Detecting Selfish Configurations in 802.11 WLANs

Pablo Serrano, *Member, IEEE,* Albert Banchs, *Member, IEEE,* Valerio Targon, and José Félix Kukielka

***Abstract*—Lately, there has been an increase in the number of IEEE 802.11 devices that provide users with the ability to modify the MAC parameters or do not conform to the standard specification. This increases the risk of having a WLAN with *selfish* stations that, through the CSMA/CA parameters, obtain a larger share of the resources at the expense of well-behaved users. In this letter we propose a mechanism to detect these selfish stations that, unlike previous approaches, is not based on heuristics nor makes any assumption about radio conditions.**

***Index Terms*—Detection, Selfish, Malicious, WLAN, 802.11**

I. Introduction

THE EDCA mechanism of IEEE 802.11e standard [1] extends the former DCF mechanism through the generalization of the MAC parameters. As these parameters control the behavior and randomness of stations when accessing the channel, EDCA supports statistical service differentiation and QoS provisioning. Nowadays there are many WLAN devices that do not fully support the EDCA mechanism, but still implement to some extent the ability to change configuration of the MAC parameters (e.g., [2]). Furthermore, even (assumed) 802.11-compliant devices have recently been reported [3] to deviate from the standard specification, leading to throughput asymmetries and unfairness. We claim that, because of the above two reasons, a mechanism to detect *selfish* configurations that try to get a larger share of throughput is needed.

Despite these risks of selfish and unfair behavior in WLANs, the design of an effective detection mechanism has received little attention. We classify the main contributions in two groups: i) changes to the MAC protocol [4], [5] that require extending the EDCA mechanism and, therefore, are of limitted applicability; and ii) detection mechanisms [6]–[8] that, based on an observed behavior, decide if a given station is acting selfishly or not. In this letter we propose a simple and robust mechanism of this second category that addresses the weaknesses of previous approaches as follows:

- DOMINO [6] is a heuristic-based approach not supported by analytical results with no means to design the trade-off between detection and false alarm probabilities.
- The approach of [7] is built on top of some strong radio assumptions that leads to unexpected poor performance for realistic scenarios.
- Our previous work of [8], based on the sampling distribution of the mean, does not take full advantage of the statistical information available and requires an optimally configured WLAN to maximize its performance.

In contrast to these, in this letter we propose a robust scheme to detect selfish configurations of standard-compliant stations that i) it is not based on heuristics, ii) it does not make any strong assumption about the scenario, and iii) it does not require the estimation of any performance parameter.

Manuscript received MONTH DAY, YEAR. The associate editor coordinating the review of this letter and approving it for publication was Dr. Chadi Assi. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^0 214994.

P. Serrano and A. Banchs are with the Universidad Carlos III de Madrid, 28911 Madrid, Spain (email: {pablo,banchs}@it.uc3m.es).

A. Banchs is also with IMDEA Networks, Avda. del Mar Mediterráneo, 28918 Madrid, Spain.

V. Targon and J. F. Kukielka are with IMDEA Networks (email: {valerio.targon,josefelix.kukielka}@imdea.org).

Digital Object Identifier XXX

II. Detecting Selfish EDCA Configurations

The EDCA mechanism is a CSMA/CA based protocol that uses channel sense to prevent simultaneous transmissions and a binary exponential backoff to react to collisions. According to the 802.11e standard, the Access Point (AP) broadcasts the values of the MAC parameters to use through beacon frames, controlling in this way the behavior of WLAN stations when contending for channel access. These parameters are:

- The transmission opportunity (TXOP), that controls the maximum time a station is allowed to spend sending data frames once channel access is granted.
- The arbitration interframe space (AIFS), i.e., the time a station has to wait once the channel is sensed idle before sending a frame or reactivating the backoff process.
- The minimum and maximum contention window (CW_{min} and CW_{max}, respectively), that control the randomness of the backoff mechanism.

Misconfigurations of the AIFS or TXOP parameters are easy to detect as they impose deterministic rules. Therefore, the challenge lies in the randomness of the backoff mechanism ruled by the CW parameters. We focus on the detection of selfish configurations of the CW_{min} parameter, because we argue it is the parameter most likely to be tuned by a selfish user: in a properly configured EDCA WLAN the collision probability will be very small, and therefore the gain from misconfigurations of the CW_{max} parameter will be small[1].

