Flow-Level Simulation of Call Admission Control schemes in EDCA-based WLANs

Cristina Cano, Boris Bellalta
Universitat Pompeu Fabra
Passeig Circumval.lació 8, 08003 Barcelona, Spain
Tel: +34 93 542 18 95
Email: *cristina.cano@upf.edu*, *boris.bellalta@upf.edu*

TD(07)015
The 8th COST 290 Management Committee Meeting
Universidad de Málaga, Spain, February 15-17, 2007

Abstract

A flow-level simulator of EDCA-based WLANs to evaluate Call Admission Control (CAC) schemes and MAC parameters tuning algorithms is presented. A novel Admission Control scheme for infrastructure WLANs is evaluated. This Admission Control uses an adaptive MAC parameter tuning algorithm which decides the best instantaneous configuration of MAC parameters in order to: *i*) guarantee the QoS requirements for sensitive flows (such as VoIP) and *ii*) maximize the best-effort throughput (web browsing and P2P transfers). Results, compared with the standard *EDCA* and *DCF*, show clearly how this proposal provides a higher performance to sensitive flows with a controlled balance for the non-sensitive traffic.

Keywords: IEEE 802.11e, WLAN, QoS.

1 Introduction

The IEEE 802.11e EDCA mode of operation [1] provides traffic differentiation capabilities based in a set of static ACs (Access Categories), which assure a hard and conservative prioritization of sensitive (or rigid) flows (*with strict bandwidth-delay requirements*), resulting in a low performance of best-effort (or elastic) flows (*with soft bandwidth-delay requirements*). However, by properly choosing the MAC defined parameters of each AC, improved performance levels can be achieved for both types of traffic, guaranteeing the required QoS for the rigid flows and, at the same time, maximizing the best-effort throughput. For example, in [2] an experimental study of the EDCA performance has been carried out. The authors remark the usefulness of using a mechanism to derive the maximum per-flow admissible sending rate and to perform admission control accordingly. They also suggest that the mechanism has to be based on realistic source models, not only for saturation conditions and including realistic un-saturated source models.

Furthermore, the IEEE 802.11e standard defines the required signalling messages to support an Admission Control mechanism. The admission control procedures are left open to be implemented by each equipment vendor. A simple algorithm which implements the admission control procedures is already included in the standard [1].

We present a flow-level simulator of IEEE 802.11e networks that could be used to evaluate multiple Admission Control schemes and MAC parameter tuning algorithms. The join performance of two simple tuning MAC parameters algorithms (iterative and non-iterative) and a model-based admission control is presented. The simulator uses an EDCA mathematical model from [3] to obtain the packet level performance of the system (flow throughput, packet delay, losses, etc.). It considers all MAC QoS enhancements, such as the transmission opportunity ($TXOP$), the use of different inter-frame spaces ($AIFS$) and different values of the back-off instance (CW_{min}, CW_{max}). The model is used to evaluate each system state and then, it decides whether a new incoming flow could be accepted or not and tests the possible MAC parameter combinations.

2 Enhanced Distributed Channel Access

The EDCA mode of operation of the IEEE 802.11e is an extension of the DCF with the goal to provide priorities and traffic differentiation in the wireless access. To achieve this traffic differentiation, the medium access control protocol classifies each traffic flow in an Access Category (AC). Four AC

are defined, each one associated to one MAC transmission queue. Each AC has its own MAC parameters and behaves independently of others. Letting $AC_{i,j}$ be the access category j of the i-STA, the basic MAC parameters of each access category are labeled as: *Arbitration Interframe Space* $AIFS_{i,j}$, *Minimum Contention Window* $CW_{min,i,j}$, *Maximum Contention Window* $CW_{max,i,j}$ and Transmission Opportunity $TXOP_{i,j}$. In Table 1, the recommended static parameters are shown for the different queues.

AC	$AIFSN_j$	$TXOP_{limit}$ (ms)	$CW_{min,j}$	$CW_{max,j}$
0 (Background: BK)	7	0	CW_{min}	CW_{max}
1 (Best effort: BE)	3	0	CW_{min}	CW_{max}
2 (Video: VI)	2	6.016	$CW_{min}/2$	CW_{min}
3 (Voice: VO)	2	3.264	$CW_{min}/4$	$CW_{min}/2$

Table 1: Default EDCA Parameter Set element parameter values for the 802.11b specification

According to the basic access (BA) mechanism, when node i has no packets to transmit and receives a packet from the network layer, it sends the packet to the corresponding $AC_{i,j}$ queue. At the same time, the node starts to sense the channel to determine its state, that can be either *busy* or *free*. If the channel is detected busy, the node waits until the channel is released. When the channel is detected free for a period of time larger than the $AIFS_{i,j}$ duration, a new backoff instance is generated, which consists on a counter set to a random value. The random value is chosen from an uniform distribution in the range $CW_{i,j}(k) = [0, min\left(2^k CW_{min,i,j} - 1, CW_{max,i,j} - 1\right)]$, where k is the current packet transmission attempt. For each packet to be transmitted, k is initially set to 0 and it is increased by one unit at each failed transmission until a maximum number of retransmissions, called Retry Limit, is reached, moment in which the packet is dropped.

