Optimal Configuration of 802.11e EDCA for Real-Time and Data Traffic

Pablo Serrano, *Member, IEEE,* Albert Banchs, *Member, IEEE,* Paul Patras, *Student Member, IEEE,* and Arturo Azcorra, *Senior Member, IEEE*

***Abstract*— The EDCA mechanism of the 802.11e standard provides QoS support through service differentiation, by using different MAC parameters for different stations. The configuration of these parameters, however, is still an open research challenge, as the standard provides only a set of fixed recommended values which do not take into account the current WLAN conditions and therefore lead to suboptimal performance. In this paper we propose a novel algorithm for EDCA that, given the throughput and delay requirements of the stations present in the WLAN, computes the optimal configuration of the EDCA parameters. We first present a throughput and delay analysis that provides the mathematical foundation upon which our algorithm is based. This analysis is validated through simulations of different traffic sources (both data and real-time) and EDCA configurations. We then propose a mechanism to derive the optimal configuration of the EDCA parameters given a set of performance criteria for throughput and delay. We assess the effectiveness of the configuration provided by our algorithm by comparing it against i) the recommended values by the standard, ii) the results from an exhaustive search over the parameter space, and iii) previous configuration proposals, both standard and non-standard compliant. Results show that our configuration outperforms all other approaches.**

I. Introduction

The IEEE 802.11e supplement [1], included in the new revision of the 802.11 standard [2], provides wireless local area networks (WLAN's) with Quality of Service (QoS) support in the two access mechanisms specified: the *Enhanced Distributed Channel Access* (EDCA) and the *HCF Controlled Channel Access* (HCCA). Our focus is on the former, which is an extended version of the widely-supported *Distributed Coordination Function* (DCF) mechanism.

Similarly to DCF, the EDCA mechanism is based on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol. The main difference is that in the new standard different stations may contend with different values of these parameters, leading to statistical service differentiation among flows (numerical values are provided in, e.g., [3]–[6]).

Manuscript received... This work was supported in part by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o} 214994. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the CARMEN project or the European Commission.

P. Serrano, A. Banchs, P. Patras and A. Azcorra are with the Department of Telematics Engineering, University Carlos III de Madrid, 28911 Leganés, Spain (e-mail: {pablo,banchs,patras,azcorra}@it.uc3m.es)

A. Banchs, P. Patras and A. Azcorra are also with IMDEA Networks, 28912 Leganés, Spain.

When deploying an EDCA WLAN, the main challenge is the configuration of the contention parameters, as the standard provides only a set of *recommended* values. However, using this configuration for every scenario, regardless of, e.g., the number of stations or the traffic patterns, leads to suboptimal performance in most circumstances. Therefore, a configuration mechanism to derive the contention parameters is needed. Furthermore, this mechanism should not be based on heuristics but rather on an analytical model that provides strong mathematical foundations in order to guarantee optimal performance.

In this paper, we build upon our previous work to achieve a two-fold objective:

- First, we present a novel analytical model of EDCA performance that accounts for generic saturated and non-saturated sources, and provides as performance figures the average throughput, average delay, and standard deviation of the delay. To our knowledge, this is the most complete model of EDCA proposed to date and the only one that has all these features.
- Second, we use this new analytical model to develop a configuration mechanism for the parameters of EDCA that, taking as input the traffic requirements from both real-time and non-real-time stations, outputs the configuration that maximizes performance: it admits as many real-time traffic stations as possible while optimizing non-real-time throughput. To the best of our knowledge, this is the first approach to configure EDCA that covers all traffic types and is sustained analytically, thereby guaranteeing optimal performance[1].

The rest of the paper is structured as follows. In Section II we review the state of the art. In Section III we describe our analytical model and validate it through exhaustive simulations. The optimal configuration mechanism is introduced in Section IV, along with the results from the numerical search to prove the effectiveness of our algorithm as well as a comparison against previous approaches. Finally, concluding remarks are given in Section V.

II. State of the Art

In this section we present the state of the art. We first summarize the behavior of the EDCA mechanism and then we review previous analyses and approaches to configure EDCA.

[1]Note that the analytical model requires a series of assumptions. Therefore, when we use the term "optimal configuration" we are referring to the configuration that provides the best performance according to this model.

A. IEEE 802.11e EDCA

This section briefly summarizes the EDCA mechanism as defined in the 802.11e standard. EDCA is a CSMA/CA-based protocol that extends DCF by means of the parameters used to access the channel. The channel access is regulated by the Channel Access Functions (CAFs). To transmit its frames, each CAF executes an independent backoff process that is regulated by a number of configurable parameters. For the configuration of these parameters, the standard groups the CAFs by Access Categories (ACs) and assigns the same configuration to all the CAFs of an AC. In this paper we assume for simplicity that each station runs only one CAF and therefore use the terms CAF and station interchangeably[2].

A station of an Access Category i ($AC\ i$) with a new frame to transmit monitors the channel activity. If the channel is sensed idle for a period of time equal to the arbitration interframe space parameter of this AC ($AIFS_i$), the station transmits. Otherwise, if the channel is sensed busy (either immediately or during the $AIFS_i$ period), the station continues to monitor the channel until it is measured idle for an $AIFS_i$ time, and, at this point, the backoff process starts. The arbitration interframe space $AIFS_i$ takes a value of the form $DIFS + nT_e$, where $DIFS$ and T_e are constants dependent on the physical layer and n is a nonnegative integer[3].

Upon starting the backoff process, the station computes a random value uniformly distributed in the range $(0, CW_i - 1)$, and initializes its backoff time counter with this value. The CW_i value is called the contention window and depends on the number of transmission attempts for the current frame. At the first transmission attempt, CW_i is set to be equal to the minimum contention window parameter (CW_i^{min}). As long as the channel is sensed idle, the backoff time counter is decremented once for each time interval T_e.

When a transmission is detected on the channel, the backoff time counter is "frozen", and reactivated again after the channel is sensed idle for a certain period. This period is equal to $AIFS_i$ if the transmission is received with a correct Frame Check Sequence (FCS). Otherwise, this period is equal to $EIFS - DIFS + AIFS_i$, where $EIFS$ is another constant dependent on the physical layer.

As soon as the backoff time counter reaches zero, the station transmits its frame in the next slot time. A collision occurs when two or more stations start a transmission simultaneously. An acknowledgment (Ack) frame is used to notify the transmitting station that the frame has been successfully received. If the Ack is not received within a timeout, the station assumes that the frame was not received and reschedules the transmission by reentering the backoff process. After each unsuccessful transmission, CW_i is doubled, up to a maximum value given by the CW_i^{max} parameter. If the number of failed attempts reaches a predetermined retry limit R, the frame is discarded.

When the station gains access to the channel, it is allowed to retain the right to access it for a duration equal to the transmission opportunity limit parameter ($TXOP_i$). Note that the impact of the $TXOP_i$ parameter is typically small in QoS-provisioned scenarios, as real-time traffic parameters are usually set such that queues never grow above one packet, while for data traffic this parameter is set such that only one packet is transmitted upon accessing the channel to avoid degrading the delay performance of real-time traffic. Following this reasoning, in the rest of this paper, we concentrate on the analysis of the other three parameters (CW_i^{min}, CW_i^{max}, and $AIFS_i$).

B. Related Work

There are several analytical models of EDCA performance available in the literature [7]–[20]. However, most of them [7]–[14] are based on the unrealistic assumption that all stations always have packets ready for transmission (commonly referred to as *saturation conditions*). While this assumption may be reasonable for data traffic, it does not hold for real-time traffic. On the other hand, previous approaches assuming non-saturated conditions [15]–[20] are typically valid only for Poisson arrivals and fixed length packets. In contrast to these previous papers, our analysis does not make any assumption about the arrival process and allows for variable packet lengths.

The analysis presented in this paper combines and extends our previous work, providing the most comprehensive analysis of EDCA to date, including generic traffic sources as well as the relevant metrics for data and real-time traffic (namely throughput, average and standard deviation of the delay). In particular, the analysis extends our previous work as follows:

- In [20], we presented an analysis of EDCA under non-saturated traffic conditions to model throughput and average delay. In this paper we also account for the standard deviation of the delay.
- In [13], we analyzed the average delay performance of EDCA. While [13] is limited to saturation conditions, the present analysis also considers non-saturation traffic.
- In [21], we analyzed the standard delay deviation when there is only voice traffic present in the WLAN. In this paper we extend this analysis to the case where there are multiple ACs.

The differences between the model presented in this paper and previous work are summarized in Table I. We observe that the proposed model is more complete than any of the previous models.

Only recently has the challenge of configuring the EDCA parameters been addressed [7], [21]–[27]. However, the existing approaches suffer from major drawbacks. Our previous works of [7] and [22] are restricted to data traffic, while our works of [21] and [23] are restricted to voice traffic. The works of [24] and [25] only consider two traffic types, voice and data, and do not allow different types of real-time and non-real-time[4] traffic. The default configuration recommended

[2]Note that, following [7], the analysis here could easily be extended to the case of multiple CAFs per station.

[3]According to the IEEE 802.11e standard terminology, $AIFS_i = SIFS + nT_e$, where $DIFS = SIFS + 2T_e$ and $n \geq 2$. Without loss of generality, in this paper we use the simplified notation $AIFS_i = DIFS + nT_e$, with $n \geq 0$.

[4]Throughout the paper we will use the terms "data" and "non-real-time" interchangeably.

TABLE I

COMPARISON BETWEEN THE ANALYTICAL MODELS OF EDCA PERFORMANCE.

	Ours	[7]–[11]	[12], [13]	[17], [19]	[15]	[16], [18]	[20]	[21]
CW	✓	✓	✓	✓	✓	✓	✓	✓
AIFS	✓	✓	✓	.	✓	.	✓	.
Average Delay	✓	.	✓	✓	✓	.	✓	✓
Standard Dev	✓	.	.	.	.	.	.	✓
Non-saturation	✓	.	.	✓	✓	✓	✓	✓
Generic sources	✓	.	.	.	.	.	✓	.

TABLE II

NOTATION USED IN THE ANALYSIS.

Variable	Description
N	Number of ACs
n_i	Number of stations of $AC\ i$
l_i	Average length of frames from $AC\ i$
A	Largest A_i in the WLAN
ρ_i	Average sending rate of $AC\ i$
τ_i	Transmission probability of a station of $AC\ i$
r_i	Throughput of a station of $AC\ i$
Δ_k	Set of ACs with $A_i \leq k$
$p(\Delta_k)$	Prob. that in a slot only set Δ_k can transmit given only the set Δ_k can transmit
$p(t_k)$	Probability that a slot is a k-slot
$p(e\|t_k)$	Prob. that a k-slot is empty
$p(e)$	Prob. that a slot is empty
$p(s_i)$	Prob. that a slot contains a success of $AC\ i$
$p(c)$	Prob. that a slot contains a collision
$p(c_i)$	Collision prob. of an attempt of $AC\ i$
$p(s)$	Prob. that a slot contains a success
$p(s_i\|\Delta_k)$	Prob. that a slot contains a success of $AC\ i$

by the standard [2], the one recommended in [26] and the adaptive mechanism of [27] consider all traffic types, but they are based on heuristics and therefore do not guarantee optimal performance. Indeed, the performance evaluation conducted shows that our proposal substantially outperforms these previous proposals.

In addition to the above, a number of modifications of the EDCA protocol have been recently proposed [28]–[32]. These proposals have the major drawback of not being standard compliant and requiring modifications to the hardware and firmware of the wireless cards, which challenges their practical deployment. The proposal in [28] applies only to one AC, while the one in [29] supports only voice and best effort traffic. The approach proposed in [30] prevents data stations from transmitting when the contention level exceeds a certain threshold, which has the shortcoming of starving them. Finally, the approaches of [31] and [32] are based on heuristics; our simulation results show that our approach, even without introducing modifications to EDCA, clearly outperforms them.

