Design and Analysis of an Adaptive Backoff Algorithm for IEEE 802.11 DCF mechanism

Mouhamad Ibrahim and Sara Alouf

INRIA – B.P. 93 – 06902 Sophia Antipolis – France
{mibrahim,salouf}@sophia.inria.fr

Abstract. This paper presents an adaptive backoff algorithm for the contention-based Distributed Coordination Function (DCF) of the IEEE 802.11 standard. Relying on on-line measurements of the number of sources, the algorithm, called Adaptive BEB, judiciously sets the size of the minimal contention window to adapt to the congestion level in the shared medium. The paper also provides an extension to Adaptive BEB for enhancing its performance over noisy channels. In this extension, a simple EWMA filter is used to derive a Packet Error Rate estimator. The performance evaluation of our proposal is addressed via simulations.

Keywords. DCF mechanism, optimal adaptation, performance evaluation, noise.

1 Introduction

Since its initial appearance in 1997, the IEEE 802.11 standard have engendered several research activities due to wireless local area networks (WLANs) specific features. One of the well-known drawbacks of IEEE 802.11 is the low performance of its MAC protocol in terms of throughput in congested networks. Several papers have pointed out this problem and proposed solutions to counter it. However, and as will be shown later, the performance of some of these degrades substantially under various scenarios.

In this paper, we aim at designing a new adaptive and robust backoff algorithm that enhances the performance of the Distributed Coordination Function (DCF) of the IEEE 802.11 standard under a wide variety of conditions. This new algorithm is meant to be simple and as close as possible to the standard. Our objectives are: maximum system performance in terms of maximum system throughput and minimum packet delay, and robustness to channel errors. Considering saturation conditions, i.e. active nodes have always packets to transmit, we derive a simple expression relating the minimum contention window size of the DCF backoff to the optimal transmission probability. The latter probability has been computed in [1], and an alternative formulation can be derived from [2, 3]. Both formulations require an estimate of the number of contending stations. In this paper, we propose a novel method for estimating the number of sources, which relies on counting *signs of life* coming from other stations to estimate their number. Despite many studies concerned with the optimization of the DCF in congested networks, the presence of errors on the channel is often disregarded. Most proposals rely on missed acknowledgments (ACK) from the destination to trigger the control of the contention window. However, missed ACKs are due either to congestion or channel errors. In this paper, we introduce a mechanism based on an Exponentially Weighted

Moving Average (EWMA) estimator of the Packet Error Rate (PER) seen on the channel to adjust the contention window, reducing thus the overhead introduced by the noise while still avoiding collisions.

The rest of the paper is as follows: Sect. 2 briefly reviews the DCF and some related work. The analytical background needed to design our algorithm is presented in Sect. 3. Section 4 is devoted to the algorithm itself, motivating every step of it and describing its design and implementation. A simple estimator of the number of contending nodes is presented in Sect. 5 and an enhancement of the backoff algorithm in noisy environments is presented in Sect. 6. Section 7 presents the simulation results and Sect. 8 studies the fairness of our algorithm. Finally, Sect. 9 summarizes our work.

2 Preliminaries on the DCF Mechanism and Related Work

The Distributed Coordination Function (DCF) is the primary access protocol in 802.11 for sharing the wireless medium between active stations. In DCF, the stations listen to the channel before transmitting: upon channel idleness for a duration greater than the Distributed InterFrame Space (DIFS) period, the station transmits directly; otherwise it waits for channel idleness. When the channel becomes idle again for a DIFS, the station enters a deferring phase by selecting a random number of time slots in the range $[0, CW-1]$ that is used to initialize a *backoff counter* [4]. Here CW refers to Contention Window and it is an integer, set at the first packet transmission attempt to the minimum contention window CW_0, and doubled as long as the transmission fails until reaching a maximum size $CW_{\max} = 2^m CW_0$, where m denotes the maximum backoff stage. The process of doubling the size of CW is called binary exponential backoff. The backoff counter is a timer decreased as long as the channel is sensed idle. When the backoff counter reaches 0, the station transmits directly, then waits for an ACK from the destination station. If the source station has not received an ACK within a specified ACK timeout, it will assume that the transmitted packet was lost due to a collision, and it will start the backoff procedure again after *doubling* its contention window size. On the other hand, if the packet is well received by the destination station, the latter will send an ACK to the source station that upon its reception will reset its contention window size to the minimal value CW_0 and proceed to the next packet in the buffer. In addition to this two-way handshaking technique, named basic access mechanism, DCF specifies another optional four-way handshaking technique called Request To Send/Clear To Send (RTS/CTS). The idea is to reserve the channel prior to a transmission by exchanging the RTS and CTS control frames between the source and the destination before the transmission of the packet.

