Evaluation of Detection Algorithms for MAC Layer Misbehavior: Theory and Experiments

Alvaro A. Cárdenas, *Member, IEEE*, Svetlana Radosavac, *Member, IEEE*, and John S. Baras, *Fellow, IEEE*

***Abstract*—We revisit 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). Our evaluation is carried out in four steps. We first derive a new analytical formulation of the SPRT that considers access to the wireless medium in discrete time slots. Then, we introduce an 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 the correctness of our theoretical analysis and validating the introduction of the new nonparametric CUSUM statistic.**

***Index Terms*—IEEE 802.11 MAC, SPRT, DOMINO, CUSUM, misbehavior, intrusion detection.**

I. Introduction

MOST 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 and so on).

In this work we derive new analytical bounds for the performance of two previously proposed protocols for detecting random access misbehavior in IEEE 802.11 networks—DOMINO [18], [17] and robust SPRT tests [16], [15]—and show the optimality of SPRT against a worst-case adversary for all configurations of DOMINO. Following the main intuitive idea of DOMINO, we also introduce a nonparametric CUSUM statistic that shares the same basic concepts of DOMINO but gives better performance. Our results are validated by theoretical analysis and experiments.

Manuscript received July 17, 2007; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor E. Knightly. This work was supported by the U.S. Army Research Office under CIP URI Grant 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. An earlier version of this document appeared in IEEE INFOCOM 2007, Anchorage, AK.

A. A. Cárdenas is with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA.

S. Radosavac is with the DoCoMo Communications Laboratories USA, Inc., Palo Alto, CA 94304 USA.

J. S. Baras is with the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA.

Digital Object Identifier 10.1109/TNET.2008.926510

A. Related Work

The current literature for preventing and detecting MAC layer misbehavior can be classified in two: (1) design of new MAC-layer protocols that discourage misbehavior, and (2) detection of misbehaving parties.

The design of MAC-layer protocols to discourage misbehavior is generally done with the help of game-theoretic ideas. The scenario usually includes a set of selfish nodes that want to maximize their access to the medium, and goal of the protocol is to motivate users to achieve a Nash equilibrium (no party will have a motivation to deviate from the protocol) [9], [12], [2], [7], [13]. Because game theoretic protocols assume that all parties are willing to deviate from the protocol (the worst case scenario), the throughput achieved is substantially less than in protocols where the honest majority cooperates with the design.

In protocols where we assume that an honest majority cooperates, we are interested only in detecting the misbehaving parties. The current literature offers two major apporaches: (1) the modification of current protocols to facilitate the detection of misbehavior, and (2) detection without modifying current protocols. The first set of approaches provide solutions based on modification of the current IEEE 802.11 MAC layer protocol. These schemes may assume a trusted receiver—e.g., an access point—that assigns back-off values to other nodes [11], or a negotiation of the backoff value among neighboring nodes [6], [14]. In these protocols it is easy to detect misbehavior because the detection agent knows the back-off time assigned to each party.

The second set of approaches attempt to detect misbehavior without modifying the underlying MAC-layer protocol. This is the most viable solution for widely deployed MAC-layer protocols (such as IEEE 802.11). Detecting misbehavior in IEEE 802.11 is, however, very challenging because each node selects their back-off value independently, and the detection agent cannot determine—with complete certainty—if a series of suspiciously small back-off values by one party was the result of chance, or if the party has deviated from the protocol specification.

In DOMINO [18], [17], the authors focus on multiple misbehavior options in IEEE 802.11, and put emphasis on detection of back-off misbehavior. The detection algorithm computes an estimate of the mean average back-off time, and raises an alarm if this estimate is suspiciously low.

A more technical approach was introduced by Rong *et al.* [19], where the detection algorithm relies on the Sequential Probability Ratio Test (SPRT). The observations of the detection agent are not the back-off times of the stations, but the inter-delivery time distribution. To use the SPRT test, the authors estimate a normal inter-delivery distribution and an

attack inter-delivery distribution. The proposed scheme does not address scenarios that include intelligent adaptive cheaters. In particular, it does not consider the flexibility that an attacker may have when designing its attack distribution.

The *robust* SPRT [15], [16] addresses the detection of an adaptive intelligent attacker by casting the problem of misbehavior detection within the min-max robust detection framework. The key idea is to optimize the performance of the detection algorithm for the worst-case attacker strategy. This process is characterized by identifying the least favorable operating point of the detection algorithm, and by deriving the strategy that optimizes the performance of the detection algorithm when operating in that point. The detection performance is measured in terms of number of required observation samples to derive a decision (detection delay) subject to a constant rate of false alarms.

B. Contributions

DOMINO and (robust) SPRT were presented independently, and without direct comparison or performance analysis. Additionally, both approaches evaluate the detection scheme performance under unrealistic conditions, such as probability of false alarm being equal to 0.01, which in our simulations results in roughly 700 false alarms per minute (uder saturation conditions), a rate that is unacceptable in any real-life implementation. In this work we address these concerns by providing a theoretical and experimental evaluation of these tests.

Our work contributes to the current literature by: (i) deriving a new strategy (in discrete time) for 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. Section II outlines the general setup of the problem. In Section III we propose a min-max robust detection model and derive an expression for the worst-case attack in discrete time. In Section IV we provide extensive analysis of DOMINO, followed by the theoretical comparison of two algorithms in Section V. Motivated by the main idea of DOMINO, we offer a simple extension to the algorithm that significantly improves its performance in Section VI.

In Section VII we present the experimental performance comparison of all algorithms. Finally, Section IX concludes our study. In subsequent sections, the terms "attacker" and "adversary" will be used interchangeably with the same meaning.

II. Problem Description and Assumptions

An adversary has no need to cheat—i.e., misbehave—for accessing the wireless medium when no one else attempts to transmit. Therefore, in order to minimize the probability of detection, an attacker will choose legitimate over selfish behavior when the level of congestion in the network is low. Similarly, the attacker will choose an adaptive selfish strategy in congested environments.

For these reasons we assume a benchmark scenario where all the participants are backlogged—i.e., have packets to send at any given time—in both, our theoretical analysis and experimental evaluations. We assume that the attacker will employ the worst-case misbehavior strategy in this setting, and consequently the detection system can estimate the maximal detection delay. Notice also that the backlogged scenario represents the worst-case scenario with regard to the number of false alarms per unit of time (because the detection algorithm is forced to make a maximum number of decisions per unit of time).

To formalize these assumptions we assume that each station generates a sequence of random back-offs $X_1, X_2, \ldots, X_i$ over a fixed period of time: the back-off values $X_1, X_2, \ldots, X_i$, of each legitimate protocol participant are distributed according to the probability mass function (pmf) $p_0(x_1, x_2, \ldots, x_i)$. The pmf of the misbehaving participants is unknown to the detection algorithm and is denoted as $p_1(x_1, x_2, \ldots, x_i)$, where $X_1, X_2, \ldots, X_i$ represent the sequence of back-off values generated by the misbehaving node over the same period of time.

We assume that a detection agent—e.g., the access point—monitors and collects the back-off values of a given station, and is asked to make a decision based on these observations. The question we face is how to design a good detection scheme based on this information.

In general, detection systems used in computer security can be classified in three approaches: (1) signature-based detection schemes, (2) anomaly detection schemes, and (3) specification-based detection schemes [20]. Signature-based detection scheme is based on the recognition of *attack signatures*. In our case, however, this is not a viable solution since there is no unique signature a misbehaving station will follow when deviating from the MAC protocol. Anomaly detection schemes consist on two phases: in the first phase, the system *learns* the normal behavior of the protocol and creates a model; in the second phase, the observations are compared with the model and flagged as anomalous if they deviate from it. The problem with anomaly detection schemes is that they tend to generate a large number of false alarms: and in general, it is very difficult to learn the normal behavior of a network. Finally, specification-based approaches attempt to capture abnormal behavior—like anomaly detection schemes—but instead of learning the "normal" model, the model is specified manually. This reduces the number of false alarms in practice, since a manual specification tries to capture all possible normal behaviors. We follow this paradigm in our work.

