This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

A Robust Kolmogorov-Smirnov Detector for Misbehavior in IEEE 802.11 DCF

Alberto Lopez Toledo and Xiaodong Wang
Electrical Engineering, Columbia University, 500 West 120th Street, New York, NY 10027, USA.
Phone: (212) 854-6592, Fax: (212) 932-9421, e-mail: {alberto,wangx}@ee.columbia.edu.

***Abstract*— The CSMA/CA protocols are designed under the assumption that all participant nodes would abide to the protocol rules. This is of particular importance in distributed protocols such as the IEEE 802.11 distributed coordinating function (DCF), in which nodes control their own backoff parameters. A selfish node may deliberately modify its random assignment and gain unfair access to the network resources. This would result in an increased observed collision probability for the rest of the nodes, that would increase their backoff windows as a result, further increasing the benefit of the selfish nodes. In this work, we develop of a robust non parametric batch detector based on the Kolmogorov-Smirnov (K-S) statistics that does not require any modification on the existing CSMA/CA protocols, and we apply it to detect misbehaviors in an IEEE 802.11 DCF network using the ns-2 simulator. We show that our method has a performance comparable to the optimum detectors with perfect information for the majority of misbehaviors, and it is able to detect any deviation from the protocol after just a few transmissions from the offending terminal.**

I. Introduction

Deviation from legitimate protocol operation in wireless networks has received considerable attention from the research community in recent years. Most of the current research deals with the case of "malicious" attacks, in which terminals do not obey the protocols with the sole objective of disrupting the operation of the network, even in their own determent. Malicious misbehaviors of this kind are often referred to as denial-of-service attacks [1]. "Selfish" misbehaviors, on the other hand, are inflicted by users who wish to increase their own share of the common transmission resources; these users are rational, and not malicious [2]. A typical selfish misbehavior may include terminals that refuse to forward packets on behalf of other hosts to conserve energy, or terminals that knowingly modify protocol parameters to gain unfair access to the channel. The threat of a selfish terminal is more credible than that of the malicious terminal (DoS), as every terminal in the network has a clear incentive to misbehave.

Detecting misbehavior in IEEE 802.11 DCF is not an easy task. The main difficulty comes from the random operation of the CSMA/CA protocol, and is exacerbated by the nature of the wireless medium itself, where channel impairment and interference make network conditions to appear different for different terminals. Various attempts have been made in the literature to attack the problem: a heuristic set of conditions is proposed in [3] for testing the manipulation of MAC protocol parameters. In [2], a modification to the IEEE 802.11 MAC protocol is proposed to detect selfish misbehavior. The approach, however, assumes a trustworthy receiver, which represents its major drawback. In [4], a minimax detection framework is employed to analyze the instance of theoretical worst-case attacks. The approach is more robust, but no operational method to detect misbehavior is proposed.

The prompt detection of such misbehaving nodes is a major security issue. In fact it is shown in [5] that an IEEE 802.11 DCF can be designed with complete stability (i.e., free of misbehavior) if there exists a way to detect terminals that deviate from the protocol in a prompt way. In this work we present a detection scheme that solves this problem.

[1]Alberto Lopez Toledo is supported by the Rafael del Pino Foundation.

II. Effect of Selfish Misbehavior in IEEE 802.11

Given its distributed nature, the IEEE 802.11 DCF bases its operation on the individual terminals correctly assigning their backoff intervals according to the protocol. A selfish terminal might try to select small backoff intervals to gain a significant advantage in channel access probability over time. By increasing their transmission probabilities, selfish terminals produce an increment in the number of collisions in the network, forcing the rest of (well-behaved) terminals to, in turn, increment their backoff intervals, further increasing the advantage for the selfish terminals. For the rest of the paper we consider a terminal to be operating correctly if it uses the binary exponential protocol with $CW_{\min} = 32$ and $CW_{\max} = 1024$, where CW_{min} and CW_{max} are the minimum and maximum contention windows respectively [5].

The effect of a misbehaving terminal can be drastic to the operation of the protocol. Figs. 1 show the collision probability of an IEEE 802.11 DCF network with one misbehaving node always using its minimum backoff window ($CW_{\min} = CW_{\max} = 8$). As we can see, the misbehaving terminal will observe a much reduced collision probability, resulting in a share of the medium as high as 5 times of those of the well-behaved terminals. The difference is notable even for more moderate misbehaviors, making even small deviations from the protocol a strong incentive for a node to misbehave.

