

# Performance Comparison of Detection Schemes for MAC Layer Misbehavior<sup>1</sup>

Alvaro A. Cárdenas<sup>2</sup>, Svetlana Radosavac and John S. Baras  
Department of Electrical and Computer Engineering  
and the Institute for Systems Research  
University of Maryland  
College Park, MD, 20740  
Email: {acardena,svetlana,baras}@isr.umd.edu

**Abstract**—This paper revisits the problem of detecting greedy behavior in the IEEE 802.11 MAC protocol by evaluating the performance of two previously proposed schemes: DOMINO and the Sequential Probability Ratio Test (SPRT). The evaluation is carried out in four steps. We first derive a new analytical formulation of the SPRT that takes into account the discrete nature of the problem. Then we develop a new tractable analytical model for DOMINO. As a third step, we evaluate the theoretical performance of SPRT and DOMINO with newly introduced metrics that take into account the repeated nature of the tests. This theoretical comparison provides two major insights into the problem: it confirms the optimality of SPRT and motivates us to define yet another test, a nonparametric CUSUM statistic that shares the same intuition as DOMINO but gives better performance. We finalize the paper with experimental results, confirming our theoretical analysis and validating the introduction of the new nonparametric CUSUM statistic.

## I. INTRODUCTION

Communication protocols were designed under the assumption that all parties would obey the given specifications. However when these protocols are implemented in an untrusted environment, a misbehaving party can deviate from the protocol specification and achieve better performance at the expense of honest participants (e.g. changing congestion parameters in TCP, free-riding in p2p networks etc.) In this paper we focus our attention to misbehavior at the IEEE 802.11 MAC layer protocol. Examples of misbehavior at the MAC layer can include a user modifying the parameters for accessing the channel in order to obtain a better throughput, or a network card with an inaccurate implementation of the protocol [1].

MAC layer protocol misbehavior has been previously studied in the literature, where it has been identified that a selfish user can implement a whole range of strategies to maximize its access to the medium. However, the most challenging detection task is that of detecting backoff manipulation [2], [1].

The current literature offers two major approaches to address this problem. The first set of approaches provides solutions based on modification of the current MAC layer protocol by making the monitoring stations aware of the backoff values of its neighbors. The approach proposed in [3] assumes existence of a trustworthy receiver that can detect misbehavior of the sender and penalize it by assigning him higher back-off values for subsequent transmissions. A decision about protocol deviation is reached if the observed number of idle slots of the sender is smaller than a pre-specified fraction of the allocated back-off. The sender is labeled as misbehaving if it turns out to deviate continuously based on a cumulative metric over a sliding window. The work in [4] attempts to prevent scenarios of colluding sender-receiver pairs using a similar approach.

A different line of thought is followed in [2], [5], [6], where the authors propose misbehavior detection schemes without making any changes to the MAC layer protocol. In [2] the authors focus on multiple misbehavior policies in the wireless environment and places emphasis on detection of backoff misbehavior. They propose a sequence of conditions on available observations for testing the extent to which MAC protocol parameters have been manipulated. The proposed scheme does not address the scenarios that include intelligent adaptive cheaters or collaborating misbehaving nodes. The authors in [5], [6] address the detection of an adaptive intelligent attacker by casting the problem of misbehavior detection within the minimax robust detection framework. They optimize the system's performance for the worst-case instance of uncertainty by identifying the least favorable operating point of a system and derive the strategy that optimizes the system's performance when operating at that point. System performance is measured in terms of number of required observation samples to derive a decision (detection delay).

However, DOMINO and SPRT were presented independently, without direct comparison or performance analysis. Additionally, both approaches evaluate the detection scheme performance under unrealistic conditions for continuous monitoring, such as probability of false alarm being equal to 0.01, which in our simulations results in roughly 700 false alarms per minute (in saturation conditions), a rate that is unacceptable in any real-life implementation. Our work contributes to the current literature by: (i) deriving a new pmf for

<sup>1</sup>Research supported by the U.S. Army Research Office under CIP URI grant No. DAAD19-01-1-0494 and by the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011.

<sup>2</sup>Now with the University of California, Berkeley.

the worst case attack using an SPRT-based detection scheme, (ii) providing new performance metrics that address the large number of alarms in the evaluation of previous proposals, (iii) providing a complete analytical model of DOMINO in order to obtain a theoretical comparison to SPRT-based tests and (iv) proposing an improvement to DOMINO based on the CUSUM test.

The rest of the paper is organized as follows. Sect. II outlines the general setup of the problem. In Sect. III we propose a minimax robust detection model and derive an expression for the worst-case attack in discrete time. In Sect. IV we provide extensive analysis of DOMINO, followed by the theoretical comparison of two algorithms in Sect. V. Motivated by the main idea of DOMINO, we offer a simple extension to the algorithm that significantly improves its performance in Sect. VI. In Sect. VII we present the experimental performance comparison of all algorithms. Finally, Sect. VIII concludes our study. In subsequent sections, the terms “attacker” and “adversary” will be used interchangeably with the same meaning.

## II. PROBLEM DESCRIPTION AND ASSUMPTIONS

Assume each station generates a sequence of random backoffs  $X_1, X_2, \dots, X_i$  in order to access the channel. The backoff values of each legitimate protocol participant are then distributed according to the probability mass function (pmf)  $p_0(x_1, x_2, \dots, x_i)$  (specified by the MAC layer protocol). Furthermore, the pmf of the misbehaving participants is unknown to the system and is denoted with  $p_1(x_1, x_2, \dots, x_i)$ .

We assume that a detection agent (e.g., the access point) monitors and collects the backoff values of a given station. It is important to note that observations are not perfect and can be hindered by concurrent transmissions or external sources of noise. It is impossible for a passive monitoring agent to know the internal exponential backoff stage of a given monitored station due to collisions, or to the fact that a station might not have anything to transmit. Furthermore, in practical applications the number of false alarms in anomaly detection schemes is very high. Consequently, instead of building a “normal” profile of network operation with anomaly detection schemes, we utilize specification based detection. In our setup we identify “normal” (i.e., a behavior consistent with the 802.11 specification) profile of a backlogged station in the IEEE 802.11 without any competing nodes, and notice that its backoff process  $X_1, X_2, \dots, X_i$  can be characterized with pmf  $p_0(x_i) = 1/(W + 1)$  for  $x_i \in \{0, 1, \dots, W\}$  and zero otherwise. We claim that this assumption minimizes the probability of false alarms due to imperfect observations. At the same time, a safe upper bound on the amount of damaging effects a misbehaving station can cause to the network is maintained.