The binary exponential backoff algorithm used in DCF has two major drawbacks. First, the contention window is increased upon transmission failure regardless of the cause of failure. Second, after a successful transmission of a packet, the contention window is reset to CW_0, thus forgetting its knowledge of the current congestion level in the network. These two points are at the basis of the inefficiency of the DCF mechanism.

As mentioned earlier, several research works have proposed modifications to IEEE 802.11 back-off algorithm to enhance its operation in congested networks. The earliest proposal [5] is based on a unique contention window CW, whose size is updated after each transmission attempt given an estimation of the number of active stations N. In order to estimate N, the authors measure the number of busy slots observed during a backoff decrease period. Another algorithm based on a unique contention window size is given in [2]. The authors establish an analogy between the standard protocol and the p-persistent IEEE 802.11 protocol, in which the backoff interval is sampled geometrically with parameter p. By maximizing the analytical expression found for the throughput of the p-persistent protocol, the authors derive the optimal p and subsequently the optimal size of the backoff interval. An estimation of the number of actual active users is required in this approach and the authors rely on measures of idle time to perform this estimation. In [6], the authors propose an extension to the DCF mechanism in order to achieve maximum throughput. Instead of directly transmitting when the backoff counter reaches 0, the authors propose to differ the transmission with a certain probability that depends, among other terms, on the slot utilization and the distribution of the packet lengths (assumed to be geometric). For this algorithm to work, each station requires a measure of the slot utilization and an inference of the packet length distribution. The authors of [7] propose to multiplicatively decrease the contention window after a successful transmission using a decrease factor δ in the range $[0, 1]$ that is set constant for all stations. The choice of δ greatly influences the performance of the algorithm. Maximum system throughput is achieved when $\delta = 0.9$ at the cost of a high system response time. A linear increase/linear decrease contention window algorithm is proposed in [8]. More precisely, upon a missed ACK, the current contention window is increased by a constant value ω. Upon receiving an ACK, it is decreased by ω with probability $1 - \delta$, and kept unchanged with probability δ. The parameters ω and δ are set heuristically to constant values. To the best of our knowledge, only [9] proposes a mechanism to enhance the DCF mechanism for noisy environments. The proposed algorithm resets the contention window when a failed transmission is assumed to be noise-corrupted. When the RTS/CTS mechanism is used, it considers as noise-corruption the absence of ACK and as collision the absence of CTS. When the basic access mechanism is used, the assumed noise-corruption probability is adjusted linearly to approach the ideal transmission probability, computed thanks to a count of the number of active stations. Last, we would like to briefly mention that [10] proposes a Kalman filter to estimate the number of active users N and relies on channel sensing to measure the collision probability (noisy channels are not considered in this study).

3 The Model

To design our algorithm, we adopt the model developed in [1]. In [1], it is assumed that every source node has always packets to send; the channel conditions are assumed to be ideal; the collision probability is constant among all sources and independent of the past. Last, it is assumed that there is no limit on the number of retransmissions of a lost packet. In [1], the author derives the following expression for the transmission probability τ in terms of the protocol parameters

$$\tau = \frac{2(1-2p)}{(1-2p)(CW_0+1) + p\,CW_0(1-(2p)^m)} \tag{1}$$

where $m = \log_2(CW_{\max}/CW_0)$ and p denotes the collision probability seen by a transmitting source node. It is equal to $1-(1-\tau)^{N-1}$, where N denotes the number of active stations. For a given N, τ and p can be computed using a fixed-point approach. The author of [1] derives also an approximation of the optimal transmission probability, where the optimality refers to maximizing the system throughput. Let T_c denote the expected time taken by an unsuccessful transmission and σ denote the slot time length, the approximate optimal transmission probability is then given by

$$\tau^* = \frac{1}{N\sqrt{T_c/(2\sigma)}} \quad . \tag{2}$$

Insights on the Optimality of the Transmission Probability

The optimal transmission probability found in [1] will not only maximize the saturation throughput but will minimize as well the system response time. In other words, the idle time wasted over the wireless channel will be minimal. In this section, we will provide some insights on the optimality of the transmission probability.

