This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

A Novel Access Method for Supporting Absolute and Proportional Priorities in 802.11 WLANs

Mohammad Nassiri, Martin Heusse, and Andrzej Duda
Grenoble Informatics Laboratory
Grenoble, France
Email: {nassiri, heusse, duda}@imag.fr

***Abstract*—Many researchers have addressed the problem of QoS differentiation in 802.11 wireless networks, however no method proposed so far benefits from all desirable properties: high aggregate throughput even for a large number of contending stations, fair allocation to all stations in the same class, fast adaptation to changing conditions, and support for absolute priorities. If we consider the IEEE 802.11e standard, its EDCA (Enhanced Distributed Channel Access) access method suffers from an increased collision rate when the number of stations increases.**

In this paper, we propose a novel access method that supports both *relative proportional* throughput allocation and *absolute priorities* in 802.11 wireless networks. The method is efficient, scalable, and fair. It builds on the idea of the *Idle Sense* method that provides the optimal throughput and fairness for 802.11 WLANs [1]: each station adjusts its contention window based on the observed average number of idle slots. We achieve absolute priority differentiation by setting the target value for the number of idle slots to a small value, so that the absolute priority class gains all the available throughput. The method also supports relative proportional throughput allocation in which several classes share the available throughput according to desired ratios.

Our simulations show that the proposed method achieves its objectives of relative and absolute differentiation both with respect to the aggregated throughput and the speed of convergence. Unlike 802.11e EDCA, it presents very good scalability—the throughput remains almost constant in function of the number of contending stations.

I. Introduction

The IEEE 802.11 standard for wireless LANs defines the basic access method called DCF (*Distributed Coordination Function*) that provides approximately equal channel access probability to all devices operating on the same channel in a given radio range. Thus, if there are N wireless stations sharing the radio channel, each of them benefits from $1/N$ channel access probability.

For many multimedia communication applications, fair sharing of the radio channel is not enough. To achieve a good level of perceived quality, multimedia traffic requires some QoS (Quality of Service) support that guarantees parameters such as a minimal throughput, a maximal delay, and a bounded jitter. Supporting QoS guarantees in wireless networks is not easy, because air medium presents a complex and time-varying behavior: path loss, signal fading and interference, time dispersion, which result in high throughput variability and significant error rate. When channel conditions are good and communicating devices are within a sufficient radio range, QoS characteristics mainly depend on the channel access method. To deal with multimedia traffic, many authors have proposed various priority access schemes to guarantee some QoS parameters [2]–[6].

The IEEE 802.11e defines an extension to the standard 802.11 operation to provide channel access differentiation for several traffic classes through new coordination functions [2]. HCF (Hybrid Coordination Function) enhances PCF (Point Coordination Function) by allowing a centralized coordinator to allocate TXOP (Transmission Opportunity) periods at any time in a polling based way. Such an access method can be implemented in an access point to operate in the infrastructure mode, however it would be difficult to use it in the ad hoc mode, because of the need for centralized coordination.

EDCA (Enhanced Distributed Channel Access) keeps the contention based operation of DCF, but extends it to give more transmission opportunity to higher priority classes. The principle of EDCA is to set different access parameters (interframe period, contention window, persistency factor, TXOP duration) for different traffic classes. Even if EDCA provides a first valuable support for QoS differentiation in 802.11 networks, it suffers from several performance problems largely reported in the literature [5]–[7].

Usually, we can achieve QoS differentiation in several ways. The first one is *relative*—we can allocate different parts of the available throughput to several traffic classes. In *proportional throughput allocation*, one class benefits from a greater throughput than another one, the proportion of the throughputs being defined by a given ratio. Such relative priorities are suitable for assigning relative weights to different types of traffic, e.g. interactive sessions such as instant messaging or remote login with respect to Web access or bulk data transfers. The advantage of relative priorities is that they do not lead to the starvation of low traffic classes when high priority traffic becomes important. However, applications that use relative priorities need to live with a possible degradation of their QoS parameters when the traffic in a given class increases.

The second way of defining QoS differentiation is *absolute*—a high priority class always benefits from the available throughput even if there is some low priority traffic. Voice-over-IP, videoconferencing, or live video broadcast are examples of applications that may require absolute priorities. In this approach, high priority traffic captures all the available throughput so that the low priority class may starve. To guarantee some QoS parameters, e.g. a minimum throughput, absolute priorities require some admission control to limit the high priority traffic so that it does not exceed the available capacity.

In this paper, we propose a novel access method that supports both relative proportional throughput allocation and absolute priorities in 802.11 wireless networks. It has all desirable properties: high aggregate throughput even for a large number of contending stations, fair allocation to all stations in the same class, fast adaptation to changing conditions, and support for absolute priorities in addition to relative proportional allocation. We build upon the idea of the *Idle Sense* method that provides the optimal throughput and fairness for 802.11 WLANs [1]. In *Idle Sense*, each station adjusts its contention window based on the observed average number of idle slots. The value of contention windows increases with the number of active stations, which results in less collisions. The original version of *Idle Sense* provides fair sharing of the radio channel with better fairness compared to 802.11 DCF. The method proposed in this paper achieves absolute priority differentiation by setting the target value for the number of idle slots to a small value, so that the absolute priority class gains all the available throughput. The proposed method also supports relative proportional throughput allocation in which several classes share the available throughput according to desired ratios. We define how stations need to adjust their contention windows to achieve relative differentiation. We keep the definition of traffic classes compatible with the IEEE 802.11e standard.

We validate our method with ns-2 simulations that show how the proposed method achieves its objectives of relative and absolute differentiation both with respect to the aggregated throughput and the speed of convergence. Unlike 802.11e EDCA, it presents very good scalability—the throughput remains almost constant in function of the number of contending stations.

The remainder of this paper is organized as follows. Section II gives an overview of relevant access methods: 802.11 DCF, *Idle Sense*, and 802.11e EDCA. Section III briefly describes the related work. We present our method that supports *proportional throughput allocation* and *absolute priorities* in Sections IV and V. Section VI shows simulation results and comparisons with 802.11e EDCA. Finally, Section VII concludes the paper.