We base our algorithm on the following observation. In order to prevent duplicates, the 802.11 standard uses a *retry bit* to mark those frames that are being retransmitted, i.e., the flag is set to 0 on the first attempt, and set to 1 on every other transmission (see Fig. 1). This way, for the case of a station always backlogged[2], the number of slots between two successfully received frames is uniformly distributed between 0 and CW_{min} if the retry bit of the second frame is set to 0[3].

[1]Using a 2-laptop testbed we confirmed that setting $CW_{max} = CW_{min}$ results in a throughput gain of only 3%.

[2]Our algorithm aims at detecting configurations that obtain more bandwidth than a well-behaved and constantly backlogged one would get.

[3]While changing the CW_{min} is easily done through a function call with commodity hardware, changing the retry bit requires the use of low-level firmware functions and therefore it can be assumed users cannot forge it.

Based on the this, our algorithm works as follows. During each observation interval T, a controller station monitors all the successful transmissions from a station under supervision, counting the number of timeslots between them. When a received frame has the retry bit set to 0, the controller adds that sample x_i to the set of collected samples. Once the observation interval is finished, a test is performed on the K collected samples to test if they were drawn from a uniform distribution between 0 and CW_{min} or not. More specifically, since we are interested in detecting selfish behaviours, we use a one-side test with the following null hypothesis

$$H_0 : F(x) \leq \mathcal{U}(CW_{min}), \text{for all } x \quad (1)$$

where $F(x)$ is the unknown distribution function of the K samples, and $\mathcal{U}(CW_{min})$ is the cumulative distribution function (cdf) of a uniform variable between 0 and CW_{min}. For this goodnes-of-fit test we use the one-side Kolmogorov-Smirnoff (K-S) test [9] as follows. First the empirical cdf $S_K(x)$, is built

$$S_K(x) = \frac{1}{K} \sum_{i=1}^{K} \mathbb{1}(x_i \leq x) \quad (2)$$

where $\mathbb{1}$ is the indicator function. Then, the maximum difference D between the two cdfs is estimated through

$$\hat{D} = max_i \{S_K(x_i) - \mathcal{U}(CW_{min})\} \quad (3)$$

and finally the significance level of the observed value $\hat{D}$ (i.e. the disproof of the null hypothesis) is approximated by [10]

$$P(D > \hat{D}) = e^{-2\lambda(\hat{D})^2} \quad (4)$$

where

$$\lambda(\hat{D}) = \left(\sqrt{K} + 0.12 + \frac{0.11}{\sqrt{K}}\right) \hat{D} \quad (5)$$

Therefore the hypothesis H_0 is rejected at a significance level α if $P(D > \hat{D}) < \alpha$, this way supporting the tune of the false alarm probability P_{FA}[4]. Note that, although [7] also uses a K-S test on the sample distribution of timeslots, there are at least two major differences between the two approaches:

1) Our proposal does not require the estimation of any WLAN parameter: in [7], authors have to compute the so called *collision factor* γ (the average number of stations involved in a collision), and then use a polynomial regression model to estimate the collision probability $\hat{p_c}$.
2) Our proposal does not make any assumption about the radio conditions. In [7], authors assume there are no losses due to noise and that in case of a collision all frames are lost. However, this is not the case for real WLANs, where the *capture effect* (in case of a collision, one of the frames may get through due to its larger power) is quite common –see, e.g., [2].

Since our approach only considers the number of slots between two consecutive successful receptions when the second frame has the retry bit set to 0, we release the assumption on the uniformity of the radio conditions. This way we achieve a two-fold objective: first, for realistic WLAN scenarios, we prevent a large false alarm rate (as we will see in the next section); second, our algorithm is simpler and better suited for a low-capacity device (e.g. an Access Point).

[4]Note that the standard K-S test is accurate only for continuous distributions, and known to be conservative for the discrete case [11]. Nevertheless, for simplicity we will assume (following [7]) that (4) leads to accurate results.

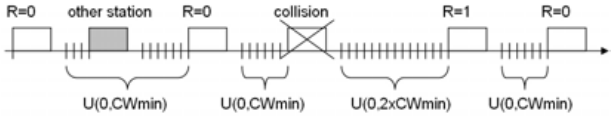

Fig. 1. Use of the retry bit $R = 0$ of frames from the station under supervision to collect backoff decrements in the $(0, CW_{min})$ range.

