MAC PARAMETR TUNING FOR BEST EFFORT TRAFFIC IN 802.11e CONTENTION-BASED NETWORKS

[(a)]Luca Scalia, [(b)]Ilenia Tinnirello and [(c)]Giuseppe Bianchi

[(a),(b)]Universita' degli Studi di Palermo, Dipartimento di Ingegneria Elettrica
Viale delle Scienze, 90128, Palermo, Italy
{luca.scalia,ilenia.tinnirello}@tti.unipa.it

[(c)]Universita' degli Studi di Roma - Tor Vergata, Dipartimento di Ingegneria Elettronica, Via del Politecnico, 1, 00133 Roma, Italy
giuseppe.bianchi@uniroma2.it

ABSTRACT[1]

This paper deals with the problem of best effort traffic delivery in 802.11e contention-based networks. Differently from most previous paper which focus on the support of quality of service (QoS) requirements, we study the tuning of the access parameters for the best effort traffic. The contribution of this paper is threefold. First, we discuss the coexistence between legacy DCF and EDCA stations, since, for guaranteeing the backward compatibility, the best effort service class defined in EDCA should correspond to the legacy DCF. We show what configurations of the access parameters are closer to the DCF protocol, by taking into account the slightly different backoff rules defined in EDCA. Second, we explore the optimizations that can be performed by dynamically tuning the access parameters, on a per-beacon basis, in the case of homogeneous best effort sources. We propose an effective algorithm able to maximize the system throughput, by adapting the minimum contention window to the network contention level. Finally, we analyze the amount of resources available for best effort traffic in presence of QoS traffic. We show that the dynamic adaptation of the minimum contention window as a function of the channel wasted times can be a valid solution to automatically regulate the best effort offered load in the network.

1. INTRODUCTION

The Distributed Coordination Function (DCF) of the IEEE 802.11 [1] MAC protocol in the last years has received significant research attention due to its robustness and popularity. The key factors of this success are to be searched in the easiness of implementation and the simplicity of the 802.11 protocol itself. However, the new market applications, basically oriented towards value-added delay-sensitive services such as VO IP or Video-On-Demand, have required further extensions to the 802.11 MAC layer in order to cope with heterogeneous traffic requirements.

Thus, the problem of quality of service and service differentiation in wireless LAN is a theme of current interest. According to the 802.11e EDCA proposal [2], which have been recently ratified, the service differentiation is provided in a completely distributed manner by giving probabilistically an higher number of channel access grants (*access priority*) to the stations involved in real-time applications. Basically, the access probability of the high priority classes is increased by setting lower contention windows (CWs) and inter-frame times (AIFSs). These differentiated settings allow to speed up the backoff expiration time of the high priority classes, thus resulting in an higher probability to win the contentions.

Great efforts have been done in literature [3, 4, 5], with both simulations and analytical models, in order to analyze the effects of the CW and AIFS differentiation and to identify, for a given traffic scenario, the most effective settings. These studies are focused on the satisfaction of the QoS requirements [6, 7, 8] of the high priority classes and, given that the CW and AIFS settings provide such requirements, they consider the performance of the best effort traffic class as a result. However, the support of best effort traffic is not a minor issue and also for this service class the tuning of the access parameters is very important. On one side, such a tuning should guarantee that the best effort class results somehow equivalent to the legacy DCF. On the other side, given that the QoS requirements for the high priority classes are maintained, it can be used for optimizing the best effort performance. Finally, the definition of a best effort traffic class should imply a feedback mechanism for regulating the best effort offered load, thus avoiding that some channel resources are destined to the best effort traffic before the priority traffic is fully satisfied. In this paper we neglect the problem of QoS support and focus our analysis on the best effort service class. In order to generalize our conclusions without any simplificative assumption (e.g. working on saturation conditions only), we base our investigations on simulations. We developed in C++ a very detailed MAC-oriented simulation tool, which follows faithfully the MAC state machine described in the standard. The simulator code is available [13] for guaranteeing an easy reproducibility of the results. The rest of the paper is organized as follow: in section 2 and 3 we briefly review the EDCA mechanisms in terms of protocol specifications and contention resolutions; in section IV we deeply investigate on the coexistence problem among

[1] This work has been partially supported by the Italian Ministry of Education, University and Research (MIUR) in the frame of the Italian project PRIN TWELVE.

AC	CWmin	CWmax	AIFSN	TXOP Limit		
				DS-CCK	OFDM	Other
AC_BK	aCWmin	aCWmax	7	0	0	0
AC_BE	aCWmin	aCWmax	3	0	0	0
AC_VI	aCWmin/2	aCWmin	2	6.016ms	3.008ms	0
AC_VO	aCWmin/4	aCWmin/2	2	3.264ms	1.504ms	0

Table I. EDCA default settings

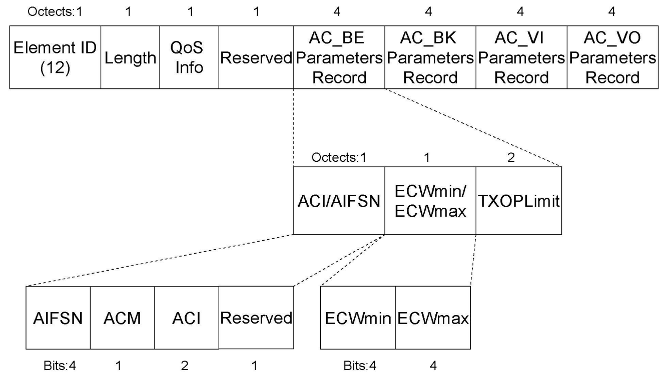

Figure 1. EDCA Parameter Set Element

EDCA and legacy DCF stations; in section V we propose a simple algorithm for optimizing the best effort performance and for giving elasticity to the best effort traffic; finally some conclusive remarks are drawn in section VI.