II. Overview of relevant access methods

Before presenting our proposal, we briefly review 802.11 DCF [8], *Idle Sense* [1], and 802.11e EDCA [2] access methods.

A. 802.11 DCF

802.11 DCF uses the *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA) principle: before initiating a transmission, a station senses the state of the channel. If the medium is sensed busy, the station waits until the channel is free during a *Distributed Interframe Space* (DIFS) interval, afterwards, it waits for an additional random contention time. The station chooses a backoff time that is an integer number of time slots distributed uniformly in the contention window $[0, CW-1]$; if another transmission occurs during this procedure, the residual backoff is kept for the next contention period. The value of CW is set to $CW_{\min}$ for the first transmission attempt and it is increased in integer powers of 2 at each failed transmission (collision or frame loss) up to $CW_{\max}$ (*exponential backoff mechanism*).

B. Idle Sense

Idle Sense optimizes 802.11 DCF for high throughput and fairness: contending stations do not perform the exponential backoff algorithm after collisions or failed transmissions, rather they make their contention windows dynamically converge in a fully distributed way to similar values solely by tracking the number of idle slots between consecutive transmissions. The method works as follows: each station measures n_i, the number of consecutive idle slots between two transmission attempts. Every $maxtrans$ transmissions, it estimates $\hat{n_i}$, the average of observed values of n_i. Then, it uses $\hat{n_i}$ to adjust its contention window to the target value $n_{\mathrm{i}}^{\mathrm{target}}$ computed numerically for a given variant of IEEE 802.11 PHY and MAC parameters—its value is 5.68 for IEEE 802.11b and 3.91 for IEEE 802.11g [1]. When stations adjust their CW so that n_i converges to ${n_i}^{\mathrm{target}}$, their throughput is optimal.

The *Idle Sense* adaptation algorithm makes $\hat{n_i}$ converge to n_i^{target} by applying AIMD (*Additive Increase Multiplicative Decrease*) [9] to contention window CW as follows[1]:

- If $\hat{n_i} \geq n_i^{target}, CW \leftarrow \alpha \cdot CW$
- If $\hat{n_i} < n_i^{target}, CW \leftarrow CW + \epsilon$

where ϵ and α are some adaptation parameters. If a station observes too many idle slots compared to the target, it needs to increase CW additively, which in turn will decrease n_{i}, whereas if it observes too few idle slots, it needs to decrease CW in a multiplicative way, which in turn will increase n_{i}.

We use the following values of the adaptation parameters, because they yield the best balance between accuracy and convergence for any number of contending wireless stations [10]:

- $\frac{1}{\alpha} = 1.0666$
- $\epsilon = 6.0$

C. 802.11e EDCA

IEEE 802.11e EDCA extends DCF for channel access differentiation. It offers eight traffic priorities mapped into four default access categories (AC): voice (AC_VO), video (AC_VI), best-effort (AC_BE), and background (AC_BK). AC_VO has the highest priority.

For each access category, an Enhanced Distributed Channel Access Function (EDCAF) contends separately for the

[1]The original adaptation algorithm applied AIMD to transmission attempt probability P_{e} [1]. When implementing *Idle Sense*, we have thoroughly tested and evaluated this approach and finally modified the adaptation algorithm so to apply AIMD to contention windows instead of transmission attempt probabilities [10].

channel. The method inherits the principles of the contention phase from DCF, but EDCAF defines its own parameter set for each access category: AIFS durations (Arbitration Inter-Frame Spaces), the minimum $CW_{\min}$ and maximum $CW_{\max}$ contention windows, and TXOP transmission opportunity limit. Like in DCF, the EDCA function of AC_j resets the value of CW_j to CW^j_{min} after a successful transmission, while after a collision or frame loss, it increases this value exponentially up to $CW_{\max}$. EDCA makes DIFS periods dependent on an access category: AIFS of a given AC is the interval during which the channel has to be idle before initiating a transmission. To support priorities between access categories, AIFS for a higher priority AC is shorter or equal to AIFS of a lower priority AC. TXOP is another parameter of each access category: a station that wins contention can exclusively use the channel during the TXOP period so it may transmit one or more data frames separated by SIFS intervals.

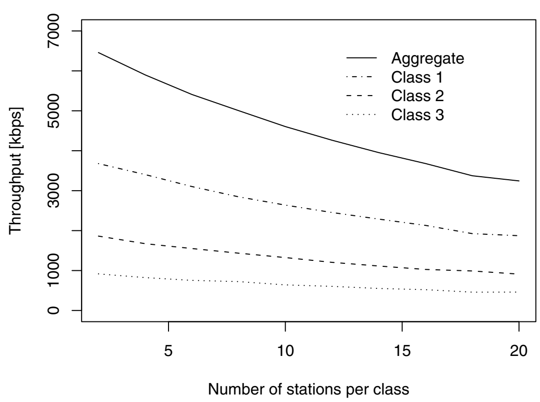

Fig. 1. Aggregate throughput per class for 802.11e EDCA, 802.11b PHY/MAC parameters.

EDCA suffers from several performance problems [5]–[7]. In particular, it does not perform well when the number of competing stations increases, because the collision rate also increases so that the total network throughput drops. Figure 1 presents an example of such a bad performance—we can observe a significant decrease in the aggregate throughput per class in a simulation of proportional differentiation with three classes (minimum and maximum contention window are $CW_1 \in [16, 48]$ for class 1, $CW_2 \in [31, 93]$ for class 2, and $CW_3 \in [61, 183]$ for class 3).

One may think that 802.11e EDCA can provide an absolute differentiation in addition to the relative one. However, it is not the case: assigning a short AIFS to the absolute priority class and a long AIFS to the lower priority one does not result in absolute differentiation, because AIFS is followed by the contention backoff that still gives the lower priority class some transmission opportunity. Using short AIFS may only work, if the absolute priority class benefits from a very small $CW_{\min}$, but in this case, collision rate increases significantly for a larger number of absolute priority stations, which leads to a sharp drop in the aggregated throughput.

III. Related Work

There is a considerable amount of research on QoS differentiation for wireless LANs and ad hoc networks. We discuss here only a small subset of closely related work, especially we are interested in solutions at the MAC layer that aim at enhancing the 802.11e type of wireless networks. We focus on distributed access methods, because they can work in the infrastructure mode as well as in an ad hoc setting.

