

# Adaptive EDCAF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks

Lamia Romdhani, Qiang Ni, and Thierry Turletti

INRIA Sophia Antipolis, 2004 Route des Lucioles, BP-93, 06902 Sophia Antipolis, France

Email: {lromdhan, qni, turletti}@sophia.inria.fr

**Abstract**— This paper describes an adaptive service differentiation scheme for QoS enhancement in IEEE 802.11 wireless ad-hoc networks. Our approach, called Adaptive Enhanced Distributed Coordination Function (AEDCF), is derived from the new EDCAF introduced in the upcoming IEEE 802.11e standard. Our scheme aims to share the transmission channel efficiently. Relative priorities are provisioned by adjusting the size of the Contention Window (CW) of each traffic class taking into account both applications requirements and network conditions. We evaluate through simulations the performance of AEDCF and compare it with the EDCAF scheme proposed in the 802.11e. Results show that AEDCF outperforms the basic EDCAF, especially at high traffic load conditions. Indeed, our scheme increases the medium utilization ratio and reduces for more than 50% the collision rate. While achieving delay differentiation, the overall goodput obtained is up to 25% higher than EDCAF. Moreover, the complexity of AEDCF remains similar to the EDCAF scheme, enabling the design of cheap implementations.

## I. INTRODUCTION

IEEE 802.11 wireless LAN specification defines two different ways to configure a wireless network: ad-hoc and infrastructure mode. In infrastructure mode an Access Point (AP) is needed to connect wireless stations to a distribution system, whereas in ad-hoc mode all wireless stations are distributed without access coordinator. In this paper, we focus on ad-hoc networks since distributed random access control are often preferred to centrally coordinated access control [10], [11]. Distributed Coordination Function (DCF) is the basic medium access mechanism of 802.11 for both ad-hoc and infrastructure modes. It uses CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocol. In this mode, if the medium is found idle for longer than a DIFS (Distributed InterFrame Space) then the station can transmit a packet. Otherwise, a backoff process is started. More specifically, the station computes a random value called backoff time, in the range of 0 and CW (Contention Window) size. The backoff timer is periodically decremented by one for every time slot the medium remains idle after the channel has been detected idle for a period greater than DIFS. As soon as the backoff timer expires, the station can access the medium. If no acknowledgment is received, the station assumes that collision has occurred, and schedules a retransmission by re-entering the backoff process.

Quality of Service (QoS) support is critical to multimedia applications. Time-bounded services such as audio and video conference typically require some specified bandwidth, delay and jitter guarantee, but can tolerate some losses. However, in

DCF all the stations in a Basic Service Set or all the flows from the same station compete the resources and channel with the same priority. There is not any differentiation mechanism to guarantee packet delay and jitter to stations or flows supporting time-bounded multimedia services. The performance evaluation results in [3], [4] show that DCF suffers from significant throughput degradation and high delay at high load conditions, which are caused by the increasing time used for channel access negotiation. Many medium access schemes have been proposed for IEEE 802.11 WLAN to provide some QoS enhancements for real-time traffics. Previous research works mainly focus on the station-based DCF enhancement scheme [1], [5], [9], [11]. Other recent works focus on queue-based enhancement schemes [2], [7], [15] since they perform more efficiently. More related works are detailed in [8].

In parallel the IEEE working group is currently working on the support of QoS in a new standard, called IEEE 802.11e [15]. A new access method called Hybrid Coordination Function (HCF) is introduced, which combines functions from the DCF and Point Coordination Function (PCF) mechanisms. Enhanced DCF (EDCAF) is a contention-based HCF channel access specified in IEEE 802.11e [7], [15]. The goal of this scheme is to enhance the DCF access mechanism of IEEE 802.11 and to provide a distributed access approach that can support service differentiation. The proposed scheme provides capability for up to eight types of traffic classes. It assigns a short CW to classes that should have higher priority in order to ensure that in most cases, high-priority classes will be able to transmit before the low-priority ones. Indeed, the  $CW_{min}$  parameter can be set differently for different priority classes, yielding higher priority classes with smaller  $CW_{min}$ . For further differentiation, in 802.11e different IFS (Inter Frame Space) can be used according to traffic classes. Instead of DIFS, an Arbitration IFS (AIFS) is used. The AIFS for a given class should be a DIFS plus some (possibly zero) time slots. Classes with the smallest AIFS will have the highest priority as it is shown in Figure 1. Each Traffic Category (TC) within the station behaves like a virtual station: it contends for access to the medium and independently starts its backoff time after sensing if the medium is idle for at least AIFS.