2. ENHANCED DISTRIBUTED CHANNEL ACCESS

We assume that the reader is familiar with the IEEE 802.11 DCF standard and the EDCA extensions. Our brief description is mainly focused on the discussion of the differences between DCF and EDCA and on the description of the mechanisms to dynamically adapt the MAC parameters. We consider an infrastructure network, in which an Access Point (AP) can centralize the MAC tuning functionalities, for both the downlink and the uplink traffic flows.

The IEEE 802.11 channel access protocol is long-term fair in terms of access probability. This means that all the stations have the same probability to win the contention and, in long term, to obtain the same amount of frame transmissions. The EDCA proposal of the IEEE 802.11e Task Group is devised to differentiate the channel access probability among different traffic sources. Packets arriving to the MAC (MSDUs) are mapped into four different access categories (ACs), which represent four different levels of service for the contention to the shared medium. Each AC contends to the medium with the same rules of standard DCF, i.e, wait until the channel is idle for a given amount of inter frame space, and then access/retry following exponential backoff rules. The access probability differentiation is provided by giving i) different Arbitration Inter-Frame Spaces AIFS, instead of the constant DIFS, and ii) different values for the minimum/maximum contention windows to be used for the backoff time extraction. Then, each AC is specified by the values AIFS[*AC*], CW_{min}[*AC*], and CW_{max}[*AC*]. The AIFS[*AC*] values differ each one for an integer number of backoff slots. In particular, AIFS[*AC*]=AIFSN[*AC*] . *aSlotTime* + *aSIFSTime*, where AIFSN[*AC*] is an integer greater than 1 for normal stations and greater than 0 for APs. Separate queues are maintained in each station for different ACs and each one behaves as a single enhanced DCF contending entity. When more than one AC of the same station expires its backoff counter, a virtual collision occurs and the highest priority packet among the internal colliding ones is selected for actual transmission on the radio channel. Once a station wins the contention and starts its transmission grant, EDCA also specifies new channel utilization operations based on the concept of transmission opportunity (TXOP), which represents a time interval in which the station is authorized to hold the channel.

Finally, as we detail in section IV-A, EDCA defines a backoff de-freezing rule slightly different from the DCF one. In standard DCF, the backoff counter is frozen during channel activity periods and resumed after the medium is sensed idle again for a DIFS interval. The resuming value is equal to the same value the station had at the starting of the busy channel period. Conversely, in EDCA, the frozen backoff counter is resumed one slot-time before the AIFS expiration. This means that, when the AIFS timer expires, the backoff counter results to be already decremented of one unit. Moreover, after the backoff counter expiration, EDCA stations have to wait for a further slot before transmitting.

Table I shows the default values of the channel access parameters defined in EDCA for each AC. In each beacon frame, the AP broadcasts the values of these parameters chosen for each AC. The per-class settings are specified in a special field of the beacon frame, called the EDCA Parameter Set Element. The most recent EDCA Parameter Set Element received by the stations are used to update the appropriate MAC values.

The detailed format of the field, which is 20 bytes long, is presented in Fig. 1. It includes a *QoS Info* field, which is used to count the updates in order to easily identify whether the EDCA parameters have changed, and four different record fields, corresponding to four different ACs. Each AC record results in turns divided into three sub-fields: the ACI/AIFSN, the ECW_{min}/ECW_{max} and $TXOP_{Limit}$ subfields.

The first subfield regards the AIFSN bits, which indicate the number of slots to add to the SIFS time (bits from 0 to 3), the enabling/disenabling of the admission control function (bit 4), the AC identification bits (bits 5 and 6), and a final reserved bit (bit 7). In the second subfield, bits from 0 to 3 and from 4 to 7 indicate, respectively, the CW_{min} and CW_{max} values using exponential notation with base 2. In particular, $CW_{min}=2^{ECW_{min}}-1$ and $CW_{max}=2^{ECW_{max}}-1$, so that the minimum encoded vales is 0 and the maximum vales is 32767.

Finally, the last subfield represents an unsigned integer corresponding to the TXOP value in units of 32 µs. The AC settings can be dynamically adapted according to the network conditions. Obviously, the smaller the AIFSN[*AC*], the CW_{min}[*AC*], and the CW_{max}[*AC*], the higher is the probability to win the contention with the other ACs. Note that these settings

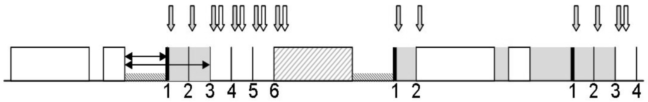

Figure 2. Channel access process: an example

only refer to the uplink traffic, while for the downlink one the AP can use arbitrary MAC values.

3. CHANNEL ACCESS OPERATIONS

In this section we describe the EDCA prioritization mechanisms in terms of low-level channel access operations. As shown in [9], whenever all the stations operate in saturation conditions, DCF channel accesses can be considered as slotted, since packet transmissions start only in discrete time instants.

These instants correspond to an integer number of backoff slots which follow the previous channel activity period plus the DIFS time. By looking only at the time instants in which a packet transmissions can be originated, the granted channel resources can be represented in terms of a sequence of idle slots, corresponding to the backoff slots in which no station accesses the channel, and busy slots, corresponding to the time interval required for the packet transmission (which includes the corresponding acknowledge in case of success) plus the DIFS. Given a channel slot, the DCF fairness property implies that each station has the same probability to start a transmission and to experience a success.

The same slotted channel can be assumed for describing the channel access operations in EDCA. The only difference is that the time instants in which the packet transmissions can be originated, which delimit the channel slots, now depend on the minimum AIFS employed by the contending traffic classes. Moreover, because of the different AIFS values, some slots can be accessed by a subset only of the competing traffic classes.