The backoff counter is decreased by one each time-slot in which the channel is sensed free, until the countdown reaches zero, instant in which the node starts the packet transmission on the channel. If, during the backoff countdown, the channel is sensed busy, the backoff is suspended until the channel is detected free again. The $AIFS_{i,j}$ value is computed using a non-negative integer $AIFSN_{i,j}$ specific for each $AC_{i,j}$: $AIFS_{i,j} = SIFS + AIFSN_{i,j}\sigma$ (where σ is an empty SLOT duration). Once a node gets the channel, it can transmit up to $B_{i,j}$ $MPDU$ packets ($TXOP_{i,j}$ limit). This limit is expressed in time units (ms) and corresponds to the consecutive time that a node can transmit few (large) or several (small) packets.

A channel collision occurs if two nodes transmit at the same time, i.e., a backoff instance from two nodes reach 0 at the same time. After the data packet is transmitted to the channel by the sender, the receiver waits for a SIFS (Short Inter-Frame Spacing) time and sends a MAC layer ACK to acknowledge the correct reception of the data packet. In the case the sender does not receive the ACK frame, it starts the retransmission procedure. After discarding or successfully transmitting a packet, if more packets are ready to be transmitted, the node starts the transmission procedure again. Otherwise, it waits for a new packet from the network layer. Another EDCA feature is the use of the different ACK policies (no ACK transmission or ACKs aggregation) which can also be used to improve the system performance.

Alternative to the BA mechanism, nodes can employ a RTS/CTS protocol to access the channel, so as to reduce the hidden terminal effect.

3 Flow-level Simulator

The flow-level simulator has been developed using the COST (Component Oriented Simulation Toolkit) simulation engine [4]. The simulator models the procedures followed by the admission control, the MAC parameters tuning algorithm and the behavior of the STAs (flow arrival and departure rates). To capture the packet level behavior, it uses the EDCA model presented in [3].

3.1 Motivation

Traffic differentiation schemes, schedulers and admission control algorithms must be evaluated at packet level in order to assess their performance in terms of throughput, packet delay or losses. However, also flow level metrics such as the blocking / dropping probabilities or flow transfer delays are fundamental to characterize a system. Moreover, some implications of those mechanisms are only understood at flow/call level. For example, a reduction of the best-effort bandwidth causes a higher transfer delay of the elastic flows, resulting in a high number of active flows competing for the channel, which in some cases could be worst than allowing the best-effort flows to transmit at maximum rate [3].

Thus, the motivation to develop a flow-level simulator resides in the need to use a general tool which will allow us to compare different call admission control schemes (with / without preemption, with guard channels based in acceptance thresholds, etc.) and different EDCA parameters combinations with heterogeneous traffic conditions (with multiple traffic classes) and with node mobility to capture the multi-rate operation mode and other channel impairments.

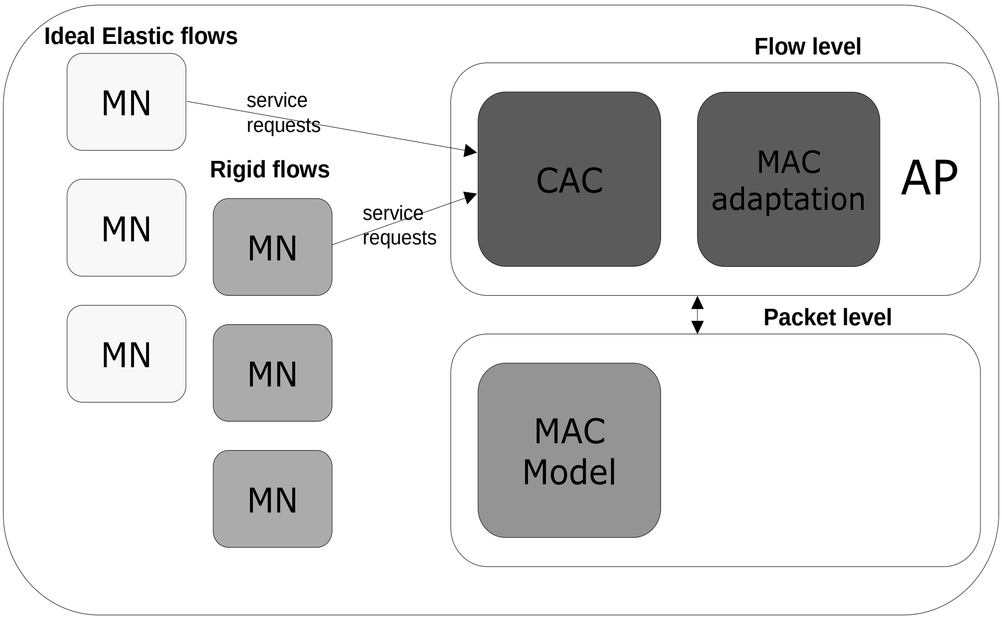


Figure 1: Flow-level simulator structure

3.2 Simulator Structure

In Figure 1 a block-scheme of the simulator is depicted. It is based on a modular structure with three main components: *i*) the MNs, which send service requests to the Admission Control module specifying the traffic profile and the requirements, *ii*) the admission control, which decides whether to accept/reject the incoming flow or to perform some action over the already active flows and *iii*) the MAC parameters tuning algorithm, which is governed by the admission control and selects / tests combinations of the EDCA MAC parameters based on the information provided by the admission control. In the next section the different blocks are described.