The time over the wireless channel can be partitioned into successive *virtual transmission times*. A virtual transmission time initiates just after a successful transmission over the wireless channel, and ends at the end of the next successful transmission, as illustrated in Fig. 1. The wasted time in a virtual transmission time, denoted as $waste_\tau$, is due to collisions and idle slot times and can be written

$$waste_\tau = \mathbf{E}\left[N_cT_c + (N_c+1)\text{Idleness}\right] = \mathbf{E}[N_c]T_c + (\mathbf{E}[N_c]+1)\mathbf{E}[\text{Idleness}] \tag{3}$$

where $\mathbf{E}[N_c]$ and $\mathbf{E}[\text{Idleness}]$ respectively represent the average number of collisions and the average length of an idle period in a virtual transmission time (N_c and Idleness are statistically independent). In (3) it is assumed that all packets have the same size, hence T_c is a constant. The first and second terms of (3) respectively account for the total time wasted in collisions and the total idle time in a virtual transmission time. Observe that the virtual transmission time is simply $waste_\tau + T_s$ (T_s being the expected time taken by a successful transmission), so the normalized system throughput can be expressed as the ratio $\mathbf{E}[\text{payload}]/(waste_\tau + T_s)$. To derive expressions for $\mathbf{E}[N_c]$ and $\mathbf{E}[\text{Idleness}]$ we look at the distribution, in a virtual transmission time, of the number of collisions, N_c, and the idle period length. Let $P_I = (1-\tau)^N$, $P_s = N\tau(1-\tau)^{N-1}$ and $P_c = 1 - P_I - P_s$ be the probabilities of having an idle slot time, a successful transmission and a collision, respectively. Let σ be the slot time duration. We have

$$P[N_c = j] = \left(\frac{P_c}{P_c+P_s}\right)^j \frac{P_s}{P_c+P_s} \quad , \qquad P[\text{Idleness} = j\sigma] = P_I^j(1-P_I)$$

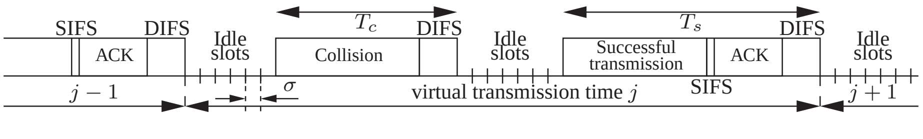

Fig. 1. An illustration of the virtual transmission time

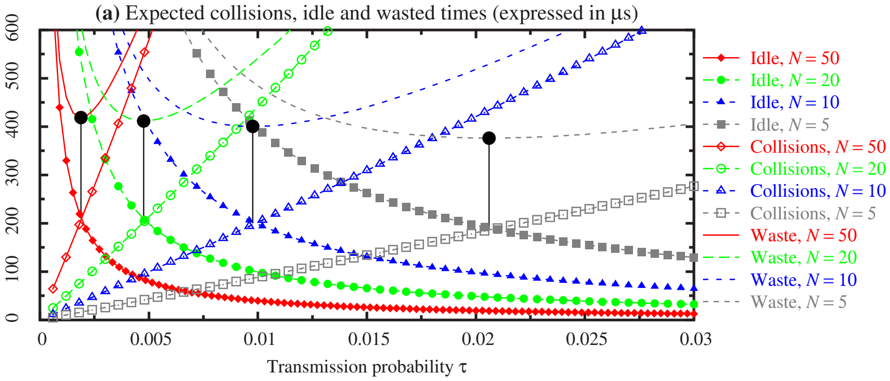

Fig. 2. Expected wasted time, expected time spent in collisions and expected idle time (all are in μs) in a virtual transmission time for different values of N ($T_c = 4335\mu$s and $\sigma = 20\mu$s).

for $j \in \mathbb{N}$, yielding $\mathbf{E}[N_c] = P_c/P_s$ and $\mathbf{E}[\text{Idleness}] = \sigma P_I/(1 - P_I)$. After some calculus, it comes that the total time wasted in collisions and the total idle time are

$$\mathbf{E}[N_c]T_c = T_c \left(\frac{1-(1-\tau)^N}{N\tau(1-\tau)^{N-1}} - 1 \right) \quad , \qquad (\mathbf{E}[N_c] + 1)\mathbf{E}[\text{Idleness}] = \frac{\sigma(1-\tau)^N}{N\tau(1-\tau)^{N-1}} \quad .$$

It can easily be proved that $\mathbf{E}[N_c]T_c$ is a monotone increasing function of τ, whereas $(\mathbf{E}[N_c]+1)\mathbf{E}[\text{Idleness}]$ is a monotone decreasing function of τ. In Fig. 2, both terms are plotted against τ for different values of the number of contending stations N. Their sum, $waste_\tau$, is also plotted and its minimal value for each N has been marked. The minimal values of $waste_\tau$ naturally correspond to the optimal transmission probability for each N. As seen in Fig. 2, the minimal values of $waste_\tau$ correspond to the intersections of both collisions and idle times. In other words, the optimality is achieved when the wasted time is equally shared between collisions and idleness. This optimal operating point has been identified in [11, p. 243] for the CSMA slotted Aloha protocol and in [3] for the p-persistent IEEE 802.11 protocol.

Observe that having the system response time minimized at the optimal transmission probability does not guarantee that the packet delay is minimized as well. This minimization also relies on the protocol fairness and its ability to equally share the medium among contending sources.

