual channel reward* $f_{k,t}$ is defined as

$$f_{k,t} \stackrel{\text{def}}{=} \begin{cases} r & \text{if channel } k \in X_t \text{ and misuse is detected,} \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where the *unit reward* r is assumed to be scaled and satisfies $rl \leq 1$ for the purpose of mathematical analysis. Note that the probability of at least one MU being detected on monitored channel k is determined by the number of radios on that channel M_k , the detection probability $p_d(M_k)$ and the action of MUs $\{P_{t,k}^m\}_{m=1}^M$. We denote the detection probability on channel k by adopting strategy X_t at timeslot t as $P_d(a_{X_t k}, p_d(M_k), \{P_{t,k}^m\}_{m=1}^M)$ which is assumed to be a non-decreasing function in $a_{X_t k}$, $p_d(M_k)$, and $P_{t,k}^m$. Thus, the channel reward $f_{k,t}$ is r with probability $P_d(a_{X_t k}, p_d(M_k), \{P_{t,k}^m\}_{m=1}^M)$. Note that the knowledge of this probability is not required.

Assume the monitor follows the strategy sequence X_1, X_2, \dots, X_T generated by any monitoring algorithm \mathbb{A} . At the end of timeslot T , the *cumulative strategy reward* is

$$G_{\mathbb{A}} \stackrel{\text{def}}{=} \sum_{t=1}^T g_{X_t, t}.$$

Meanwhile, the monitor incurs *cumulative switching cost*

$$L_{\mathbb{A}} \stackrel{\text{def}}{=} \sum_{t=1}^T c(X_{t-1}, X_t).$$

Thus, the *utility* of the monitor by choosing algorithm \mathbb{A} is

$$U_{\mathbb{A}} = G_{\mathbb{A}} - L_{\mathbb{A}}. \quad (3)$$

To measure the performance of algorithm \mathbb{A} , we use a special case of the worst-case regret, *weak regret* [17], as the metric. The weak regret of algorithm \mathbb{A} is the difference between the utility by using *best fixed algorithm* and the actual utility by using algorithm \mathbb{A} . A fixed algorithm chooses only one strategy for all timeslots and never switches. The best fixed algorithm is the one resulting the highest utility among all fixed algorithms. The strategy chosen in best fixed algorithm is called the *best strategy*, denoted by Z_{best} . Formally, the utility by using best fixed algorithm is

$$U_{\text{best}} \stackrel{\text{def}}{=} \max_{s \in \mathcal{S}} \left(\sum_{t=1}^T g_{s,t} - c(X_1 = s) \right), \quad (4)$$

where $c(X_1 = s)$ represents the switching cost incurred at the first timeslot by choosing strategy $X_1 = s$. Since no switches will happen in future timeslots, the cumulative switching cost $L_{\mathbb{A}} = c(X_1 = s) + \sum_{t=2}^T c(X_{t-1} = s, X_t = s) = c(X_1 = s)$. Note that the best strategy can only be found in hindsight. Now we can define the weak regret of algorithm \mathbb{A} as

$$R_{\mathbb{A}} \stackrel{\text{def}}{=} U_{\text{best}} - U_{\mathbb{A}}. \quad (5)$$

Problem Statement: Given K channels, time horizon T , and a monitor with l radios, our objective is to design online spectrum usage monitoring algorithms such that the weak regret is minimized, with different adversaries. We make no assumption on the knowledge of the probability functions $p_d(M_k)$ and $P_d(a_{X_t k}, p_d(M_k), \{P_{t,k}^m\}_{m=1}^M)$. In addition, the attack distribution \mathbf{P}_t^m and the reward of choosing a strategy are unknown a priori. Furthermore, any desired algorithm need to balance not only the trade-off between exploration and exploitation, but also that between strategy rewards and switching costs. Therefore, this is a very challenging problem.

IV. MONITORING ALGORITHMS AND PERFORMANCE ANALYSIS

To answer those questions, we design two effective online algorithms, SpecWatch and SpecWatch⁺. In SpecWatch, we select strategies based on monitoring history, and repeat the same strategy for certain timeslots to reduce switching costs. Whereas, in SpecWatch⁺, we select strategies more strategically to improve the performance. We also rigorously bound the weak regrets of both algorithms. Due to space limitation, all proofs are omitted and can be found in [40].

A. Design Rationale

Both SpecWatch and SpecWatch⁺ are based on the batching version of exponential-weight algorithm for exploration and exploitation, where the idea of batching is inspired by [41]. Our algorithms work as follows: The timeslots $1, 2, \dots, T$ are grouped into consecutive and disjoint batches of size τ . The set of batches is $\mathcal{J} = \{1, 2, \dots, j, \dots, J\}$, where $J = \lceil T/\tau \rceil$. Let $t_j = (j-1)\tau$, then the j -th batch starts from timeslot $t_j + 1$ and ends at timeslot $t_j + \tau$ as shown in Fig. 1. In both algorithms, strategies are selected according to their weights. Within each batch, the chosen strategy remains the same. At the end of each batch, strategy weights are updated according to the strategy reward. The main notations are summarized in Table I.



Fig. 1 Example of batching timeslots with batch size $\tau = 4$

B. Primary Spectrum Usage Monitoring Algorithm

1) *Algorithm Design:* In this section, we elaborate the details of SpecWatch, a primary spectrum usage monitoring algorithm. The pseudo-code of SpecWatch is illustrated in Algorithm 1.

