Building The Next MAC for WLANs

Luis Sanabria-Russo

NeTS Research Group at

Universitat Pompeu Fabra, Barcelona, Spain

`Luis.Sanabria@upf.edu`

Abstract—Collisions are a main cause of throughput degradation in WLANs. The current contention mechanism used in IEEE 802.11 networks is called Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). It uses a Binary Exponential Backoff (BEB) technique to randomize each contender attempt of transmitting, effectively reducing the collision probability. Nevertheless, CSMA/CA relies on a random backoff that while effective and totally distributed, in principle is unable to completely eliminate collisions; degrading the network throughput as more contenders attempt to share the channel. Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) is able to create a collision-free schedule in a totally distributed manner by means of picking a deterministic backoff after successful transmissions. Further, by applying simple extensions to the backoff mechanism of CSMA/ECA it is able to support many contenders in a collision-free schedule, surpassing the achieved throughput of CSMA/CA and providing short-term throughput fairness among contenders.

This work describes CSMA/ECA, its mechanisms to achieve a collision-free schedule with many contenders as well as its performance when coexisting with CSMA/CA nodes. Further, the effects of imperfect clocks over CSMA/ECA's deterministic backoff mechanism and its consequences when attempting to implement the protocol in real hardware are also analysed.

Index Terms—CSMA/ECA, WLAN, MAC, Collision-free, Clock Drift.

I. INTRODUCTION

Wireless Local Area Networks (WLANs or IEEE 802.11 networks [1]) are a popular solution for wireless connectivity, whether in public places, work environments or at home. This technology works over an unlicensed spectrum in the Industrial, Scientific and Medical (ISM) radio bands (at around 2.4 or 5 GHz), offering a good tradeoff between performance and costs; which is a main reason for its popularity.

The Medium Access Control (MAC) scheme used in WLANs is called Distributed Coordination Function (DCF) and is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. It has been widely adopted by manufacturers and consumers, making it very cheap to implement and an ubiquitous technology[1]. Nevertheless, the ever-growing throughput demands from upper layers have proven to be limited by WLANs' MAC [2], which by its nature is prone to collisions that degrade the overall performance as more nodes join the network.

The research community has pushed forward many alternatives to the current MAC in WLANs [3]–[13], but when a proposal deviates too much from CSMA/CA or time-critical operations are modified, its hardware implementation

---

[1]DCF and CSMA/CA will be used interchangeably throughout this work

as part of WLANs' MAC often becomes unlikely [14]; the standardization process taking many years without certainty of approval [2].

A CSMA/CA replacement should be able to provide advantages in terms of throughput, spectrum efficiency and handle many contenders without sacrificing short-term throughput fairness. Furthermore, to support the existing user base and ease its implementation on real hardware, the new MAC protocol should be backwards compatible.

A suitable candidate, and the one to be evaluated in this work, is called Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) [7]. It is capable of attaining higher throughput than CSMA/CA by making a modification to the contention mechanism. In CSMA/ECA, nodes pick a deterministic backoff after successful transmissions; constructing a collision-free schedule among successful contenders. This backoff mechanism ensures that more channel time is spent on successfull transmissions rather than recovering from collisions, thus increasing the throughput of the network. Further enhancements, like *Hysteresis* and *Fair Share* [15] allowed CSMA/ECA to support many more contenders in a collision-free schedule without compromising short-term fairness.

Although many studies have been made analysing the performance of CSMA/ECA [7], [8], [15], [16], neither assesses the protocol's backwards compatibility property under different traffic conditions. Furthermore, the impact of imperfect clocks over the deterministic backoff mechanism is also lacking.

This work serves as an extension to [15], and provides the first simulation results on achieved throughput and suffered delay of CSMA/ECA under different traffic conditions. Further, a short overview of a real hardware implementation of CSMA/ECA alongside future challenges is provided prior the conclusions.

An overview of similar distributed and collision-free MAC protocols for WLANs is provided in Section II. CSMA/ECA, as well as its properties for allocating many contenders in a collision-free schedule are explained in Section III. Section IV details the simulation environment for testing CSMA/ECA, while Section V explains the results. An overview a CSMA/ECA real-hardware implementation is compiled in Section VI, followed by conclusions in Section VII.