Although our theoretical results utilize the above expression for  $p_0$ , the experimental setting utilizes the original implementation of the IEEE 802.11 MAC. In this case, the detection agent needs to deal with observed values of  $x_i$  larger than  $W$ , which can be due to collisions or due to the exponential

backoff specification in the IEEE 802.11. We further discuss this issue in Sect. VII.

## III. SEQUENTIAL PROBABILITY RATIO TEST (SPRT)

A monitoring station observing the sequence of backoffs  $X_1, X_2, \dots, X_N$  will have to determine how many samples ( $N$ ) it is going to observe before making a decision. It is therefore clear that two quantities are involved in decision making: a stopping time  $N$  and a decision rule  $d_N$  which at the time of stopping decides between hypotheses  $H_0$  (legitimate behavior) and  $H_1$  (misbehavior). We denote the above combination with  $D=(N, d_N)$ .

In order to proceed with our analysis we first define the properties we want our detector to satisfy. Intuitively, we want to minimize the probability of false alarms  $\mathbb{P}_0[d_N = 1]$  and the probability of deciding that a misbehaving node is acting normally  $\mathbb{P}_1[d_N = 0]$  (missed detection). Additionally, we want to minimize the average number of samples we collect from a misbehaving station  $\mathbb{E}_1[N]$  before calling the decision function. It is now easy to observe that  $\mathbb{E}_1[N]$ ,  $\mathbb{P}_0[d_N = 1]$  and  $\mathbb{P}_1[d_N = 0]$  form a multi-criteria optimization problem. However, not all of the above quantities can be optimized at the same time. Therefore, a natural approach is to define the accuracy of each decision a priori and minimize the number of samples collected:

$$\inf_{D \in \mathcal{T}_{a,b}} \mathbb{E}_1[N] \quad (1)$$

where

$$\mathcal{T}_{a,b} = \{(N, d_N) : \mathbb{P}_0[d_N = 1] \leq a \text{ and } \mathbb{P}_1[d_N = 0] \leq b\}$$

The solution  $D^*$  (optimality is assured when the data is i.i.d. in both classes) to the above problem is the SPRT [5]:

$$N = \inf_n S_n \in [L, U] \text{ and } d_N = \begin{cases} 1 & \text{if } S_N \geq U \\ 0 & \text{if } S_N \leq L, \end{cases}$$

where

$$S_n = \ln \frac{p_1(x_1, \dots, x_n)}{p_0(x_1, \dots, x_n)} \quad (2)$$

and where  $L \approx \ln \frac{b}{1-a}$  and  $U \approx \ln \frac{1-b}{a}$ . Furthermore, by Wald’s identity:

$$\mathbb{E}_j[N] = \frac{\mathbb{E}_j[S_N]}{\mathbb{E}_j \left[ \ln \frac{p_1(x)}{p_0(x)} \right]} = \frac{\mathbb{E}_j[S_N]}{\sum_{x=0}^W p_j(x) \ln \frac{p_1(x)}{p_0(x)}} \quad (3)$$

with  $\mathbb{E}_1[S_N] = Lb + U(1 - b)$  and  $\mathbb{E}_0[S_N] = L(1 - a) + Ua$ , where the coefficients  $j = 0, 1$  in Eq.(3) correspond to legitimate and adversarial behavior respectively.

### A. Adversary Model

In this section we set a theoretical framework to address the discrete time nature of the MAC layer protocol. Due to the different nature of the problem, the relations derived in [5], [6] no longer hold and a new pmf  $p_1^*$  that maximizes the performance of the adversary is derived.

We assume the adversary has full control over the probability mass function  $p_1$  and the backoff values it generates. In addition to that we assume that the adversary is intelligent, i.e. the adversary knows everything the detection agent knows and can infer the same conclusions as the detection agent.

Now before stating the objective of the adversary we require the following result.

*Lemma 1:* The probability that the adversary accesses the channel before any other terminal when competing with  $n$  neighboring (honest) terminals for channel access in saturation condition is:

$$\Pr[\text{Access}] \equiv P_A = \frac{1}{1 + n \frac{\mathbb{E}_1[X]}{\mathbb{E}_0[X]}} \quad (4)$$

Note that when  $\mathbb{E}_1[X] = \mathbb{E}_0[X]$  the probability of access is equal for all  $n + 1$  competing nodes (including the adversary), i.e., all of them will have access probability equal to  $\frac{1}{n+1}$ . We omit the proof of this result and refer the reader to [6] for the detailed derivation.

Since we want to prevent the misbehaving station from stealing bandwidth unfairly from the contending honest nodes, we consider worth of detection any adversarial strategy that causes enough damage to the network, where “damage” is quantified by a parameter  $G$  in the following relation:  $P_A \geq G$ . The goal of the adversary is therefore to find a strategy  $p_1$  such that  $P_A \geq G$  while minimizing the probability of detection.

By solving  $P_A \geq G$  for  $\mathbb{E}_1[X]$  we obtain:  $\mathbb{E}_1[X] \leq g\mathbb{E}_0[X]$ , where  $g = \frac{1-G}{nG}$ . Notice that for  $G \in (\frac{1}{1+n}, 1)$ ,  $g \in (0, 1)$ , so  $g = 0$  corresponds to complete misbehavior and  $g = 1$  correspond to legitimate behavior. Therefore, for any given  $g$ ,  $p_1$  must belong to the following class of feasible probability mass functions:

$$\mathcal{A}_g \equiv \left\{ q : \sum_{x=0}^W q(x) = 1 \text{ and } \sum_{x=0}^W xq(x) \leq g\mathbb{E}_0[X] \right\} \quad (5)$$

Knowing  $g$ , the objective of the attacker is to maximize the amount of time it can misbehave without being detected. Assuming that the adversary has full knowledge of the employed detection test, it attempts to find the access strategy (with pmf  $p_1$ ) that maximizes the expected duration of misbehavior before an alarm is fired. By looking at equation Eq.(3), the attacker thus needs to minimize the following objective function

$$\min_{p_1 \in \mathcal{A}_g} \sum_{x=0}^W p_1(x) \ln \frac{p_1(x)}{p_0(x)} \quad (6)$$

