Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs

Martin Heusse, Franck Rousseau, Romaric Guillier, Andrzej Duda

LSR-IMAG Laboratory[*]
Grenoble, France

{Martin.Heusse, Franck.Rousseau, Romaric.Guillier, Andrzej.Duda}@imag.fr

ABSTRACT

We consider wireless LANs such as IEEE 802.11 operating in the unlicensed radio spectrum. While their nominal bit rates have increased considerably, the MAC layer remains practically unchanged despite much research effort spent on improving its performance. We observe that most proposals for tuning the access method focus on a single aspect and disregard others. Our objective is to define an access method optimized for throughput and fairness, able to dynamically adapt to physical channel conditions, to operate near optimum for a wide range of error rates, and to provide equal time shares when hosts use different bit rates.

We propose a novel access method derived from 802.11 DCF [2] (*Distributed Coordination Function*) in which all hosts use similar values of the contention window CW to benefit from good short-term access fairness. We call our method *Idle Sense*, because each host observes the mean number of idle slots between transmission attempts to dynamically control its contention window. Unlike other proposals, *Idle Sense* enables each host to estimate its frame error rate, which can be used for switching to the right bit rate. We present simulations showing how the method leads to high throughput, low collision overhead, and low delay. The method also features fast reactivity and time-fair channel allocation.

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks—*Access schemes*

General Terms

Algorithms, Performance

Keywords

Wireless LANs, 802.11, Access Methods, Fairness

1. INTRODUCTION

Since the advent of the first 802.11 wireless LANs, much research effort has been spent on improving their performance. Successive variants have increased the nominal bit rate at the physical layer. However, the MAC layer remains practically unchanged despite many proposals for tuning its performance. Most of this work focuses on a single aspect while disregarding others. For example, many researchers try to improve throughput, but they neglect other performance aspects such as short-term fairness, adaptation to channel conditions, or handling multiple bit rates. Unlike these proposals, we follow a global approach to the design of an access method for wireless LANs by taking into account all aspects and trying to find the best tradeoff between antagonistic objectives.

We elaborate the principle of our method by considering a modification to the basic CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*) access method: contending hosts do not perform the exponential backoff algorithm after collisions, but rather dynamically converge in a fully distributed way to similar values of their contention windows. We optimize the throughput of the modified access method to establish relations for estimating the current state of the network and for dynamically controlling the contention window CW. We call our access method *Idle Sense*, because each host observes the mean number of idle slots between two transmission attempts. By comparing the estimate with a theoretically derived value, hosts adjust their contention windows CW in a fully distributed way using an AIMD (*Additive Increase Multiplicative Decrease*) control algorithm. It is the first access method to our knowledge that decouples the dynamic load control from collision perception—instead of adjusting the contention window upon a collision, *Idle Sense* controls transmission opportunities of hosts based on the observed number of idle slots. Load control based on collisions results in lower performance and less fairness, because some collisions are just failed transmissions. Our method approaches the optimal tradeoff between high throughput, low collision overhead, and good short-term fairness, which

---

[*] LSR-IMAG is a joint research laboratory of CNRS (*Centre National de la Recherche Scientifique*), INPG (*Institut National Polytechnique de Grenoble*), and UJF (*Université Joseph Fourier*).

SIGCOMM'05, August 22–26, 2005, Philadelphia, Pennsylvania, USA.

results in low delay. It is based on a surprisingly simple principle, yet it results in unprecedented overall performance.

Our method also enables a host to estimate the collision rate and distinguish it from the frame error rate due to poor channel conditions. This feature solves other problems, usually overlooked, that arise in wireless environments: adaptation of the bit rate to channel conditions and handling hosts with multiple bit rates. The information on the frame error rate is essential for deciding (or not) on switching to a different bit rate. The method can handle hosts operating at various bit rates by scaling their contention windows so that they obtain equal channel time shares. In this way, the method provides a solution to performance anomaly in which the rate of a slower host limits the throughput of a fast host [12]. We validate our method with simulations showing very good performance results in terms of efficiency, fairness, and reactivity.

The rest of the paper is organized as follows. We first analyze the contention in wireless LANs from which we derive our access method (Section 2). We then discuss the properties of our access method (Section 3) and present simulation results that show how the method meets the desired requirements (Section 4). We compare our method with other proposals (Section 5) and present some conclusions (Section 6).

2. IDLE SENSE ACCESS METHOD

Our objective is to define an access method optimized for throughput and fairness, able to adapt dynamically to physical channel conditions, to operate optimally for a wide range of error rates, and to provide equal time shares when hosts use different bit rates. Usually, optimizing a CSMA/CA method is limited to only considering the best possible use of the radio channel, which results in maximizing throughput. However, we also need to consider other performance properties such as short-term fairness and delay.

We start with a short discussion on the ways of optimizing the behavior of the standard 802.11 DCF access method. Then, we analyze channel contention in a modified 802.11 DCF method with equal contention windows CW. We optimize the throughput of such a method and establish relations for estimating the current state of the network and dynamically controlling the contention window CW. From this analysis, we derive the *Idle Sense* access method and show how it can provide high throughput and low collision overhead along with good fairness.

2.1 Optimizing DCF

How can we optimize a CSMA/CA access method such as 802.11 DCF? Its principles are as follows: a host wishing to transmit senses the carrier on the channel, waits for a period of time (DIFS—*Distributed Inter Frame Space*) and then transmits if the medium is still free. If the frame is correctly received, the receiving host sends an ACK frame after another fixed period of time (SIFS—*Short Inter Frame Space*). If the channel is perceived to be busy, the emitting host waits until the channel is free and after the DIFS interval, it waits for a random contention time: it chooses backoff b, an integer distributed uniformly in the window $[0, CW[$ and waits for b time slots before attempting to transmit. If two hosts have the same remaining value of backoff at any given instant, they transmit at the same time and eventually collide. When a host detects a failed transmission (it does not receive the ACK of a frame), it executes the exponential backoff algorithm—it doubles CW (the contention window CW may vary between $CW_{\min}$ and $CW_{\max}$).

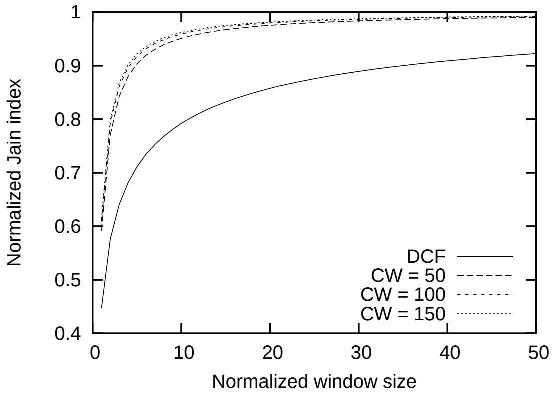

Figure 1: Fairness of 802.11 DCF with equal contention windows CW.

The main source of short-term unfairness in this access method is the exponential backoff algorithm applied after collision: the colliding hosts double their contention windows and have higher probability of choosing a larger backoff during which other hosts may benefit from channel access. This also means increased delay for hosts that doubled their CW. In this way, the standard DCF method controls the load on the channel by reducing the number of contending hosts, because the hosts that have failed their transmission are likely to attempt to access the channel later on. Moreover, hosts consider all failed transmissions as collisions, whereas only a part of them are really collisions. So, DCF bases its load control on a biased indicator, which leads to lower performance and increased unfairness. By making contention window CW equal for all hosts we can alleviate these problems and even more importantly, we can decouple the load control from handling failed transmissions. Obviously, once we have decided to set contention windows equal, we also need to propose another control algorithm for adjusting the contention window to the current load on the channel, which is the main objective of the *Idle Sense* method.

To illustrate this discussion, we have evaluated the short-term fairness of a modified 802.11 DCF with CW equal for all hosts by using the sliding window method that observes the patterns of transmissions to compute the average Jain fairness index in a window of an increasing size[1] [14, 18]. Figure 1 compares the Jain fairness index for the standard 802.11 DCF access method with the case in which CW is kept equal for all hosts. We can observe that equal CW results in significant improvement of short-term fairness, which means that the delay is also improved[2].