II. RELATED WORK

Time in WLANs is divided into tiny empty slots of fixed length σ_e, collisions and successful slots of length σ_c and σ_s,

respectively. Collision and successful slots contain collisions or successful transmissions, making them several orders of magnitude larger than empty slots ($\sigma_e \ll \min(\sigma_s, \sigma_c)$). One of the effects of collisions is the degradation of the network performance by wasting channel time on collisions slots.

Big advances in the WLANs PHY [2], [17] push the community towards the development of MAC protocols able to take advantage of a much faster PHY. By reducing the time spent in collisions nodes are able to transmit more often, which in turn translates to an increase in the network throughput. Further, the upcoming MAC protocols for WLANs should work without message exchange between contenders, that is, work in a totally distributed fashion in order to avoid injecting extra control traffic that may reduce the data throughput.

The followings are MAC protocols suitable for replacing the current standard in WLANs, distributed and capable of attaining greater throughput than CSMA/CA by constructing a collision-free schedule.

A. Zero Collision MAC

Zero Collision MAC (ZC-MAC) [18] achieves a zero collision schedule for WLANs in a totally distributed way. It does so by allowing contenders to reserve one empty slot from a predefined virtual schedule of N-slots in length. Backlogged stations pick a slot in the virtual cycle to attempt transmission. If two or more stations pick the same slot in the cycle, their transmissions will eventually collide; forcing the involved contenders to randomly and uniformly select other empty slot from those detected empty in the previous cycle plus the slot where they collided. When all M stations reserve a different slot, a collision-free schedule is achieved.

ZC-MAC is able to outperform CSMA/CA under different scenarios. Nevertheless, given that the length of ZC MAC's virtual cycle has to be predefined without actual knowledge of the real number of contenders in the deployment, the protocol is unable to provide a collision-free schedule when $M > N$. Furthermore, if N is overestimated ($N \gg M$), the fixed-width empty slots between each contender's successful transmission are no longer negligible and contribute to the degradation of the network performance.

B. Learning-MAC

Learning-MAC [19] is another MAC protocol able to build a collision-free schedule for many contenders. It does so defining a *learning strength* parameter, $\beta \in (0, 1)$. Each contender starts by picking a slot for transmission s of the schedule n of length C at random with uniform probability. After a contender picks slot $s(n)$, its selection in the next schedule, $s(n + 1)$, will be conditioned by the result of the current attempt. (1) and (2) extracted from [19] show the probability of selecting the same slot $s(n)$ in cycle $n + 1$.

$$\left.\begin{array}{r} p_{s(n)}(n+1) = 1, \\ p_j(n+1) = 0, \end{array}\right\} \quad Success \quad (1)$$

$$\left.\begin{array}{r} p_{s(n)}(n+1) = \beta p_{s(n)}(n), \\ p_j(n+1) = \beta p_j(n) + \dfrac{1-\beta}{C-1}, \end{array}\right\} \quad Collision \quad (2)$$

for all $j \neq s(n)$, $j \in \{1, \ldots, C\}$. That is, if a station successfully transmitted in $s(n)$, it will pick the same slot on the next schedule with probability one. Otherwise, it follows (2).

The selection of β implies a compromise between fairness and convergence speed, which the authors determined $\beta = 0.95$ to provide satisfactory results.

L-MAC is able to achieve higher throughput than the current MAC with a very fast convergence speed. Nevertheless, the choice of β suppose a previous knowledge of the number of empty slots ($C - N$, where N is the number of contenders), which is not easily available to the current MAC or may require a centralised entity [13].

Further extensions to L-MAC introduced an *Adaptative* schedule length in order to increase the number of supported contenders in a collision-free schedule. This adaptive schedule length is doubled or halved depending on the presence of collisions or many empty slots per schedule, respectively. The effects of reducing the schedule length may provoke a re-convergence phase which can result in short-term fairness issues.

III. CARRIER SENSE MULTIPLE ACCESS WITH ENHANCED COLLISION AVOIDANCE (CSMA/ECA)

CSMA/ECA [7] is a totally distributed and collision-free MAC for WLANs. It differs from DCF in that it picks a deterministic backoff, $B_d = CW_{\min}/2$ after successful transmissions; where $CW_{\min}$ is the minimum contention window of typical value $CW_{\min} = 16$. By doing so, contenders that successfully transmitted on schedule n, will transmit without colliding with other successful nodes in future cycles.