4 Adaptive Binary Exponential Backoff (BEB)

According to equation (2), optimal system performance can be achieved in different network topologies by fine-tuning the transmission probability of the stations to the optimal transmission probability τ^* for each network topology. For a given network topology, transmitting at the optimal transmission probability can be achieved by well adjusting the size of the minimum starting contention window CW_{min}. Note that a one-to-one relation between these two parameters can be obtained by inverting equation (1), and CW_{min} will then be expressed in terms of τ^* as follows:

$$CW_{\text{min}} = \frac{(2-\tau^*)(1-2p)}{\tau^*(1-p-p(2p)^m)} \quad . \tag{4}$$

Algorithm 1 Adjustment algorithm.

```
Require: An estimation $\overline{N}$ of the number of active nodes
Ensure: Sub-optimal values of $CW_{\min}$ and $m$
 1: Set $\tau = \tau^* = \frac{1}{\overline{N}\sqrt{T_c/2\sigma}}$
 2: Set $p = 1 - (1 - \tau^*)^{\overline{N}-1}$
 3: Compute $cw = \frac{(2-\tau^*)(1-2p)}{\tau^*(1-p-p(2p)^m)}$
 4: Select $j$ such that $2^j CW_0$ is the closest to $cw$, $j \in [0, m]$
 5: Set $CW_{\min} = 2^j CW_0$
 6: Set $m = \log_2 \frac{CW_{\max}}{CW_{\min}}$ $\{CW_{\max}$ is fixed$\}$
```

Unfortunately, the parameter m will still depend on the previous value of $CW_{\min}$ as follows $m = \log_2(CW_{\max}/CW_{\min})$. However, this practically will not impact the new selected value of $CW_{\min}$ since the term $p(2p)^m$ can be neglected with respect to 1.

The operation of our algorithm can then be summarized as follows. After a failed transmission, the behavior is the same as in the standard. However, after a successful transmission, the initial window size of the binary exponential backoff mechanism is set to an optimal value, hereafter denoted $CW_{\min}$. The selection of this value is detailed in Algo. 1. Given an estimation of N, the approximate optimal transmission probability is computed using (2) (line 1), enabling the computation of the collision probability p (line 2). The optimal minimal size of the contention window size is computed according to (4) (line 3). Note that to be as close as possible to the standard, $CW_{\min}$ is imposed to take only those values defined by the standard $\{2^j CW_0, j = 0, \ldots, m\}$ (recall that CW_0 is the size of the minimal contention window in the standard) (lines 4–5). Last, the value of m is updated (line 6). Observe that the values of $CW_{\min}$ and m will be sub-optimal due to the fact that $CW_{\min}$ is made discrete.

The analytical throughput achieved by our adaptive algorithm, hereafter referred to as “Adaptive BEB”, can be obtained by substituting the new values of $CW_{\min}$ and m and the optimal value of p given in line 2 of Algo. 1 (1) to compute the sub-optimal value of τ, and therefore the throughput achieved by Adaptive BEB will be $T = \mathbf{E}[\text{payload}]/(waste_\tau + T_s)$.

5 Estimation of the Number of Contending Stations

To adjust $CW_{\min}$, we need an estimation of N, hereafter denoted $\overline{N}$, that tracks its time-evolution. However, since the value of $CW_{\min}$ is made discrete, a reactive and adequately accurate estimator is sufficient. For instance, consider the case when $N = 30$. When applying Algo. 1, we would obtain the correct optimal value $CW_{\min} = 2^4 CW_0$ for $\overline{N} \in [20.45, 40.95]$.

As opposed to previously proposed methods (e.g. [2, 5, 10]) which estimate the current number of sources by measuring the channel activity, we propose to estimate the number of active stations directly by counting as active each station from which the concerned measuring station receives a *sign of life*, i.e. error-free data or RTS packets. The measurement period will be the virtual transmission time of the station, i.e. the time between its consecutive successful transmissions. The idea can be summarized as follows: since each active node is always filtering received packets whether they are destined to it or not, it can thus keep trace of a large number of active nodes in a given time interval by counting the number of *distinct* signs of life received in that time interval from these nodes. Let $\widehat{N}_n$ denote the count of distinct signs of life at the nth measurement period of a given station. For multiple reasons, these measurements cannot be used directly in Algo. 1 as an estimation of N. First, the count of signs of life cannot account for stations whose transmitted packets are corrupted, either by noise or collisions. Second, variable and relatively small measurement periods result in counting only a variable portion of all active stations. Therefore, based on the measurements $\{\widehat{N}_n\}_{n\in\mathbb{N}^*}$, an unbiased estimator should be devised. Observe that measurements collected over relatively large measurement times are more "accurate" than those collected over small measurement times, and should be preferentially treated. By observing that the expected length of a measurement period is reflected in cw, the value reached by the contention window at the end of the measurement time, one can use this value to proportionally weight the corresponding measurement, so that the following filter $\sum_{i=0}^{q-1} cw_{n-i}\widehat{N}_{n-i}/\sum_{i=0}^{q-1} cw_{n-i}$ can be used. Simulation analysis has shown that low values of the filter order q will suffice for a convenient performance, so hereafter, we set $q = 3$ so as to heighten reactiveness. However, and because of the fact that corrupted packets cannot contribute to the measurements, the previous ratio, henceforth referred to as the observation, underestimates N and needs thus to be corrected. To derive an appropriate correction on the observations, we have performed several simulations with different values of N. Figure 3(a) depicts the evolution of the observation against the actual number of sources. Obviously, a linear correction is needed, resulting in the following estimator