*Theorem 2:* The pmf  $p_1^*$  that minimizes Eq.(6) is:

$$p_1^*(x) = \begin{cases} \frac{r^x(r^{-1}-1)}{r^{-1}-r^W} & \text{for } x \in \{0, 1, \dots, W\} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where  $r$  is the solution to the following equation:

$$\frac{Wr^W - r^{-1}(Wr^W + r^W - 1)}{(r^{-1} - 1)(r^{-1} - r^W)} = g \frac{W}{2} \quad (8)$$

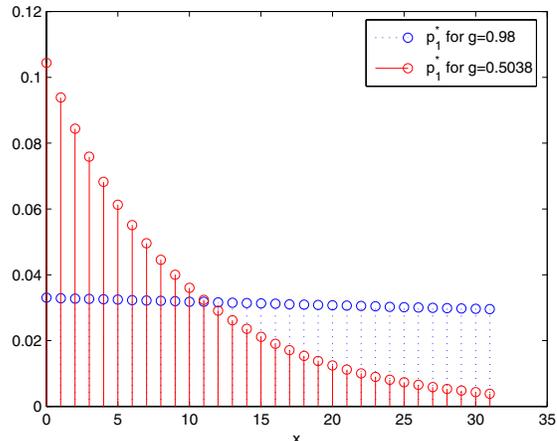


Fig. 1. Form of the least favorable pmf  $p_1^*$  for two different values of  $g$ . When  $g$  approaches 1,  $p_1^*$  approaches  $p_0$ . As  $g$  decreases, more mass of  $p_1^*$  concentrated towards the smaller backoff values.

*Proof:* Notice first that the objective function is convex in  $p_1$ . We let  $q^\epsilon(x) = p_1^*(x) + \epsilon h(x)$  and construct the Lagrangian of the objective function and the constraints

$$\sum_{x=0}^W q^\epsilon(x) \ln \frac{q^\epsilon(x)}{p_0(x)} + \mu_1 \left( \sum_{x=0}^W q^\epsilon(x) - 1 \right) + \mu_2 \left( \sum_{x=0}^W xq^\epsilon(x) - g\mathbb{E}_0[X] \right) \quad (9)$$

By taking the derivative with respect to  $\epsilon$  and equating this quantity to zero for all possible sequences  $h(x)$ , we find that the optimal  $p_1^*$  has to be of the form:

$$p_1^*(x) = p_0(x)e^{-\mu_2 x - \mu_0} \quad (10)$$

where  $\mu_0 = \mu_1 + 1$ . In order to obtain the values of the Lagrange multipliers  $\mu_0$  and  $\mu_2$  we utilize the fact that  $p_0(x) = \frac{1}{W+1}$ . Additionally, we utilize the constraints in  $\mathcal{A}_g$ . The first constraint states that  $p_1^*$  must be a pmf and therefore by setting Eq.(10) equal to one and solving for  $\mu_0$  we have

$$\mu_0 = \ln \sum_{x=0}^W p_0(x)r^x = \ln \frac{1}{W+1} \frac{r - r^{W+1}}{r - 1} \quad (11)$$

where  $r = e^{-\mu_2}$ . Replacing this solution in Eq.(10) we get

$$p_1^*(x) = \frac{r^x(r^{-1} - 1)}{r^{-1} - r^W} \quad (12)$$

The second constraint in  $\mathcal{A}_g$  must be satisfied with equality and is therefore rewritten in terms of Eq.(12) as

$$\frac{r^{-1} - 1}{r^{-1} - r^W} \sum_{x=0}^W xr^x = g\mathbb{E}_0[X] \quad (13)$$

from where Eq.(8) follows. ■

Fig. 1 illustrates the optimal distribution  $p_1^*$  for two values of the parameter  $g$ .

### B. SPRT Optimality for any Adversary in $\mathcal{A}_g$

Let  $\Phi(D, p_1) = \mathbb{E}_1[N]$ . We notice that the above solution was obtained in the form

$$\max_{p_1 \in \mathcal{A}_g} \min_{D \in \mathcal{T}_{a,b}} \Phi(D, p_1) \quad (14)$$

That is, we first minimized  $\Phi(D, p_1)$  with the SPRT (minimization for any given  $p_1$ ) and then found the  $p_1$  that maximized  $\Phi(\text{SPRT}, p_1)$ . However this assumes a non-adaptive adversary, since the SPRT was derived assuming a given  $p_1$ . In a real scenario we can expect the adversary to select its strategy *after* the detection algorithm has been selected; that is, the problem we are interested in solving is:

$$\min_{D \in \mathcal{T}_{a,b}} \max_{p_1 \in \mathcal{A}_g} \Phi(D, p_1) \quad (15)$$

Fortunately, our solution also satisfies this optimization problem since it forms a saddle point equilibrium:

*Theorem 3:* For every  $D \in \mathcal{T}_{a,b}$  and every  $p_1 \in \mathcal{A}_g$

$$\Phi(D^*, p_1) \leq \Phi(D^*, p_1^*) \leq \Phi(D, p_1^*) \quad (16)$$

We omit the proof of the theorem since its derivation follows a reasoning similar to the one in [6]. As a consequence of this theorem, there is no incentive for deviation from  $(D^*, p_1^*)$  for any of the players (the detection agent or the misbehaving node).

### C. Evaluation of Repeated SPRT

The original setup of SPRT-based misbehavior detection proposed in [5] was better suited for on-demand monitoring of suspicious nodes (e.g., when a higher layer monitoring agent requests the SPRT to monitor a given node because it is behaving suspiciously, and once it reaches a decision it stops monitoring) and was not implemented as a repeated test.

On the other hand, the configuration of DOMINO is suited for continuous monitoring of neighboring nodes. In order to obtain fair performance comparison of both tests, a repeated SPRT algorithm is implemented: whenever  $d_N = 0$ , the SPRT restarts with  $S_0 = 0$ . This setup allows a detection agent to detect misbehavior for both short and long-term attacks. The major problem that arises from this setup is that continuous monitoring can raise a large number of false alarms if the parameters of the test are not chosen appropriately.

This section proposes a new evaluation metric for continuous monitoring of misbehaving nodes. We believe that the performance of the detection algorithms is appropriately captured by employing the expected time before detection  $\mathbb{E}[T_D]$  and the average time between false alarms  $\mathbb{E}[T_{FA}]$  as the evaluation parameters.

