SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 0000; **00**:1–14
DOI: 10.1002/sec

Research article

Improving QoS and security in wireless ad-hoc networks by mitigating the impact of selfish behaviors: a game-theoretic approach

Szymon Szott*, Marek Natkaniec, and Andrzej R. Pach

AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunications, al. A. Mickiewicza 30, 30-059 Krakow, Poland

ABSTRACT

Selfish users are known to be a severe security threat for wireless ad-hoc networks. In particular, they can exploit mechanisms designed to assure QoS in the network. In this paper the problem of backoff misbehavior in IEEE 802.11 EDCA networks is studied using a game-theoretic approach. First, it is shown how this selfish behavior can disrupt traffic differentiation. Then, a solution is proposed which encourages standard-compliant behavior and thus proper QoS provisioning. This solution is based on punishing selfish nodes by degrading their throughput proportionally to the degree of misbehavior. A practical application of the solution is proposed, which is verified through simulations. Results show that the suggested mechanism considerably improves QoS provisioning in IEEE 802.11 EDCA ad-hoc networks in the presence of selfish nodes. Furthermore, it is shown that the the mechanism is adaptive, does not have a negative impact on the throughput of well-behaving nodes, and provides legacy node support. Copyright © 0000 John Wiley & Sons, Ltd.

KEYWORDS

EDCA; game theory; misbehavior; QoS; security

***Correspondence**

Szymon Szott, AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunications, al. A. Mickiewicza 30, 30-059 Krakow, Poland. E-mail: szott@kt.agh.edu.pl

Received . . .

1. INTRODUCTION

The IEEE 802.11 standard [1] defines a wireless local area network technology which has achieved widespread success due to inexpensive equipment, simple deployment, and high transmission speeds. This standard supports Quality of Service (QoS) through the widely investigated and well described Enhanced Distributed Channel Access (EDCA) function. Even though medium access control (MAC) is based on a modified carrier sense multiple access with collision avoidance (CSMA/CA) scheme, traffic prioritization is achieved through the use of access categories (ACs) which have different medium access parameters.

For the network to operate correctly, cooperation between wireless nodes is required in accessing the shared radio channel. However, IEEE 802.11 does not contain any security measures to ensure that nodes conform to the standard. Nodes can misbehave (i.e., act selfishly) by manipulating access parameters in order to assure a higher probability of data transmission. Though several parameters may be changed, modification of the contention window parameters (known in the literature as *backoff misbehavior*) are the most difficult to detect because of their random nature. Furthermore, backoff misbehavior is hidden from detection schemes working at the network layer and can be combined with misbehavior in upper layers. Studies have shown that, e.g., in a five node network if one node decreases its contention window parameters it can increase its throughput seven times in [2]. Such parameter modification can easily be performed with the use of either modern wireless drivers [3] or the emerging, flexible "soft-MAC" drivers [4]. Even equipment vendors make non-standard modifications to increase the performance of their cards [5]. Therefore, the problem of selfish behavior has become pressing and requires a prompt resolution.

Prepared using secauth.cls [Version: 2010/06/28 v2.00]

Existing solutions to this problem have mostly focused either on optimizing network utilization under the Distributed Coordination Function (DCF) of IEEE 802.11 or on introducing fundamental changes to the standard. These approaches cannot be applied to EDCA (cf. Section 2), which is currently the only QoS mechanism standardized for multi-hop ad-hoc networks. Therefore, in this paper we present a new solution which fulfills the following set of requirements: (1) it encourages standard-compliant behavior, (2) does not have a negative impact on the performance of well-behaving nodes, (3) provides support for legacy nodes, and (4) assures EDCA compatibility.

We assume that the selfish, misbehaving users (i.e., the *cheaters*) are not networking experts — they can perform simple modifications of EDCA backoff parameters. In reality users could exploit more elaborate attacks (such as dynamic selection of these parameters) by using drivers prepared by a networking expert or by purchasing non-compliant hardware. The method proposed in this paper can easily be extended to cope with such attacks. However, other attacks, such as re-mapping of packets to ACs [6], are considered out of the scope of this paper. Furthermore, we assume that users misbehave to maximize throughput which is a straightforward and immediately beneficial goal.

In Section 2 we show how our approach differs from existing state-of-the-art solutions. In the subsequent sections, we provide the following original contributions:

- An EDCA model for saturation conditions is described in Section 3. This model of EDCA is simpler than those available in the literature* in order to rapidly calculate the network saturation throughput. However, despite its simplicity, it supports the most important EDCA features: multiple ACs, standard-compliant parameters, and Arbitration Inter-Frame Space (AIFS) differentiation. Therefore, it is able to correctly model EDCA (as verified by simulations). Additionally, the model supports the analysis of backoff misbehavior. The analytical results obtained from this model are utilized in the subsequent section.
- We analyze the impact of selfish behavior using a game theoretic framework (Section 4). To the best of our knowledge, this is the first such analysis performed for EDCA networks. It provides new insights related to EDCA, which show that current state-of-the art approaches cannot be applied. A solution to the problem of misbehavior is provided in the form of a penalty mechanism which is proportional to the degree of misbehavior. Theoretical results prove that the proposed solution improves QoS provisioning by encouraging the use of standard EDCA parameters.
- In Section 5 we discuss the application of the proposed penalty mechanism and perform a simulation analysis. The simulation results show that the mechanism provides incentives for nodes to choose standard-compliant parameters, does not have a negative impact on the throughput of well-behaving nodes and is compatible with EDCA. We also show that the proposed penalty mechanism cannot be exploited by selfish users.

Finally, we conclude the paper and provide possible directions of future work in Section 6. The nomenclature used throughout the paper can be found in Table I.

2. STATE OF THE ART

In recent years, numerous QoS solutions have been proposed for ad-hoc networks. An overview of challenges and solutions can be found in [8]. However, the EDCA function of IEEE 802.11 (described in Section 3) remains the only standardized QoS MAC protocol for such networks. It has been widely studied in the literature and several extensions have been proposed, e.g., to provide support for hidden nodes [9]. Furthermore, it has been shown that EDCA can be used within a comprehensive cross-layer QoS solution for ad-hoc networks [10]. For these reasons, EDCA is the focus of this paper.

The use of game theory for modeling different aspects of wireless networks has been reported in the literature for several years. A comprehensive overview of the basics of game theory and its application to IEEE 802.11 networks can be found in [11] and [12]. One of the first papers to employ game theory to optimize medium access in wireless networks was [13]. Similarly, other papers have used a game theoretic framework to enhance the performance of IEEE 802.11 networks [14, 15, 16].

Security aspects related to selfish behavior at the MAC layer have also been the subject of contemporary research [18]. In our opinion one of the most fundamental works with respect to selfishness and game theory was written by Cagalj et. al [19]. In this paper the authors focus on the coexistence of several cheaters in an IEEE 802.11 network. The strategy of the cheaters is to manipulate contention window values to achieve the highest throughput. The authors prove that the network suffers from a *tragedy of the commons* [20] in the presence of multiple cheaters. Then, they introduce a method to guide the cheaters to choose a contention window value which assures optimal and fair distribution of throughput. This approach is implemented through a detection mechanism (based on observing throughput deviations) and a penalization scheme (based on selectively jamming the frames of cheaters). The proposed solution is anonymous, distributed, self-adaptive, and it does not encourage the abuse of the penalization scheme. Despite these indisputable benefits the work presented in [19] cannot be applied to EDCA networks for the following reasons. In such networks it is impossible

*A comparison of EDCA models can be found in [7].