Pattara-Atikom *et al.* proposed a survey of distributed MAC schemes that support QoS in 802.11 networks [5]. They categorize access methods into two groups: priority oriented and fair scheduling based. The priority schemes [2], [11], [12] provide service differentiation by allowing privileged access to the channel for traffic classes with higher priority while mechanisms based on fair scheduling aim at allocating weighted fair share of throughput among different demands [4], [13]. In all the above mechanisms, QoS is usually provided by tuning access parameters like IFS (Inter-Frame Spaces) and contention windows. The authors also compared three approaches based on fair scheduling: Distributed Weighted Fair Queueing (DWFQ) [13], Distributed Fair Scheduling (DFS) [4], and their own proposal—the Distributed Deficit Round Robin (DDRR) [14] based on the concept of the Deficit Round Robin (DRR) [15]. In DDRR, each traffic class determines the allotted service quantum rate based on its throughput requirements and maintains a deficit counter of accumulated quanta. The deficit counter decreases by the size of a transmitted frame and a traffic class can transmit only when the counter is positive.

Qiao *et al.* proposed a Priority-based Fair Medium Access Control protocol (P-MAC) to maximize the wireless channel utilization subject to weighted fairness among multiple traffic flows [3]. The basic idea of P-MAC is to optimize the value of the contention window for each wireless station to reflect the relative weights among traffic flows as well as the number of stations contending for the wireless medium. The proposed method is sensitive to the convergence to a desirable operating point. Hu *et al.* have shown that P-MAC may not converge even in the presence of only a single traffic class [6].

Hu *et al.* proposed *MAC Contention Control* (MCC) to achieve proportional throughput allocation [6]. The proposal does not modify the original IEEE 802.11e EDCA, but rather it adds a thin layer above MAC that adjusts the rate of dequeuing frames in function of relative priorities. MCC dequeues frames from layer 3 according to an AIMD algorithm based on one of two MAC layer channel state indicators: the average number of collisions between successful transmissions and the number of idle slots between transmission attempts. The first indicator has major drawback of being dependent on the channel error rate (recall that stations cannot distinguish between a collision and a failed transmission). Moreover, the authors admit that it significantly varies with changing traffic characteristics, which makes it a poor reference. The second indicator has already been used as the basis of the Asymptotically Optimal Backoff (AOB) [16] and *Idle Sense* access methods [1]. The adaptation algorithm of MCC also uses the AIMD algorithm for adjusting the transmission attempt rate similarly to *Idle Sense*. As MCC operates above an unchanged 802.11e layer, it fails to maintain both high reactivity and good channel efficiency: performance evaluation done by the authors [6] shows that the adaptation time is of the order of seconds and the aggregated throughput decreases in function of the number of stations. In particular, the simulation results show two cases: for small AIMD parameters, MCC does obtain a better aggregate throughput than 802.11e, but responds very slowly to the network dynamics. For large AIMD parameters, MCC becomes more responsive to the network dynamics, but it does not attain high aggregate throughput.

Although many authors have addressed the problem of QoS differentiation in 802.11 wireless networks, no method proposed so far offers all the desirable characteristics at the same time: high aggregate throughput for an increasing number of contending stations, fair allocation to all stations in the same class, fast adaptation to changing conditions, and support for absolute priorities.

IV. Relative proportional throughput allocation

In our proposed method, each station manages one class with an absolute priority (denoted by index 0) and M lower priority classes (with an index from 1 to M) that benefit from relative proportional throughput allocation. Proportionally means that throughput shares follow some predefined ratios. In this section, we focus on proportional allocation, so we assume at the beginning that there is no traffic with absolute priority.

Our first goal is thus to proportionally allocate throughput to different classes. The mechanism needs to achieve this goal even in the presence of a large number of contending stations. The good properties of *Idle Sense* [1] suggest that by building on its principles and defining a differentiation method, we can benefit from its optimal behavior for a large range of contending stations. The idea for the differentiation method comes from the relation between the channel access probability and the contention window: if class i wants to obtain for example a double throughput of class j, it needs to benefit from channel access probability, which is double compared to the probability of class j. This condition translates into a relation between contention windows: *CW* of class i should be a half of *CW* for class j. Unlike 802.11e, all traffic classes in our method use the same inter-frame space (DIFS) before choosing a backoff from the contention window.

To derive the contention windows leading to proportional allocation, we consider that each station uses the *Idle Sense* access method to adjust CW_{ref}, the reference contention window. CW_{ref} corresponds to the fair throughput share each station obtains when there is no absolute priority traffic. Every $maxtrans$ transmissions, the *Idle Sense* access method adjusts CW_{ref} to the predefined target value n_i^{target} as in the original *Idle Sense* method. The value of CW_{ref} varies in function of the number of active stations so that they obtain their fair shares of throughput. This part of throughput per station is then distributed over M traffic classes according to desired ratios: class i, resp. class j, uses contention window CW_i, resp. CW_j to obtain throughput X_i, resp. X_j, so that

$$\frac{X_i}{r_i} = \frac{X_j}{r_j}, \tag{1}$$

where r_i and r_j are the throughput ratios of traffic classes i and j, $0 \leq r_i, r_j < 1$ for $i \in \{1, ...M\}$ and $j \in \{2, ...M\}$. We assume $r_1 = 1$ for the highest relative priority class.