III. PERFORMANCE ANALYSIS

In this section we consider a WLAN operating under the EDCA mechanism and analyze the throughput and the delay of each AC in the WLAN.

A. Definitions, Terminology and Assumptions

In the following we present the key definitions, terminology and assumptions upon which our analysis is based. A summary of the notation and variables used in the analysis is provided in Table II. In particular, our analytical model takes the following input variables:

- The number of ACs in the WLAN (N).
- The number of stations of each AC (n_i is the number of stations of $AC\ i$).
- The average sending rate of the stations of each AC (ρ_i), their frame length distribution, and the average frame length (l_i).
- The configuration $\{CW_i^{min}, m_i, A_i\}$ of each AC, where m_i is defined such that $CW_i^{max} = 2^{m_i} CW_i^{min}$, and A_i such that $AIFS_i = DIFS + A_i T_e$.

and provides as output the throughput, average delay and standard deviation for each AC.

Note that our model can be applied to analyze generic source models. The only restriction imposed on the sources is that they are ergodic, as otherwise the analysis could not rely on the stations' average sending rate.

Our analysis is based on the following definitions:

Definition 1: A *slot time* is the time interval between two consecutive backoff counter decrements of a station with minimal $AIFS_i$ (i.e., $DIFS$). We say that a slot time is nonempty when it contains a collision or a successful transmission and that it is empty otherwise.

Definition 2: A slot time is a k*-slot time* if it is preceded by k or more empty slot times.

Definition 3: The *saturation rate* of an AC is the rate that the stations of this AC would obtain if they always had a packet ready for transmission.

Based on these definitions, our analysis relies on a number of assumptions. First, we make the following two key approximations around the notion of saturation rate to compute the stations' rates in the WLAN:

- As long as the average sending rate of the stations of a given AC falls below their saturation rate, we assume that the stations of this AC see all their packets served (i.e., their transmission queue never overflows). We refer to such an AC as a *non-saturated* AC.
- On the other hand, if the average sending rate of the stations of the AC exceeds the saturation rate, we consider that the stations of this AC always have packets ready for transmission (i.e., their transmission queue never empties). We refer to such an AC as *saturated*.

In addition to the above two approximations, our analysis further relies on the following additional assumptions which have already been used in previous works in the literature:

- Backoff times follow a geometric distribution (i.e. a station transmits upon decrementing its counter with an independent probability). This assumption was first used in the analysis of [33] for 802.11 DCF, and since then it has been used in most of the analyses of EDCA (see e.g. [8]–[10]). Even though backoff times actually follow a uniform distribution, all these works have shown that this assumption leads to accurate results.

- Each packet transmission attempt collides with an independent probability. This assumption, initially used in [34], has been the basis of most of the WLAN performance analyses so far.
- The length of a slot time can be modeled with a random variable that depends only on the stations that could potentially transmit in this slot time. This is also a common assumption when analyzing the delay performance of EDCA (see e.g. [11]).
- Finally, at each transmission attempt the packet length follows a random variable that depends only on the considered $AC\ i$. This assumption is necessary for the tractability of the analysis, and has been used and shown to be accurate in previous analyses dealing with variable packet lengths in WLAN (see e.g. [20], [34]).

We build our analysis upon the variable τ_i, defined as the probability that a station of $AC\ i$ transmits upon a backoff counter decrement. Note that, since a station with $A_i = k$ starts decrementing its backoff counter only after k empty slot times following a nonempty slot time, we see that the backoff counter decrements of this station coincide with the boundaries of the k*-slot times*. Therefore, a station of $AC\ i$, with $A_i = k$, transmits in a k*-slot time* with probability τ_i, and does not transmit in any other slot time (see Fig. 1).

In the following, we first analyze separately the τ_i of a saturated AC and the τ_i of a non-saturated AC and combine both analyses to compute the τ_i values of all the ACs in the WLAN. Then, based on these values, we calculate the throughput and delay performance of each AC.

B. Point of Operation of the WLAN

We first compute the point of operation of the WLAN as given by the transmission probabilities τ_i's of all the AC's. We start with the case of a saturated AC [13]. With the assumption that each transmission attempt collides with a constant and independent probability, we can model the behavior of this AC with the same Markov chain as Fig. 5 of [35]. Then, the probability that a station of a saturated AC transmits upon a backoff counter decrement can be computed by means of the following equation given by [35]

$$\tau_i^{sat} = \frac{2(1-2p(c_i)(1-p(c_i)^{R+1})}{CW_i^{min}(1-(2p(c_i)^{m_i+1})(1-p(c_i))+(1-2p(c_i)(1-p(c_i)^{R+1})+ +CW_i^{min}2^{m_i}p(c_i)^{m_i+1}(1-2p(c_i)(1-p(c_i)^{R-m_i})} \quad (1)$$

where $p(c_i)$ is the probability that a transmission attempt of a station of $AC\ i$ collides.

We next focus on the analysis of a non-saturated AC. The goal of this analysis is to compute the probability that a non-saturated station transmits in a slot time, τ_i^{nonsat}. Note that, in contrast to the τ_i of a saturated station which depends exclusively on the backoff process, τ_i^{nonsat} also accounts for the inactivity periods of the station caused by its queue being empty. The following lemma let us compute the τ of a non-saturated AC based on variables that, as shown in Appendix, can be expressed as a function of the τ_i's and $p(c_i)$'s.

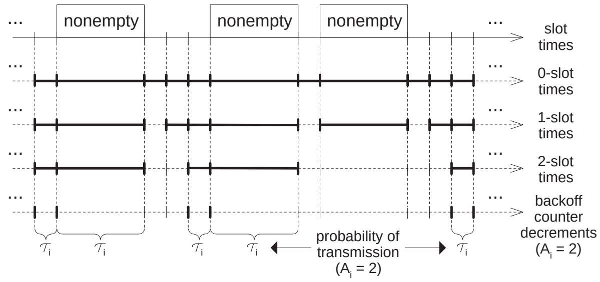


Fig. 1. k*-slot times* and probability of transmission (example with $k = 2$).

Lemma 1: The τ_i of a non-saturated AC is given by[5]

$$\tau_i^{nonsat} = \frac{\rho_i(1-p(c_i)^{R+1})\,(p(s)T_s + p(e)T_e + p(c)T_c)}{l_i(1-\tau_i)^{n_i-1}\sum_{k=A_i}^{A} p(\Delta_k)\prod_{j\in\Delta_k\setminus i}(1-\tau_j)^{n_j}} \quad (2)$$

where $p(s)$, $p(c)$, and $p(e)$ are the probabilities that a slot time contains a successful transmission, a collision, or is empty, respectively, and T_s, T_c, and T_e are the average slot time durations in each case. Δ_k is the set of ACs with $A_i \leq k$, and $p(\Delta_k)$ is the probability that a randomly chosen slot time is allowed for transmission to the set Δ_k.

With the above, we can express the τ_i's as a function of the rest of the τ_i's and $p(c_i)$. In order to build a system of equations, we need to express $p(c_i)$ as a function of the rest of the τ_i's. We compute $p(c_i)$ as a function of the probability of an empty k*-slot time* (denoted by $p(e|t_k)$) as follows. A k*-slot time* is empty as long as i) the considered station does not transmit, and ii) no other station transmits. The latter can be expressed as a function of $p(c_i)$ by noting that the probability of a collision corresponds to the case when some other station transmits. Thus,

$$p(e|t_k) = (1-\tau_i)(1-p(c_i)) \quad (3)$$

which yields

$$p(c_i) = 1 - \frac{p(e|t_k)}{1-\tau_i} \quad (4)$$

Now let us focus on the probability that a given k*-slot time* is empty. If the previous k*-slot time* was nonempty, in this k*-slot time* only the ACs with $A_i \leq k$ may transmit. If the previous k*-slot time* was empty, the given k*-slot time* is preceded by $k+1$ or more empty slot times, which is exactly the definition of (k+1)*-slot time*, and therefore such a k*-slot time* is empty with probability $p(e|t_{k+1})$. Applying this reasoning (see Fig. 2), $p(e|t_k)$ can be written as

$$p(e|t_k) = (1-p(e|t_k))\prod_{j\in\Delta_k}(1-\tau_j)^{n_j} + p(e|t_k)p(e|t_{k+1}) \quad (5)$$

[5]The proofs of all lemmas are derived in the Appendix.

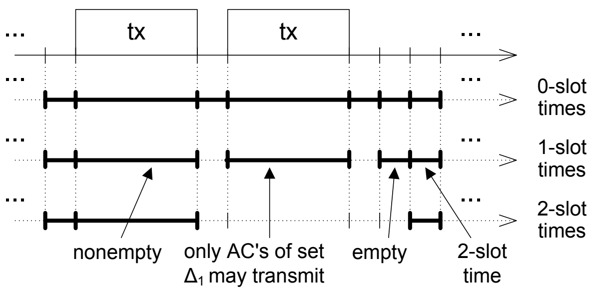


Fig. 2. Probability of an empty k-*slot time* (example with $k = 1$).

Note that, if A is the largest A_i in the WLAN, in a A-*slot time* all stations may transmit, therefore the following equation holds

$$p(e|t_A) = \prod_{j \in \Delta_A} (1 - \tau_j)^{n_j} \tag{6}$$

Starting from $\tau_i \ \forall i$, with (6) we can compute $p(e|t_A)$. Then, with (5) we can compute $p(e|t_{A-1})$. Applying this recursively, we can compute $p(e|t_k) \ \forall k$. Then, $p(c_i)$ can be computed using (4) and, finally, τ_i can be obtained from (1).

We next combine the analyses for a saturated and a non-saturated AC in order to obtain all the τ_i's in the WLAN under stationary conditions. From the above we have a method to compute the τ_i of a saturated and of a non-saturated AC; the remaining challenge lies in determining which ACs are saturated and which are not. For this purpose, we proceed step by step as follows:

- In the first step, we consider that all ACs are saturated. Note that, from (1) and (2), we can express each τ_i of a saturated (or non-saturated) AC as a function of all the τ_i's. Therefore, we have a system of N non-linear equations on the τ_i's that can be resolved using numerical techniques. Once the τ_i values have been derived, we compute the throughput of all ACs by using Lemma 2 in Section III-C. We next compare the throughputs against the sending rates. If the throughput of an AC is larger than its sending rate, we consider from this step on that this AC is not saturated, and move it to the set of non-saturated ACs.
- In the second step, we take the new sets of saturated and non-saturated ACs resulting from the first step and repeat the throughput computation. Next, we compare again the throughputs obtained in the previous step for the saturated ACs against their sending rates, and move those ACs whose throughputs are larger than their sending rates to the set of non-saturated ACs[6].
- The above is done iteratively until the resulting throughputs of all the saturated ACs are smaller than their sending rates. This last scenario represents a stable solution, and therefore the values from this step give us the throughput that each AC will obtain in the WLAN under stationary conditions.

Note that, as the number of ACs (N) is limited to 4 by the standard, the above procedure requires in the worst case that we resolve at most 4 times a system of no more than 4 equations, and therefore the computational complexity is low.

C. Throughput and Delay Analysis

Once the values τ_i's have been derived, we can analyze the throughput and delay performance of the WLAN. More specifically, in the following we analyze the average throughput, the average service delay and the standard deviation of the delay[7].

The throughput r_i is given by the following lemma

Lemma 2: The average throughput r_i a station from $AC\ i$ experiences is given by

$$r_i = \frac{l_i \sum_{k=A_i}^{A} p(\Delta_k) p(s_i|\Delta_k)}{p(s)T_s + p(c)T_c + p(e)T_e} \tag{7}$$

where $p(s_i|\Delta_k)$ is the probability that, given only stations form set Δ_k can transmit, there is a successful transmission from $AC\ i$.