2.2 Analysis of channel contention

In this section, we analyze contention of a radio channel

---

[1] We have normalized the window size with respect to the number of hosts and computed the Jain index for the window sizes that are multiples of N.

[2] The present discussion assumes that the competing hosts transmit at the same bit rate; however the notion of fairness may need to be changed if hosts use different bit rates (cf. discussion in Section 3.2).

shared by multiple hosts by means of the modified 802.11 DCF access method (no exponential backoff, equal CW for all hosts). We consider the case of greedy hosts—they always have a frame to transmit. The analysis below assumes that all hosts can hear each other, so we do not consider the use of RTS/CTS frames. However, our proposed access method may use RTS/CTS just like 802.11 DCF to relax this condition by requiring that all hosts hear at least one of the two hosts involved in a frame exchange (typically, all hosts are within the reach of an access point).

We represent the contention in the modified 802.11 DCF access method as a discrete time stochastic process evolving between three states (see Figure 2 that illustrates the contention process for two hosts):

- an idle contention slot,
- a successfully acknowledged transmission,
- a collision.

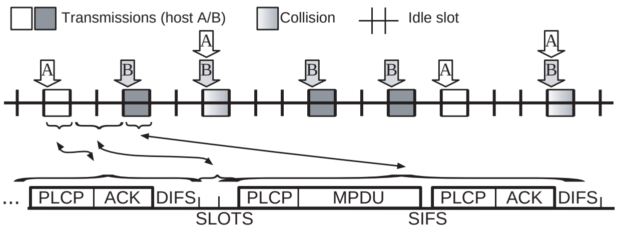

Figure 2: Two hosts contending for the channel.

Actually, all time intervals have different durations; for example, an idle contention slot is much shorter than a collision, which in turn may be shorter than a successful transmission. However, we represent them as equal intervals, because it is sufficient for our analysis as we are only interested in probabilities of being in a given state. Note that it is easy for a host to distinguish between an idle slot (no carrier) and two other states (carrier is sensed).

Consider the event consisting of a host attempting to transmit in a given time slot. Denote the attempt probability by P_e. Thus it is straightforward to express P_t, the probability of a successful transmission in a given slot, if N hosts contend for the channel: such an event requires a transmission attempt by a single host and the absence of all the others:

$$P_t = N\ P_e\ (1 - P_e)^{N-1}. \tag{1}$$

The collision probability in a slot can be expressed in a similar way:

$$P_c = 1 - P_t - P_i \tag{2}$$

where

$$P_i = (1 - P_e)^N \tag{3}$$

is the probability of an idle slot.

Let us denote by n_i the number of consecutive idle slots between two transmission attempts (transmission or collision). Its mean depends on the idle probability in the following way:

$$\overline{n_i} = \frac{P_i}{1 - P_i}. \tag{4}$$

If all hosts share the same CW and consider the successful transmission and collision intervals like any other intervals during which they wait[3], the attempt probability P_e can be computed as [5, 9, 7]:

$$P_e(CW) = \frac{2}{CW + 1}. \tag{5}$$

Anyway, Eq. 5 remains a good approximation for the attempt probability in an access method with a given CW. Note also that even if we know the attempt probability P_e, we need to know N, the number of contending hosts, to find the other probabilities.

To derive the principles of our access method, we will optimize the throughput by minimizing collision overhead and the time spent in idle slots. We can express the throughput as a function of P_e (Eqs. 1, 2, and 3) as follows:

$$X(P_e) = \frac{P_t\ s_d}{P_t\ T_t + P_c\ T_c + P_i\ T_{\mathrm{SLOT}}}. \tag{6}$$

In this expression, we assume as known: the average transmission duration T_t, the average collision duration T_c, and the slot duration T_{SLOT}. T_t depends on parameters of the PHY and MAC layers (802.11a/b/g/n) and the average data frame size s_d, which can be obtained from traffic measurements or, for analyzing the worst case, we can assume equal to the maximum frame size used in practice—the Ethernet MTU of 1500 bytes. Similarly, it is difficult to determine T_c in real conditions, so we use an upper bound—the duration of a collision involving a frame of the maximum size (it corresponds to the transmission of a data frame, the PLCP header included, without an ACK frame and without EIFS—we consider that not all stations wait for EIFS after a collision)[4]. T_{SLOT} is defined in a given standard (802.11a/b/g/n).

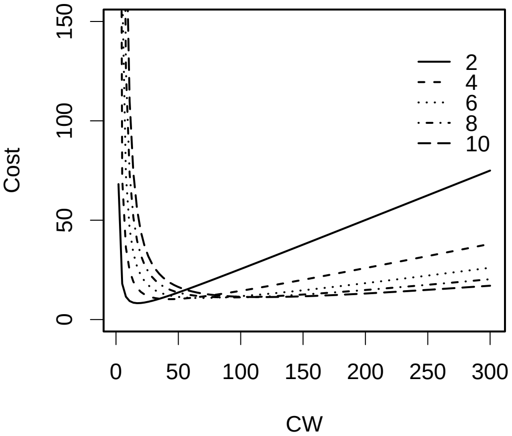

Figure 3: Cost function with respect to contention window CW.

Our objective is to find the optimal attempt probability P_e^{opt} that maximizes the throughput by minimizing the time

---

[3] This is not true for the standard 802.11 DCF access method, because hosts do not decrement the value of their contention window intervals during transmissions and collisions (decrementing the backoff after a period of carrier sense was proposed in 802.11e).

[4] Another possible approach to precisely estimate T_c is to use RTS/CTS. In this case, a collision involves frames of known size.

spent in collisions and in contention while maximizing the time spent in transmissions. So, maximizing $X(P_\mathrm{e})$ is equivalent to minimizing the following cost function:

$$Cost(P_\mathrm{e}) = \frac{\frac{T_\mathrm{c}}{T_\mathrm{SLOT}}\ P_\mathrm{c} + P_\mathrm{i}}{P_\mathrm{t}} \tag{7}$$

Figure 3 plots the $Cost \circ P_\mathrm{e}(CW)$ function for several values of N, the number of hosts, with respect to the contention window CW. As expected, the optimal value of CW increases with N while the cost function becomes less and less sensitive to the variations of CW, which leads to subrange estimation of the number of competing stations [22] as an enhancement of the proposal by Calì *et al.* [9]. Setting the first derivative of the cost function to zero leads to:

$$1 - NP_\mathrm{e}^\mathrm{opt} = \eta(1 - P_\mathrm{e}^\mathrm{opt})^N \tag{8}$$

where

$$\eta = 1 - \frac{T_\mathrm{SLOT}}{T_\mathrm{c}}$$

can be computed for a given variant of 802.11 from the parameters of the MAC and PHY layers. Example values are: 802.11b (11 Mb/s bit rate): $\frac{T_\mathrm{c}}{T_\mathrm{SLOT}} = 68.17$[5] and 802.11g (54 Mb/s bit rate) : $\frac{T_\mathrm{c}}{T_\mathrm{SLOT}} = 31.0$.

By denoting

$$\zeta = NP_\mathrm{e}^\mathrm{opt}, \tag{9}$$

we obtain: $1 - \zeta = \eta(1 - \zeta/N)^N$. If we consider the limit for $N \to \infty$, it gives:

$$1 - \zeta = \eta e^{-\zeta} \tag{10}$$

We can solve this equation numerically for the value of η corresponding to a given variant of 802.11, for example if $\frac{T_\mathrm{c}}{T_\mathrm{SLOT}} = 68.17$ (802.11b), we obtain $\zeta = NP_\mathrm{e}^\mathrm{opt} = 0.1622$.

N	CW^opt	$\overline{n}_\mathrm{i}^\mathrm{opt}$
2	18	4.01
3	30	4.51
4	43	4.89
5	55	5.01
6	68	5.18
7	80	5.23
8	92	5.26
9	105	5.35
10	117	5.36
11	129	5.38
12	142	5.43
13	154	5.44
14	166	5.44
15	179	5.48
16	191	5.48
17	203	5.48
18	216	5.51
19	228	5.51
20	240	5.51
21	253	5.54

Table 1: Optimal values of the contention window CW^opt and the number of idle slots $\overline{n}_\mathrm{i}^\mathrm{opt}$ (MAC and PHY parameters for 802.11b).