$$\overline{N}_n = a\left(\sum_{i=0}^{q-1} cw_{n-i}\widehat{N}_{n-i}\right) / \left(\sum_{i=0}^{q-1} cw_{n-i}\right) + b \tag{5}$$

with $a = 1.35405$, $b = 1.75998$ for $q = 3$. These latter values, which are independent of the nodes distribution, need to be adjusted in the case of noise over the channel; this issue is left for future work. Figure 3(b) illustrates the estimation of N over a simulation in which nodes arrivals are Poisson and activity time is exponentially distributed. The estimator exhibits good reactiveness to changes in N at the cost of large fluctuations,

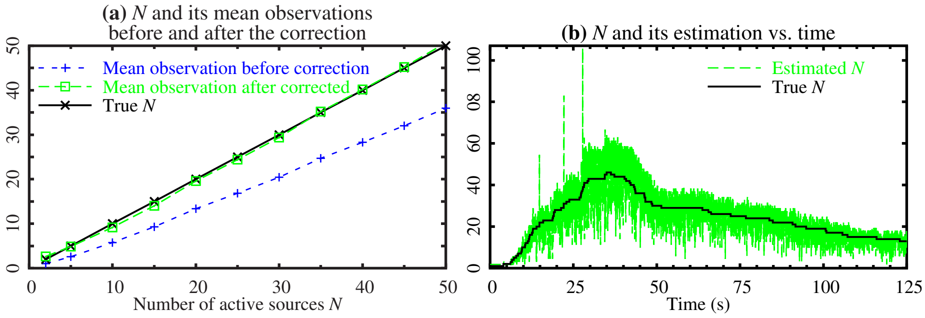

Fig. 3. (a) Mean observations of N vs. N. **(b)** Estimation of N vs. time in sample simulation

but these have only a limited impact on the performance of Algo. 1 because of the discretization process.

6 Enhancement of the Backoff Algorithm in Noisy Environments

The assumption of ideal channel conditions is in general unrealistic due to the existence of various “noise” factors (e.g. fading, shadowing, interference) that perturb the state of the wireless medium. Whenever a frame is noise-corrupted, both the standard and Adaptive BEB behave as if the loss is due to a collision, and the contention window of the backoff algorithm will be accordingly increased. Clearly, there is a flaw in the design of these algorithms due to the automatic invocation of the contention window increase process in the absence of CTS/ACK. Both algorithms lack to identify the cause of a missed CTS/ACK, and fail in adapting to error-prone environments. In the following, we present an optional extension that can be applied to Algo. 1 to enhance its performance in noisy environments. In this extension, we propose a Packet Error Rate (PER) estimator (Sect. 6.1), and a persistence mechanism that makes use of this estimator (Sect. 6.2). The resulting algorithm will be denoted as “Adaptive BEB^{++}”.