We next compute the delay performance of the WLAN. For this purpose, we define $B_{i,r}$ as the average backoff counter before retry r, $T^i_{slot,k}$ as the average duration of a k-*slot time* in which the considered station of $AC\ i$ does not transmit, $T^i_{inter\_tx,k}$ and $T^i_{inter,k}$ as the average durations of the time between two k-*slot times* when the considered station transmits and does not transmit in the first one, respectively, and $T_{s,i}$ and $T_{c,i}$ as the average durations of a slot time that contains a success and a collision involving a station of $AC\ i$. In Figs. 3 and Fig. 4 we illustrate these delay components for a given sequence of slot times. Based on these variables, lemma 3 provides the average value of the delay d_i.

Lemma 3: The *average delay* experienced by a non dropped packet of a station of $AC\ i$ is given by

$$d_i = \frac{1}{\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j} \sum_{j=0}^{R} (1-p(c_i))p(c_i)^j (jT_{c,i} + T_{s,i} + \sum_{r=0}^{j} (T^i_{inter\_tx,k} + B_{i,r}(T^i_{slot,k} + T^i_{inter,k}))) \tag{8}$$

Finally, the following lemma gives the value of the standard deviation of the delay.

Lemma 4: The *standard deviation* of the delay is given by

$$\sigma^2_{d_i} = \frac{\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j E[(d_{i,j})^2]}{\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j} - (d_i)^2 \tag{9}$$

where $d_{i,j}$ is the delay that a frame from $AC\ i$ suffers in case of j retries.

[6]Note that an AC that was not saturated in the previous step can never become saturated again. In fact, if such an AC always had packets ready for transmission, it would obtain a throughput even larger than in the step where it became non-saturated (since in the current step there are fewer saturated ACs).

[7]Given the average delay and its standard deviation, it is possible to provide guarantees on the delay distribution by means of the Chebyshev inequality [36]. In this paper we do not further discuss this and simply assume that the average delay and the standard deviation are sufficient to provide real-time traffic with the desired service guarantees.

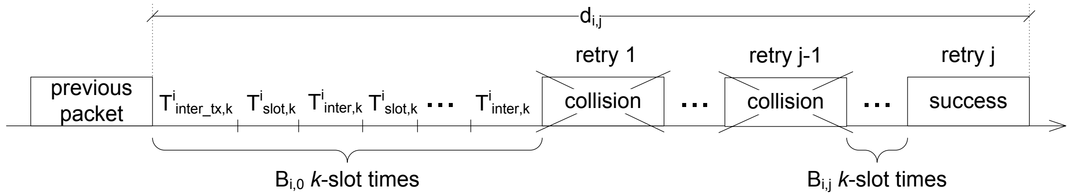


Fig. 3. Average delay components for the case of j retries.

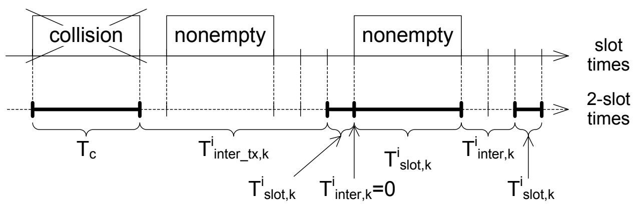


Fig. 4. Components of the delay (example with $k = 2$).

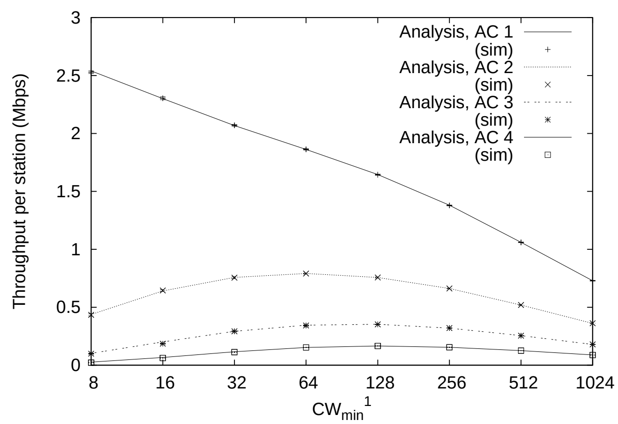


Fig. 5. Validation of the throughput model for a scenario with 4 ACs, each using a different $AIFS$ and CW configuration.

The above lemma terminates our performance analysis of EDCA. The following section is devoted to the assessment of its accuracy under different traffic sources and EDCA configurations.

D. Performance Analysis Validation

We validate the accuracy of the model by comparing the analytical values against those obtained by means of simulations. For this purpose we have implemented the 802.11e EDCA protocol in OMNeT++[8]. The source code of our simulations is available in our website[9]. The simulations are performed for a WLAN with the MAC layer parameters of IEEE 802.11b [37]. We assume a channel in which frames are only lost due to collisions. The queue size of all of the stations is set equal to 100 packets. For the simulation results, average and 95% confidence interval values are given, although in many cases confidence intervals are too small to be appreciated in the graphs. The values obtained analytically are plotted with lines, and the simulation results are plotted with points.

1) Data traffic: First, we analyze our throughput model for the case when only data traffic is present in the network, with no delay requirements. We have taken a fixed frame payload size of 1500 bytes and $m_i = 5$ (i.e., $CW_i^{max} = 2^5 CW_i^{min}$).

We consider a scenario with 4 ACs, $i \in \{1, \ldots, 4\}$, with $n_i = 2$ stations each, sharing the channel with a different CW_i^{min} and $AIFS_i$ each. Specifically, we take $CW_i^{min} = 2^{i-1} CW_1^{min}$ for $i \in \{2, 3, 4\}$ and $A_i = i - 1$ for $i \in \{1, \ldots, 4\}$. Results are given in Fig. 5. The simulations performed validate our model for data traffic, as simulation results match the analytical ones well.

[8]`http://www.omnetpp.org`
[9]`http://enjambre.it.uc3m.es/~ppatras/owsim/`

2) Voice traffic: Next, we validate the accuracy of our analysis by comparing analytical results against simulations in a scenario where only voice traffic is present. Following the behavior of standard PCM codecs (e.g., G.711), voice sources generate one 80 byte packet every 10 ms.

Figs. 6 and 7 plot the average and standard deviation of the delay, respectively, for different configurations of the CW_{min} parameter and different numbers of voice stations. The three values chosen for the number of voice stations, $n \in \{10, 15, 20\}$, correspond to a low, medium, and heavily loaded WLAN, respectively. We observe that the analytical results match the simulations remarkably well, which confirms the accuracy of our analysis.

We observe from Fig. 6 that the evolution of the delay vs. the CW shows a non-monotonous behavior. Indeed, there is at first a steep decrease of the delay, reaching the minimum value, and then there is a slow increase. This is caused because, with small CW values, there are many collisions in the WLAN which causes congestion. When using larger CW, collisions take place less frequently and the WLAN moves out of a congested situation; the steep decrease corresponds to this change from congestion to out of congestion. Then, after reaching the minimum value, there is a "graceful degradation" of the delay, caused by the use of larger backoff counters than needed to prevent congestion[10].

3) Voice and data traffic: Next, we validate the model for the case of a WLAN operating with both data and voice traffic.

[10]For a detailed analysis of this behavior, see [21].

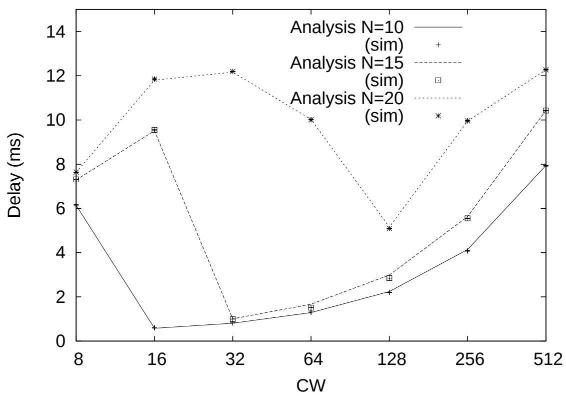


Fig. 6. Validation of the average delay model for a voice traffic scenario with different configurations of the CW used.

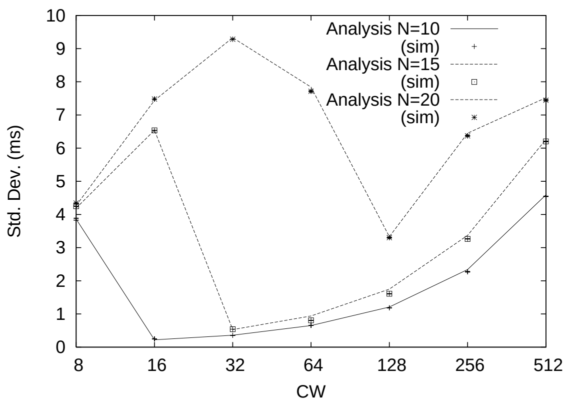


Fig. 7. Validation of the model for the standard deviation of the delay under voice traffic and different CW configurations.

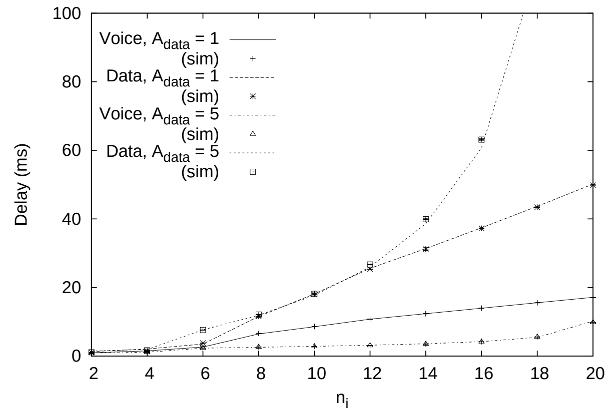


Fig. 8. Validation of the average delay model for a mixed scenario with voice and data stations and different $AIFS$ differentiation.

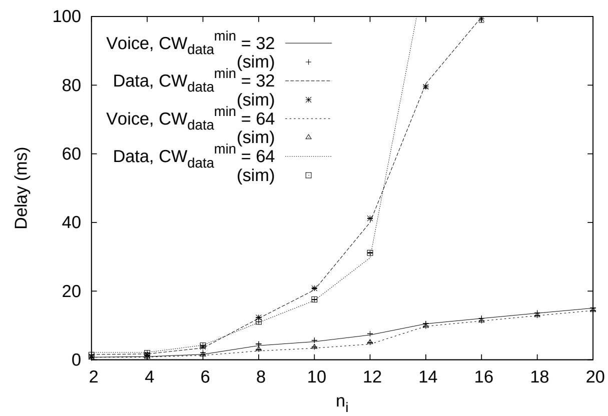


Fig. 9. Validation of the average delay model for a mixed scenario with voice and data stations and different CW differentiation.

The validation is performed using two ACs, both with the same number of stations:

- The first group (voice stations) transmits 80–byte packets every 10 milliseconds.
- The second group (data stations) transmits according to a Poisson process with an average rate of 500 Kbps and the packet lengths derived from the measurements in [38].

For validating our model, we perform the following experiments:

- First, an experiment to validate the analysis of the differentiating effect of the $AIFS$ parameter. To this aim, both ACs have the same contention window configuration $CW_{min} = 32,\ m = 5$. Regarding the A_i parameter, the voice AC is always configured with $A_i = 0$, while for the configuration of the data AC we use two different values: $A_i = 1$ (small differentiation) and $A_i = 5$ (large differentiation).
- Similarly, we assess whether our model captures the differentiating effect of the CW parameter by means of the following configuration: $A_i = 0$ for the two ACs and $CW_{min} = 16,\ m = 1$ for voice, while for data traffic we use $CW_{min} = 32,\ m = 4$ in one case and $CW_{min} = 64,\ m = 4$ in the other case.

The results for the average delay are shown in Figs. 8 and 9. As the results from the analytical model closely follow the simulation values, we conclude that the proposed model is valid also for this case. It is worth remarking the degree of service differentiation that the $AIFS$ and CW parameters provide: for the case of $AIFS$, the differentiation is strong only when there is enough traffic on the WLAN (i.e., n_i is relatively large). On the other hand, the CW parameter provides a larger level of differentiation.