The above quantities are straightforward to compute for the SPRT. Namely, each time the SPRT stops the decision function can be modeled as a Bernoulli trial with parameters  $a$  and  $1-b$ ; the waiting time until the first success is then a geometric random variable. Therefore:

$$\mathbb{E}[T_{FA}] = \frac{\mathbb{E}_0[N]}{a} \quad \text{and} \quad \mathbb{E}[T_D] = \frac{\mathbb{E}_1[N]}{1-b} \quad (17)$$

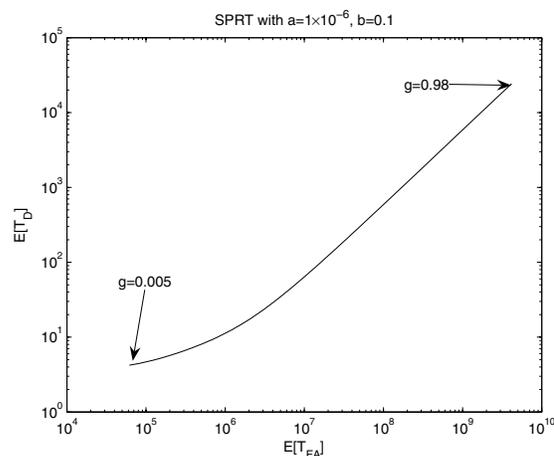


Fig. 2. Tradeoff curve between the expected number of samples for a false alarm  $E[T_{FA}]$  and the expected number of samples for detection  $E[T_D]$ . For fixed  $a$  and  $b$ , as  $g$  increases (low intensity of the attack) the time to detection or to false alarms increases exponentially.

Fig. 2 illustrates the tradeoff between these variables for different values of the parameter  $g$ . It is important to note that the chosen values of the parameter  $a$  in Fig. 2 are small. We claim that this represents an accurate estimate of the false alarm rates that need to be satisfied in actual anomaly detection systems [7], [8], a fact that was not taken into account in the evaluation of previously proposed systems.

### IV. PERFORMANCE ANALYSIS OF DOMINO

We now present the general outline of DOMINO [2]. The first step of the algorithm is based on computation of the average value of backoff observations:  $X_{ac} = \sum_{i=1}^m X_i/m$ . In the next step, the averaged value is compared to the given reference backoff value:  $X_{ac} < \gamma B$ , where the parameter  $\gamma$  ( $0 < \gamma < 1$ ) is a threshold that controls the tradeoff between the false alarm rate and missed detections. The algorithm utilizes the variable `cheat_count` which stores the number of times the average backoff exceeds the threshold  $\gamma B$ . DOMINO raises an alarm after the threshold is exceeded more than  $K$  times. A forgetting factor is considered for `cheat_count` if the monitored station behaves normally in the next monitoring period. That is, the node is partially forgiven: `cheat_count=cheat_count-1` (as long as `cheat_count` remains greater than zero).

More specifically, let `condition` be defined as  $\frac{1}{m} \sum_{i=1}^m X_i \leq \gamma B$  and let the algorithm be initialized with `cheat_count = 0`. After collecting  $m$  samples, the following routine is executed:

```

if condition
    cheat_count = cheat_count + 1
    if cheat_count > K
        raise alarm
    end
elseif cheat_count > 0
    cheat_count = cheat_count - 1
    
```

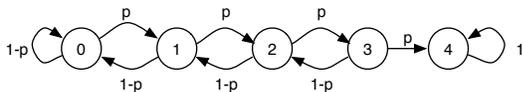


Fig. 3. For  $K=3$ , the state of the variable `cheat_count` can be represented as a Markov chain with five states. When `cheat_count` reaches the final state (4 in this case) DOMINO raises an alarm.

end

It is now easy to observe that DOMINO is a sequential test, with  $N = m * N_t$ , where  $N_t$  represents the number of steps `cheat_count` takes to exceed  $K$  and  $d_N = 1$  every time the test stops. Therefore we can evaluate DOMINO and SPRT with the same performance metrics. However, unlike SPRT where  $a$  controls the number of false alarms and  $b$  controls the detection rate, the parameters  $m$ ,  $\gamma$  and  $K$  in DOMINO are difficult to tune because there has not been any analysis of their performance.

In order to provide an analytical model for the performance of DOMINO, we proceed in the following two steps:

- 1) We first compute  $p := \Pr \left[ \frac{1}{m} \sum_{i=1}^m X_i \leq \gamma B \right]$
- 2) Secondly, we define a Markov chain with transition probabilities  $p$  and  $1-p$ . The absorbing state represents the case when `cheat_count`  $> K$ . A Markov chain for  $K = 3$  is shown in Fig. 3.

We can now write

$$p = p^j = \mathbb{P}_j \left[ \frac{1}{m} \sum_{i=1}^m X_i \leq \gamma B \right], j \in 0, 1$$

where  $j = 0$  corresponds to the scenario where the samples  $X_i$  are generated by a legitimate station  $p_0(x)$  and  $j = 1$  corresponds to the samples being generated by  $p_1^*(x)$ . In the remainder of this section we assume  $B = \mathbb{E}_0[X_i] = \frac{W}{2}$ .

We now derive the expression for  $p$  for the case of a legitimate monitored node. Following the reasoning from Sect. II, we assume that each  $X_i$  is uniformly distributed on  $\{0, 1, \dots, W\}$ . It is important to note that this analysis provides a lower bound on the probability of false alarms when the minimum contention window of size  $W + 1$  is assumed. Using the definition of  $p$  we derive the following expression:

$$\begin{aligned} p &= \mathbb{P}_0 \left[ \sum_{i=1}^m X_i \leq m\gamma B \right] \\ &= \sum_{k=0}^{\lfloor m\gamma B \rfloor} \mathbb{P}_0 \left[ \sum_{i=1}^m X_i = k \right] \\ &= \sum_{k=0}^{\lfloor m\gamma B \rfloor} \sum_{\{(x_1, \dots, x_m) : \sum_{i=1}^m x_i = k\}} \frac{1}{(W+1)^m} \end{aligned} \quad (18)$$

where the last equality follows from the fact that the  $X_i$ 's are i.i.d. with pmf  $p_0(x_i) = \frac{1}{W+1}$  for all  $x_i \in \{0, 1, \dots, W\}$ .

The number of ways that  $m$  integers can sum up to  $k$  is  $\binom{m+k-1}{k}$  and  $\sum_{k=0}^L \binom{m+k-1}{k} = \binom{m+L}{L}$ . However, an additional constraint is imposed by the fact that  $X_i$  can only take values up to  $W$ , which is in general smaller than  $k$ , and thus the above combinatorial formula cannot be applied. Furthermore, a direct computation of the number of ways  $x_i$  bounded integers sum up to  $k$  is very expensive. As an example, let  $W+1 = 32 = 2^5$  and  $m = 10$ . A direct summation needed for calculation of  $p$  yields at least  $2^{50}$  iterations.