6.1 PER Estimation

To estimate the Packet Error Rate, we propose to infer it from the measured corruption probability in a station virtual transmission time, and filter these inferred values through a simple Exponentially Weighted Moving Average (EWMA) filter. This method works well in both RTS/CTS and basic access modes. When a transmission occurs on the channel, all other stations within communication range receive the transmitted packet. Upon receiving a packet, every station verifies first its Check Redundancy Code (CRC) so that corrupted packets could be disregarded. Therefore, every station is capable of counting how many received packets are corrupted out of all of them. The ratio between the two counts, noted $\hat{p}_{cr}$, is nothing but a measure of the corruption probability, p_{cr}, perceived in a measurement time. The corruption probability can be written $p_{cr} = p_c + p_e(1 - p_c)$ where p_e is the PER and p_c is the collision probability seen on the channel by a measuring station. We have $p_c = 1-(1-\tau)^{N-1}-(N-1)\tau(1-\tau)^{N-2}$. Using $\overline{N}$ and τ^* we can infer p_c; the inferred value is denoted $\hat{p}_c$. Therefore, p_e can be inferred as follows $\hat{p}_e = (\hat{p}_{cr} - \hat{p}_c)/(1 - \hat{p}_c)$ and is used to feed the following EWMA filter: $\bar{p}_{e,n} = \alpha\bar{p}_{e,n-1} + (1-\alpha)\hat{p}_{e,n}$, where $\bar{p}_{e,n}$ and $\hat{p}_{e,n}$ respectively denote the estimated and inferred PER in the nth measurement time at a given station. As for α, we have performed several simulations with $N = 25$ and an abruptly varying PER, as illustrated in Fig. 4(d). We have computed both expectation and variance of the error $p_e - \bar{p}_{e,n}$ for different values of α, and selected the value minimizing the mean error. This value is $\alpha = 0.95$ as can be seen from the table below. Observe in Fig. 4(d) how $\alpha = 0.95$ yields a highly reactive estimator.

α	0.9	0.95	0.99	0.995	0.999
$\mathbf{E}[p_e - \bar{p}_{e,n}]$	0.0174836	0.0153285	0.0192210	0.0259682	0.0714458
Var$[p_e - \bar{p}_{e,n}]$	0.0494257	0.0498566	0.0479849	0.0453496	0.0282030

6.2 The Persistence Algorithm

Unlike the standard and the Adaptive BEB mechanisms that upon a missed CTS/ACK blindly defer the transmission, we propose a persistence probability P_p such that the former behavior is undertaken only with probability $1 - P_\mathrm{p}$. Upon a missed CTS/ACK and with probability P_p, the station retransmits the packet when the CTS/ACK timeout expires. In other words, P_p can be regarded as the probability of assuming that a failed transmission is due to a noise-corruption, not a collision. Obviously, the choice of this persistence probability is crucial for the algorithm to perform well. P_p should account for $\bar{p}_e$, the backoff stage, and the count of persistent trials within the current backoff stage. Clearly, it should increase with $\bar{p}_e$. Also, as a station increments its window size for a given packet retransmission, the collision probability decreases. If, however, the retransmission still fails, then it is more likely that it was due to a noise-corruption rather than a collision, and therefore, P_p should increase with the actual backoff stage that is equal to $\log_2(CW/CW_\mathrm{min})$. Last, if, within the same stage, persistent retransmissions are repeatedly being unsuccessful, then it is more likely that the station is observing collisions and not corruption by the noisy channel, and as such, the persistence probability should decrease with $trials$, the count of persistent retransmissions within the same backoff stage. In our implementation, we have used the following expression of the persistence probability which exhibits all above-mentioned desired trends:

$$P_\mathrm{p} = \bar{p}_e^{\frac{trials}{1+\log_2(CW/CW_\mathrm{min})}} \quad .$$

7 Simulation Results

In this section, we investigate the effectiveness of our proposal through simulations conducted in `ns-2` [12]. Simulated nodes are uniformly distributed in a 100m×100m square and the power transmission is sufficiently high so that all the nodes are within communication range. All sources generate Constant Bit Rate (CBR) traffic. The protocol parameters are as follows: $CW_0 = 32$, $CW_\mathrm{max} = 1024$ and the slot size is $\sigma = 20\mu$s. The expected collision duration T_c used in (2) corresponds to the maximum length of the data packet delivered by the MAC layer to the PHY layer and depends on the data rate. The proposed algorithms have been thoroughly tested considering both ad hoc and infrastructure modes of operation, fixed and abruptly changing number of sources as well as Poisson source arrivals. Both error-free and error-prone channels have been considered. Due to space limitations, we will discuss only 2 scenarios exhibiting the most relevant properties of the proposed algorithms.

Scenario A: N nodes are uploading CBR traffic to an access point. Starting with 10 active nodes for a duration of 25s, two 20-node bursts join the network at instants 25s and 50s, and then leave the network consecutively at instants 75s and 100s. Each node generates 1050B-packets every 5ms. Data rate at the physical layer is 2Mbps. There is no error on the wireless channel and the basic access mechanism is used. We have investigated the performance of 3 algorithms beside the standard: the Adaptive BEB, the additive increase/additive decrease algorithm [8] and the multiplicative slow decrease algorithm [7] referred to “Additive” and “Multiplicative” respectively. Figures