III. Problem Formulation

The strategy used by the misbehaving terminal is completely unpredictable. This uncertainty makes the problem of detecting misbehaving terminals a difficult one. Unlike other approaches to misbehavior based on throughput difference (channel conditions may cause certain terminals to have more throughput than others while still abiding by the protocol; also, throughput

1-4244-0353-7/07/$25.00 ©2007 IEEE

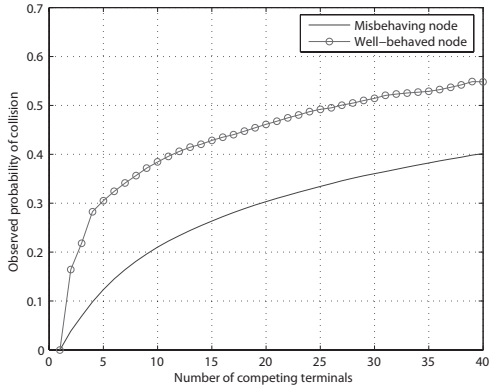

Fig. 1: Effect of having just one misbehaving node in the observed probability of collision of an IEEE 802.11 DCF network.

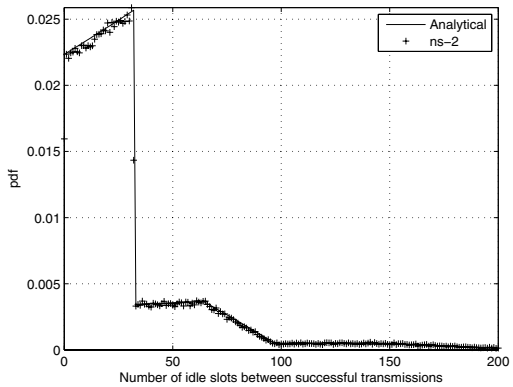

Fig. 2: The pdf of the number of idle slots between successful transmissions for an IEEE 802.11 DCF network with 10 nodes.

is protocol parameter-dependent), we are interested in knowing the sequence of backoff intervals selected by a given terminal, in particular, how many idle slots the terminal waited since its last transmission before attempting a new transmission, so that we can then check if that sequence corresponds to the case of a binary exponential increase with the correct $CW_{\min}$ and $CW_{\max}$ parameters. However, the sequence of backoff intervals selected by a terminal, i.e., its transmission *attempts*, is not directly observable in a CSMA/CA system, and in particular in an IEEE 802.11 system, because the only observable transmissions from a terminal are *successful* transmissions. Attempted transmissions that result in collisions can be observed, but it is not possible to distinguish which terminals *are involved* in them. So our observation events are the specific times at which a given terminal transmits. In particular, and because the terminals only decrement their backoff counters when the channel is idle, we focus on the number of idle slots between two consecutive successful transmissions of a certain terminal.

A. Hypothesis Test

Let $x_1, ..., x_K$ be a sequence of observations related to the operation of a CSMA/CA terminal. We consider two hypotheses, the *null* hypothesis H_0 corresponds to the observed terminal *not misbehaving*, while the alternate hypothesis H_1 corresponds to the case that the terminal *is misbehaving*. We bias towards the not misbehaving case because the cost of a false alarm is high, as it is more important to guarantee that the well-behaved terminals are not accused of misbehaving (and potentially being disconnected from the network). We write this problem as a standard hypothesis test

$$\text{choose} \begin{cases} H_0 : x_1, ..., x_K \sim f_0 \\ H_1 : x_1, ..., x_K \sim f_1, \end{cases} \quad (1)$$

where f_0 and f_1 are the probability distributions of the observations when a node is not misbehaving and misbehaving respectively. We refer to these distributions as the *strategy* of a terminal. We want to design a decision rule $\delta(x_1, ..., x_K) \in \{0, 1\}$ to discriminate between the two hypotheses.