TABLE I ALGORITHM NOTATIONS

$f_{k,t}$	actual channel reward of channel k in timeslot t
$g_{s,t}$	actual strategy reward of strategy s in timeslot t
$f_{k,j}$	actual channel reward of channel k in batch j
$g_{s,j} = \sum_{k \in \mathcal{S}} f_{k,j}$	actual strategy reward of strategy s in batch j
$\bar{f}_{k,j}$	average channel reward of channel k in batch j
$\bar{g}_{s,j} = \sum_{k \in \mathcal{S}} \bar{f}_{k,j}$	average strategy reward of strategy s in batch j
$\bar{f}'_{k,j}$	virtual channel reward of channel k in batch j
$\bar{g}'_{s,j} = \sum_{k \in \mathcal{S}} \bar{f}'_{k,j}$	virtual strategy reward of strategy s in batch j
$h_{k,j}$	channel weight of channel k in batch j
$w_{s,j} = \prod_{k \in \mathcal{S}} h_{k,j}$	strategy weight of strategy s in batch j
$W_j = \sum_{s \in \mathcal{S}} w_{s,j}$	total weight of all strategies in batch j
$q_{k,j}$	channel probability of channel k in batch j
$p_{s,j}$	strategy probability of strategy s in batch j

Algorithm 1: SpecWatch

```

1 Parameters:  $\gamma \in (0, 1]$ ,  $\tau \in [1, T]$ ,  $J = \lceil \frac{T}{\tau} \rceil$ .
2 Initialization: set  $w_{s,1} = 1$  for all  $s \in \mathcal{S}$ .
3 for  $j = 1, \dots, J$  do
4   Calculate the probability of each  $s \in \mathcal{S}$  using (6).
5   Choose strategy  $Z_j \in \mathcal{S}$  randomly accordingly to the
   probability distribution  $p_{1,j}, \dots, p_{S,j}$  and incur
   switching cost  $c(Z_{j-1}, Z_j)$ .
6   Monitor using this strategy for  $\tau$  timeslots, i.e.,
    $X_{t_j+i} = Z_j$  for  $1 \leq i \leq \tau$ , and receive the reward of
   the whole batch.
7   Update strategy weight for each  $s \in \mathcal{S}$  using (14).
8 end
```

The initial step is to set the weight of each strategy to 1. SpecWatch proceeds batch by batch. For each batch j , we follow three phases as shown below.

Strategy Probability Calculation. We first calculate the probability of choosing strategy $s \in \mathcal{S}$ using

$$p_{s,j} = (1 - \gamma) \frac{w_{s,j}}{W_j} + \frac{\gamma}{S}, \quad (6)$$

where $\gamma \in (0, 1]$ is a parameter in, $w_{s,j}$ is the strategy weight of strategy s in batch j , and $W_j = \sum_{s \in \mathcal{S}} w_{s,j}$. It is a weighted average of two terms, where the first is to exploit strategies with good reward history, and the second guarantees the exploration over all strategies. γ controls the balance between them.

Strategy Selection and Spectrum Monitoring. We then select a strategy $Z_j \in \mathcal{S}$ randomly according to the probabilities calculated above. The monitor keeps using Z_j for all τ timeslots in batch j , i.e., $X_{t_j+i} = Z_j$ for $1 \leq i \leq \tau$. Therefore, the monitor only incurs switching cost $c(Z_{j-1}, Z_j)$ once for the whole batch j .

Depending on the misuse behavior of MUs (discussed in Section III), the monitor receives rewards on monitored channels accordingly. The strategy reward gained by the monitor

is the summation of rewards over all monitored channels. Specifically, let $f_{k,j}$ be the total actual reward of channel k throughout batch j , i.e.,

$$f_{k,j} = \sum_{i=1}^{\tau} f_{k,t_j+i}, \quad (7)$$

where f_{k,t_j+i} is either r or 0 as defined in (2); then the actual strategy reward $g_{Z_j,j}$ of strategy Z_j throughout batch j is

$$g_{Z_j,j} = \sum_{k \in Z_j} f_{k,j} = \sum_{k \in Z_j} \sum_{i=1}^{\tau} f_{k,t_j+i}. \quad (8)$$

Strategy Weight Update. In the end, we update the weight of each strategy in the following steps. First, we calculate the average channel reward of each monitored channel $k \in Z_j$,

$$\bar{f}_{k,j} = \frac{1}{\tau} \sum_{i=1}^{\tau} f_{k,t_j+i}, \quad (9)$$

so that $\bar{f}_{k,j} \in [0,1]$. We also calculate the probability of choosing channel $k \in Z_j$ by summing up the probabilities of strategies containing that channel,

$$q_{k,j} = \sum_{s:k \in s} p_{s,j}. \quad (10)$$

Based on (9) and (10), we calculate the *virtual channel reward*,

$$\bar{f}'_{k,j} = \frac{\bar{f}_{k,j}}{q_{k,j}}. \quad (11)$$

Then we update each *channel weight* by

$$h_{k,j+1} = h_{k,j} \exp(\gamma \bar{f}'_{k,j} / S), \quad (12)$$

where $h_{k,1} = 1$ for all $k \in \mathcal{K}$. Note that using virtual channel rewards instead of average channel rewards compensates the rewards of channels with low probabilities. Among the channels receiving rewards, those with lower probabilities can obtain higher virtual channel rewards, and therefore higher channel weights.