Since the IEEE 802.11 access distribution is known, it should—in principle—be the best manual specification for the normal access pmf p_0. However, the back-off observations seen by the monitoring agent cannot be perfect: not only can they be hindered by concurrent transmissions or external sources of noise, but it is impossible for a passive monitoring agent to know the back-off stage of a given monitored station because of collisions, and because in practice, nodes might not be constantly backlogged. Consequently, 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 back-off process $X_1, X_2, \ldots, X_i$ can be characterized with pdf $p_0(x_i) = 1/(W+1)$ for $x_i \in \{0, 1, \ldots, W\}$ and zero

otherwise. It should be clear that this assumption minimizes the probability of false alarms due to imperfect observations. At the same time, we maintain a safe upper bound on the amount of damaging effects a misbehaving station can cause to the network.

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 back-off specification in IEEE 802.11. We further discuss this issue in Section VII.

III. Sequential Probability Ratio Test (SPRT)

A monitoring station observing the sequence of backoffs $X_1, X_2, \ldots, X_N$ will have to determine how many samples (N) it is going to observe before making a decision (d_N). 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 stopping time, 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 of an efficient detector. Intuitively, we want to minimize the probability of false alarms $\mathbb{P}_0[d_N = 1]$, and also, the probability of deciding that a misbehaving node is acting normally $\mathbb{P}_1[d_N = 0]$ (missed detections). Additionally, each detector should be able to derive the decision as soon as possible; so we would like to minimize the number of samples we collect from a misbehaving station ($\mathbb{E}_1[N]$) before calling the decision function.

Therefore $\mathbb{E}_1[N], \mathbb{P}_0[d_N = 1], \mathbb{P}_1[d_N = 0]$ form a multi-criteria optimization problem. Since not all of the above quantities can be optimized at the same time, a natural approach is to define the accuracy of each decision *a priori* and minimize the number of samples collected:

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

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 $D^\star$ (optimality is assured when the data is i.i.d. in both classes) to the above problem is the SPRT [21]. Let

$$S_n = \ln \frac{p_1(x_1, \ldots, x_n)}{p_0(x_1, \ldots, x_n)} \quad \text{and} \quad N = \inf_n S_n \in [L, U].$$

The SPRT decision rule d_N is defined as

$$d_N = \begin{cases} 1 & \text{if } S_N \geq U \\ 0 & \text{if } S_N \leq L, \end{cases} \tag{2}$$

where $L \approx \ln \frac{b}{1-a}$ and $U \approx \ln \frac{1-b}{a}$.

The performance of the SPRT can be formally analyzed by Wald's identity:

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

where $\mathbb{E}_1[S_N] = Lb + U(1-b)$ and $\mathbb{E}_0[S_N] = L(1-a) + Ua$; furthermore, the coefficients $j = 0, 1$ in (3) correspond to whether our observations are distributed with the legitimate distribution p_0 or the adversarial behavior p_1 (respectively).

A. Adversary Model

In this section we find the least favorable p_1 for the SPRT. We begin by stating our assumptions on the adversary class we consider.

Capabilities of the Adversary: We assume the adversary has full control over the probability mass function p_1 and the back-off values it generates.

Knowledge of the Adversary: We assume the adversary knows everything the detection agent knows and can infer the same conclusions as the detection agent. In other words, we assume there is *no secret information* for the adversary.

Goal of the Adversary: We assume the objective of the adversary is to design p_1 in order to obtain access to the medium with probability P_A, while at the same time, minimizing the probability of being detected.

Theorem 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

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

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). More specifically, all of them will have access probability equal to $\frac{1}{n+1}$.

The proof of the theorem can be found in Appendix I.

Because we want to prevent a misbehaving station from stealing bandwidth unfairly from the contending honest nodes, we consider "worthy" of detection only those adversarial strategies that cause enough "damage" to the network (when $P_A \geq G$), where "damage"—quantified by G—denotes a lower bound on the probability of access by the adversary under saturation conditions. In practice, if the real gain of the adversary P_A is greater than G, then our detection mechanism will detect faster this misbehavior. If P_A is less than G, then we expect that the effect of this type of adversary is not damaging.

An example of this last case is an adversary that never fires an alarm because it selects p_1 such that the detection statistic S_n never reaches the upper bound U. However, under these conditions we know that P_A is asymptotically no different than the probability of access by a legitimate node, and thus there is no need to detect this type of misbehavior.

1) Finding $P_1^\star$: Now we turn our attention to finding the least-favorable distribution $p_1^\star$.

Let $g = \frac{1-G}{nG}$. Solving $P_A \geq G$ for $\mathbb{E}_1[X]$ we obtain

$$\mathbb{E}_1[X] \leq g\mathbb{E}_0[X]. \tag{5}$$

Notice that when $G \in (\frac{1}{1+n}, 1), g \in (0, 1)$, so $g = 0$ corresponds to complete misbehavior and $g = 1$ correspond to legitimate behavior.

Now, for any given g, p_1 must belong to the following class of feasible probability mass functions:

$$\mathcal{A}_g \equiv \left\{ q : \forall x q(x) \geq 0, \sum_{x=0}^{W} q(x) = 1 \text{ and } \sum_{x=0}^{W} x q(x) \leq g \mathbb{E}_0[X] \right\}. \quad (6)$$

The first two constraints guarantee that q is a probability mass function. The last constraint guarantees that q belongs to the class of "dangerous" probability distributions—the ones we are interested in detecting, as previously explained.

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 (p_1) that maximizes the expected duration of misbehavior before an alarm is fired. By looking at (3), we conclude that the attacker needs to minimize the following objective function:

$$\min_{p_1 \in \mathcal{A}_g} \sum_{x=0}^{W} p_1(x) \ln \frac{p_1(x)}{p_0(x)}. \quad (7)$$

Theorem 2: The pmf $p_1^\star$ that minimizes (7) is

$$p_1^\star(x) = \begin{cases} \frac{r^x(r^{-1}-1)}{r^{-1}-r^W} & \text{for } x \in \{0,1,\ldots,W\} \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where r is the solution to

$$\frac{W r^W - r^{-1}(W r^W + r^W - 1)}{(r^{-1}-1)(r^{-1}-r^W)} = g\frac{W}{2}. \quad (9)$$

Proof: We use variational methods for the derivation of p_1.

First, notice that the objective function is convex in p_1. Now let $q^\epsilon(x) = p_1^\star(x) + \epsilon h(x)$ and construct the Lagrangian of the objective function and the constraints:

$$L(\epsilon) = \sum_{x=0}^{W} q^\epsilon(x) \ln \frac{q^\epsilon(x)}{p_0(x)} + \mu_1 \left(\sum_{x=0}^{W} q^\epsilon(x) - 1 \right) + \mu_2 \left(\sum_{x=0}^{W} x q^\epsilon(x) - g\mathbb{E}_0[X] \right). \quad (10)$$

Next we take the derivative of the Lagrangian with respect to ϵ. Then we evaluate this quantity at $\epsilon = 0$, for all possible sequences $h(x)$:

$$\left. \frac{dL(\epsilon)}{d\epsilon} \right|_{\epsilon=0} = 0 \text{ for all } h(x) \quad (11)$$

and obtain:

$$\ln \frac{p_1^\star(x)}{p_0(x)} + 1 + \mu_1 + \mu_2 x = 0. \quad (12)$$

Therefore, the optimal $p_1^\star$ has to be of the form:

$$p_1^\star(x) = p_0(x) e^{-\mu_2 x - \mu_0} \quad (13)$$

where $\mu_0 = \mu_1 + 1$.

