Configuration of IEEE 802.11e EDCA for Voice and Data traffic: An Experimental Study

Pablo SERRANO[1], Albert BANCHS[1], José Félix KUKIELKA[1], Giuseppe D'AGOSTINO[1], Seán MURPHY[2]
[1]*University Carlos III de Madrid, Avda. Universidad 30, Leganes, Madrid, 28911, SPAIN*
Tel: +34 91624 6236, Fax: +34 916248803,
Email:{pablo,banchs,kukielka,gdagostino}@it.uc3m.es
[2]*University College Dubline, Ireland*
Tel: +countrycode localcode number, Fax: + countrycode localcode number, Email:

Abstract: In this article we conduct an experimental study of the Enhanced Distributed Channel Access (EDCA) mechanism of the IEEE 802.11e standard with a testbed consisting of 1 Access Point and 15 WLAN stations. The focus of our study is the proposal of guidelines for the configuration of the open parameters in the EDCA mechanism. Specifically, we aim at supporting two widely deployed applications nowadays: *voice* and *data* traffic. To our knowledge, this is the first attempt to propose concrete configuration guidelines for the support of these two application types. From the comparison of our proposed configuration against the one recommended by the standard, we show that our guidelines outperform the standard's by 20% to 40%, depending on the number of data stations present in the WLAN.

Keywords: Wireless LAN, 802.11e.

1. Introduction

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 DCF mechanism of the IEEE 802.11 standard. Recently, the IEEE 802 Working Group has approved a new standard called 802.11e [1]. The Enhanced Distributed Channel Access (EDCA) mechanism of 802.11e extends the basic 802.11 DCF algorithm with Quality of Service capabilities. This new mechanism is based on a number of open parameters whose configuration is yet an unsolved research issue. Although the standard includes some recommendations for the parameters configuration, these are statically set and do no guarantee optimized performance.

The focus of the present article is on the configuration of the open parameters of the EDCA mechanism. To date, a considerable amount of work has been directed to study analytically and via simulation the performance of this mechanism as a function of the parameters configuration (see e.g. [2],[3],[4],[5]). However, it is well known that WLANs suffer from a number of non-ideal effects which have a non-negligible impact on performance and are not accounted for in such analytical and simulation studies [6].

A number of recent articles in the literature have studied the performance of EDCA experimentally. In [6] we analyzed experimentally the performance of EDCA under voice and data traffic with an early version of the testbed that we have used here (in [6] the testbed was limited to 3 stations while in this article we use up to 15 stations). In [7], the authors studied experimentally the impact and configuration of CW_{min} parameter, but their study was limited to only this EDCA parameter and they only considered data stations. In [8], an experimental analysis of EDCA under voice and data traffic was conducted, but the focus of the analysis was understanding the impact of the various parameters rather than studying their optimal configuration.

None of the above referenced works proposes concrete guidelines for the configuration of the EDCA parameters. Indeed, to the best of our knowledge, there is no work in the literature to date that addresses the problem of finding the optimal EDCA parameter configuration that satisfies a given set of throughput and delay requirements. In this article, we tackle this issue by proposing a number of configuration rules for EDCA under voice and data traffic. The objective of the proposed rules is to provide data stations with the maximum possible throughput while admitting as many voice stations as possible.

In this paper we focus on voice and data applications for 802.11b WLANs. We argue that voice and data are two of the most widely used applications, and that 802.11b is widely deployed, specially for VoIP phones. We note, however, that the same methodology as the one presented here can be used to obtain configuration rules for other application types as well as other physical layers like e.g. 802.11g.

2. IEEE 802.11e EDCA

We now briefly summarize the EDCA mechanism as defined in the 802.11e standard. EDCA controls the access to the wireless channel on the basis of the Channel Access Functions (CAF's). To transmit its frames, each CAF executes an independent backoff process which is regulated by a number of configurable parameters. For the configuration of these parameters, the standard groups the CAF's by Access Categories (AC's) and assigns the same configuration to all the CAF's of an AC. In this article, we have configured stations to run only one CAF and use indistinctly the terms CAF and station. The only two AC's that we consider are the ones corresponding to voice and data stations.

The operation of a station running EDCA is described as follows. When the station has a new frame to transmit, it monitors the channel activity. If the channel is idle for a period of time equal to the arbitration interframe space parameter of this station (AIFS), it transmits. Otherwise, if the channel is sensed busy, it continues to monitor the channel until it is measured idle for an AIFS time, and, at this point, the backoff process starts.