B. Probability Distribution of Legitimate Terminals

To calculate the distribution of the samples x_i under H_0, we consider a typical IEEE 802.11 DCF, where $CW_{\min} = 32$ and $CW_{\max} = 1024$. We derive the distribution f_0 of the number of idle slots a terminal would wait between successful transmissions as follows. Let us assume that the legitimate terminal is saturating, i.e., it always has a packet to send, and let p_c be the probability that the terminal will suffer from a collision if it transmits in the current slot. After a successful transmission of a terminal, the next attempt to transmit will happen after τ_1 idle slots where $\tau_1 \sim \mathcal{U}[0, 32]$ and $\mathcal{U}$ denotes the uniform probability distribution. That transmission will be successful with probability $(1 - p_c)$, and hence $x_i = \tau_1$. If there is a collision, with probability p_c, then the terminal would double its window size and make another attempt after $\tau_2 \sim \mathcal{U}[0, 64]$ slots. If that last transmission is successful then the number of idle slots after the last successful transmission is $x_i = \tau_1 + \tau_2 \sim \mathcal{U}[0, 32] + \mathcal{U}[0, 64]$ with probability $p_c(1 - p_c)$. If there is a further collision the number of idle slots is $x_i = \mathcal{U}[0, 32] + \mathcal{U}[0, 64] + \mathcal{U}[0, 128]$ with probability $p_c^2(1 - p_c)$. Following the above argument we can easily obtain the distribution of the number of idle slots between successful transmissions, f_0, assuming p_c does not vary between successful transmissions. We denote the pdf f_0 as the strategy of a saturating legitimate terminal. Fig. 2 shows f_0 compared to the histogram of the number of idle slots between successful transmissions in an IEEE 802.11 DCF network with 10 saturating terminals using the ns-2 simulator.

C. Characterizing Misbehaving Terminals

Unlike the strategy of a legitimate terminal, the unknown strategy of a (potentially) misbehaving terminal is not unique. Let us define f_1 as the unknown strategy of the observed terminal for which we are interested in determine whether or nor it is misbehaving. In order to characterize and quantify misbehavior we will compare f_1 to the strategy of a saturating legitimate node f_0. Denote $F_1(x)$ and $F_0(x)$ as the cumulative distribution functions (cdf) for f_1 and f_0 respectively.

Consider a legitimate terminal. If the terminal is saturating, then it is clear that $F_1(x) = F_0(x)$. If the terminal is not

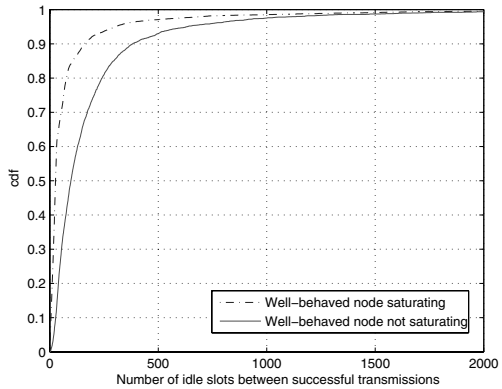

Fig. 3: The cdf of the number of idle slots between successful transmission for a saturating and non-saturating legitimate terminals.

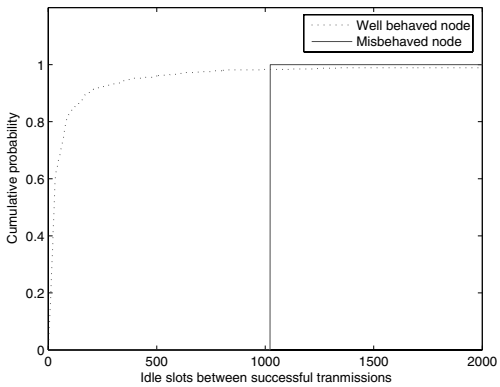

Fig. 4: Example of a misbehaving terminal with transmission probability lower than that of a legitimate terminal.

saturating, e.g., it rests for an unknown time $\Delta > 0$ after each or some transmissions, then obviously its cdf satisfies $F_1(x) < F_0(x)$, $\forall x$. In general, for any terminal using the correct protocol, either saturating or not, we have $\forall x$, $F_1(x) \leq F_0(x)$. Intuitively, if the cdf of a terminal is always on or below the cdf of a well behaved terminal that is always transmitting, then the terminal is definitely not misbehaving (see Fig. 3).