Security Comm. Networks 0000; **00**:1–14 © 0000 John Wiley & Sons, Ltd.
DOI: 10.1002/sec
Prepared using secauth.cls

Table I. Nomenclature

	Acronyms
AC	Access Category
AIFS	Arbitration Inter-Frame Space
BE	Best Effort
BK	Background
C	The cooperation strategy
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CW	Contention Window
DCF	Distributed Coordination Function
EDCA	Enhanced Distributed Channel Access
HR/DSSS	High Rate/Direct Sequence Spread Spectrum
M	The misbehaviour strategy
MAC	Medium Access Control
PD	Prisoner's Dilemma
QoS	Quality of Service
Vi	Video
Vo	Voice
	Variables
α_p	Penalty factor
δ	Propagation delay
τ_i	Transmission probability in a slot time for the i-th AC
$AIFS_i$	AIFS for the i-th AC
$AIFSN_i$	AIFS Number for the i-th AC
$b_i(t)$	Value of the backoff counter for the i-th AC at time t
$b_{i,j}$	Stationary distribution for $j \geq 0$
C_m	Payoff for a cooperating node when m other nodes are misbehaving
CW_i^{MIN}	CW maximum size for the i-th AC
CW_i^{MAX}	CW minimum size for the i-th AC
$DIFS$	DCF Inter-Frame Space
$EIFS$	Extended Inter-Frame Space
i	AC number
j	Retransmission counter
m	Number of misbehaving players
M_m	Payoff for a misbehaving node when m nodes are misbehaving
N_C	Number of ACs
n_i	Number of nodes using the i-th AC
N	Total number of nodes in the network
U	Payoff when both players misbehave in a PD (uncooperative payoff)
p^B	Probability that the wireless channel is busy
P^S	Probability of a successful transmission in any AC
p_i^B	Frame blocking probability for the i-th AC
R	Payoff when both players cooperate in a PD (reward)
S	Payoff for cooperating player in a PD if the other player misbehaves (sucker's payoff)
S_i	Throughput value for the i-th AC
$SIFS$	Short Inter-Frame Space
T	Payoff for misbehaving player in a PD if the other player cooperates (temptation to defect)
T^C, T^{CS}, T^S	Average duration of a collision/contention slot/successful transmission, respectively
T^{DATA}	Average time required to send a DATA frame
T^H	Time required to send the PHY and MAC headers
T_e	Slot time
$TXOP_{Limit}$	Transmission Opportunity Limit

to find a single optimal contention window value because each AC has different access parameters, which result in differences in throughput. These differences are necessary to provide QoS. Furthermore, the approach presented in [19] ensures that the misbehaving nodes will have the same throughput but it will be significantly higher than that of the well-behaving nodes. This does not comply with the standard. Additionally, the authors acknowledge that the misbehavior detection mechanism that they use does not work with different traffic constraints (such as appear in EDCA). They suggest using backoff detection, i.e., comparing the measured and expected distributions of backoff values, to determine which nodes are misbehaving. We propose such a method in [21] while an alternative can be found in [22]. Additionally, as pointed out by Konorski in [15], the method presented in [19] assumes genuine node identities and uses a non-standard penalization method (frame jamming). The former issue is out of the scope of this paper while the latter is addressed in Section 5.

The authors of [17] also use game theory to study EDCA networks with selfish nodes. However, their goal is not achieving standard-compliance but rather increasing the performance of EDCA by dynamically adjusting backoff values based on current network conditions. This proposal requires additional information to be exchanged between nodes and, unlike our solution, it does not provide support for legacy nodes.

An alternative approach to addressing the problem of misbehavior is to use a preventive strategy. This requires modifying the medium access function so that misbehavior becomes completely or nearly impossible. An example of this is the negotiation of contention window parameters presented in [23]. Other examples can be found in [24, 25, 26]. The disadvantage of these approaches, however, is that they do not provide support for legacy nodes and in many cases cannot be applied to EDCA.

To summarize, this paper distinguishes itself from the state of the art because our goal is to provide incentives for nodes to behave according to the EDCA function of the IEEE 802.11 standard while assuring legacy node support.

3. SIMPLIFIED EDCA MODEL

This section begins with a brief description of the operation of EDCA. This is followed by the presentation of a simplified EDCA model which is then used to estimate the throughput values required in the game theoretic analysis presented in Section 4. This model is a simplified and saturation-only version of our previous work reported in [27]. We analyze only saturation conditions, because in a non-saturated network, the impact of backoff misbehavior is not significant [2].

EDCA introduces four ACs to provide appropriate QoS: Voice (Vo), Video (Vi), Best effort (BE), and Background (BK). Each category has its own set of medium access parameters, which are responsible for traffic differentiation. These parameters are: the Arbitrary Inter-frame Space Number ($AIFSN$), the Contention Window Minimum and Maximum values (CW^{MIN} and CW^{MAX}), and the optional Transmission Opportunity Limit ($TXOP_{Limit}$).

In EDCA, medium access is regulated by the following backoff mechanism. To access the channel a node randomly selects a value from the range $[0, CW]$ (initially $CW = CW^{MIN}$). The chosen backoff value denotes the time slot in which the node will begin its transmission. The decreasing of this value begins when the channel has been idle for an $AIFS$ period. The countdown is paused when the channel is sensed busy. When the backoff value reaches zero, the node starts to transmit. In order to avoid collisions the following binary exponential mechanism is used: if a collision occurs, CW is doubled until it reaches CW^{MAX}. In the case of a successful transmission CW is reset to the value of CW^{MIN}. Otherwise, after a given number of unsuccessful transmission attempts, the frame is dropped. To summarize, the backoff mechanism decreases the probability that two nodes will transmit simultaneously and thus cause a collision.

We model an EDCA wireless network in which each node transmits traffic of one AC. To provide rapid calculations, we assume that there are no retransmissions, there is no binary exponential mechanism (i.e., $CW^{MIN} = CW^{MAX} = CW_i$), the RTS/CTS exchange is not used, $TXOP_{Limit}$ is set to zero, the medium is error-free, there are no hidden or exposed nodes, and frames are of equal length. These simplifications do not affect the saturation throughput of nodes and therefore do not have a qualitative impact on the study.

The input parameters for our analysis are the number of ACs in the network (N_C), the number of nodes in the i-th AC (n_i), and the total number of nodes in the network N ($N = \sum_{i=0}^{N_C-1} n_i$). Backoff misbehavior is modeled by using an additional AC (denoted by the index *misb*) which has a non-standard contention window value (CW_{misb}).

The goal of our analysis is to derive the overall throughput in each AC (S_i). It is defined as the quotient of the average duration of a successful transmission of a frame in the i-th AC and the average duration of a contention slot (T^{CS}), in which the frame competes for medium access with other frames:

$$S_i = \frac{p_i^S T^{DATA}}{T^{CS}}, \tag{1}$$

where p_i^S is the probability of a successful transmission for the i-th AC and T^{DATA} is the average time spent on transmitting a frame (without the PHY and MAC headers).

If we define τ_i as the transmission probability in a slot time for the i-th AC we can compute p_i^S as the probability that only one node is transmitting in a given slot time:

$$p_i^S = n_i \, \tau_i \, (1-\tau_i)^{n_i - 1} \prod_{\substack{j=0 \\ j \neq i}}^{N_c - 1} (1-\tau_j)^{n_j}. \tag{2}$$

We calculate T^{CS} using the following equation:

$$T^{CS} = (1-p^B)T_e + P^S T^S + (p^B - P^S)T^C, \tag{3}$$

where p^B is the probability that the channel is busy, $1-p^B$ is the probability of a free channel, T_e is the slot time, P^S is the overall probability of a successful transmission in any AC ($P^S = \sum_{i=0}^{N_c-1} p_i^S$), and T^S (T^C) is the duration of a successful transmission (collision). T^S and T^C can be calculated as

$$T^S = AIFS^{MIN} + T^H + T^{DATA} + SIFS + T^{ACK} + 2\delta, \tag{4}$$

$$T^C = T^H + T^{DATA} + \delta + EIFS, \tag{5}$$

where $EIFS$ is the Extended Inter-Frame Space, $AIFS^{MIN}$ is the minimum $AIFS$ value among all ACs,

Security Comm. Networks 0000; **00**:1–14 © 0000 John Wiley & Sons, Ltd.
DOI: 10.1002/sec
Prepared using secauth.cls

T^{ACK} is the time required to send the ACK frame, δ is the propagation delay, T^H is the time required to send the PHY and MAC headers, and $SIFS$ is the Short Inter-Frame Space.