Upon starting the backoff process, the station computes a random value uniformly distributed in the range $(0,CW-1)$, and initializes its backoff time counter with this value. The CW value is called the contention window, and depends on the number of transmissions failed for the frame. At the first transmission attempt, CW is set equal to the minimum contention window parameter (CW_{min}). As long as the channel is sensed idle the backoff time counter is decremented once every slot time. When a transmission is detected on the channel, the backoff time counter is "frozen", and reactivated again after the channel is sensed idle for an AIFS time.

As soon as the backoff time counter reaches zero, the station transmits its frame in the next slot time. A collision occurs when two or more stations start transmission simultaneously. An acknowledgement (Ack) frame is used to notify the transmitting station that the frame has been successfully received. If the Ack is not received within a given timeout, the station assumes that the frame was not received and reschedules the transmission by reentering the backoff process. After each unsuccessful transmission CW_i is doubled, up to a maximum value given by the CW_{max} parameter. Once the backoff process is completed, CW is set again to CW_{min}.

When the station gains access to the channel, it is allowed to retain the right to access it for a duration equal to the transmission opportunity limit parameter ($TXOP_{limit}$). If this parameter is set to zero, the station is allowed to transmit only one packet upon accessing the channel.

As it can be seen from the above description of EDCA, the operation of this protocol depends on a number of open parameters (namely CW_{min}, CW_{max}, AIFS and $TXOP_{limit}$) whose configuration is yet an unresolved issue. The rest of the paper is devoted to finding the optimal configuration of these parameters in order to meet the requirements of voice and data stations.

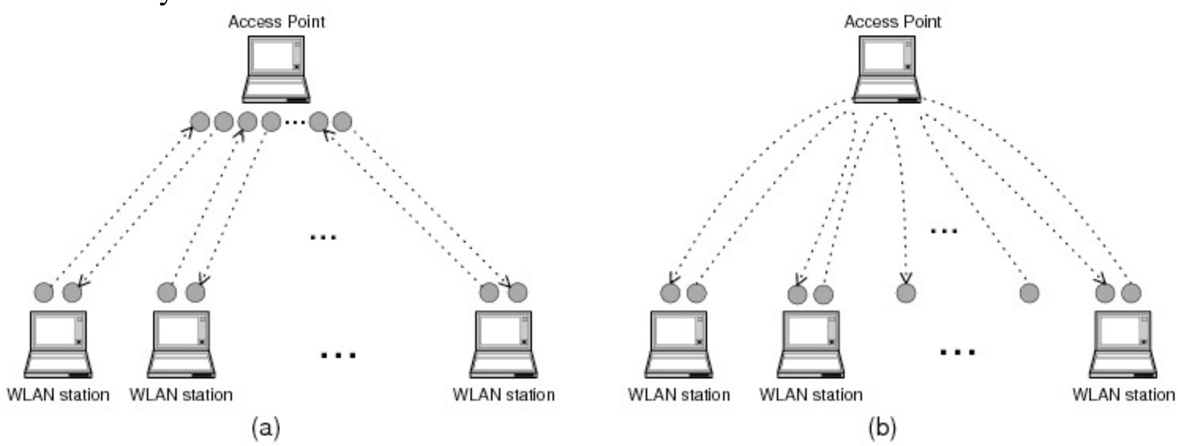
3. Experimental setup

To perform our experiments we built a testbed composed of 16 laptop and desktop PCs, one of them configured as the Access Point (AP) and the other 15 as WLAN stations running under the infrastructure mode. All PCs were equipped with Atheros based 802.11b wireless cards which fully implement the EDCA mechanism of the standard. We used the MADWIFI driver (Multiband Atheros Driver for WiFI), which provides full support for wireless adapters using Atheros chipsets on Linux platforms.

The configuration of the CW_{min}, CW_{max}, AIFS and $TXOP_{limit}$ parameters in the Atheros wireless cards is performed with some functions that the MADWIFI driver provides for this purpose. The configuration of the CW_{min} and CW_{max} parameters is set equal to 2^n and 2^m, respectively, where n and m are integer values provided to MADWIFI. Note that this restricts the possible values of these two parameters to powers of 2. The configuration of the AIFS parameter is set equal to SIFS+Aσ, where SIFS and σ are constants defined by the physical layer and A is an integer value in the range between 2 and 15 provided to MADWIFI. For simplicity, hereafter we indicate the value of the AIFS parameter by giving the corresponding value of A. Finally, the parameter $TXOP_{limit}$ is directly set to some integer value in the range between 0 and 65535 us.

For traffic generation, we used the *iperf*[1] tool, configured according to the desired traffic patterns in order to emulate voice and data applications. One-way delays were obtained from this tool by synchronizing all stations via NTP and measuring the time elapsed between the departure of a packet at the sending application and its arrival at the receiver. Note that the delay values obtained with this method correspond to the total one-way delay in the WLAN including the queuing delay at the sender.