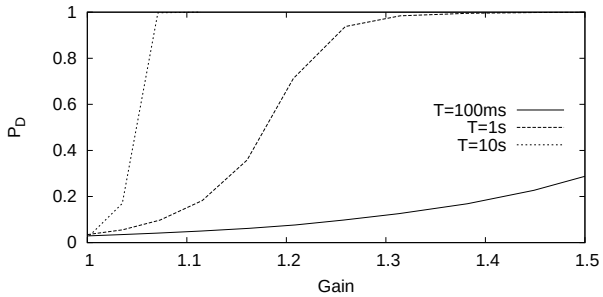

Fig. 2. Probability of detection vs. selfish gain

III. Performance Evaluation

We asses the effectiveness of our proposal to detect selfish configurations by means of simulations. We first consider a WLAN scenario with an AP and $N = 10$ stations. Stations use the parameters of the 802.11b physical layer (in particular, $CW_{min} = 32$) and always have 1500-byte frames ready for transmission. The AP runs our detection algorithm every T seconds, while the probability of false alarm P_{FA} is set through a significance level of $\alpha = 0.05$. To compute the probability of detection P_D, we assume one of the users reduces his CW_{min} parameter and run simulations for more than 20k observation intervals. We also compute the gain the selfish user gets over the rest of the users of the WLAN, to quantify the *threat* and relate it to the detection probability:

$$Gain = R_{sel}/R_{well}$$

where R_{sel} and R_{well} are the throughput experienced by a selfish and a well-behaved user, respectively.

Results for P_D vs. gain are depicted in Fig. 2 for different values of T. Note that the case $Gain = 1$ corresponds to the case when the user is well behaved ($CW_{min} = 32$), so in this case P_D corresponds to P_{FA}. The results can be summarized as follows. First, the typical *beacon interval* ($T = 0.1s$) is not well suited to detect malicious configurations, even when the selfish user is getting more than 1.5 times the bandwidth of a well behaved user. Therefore policy decisions cannot be taken in a beacon time, but rather some memory is needed to achieve enough certainty. In case the timescale is $T = 1s$, a selfish user may get around 20% more bandwidth than a regular user before being detected with a 0.5 probability, a result that quantifies the trade-off between detection certainty

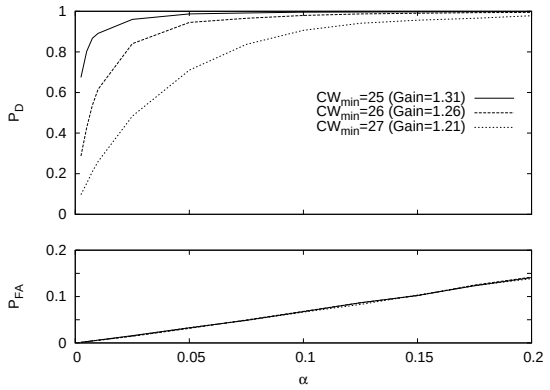

Fig. 3. Impact of α on P_D and P_{FA}

and unfairness risk. Only for very large intervals ($T = 10s$) a selfish user will be practically always detected before getting more than 10% the bandwidth of a regular user.

To analyze the trade-off between the P_D and P_{FA} we now supervise a selfish and a well-behaved station, and plot in Fig. 3 the resulting probabilities for different values of α and gain (we set $T = 1s$). Considering the gain a selfish user may get, results show there is little advantage in using values of $\alpha > 0.10$, as the growth of P_D is not compensated by the one of P_{FA}. Note that the P_{FA} values are quite similar for the three cases, and always below α –a result expected because of the discrete nature of $\mathcal{U}(CW_{min})$ [11].

Next we compare our approach against previous proposals to detect selfish CW_{min} values. We first want to assess the extent to which realistic radio conditions impact detection performance. To this aim, we assume that all the N stations are well-behaved and one is closer to the AP, this resulting in a capture effect that benefits this station as follows: colliding frames from this station are successfully received with a probability p_c, while the other(s) transmission(s) are lost.