The probability that the channel is busy p^B is equal to the probability that at least one node in the network is transmitting:

$$p^B = 1 - \prod_{i=0}^{N_c-1} (1-\tau_i)^{n_i}. \quad (6)$$

The remaining unknown variables of (2) and (3) can be found using numerical analysis of the Markov chain presented in Figure 1. The brevity of the model follows the simplifying assumptions stated at the beginning of this section. Even if a more detailed model was applied, the conclusions derived in the subsequent sections would not change.

We denote CW_i as the contention window value of the i-th AC. Furthermore, we define the probability that, for a given node, at least one other node is transmitting during the given node's backoff. This is the frame blocking probability for the i-th AC (p_i^B). We also need to take into account the different values of $AIFS_i$ because nodes transmitting with a lower priority AC need to wait for more empty slots than nodes transmitting with a higher priority AC. We calculate p_i^B using the following equation:

$$p_i^B = 1 - \left[(1-\tau_i)^{n_i-1} \prod_{j=0, j\neq i}^{N_c-1} (1-\tau_j)^{n_j}\right]^a, \quad (7)$$

where $a = AIFSN_i - AIFSN^{MIN} + 1$, $(1-\tau_i)^{n_i-1}$ is the probability that no other nodes are transmitting data in the i-th AC, $\prod_{j=0,j\neq i}^{N_c-1}(1-\tau_j)^{n_j}$ is the probability that none of the nodes are transmitting data in the other ACs, and $AIFSN_i$ is the AIFS Number for the i-th AC.

Let $b_i(t)$ be the value of the backoff counter for a given node and the i-th AC, where t is given in slot times. We model the process $b_i(t)$ with the discrete Markov chain presented in Figure 1. We assume the notation that $b_{i,j} = \lim_{t\to\infty} P\{b_i(t) = j\}$ ($i \in 0, ..., N_c - 1$ and $j \in 0, ..., CW_i$). This is the stationary distribution of the Markov chain.

From the chain analysis, every $b_{i,j}$ state can be represented as a function of $b_{i,0}$:

$$b_{i,j} = \frac{CW_i + 1 - j}{CW_i + 1} \frac{b_{i,0}}{1 - p_i^B}, \text{for } j \geq 1. \quad (8)$$

From (8) and the normalization property ($\sum_{j=0}^{CW_i} b_{i,j} = 1$) we can derive the transmission probability in a slot time for the i-th AC:

$$\tau_i = b_{i,0} = \frac{(1-p_i^B)(CW_i+1)}{\sum_{j=0}^{CW_i}(CW_i+1-j)}. \quad (9)$$

Finally, using equations (1) to (9) we can compute the overall throughput in each AC (S_i). In order to obtain numerical throughput values we assume that IEEE 802.11 HR/DSSS (known as 802.11b) [1] is used. Even though this standard has been (in recent years) superseded by OFDM-based schemes, this choice does not qualitatively impact the analysis but rather allows to achieve network saturation faster. The following values are set according to 802.11b: $N_C = 4$, $M = 4$, $SIFS = 10$ μs, $EIFS = 318$ μs, $DIFS = 50$ μs, and $T_e = 20$ μs. The EDCA parameters for 802.11b are given in Table II. Furthermore, we assume that the average frame size is 1000 B and that $\delta = 2$ μs. Under this last presumption the distances between nodes cannot exceed 600 m. We then use Wolfram Mathematica 7 [28] to provide the numerical calculations for the study of EDCA games in Section 4.

Table II. Default EDCA Parameters of IEEE 802.11 HR/DSSS (802.11b)

Access Category (i)	$AIFSN_i$	CW_i^{MIN}	CW_i^{MAX}
Vo	2	7	15
Vi	2	15	31
BE	3	31	1023
BK	7	31	1023

4. STUDY OF EDCA GAMES

We analyze the EDCA game which is a game-theoretic model of a single-hop network in which all nodes use the EDCA function (as described and modeled in Section 3). Each node in the network is a player in the game. Players have two available strategies: **cooperate** (C) and conform to the standard or **misbehave** (M) and deviate from the standard. Furthermore, we assume that in the second case the misbehaving nodes set $CW_{misb} = 1$[†]. The payoff of each player is the throughput they achieve and the goal of each player is to maximize throughput. To simplify the analysis we assume that the nodes are not energy-constrained. Therefore, there is no cost of transmission. This assumption is valid if nodes are connected to a mains power supply or lack energy awareness. As a final consideration, we assume that players do not collude.

We denote one stage of the game as a fixed period of time T^G during which the players *play* for the throughput. We use the classical prisoner's dilemma (PD) notation where R – reward for cooperation, S – sucker's payoff, T – temptation payoff, and U – uncooperative payoff [20]. Table III presents the canonical PD payoff matrix.

A game is a PD if

$$T > R > U > S. \quad (10)$$

[†] $CW_{misb} = 1$ is the lowest value possible with the use of available drivers [3]. Even if $CW_{misb} = 0$ was a feasible option, the network would collapse if more than one node were to set such a CW value. This is because all nodes would transmit simultaneously (i.e., without backoff).

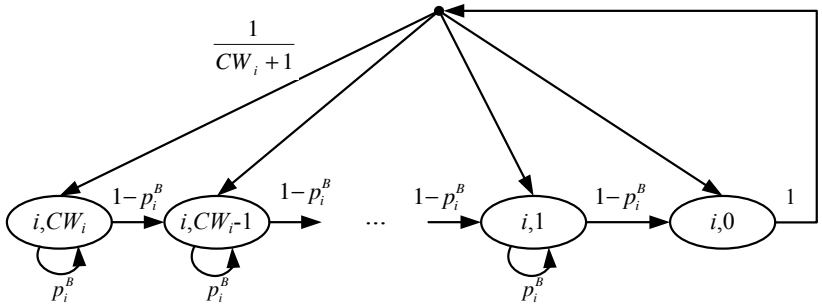

Figure 1. Markov chain of the proposed model

Table III. PD payoff matrix

N_1 \ N_2	C	M
C	(R, R)	(S, T)
M	(T, S)	(U, U)

Table IV. Normalized throughput in EDCA game for $N = 2$ and BE AC

N_1 \ N_2	C	M
C	$(0.237, 0.237)$	$(0.006, 0.526)$
M	$(0.526, 0.006)$	$(0.206, 0.206)$

Table V. Normalized throughput in EDCA game for $N = 2$ and different ACs (N_1: Vo and N_2: BE)

N_1 \ N_2	C	M
C	$(0.449, 0.064)$	$(0.045, 0.454)$
M	$(0.546, 0.002)$	$(0.457, 0.039)$

Table VI. Normalized throughput in EDCA game for $N = 5$ and BE AC

N_1 \ m	0	1	2	3	4
C	0.094	0.007	0.006	0.005	0.004
M	0.472	0.189	0.115	0.081	0.061

Neither player has any incentive to deviate from strategy M because neither of them can gain more throughput by unilaterally changing their strategy. This is known as the Nash equilibrium [20]. The *dilemma* in this case is that the players would gain more by cooperating. This, however, would require a bilateral change of strategy.

