This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

Detection of Malicious Parameter Configurations in 802.11e EDCA

Pablo Serrano, Albert Banchs and José Félix Kukielka
Universidad Carlos III de Madrid, Spain
E-mail: {pablo,banchs,kukielka}@it.uc3m.es

***Abstract*— The service quality experienced by a user with the EDCA mechanism of the upcoming 802.11e standard depends on a number of configurable parameters, namely CW_{min}, CW_{max}, $AIFS$ and $TXOP\_limit$. WLAN stations are supposed to use the parameter configurations that the Access Point (AP) of the WLAN distributes with the beacon frames. However, a user can maliciously configure the parameters of his WLAN station in order to obtain a better service. In this paper, we address the issue of detecting malicious parameter configurations of EDCA. The $AIFS$ and $TXOP\_limit$ parameters are relatively easy to control because of their deterministic nature. Given the capture effect, the CW_{max} parameter may be unused by some users. Therefore, the key challenge lies in detecting if the CW_{min} parameter of a station is well configured. The main contribution of this paper is the proposal of an algorithm to detect malicious CW_{min} configurations. We show that, for an optimally configured WLAN, our algorithm is effective in avoiding that a user can substantially benefit from maliciously configuring his WLAN station without being detected.**

I. Introduction

In recent years, much interest has been devoted to the design of wireless local area networks (WLANs) with Quality of Service (QoS) support. The Enhancements Task Group (TGe) was formed under the IEEE 802.11 project to recommend an international WLAN standard with QoS. This standard is called 802.11e and is being built as an extension of the basic WLAN 802.11 standard. While the standardization process of 802.11e is still ongoing, the main features of the upcoming standard have already been agreed upon and are unlikely to change. These features are described in the latest version of the 802.11e standard draft [1], where two different access mechanisms are defined: the *Enhanced Distributed Channel Access* (EDCA) and the *HCF Controlled Channel Access* (HCCA). The focus of this paper is on the former.

With EDCA, the service received by a station depends on a number of parameters, namely CW_{min}, CW_{max}, $AIFS$ and $TXOP\_limit$. These are open parameters that can be configured to different values for each station. According to the standard draft, the values of these parameters are distributed by the Access Point (AP) with beacon frames, and stations are configured with the received values. In this way, the AP can control the service quality provided to each station, based e.g. on the service contracted by the user.

One of the main challenges that still need to be addressed with EDCA is to avoid that a user can obtain a better service by maliciously changing the parameters of his WLAN station to different values from the ones assigned by the AP. Indeed, with current WLAN products with EDCA support, it is very easy for a user to modify his EDCA parameters [2] and thus obtain throughput advantages from the WLAN.

In this paper, we address the issue of detecting if the EDCA parameters of a WLAN station are maliciously configured. Because of the random nature of EDCA, there is always some probability that a false alarm occurs in the detection (i.e. that a well-behaved user is erroneously detected as malicious). This false alarm probability is taken as an input parameter to the proposed detection algorithm. Specifically, given a fixed observation time interval there exists a tradeoff between accuracy and false alarm probability in our algorithm: the lower the additional service a malicious user can receive without being detected, the higher the probability of erroneously detecting a well-behaved user as malicious.

The issue of detecting malicious EDCA configurations has been previously addressed in the literature [2], [3]. The main novelty of our algorithm with respect to those works lies in the probabilistic nature of our analysis. In particular, while our algorithm can be tuned to detect malicious configurations with a desired false alarm probability, the previously published mechanisms have been designed heuristically and give only an "indication" of malicious configuration, and the probability with which this indication can be trusted is not known.

We note that the focus of this work is on the *detection* of malicious configurations, and that the *reaction* adopted upon detecting a malicious configuration is out of the scope of the paper. One possible reaction could be closing the user's connection to the WLAN. Another could be dropping the excess traffic sent by a user during an observation interval in the next interval. Obviously, it is important to be able to set the false alarm probability depending on the reaction adopted; typically, the more severe the reaction, the lower the probability of a false alarm, in order to avoid punishing a well-behaved user.