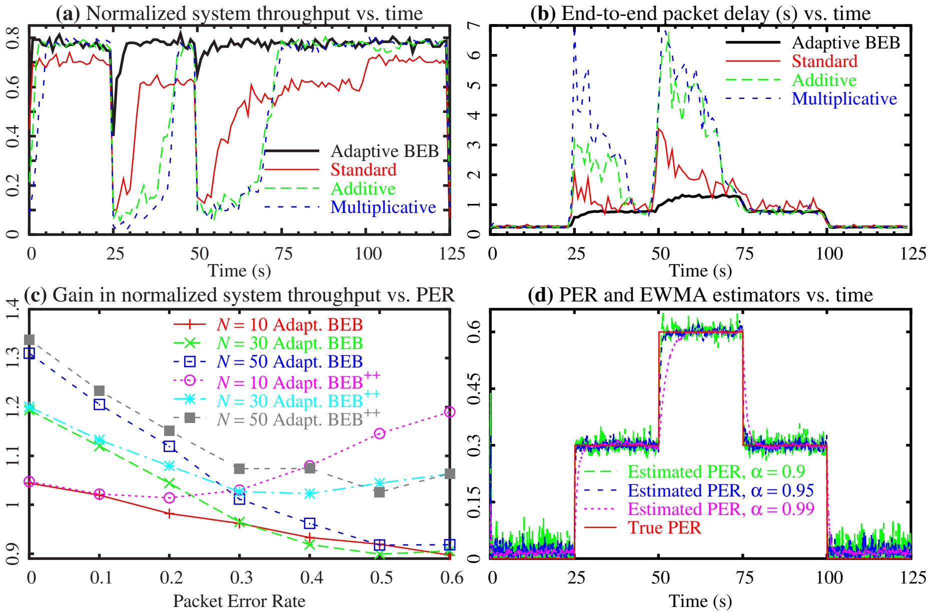

Fig. 4. (a) Simulative system throughput and **(b)** end-to-end delay (in s) over time (scenario A). **(c)** Throughput gain of Adaptive BEB and Adaptive BEB^{++} against the standard (scenario B). **(d)** PER and EWMA estimators vs. time for several α values

4(a) and 4(b) respectively illustrate the system throughput and the packet delay (in seconds) over time, when using the standard and the three adaptive algorithms. As can be seen, Adaptive BEB exhibits a steady performance over the simulation time whether in terms of throughput or packet delay, and is the only one to rapidly recover after the abrupt increase of N at times 25s and 50s. This can be explained by the fact that an optimal $CW_{\min}$ is rapidly selected, consequently avoiding the high number of collisions that are experienced by the standard. The slow decrease approaches [7, 8] perform very well when the number of sources is constant or changes smoothly. However, their performance degrades severely in the scenario at hand for 2 reasons. First, their control algorithms are relatively slow compared to the important change in the network state. Second, in the infrastructure mode of operation, some congestion occurs at the access point buffer yielding lost packets and subsequent missed ACKs. Buffer overflow is penalizing more the slow decrease approaches than the Adaptive BEB as in these approaches the contention window after a successful transmission will be unnecessarily large. In contrast, the Adaptive BEB controls the $CW_{\min}$ by using $\overline{N}$, so that only retransmissions affect the system performance.

Scenario B: There are $2N$ nodes in the network: N sources and N destinations operating in ad hoc mode with a data rate at the physical layer set to 11Mbps. The basic access mechanism is used. Each source generates 1000B-packets every 0.8ms. The error on the channel is modeled as a Bernoulli loss process. Packets are noise-corrupted with a probability PER set constant throughout each simulation runtime (100s). Two algorithms are investigated: the Adaptive BEB and the Adaptive BEB^{++}. Figure 4(c) plots, for different values of N, the throughput gain (defined as the ratio between both throughputs) achieved by each algorithm with respect to the standard versus the PER. Due to its persistence mechanism, Adaptive BEB^{++} shows a high robustness to error on the channel, and the gain obtained is always greater than 1 for different values of PER and N. For instance there is as much as 19% more throughput with Adaptive BEB^{++} when $N = 10$ and $\text{PER} = 0.6$. The Adaptive BEB has previously shown better performance than the standard. This is no longer true at high PER values. Actually, the Adaptive BEB is much more penalized by high PERs since in this case the average backoff time will be much higher than in the standard due to a larger starting contention window. There will be simply too much idle times over the channel.