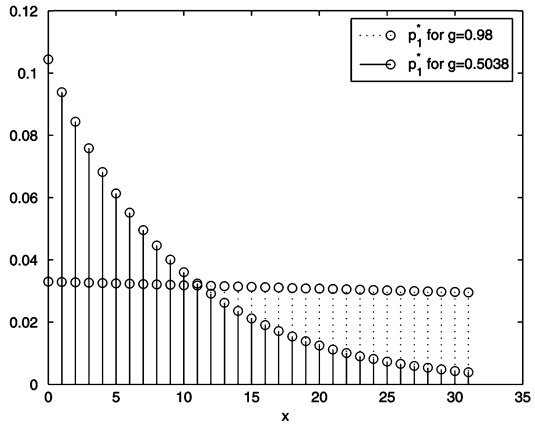


Fig. 1. Form of the least favorable pmf $p_1^\star$ for two different values of g. When g approaches 1, $p_1^\star$ approaches p_0. As g decreases, more mass of $p_1^\star$ is concentrated towards the smaller backoff values.

In order to obtain the values of the Lagrange multipliers μ_0 and μ_2 we utilize the fact that $p_0(x) = \frac{1}{W+1}$. Additionally, we utilize the constraints in $\mathcal{A}_g$. One constraint states that $p_1^\star$ must add up to 1, and therefore, by setting (13) equal to one and solving for μ_0 we obtain the following expression:

$$\mu_0 = \ln \sum_{x=0}^{W} p_0(x) r^x = \ln \frac{1}{W+1} \frac{r - r^W}{r-1} \quad (14)$$

where $r = e^{-\mu_2}$. Replacing this solution in (13) we obtain

$$p_1^\star(x) = \frac{r^x(r^{-1}-1)}{r^{-1}-r^W}. \quad (15)$$

Now we need to find the value of the other Lagrange multiplier: μ_2, or alternatively, the value of r. To solve for this value we use the constraint on the mean for $p_1^\star \in \mathcal{A}_g$. Notice that this constraint must be satisfied with equality. Rewriting this constraint in terms of (15) we obtain

$$\frac{r^{-1}-1}{r^{-1}-r^W} \sum_{x=0}^{W} x r^x = g\mathbb{E}_0[X] \quad (16)$$

from where (9) follows. ∎

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

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

Let $\Phi(D, p_1) = \mathbb{E}_1[N]$. The previously-discussed solution was obtained in the form

$$\max_{p_1 \in \mathcal{A}_g} \min_{D \in \mathcal{I}_{a,b}} \Phi(D, p_1). \quad (17)$$

In other words, we first minimized $\Phi(D, p_1)$ by using the SPRT (minimization for any p_1) and then found the $p_1^\star$ that maximizes $\Phi(\text{SPRT}, p_1^\star)$.

This solution, however, puts the misbehaving station at a disadvantage, since it is implicitly assumed (by the optimization ordering) that the detecting algorithm knows p_1 and then minimizes the number of samples by using the SPRT on this p_1.

In practice, however, it is expected to be easier for a misbehaving station to learn which detection algorithm we use, rather than the detection algorithm learning the attack distribution a priori. We are therefore interested in finding a detection algorithm resistant to *adaptive attackers*—those who can select their response based on our defenses.

Formally, the problem we are interested in solving must reverse the ordering from maximin to minimax:

$$\min_{D\in\mathcal{T}_{a,b}} \max_{p_1\in\mathcal{A}_g} \Phi(D,p_1) \tag{18}$$

Fortunately, our solution also satisfies this optimization problem since it forms a saddle point equilibrium, resulting in the following theorem:

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

$$\Phi(D^\star,p_1) \le \Phi(D^\star,p_1^\star) \le \Phi(D,p_1^\star). \tag{19}$$

We omit the proof of this result; the details can be found in [15].

As a consequence of this theorem, there is no incentive for deviation from $(D^\star,p_1^\star)$ for any of the players (the detection agent or the misbehaving node).

C. A Less Powerful Adversary Model

So far we have assumed that the adversary has the knowledge of the detection algorithm used (the SPRT in our case) in order to find the least favorable distribution $p_1^\star$. Nevertheless, $p_1^\star$ can be argued to be a good adversarial strategy against any detector (in the asymptotic observation case $n \to \infty$).

Information theory has given bounds on the probability of detection and false alarm for an optimal detector in terms of the Kullback-Leibler divergence between the distribution of the two hypothesis [8], [3].

In our case, the Kullback-Leibler divergence between p_0 and p_1,denoted as $D(p_1||p_0)$, is given by (7) (up to a scaling factor). Applying the results from information theory, the probability of detection of the optimal decision algorithm (when the false alarm rate tends to zero, and n is large enough) is lower bounded by $1-2^{-nD(p_1||p_0)}$.

It is now clear that an adversary that tries to minimize the probability of detection, under these conditions, will attempt to minimize (7), leading to the same $p_1^\star$ we obtained in (8)

D. Evaluation of Repeated SPRT

The original setup of SPRT-based misbehavior detection proposed in [16] 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 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. Monitoring transmitting stations continuously, however, can raise a large number of false alarms if the parameters of the test are not chosen appropriately.

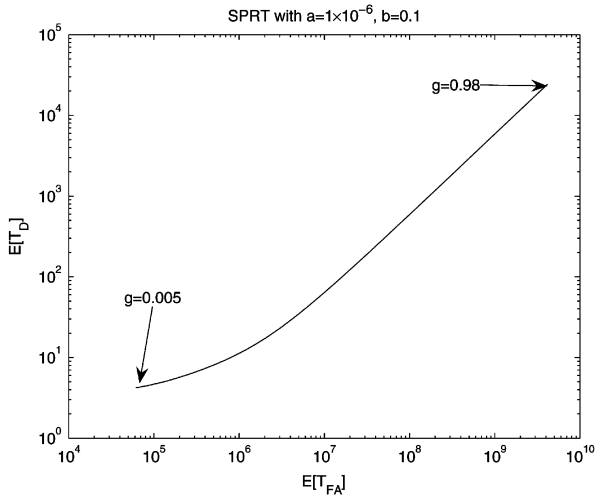


Fig. 2. Tradeoff curve between the expected number of samples for a false alarm $E[T_{\rm FA}]$ and the expected number of samples for a detection $E[T_D]$. For fixed a and b, as g increases the time to detection or to false alarms increases exponentially.

In this section we propose 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 $\mathbb{E}[T_{\rm FA}]$ as the evaluation parameters.

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

$$\mathbb{E}[T_{\rm FA}] = \frac{\mathbb{E}_0[N]}{a} \text{ and } \mathbb{E}[T_D] = \frac{\mathbb{E}_1[N]}{1-b}. \tag{20}$$

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 [5], [1], 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 the DOMINO detection algorithm. The first step of the algorithm is based on computation of the average value of back-off observations: $X_{ac} = \sum_{i=1}^{m} X_i/m$. In the next step, the averaged value is compared to the given reference back-off 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 back-off exceeds the threshold

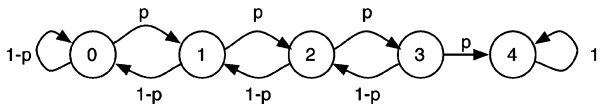


Fig. 3. For $\mathrm{K} = 3$, the state of the variable `cheat_count` can be represented as a Markov chain with five states. When `cheat_count` reaches the final state (4 in this case) DOMINO raises an alarm.

