Abstract—This paper presents CSMA/ECA, which combines the efficiency of reservation-based protocols and the simplicity of random access mechanisms. The maximum efficiency of CSMA/CA with optimal parameter adjustment is easily exceeded by CSMA/ECA, even when fixed parameters are used by the latter. CSMA/ECA stations fairly coexist with legacy CSMA/CA and increase the portion of time that is devoted to successful transmissions while decreasing the number of collisions and empty slots. We show that the performance of CSMA/ECA is also clearly superior to the legacy one in lossy channels. The proposed mechanism initially behaves as a CSMA/CA network, but it progressively converges to a collision-free deterministic operation. The convergence process can be modelled as a Markov Chain to assess the duration of the transitory phase.

I. INTRODUCTION

In many communications systems, a broadcast channel is shared by a set of stations. There are different strategies to arrange the sharing, which are called multiple access mechanisms. One option is to divide the resources (time, frequency, carriers or codes) among the different participating nodes. The nodes can also take turns in transmitting, and explicitly signal the end of each turn. Those alternatives prevent that two stations simultaneously transmit.

A popular medium access technique in local area networks is Carrier Sense Multiple Access (CSMA) [1]. The key property of CSMA networks is that the stations listen before transmitting. A station with data ready to transmit senses the channel for a given amount of time and, if the channel is detected idle, the station transmits. It is still possible that collisions occur in CSMA because the propagation of the communication signals is not instantaneous, and real communication systems require a certain amount of time to switch from a listening mode to a transmitting mode.

In CSMA with Collision Avoidance (CSMA/CA), the stations defer their transmission a random number of slots. The efforts to reduce the number of collisions are motivated by the fact that collisions represent a significant waste of resources in wireless networks, since it is not feasible to immediately detect a collision and interrupt the transmission. The stations either transmit or receive, and cannot collect any feedback from the radio channel while they are transmitting.

CSMA/CA combined with truncated Binary Exponential Backoff (BEB) is at the core of the Medium Access Control (MAC) specification in the suite of protocols IEEE 802.11 [2]. These protocols are widely used in Wireless Local Area Networks (WLANs) and, for this reason, they have been the subject of extensive research with the goal of reducing collisions and improving performance.

In spite of the possibility of collisions, CSMA/CA is still an appealing protocol for WLANs. It is lightweight, it takes advantage of statistical multiplexing to accommodate bursty traffic and it can be executed in a distributed fashion. CSMA/CA is especially fitted for networks with a large number of stations that sporadically send one packet. However, CSMA/CA was not designed to benefit from the fact that some stations have multiple-packet messages [3], [4], i.e., stations that store several packets in their transmission queues.

When stations send multiple consecutive packets, it is possible to use the feedback obtained from previous transmissions attempts to adequately schedule future transmissions. For this reason, we suggest a modification to the CSMA/CA protocol that further reduces the number of collisions while maintaining all its versatility and power. We call the new protocol CSMA with Enhanced Collision Avoidance (CSMA/ECA).

The main features of the presented CSMA/ECA protocol are the following:

- The maximum theoretical performance of CSMA/CA can be exceeded using CSMA/ECA.
- It provides a collision-free medium access after a transitory phase.
- It fairly coexists with legacy CSMA/CA.
- It works in a distributed fashion.
- It does not require additional computational efforts and can be easily implemented.
- It is robust against channel errors.

The rest of the paper is organized as follows: Section II defines the CSMA/ECA algorithm, then in Section III a Markov Chain model to predict the length of the transitory phase is described. Implementation issues and the performance evaluation results are discussed in Section IV while Section V presents an overview of the related work in the area. Finally, some conclusions are given.

II. ENHANCED COLLISION AVOIDANCE

In CSMA/CA, whenever there are backlogged stations with a packet ready to be transmitted, the channel time is implicitly divided into slots. Three different kinds of slots are differentiated: empty, successful and collision. A slot is empty when no station attempts transmission; successful if one (and only
one) station transmits; and collision if more than one station simultaneously transmit. The channel time spent in empty slots or collision slots is wasted.