One of our aims was to study the performance of a number of voice conversations running under the infrastructure mode. A natural way of emulating this with the above tool would be to run two unidirectional instances of *iperf* between each WLAN station and the AP (see Figure 1a). However, we observed that, with this configuration, the AP could not handle such a large number of *iperf* instances while preserving timing accuracy, and packets were accumulated at the AP yielding artificially long delays. Note that in a real situation this would not occur as the end-points of the voice conversations would be terminals beyond the AP and not the AP itself.



[1]http://dast.nlanr.net/Projects/Iperf/

Figure 1. Testbed setup

In order to avoid the above problem, we used the configuration shown in Figure 1b. With this configuration, each station executes one *iperf* instance for sending traffic via the AP to another station, and another *iperf* instance for receiving traffic, while the AP only forwards traffic. By building the chain structure shown in the figure, we have that the traffic sent through the WLAN is exactly the same as if each station was running a bidirectional voice conversation against the AP. Note that, with this setup, none of the PCs needs to run more than two *iperf* instances, which does not raise timing problems.

For each of the experiments presented in the paper, 5 runs of 60 seconds each were executed. The maximum, minimum and average values of the 5 runs are given for each case.

4. Configuration guidelines of EDCA for voice traffic

We now address the issue of finding the optimal EDCA configuration and associated admission control in order to meet the requirements of *voice* traffic, which is one one of the most widely deployed applications nowadays. We assume a standard G.711 voice codec that generates one 80 byte packet every 10 ms with no silence detection. Note that this is the most resource hungry possible behavior and therefore represents the *worst-case* conditions. The reason for this choice is that it ensures the generality of the resulting guidelines for all voice applications. Indeed, the configuration and admission control guidelines obtained under these assumptions will also satisfy the requirements of other voice applications that use more efficient voice codecs.

According to the ITU-T G.114 [9] recommendation, a 150 ms one-way delay with a 5% packet loss ensures good quality for voice traffic. Assuming that a VoIP flow may traverse up to two WLAN hops in addition to a wired core network, we divide the technologies traversed as follows. We consider that core network delay should not exceed 50 ms with 99% probability, while WLAN delay should be below 50 ms with 98% probability. Note that, in this way, we obtain a delay bound of 50+50+50=150 ms with probability 98%x99%x98%=95%[2]. Based on this, hereafter we measure the 98% percentile of the delay and impose the requirement that this value cannot exceed 50 ms[3].

Before addressing the issue of finding the EDCA configuration guidelines that best meet the above requirement, we first made the following experiment using the default DCF configuration. We set up a scenario with n_v voice calls and measured the delay in each direction for increasing n_v values ("uplink" for traffic sent by the STAs, and "downlink" for traffic sent by the AP). The results are shown in Figure 2. Given the requirement that delay cannot exceed 50 ms for 98% of the packets, from these results we have that with DCF we can admit as many as 8 simultaneous voice calls in the WLAN.

---

[2] Note that this bound corresponds to a worst case, as we impose a delay below 50 ms in each part in order to guarantee a total delay below 150 ms.

[3] Note that with this requirement we impose a limit not only on the average delay but also on its distribution.