γB. DOMINO raises a false 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
    }
}
elseif cheat_count > 0 {
    cheat_count = cheat_count − 1
}
```

It is now easy to observe that DOMINO is a sequential test (N, d_N), with stopping time N equal to $m * N_t$ (where N_t represents the number of steps `cheat_count` takes to exceed K) and the decision rule each time DOMINO stops is $d_N = 1$.

In order to compare the performance of this sequential test with our SPRT, we need to derive new expressions for DOMINO; mainly, the average time between false alarms $\mathbb{E}[T_{\mathrm{FA}}]$ and the average waiting time for a detection $\mathbb{E}[T_D]$. However, unlike the SPRT case, where these expressions are easy to derive, in DOMINO we need to do some more work because (1) we are not aware of an analytical model for DOMINO, and (2) the parameters m, γ and K in DOMINO are difficult to tune because there has not been any analytical study of their influence on $\mathbb{E}[T_{\mathrm{FA}}]$ and $\mathbb{E}[T_D]$. The correlation between DOMINO and SPRT parameters is further addressed in Section VII.

In order to provide an analytical model for the performance of the algorithm, we model the detection mechanism in two steps:

1) We first define $p := \Pr[\frac{1}{m}\sum_{i=1}^{m} X_i \leq \gamma B]$
2) We define a Markov chain with transition probabilities p and $1 - p$. The absorbing state represents the case when misbehavior is detected (note that we assume m is fixed, so p does not depend on the number of observed back-off values). A Markov chain for $K = 3$ is shown in Fig. 3.

A. Computing the Transition Probabilities: P^0 and P^1

We can now write

$$p = p^0 = \mathbb{P}_0\left[\frac{1}{m}\sum_{i=1}^{m} X_i \leq \gamma B\right] \tag{21}$$

when the samples X_i are generated by a legitimate station. Otherwise, if the samples X_i are generated by $p_1(x)$ we need to compute:

$$p = p^1 = \mathbb{P}_1\left[\frac{1}{m}\sum_{i=1}^{m} X_i \leq \gamma B\right]. \tag{22}$$

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 Section II, we assume that each X_i is uniformly distributed on $\{0, 1, \ldots, W\}$. Therefore, the mean of p_0 is $\mathbb{E}_0[X_i] = \frac{W}{2}$ and its variance $\mathbb{VAR}_0(X_i) = \frac{W(W+2)}{12}$. Recall 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^0 we derive the following expression:

$$\begin{aligned} p^0 &= \mathbb{P}_0\left[\sum_{i=1}^{m} X_i \leq m\gamma B\right] \\ &= \sum_{k=0}^{\lfloor m\gamma B\rfloor} \mathbb{P}_0\left[\sum_{i=1}^{m} X_i = k\right] \\ &= \sum_{k=0}^{\lfloor m\gamma B\rfloor} \sum_{\{(x_1,\ldots,x_m):\sum_{i=1}^{m} x_i = k\}} \frac{1}{(W+1)^m} \end{aligned} \tag{23}$$

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 $x_i \in \{0, 1, \ldots W\}$.

In general, there are three ways of obtaining the value for (23): (1) we can try to derive an analytical expression via a combinatorial formula, (2) we can use the moment generating function for obtaining the exact numerical value for p^0, or (3) we obtain an *approximate* value by using the Central Limit Theorem.

Following the combinatorial approach, the number of ways that m integers can sum up to k is

$$\binom{m+k-1}{k}$$

and therefore,

$$\sum_{k=0}^{\lfloor m\gamma B\rfloor} \binom{m+k-1}{k} = \binom{m+\lfloor m\gamma B\rfloor}{\lfloor m\gamma B\rfloor}$$

An additional constraint is, however, 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, using the moment generating function we can obtain an efficient alternative way for computing $\mathbb{P}_0[\sum_{i=1}^{m} X_i = k]$. We first define $Y := \sum_{i=1}^{m} X_i$. It

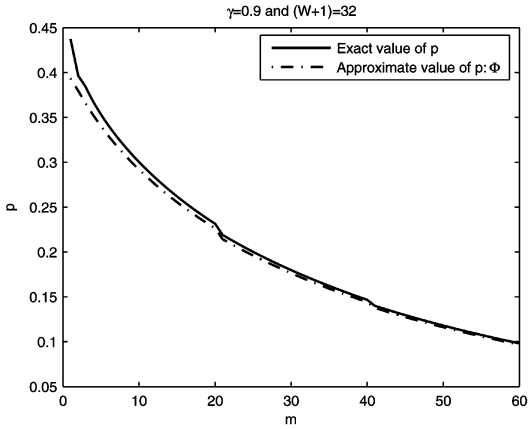


Fig. 4. Exact and approximate values of p as a function of m.

is well known that the moment generating function of $Y, M_Y(s) = M_X(s)^m$, can be computed as follows:

$$M_Y(s) = \frac{1}{(W+1)^m}(1 + e^s + \cdots + e^W)^m$$
$$= \frac{1}{(W+1)^m} \times \sum_{\left\{ k_0, \ldots, k_W : \atop \sum k_i = m \right\}} \binom{m}{k_0; \cdots; k_W} 1^{k_0} e^{sk_1} \cdots e^{sWk_W}$$

where $\binom{m}{k_0; k_2; \cdots; k_W}$ is the multinomial coefficient $\frac{m!}{k_0! k_1! \cdots k_W!}$.

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 (24). This result can be used for the efficient computation of p by using (23).

Alternatively, we can approximate the computation of p for large values of m. The approximation arises because 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 is therefore omitted.

B. Expected Time to Absorption in the Markov Chain

We now derive the expression for the expected time to absorption for a Markov Chain with $K+1$ states. Let μ_i be the expected number of transitions until absorption given that the process starts at state i. In order to compute the stopping times $\mathbb{E}[T_D]$ and $\mathbb{E}[T_{\text{FA}}]$, it is necessary to find the expected time to absorption starting from state zero, μ_0. Therefore, $\mathbb{E}[T_D] = m \times \mu_0$ (computed under $p = p^1$) and $\mathbb{E}[T_{\text{FA}}] = m \times \mu_0$ (computed under $p = p^0$).

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

$$\mu_{K+1} = 0$$
$$\mu_i = 1 + \sum_{j=0}^{K+1} p_{ij}\mu_j \text{ for } i \in \{0, 1 \ldots, K\}$$

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

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

For example, for $K = 3$ we obtain

$$\begin{bmatrix} -p & p & 0 & 0 \\ 1-p & -1 & p & 0 \\ 0 & 1-p & -1 & p \\ 0 & 0 & 1-p & -1 \end{bmatrix} \begin{bmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

and the solution we are interested is

$$\mathbb{E}[\text{time to absoption}] = \mu_0 = \frac{1 - p + 2p^2 + 2p^3}{p^4}.$$

V. Theoretical Comparison

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

For the 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 [5]). However, in DOMINO it is not clear how the parameters m, K, and γ affect our metrics, so we vary all the available parameters to explore and find the best possible performance of DOMINO.

Fig. 5 illustrates the performance of DOMINO for $K = 3$ (the default value used in [18]). Each curve for γ has m ranging between 1 and 60. Observing the results in Fig. 5, we conclude that the best performance of DOMINO is obtained for $\gamma = 0.7$, regardless of m. Therefore, this value of γ 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 notice that with the increase of K, the point with $m = 1$ forms a performance curve that is better than any other point with $m > 1$.