The paper is outlined as follows. In Section II we briefly review the EDCA mechanism of the 802.11e standard draft. In Section III we identify which are the challenges for the

The work described in this paper is based on results of OPTINET6 and DAIDALOS. OPTINET6 receives research funding from the Spanish Ministry of Science and Technology. DAIDALOS receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper.

detection of malicious configurations. In Section IV we present our algorithm for detecting malicious configurations. We argue that, because of the *capture effect*, which occurs often in WLANs, our algorithm is specially effective when the optimal configuration of Section V is used. Simulation results in Section VI show that our algorithm, when combined with the optimal parameter configuration, effectively prevents that a user can unfairly gain substantial service improvements. Finally, our conclusions are given in Section VII.

II. 802.11E EDCA

This section briefly summarizes the EDCA mechanism that regulates the access to the wireless channel as defined in the 802.11e standard draft [1].

A station with a new frame to transmit monitors the channel activity. If the channel is idle for a period of time equal to the arbitration interframe space ($AIFS$), the station transmits. Otherwise, if a transmission is detected on the channel (either immediately or during the $AIFS$), the station continues to monitor the channel until it is measured idle for an $AIFS$ (if the transmission is correct) or an $EIFS - DIFS + AIFS$ (otherwise), and, at this point, the backoff process starts. The arbitration interframe space $AIFS$ takes a value of the form $DIFS + n\sigma$, where n is a nonnegative integer.

Upon starting the backoff process, the station computes a random value uniformly distributed in the range $(0, CW - 1)$, and initializes its backoff time counter with this value. The CW value is called contention window, and depends on the number of failed transmissions for the frame. At the first transmission attempt, CW is set equal to a value CW_{min}, called minimum contention window.

The backoff time counter is decremented once every time interval σ as long as the channel is sensed idle, "frozen" when a transmission is detected on the channel, and reactivated when the channel is sensed idle again for an $AIFS$ (if the transmission is correct) or an $EIFS - DIFS + AIFS$ (otherwise). As soon as the backoff time counter reaches zero, the station transmits its frame. A collision occurs when two or more stations start transmission simultaneously. An acknowledgement (Ack) frame is used to notify the transmitting station that the frame has been successfully received.

If the Ack is not received within a specified Ack Timeout, the station assumes that the transmitted frame was not received successfully and schedules a retransmission reentering the backoff process. After each unsuccessful transmission, CW is doubled, up to a maximum value CW_{max}. If the number of failed attempts reaches a predetermined retry limit, the frame is discarded.

After a (successful or unsuccessful) frame transmission, before transmitting the next frame, the station must execute a new backoff process. As an exception to this rule, the protocol allows the continuation of an EDCA transmission opportunity ($TXOP$). A continuation of an EDCA $TXOP$ occurs when a station retains the right to access the channel following the completion of a successful transmission. In this case, the station transmits a new frame at a SIFS period following the completion of the successful transmission. The period of time a station is allowed to retain the right to access the channel is limited by the parameter $TXOP\_limit$.

As it can be seen from the description of EDCA given in this section, the behavior of a station depends on a number of parameters, namely CW_{min}, CW_{max}, $AIFS$ and $TXOP\_limit$. These are configurable parameters that can be set to different values for each station. According to the standard draft, stations are grouped by Access Categories (AC's), all the stations of an AC having the same configuration, and the configuration of all the AC's is distributed by the AP with beacon frames.

III. On the detection of malicious EDCA configurations

Detecting malicious EDCA parameter configurations is important in order to avoid rogue wireless hosts to unfairly obtain a better service than the one they are entitled to. Indeed, current EDCA compatible wireless cards[1] already allow manually configuring the EDCA parameters, and therefore it will not be possible to prevent users in future EDCA WLAN's from maliciously configuring their EDCA parameters, using these or future wireless cards that also allow manual configuration.

In this section we address the issue of detecting a user that is obtaining additional throughput from the network by maliciously configuring his EDCA parameters. To detect misbehaviour with a high degree of accuracy, we need to measure the relevant EDCA metrics whithin a configurable observation interval.