Finally, we give the formal definition of *strategy weight*, which is defined as

$$w_{s,j} = \prod_{k \in s} h_{k,j}. \quad (13)$$

Combining (12) and (13), we can directly update the strategy weight for each $s \in \mathcal{S}$ by

$$w_{s,j+1} = w_{s,j} \exp(\gamma \bar{g}'_{s,j} / S), \quad (14)$$

where $\bar{g}'_{s,j}$ is the *virtual strategy reward* for each $s \in \mathcal{S}$, i.e., $\bar{g}'_{s,j} = \sum_{k \in s} \bar{f}'_{k,j}$. Note that by combining (9), (10) and (11), we can directly calculate $\bar{g}'_{s,j}$ directly by

$$\bar{g}'_{s,j} = \sum_{k \in s} \bar{f}'_{k,j} = \sum_{k \in s} \frac{\bar{f}_{k,j}}{q_{k,j}} = \frac{1}{\tau} \sum_{k \in s} \frac{\sum_{i=1}^{\tau} f_{k,t_j+i}}{\sum_{s:k \in s} p_{s,j}}. \quad (15)$$

Remark. We do not update strategy weights based on actual strategy rewards (8), but instead calculate channel weights (13) first. This is because the rewards of monitored channels provide useful information on those unchosen strategies containing these channels.

2) Performance Analysis: To analyze the performance of SpecWatch, we first bound the difference between the reward achieved by SpecWatch and that by best fixed algorithm (Lemma 1), and then prove the bound of the expected weak regret (Theorem 1). For better understanding of Theorem 1, we present a specific bound obtained by a particular choice of parameters γ and τ (Corollary 1).

Recalling (3), (4) and (5), the expected weak regret of SpecWatch is

$$R_{SW} = (G_{best} - L_{best}) - (G_{SW} - L_{SW}), \quad (16)$$

where SW is short for SpecWatch. Since best fixed algorithm never switches, and SpecWatch only switches between batches for at most J times, their cumulative switching costs are

$$L_{best} = c(Z_{best}) \geq 0 \quad \text{and} \quad L_{SW} = \sum_{j=1}^J c(Z_{j-1}, Z_j) \leq J.$$

Thus, we have

$$L_{SW} - L_{best} \leq J. \quad (17)$$

Now it suffices to only consider the difference between rewards. An important observation is that we group the time horizon into batches, update the strategy probabilities only at the beginning of each batch, and use the average value of entire batch to update weight. Therefore, each batch can be considered as a round in conventional MAB. With this consideration, we introduce the notations below for our proofs,

$$\bar{G}_{SW} \stackrel{\text{def}}{=} \sum_{j=1}^J \bar{g}_{Z_j,j} \quad \text{and} \quad \bar{G}_{best} \stackrel{\text{def}}{=} \max_{s \in \mathcal{S}} \sum_{j=1}^J \bar{g}_{s,j}.$$

Note that

$$\begin{aligned} \bar{g}_{s,j} &= \sum_{k \in s} \bar{f}_{k,j} = \frac{1}{\tau} \sum_{k \in s} f_{k,j} = \frac{1}{\tau} \sum_{k \in s} \sum_{i=1}^{\tau} f_{k,t_j+i} \\ &= \frac{1}{\tau} \sum_{i=1}^{\tau} \sum_{k \in s} f_{k,t_j+i} = \frac{1}{\tau} \sum_{i=1}^{\tau} g_{s,t_j+i}. \end{aligned}$$

Thus we have

$$G_{SW} = \sum_{t=1}^T g_{X_t,t} = \tau \sum_{j=1}^J \bar{g}_{s,j} = \tau \bar{G}_{SW}. \quad (18)$$

Similarly,

$$G_{best} = \tau \bar{G}_{best}. \quad (19)$$

We now give the bound of the expected difference between \bar{G}_{SW} and \bar{G}_{best} .

Lemma 1. For any type of adversaries, any $T > 0$, and any $\gamma \in (0, 1]$, we have

$$\mathbb{E} [\bar{G}_{best} - \bar{G}_{SW}] \leq (e - 1) \gamma \bar{G}_{best} + \frac{S \ln S}{\gamma}, \quad (20)$$

where e is the base of natural logarithm.

Now taking the bound of switching costs into consideration, we have the following theorem:

Theorem 1. The expected weak regret of SpecWatch is $O(T^{2/3})$ with parameters

$$\gamma = A_\gamma T^{-1/3} \in (0, 1] \quad \text{and} \quad \tau = A_\tau T^{1/3} \in [1, T],$$

where A_γ and A_τ are constants. Specifically,

$$\mathbb{E}[R_{SW}] \leq \left((e-1)A_\gamma + \frac{A_\tau S \ln S}{A_\gamma} + \frac{1}{A_\tau} \right) T^{\frac{2}{3}}. \quad (21)$$

For better understanding of Theorem 1, we now give a specific bound by choosing particular parameters.

Corollary 1. When $T \geq (e-1)S \ln S$, with parameters

$$\gamma = \sqrt[3]{\frac{S \ln S}{(e-1)^2 T}} \quad \text{and} \quad \tau = \sqrt[3]{\frac{T}{(e-1)S \ln S}},$$

the expected weak regret of SpecWatch is

$$\mathbb{E}[R_{SW}] \leq 3((e-1)S \ln S)^{\frac{1}{3}} T^{\frac{2}{3}}. \quad (22)$$

Remark. The expected weak regret of SpecWatch is bounded to $O(T^{2/3})$, which matches the lower bound proved in [21]. Thus, SpecWatch is asymptotically optimal. If we calculate the normalized weak regret R_{SW}/T , i.e., amortizing the regret to every timeslot, then it is clear that the expected value of normalized weak regret converges to 0 as T approaches to ∞ .

