In: Progress in Wireless Communications Research ISBN 978-160021-675-6
Editor: Alfred P. Martinhoff, pp. 121-149

Chapter 8

APPLICATIONS AND CONFIGURATION OF 802.11E WIRELESS LANS

Albert Banchs and Pablo Serrano
University Carlos III of Madrid

Abstract

Nowadays Wireless LANs (WLANs) have become a very popular technology for Internet access. The Medium Access Control algorithm used by today's WLANs is the one defined by the IEEE 802.11 standard. Recently, the IEEE 802 Working Group has approved a new standard called 802.11e that extends the basic 802.11 algorithm with Quality of Service capabilities. This new standard is based on a number of open parameters whose configuration is yet an unresolved research issue. In this article we take up this subject by reviewing existing guidelines for the configuration of the 802.11e parameters as well as proposing new ones.

1. Introduction

In recent years, much interest has been devoted to the design of wireless local area networks (WLAN's) with Quality of Service (QoS) support. The Enhancements Task Group (TGe) was formed under the IEEE 802.11 project to recommend an international WLAN standard with QoS support. Recently, this group has approved a new standard called 802.11e [1] that extends the basic 802.11 [2] algorithm with Quality of Service capabilities.

The 802.11e standard defines two different access mechanisms: the *Enhanced Distributed Channel Access* (EDCA) and the *HCF Controlled Channel Access* (HCCA). This article focuses on the former. As EDCA is based on several open configurable parameters (namely CW_{min}, CW_{max}, $AIFS$ and $TXOP\_limit$), the challenge with this mechanism lies in its configuration. While there are some configuration recommendations for EDCA in the standard, these are not sustained analytically and do not guarantee optimized performance.

In this article we take up the issue of the configuration of the open parameters in the standard by reviewing existing configuration guidelines as well as proposing new ones. The various proposals addressed in the article respond to scenario driven strategies derived from 802.11e usage. In particular, in the article we consider the following situations:

- *Service Guarantees in a friendly environment.* In this scenario the objective is to compute the 802.11e configuration parameters that provide clients with the requested service guarantees. It is assumed that all users behave friendly, and therefore configurations do not have to prevent that users obtain a better service than requested by misbehaving. A typical example of this scenario could be an office setting.

- *Fair Resource Allocation.* In contrast to the above scenario, in this case the objective is not to provide service guarantees but to fairly distribute the network resources among the various WLAN users. This strategy could be used e.g. in a WLAN in which the applications' requirements are not known.

- *Service Guarantees in an unfriendly environment.* In this scenario the objective is to provide service guarantees, like in the first scenario. However, in this case the configurations have to prevent that potential misbehaving users can obtain a better service thereby disrupting other clients. This requires a different strategy from the above for the configuration of 802.11e parameters. An example of such a scenario could be a WLAN hot-spot.

The article discusses and proposes parameter configuration guidelines for the above scenarios and validates them via simulation. The rest of this article is devoted to the analysis of EDCA configurations in order to satisfy the requirements of each of the above scenarios. It is structured as follows. In section 2. we provide a short summary of the IEEE 802.11e EDCA protocol. Section 3. presents some configuration rules to provide *service guarantees in a friendly environment*. In particular, the focus of that section is on voice over IP applications in this environment. Next, Section 4. proposes some configuration guidelines for the second of the target scenarios, *fair resource allocation*. Finally, Section 5. addresses the scenario of *service guarantees in an unfriendly environment* and proposes configuration rules for this scenario. The article closes with some final remarks in Section 6..

2. IEEE 802.11e EDCA

This section briefly summarizes the EDCA mechanism as defined in the 802.11e standard [1]. EDCA regulates the access to the wireless channel on the basis of the *channel access functions* (CAF's). A station may run up to 4 CAF's, and each of the frames generated by the station is mapped to one of these CAF's. Then, each CAF executes an independent backoff process to transmit its frames.

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

Upon starting the backoff process, the CAF computes a random value uniformly distributed in the range $(0, CW - 1)$, and initializes its backoff time counter with this value. The CW value is called contention window, and depends on the number of transmissions

failed for the frame. At the first transmission attempt, CW is set equal to a value CW_{min}, called minimum contention window.

The backoff time counter is decremented once every time interval σ as long as the channel is sensed idle, "frozen" when a transmission is detected on the channel, and reactivated when the channel is sensed idle again for an $AIFS$. As soon as the backoff time counter reaches zero, the CAF transmits its frame. A collision occurs when two or more CAF's start transmission simultaneously. An acknowledgement (Ack) frame is used to notify the transmitting CAF that the frame has been successfully received. The Ack is immediately transmitted at the end of the frame, after a period of time called short interframe space (SIFS).

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

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

The RTS/CTS mechanism is defined as optional for EDCA. With this mechanism, a CAF that has a frame to transmit follows the same backoff procedure as described above, and then, instead of the frame, preliminarily transmits a special short frame called Request To Send (RTS). When the receiving station detects an RTS frame, it responds, after a SIFS, with a Clear To Send (CTS) frame. The CAF is allowed to transmit its frame only if the CTS frame is correctly received; in this case, the frame transmission proceeds after a SIFS, and it is followed by an Ack.

In the case of a single station running more than one CAF, if the backoff time counters of two or more CAF's of the station reach zero at the same time, a scheduler inside the station avoids the *internal collision*, granting the access to the channel to the highest priority CAF. The other CAF's of the station involved in the internal collision react as if there had been a collision on the channel, doubling their CW and restarting the backoff process.

As it can be seen from the description of EDCA given in this section, the behavior of a CAF depends on a number of parameters, namely CW_{min}, CW_{max}, $AIFS$ and $TXOP\_limit$. These are configurable parameters that can be set to different values for different CAF's. The standard draft groups CAF's by Access Categories (AC's), having all the CAF's of an AC the same configuration, and limits the maximum number of AC's in the WLAN to 4.

The rest of the article is devoted to the analysis of the performance of a WLAN as a function of the above EDCA parameters and to the search for their appropriate configuration under the three identified scenarios. For simplicity, in the rest of this article we assume that station only execute one CAF and use indistinctly the terms station and CAF.

3. Friendly Environment Configuration

In this section we address the issue of finding the optimal configuration of a 802.11e EDCA WLAN in a friendly environment in which there exists a trust relationship with the users. In this scenario, users declare the traffic specifications of their applications and their QoS requirements, and the optimal configuration is computed based on these data. The fundamental difference between this scenario and the one of Section 5. is that here, in contrast to 5., users are trusted to adhere to the declared traffic specifications.

In this article we focus on a specific target application to illustrate the configuration of EDCA WLANs in a friendly environment. Specifically, we concentrate in a scenario in which all stations run a voice over IP (VoIP) application. This is indeed a very relevant scenario as voice traffic is one of the main targets of EDCA. We note, however, that this scenario is only taken as an example of EDCA configurations under a friendly environment, and that a similar analysis to the one presented here should be conducted in order to find the optimal configuration for other applications.

The rest of this section is organized as follows. We first present a number of considerations that allow us to fix the configuration of three of the parameters (CW_{max}, $AIFS$ and $TXOP$). Then, we present a model for the throughput and delay of EDCA as a step towards finding the optimal EDCA configuration. Our model, unlike previous analyses (see [3–5] and references therein), does not only account for the throughput and average delay characterization but also for the standard deviation of the delay. Indeed, we argue that variance is a fundamental measure in order to provide a real-time application such as voice traffic with meaningful QoS guarantees. Finally, we propose a concrete algorithm for the configuration of the EDCA parameters for voice traffic. Our algorithm takes as input parameters the number of voice stations, their arrival rate and the desired service quality criterion (namely, average delay and standard deviation), and provides as output the EDCA parameter values (if they exist) that satisfy this criterion. The results of this section can be found in [6].

3.1. Considerations on the Configuration

Our focus here is on a WLAN operating under voice traffic. As in this scenario we only have one traffic class (namely voice), there is no need for introducing any type of differentiation, and only AC needs to be used in the WLAN, with the same EDCA parameter values for all stations.

As a result of the above, we have that all the stations use the same $AIFS$ configuration. From this, it follows that the optimal setting for this parameter is its minimum possible value, namely $AIFS = DIFS$, as otherwise some extra time is unnecessarily lost after every transmission. This fixes the value of one of the four parameters.

We next consider the configuration of the CW_{max} parameter. 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 set so that the resulting collision probability corresponds to optimal operation. In addition, if we set CW_{max} larger than CW_{min}, the delay of the

packets that suffer one or more collision drastically grows, which harms jitter performance. Based on these arguments, we set $CW_{min} = CW_{max}$, which fixes another parameter.