The evaluation of the analysis of the standard deviation of the delay is depicted in Figs. 10 and 11. As in the previous case, the model follows closely the simulation results, which confirms the validity of our analysis for this performance metric as well. We further observe that, compared to the average delay, the standard deviation is more sensitive to the increase in load.

4) Mixed traffic: We finally validate our model for the more general case, with up to four traffic classes of different characteristics, which we name “voice”, “video”, “data”, and “background”, respectively:

- In the first AC (voice), 80 byte packets are generated every 10 ms.
- In the second AC (video), we model video traffic with a variable bit rate source sending variable size packets at

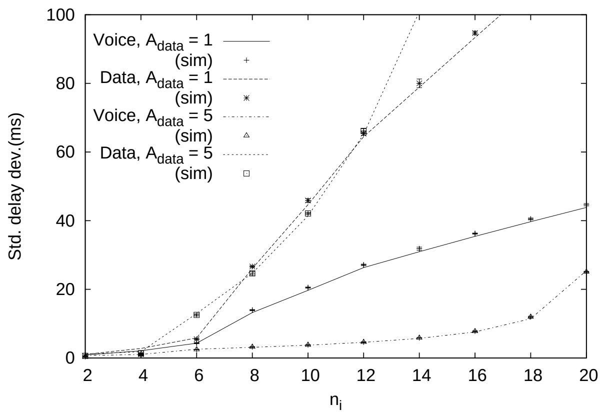


Fig. 10. Validation of the model of the standard deviation of the delay for a mixed scenario with voice and data stations and different $AIFS$ configurations.

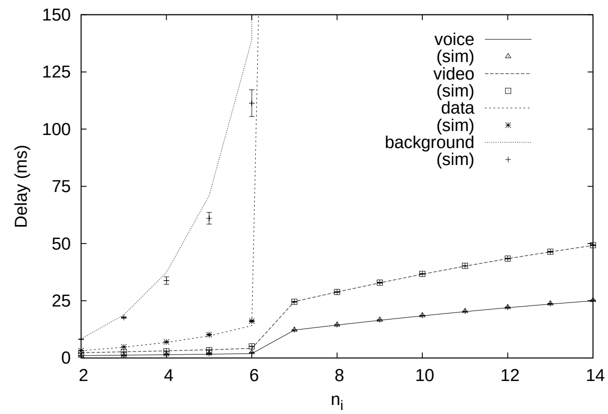


Fig. 12. Validation of the average delay model for a scenario with 4 ACs configured according to the standard recommended values.

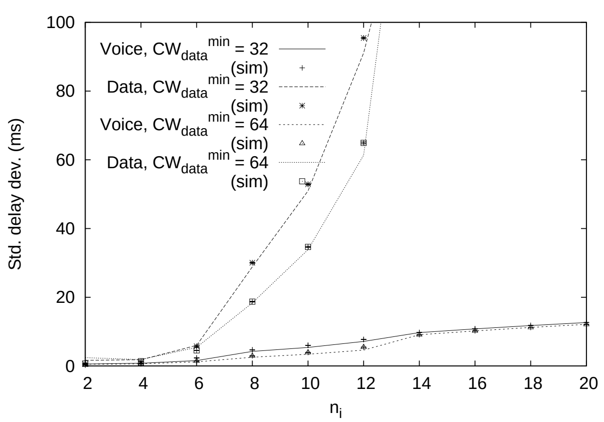


Fig. 11. Validation of the model of the standard deviation of the delay for a mixed scenario with voice and data stations and different CW configurations.

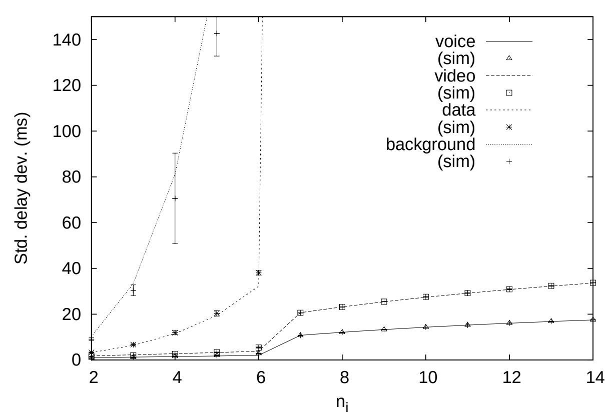


Fig. 13. Validation of the model for the standard deviation of the delay for a scenario with 4 ACs configured according to the standard recommended values.

a constant interarrival time. The average bit rate of the source is set equal to 250 Kbps and the packet length distribution is taken from the video traffic measurements of [39].

- In the third AC (data), stations always have a packet ready for transmission, modeling the behavior of a data transfer. Packet sizes are taken from the data traffic measurements of [38].
- In the fourth AC (background), stations always have 1000 byte packets ready for transmission.

The configuration of each AC is derived from the recommendations given in the 802.11e standard for 802.11b (Table III). Experiments are performed for a varying number of stations per AC (each AC has n_i stations). Figs. 12 and 13 plot the average and standard deviation of the delay, respectively. The validation of the throughput model is depicted in Fig. 14.

We observe from the figures that EDCA is effective in providing service differentiation. Both in terms of throughput and delay, higher priority ACs always perform better than lower priority ones. Furthermore, higher priority ACs also saturate later: AC 3 (data) saturates for $n_i > 4$ while ACs 1 and 2 (voice and video) saturate for $n_i > 6$ (AC 4 is by definition always saturated). Beyond this saturation point, the throughput of all ACs decreases gradually with n_i, while delay increases drastically. For all cases, the analytical results match the simulations remarkably well, confirming the accuracy of our model.

IV. Optimal Configuration

In this section, we present an algorithm to find the optimal configuration of the EDCA parameters under a general scenario with multiple real-time and data ACs. The objectives of our algorithm are:

1) Meet the requirements of the real-time traffic. More specifically, the configuration should provide real-time stations with the required throughput and delay guarantees.

TABLE III
EDCA Configuration

	A_i	CW_i^{min}	CW_i^{max}
voice	0	8	16
video	0	16	32
data	1	32	1024
background	5	32	1024

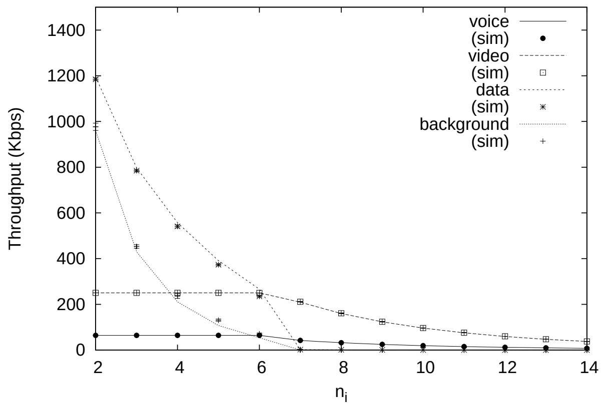


Fig. 14. Validation of the throughput model for a scenario with 4 ACs configured according to the standard recommended values.

2) Maximize the admissibility in the network. Specifically, we aim to admit as many real-time stations as possible, while satisfying the previous objective.
3) Maximize the throughput received by the data traffic while meeting the previous two objectives. For throughput allocation we use the common weighted max-min fair allocation criterion [40]–[42]. This maximizes the minimum r_i/w_i in the system, r_i being the throughput allocated to entity i and w_i the entity's weight. In our case, the *entities* are the WLAN stations and the *allocated throughput* is the saturation throughput of a station.

A. Considerations for Optimal Configuration

Before proceeding with the design of our algorithm we make the following considerations on the configuration of some of the EDCA parameters. This simplifies the design of the algorithm and reduces its computational cost.

The considerations for the *data ACs* are the following:

- From (1), we have that τ_i can be adjusted as a function of two parameters, CW_i^{min} and m_i. As a consequence, we have one degree of freedom when setting these parameters in order to obtain the desired τ_i. Following this, we fix $m_i = 0$. Then, by substituting m_i with 0 in (1), we compute the CW_i^{min} value that leads to τ_i^{opt} as follows
$$CW_i = \frac{2}{\tau_i^{opt}} - 1. \tag{10}$$
- $A_i = A_d \ \forall i$: Following the proof in [7], the throughput of the data stations is maximized when they all use the same A_i setting.
- $TXOP = 1$ packet. This ensures that a station transmitting data sends only one packet upon accessing the channel and thus reduces the delay inflicted on real-time traffic.

For the case of *real-time ACs*, we make the following considerations:

- $A_i = 0$: The optimal setting for this parameter is its minimum possible value, namely, $AIFS = DIFS$, as otherwise additional time is unnecessarily lost after every transmission.
- $CW_j^{min} = CW_j^{max} = CW_j$: When the number of stations in the channel is unknown, CW_{max} is typically set larger than CW_{min}, so that after a collision the CW increases and thus the probability of a new collision is reduced. However, this is not necessary in our case, as the number of stations is known and therefore their CW_{min} can be directly configured for optimal operation. In addition, if we set CW_{max} larger than CW_{min}, the delay of the packets that suffer one or more collisions drastically grows, which harms jitter performance.
- $TXOP_j = TXOP_{max}$: Considering the strict delay requirements of real-time traffic, the EDCA parameters will be chosen such that the transmission queues of the stations almost never grow to more than one packet (in particular, this holds for the configurations that we later propose). In the eventual case that queues grow above one packet, it is desirable that, upon accessing the channel, all waiting packets are transmitted in order to minimize their delay. To achieve this, we set the $TXOP$ parameter to its maximum allowed value.

Following the above considerations, our algorithm provides the configuration for the following set of parameters which are left open: the A_i configuration for the data stations, the CW_i parameters for each of the real-time ACs, and the CW_i for each data AC.

B. Optimal Configuration Algorithm

The proposed algorithm is based on a numerical search we perform over the τ_i of one data AC, which we take as reference. In each step of the search, given the value of the τ_i of this reference AC, we need to compute the τ_j of the other ACs. In order to obtain the τ_j of the other data ACs, we apply the max-min fair allocation criterion to the throughput expression given in (7), which yields [7]:

$$\frac{\tau_i(1-\tau_j)}{\tau_j(1-\tau_i)} = \frac{w_i}{w_j} \tag{11}$$

Once the τ_j's of each data AC are known, the other τ_l's can be obtained as follows. Neglecting the probability of a drop due to reaching the maximum retry limit, we have $r_i \approx \rho_i$. Furthermore, by applying (7) to r_l/r_k and making the approximation $p(s_l)/p(s_k) \approx \tau_l/\tau_k$, we obtain

$$\frac{\tau_l}{\tau_k} \approx \frac{\rho_l/l_l}{\rho_k/l_k} \tag{12}$$

where, hereafter, we will denote the rhs of the above equation by K_l.

With the above, we can derive a third-order equation to calculate approximately the τ_k of a reference real-time AC, given all the τ_i's of the data ACs. This third-order equation is obtained from setting the output rate of a station of AC k equal to its input rate:

$$r_k = \rho_k \tag{13}$$

where r_k is computed by making the following simplification to the expression of (7): we distinguish two types of slots, the ones where only real-time stations can transmit and the ones where data stations can transmit, and then compute the numerator and denominator of (7) by conditioning them to these two types of slots. This yields

$$r_k = \frac{p(\Delta_{real})p(s_k|\Delta_{real}) + p(\Delta_{data})p(s_k|\Delta_{data})}{p(\Delta_{real})T_{slot,r} + p(\Delta_{data})T_{slot,d}} l_k \quad (14)$$

where $p(\Delta_{real})$ is the probability that in a slot time only real-time ACs can transmit, $p(\Delta_{data})$ is the probability that data stations can also transmit, $p(s_k|\Delta_{real})$ and $p(s_k|\Delta_{data})$ are the success probabilities of a station of AC k for each of the two cases, and $T_{slot,r}$ and $T_{slot,d}$ are the average slot durations, respectively.