Fig. 2 shows an example of slotted EDCA channel. The discrete time instants in which the channel can be granted are indicated by some arrows and numbered according to the time elapsed by the last channel activity. A transmission originated after the minimum AIFS employed in the network belongs to the transmission slot 0, while a transmission originated after x idle backoff slots belongs to the transmission slot x. Each arrow represents the probability that a station belonging to a given priority class transmits on the channel. Only two classes are considered. Since each class employs a different AIFS value, (in the example, the difference among the two values is equal to two backoff slots) some slots can be accessed by one class only. We define these slots, which are shaded and pointed by a single arrow, as *protected*. Note that protected slots occur after each busy slot, and then the percentage of protected slots grows as the network congestion increases. At the end of each channel access, the stations contend for acquiring the right of the next transmission grant. The contention is based on the comparison of the backoff counter values of each contending station, since the station with the lowest backoff expiration time acquires the right to initiate the next transmission.

The backoff expiration time does not depend only on the backoff counter value, but also on the specific AIFS setting, since the resumes of the backoff counters after each channel activity are not synchronous among the stations. Whenever two or more stations access the channel simultaneously, a collision occurs. If n is the total number of competing stations, b_k is the backoff counter for station k after the last transmission, and δ_k is the number of slots required for the backoff resuming (i.e. $\delta_k = AIFSN_k - 1$[2]) for the same station k, the number of idle slots in each contention results:

$$\min_{k=1\ldots n} (b_k + \delta_k) - \min_{k=1\ldots n} AIFSN_k + 1$$

In the previous computation, we included the minimum AIFS employed in the network in the busy slot definition, and added a final empty backoff slot before transmission as stated in the standard (see [2], clause 9.9.1.3). The index for which the minimum is obtained represents the station which wins the contention. The contention is successful if such a minimum is unique. Note that the prioritization mechanisms introduced in EDCA basically work on differentiating the b_k extraction ranges and the backoff resuming offset δ_k. In the following, we refer to this slotted contention resolution modelling in order to describe EDCA operations and performance in terms of "internal" low-level protocol operations.

4. COEXISTENCE OF EDCA AC_BE AND LEGACY DCF STATIONS

In this section, we try to clarify the rationale of the AC_BE default settings suggested in the standard. Since EDCA is backward compatible with standard DCF, we expect that the best effort traffic category is somehow equivalent to the legacy DCF traffic. However, from table I, we see that the access parameters have some differences. Despite of the same minimum and maximum contention window value, the inter frame time value for the AC_BE is higher than a DIFS (we recall that a DIFS is equal to an AIFS with AIFSN=2).

Figure 3 shows throughput results in a scenario in which N legacy DCF stations share the channel with the same number N of EDCA stations. Curves with the same symbol refer to the same simulation. The shaded symbols represent the aggregate EDCA throughput, while the empty symbols represent the aggregate DCF throughput. EDCA stations have been configured with the standard DCF backoff parameters (CW_{min}=31 and CW_{max}=1023). The packet size has been fixed to 1500 bytes (Ethernet MTU) and the retransmission limit is set to 7 for all the stations. Control frames are transmitted at a basic rate equal to 1 Mbps, while the MPDU is transmitted at 11 Mbps. Whenever no otherwise specified, these settings have been maintained in all the simulations. We measured performance in saturation conditions. Although this assumption is not realistic for real-time application, it represents a very good representation of elastic data traffic and an interesting case study to derive the limit performance, i.e., the maximum amount of bandwidth that AC_BE can obtain sharing the channel with best effort DCF stations. From the figure we see that in the case AIFSN=2 the EDCA stations receive much more resources than DCF stations, while in the case AIFSN=3 they achieve performance close to that of legacy DCF stations.

[2] We subtract one slot because, as we specify in the next section, EDCA access categories resume the backoff counter one slot before the AIFS expiration. Thus, after the AIFS the backoff counter is already decremented of one unit.

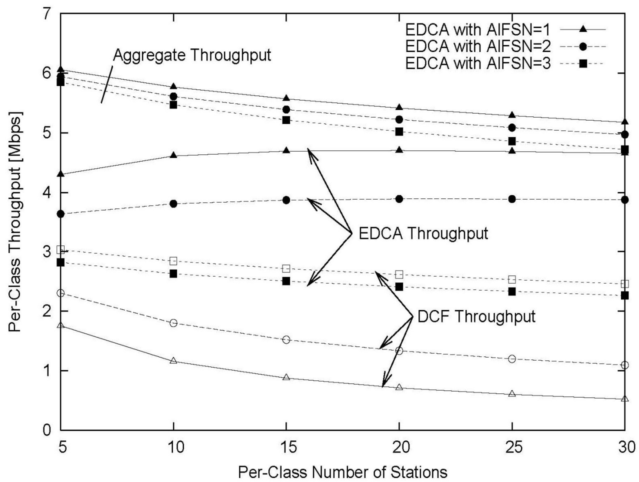

Figure 3. DCF versus EDCA throughput with AIFS differentiation

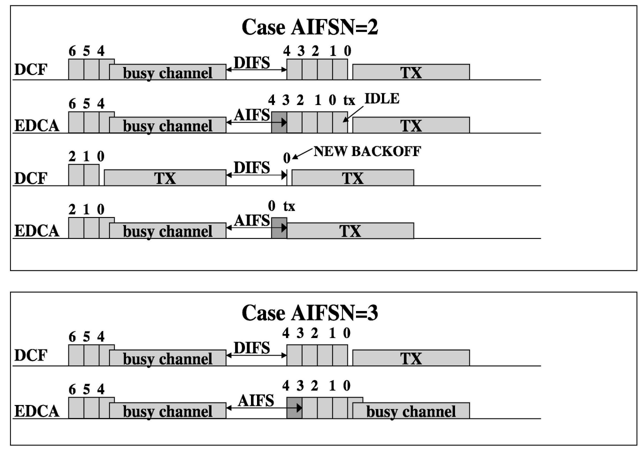

Figure 4. Backoff counter management in EDCA and DCF

This counter-intuitive result confirms that the default settings have been chosen in order to guarantee backward compatibility with the DCF. However, we need a detailed analysis of the channel access operations in DCF and EDCA to fully understand how this compatibility is provided.