The above discussion leads to our definition of misbehavior: a terminal using a unknown strategy f_1 with cdf F_1 is misbehaving, if $\exists x$, s.t. $F_1(x) > F_0(x)$, where F_0 is the cdf of the strategy of a legitimate terminal that is saturating.

Note that our definition of misbehavior does not take into account the transmission probability (and hence, the throughput) of the terminals. It is easy to find a terminal satisfying $\exists a\ F_1(a) > F_0(a)$ and $\forall x \neq a\ F_1(x) \leq F_0(x)$, that has a transmission probability lower than that of the legitimate saturating terminal, and therefore appears non-misbehaving (see an extreme example Fig. 4). However, the CSMA/CA protocol is designed so that the transmissions of a terminal are distributed as uniformly as possible in time to avoid collisions. A terminal transmitting less than a legitimate terminal but using a different strategy such as the one in Fig. 4, may produce a disruption in the service at its transmission attempts, perturbing the normal operation of the protocol. Those terminals should be considered as misbehaving terminals. Our definition of misbehavior is general enough to capture this often overlooked type of misbehavior.

IV. Misbehavior Detection

We are interested in developing a detector that can discriminate between a legitimate terminal using f_0 and a misbehaving terminal that does not. Because the distribution f_1 of the number of idle slots between successful transmissions of a potential misbehavior, or any other parameter about its operation is unknown, it is necessary to use *distribution-free* or *nonparametric* approaches to perform the detection. In what follows we will present a nonparametric test based on the Kolmogorov-Smirnov statistic, that fits seamlessly with our definition of misbehavior.

A. Collision Probability Estimation

As seen in Section III, in order to obtain the distribution f_0 of the idle slots between successful transmissions for a saturating legitimate terminal, the probability of collision in the network p_c has to be estimated. A terminal can keep track of its own transmissions and count how many of them resulted in collisions. However, that is not suitable for terminals that do not have anything to send, and moreover, the transmission rate of a legitimate terminal will be slowed in the presence of a misbehaving terminal. A faster estimate can be obtained if a terminal does not count how many of its own transmissions resulted in collisions, but instead how many of the total number of transmissions in the network resulted in collisions. Note that it is not possible to observe how many terminals attempted a transmission for any give collision, because the identity and the number of the colliding terminals is hidden by the collision itself. However, the average number of terminals colliding (*collision factor*) γ is a function of the protocol and the number of terminals competing in the network. We define this new estimator as

$$\tilde{p}_c = \frac{C\gamma}{T' + C\gamma}, \tag{2}$$

where C is the number of collisions, T' is the number of successful transmissions observed by the terminal in the network,

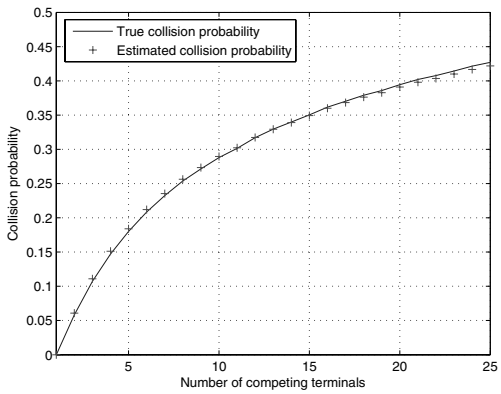

Fig. 5: Estimated collision probability using the collision factor for $U = 15$.

and γ is the collision factor. Note that $T' + C\gamma$ is the average number of transmission attempts in the network, and $C\gamma$ is, on average, the proportion of these transmissions resulting in collisions. As before, the measuring terminal may increase or decrease the observation interval T'. In our simulations we set $T' = 30$.

The estimate $\tilde{p}_c$ requires the use of a γ corresponding to the number of competing terminals U. While U can be estimated with techniques such as those in [6], in an IEEE 802.11 DCF network with $U \leq 25$ (note that these are terminals that are simultaneously sending at any given point in time, not the total number of terminals in the network), it is possible to select a fixed $\gamma = 2.14$, corresponding to $U = 15$, such that $\tilde{p}_c \approx p_c$ for any given U, as Fig. 5 shows. The small error in the estimation of the collision probability has virtually no effect on the false alarm probability of the detectors, and its simplicity comes at the cost of reducing the probability of detection, although only for misbehavior cases that are extremely close to the legitimate operation of the protocol, which are undoubtedly of less interest.