Per priority differentiation used by EDCAF ensures better services to high priority class while offering a minimum service for low priority traffic. Although this mechanism improves the quality of service of real-time traffic, the performance obtained are not optimal since EDCAF parameters cannot be

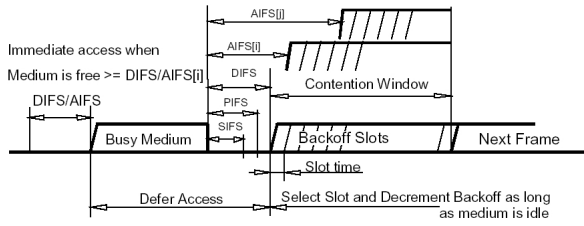


Fig. 1. Some IFS relationships

adapted to the network conditions. In fact, since each TC is implemented as a virtual station, the collision rate increases very fast when the contentions to access the shared medium are very high, which significantly affects the goodput, the latency and thus, decreases the performance of delay-bounded traffic [6]. This motivates us to propose a scheme that adapts the  $CW$  parameter according to the network conditions. Our new scheme called AEDCF aims to provide real-time support in 802.11 ad-hoc networks.

The remainder of this paper is organized as follows. In Section II, we describe AEDCF in detail. Simulation methodology and performance evaluation of our proposal are detailed in Section III. Section IV concludes the paper by summarizing results and outlining future works.

## II. THE ADAPTIVE EDCF (AEDCF) SCHEME

Figure 2 compares the 802.11e architecture that supports queue-based differentiation with the original one queue based DCF access mechanism. To improve the performance under different load rates and to increase the service differentiation in EDCF-based networks, we propose a new scheme called Adaptive EDCF (AEDCF). This scheme extends the basic EDCF by making it more adaptive taking into account network conditions.

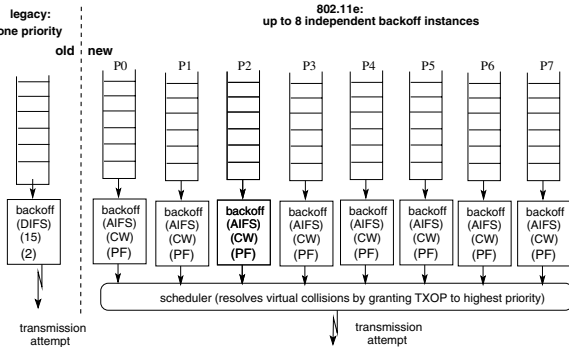


Fig. 2. Queue-based EDCF vs. basic DCF

We assume that  $n$  stations are sending packets through the wireless media. The flows sent by each station may belong to different classes of service with various priority levels. In each station and for each class  $i$ , the scheme maintains: the current contention window value ( $CW[i]$ ), the minimum contention window value ( $CW_{min}[i]$ ), and the maximum contention

window value ( $CW_{max}[i]$ ). Note that  $i$  varies from 0 (the highest priority class) to 7 (the lowest priority class).

### A. Scheme Description

In order to efficiently support time-bounded multimedia applications, we use a dynamic procedure to change the contention window value after each successful transmission or collision. We believe that this adaptation will increase the total goodput of the traffic which becomes limited when using the basic EDCF, mainly for high traffic load.

In the basic EDCF scheme for ad-hoc networks [15], the  $CW_{min}[i]$  and  $CW_{max}[i]$  values are statically set for each priority level. After each successful transmission, the  $CW[i]$  values are reset to  $CW_{min}[i]$ . We propose to reset the  $CW[i]$  values more slowly to adaptive values (different to  $CW_{min}[i]$ ) taking into account their current sizes and the average collision rate while maintaining the priority-based discrimination. In other words, we ensure that at each instant, the highest priority class has the lowest contention window value so that it has the highest priority to access the media. The adaptive slow  $CW$  decrease is a tradeoff between wasting some backoff time and risking a collision followed by the whole packet transmission.

After each collision, the source has to wait for a timeout to realize that the packet has collided, and then doubles its  $CW$  to reduce the number of collisions [15]. We propose to change the mechanism and differentiate between classes using different factors to increase their  $CW$ s.

In the next sub-sections, we explain in detail how the contention window of each priority level is set after each successful transmission and after each collision.