For the $AIFS$ and $TXOP\_limit$ parameters, because of their deterministic nature, it is possible to detect their malicious configuration simply by observing the controlled user's transmission patterns: if the value of $AIFS$ would be, for example, $DIFS + 2\sigma$, and the controlled station ever transmits at $DIFS + \sigma$ resulting in an earlier than expected transmission, or the length of its transmission is greater than $TXOP\_limit$, then we can be sure that the parameters have been misconfigured. Without loss of generality, in the rest of the paper we assume that all stations are configured with $AIFS = DIFS$ and transmit a single packet of fixed length when they access the channel.

Detecting a maliciously configured CW is more difficult as the behavior of this parameter is random. Therefore the detection in this case will always involve a certain error probability. Indeed, backoff times are computed from a random distribution between 0 and the $CW_{min} - 1$ and therefore, there is always some chance that a well configured station draws small backoff times from this uniform distribution and, as a result, obtains more service over a certain observation interval.

In addition to the above, one effect that needs to be considered when analyzing the behavior of a user is the so called *capture effect*. In case of a collision between two frames, in some occasions we have that the frame received

[1] Atheros chipset based WLAN cards implement a subset of EDCA and allow manually configuring the EDCA parameters. The reader is referred to [2] for the details on this manual configuration.

with the strongest signal survives the collision and it *captures* the channel. Note that, in this case, the collision results in a successful transmission of the winning station, which does not increase its CW in the next backoff process. Indeed, in our experimental study of [4] we found that the capture effect occurs relatively frequently in a WLAN environment.

Note that the worst case for the controlled user (i.e. the case in which he obtains the highest throughput) is when the user continuously has packets ready for transmission and always captures the channel in case of a collision. If the user obtains more throughput than he should given these conditions, we can be sure he is behaving maliciously. In the following section, we present an algorithm that detects if a user under these conditions is misbehaving. Simulation results in Section VI show that this algorithm is also effective for controlling any other user.

IV. Detection of a misconfigured CW_{min} including capture effect

According to the explanations provided in the previous section, the main challenge in the detection of malicious parameter configurations lies in finding if the CW_{min} of a station that always captures the channel in case of a collision is misconfigured. We now present an algorithm that achieves this goal.

Our algorithm can be executed by any station that listens to the controlled WLAN, and this station can monitor any subset of the stations sending to the WLAN. Specifically, the algorithm can be executed at the Access Point (AP) to control all the other stations in the WLAN. In the remaining of this paper we assume, without loss of generality, that there exists one controlling station and one controlled station.

The algorithm works as follows. The channel is monitored over observation intervals of a configurable duration T_{obs}. The objective is to find, from the behavior of the controlled station in the interval, if it is maliciously configured. In order to achieve this objective, our algorithm measures the following two metrics in the time interval:

- N: the number of slot times contained in the interval, where a slot time is defined as the interval between two backoff counter decrements. Note that, since the controlling station already decrements its backoff counter in order to transmit packets, counting the number of slot times in an interval does not introduce any additional complexity.
- S: the number of successful transmissions of the controlled station in the time interval.

We observe that, as a station waits for a number of slot times uniformly distributed in the range $(0, CW_{min} - 1)$ between each successful transmission, the total number of slot times for a controlled station that has performed S successful transmissions in the time interval follows the random variable (see Fig. 1),

$$N_{total} = \sum_{i=1}^{S} Unif(1, CW_{min}) \tag{1}$$

and the average number of slot times between one success and the next one follows the random variable,

$$N_{avg} = \frac{1}{S} \sum_{i=1}^{S} Unif(1, CW_{min}) \tag{2}$$

Note that, since N_{avg} is the mean of a large number of independent random variables, according to the Central Limit Theorem it closely follows a gaussian distribution, whose mean and typical deviation can be computed as follows,

$$m = \sum_{i=1}^{CW_{min}} \frac{i}{CW_{min}} = \frac{CW_{min} + 1}{2} \tag{3}$$

$$\sigma^2 = \frac{1}{S} \left(\sum_{i=1}^{CW_{min}} \frac{i^2}{CW_{min}} - \left(\frac{CW_{min} + 1}{2} \right)^2 \right) \tag{4}$$