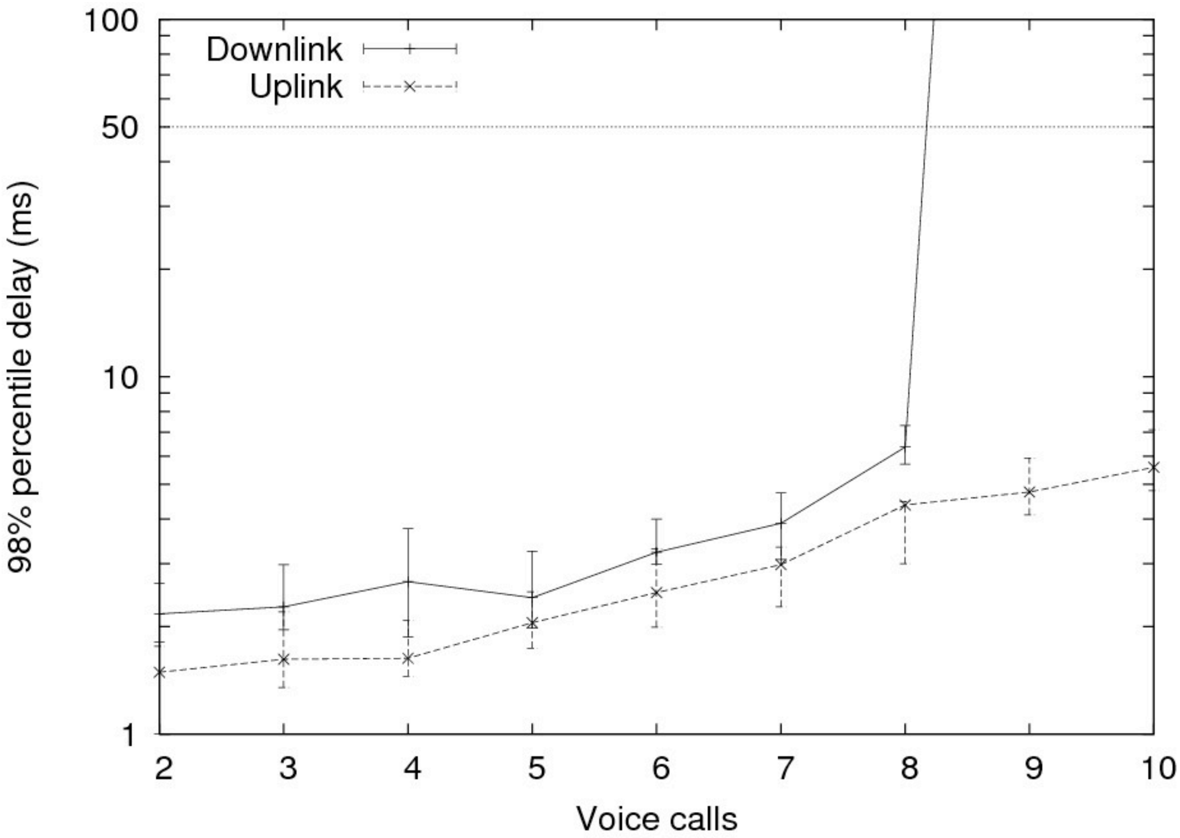


Figure 2. Maximum number of voice stations with DCF

The above experiment was run without using any of the EDCA parameters. Starting from the above experiment as benchmark, we now present a number of configuration rules (derived from a combination of reasoning and experiments) and see how many additional stations can be admitted with the proposed rules. We start with the configuration of the AIFS parameter, which should clearly be the smallest possible value for voice traffic as this maximizes delay performance:

Rule 1: $A_{voice} = 2$

With large TXOP values, a station with several packets in its transmission queue does not have to wait several backoff processes before transmitting them. Instead, all the packets can be transmitted back-to-back with a single backoff process, which reduces the queuing delays. Based on this rationale, we derive the second of our configuration guidelines, according to which the $TXOP_{limit}$ parameter of voice stations is set to the maximum possible value:

Rule 2: $TXOP_{limit\_voice} = 65535$ us

In order to assess the effectiveness of the above rule, we repeated the experiment of Figure 2 but with the new $TXOP_{limit}$ setting. The results showed that with this new configuration, two additional voice calls could be served while meeting the given quality criterion, i.e. a total of 10 calls can be admitted in this case.

Next, we address the configuration of the two remaining parameters, CW_{min} and CW_{max}. We start by reasoning that in the optimal configuration for voice traffic the CW_{max} parameter should be set equal to CW_{min} so that the CW is not doubled after each unsuccessful transmission. Indeed, if the CW was doubled after a collision, this would result in very large delays for those packets that suffer one or more collisions, and this would severely degrade voice performance. This yields the next of our configuration guidelines:

Rule 3: $CW_{max}(voice) = CW_{min}(voice)$

In order to find the optimal value of the remaining parameter (CW_{min}), we proceeded as follows. Given n_v voice calls, we performed an exhaustive search in the {CW_{min}(voice, AP), CW_{min}(voice, STA)} space. If the given delay criterion was met for some configuration, we repeated the experiment for n_v+1 voice calls, until no configuration provided the desired quality. From the above experiment, we obtained the maximum number of voice calls that can be admitted in the WLAN (n_v = 11). The performance of the different CW configurations for this n_v value in terms of average delay are given in Table 1. Results are given in ms, and those configurations whose delay exceeded 1 second are marked with "-". From the table, it can be seen that the optimal CW_{min} configuration that allows admitting the maximum number of voice calls while minimizing delay is of 32 for the AP and 64 for the stations. Note that, as the AP is sending n_v voice flows while stations are only sending one, a smaller CW_{min} is needed for the AP in order to optimally handle this larger traffic volume.