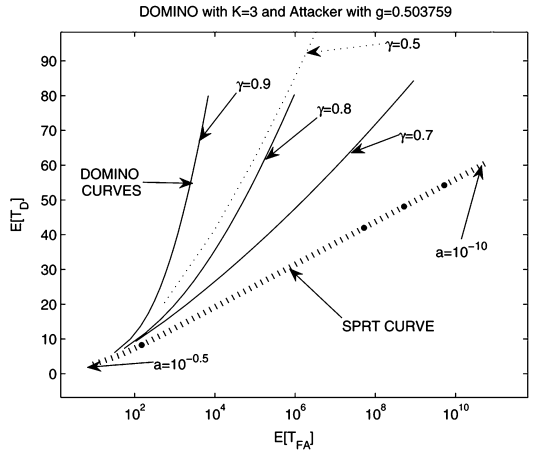


Fig. 5. DOMINO performance for $K = 3$, m ranges from 1 to 60. γ is shown explicitly. As γ tends to either 0 or 1, the performance of DOMINO decreases. The SPRT outperforms DOMINO regardless of γ and m.

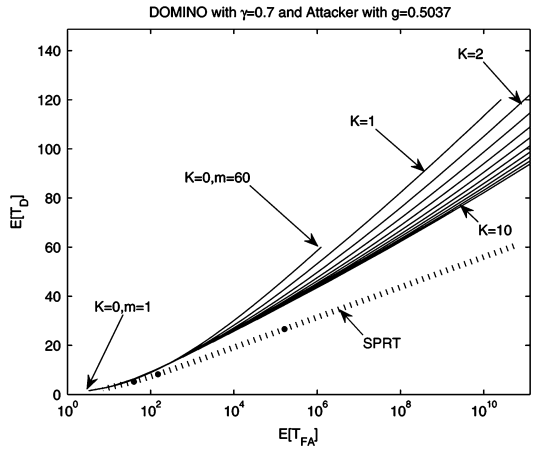


Fig. 6. DOMINO performance for various thresholds $K, \gamma = 0.7$ and m in the range from 1 to 60. The performance of DOMINO decreases with increase of m. For fixed γ, the SPRT outperforms DOMINO for all values of parameters K and m.

Consequently, Fig. 7 represents the best possible performance for DOMINO; that is, we let $m = 1$ and change K from 1 up to 100. We again test different γ values for this configuration, and conclude that the best γ is still close to the optimal value of 0.7 derived from experiments in Fig. 5. Even with the optimal setting, DOMINO is outperformed by the SPRT.

Since m was not considered as a tuning parameter in the original DOMINO algorithm (m was random in [18], 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 according to our analysis, any other value of m is suboptimal. Notice that O-DOMINO can be expressed as

$$K_i = (K_{i-1} + (1_{X_i \leq \gamma B} - 1_{X_i > \gamma B}))^+ \tag{24}$$

where 1_R is the indicator random variable for event R ($1_R = 1$ if the outcome of the random experiment is event R, and $1_R = 0$ otherwise), and $(x)^+ = x$ if $x \geq 0$ and 0 otherwise.

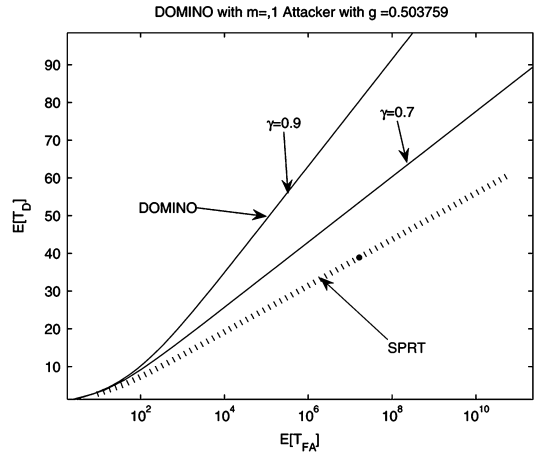


Fig. 7. The best possible performance of DOMINO is when $m = 1$ and K changes in order to accommodate for the desired level of false alarms. The best γ must be chosen independently.

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.

By looking at O-DOMINO's behavior (24), we were reminded of *quickest change-detection* nonparametric statistics. One particular nonparametric statistic that has a very similar behavior to DOMINO is the nonparametric cumulative sum (CUSUM) statistic [4]. Nonparametric CUSUM is initialized with $Y_0 = 0$ and updates its value as follows:

$$Y_i = (Y_{i-1} + (\gamma B - X_i))^+. \tag{25}$$

An alarm is fired whenever $Y_i > c$, where c is a threshold that can be used as a parameter to control the tradeoff between the rate of false alarms and the rate of missed detections.

A. Properties of the Nonparametric CUSUM Statistic

Assuming $\mathbb{E}_0[X] > \gamma B$ and $\mathbb{E}_1[X] < \gamma B$—i.e., the expected back-off value of an honest node is 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 $c \to \infty$

$$\sup_i |\ln(\mathbb{P}_0[Y_i > c])| = \mathcal{O}(c). \tag{26}$$

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

$$T_D = \frac{c}{\gamma B - \mathbb{E}_1[X]}. \tag{27}$$

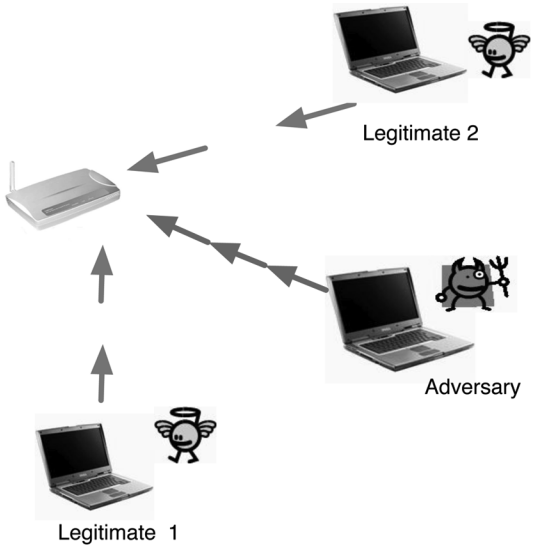


Fig. 8. Simulations: two legitimate participants compete with the adversary.

The proof is a straightforward extension of the case originally considered in [4].

B. Relationship Between Nonparametric CUSUM and DOMINO

It is easy to observe that the CUSUM test is similar to O-DOMINO; c in CUSUM being equivalent to the upper threshold K in DOMINO, and the statistic y in CUSUM being equivalent to the variable `cheat_count` in O-DOMINO.

The main difference between O-DOMINO 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 can be expressed as $\gamma B - x_i$ (or a decrease of y_i if $y_i - \gamma B - x_i < 0$); in other words, it is proportional to the amount of time the station did not attempt to access the channel.

VII. Experimental Results

A. Assumptions and Experimental Setup

We now proceed to experimental evaluation of the analyzed detection schemes. It has already been mentioned that we assume existence of an intelligent adaptive attacker that is able to adjust its access strategy depending on the level of congestion in the environment. Namely, we assume that, in order to minimize the probability of detection, the attacker chooses legitimate over selfish behavior when the congestion level is low, and an adaptive selfish strategy in congested environments. Due to these reasons, when constructing the experiments, we assume that all stations have packets to send at any given time. We assume that the attacker will employ the *least-favorable* misbehavior strategy for our detection algorithm, enabling us to estimate the maximal detection delay. It is important to mention that this setting also represents the worst-case scenario with regard to the number of false alarms per unit of time because the detection algorithm is forced to make a maximum number of decisions per unit of time. (We expect the number of alarms to be smaller in practice.)

The back-off distribution of an optimal attacker was implemented in the network simulator Opnet[1] and tests were performed for various levels of false alarms. We note that the simulations were performed with nodes that followed the standard IEEE 802.11 access protocol (with exponential back-off). The results presented in this work correspond to the scenario consisting of two legitimate and one selfish node competing for channel access. The corresponding scenario is presented in Fig. 8. We consider the scenario where one adaptive intelligent adversary competes with two legitimate stations for channel access. Consequently, in a fair setting, each protocol participant should be allowed to access the medium for 33% of time under the assumption that each station is backlogged and has packets to send at any given time slot. The detection agent was implemented such that any observed back-off value $X_i > W$ was set up to be W. Our experiments show that it works well in practice.

The resulting 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 for more aggressive attacks, 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).

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 seconds—a quantity that is more meaningful and intuitive in practice.

B. Results

1) Testing the Detection Schemes: 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.