Collisions are handled as in DCF, which is described in Algorithm 1. In Algorithm 1, the node starts by setting the retransmissions counter and backoff stage to zero ($r \in [0, R]$ and $k \in [0, m]$ respectively; m is the maximum backoff stage of typical value $m = 5$, and R is the retransmissions limit with typical value $R = m + 1$), then generates a random backoff, B. When the Acknowledgement (*ack*) for a sent packet is not received by the sender a collision is assumed. Upon collision, the involved nodes will double their contention window by incrementing their backoff stage in one and picking a random backoff, $B \in [0, 2^k CW_{\min}]$. This procedure is described between Line 10 and 13 of Algorithm 1.

Algorithm 2 provides an explanation of CSMA/ECA deterministic backoff mechanism, which main difference with CSMA/CA (and therefore with Algorithm 1) relies on the selection of a deterministic backoff after a successful transmission (compare Line 18 in Algorithm 1 with Line 18 in Algorithm 2). Figure 1 shows an example of CSMA/ECA dynamics with four contenders.

In Figure 1, the *STA-#* labels represent stations willing to transmit. The horizontal lines represent a time axis with each number indicating the amount of empty slots left for the backoff to expire. Stations willing to transmit begin the contention for the channel by picking a random backoff, B. The first outline highlights the fact that stations STA-3 and
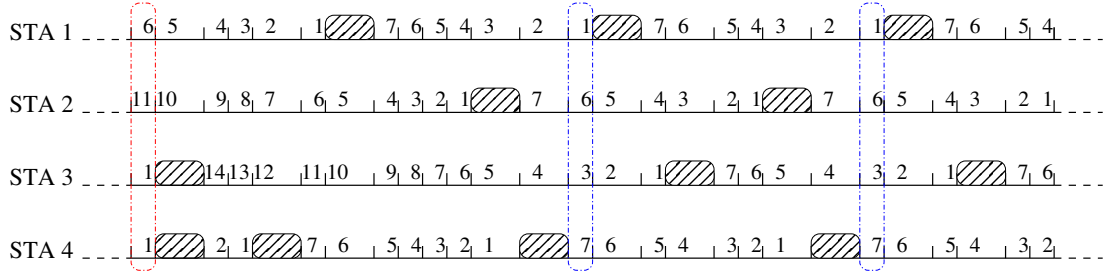
Fig. 1. CSMA/ECA example in saturation

Algorithm 1:

1 **while** *the device is on* **do**
2    $r \leftarrow 0; k \leftarrow 0$;
3    $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
4   **while** *there is a packet to transmit* **do**
5   **repeat**
6   **while** $B > 0$ **do**
7   wait 1 slot;
8          $B \leftarrow B - 1$;
9   Attempt transmission of 1 packet;
10   **if** *collision* **then**
11          $r \leftarrow r + 1$;
12          $k \leftarrow \min(k + 1, m)$;
13          $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
14   **until** *(r = R) or (success)*;
15      $r \leftarrow 0$;
16      $k \leftarrow 0$;
17   **if** *success* **then**
18        $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
19   **else**
20   Discard packet;
21        $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
22   Wait until there is a packet to transmit;

Algorithm 1: CSMA/CA. r indicates the number of retransmission attempts, while R is the maximum retransmission attempts limit; when it is reached, the packet waiting for transmission is dropped.

1 **while** *the device is on* **do**
2    $r \leftarrow 0 ; k \leftarrow 0$;
3    $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
4   **while** *there is a packet to transmit* **do**
5   **repeat**
6   **while** $B > 0$ **do**
7   wait 1 slot;
8          $B \leftarrow B - 1$;
9   Attempt transmission of 1 packet;
10   **if** *collision* **then**
11          $r \leftarrow r + 1$;
12          $k \leftarrow \min(k + 1, m)$;
13          $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
14   **until** *(r = R) or (success)*;
15      $r \leftarrow 0$;
16      $k \leftarrow 0$;
17   **if** *success* **then**
18        $B_d \leftarrow (2^k \mathrm{CW}_{\min})/2 - 1$;
19        $B \leftarrow B_d$;
20   **else**
21   Discard packet;
22        $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
23   Wait until there is a packet to transmit;

Algorithm 2: CSMA/ECA.