The noise in the estimate $\tilde{p}_c$ may overshoot p_c so that a legitimate terminal may appear as misbehaving. Because the cost of a false alarm is very high, we filter the data to reduce the noise, using a robust locally weighted polynomial regression model (rloess). Let $\tilde{p}_c^{(1)}, .., \tilde{p}_c^{(q)}$ be the sequence of collision probabilities estimated using (2) and filtered as indicated above. The cdf of the number of idle slots between successful transmissions for a legitimate saturating terminal in that period can be calculated as the average of the cdfs for each of the observed $\hat{p}_c^{(i)}$, i.e.,

$$\hat{F}_0 = \frac{1}{q} \sum_{i=1}^{q} F_0 \left(\tilde{p}_c^{(i)} \right), \tag{3}$$

where $F_0(p_c^{(i)})$ is the cdf of f_0 with collision probability $p_c^{(i)}$.

B. The Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov (K-S) test [7], is the most widely used goodness-of-fit test for continuous data. It is based on the empirical distribution function (edf), which converges uniformly almost surely to the real population cdf (Glivenko-Cantelli Theorem) [8]. The K-S test determines whether the underlying distribution f_1, from which samples are drawn, differs from an hypothesized distribution f_0. The K-S test compares the edf $\hat{F}_1$ obtained from the data samples with the hypothesized cdf F_0, and determines whether $F_1 = F_0$, or $F_1 < F_0$, or $F_1 > F_0$. For the misbehavior detection problem, we define the null hypothesis as the event where a node is not misbehaving, and hence we will use the one-sided test

$$\text{choose} \begin{cases} H_0 : F_1 \leq F_0 & \text{(not misbehaving)} \\ H_1 : F_1 > F_0 & \text{(misbehaving).} \end{cases} \tag{4}$$

Let $x_1, x_2, ..., x_K$ be the observations of the number of idle slots between successful transmissions from a terminal using an unknown strategy f_1. The edf of the observations is

$$\hat{F}_1(x) = \frac{1}{K} \sum_{i=1}^{K} \mathbb{1} \{x_i \leq x\}, \tag{5}$$

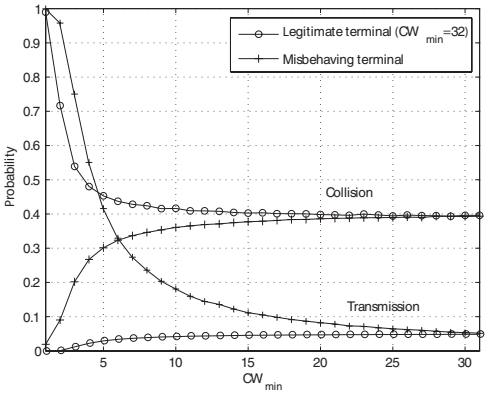

Fig. 6: Probability of transmission and collision when a misbehaving terminal uses IEEE 802.11 DCF with different $CW_{\min}$.

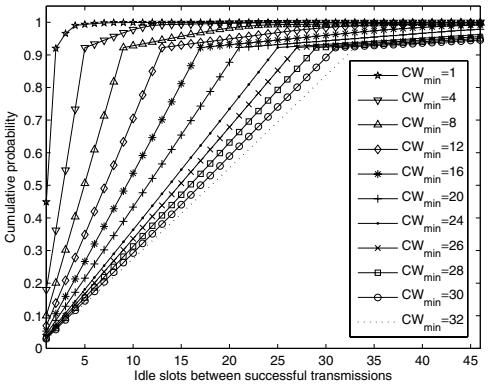

Fig. 7: Cdf of the number of idle slots between successful transmissions when a misbehaving terminal uses IEEE 802.11 DCF with different $CW_{\min}$.

where $\mathbb{1}(\cdot)$ is the indicator function.

The one-sided K-S test statistic D is defined as the maximum value of the difference between the two cdfs as

$$D \triangleq \max_{-\infty < x < +\infty} \{F_1(x) - F_0(x)\}, \tag{6}$$

and can be calculated as

$$\hat{D} = \max_{1 \leq i \leq K} \left\{ \hat{F}_1(x_i) - \hat{F}_0(x_i) \right\}, \tag{7}$$

where $\hat{F}_0$ and $\hat{F}_1$ are given respectively by (3) and (5).