4.1 Backoff Counter Decrement Rules

EDCA slightly differs from DCF in terms of how the backoff counter is managed (decremented, frozen, resumed). However, such apparently minor difference (which might perhaps appear as a technicality) indeed has some important consequences in terms of performance of EDCA access categories, especially when they compete with legacy DCF stations.

In standard DCF, the backoff counter is decremented at each idle slot-time, frozen during channel activity periods, and resumed after the medium is sensed idle again for a DIFS interval. This implies that a legacy DCF station, after a DIFS, resumes the backoff counter to the discrete value the station had at the instant of time the busy channel period started. An illustrative example is shown in figure 4. Here, a busy channel period (i.e. a transmission from one or more other stations) starts while the backoff counter of the considered DCF station is equal to 4. This value will be frozen during the busy channel period, and will be resumed, again to the value 4, only a DIFS after the end of the busy period. As a consequence, it will be decremented to the value 3 only a slot after the DIFS. In EDCA, the backoff counter is also decremented at every idle slot-time and frozen during channel activity periods. But it is resumed one slot-time before the AIFS expiration. This means that, when the AIFS timer elapses, the backoff counter will result to be already decremented of one unit.

Moreover, since a single MAC operation per-slot is permitted (backoff decrement or packet transmission, see [2], clause 9.9.1.3), when the counter decrements to 0, the station cannot transmit immediately, but it has to wait for a further backoff slot if the medium is idle, or for a further AIFS expiration if the medium is busy. In order to understand how these different rules affect the channel access probability, refer to the example shown in figure 4.

Let's first focus on the case AIFSN=2 (top figure), which correspond to using an AIFS equal to the DCF DIFS. In the example, two stations encounter a busy channel period with the same backoff counter value. However, at the end of the channel activity, we see that the DCF station resumes its counter to a value equal to the frozen value (4 in the example), while the EDCA station resumes and decrements its counter. In the case of a single busy channel period encountered during the backoff decrement process, this difference will be compensated by the fact that the EDCA station will have to wait for an extra slot, i.e., unlike the DCF station, it will transmit in the slot following the one in which the backoff counter is decremented to 0 (this is the case illustrated in the two top diagrams of figure 4).

However, in the presence of several busy channel periods encountered during the backoff decrement process (which is very likely to happen in the presence of several competing stations), the EDCA station will gain a backoff counter decrement advantage for every encountered busy period with respect to the DCF station. This implies that, for an AIFSN equal to a DIFS, the EDCA station has an advantage over DCF.

Indeed, figure 4 shows that there is also a second reason why, with same Inter Frame Space AIFSN=2, the EDCA gains priority over DCF stations.

In fact, as shown in the figure, an EDCA station may actually transmit in the slot immediately following a busy channel period (it is sufficient that the busy channel period was encountered while the backoff counter was equal to 0 - last case in the figure). Conversely, a DCF station cannot de-freeze a backoff counter value equal to zero. Thus, the only case in which it can access the slot immediately following a busy period is when it extracts a new backoff counter, after a successful transmission, exactly equal to 0.

In order to synchronize the EDCA and DCF backoff decrements, it appears appropriate to set the AIFSN value equal to 3. In this case, as we can see in figure 4, although the EDCA station has an higher inter frame space, after each busy slot the backoff evolution of the two target stations is the same. However, since the EDCA station has to wait for a further channel slot after the counter expiration, the access probabilities of the two stations does not coincide, since, for a given extraction, the EDCA station has always to wait for a slot more than the DCF station. However, this results in just a slightly higher access probability for the DCF station (loosely speaking, the EDCA station resembles the

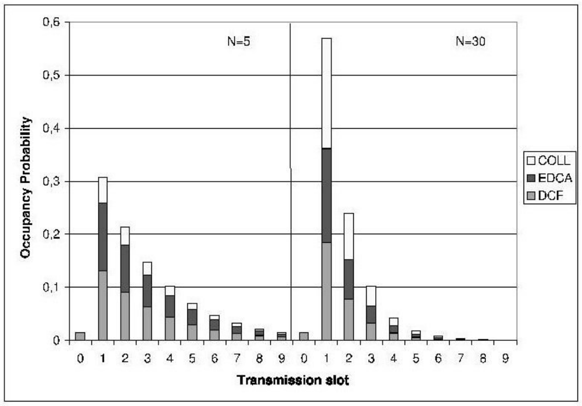

Figure 5. Per-slot occupancy probability: AC_BE vs. DCF

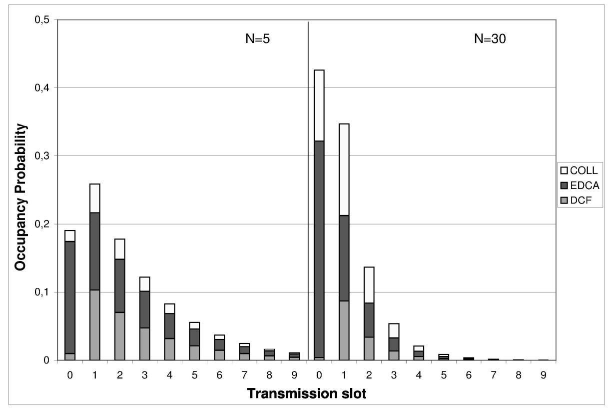

Figure 6. Per-slot occupancy probability: AIFS differentiation

operation of a DCF station with a CW_{min} value increased of just one unity).

4.2 Analysis of AC_BE Default Settings

The throughput results shown in the earlier figure 3 show that, for the same contention window parameters, EDCA throughput performance are similar to that of legacy stations when the AIFSN parameter is set to the value 3 (i.e. the EDCA AC_BE Access Category, see table I), rather than to a legacy DIFS (i.e. AIFSN=2). The discussion carried out in the previous section has provided a qualitative justification.