Notice that the packet level metrics, such as the achieved throughput or packet delays for each flow are obtained from an analytical model. The combination of simulation / mathematical models provides both a high flexibility (at flow level) and fast computation for the packet-level metrics, allowing long simulations without and excessive computational cost.

4 Call Admission Control Procedure in EDCA

The standard procedure defined in [1] specifies that when the admission control mandatory (ACM) field is activated for an specific AC, all MNs which want to use this AC must send an ADDTS

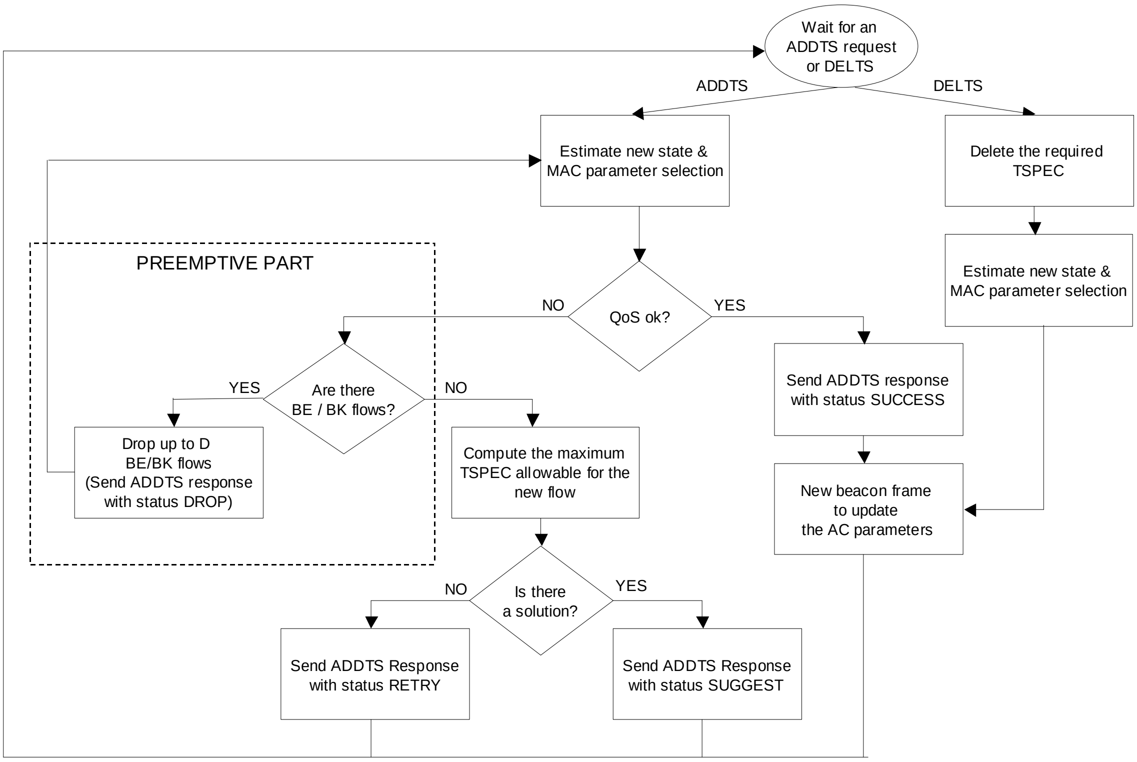


Figure 2: Call Admission Control procedure at the AP

(Add Traffic Stream) request containing the TSPEC (Traffic SPECification) for the new flow. Once the CAC receives the ADDTS frame it decides whether to *i*) *admit the new flow with the requested TSPEC*, *ii*) *reject the flow* or *iii*) *suggest another TSPEC alternative, which is expected to be acceptable for the network*. Once a decision is taken, the CAC sends to the MN the corresponding ADDTS response. When the station receives the response it decides if it satisfies the MN flow requirements or not. If both the CAC and the MN accept, the flow becomes active. Otherwise, the requesting process can be repeated. When the flow finishes, the MN must send a DELTS (DELete Traffic Stream) packet, then the CAC can release the resources used by the flow.

According to the previous signalling protocol, we have developed a novel Admission Control scheme based on the adaptation of the MAC parameters to each network state (number and type of flows). Using the information in the TSPEC of each ADDTS request, the Call Admission Control located at the Access Point decides whether the new state of the network is feasible and compute the most suitable MAC parameters (using the estimations provided by the analytical model).

In Figures 2 and 3 our scheme of the Call Admission Control at the AP and at MNs respectively is shown.