1) *Setting CW After Each Successful Transmission:* After each successful transmission, the basic EDCF mechanism simply sets the contention window of the corresponding class to its minimum contention window regardless the network conditions. Motivated by the fact that when a collision occurs, a new one is likely to occur in the near future, we propose to update the contention window slowly (not reset to  $CW_{min}$ ) after successful transmission to avoid bursty collisions. The simplest scheme we can use to update the  $CW$  of each class  $i$  is to reduce it by a static factor such as  $0.5 * CW_{old}$ . In the remainder of this paper, we denote this approach the Slow Decrease (SD) scheme. However, a static factor cannot be optimal in all network conditions. In our scheme, we propose that every class updates its  $CW$  in an adaptive way taking into account the estimated collision rate  $f_{curr}^j$  in each station. Indeed, the collision rate can give an indication about contentions in a distributed network. The value of  $f_{curr}^j$  is calculated using the number of collisions and the total number of packets sent during a constant period (i.e. a fixed number of slot times) as follows:

$$f_{curr}^j = \frac{E(collisions_j[p])}{E(data\_sent_j[p])}, \quad (1)$$

where  $E(collisions_j[p])$  is the number of collisions of station  $p$  which occurred at step  $j$ , and  $E(data\_sent_j[p])$  is the total number of packets that have been sent in the same period  $j$

by flows belonging to the station  $p$ . Note that the above ratio  $f_{curr}^j$  is always in the range of  $[0, 1]$ .

To minimize the bias against transient collisions, we use an estimator of Exponentially Weighted Moving Average (EWMA) to smoothen the estimated values. Let  $f_{avg}^j$  be the average collision rate at step  $j$  (for each update period) computed according to the following iterative relationship:

$$f_{avg}^j = (1 - \alpha) * f_{curr}^j + \alpha * f_{avg}^{j-1} \quad (2)$$

where  $j$  refers to the  $j^{th}$  update period and  $f_{curr}^j$  stands for the instantaneous collision rate,  $\alpha$  is the *weight* (also called the *smoothing factor*) and effectively determines the memory size used in the averaging process.

The average collision rate is computed dynamically in each period  $T_{update}$  expressed in time-slots. This period should not be too long in order to get good estimation and should not be too short in order to limit the complexity.

To ensure that the priority relationship between different classes is still fulfilled when a class updates its  $CW$ , each class should use different factor according to its priority level (we denote this factor by Multiplier Factor or  $MF$ ). Keeping in mind that the factor used to reset the  $CW$  should not exceed the previous  $CW$ , we limit the maximum value of  $MF$  to 0.8. We have fixed this limit according to an extensive set of simulations done with several scenarios. In AEDCF, the  $MF$  of class  $i$  is defined as follows:

$$MF[i] = \min((1 + (i * 2)) * f_{avg}^j, 0.8). \quad (3)$$

This formula allows the highest priority class to reset the  $CW$  parameter with the smallest  $MF$  value (i.e., priority level 0, see P0 in Figure 2). After each successful transmission of packet of class  $i$ ,  $CW[i]$  is then updated as follows:

$$CW_{new}[i] = \max(CW_{min}[i], CW_{old}[i] * MF[i]) \quad (4)$$

The equation above guarantees that  $CW[i]$  is always greater than or equal to  $CW_{min}[i]$  and that the priority access to the wireless medium is always maintained.

2) *Setting CW After Each Collision* : In the current version of EDCF[15], after each unsuccessful transmission of packet of class  $i$ , the  $CW$  of this class is doubled, while remaining less than the maximum contention window  $CW_{max}[i]$ :

$$CW_{new}[i] = \min(CW_{max}[i], 2 * CW_{old}[i]). \quad (5)$$

In AEDCF, after each unsuccessful transmission of packet of class  $i$ , the new  $CW$  of this class is increased with a Persistence Factor  $PF[i]$ , which ensures that high priority traffic has a smaller value of  $PF[i]$  than low priority traffic:

$$CW_{new}[i] = \min(CW_{max}[i], CW_{old}[i] * PF[i]). \quad (6)$$

In fact, this PF parameter has been proposed in a previous version of the draft, but it has been removed from draft [15]. In this paper, we introduce PF in our AEDCF scheme because by this way we can reduce the probability of a new collision and consequently decrease delay.