We first compare O-DOMINO with the original configuration suggested for DOMINO. The original DOMINO algorithm, as suggested in [18], assumes $K = 3$ and $\gamma = 0.9$. Furthermore, as we have already mentioned, the original DOMINO takes the back-off 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, we first

[1]http://www.opnet.com/products/modeler/home.html

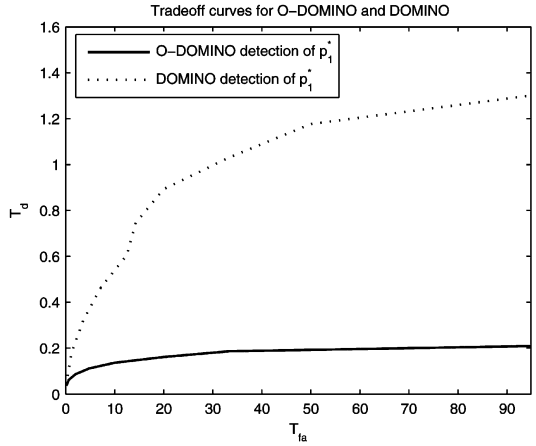


Fig. 9. Tradeoff curves for the original DOMINO algorithm with $K = 3, \gamma = 0.9$ and different values of m versus O-DOMINO with $\gamma = 0.7$ and different values of K.

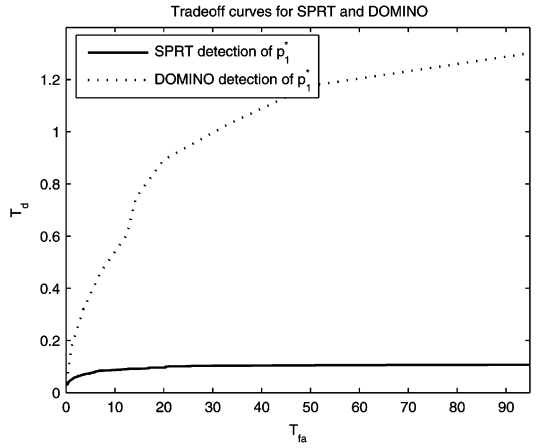


Fig. 10. Comparison of the SPRT test against DOMINO using the optimal (exponential) attack. The gain of the attacker is identical in both cases.

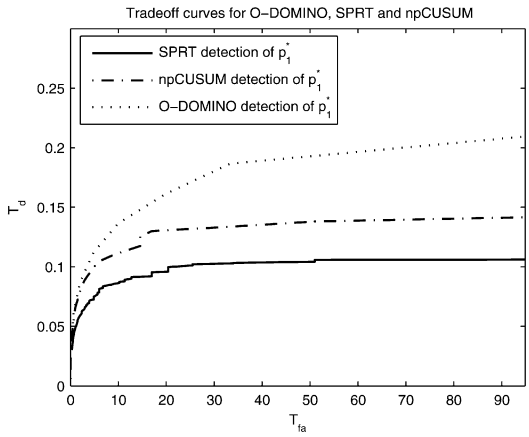


Fig. 11. Tradeoff curves for SPRT with $b = 0.1$ and different values of a versus nonparametric CUSUM and O-DOMINO with $\gamma = 0.7$ and different values of K.

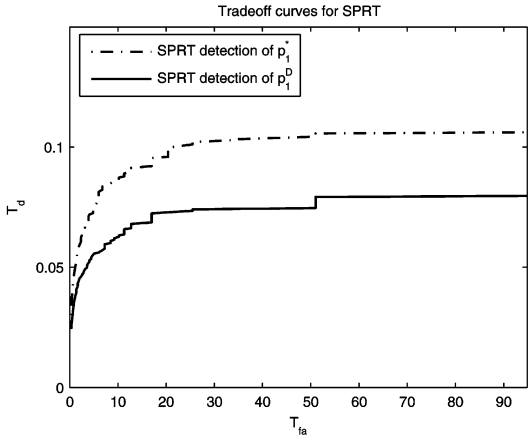


Fig. 12. Tradeoff curves for SPRT with $b = 0.1$ and different values of a. One curve shows its performance when detecting an adversary that chooses p_1^D and the other is the performance when detecting an adversary that chooses $p_1^\star$.

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), versus O-DOMINO with $\gamma = 0.7$ (the suggested optimal performance achievable by the O-DOMINO algorithm according to our analysis). This comparison can be seen in Fig. 9; similar performance was also observed for other configurations of γ and K in 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 γ. Therefore, our experiments suggest that having γ close to 0.7 is the optimal setting for DOMINO; a result that coincides with our analytical derivations.

We also test the performance of DOMINO and the SPRT in the presence of the worst-case attack strategy $p_1^\star$. Fig. 10 shows that SPRT significantly outperforms DOMINO in the presence of an optimal attacker.

We now test how our three proposed algorithms compare to each other. Fig. 11 provides experimental evidence confirming our predictions. In general, since the SPRT is optimal, it performs better than the nonparametric CUSUM statistic, and because the nonparametric CUSUM statistic takes into account the level of misbehavior observed (or normal behavior) for each sample, then it outperforms the restricted addition and substraction in O-DOMINO.

2) Testing the Optimality of $P_1^\star$: We have therefore shown how SPRT is the best test when the adversary selects $p_1^\star$. We now show that if the adversary deviates from $p_1^\star$ 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 [18]; 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 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. 12 shows the performance of SPRT when the adversary uses p_1^D and $p_1^\star$. In our mathematical analysis we proved that $p_1^\star$ is the worst possible distribution our detection algorithm (SPRT) can face, i.e., any other distribution will generate shorter detection delay. The results presented in Fig. 12 support this statement, since p_1^D is detected faster than $p_1^\star$ when the SPRT is used as detection algorithm.

Note that the same phenomenon happens for DOMINO. As can be seen in Fig. 13, an adversary using $p_1^\star$ against DOMINO

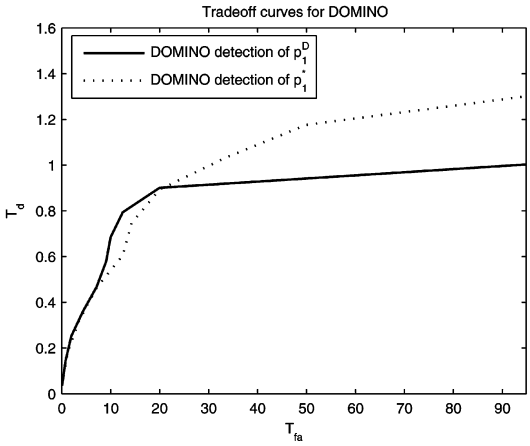


Fig. 13. Comparison of the DOMINO test against p_1^D versus the optimal attack $p_1^\star$. The gain of the attacker is identical in both cases.

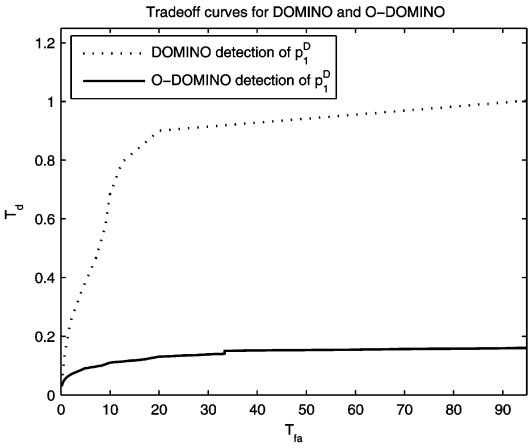


Fig. 14. Tradeoff curves for DOMINO and O-DOMINO with the same parameters as in Fig. 9. However this time instead of detecting an adversary that chooses $p_1^\star$ we measure their performance against an adversary that chooses p_1^D.

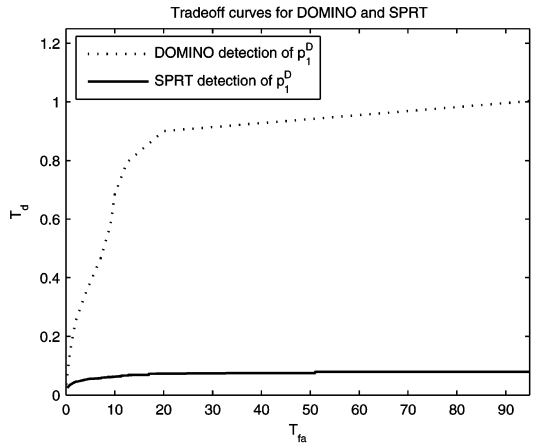


Fig. 15. Comparison of the SPRT against DOMINO, using the optimal attack from DOMINO paper. The gain of the attacker is identical in both cases.

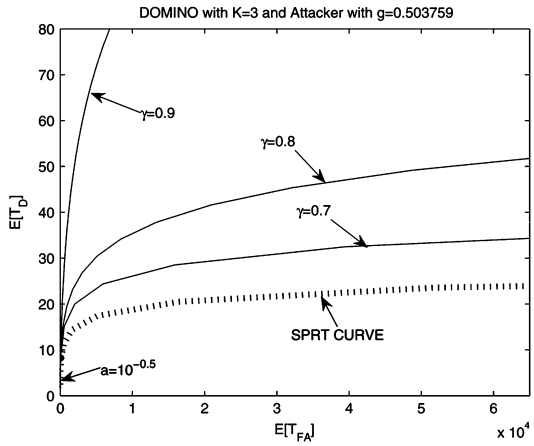


Fig. 16. Comparison between theoretical and experimental results: theoretical analysis with linear x axis closely resembles the experimental results.

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 $p_1^\star$ against DOMINO.

As we described before, however, $p_1^\star$ can be argued to be a good adversarial strategy against any detector in the asymptotic observation case because it minimizes the Kullback-Leibler divergence between p_1 and p_0. On the other hand we could not find any theoretical motivation for the definition of p_1^D.

We now test the performance of our algorithms against p_1^D. Fig. 14 compares the performance of DOMINO and O-DOMINO with respect to p_1^D. When compared to Fig. 9, it is evident that DOMINO and O-DOMINO perform better when the adversary chooses p_1^D.

We also compare the performance of SPRT and DOMINO when the adversary chooses p_1^D. The results are presented in Fig. 15. As expected, a sub-optimal attack p_1^D is detected with a substantially smaller detection delay with the SPRT. More specifically, we observe that the detection delay for a sub-optimal strategy is more than 10 times larger than the one for the optimal strategy.

Note also how close the theoretical shape of the tradeoff curves is to the actual experimental data. Fig. 16 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. Discussion on Parametric, Nonparametric, and Robust Statistics

Tests based on *nonparametric* statistics do not consider the distribution itself, but only certain parameters that characterize it, such as mean, median or variance. Since such tests consider only certain parts of distribution, they allow a very large class of probability distributions. Furthermore, nonparametric tests let us deal with unknown probability distributions. The disadvantage is that they throw away a lot of information about the problem that can help improve the performance of the detection test.

On the other hand, tests based on *parametric* statistics assume precise knowledge of the distributions. The advantage is that

if we know the distributions precisely, parametric statistics are optimal in the sense that they perform better (in general) than nonparametric tests. The disadvantage is that if our knowledge of the distributions is incorrect, then there is no guarantee that the test will perform well.