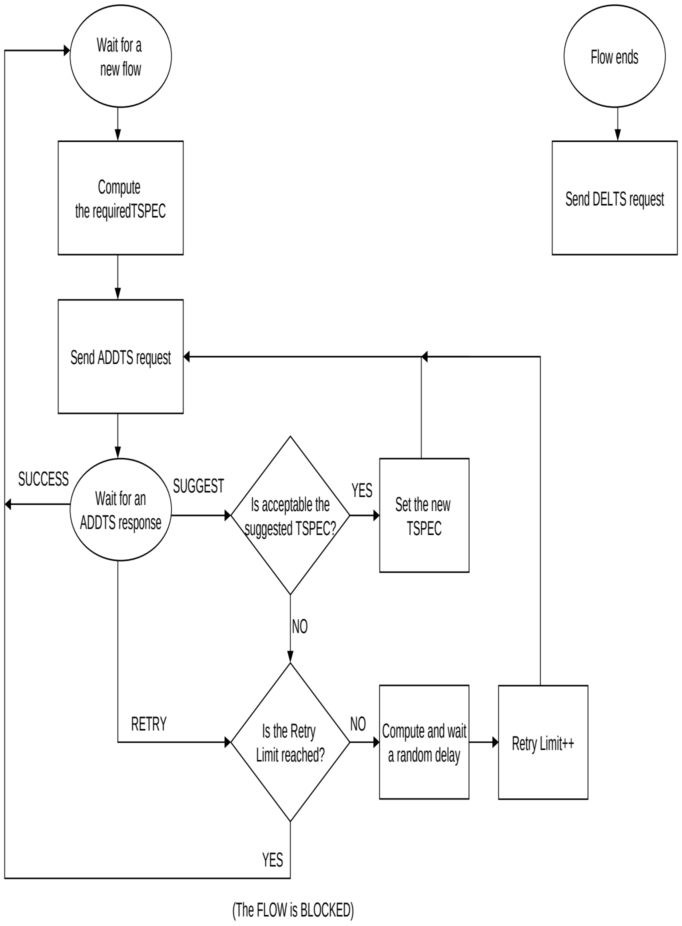


Figure 3: Call Admission Control procedure at MNs

5 Tuning the EDCA MAC parameters

5.1 Iterative MAC Parameters Tuning Algorithm (EDCA-iter)

The first algorithm was presented in [5] and it is based on the outcomes produced by a multi-goal optimization function, which using the analytical EDCA model, was evaluated under four different objectives: *i*) *to guarantee the throughput of the streaming flows*, *ii*) *a queue utilization of rigid flows near* 0.8, *iii*) *an average queuing delay (including service time) lower than* 150 *ms* and *iv*) *to achieve the maximum allowable elastic throughput.* The optimization problem is solved by using the MATLAB [1] package. The allowed range of MAC parameters was: $CW_{min,BE} = [32 - 1024]$, $CW_{min,VO} = [8 - 32]$, $A_{BE} = [3 - 7]$, $A_{VO} = [2]$, $TXOP_{BE} = [1 - 7]$ and $TXOP_{VO} = [1 - 7]$.

From that outcomes, an iterative algorithm was developed:

1. Initialize the parameter vector to the EDCA defined parameters, so $\xi = [BE : (32, 7, 3), VO : (8, 1, 2)]$ where $([AC : CW_{min}, TXOP, AIFS])$. Set the goals vector $\mathbf{G} = (S_{VO}, \mathbf{D})$ to the desired values of throughput S_{VO} and delay $\mathbf{D}$ (mean and jitter). Go to 2.
2. Using the current parameters ξ. Check whether the goals (allowing a 1 % of packet losses) are achieved or the maximum number of iterations have been reached. If so, go to Step 4. Else, go to Step 3.
3. Increase/Decrease sequentially (only one change each iteration):
   (a) $TXOP_{VO} \rightarrow TXOP_{VO} + 1$. Update ξ. Go to Step 2.
   (b) $A_{BE} \rightarrow A_{BE} - 1$. Update ξ. Go to Step 2.
   (c) $TXOP_{BE} \rightarrow TXOP_{BE} - 1$. Update ξ. Go to Step 2.

   If the maximum values of all three parameters have been reached, double at each iteration the value of $CW_{min,BE}$ until its maximum value is reached. Update ξ. Go to Step 2.
4. This algorithm is only suitable for a single-hop ad-hoc network. To solve that limitation, allowing to use it in an infrastructure WLAN, different parameters for the AP are used:
   (a) $AIFSN_{BE,DL} \rightarrow max(1, AIFSN_{BE,UL} - 1)$.
   (b) $TXOP_{BE,DL} \rightarrow min(10, TXOP_{BE,UL} \cdot n_{BE,DL})$.

---

[1] © http://www.mathworks.com

(c) $TXOP_{VO,DL} \rightarrow min(10, TXOP_{VO,UL} \cdot n_{VO,DL})$.

(d) $CW_{min,BE,DL} \rightarrow CW_{min,BE,UL}$.

5. Set the working parameters to ξ.

5.2 Real Time MAC Parameters Tuning Algorithm (EDCA No-iter)

An heuristic real-time algorithm to select the most suitable combination of the MAC parameters is presented in this section. The development of real-time algorithms is crucial to provide practical solutions that could be implemented in commercial APs and wireless network cards. Moreover, they have to use only the information that the Admission Control can store, such as the number of active flows at both uplink and downlink (n_{DL}, n_{UL}), the bandwidth requested by each rigid flow (B_r) or, in case of an elastic flow, the minimum bandwidth required ($B_{e,min}$).