Given the stringent delay requirements of voice traffic, the parameters setting for voice stations will typically be chosen such that their transmission queue never grows to more than one packet (in particular, this holds for the configurations that we propose in Section 3.4.). As a result of this, the $TXOP$ parameter will rarely be used. In the rest of this section, we do not further consider this parameter and simply assume that stations transmit only one packet when they access the channel.

Based on the above considerations, we have that three out of the four parameters of EDCA are fixed (namely $AIFS$, CW_{max} and $TXOP$); the rest of this section is devoted to finding the optimal configuration of the remaining parameter (CW_{min}).

3.2. Throughput Analysis

We next analyze the throughput performance of an EDCA WLAN with N voice stations as a function of the CW_{min} configuration. Following the behavior of many of today's most popular voice applications (like e.g. Skype), which do not use silence suppression, we model voice stations as CBR traffic sources that generate a voice packet of size L every time interval T.

The key variable upon which we base the throughput analysis is τ, defined as the probability that a station transmits in a randomly chosen slot time. Based on this variable, the throughput r experienced by a given station is computed as follows:

$$r = \frac{P_g L}{P_s T_s + P_c T_c + P_e T_e} \tag{1}$$

where P_g is the probability that a randomly chosen slot time contains a successful transmission of the given station, 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 slot time durations in each case.

The above probabilities are computed as a function of τ as follows:

$$P_g = \tau(1-\tau)^{N-1} \tag{2}$$

$$P_s = N\tau(1-\tau)^{N-1} \tag{3}$$

$$P_e = (1-\tau)^N \tag{4}$$

$$P_c = 1 - P_e - P_s \tag{5}$$

Given $\tau \ll 1$, these probabilities can be accurately approximated by

$$P_g \cong \tau(1-(N-1)\tau) \tag{6}$$

$$P_s \cong N\tau(1-(N-1)\tau) \tag{7}$$

$$P_e \cong (1-N\tau) \tag{8}$$

$$P_c \cong 1 - N\tau(1-(N-1)\tau) - (1-N\tau) \tag{9}$$

From the above, we have a formula to compute the throughput as a function of τ, $r(\tau)$. Based on this, we can obtain the throughput performance as a function of CW_{min} as follows. We say that a station is saturated when it always has packets ready for transmission. The τ value of such a station will be [4]

$$\tau_{sat} = \frac{2}{CW_{min} + 1} \tag{10}$$

If, for a given CW_{min} configuration we have that $r(\tau_{sat}) < L/T$, then stations will be saturated[1], as their incoming rate L/T will be larger than the outgoing rate $r(\tau_{sat})$. Throughput performance in this case will be the given by $r(\tau_{sat})$. On the other hand, if the CW_{min} configuration is such that $r(\tau_{sat}) \geq L/T$, then stations will not be saturated and their throughput will be equal to the incoming rate, L/T.

Based on the above throughput analysis, we now analyze the τ value at which stations operate. In case of saturation, the value of τ is directly given by Eq. (10). In case of non saturation, the throughput experienced by the stations is equal to their incoming rate, and therefore the τ of operation has to satisfy the following second order equation:

$$r(\tau) = L/T \tag{11}$$

From Figure 1, which plots $r(\tau)$ as a function of τ, it can be seen that the above equation has two solutions: τ_1 and τ_2. We next show that the τ of operation corresponds to the smallest of the two, i.e. τ_1.

From the fact that under non saturation $r(\tau_{sat}) > L/T$, we have that the value of τ_{sat} surely falls between τ_1 and τ_2. Note that τ_{sat} corresponds to the extreme case when a station always has packets ready for transmission and only waits one backoff process between each transmission and the next one. Therefore, τ_{sat} represents an upper bound on the maximum τ at which the station can possibly operate. As a consequence of this reasoning, we have that τ_2 cannot be the point of operation, which leaves τ_1 as the only possible solution.

3.3. Delay Analysis

We next analyze, as a function of the τ of operation obtained in the previous section, the delay performance of the WLAN. Specifically, our focus is on the time elapsed between the beginning of the backoff process and the successful transmission of a packet. Given our assumption of Section 3.1. that EDCA parameters are set such that transmission queues do not grow to more than one packet, this corresponds to the total delay of the WLAN.

We start by analyzing the average value of the delay. This can be computed as follows:

$$E[d] = \sum_{j=0}^{R} P_{tx}(j) E[d_j] \tag{12}$$

where $P_{tx}(j)$ is the probability that a packet is successfully transmitted after j retries and $E[d_j]$ is the expected delay in this case. $P_{tx}(j)$ is computed as

$$P_{tx}(j) = (1-p)p^j \tag{13}$$

[1]The reader is referred to [5] for a more detailed discussion on the throughput behavior as a function of the CW_{min} configuration.

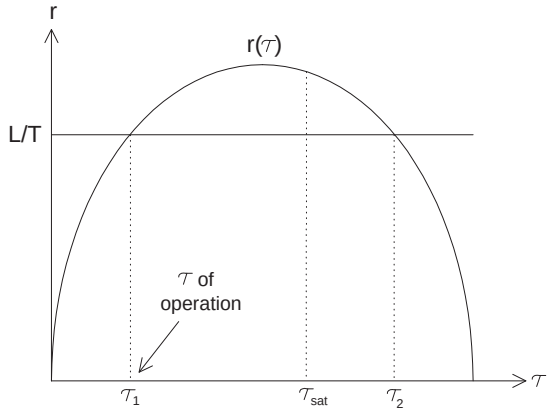

Figure 1. τ of operation.

where p is the probability that a transmission attempt collides, which is given by

$$p = 1 - (1 - \tau)^{N-1} \tag{14}$$

$E[d_j]$ is computed as follows:

$$E[d_j] = T_s + jT_c + E[d_{bo}(j)] \tag{15}$$

where $E[d_{bo}(j)]$ is the total time spent in average with backoff counter decrements for the case of j collisions. This is calculated as

$$E[d_{bo}(j)] = jE[c_{bo}]E[T_{slot}] \tag{16}$$

where $E[c_{bo}]$ is the expected backoff time counter drawn at the beginning of a backoff process and $E[T_{slot}]$ is the average duration of a slot time when the considered station does not transmit.

Since the backoff time counter is calculated from a uniform distribution between 0 and $CW_{min} - 1$, $E[c_{bo}]$ is equal to

$$E[c_{bo}] = \frac{CW_{min} + 1}{2} \tag{17}$$

Finally, $E[T_{slot}]$ is calculated as follows by noting that during those slot times the considered station does not transmit:

$$E[T_{slot}] = P_{e,N-1}T_e + P_{s,N-1}T_s + P_{c,N-1}T_c \tag{18}$$

where

$$P_{e,N-1} = (1 - \tau)^{N-1} \tag{19}$$

$$P_{s,N-1} = (N - 1)\tau(1 - \tau)^{N-2} \tag{20}$$

$$P_{c,N-1} = 1 - P_{e,N-1} - P_{s,N-1} \tag{21}$$

which terminates the analysis of the average delay.

Next, we analyze the standard deviation of the delay. The analysis follows the same lines as the computation of the average delay in the previous section. The standard deviation of the delay can be computed as a function of the first and second moments of the delay as follows:

$$\sigma_d = \sqrt{E[d^2] - E[d]^2} \tag{22}$$

$E[d]$ has already been computed above. To compute $E[d^2]$, we proceed similarly as in Eq. (12):

$$E[d^2] = \sum_{j=0}^{R} P_{tx}(j) E[d_j^2] \tag{23}$$

$P_{tx}(j)$ has already been obtained in Eq. (13). By definition, $E[d_j^2]$ can be expressed as

$$E[d_j^2] = E[d_j]^2 + \sigma_{d_j}^2 \tag{24}$$

where $E[d_j]$ has already been computed in Eq. (15).