## B. Evaluation of Complexity

Our mechanism is easy to implement, and needs very few resources: it requires four registers to buffer the parameters defined above:  $f_{avg}^{j-1}$ ,  $T_{update}$ ,  $MF[i]$  and  $\alpha$ . To reset the  $CW[i]$  values,  $f_{avg}^j$  and  $MF[i]$  parameters defined in Equations 2 and 3, are updated only at the beginning of each new update period  $T_{update}$ . The calculation of  $MF[i]$  requires one addition, two multiplications and one comparison for each active class. Then, two multiplications and two additions are required to compute  $f_{avg}^j$  and one more division to obtain  $f_{curr}^j$  (which is defined in Equation 1 for all the active TCs).

One comparison and one multiplication are required to compute  $MF[i]$  and to decide which value will be used to reset the  $CW[i]$  (see Equation 4). Finally, during the update period, we need two counters to increment collisions and data sent, one comparison and one multiplication that are introduced in Equation 6 to calculate the  $CW_{new}[i]$  and to decide which value will be used to reset the  $CW$ .

## III. SIMULATION METHODOLOGY AND RESULTS

We have implemented AEDCF in the ns-2 simulator, our ns source codes are available in [12]. We report in this section part of simulations we have done with different network topologies and source characteristics. An analysis of performance is presented in detail. In order to show advantages of using an adaptive factor ( $MF[i]$ ) to decrease the  $CW$  after successful transmission, we also present the results of the static Slow Decrease scheme.

### A. Impact of $T_{update}$ and $\alpha$ parameters

As mentioned in Section II-A, our scheme uses an update period (defined in number of time slots) after which it should update the estimated collision rate ( $f_{avg}^j$ ). We have done several set of simulations to observe the effect of the update period on the delay and on the goodput performances [12]. Results obtained show that the update period has a slight impact on the obtained total goodput. However, we have noted that the delay significantly increases when  $T_{update}$  is greater than 6000 time-slots. In the following simulations we have chosen  $T_{update}$  equal to 5000 time-slots which provides a good tradeoff between goodput and latency.

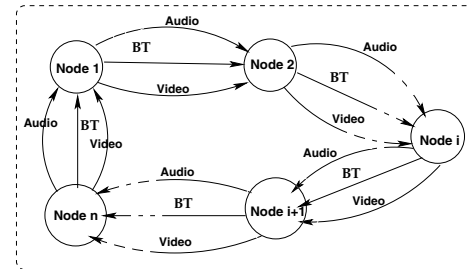


Fig. 3. Simulation Topology

Let us now analyze the impact of the smoothing factor  $\alpha$  on the performance of AEDCF. Note that this factor is

used to estimate the average collision rate defined in Equation 2. For this purpose, we use the topology shown in Figure 3, which consists of  $n$  stations indexed from 1 to  $n$ . Each station generates the same traffic of three data streams, labeled with high, medium and low, according to their priorities. Station  $n$  sends packets to station number 1. Station  $i$  sends to station  $i + 1$  three flows belonging to the three classes of service: Audio (high priority), Video (medium priority), and Background Traffic (denoted by BT for low priority). We use CBR sources to simulate BT, video, and audio traffics.

In the following simulations, we assume that each wireless station operates at IEEE 802.11a PHY mode-6 [14], see network parameters shown in Table I.

TABLE I  
IEEE 802.11A PHY/MAC PARAMETERS USED IN SIMULATION

SIFS	16 $\mu$ s
DIFS	34 $\mu$ s
ACK size	14 bytes
Data rate	36 Mbits/s
Slot_time	9 $\mu$ s
CCA Time	3 $\mu$ s
MAC Header	28 bytes
Modulation	16-QAM
Preamble Length	20 $\mu$ s
RxTxTurnaround Time	1 $\mu$ s
PLCP header Length	4 $\mu$ s

Table II shows the network parameters selected for the three TCs.

TABLE II  
MAC PARAMETERS FOR THE THREE TCs.