We show below how to compute CW_j to obtain the required allocation defined by Eq. 1. If all contending entities send frames of the same size, Eq. 1 translates into the following relation between the probabilities of a successful transmission P_{e} of class i and j:

$$\frac{P_{\text{t}}^i}{r_i} = \frac{P_{\text{t}}^j}{r_j}. \tag{2}$$

A transmission of class i succeeds, if a station attempts to transmit with probability P_{e}^i and no collision occurs, which is the case if no other station attempts to transmit in the same slot. Thus, we have:

$$P_{\text{t}}^i = P_{\text{e}}^i \ \frac{1}{1 - P_{\text{e}}^i} \ \prod_{k=1}^{M} (1 - P_{\text{e}}^k)^{N_k},$$

where N_k is the number of stations in class k and $\prod_{k=1}^{M}(1 - P_{\text{e}}^k)^{N_k}$ is the probability that a slot is idle. Thus, from Eq. 2, we obtain:

$$\frac{P_{\text{e}}^i}{r_{\text{i}}} \ \frac{1}{1 - P_{\text{e}}^i} = \frac{P_{\text{e}}^j}{r_{\text{j}}} \ \frac{1}{1 - P_{\text{e}}^j}$$

Typically, the slot time is as small as possible and transmissions are rare to avoid collisions, so that $\forall i, P_{\text{e}}^i \ll 1$ and thus:

$$\frac{P_{\text{e}}^i}{r_i} \approx \frac{P_{\text{e}}^j}{r_j}. \tag{3}$$

When stations do not perform exponential backoff mechanism after collisions, the transmission attempt probability is as follows according to the Bianchi model [17]:

$$P_e = \frac{2}{CW + 1}. \tag{4}$$

The reference contention window CW_{ref} corresponds thus to the aggregate transmission attempt probability of multiple traffic classes in one station P_e^{ref} that we want to distribute over M classes:

$$\sum_{i=1}^{M} P_e^i = P_e^{\text{ref}} \tag{5}$$

This yields the following expression for P_{e}^1:

$$P_{\text{e}}^1 = \frac{P_{\text{e}}^{\text{ref}}}{\sum_{i=1}^{M} r_i} \tag{6}$$

and we obtain:

$$P_{\text{e}}^j = \frac{r_j P_{\text{e}}^{\text{ref}}}{\sum_{i=1}^{M} r_i}. \tag{7}$$

Knowing P_e^j we can derive CW_j to achieve the allocation defined by Eq. 1:

$$CW_j = \frac{\sum_{i=1}^{M} r_i}{r_j}(CW_{\text{ref}} + 1) - 1. \quad (8)$$

The proposed access method updates the values of CW_j when it computes the new value of CW_{ref}. Algorithm 1 in the next section specifies more formally the proposed method after the discussion of the absolute priority part.

Note that in addition to providing relative differentiation between several traffic classes, the proposed method can provide good fairness to the traffic in the same priority class: stations just behave as in the original *Idle Sense* method with contention values CW_j converging to the same target value thus inheriting the good fairness property of *Idle Sense*.

V. Absolute Priority Access

Besides proportional throughput allocation, we want to provide *absolute priority* access that enables a high priority class to always benefit from the available throughput even if there is some low priority traffic. In other words, we want that as long as there are packets to send in the class with the absolute priority, low priority traffic is suspended. Moreover, traffic of absolute priority coming from different stations benefits from fair allocation of the channel capacity. If there is no traffic with the absolute priority, M lower priority classes share the available throughput proportionally.

Our idea is to exploit another possibility offered by the *Idle Sense* access method: assign a small value of n_i^{target} to the absolute priority class. Thus, to offer both proportional throughput allocation and absolute priorities, we define two target values:

- $n_i^{target} = n_i^a$ for the absolute priority traffic class,
- $n_i^{target} = n_i^p, n_i^p < n_i^a$ for all classes that benefit from proportional throughput allocation.

The superscripts a and p refer here to the absolute and proportional priority access, respectively. If n_i^a is smaller than n_i^p, the contention window of the absolute priority traffic converges to a much smaller value than the contention window of low priority classes. At the same time, the adaptation mechanism of *Idle Sense* adjusts the small contention window of the absolute priority traffic so that all stations with this priority class benefit from a fair share of the available throughput.

There is still one issue left: how to choose a suitable target value of n_i^a and n_i^p? Proportional throughput allocation needs to obtain the whole available capacity in the absence of the absolute traffic. So the target of proportional priorities n_i^p should be the target of the basic version of *Idle Sense*, that is 5.68 for IEEE 802.11b and 3.91 for IEEE 802.11g. For the absolute priority class, we need to choose a smaller target value, but not too small, because of an increasing collision rate.

To choose the right value of n_i^a, we analyze the conditional probability $P[\text{coll}|\ \text{transmission}]$ that a transmission attempt results in a collision, which corresponds to the collision probability. This analysis is fairly general and holds for any variant of IEEE 802.11. Recall that for a large number of stations N, the network attains the optimal throughput when the transmission attempt probability in a given slot P_{e} is such that $NP_{\text{e}} = \zeta$, where ζ is a constant [1]. Furthermore, we have $\zeta = \log(\frac{1}{n_{\text{i}}} + 1)$; $P_{\text{t}} = NP_{\text{e}}(1 - P_{\text{e}})^{N-1} \approx \zeta e^{-\zeta}$; $P_{\text{i}} = (1-P_{\text{e}})^N \approx e^{-\zeta}$; $P_{\text{c}} = 1-P_{\text{t}}-P_{\text{i}} \approx 1-(\zeta-1)e^{-\zeta}$ [1]. Finally, the collision probability is the following:

$$P[\text{coll}|\ \text{transmission}] = \frac{P_{\text{c}}}{P_{\text{c}} + P_{\text{t}}} \approx 1 - n_{\text{i}} \log(\frac{1}{n_{\text{i}}} + 1) \quad (9)$$

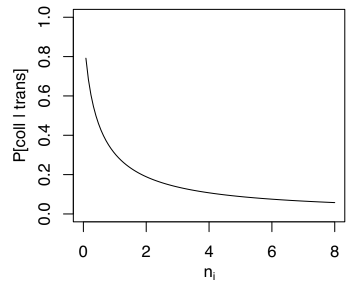

Fig. 2. Conditional collision probability as a function of the target number of idle slots.

Figure 2 presents the collision probability in function of the target number of idle slots. We can observe that it steeply increases for low values of n_i. We have observed in our simulations that the choice of $n_i^a = 3$ for 802.11b leads to a high network throughput, while the collision probability is still low as it can be seen in Figure 2. For 802.11g, our simulations show that the best choice is $n_i^a = 2.3$.