4.1. Two player EDCA Games

First a single-stage two player game ($N = 2$) is analyzed. Each node transmits to exactly one receiver and both players transmit traffic using the same AC. We choose BE because it is the default AC in the IEEE 802.11 standard. The normalized throughput results derived from the EDCA model (Section 3) are presented as a payoff matrix in Table IV. If both players cooperate they receive 0.237 of the normalized throughput. If both misbehave they receive 0.206. If only one of them misbehaves, the cooperating player receives only 0.006 while the misbehaving one receives 0.526. These results fulfill condition (10) and prove that the EDCA game in which nodes choose the same AC is a PD. This has been shown previously for DCF [19], which can be considered as "EDCA with one AC". A more interesting case, presented next, is when there are various traffic sources in the network and different ACs are used.

A very simple example of an EDCA game with different ACs is when there are two nodes (N_1 and N_2) sending Vo and BE traffic, respectively. If N_2 is selfish, then N_1 receives a lower level of service (Table V). However, this game is not a PD since for N_1: $U > R$ ($0.457 > 0.449$). This difference is caused by the fact that if both nodes use $CW_{misb} = 1$ then the difference in throughput is based solely on $AIFS$. This simple example shows that the state-of-the-art approach (i.e., choosing CW as the strategy and aiming for throughput fairness [19]) cannot be applied to EDCA networks.

4.2. Multiplayer Games

Having analyzed two player games, multiplayer games ($N > 2$) are studied next. We assume that all nodes send BE traffic. Tables VI and VII provide exemplary results for $N = 5$ and $N = 100$, respectively. These are the results of the normalized throughput of player N_1 with respect to the strategy chosen by it and the other players. The number of other misbehaving players is represented by the variable

Security Comm. Networks 0000; **00**:1–14 © 0000 John Wiley & Sons, Ltd.
DOI: 10.1002/sec
Prepared using secauth.cls

Table VII. Normalized throughput in EDCA game for $N = 100$ and BE AC

N_1 \ m	0	1	2	...	99
C	0.0025	0.0021	0.0018	...	0.0001
M	0.0333	0.0279	0.0240	...	0.0008

Table VIII. Normalized throughput in EDCA game for $N = 5$ (N1: BE, N2–N5: Vo)

N_1 \ m	0	1	2	3	4
C	0.0120	0.0029	0.0022	0.0016	0.0012
M	0.1568	0.0434	0.0283	0.0199	0.0149

m. Such a representation is sufficient because the game is symmetric, i.e., each player has the same perspective of the game and the choice of which node is N_1 does not influence the analysis.

Let us assume that C_m and M_m represent the two possible payoffs for N_1 when m other users are misbehaving. A multiplayer EDCA game in which all nodes use the same AC is a PD if the following conditions are satisfied [20]:

- $M_m > C_m$ for $0 \leq m \leq N-1$ (M is the dominant strategy, i.e., for any number of m misbehaving nodes it is always more beneficial to choose the M strategy),
- $M_{m+1} < M_m$ and $C_{m+1} < C_m$ for $0 \leq m \leq N-1$ (the payoff decreases with the increase of m, i.e., the throughput of any node decreases with the increase of misbehaving nodes in the network),
- $C_0 > M_{N-1}$ (universal cooperation is superior to universal misbehavior, i.e., nodes achieve higher throughput if they all cooperate than if they all misbehave).

First, we prove that $M_m > C_m$ for $0 \leq m \leq N-1$. Assuming that node N_1 uses its own AC (a separate AC indexed as s)[‡] and (similarly to [19]) that S_s is a continuous function of CW_s we can calculate the following:

$$\frac{\partial S_s}{\partial CW_s} = \frac{\partial S_s}{\partial \tau_s} \frac{\partial \tau_s}{\partial CW_s}. \tag{11}$$

The first derivative of Equation 1 can be computed as:

$$\frac{\partial S_s}{\partial \tau_s} = \frac{(c + 2T^C)T^{DATA}}{[(1-\tau_s)T_e + c + \tau_s T^S + (1-2\tau_s)T^C]^2}, \tag{12}$$

where $c = \sum_{\substack{j=0 \\ j \neq s}}^{N_c - 1} n_j (1-\tau_j)^{n_j - 1}(T^S + T^C)$.

Similarly, we calculate

$$\frac{\partial \tau_s}{\partial CW_s} = \frac{2(p_s^B - 1)\left[\sum_{i=0}^{4}(p_s^B)^i\right]^2}{[3 - p_s^B(d + eCW_s) + CW_s]^2}. \tag{13}$$

where $d = \left(3 + p_s^B + (p_s^B)^2 + (p_s^B)^3 + 2(p_s^B)^4\right)$ and $e = (1 + p_s^B)\left(1 + (p_s^B)^2\right)$.

[‡]This means that nodes in the network use three ACs: $N - m - 1$ nodes use BE, m misbehaving nodes use an AC indexed as *misb*, and node N_1 uses an AC indexed as s. This assumption simplifies the analysis without changing the overall conclusions.

We conclude that $\frac{\partial S_s}{\partial \tau_s} > 0$ and $\frac{\partial \tau_s}{\partial CW_s} < 0$, which means that throughput is a decreasing function of the contention window size. Therefore, choosing strategy M over strategy C (i.e., decreasing the contention window size) always provides an increase in throughput.

Secondly, we prove that $M_{m+1} < M_m$ and $C_{m+1} < C_m$ for $0 \leq m \leq N-1$. Under the same assumptions as before, we calculate the following:

$$\frac{\partial S_s}{\partial m} = \frac{\partial S_s}{\partial \tau_s} \frac{\partial \tau_s}{\partial p_s^B} \frac{\partial p_s^B}{\partial m}. \tag{14}$$

The first derivative of Equation 9 can be computed as:

$$\frac{\partial \tau_s}{\partial p_s^B} = -\frac{CW_s + 1}{\sum_{j=0}^{CW_s}(CW_s + 1 - j)}. \tag{15}$$

Furthermore, the first derivative of (7) can be calculated as shown in (16). Based on (13) we know that $\tau_{misb} > \tau_{BE}$ and therefore $\frac{\partial p_s^B}{\partial m} > 0$. From (12) and (15) we have $\frac{\partial S_s}{\partial \tau_s} > 0$ and $\frac{\partial \tau_s}{\partial p_s^B} < 0$, respectively. Applying these findings to (14) we conclude that the throughput of any node decreases with the increase of the number of misbehaving nodes in the network.

Thirdly, we study if $C_0 > M_{N-1}$, i.e., if there will be an increase in throughput when all nodes in the network perform a uniform change of CW from the standard value to a non-standard one. This depends on the network size and it has been shown in the literature that the standard contention window values are not always optimum in terms of throughput [19, 29, 30, 31]. This is confirmed by results from the EDCA saturation model proposed in this paper (Figure 2). These results show that for $N \geq 3$ non-standard CW_{misb} values for the BE AC provide lesser throughput than the standard values. Therefore, we conclude that universal cooperation is superior to universal misbehavior.

Having shown that multiplayer EDCA games in which all nodes use the same AC are PDs we analyze multiplayer games in which nodes use different ACs. Table VIII presents normalized throughput results for a network consisting of five nodes in which one node sent traffic using BE while the other nodes sent traffic using Vo. This exemplary scenario shows that not all multiplayer EDCA games are PDs. In this case the condition $C_0 < M_{N-1}$ was not met for node N1. However, studies have shown that this condition was met for the remaining nodes (N2–N5). Therefore, despite the fact that there exist configurations of EDCA games for which the formal PD conditions are not

$$\frac{\partial p_s^B}{\partial m} = \begin{cases} (1-\tau_{BE})^{N-m}(1-\tau_{misb})^{m-1} ln \frac{1-\tau_{BE}}{1-\tau_{misb}}, \text{if } N_1 \text{ choses strategy } M, \\ (1-\tau_{BE})^{N-m-1}(1-\tau_{misb})^{m} ln \frac{1-\tau_{BE}}{1-\tau_{misb}}, \text{if } N_1 \text{ choses strategy } C. \end{cases} \quad (16)$$