Whenever a station has to defer its transmission, it chooses a random backoff value $B$ from a contention window.

$$B \sim U[0, CW - 1],$$

where $U$ is the uniform distribution and $CW$ is the contention window.

We consider that the stations are saturated (i.e. the stations always have a packet ready to transmit). As a consequence, the stations are either transmitting, receiving or backing off; they are never idle. After each transmission attempt, the stations choose a backoff value. The stations have to backoff both after collisions and successful transmissions. For the first case, the backoff has to be necessarily random to prevent a new collision in the retransmission attempt. However, for the second case, the backoff value can be deterministically selected.

### A. Deterministic Backoff After Successful Transmissions

By choosing a deterministic backoff after a successful transmission and a random backoff otherwise, the system converges to a collision-free operation when the number of active stations is not greater than the value of the deterministic backoff. In the case of a successful transmission, the deterministic behaviour stabilizes the system (hopefully leading to another success). Conversely, if there is a collision, the randomness of the backoff provides a change that would (desirably) avoid more collisions. The system exploits the information gathered from previous transmission attempts to further reduce the collisions, thus we call it Enhanced Collision Avoidance (ECA). The terminals perform a random search to find free slots, until collisions disappear.

It has to be clear that a station keeps using a deterministic backoff after each successful transmission, until a collision occurs. As soon as it suffers a collision, the station moves back to the random behaviour. The collision will always be caused by a station that randomly selected its transmission slot, since collisions among stations that behave deterministically are not possible.

This principle can be better understood by means of an example. Consider the simplest case of two stations ($STA 0$ and $STA 1$) contending for a channel, as shown in Fig. 1. The channel time advances from left to right and it is divided in slots. Even though the actual duration of empty, successful and collision slots differ, all the slots are equally represented in Fig. 1 for simplicity reasons. The upper channel time line corresponds to legacy CSMA/CA, while the lower one incorporates the modifications we have proposed for CSMA/ECA. The deterministic backoff after successes is a value that depends on the 802.11 flavor, as will be explained in subsection IV-A. In the example depicted in Fig. 1, a value equal to sixteen is used.

In the figure, the transmission attempts are represented as shaded boxes. Additionally, the figure also shows the backoff value chosen by each station (between brackets). The label indicates whether the backoff has been chosen randomly or deterministically.

In the example, the two CSMA/CA stations collide, then successfully transmit and, finally, collide again. When CSMA/ECA is used, collisions disappear after all stations have successfully transmitted, because the backoff is selected deterministically. It is useful to imagine a virtual frame$^1$ of $V$ slots (represented with a dotted line in the figure) and observe that, after collisions disappear, the stations transmit in fixed slot positions within the virtual frame, similarly to a TDMA operation.

Algorithm 1 represents the protocol that is distributedly executed in each of the contending stations. The meaning of each of the variables is as follows:

- $b$ is the backoff counter.
- $CW_{\text{min}}$ is the minimum contention window.
- $CW_{\text{max}}$ is the maximum contention window.
- $a$ is the number of transmission attempts.
- $A$ is the maximum number of transmission attempts.
- $V$ is the deterministic backoff value after successful transmissions.

```plaintext
Algorithm 1: CSMA/ECA

1 /* Initialize $b$. */
2 $b \leftarrow U[0, CW_{\text{min}} - 1]$;
3 while there is a packet to transmit do
4    /* Initialize $a$. */
5    $a \leftarrow 0$;
6    while $a < A$ do
7        /* First, backoff. */
8            while $b > 0$ do
9                wait 1 slot;
10               $b \leftarrow b - 1$;
11            end
12        Attempt transmission;
13        if success then
14            /* Deterministic backoff. */
15                $b \leftarrow V$;
16            break;
17        else
18            /* If transmission fails. */
19                $a \leftarrow a + 1$;
20                /* Random backoff value. */
21                $b \leftarrow U[0, \min(CW_{\text{min}} * 2^a, CW_{\text{max}}) - 1]$;
22        end
23    end
24 end
```

### B. Efficiency of CSMA/ECA during Steady-State Operation

Let us define the channel efficiency ($\phi$) as the fraction of channel time that is devoted to successful transmissions.