C. Improved Spectrum Usage Monitoring Algorithm

We have already proved that SpecWatch is an effective online spectrum usage monitoring algorithm with expected normalized regret converging to 0. Though the *expectation* provides a quite legitimate estimate on the performance of SpecWatch, the actual value of weak regret may sometimes deviate a lot from the expected bound as expectation just represents the mean. In this section, we present an improved algorithm, SpecWatch⁺, and prove the upper bound of its weak regret with arbitrary confidence level.

Algorithm 2: SpecWatch⁺

- 1 **Parameters:** $\gamma \in (0, \frac{1}{2})$, $\eta > 0$, $\beta \in (0, 1)$, $\delta \in (0, 1)$, $\tau \in [1, T]$, $J = \lceil \frac{T}{\tau} \rceil$.
 - 2 **Initialization:** set $w_{s,1} = 1$ for all $s \in \mathcal{S}$.
 - 3 **for** $j = 1, \dots, J$ **do**
 - 4 Calculate the probability of each $s \in \mathcal{S}$ using (23).
 - 5 Choose strategy $Z_j \in \mathcal{S}$ randomly accordingly to the probability distribution $p_{1,j}, \dots, p_{S,j}$ and incur switching cost $c(Z_{j-1}, Z_j)$.
 - 6 Monitor using this strategy for τ timeslots, i.e., $X_{t_j+i} = Z_j$ for $1 \leq i \leq \tau$, and receive the reward of the whole batch.
 - 7 Update strategy weight for each $s \in \mathcal{S}$ using (25).
 - 8 **end**
-

1) *Algorithm Design:* The pseudo-code of SpecWatch⁺ is illustrated in Algorithm 2. The basic three phases remain the same as SpecWatch. We only modify strategy probability

calculation, virtual channel reward calculation, and channel weight calculation.

For calculating strategy probabilities, we introduce a new concept called *covering strategy set*. A covering strategy set $\mathcal{C} \in \mathcal{S}$ is a set of strategy that *covers* all channels \mathcal{K} , where a channel $k \in \mathcal{K}$ is covered if there is a strategy $s \in \mathcal{C}$ such that $k \in s$. Note that the selection of \mathcal{C} does not affect the performance of SpecWatch⁺. Thus, we randomly construct the covering strategy set under the constraint that its size $C \stackrel{\text{def}}{=} |\mathcal{C}|$ is less than or equal to K . The probability of each strategy s is calculated by

$$p_{s,j} = \begin{cases} (1-\gamma) \frac{w_{s,j}}{W_j} + \frac{\gamma}{C} & \text{if } s \in \mathcal{C}, \\ (1-\gamma) \frac{w_{s,j}}{W_j} & \text{otherwise.} \end{cases} \quad (23)$$

In this way, the strategies in the covering set are more likely to be chosen than others. As a result, SpecWatch⁺ can cover all channels more quickly, and thus reveal the best channels sooner, which expedites the exploration for the best strategy. Note that this modification also has impact on the calculation of channel probabilities. In particular, the channel probability of $k \in \mathcal{K}$ in batch j is

$$q_{k,j} = \sum_{s:k \in s} p_{s,j} = (1-\gamma) \frac{\sum_{s:k \in s} w_{s,j}}{W_j} + \frac{\gamma |\{s \in \mathcal{C} : k \in s\}|}{C}.$$

For calculating virtual channel rewards, compared to (11), we introduce a new parameter β and have

$$\bar{f}'_{k,j} = \begin{cases} \frac{\bar{f}_{k,j} + \beta}{q_{k,j}} & \text{if } k \in Z_j, \\ \frac{\beta}{q_{k,j}} & \text{otherwise.} \end{cases}$$

Note that in SpecWatch, every unmonitored channel has average channel reward $\bar{f}_{k,j} = 0$, thus its virtual channel reward is also 0. However, we could also gain rewards on an unmonitored channel if we had monitored it, which indicates that the virtual channel reward of that channel should be positive. With this concern, we use parameter β to reduce the bias between monitored and unmonitored channels.

For calculating channel weights, we introduce a new parameter η . In SpecWatch, the parameter γ controls both the influence of strategy weight on strategy probability (6) and the effect of channel reward on channel weight (12). However, it is more reasonable to control them separately. We keep using γ in calculating strategy probability in SpecWatch⁺ and use parameter η in calculating channel weight. Specifically, the channel weight of k is

$$h_{k,j+1} = h_{k,j} \exp(\eta \bar{f}'_{k,j-1}), \quad (24)$$

where $h_{k,1} = 1$ for all $k \in \mathcal{K}$. Thus, the strategy weight of s is updated by using

$$w_{s,j+1} = w_{s,j} \exp(\eta \bar{g}'_{s,j}). \quad (25)$$

Similar to (15), the virtual strategy reward of s is

$$\bar{g}'_{s,j} = \frac{1}{\tau} \sum_{k \in s} \frac{\sum_{i=1}^{\tau} f_{k,t_j+i}}{(1-\gamma) \frac{\sum_{s:k \in s} w_{s,j}}{W_j} + \frac{\gamma |\{s \in \mathcal{C} : k \in s\}|}{C}}. \quad (26)$$

2) *Performance Analysis*: We analyze SpecWatch⁺ with a similar approach as for SpecWatch. We first present two lemmas (Lemma 2 and Lemma 3) to analyze rewards, then we consider the switching costs and claim the bound of weak regret in Theorem 2. We also provide a specific choice of parameters and show the resulting bound in Corollary 2.