Parameters	High	Medium	Low
$CW_{min}$	5	15	31
$CW_{max}$	200	500	1023
AIFS( $\mu$ s)	34	43	52
PF	2	4	5
Packet Size(bytes)	160	1280	200
Packet Interval(ms)	20	10	12.5
Sending rate(Kbit/s)	64	1024	128

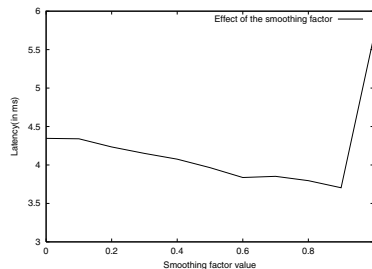


Fig. 4. Effect of the smoothing factor on mean delay

Figure 4 and Figure 5 show the average delay and goodput as a function of the smoothing factor  $\alpha$ , respectively. All the results are averaged over 20 simulations. Twenty-five stations are used to evaluate the effect of smoothing factor, which

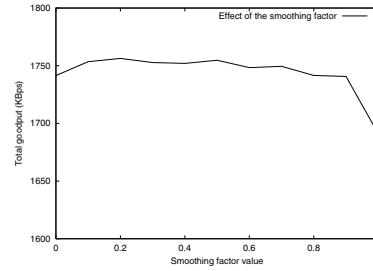


Fig. 5. Effect of the smoothing factor on goodput

corresponds to a load rate of 84.5%. We can note that choosing a value of  $\alpha$  in the range of [0.75, 0.9] achieves a good tradeoff between goodput and mean delay. So in the following simulation scenarios we set  $\alpha$  to 0.8.

### B. Effect of traffic load

To evaluate the performance of AEDCF, we investigate in this section the effect of the traffic load and compare it with the basic EDCF and SD schemes. Our simulations use different types of traffics to evaluate service differentiation. Three queues are used in each station. The highest priority queue in each station generates packets with packet size equal to 160 bytes and inter-packet interval of 20 ms, which corresponds to 64 Kbit/s PCM audio flow. The medium traffic queue generates packets of size equal to 1280 bytes each 10 ms which corresponds to an overall sending rate of 1024 Kbit/s. The low priority queue in each station generates packets with sending rate equal to 128 Kbit/s, using a 200 bytes packet size. To increase the load of the system, we gradually increase the number of stations. All the stations are located within an Independent Basic Service Set such that every station is able to detect a transmission from any other station, and stations are not moving in the simulation. The topology is shown in Figure 3. We start simulations with two wireless stations, then we increase the load rate by increasing the number of stations by one every eight seconds. Figures 6 - 9 show the average obtained over 5 simulations. We increase the number of stations from 2 to 44 which correspond to load rates from 6.7% to 149%.

To evaluate the performance of the different schemes, the following metrics are used:

- **Gain of goodput:** This metric stands for the gain (in %) on the average goodput of new schemes (SD or AEDCF), compared with the basic EDCF. It is calculated as follows:  

$$Gain\_of\_goodput = \frac{AG_{new} - AG_{EDCF}}{AG_{EDCF}} * 100\%$$
- **Mean delay:** It is the average delay of all the flows that have the same priority in the different stations. This metric is used to evaluate how well the schemes can accommodate real-time flows. However, real-time flows also require low average delay and bounded delay jitter.
- **Latency distribution:** Latency distribution allows to trace the percentage of packets that have latency less than the maximum delay required by the applications.

- **Medium utilization ( $M_u$ ):** Due to the scarcity of wireless bandwidth, we also study the medium utilization of the different schemes by computing the percentage of time used for transmission of data frames:

$$M_u = \frac{TotalTxTime - CollisionTime - IdleTime}{TotalTxTime} * 100\%$$

- **Collision rate:** Collisions in wireless LAN cause additional delays by increasing the delay that stations should wait, before initiating a new transmission attempt. This rate is calculated as the average number of collisions that occur per second.

Figure 6 shows the mean delay of the audio flow corresponding to the high priority class. The AEDCF scheme is able to keep the delay low even when the traffic load is very high, i.e., with a large number of stations. We can see that the mean delay of audio for AEDCF is 51% smaller than that for the basic EDCF when the load rate is up to 88% (26 stations). Moreover, the mean delay of audio of AEDCF is still 38% smaller than that of the basic EDCF when the load rate reaches 149% (44 stations). Indeed, when the number of stations is more than 13, the delay obtained by the basic EDCF increases faster than AEDCF and SD scheme, while AEDCF always keeps a lower mean access delay less than 10 ms. We can also note that AEDCF has a mean delay 30% less than the static slow decrease scheme when the load rate reaches 149%.

In Figure 7, we plot the gain on goodput as a function of the traffic load of AEDCF and SD. We observe that the goodput gain of AEDCF increases when the traffic load increases. It reaches about 28% when the load rate is about 119% (i.e. for 35 stations). Moreover, the goodput of AEDCF is 10% higher than the SD scheme when the load rate is 149%. Indeed, AEDCF is much more efficient during high load rates.