$^1$Some works refer to data-link layer PDUs as frames. In this article, a frame is a group of slots. Data-link layer PDUs are called packets.
The concept of active (or contending) station deserves some further clarification. In a typical network, some of the stations are registered to a given access point. As an example, if 50 stations with an activity rate of 10% share a given frequency band, the expected number of simultaneous contenders is 5.

The efficiency of CSMA/ECA is computed as presented in (3) using $V = 16$, the upper bound for CSMA/CA with dynamic parameter adjustment is obtained from [5], and the performance of plain CSMA/CA is computed using the approach in [6]. The simulation results are obtained using a custom simulator$^2$ in Octave, that includes only the MAC layer.

Before reaching the steady-state and obtaining the efficiency as presented in (3), the system goes through a transitory operation. The efficiency obtained in the transitory operation is a value between the efficiency delivered by CSMA/CA and $\phi = \frac{P_s T_s}{P_c T_c + P_s T_s + P_c T_c}$.

where $P_c$, $P_s$ and $P_e$ are the empty, success and collision probabilities, respectively. And $T_c$, $T_s$ and $T_e$ are the duration of an empty, successful and collision slot, respectively.

Then, for a number of contending stations ($\varsigma$) not greater than the size of the virtual frame, the efficiency that can be obtained from CSMA/ECA in steady-state collision-free operation is:

$$\phi = \frac{\varsigma \cdot T_s}{\varsigma \cdot T_s + (V - \varsigma) \cdot T_e}; \varsigma \leq V.$$  

$$P(X(t)) = \sum_{i=0}^{n} P_i e^{-\lambda(t)} \left( \frac{\lambda(t)}{i!} \right)$$

The upper theoretical maximum of CSMA/CA with transitory and stationary operation are represented as thick dots with bars to account for the 95% confidence interval. The IEEE 802.11b standard has been assumed, together with a data rate of 2 Mbps and a packet size equal to 1500 bytes. The efficiency is represented for an increasing number of contending stations. The concept of active (or contending) station deserves some further clarification. In a typical network, some of the stations are actively sending data while others are idle. Usually, the number of active stations is only a fraction of the stations that are registered to a given access point. As an example, if 50 stations with an activity rate of 10% share a given frequency band, the expected number of simultaneous contenders is 5.

The efficiency in (3), because only a fraction of the collisions is avoided. During this transitory phase, the number of stations that successfully transmit (and thus use a deterministic backoff) is a random variable. In the next section, the evolution of this number is modelled as a Markov Chain in order to draw additional conclusions about the transition process.

### III. A Dissection of the Convergence Process

Consider a scenario with $\varsigma$ saturated stations and a virtual frame size of $V$ slots, $2 \leq \varsigma \leq V$. We will assume that the transition process occurs in a frame-by-frame basis. Let $X_n$ be the random variable that represents the number of stations that successfully transmitted in the frame $n$. Then we can model the transition process as a time-homogeneous Markov Chain whose state space is

$$S = \{S_i | 0 \leq i \leq \varsigma\}$$

As the system runs, it transitions from an initial state $S_0$ to a (stable) state $S_\varsigma$.