As in Section IV-B, we first define the cumulative average reward to be

$$\bar{G}_{SW+} \stackrel{\text{def}}{=} \sum_{j=1}^J \bar{g}_{Z_j, j}, \quad (27)$$

where Z_1, Z_2, \dots, Z_J is strategy sequence generated by SpecWatch⁺. Similar to (18), we have

$$G_{SW+} = \tau \bar{G}_{SW+}. \quad (28)$$

For the ease of our remaining analysis, we introduce the following notations,

$$\bar{F}_{k,n} \stackrel{\text{def}}{=} \sum_{j=1}^n \bar{f}_{k,j} \quad \text{and} \quad \bar{F}'_{k,n} \stackrel{\text{def}}{=} \sum_{j=1}^n \bar{f}'_{k,j} \quad \text{for } k \in \mathcal{K}, \quad (29)$$

$$\bar{G}_{s,n} \stackrel{\text{def}}{=} \sum_{j=1}^n \bar{g}_{s,j} \quad \text{and} \quad \bar{G}'_{s,n} \stackrel{\text{def}}{=} \sum_{j=1}^n \bar{g}'_{s,j} \quad \text{for } s \in \mathcal{S}, \quad (30)$$

where n is an arbitrary batch.

We first show the deviation of the actual cumulative reward from the virtual cumulative reward of each channel.

Lemma 2. *For any $\delta \in (0, 1)$, $\beta \in (0, 1)$ and $k \in \mathcal{K}$ in SpecWatch⁺, we have*

$$\Pr \left[\bar{F}_{k,n} \geq \bar{F}'_{k,n} + \frac{1}{\beta} \ln \frac{K}{\delta} \right] \leq \frac{\delta}{K}. \quad (31)$$

Next, we bound the difference between the cumulative reward gained by best fixed algorithm and that by SpecWatch⁺.

Lemma 3. *For any type of adversaries, any $T > 0$, $\gamma \in (0, 1/2)$, $\tau \in [1, T]$, $\beta \in (0, 1)$, and $\eta > 0$ satisfying $2\eta lC \leq \gamma$, we have*

$$\bar{G}_{\text{best}} - \bar{G}_{SW+} \leq \gamma J + 2\eta lCJ + \frac{l}{\beta} \ln \frac{K}{\delta} + \frac{\ln S}{\eta} + \beta KJ.$$

with probability at least $1 - \delta$ for any $\delta \in (0, 1)$.

Taking the the bound of switching costs into consideration, we now bound the weak regret of SpecWatch⁺.

Theorem 2. *With probability at least $1 - \delta$, the weak regret of SpecWatch⁺ is bounded by $O(T^{2/3})$. In particular, choosing*

$$\begin{aligned} \tau &= B_\tau T^{1/3} \in [1, T], \\ \gamma &= B_\gamma T^{-1/3} \in (0, 1/2), \\ \beta &= B_\beta T^{-1/3} \in (0, 1), \\ \text{and } \eta &= \frac{B_\gamma}{2lC} T^{-1/3}, \end{aligned}$$

where B_τ , B_γ , and B_β are constants, we have

$$R_{SW+} \leq \left(2B_\gamma + B_\beta K + B_\tau \left(\frac{l \ln \frac{K}{\delta}}{B_\beta} + \frac{\ln S}{B_\eta} \right) + \frac{1}{B_\tau} \right) T^{\frac{2}{3}}.$$

Note that finding the optimal choice of parameters is challenging because it is depending on the real values of C , K , S , as well as the confidence level $1 - \delta$. We now give an example choice of parameters to reach a specific bound.

Corollary 2. *Under the condition of*

$$T \geq \max \left\{ B^2, \frac{8(lC \ln S)^{3/2}}{B}, \frac{(\frac{l}{K} \ln \frac{K}{\delta})^{3/2}}{B} \right\}, \quad (32)$$

using parameters

$$\begin{aligned} \tau &= B^{-2/3} T^{1/3}, \\ \gamma &= \sqrt{lC \ln S} \cdot B^{-1/3} T^{-1/3}, \\ \beta &= \sqrt{\frac{l}{K} \ln \frac{K}{\delta}} \cdot B^{-1/3} T^{-1/3}, \\ \text{and } \eta &= \sqrt{\frac{\ln S}{4lC}} \cdot B^{-1/3} T^{-1/3}, \end{aligned}$$

where $B = 4\sqrt{lC \ln S} + 2\sqrt{lK \ln \frac{K}{\delta}}$, we have

$$R_{SW+} \leq 2 \left(4\sqrt{lC \ln S} + 2\sqrt{lK \ln \frac{K}{\delta}} \right)^{\frac{2}{3}} T^{\frac{2}{3}}, \quad (33)$$

with probability at least $1 - \delta$.

Remark. Note that SpecWatch and SpecWatch⁺ have the same theoretical upper bound on the weak regret. Thus it is not guaranteed that SpecWatch⁺ always outperforms SpecWatch in reality, as shown in Section V. The improvement over SpecWatch is the fact that SpecWatch⁺ guarantees the actual weak regret is bounded with any given confidence level.