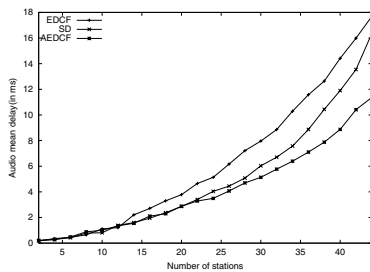


Fig. 6. Mean Delay

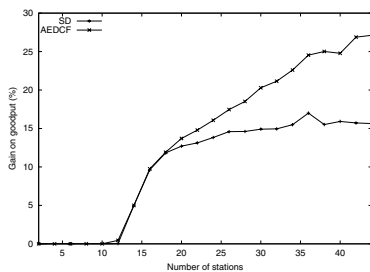


Fig. 7. Gain on goodput

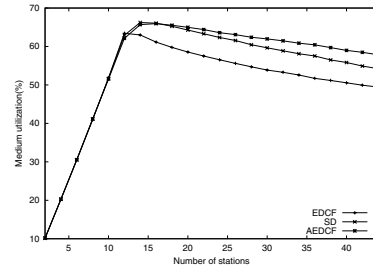


Fig. 8. Medium utilization

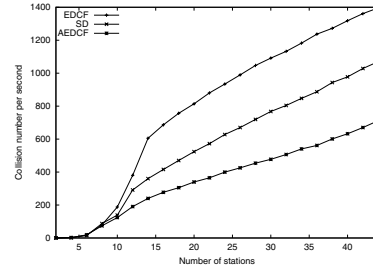


Fig. 9. Collision rate

Figure 8 shows the medium utilization as a function of the traffic load. For the three schemes the medium utilization decreases when the traffic load increases. However, our scheme achieves better medium utilization than the basic scheme whatever the number of stations.

The obtained collision rate is shown in Figure 9. The collision rates achieved by both schemes are similar when the traffic load is low, i.e. the number of stations is less than 8. However, when the traffic load increases, AEDCF is able to maintain a lower collision rate than EDCF and SD schemes. We can explain this behavior by the fact that we use an adaptive technique to change the contention windows according to the collision rate. The reduction of collision rate of AEDCF leads to significant goodput improvement and delay decrease. Moreover, AEDCF achieves an efficient tradeoff between collision decrease and idle time increase.

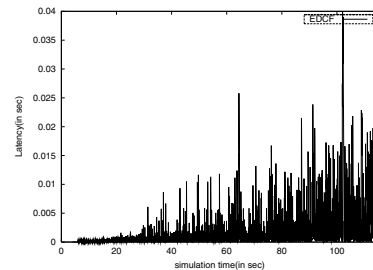


Fig. 10. The audio delay for EDCF

We have done a different simulation to study performance on delay and jitter. This new experiment has the same topology than Figure 3, but the number of stations is increased from 2 (4 sec) to 25 (100 sec) and the simulation stops at 115 sec. The delay variations of both EDCF and AEDCF schemes are

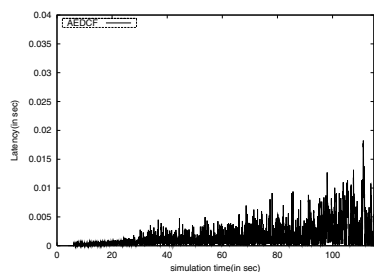


Fig. 11. The audio delay for AEDCF

plotted in Figures 10 and 11, respectively. AEDCF meets the target by maintaining the delay lower than EDCF and stable during the live of audio sessions. However, we can note that both delay and jitter for EDCF are twice higher than AEDCF, which degrade the quality of audio.

We show the latency distribution for each TC in Figures 12 and 13, in which a fixed number of 25 stations is used to show the delay performance. On a cumulative distribution plot, an ideal result would coincide with the y-axis, representing 100% of results with zero latency. Although we cannot reasonably expect zero latency, we would like to obtain constant performance, corresponding to a vertical line.

Figure 12 shows that AEDCF always outperforms EDCF. Note that the maximum delay of audio packets of AEDCF is less than 20ms, whereas for EDCF, the maximum value is more than 30ms.