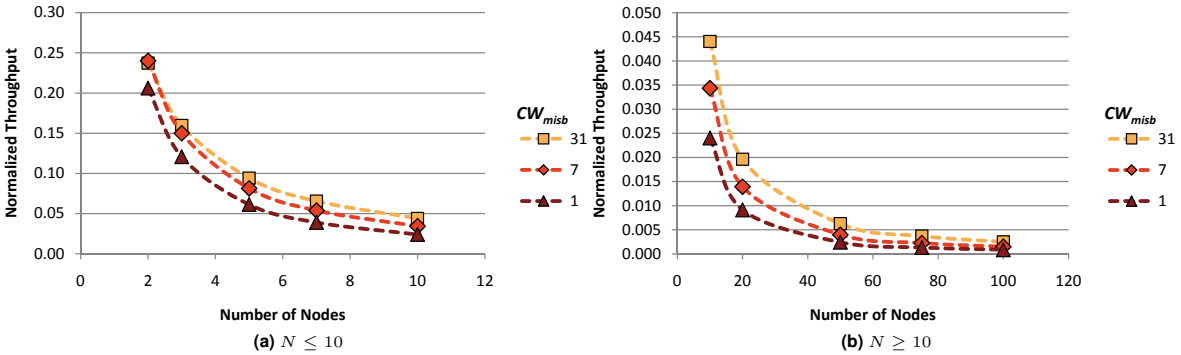

Figure 2. Impact of number of nodes in the network and CW_{misb} on normalized per-node throughout in a network in which all nodes send BE traffic

met for all players, the M strategy remains dominating for all players.

4.3. Proposed Solution

The analysis of EDCA games has provided the following conclusions:

- if all nodes use the same AC then the game is a PD,
- if nodes use different ACs then the game may not be a strict PD (depending on AC),
- M is the dominant strategy and causes lack of proper traffic differentiation,
- a large number of misbehaving nodes severely degrades QoS provisioning in the network.

In the literature, the solution to a multiplayer PD is to force players into cooperation [20]. This has been phrased by Hardin as "mutual coercion mutually agreed upon" [32]. Therefore, a distributed method of reacting to misbehavior that enforces cooperation is required. Furthermore, we want the method to be compatible with the IEEE 802.11 standard. To this end **we propose punishing selfish nodes by degrading their throughput proportionally to the degree of misbehavior**. We employ this *penalty method* by introducing a penalty factor α_p to (1) in the following way:

$$S_{misb} = \begin{cases} \frac{p_{misb}^S T^{DATA}}{T^{CS}}, & \text{if } CW_{misb} \geq CW_{std}, \\ \alpha_p \frac{p_{misb}^S T^{DATA}}{T^{CS}}, & \text{if } CW_{misb} < CW_{std}, \end{cases} \quad (17)$$

where CW_{std} is the standard CW^{MIN} value for the manipulated AC and α_p is defined as

$$\alpha_p = \frac{CW_{misb} - 1}{CW_{std} - 1}. \quad (18)$$

The amount of penalty depends on how much the player has deviated from the standard. If the misbehaving node chooses the smallest possible contention window ($CW_{misb} = 1$) then it will not receive any throughput. Naturally, other, more complex (non-linear) penalty functions may be envisaged. For example, in a real-world implementation, the penalty function might need to be made more aggressive, depending on the throughput awareness of the user. However, as long as the penalty function assures that a node using non-standard parameters has decreased throughput, then the conclusions drawn below are correct. Additionally, for now, we assume that this penalty is applied automatically. Later, in Section 5, we discuss how to apply this mechanism to EDCA networks in a distributed manner.

An example of how the penalty mechanism affects the throughput of the misbehaving node is shown in Figure 3. This figure presents analytical results for $N = 5$, $m = 1$ and all nodes transmitting either Vo or BE traffic. If no penalty is applied, low contention window values lead to high throughput. This follows from the analysis of (11). If the penalty factor α_p is used then the misbehaving node achieves the highest throughput only if it uses the standard parameter values (i.e., CW_{misb} equal to 7 for Vo or 31 for BE). Therefore, cooperation becomes the optimal strategy for a misbehaving node.

The penalty method was also validated in two scenarios in which only BE traffic was present. In the first, there was one misbehaving node but the size of the network varied (Figure 4). In the second, the network consisted of 20 nodes but had a varying percentage of misbehaving nodes (Figure 5). In both cases the penalty method can correctly decrease the throughput. In Figure 4 the highest throughput is achieved for the standard parameter values (i.e., $CW_{misb} = 31$). Similarly, in Figure 5 the

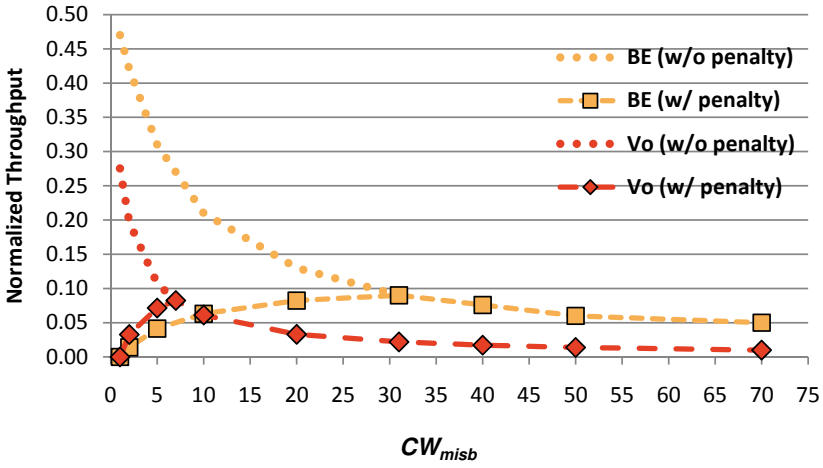

Figure 3. Throughput of misbehaving node with and without penalization

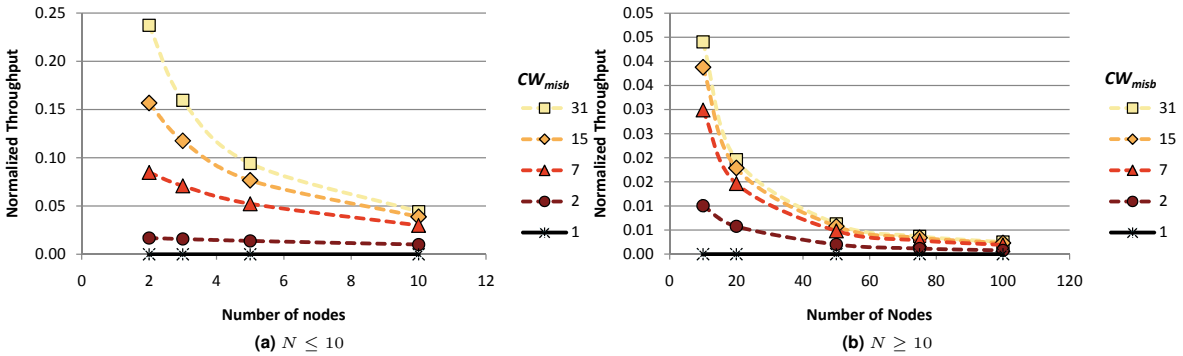

Figure 4. Impact of penalty on misbehaving node throughput for a varying number of nodes in the network and BE traffic

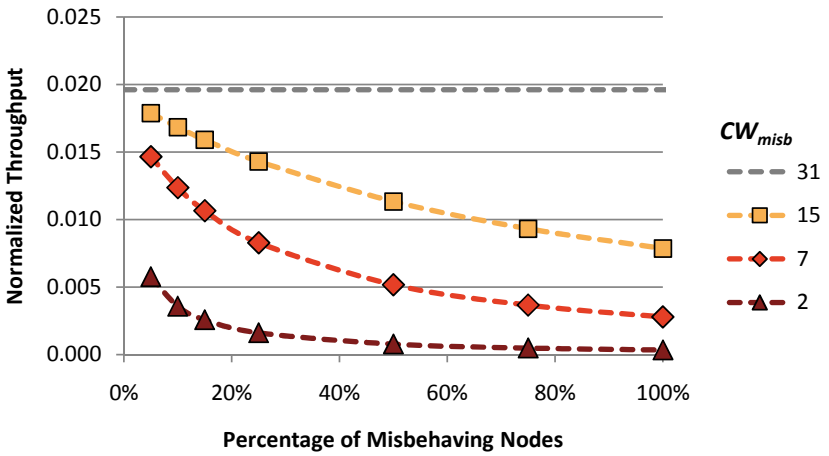

Figure 5. Impact of penalty on misbehaving node throughput for a varying number of misbehaving nodes and BE traffic

misbehaving nodes receive much less throughput than in the case where all nodes cooperate (illustrated in the figure as the reference level of $CW_{misb} = 31$).