Based on the above results, the following two guidelines are derived. The first deals with the optimal CW_{min} configuration:

Rule 4: CW_{min}(voice, AP) = 32, CW_{min}(voice, STA) = 64

and the second one with the admission control algorithm:

Rule 5: Admit no more than 11 voice conversations

Table 1. Optimal CW configuration for voice traffic

	CW_{min}(voice, AP)					
CW_{min}(voice, STA)	4	8	16	32	64	128
4	-	-	-	-	-	-
8	287.55	-	-	-	-	-
16	105.79	102.96		-	-	-
32	33.61	45.87	36.3	69.63	-	-
64	23.4	29	25.08	22.56	-	-
128	28.08	29.53	29.61	26.94	-	-
256	39.76	44.97	47.77	43.48	42.67	-
512	69.18	68.54	76.43	73.45	-	-

5. Configuration guidelines of EDCA for data traffic

We now address the issue of finding the optimal configuration for data stations when both voice and data stations are present in the WLAN. Our goal in this case is to provide data stations with the largest possible throughput while ensuring that voice performance is not degraded. Following the same rationale as for voice traffic, we assume \emph{worst-case} conditions for data stations in order to ensure the generality of the proposed guidelines; specifically we assume that data stations are constantly backlogged and always have a 1500 byte packet ready for transmission.

We start by discussing the configuration of the $TXOP_{limit}$ parameter of data stations. Note that setting this parameter to a large value would harm the performance of voice stations as they would see their delay increased. To avoid this undesirable situation, we set the the $TXOP_{limit}$ parameter such that data stations can transmit only one packet when they access the channel, i.e.:

Rule 6: $TXOP_{limit}$(data) = 0

For data stations, since delay performance is not as critical as for voice stations, the CW can be doubled after each unsuccessful transmission. Indeed, it is well known that doubling the CW after a collision increases throughput performance. Based on this rationale, we use the same collision avoidance mechanism as the default DCF standard configuration of 802.11, where the CW is doubled up to 5 times until reaching the maximum value:

Rule 7: $CW_{max}(data) = CW_{min}(data)\ 2^5$

The two remaining parameters for the data stations are the CW_{min} and the AIFS. To study the configuration of these parameters, we performed the following experiment, with one data station and the maximum number of admissible voice stations obtained above. We first set the AIFS parameter of the data station to the minimum allowed value and tried increasing CW_{min} values until the delay bound for voice was met. Note that, in this way, we obtain the minimum possible CW_{min} value that can be assigned to the data station, which corresponds to the maximum throughput. We repeated the same experiment but setting the AIFS parameter to the maximum value.

The CW_{min} values that we obtained from the above experiments were 128 and 16, respectively. This gives us two possible {CW_{min}, AIFS} configurations for the data station that guarantee the delay criterion for voice traffic, namely {128,2} and {16,15}$. In order to find which of the two configurations is the most appropriate one, we performed the following experiment: we measured with both configurations the throughput obtained by the data station for an increasing number of voice calls (n_v). The results, given in Figure 3, show that the configuration that provides the data station with the greatest throughput is the one with the largest AIFS.

The above result is explained as follows. For $n_v = 11$ the channel is at the limit of its capacity and approximately the same throughput is obtained with both configurations. Under the maximum AIFS configuration, such a high load yields frequent transmissions in the channel, which forces the data station to freeze its backoff counter often thereby protecting voice traffic. As n_v decreases, there are less transmissions and the backoff counter is frozen less frequently, yielding thus a more aggressive behavior of the data station. In contrast, under the minimum AIFS configuration, protection is given by the CW_{min} parameter, which does not modify its behavior as n_v decreases.

The above reasoning shows that the maximum AIFS configuration behaves better as a result of better adapting the aggressiveness of data traffic to the number of voice stations present. This yields the following guideline for the configuration of the data station:

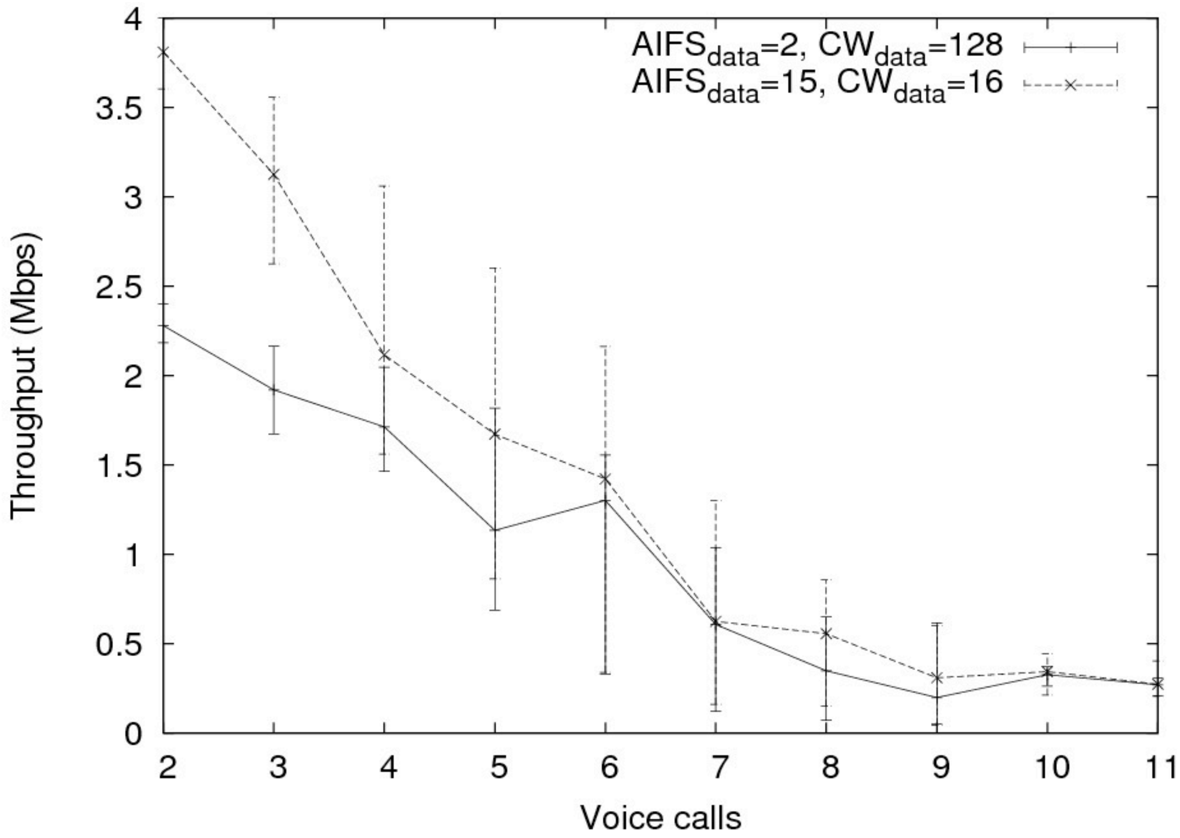


Figure 3. Throughput of data station vs. number of voice calls

Rule 8: A_{data} = 15 and CW_{min}(data)=16

The rule above gives the configuration for one data station. In order to find the optimal configuration for more than one data station, we argue as follows. The effect of one saturated station is roughly equivalent to the effect of n_d data stations with a CW_{min} value n_d times larger. Indeed, in the latter case we have n_d times more stations but each of them accessing the channel at a rate n_d times smaller. In order to validate this argument, we ran an experiment with the maximum number of voice calls and an increasing number n_d of data stations, each of them with a CW_{min} value n_d times larger than the one given by the above rule. The maximum of the downlink and uplink voice delay is shown in Figure 4. Results confirm our argument, as voice performance (both uplink and downlink) is preserved in all experiments. This yields our final rule:

Rule 9: For n_d data stations, $CW_{min}^{data}= n_d \cdot CW_{min}^{rule\,\,8}$}