The probability $p(\Delta_{real})$ is computed as follows. We first calculate the exact expression for the probability of being in a state in which only real-time ACs can transmit, and then we perform a first order approximation of the Taylor expansion of this expression. The result is the following:

$$\begin{aligned} p(\Delta_{real}) \approx & \frac{A_d(1-p(e|\Delta_{data})}{A_d(1-p(e|\Delta_{data}))+1} + \\ & \frac{(1-p(e|\Delta_{data}))A_d^2}{2((1-p(e|\Delta_{data})A_d+1)^2}\tau_k + \\ & \frac{(1+p(e|\Delta_{data}))A_d\sum_{l\in\Delta_{real}} n_l K_l}{2((1-p(e|\Delta_{data})A_d+1)^2}\tau_k \end{aligned} \quad (15)$$

where Δ_{real} is the set of the real-time ACs, and $p(e|\Delta_{data})$ is the probability that a slot in which all ACs can transmit is empty,

$$p(e|\Delta_{data}) = \prod_k (1-\tau_k) \quad (16)$$

$p(\Delta_{data})$ is simply computed as

$$p(\Delta_{data}) = 1 - p(\Delta_{real}) \quad (17)$$

The probability $p(s_k|\Delta_{real})$ corresponds to the probability that a station of AC k transmits and no other real-time station transmits:

$$\begin{aligned} p(s_k|\Delta_{real}) = \tau_k(1-\tau_k)^{n_k-1}\textstyle\prod_{l\in\Delta_{real}\setminus k}(1-\tau_l)^{n_l} \approx \\ \tau_k\left(1-(n_k-1)\tau_k - \textstyle\sum_{l\in\Delta_{real}\setminus k} n_l K_l \tau_k\right) \end{aligned} \quad (18)$$

By considering that the probability that no other station transmits is approximately $p(e|\Delta_{data})$, the probability $p(s_k|\Delta_{data})$ corresponds to the probability that a station of AC k transmits and no other station, real-time or data, transmits:

$$p(s_k|\Delta_{data}) \approx \tau_k p(e|\Delta_{data}) \quad (19)$$

Finally, we can calculate $T_{slot,r}$ and $T_{slot,d}$ as a second-order expression in τ_k, by considering the different lengths of the transmissions that we can have in a slot time and the corresponding probabilities.

Based on the above analysis, our optimal configuration mechanism is described by Algorithm 1 and is summarized as follows:

- Given a reference data AC i and a reference real-time AC k, a search is performed on all A_i's values specified in the standard (line 4). For each A_i value, a *golden section* search is performed on the τ_i to maximize the throughput allocation criterion (line 5).
- For each value of τ_i, the τ_j of the remaining data ACs is computed with (11) according to the allocation criterion (line 7).
- Next, the transmission probability τ_k of the reference real-time AC is computed with (13) (line 9), and, from this, the remaining τ_l's of the other real-time ACs are then computed by applying (12) (line 11).
- With all the τ's, we proceed to compute the CW values that guarantee delay performance to real-time stations. Following the explanations of [21], there is a range of CW values that provide the desired QoS performance. To compute this range of CW values, we use the delay analysis of Lemma 3 and 4 to obtain the configurations that lead to the desired delay performance. With the already-computed τ_i's and the setting $CW_{min} = CW_{max}$, this can be efficiently done using (8) and (9).
- From all the CW values, we choose the maximum one for each AC (line 14) since, following the discussion of [21], these are the ones that lead to a WLAN operating as far as possible from instability.
- We next check that the values of CW obtained in the previous step satisfy the requirement that, even in the cases where the real-time stations become saturated, their throughput in saturation $r_j(sat)$ is larger than their input rate ρ_j (line 18) since, following [43], this guarantees that the proposed configuration is indeed stable. If this condition is not met, then this configuration is not further considered in the search.
- Next, the τ's are used to compute the weighted rate r_i/w_i of each data AC (line 23). Note that the golden section search of line 5 maximizes the minimum of these values. Therefore, if the current configuration provides better performance than the ones evaluated previously in the search, it is saved (lines 25–30).
- Finally, once the search ends, the best configuration is returned through the EDCA parameters (line 37). If the search does not provide any configuration, this means that there exists no configuration that satisfies the sources' requirements, and therefore the request that triggered this search has to be rejected.

C. Optimal Configuration Validation

In this section, we validate our optimal configuration algorithm by means of simulations for different traffic scenarios. More specifically, we assess through simulations the performance of the configuration resulting from our algorithm, and compare it against the best performance obtained by performing an exhaustive search over the EDCA parameters.

1) Data traffic: We first assess the performance of our algorithm for a scenario where only data stations are present. We consider a scenario where 4 ACs with n_i stations each always have a 1500-byte frame ready for transmission. In these circumstances, and with no real-time traffic in the WLAN, the only relevant metric of performance is the maximum $\min(r_i/w_i)$, according to the max-min fair allocation criterion.

Algorithm 1 Optimal configuration of EDCA parameters

Take data AC i as a reference
Take real-time AC k as a reference
$max \leftarrow 0$
for $A_i = 0$ to 15 **do**
  while *Golden section search* on τ_i **do**
    for each data AC j **do**
      $\tau_j \leftarrow w_j \tau_i / (w_i + \tau_i(w_j - w_i))$ ▷ (Eq. 11)
    end for
    Compute τ_k ▷ (Eq. 13)
    for each real-time AC j **do**
      $\tau_l \leftarrow \tau_k K$ ▷ (Eq. 12)
    end for
    for each real-time AC j **do**
      Compute CW_j to fulfill
      the delay requirement ▷ Lemmas 3 and 4
    end for
    for each real-time AC j **do**
      if $r_j(sat) < \rho_j$ **then**
        The τ_i value is not a possible value. Skip. ▷ CW_j corresponds to saturation.
      else
        for each data AC j **do**
          Compute r_j/w_j
        end for
        if $min\{(r_i/w_i)\} > max$ **then** ▷ Save configuration
          $max \leftarrow min\{(r_i/w_i)\}$
          $A_{max} \leftarrow A_i$
          $\tau_{max}^{data} \leftarrow \tau^{data}$
          $CW_{max}^{real-time} \leftarrow CW^{real-time}$
        end if
      end if
    end for
  end while
end for
$CW^{data} \leftarrow 2/\tau_{max}^{data} - 1$
return A_i, CW^{data}, $CW^{real-time}$

Results are shown in Table IV, for $n_i = \{2, 10\}$ stations per AC and different throughput allocation weights w_i's. They show that the configuration algorithm maximizes throughput performance, as the gain obtained when using the exhaustive search is negligible.

2) Voice traffic: Next, we evaluate the performance of our algorithm for voice traffic. We validate the algorithm by comparing the performance of our configuration ($CW_{algorithm}$) against the result of performing an exhaustive search over the CW_{min} space and choosing the best CW_{min} value ($CW_{exhaustive}$). We perform this experiment for three different quality criteria, ranging from a more stringent requirement ($E[d_{max}], \sigma_{max} \leq$ 2.5ms) to a more relaxed one ($E[d_{max}], \sigma_{max} \leq$ 5ms) [21]. Simulation results, presented in Table V, show that i) the proposed configuration is always very close to the one obtained from the exhaustive search and ii) our algorithm admits as many voice calls as the exhaustive search while meeting the desired quality criteria.

TABLE IV
ALGORITHM VALIDATION FOR DATA TRAFFIC SCENARIO

n_i	w_1	w_2	w_3	w_4	min(r_i/w_i)	min(r_i/w_i) exhaustive search
2	1	2	3	4	212.21	212.52
	1	3	5	7	124.20	124.66
	1	4	7	10	88.12	88.41
	1	5	9	13	66.25	66.74
10	1	2	3	4	41.79	41.85
	1	3	5	7	24.81	24.96
	1	4	7	10	18.62	18.71
	1	5	9	13	13.02	13.29

TABLE V
ALGORITHM VALIDATION FOR VOICE TRAFFIC SCENARIO

$E[d_{max}]$	σ_{max}	n	$CW_{algorithm}$	$E[d_{algorithm}]$	$\sigma_{algorithm}$	$CW_{exhaustive}$	$E[d_{exhaustive}]$	$\sigma_{exhaustive}$
5 ms	5 ms	10	314	4.95	2.78	317	4.99	2.82
		15	225	4.91	2.87	229	4.99	2.92
		20	118	4.72	3.02	125	4.99	3.25
5 ms	2.5 ms	10	274	4.35	2.43	281	4.45	2.49
		15	186	4.07	2.36	196	4.28	2.49
		20	89	3.65	2.48	91	4.31	2.49
2.5 ms	2.5 ms	10	145	2.45	1.32	148	2.49	1.35
		15	104	2.32	1.29	111	2.47	1.39
		19	66	2.29	1.42	72	2.49	1.54

3) Voice and Data Traffic: To validate the proposed algorithm for a scenario in which the WLAN operates under both data and voice traffic, we perform the following experiment. We consider two ACs, both with the same number of stations and the following characteristics:

- The first AC transmits 80 byte packets every 10ms. We consider two different delay requirements for this AC: i) $E[d], \sigma_d \leq$ 2.5ms, and ii) $E[d], \sigma_d \leq$ 5ms.
- The second AC models the behavior of a data transfer by always having a 1000 byte packet ready for transmission.

Again, we compare the results from our configuration against those provided by the best configuration found by means of an exhaustive search over the CW_{min} of the data and voice ACs and the A_i parameter of the data AC. The results are shown in Table VI: with the proposed configuration, the quality criteria are always met and the throughput obtained by data stations is very close to the one provided by the configuration resulting from the exhaustive search. We therefore conclude that the proposed configuration algorithm maximizes the performance of the WLAN.

4) Mixed Traffic: Finally, to validate the proposed algorithm under the most generic scenario, we consider a WLAN with the four ACs defined in Section III-D.4, each of them with the same number of stations n_i. For the real-time ACs, we consider the following delay requirements: $E[d_1], \sigma_{d,1} \leq$ 5ms, $E[d_2], \sigma_{d,2} \leq$ 20ms, and for the data ACs the following weights: $w_3 = 2, w_4 = 1$.

The throughput and delay results obtained with the proposed algorithm are given in Table VII. The results validate the

TABLE VI
ALGORITHM VALIDATION FOR DATA AND VOICE TRAFFIC SCENARIO

n_i	$E[d_{max}]$, $\sigma_{d_{max}}$	$E[d]$	σ_d	r_{data}	r_{data} exhaustive search
5	2.5	2.47	2.22	971.62	973.29
10		2.49	2.37	319.04	324.23
15		2.47	2.45	113.85	117.14
5	5	4.85	4.30	974.62	976.83
10		4.83	4.37	324.33	327.14
15		4.69	4.25	113.11	115.44

TABLE VII
ALGORITHM VALIDATION FOR MIXED TRAFFIC SCENARIO

n_i	$E[d_1]$	$\sigma_{d,1}$	$E[d_2]$	$\sigma_{d,2}$	$\min(r_i/w_i)$	$\min(r_i/w_i)$ exhaustive search
2	4.91	3.48	19.81	13.09	793.21	793.93
4	4.96	3.55	19.78	13.00	304.34	304.82
6	5.00	3.59	19.88	13.35	144.17	144.57
8	5.00	3.59	19.54	13.04	64.87	65.15
10	4.74	3.45	18.81	12.73	18.24	18.34

proposed algorithm, since they satisfy all the requirements:

- In all scenarios, the average and standard deviation of the delay obtained with our configuration are below the desired values, which shows that the configuration meets the required delay guarantees.
- Furthermore, an exhaustive search has been conducted which has shown that no other configuration can admit more stations while satisfying the delay requirements. This proves that the configuration maximizes the admissibility region by admitting as many stations as possible.
- Finally, by comparing the throughput performance obtained through exhaustive search against our results, we conclude that we also maximize the $\min(r_i/w_i)$ for data ACs.