V. PERFORMANCE EVALUATION

We conduct extensive simulations to demonstrate the performance of our proposed online spectrum monitoring algorithms, SpecWatch and SpecWatch⁺. We first show the convergence of normalized weak regrets of both algorithms and then compare the actual cumulative utilities returned by each algorithm under different adversary settings. We also demonstrate the impact of the detection probability, the number of radios, the number of MUs, and adversary settings, on the algorithm performance.

In the simulation setting, we consider $K = 8$ channels, and we deploy a monitor with $l = 3$ radios. We set the unit reward of successfully detecting on a single channel to be $r = 0.3$ and the unit switching cost of tuning one radio to be $c = 0.03$. If not specified, the detection probability of each radio is set to be $p_d = 0.9$ as it is the recommended detection accuracy in consistent with [42]. The parameters of both algorithms are chosen as in the corollaries. If the monitor uses SpecWatch⁺, we set $\delta = 0.5$ so that the weak regret is relatively small with an acceptable confidence level.

We assume there are $m = 3$ MUs attacking channels either obliviously or adaptively. Specifically, we consider four adversary settings,

- **Fixed adversary (Fixed):** Each MU selects a fixed channel and never switches throughout the time horizon T .

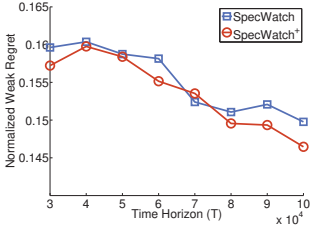


Fig. 2 Convergence of weak regrets of SpecWatch and SpecWatch⁺

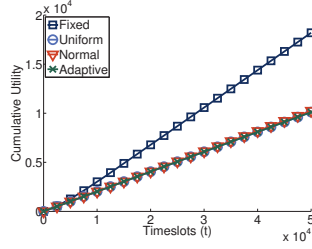


Fig. 3 Cumulative utility of SpecWatch under different adversary settings

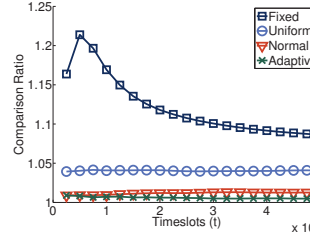


Fig. 4 Comparison of SpecWatch and SpecWatch⁺ under different adversary settings

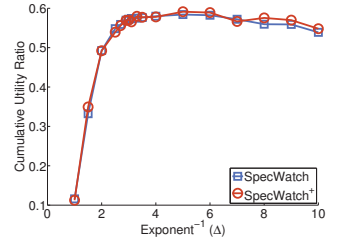


Fig. 5 Impact of batch size on cumulative utility

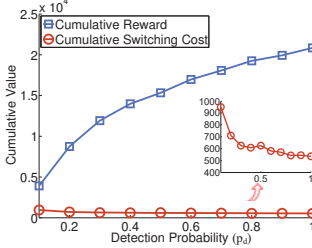


Fig. 6 Impact of detection probability on cumulative reward and cumulative switching cost

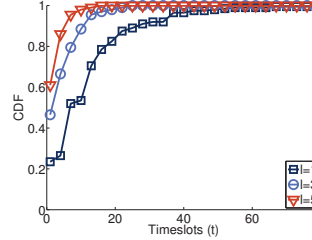


Fig. 7 CDF of expected number of timeslots to detect the first MU with different number of radios

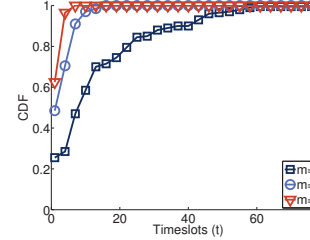


Fig. 8 CDF of expected number of timeslots to detect the first MU with different number of MUs

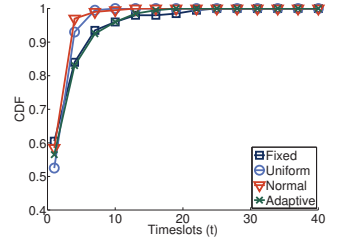


Fig. 9 CDF of expected number of timeslots to detect the first MU under different adversary settings

- Uniform adversary (*Uniform*): In every timeslot, each MU selects a channel uniformly at random.
- Normal adversary (*Normal*): In every timeslot, each MU selects a channel following the same normal distribution.
- Adaptive adversary (*Adaptive*): Each MU adopts modified SpecWatch, where the actual channel reward is r if the MU is not captured on that channel, and 0 otherwise.

The simulation results are shown below, and each of them is averaged over 100 trials.

Convergence. Fig. 2 shows the normalized weak regrets of both algorithms decrease with T , which supports our theoretical analysis that the normalized weak regret converges to 0 as $T \rightarrow \infty$. Here we only show the result with adaptive adversary since in other adversary settings, the results are similar. Note that when $T = 7000$, the convergence of SpecWatch is even better than SpecWatch⁺.