We are interested in computing the transition probability matrix $P$ which is the matrix of one step transition probabil-
properties $p_{i,j}$ defined by
\[ p_{i,j} = Pr(X_{n+1} = j | X_n = i) \quad 0 \leq i, j \leq \varsigma. \quad (5) \]

Before dealing with the general computation of $p_{i,j}$, we will analyze some results that immediately arise from the definition of the problem and provide some insights about the behaviour of the model. Note that the following properties apply only to the model, and not necessarily to the system that is being modelled. However, they are helpful in computing the transition matrix for the model.

**Claim 1:** The system is stable when $X_n = \varsigma$, i.e. state $S_\varsigma$ is absorbing.

\[ Pr(X_{n+1} = \varsigma | X_n = \varsigma) = 1. \quad (6) \]

**Proof:** $X_n = \varsigma$ implies that all the stations successfully transmitted in virtual frame $n$. Therefore, all the stations will deterministically choose the transmission slot in virtual frame $n+1$, specifically they will transmit in the same position in the frame $n+1$ as they did in virtual frame $n$. As there were no collisions in frame $n$, there will be no collisions in frame $n+1$.

**Claim 2:** It is not possible that there is one and only one station that randomly selects the transmission slot in a given virtual frame.

\[ Pr(X_n = \varsigma - 1) = 0; \quad n > 0. \quad (7) \]

**Proof:** Seeking a contradiction we assume that there is only one station that randomly selects the transmission slot in virtual frame $n$. This implies that this station suffered a collision in the previous frame $n-1$. Since a collision occurs when a minimum of two stations transmit in the same slot, there are at least two stations that will randomly select the transmission slot in virtual frame $n$. This contradicts our assumption.

A. Computing the Transition Probability Matrix

After these preliminary results, we face the general problem of computing $p_{i,j}$, i.e. the probability that we have $j$ successful transmissions in the current virtual frame given that there were $i$ successes in the previous frame. There are $i$ stations that deterministically transmit in $i$ different slots, while the rest of the stations ($\varsigma - i$) randomly transmit in any of the $V$ slots.

Note that for the special case $i = 0$, the problem is reduced to the computation of the number of successes that are obtained when $\varsigma$ stations transmit in $V$ slots and can be solved using the model suggested in [7]. For any other value of $i$ ($i \neq 0$), the approach in [7] is no longer applicable, since it assumes that there are slots reserved for the stations that successfully transmitted in the previous frame. Hence, we are interested in finding another scheme that can be used for any value of $i$, $V$ and $\varsigma$.

For large values of $V$ and $\varsigma$, a brute force approach that sweeps all the different combinations to obtain the transition probability matrix $P$ is computationally impractical. To compute the first row of the transition matrix, it would be necessary to consider $\varsigma$ stations that could transmit in any of the $V$ available slots, which would account for $V^\varsigma$ possibilities.

Nevertheless, certain shortcuts are possible to accelerate the computation of $P$. The reason is that we are interested only in the number of successful slots in a virtual frame, but not in which are those successful slots. In other words, the slots are interchangeable. Similarly, we are not interested in which are the stations that successfully transmitted; all the stations are equivalent from our point of view.

By using the aforementioned interchangeability properties, we propose the following method to compute $P$. Assume that the previous state is $S_0$ and we want to compute the probabilities $p_{i,j}$ for all values $0 \leq j \leq \varsigma$. Now consider a transmission in the current frame. This transmission can be in any of the $V$ (for now, empty) slots. Since all these slots are empty, the $V$ possible outcomes are equivalent for our analysis. Each of the $V$ outcomes consists of a slot with one transmission and $V - 1$ empty slots. Following the same reasoning, for a second transmission in the same virtual frame, there are only two possible outcomes: $a)$ that the transmission slot is the same as the one selected for the first transmission (which occurs with probability $1/V$) or $b)$ the two transmissions are in different slots (which occurs with probability $(V - 1)/V$). These steps can be repeated to build a tree and obtain all the possible outcomes of interest and the probabilities associated with each outcome. A graphical example is presented in Fig. 3.