D. Comparison against other approaches

We next compare the performance of the configuration resulting from our algorithm against the following approaches:

- Two other available approaches for the configuration of the EDCA parameters, namely, the standard recommended set of values [2] and the recent proposal of [26].
- The adaptive configuration schemes of [27], [31] and [32], which aim to provide QoS guarantees in EDCA WLANs. It is worth noting that the approaches of [31] and [32] require introducing changes to the 802.11e standard, which challenges their practical use.

The scenario that we choose for this comparison is the mixed traffic scenario of Section IV-C.4, since this is the most complete of the scenarios used in the validation of the algorithm. Table VIII gives the average delay of voice and video flows (in milliseconds) as well as the total throughput given to data stations (in Kbps) resulting from our algorithm and from the other five mentioned approaches.

From the results given in the table, we conclude that our algorithm clearly outperforms the other proposals, since $i)$ with our approach, real-time stations always see their delay guarantees satisfied, $ii)$ our approach provides data stations with a substantially larger throughput than any other approach meeting the delay requirements[11], and $iii)$ it also provides a much larger admissibility region; in particular, with our approach we can admit up to $n_i = 10$ stations, while none of the other proposals can admit more than $n_i = 6$ stations (i.e., our approach can admit at least 66% more stations). The reason for this is that the other approaches are based on heuristics that do not guarantee optimal performance, in contrast to ours which is based on an analytical model that guarantees optimal performance.

E. Implementation Considerations

We assess the computational cost of the algorithm by measuring the number of flops (floating point operations) required by a *MATLAB* implementation to execute it. For all the presented experiments, the algorithm requires approximately 90 Kflops. Assuming a WLAN Access Point with a 10 MFlops/sec CPU, it would take 9 ms to perform an admission control decision, which is fully acceptable in a realistic scenario. We conclude from this experiment that the computational complexity of the proposed algorithm is sufficiently low to allow its practical use in today's hardware platforms.

V. CONCLUSIONS

As the EDCA mechanism of 802.11e becomes widely available, the need for a configuration algorithm to tune the MAC parameters and boost WLAN performance arises. We have shown that a proper configuration of EDCA can lead to performance gains of 66% over the standard recommended values. We believe these gains represent a strong motivation for the deployment of EDCA WLANs in order to efficiently use the scarce wireless medium.

In this paper, we have presented an algorithm to configure an EDCA WLAN that achieves a two-fold objective: $i)$ it maximizes the admissibility region of real-time traffic, and $ii)$ it optimizes throughput performance of data traffic. To build this algorithm, we have presented the most comprehensive analysis to date of EDCA performance. This analysis, as proven by exhaustive simulations, can accurately model throughput and delay performance of real-time and non real-time traffic.

We have used this analysis to design an optimal configuration algorithm for EDCA. In contrast to previous work, typically heuristic or measurement-based, ours is a mathematically supported mechanism that tunes the EDCA parameters to maximize performance. By means of the analytical model, we have derived an efficient algorithm whose complexity is well suited for low computation capacity devices and can be implemented in realistic scenarios. We have shown that the performance of our algorithm is almost identical to the one obtained through exhaustive numerical searches.

[11] Although for the $n_i = 10$ case data stations receive a larger throughput with [31] than with our configuration, voice and video stations suffer much larger delays. Additionally, they also suffer a drop rate above 20% (not shown in the table), which results in our approach actually providing a better total throughput performance.

TABLE VIII

COMPARISON AGAINST OTHER APPROACHES

n_i	Our algorithm			Standard			[26]			[31]			[32]			[27]		
	d_{vo}	d_{vi}	r_{data}	d_{vo}	d_{vi}	r_{data}	d_{vo}	d_{vi}	r_{data}	d_{vo}	d_{vi}	r_{data}	d_{vo}	d_{vi}	r_{data}	d_{vo}	d_{vi}	r_{data}
2	4.9	19.8	4759	0.9	2.1	4728	0.4	1.9	3808	5.9	6.6	2127	1.1	2.6	2990	2.1	5.0	3724
4	4.9	19.7	3652	1.5	2.9	3474	0.6	2.5	2576	9.9	11.2	2327	1.9	5.4	2752	4.9	12.2	1713
6	5.0	19.8	2595	2.5	5.1	2093	0.8	4.9	1478	9.9	19.5	1868	4.0	10.0	1616	18.7	41.7	104
8	5.0	19.5	1556	14.5	28.9	9	1.3	32.4	1	10.0	26.6	894	11.6	32.2	347	33.4	76.4	65
10	4.7	18.8	547	18.6	36.6	4	1.4	46.9	0	15.9	27.0	831	14.9	42.3	278	50.6	48.5	146

REFERENCES

[1] IEEE 802.11 WG, *Amendment to Standard for Information Technology. LAN/MAN Specific Requirements - 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] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1–1184, 12 2007.

[3] S. Mangold, S. Choi, G. Hiertz, O. Klein, and B. Walke, "Analysis of IEEE 802.11e for QoS Support in Wireless LANs," *IEEE Wireless Communications*, vol. 10, no. 6, pp. 40–50, Diciembre 2003.

[4] S. Choi, J. Prado, and S. Shankar, "IEEE 802.11e Contention-Based Channel Access (EDCF) Performance Evaluation," in *Proceedings of IEEE ICC'03*, 2003.

[5] A. Grilo and M. Nunes, "Performance evaluation of IEEE 802.11e," in *Proceedings of PIMRC'02*, Lisboa, Portugal, September 2002.

[6] D. Gu and J. Zhang, "A New Measurement-Based Admission Control Method for IEEE802.11 Wireless Local Area Networks," in *Proceedings of IEEE PIMRC '03*, Septiembre 2003, pp. 2009—2013.

[7] A. Banchs and L. Vollero, "Throughput Analysis and Optimal Configuration of 802.11e EDCA," *Computer Networks*, vol. 50, no. 11, August 2006.

[8] J. W. Robinson and T. S. Randhawa, "Saturation Throughput Analysis of IEEE 802.11e Enhanced Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 5, pp. 917–928, June 2004.

[9] Z. Kong, D. H. K. Tsang, B. Bensaou, and D. Gao, "Performance Analysis of IEEE 802.11e Contention-Based Channel Access," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 10, pp. 2095–2106, December 2004.

[10] V. Ramaiyan, A. Kumar, and E. Altman, "Fixed point analysis of single cell IEEE 802.11e WLANs: uniqueness, multistability and throughput differentiation," in *Proceedings of ACM SIGMETRICS 2005*, June 2005.

[11] J. Hui and M. Devetsikiotis, "A unified model for the performance analysis of IEEE 802.11e EDCA," *IEEE Transactions on Communications*, vol. 53, no. 9, pp. 1498–1510, September 2005.

[12] H. Zhu and I. Chlamtac, "Performance Analysis for IEEE 802.11e EDCF Service Differentiation," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, June 2005.

[13] A. Banchs and L. Vollero, "A Delay Model for IEEE 802.11e EDCA," *IEEE Communications Letters*, vol. 9, no. 6, pp. 508–510, Junio 2005.

[14] H. Y. Hwang, S. J. Kim, D. K. Sung, and N.-O. Song, "Performance analysis of ieee 802.11e edca with a virtual collision handler," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 2, pp. 1293–1297, March 2008.

[15] P. E. Engelstad and O. N. Osterbo, "Non-saturation and saturation analysis of IEEE 802.11e EDCA with starvation prediction," in *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWIM'05)*, October 2005.

[16] G.-R. Cantieni, Q. Ni, C. Barakat, and T. Turletti, "Performance Analysis of Finite Load Sources in 802.11b Multirate Environments," *Computer Communications*, vol. 28, no. 10, pp. 1095–1109, June 2005.

[17] J. W. Tantra, C. H. Foh, I. Tinnirello, and G. Bianchi, "Analysis of the IEEE 802.11e EDCA under Statistical Traffic," in *Proceedings of ICC'06*, Istanbul, Turkey, June 2006.

[18] K. Duffy, D. Malone, and D. Leith, "Modeling the 802.11 Distributed Coordination Function in non-saturated conditions," *IEEE Communications Letters*, vol. 9, no. 8, August 2005.

[19] B. Li and R. Battiti, "Analysis of the 802.11 DCF with Service Differentiation Support in Non-saturation Conditions," in *Proceedings of the Fifth International Workshop on Quality of future Internet Services (QofIS'05)*, September 2004.

[20] P. Serrano, A. Banchs, and A. Azcorra, "A throughput and delay model for IEEE 802.11e EDCA under non saturation," *Wirel. Pers. Commun.*, vol. 43, no. 2, pp. 467–479, 2007.

[21] P. Serrano, A. Banchs, and J. Kukielka, "Optimal configuration of 802.11e edca under voice traffic," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, 2007, pp. 5107–5111.

[22] Y. Ge, J. C. Hou, and S. Choi, "An analytic study of tuning systems parameters in IEEE 802.11e enhanced distributed channel access," *Comput. Netw.*, vol. 51, no. 8, pp. 1955–1980, 2007.

[23] D. Gao, J. Cai, C. H. Foh, C.-T. Lau, and K. N. Ngan, "Improving wlan voip capacity through service differentiation," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 1, pp. 465–474, Jan. 2008.

[24] M. Narbutt and M. Davis, "Experimental tuning of AIFSN and CWmin parameters to prioritize voice over data transmission in 802.11e WLAN networks," in *IWCMC '07: Proceedings of the 2007 international conference on Wireless communications and mobile computing*. New York, NY, USA: ACM, 2007, pp. 140–145.

[25] C. Cano, B. Bellalta, and M. Oliver, "Adaptive admission control mechanism for ieee 802.11e wlans," *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pp. 1–5, Sept. 2007.

[26] J. Lee, W. Liao, J.-M. Chen, and H.-H. Lee, "A practical QoS solution to voice over IP in IEEE 802.11 WLANs ," *Communications Magazine, IEEE*, vol. 47, no. 4, pp. 111–117, April 2009.

[27] J. Freitag, N. L. S. da Fonseca, and J. F. de Rezende, "Tuning of 802.11e Network Parameters," *IEEE Communications Letters*, vol. 10, no. 8, pp. 611–613, August 2006.

[28] W.-T. Chen, "An effective medium contention method to improve the performance of ieee 802.11," *Wirel. Netw.*, vol. 14, no. 6, pp. 769–776, 2008.

[29] K. A. Meerja and A. Shami, "Analysis of enhanced collision avoidance scheme proposed for ieee 802.11e-enhanced distributed channel access protocol," *IEEE Transactions on Mobile Computing*, vol. 8, no. 10, pp. 1353–1367, 2009.

[30] J.-C. Chen and K.-W. Cheng, "Edca/ca: Enhancement of ieee 802.11e edca by contention adaption for energy efficiency," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 8, pp. 2866–2870, August 2008.

[31] A. Nafaa and A. Ksentini, "On sustained qos guarantees in operated ieee 802.11 wireless lans," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 8, pp. 1020–1033, Aug. 2008.

[32] Y. Xiao, F. H. Li, and B. Li, "Bandwidth Sharing Schemes for Multimedia Traffic in the IEEE 802.11e Contention-Based WLANs," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 7, pp. 815–831, July 2007.

[33] F. Calì, M. Conti, and E. Gregori, "Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Netw.*, vol. 8, no. 6, pp. 785–799, 2000.

[34] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[35] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of reliable transport protocol over IEEE 802.11 wireles LAN: Analysis and enhancement," in *Proceedings of IEEE INFOCOM'02*, June 2002.

[36] G. Grimmett and D. Stirzaker, *Probability and Random Processes, 3rd ed.* Oxford Press, 2001.

[37] IEEE 802.11 WG, *Information Technology - Telecommun. and Information Exchange between Systems. Local and Metropolitan Area Networks. Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical*

Layer Extension in the 2.4 GHz Band. Supplement to IEEE 802.11 Standard, September 1999.

[38] K. Claffy, G. Miller, and K. Thompson, "The nature of the beast: Recent traffic measurements from an Internet backbone," in *Proceedings of INET'98*, Geneve, Switzerland, July 1998.