Fortunately, an efficient alternative way for computing  $\mathbb{P}_0 \left[ \sum_{i=1}^m X_i = k \right]$  exists. We first define  $Y := \sum_{i=1}^m X_i$ . Therefore the moment generating function of  $Y$ ,  $M_Y(s) = M_X(s)^m$  can be computed as follows:

$$\begin{aligned} M_Y(s) &= \frac{1}{(W+1)^m} (1 + e^s + \dots + e^W)^m \\ &= \frac{1}{(W+1)^m} \times \\ &\quad \sum_{\left\{ \begin{array}{l} k_0, \dots, k_W : \\ \sum k_i = m \end{array} \right\}} \binom{m}{k_0; \dots; k_W} 1^{k_0} e^{sk_1} \dots e^{sWk_W} \end{aligned}$$

where  $\binom{m}{k_0; k_2; \dots; k_W}$  is the multinomial coefficient.

By comparing terms with the transform of  $M_Y(s)$  we observe that  $\Pr[Y = k]$  is the coefficient that corresponds to the term  $e^{ks}$  in Eq.(19). This result can be used for the efficient computation of  $p$  by using Eq.(18).

Alternatively, we can approximate the computation of  $p$  for large values of  $m$ . The approximation arises from the fact that as  $m$  increases,  $Y$  converges to a Gaussian random variable, by the Central Limit Theorem. Thus,

$$p = \Pr[Y \leq m\gamma B] \approx \Phi(z)$$

where

$$z = \frac{m\gamma B - m\frac{W}{2}}{\sqrt{(W)(W+2)m/12}}$$

and  $\Phi(z)$  is the error function.

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-x^2/2} dx$$

Fig. 4 illustrates the exact and approximate calculation of  $p$  as a function of  $m$ , for  $\gamma = 0.9$  and  $W+1 = 32$ . This shows the accuracy of the above approximation for both small and large values of  $m$ .

The computation of  $p = p^1$  follows the same steps (although the moment generating function cannot be easily expressed in analytical form, it is still computationally tractable) and its derivation is therefore omitted.

#### A. Expected Time to Absorption in the Markov Chain

We now derive the expression for expected time to absorption for a Markov Chain with  $K+1$  states. Let  $\mu_i$  be the expected number of transitions until absorption, given

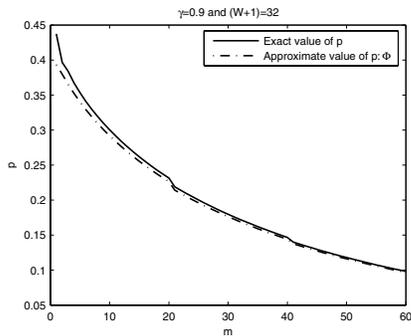


Fig. 4. Exact and approximate values of  $p$  as a function of  $m$ .

that the process starts at state  $i$ . In order to compute the stopping times  $\mathbb{E}[T_D]$  and  $\mathbb{E}[T_{FA}]$ , it is necessary to find the expected time to absorption starting from state zero,  $\mu_0$ . Therefore,  $\mathbb{E}[T_D] = m \times \mu_0$  (computed under  $p = p^1$ ) and  $\mathbb{E}[T_{FA}] = m \times \mu_0$  (computed under  $p = p^0$ ).

The expected times to absorption,  $\mu_0, \mu_1, \dots, \mu_{K+1}$  represent the unique solutions to the equations

$$\begin{aligned} \mu_{K+1} &= 0 \\ \mu_i &= 1 + \sum_{j=0}^{K+1} p_{ij} \mu_j \text{ for } i \in \{0, 1, \dots, K\} \end{aligned}$$

where  $p_{ij}$  is the transition probability from state  $i$  to state  $j$ . For any  $K$ , the equations can be represented in matrix form:

$$\begin{bmatrix} -p & p & 0 & \dots & 0 \\ 1-p & -1 & p & 0 & 0 \\ 0 & 1-p & -1 & p & 0 \\ & & \vdots & & \\ 0 & \dots & 0 & 1-p & -1 \end{bmatrix} \begin{bmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \vdots \\ \mu_K \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}$$

For example, solving the above equations for  $\mu_0$  with  $K = 3$ , the following expression is derived

$$\mu_0 = \frac{1 - p + 2p^2 + 2p^3}{p^4}$$

## V. THEORETICAL COMPARISON

In this section we compare the tradeoff curves between  $\mathbb{E}[T_D]$  and  $\mathbb{E}[T_{FA}]$  for both algorithms. For the sake of concreteness we compare both algorithms for an attacker with  $g = 0.5$ . Similar results were observed for other values of  $g$ .

For SPRT we set  $b = 0.1$  arbitrarily and vary  $a$  from  $10^{-1/2}$  up to  $10^{-10}$  (motivated by the realistic low false alarm rate required by actual intrusion detection systems [7]). Due to the fact that in DOMINO it is not clear how the parameters  $m$ ,  $K$  and  $\gamma$  affect our metrics, we vary all the available parameters in order to obtain a fair comparison. Fig. 5 illustrates the performance of DOMINO for  $K = 3$  (the default threshold used in [2]). Each curve for  $\gamma$  has  $m$  ranging between 1 and 60. Under these settings, we conclude that the best performance of DOMINO is obtained for  $\gamma = 0.7$ , regardless of  $m$ . Therefore,

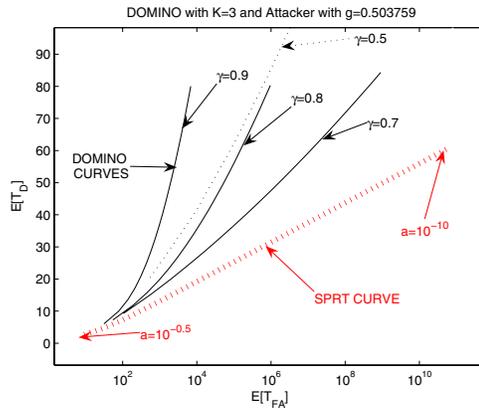


Fig. 5. DOMINO performance for  $K = 3$ ,  $m$  ranges from 1 to 60.  $\gamma$  is shown explicitly. As  $\gamma$  tends to either 0 or 1, the performance of DOMINO decreases. The SPRT outperforms DOMINO regardless of  $\gamma$  and  $m$ .