The remaining challenge is the computation of $\sigma_{d_j}^2$. Since T_s and T_c are constants, from Eq. (15) it follows

$$\sigma_{d_j}^2 = \sigma_{d_{bo}(j)}^2 \tag{25}$$

Since in case of j retransmission, the total backoff delay is composed of j backoff components, we have

$$\sigma_{d_{bo}(j)}^2 = j\sigma_{d_{bo}}^2 \tag{26}$$

where $\sigma_{d_{bo}}$ can be expressed as

$$\sigma_{d_{bo}}^2 = E[d_{bo}^2] - E[d_{bo}]^2 \tag{27}$$

$E[d_{bo}]$ has already been obtained above. $E[d_{bo}^2]$ can be calculated as

$$E[d_{bo}^2] = \sum_{k=0}^{CW_{min}-1} P_{bo}(k) E[(\underbrace{T_{slot} + T_{slot} + \cdots + T_{slot}}_{k \text{ times}})^2] \tag{28}$$

where $P_{bo}(k) = 1/CW_{min}$ is the probability that the backoff counter drawn is equal to k and

$$E[(\underbrace{T_{slot} + T_{slot} + \cdots + T_{slot}}_{k \text{ times}})^2] = k^2 E[T_{slot}]^2 + k\sigma_{T_{slot}}^2 \tag{29}$$

Finally, by combining the above two equations,

$$\begin{aligned} E[d_{bo}^2] &= \frac{E[T_{slot}]^2}{CW_{min}} \sum_k k^2 + \frac{\sigma_{T_{slot}}^2}{CW_{min}} \sum_k k = \\ &= E[T_{slot}]^2 \frac{(CW_{min}-1)(2CW_{min}-1)}{6} + \sigma_{T_{slot}}^2 \frac{CW_{min}-1}{2} \end{aligned} \tag{30}$$

where

$$\sigma_{T_{slot}}^2 = E[T_{slot}^2] - E[T_{slot}]^2 \tag{31}$$

$$E[T_{slot}^2] = P_{e,N-1}T_e^2 + P_{s,N-1}T_s^2 + P_{c,N-1}T_c^2 \quad (32)$$

which terminates the delay standard deviation analysis.

We validated the accuracy of our analysis by comparing analytical results against simulations. For the simulations, we used an event-driven simulator that closely follows the 802.11e EDCA behavior for each station. The experiments were performed for a WLAN with the system parameters of the IEEE 802.11b physical layer. Following the behavior of standard PCM codecs, voice sources generated one 80 byte packet every 10 ms.

Figures 2 and 3 plot the average and standard deviation of the backoff delay for different configurations of the CW_{min} parameter as well as 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 heavy loaded WLAN, respectively. Simulation results are plotted with 95% confidence intervals, although these are so small that can barely be appreciated in the graphs.

From the figures, we observe that analytical results match simulations remarkably well, which confirms the accuracy of our analysis. We further observe that delays show the following behavior:

- For too low CW_{min} values, the WLAN is saturated and delays are very large.
- As CW_{min} increases, after crossing a certain threshold (which varies for different N values) the WLAN leaves saturation and delays decrease sharply.
- After this threshold, delays increase gradually with the CW_{min}. The reason for this gradual increase is that, the larger the CW_{min}, the longer the completion of the backoff process takes.

From the above, it can be intuitively seen that the CW_{min} values that provide the best performance are the ones close to the saturation threshold. In the following, we address the issue of finding this optimal configuration.

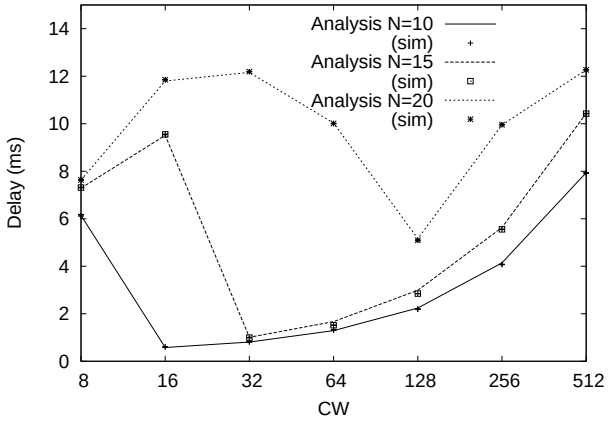

Figure 2. Validation of the average delay.

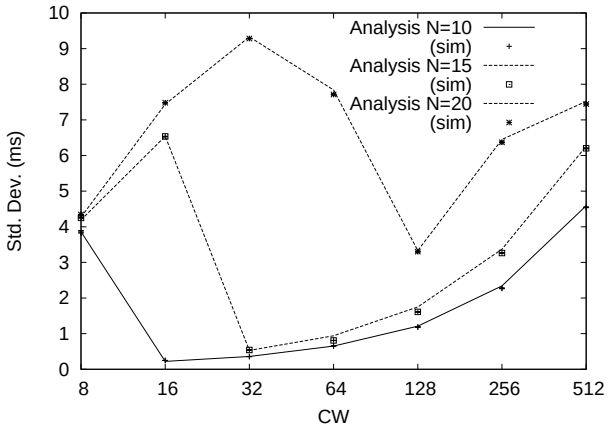

Figure 3. Validation of the delay standard deviation.

3.4. Optimal Configuration

We next present an algorithm that, given the desired performance for voice traffic, finds the optimal configuration that satisfies this quality criterion. Specifically, our algorithm takes as input the desired upper bound values for the average delay and its standard deviation (D_{max} and σ_{max}) and provides the following output: i) it determines if there exists any CW_{min} configuration that meets the given requirements, and ii) if it exists, it gives the optimal CW_{min} configuration.

In the following, we first obtain some lower and upper bounds for CW_{min} and then, based on these bounds, we propose an algorithm to calculate the optimal CW_{min}. We start by analyzing the CW_{min} range that provides good throughput performance.

According to Section 3.2., the WLAN will not be saturated as long as CW_{min} is set such that the following condition holds: $r(\tau_{sat}) \geq L/T$, where τ_{sat} is a function of CW_{min} as given by Eq. (10). For any CW_{min} that does not meet this condition, the outgoing rate will be smaller than the incoming one and as a result throughput performance will be degraded. As it can be observed from Figure 4, this imposes a lower and an upper bound on CW_{min}. Hereafter, we refer to these bounds as CW_1 and CW_2, respectively.

We now analyze the CW_{min} range to meet the given delay performance requirements. According to the average delay analysis of Section 3.3., as long as the WLAN is not saturated (which is given by the above bounds) average delay is an increasing function of CW_{min}. As a result, the requirement that average delay cannot exceed a given D_{max} value imposes an additional upper limit on CW_{min}, which we refer to with CW_3. Indeed, as it can be seen from Figure 5, for any CW_{min} value larger than CW_3 the average delay will not meet the given criterion. Following a similar reasoning as above, we have that the requirement on the delay standard deviation imposes yet an additional upper limit, which we refer to with CW_4.

We next propose an algorithm to compute the optimal CW_{min} based on the lower bound (CW_1) and three upper bounds (CW_2, CW_3 and CW_4) obtained above. From above, we have that any CW_{min} that falls within the bounds meets the given quality criterion. The remaining challenge is to choose one CW_{min} value within this range. Based on the follow-

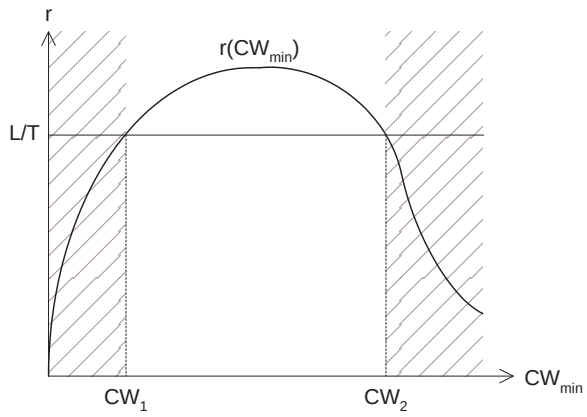

Figure 4. CW_{min} bounds for throughput.

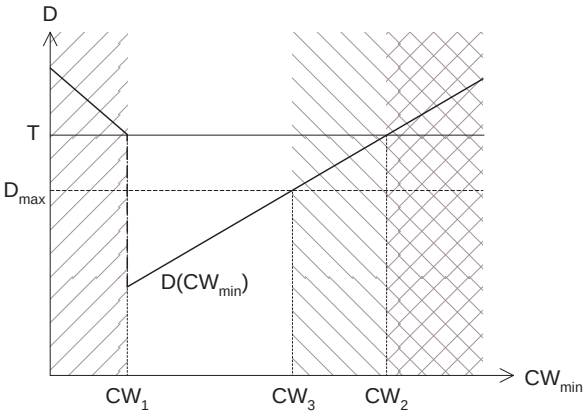

Figure 5. CW_{min} bound for average delay.

ing argument, we choose the largest possible value. As it can be observed from Figure 5, in the given range delay performance improves as CW_{min} decreases. The problem, however, is that as CW_{min} approaches CW_1, there is the risk of suffering a sharp performance decrease. In order to avoid this, we choose the CW_{min} value that, while meeting the given criterion, falls as far as possible from this critical point.