Moreover, when $N \to \infty$, Eq. 3 yields

$$P_\mathrm{i}^\mathrm{opt} = (1 - \zeta/N)^N \to e^{-\zeta}. \tag{11}$$

This means that for a given ratio $\frac{T_\mathrm{c}}{T_\mathrm{SLOT}}$, the throughput is optimal for the probability of an idle slot $P_\mathrm{i}^\mathrm{opt}$ that tends towards an easily computable constant. Similarly, Eq. 4 gives the optimal number of idle slots between two transmission attempts when $N \to \infty$ (it is equal to 5.68 for the 802.11b parameters, 3.91 for 802.11g):

$$\overline{n}_{\mathrm{i}\infty}^\mathrm{opt} = \frac{e^{-\zeta}}{1 - e^{-\zeta}}. \tag{12}$$

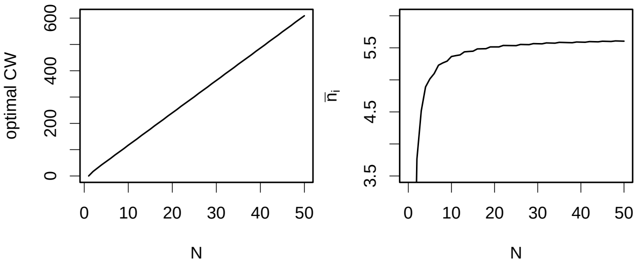

Figure 4: Optimal values of contention window CW^opt and corresponding numbers of idle slots $\overline{n}_\mathrm{i}^\mathrm{opt}$.

We build our access method (Section 2.3) on this result: we can observe the mean number of idle slots between two transmission attempts (the name of *Idle Sense* comes from this way of using Eq. 12) to see if the channel is used in an optimal manner. If the observed value is less than $n_{\mathrm{i}\infty}^\mathrm{opt}$, the operating point is not optimal due to excessive collisions, we thus need to increase CW; conversely, if the observed value is greater than $n_{\mathrm{i}\infty}^\mathrm{opt}$, the operating point is not optimal due to too much time spent waiting in idle slots, we thus need to decrease CW.

Consider now the case of $N < \infty$. We can obtain CW^opt and $\overline{n}_\mathrm{i}^\mathrm{opt}$ for a given number of competing hosts by finding the only root of the polynomial defined by Eq. 8 that is real and in $[0, 1]$ (Table 1 and Figure 4 present their values). However, note how quickly $\overline{n}_\mathrm{i}^\mathrm{opt}$ converges to an asymptotic value. We can thus use $\overline{n}_{\mathrm{i}\infty}^\mathrm{opt}$ as the target value to control CW (note also that the initial value of contention window $CW_\mathrm{min} = 32$ in 802.11 DCF is optimal for N between 3 and 4).

2.3 Principles of the Idle Sense access method

The idea of *Idle Sense* is simple: each host estimates $\widehat{n}_\mathrm{i}$, the number of consecutive idle slots between two transmission attempts and uses it to compute its contention window CW. By adjusting CW, a host makes n_i converge to $\overline{n}_\mathrm{i}^\mathrm{target}$, a common value for all hosts.

Ideally, if we know the number of contending hosts N, we can determine $\overline{n}_\mathrm{i}^\mathrm{opt}$ (Table 1) and use it as the target value $\overline{n}_\mathrm{i}^\mathrm{target}$. This requires knowing or estimating the number of contending hosts N, which we want to avoid. We observe that the number of idle slots between two transmission attempts $\overline{n}_\mathrm{i}^\mathrm{opt}$ converges quickly to the asymptotic value $\overline{n}_{\mathrm{i}\infty}^\mathrm{opt}$ (Eq. 12). We can thus use a value close to it as the target value $\overline{n}_\mathrm{i}^\mathrm{target}$ for all hosts. For example, we have chosen the value of $\overline{n}_\mathrm{i}^\mathrm{target}$ to be 5.68 for 802.11b, because it gives satisfactory results for a wide range of number of active hosts. The method is not sensitive to this value, as shown in Figure 5. This figure also shows how small is the additional cost of using a fixed $\overline{n}_\mathrm{i}^\mathrm{target}$ instead of $\overline{n}_\mathrm{i}^\mathrm{opt}$.

Finally, we need a control algorithm to track the value of n_i and make it converge to the target value $\overline{n}_\mathrm{i}^\mathrm{target}$. As we want to share the attempt probability P_e equally among all hosts, a natural choice of a control mechanism is the AIMD principle applied to the attempt probability P_e. AIMD has the property of converging to equal values of the control variable [10]. Our control algorithm is the following: if we observe too many idle intervals compared to the target (large $\widehat{n}_\mathrm{i}$), we need to increase P_e additively, which in turn will decrease the expected number of consecutive idle slots n_i,

---

[5] 83.87 if EIFS is added

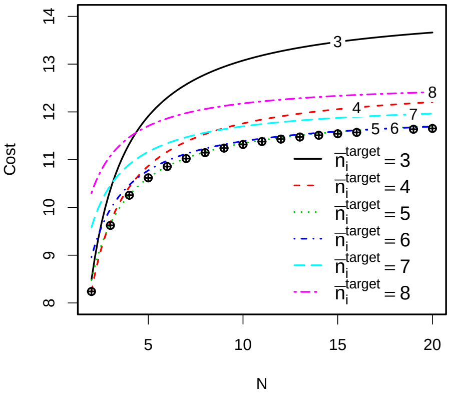

Figure 5: Cost function for various $\overline{n}_i^{\text{target}}$. The dotted line gives the minimal values of the cost function obtained for $\overline{n}_i^{\text{opt}}$.

whereas if we observe too few idle intervals (small $\widehat{n}_i$), we need to decrease P_e in a multiplicative way, which in turn will increase the expected number of consecutive idle slots n_i. This yields the following algorithm:

- If $\widehat{n}_i > \overline{n_i}^{\text{target}}$, $P_e \leftarrow P_e + \epsilon$
- If $\widehat{n}_i < \overline{n_i}^{\text{target}}$, $P_e \leftarrow \alpha P_e$

Using a simplified equation for P_e (cf. Eq. 5):

$$P_e = \frac{2}{CW} \tag{13}$$

we can easily transform the AIMD rules for P_e into the equivalent rules for CW:

- If $\widehat{n}_i > \overline{n_i}^{\text{target}}$, $CW \leftarrow \frac{2\ CW}{2+\epsilon\ CW}$
- If $\widehat{n}_i < \overline{n_i}^{\text{target}}$, $CW \leftarrow \frac{CW}{\alpha}$

The *Idle Sense* access method is described more formally in Figure 6.

N	CW	N	CW
2	24.7	12	148.0
3	37.0	13	160.3
4	49.3	14	172.7
5	61.7	15	185.0
6	74.0	16	197.3
7	86.3	17	209.7
8	98.7	18	222.0
9	111.0	19	234.3
10	123.3	20	246.7
11	135.7	21	259.0

Table 2: Values of CW for $\overline{n}_i^{\text{target}} = 5.68$.

Consider an example of $N = 5$ contending hosts that use the current value of $CW = 60$. The AIMD coefficients are set to $\frac{1}{\alpha} = 1.2$ and $\epsilon = 0.001$. We can see that the algorithm keeps CW oscillating above or below the target value from Table 2 61.7.

1. First step: $CW = 60$ implies $P_e = 0.033$ and $P_i = 0.870$, so that $\overline{n}_i = 5.51$. Thus, the test in the algorithm will increase CW because $\widehat{n}_i$ is smaller than 5.68: the new value of CW becomes $\frac{CW}{\alpha} = 1.2 \times 60 = 72$.

```
After each transmission
{
  /* **Idle Sense**: host observes $n_i$
     idle slots before a transmission */
  $sum \leftarrow sum + n_i$
  $ntrans \leftarrow ntrans + 1$

  if ($ntrans >= maxtrans$) {
    /*  Compute the estimator */
    $\widehat{n}_i \leftarrow sum/ntrans$
    /* Reset variables */
    $sum \leftarrow 0$
    $ntrans \leftarrow 0$
    if ($\widehat{n}_i < \overline{n}_i^{target}$) {
      /* Increase $CW$ */
      $CW \leftarrow \frac{CW}{\alpha}$
    }
    else {
      /* Decrease $CW$ */
      $CW \leftarrow \frac{2CW}{2+\epsilon CW}$
    }
  }
}
```

Figure 6: *Idle Sense* access method.