It must be noted, however, that the penalty mechanism does not ensure that cooperating nodes automatically achieve more throughput. This is because the penalty factor α_p is applied to the throughput (S_{misb}) and not to the transmission probability (τ_{misb}). This means that misbehaving nodes still occupy the channel by generating DATA frames.

Table IX. Normalized throughput in EDCA game for $N = 5$ and BE traffic with penalty mechanism enabled

N_1 \ m	0	1	2	3	4
C	0.094	0.007	0.006	0.005	0.004
M	0	0	0	0	0

We again study the EDCA game for $N = 5$ and BE traffic but with the penalty mechanism enabled. Since misbehaving nodes set $CW_{misb} = 1$ then according to (17) they achieve zero throughput (Table IX). The domination of the cooperation strategy is true not only for the extreme case of $CW_{misb} = 1$. This occurs independently of the chosen CW_{misb} and the configuration of AC traffic in the network because only cooperation provides the highest throughput for each node (Figure 3).

This means that in repeated games (i.e., games consisting of multiple T^G periods which are equivalent to a session in which the user accesses the network) all players use an adaptive maximization strategy: they modify CW_{misb} to maximize their throughput. Thus, with the penalty mechanism enabled universal cooperation becomes the Nash equilibrium. In conclusion, we can state that the goal of encouraging players to use standard EDCA parameters has been achieved. Obviously, the problem of misbehavior on the penalty method remains. We address this issue in Section 5.4.

5. APPLICATION OF PENALTY MECHANISM

In this section we discuss how the theoretical approach described in the previous section can be realized in an IEEE 802.11 EDCA network. Before the penalty mechanism can be applied, however, the misbehaving nodes need to be first detected. Because this issue is out of the scope of this paper, we refer the reader to [21] were we discuss detection methods for EDCA and propose a novel one. Our method can successfully detect nodes which use non-standard values of contention window parameters. Therefore, we assume that the misbehaving nodes have been identified.

The penalty mechanism of (17) requires the cheater's throughput to be decreased in an adaptive manner without affecting the throughput of both the node imposing the penalty (the *penalty* node) and the other, well-behaving nodes. In EDCA networks a successful transmission occurs only when one node is transmitting, therefore, the penalty can be inflicted on one node without affecting the other nodes. Since single-hop networks are considered and support for legacy 802.11 nodes is required there are two reasonable reaction methods:

Table X. Simulation parameters

Basic rate	1 Mb/s	Data rate	11 Mb/s
δ	2 μs	Frame Size	1000 B
Transport protocol	UDP	Traffic generator	CBR
PHY overhead	192 bits	MAC header	32 B
SIFS	10 μs	EIFS	318 μs

- selective frame jamming [19] — any node jams the payload of the DATA frames of the misbehaving node for a short duration of time,
- refusing ACK frames [33] — the node which is the receiver of frames transmitted by the misbehaving node refuses to send ACK frames.

Even though these techniques have been already known in the literature, they have not yet been applied to EDCA. Furthermore, they are the only ones which are relatively simple to implement, operate in a distributed manner, work at the MAC layer, and can be used in single-hop networks. However, more importantly, both these methods fulfill the previously mentioned requirements:

- encouraging standard-compliant behavior — frame jamming or refusing ACK frames occurs with a probability of α_p (for $CW_{misb} < CW_{std}$) which means that the highest possible throughput is achieved only for standard parameter values,
- having a negligible impact on other nodes — the throughput of the node executing the penalty mechanism and the legacy nodes is not negatively affected by frame jamming or refusing ACK frames,
- providing support for legacy nodes — nodes which are unaware of frame jamming or refusing ACK frames operate normally,
- assuring EDCA compatibility — both frame jamming and refusing ACK frames can be applied regardless of the AC used to transmit traffic in the network.

The simulation results presented in this section prove that these requirements are met.

In single-hop EDCA networks the two methods cause the same effects and are, in fact, indistinguishable from the perspective of an uninvolved node. Therefore, we conduct a simulation analysis of only one of these methods (i.e., refusing ACKs).

The simulations were performed using the ns-2 simulator with our modified version of the EDCA patch described in [34]. These modifications allow simulating networks with both misbehaving and well-behaving nodes. Each simulation run was repeated multiple times to assure the defined confidence level. The 95% confidence intervals of the results are either presented in the figures or were too small for graphical representation.

In the following subsections we consider several ad-hoc scenarios. In each scenario there is a single-hop network

Security Comm. Networks 0000; **00**:1–14 © 0000 John Wiley & Sons, Ltd.
DOI: 10.1002/sec
Prepared using secauth.cls

Table XI. Throughput comparison for scenario with BE traffic. Case A — reference (no misbehavior). Case B — misbehavior but no penalty mechanism. Case C — penalty mechanism is enabled. Case D — Penalty mechanism is enabled and cheater uses an adaptive maximization strategy.

Node	Case			
	A	B	C	D
Cheater	0.10	0.35	0.05	0.10
Penalty	0.10	0.04	0.05	0.10
Legacy (average)	0.10	0.04	0.05	0.10

Table XII. Throughput comparison for scenario with different ACs. Case A — reference (no misbehavior). Case B — misbehavior but no penalty mechanism. Case C — penalty mechanism is enabled. Case D — Penalty mechanism is enabled and cheater uses an adaptive maximization strategy.

Node	Case			
	A	B	C	D
Cheater (BK)	0.005	0.100	0	0.005
Penalty (Vo)	0.107	0.072	0.095	0.107
Legacy (Vo) (average)	0.104	0.066	0.067	0.104

using the 802.11b physical layer. Nodes generate enough traffic to saturate the network. Tables II and X list the most important EDCA and simulation parameters, respectively. Other parameters are set according to the IEEE 802.11 standard.

5.1. Single AC Traffic

First, a small network is considered to illustrate the basic properties of the penalty mechanism of (17). There are five nodes in the network and all transmit BE traffic. However, one node is a cheater (with $CW_{misb} = 5$), one is a penalty node, and the other three are legacy 802.11 nodes. Figure 6 shows how the throughput of the cheater is degraded once the penalty mechanism is turned on after 20 s. The loss in throughput is proportional to the degree of misbehavior. In the presented case

$$\alpha_p = \frac{CW_{misb} - 1}{CW_{std} - 1} = \frac{5 - 1}{31 - 1} = \frac{2}{15}, \qquad (19)$$

which means that for only two out of every 15 DATA frames does the cheater receive an ACK. The penalty mechanism does not have a negative impact on the throughput of neither the penalty nor the legacy nodes. In fact, these nodes observe a slight increase in throughput. This is because these nodes can access the channel slightly sooner than the misbehaving node which has to wait for the EIFS time interval after not receiving an ACK.