We next present our algorithm resulting from all the above considerations. Note that the algorithm is extremely efficient as each of the steps only involves the calculation of one equation of first or second order:

- In the first step, we compute CW_1 and CW_2 by solving $r(CW_{min}) = L/T$, using the expression for $r(CW_{min})$ obtained in Section 3.2. with the τ of operation for saturation.

- Next, we compute CW_3 by solving $E[d] = D_{max}$, using the $E[d]$ expression of Section 3.3. with the τ of operation for non saturation.

Table 1. Algorithm validation.

D_{max}	σ_{max}	N	CW_{alg}	D_{alg}	σ_{alg}	CW_{exh}	D_{exh}	σ_{exh}
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

- We then obtain CW_4 by solving $\sigma_d = \sigma_{max}$, using in this case the expression for σ_d of Section 3.3..

- As a final step, the algorithm compares the lower bound (CW_1) with the minimum of all upper bounds (CW_2, CW_3 and CW_4): if $CW_1 > min(CW_2, CW_3, CW_4)$, there exists no CW_{min} value that satisfies the desired quality criterion and the algorithm indicates that it is not possible to admit the given number of voice calls.

- Otherwise, the algorithm terminates by giving the following optimal configuration: $CW_{min} = min(CW_2, CW_3, CW_4)$.

We validated our algorithm by comparing the performance of our configuration given by the algorithm ($CW_{algorithm}$) against the result of performing an exhaustive search over the CW_{min} space ($CW_{exhaustive}$). Specifically, for the exhaustive search we evaluated by means of simulation the delay performance of all possible CW_{min} values and took the largest CW_{min} that met the given quality criterion. We performed this experiment for three different quality criteria ranging from a more stringent criterion ($D_{max} = \sigma_{max} = 2.5ms$) to a more relaxed one ($D_{max} = \sigma_{max} = 5ms$).

Simulation results are presented in Table 1. It can be seen that the proposed configuration is always very close to the one obtained from the exhaustive search. In all three experiments, our algorithm admits as many voice calls as the exhaustive search (20 for the first two experiments and 19 for the third one). In addition, the desired quality criteria are always met by our configuration.

We conclude that our algorithm is effective in admitting as many voice calls as possible while guaranteeing the desired performance. The proposed algorithm finds therefore the best possible configuration in a friendly environment in which the application behavior is known and can be trusted and the application requirements are also known. In the following two sections we address other scenarios in which these data are not known (Section 4.) or cannot be trusted (Section 5.).

4. Configuration for Fair Resource Allocation

In this section we address the issue of finding the optimal configuration of EDCA when the information available about the stations connected to the WLAN is very reduced. Specifically, we assume that (in contrast to the previous section) we have neither information about the applications running in each station nor about their QoS requirement. The only information that we assume is the *weight* assigned to each station, which is statically set and represents the priority of the station as explained in the throughput allocation criterion below. Specifically, as stations are grouped by AC's, we take the *weight* assigned to each of the AC's in the WLAN as the input to find the appropriate configuration. If not even this information on *weights* was available, the configuration proposed here can also be used by simply assigning the same weight to all AC's. The results presented here have been published in [4].

4.1. Throughput Allocation Criterion

While there are many different criteria proposed in the literature for throughput allocation, *weighted max-min fairness* [7–9] is a widely accepted one[2]. The weighted max-min fair allocation is the one that 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 this section we set our objective to find the configuration that provides weighted max-min fairness in the WLAN, the WLAN stations being our *entities*, and the saturation throughput of a WLAN station its *allocated throughput*. Note that the saturation throughput in a WLAN [10] corresponds to the notion of *allocated throughput* in weighted max-min fairness: the former assumes that all stations always have packets to transmit, while the latter assumes that all entities are using all the throughput to which they are entitled.

In the rest of the section we present an analysis that finds the configuration of each AC that maximizes $min(r_i/w_i)$. We refer to the parameters of AC i with $AIFS_i$, CW_i^{min} and CW_i^{max} and $TXOP\_limit_i$, respectively.

According to Theorem 2 of [4], it can be seen that the $AIFS_i$ parameter should be set to the minimum possible value for all AC's in order to optimize throughput performance. In the following, we take this configuration for the $AIFS_i$ parameter and search for the optimal values of the remaining parameters.

4.2. CW_i^{min} and CW_i^{max} Configuration

Following [4], it can be seen that the search for the optimal parameter configuration can be restricted to the solutions that satisfy

$$\frac{r_i}{r_j} = \frac{w_i}{w_j} \ \forall i, j \tag{33}$$

since, according to Theorem 3 of [4], for any configuration that does not satisfy the above condition, there exists a configuration that satisfies the condition and provides equal or

[2]Weighted max-min fairness is e.g. the criterion provided by Weighted Fair Queueing, which is the most widely implemented mechanism for throughput allocation in wired links. Many works in the literature have aimed at providing weighted max-min fairness in WLAN (see e.g. [11–14]).

better throughput performance. This reduces our problem to finding the configuration that maximizes the $min(r_i/w_i)$ under the constraint of Eq. (33).

The throughput of a station of AC i can be computed as

$$r_i = \frac{P_i l_i}{P_s T_s + P_c T_c + P_e T_e} \tag{34}$$

where P_i is the probability that a slot time contains a successful transmission of a given station of AC i, l_i is the average length of the packets of that station[3], 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.

The probability P_i is computed as

$$P_i = \tau_i (1-\tau_i)^{n_i - 1} \prod_{j \in S \setminus i} (1-\tau_j)^{n_j} \tag{35}$$

where τ_i is the probability that a station of AC i transmits in a slot time, n_i is the number of stations that belong to AC i and S is the set of AC's in the WLAN.

The other probabilities are computed as follows:

$$P_s = \sum_{i \in S} n_i P_i \tag{36}$$

$$P_e = \prod_{j \in S} (1-\tau_i)^{n_i} \tag{37}$$

$$P_c = 1 - P_e - P_s \tag{38}$$

From the above expression for r_i, it can be seen that the condition of Eq. (33) can be rewritten as

$$\frac{\tau_i (1-\tau_j)}{\tau_j (1-\tau_i)} = \frac{w_i}{w_j} \tag{39}$$

Under the assumption of $\tau_i \ll 1 \; \forall i$ — which is reasonable in optimal operation, as large τ_i values would lead to a high collision probability — Eq. (39) is approximately equivalent to

$$\frac{\tau_i}{\tau_j} \approx \frac{w_i}{w_j} \tag{40}$$

Next, we use Eqs. (33) and (40) to find the optimal τ_i's. From Eq. (33) we have

$$r_j = \frac{w_j}{\sum_{i \in S} n_i w_i} r \tag{41}$$

where r is the total throughput in the WLAN.

With the above, the problem of finding the optimal configuration can be reformulated as to find the τ_i values that maximize r subject to the condition of Eq. (33), as any other set that complies with this condition will lead to smaller $r_i \; \forall i$, and therefore smaller $min(r_i/w_i)$.

[3]For simplicity hereafter we assume that l_i takes the same value for all stations and refer to it with l.

The total throughput r can be expressed as

$$r = \frac{p(s)\,l}{p(s)T_s + p(c)T_c + p(e)\sigma} = \frac{l}{T_s - T_c + \frac{p(e)(\sigma - T_c) + T_c}{p(s)}} \tag{42}$$

As l, T_s, and T_c are constant, maximizing the following expression will result in the maximization of r,

$$\hat{r} = \frac{p(s)}{p(e)(\sigma - T_c) + T_c} \tag{43}$$

From Eq. (40) we have $\hat{r}$ can be approximated by

$$\hat{r} \approx \frac{a(\tau_1/w_1) - b(\tau_1/w_1)^2}{c(\tau_1/w_1) + \sigma} \tag{44}$$

where AC 1 is taken as reference, with

$$a = \sum_{i \in S} n_i w_i \tag{45}$$

$$b = \sum_{i \in S} \sum_{j \in S \setminus \{1,\ldots,i\}} n_i n_j w_i w_j \tag{46}$$

$$c = \sum_{i \in S} n_i w_i (T_c - \sigma) \tag{47}$$

The optimal value of τ_1, τ_1^{opt}, that maximizes $\hat{r}$ can then be obtained by

$$\frac{d\,\hat{r}}{d\,\tau_1}\Big|_{\tau_1 = \tau_1^{opt}} = 0 \implies bc\left(\frac{\tau_1^{opt}}{w_1}\right)^2 + 2b\sigma\left(\frac{\tau_1^{opt}}{w_1}\right) - a\sigma = 0 \tag{48}$$

which yields

$$\tau_1^{opt} = w_1 \frac{\sqrt{(b\sigma)^2 + abc\sigma} - b\sigma}{bc} \tag{49}$$