2. Second step: This leads to $\widehat{n}_i = 6.71$. As this value is now greater than the target value 5.68, the algorithm will decrease CW according to $\frac{2\ CW}{2+\epsilon\ CW}$ giving the new value of $CW = 69$.

3. Next step: for $CW = 69$ we can find the new value $\overline{n}_i = 6.41$. It is a lower value, but as in the previous step, it is still greater than the target value 5.68, so the algorithm will decrease CW according to $\frac{2\ CW}{2+\epsilon\ CW}$ giving the new value of $CW = 67$. Step by step, the algorithm approaches and oscillates around the target value.

The AIMD control mechanism results in oscillations around the target with amplitude that may increase for larger values of the target. However, this has low impact on the throughput, because of the flat shape of the cost function for a large number of hosts N (cf. Figure 3).

3. PROPERTIES OF IDLE SENSE

Besides achieving high throughput and guaranteeing short-term fairness, an access method dedicated to wireless LANs also needs to adapt the bit rate to channel conditions and deal with the issue of time-fairness. We discuss all these problems in this section.

3.1 Channel adaptation

Many studies have shown that wireless channels exhibit *time varying* characteristics—the quality of received signals changes dramatically even over short time intervals due to multiple causes: noise, attenuation, interference, multipath propagation, and host mobility [23, 1]. Poor transmission conditions result in incorrectly received frames detected by CRC. The frame error rate may be fairly high, for example in the case of hosts too far away from the receiver or when transmission is subject to interferences and fading. At some threshold frame error rate, a host may lower its bit rate to obtain better throughput—transmission at the lower bit rate uses more robust modulation schemes that decrease the frame error rate (e.g. the 802.11g and 802.11a standards define several bit rates ranging from 6 to 54 Mb/s). Obviously, transmission takes longer, but a host expects this to be compensated by a decrease in the frame error rate, which globally results in better throughput. The main issue is to decide when it is beneficial to switch to a lower bit rate.

Let us consider a single host transmitting at the higher bit rate (lower bit rate) r_{h} (respectively r_{l}). When switching to the lower bit rate, we expect to lower the frame error rate from e_{h} to e_{l}. We assume that some part of the frame transmission time does not depend on the bit rate: σ_{h} (respectively σ_{l}) denote the proportion of useful throughput when using the higher bit rate (respectively lower bit rate). To find the threshold value of the frame error rate for which a host needs to lower its bit rate, we need to have at least the same throughput for both bit rates:

$$r_{\mathrm{h}}\sigma_{\mathrm{h}}(1-e_{\mathrm{h}}) = r_{\mathrm{l}}\sigma_{\mathrm{l}}(1-e_{\mathrm{l}}). \qquad (14)$$

The threshold frame error rate is thus

$$e_{\mathrm{h}} = 1 - \frac{r_{\mathrm{l}}\sigma_{\mathrm{l}}}{r_{\mathrm{h}}\sigma_{\mathrm{h}}}(1-e_{\mathrm{l}}). \qquad (15)$$

As a first approximation, we can assume that lowering the bit rate will reduce the frame error rate to 0 and the proportions of useful throughput are equal ($\sigma_{\mathrm{h}} = \sigma_{\mathrm{l}}$). In this case, we simply obtain

$$e_{\mathrm{h}} = 1 - \frac{r_{\mathrm{l}}}{r_{\mathrm{h}}} \qquad (16)$$

This means that for 802.11b, we need to switch from 11 Mb/s to 5.5 Mb/s when the frame error rate exceeds 50%. If the frame error rate at the lower bit rate is not 0, this value should be increased, e.g. if $e_{\mathrm{l}} = 0.2$, the threshold increases to 60%. A similar observation comes from measurement studies [1]: it is generally better to wait for a significantly high loss rate before switching to a lower bit rate.

We can thus see that selecting the optimal bit rate is not easy, because the rate used by a sender depends on the frame error rate perceived by the receiver. It might return its measured value, if such a measure exists, to the sender, but this requires a reverse signaling channel. One may think about deriving the frame error rate from channel parameters such as the signal to noise ratio (SNR), however it is hard to relate the frame error rate to SNR (measurements in real life deployed networks show that the correlation between SNR and the frame error rate is fairly low [1]). An ideal access method needs to evaluate the frame error rate to control switching to a lower bit rate. The correctly evaluated error rate may also be useful for cross-layer optimizations, for instance for influencing TCP congestion control.

The *Idle Sense* access method provides a means for adapting transmission bit rate by being able to infer the frame error rate from the observed rate of successfully transmitted frames. This is challenging in 802.11 wireless LANs because collisions and missed transmissions usually cannot be distinguished (however, it can be done when using RTS/CTS for each data frame: a failed RTS/CTS exchange may be considered as a collision and no ACK for a frame as a loss).

We can express the conditional collision probability $P[\text{collision}|\text{transmission}]$ for a given transmission attempt as $P[\text{coll}|\text{transmission}] = \frac{P_{\mathrm{c}}}{P_{\mathrm{t}}+P_{\mathrm{c}}}$. Its limit for $N \to \infty$ is

$$P[\text{coll}|\text{transmission}]^{\infty} = 1 - \frac{\zeta}{e^{\zeta}-1}.$$

Its typical values are low (it is significantly less than 10% for $\zeta = 0.1622$). We can deduce from this relation the part of frame losses due to incorrectly received frames and use it for deciding to switch to a different bit rate.

We derive frame error rate P_{err} (that "adds" up to losses due to collisions) from the observed rate of successfully transmitted frames P_{ok}:
$P_{\mathrm{ok}} = (1-P_{\mathrm{c}})(1-P_{\mathrm{err}}) \Rightarrow P_{\mathrm{ok}} = 1 - P_{\mathrm{c}} - P_{\mathrm{err}}(1-P_{\mathrm{c}})$.
Considering that $1-P_{\mathrm{c}} \approx 1$, frame error rate $P_{\mathrm{err}} = 1-P_{\mathrm{c}} - P_{\mathrm{ok}}$ can be used in Eq. 15 to set thresholds that determine when to switch to a higher or to a lower bit rate.

During an initial phase of converging to a fixed operation point, a station may have a smaller value of CW than the optimal one, which may result in a higher number of collisions. This phenomenon is however alleviated by the fact that the convergence latency of Idle Sense is small compared to the time it takes to estimate frame losses: $\widehat{n_{\mathrm{i}}}$ is updated after any host transmission, while the loss probability may only be evaluated after a significant number of transmission attempts from a given host.

3.2 Time-fairness

Rate diversity in wireless LANs leads to performance anomaly: the rate of a slower host limits the throughput of a fast host, because it takes more time to transmit frames at a lower bit rate, which decreases the channel time available for the transmission of fast hosts [12]. It was shown that this phenomenon results from applying max-min fairness to the case of competing hosts with different bit rates [26]. In the method called *Opportunistic Auto Rate*, Sadeghi *et al.* proposed to grant hosts the same temporal-share of channel access as under single-rate 802.11 DCF [27]. Tan *et al.* reconsidered performance objectives in 802.11 DCF by proposing another definition of fairness that focuses on time shares instead of rate shares: *time-fairness* [29].

Using time-fairness as a performance objective does away with the performance anomaly as each host may benefit from a time share corresponding to its transmission conditions. We can thus eliminate two pathological situations: (i) performance anomaly in which the rate of a slower host limits the throughput of a fast host and (ii) starvation of slow hosts that may occur if an access point does not allow switching to a lower bit rate (the last approach is commonly used to deal with performance anomaly in current products).