In Fig. 3 we show an example for $\varsigma = 3$ and $V = 4$. It is a tree with $\varsigma + 1$ levels. The root represents the $V = 4$ empty slots, and in every level, a new transmission (represented as a ball) is included. The levels are labeled as $\{0\}$, $\{1\}$, $\{2\}$ and $\{3\}$. The edges of the tree are labeled with probability values. At the first level, there is only one node, since the only possible situation (with only one transmission) is one success and three empty slots. Therefore, the edge from the root to the node at the first level is labeled with probability 1. In the transition from level $\{1\}$ to level $\{2\}$ there are two possible options: $a)$ that the two transmissions occur in the same slot (with probability $1/4$) and $b)$ that the transmissions occur in different slots (with probability $3/4$). This process is iterated until all the transmissions are included, and 4 leafs are obtained. By following the path from the root to the leaf,
the probability of each leaf is computed. The probability that no station successfully transmits can be obtained from the first leaf. From the tree it can be observed that the transition probability from state $S_0$ to state $S_0$ (i.e. the probability of having zero successes in frame $n+1$ given the fact that there were zero successes in the frame $n$) is

$$p_{0,0} = Pr(X_{n+1} = 0 | X_n = 0) = \frac{1}{16}. \quad (8)$$

The probability that there is only one success is $p_{0,1} = \frac{3}{16} + \frac{6}{16} = \frac{9}{16}$. The probability of two successes is zero $p_{0,2} = 0$ and the probability of three successes is $p_{0,3} = \frac{6}{16}$. With these values, we have already completed the first row of the transition matrix $P$. To obtain the values for the second row, one has to assume that there was a successful collision in the previous virtual frame. Therefore, we consider only a subtree of the tree represented in Fig. 3, particularly the one with the root at the node of level $\{1\}$. To compute the third row of the matrix we use as a root the lower node of level $\{2\}$. The last row is computed using only one node, which is the lowest leaf. The transition matrix which is obtained for this example is:

$$P_{c=3, V=4} = \begin{pmatrix}
\frac{1}{16} & \frac{9}{16} & 0 & \frac{6}{16} \\
\frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\
0 & \frac{1}{2} & 0 & \frac{1}{2} \\
0 & 0 & 0 & 1
\end{pmatrix} \quad (9)$$

It is not a coincidence that the first two rows of $P_{c=3, V=4}$ are the same. Actually, since in level $\{0\}$ all the slots are empty and thus equivalent, there is only one way to place the first ball (transmission). As a consequence, there is always only one node in level $\{1\}$, and the single edge from level $\{0\}$ to level $\{1\}$ takes the value 1.

**Claim 3:** The first two rows of the transition probability matrix $P$ are equal.

$$p_{0,j} = p_{1,j} \quad 0 \leq j \leq \varsigma \quad (10)$$

**Proof:** Consider a tree as the one exemplified in Fig. 3. Then take the subtree with the node of level $\{1\}$ as a root. From this tree, we can obtain the values of the second row (indexed as 1) of the transition matrix $P_{1,j}$. Now, to obtain the first row (indexed as 0), we observe that we use exactly the same tree, but with an additional edge with value 1 and an additional node as a root. Then we can obtain the values of the first row by multiplying the values of the second row by one.

We are interested in evaluating how long does it take for the system to leave the transitory phase and begin the collision-free operation. We consider an initial state $S_0$ in which all the stations randomly choose their transmission slot and then we use the transition matrix $P$ to evaluate the marginal distributions in subsequent frames. Let

$$\pi_n = \{Pr(X_n = i), 0 \leq i \leq \varsigma\} \quad (11)$$

be the vector of the marginal probabilities at stage $n$, and $\pi_0 = [1, 0, ..., 0]$ the initial vector. This means that the initial state is $S_0$ with probability 1. Then the vector $\pi_n$ can be obtained by:

$$\pi_n = \pi_0 P^n. \quad (12)$$

The last component of vector $\pi_n$ is precisely the value of interest for our study $Pr(X_n = \varsigma)$, which is the probability that the system has reached the stable collision-free state. One particularity of our evaluation of the transition curve is that we have considered that the transition step contains 2 * $V$ slots i.e. two virtual frames. This is an approximation of the expected backoff of those stations that suffered a collision. We are implicitly assuming that the probability that the same station suffers multiple successive collisions is low, which is true for low values of $\varsigma$. As the value of $\varsigma$ approaches the value of $V$, the assumption is no longer valid.