Finally, applying Eq. (39) to τ_1^{opt}, we obtain our approximation to the optimal τ_i values,

$$\tau_i^{opt} = \frac{w_i \tau_1^{opt}}{w_1(1 - \tau_1^{opt}) + w_i \tau_1^{opt}} \tag{50}$$

The remaining challenge is to find the CW_i^{min} and CW_i^{max} configuration that leads to the optimal τ_i values obtained above. From [15], the probability τ_i in saturation conditions can be expressed as follows:

$$\tau_i = \frac{2(1-2p_)(1-p_)^{R+1})}{CW_i^{min}(1-(2p_i)^{m_i+1})(1-p_i)+(1-2p_i)(1-p_)^{R+1})+CW_i^{min}2^{m_i}p_)^{m_i+1}(1-2p_i)(1-p_i^{R-m_i})} \tag{51}$$

where p_i is the probability that a transmission attempt of a station of AC i collides and m_i is defined such that $CW_i^{max} = 2^{m_i} CW_i^{min}$.

From the above, 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 level of freedom to adjust these parameters in order to obtain the desired τ_i. If we fix m_i, then the CW_i^{min} value that leads to τ_i^{opt} can be computed from Eq. (51) as follows,

$$CW_i^{min} = \frac{(1-2p_i)(1-p_i^{R+1})}{(1-(2p_i)^{m_i+1})(1-p_i)+2^{m_i}p_i^{m_i+1}(1-2p_i)(1-p_)^{R-m_i})} \left(\frac{2}{\tau_i^{opt}} - 1\right) \tag{52}$$

where p_i is computed from the τ_i^{opt} values according to

$$p_i = (1-\tau_i^{opt})^{n_i-1} \prod_{j \in S \setminus i} (1-\tau_j^{opt})^{n_j} \tag{53}$$

Note, however, that Eq. (52) does not necessarily yield an integer CW_i^{min} value; to meet the requirement that contention windows must take integer values, we round CW_i^{min} to the closest integer, i.e.[4]

$$CW_i^{min} = \text{round int}\left(CW_i^{min}(Eq.\ (52))\right) \tag{54}$$

In the following, unless otherwise specified, we set (following similar arguments to the ones given in Section 3.) $m_i = 0$, from which $CW_i^{min} = CW_i^{max} = CW_i$. With this setting, Eq. (52) is simplified to

$$CW_i = \frac{2}{\tau_i^{opt}} - 1 \tag{55}$$

4.3. $TXOP\_limit_i$ Configuration

The remaining open issue is to find the optimal $TXOP\_limit_i$ configuration. Throughput performance increases with larger $TXOP\_limit_i$ values, since larger transmission times means lower overhead for each transmitted bit. However, $TXOP\_limit_i$ can not be set based only on throughput performance considerations, as throughput performance would be optimized with infinite payload size transmissions, but this would lead to infinite delays, which is clearly undesirable.

Based on the above, we propose to set the $TXOP\_limit_i$ of all the AC's to the maximum acceptable value according to delay and/or other considerations, while configuring the other three parameters (CW_i^{min}, CW_i^{max} and $AIFS_i$) following the algorithm given here. For example, given a maximum allowed delay, if the other EDCA parameters are configured according to the formulae given here, then the value of the $TXOP\_limit_i$ parameter can be computed from our delay model of EDCA in [16].

Hereafter we assume that the configuration of the $TXOP\_limit_i$ parameter is set to a fixed value (corresponding to a certain payload size) and do not further consider this parameter.

[4]Note that, for $\tau_i^{opt} \ll 1$, the error resulting from the rounding operation is very small.

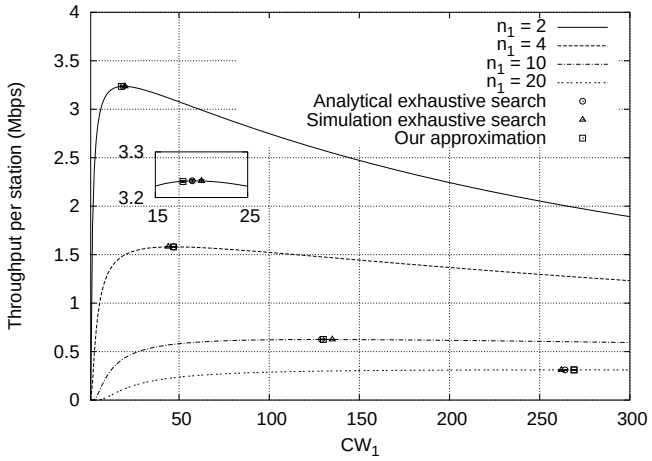

Figure 6. One AC.

4.4. Optimal Configuration Validation

The optimal configuration proposed here is based on a number of approximations. To assess the validity of the configuration proposed, we next compare it with the result of performing an exhaustive search over the entire configuration space. Specifically, we evaluate (analytically or via simulation) the throughputs resulting from all possible configurations (or a range wide enough) and choose the one that leads to the maximum $min(r_i/w_i)$, against which we compare our configuration.

In the following, we refer to the three methods mentioned above as "our approximation" to the optimal configuration, "analytical exhaustive search" and "simulation exhaustive search". Note that the analytical and simulation exhaustive search methods are unfeasible for practical use, as they require a large amount of time and computational resources to find the optimal configuration; our intent here is to use them as a benchmark to assess the accuracy of our approximation.

We first study the simplest possible scenario in which there is only one AC present in the WLAN with weight $w_1 = 1$. In this case, our objective of maximizing the minimum r_i/w_i is equivalent to finding the CW_1 value that, configuring the AC with this value, maximizes the individual throughputs.

Fig. 6 shows the optimal configuration resulting from our approximation, analytical exhaustive search and simulation exhaustive search, for different numbers of stations. The resulting throughputs are given for each case; the throughputs obtained analytically are represented with lines, and the throughputs obtained via simulation with points and confidence intervals. For the simulation exhaustive search, the simulation throughputs have been obtained by rerunning the simulations for the selected configuration with different seed values.

We observe that the optimal CW_1 values given by our approximation are very close to the ones obtained with the exhaustive search methods, and the resulting throughputs are practically identical. As throughput is the only relevant metric for our objective of maximizing $min(r_i/w_i)$, we conclude that these results validate our approach.

To assess performance in case of multiple AC's, we study a scenario with 4 AC's,

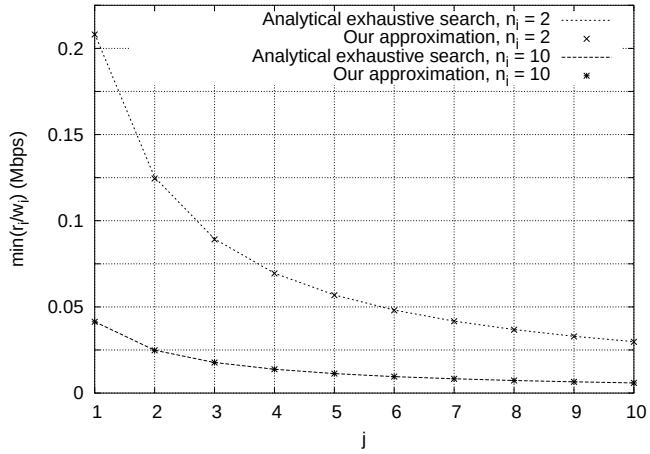

Figure 7. Multiple AC's.

$i \in \{1, \ldots, 4\}$, each AC i with n_i stations ($n_i = 2$ and 10) and a weight $w_i = w_1 + (i-1)j$, with $w_1 = 1$ and $j \in [1, 10]$. Fig. 7 shows the $min(r_i/w_i)$ values, obtained analytically, corresponding to the analytical exhaustive search method and our approximation to the optimal configuration. Results validate our approach also for this case, as the throughput performance given by our approximation is very close to the performance of the analytical exhaustive search method.

5. Configuration in an Unfriendly Environment

In this section we address the issue of configuring EDCA WLANs in order to provide users with service guarantees in an *unfriendly scenario*. By unfriendly, we mean that applications cannot be assumed to be well-behaved, and they may behave differently than the traffic specifications declared by them in order to gain extra resources. This contrasts with the friendly scenario assumed in the Section 3., in which applications were supposed to conform to the declared traffic specifications.