In general, time-fairness can be achieved by proportionally scaling either the MTU [11] or the access probability according to bit rates. If we want to provide time-fairness at the MAC layer, it implies the latter approach and *Idle Sense* naturally allows to control the access probability of hosts. In this case, if they transmit at the maximum available rate $r_{\max}$, they compute the contention window CW as described in Section 2.3. If a host transmits at a lower bit rate r_{current}, it uses a modified contention window scaled with respect to the maximum available rate:

$$CW' = CW \times \frac{r_{\max}}{r_{\mathrm{current}}}.$$

This gives the following probability of access attempt for the slower host:

$$P_{\mathrm{e}}(CW') = \frac{2}{CW'+1} \approx \frac{2}{CW'} \approx P_{\mathrm{e}}(CW) \times \frac{r_{\mathrm{current}}}{r_{\max}}.$$

In this way, the slower hosts access the channel less often than fast hosts by a factor that corresponds to the ratio between the time fast and slow hosts use the channel (if we neglect transmission overhead), which corresponds to equal time shares of the channel.

We can note that using such a scaling of CW does not impact overall network performance, because even if only slow hosts compete for the channel, they will all have the same value of CW and the operating point will be the same regardless of the actual bit rate. The only issue here is that channel utilization may slightly drop, because $\overline{n}_{\mathrm{i}}^{\mathrm{target}}$ is computed for a given bit rate.

3.3 Final remarks

We can note that observing idle slots merely depends on the absence of the carrier signal, so it is independent of any mechanism for collision detection. It is an advantage in wireless LANs, because a collision may be unequally perceived by the hosts involved in it (cf. the discussion on the physical layer capture effect [17] in Section 5). Our method makes it possible to decouple the dynamic load control from perceiving collisions. In this way, load estimation comes for free, because hosts always sense the channel during contention: we simply require them to take into account the time elapsed since the last contention started, at the instant the host transmits or senses another transmission. This observation is one of the most important characteristics of our method.

The fact of using a constant target value $\overline{n}_{\mathrm{i}}^{\mathrm{target}}$ makes the *Idle Sense* access method simple, elegant, and efficient. Based on simple computations, each host is able to determine a new value of CW based on its local estimation of the number of consecutive idle slots.

Using the expected number of consecutive idle slots between two transmission attempts $\overline{n}_{\mathrm{i}}$ as a target value for the control algorithm guarantees that hosts converge to an optimal state with minimal collision or contention overhead, and maximal transmission probability—we express these objectives in our cost function (Eq. 7). As hosts use similar values of the contention window, they benefit from short-term fairness and short delays.

Choosing the asymptotic value as the target means that efficiency may be slightly lower for a small number of hosts. For instance, for two hosts, the optimum is reached for $\overline{n}_{\mathrm{i}} = 4.01$, so that time may be wasted in unnecessary backoff. However, it has little impact on perceived performance, because when few hosts compete for the channel, each of them benefits from a large share of the available bandwidth.

The *Idle Sense* access method provides near optimal channel utilization for a given ratio of $\frac{T_{\mathrm{c}}}{T_{\mathrm{SLOT}}}$. If the real traffic characteristics differ from what is expected (for example the average collision duration is different from the maximal transmission time, this upper bound being used in our optimization) the throughput may drop. Other refinements are still possible; for example, we can use RTS/CTS as indicated previously.

4. PERFORMANCE

To evaluate the performance of our access method, we have developed a discrete-event simulator that implements the standard 802.11 DCF method (no RTS/CTS) and *Idle Sense*. We have implemented several variants of 802.11: 11 Mb/s 802.11b, 54 Mb/s 802.11g, and its modification with the nominal bit rate of 100 Mb/s (we assume that other parameters of MAC layer for this rate is the same as for 802.11g). The parameters for setting optimal values for *Idle Sense* are the following: $\frac{T_{\mathrm{c}}}{T_{\mathrm{SLOT}}} = 68.17$ for 802.11b, $\frac{T_{\mathrm{c}}}{T_{\mathrm{SLOT}}} = 31.0$ for 802.11g, and $\frac{T_{\mathrm{c}}}{T_{\mathrm{SLOT}}} = 19.3$ for an extrapolated 100 Mb/s 802.11. Unless stated otherwise, hosts behave like greedy sources transmitting at a given bit rate. A simulation runs for 10^6 transmissions to obtain precision of the order of 10^{-3}. After several tests, we set the following values of the control parameters: $\epsilon = 0.001$ and $\frac{1}{\alpha} = 1.20$; $ntrans = 5$.

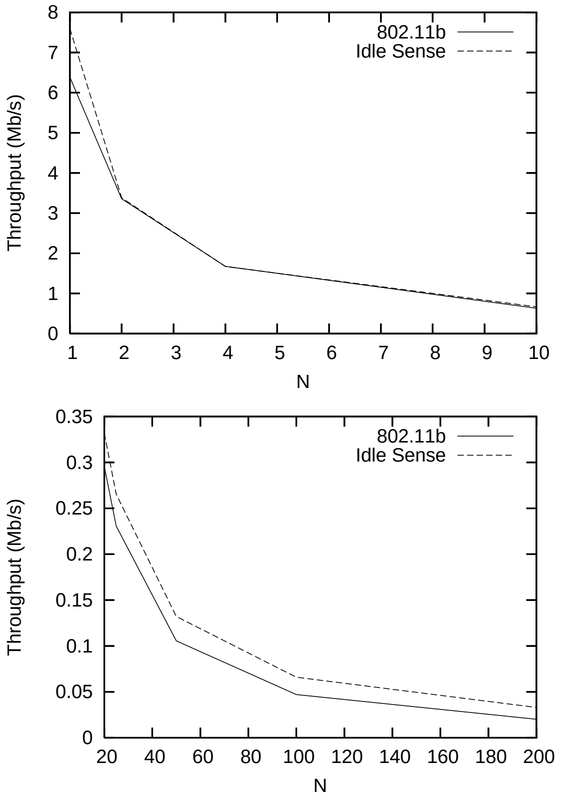

Figure 7: Throughput comparison.

4.1 Throughput

Let us bear in mind that our goal is to maximize throughput while providing fairness by setting the contention window CW equal for each host. So, in this part, we evaluate the throughput of our method, but we do not expect its significant increase. We compare throughput achieved by the 802.11b DCF, *Slow Decrease*[6] [25], *Asymptotically Optimal Backoff* (AOB) [7], and the *Idle Sense* methods for an increasing number of hosts (see Section 5 for descriptions of *Slow Decrease* and AOB). The throughput is the average of the throughputs of all hosts active in the network. Table 3 and Figure 7 show that for a small number of hosts, *Idle Sense* provides the throughput comparable to that of obtained in 802.11b. The improvement is significant for an increasing number of hosts even up to 60%. Such a result was expected, because we set the target number of idle slots $\overline{n}_{\mathrm{i}}^{\mathrm{target}}$ to 5.68. The gain is smaller for lower values of N.

---

[6] We use $CW_{min} = 8$ and $CW_{max} = 1024$ for simulations of *Slow Decrease*

N	1	2	4	10	15	20	25	50	100	200
Throughput, 802.11b (Mb/s)	6.39	3.35	1.67	0.63	0.41	0.29	0.23	0.10	0.05	0.02
Throughput, *Idle Sense* (Mb/s)	7.59	3.38	1.67	0.62	0.42	0.32	0.27	0.13	0.07	0.03
Throughput gain	19%	1%	0%	5%	9%	12%	15%	25%	40%	63%
Throughput, *Slow Decrease* (Mb/s)	7.32	3.40	1.65	0.63	0.41	0.31	0.24	0.12	0.05	0.03
Throughput, AOB (Mb/s)	5.64	3.04	1.57	0.37	0.28	0.22	0.25	0.12	0.06	0.03
Collision rate, 802.11b	0.0%	3.1%	7.8%	15.9%	20.0%	22.8%	25.1%	32.4%	40.5%	49.9%
Collision rate, *Idle Sense*	0.0%	3.0%	4.7%	6.1%	6.6%	6.9%	7.3%	8.4%	9.2%	9.7%

Table 3: MAC-level throughput and collision rate for 802.11b DCF, *Idle Sense*, AOB, and *Slow Decrease*.