Algorithm 1 *Priority Idle Sense*

```
maxtrans ← 5 ; sum ← 0 ; ntrans ← 0
sumratio ← r_1 + ... + r_M
After each transmission {
/* Station observes n_i idle slots before a transmission */
sum ← sum + n_i
ntrans ← ntrans + 1
if (ntrans >= maxtrans) then
   /* Compute the estimator */
   n̂_i ← sum/ntrans
   /* Reset variables */
   sum ← 0
   ntrans ← 0
   A. Adjust CW_0 to n_i^a with respect to n̂_i
   B. Adjust CW_ref to n_i^p with respect to n̂_i
   /* Update contention window of low priority classes */
   for j = 1 to M do
      CW_j ← (sumratio / r_j)(CW_ref + 1) − 1
   end for
end if
}
```

Algorithm 1 formally specifies our access method that provides both proportional throughput allocation and absolute priority access. The adaptation mechanism adjusts the contention window of absolute priority class CW_a to the target value n_i^a and the reference contention window CW_{ref} to n_i^p. The effect of this control is a fast increase of the contention

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

window for lower priority classes when the absolute traffic appears. Conversely, the contention window needs to decrease fast when the absolute traffic disappears. To obtain the fast decrease, we need to limit the upper bound $CW_{\max}$ of lower priority classes. This limitation is not presented in the pseudo-code of the algorithm for the sake of conciseness. Notice also that the adaptation of contention window (lines A and B) follows the AIMD mechanism described in section II-B.

Similarly to proportional allocation, our method for absolute differentiation also can provide good fairness when several stations generate traffic with absolute priority. This results from the properties of the *Idle Sense* method—the station adjust their contention windows CW_0 to the same target value so that their long-term average converge to the same value, which results in almost the same throughput.

In this paper we focus on providing a single absolute priority class, although the method can be extended to accommodate several absolute priority levels using smaller and smaller n_i^a target values. Nevertheless, this may lead to an increasing collision rate and needs further investigation.

VI. Performance Evaluation

In this section, we evaluate our access method by means of simulation. We have implemented it in the NS2 network simulator version 2.29 [18] and compare to the 802.11e EDCA implementation by Wiethoelter *et al.* [19]. We report on comparisons with 802.11e, because we have designed our method to be its enhancement. We do not compare with MCC, because it places its functionality above the standard 802.11e MAC layer and its reported performance is not on a par in terms of convergence speed [6].

Our numerical examples use the MAC and PHY parameters of IEEE 802.11b. We assume ideal channel conditions, i.e. frame losses are only due to collisions. We analyze saturated conditions in which an active station has always a data frame of 1500 bytes to send. We set the transmission and carrier sensing ranges to 160 m and 400 m, respectively. We place stations randomly in a rectangle of 100 m by 100 m so they are all in the transmission range. To evaluate the performance only at the MAC layer, we use a static routing agent without any other traffic sources such as ARP. We present simulation results without the RTS/CTS exchange. Each simulation runs for 30 seconds during which approximately 15000 frames are transmitted.

We use several performance metrics to evaluate our method: *aggregate throughput*, *per class throughput* and *collision rate*.

A. Proportional Throughput Allocation

In this section, we evaluate our method with respect to proportional throughput allocation in several scenarios. To only focus on the properties of the relative priority scheme, we assume that there is no traffic with absolute priority.

1) One active class in half of the stations: In this scenario, we consider two traffic classes and set the desired throughput ratios to $r_1 = 1$, $r_2 = 0.5$. Only one class is active in a given station: a half of the stations generate traffic of class 1 while the other half sends class 2 traffic. To obtain this allocation, we set the following parameters of 802.11e: $CW_1 \in [16, 48]$ for class 1, $CW_2 \in [31, 93]$ for class 2.

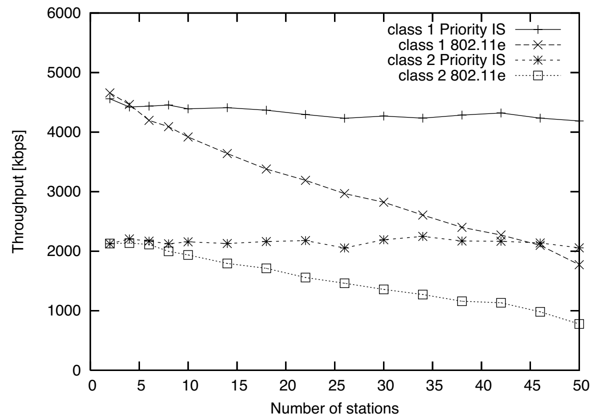

Fig. 3. Aggregate throughput per class *vs.* number of stations, proportional throughput allocation. Half of stations generate class 1 traffic while the other half send class 2 traffic.

We can expect that the aggregate throughput of stations with class 2 traffic being a half of the class 1 throughput, because of the chosen throughput ratios and the equal number of sources in each class. We can see in Figure 3 that this is effectively the case. Moreover, we can observe that the aggregate throughput does not depend on the number of contending stations, which is the sign of good efficiency. 802.11e presents an opposite behavior—the aggregate throughput decreases.

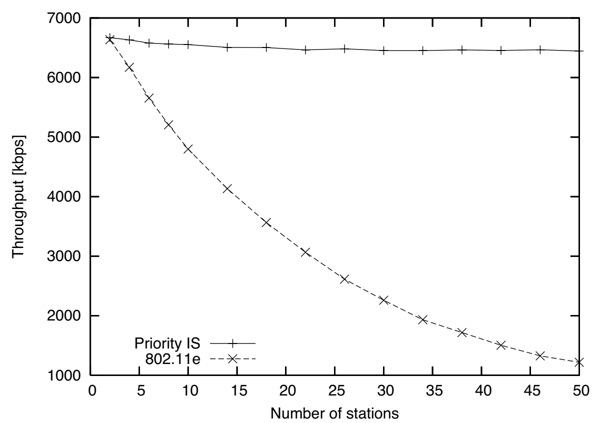

Fig. 4. Aggregate throughput *vs.* number of stations. Proportional throughput allocation to stations with three active classes.