The only means of controlling the load offered by a station in the above scenario is by configuring appropriately its EDCA parameters. Note that indeed in EDCA there is no other way of controlling the traffic sent by an application. Therefore, we are imposing here an extra requirement to the EDCA configuration as compared to Section 3., and consequently the resulting optimal configuration will differ from the optimal one of Section 3.. For instance, a low rate application with stringent delay requirements was assigned very small $AIFS$ and CW parameters in Section 3., since this configuration provides the station with small delays. However, this cannot be done in an unfriendly scenario, since a station with very small $AIFS$ and CW parameters could possible consume the entire capacity of the WLAN if it sent more traffic than declared, and would thereby starve the other stations.

The focus of this section is on the computation of the optimal configuration in an unfriendly scenario such as the one described above. Our goal in the computation of the EDCA configuration is to provide a target station with its service guarantees independent of the behavior of the other stations.

5.1. EDCA Configuration

The problem that we address in this section is formulated as follows. Given a set of stations S, each station $i \in S$ with certain service requirements, our objective is to find the configuration of the parameters of all stations ($AIFS_i$, $TXOP\ limit_i$, CW_i^{max} and CW_i^{min}) that guarantees to each station its service requirements *independent of the behavior of the other stations*.

The objective of the configuration proposed is the following. For an application that generates packets according to some given traffic specifications, we want to guarantee that the station's delay requirements will be met. Note that we cannot control that the application conforms to the given traffic specifications; however, the station's delay requirements only need to be met *as long as the application's sending behavior conforms to these specifications*.

Let us define the saturation throughput of a station i (r_i^{sat}) as the throughput that this station would obtain if all stations (including station i), with their respective EDCA configurations, saturated their channel, i.e., always had a packet ready for transmission. The key assumption upon which the optimal configuration obtained in this section is based is the following: we assume that, given a certain arrival process and delay requirements of a station, there exists a r_i^{sat} value such that, if the station is provided with a saturation throughput equal to r_i^{sat}, this ensures that the station will receive its desired guarantees *independent of the behavior of the other stations*.

With the above assumption, a station with certain service requirements does not need to inform the entity responsible for computing the optimal configuration (hereafter the *configuration server*) of its arrival process and delay requirements, but can *simply compute the r_i^{sat} value that guarantees its service requirements and request the corresponding saturation throughput to the configuration server*.

The above simplifies greatly both the communication between the stations and the configuration server and the computations to be performed by the configuration server. Specifically, only operations dealing with the saturation throughput need to be performed. One additional piece of information that stations need to provide to the configuration server is their transmission length, since this information is necessary in the computation of saturation throughputs.

With the above, the problem of finding the optimal configuration is reduced to finding the configuration that satisfies a set of saturation throughput requirements. Specifically, given a set of stations and their requirements in terms of saturation throughput (r_i^{sat}) and transmission length (l_i), we need to find (if it exists) the configuration of all stations ($AIFS_i$, $TXOP\ limit_i$, CW_i^{max} and CW_i^{min}) that satisfies these requirements.

The problem of finding the EDCA configuration that satisfies a set of saturation throughput requirements is precisely the one that has been addressed in Section 4.. Therefore, we can reuse here the analysis of that section in order to compute the optimal configuration of the EDCA parameters. Note that, although the saturation throughput analysis is shared by the two scenarios, the way of determining the saturation throughput is fundamentally different for the two cases: in Section 4. the saturation throughputs were computed as a function of some pre-assigned *weights* for throughput allocation, while here they are computed as a function of the *service guarantees* desired for a given station.

The optimal configuration proposed here is therefore based on the following two premises:

- Our key assumption is that it is enough for a station to request a certain saturation throughput (given the length of its transmissions) in order to guarantee that the requirements of the applications will be met.
- Stations need to be able to map the service guarantee requirements of their applications into requesting the appropriate l_i and r_i^{sat} parameters.

Next, we first proof our key assumption, and then we propose some configuration guidelines for the stations to compute (given the applications requirements) the l_i and r_i^{sat} parameters to request.

5.2. Key Assumption

We now proceed to demonstrate our key assumption, which states that, given the requirements of a station, there exists a saturation throughput value that guarantees these requirements independent of the behavior of the other stations. Specifically, if we can find a saturation throughput value that provides the station with the desired guarantees when all other stations are saturated *independent of the number of the stations in the WLAN and their requirements*, then we will have proved our assumption.

Let us analyze the distribution of the service time of a station (i.e., the time elapsed between a packet reaching the first position in the transmission queue and its transmission),

$$P(d_s > D_s) = \sum_{j=0}^{R-1} (1-p_i)p_i^j P(d_{s,j} > D_s) \tag{56}$$

where p_i is the probability that a transmission attempt of station i collides, j is the number of attempts before the station transmits successfully and $d_{s,j}$ is the service time given that the station has performed j attempts before transmitting successfully.

Based on the results of [17], we can obtain an accurate estimation of the service delay distribution by considering that the duration of all slot times is fixed and equal to the average slot time duration T_{slot}. By using this approximation,

$$P(d_{s,j} > D_s) = P\left(T_{slot}U_j(CW_i^{min}) > D_s\right) \tag{57}$$

where $U_j(x) = \sum_{k=1}^{j} unif(0,x)$, $unif(0,x)$ being a random variable uniformly distributed between 0 and x.

From the above,

$$P(d_{s,j} > D_s) = P\left(T_{slot}CW_i^{min}U_j(1) > D_s\right) \tag{58}$$

From the fact that the average backoff delay of the station if it was saturated would be to l_i/r_i^{sat}, r_i^{sat} being its saturation throughput, we have that

$$\sum_{j=0}^{R-1} (1-p_i)p_i^j j\frac{CW_i^{min}}{2}T_{slot} = \frac{l_i}{r_i^{sat}} \tag{59}$$

from which $CW_i^{min}T_{slot}$ can be expressed as a function of l_i, r_i^{sat} and p_i,

$$CW_i^{min}T_{slot} = f(l_i, r_i^{sat}, p_i) \tag{60}$$

Combining the above equations it follows that

$$P(d_s > D_s) = f(l_i, r_i^{sat}, p_i) \tag{61}$$

The probability p_i is computed as follows

$$p_i = 1 - \prod_{j \in S \setminus i} (1 - \tau_j) \tag{62}$$

which, under the assumption $\tau_j \ll 1$, can be approximated by

$$p_i \approx 1 - \left(1 - \sum_{j \in S \setminus i} \tau_j\right) = \sum_{j \in S \setminus i} \tau_j \tag{63}$$

From Eq. (50),

$$\tau_j \approx w_j \left(\sqrt{\left(\frac{\sigma}{c}\right)^2 + \frac{a\sigma}{bc}} - \frac{\sigma}{c}\right) \tag{64}$$

Given $T_c \gg \sigma$, we have that the term σ/c is negligible compared to $a\sigma/bc$ and therefore

$$\tau_j \approx w_j \sqrt{\frac{a\sigma}{bc}} \tag{65}$$

Substituting a, b and c and using the approximation $b \approx (\sum_i w_i)^2/2$,

$$\tau_j \approx \frac{w_j}{\sum_i w_i} \sqrt{\frac{2\sigma}{T_c}} \tag{66}$$

Finally, substituting the above into Eq. (63) yields

$$p_i \approx \sqrt{\frac{2\sigma}{T_c}} \tag{67}$$

which shows that (under our optimal configuration) the collision probability is *independent of the number of stations and their requirements* and depends only on the average duration of the collisions.

Applying the above results to Eq. (61) leads to

$$P(d_s > D_s) = f(l_i, r_i^{sat}, T_c) \tag{68}$$

From the above, by taking a lower bound on T_c (like e.g. 100 bytes packet length) we obtain an upper bound for the distribution of the service *which depends only on the station's packet length and its provisioned saturation throughput, and is independent of the other stations*.

We have therefore our assumption proofed. Indeed, the end-to-end delay distribution is a function of the station's arrival process and the service time distribution. Since the service time distribution depends only on the station's packet length and the saturation throughput provisioned, we have that the end-to-end delay distribution depends only on the station's arrival process, the packet length and the saturation throughput. Therefore, a station, given its arrival process and packet length, simply by requesting the appropriate saturation throughput can be sure that its service requirements will be met *independent of the other stations in the WLAN and their requirements*.