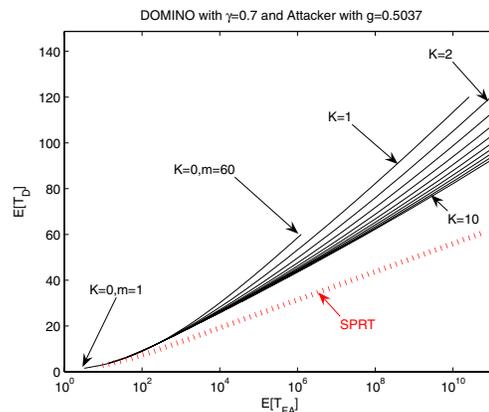


Fig. 6. DOMINO performance for various thresholds  $K$ ,  $\gamma = 0.7$  and  $m$  in the range from 1 to 60. The performance of DOMINO decreases with increase of  $m$ . For fixed  $\gamma$ , the SPRT outperforms DOMINO for all values of parameters  $K$  and  $m$ .

this value of  $\gamma$  is adopted as an optimal threshold in further experiments.

Fig. 6 represents the evaluation of DOMINO for  $\gamma = 0.7$  with varying threshold  $K$ . For each value of  $K$ ,  $m$  ranges from 1 to 60. In this figure, however, we noticed that with the increase of  $K$ , the point with  $m = 1$  forms a performance curve that is better than any other point with  $m > 1$ .

Consequently, Fig. 7 represents the best possible performance for DOMINO; that is, we let  $m = 1$  and change  $K$  from one up to one hundred. We again test different  $\gamma$  values for this configuration, and conclude that the best  $\gamma$  is still close to the optimal value of 0.7 derived from experiments in Fig. 5. However, even with the optimal setting, DOMINO is outperformed by the SPRT.

Due to the fact that  $m$  was not considered as a tuning parameter in the original DOMINO algorithm ( $m$  was random in [2], depending only on the number of observations in a given unit of time,) we refer to the new configuration with  $m = 1$  as O-DOMINO, for Optimized-DOMINO, since

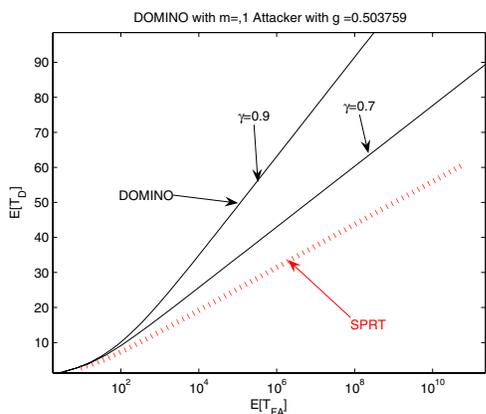


Fig. 7. The best possible performance of DOMINO is when  $m = 1$  and  $K$  changes in order to accommodate for the desired level of false alarms. The best  $\gamma$  must be chosen independently.

according to our analysis, any other value of  $m$  is suboptimal. Notice that O-DOMINO can be expressed as:

$$K_i = (K_{i-1} + (1_{X_i \leq \gamma B} - 1_{X_i > \gamma B}))^+ \quad (19)$$

where  $1_R$  is the indicator random variable for event  $R$ .

## VI. NONPARAMETRIC CUSUM STATISTIC

As concluded in the previous section, DOMINO exhibits suboptimal performance for every possible configuration of its parameters. However, the original idea of DOMINO is very intuitive and simple; it compares the observed backoff of the monitored nodes with the expected backoff of honest nodes within a given period of time.

In this section we extend the above idea by proposing a test that exhibits better performance than O-DOMINO, while still preserving its simplicity.

Inspired by the notion of nonparametric statistics for change detection by looking at Eq.(19), we adapt the nonparametric cumulative sum (CUSUM) statistic and apply it in our analysis. Nonparametric CUSUM is initialized with  $Y_0 = 0$  and updates its value as follows:

$$Y_i = (Y_{i-1} + (\gamma B - X_i))^+ \quad (20)$$

An alarm is fired whenever  $Y_i > c$ .

Assuming  $\mathbb{E}_0[X] > \gamma B$  and  $\mathbb{E}_1[X] < \gamma B$  (i. e. the expected backoff value of an honest node is always larger than a given threshold and vice versa), the properties of the CUSUM test with regard to the expected false alarm and detection times can be captured by the following theorem.

*Theorem 4:* The probability of firing a false alarm decreases exponentially with  $c$ . Formally, as  $c \rightarrow \infty$

$$\sup_i |\ln(\mathbb{P}_0[Y_i > c])| = \mathcal{O}(c) \quad (21)$$

Furthermore, the delay in detection increases only linearly with  $c$ . Formally, as  $c \rightarrow \infty$

$$T_D = \frac{c}{\gamma B - \mathbb{E}_1[X]} \quad (22)$$

The proof is a straightforward extension of the case originally considered in [9].

It is easy to observe that the CUSUM test is similar to DOMINO, with  $c$  being equivalent to the upper threshold  $K$  in DOMINO and the statistic  $y$  in CUSUM being equivalent to the variable `cheat_count` in DOMINO when  $m = 1$ .

The main difference between DOMINO when  $m = 1$  and the CUSUM statistic is that every time there is a “suspicious event” (i.e., whenever  $x_i \leq \gamma B$ ), `cheat_count` is increased by one, whereas in CUSUM  $y_i$  is increased by an amount proportional to the level of suspected misbehavior. Similarly, when  $x_i > \gamma B$ , `cheat_count` is decreased only by one (or maintained as zero), while the decrease in  $y_i$  is proportional to the amount of time the station did not attempt to access the channel.

## VII. EXPERIMENTAL RESULTS

We now proceed to the experimental evaluation of the analyzed detection schemes. The backoff distribution of the optimal attacker from Eq. (7) was implemented in the network simulator Opnet and tests were performed for various levels of false alarms. We note that the simulations were performed with honest nodes that followed the standard IEEE 802.11 access protocol (with exponential backoff). Therefore the detection agent was implemented such that any observed backoff value greater than  $W$  ( $X_i > W$ ) was scaled down to  $W$ . Our experiments show that this decision works well in practice.

The results presented in this work correspond to the scenario consisting of two legitimate and one selfish node competing for channel access. It is important to mention that the resulting performance comparison of DOMINO, CUSUM and SPRT does not change for any number of competing nodes. SPRT always exhibits the best performance.