2) Three active classes per station: In this scenario, all stations generate traffic of three classes: 1, 2, and 3. We set the desired throughput ratios for different traffic classes in a station to $r_1 = 1$, $r_2 = 0.5$, and $r_3 = 0.25$, so that the aggregate throughput of class 2 should be a half of what obtains class 1, while class 3 traffic needs to obtain one quarter of the class 1 throughput. Figures 4 and 5 show simulation comparisons with 802.11e in this scenario. To achieve proportional throughput allocation in 802.11e, we set the appropriate values of the minimum and maximum contention window for different access

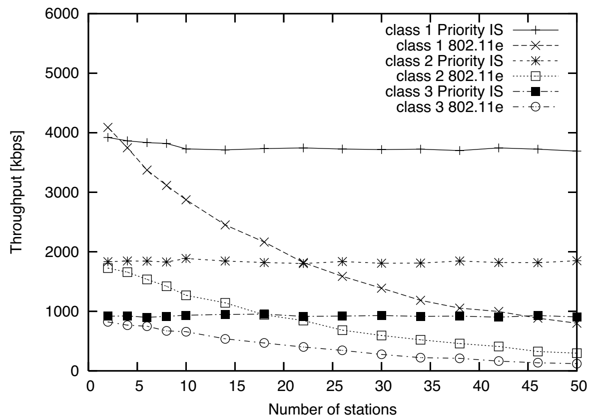

Fig. 5. Aggregate throughput per class *vs.* number of stations. Proportional throughput allocation to stations with three active classes.

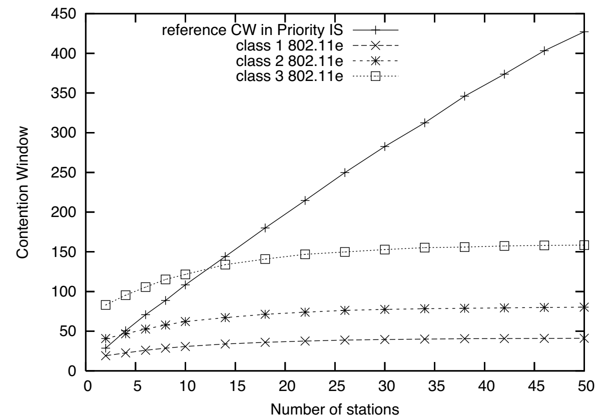

Fig. 7. Contention window *vs.* number of stations. Proportional throughput allocation to stations with three active classes.

categories: $CW_1 \in [16, 48]$ for class 1, $CW_2 \in [31, 93]$ for class 2, and $CW_3 \in [61, 183]$ for class 3 to obtain throughput allocation in the proportions defined for our method.

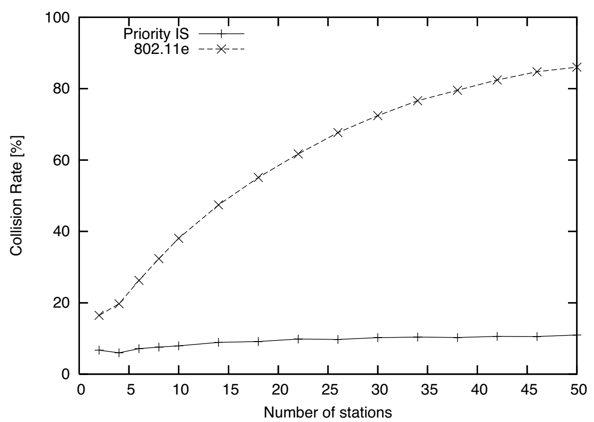

Fig. 6. Collision rate *vs.* number of stations. Proportional throughput allocation to stations with three active classes.

Figure 4 shows that our method provides consistently high aggregated throughput independently of their number. For 802.11e EDCA, the aggregated throughput quickly drops with the number of competing stations. We can also see in Figure 5 that for our method, the ratio of class 2 throughput to class 1 is 1/2 regardless of the number of active stations, which is our objective. The EDCA access method of 802.11e keeps the ratios right, but fails to achieve high aggregated throughput for an increasing number of stations. We can also see that the collision rate for 802.11e significantly increases with the number of stations (cf. Figure 6), which explains the degradation in aggregated throughput.

Figure 7 shows how the average contention window changes when the number of active stations increases. For the proposed mechanism, we just show the reference contention window CW_{ref} in the figure, because the contention window for each class is a linear function of CW_{ref}. We also present the average contention windows for all three 802.11e classes. We can observe that CW_{ref} adapts fairly well to the number of competing stations. The average contention windows in 802.11e do not increase sufficiently, which results in an increased collision rate shown in Figure 6.

Another well known aspect of EDCA is its rather poor short-term fairness. Figure 8 presents the Jain fairness index [20]. We use the sliding window method that observes the patterns of transmissions to compute the Jain index in a window of an increasing size. The smaller the window, the index represents the shorter term fairness with the perfect fairness being 1 (we normalized the window size with respect to the number of stations so that the window sizes are multiples of N).

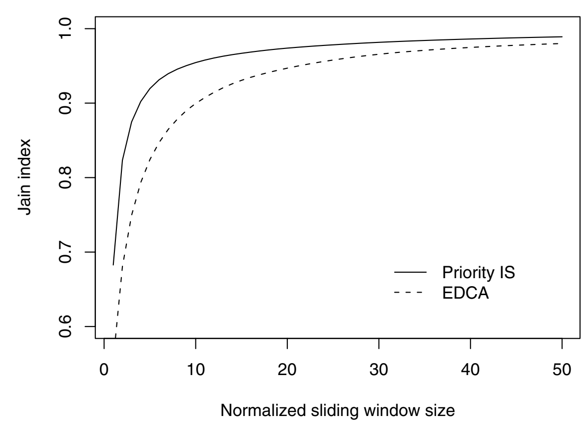

Fig. 8. Jain fairness index for 10 stations, each station sending traffic of all 3 classes (we only show the Jain index for the highest priority traffic, because it is almost the same for other classes).