It is a major finding that in an optimally configured WLAN (following the optimal configuration given the previous section) its delay behavior can be guaranteed based only on its saturation throughput (i.e., the throughput it would obtain if it always had a packet ready for transmission) independently of the other stations in the WLAN. Following this finding, in our configuration each station calculates the saturation throughput it needs according to its service requirements and issues the corresponding request to some configuration server, and this configuration server only needs to compute the configuration that satisfies the received saturation throughput requests. Note that this greatly simplifies the computation of the optimal configuration at the configuration server.

5.3. Stations' Configuration Guidelines

We next provide some guidelines to stations on which values to request. According to the above, by requesting appropriate values for saturation throughput and transmission length, applications requests can be satisfied independent of the other stations in the WLAN. It is important to note that the guidelines provided here are only recommendations and stations do not need to follow them.

Let l_i^* be the station's packet length and r_i^{sat} its requested saturation throughput, and let $T_i^{sat} = l_i^*/r_i^{sat}$ be the packet interarrival time under saturation conditions. If the transmission is long enough to fit several packets (say x packets), this means that the station can only be allowed to transmit once every xT_i^{sat} interval, in order to limit the station's throughput to r_i^{sat}. Every time it accesses the channel, the station then transmits bursts of x packets.

Note that with the above setting the station transmits its data in bundles of size xl_i^*, which is equivalent to using a larger packetization size, namely xl_i^*. Packetization sizes are chosen following the applications' delay requirements: if packets of length l_i^* are used, this means that the application cannot afford waiting until more data is generated before transmitting it.

Based on the above, we conclude that setting the transmission length such that a station can transmit more than one packet upon accessing the channel yields unacceptably long delays, and therefore we recommend to set the transmission length data size of station i equal to the packet length of the station ($l_i = l_i^*$). Hereafter we use l_i and l_i^* indistinctly.

The above configuration basically "disables" the $TXOP\ limit$ mechanism since it only allows that each station transmits a single packet upon accessing the channel. Note that this setting is necessary given our starting assumption of an unfriendly scenario; if we assigned a $TXOP\ limit$ too long to a station, this could be used by the station to persistently transmit for long durations and thus starve other stations. Instead, if the applications behavior could

be trusted, we could assign a long $TXOP\ limit$ to a given station in order to absorb eventual data bursts generated by the station without risking starving the other stations. Indeed, if the application's behavior is well-behaved, we can be sure that the station will not use this long $TXOP\ limit$ to persistently transmit for long durations.

We note that, although following the above reasoning we advocate setting the $TXOP\ limit$ equal to the duration of a packet, it is up to the station to request a longer $TXOP\ limit$ value. Specifically, since the transmission length l_i is reported by the station, the station can obtain a longer $TXOP\ limit$ simply by reporting a larger l_i. This, however, will harm the station's delay, and therefore it is not the configuration recommended in this paper.

We now study the other parameter, the saturation throughput, that a station needs to request in order to ensure that its service requirements will be met. Note that, although in an unfriendly scenario a target station is not forced to conform to its declared specifications, the service guarantees provided only need to be met *as long as the station conforms to these specifications*. The worst-case for the target station corresponds to the case when all other stations saturate their channel (i.e., always have a packet ready for transmission). Therefore, the scenario we are interested in to achieve our goal of providing service guarantees to a target station is the one in which the target station conforms to its traffic specifications while all other stations saturate their channel. If under these conditions the requirements of the target station are met, this means that our configuration is effective in providing the station with the desired service guarantees.

The service guarantees of an application are expressed based on the *end-to-end delay*, defined as the time elapsed between the generation of a packet and its transmission in the WLAN; note that this includes the *queuing time* at the station's transmission queue. We consider that the delay of a lost packet is infinite; applications requirements on packet losses are thus captured by this metric.

The remaining challenge is to find the saturation throughput that a station needs to request in order to see its service requirements met. Note that the scheme proposed below for this purpose does not aim to serve for all possible situations and it is up to a station to issue its saturation throughput requests based on any other criterion. However, we believe that the scheme proposed hereafter does cover the most common applications.

In our scheme, we express the service requirements of an application based on two parameters: P and D. Specifically, we consider that the requirements of an application are met as long as its packets suffer an *end-to-end delay* smaller than D with probability P, where P and D are values dependent on the application's nature. We measure D in T_i units, where T_i corresponds therefore to the average interarrival time between two consecutive packets (defined as l_i/r_i, where r_i is the station's average sending rate).

The rationale for measuring D in T_i units is the following. Interarrival times of an application are typically chosen based on the application's delay requirements. If delay requirements are stringent, the application cannot wait until more data is generated before sending a packet, and therefore sends its packet using a shorter interarrival time. Following this, it is reasonable to take the station's interarrival time as the benchmark against which to measure the end-to-end delay requirements, and aim at the end-to-end delay being a fraction of the interarrival time. For applications that go through inactive periods, such as ON/OFF sources, we consider the interarrival time of the *active periods*.

Let us define Δ such that $r_i^{sat} = (1+\Delta)r_i$. Then, Eq. (59) can be rewritten as

$$\sum_{j=0}^{R-1}(1-p_i)p_i^j j\frac{CW_i^{min}}{2}T_{slot} = \frac{l_i}{(1+\Delta)r_i} = \frac{T_i}{1+\Delta} \tag{69}$$

from which

$$CW_i^{min}T_{slot} = T_i f(p_i,\Delta) \tag{70}$$

Substituting this into Eq. (58)

$$P(d_{s,j} > D_s) = P\left(T_i f(T_c,\Delta) > D_s\right) \tag{71}$$

which yields

$$P(d_{s,j} > D_s/T_i) = f(T_c,\Delta) \tag{72}$$

Therefore, we have that the service time distribution measured in T_i units *depends only on T_c and Δ, and does not depend on application specific parameters, such as the average arrival rate, r_i, or the packet length, l_i*. Consequently, the end-to-end delay distribution in T_i units depends only on T_c and Δ, in addition to the application's arrival process normalized into T_i units. Note that, with this normalization, we lose dependency of the arrival process on the application specific parameters (r_i and l_i), and the only dependency is on the arrival process nature (CBR, ON/OFF, Poisson).

The above result is very important. Indeed, following this result, we can obtain the required Δ values for a limited number of cases, and then these results can be used to determine the Δ of any application by mapping this application into one of the cases studied. Specifically, we can analyze the Δ values for a variety of common sources (CBR, ON/OFF, Poisson) and applications delay requirements (audio, video, data), and then, the Δ required by an application can be found simply by taking from this set of Δ values the one that corresponds to the same source nature and delay requirements as the application, independently of application's specific parameters such as the packet length or the average sending rate.

We now proceed to validate the above results via simulation, as well as to find the Δ values required for the most typical applications. For this purpose, we consider the following scenarios:

1. *Homogeneous scenario with 5 stations*. In this scenario, all the stations have requested the same saturation throughput and use packet lengths of 100 bytes.
2. *Homogeneous scenario with 10 stations*. The same as above but with 10 stations.
3. *Homogeneous scenario with 10 stations and 500 byte packet lengths*. The same as above but with all stations sending packets of 500 byte length, instead of 100 bytes.
4. *Homogeneous scenario with 10 stations and 1000 byte packet lengths*. The same as above but with 1000 byte length packets.
5. *Homogeneous scenario with 10 stations*. The same as scenario 1) but with 20 stations.
6. *Heterogeneous scenario with 10 stations and $w_i = \{2,1\}$*. In this scenario stations are divided in two groups of 5 stations each, with stations of the first group requesting twice as much saturation throughput as stations of the second group.

7. *Heterogeneous scenario with 10 stations and* $w_i = \{4, 1\}$. Same as above but with stations of the first group request four times as much saturation throughput as stations of the second group.

8. *Heterogeneous scenario with 20 stations and* $w_i = \{10, 5, 2, 1\}$. In this scenario stations are divided in four groups of 5 stations each, with stations of the first group requesting 10 times as much saturation throughput as stations of the fourth group, stations of the second group requesting 5 times and stations of the third group twice.

Following the rationale exposed at the beginning of this section, we consider the case in which all stations saturate their channel but one, which sends at a rate $r_i = r_i^{sat}/(1 + \Delta)$. Our goal is to find the Δ value required for different applications and arrival processes. We consider the following common arrival processes: CBR, ON/OFF and Poisson. Results for these three cases are given, respectively, in Figures 8, 9 and 10. In the three graphs, 95% percentile values of the end-to-end delay are given (in T_i units).