The algorithm is based on a well-known performance result of the *DCF*, which states that the maximum load (utilization) that a WLAN is able to carry is below the 80% [6]. However, notice that this assumption it's not general and depend on the different traffic profiles parameters (for example, the packet length). Clearly, it does not hold for VoIP calls as the AP saturates at lower loads than the 80% stated. Therefore, the presented algorithm is designed to be general and as a particular case, here it is evaluated with VoIP and elastic traffic.

Then, the algorithm (Algorithm 1) starts computing the minimum aggregated bandwidth, β, required by all flows (including the flow which is requesting service). Using this value and the maximum physical bandwidth achievable, the data rate R_{data} (a single data rate is considered for the AP and all STAs), the MAC parameters are adjusted based on a set of predefined thresholds. As can be seen only best-effort (BE) and voice (VO) access categories have been considered. Note that in the computation of β, and assuming that the best-effort flows have a very low minimum bandwidth, the β value reflects the load of real-time traffic.

When β is low, the best-effort MAC parameters (AC_BE) allow that best-effort flows could achieve a high throughput performance (with high TXOP values and lower AIFS). However, as the load of sensitive flows increases, the AC_BE parameters reduce the throughput of best-effort flows. For the streaming MAC parameters (AC_VO) only the TXOP is adjusted, which is also increased with β. The α is set to 9 and γ is equal to 0.4, both values are derived from the iterative algorithm and they are related with the amount of bandwidth required by a real-time flow and the R_{data} value, in the following way: $\frac{\alpha}{\gamma} < \frac{B_{VO}}{R_{data}}$.

Table 2: Values of MAC parameters

Parameter	Value	Parameter	Value
$CW_{min,BE,UL}$	[32-1024]	$CW_{min,BE,DL}$	[32-1024]
$CW_{min,VO,UL}$	8	$CW_{min,VO,DL}$	8
$CW_{max,BE,UL}$	1024	$CW_{max,BE,DL}$	1024
$CW_{max,VO,UL}$	32	$CW_{max,VO,DL}$	32
$AIFSN_{BE,UL}$	[3-10]	$AIFSN_{BE,DL}$	[2-9]
$AIFSN_{VO,UL}$	2	$AIFSN_{VO,DL}$	1
$TXOP_{UL}$	[1-10]	$TXOP_{DL}$	[1-10]

The highest threshold is set conservatively to 70% (note that it is lower than the previous 80%) of the R_{data}, assuming that it is the maximum efficiency that the Hotspot can achieve. The other thresholds are uniformly distributed between the interval from 35% to 70% in order to reduce progressively the best-effort traffic contention.

The values of the fixed MAC parameters and the ranges of the dynamic ones are shown in Table 2.

The problem of the downlink unfairness in WLANs using the *DCF* random access protocol has been intensively studied as one of the limitations of this type of networks [7, 8]. It appears because the AP and the STAs have the same probability to access the channel as the MAC protocol is designed to provide a node-based fairness (two nodes with a packet ready to be transmitted have the same probability to access the channel). However, the AP usually carries more traffic than any other node in the cell. For example, in a symmetrical-flow scenario the AP carries the same amount of data than the rest of STAs requiring to access the channel more often.

To mitigate the uplink versus downlink unfairness, the proposed algorithm selects different MAC parameters for the AP and for the STAs (Algorithm 1). For each *AC* the AIFS of the downlink will be one unit lower than the one used for the STAs and the TXOP is increased proportionally as the number of flows in the downlink ($n_{BE/VO,DL}$). Finally, the CW_{min} of elastic uplink flows is also set depending on the number of active STAs contending for the channel with the AP ($n_{BE,UL}$).

Once the MAC parameters are selected, the admission control estimates if the new state is feasible (all flows achieve their required QoS). Notice that a single estimation step is done, reducing the high computational cost of using an iterative algorithm, which requires to estimate at each step if the selected MAC parameters are suitable to accept the new incoming flow. However, there is a performance cost that will be evaluated in the following section.

Algorithm 1 Real Time MAC Parameters Tuning Algorithm

```
$\beta = \sum_{n_{DL/UL,BE}} B_{e,min} + \sum_{n_{DL/UL,VO}} B_r$
STAs $AC$ Parameters:
    $TXOP_{BE,UL} \rightarrow max(1, \lfloor 10 - \frac{\alpha \cdot \beta}{\gamma \cdot R_{data}} \rfloor)$
    $TXOP_{VO,UL} \rightarrow min(10, 12 - TXOP_{BE,UL})$
    $AIFSN_{BE,UL} \rightarrow min(10, \lceil 3 + \frac{\alpha \cdot \beta}{\gamma \cdot R_{data}} \rceil)$
    Set the $CW_{min,BE,UL}$ to:
    $\rightarrow$ 1024 if $(\beta > 0.7 \cdot R_{data})$
    $\rightarrow$ 512 if $(\beta > 0.5 \cdot R_{data})$
    $\rightarrow$ 256 if $(\beta > 0.45 \cdot R_{data})$
    $\rightarrow$ 128 if $(\beta > 0.4 \cdot R_{data})$
    $\rightarrow$ 64 if $(\beta > 0.35 \cdot R_{data})$
    $\rightarrow$ 32 if $(\beta < 0.35 \cdot R_{data})$
    if$(n_{BE/VO,DL} > 0)$
    $CW_{min,BE,UL} \rightarrow min(1024, 32 \cdot n_{BE,UL})$
AP $AC$ parameters:
    $AIFSN_{BE,DL} \rightarrow max(1, AIFSN_{BE,UL} - 1)$
    $TXOP_{BE,DL} \rightarrow min(10, TXOP_{BE,UL} \cdot n_{BE,DL})$
    $TXOP_{VO,DL} \rightarrow min(10, TXOP_{VO,UL} \cdot n_{VO,DL})$
    $CW_{min,BE,DL} \rightarrow CW_{min,BE,UL}$
```