N	1	2	4	10	15	20	25	50	100	200
802.11g, 54 Mb/s	31.79	16.18	7.85	2.92	1.87	1.36	1.06	0.49	0.22	0.09
Idle Sense ($\epsilon = 0.001$)	38.12	15.49	7.86	3.12	2.08	1.56	1.25	0.62	0.31	0.15
Idle Sense ($\epsilon = 0.1$)	38.12	16.53	7.94	3.00	1.96	1.46	1.17	0.57	0.28	0.14
802.11, 100 Mb/s	43.94	23.09	11.43	4.31	2.78	2.03	1.59	0.73	0.33	0.14
Idle Sense, 100 Mb/s	57.04	21.50	11.42	4.52	3.01	2.26	1.81	0.90	0.45	0.23

Table 4: Throughput comparisons between high bit rate variants of 802.11 and *Idle Sense*.

Table 4 shows the comparison between high performance variants of 802.11 and the *Idle Sense* method. We can also see that *Idle Sense* provides significant improvement of throughput for $N > 4$. For $N \leq 4$, there is no gain, because DCF is optimal. The advantage of *Idle Sense* is more in providing better fairness along with similar level of throughput.

For rates greater than 54 Mb/s, the 802.11 method in our simulator uses a smaller CW_{min} that makes it more efficient for a small number of hosts. To obtain better performance of *Idle Sense*, we can use a greater value of ϵ, which would result in a smaller CW attainable by our method (currently for $\epsilon = 0.001$ CW can only go down to 15). The line $\epsilon = 0.1$ gives an example of such a setting. It illustrates how insensitive throughput is to AIMD parameters.

The case $N = 1$ is a specific one: in *Idle Sense*, when a host does not sense any carrier from other hosts for a significant period of time, it sets CW to 2, which improves throughput. This value is the smallest one that allows any other host which becomes active to enter the competition for channel access.

4.2 Fairness: the Jain index

As in Figure 1, we evaluate short-term fairness by using the normalized Jain fairness index [14, 18]. We can see from Figure 8 that *Idle Sense* provides much better short-term fairness than 802.11b and other proposed modifications of 802.11 DCF such as *Slow Decrease* and AOB.

4.3 Delay

Bit rate (Mb/s)	11	54	100
802.11	1484	2600	2700
Idle Sense	94	93	87

Table 5: Maximum values of K, the number of inter-transmissions, for $N = 10$, observed over 10^6 transmissions.

To have some idea of the delay, we use another fairness index: K, the number of inter-transmissions that other hosts may perform between two transmissions of a given host [4].

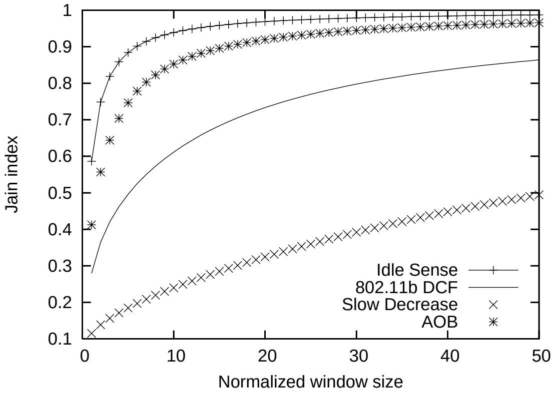

Figure 8: Fairness comparison for $N = 50$ competing hosts.

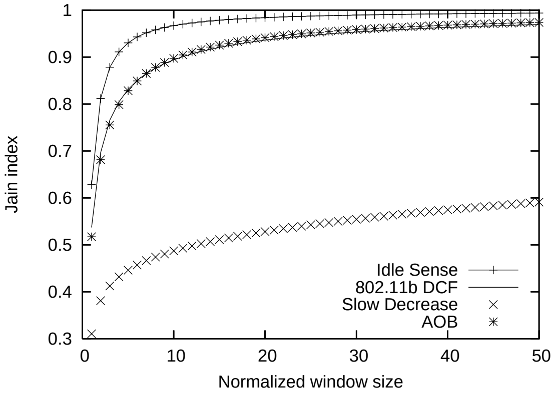

Figure 9: Fairness comparison for $N = 5$ competing hosts.

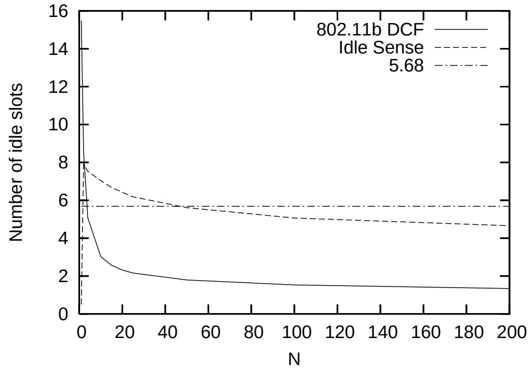

Figure 10: Average number of idle slots between transmissions.

We can notice that the number of inter-transmissions is directly related to delays perceived by a host competing with other hosts for channel access: when a host experiences large values of K, it also suffers from long delays, because it has to wait for channel access while other hosts transmit several frames.

Table 5 presents the maximum values of K, the number of inter-transmissions observed over 10^6 simulated transmissions for $N = 10$. This index is significantly lower for *Idle Sense* than for 802.11.

4.4 Collision overhead

Table 3 also presents the comparison of the collision overhead between 802.11 and *Idle Sense*. We measure the percentage of collisions (all collisions) per transmission experienced by hosts for each variant of 802.11 and *Idle Sense*. We can observe that even for $N \leq 4$, *Idle Sense* results in lower collision overhead, which means less waste of transmission capacity. Actually, the optimization of the cost function in Section 2.2 does not take into consideration the fact that the time wasted in collisions is even more expensive than the time spent in empty slots: transmission may prevent other hosts from using the channel and consumes energy. Moreover, each collision contributes to a packet loss above the MAC layer when the frame retry counter incremented at each transmission attempt reaches the retry limit.

Figure 10 presents the average number of idle slots in the *Idle Sense* method. For the special case of $N = 1$, we use $CW = 2$, which explains the singular value that can be observed in the figure. For greater values of N, the average number of idle slots stays around the target value. The observed bias is related to the AIMD mechanism and the chosen parameters correspond to a slight overestimation of CW for less than 50 hosts. Our objective here is to prefer a low collision rate: this choice is motivated by the shape of the cost function presented in Figure 3, which is much steeper on the left of the minimum than on its right.

4.5 Convergence speed

The convergence speed of the *Idle Sense* method is the next aspect to evaluate. For given traffic conditions (number of active hosts and their frame sizes), there is an optimal value of CW for which hosts benefit from short-term fairness and high throughput: if CW is too small, collisions are more frequent, and if CW is too large, hosts spend too much time waiting for transmission. Thus, an access method should adapt the value of CW to current traffic conditions.

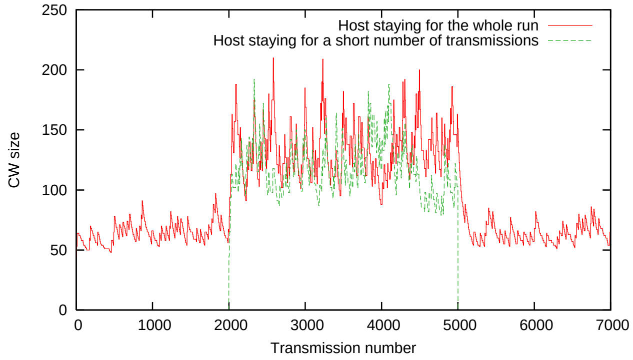

Figure 11: Convergence of the *Idle Sense* method.

We consider the following scenario. At the beginning, there are five greedy hosts competing for the channel; the parameters of their control algorithm have already converged to a stable value. After 2000 transmissions, five more hosts start competing for the channel. Then, the five hosts leave after 3000 transmissions (cf. Figure 11). We can see from this figure that the AIMD control method makes *Idle Sense* quickly adapt to the changes in traffic conditions—only few periods of idle slot estimation are required to reach a new stable state.

4.6 Time fairness

We have implemented the mechanism for scaling contention window CW with respect to a lower bit rate as described in Section 3.2. We have simulated the following scenario: one slow host transmitting at 1 Mb/s competes with $N - 1$ fast hosts transmitting at 11 Mb/s.