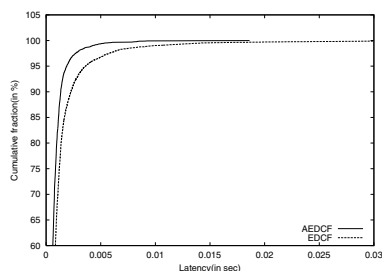


Fig. 12. Latency distributions of audio

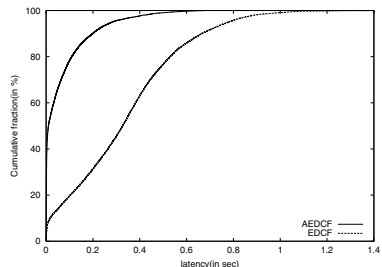


Fig. 13. Latency distributions of video

Figure 13 shows the latency measured for video (medium priority) traffic for both AEDCF and EDCF. There are considerable differences between them: more than 85% of video packets for AEDCF have delay less than 200ms, whereas

only 30% of video packets for EDCF have delay less than 200ms. These results are due to incorrect fast decreases of the  $CW$ s which increase the collision probability for next packet transmissions and thus increase end-to-end delays. Since our scheme outperforms EDCF especially at high load, we believe that the adaptive slow decrease of the  $CW$  is very efficient to estimate the network status and reduce collisions.

From the simulations, we can conclude that both AEDCF and the static slow decrease scheme outperform EDCF. Using an adaptive slow decrease, AEDCF get much higher goodput than the SD scheme. Moreover, the AEDCF scheme can improve the performance of all types of traffics.

#### IV. CONCLUSION

Our main contribution in this paper is the design of a new adaptive scheme for Quality of Service enhancement for IEEE 802.11 WLANs. We extend the basic 802.11e EDCF scheme by dynamically varying the contention window of each active class of service. Simulation results demonstrated that our scheme achieves better performance of throughput, delay and jitter. We validate our results by analyzing the impact of sources and network dynamics on the performance metrics and compare the results obtained with the basic EDCF and the static slow decrease schemes. Although AEDCF is intended to improve performance of wireless ad-hoc networks, the same idea can be used in the infrastructure mode with some changes. Future works could include adapting other parameters such as  $CW_{max}$ , the maximum number of retransmissions and the packet burst length according to the network load rate.

#### ACKNOWLEDGMENTS

The authors would like to thank Imad Aad for early discussions concerning slow decrease schemes.

#### REFERENCES

- [1] I. Aad and C. Castelluccia, "Differentiation mechanisms for IEEE 802.11", *IEEE Infocom*, Apr. 2001.
- [2] I. Aad and C. Castelluccia, "Remarks on per-flow differentiation in IEEE 802.11", *Proc. of European Wireless*, Feb. 2002.
- [3] H.S. Chahalaya and S. Gupa, "Throughput and fairness properties of asynchronous data transfer methods in the IEEE 802.11 MAC protocol", *PIMRC*, 1995.
- [4] K.C. Chen, "Medium access control of wireless LANs for mobile computing", *IEEE Network*, Vol. 8, No. 5, pp. 50-63, Sep. 1994.
- [5] J. Deng and R. S. Chang, "A priority Scheme for IEEE 802.11 DCF Access Method", *IEICE*, Vol. 82-B, No. 1, Jan. 1999.
- [6] A. Lindgren, et al., "Evaluation of Quality of Service Schemes for IEEE 802.11", *In Proc. of IEEE LCN*, Nov. 2001.
- [7] S. Mangold, et al., "IEEE 802.11e Wireless LAN for Quality of Service", *European Wireless*, Feb. 2002.
- [8] Q.Ni, et al., "QoS issues and enhancements for IEEE 802.11 Wireless LAN", *INRIA Research Report* No. 4612, Nov. 2002.
- [9] J.L. Sobrinho and A.S. Krishnakumar, "Real-time traffic over the IEEE 802.11 medium access control layer", *Bell Labs Technical Journal*, 1996.
- [10] N. H. Vaiday, et al, "Distributed Fair Scheduling in a Wireless LAN", *ACM Mobicom*, Aug. 2000.
- [11] A. Veres, et al., "Supporting service differentiation in wireless packet networks using distributed control", *JSAC*, Vol. 19, No. 10, Oct. 2001.
- [12] "AEDCF", see "<http://www-sop.inria.fr/planete/qni/Research/AEDCF/>".
- [13] IEEE WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", *IEEE 802.11 Standard*, 1999.
- [14] IEEE WG, "802.11a, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", Sep. 1999.
- [15] IEEE WG, "802.11e Draft 3.1", May 2002.