Goal of figure 5 is to back-up the previous qualitative explanation with quantitative results. To this purpose, we have numbered slots according to our previous description of the channel access operations. The slot immediately following a DIFS is indexed as slot 0. In the assumption of ideal channel conditions, a successful transmission occurs if, in a transmission slot, only one station transmits; otherwise a collision occurs.

Figure 5 reports the probability distribution that a transmission occurs at a given slot, for two different load scenarios: *N=5*, i.e. 5 EDCA stations competing with 5 DCF stations, and *N=30*. Only the first 10 slots are plotted, since most transmissions are originated after very few idle backoff slots. In addition, the figure further details in different colors the probability that a transmission occurring at a given slot results in a collision, in a success for an EDCA station, or in a success for a DCF station.

Figure 5 shows that DCF stations are the only ones that can transmit in the slot immediately following the last busy period.

Also, it confirms that a transmission in slot 0 is always successful (as it is originated by a station that has just terminated a successful transmission). Indeed, a transmission in the slot immediately following a busy period is a rare event, since it requires that the station that has just experienced a successful transmission extracts a new backoff counter exactly equal to 0. Thus, the slot 0 is a "protected slot" for the DCF stations, but it is rarely[3] granted. The figure also shows that, in the slots with index greater than 0, DCF and EDCA stations experience almost the same success probability, with a negligible advantage for DCF. For example, inthe case *N=5* a DCF success occurs, almost constantly through the various slot indexes, in about the 42.5% of the cases versus the 41% of EDCA, while for *N=30* these numbers reduce to, respectively, about 32.5% and 31.3% due to the increased probability of collision.The fundamental conclusion is that *by using AIFSN=3, an EDCA station can be set to approximately operate as a legacy DCF station*. With reference to the proposed EDCA parameter settings reported in table I, we thus conclude that an EDCA station belonging to the Access Category AC_BE will experience similar performance than a legacy DCF station. The above quantitative analysis also justifies why DCF shows a slightly superior throughput performance over EDCA AC_BE, as found in the figure 3 under the case of AIFSN=3.

4.3 AIFSN=2 and Legacy DCF Stations

As shown in table I, AIFSN=2 is the minimal setting allowed for an EDCA station. The rationale is that both AIFSN=0 and AIFSN=1 are already reserved in the 802.11 standard for, respectively, the Short Inter Frame Space (SIFS), and for the Point Coordination Function InterFrame Space (PIFS). However, as discussed above, the different mechanism employed in EDCA for decrementing the backoff counter suggests that, by using AIFSN=2 (i.e. AIFS=DIFS), an EDCA station is nevertheless expected to gain priority over a legacy DCF station. This was indeed shown in the previous figure 3, and is strikingly confirmed by figure 6, which, similarly to figure 5, reports the probability distribution that a transmission occurs at a given slot for the scenario of *N* DCF stations competing with N EDCA stations configured with AIFSN=2 and standard contention window parameters (CW_{min}=31 and CW_{max}=1023).

Figure 6 shows, for two different load conditions (N=5 and N=30), how the channel slots are occupied by the contending stations.

[3] Quantification is easy: after a successful transmission, a DCF station transmits in the slot 0 only if it exactly extracts a backoff counter equal to 0. This occurs with probability $1/(1+CW_{min}) \approx 3.1\%$. This conditional probability is consistent with the absolute probability value reported in figure 5 (about a half of this), since about half of the busy periods consist in a successful DCF transmission.

From the figure, we see that the slot 0, which, as shown before, is only rarely used by DCF stations, results almost protected for EDCA stations. Channel slots with index higher than 0 are instead accessed by both the classes with comparable probability. Figure 6 allows to draw a number of interesting considerations. First, the probability of collision in the protected slots (specifically, slot 0) is lower than in the other slots (e.g. for the case *N=5*, a collision in slot 0 occurs only in about 8.5% of the cases, versus an average of 17% in the remaining slots, and these numbers for the case *N=30* become 24.5% versus 38.5%), due to the reduced number of competing stations. Second, and most interesting, as the network load increases, the probability of accessing low-indexed slots significantly increases. The reason is that the number of slots between two consecutive busy channel periods significantly reduces in high load. But this implies that a large amount of accesses occurs in slots 0 (more than 40% in the case of N=30, see figure 6), and thus are almost exclusively dedicated to EDCA stations, with a definite gain in terms of service differentiation effectiveness (as earlier shown in figure 3). As a conclusion, *the usage of AIFSN=2 in EDCA (i.e. AIFS=DIFS) provides a significant priority of EDCA stations over legacy DCF stations*. This is an extremely important fact, as it allows to effectively deploy AIFS differentiation even when DCF stations share the same channel, and thus, apparently, there seems to be no room for AIFS levels intermediate between the Inter Frame spaces reserved by the standard (SIFS and PIFS), and the legacy DIFS.

5. TUNING OF AC_BE CONTENTION WINDOW

In this section, after having clarified the rationale of the AC_BE AIFSN setting, we analyze the exploitations of the contention window tuning for optimizing the performance of the best effort traffic. It is well known, that the contention window (and especially the minimum window value) is a very critical parameter, in saturation conditions, for the performance of contention-based networks [9], [10]. In contention-based networks, the distributed management of the channel access allows to avoid the overheads of any polling scheme, but introduces some inefficiencies because of the time wasted for the contention resolution. This time includes the idle time wasted for the backoff expiration and the transmission time wasted in the case of collisions. The settings of the minimum contention window have different effects on the two sources of channel waste. On one side, as the contention window increases the collision probability is reduced because of the increment of the backoff extraction ranges. On the other side, the idle time spent for the backoff expiration grows. Then, it exists an optimum contention window value which maximizes the throughput as a trade-off between the increment of the backoff times and reduction of the collision times. The optimum depends on the number of contending stations and on the time wasted during a collision [9]. Unfortunately, DCF does not allow to dynamically adapt the minimum contention window values to the network congestion level or to the transmission rate, thus performing, in general, far from the optimal conditions. However, this limitation has been removed in the case of EDCA, in which the access parameters can be tuned at least on a per-beacon basis. Assuming that the stations working in saturation conditions are the data best effort stations, we study some solutions and some effects about the AC_BE minimum contention window tuning.