With this definitions it might be intuitive to think of our problem of detecting misbehavior as a nonparametric problem, since the distribution of the adversary is unknown. This was the approach followed by DOMINO (and by the nonparametric CUSUM statistic): although DOMINO does not specify the exact distribution of the adversary, it specifies a test that compares the *sample mean* of the observed process to a constant, and is thus specifying a constraint on the mean for the adversary distribution.

Our SPRT formulation attempts to use the precision given by parametric tests while avoiding the specification of the distribution of the adversary. We achieve this by using *robust* statistics: a collection of related theories on the use of approximate parametric models [10].

The usual problem with robust statistics is that even simple problems can become intractable very easily, and thus, finding a solution to the robust formulation is usually hard.

The main advantage of robust statistics is that they produce estimators that are not excessively affected by small departures from model assumptions. By defining a class of adversary distributions $\mathcal{A}_g$ and then selecting the saddle point strategy between the detector and the least-favorable conditions, we are guaranteed that we can tolerate any deviation of the adversary distribution (since by the saddle point condition any other adversary distribution p_1 will make our test perform better than with $p_1^\star$). At the same time we are incorporating more information about the problem than with nonparametric statistics, and thus we expect our system to perform better than nonparametric tests.

IX. Conclusion

In this work, we performed an extensive analytical and experimental comparison of the existing misbehavior detection schemes in the IEEE 802.11 MAC. We confirmed the optimality of the SPRT-based detection schemes and provided an analytical and intuitive explanation of why the other schemes exhibit suboptimal performance when compared to the SPRT schemes. In addition to that, we offered an extension to DOMINO: preserving its original idea and simplicity, while significantly improving its performance.

Our results show the value of doing a rigorous formulation of the problem and providing a formal adversarial model since it can outperform heuristic solutions. We believe our model applies not only to MAC-layer problems, but to a more general adversarial setting. In several practical security applications such as in biometrics, spam filtering, watermarking etc., the attacker has control over the attack distribution and this distribution can be modeled in similar fashion as in our approach: with the use of robust statistics and minimax games.

An issue of further study concerns the response mechanisms. When an alarm is raised, we must consider the effects of our reaction, such as, denying access to the medium for a limited period of time. If we observe constant misbehavior (even after penalizing the station), we might consider more severe penalties, such as revocation of the station from the network. Alternatively, our misbehavior detection algorithm might be part in a larger misbehavior detection engine. In this case the problem is one of combining and correlating alerts from different nodes. Finding a way to integrate our detection algorithm to larger alarm systems, and testing the effects of our response mechanisms is an area of research that must be explored further.

Appendix I

In IEEE 802.11 protocol, the back-off counter of any node freezes during the transmissions and reactivates during free periods. We observe the back-off times during a fixed period T that *does not include* transmission intervals. Consider first the case of one misbehaving and one legitimate node and assume that within the time period T, we observe $X_1, \ldots, X_N, N$ samples of the attacker's back-off and $Y_1, \ldots, Y_M, M$ samples of the legitimate node's back-offs. The attacker's percentage of accessing the channel during the period T is $\frac{N}{N+M}$. In order to obtain P_A we need to compute the limit of this ratio as $T \to \infty$. Notice that

$$X_1 + \cdots + X_N \leq T < X_1 + \cdots + X_{N+1}$$
$$Y_1 + \cdots + Y_M \leq T < Y_1 + \cdots + Y_{M+1}$$

which yields the following two equations:

$$\frac{\frac{N}{X_1+\cdots+X_N}}{\frac{N}{N+1}\frac{N+1}{X_1+\cdots+X_{N+1}} + \frac{M}{M+1}\frac{M+1}{Y_1+\cdots+Y_{M+1}}} \geq \frac{\frac{N}{T}}{\frac{N}{T}+\frac{M}{T}}$$
$$\frac{\frac{N}{N+1}\frac{N+1}{X_1+\cdots+X_{N+1}}}{\frac{N}{X_1+\cdots+X_N} + \frac{M}{Y_1+\cdots+Y_M}} \leq \frac{\frac{N}{T}}{\frac{N}{T}+\frac{M}{T}}.$$

Letting $T \to \infty$ results in $N, M \to \infty$ and from the previous double inequality, by applying the Law of Large Numbers, we conclude that for the case of one misbehaving node against n legitimate ones

$$\mathrm{P}_A = \lim_{N,M\to\infty} \frac{N}{N+M} = \frac{\frac{1}{\mathbb{E}_1[X]}}{\frac{1}{\mathbb{E}_1[X]} + \frac{n}{\mathbb{E}_0[Y]}} = \frac{1}{1+n\frac{\mathbb{E}_1[X]}{\mathbb{E}_0[Y]}}.$$

References

[1] S. Axelsson, "The base-rate fallacy and its implications for the difficulty of intrusion detection," in *Proc. 6th ACM Conf. Computer and Communications Security (CCS'99)*, Nov. 1999, pp. 1–7.

[2] N. BenAmmar and J. S. Baras, "Incentive compatible medium access control in wireless networks," presented at the 2006 Workshop on Game Theory for Communications and Networks, Pisa, Italy, 2006.

[3] R. E. Blahut, *Principles and Practice of Information Theory*. Reading, MA: Addison-Wesley, 1987.

[4] B. E. Brodsky and B. S. Darkhovsky, "Nonparametric methods in change-point problems," in *Mathematics and Its Applications*. Boston, MA: Kluwer Academic , 1993, vol. 243.

[5] A. A. Cárdenas, J. S. Baras, and K. Seamon, "A framework for the evaluation of intrusion detection systems," presented at the 2006 IEEE Symp. Security and Privacy, Oakland, CA, May 2006.

[6] A. A. Cárdenas, S. Radosavac, and J. S. Baras, "Detection and prevention of MAC layer misbehavior in ad hoc networks," in *Proc. 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04)*, Washington, DC, 2004, pp. 17–22.