If the CW_{min} of the station is properly configured, the measured number of slot times in the time interval divided by the number of successes of the controlled station (N/S) will be a sample of the random variable N_{avg}. The probability that this sample takes a value below a given threshold $m - K\sigma$ can easily be computed from the gaussian distribution,

$$P(N_{avg} < m - K\sigma) = 0.5\, erfc \left(-\frac{K}{\sqrt{2}} \right) \tag{5}$$

Notice that, for a user that has configured his CW_{min} to a smaller value in order to obtain greater shares of throughput, the measured N/S will typically be smaller than it should. Based on this, our algorithm proceeds as follows to detect if the CW_{min} of a station is misconfigured. Given S, m and σ are computed from Eqs. (3) and (4). Then, N/S is compared against $m - K\sigma$. If $N/S < m - K\sigma$, the algorithm identifies the user as malicious. Otherwise, the user is considered well-behaved.

The above algorithm is based on the following parameters:

- Probability of erroneously detecting a well-behaved user as malicious, referred to as the *probability of false alarm*.
- Length of the observation interval. This parameter relates to the *time granularity*. Its length should be small enough to avoid that a user can benefit from misbehaving over short time scales.
- K. This parameter relates to the *accuracy* of the algorithm; the lower K, the smaller the additional throughput a user can gain without being detected.

Note that the above parameters are interrelated (setting two of them, the third is given) and they have to be traded off:

- The smaller the probability of false alarm, the lower the accuracy. Indeed, for the probability of false alarm to be small, K needs to be higher and therefore the difference between the expected throughput and the throughput above which we detect the user as malicious is larger. This tradeoff is illustrated by Fig. 2.
- The larger the length of the observation interval, the higher the accuracy. In fact, for a larger length, we have a larger S and therefore a smaller σ (see Eq. (4)), which yields a higher accuracy (for the same probability of false alarm). This is also illustrated in Fig. 2.

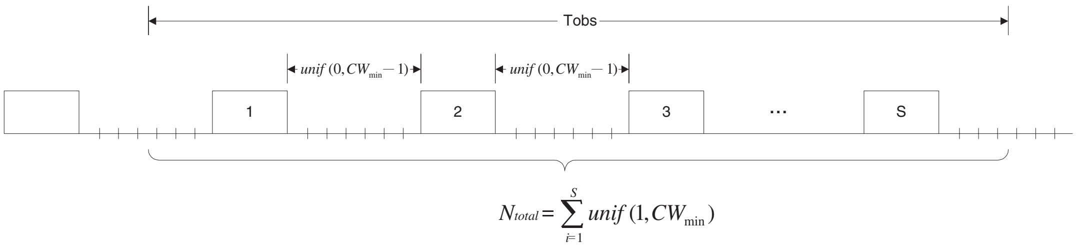

Fig. 1. N_{total} vs. S

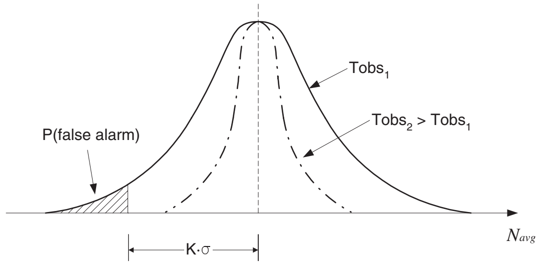

Fig. 2. Relationship between probability of false alarm, K and observation interval length.

V. Optimal Configuration of the CW_{min} and CW_{max} Parameters

The problem of assuming that the station always captures the channel in case of collision is that, if the configuration is such that collisions occur often, a malicious user that is not capturing the channel may go undetected, as he may be obtaining a throughput gain similar to the one he would obtain by capturing the channel.

In order to avoid the above problem, we propose to use the configuration given in [5] for the CW_{min} and CW_{max} parameters. This configuration has been computed to maximize the throughput performance of the WLAN, and it is characterized by a very small probability of collision, which in turn reduces considerably the impact of the capture effect. As a result, a user cannot take advantage of the capture effect to gain substantial throughput.