Table XI presents simulation results for the following four cases of the described scenario: A — reference case with no misbehavior, B — misbehavior but no penalty mechanism, C — penalty mechanism enabled, and D — penalty mechanism enabled and the cheater uses an adaptive maximization strategy. These results show that, through the use of the penalty mechanism, the status of the network moves from being dominated by the cheater (Case B) to a fair sharing of network resources through standard compliance (Case D). Furthermore, as described before, the penalty mechanism provides a slight increase in throughput for the penalty and the legacy nodes. Finally, we present results to show how the penalty mechanism adopts to various levels of misbehavior (Figure 7). These results show that the implementation of the penalty mechanism is satisfactory. The slight difference between analysis and simulation is a result of the simplification assumptions (Section 3).

5.2. Multiple AC Traffic

In the next scenario, a situation is analyzed in which nodes transmit traffic of different ACs to show that the penalty mechanism can be used regardless of the traffic patterns in the EDCA network. This scenario is similar to the previous one. The difference is that the cheater transmits BE traffic and the other nodes transmit Vo traffic. Table XII presents the throughput results for the same four cases as previously. In the first case, the node sending BE traffic receives much less throughput than the nodes sending Vo traffic. This is consistent with the IEEE 802.11 standard. However, once the cheater starts misbehaving (by setting $CW_{misb} = 1$) it obtains higher throughput than the well-behaving nodes. Therefore, the misbehaving node can use contention window manipulation to gain even higher throughput than Vo traffic, even if it transmits using an AC of the lowest priority. Again, the penalty mechanism provides incentives for nodes to behave in a standard compliant manner without degrading the throughput of other nodes. These nodes observe a slight increase in throughput because they can access the channel slightly sooner than the misbehaving node which has to wait for the EIFS time interval.

5.3. Multiple Misbehaving Nodes

In the final simulation scenario, we analyze the performance of the penalty mechanism in the presence of multiple misbehaving nodes. We consider a single-hop network consisting of 20 nodes which transmit BE traffic. The misbehaving nodes set $CW_{misb} = 5$. Figure 8 presents the normalized average throughput of nodes with respect to the percentage of misbehaving nodes in the network.

The throughput achieved by all nodes strongly depends on the number of cheaters in the network. For both types of nodes (cheaters and other nodes) the throughput decreases exponentially with the increase of the percentage of misbehaving nodes in the network. This is because of the large number of collisions which result from the low contention window values set by the cheaters.

Applying the penalty mechanism assures that the throughput of misbehaving nodes is considerably reduced, regardless of their percentage. Most importantly, the goal of the proposed penalization has been achieved —

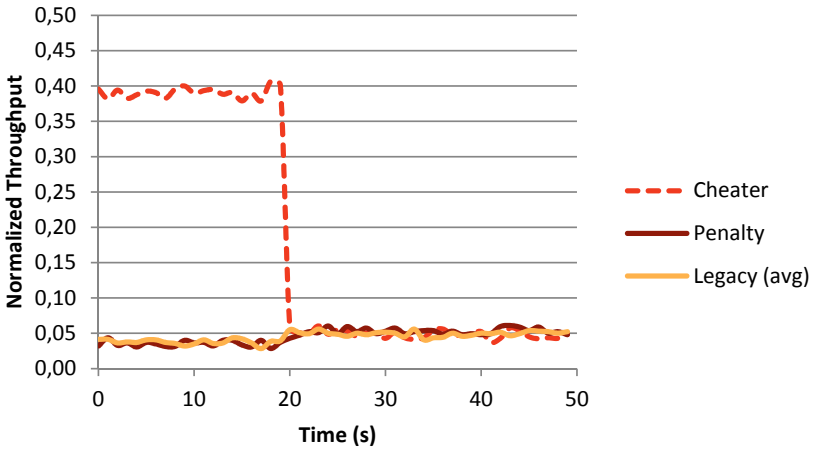

Figure 6. Node throughput before and after enabling penalization

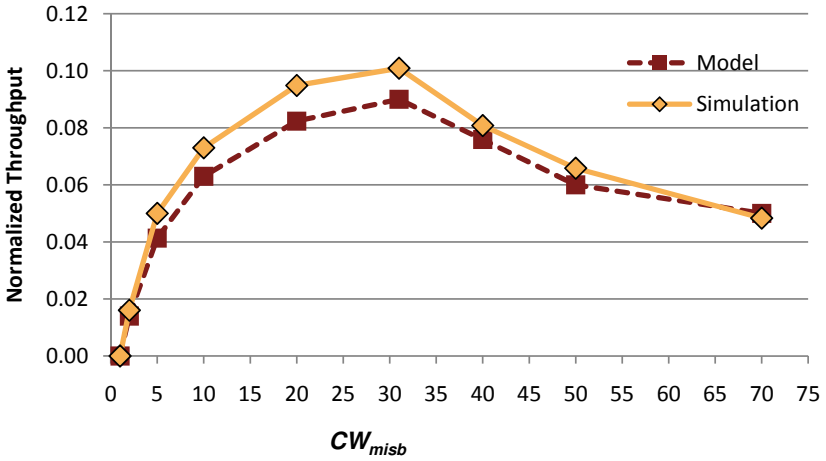

Figure 7. Throughput of penalized misbehaving node

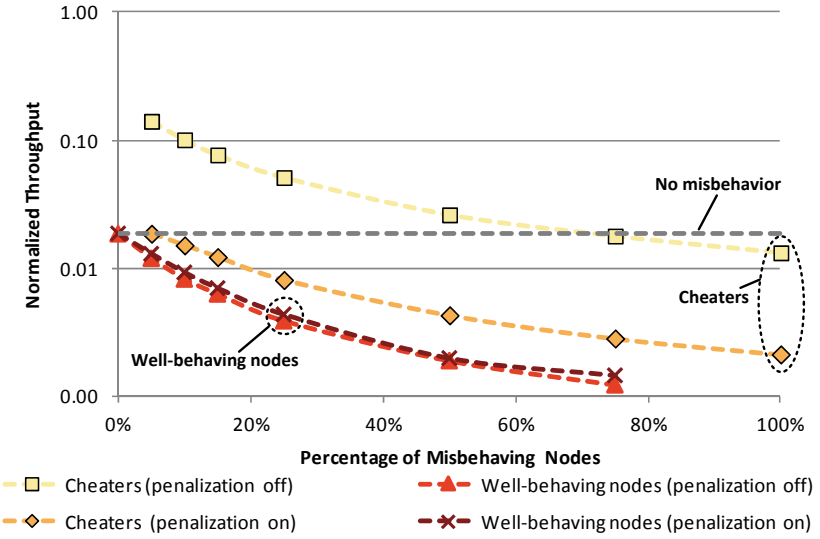

Figure 8. Impact of the percentage of misbehaving nodes in the network. Throughput is averaged over the cheaters and other (well behaving) nodes. Case A — reference (no misbehavior). Case B — misbehavior but no penalty mechanism. Case C — penalty mechanism is enabled. Case D — Penalty mechanism is enabled and cheater uses an adaptive maximization strategy.

Security Comm. Networks 0000; **00**:1–14 © 0000 John Wiley & Sons, Ltd.
DOI: 10.1002/sec
Prepared using secauth.cls

the cheaters receive up to 88% less throughput than if none of them had been misbehaving. Simultaneously, the throughput of the well-behaving nodes is not degraded.

5.4. Misbehavior on the Penalty Mechanism

The proposed penalty mechanism (whether in the form of frame jamming or refusing ACK frames) requires a slight modification of the EDCA function. While nodes which apply this penalty mechanism remain compatible with EDCA, there exists the possibility that the introduced mechanisms can become exploited by misbehaving nodes. Fortunately, this is not a severe threat for two reasons. Firstly, it can be safely assumed that players can modify only their CW values and not the implemented EDCA mechanism. This assumption is valid for modern wireless drivers, e.g., [3]. Secondly, the penalty mechanism does not provide any significant throughput gains for cooperating nodes so there is no incentive to misbehave this way. Therefore, we conclude that misbehavior on the reaction method is not a threat which can be expected from selfish nodes. It could be expected from malicious players (i.e., from users who want to attack/destabilize the network), however, such denial of service attacks can be performed through much simpler methods, e.g., by jamming the whole channel.