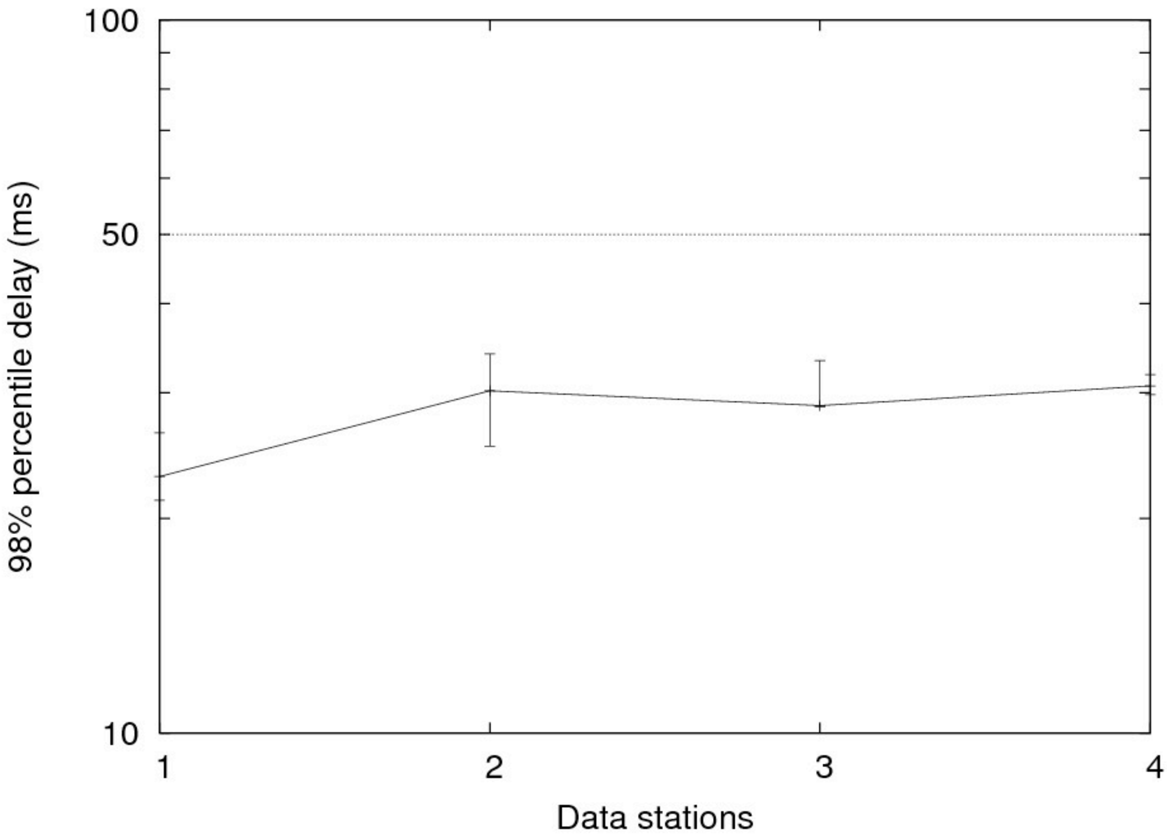


Figure 4. Voice delay with the configuration of Rule 9

6. Comparison against Standard Recommendation

In order to assess the effectiveness of the configuration guidelines proposed here for voice and data traffic, we compared them against the configuration recommended by the 802.11e standard [1]. Specifically, in Figure 5 we show the maximum of uplink and downlink delay for our configuration and the standard's as a function of the number of voice stations (n_v) when there are no data stations present in the WLAN.