5.1 Identification of the Optimal Working Conditions

It has been shown [9] that in DCF the optimal CW_{min} value should be set as a function of the number of competing stations and collision times.
For example, in [10] the optimal window is expressed in a closed form solution in the case of saturated contending stations. Although these results could be very easily adapted to the case of a single EDCA access category, the extension to general situations does not appear immediate. In fact, on one side, it is not possible to correctly estimate the number of per-class competing stations by simple monitoring the channel activity, such as in [11]. On the other side, it is not reasonable to assume that all the stations work in saturation conditions. Thus, in this paper, we propose a different approach for the optimal window tuning.
Instead of analyzing the network status to compute the optimal CW_{min} value a function of this status, we analyze the network status to correct the AC_BE behavior and identify the attainment of the optimal working conditions. It can be easily shown that the optimum is reached when the backoff time derivative is, in modulo, equal to the derivative of the collision times. However, in [12] it is observed that this condition is approximately reached whenever the backoff times are equal to the collision times, regardless the number of competing stations and on the status of the queues. Thus, since the collision times are decreasing monotonic functions of the contention windows and the backoff times are increasing monotonic functions, we can recognize whenever the current CW_{min} value is higher or lower than the optimal one, by simply comparing the two sources of wasted times.
Specifically, if the collision time is higher than the backoff time, it is necessary to further reduce the collision probability by increasing the CW_{min} parameter. Conversely, if the backoff time is higher than the collision time, it is necessary to avoid unuseful idle times by decreasing the CW_{min} values. In principle, this operation can be performed by the AP on a per-beacon basis and allows to incrementally correct the AC_BE contention window values, without requiring any network load estimator and any preliminarily assumption about the data sources. The capability to optimize the CW_{min} value by tracking the network dynamics, in the case of station activations/deactivations or in the case of not-saturated data stations, depends on the beacon interval value. However, the tracking is generally not critical because the performance do not degrade significantly around the optimal CW_{min} settings.

5.2 Example of AC_BE Dynamic Contention Window Corrections

In this section, we propose a very simple correction algorithm for the AC_BE minimum contention window.
We choose to define the algorithm as much simple as possible, in order to show the robustness of the optimization criteria. During each beacon interval i, the AP counts the overall time B_i spent in backoffs and the overall time C_i spent in collisions. Then, it updates the AC_BE CW_{min} value as follows:

$$CW_{min}(i) = CW_{min}(i-1) \cdot 2 \qquad C_i > B_i$$
$$CW_{min}(i) = CW_{min}(i-1)/2 \qquad C_i \le B_i$$

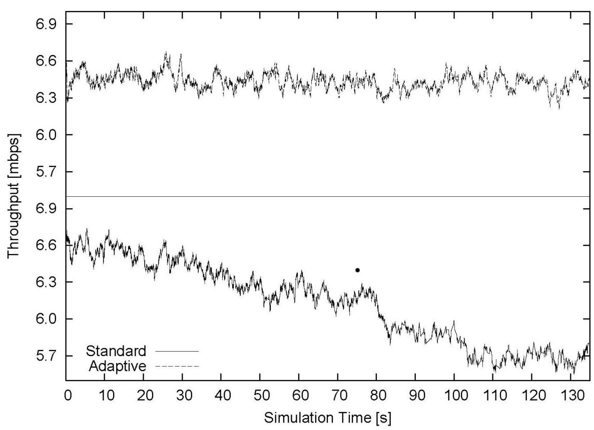

Figure 7. Overall throughput in the case of standard and adaptive protocol in dynamic scenario

	5	10	20	30	40
Standard	6.53	6.24	5.80	5.50	5.24
Adaptive	6.52	6.47	6.45	6.43	6.44

Table II. Default vs. Adaptive CW_{min} setting in static scenario

The new contention window value is finally broadcasted through the EDCA parameter set field. No filtering operation on the channel wasted times and no hysteresis for the contention window updates are considered. Table II shows the performance improvement achievable with the proposed algorithm in a static network scenario. We compare the throughput performance in the case of fixed CW_{min} and in the case of automatic CW_{min} updates. The fixed CW_{min} is set to 32, which represents the default AC_BE value. We assume to be in saturation conditions, with the number of stations taken as a simulation parameter. From the table, it is evident that the network throughput is poorly sensitive to the network load in the case of automatic adaptive CW_{min} setting. In fact, for each load condition, the algorithm is able to obtain almost the same channel utilization, by making a trade-off between the times wasted in collisions and in backoffs. In order to test the algorithm behaviour in dynamic load conditions, in Fig. 7 we show the throughput temporal variations in a scenario in which some stations activate asynchronously during the simulation time. We start the simulation with a fixed number of 5 AC_BE stations, and subsequently activate new AC_BE stations. We consider a single station entry at time instants 20, 30, 40, 50 and 60 seconds; then, two groups of 5 stations simultaneously join the network at the time instants 80 and 100 seconds. The entire simulation lasts 130 seconds. The figure shows how this algorithm is able to immediately adapt to load changes, thus providing an aggregated throughput which results almost constant. Conversely, for the standard fixed CW_{min} value, the throughput significantly degrades as the number of contenting stations grows. In Fig. 8 we compare the channel wasted times in the case of standard protocol (lower trace), with CW_{min} set to 32, and in the case of adaptive CW_{min} tuning (upper trace). In order to improve the figure readability, the collision times and the backoff times are plotted t intervals of 5