Cumulative Utility. Fig. 3 plots the actual utilities gained by SpecWatch. In the simulation, we fix the time horizon to be $T = 50000$ and study utilities change with time. The figure for SpecWatch⁺ is very similar and thus omitted. We observe that the cumulative utilities under fixed adversary greatly exceed the other three settings, and the monitor gains lowest cumulative utilities when confronting adaptive adversaries. We also compare the value of utilities gained by SpecWatch and SpecWatch⁺ in Fig. 4, where the comparison ratio represents U_{SW+}/U_{SW} . Since this ratio is always above 1, it indicates that SpecWatch⁺ always outperforms SpecWatch in such certain simulation setting. The outperformance is most obvious with fixed adversary because it is easier for SpecWatch⁺ to reveal the best strategy when MUs fix their channels. Whereas, the outperformance is least obvious with adaptive adversary because of the adaptiveness of MUs. Another observation is the decline of comparison ratio in fixed adversary setting,

which indicates that, as time goes by, SpecWatch eventually pinpoints the attacked channels and thus has the same utility with SpecWatch⁺.

Impact of Algorithm Parameters. Among all parameters of the two algorithms, the most important one is the batch size τ , which controls the trade-off between cumulative reward and cumulative switching cost. As shown in Fig. 5, we conducted simulations where the batch size τ was set to be exactly $T^{1/\Delta}$ and plotted how the cumulative utility ratio (the ratio between cumulative utility gained by SpecWatch or SpecWatch⁺ and the best fixed algorithm) changes with Δ , under the adaptive adversary. It is shown that both SpecWatch and SpecWatch⁺ achieve highest cumulative utility ratio when τ is around $T^{1/3}$, which is in consistency with our theoretical analysis.

Impact of System Parameters and Adversary Settings. In simulations, we fix the time horizon to be $T = 50000$. Since the impacts on both algorithms are similar, we only present results of SpecWatch⁺.

Fig. 6 shows the impact of detection probability p_d on the cumulative rewards and the cumulative switching cost. As expected, the cumulative reward grows with decreasing slope as the detection probability increases. The cumulative switching cost, however, has a decreasing trend. This is because the larger the detection probability, the more accurate for the monitor to evaluate each strategy; thus the best strategy is revealed more quickly, avoiding unnecessary switches and reducing cumulative switching cost.

We also study the impact the number of radios l , the number of MUs m , and the types of adversary on the performance of our algorithms. Fig. 7, Fig. 8, and Fig. 9 illustrate the cumulative distribution function (CDF) of expected number of timeslots to detect the first MU. In general, more radios or more MUs make it sooner for the monitor to detect

successfully. In Fig. 9, the monitor takes longest time to detect the first MU under adaptive adversary setting, which is as expected. Another observation is that the monitor also takes a long time in fixed adversary setting. This is because under uniform and normal adversary settings, both MUs and the monitor select channels randomly and thus have higher chance to select the same channel compared to that under fixed adversary setting.

VI. CONCLUSION

In this paper, we studied the adversarial spectrum usage monitoring problem with unknown statistics by formulating it as an adversarial multi-armed bandit problem with switching costs. To solve this problem, we designed two effective on-line algorithms, SpecWatch and SpecWatch⁺. We rigorously proved that their weak regrets are bounded by $O(T^{2/3})$, which matches the problem's lower bound. Thus, they are asymptotically optimal. Moreover, our algorithms can guarantee the proved performance under any adversary setting and are independent of the underlying misuse detection technique.