Specifically, [5] proposes to set the CW_{min} and CW_{max} to the following value for all stations:

$$CW_{max} = CW_{min} = \frac{2n^2(n-1)(T_c-\sigma)}{\sqrt{(n(n-1)\sigma)^2 + n^3(n-1)(T_c-\sigma)\sigma} - n(n-1)\sigma} \tag{6}$$

where σ is the duration of an idle slot time, T_c is the duration of a slot time that contains a collision and n is the number of stations in the WLAN.

VI. Simulation Results

In this section, we evaluate via simulation the effectiveness of our algorithm to detect misconfigured users, with an event-driven simulator that closely follows the EDCA protocol.

The values of the various system parameters used in the simulations have been taken from the 802.11b standard. All simulations are performed for a packet length of 1500 bytes and a number of stations $n = 10$.

A malicious user may misbehave in different ways to gain more throughput. The behavior that we have chosen here is the following. The malicious user always has packets ready for transmission, which provides him with the largest possible throughput given his EDCA configuration. The EDCA parameters are maliciously set as follows: CW_{min} and CW_{max} are set to the same value (such that the CW is not doubled after a collision), this value being smaller than the CW_{min} of the other stations. The stations other than the malicious one are configured with the correct EDCA parameters. We assume that these stations also always have packets ready for transmission.

For each of the different scenarios studied, 100.000 observation intervals are analyzed. The reported probability of detection is obtained from measuring in how many intervals out of these 100.000 the algorithm detects the user as malicious.

In order to better understand the impact of the various parameters on our algorithm, we performed a number of experiments for different values of K and T_{obs}. In these experiments, all the users but the controlled one implement the default CW configuration of 802.11 DCF ($CW_{min} = 32$ and $CW_{max} = 1024$). The controlled user always captures the channel and uses the malicious configuration described above. We study the probability of detection as a function of the CW_{min} of the controlled user. Note that, when $CW_{min} = 32$, the controlled user can be considered as well-behaved, as a station that always captures the channel never doubles its CW.

Table I gives the probability of false alarm (i.e. the probability of detecting a well configured controlled user as a malicious one), for different values of K and T, both according to simulations and analysis (computed from Eq. (5)). Results confirm that our algorithm allows tuning the probability of false alarm, since in all cases the analytical computation of this probability is very close to the simulation outcome. Note that the difference between simulation and analysis decreases with increasing T_{obs}. In fact, the greater the observation interval, the more random variables are included in the sum of Eq. (2), and, according to the Central Limit Theorem, the more accurate our gaussian approximation becomes.

For the rest of this section we set $K = 2$ and $T = 5s$ and perform an evaluation of the detection algorithm in three

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

TABLE I

PROBABILITY OF FALSE ALARM

K	T(s)	P(false alarm) analysis	P(false alarm) simulations
1	5	0.1587	0.1567
2	1	0.0228	0.0218
∥	5	∥	0.0224
∥	10	∥	0.0227
3	5	0.0013	0.0012

TABLE II

RESULTS FOR DEFAULT CONFIGURATION, CAPTURE EFFECT

CW_{min}	$P_{detection}$	$r_{malicious}(Kbps)$	$r_{rest}(Kbps)$
32	0.0224	1572.68	656.03
31	0.2666	1617.71	652.13
30	0.8101	1665.95	647.88
29	0.9934	1717.12	643.36
28	0.9999	1770.96	638.67
27	1	1828.40	633.62

TABLE III

RESULTS FOR DEFAULT CONFIGURATION, NO CAPTURE EFFECT

CW_{min}	$P_{detection}$	$r_{malicious}(Kbps)$	$r_{rest}(Kbps)$
32	0	1127.61	656.04
23	0.0397	1522.15	609.49
22	0.4273	1583.87	602.28
21	0.9319	1650.70	594.50
20	0.9995	1722.85	586.15
19	1	1802.29	576.96

TABLE IV

RESULTS FOR OPTIMAL CONFIGURATION, NO CAPTURE EFFECT

CW_{min}	$P_{detection}$	$r_{malicious}(Kbps)$	$r_{rest}(Kbps)$
174	0	738.01	738.05
153	0.1267	829.32	728.57
147	0.5158	859.51	725.43
141	0.9029	892.37	722.01
130	0.9998	959.27	715.09
129	1	965.79	714.42

different situations: with capture effect and default 802.11 DCF parameter configuration, no capture effect and default 802.11 DCF parameter configuration, and finally, no capture effect and optimal parameter configuration.