Define [7]

$$\lambda(\hat{D}) = \max \left\{ \left(\sqrt{K} + 0.12 + \frac{0.11}{\sqrt{K}} \right) \hat{D}, 0 \right\}. \tag{8}$$

Then, the hypothesis H_0 is rejected at a significance level α if $P(D > \hat{D}) < \alpha$, where [9]

$$P(D > \hat{D}) = e^{-2\lambda(\hat{D})^2}. \tag{9}$$

We can now give the algorithm to test if a terminal is misbehaving (Algorithm 1): fixing the number of samples K, the measuring terminal calculates a new number x_i of idle slots since the last successful transmissions for each successful transmission of the observed terminal. Simultaneously, it calculates a new estimate of the collision probability $\tilde{p}_c^{(j)}$ using (2), every T' successful transmissions in the network (from *any* terminal). After the K-th successful transmission of the observed terminal, the algorithm uses the collected sequence of

idle slots between successful transmissions $x_1, ..., x_K$ and the calculated sequence of estimates of the probability of collision $\tilde{p}_c^{(1)}, .., \tilde{p}_c^{(q)}$, to perform the hypothesis test in (4) with a false alarm probability $P_{FA} = \alpha$.

Algorithm 1 K-S test for a fixed number of samples with $P_{FA} = \alpha$

1: Calculate K observations of the number of idle slots between transmissions of the observed terminal $x_1, ..., x_K$. Calculate the edf $\hat{F}_1$ from the samples of the observed terminal using (5).
2: Simultaneously, collect the estimates $\tilde{p}_c^{(1)}, .., \tilde{p}_c^{(q)}$ using (2), and filter as indicated in Section IV-A. Calculate the cdf of a legitimate terminal $\hat{F}_0$ using (3).
3: Perform the one sided K-S test for $\hat{F}_1 > \hat{F}_0$ and obtain the significance level P using (9).
4: **if** $P \leq \alpha$ **then**
5: reject H_0. The terminal is misbehaving.
6: **else**
7: do not reject H_0. The terminal is not misbehaving.
8: **end if**

The algorithm can be used by any terminal in the network. The access point (AP) can implement the algorithm for each terminal and take appropriate actions, such as disconnecting the offended terminal from the network. Terminals may also monitor their neighbors and implement the Nash equilibrium punishing strategy described in [5], to dissuade terminals from misbehaving.

V. Simulation Results

A. Simulation Setup and Performance Benchmarks

For the simulations we consider the IEEE 802.11 DCF where a legitimate terminal uses $CW_{\min} = 32$ and $CW_{\max} = 1024$. The simulations are performed using the ns-2 network simulator version 2.28 [10]. We modified the 802.11 implementation so the nodes measure the number of idle slots in the network. The parameters used in the simulation are typical for a 11 Mbps 802.11b WLAN. No packet fragmentation occurs, and the nodes are located close to each other to avoid capture or hidden terminal problems. The propagation delay is 1 μs. The packet size is fixed with a payload of 1024 bytes. The MAC and PHY headers use respectively 272 and 192 bits. The ACK length is 112 bits. The Rx/Tx turnaround time is 20 μs and the busy detect time 29 μs. The short retry limit and long retry limit are set to 7 and 4 retransmissions respectively. Finally, the slot time is 20 μs, the SIFS is 10 μs, and the DIFS is 50 μs. We implemented the detectors in MATLAB.

For simplicity, in our simulations the misbehaving terminals are assumed to use the binary exponential strategy with $CW_{\max} = 2^5 CW_{\min}$, and $CW_{\min} \in \{1, 2, ..., 32\}$. The case of $CW_{\min} = 32$ corresponds to the legitimate terminal. The case of $CW_{\min} = 16$ corresponds to the moderate misbehavior described in Section II. Finally the case of $CW_{\min} = 1$ corresponds to a case of extreme misbehavior. Fig. 6 show

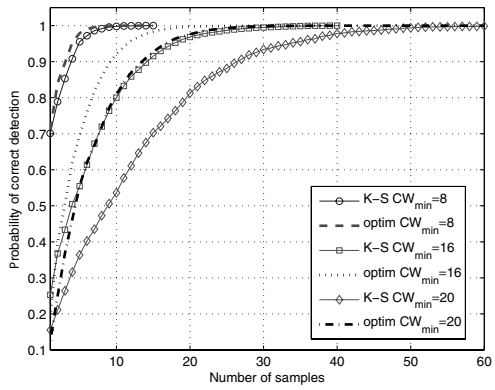

Fig. 8: Performance of the K-S detector vs. optimum detector when a misbehaving terminal uses IEEE 802.11 DCF with different $CW_{\min}$.