REFERENCES

- [1] F. S. P. T. Force, "Report of the spectrum efficiency working group," Tech. Rep., 2002.
- [2] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, 2007.
- [3] A. Anandkumar, N. Michael, and A. Tang, "Opportunistic spectrum access with multiple users: learning under competition," in *Proc. of INFOCOM*, 2010, pp. 1–9.
- [4] C. Santivanez, R. Ramanathan, C. Partridge, R. Krishnan, M. Condell, and S. Polit, "Opportunistic spectrum access: Challenges, architecture, protocols," in *Proc. of WICON*, 2006.
- [5] C. Tekin and M. Liu, "Online learning in opportunistic spectrum access: A restless bandit approach," in *Proc. of INFOCOM*, 2011, pp. 2462–2470.
- [6] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao, "Opportunistic spectrum access in unknown dynamic environment: A game-theoretic stochastic learning solution," *IEEE Trans. Wireless Commun.*, vol. 11, no. 4, pp. 1380–1391, 2012.
- [7] Q. Wang, K. Ren, and P. Ning, "Anti-jamming communication in cognitive radio networks with unknown channel statistics," in *Proc. of ICNP*, 2011, pp. 393–402.
- [8] L. Yang, Z. Zhang, B. Y. Zhao, C. Kruegel, and H. Zheng, "Enforcing dynamic spectrum access with spectrum permits," in *Proc. of MobiHoc*, 2012, pp. 195–204.
- [9] M. B. Weiss, M. Altamimi, and M. McHenry, "Enforcement and spectrum sharing: A case study of the 1695–1710 mhz band," in *Proc. of CROWNCOM*, 2013, pp. 7–12.
- [10] C. Sorrells, P. Potier, L. Qian, and X. Li, "Anomalous spectrum usage attack detection in cognitive radio wireless networks," in *Proc. of HST*, 2011, pp. 384–389.
- [11] G. Atia, A. Sahai, and V. Saligrama, "Spectrum enforcement and liability assignment in cognitive radio systems," in *Proc. of DySPAN*, 2008, pp. 1–12.
- [12] P. Kyasanur and N. H. Vaidya, "Detection and handling of mac layer misbehavior in wireless networks," in *Proc. of DSN*, 2003, pp. 173–182.
- [13] Y. Zhang and L. Lazos, "Countering selfish misbehavior in multi-channel mac protocols," in *Proc. of INFOCOM*, 2013, pp. 2787–2795.
- [14] S. Liu, L. J. Greenstein, W. Trappe, and Y. Chen, "Detecting anomalous spectrum usage in dynamic spectrum access networks," *Ad Hoc Networks*, vol. 10, no. 5, pp. 831–844, 2012.
- [15] J. Tang and Y. Cheng, "Selfish misbehavior detection in 802.11 based wireless networks: An adaptive approach based on markov decision process," in *Proc. of INFOCOM*, 2013, pp. 1357–1365.
- [16] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "Specguard: Spectrum misuse detection in dynamic spectrum access systems," in *Proc. of INFOCOM*, 2015, pp. 172–180.
- [17] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM J. on Computing*, vol. 32, no. 1, pp. 48–77, 2002.
- [18] H. Robbins, "Some aspects of the sequential design of experiments," *Bull. Amer. Math. Soc.*, vol. 58, no. 5, pp. 527–535, 1952.
- [19] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, no. 1, pp. 4–22, 1985.
- [20] L. Chen, S. Iellamo, and M. Coupechoux, "Opportunistic spectrum access with channel switching cost for cognitive radio networks," in *Proc. of ICC*, 2011, pp. 1–5.
- [21] O. Dekel, J. Ding, T. Koren, and Y. Peres, "Bandits with switching costs: $T^{2/3}$ regret," in *Proc. of STOC*, 2014, pp. 459–467.
- [22] Y. Song, X. Chen, Y.-A. Kim, B. Wang, and G. Chen, "Sniffer channel selection for monitoring wireless LANs," in *Proc. of WASA*, 2009, pp. 489–498.
- [23] J. Yeo, M. Youssef, and A. Agrawala, "A framework for wireless LAN monitoring and its applications," in *Proc. of WiSe*, 2004, pp. 70–79.
- [24] Y.-C. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: Solving the puzzle of enterprise 802.11 analysis," in *Proc. of SIGCOMM*, 2006, pp. 39–50.
- [25] Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benkö, J. Chiang, A. C. Snoeren, S. Savage, and G. M. Voelker, "Automating cross-layer diagnosis of enterprise wireless networks," in *Proc. of SIGCOMM*, 2007, pp. 25–36.
- [26] D.-H. Shin and S. Bagchi, "Optimal monitoring in multi-channel multi-radio wireless mesh networks," in *Proc. of MobiHoc*, 2009, pp. 229–238.
- [27] D.-H. Shin, S. Bagchi, and C.-C. Wang, "Toward optimal sniffer-channel assignment for reliable monitoring in multi-channel wireless networks," in *Proc. of SECON*, 2013, pp. 203–211.
- [28] H. Nguyen, G. Scalosub, and R. Zheng, "On quality of monitoring for multichannel wireless infrastructure networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 664–677, 2014.
- [29] S. Chen, K. Zeng, and P. Mohapatra, "Efficient data capturing for network forensics in cognitive radio networks," in *Proc. of ICNP*, 2011, pp. 176–185.
- [30] D.-H. Shin, S. Bagchi, and C.-C. Wang, "Distributed online channel assignment toward optimal monitoring in multi-channel wireless networks," in *Proc. of INFOCOM*, 2012, pp. 2626–2630.
- [31] Q. Yan, M. Li, F. Chen, T. Jiang, W. Lou, Y. T. Hou, and C.-T. Lu, "SpecMonitor: Towards efficient passive traffic monitoring for cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 10, pp. 5893–5905, 2014.
- [32] P. Arora, C. Szepesvári, and R. Zheng, "Sequential learning for optimal monitoring of multi-channel wireless networks," in *Proc. of INFOCOM*, 2011, pp. 1152–1160.
- [33] R. Zheng, T. Le, and Z. Han, "Approximate online learning algorithms for optimal monitoring in multi-channel wireless networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 1023–1033, 2014.
- [34] T. Le, C. Szepesvari, and R. Zheng, "Sequential learning for multi-channel wireless network monitoring with channel switching costs," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5919–5929, 2014.
- [35] S. Yi, K. Zeng, and J. Xu, "Secondary user monitoring in unslotted cognitive radio networks with unknown models," in *Proc. of WASA*, 2012, pp. 648–659.
- [36] J. Xu, Q. Wang, R. Jin, K. Zeng, and M. Liu, "Secondary user data capturing for cognitive radio network forensics under capturing uncertainty," in *Proc. of MILCOM*, 2014, pp. 935–941.
- [37] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2-3, pp. 235–256, 2002.
- [38] "IEEE Std 802.11af-2013-Amendment 5: Television white spaces operation," *IEEE-SA*, pp. 1–198, 2014.
- [39] "IEEE Std 802.11n-2009-Amendment 5: Enhancements for higher throughput," *IEEE-SA*, pp. 1–565, 2009.
- [40] SpecWatch: Adversarial spectrum usage monitoring in CRNs with unknown statistics. [Online]. Available: <http://inside.mines.edu/~7Emili/papers/SpecWatch/>
- [41] R. Arora, O. Dekel, and A. Tewari, "Online bandit learning against an adaptive adversary: from regret to policy regret," in *Proc. of ICML*, 2012, pp. 1503–1510.
- [42] J.-K. Choi and S.-J. Yoo, "Time-constrained detection probability and sensing parameter optimization in cognitive radio networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2013, no. 1, pp. 1–12, 2013.