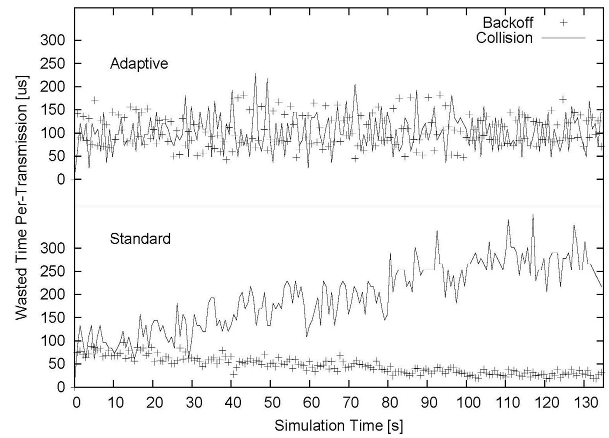

Figure 8. Percent (%) of Backoff expirations and collision overheads, for each successful trx vs. AC_BE CW*min* settings, for different N values

beacons. From the figure we see that, in the case of fixed contention window setting, as the number of station increases (i.e., as the simulation time advances) the collisions waste more and more resources, while the backoff times are slightly reduced.Indeed, our adaptive algorithm, allows to equalize these wasted times.Thus, from Fig. 7 and Fig. 8 we notice that this operation really corresponds to the maximization of the overall network throughput.

5.3 Elastic Behaviour of AC_BE in Presence of Priority Traffic

The dynamic adaptation of the CW_{min} parameter of the best effort class has other important consequences in the case of coexistence with priority traffic classes. In fact, the setting of the AC_BE CW_{min} as a function of the channel resource wastes allows to introduce a reactive mechanism for regulating the AC_BE access probability, able to give elasticity to the best effort traffic. The elasticity, i.e. the capability of data applications to adapt to the available bandwidth, is a desirable property for the best effort traffic.

This property implies that in presence of priority traffic, the best effort traffic should access only the radio resources which are not wasted for the priority traffic delivery. However, the fixed settings of the access parameters do not provide such adaptability, since a minimum amount of resources is granted even in presence of saturated priority traffic. Fig. 9 shows the resource repartition between the AC_BE and the AC_VI, in a scenario in which an equal number of 10 stations contend for the channel access. We assume that the best effort stations are permanently in the contending state, i.e. have the transmission queues never empty. The source rate of the AC_VI stations is equal for all the stations and is considered as a simulation parameter. Each bar of the figure corresponds to a different source rate and is labelled in terms of aggregated AC_VI offered load. The y-axis represents the overall network throughput and, in different colors, the throughput repartition between the two access categories AC_BE and AC_VI. In the left part of the figure, we assume that the CW_{min} parameters

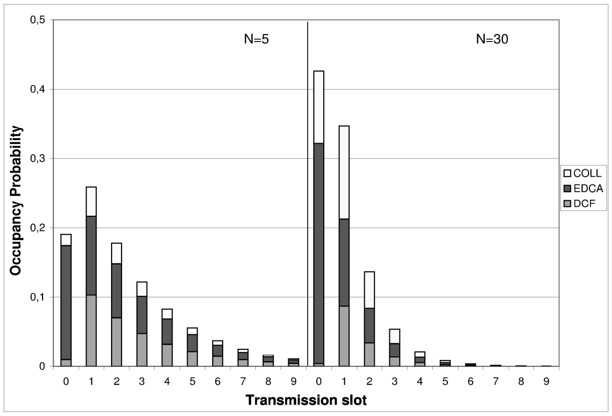

Figure 9. Throughput vs. offered priority traffic in the case of standard and adaptive CW*min* update

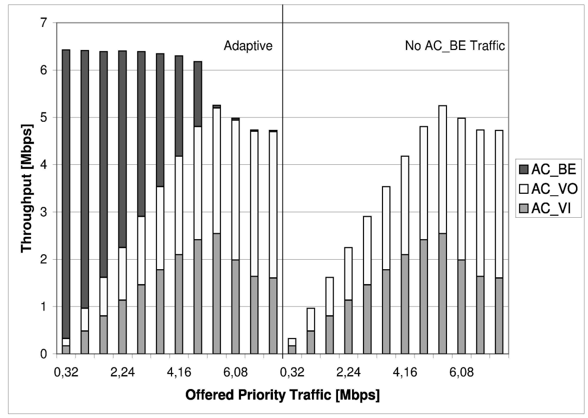

Figure 10. Throughput vs. offered priority traffic in presence and in absence of best effort traffic