[7] L. Chen and J. Leneutre, "Selfishness, not always a nightmare: Modeling selfish MAC behavior in wireless mobile ad hoc networks," presented at the 27th IEEE Int. Conf. Distributed Computing Systems (ICDCS), Toronto, ON, Canada, Jun. 25–29, 2007.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[9] E. Altman, R. E. Azouzi, and T. Jiménez, "Slotted Aloha as a game with partial information," *Comput. Netw.*, vol. 45, no. 6, pp. 701–703, 2004.
[10] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*, rev. ed. New York: Wiley-Interscience, 2005.
[11] P. Kyasanur and N. Vaidya, "Selfish MAC layer misbehavior in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 5, pp. 502–516, 2005.
[12] A. B. MacKenzie and S. B. Wicker, "Stability of multipacket slotted Aloha with selfish users and perfect information," in *Proc. IEEE INFOCOM'03*, San Francisco, CA, 2003, pp. 1583–1590.
[13] I. A. Mario Cagalj, S. Ganeriwal, and J.-P. Hubaux, "On selfish behavior in CSMA/CA networks," in *Proc. IEEE INFOCOM'05*, Miami, FL, 2005, pp. 2513–2524.
[14] S. Radosavac, A. Cárdenas, J. S. Baras, and G. V. Moustakides, "Detecting IEEE 802.11 MAC-layer misbehavior in ad hoc networks: Robust strategies against individual and colluding attackers," *J. Computer Security*, vol. 15, no. 1, pp. 103–128, Jan. 2007, Special Issue on Security of Ad Hoc and Sensor Networks.
[15] S. Radosavac, G. V. Moustakides, J. S. Baras, and I. Koutsopoulos, "An analytic framework for modeling and detecting access layer misbehavior in wireless networks," *ACM Trans. Information and System Security (TISSEC)*, vol. 11, no. 4, Nov. 2008, to be published.
[16] S. Radosavac, J. S. Baras, and I. Koutsopoulos, "A framework for MAC protocol misbehavior detection in wireless networks," in *Proc. 4th ACM Workshop on Wireless Security (WiSe'05)*, Cologne, Germany, 2005, pp. 33–42.
[17] M. Raya, I. Aad, J.-P. Hubaux, and A. E. Fawal, "DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1691–1705, Dec. 2006.
[18] M. Raya, J.-P. Hubaux, and I. Aad, "DOMINO: A system to detect greedy behavior in IEEE 802.11 hotspots," presented at the 2nd Int. Conf. Mobile Systems, Applications and Services (MobiSys2004), Boston, MA, Jun. 2004.
[19] Y. Rong, S. K. Lee, and H.-A. Choi, "Detecting stations cheating on backoff rules in 802.11 networks using sequential analysis," in *Proc. 25th IEEE INFOCOM*, Barcelona, Spain, Apr. 23–29, 2006, pp. 1–13.
[20] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, "Specification-based anomaly detection: A new approach for detecting network intrusions," in *Proc. 9th ACM Conf. Computer and Communications Security*, Washington, DC, 2002, pp. 265–274.
[21] A. Wald, *Sequential Analysis*. New York: Wiley, 1947.

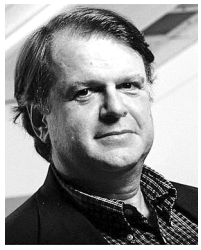
Alvaro A. Cárdenas (M'06) received the B.S. degree with a major in electrical engineering and a minor in mathematics from the Universidad de los Andes, Bogotá, Colombia, in 2000, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, in 2002 and 2006, respectively.

He is currently a postdoctoral scholar in the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA. His research interests include intrusion detection, the security of control systems, and statistical machine learning approaches for computer security.

Dr. Cárdenas is the recipient of a graduate school fellowship from the University of Maryland from 2000 to 2002, a distinguished research assistantship from the Institute for Systems Research from 2002 to 2004, and a best paper award from the 23rd Army Science Conference in 2002. He is a member of the ACM.

Svetlana Radosavac (M'01) received the B.S. degree in electrical engineering from the University of Belgrade in 1999, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, in 2002 and 2007, respectively.

From June to October 2007, she was a research associate with the Institute for Systems Research at the University of Maryland, where she worked on the analysis of Byzantine behavior of users in wireless networks and misbehavior detection in the IEEE 802.11. She is currently a Research Engineer at DoCoMo Communications Laboratories USA, Inc., Palo Alto, CA. Her research interests include network security, game theory, network economy and virtualization.



John S. Baras (M'73–SM'73–F'84) received the B.S. degree in electrical engineering with highest distinction from the National Technical University of Athens, Greece, in 1970, and the M.S. and Ph.D. degrees in applied mathematics from Harvard University, Cambridge, MA, in 1971 and 1973, respectively.

Since 1973, he has been with the Department of Electrical and Computer Engineering, University of Maryland at College Park, where he is currently Professor, member of the Applied Mathematics and Scientific Computation Program Faculty, and Affiliate Professor in the Department of Computer Science. From 1985 to 1991, he was the Founding Director of the Institute for Systems Research (ISR) (one of the first six NSF Engineering Research Centers). In February 1990, he was appointed to the Lockheed Martin Chair in Systems Engineering. Since 1991, he has been the Director of the Maryland Center for Hybrid Networks (HYNET), which he co-founded. He has held visiting research scholar positions with Stanford, MIT, Harvard, the Institute National de Reserche en Informatique et en Automatique, the University of California at Berkeley, Linkoping University and the Royal Institute of Technology in Sweden. His research interests include control, communication and computing systems.

Dr. Baras' awards include: the 1980 George S. Axelby Prize of the IEEE Control Systems Society; the 1978, 1983 and 1993 Alan Berman Research Publication Award from NRL; the 1991 and 1994 Outstanding Invention of the Year Award from the University of Maryland; the 1996 Engineering Research Center Award of Excellence for Outstanding Contributions in Advancing Maryland Industry; the 1998 Mancur Olson Research Achievement Award, from the University of Maryland College Park; the 2002 Best Paper Award at the 23rd Army Science Conference; the 2004 Best Paper Award at the Wireless Security Conference WISE04; the 2007 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communication Systems. He is a Foreign Member of the Royal Swedish Academy of Engineering Sciences (IVA).