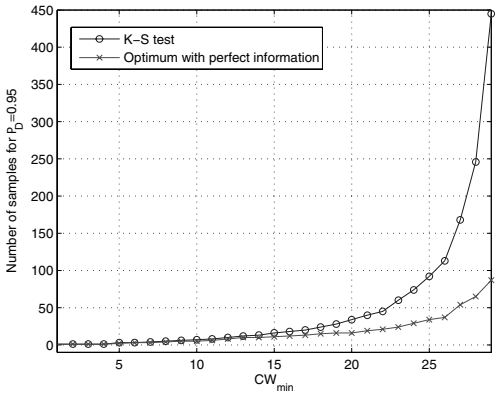

Fig. 9: Number of samples needed to detect a misbehaving terminal using IEEE 802.11 DCF with different $CW_{\min}$ and with $P_D = 0.95$.

the difference in probability of collision and probability of transmission observed by a selfish terminal using the above misbehavior strategies for a network with 20 terminals. Fig. 7 shows the cdf for some misbehavior strategies compared to the strategy of a legitimate terminal when $p_c = 0.1$. All these cases represent a good overview of the different intensity of misbehaviors, and provide a benchmark for the subjective performance of our algorithms (i.e., delay until making a decision). Note that for $CW_{\min} > 25$ the effect of misbehavior is minimal, so we are interested in a fast detection of the strategies with $CW_{\min} \leq 25$.

To the best of our knowledge there is no existing comparable robust method to detect a misbehaving terminal without modifications of the IEEE 802.11 DCF protocol. We decided to compare our method to the optimal Neyman-Pearson detector for the same $P_{FA} = \alpha$. While in practice the misbehavior strategy f_1 is not known, we can arbitrarily specify it in our simulations. The performance of the optimal detectors with known f_1 is an upper bound for the performance.

B. K-S Performance

Consider a network of 10 terminals. Fig. 8 show the probability of detection of our K-S detector, and the optimal Neyman-Pearson detector with perfect information for the misbehavior cases $CW_{\min} = 8$, $CW_{\min} = 16$ and $CW_{\min} =$

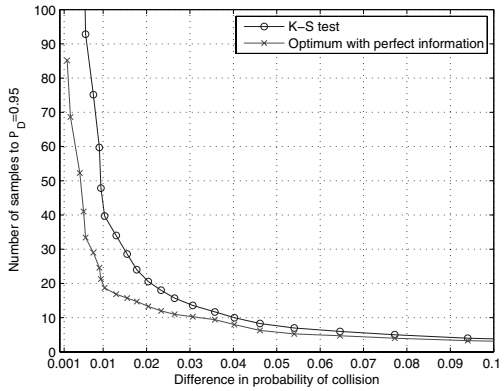

Fig. 10: Number of samples needed to detect a shift in collision probability with $P_D = 0.95$.

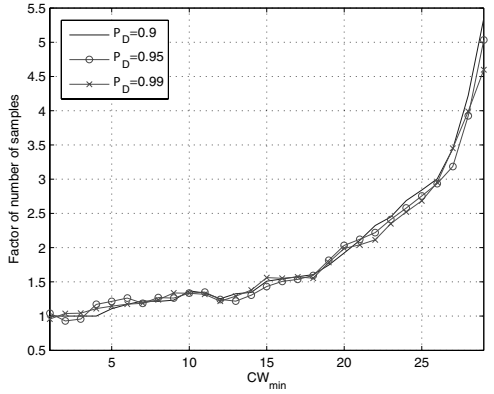

Fig. 11: Factor of the number of samples needed by K-S vs. optimum.

20, with $P_{FA} = 0.05$. The K-S detector is able to detect the misbehavior terminals very fast, requiring less than twice the samples needed by the optimum detector with perfect information.