**B. Validation by Simulation**

The model presented above is based on two approximations with respect to the actual CSMA/ECA operation. The first one is that, in the model, the convergence process occurs in a frame-by-frame basis. In contrast, the CSMA/ECA algorithm allows that the same station re-attempts transmission (and eventually succeeds) in the same virtual frame. Actually, the virtual frame concept is not intrinsic of CSMA/ECA and it is an abstraction we have used for the analysis. The second concession to simplicity is that the exponential growing of the contention window has been neglected in our model. As a consequence of these two concessions (frame-by-evolution and static contention window), our model provides only an approximation to the expected behaviour of CSMA/ECA.

At this point it should be clear that, being the transitory operation a random convergence process, only probabilistic guarantees can be offered regarding its duration. Fig. 4 plots the probability that the system has reached the collision-free operation in a given slot. The results obtained from the model are compared to those obtained from simulation. It can be observed that the transition process is slower for higher values of $\varsigma$.

**C. Disruption of the Stationary Operation**

Although the system is expected to run in the collision-free mode of operation for most of the time, there are two events that can disrupt the stationary operation: a channel error and a new entrant. The model can be used to assess the recovery curves associated with these events. It is necessary to force the initial state to $S_{-1}$. Regardless of the fact that the system will never transition to $S_{-1}$, it is possible to use it as an initial state. It precisely reflects the fact that all stations but one are using a deterministic backoff. The initial vector under consideration is: $\pi_0^D = [0, ..., 0, 1, 0]$.

And the marginal probabilities of subsequent steps:

$$\pi_n^D = \pi_0^D P^n. \quad (13)$$
Provided that current state is $S_i$, we use the maximum number of collisions (worst case) in the previous step as an approximation of the actual number of collisions in the previous step:

$$\kappa_i \approx \lceil \frac{c - i}{2} \rceil. \tag{14}$$

where $\lceil \cdot \rceil$ is the floor operator. Then, using the approximation $T_c \approx T_s$, the efficiency of the system in the step $n - 1$ is:

$$\phi_{n-1} \approx \sum_{i=0}^{c} \frac{2 \cdot i \cdot T_s}{(2 \cdot i + \kappa_i) \cdot T_s + (2 \cdot V - 2 \cdot i - \kappa_i) \cdot T_c} \approx D_n(i), \tag{15}$$

where the expectation of the backoff of those stations that suffer collisions is considered to be twice as much as $V$.

Fig. 5 shows the recovery curves obtained from (13)-(15). The transitory phase associated with new incorporations to the contention can be avoided by means of Smart Entry, which will be described in Subsection IV-B.

IV. IMPLEMENTATION ISSUES

In this section we address the coexistence of CSMA/ECA with the legacy protocol. We also propose a rational way to join the contention and study the performance of the system when we release the ideal channel assumption.

A. Coexistence with legacy CSMA/CA

A promising field of application of the proposed CSMA/ECA is the successful protocol suite IEEE 802.11. Nevertheless, given the large number of deployed networks and terminals, any new version of the medium access control algorithm should be backward compatible with the already existing equipment. Furthermore, to guarantee the smooth coexistence of new and legacy stations, those stations running CSMA/ECA should consume a fair amount of the available bandwidth.