We set $T = 1s$ and $\alpha = 0.05$ and count the number of times the detection algorithm (wrongly) classifies a behavior as selfish. Results are presented in Table I for the algorithm of [7] (TLW) and the one presented in this letter (Ours). We perform the test when the station is near and far from the AP ($near$ and far, respectively), and for p_c ranging from 0 (no capture) to 1 (the frame from the near station always captures the medium). The results show that the assumptions made in [7] lead to quite low performance if a station benefits from the radio conditions. More specifically, the TLW mechanisms largely deviates from the target P_{FA} if a station captures the channel in just 25% of the collisions, leading to $P_{FA} = 0.22$. If the station is so close to the AP that it captures the channel in 75% of the collisions, the TLW mechanism will mark it as misbehaving with practically no doubt ($P_{FA} = 0.999$). We conclude that the TLW algorithm is poorly suited for realistic scenarios, while our proposal is oblivious to radio conditions, with practically the same results for the $near$ and far case.

Lastly, we compare the mechanism proposed against our previous proposal based on the Central Limit Theorem (CLT) [8] and DOMINO [6]. To that aim we use the same scenario with one selfish user, and compare the minimum time needed to obtain a $P_D \geq 0.90$ for the same P_{FA} and different

TABLE I
IMPACT OF RADIO CONDITIONS ON P_{FA}

p_c	Ours		TLW	
	$near$	far	$near$	far
0.00	0.034	0.032	0.032	0.032
0.25	0.032	0.032	0.220	0.029
0.50	0.034	0.032	0.783	0.027
0.75	0.030	0.032	0.999	0.025
1.00	0.032	0.032	1.000	0.024

TABLE II
TIME REQUIRED FOR $P_D \geq 0.90$, $P_{FA} = 0.05$

N	CW_{min}	Gain	Ours [s]	CLT [s]	DOMINO [s]
5	30	1.07	3.3	6.0	11.9
	28	1.16	0.9	1.4	2.8
	26	1.26	0.4	0.6	1.3
10	30	1.07	8.1	14.1	> 60
	28	1.16	2.1	3.2	30.0
	26	1.26	1.0	1.4	12.6
20	30	1.07	20.6	36.2	> 60
	28	1.15	5.3	8.3	> 60
	26	1.25	2.4	3.4	> 60

values of N. Results, in Table II, show that the K-S test outperforms both proposals, with average time savings of 36% compared to CLT and more than 80% compared to DOMINO[5]. These time savings are caused by i) the use of more statistical information, i.e., the cdf of the random variable, and ii) the ability to collect more samples by looking at the retry bit.

As compared to previous work, then, ours is an effective approach well suited to be implemented in real devices, due to its analytical foundations, the absence of assumptions about radio conditions, and its low complexity.

REFERENCES

[1] IEEE 802.11e, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*. Supplement to IEEE 802.11 Standard, November 2005.

[2] A. Banchs, A. Azcorra, C. García, and R. Cuevas, "Applications and Challenges of the 802.11e EDCA Mechanism: An Experimental Study," *IEEE Network Magazine*, vol. 19, pp. 52 – 58, July and August 2005.

[3] G. Bianchi, A. Di Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello, "Experimental Assessment of the Backoff Behavior of Commercial IEEE 802.11b Network Cards," in *IEEE INFOCOM*, 2007.

[4] A. A. Cárdenas, S. Radosavac, and J. S. Baras, "Detection and Prevention of MAC Layer Misbehavior in Ad Hoc Networks," in *SASN '04*. New York, NY, USA: ACM, 2004, pp. 17–22.

[5] P. Kyasanur and N. H. Vaidya, "Selfish MAC Layer Misbehavior in Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 5, pp. 502–516, 2005.

[6] M. Raya, I. Aad, J.-P. Hubaux, and A. E. Fawal, "DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1691–1705, 2006.

[7] A. L. Toledo and X. Wang, "Robust Detection of Selfish Misbehavior in Wireless Networks," *IEEE JSAC*, vol. 25, no. 6, pp. 1124–1134, 2007.

[8] P. Serrano, A. Banchs, and J. Kukielka, "Detection of Malicious Parameter Configurations in 802.11e EDCA," *IEEE GLOBECOM*, 2005.

[9] F. J. Massey Jr., "The Kolmogorov-Smirnov Test for Goodness of Fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[10] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge University Press, 1992.

[11] W. J. Conover, "A Kolmogorov Goodness-of-Fit Test for Discontinuous Distributions," *Journal of the American Statistical Association*, vol. 67, no. 339, pp. 591–596, 1972.

[5]Note that DOMINO does not provide a way to set the α value for a desired P_{FA}, so we had to run a numerical search per CW_{min} value to tune it.