In order to demonstrate the performance of all detection schemes, we choose to present the results for the scenario where the attacker attempts to access channel for 60% of the time (as opposed to 33% if it was behaving legitimately). This corresponds to  $g = 1/3$ . The backlogged environment in Opnet was created by employing a relatively high packet arrival rate per unit of time: the results were collected for the exponential(0.01) packet arrival rate and the packet size was 2048 bytes. The results for both legitimate and malicious behavior were collected over a fixed period of 100s.

The evaluation was performed as a tradeoff between the average time to detection and the average time to false alarm. It is important to mention that the theoretical performance evaluation of both DOMINO and SPRT was measured in number of samples. Here, however, we take advantage of the experimental setup and measure time in number of seconds, a quantity that is more meaningful and intuitive in practice.

The first step in our experimental evaluation is to test the optimality of the SPRT, or more generally, the claim that O-DOMINO performs better than the original DOMINO, that the nonparametric CUSUM statistic performs better than O-DOMINO and that the SPRT performs better than all of the above.

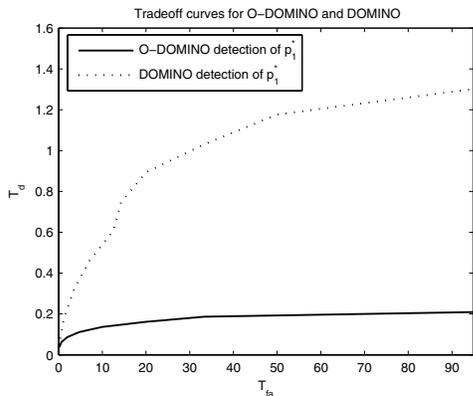


Fig. 8. Tradeoff curves for the original DOMINO algorithm with  $K = 3$ ,  $\gamma = 0.9$  and different values of  $m$  vs. O-DOMINO with  $\gamma = 0.7$  and different values of  $K$

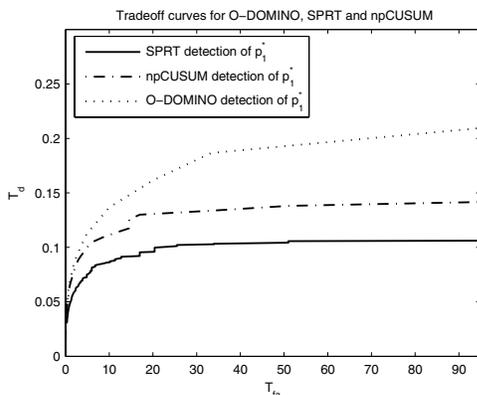


Fig. 9. Tradeoff curves for SPRT with  $b = 0.1$  and different values of  $a$  vs. nonparametric CUSUM and O-DOMINO with  $\gamma = 0.7$  and different values of  $K$

The original DOMINO algorithm, as suggested in [2], assumes  $K = 3$  and  $\gamma = 0.9$ . As we have already mentioned the original DOMINO takes averages over a fixed unit of time, so the number  $m$  of observed samples for taking the average is different for every computed average backoff. Therefore in Fig. 8 we compare DOMINO with  $K = 3$ ,  $\gamma = 0.9$  and  $m$  varying (representing the fact that the performance of the original DOMINO algorithm can be any point on that tradeoff curve, depending on the number of samples observed  $m$ ), vs. O-DOMINO with  $\gamma = 0.7$  (the suggested algorithm according to our analysis). This performance was also observed for other configurations of DOMINO. In particular we noticed that as long as DOMINO takes averages of the samples, i.e., as long as  $m > 1$ , DOMINO is outperformed by O-DOMINO, even if they assume the same  $\gamma$ . Our experiments also suggest that having  $\gamma$  close to 0.7 is the optimal setting. Notice that this is true in our theoretical analysis (done with  $g=0.503$ ) and in our experimental analysis, where  $p_1^*$  was obtained for  $g = 1/3$ .

We now test how our three proposed algorithms compare to

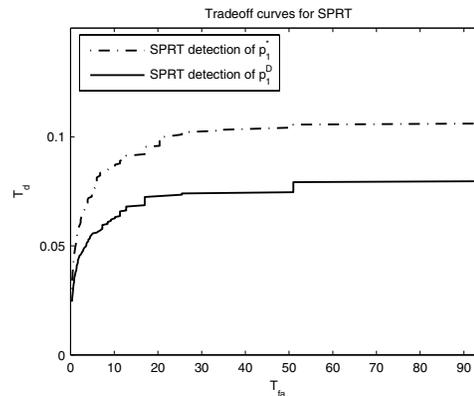


Fig. 10. Tradeoff curves for SPRT with  $b = 0.1$  and different values of  $a$ . One curve shows its performance when detecting an adversary that chooses  $p_1^D$  and the other is the performance when detecting an adversary that chooses  $p_1^*$

each other. Fig. 9 provides experimental evidence confirming our predictions. In general, since the SPRT is optimal, it performs better than the nonparametric CUSUM statistic, and since nonparametric CUSUM takes into account the level of the misbehavior (or normal behavior) for each sample, then it outperforms the restricted addition and subtraction in O-DOMINO.

We have therefore shown how SPRT is the best test when the adversary selects  $p_1^*$ . We now show that if the adversary deviates from  $p_1^*$  it will be detected faster. In order to come up with another strategy  $p_1$  in  $\mathcal{A}_{1/3}$  we decided to use the attack distribution considered in [2], mainly a uniform distribution with support between 0 and  $aW$ , where  $a$  denotes the misbehavior coefficient of the adversary (and  $W$  is the contention window size). We will call this pmf  $p_1^D$ . In order to make a fair comparison, we require  $p_1^D \in \mathcal{A}_{1/3}$ , and thus we set  $a = 1/3$ .

Fig. 10 shows the performance of SPRT when the adversary uses  $p_1^D$  and when it uses  $p_1^*$ . This figure supports the analysis that  $p_1^*$  is the worst possible distribution SPRT can face.

It is also interesting to see that the same phenomenon happens for DOMINO. As can be seen in Fig. 11, an adversary using  $p_1^*$  against DOMINO can misbehave for longer periods of time without being detected than by using  $p_1^D$ . Notice however that we did not derive the optimal adversarial strategy against DOMINO, and therefore there might be another distribution  $p_1^O$  which will yield a better gain to the adversary when compared to using  $p_1^*$  against DOMINO.