N	2	4	10	15	20
802.11b (slow host)	0.77	0.60	0.35	0.25	0.20
802.11b (fast host)	0.77	0.60	0.35	0.25	0.20
802.11b (average)	0.77	0.59	0.35	0.25	0.20
Idle Sense (slow host)	0.34	0.18	0.06	0.04	0.03
Idle Sense (fast host)	3.90	2.16	0.68	0.45	0.34
Idle Sense (average)	2.12	1.67	0.62	0.42	0.32

Table 6: Performance anomaly: throughput when a single host transmits at a lower bit rate (1 Mb/s vs. 11Mb/s).

Table 6 presents the throughput obtained by the slow host and the fast one. The first two lines show the performance anomaly: the throughput of the slow host is the same as that of the fast host, even if the latter transmits at a more than 10 times higher bit rate. We can observe that when *Idle Sense* is used, the fast host benefits from much higher throughput, which solves the problem—in this case, the access method behaves like TDMA by providing time-fairness: the channel time used by all hosts (slow or fast) is equal.

5. RELATED WORK AND COMPARISONS

We review below the main aspects related to access methods in wireless LANs: the characteristics of wireless LANs, performance analysis of 802.11 DCF, various proposals for enhancing access methods, and bit rate adaptation techniques.

Several experiments in real-life environments have studied the nature of wireless LANs. Measurements on an 802.11 testbed in the *Divert* project show significant frame losses due to time-varying behavior of radio channels even in the close vicinity of an access point [23]. The authors observe that frame losses occur in bursts and their rate strongly depends on the path between an access point and a host. Extensive measurements in *Roofnet* show that the distribution of frame loss rates is relatively uniform over the whole range of loss rates [1]. Moreover, SNR and distance have little predictive value for loss rate. Another study in ad hoc environments has found that wireless LANs exhibit complex behavior: the transmission range is not circular, communications are not symmetric, and the average signal strength varies widely even among positions close to an access point [19]. In another experiment, the authors show the importance of rate diversity: in an indoor environment with hosts relatively close to an access point, more than 50% of bytes were transferred using the lowest bit rate [29]. All this experimental evidence shows the importance of the aspects formulated in the design principles.

Other experiments have investigated the *physical layer capture* effect in 802.11 wireless LANs: a host may successfully receive a stronger frame (sent using a stronger signal) involved in a collision. The effect causes serious imbalance in throughputs of sources and significant unfairness [17], because the host that detects the collision performs the exponential backoff thus reducing its transmission opportunity. The other host succeeds on two fronts—it transmits its frame and continues to operate using the initial contention window. Our method alleviates this effect, because a host does not adjust its contention window when perceiving a collision.

Several authors have extensively analyzed the performance of 802.11 DCF [8, 9]. We build upon their main results on throughput optimization.

Much work proposes various enhancements to 802.11 DCF. Some of them consist of dynamic adjustment of CW. Calì *et al.* compute the optimal value of CW using three levels of estimators [9]. Bianchi *et al.* define a method for estimating the number of active hosts by means of a Kalman filter to set suitable values for CW [6]. Ma *et al.* proposed a way to overcome the complexity of such solutions by using a centralized approach: an access point measures the number of contending hosts and broadcasts the optimal value of CW [22].

Bononi *et al.* [7] proposed an improvement to 802.11 DCF called *Asymptotically Optimal Backoff* (AOB) that aims at similar performance objectives as *Idle Sense*. However, their asymptotic results are inaccurate: $(1 - P)^N$ does not converge to $1 - NP$ when $NP \approx \zeta$. Moreover, AOB keeps the exponential backoff mechanism of DCF, so it does not completely decouple collision detection from load control.

Aad *et al.* have introduced a *Slow Decrease* method [25]. dividing CW by 2 instead of resetting it to initial value $CW_{\min}$ after a successful transmission. The method was intended for improving efficiency and fairness, because the values of CW used by each host are less disproportionate. However, simulations in Section 4 show that even if the method obtains throughput similar to *Idle Sense*, it presents much worse fairness. Kwon *et al.* have defined *Fast Collision Resolution* [20]. The main idea is to double the contention window CW of any host that either experiences a collision or loses a contention; it then draws a new backoff counter. To decrease the time spent in backoff, hosts can exponentially decrease their backoff timer after observing a number of empty slots. This method presents a significant improvement of throughput compared to the standard 802.11 DCF method. However, as only the host that has just succeeded a transmission decreases its CW to the minimal value, the method causes high short-term unfairness that we want to avoid. The *Fairly Scheduled Fast Collision Resolution* (FS-FCR) variant [20] addresses this issue by setting a limit on the number of successive retransmissions that a host may perform: when a station reaches the limit, it sets its CW to $CW_{\max}$.

The *Binary Countdown Method* [30] can reduce collision overhead. As collisions significantly limit throughput, the method is more efficient than the standard 802.11 DCF. However, it requires a control channel for transmitting management messages to schedule each transmission. As the channel consumes 20% of the available bandwidth, it is not a method that compares favorably with our approach.

Some proposals were made before the emergence of the 802.11 standard. The hidden terminal problem motivated MACA (*Multiple Access Collision Avoidance*), which proposed to use RTS/CTS for collision avoidance on the shared channel [16]. The objective of MACAW (*MACA for Wireless*) was to achieve high throughput and fair channel allocation [3]. It suggested the use of link-layer ACKs with RTS/CTS and an additional DS (Data Sending) frame. It proposed a backoff control mechanism based on the MILD (*Multiplicative Increase, Linear Decrease*) principle: the backoff counter is increased by a factor upon a collision and decreased by 1 after a successful transmission. The method uses the same value of backoff counters for all hosts (as in our method), which is distributed in the packet header and copied by the receiver into its own counter. Nandagopal *et al.* explored a framework for fair access methods in wireless LANs [24]. They focused on studying and enforcing fairness when flows in the network face diverse spatial and contention conditions. We are not convinced that fairness at the level of flows needs to be enforced at the MAC layer—it is more the problem of traffic management at upper layers such as IP. The authors proposed *Proportionally Fair Contention Resolution* (PFCR) in which hosts control a transmission probability by means of the MILD principle. Similarly, Song *et al.* proposed a new backoff algorithm based on the EIED (*Exponential Increase Exponential Decrease*) principle: CW is increased by a factor upon a collision and decreased by another factor after a successful transmission [28]. Their method compares favorably with MILD and the standard exponential backoff of 802.11 DCF. Although these methods and ours have similar control schemes in common (MILD, EIED, and AIMD), our control algorithm acts upon a different variable, i.e. the mean number of idle slots.

The main problem with all these proposals is related to the core principle of dynamic load control: in all the methods a host increases its contention window after it experiences a collision, which is considered as a signal to decrease the rate of transmission attempts. This leads to degraded performance, because the methods cannot distinguish collisions from corrupted frames nor can they handle the capture effect [17] correctly: when a method adjusts the contention window after a collision, its operation is not optimal. As our method relies on only observing idle periods in channel activity, it is insensitive to all the problems that arise in methods based on inferring the channel load from collisions.

TCF (*TDM-based Coordination Function*) is an original approach, much different from the contention based methods [21]. It eliminates contention periods by allocating the channel dynamically using a TDMA scheme. The method offers high throughput and good fairness if the number of contending hosts remains stable; otherwise, it presents similar problems as other proposals, because the phase allowing hosts to join, is based on contention.

A different approach is represented by Tan *et al.* who proposed placing a regulator above the MAC layer in an access point to control the cell and set rates for hosts according to some performance objective (throughput or time-fairness) [29]. Although the idea is interesting from the point of view of providing equal time shares to hosts, it relies on a central coordinator.

802.11e mechanisms support quality of service with several priority classes. the same as in 802.11, however 802.11e introduces priorities by using the *Extended DCF* (EDCF) in which inter-frame intervals and the contention windows depend on traffic classes: $AIFS(class)$ and $CW(class)$. It also extends the standard coordinator based method PCF to the HCF (*Hybrid Coordinator Function*). As the standard is currently being developed, it is not clear whether the values of CW will result in optimal throughput. Inside one class of traffic, the respective fairness properties of the method remain the same as in 802.11 DCF. Even if the present paper does not look at the problem of service differentiation, it is straightforward to add our method to 802.11e mechanisms so that the traffic of the same class benefits from optimal performance.