6 Performance Results

Two scenarios are defined to evaluate the presented admission control (without considering preemption) and the two EDCA parameters tuning algorithms.

1. The number of nodes with VoIP (bi-directional rigid flows), downlink P2P and Web flows are fixed to $n_{VO} = 4$, $n_{P2P,DL} = 2$ and $n_{WEB,DL} = 2$ respectively, while the number of nodes with P2P uplink flows ranges from 0 to 20. As has been proved, the uplink BE flows cause the worst performance degradation to rigid flows (especially to the downlink ones), see [3]. Therefore, in this scenario, we show how the *EDCA* and the the algorithms presented are able to solve the rapid performance degradation of the rigid flows.

2. In the second scenario we change the number of VoIP calls, ranging n_{VO} from 0 to 25. The number of downlink/uplink P2P and downlink Web flows are maintained constant and equal to $n_{WEB,DL} = 2$, $n_{P2P,DL} = 2$ and $n_{P2P,UL} = 5$ respectively. Here, the goal is to observe if the system is able to allocate enough resources for new rigid flows and how to manage the bandwidth reduction for the best-effort flows.

In Table 3, the considered traffic profiles are shown. The elastic flows (HTTP and P2P) will use the AC_BE queue and the VoIP calls the AC_VO. In both scenarios, the flow arrival rates

Characteristic	Web	P2P	VoIP
Application/Codec	Web browsing	File sharing	G.711
Type	Elastic	Elastic	Rigid
Bandwidth	$> 10Kbps$	$> 10Kbps$	$80Kbps$
Packet Length	$1500bytes$	$1500bytes$	$200bytes$
Average flow duration/Total transfer	$50Kbytes$	$10Mbytes$	$120s$
Average Inter-arrival Time	$30s$	$30s$	$600s/300s$

Table 3: Traffic Characteristics

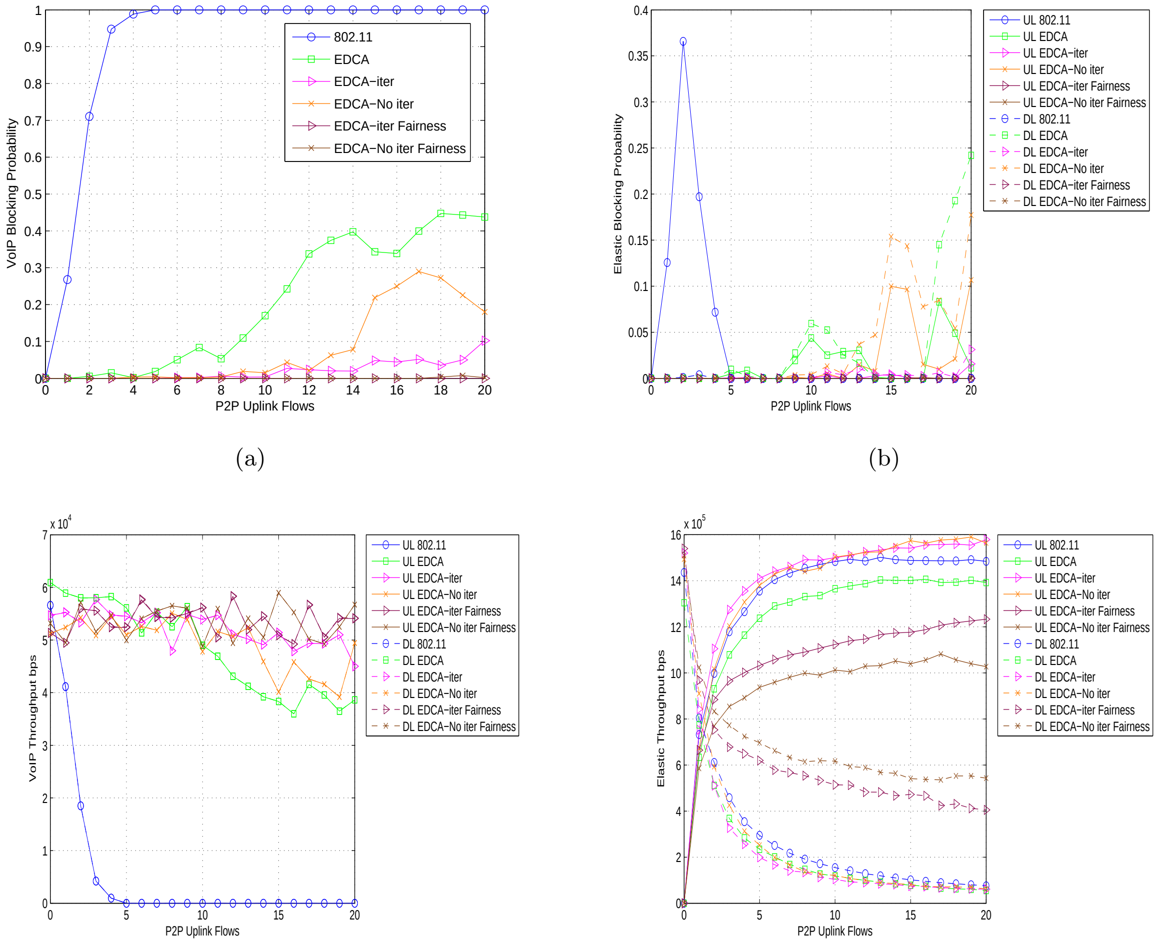
follow a Poisson process, the VoIP call duration is exponentially distributed and both P2P and Web flow lengths depends on the bandwidth used by them, which changes dynamically with time (this bandwidth is changed by the CAC each time the network state change).