The only difference between CSMA/CA and CSMA/ECA as presented in Algorithm 1 can be found in line 11. CSMA/CA randomly chooses the backoff value from the minimum contention window ($b \leftarrow U[0, CW_{\text{min}} - 1]$), while CSMA/ECA deterministically chooses as a value the size of the virtual frame ($b \leftarrow V$). In order to fairly compete with legacy stations, it is desired that

$$V = \lceil E[U[0, CW_{\text{min}} - 1]] \rceil, \tag{16}$$

where $E[\cdot]$ represents the expectation operator and $\lceil \cdot \rceil$ is the ceiling operator. This selection of the virtual frame size guarantees that the expected number of slots that a station waits after a successful transmissions is approximately the same for both CSMA/CA and CSMA/ECA.

To validate this idea, we performed simulations for a scenario in which half of of the stations run CSMA/CA while the other half use CSMA/ECA. The values chosen for the MAC parameters are $CW_{\text{min}} = 32$ and $V = 16$. The rest of the parameters are taken from the IEEE 802.11b specification. Each simulation runs for 10000 slots and each scenario is repeated ten times. The number of competing stations range from two to forty (only even values are considered). When a value of 40 stations is indicated, it actually means 20 CSMA/ECA stations plus 20 CSMA/CA stations.

The plot in Fig. 6 presents the results in three different curves. The first curve, marked with boxes, shows the overall channel efficiency. The second curve, marked using circles, is the amount of channel time devoted to successful transmissions of the CSMA/ECA stations. Finally, the third curve is marked using triangles and is the amount of channel time devoted to successful transmissions of CSMA/CA stations.

It can be observed that CSMA/ECA flows obtain higher channel utilization than CSMA/CA flows thanks to the reduced collision probability. This small advantage can be seen as an incentive for legacy stations to shift to CSMA/ECA for the greater benefit of the network. Jain’s index [8] can be used to assess the fairness of the system:

$$\text{fairness} = \frac{(\phi_{\text{csma/eca}} + \phi_{\text{csma/ca}})^2}{2(\phi_{\text{csma/eca}}^2 + \phi_{\text{csma/ca}}^2)}. \tag{17}$$
Fig. 6. Half of the stations run CSMA/ECA, while the other half run CSMA/CA. The figure shows the channel utilization achieved by each group.

The possible outcomes range from 0.5 (worst case) to 1 (best case). We obtained results higher than 0.98 when comparing the channel utilization of CSMA/ECA and CSMA/CA in a mixed scenario.

The benefits of using CSMA/ECA are greatly diminished in the presence of legacy stations since the collision-free operation is never reached. Nevertheless, a network running a mixture of CSMA/CA and CSMA/ECA stations will offer equal or better performance than a pure CSMA/CA network, since some of the collisions will be avoided.

To assess the benefits of using CSMA/ECA we simulate three different scenarios, namely, a pure CSMA/ECA, a mixed CSMA/ECA and CSMA/CA and a pure CSMA/CA. The results are compared in Fig. 7. Note that there is a curve (marked with squared boxes) that is common in Fig. 6 and in Fig. 7.

It can be observed that, thanks to the enhanced collision avoidance mechanism, a larger fraction of the channel time is devoted to successful transmissions when only CSMA/ECA is used. For a number of active stations up to the size of the virtual frame size $V$, the efficiency is almost 1.

B. Smart Entry

So far we have assumed that the number of contenders is fixed. Nevertheless, in a real network, the stations join and leave the contention depending on the load that they receive from the upper layers of the protocol stack.

Ideally, the system will run in the collision-free stable mode of operation. At this point, if a station that joins the contention selects the first transmission slot randomly, it poses the collision-free mode of operation of the system at risk: it may provoke a collision and move the system back to its transitory (collision-prone) mode of operation. To avoid this situation, the stations that are not actively contending for the channel should keep track of the empty slots in each virtual frame. When one of those stations receives a packet from the upper layer, it already knows which slots are expected to be empty, and can schedule the first transmission accordingly.