B. Absolute Priority Access

To evaluate our support for absolute priority access, we use a simulation set up of 10 active stations each with two classes: one with an absolute high priority and the other with a lower priority. We are not only interested in observing the desired property of absolute traffic differentiation, but we also want to assess the speed of convergence of our proposed method. Thus, instead of only observing the stationary throughput obtained

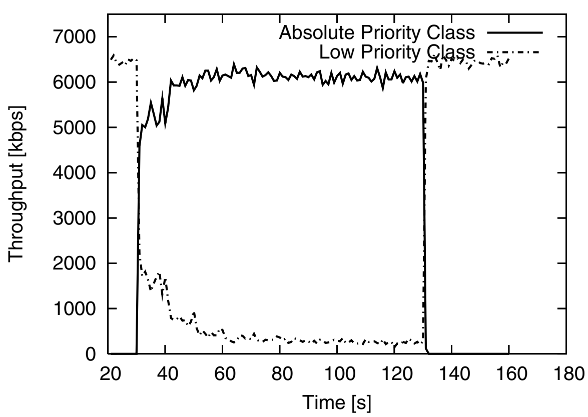

Fig. 9. Evolution of the aggregate throughput per class in time. 10 stations generate low priority traffic all of the time while one station joins the network every 10 seconds and starts sending absolute priority traffic.

by the respective priority classes, we consider a dynamically varying workload: 10 stations generate low priority traffic all of the time; one station with absolute priority class joins the network every 10 seconds beginning at instant 30 s and starts sending absolute priority traffic. Figure 9 shows the evolution of the aggregate throughput in time. We can observe that even if there is only one active station with absolute priority traffic, it obtains almost 90% of the channel throughput (5.1 Mbps) while all ten low priority classes share the remaining 10% of the throughput. When the second absolute class starts generating traffic, the low priority class becomes almost starved with very low throughput. At instant 130, the absolute priority traffic stops, so that the lower priority class quickly regains the full capacity of the network.

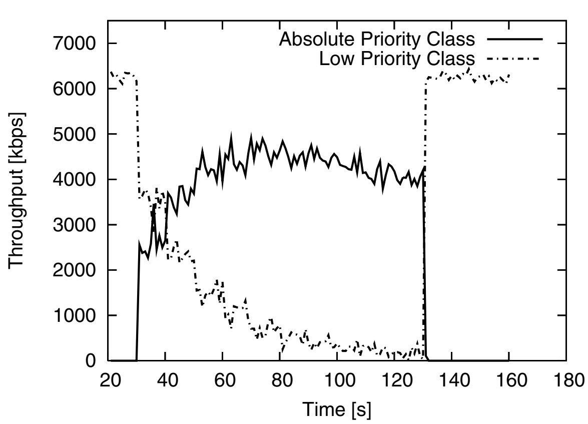

Fig. 10. Evolution of the aggregate throughput per class in time for 802.11e EDCA. The simulation scenario as in Figure 9.

Figure 10 shows the results of the same simulation for the 802.11e EDCA. We can see that the absolute priority class obtains much lower throughput compared to our method.

Figure 11 shows the evolution of the contention window for the absolute and relative low priority classes at a chosen station. We can see that when the station starts to send absolute

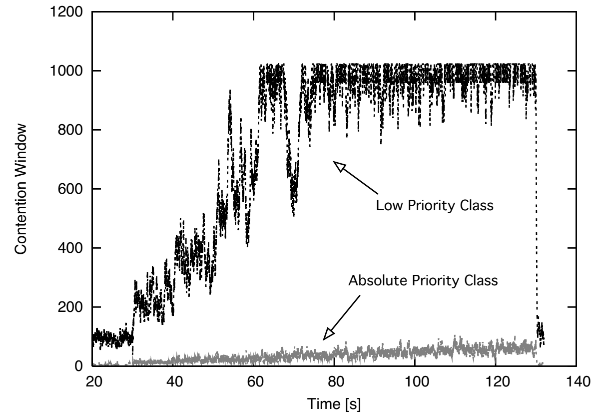

Fig. 11. Evolution of the contention window for the absolute and low priority classes in time.

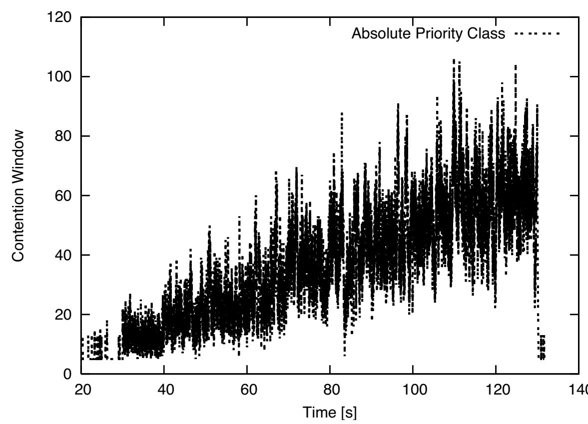

Fig. 12. Evolution of the contention window for the absolute class in time.

priority traffic, the contention window of the relative class sharply increases up to the limit of 1024, so that it cannot compete with the absolute priority class. Note also that the contention window of the absolute priority class increases to adapt to an increasing number of contending absolute priority stations. Figure 12 presents a zoom into the behavior of the contention window for the absolute priority class to show its increase in function of the number of contending stations in this class.

An important feature of our method is its rapid convergence. Figure 13 shows how quickly the contention window of the low priority class converges after the end of the absolute priority traffic: it takes roughly 300 ms to adjust to the appropriate value.

Finally, we have analyzed the collision rate when stations use our access method offering absolute priority (cf. Figure 14). It increases a little bit, because the target number of idle slots ($n_i^a = 3$) is smaller than the corresponding value used in *Idle Sense* (5.68). Nevertheless, it always increases very slowly with the number of active stations, which results in a very good aggregated throughput per class (cf. Figure 9).

By selecting different target values for absolute and proportional priorities, there is no need for using different AIFS as

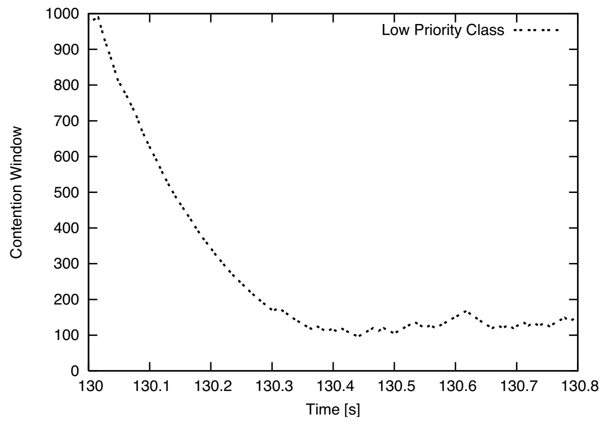

Fig. 13. Zoom into the behavior of the low priority class at instant 130.