In order to evaluate the effectiveness and the importance of the uplink / downlink fairness part of the presented algorithms, it could be disabled (it is not considered). When it is active, the plots are labeled with the term "fairness".

In Fig. 4 the results obtained for the Scenario 1 are shown. Notice that the blocking probability of VoIP flows using *EDCA-Iter* is lower than the one obtained by *EDCA*. Obviously, in both cases, the gain compared with *DCF* is clear. To asses this improvement *EDCA-Iter* (the iterative algorithm to tune the MAC parameters) decrease the instantaneous elastic throughput by adjusting the affordable bandwidth of BE flows by means of tuning the MAC parameters of the AC_BE. Notice also that the results obtained using the real-time algorithm, *EDCA-No iter*, are closer to the ones obtained by the iterative one. In the uplink, using *EDCA-Iter* the average BE throughput is also higher than the obtained by *EDCA* or *DCF*. As the proposed algorithm increase the uplink / downlink unfairness, the downlink throughput obtained by using them is lower than the achieved by *EDCA* and the *DCF*.

When the uplink / downlink *fairness* is used, the blocking probability decrease to zero, showing the importance of that mechanism, as the AP is the network bottleneck (new flows could not be accepted because the throughput in the downlink could not be guaranteed). Notice also that with the fairness part of the algorithm active, the throughput in the both the uplink and downlink are better balanced than without it.

Figure 5 shows the results from the Scenario 2. Notice how using the *DCF* the system is unable to accept any VoIP call as it is already saturated by the BE flows. However, using both *EDCA* and *EDCA-Iter*, a clear VoIP blocking probability gain is obtained, which is translated into a higher throughput for the VoIP flows. As expected, using the *EDCA-Iter* the VoIP performance is clearly