of both the access categories are equal to the default standard value. In these conditions, we see that the AC_VI traffic saturates around 4 Mbps. For higher offered loads, the resources allocated to the AC_VI are lower than the requested ones and a fixed amount of bandwidth (about 1.8 Mbps) is granted to the AC_BE traffic. In the right part of the figure, we assume that the AC_VI CW_{min} is fixed to the default value, and the AC_BE CW_{min} changes adaptively according to our proposed algorithm devised to equalize backoff and collision times. Two interesting remarks are evident from the figure: first, in this case the AC_VI throughput saturates around 5.3 Mbps, which is about 30% more than in the previous case; second, the AC_BE throughput is approximately equal to the portion of bandwidth available to reach a total throughput of 6.4Mbps before the AC_VI saturation, and equal to 0 after the AC_VI saturation. Note that whenever the AC_VI traffic saturates, the aggregated throughput in the network is lower than the maximum achievable one (i.e. 6.4Mbps), since for this class we do not optimize the minimum contention window settings. In order to better understand the effects of the dynamic CW_{min} adaptation for the AC_BE traffic, Fig. 10 plots the throughput repartition between priority stations in presence and in absence of best effort traffic. As priority stations, we considered 5 stations employing the default AC_VI access parameters and 5 stations employing the AC_VO access parameters. All the priority stations have the same source rate which is represented in terms of aggregated rate in the x-axis. In the leftmost plot of the figure we observe the throughput repartition in presence of 10 AC_BE stations with adaptive CW_{min} settings. Again, until the priority sources are saturated, the best effort traffic receives an amount of resources which allow to maintain a fixed overall throughput equal to 6.4Mbps. No differentiation is evident between the AC_VI and the AC_VO stations. The priority traffic saturates whenever the offered load is greater than the perceived throughput. In this case, the best effort traffic does not receive any resource and the priority throughput starts to be differentiated according to the ratio between the AC_VI and AC_VO CW_{min} values. Specifically, the AC_VO throughput is almost twice the AC_VI one, since the default value of the AC_VO and AC_VI CW_{min} is, respectively, equal to 8 and 16. In the rightmost plot of the figure we observe the throughput repartition among the priority stations in absence of best effort traffic. We see that the figure is almost identical to the leftmost plot. Thus, we draw a very important conclusion: the best effort traffic does not subtract esources to priority traffic whenever the adaptive CW_{min} settings are employed. In fact, the CW_{min} adaptation tries to equalize the backoff and collision times experienced on the channel. Before the priority traffic saturation, the channel wastes are mainly due to the best effort traffic and the adaptation is able to force such equalization. As the priority traffic gets higher, more and collisions are due to the priority traffic. The CW_{min} regulation algorithm tries to reduce the collisions by increasing the CW_{min} value. However, since the adaptation only regards the AC_BE stations, the CW_{min} corrections do not have any influence on the channel wastes. Given that the corrections are ineffective, the algorithm react by increasing again the AC_BE CW_{min}, thus progressively reducing the AC_BE access probability down to 0.

6. CONCLUSIONS

This paper has tackled the issue of best effort traffic support in EDCA networks. In fact, the lack of minimum requirements or guarantees for the best effort services does not imply that the tuning of the AC_BE access parameters is not critical. Specifically, we faced several problems which arise in presence of best effort traffic. On one side, we analyzed the MAC settings which make the EDCA protocol backward compatible with the DCF one.

To this purpose, we have first detailed the basic operations of the EDCA differentiation mechanisms, in terms of contention resolutions. On the basis of such a description, we have shown that the backoff decrement rules adopted in EDCA has some not obvious consequences in the setting of the AIFS value which equalizes the DCF performance. In fact, by setting the AIFS value equal to the standard DIFS, the EDCA stations gain a significant amount of priority on the DCF stations. The resource repartition among EDCA and DCF stations is surprisingly equalized by including an extra backoff slot in the EDCA AIFS value. These considerations justify the default AIFS=3 setting suggested by the standard for the AC_BE. On the other side, we analyzed what kind of optimizations are possible in presence of uniform EDCA best effort sources.

Thanks to the EDCA capability to change the access parameters on a per-beacon basis, we proposed to dynamically adapt the AC_BE minimum contention window to the network contention level. In fact, in literature it has been shown that the CSMA/CA

performance are optimized whenever the minimum contention window is set as a function of the number of contending stations.
Since for the AC_BE the minimum contention window is not a prioritization parameter, we can easily exploit the EDCA tuning capability for optimization purposes.
We have shown that it is possible make automatic the contention window setting by simply monitoring the channel activity, without requiring any a priori traffic model or any network load estimator. This automatic setting is also effective in presence of QoS traffic, since it introduces a feedback mechanism which regulates the best effort offered load.
The feedback is able to make the best effort traffic elastic, i.e. to regulate the offered best effort traffic according to the available resources, given that the QoS requirements are satisfied.

REFERENCES

[1] IEEE 802.11 WG, IEEE Std 802.11, 1999 edition. International 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, 1999.

[2] IEEE 802.11 WG, IEEE Std 802.11e, Amendment to STANDARD for Information Technology - Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements, October 2005.

[3] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11", Proc. of IEEE INFOCOM 01, Anchorage, AK, USA, April 2001, pp. 209-218.

[4] S. Choi, J. del Prado, S. Mangold, and S. Shankar, "IEEE 802.11e Contention-Based Channel Access (EDCF) Performance Evaluation", Proc. of IEEE ICC 03, Anchorage, AK, USA, May 2003, vol. 2, pp. 1151-1156.

[5] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service", Proc. of European Wireless 2002, Florence, Italy, February 2002, vol. 1, pp. 32-39.

[6] S. Choi, "IEEE 802.11e MAC-Level FEC Performance Evaluation and Enhancement", Proc. of IEEE GLOBECOM 02, Taipei, Taiwan, November 2002, vol. 21, pp. 782-786.

[7] H.L Truong, G.Vannuccini, "The IEEE 802.11e MAC for Quality of Service in Wireless LANs", Proc. of SSGRR 2003, L'Aquila, Italy, January 2003.

[8] Y. Xiao, H. Li, and S. Choi, "Protection and Guarantee for Voice and Video Traffic in IEEE 802.11e Wireless LANs", Proc. of IEEE INFOCOM 04, Hong Kong, China, March 2004, vol. 3, pp. 2152-2162.

[9] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function", IEEE Journal on Selected Area and Communications, March 2000, vol.18, no.3, pp. 535-47.

[10] L. Bononi, M. Conti, E. Gregori, "Runtime Optmizations of IEEE 802.11 Wireless LANs Performance", IEEE Trans. on Parallel and Distributed Systems, January 2004, vol.15, no.1, pp.66-80.

[11] G. Bianchi, I. Tinnirello, "Kalman Filter Estimation of the number of competing terminals in IEEE 802.11 networks", Proc. of IEEE INFOCOM 03, 30 March - 3 April 2003, vol.2, pp. 844-852.

[12] R.G. Gallagher, "A perspective on Multiaccess Channels", *IEEE Trans. on Information Theory*, March 1985, vol.31, no.2, pp. 124-142.

[13] The TWELVE Project, available at [twelve.unitm.it/tools-it.html]