Fig. 9 shows the number of samples needed to detect a misbehaving terminal for different $CW_{\min}$ strategies with $P_{FA} = 0.05$ and $P_D = 0.95$. Note that the performance of the K-S detector starts to degrade only for $CW_{\min} > 29$, which is very close to the strategy of a legitimate terminal.

A more interesting view of that result is shown in Fig. 10, that shows the number of samples needed to detect a difference in collision probability between the attacker and the legitimate terminal. The K-S test stays within the same performance range of the optimum test. In fact, performance only degrades for differences in collision lower than 0.005, and even in that case it only requires 100 samples for 95% detection probability. Note that we are more interested in the class of misbehavior that results in larger gains for the misbehaving terminal. Such a misbehavior would have the most devastating effects on the network, in the sense that it would deny channel access to the other terminals and would lead to unfair sharing of the channel [4]. Hence, it is more valuable, as our detector does, to perform faster for the more severe misbehaviors. Fig. 11 shows the factor of the number of samples that our K-S detector requires more than the optimum Neyman-Pearson test. Note that the performance of our detector is consistent as P_D increases.

Overall, the detection speed of the our K-S detector is high. Under good SNR conditions, a typical IEEE 802.11g network can deliver approximately 24Mbps to the upper layers [11], resulting in an approximate throughput of 2230 packets per second, assuming packets of 1400 bytes. On such a network, and taking into account the throughput of the misbehaving terminal for 10 competing terminals, our K-S algorithm is able to detect the $CW_{\min} = 29$ strategy in slightly less than 2 seconds, and all the misbehavior strategies $CW_{\min} < 29$ in less than a second. These times are comparable to the time a terminal needs to subscribe (and acquire an IP address) to an IEEE 802.11 network.

VI. Conclusions

We have proposed a method for detecting misbehaving terminals in a CSMA/CA network, based on measuring the number of idle slots between successful transmissions. The Kolmogorov-Smirnov (K-S) test is employed to determine whether the samples are consistent with the hypothesis that the terminal abides by the protocol rules. We have proposed a K-S detector to detect misbehaviors in the IEEE 802.11 DCF protocol. The performance obtained is close to that of the optimum detectors that assume perfect knowledge about the misbehavior strategy. The proposed technique is, to the best of our knowledge, the first robust misbehavior detector that can operate without modifying the protocol implementation.

References

[1] L. Buttyan and J. Hubaux, "Report on a working session on security in wireless ad hoc networks," *Mobile Computing and Communications Review*, vol. 6, no. 4, Nov. 2002.

[2] P. Kyasanur and N. H. Vaidya, "Selfish MAC layer misbehavior in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 5, pp. 502–516, 2005.

[3] 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.

[4] S. Radosavac, J. Baras, and I. Koutsopoulos, "A framework for MAC protocol misbehavior detection in wireless networks," in *WiSe '05: Proc. ACM Workshop on Wireless Security*, 2005, pp. 33–42.

[5] A. Lopez Toledo, T. Vercauteren, and X. Wang, "Adaptive optimization of IEEE 802.11 DCF based on bayesian estimation of the number of competing terminals," *IEEE Trans. Mobile Comput.*, vol. 5, no. 9, pp. 1283–1296, Nov. 2006.

[6] T. Vercauteren, A. Lopez Toledo, and X. Wang, "Batch and sequential bayesian estimators of the number of active terminals in an IEEE 802.11 network," *IEEE Trans. Signal Processing*, vol. 55, no. 2, pp. 437–450, Feb. 2007.

[7] F. Massey, "The Kolmogorov-Smirnov test for goodnes of fit," *J. Amer. Stat. Assoc.*, vol. 46, no. 253, pp. 68–78, 1951.

[8] H. Khamis, "The δ-corrected kolmogorov-smirnov test for goodness of fit," *J. Statistical Planning and Inference*, vol. 24, pp. 317–335, 1990.

[9] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 1992.

[10] S. McCanne and S. Floyd, network simulator 2. http://www.isi.edu/nsnam/ns.

[11] K. Medepalli, P. Gopalakrishnan, D. Famolari, and T. Kodama, "Voice capacity of ieee 802.11b, 802.11a and 802.11g wireless LANs," in *Proc. 2004 IEEE Globecom*, 2004, pp. 1459–1553.