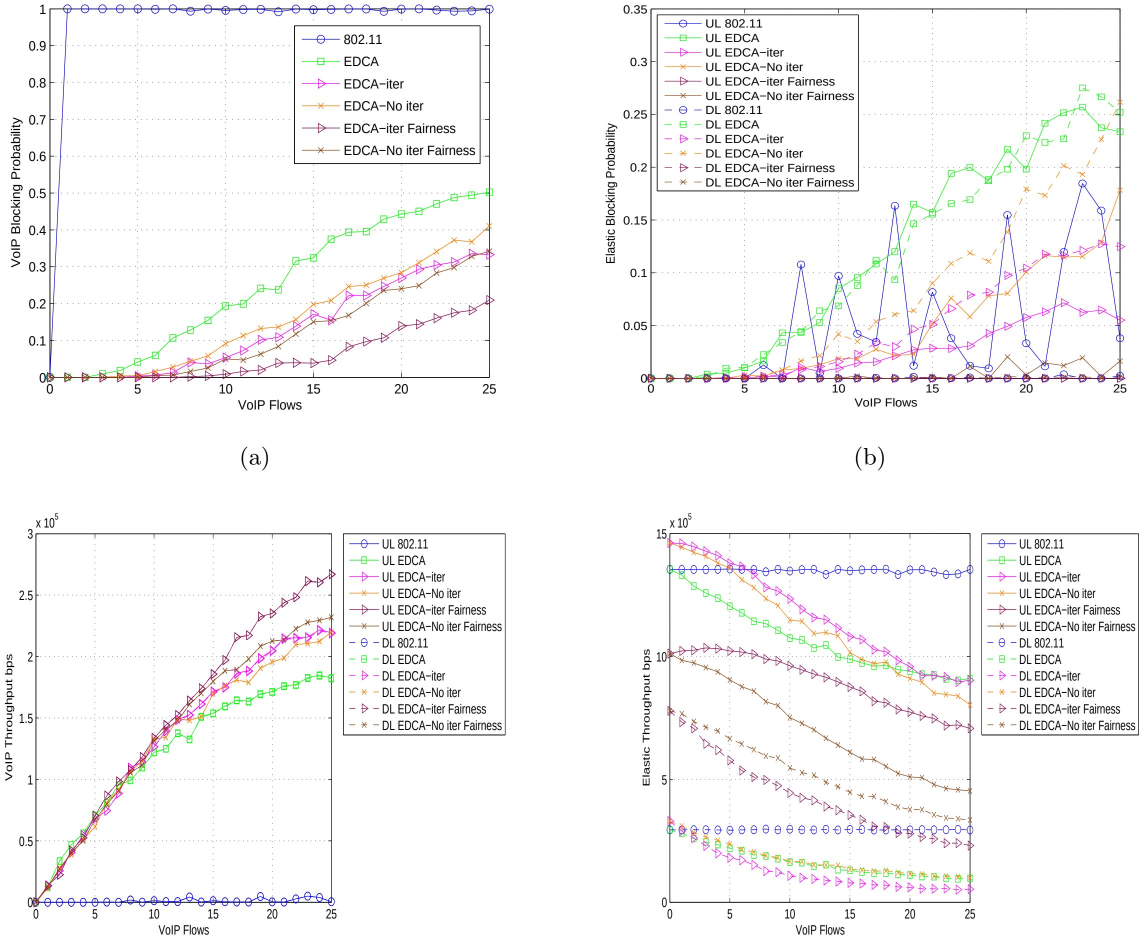


(a) (b) (c) (d)

Figure 4: Scenario 1. (a) VoIP Blocking Probability (b) Elastic Blocking Probability (c) VoIP Throughput (d) Elastic Throughput

improved due to the greater flexibility to adapt the BE instantaneous rate, as we mentioned before. The use of the fairness algorithm leads to an improvement in the blocking probability for the VoIP flows. Using the *EDCA-NoIter*, the results are worst than the ones from the *EDCA-Iter* but still considerable better than using *EDCA*. With regards to the elastic flows, notice that the results follow the expected the results.

Results clearly show that the overall cell performance is improved due to a better use of the transmission resources, which are allocated properly to satisfy the requirements of each traffic flow. Then, two main results are achieved i) *the performance of sensitive traffic is significatively improved* and ii) *the average best-effort traffic is maximized in a wide range of possible situations.*



(a) (b) (c) (d)

Figure 5: Scenario 2. (a) VoIP Blocking Probability (b) Elastic Blocking Probability (c) VoIP Throughput (d) Elastic Throughput

7 Conclusions and future work

This report presents a flow-level simulator of a WLAN Hotspot. The simulator has been developed to study the performance of join admission control schemes and EDCA MAC parameters tuning algorithms at both flow-level (blocking probability, average number of active flows, average bandwidth used etc.) and at packet level (flow throughput, packet delays, losses, etc.). The preliminary results presented allow to compare the DCF, the EDCA and two adaptive versions (iterative and non-iterative). Note the remarkable gain which is achieved by using the adaptive solutions.

This study can be further expanded with the evaluation of more specific characteristics of the admission control algorithm (such the retry-request mechanism, the computation of the suitable TSPEC by the Admission Control or the possibility to preempt calls), the impact of mobility and/or considering multiple transmission rates and other channel impairments.

Finally, this work is intended to be validated in a test-bed to show that similar gains can also be obtained in real conditions, where a computational and efficient solution for the admission control and EDCA parameter tuning algorithm is required to allow the system work in real time.

References

[1] IEEE Std 802.11e. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications; Amendment: Medium Access Control(MAC) Quality of Service Enhancements. *IEEE Std 802.11e*, 2005.

[2] Albert Banchs, Arturo Azcorra, Carlos Garcï¿½, and Rubï¿½ Cuevas. Applications and Challenges of the 802.11e EDCA Mechanism: An Experimental Study. *IEEE Network, p.52-58*, July/August 2005.

[3] B. Bellalta, M. Meo, and M. Oliver. VoIP Call Admission Control in WLANs in Presence of Elastic Traffic. *Journal of Communications Software and Systems*, To be published in January 2007.

[4] Gilbert (Gang) Chen. Component Oriented Simulation Toolkit. *http://www.cs.rpi.edu/~cheng3/*, 2004.

[5] B. Bellalta, C. Cano, M. Oliver, and M. Meo. Modeling the IEEE 802.11e EDCA for MAC parameter optimization. *Het-Nets'06, IlKley, UK*, September 2006.

[6] A. Ksentini, A. Guï¿½oui, and M. Naimi. Adaptive Transmission Opportunity with Admission Control for IEEE 802.11e Networks. *ACM/IEEE MSWIM 2005, Montreal, Quebec, Canada*, October 2005.

[7] C. Casetti and C.-F. Chiasserini. Improving Fairness and Throughput for Voice Traffic in 802.11e EDCA. *IEEE PIMRC 2004, Barcelona, Spain*, September 2004.

[8] Jiwoong Jeong, Sunghyun Choi, and Chong kwon Kim. Achieving Weifhted Fairness between Uplink and Downlink in IEEE 802.11 DCF-Based WLANs. *IEEE QoS in Heterogeneous Wired/Wireless Networks (QShine'05), Orlando, FL, (USA)*, August 2005.