[39] L. Muscariello, M. Mellia, M. Meo, R. L. Cigno, and M. Ajmone, "A Simple Markovian Approach to Model Internet Traffic at Edge Routers," COST279, Technical Document, No. TD(03)032, May 2003.

[40] D. Bertsekas and R.G. Gallager, *Data Networks*. Prentice Hall, 1987.

[41] L. Massoulié and J. Roberts, "Bandwidth sharing: objectives and algorithms," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 320–328, 2002.

[42] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *Mobile Computing and Networking*, 2000, pp. 167–178.

[43] P. Serrano, A. Banchs, T. Melia, and L. Vollero, "Performance Anomalies of Nonoptimally Configured Wireless LANs," in *IEEE Wireless Communications and Networking Conference (WCNC), Las Vegas, Nevada*, April 2006.

APPENDIX I

Lemma 1: The τ_i of a non-saturated AC is given by

$$\tau_i^{nonsat} = \frac{\rho_i(1-p(c_i)^{R+1})\left(p(s)T_s + p(e)T_e + p(c)T_c\right)}{l_i(1-\tau_i)^{n_i-1}\sum_{k=A_i}^{A} p(\Delta_k)\prod_{j\in\Delta_k\setminus i}(1-\tau_j)^{n_j}} \tag{20}$$

Proof: According to Section III-A, a station of a non-saturated AC sees all the traffic it sends served, either because its packets are transmitted successfully or because they are discarded when reaching the retry limit due to suffering $R+1$ collisions. Hence, the following equation holds,

$$\rho_i(1-p(c_i)^{R+1}) = r_i \tag{21}$$

where r_i is the throughput experienced by a station of $AC\ i$, given by (7), ρ_i is its average sending rate, and $p(c_i)^{R+1}$ corresponds to the probability that a packet of this station is discarded upon reaching the retry limit.

In order to prove the lemma, we need to derive the different variables in (7). The probability $p(e)$ is, by definition, $p(e|t_0)$, as all slot times are *0-slot time*s. This has already been computed in Section III-B. To compute the rest of the variables in (7), we proceed as follows. First, let us define $p(t_k)$ as the probability that a slot time is a k*-slot time*. Since a slot time is a k*-slot time* if and only if the previous slot time is a $(k-1)$*-slot time* and it is empty, which occurs with a probability $p(e|t_{k-1})$, this probability can be expressed as

$$p(t_k) = p(t_{k-1})p(e|t_{k-1}) \tag{22}$$

Starting from $p(t_0) = 1$ (which holds by definition), and applying the above recursively, it follows that

$$p(t_k) = \prod_{j=0}^{k-1} p(e|t_j) \tag{23}$$

The probability that a random slot time contains a success of a given station of $AC\ i$ can be computed (by applying the total probability theorem) as

$$p(s_i) = \sum_{k=A_i}^{A} p(\Delta_k)p(s_i|\Delta_k), \tag{24}$$

where $p(s_i|\Delta_k)$ is the probability that a slot time in which this set of ACs may transmit contains a success of a given station of $AC\ i$.

A slot time is allowed for transmission to the set Δ_k (with $k < A$) if the slot time is a k*-slot time* but not a $(k+1)$*-slot time*[12]. For $k = A$, we have that in a A*-slot time* all ACs are allowed to transmit. Thus,

$$p(\Delta_k) = \begin{cases} p(t_k) - p(t_{k+1}) & k < A, \\ p(t_A) & k = A. \end{cases} \tag{25}$$

The probability $p(s_i|\Delta_k)$ corresponds to the case when the considered station transmits and no other station of set Δ_k does:

$$p(s_i|\Delta_k) = \tau_i(1-\tau_i)^{n_i-1}\prod_{j\in\Delta_k\setminus i}(1-\tau_j)^{n_j} \tag{26}$$

The probability that a slot time contains a success can be computed as the sum of the individual success probabilities

$$p(s) = \sum_{i\in\Delta_A} n_i p(s_i) \tag{27}$$

where, with our definition of A, Δ_A denotes the set of all ACs. The probability that a slot time contains a collision can be obtained from

$$p(c) = 1 - p(e) - p(s) \tag{28}$$

The average duration of a success T_s can be computed by summing the different possible durations weighted by their probabilities

$$T_s = \sum_{i\in\Delta_A} \frac{n_i p(s_i)}{p(s)} T_{s,i} \tag{29}$$

where $T_{s,i}$ is the average duration of a success of a station of AC i, which is calculated according to the following expression given by [7]

$$T_{s,i} = T_{PLCP} + \frac{H+l_i}{C} + SIFS + T_{PLCP} + \frac{ACK}{C} + DIFS \tag{30}$$

where T_{PLCP} is the Physical Layer Convergence Protocol preamble and header transmission time, H is the MAC overhead (header and FCS), ACK is the size of the acknowledgment frame, and C is the channel bit rate.

In order to compute the average duration of a collision T_c, we note that this is given by the largest packet length involved. Following this, we can compute T_c by summing the possible collision durations weighted by their probabilities,

$$T_c = \sum_{l\in L} \frac{p(c, t=l)}{p(c)} T_c^l \tag{31}$$

where $p(c, t = l)$ is the probability that a slot time contains a collision in which the length of the longest packet involved is equal to l, T_c^l is the duration of this collision, and L is the set of packet lengths.

T_c^l is computed as (see [7])

$$T_c^l = T_{PLCP} + \frac{H+l}{C} + EIFS \tag{32}$$

[12] Note that a slot time that is a k*-slot time* but not a (k+1)*-slot time* is preceded by exactly k empty slot times, and therefore only the ACs with $A_i \leq k$ (i.e., the ACs of set Δ_k) may transmit in such a slot time.

and $p(c,t=l)$ is computed, applying the total probability theorem, as

$$p(c,t=l)=\sum_{k=0}^{A}p(\Delta_k)p(c,t=l|\Delta_k) \tag{33}$$

where $p(c,t=l|\Delta_k)$ is the probability that, given that only stations of set Δ_k may transmit, a slot time contains a collision in which the longest packet involved is of length l.

To obtain $p(c,t=l|\Delta_k)$ we sweep along all the stations that may transmit and compute the probability that i) the considered station transmits a packet of length l and ii) some other station transmits a packet, but with length no longer than l. Let us define S_k as the set of stations of Δ_k and $p(t_j=l)$ as the probability that the length of a transmission from station j is l. Then,

$$p(c,t=l|\Delta_k)=\sum_{j\in S_k}\tau_j\, p(t_j=l)p(tx\leq l|S_k,j) \tag{34}$$

where $p(tx\leq l|S_k,j)$ accounts for the probability that there is at least one transmission from the set S_k (without station j), but of size less than or equal to l. To compute this probability, we calculate the probability that no station transmits a packet longer than l and subtract from this the probability that no station transmits. In particular, for the computation of the first term, we index all the stations and refer with $S_{k,j}$ to the set of stations of S_k with index smaller than j; then, we compute the probability that stations of $S_{k,j}$ do not transmit a packet longer than or equal to l, and the probability that stations with higher index than j do not transmit a packet longer than l[13],

$$p(tx\leq l|S_k,j)=\prod_{m\in S_{k,j}}\left(1-\tau_m p(t_m\geq l)\right) \prod_{m\in S_k\setminus S_{k,j}\cup j}\left(1-\tau_m p(t_m>l)\right)-\prod_{m\in S_k\setminus j}(1-\tau_m) \tag{35}$$

Finally, expressing r_i as a function of the variables computed in (22)–(35) and substituting these into (20) yields

$$\rho_i(1-p(c_i)^{R+1})=\tau_i(1-\tau_i)^{n_i-1}l_i \frac{\sum_{k=A_i}^{A}p(\Delta_k)\prod_{j\in\Delta_k\setminus i}(1-\tau_j)^{n_j}}{(p(s)T_s+p(e)T_e+p(c)T_c)} \tag{36}$$

The proof follows. ∎

Lemma 2: The average throughput r_i a station from $AC\ i$ experiences is given by

$$r_i=\frac{l_i\sum_{k=A_i}^{A}p(\Delta_k)p(s_i|\Delta_k)}{p(s)T_s+p(c)T_c+p(e)T_e} \tag{37}$$

Proof: We compute the throughput r_i following (12) of [34]: we divide the average payload information transmitted by $AC\ i$ in a slot time $E[payload_i\ per\ slot]$ over the average duration of a slot time $E[slot\ length]$.

$$r_i=\frac{E[payload_i\ per\ slot]}{E[slot\ length]} \tag{38}$$

[13]The distinction in (35) between the stations with indexes smaller and larger than j is made in order to avoid counting more than once the event when two or more stations transmit a packet of length l.

The average payload information transmitted by $AC\ i$ is given by

$$E[payload_i\ per\ slot]=l_i p(s_i) \tag{39}$$

while the average length of a slot time is given by

$$E[slot\ length]=p(s)T_s+p(c)T_c+p(e)T_e \tag{40}$$

where the probabilities and average durations have already been derived in the proof of Lemma 1 above. By combining the above equations we obtain

$$r_i=\frac{l_i p(s_i)}{p(s)T_s+p(c)T_c+p(e)T_e} \tag{41}$$

The proof follows. ∎

Lemma 3: The *average delay* experienced by a non dropped packet of a station of $AC\ i$ is given by

$$d_i=\frac{1}{\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j}\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j(jT_{c,i}+ T_{s,i}+\sum_{r=0}^{j}(T^i_{inter\_tx,k}+B_{i,r}(T^i_{slot,k}+T^i_{inter,k}))) \tag{42}$$

Proof: To compute the average delay of a non-dropped packet d_i we use the total probability theorem as follows

$$d_i=\frac{\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j d_{i,j}}{\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j} \tag{43}$$

where $d_{i,j}$ is defined as the average delay of a station of $AC\ i$ in case the frame suffers j retries. This delay is computed as (see Fig. 3)

$$d_{i,j}=\sum_{r=0}^{j}\left(T^i_{inter\_tx,k}+B_{i,r}(T^i_{slot,k}+T^i_{inter,k})\right)+jT_{c,i}+T_{s,i} \tag{44}$$

In order to complete the proof of the Lemma, we need to compute the components of (44). $T_{s,i}$ is given by (30). $B_{i,r}$ is computed using the following equation given in [13]

$$B_{i,r}=\frac{CW_i^{min}2^{min(m_i,r)}-1}{2} \tag{45}$$

$T_{c,i}$ is computed by applying the total probability theorem

$$T_{c,i}=\frac{\sum_{k=A_i}^{A}T_{c,i,k}p(\Delta_k)}{\sum_{k=A_i}^{A}p(\Delta_k)} \tag{46}$$

where $T_{c,i,k}$ is the average duration of a collision in which a station of AC i is involved when only the ACs of set Δ_k may transmit. This is computed as follows.

$$T_{c,i,k}=\frac{\sum_{l\in L}T_c^l p(c_i,t=l|S_k)}{\sum_{l\in L}p(c_i,t=l|S_k)} \tag{47}$$

where $p(c_i,t=l|S_k)$ is the probability that a slot time in which a station of AC i transmits and the stations of set S_k may transmit contains a collision of length l. This is computed by distinguishing between the case that a station of AC i

transmits a frame of size l (with probability $p(t_i = l)$) or smaller (with probability $p(t_i < l)$):

$$p(c_i, t = l|S_k) = p(t_i = l)\cdot \left(\prod_{m\in S_k\setminus i}(1-\tau_m p(t_m > l)) - \prod_{m\in S_k\setminus i}(1-\tau_m)\right) + p(t_i < l)\left(\sum_{j\in S_k\setminus i}\tau_j p(t_j = l)\prod_{m\in S_{k,j}\setminus i}(1-\tau_m p(t_m \geq l)) \prod_{m\in S_k\setminus\{S_{k,j}\cup i,j\}}(1-\tau_m p(t_m > l))\right) \quad (48)$$

$T^i_{slot,k}$ is computed as the sum of probabilities of success, empty, and collision multiplied by the average slot time duration in each case,

$$T^i_{slot,k} = p(s|S_k,i)T^i_{s,k} + p(e|S_k,i)T_e + (1 - p(s|S_k,i) - p(e|S_k,i))\, T^i_{c,k} \quad (49)$$

where $p(e|S_k,i)$ and $p(s|S_k,i)$ are the probabilities that a *k-slot time* in which the considered station does not transmit[14] is empty and contains a success, respectively, and $T^i_{s,k}$ and $T^i_{c,k}$ are the average slot time durations of a success and a collision.