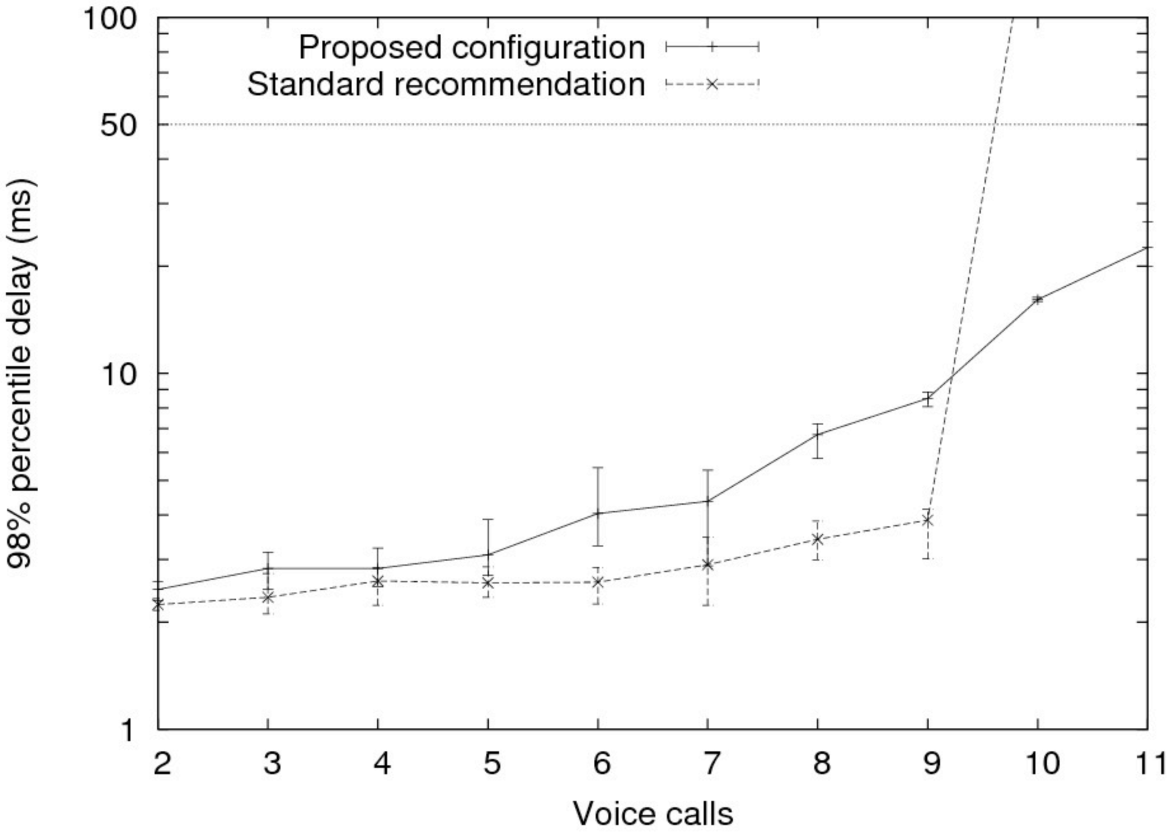


Figure 5. Comparison against Standard Recommendation

From the above results, it can be seen that our guidelines outperform the standard in terms of supported voice calls: with our configuration we can admit as many as 11 stations while preserving voice traffic performance, and with the standard recommended configuration only 9 stations can be admitted. The price that we have to pay for this is slightly larger delays for n_v<8; however, this does not harm voice performance as delays in all cases keep well below the given criterion.

We therefore conclude that with our rules we have a gain of about 20% in the number of voice stations that can be admitted without any significant loss in performance.

In order to understand how the above gain changes when data stations are present, we repeated the experiment of Figure 5 with 4 data stations. With this new setup, we saw the number of voice stations that could be admitted with our configuration was still equal to 11 (as it can bee seen from the results of Figure 4) while with the standard recommendations only 7 stations could be admitted in this case. We conclude that the gain obtained with our guidelines is higher as the number of data stations increase -specifically, a gain above 40% is achieved when 4 data stations are present.

7. Conclusions

In this article we have proposed a number of configuration guidelines for 802.11e EDCA in order to support two of the most widely used applications today, namely voice and data. To our knowledge, no previous work (other than the standard's recommendations) has attempted to provide concrete configuration rules for EDCA. Our experiments have shown that our guidelines outperform the standard's by 20% to 40%, depending on the number of data stations present.

The proposed configuration rules have been derived by combining reasoning and real-life experiments. As there are a large number of variables in our scenario (including four EDCA parameters plus the number of stations for each application type), reasoning is a necessary tool; otherwise, an unfeasibly large variable space would have been needed to be explored experimentally. Running real-life experiments is essential in order to guarantee the applicability of the results; indeed, admission control algorithms derived via simulations tend to admit more stations than they should as a a result of neglecting some non-ideal effects.

In [10] we conducted a simulation analysis of EDCA with similar configurations to the ones proposed here. According to those simulations, as many as 9 voice stations could be admitted with DCF, 11 when the $TXOP_{limit}$ parameter was set to a large value, and 12 with the CW parameters optimally set. The figures that we have obtained here experimentally under the same conditions are respectively 8, 10 and 11, i.e. exactly one station less for each case. These results confirm the above statement that simulations are not appropriate for designing admission control rules, although it is worthwhile observing that there is a clear correlation between simulation and experimental results.

References

[1] IEEE 802.11e, 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] D. P. Hole and F. A. Tobagi, Capacity of an IEEE 802.11b WLAN supporting VoIP, in Proc. of the IEEE International Conference on Communications, June 2004.

[3] A. Banchs and L. Vollero, Throughput Analysis and Optimal Configuration of 802.11e EDCA, Computer Networks, vol. 50, no. 11, pp. 1749--1768, August 2006.

[4] J. W. Robinson and T. S. Randhawa,, Saturation Throughput Analysis of IEEE 802.11e EDCF, IEEE Journal on Selected Areas in Communications, vol. 22, no. 5, pp. 917--928, June 2004.

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

[6] A. Banchs, A. Azcorra, C. Garcíia, and R. Cuevas, Applications and Challenges of the 802.11e EDCA Mechanism, IEEE Network, vol. 19, no. 4, pp. 52--58, July 2005.

[7] V. A. Siris and G. Stamatakis, Optimal CWmin Selection for Achieving Proportional Fairness in Multi-Rate 802.11e WLANs: Test-bed Implementation and Evaluation, Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization, September 2006.

[8] I. Dangerfield, D. Malone, and D. J. Leith, Experimental Evaluation of 802.11e EDCA for Enhanced Voice over WLAN Performance, Proceedings of WiNMee, April 2006.

[9] International Telecommunication~Union (ITU), Transmission systems and media, general recommendation of the transmission quality for an entire international telephone connection; one-way transmission time. Recommendation G.114, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, March 1993.

[10] A. Banchs, G. Iannello, P. Serrano, and L. Vollero, A CAPWAP Architecture for Dynamic Configuration of QoS MAC Parameters, Proc. of the First Workshop on multiMedia Applications over Wireless Networks (MediaWiN 2006), April 2006.