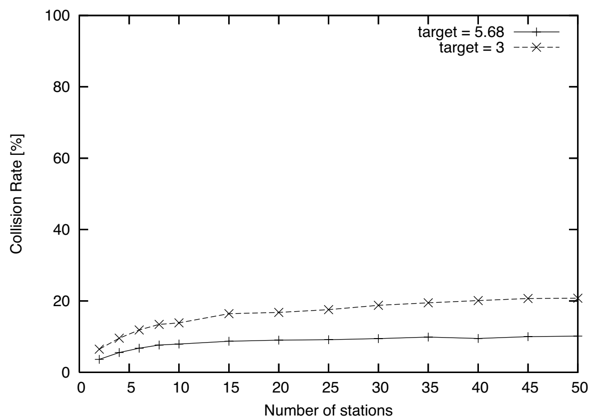

Fig. 14. Collision rate *vs.* number of stations for different values of the target number of idle slots n_i^{target}.

in 802.11e, because as our simulation results have shown, it is possible to proportionally allocate throughput as well as to provide absolute priority access only by adjusting contention windows. In particular, it is not necessary to use a larger AIFS for the low priority classes, because it leads to a lower channel utilization in the absence of absolute priority traffic.

Note that we have set the frame size of the absolute priority traffic to a large value of 1500 bytes to exhibit its contention with low priority traffic. For short frame traffic such as VoIP, we can observe a similar behavior with the absolute priority traffic gaining most of the channel capacity, but with a lower throughput due to an increased overhead. For proportional throughput allocation, we can adapt the expressions in Section IV to take the difference in frame sizes into account.

VII. Conclusion

In this paper, we have proposed an access method that supports both relative proportional throughput allocation and absolute priorities in 802.11 wireless networks. Our simulations show that the proposed method achieves its objectives of relative and absolute differentiation both with respect to the aggregated throughput and the speed of convergence. Unlike 802.11e EDCA, it presents very good scalability—the throughput remains almost constant in function of the number of contending stations. As our mechanism is based on *Idle Sense* access method, it provides good fairness for traffic of the same priority.

Acknowledgments

This work was partially supported by the European Commission project WIP under contract 27402 and the French Ministry of Research project AIRNET under contract ANR-05-RNRT-012-01.

References

[1] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, “Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs,” in *Proc. of ACM SIGCOMM 2005*, vol. 35, no. 4, August 2005, pp. 121–132.

[2] IEEE, “IEEE 802.11e: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: MAC Enhancements for Quality of Service (QoS),” January 2005.

[3] D. Qiao and K. Shin, “Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF,” in *IEEE Intĺ Workshop on QoS*, 2002, pp. 227–236.

[4] N. H. Vaidya, A. Dugar, S. Gupta, and P. Bahl, “Distributed Fair Scheduling in a Wireless LAN.” *IEEE Trans. Mob. Comput.*, vol. 4, no. 6, pp. 616–629, 2005.

[5] W. Pattara-Atikom, P. Krishnamurthy, and S. Banerjee, “Distributed Mechanisms for Quality of Service in Wireless LANs.” *IEEE Wireless Communications Magazine*, June 2003.

[6] C. Hu and J. C. Hou, “A Novel Approach to Contention Control in IEEE 802.11e-Operated WLANs,” in *Proc. IEEE INFOCOM’07*, May 2007.

[7] W. Pattara-Atikom, S. Banerjee, and P. Krishnamurthy, “A-DRAFT: an adaptive QoS mechanism to support absolute and relative throughput in 802.11 wireless LANs.” in *Proc. MSWiM’04*. ACM, 2004, pp. 117–126.

[8] IEEE, “IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications,” *ISO/IEC 802-11: 1999(E)*, August 1999.

[9] D. Chiu and R. Jain, “Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks,” *Journal of Computer Networks and ISDN*, vol. 17, no. 1, June 1989.

[10] Y. Grunenberger, M. Heusse, F. Rousseau, and A. Duda, “Experience with an Implementation of the Idle Sense Wireless Access Method,” in *Proc. of CoNEXT 2007*, New York, 2007.

[11] J. Deng and R. S. Chang, “A priority scheme for IEEE 802.11 DCF Access method.” *IEICE Trans. on Comm.*, vol. E82-B, no. 1, pp. 96–102, 1999.

[12] I. Aad and C. Castelluccia, “Differentiation Mechanisms for IEEE 802.11.” in *Proc. IEEE INFOCOM’01*, 2001, pp. 209–218.

[13] A. Banchs and X. Perez, “Distributed Weighted Fair Queuing in 802.11 Wireless LAN.” in *Proc. IEEE ICC’02*, vol. 5, 2002, pp. 3121–3127.

[14] W. Pattara-Atikom, S. Banerjee, and P. Krishnamurthy, “Starvation Prevention and Quality of service in Wireless LANS,” in *Proc. IEEE WPMC’02*, 2002.

[15] M. Shreedhar and G. Varghese, “Efficient Fair Queueing Using Deficit Round Robin,” in *Proc. ACM SIGCOMM’95*, 1995, pp. 231–242.

[16] L. Bononi, M. Conti, and E. Gregori, “Runtime optimization of IEEE 802.11 wireless LANs performance,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 1, pp. 66–80, 2004.

[17] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, 2000.

[18] The Network Simulator NS-2 (v2.29). [Online]. Available: http://www.isi.edu/nsnam/ns/

[19] S. Wiethoelter, M. Emmelmann, C. Hoene, and A. Wolisz, “TKN EDCA Model for NS2,” Technische Universität Berlin, Tech. Rep. TKN-06-003, Jun. 2006.

[20] R. Jain, D. Chiu, and W. Hawe, “A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems,” *DEC Research Report TR-301*, September 1984.