We first assess the performance of our algorithm in the case when the controlled station captures the channel and all stations but the controlled one use the default configuration. The outcome for this case is given in Table II. According to these results, when the controlled user sets his CW_{min} to 32 (i.e. he is not misbehaving) the throughput he is obtaining ($r_{malicious}$) is already 2,4 times greater than the rest of the stations (r_{rest}), as a consequence of the capture effect. When he changes to a $CW_{min} = 27$ or below (misbehaving and being always detected), he is capturing 2,9 times more throughput than the rest of the stations. So he is benefiting, before he is always detected by the algorithm, from about 20% more throughput than a well-behaved user that always captures the channel. However, in absolute terms, he is obtaining about 3 times the throughput of any other user.

We next assess the performance of our algorithm with the previous configuration when the controlled station does not capture the channel. The outcome of this case is shown in Table III. It can be seen from these results that a malicious user is able to decrement his CW_{min} to a lower value than in the previous case before being always detected. Since the user does not capture the channel, and his throughput is about 3 times as large as the one for the rest of the users, we conclude that the algorithm is not effective in this case.

Our last experiment relates to the optimal configuration case with no capture effect. The results of this case are given in Table IV. According to this outcome, it can be seen that a malicious user has little chances of getting through unnoticed. Specifically, the controlled station is always detected when getting 35% or more throughput than the rest of the stations. We conclude that our algorithm combined with the optimal configuration is effective in avoiding that a user can benefit from the misconfiguration of his EDCA parameters.

VII. CONCLUSIONS

In this paper we addressed the issue of detecting a user that maliciously configures his EDCA parameters to obtain more throughput. In order to achieve this goal the challenge lies in detecting malicious CW_{min} and CW_{max} configurations.

One of the aspects that needs to be considered in the detection of malicious configuration of the CW parameters is the capture effect. To overcome this, we proposed to use the optimal configuration provided in [5]. With this arrangement, the impact of the capture effect is smaller and the difference between the throughput of a user that captures the channel and a user that does not is reduced. We showed that, with this setup, we can effectively restrain a malicious user trying to gain substantial throughput without being detected.

The key limitation of the procedure is that, because of the random nature of the EDCA mechanism, it is not possible to determine with 100% certainty that a user is misbehaving. Furthermore, there exists a fundamental tradeoff between this degree of certainty and the accuracy and time granularity of the detection. One of the main merits of the algorithm proposed here is that its parameters can be tuned in order to provide the desired tradeoff between these aspects.

REFERENCES

[1] IEEE 802.11e/D13.0, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS).* Draft Supplement to IEEE 802.11 Standard, January 2005.

[2] M. Raya, J.-P. Hubaux, and I. Aad, "DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots," in *Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (ACM MobiSys'04)*, Boston, MA, June 2004.

[3] P. Kyasanur and N. H. Vaidya, "Detection and Handling of MAC Layer Misbehavior in Wireless Networks," in *Proceedings of 2003 International Conference on Dependable Systems and Networks (DSN'03)*, San Francisco, CA, June 2003.

[4] A. Banchs, A. Azcorra, C. García, and R. Cuevas, "Applications and Challenges of the 802.11e EDCA mechanism: An Experimental Study," *IEEE Network*, vol. 19, no. 4, July/August 2005.

[5] A. Banchs, X. Pérez, and D. Qiao, "Providing Throughput Guarantees in IEEE 802.11e Wireless LANs," in *Proceedings of the 18th International Teletraffic Congress (ITC18)*, Berlin, Germany, September 2003.