6. CONCLUSIONS

In this paper we have shown how QoS and security can be improved in IEEE 802.11 ad-hoc networks by mitigating the impact of selfish behavior. The presented study showed that current state-of-the-art methods for dealing with such behavior cannot be applied to the EDCA function. A penalty method was introduced to solve the analyzed problem. Using game theory analysis it was shown that the proposed method encourages standard-compliant behavior. Furthermore, the practical application of the penalty mechanism was discussed. Simulation studies confirmed that when the penalty method is used each node receives the highest throughput only if it cooperates. Additionally, results show that the method adapts to the degree of misbehavior, provides legacy node support, assures compatibility with EDCA and does not have a negative impact on the penalty or the legacy nodes. Finally, we have shown that the application of the penalty mechanism does not provide incentives to misbehave.

As future work we envision the analysis of similar security aspects in multi-hop networks. Our previous studies have shown that it is profitable for a selfish node to increase the contention window values of forwarded traffic [35]. New reaction methods are required for such networks and this is certainly a challenging research problem.

ACKNOWLEDGEMENT

This work has been partially supported by the Polish Ministry of Science and Higher Education under the European Regional Development Fund, Grant No. POIG.01.01.02-00-045/09-00 Future Internet Engineering.

REFERENCES

1. Group IW, *et al.*. IEEE 802.11-2007: Wireless LAN medium access control (MAC) and physical layer (phy) specifications 2007.
2. Szott S, Natkaniec M, Canonico R, Pach AR. Impact of Contention Window Cheating on Single-hop IEEE 802.11e MANETs. *Proc. of IEEE WCNC*, 2008.
3. Madwifi Project. URL `http://madwifi-project.org/`.
4. Sharma A, Belding EM. FreeMAC: framework for multi-channel MAC development on 802.11 hardware. *Proc. of ACM PRESTO*, 2008.
5. Bianchi G, Di Stefano A, Giaconia C, Scalia L, Terrazzino G, Tinnirello I. Experimental Assessment of the Backoff Behavior of Commercial IEEE 802.11b Network Cards. *Proc. of IEEE INFOCOM*, 2007.
6. Szott S, Konorski J. A game-theoretic approach to edca remapping attacks. *Proc. of International Conference on Wireless Communications, Networking and Mobile Computing (WiCom)*, 2012.
7. Kosek-Szott K, Natkaniec M, Pach AR. A Simple but Accurate Throughput Model for IEEE 802.11 EDCA in Saturation and Non-saturation Conditions. *Computer Networks* 2011; **55**:622–635.
8. Natkaniec M, Kosek-Szott K, Szott S. QoS Support in Multi-hop Ad-hoc Networks. *Wireless Network Traffic and Quality of Service Support: Trends and Standards*, Lagkas T, Angelidis P, Georgiadis L (eds.). IGI Global, 2010.
9. Kosek-Szott K, Natkaniec M, Pach A. BusySiMOn — a New Protocol for IEEE 802.11 EDCA-Based Ad-Hoc Networks with Hidden Nodes. *Proc. of IEEE GLOBECOM*, 2010.
10. Natkaniec M, Kosek-Szott K, Szott S, Gozdecki J, Glowacz A, Sargento S. Supporting QoS in Integrated Ad-Hoc Networks. *Wireless Personal Communications* 2011; **56**:183–206.
11. Charilas DE, Panagopoulos AD. A survey on game theory applications in wireless networks. *Elsevier Computer Networks* 2010; **54**:3421–3430.
12. Srivastava V, Neel J, MacKenzie A, Menon R, DaSilva L, Hicks J, Reed J, Gilles R. Using game theory to analyze wireless ad hoc networks. *IEEE Communications Surveys and Tutorials* 2005; **7**.
13. MacKenzie A, Wicker S. Selfish users in Aloha: a game-theoretic approach. *Proc. of IEEE VTC*, 2001.

14. Chen L, Leneutre J. Selfishness, Not Always A Nightmare: Modeling Selfish MAC Behaviors in Wireless Mobile Ad Hoc Networks. *Proc. of ICDCS*, 2007.
15. Konorski J. A game-theoretic study of CSMA/CA under a backoff attack. *IEEE/ACM Transactions on Networking* 2006; **14**:1167–1178.
16. Kunz L. Contention based access misbehavior in WLAN networks 2005.
17. Zhao L, Cong L, Zhang H, Ding W, Zhang J. Game-Theoretic EDCA in IEEE 802.11e WLANs. *Proc. of IEEE VTC*, 2008.
18. Guang L, Assi C, Benslimane A. MAC layer misbehavior in wireless networks: challenges and solutions. *IEEE Wireless Communications* 2008; **15**:6–14, doi:10.1109/MWC.2008.4599216.
19. Cagalj M, Ganeriwal S, Aad I, Hubaux JP. On Selfish Behavior in CSMA/CA Networks. *Proc. of IEEE INFOCOM*, 2005.
20. Straffin P. *Game Theory and Strategy*. Mathematical Association of America, 1993.
21. Szott S, Natkaniec M, Canonico R. Detecting backoff misbehaviour in IEEE 802.11 EDCA. *Wiley European Transactions on Telecommunications* 2011; **22**:31–34.
22. Serrano P, Banchs A, Targon V, Kukielka J. Detecting selfish configurations in 802.11 WLANs. *Communications Letters, IEEE* 2010; **14**:142–144.
23. Cárdenas A, Radosavac S, Baras J. Detection and prevention of MAC layer misbehavior in ad hoc networks. *Proc. of SASN*, 2004.
24. Guang L, Assi CM, Benslimane A. Enhancing IEEE 802.11 Random Backoff in Selfish Environments. *IEEE Transactions on Vehicular Technology* 2008; **57**:1806–1822.
25. Kyasanur P, Vaidya NH. Detection and Handling of MAC Layer Misbehavior in Wireless Networks. *Proc. of Int. Conf. on Dependable Systems and Networks*, 2003.
26. Nuggehalli P, Sarkar M, Kulkarni K, Rao R. A Game-Theoretic Analysis of QoS in Wireless MAC. *Proc. of IEEE INFOCOM*, 2008.
27. Szott S, Natkaniec M, Pach AR. An IEEE 802.11 EDCA Model with Support for Analysing Networks with Misbehaving Nodes. *EURASIP Journal on Wireless Communications and Networking* 2010; **2010**:13.
28. Wolfram. Mathematica 7. URL `http://www.wolfram.com`.
29. Banchs A, Vollero L. Throughput analysis and optimal configuration of 802.11e EDCA. *Computer Networks* 2006; **50**:1749–1768.
30. Bianchi G, Fratta L, Oliveri M. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. *Proc. of IEEE PIMRC*, 1996.
31. Qiao D, Shin K. Achieving efficient channel utilization and weighted fairness for data communications in IEEE 802.11 WLAN under the DCF. *Proc. of IEEE IWQoS*, 2002.
32. Hardin G. The Tragedy of the Commons. *Science* 1968; **162**:1243–1248.
33. Vollero L, Iannello G. Frame dropping: A QoS mechanism for multimedia communications in WiFi hot spots. *Proc. of ICPP Workshops*, 2004, doi: 10.1109/ICPPW.2004.1327996.
34. Wiethoelter S, Emmelmann M, Hoene C, Wolisz A. TKN EDCA Model for ns-2. *Technical Report TKN-06-003*, Telecommunication Networks Group, Technische Universität Berlin 2006.
35. Szott S, Natkaniec M, Banchs A. Impact of Misbehaviour on QoS in Wireless Mesh Networks. *Proc. of IFIP Networking*, 2009.

DOI: 10.1002/sec
Prepared using secauth.cls