Nevertheless,  $p_1^*$  can be argued to be a good adversarial strategy against any detector in the asymptotic observation case, since  $p_1^*$  is in fact minimizing the Kullback-Leibler divergence from the specified pmf  $p_0$ , as can be seen from Eq.(6). The result is that the probability of detection of any algorithm (when the false alarm rate goes to zero) is upper bounded by  $2^{D(p_1||p_0)}$ , where  $D(p||q)$  denotes the Kullback-Leibler divergence between two pmfs [10]. On the other hand we could not find any theoretical motivation for the definition

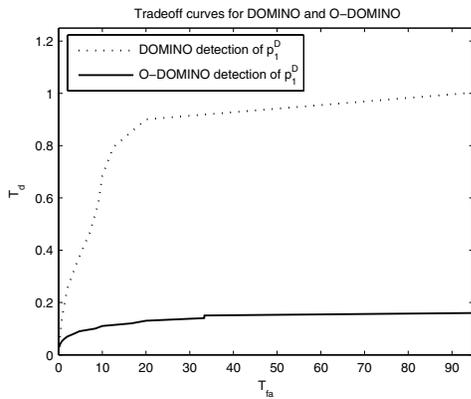


Fig. 11. Tradeoff curves for DOMINO and O-DOMINO with the same parameters as in Fig. 8. However this time instead of detecting an adversary that chooses  $p_1^*$  we measure their performance against an adversary that chooses  $p_1^D$ . When compared to Fig. 8, it is evident that DOMINO and O-DOMINO perform better when the adversary chooses  $p_1^D$ .

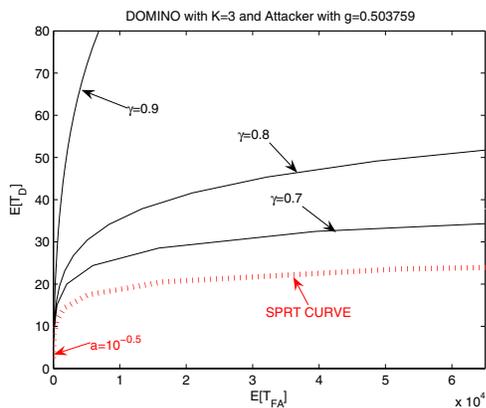


Fig. 12. Comparison between theoretical and experimental results: theoretical analysis with linear x-axis closely resembles the experimental results.

of  $p_1^D$ .

It is also interesting to note how close the theoretical shape of the tradeoff curves is to the actual experimental data. Fig. 12 supports the correctness of our theoretical analysis since if the logarithmic x-axis in the tradeoff curves in Section V is replaced with a linear one, our theoretical curves closely resemble the experimental data.

## VIII. CONCLUSIONS AND FUTURE WORK

In this work, we performed extensive analytical and experimental comparisons of several misbehavior detection schemes in the MAC layer of IEEE 802.11. We confirmed the optimality of the SPRT-based detection schemes and provided analytical explanations of why the other schemes exhibit suboptimal performance when compared to the SPRT. In addition to that, we offered two extensions to DOMINO: O-DOMINO and nonparametric CUSUM. These extensions still preserve the original intuition and simplicity of the algorithm, while significantly improving its performance. Our results

show the value of performing a rigorous formulation of the problem with the help of advanced statistical and game-theoretic techniques, since these techniques can outperform heuristic solutions in very practical scenarios.

It is also important to point out that the SPRT is a *parametric* statistic, while the other algorithms presented in this work are *nonparametric*. Nonparametric statistics are easier to apply since they do not require exact models of the normal or adversarial distributions. They only require knowledge of some of its parameters. DOMINO, O-DOMINO and nonparametric CUSUM, for example only assume knowledge of some *nominal backoff*. This nominal backoff represents the normal backoff expected by honest stations. A parametric statistic, on the other hand, needs a model for the distributions  $p_0$  and  $p_1$ . If it has both models it should in principle perform better than its corresponding nonparametric statistic. In order to obtain the model for  $p_1$ , we used the typical idea of robust statistics: find the least favorable distribution  $p_1$ . It should be interesting to see the comparison of our robust SPRT statistic with other robust versions of sequential parametric statistics, such as the parametric version of CUSUM, or the Shiryaev-Roberts statistic.

Another important aspect that needs to be addressed is the response to an alarm. The effect of this responses should be analyzed with respect to the network performance degradation due to false alarms and the effectiveness in thwarting misbehavior.

## REFERENCES

- [1] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions," in *Proceedings of the USENIX Security Symposium*, Washington D.C., August 2003.
- [2] M. Raya, J.-P. Hubaux, and I. Aad, "DOMINO: A system to detect greedy behavior in IEEE 802.11 hotspots," in *Proceedings of the Second International Conference on Mobile Systems, Applications and Services (MobiSys2004)*, Boston, Massachusetts, June 2004.
- [3] P. Kyasanur and N. Vaidya, "Detection and handling of mac layer misbehavior in wireless networks," in *Proceedings of the International Conference on Dependable Systems and Networks*, June 2003.
- [4] A. A. Cárdenas, S. Radosavac, and J. S. Baras, "Detection and prevention of mac layer misbehavior in ad hoc networks," in *Proceedings of the 2nd ACM workshop on security of ad hoc and sensor networks (SASN 04)*, 2004.
- [5] S. Radosavac, J. S. Baras, and I. Koutsopoulos, "A framework for MAC protocol misbehavior detection in wireless networks," in *Proceedings of the 4th ACM workshop on Wireless Security (WiSe 05)*, 2005, pp. 33–42.
- [6] S. Radosavac, G. V. Moustakides, J. S. Baras, and I. Koutsopoulos, "An analytic framework for modeling and detecting access layer misbehavior in wireless networks," *submitted to ACM Transactions on Information and System Security (TISSEC)*, 2006.
- [7] A. A. Cárdenas, J. S. Baras, and K. Seamon, "A framework for the evaluation of intrusion detection systems," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, Oakland, California, May 2006.
- [8] S. Axelsson, "The base-rate fallacy and its implications for the difficulty of intrusion detection," in *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS '99)*, November 1999, pp. 1–7.
- [9] B. E. Brodsky and B. S. Darkhovsky, *Nonparametric Methods in Change-Point Problems*, ser. Mathematics and Its Applications. Kluwer Academic Publishers, 1993, vol. 243.
- [10] R. E. Blahut, *Principles and Practice of Information Theory*. Addison-Wesley, 1987.