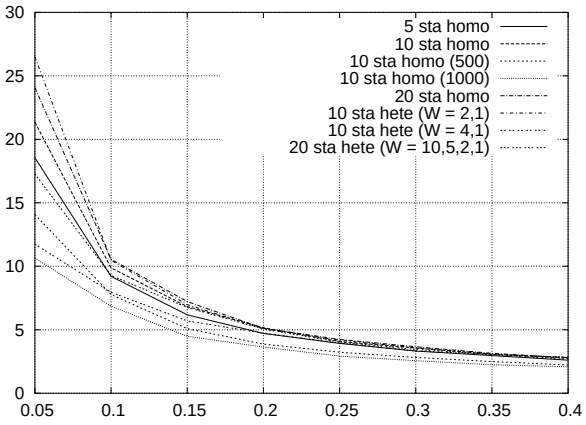

Figure 8. CBR traffic.

Simulation results confirm our key assumption that the delay distribution is independent of the number of stations and their saturation throughput, and depends only on the average collision length. Indeed, the 95 percentile delay in all three graphs is almost the same for all the scenarios but the ones with different packet length, except for some deviations for small Δ values. As predicted by the analysis, we observe that the smaller the packet length, the worse the delay performance. As mentioned before, we take the 100 byte packet length as worst case.

Based on the above results, we can derive the Δ values required for different applications based on the nature of their arrival process (CBR, Poisson, ON/OFF) and their delay requirements (audio, video, data). Specifically, we consider that audio applications require 95% of their packets to arrive with a delay 5 times the interarrival time, video applications require 95% of their packets to arrive with a delay 15 times the interarrival time, and data applications have no delay requirements. Note that, with these values, a voice application with an interarrival time of 10 ms suffers delays below 50 ms with 95% probability, and

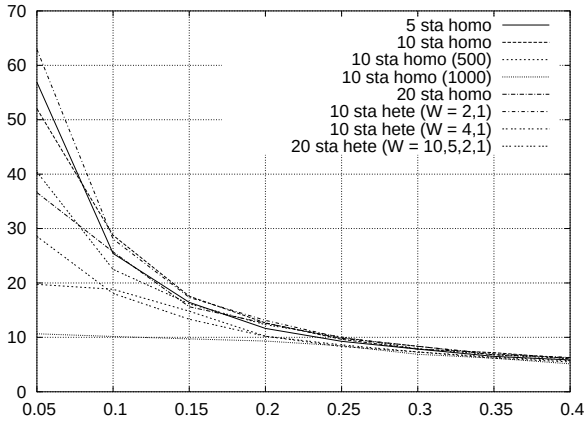

Figure 9. Poisson traffic.

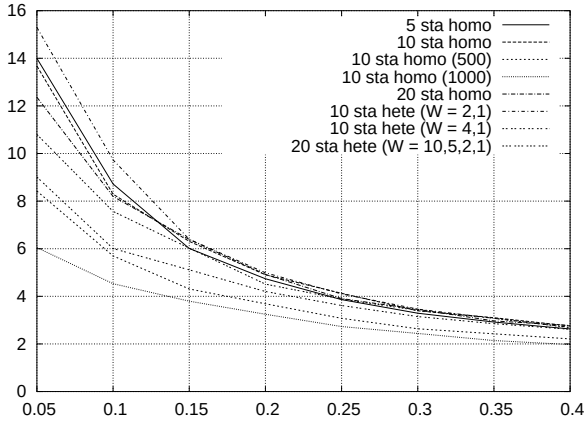

Figure 10. ON/OFF traffic.

a video application with an interarrival time of 20 ms suffers (with the same probability) delays below 300 ms.

The Δ values resulting from the above considerations are given in Table 2. These are the recommended Δ values for the various applications depending on traffic nature and arrival process. We note that, based on our finding that the delay distribution is independent of the scenario, it is very easy to extend the recommended Δ's to additional delay requirements and/or arrival processes, since no exhaustive study is required but it is enough to simulate one single scenario.

With the above, we have that a station with certain services requirements and a given arrival process can obtain its Δ from Table 2, and from this it can compute the required saturation throughput by considering its average sending rate and applying $r_i^{sat} = (1+\Delta)r_i$. Then, the optimal configuration can be obtained from all the saturation throughput requests; following the above arguments, this is the optimal configuration that ensures the desired service guarantees regardless of the behavior of the other stations.

Table 2. Recommended Δ values.

	CBR	Poisson	ON/OFF
audio	0.2	0.4	0.2
video	0.1	0.25	0.1
data	0	0	0

6. Conclusion

The EDCA mechanism of the IEEE 802.11e standard is based on a number of parameters whose configuration has been left open in the standard. Finding the optimal configuration of these parameters is crucial in order to make an efficient and satisfactory use of the mechanism. This article has addressed this issue with a number of contributions.

The first contribution of this article has been to identify a number of scenarios that require different configuration guidelines. Indeed, we have shown that, depending on the information available about the applications, as well as the friendly or unfriendly behavior of the users, the requirements and objectives for the EDCA parameters is different and as a result the appropriate configuration is also different in each case.

The second and main contribution of the article has been to propose concrete configuration guidelines for each of the scenarios identified. The various scenarios and the conclusions of the corresponding configuration guidelines proposed in this article are summarized next:

- *Service Guarantees in a friendly environment.* In this scenario stations are well-behaved and can be assumed to conform to the declared traffic specifications. Based on the traffic specifications and QoS requirements, in this article we have addressed the issue of finding the appropriate EDCA configuration. Specifically, we have focused on one use case, Voice over IP, and we have presented a method to compute the optimal configuration such that the given QoS requirements are met while admitting as many stations as possible.

- *Fair Resource Allocation.* In this case the objective is not to provide service guarantees but to fairly distribute the network resources among the various WLAN users. This configuration is adequate if the QoS requirements of the applications are not known but we still want to make sure that one aggressive station cannot utilize all the WLAN resources while the other stations starve. In addition, different priorities for the WLAN stations can be introduced.

- *Service Guarantees in an unfriendly environment.* While the goal of this case coincides with the first scenario, the assumptions on the users behavior is different. In particular, here we do not assume that users are well-behaved, and therefore the configuration must protect a given user from other malicious ones. As a result, the configuration will be different from the one of the first scenario. In the article, we have given configuration guidelines that can be used to satisfy the requirements of the majority of today's applications under this scenario.

References

[1] IEEE 802.11, *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, 2005.

[2] IEEE 802.11, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, IEEE Standard, IEEE, 1999.

[3] G.-R. Cantieni *et al*. Performance Analysis of Finite Load Sources in 802.11b Multirate Environments. *Computer Communications*. June 2005.

[4] A. Banchs and L. Vollero. Throughput Analysis and Optimal Configuration of 802.11e EDCA. *Computer Networks*. August 2006.

[5] P. Serrano *et al*. Performance Anomalies of nonoptimally configured WLANs. In *Proceedings of WCNC*. April 2006.

[6] P. Serrano, A. Banchs and J. F. Kukielka. Analysis and Configuration of IEEE 802.11e EDCA under Voice Traffic. submitted.

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

[8] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: the Multiple Node Case. *IEEE/ACM Transactions on Networking*. April 1994.

[9] L. Massoulie and J. Roberts. Bandwidth Sharing: objectives and algorithms. *IEEE/ACM Transactions on Networking*. June 2002.

[10] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *Journal of Selected Areas in Communications*. March 2000.

[11] T. Nandagopal, S. Lu and V. Bharghavan. A Unified Architecture for the Design and Evaluation of Wireless Fair Queuing Algorithms. In *Proceedings of ACM MOBICOM'99*. August 1999.

[12] S. Lu, V. Bharghavan and R. Srikant, Fair Scheduling in Wireless Packet Networks. In *Proceedings of ACM SIGCOMM'97*. August 1997.

[13] N. H. Vaidya, P. Bahl and S. Gupta. Distributed Fair Scheduling in Wireless LAN. In *Proceedings of ACM MOBICOM'00*. August 2000.

[14] A. Banchs and X. Pérez-Costa. Distributed Weighted Fair Queuing in 802.11 Wireless LAN. In *Proceedings of IEEE ICC'02*. April 2002.

[15] H. Wu, Y. Peng, K. Long, S. Cheng and J. Ma. Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement. In *Proceedings of IEEE INFOCOM 2002*. June 2002.

[16] A. Banchs and L. Vollero. A Delay Model for IEEE 802.11e EDCA. *IEEE Communications Letters*. June 2005.

[17] A. Banchs, P. Serrano and A. Azcorra. End-to-end Delay Analysis and Admission Control in 802.11 DCF WLANs. *Computer Communications*. April 2006.