If Smart Entry is to be used, the first line of Algorithm 1 has to be substituted by Algorithm 2. It includes an array called slotNumber[] to keep track of the status of each slot of the frame. The size of this array is precisely the size of the virtual frame $V$. With the modification presented in Algorithm 2, a station joining the contention transmits in the first empty slot.

Note that while the station is delaying the first transmission attempt, it marks the positions in the array as free. This behaviour prevents a deadlock in the case in which all the slots are busy. If there are no free slots, the station will delay its transmission attempt $V$ slots, and then deliberately prompt a collision in order to free some slots for a future transmission attempt.

C. Releasing the Ideal Channel assumption

Stations cannot distinguish channel errors from collisions, and therefore use a random backoff after all packet losses. We want to stress the proposed protocol by introducing packet errors with probability of $10^{-2}$. This 1% threshold was used as a standard measure of robustness by the IEEE 802.11 committee [9].

The simulations in Fig. 8 are performed in the presence of imperfect channel conditions. The packet errors are treated as collisions by the stations and, hence, interfere in the enhanced collision avoidance mechanism. It can be observed that CSMA/ECA clearly outperforms CSMA/CA in lossy channels.

V. RELATED WORK

In [5] it is shown that there is a fundamental limit on the efficiency of completely random access protocols, in which the transmission slot is chosen without using any prior information. Then, it is explained that CSMA/ECA can overcome this limit by using a random behaviour after failures (to trigger
/* Initialize slotNumber[] */
for i ← 0 to V - 1 do
    slotNumber[i] ← unknown;
end
i ← 0;
/* Scan the channel while waiting for a packet from the upper layers. */
while True do
    if there is a packet ready to transmit then
        if slotNumber[i] is free then
            transmit;
            /* Leave Smart Entry and move to normal CSMA/ECA operation. */
            break;
        else
            wait 1 slot;
            slotNumber[i] ← free;
        end
    else
        if channel sensed busy then
            slotNumber[i] ← busy;
        else
            slotNumber[i] ← free;
        end
    end
i ← (i + 1) (mod V);
end

Algorithm 2: Smart Entry

Fig. 8. The channel efficiency delivered by CSMA/ECA and CSMA/CA in an unreliable channel.

a change) and a deterministic behaviour after successes (to stabilize the system).

In [10], simulations are used to assess the performance of CSMA/ECA in saturated, non-saturated and hybrid (a combination of saturated and non-saturated) scenarios. CSMA/ECA is shown to perform equal or better than CSMA/CA in all the considered scenarios. Specifically, the two protocols deliver the same throughput in those scenarios in which the network is able to absorb all the offered traffic. However, when the traffic load overwhelms the network, CSMA/ECA performs better than CSMA/CA. Traffic prioritization in CSMA/ECA is addressed in [11]. An analytical model to capture the behaviour of CSMA/ECA in stationary operation for both rigid and elastic flows is presented in [12].

VI. CONCLUSIONS

In this article we address the problem of collisions in CSMA networks. Our finding is that, instead of using a random backoff after all transmission attempts, it is better to use a random one after collisions and a deterministic one after successes. It reduces the chances of collisions as soon as two or more stations successfully transmit. As the system runs, it progressively converges to a collision-free operation that considerably improves the channel efficiency.

The proposed protocol outperforms CSMA/CA and, in the most typical scenarios, it even surpasses the theoretical upper bound associated with CSMA/CA networks that allow for dynamic parameter adjustment. Additionally, CSMA/ECA does not add any additional complexity to the implementation, it can fairly coexist with already deployed networks and it is robust against unreliable channel conditions.

ACKNOWLEDGMENTS

The authors benefited from the discussion with A. Banchs, B. Bellalta, A. Lozano and A. Vinel. This work was partially supported by the Spanish Government under grant TEC2008-06055/TEC.

REFERENCES