$T^i_{s,k}$ is computed by applying the total probability theorem

$$T^i_{s,k} = \frac{\sum_{j=k}^{A} p(\Delta_j)\sum_{m\in\Delta_j} n_{m,i} p(s_m|\Delta_j,i)T_{s,m}}{p(s|S_k,i)} \quad (50)$$

where $n_{m,i} = n_m - \delta_{im}$ (the Kronecker function δ_{im} accounts for the fact that the considered station does not transmit), $p(s_m|\Delta_j,i)$ is the probability that, given the set Δ_j can transmit but station i did not transmit, there is a success from $AC\ m$,

$$p(s_m|\Delta_j,i) = \tau_m(1-\tau_m)^{n_{m,i}-1}\prod_{j\in\Delta_k\setminus m}(1-\tau_j)^{n_{j,i}} \quad (51)$$

and $p(s|S_k,i)$ is computed by adding the success probabilities of each AC,

$$p(s|S_k,i) = \sum_{j=k}^{A} p(\Delta_j)\sum_{m\in\Delta_j} n_{m,i} p(s_m|\Delta_j,i)T_{s,m} \quad (52)$$

$T^i_{c,k}$ is computed similarly to (50)

$$T^i_{c,k} = \frac{\sum_{j=k}^{A}\sum_{l\in L} T^l_c p(c,t=l|S_j,i)p(\Delta_j)}{\sum_{j=k}^{A}\sum_{l\in L} p(c,t=l|S_j,i)p(\Delta_j)} \quad (53)$$

where

$$p(c,t=l|S_k,i) = \sum_{j\in S_k\setminus i}\tau_j\, p(t_j = l)p(tx \leq l|S_k,i,j) \quad (54)$$

is the probability of a collision of size l in a k-slot, with $p(tx \leq l|S_k,i,j)$ being the probability that at least one station other than i and j transmits a frame of smaller than or equal to l, computed following (35). Finally, we compute $p(e|S_k,i)$ by applying a similar reasoning to (3),

$$p(e|S_k,i) = \frac{p(e|t_k)}{1-\tau_i} \quad (55)$$

$T^i_{inter,k}$ is computed as follows. If the given slot time is empty, which occurs with probability $p(e|S_k,i)$, then $T^i_{inter,k} = 0$. Otherwise, $T^i_{inter,k}$ is by definition equal to $T^i_{inter\_tx,k}$. Thus,

$$T^i_{inter,k} = (1 - p(e|S_k,i))\, T^i_{inter\_tx,k} \quad (56)$$

The above relies on $T^i_{inter\_tx,k}$, which is the time between a nonempty timeslot and the next k-slot. To compute it, we consider the number of j empty slots that follow the transmission(s) and distinguish two cases: i) when the number of j empty timeslots is equal to k, and therefore the time until the next k-slot is composed of exactly k empty slot times, and ii) when $j < k$, and therefore the time is composed by j empty slot times, a non-empty slot where only stations from S_j can transmit, and an additional time which is, by definition, $T^i_{inter\_tx,k}$. This way,

$$T^i_{inter\_tx,k} = \prod_{j=0}^{k} p(e|S_j,i)kT_e + \sum_{j=0}^{k-1}\left(\prod_{l=0}^{j} p(e|S_l,i)(1-p(e|S_{j+1},i))\right)\cdot \left(jT_e + T^i_{slot\_tx,j} + T^i_{inter\_tx,k}\right) \quad (57)$$

where $T^i_{slot\_tx,j}$ is the average duration of a nonempty slot time preceded by a nonempty *k-slot time* followed by j empty slot times, computed as the probability that such a slot time contains a collision multiplied by the average duration in this case, plus the probability that it contains a success multiplied by the corresponding average duration,

$$T^i_{slot,tx,j} = \left(1 - \frac{\sum_{m\in S_j} n_m\tau_m(1-\tau_m)^{n_m-1}\prod_{p\in S_j\setminus m}(1-\tau_p)^{n_p}}{1-\prod_{m\in S_j}(1-\tau_m)^{n_m}}\right)\cdot T^i_{c,j} + \frac{\sum_{m\in S_j} n_m\tau_m(1-\tau_m)^{n_m-1}\prod_{p\in S_j\setminus m}(1-\tau_p)^{n_p}}{1-\prod_{m\in S_j}(1-\tau_m)^{n_m}} T^i_{s,j} \quad (58)$$

Equations (57)–(58) can be reduced to a first order equation on $T^i_{inter\_tx,k}$, from which we can isolate this term and then derive $T^i_{inter,k}$. By combining all the above equations, we obtain the expression for the average delay given by the lemma, as well as the computation of all the terms of this expression. The proof follows. ■

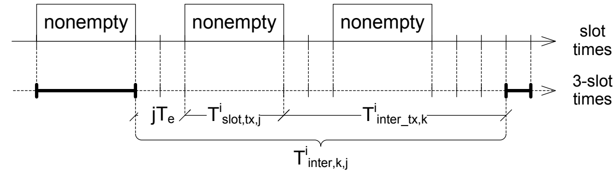


Fig. 15. Components of $T^i_{inter,k,j}$ (example with $k = 3$ and $j = 2$).

[14]The condition that the considered station does not transmit holds until the end of the proof.

Lemma 4: The *standard deviation* of the delay is given by

$$\sigma_{d_i}^2 = \frac{\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j E[(d_{i,j})^2]}{\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j} - (d_i)^2 \qquad (59)$$

Proof: To compute the standard deviation of the delay, $\sigma_{d_i}^2$, we use the following statistical relationship between the average and the second order moment

$$\sigma_{d_i}^2 = E[(d_i)^2] - (d_i)^2 \qquad (60)$$

We already have computed d_i in (8), so the remaining challenge is to compute the second order of the average delay, i.e., $E[(d_i)^2]$. To this aim, we proceed similarly to (43)

$$E[(d_i)^2] = \frac{\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j E[(d_{i,j})^2]}{\sum_{j=0}^{R}(1-p(c_i))p(c_i)^j} \qquad (61)$$

In order to compute $E[(d_{i,j})^2]$, we rewrite $d_{i,j}$ in (44) as

$$d_{i,j} = T_{s,i} + jT_{c,i} + jT^i_{inter\_tx,k} + d^{i,j}_{bo} \qquad (62)$$

where $d^{i,j}_{bo}$ is the average time spent in backoff counter decrements for $AC\ i$ in case of j retries,

$$d^{i,j}_{bo} = \sum_n p(bo = n|AC\ i, j\ retx) \cdot \underbrace{(T^i_{slot,k} + T^i_{inter,k}) + ... + (T^i_{slot,k} + T^i_{inter,k})}_{n \text{ times}} \qquad (63)$$

where $p(bo = n|AC\ i, j\ retx)$ is the probability that the total number of backoff counter decrements after j retries is n. This is computed through j convolutions of the different uniform distributions the station may use to compute its backoff counter:

$$p(bo = n|AC\ i, j\ retx) = U(0, CW_i^{min} - 1) * \cdots * U(0, 2^{min(j,m_i)}CW_i^{min} - 1) \qquad (64)$$

With the above, we proceed as follows to compute $E[(d_{i,j})^2]$

$$E[(d_{i,j})^2] = (d_{i,j})^2 + \sigma^2_{d_{i,j}} \qquad (65)$$

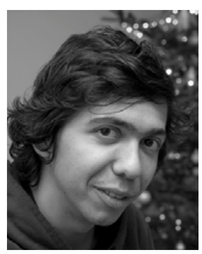
where $\sigma^2_{d_{i,j}}$ is given by the sum of the variances of the components of (62). With our assumption that slot time durations are independent:

$$\sigma^2_{d_{i,j}} = j\sigma^2_{T_{c,i}} + \sigma^2_{T_{s,i}} + (j+1)\sigma^2_{T^i_{inter\_tx,k}} + \sigma^2_{d^{i,j}_{bo}} \qquad (66)$$

Given the previous expressions, the computation of $E[(d_i)^2]$ (and therefore the analysis of the standard deviation of the delay) is laborious but straightforward, as it basically involves redoing the analysis of the average delay but computing second order moments and variances. By combining all the above equations, we obtain the expression for the average delay given by the lemma, as well as the computation of all the terms of this expression. The proof follows. ∎

Pablo Serrano got his Telecommunication Engineering degree and his PhD from the University Carlos III of Madrid (UC3M) in 2002 and 2006, respectively. He has been with the Telematics Department of UC3M since 2002, where he currently holds the position of Assistant Professor. In 2007 he was a Visiting Researcher at the Computer Network Research Group at Univ. of Massachusetts Amherst partially suported by the Spanish Ministry of Education under a José Castillejo grant. His current work focuses on performance evaluation of wireless networks. He has over 20 scientific papers in peer-reviewed international journal and conferences. He also serves as TPC member of several international conferences, including IEEE Globecom and IEEE INFOCOM.

Albert Banchs received his Telecommunications Engineering degree from the Polytechnical University of Catalonia in 1997, and the PhD degree from the same university in 2002. His Ph.D. received the national award for best thesis on broadband networks. He was a visitor researcher at ICSI, Berkeley, in 1997, worked for Telefonica I+D, in 1998, and for NEC Europe Ltd., Germany, from 1998 to 2003. Since 2003, he is with the University Carlos III of Madrid and since 2009 he is also Deputy Director of IMDEA Networks. A. Banchs authors over 50 publications in peer-reviewed journals and conferences and four patents (two of them granted). He is associated editor for IEEE Communications Letters and has been guest editor for IEEE Wireless Communications and Computer Networks. He has served on the TPC of a number of conferences and workshops including IEEE INFOCOM, IEEE ICC and IEEE Globecom, and is TPC chair for European Wireless 2010.



Paul Patras received his Telecommunications Engineering degree from the Technical University of Cluj-Napoca in 2006, and his M.Sc. degree in Telematics Engineering from University Carlos III of Madrid in 2008. Since 2007, he is a research assistant at IMDEA Networks and a Ph.D. candidate at University Carlos III of Madrid. His current research interests include performance optimization in IEEE 802.11 wireless LANs, adaptive MAC mechanisms, quality of service provisioning in wireless mesh networks, prototype implementation and testbeds.

Arturo Azcorra received his M.Sc. degree in Telecommunications Engineering from the Universidad Politécnica de Madrid in 1986 and his Ph.D. from the same university in 1989. In 1993, he obtained an M.B.A. from the Instituto de Empresa. Dr. Azcorra has recently been appointed Director General for Technology Transfer and Entrepreneurial Development at the Spanish Ministry of Science and Innovation. As a result, he is on leave from his double appointment as Full Professor (with chair) at the Telematics Engineering Department of University Carlos III of Madrid and as Director of IMDEA Networks. He founded the not-for-profit institute in 2006 and has conducted his research activities there since its inception. Arturo Azcorra is an IEEE Senior Member and an ACM SIGCOMM Member. He has participated in and directed 49 research and technological development projects, including European ESPRIT, RACE, ACTS and IST programs. Prof. Azcorra has coordinated the CONTENT and E-NEXT European Networks of Excellence. He has served as a Program Committee Member in numerous international conferences, including several editions of IEEE PROMS, IDMS, QofIS, CoNEXT and IEEE INFOCOM. He has published over 100 scientific papers in books, international magazines and conferences.