8 Fairness Analysis

Fairness describes the MAC protocol capability to distribute the available resources equally among communicating terminals. There are a variety of fairness definitions intended to support different QoS and service differentiation. In our study, fairness is simply the equal partitioning of the bandwidth among all flows at hand. We have run simulations when either one of the following algorithms is used: Adaptive BEB or the standard when $\text{PER} = 0$, and Adaptive BEB^{++} or the standard when $\text{PER} = 0.4$. We considered 3 different values of N, namely, 10, 30 and 50. For each simulation, we have computed the index of fairness of Jain et al. [13] $f(x_1, x_2, \ldots, x_I) = \left(\sum_{i=1}^{I} x_i\right)^2 / \left(I \sum_{i=1}^{I} x_i^2\right)$ where x_i denotes user i allocation, and I the number of users sharing the bandwidth. This index returns the percentage of flows being treated fairly. To compute the index of fairness, we use the Sliding Window Method (SWM) introduced in [14]. A small sliding window size allows the study of the short-term fairness whereas large sliding window sizes enable the long-term fairness one. The results are plotted in Fig. 5. Observe that Adaptive BEB (respectively Adaptive BEB^{++}) achieves better fairness, i.e. higher index value, than the standard when $\text{PER} = 0$ (respectively when $\text{PER} = 0.4$), for any investigated value of N. The fact that our algorithm assigns

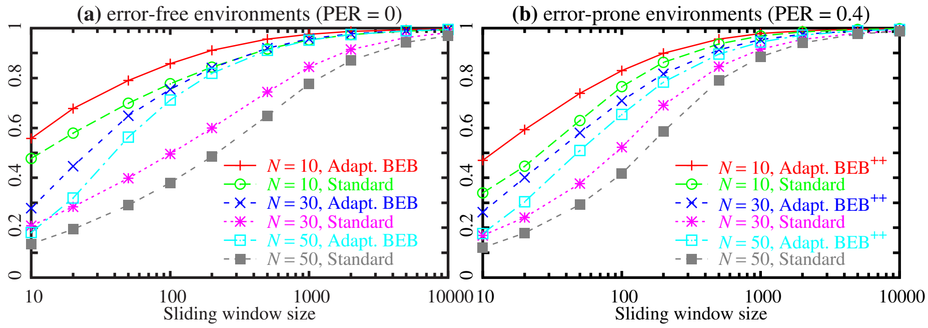

Fig. 5. Fairness index versus the sliding window size

to the contending nodes optimal and relatively equal $CW_{\min}$ allows on one hand to minimize the number of collisions, and consequently the risk of having largely different backoff intervals, and on the other hand, to obtain equal opportunities of accessing the channel for the various nodes.

9 Conclusion

In this paper, we have proposed and evaluated a simple, efficient and robust adaptive mechanism that can be easily incorporated within the IEEE 802.11 DCF function in order to optimize its performance. The proposed algorithm includes two mechanisms. The first optimally adjusts the minimum contention window size to the current network congestion level by using an on-line estimation of the number of active stations. The other mechanism extends the first to enhance its performance in the presence of noisy channels. Simulation results have shown that our proposed algorithm outperforms the standard in terms of throughput, packet delay, fairness and robustness to noise, for different scenarios and configurations. When compared to other proposals, our algorithm has shown better performance under several configurations.

Acknowledgements The second author wishes to thank I. Aad and R. van der Mei for fruitful discussions at an early stage of this work.

References

1. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications **18**(3) (2000) 535–547
2. Cali, F., Conti, M., Gregori, E.: Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. IEEE/ACM Trans. on Networking **8**(6) (2000) 785–799
3. Cali, F., Conti, M., Gregori, E.: IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism. IEEE J. on Sel. Areas in Comm. **18**(9) (2000) 1774–1786
4. IEEE Std. 802.11b, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. (1999)
5. Bianchi, G., Fratta, L., Oliveri, M.: Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In: Proc. of PIMRC ’96. (1996)
6. Bononi, L., Conti, M., Gregori, E.: Runtime optimization of IEEE 802.11 wireless LANs performance. IEEE Transactions on Parallel and Distributed Systems **15**(1) (2004) 66–80
7. Aad, I., Ni, Q., Barakat, C., Turletti, T.: Enhancing IEEE 802.11 MAC in congested environments. In: Proc. of IEEE ASWN ’04, Boston, Massachusetts. (2004)
8. Galtier, J.: Optimizing the IEEE 802.11b performance using slow congestion window decrease. In: Proc. of 16th ITC Specialist Seminar, Anvers, Belgique. (2004)
9. Nadeem, T., Agrawala, A.: IEEE 802.11 DCF enhancements for noisy environments. In: Proc. of PIMRC ’04, Barcelona, Spain. (2004)
10. Bianchi, G., Tinnirello, I.: Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network. In: Proc. of IEEE INFOCOM ’03. (2003)
11. Bertsekas, D., Gallager, R.: Data Network. Prentice Hall (1992)
12. The network simulator, version 2.28 (2005) `http://www.isi.edu/nsnam/ns/`.
13. Jain, R., Hawe, W., Chiu, D.: Quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report DEC-TR-301, DEC (1984)
14. Koksal, C.E., Kassab, H., Balakrishnan, H.: An analysis of short-term fairness in wireless media access protocols (extended abstract). Perf. Eval. Rev., **28**(1) (2000) 118–119