As we can use *Idle Sense* for selecting the best bit rate, we briefly review other related methods. 802.11 products commonly use a bit rate adaptation method based on *Auto Rate Fallback* (ARF) first used in Lucent's WaveLAN II cards [15]. In ARF, the sender chooses the best bit rate by incrementally decreasing the rate after a number of frame losses and increasing it after several successful transmissions. ARF performs poorly, because of the confusion between collisions and incorrectly received frames: hosts may frequently degrade the bit rate only because of an increased collision rate resulting from a higher channel load. Much experimental evidence has been gathered on this effect: in a cell with many hosts (20) placed closed to an access point, the proportion of frames transmitted at a low rate (1 Mb/s for 802.11b) is fairly high (around 30%) [29].

Another method, the *Receiver Based Auto Rate* (RBAR), makes use of the RTS/CTS mechanism: the receiver reports the SNR of the RTS frame in a replying CTS frame [13]. Even if simulations show fairly good performance of RBAR, this relies on the strong correlation between SNR and the error rate, which is weak in reality [1]; it assumes that the channel conditions remain constant during transmission of the RTS/CTS and data frames, and derives the measure of the SNR from transmission at a lower bit rate (for instance RTS is sent at 2 Mb/s in 802.11b).

Our rate adaptation mechanism provides feedback information on the channel quality, so it may coexist with other schemes such as RBAR. An extension of this work would be to combine the information coming from multiple schemes (e.g. the decision may be based on both the frame error rate from *Idle Sense* and on SNR from RBAR).

6. CONCLUSIONS

We have presented *Idle Sense*, a novel access method that adapts the contention windows of contending hosts so they converge to a common value, which is near optimal for given network conditions. In this way, it offers high throughput, low collision and contention overhead, and good short-term fairness. Such optimal behavior is desired for next generation wireless LANs operating in unlicensed spectrum bands at high data rates.

The method relies on the AIMD adjustment of contention windows based on an estimator of the number of idle slots. It is fully distributed and does not require any centralized point of coordination. Our simulations show very good results in terms of efficiency and fairness.

Idle Sense solves, in a natural way, the performance anomaly problem observed when contending hosts use various bit rates. It also provides a criterion for detecting adverse transmission conditions in order to degrade the transmission bit rate: this is of major interest for the new wireless LANs that feature considerable transmission rate diversity.

Future investigations will focus on giving more weight to the access point if present, because usually downlink traffic is greater then the traffic of mobile hosts. *Idle Sense* easily enables such optimization as it can estimate the number of contending hosts.

Another research direction is to find better solutions to the problem of exposed terminals—a station in the range of two other stations will give in with respect to its neighbors by increasing its CW, so it will access the channel less often. The problem may be even worse if the stations in its neighborhood do not hear each other, so that they may transmit simultaneously and successfully. In this case, the station in between will hardly ever sense the channel idle and it will almost never transmit. Any CSMA access method at least partially faces this problem and current solutions include properly placing access points so that cells do not overlap. We plan to address this problem in our future work.

7. REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-Level Measurements from an 802.11b Mesh Network. In *Proc. of ACM SIGCOMM*, Boston, USA, August 2004.

[2] ANSI/IEEE. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2003.

[3] V. Barghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a Media Access Protocol for Wireless LAN's. In *Proc. of ACM SIGCOMM*, August 1994.

[4] G. Berger-Sabbatel, A. Duda, O. Gaudoin, M. Heusse, and F. Rousseau. Fairness and its Impact on Delay in 802.11 Networks. In *Proc. of GLOBECOM 2004*, Dallas, USA, 2004.

[5] G. Bianchi, L. Fratta, and M. Oliveri. Performance Analysis of IEEE 802.11 CSMA/CA Medium Access Control Protocol. In *Proc. of PIMRC 1996*, 1996.

[6] G. Bianchi and I. Tinnirello. Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 Network. In *Proc. of INFOCOM'03*, 2003.

[7] L. Bononi, M. Conti, and E. Gregori. Runtime Optimization of IEEE 802.11 Wireless LANs Performance. *IEEE Transactions on Parallel and Distributed Systems*, 15(1):66–80, January 2004.

[8] F. Calì, M. Conti, and E. Gregori. IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement. In *Proc. of INFOCOM*, 1998.

[9] F. Calì, M. Conti, and E. Gregori. Dynamic Tuning of the IEEE 802.11 Protocol. *ACM/IEEE Transactions on Networking*, 8(6), 2000.

[10] D. Chiu and R. Jain. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN*, 17(1), June 1989.

[11] J. Dunn, M. Neufeld, A. Sheth, D. Grunwald, and J. Bennett. A Practical Cross-Layer Mechanism for Fairness in 802.11 Networks. In *Proc. of BROADNETS'04*, 2004.

[12] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *Proc. of INFOCOM'03*, San Francisco, USA, March-April 2003.

[13] G. Holland, N. H. Vaidya, and P. Bahl. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Proc. of ACM MOBICOM 2001*, Rome, Italy, July 2001.

[14] 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.

[15] A. Kamerman and L. Monteban. WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band. *Bell Labs Tech. Journal*, 2:118–133, 1997.

[16] P. Karn. MACA: A New Channel Access Method for Packet Radio. In *Proc. of ARRL/CRRL Amateur Radio 9th Computer Networking Conf.*, 1990.

[17] A. Kochut, A. Vasan, U. Shankar, and A. Agrawala. Sniffing Out the Correct Physical Layer Capture Model in 802.11b. In *Proc. of ICNP'04*, Berlin, Germany, October 2004.

[18] C.E. Koksal, H. Kassab, and H. Balakrishnan. An Analysis of Short-Term Fairness in Wireless Media Access Protocols. In *Proc. of ACM SIGMETRICS*, 2000.

[19] D. Kotz, C. Newport, R.S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental Evaluation of Wireless Simulation Assumptions. Technical Report TR2004-507, Dept. of Computer Science, Dartmouth College, June 2004.

[20] Y. Kwon, Y. Fang, and H. Latchman. A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs. In *Proc. of INFOCOM'03*, San Francisco, USA, March-April 2003.

[21] C. Lim and C. Choi. TDM-based Coordination Function(TCF) in WLAN for High Throughput. In *Proc. of GLOBECOM 2004*, Dallas, USA, November 2004.

[22] H. Ma, H. Li, P. Zhang, S. Luo, C. Yuan, and X. Li. Dynamic Optimization of IEEE 802.11 CSMA/CA Based on the Number of Competing Stations. In *Proc. of ICC'04*, Paris, France, June 2004.

[23] A. Miu, G. Tan, H. Balakrishnan, and J. Apostolopoulos. Divert: Fine-Grained Path Selection for Wireless LANs. In *Proc. of MOBISYS'04*, pages 203–216, 2004.

[24] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving MAC Layer Fairness in Wireless Packet Networks. In *Proc. of ACM MOBICOM'00*, August 2000.

[25] Q. Ni, I. Aad, C. Barakat, and T. Turletti. Modeling and Analysis of Slow CW Decrease for IEEE 802.11 WLAN. In *Proc. of PIMRC*, Beijing, China, 2003.

[26] B. Radunovic and J. Y. Le Boudec. Rate Performance Objectives of Multi-hop Wireless Networks. In *Proc. of INFOCOM'04*, Hong Kong, March 2004.

[27] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic Media Access for Multirate Ad Hoc Networks. In *Proc. of ACM MOBICOM'02*, September 2002.

[28] N. O. Song, B. J. Kwak, J. Song, and L. E. Miller. Enhancement of IEEE 802.11 Distributed Coordination Function with Exponential Increase Exponential Decrease Backoff Algorithm. In *Proc. of VTC 2003 (Spring)*, New Orleans, 2003.

[29] G. Tan and J. Guttag. Time-based Fairness Improves Performance in Multi-rate Wireless LANs. In *Proc. of USENIX'04*, Boston, June 2004.

[30] T. You, C.-H. Yeh, and H. Hassanein. A New Class of Collision-Prevention MAC Protocols for Ad Hoc Wireless Networks. In *Proc. of ICC'03*, San Francisco, USA, May 2003.