STA-4 will eventually collide because they have selected the same B. After recomputing the random backoff, STA-4's attempt results in a successful transmission, which instructs the node to pick a deterministic backoff, $B_d = 7$ in this case. By doing so, all successful STAs will not collide among each other in future cycles.

Collision slots being orders of magnitude larger than empty slots degrade the network performance. When CSMA/ECA builds the collision-free schedule all contenders are able to successfully transmit more often, increasing the aggregated throughput beyond DCF's. Figure 2 shows the achieved throughput of CSMA/ECA and CSMA/CA, alongside the Jain's Fairness Index (JFI) [20].

Referring to Figure 2, CSMA/ECA is able to achieve an aggregated throughput that goes beyond CSMA/CA up until

the number of contenders (N) is greater than $B_d = 7$. Beyond this point, the network will have a mixed behavior relating to backoff mechanisms: some nodes will successfully transmit and pick a deterministic backoff while others will collide due to the lack of empty slots and return to a random backoff. As more contenders join the network, CSMA/ECA performance will approximate to CSMA/CA's.

The *JFI for CSMA/ECA* and *JFI for CSMA/CA* curves in Figure 2 show the Jain's Fairness index for both protocols. Both protocols have a JFI equal to one, which suggests that the available throughput is shared evenly among all stations.

A. Supporting many more contenders

As was mentioned before, CSMA/ECA is only able to build a collision-free schedule if the number of contenders N, is less or equal than B_d. When $N > B_d$, collisions reappear.
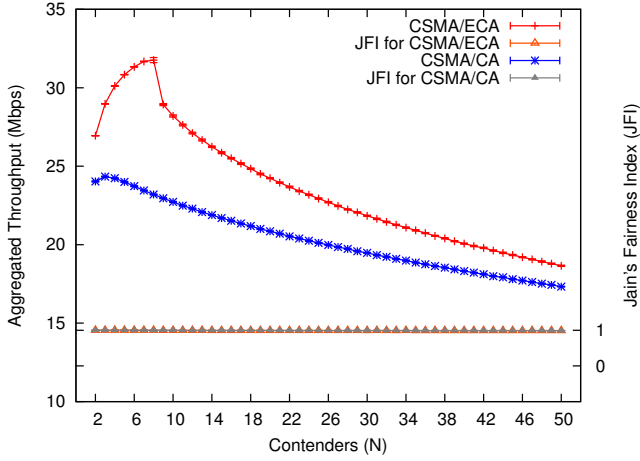
Fig. 2. CSMA/ECA example in saturation

Fig. 3. Fairness comparison with nodes under saturation

To be able to attain a collision-free schedule even when the number of contenders exceeds B_d, we introduce *Hysteresis*. Hysteresis is a property of the protocol that instructs nodes not to reset their backoff stage (k) after successful transmissions, but to pick a deterministic backoff $B_d = CW(k)/2$; where $CW(k) = 2^k CW_{\min}$. This measure allows the adaptation of the schedule length, admitting many more contenders in a collision-free schedule. This idea of a schedule is significantly different from the virtual schedule required by the protocols described in Section II, that is, CSMA/ECA with Hysteresis does not require a previous knowledge of neither the number of contenders nor the result of previous transmissions, easing its implementation in real hardware.

Hysteresis enables CSMA/ECA nodes to have different schedules (B_d), carrying the undesired effect of unevenly dividing the channel time among contenders (i.e., some nodes will have to wait more in order to attempt transmissions).

This unfairness issue is solved by instructing nodes at backoff stage k to transmit 2^k packets on each attempt, thus proportionally compensating those nodes at higher backoff stages. This additional extension to CSMA/ECA is called *Fair Share*. CSMA/ECA with Hysteresis and Fair Share will be referred to as CSMA/ECA$_{Hys+FS}$ in order to distinguish it from what was described until this point.

The idea of allowing the transmission of more packets to stations that transmit less often was initially proposed by Fang et al. in [19]. It was later adapted to CSMA/ECA$_{Hys}$ and named Fair Share in [15]. Figure 3 shows the JFI for CSMA/CA as well as for CSMA/ECA$_{Hys+FS}$.

In Figure 3, the only curve deviating from JFI = 1 is *CSMA/ECA with Hysteresis*; suggesting an uneven partition of the channel access time among contenders (which is fixed with Fair Share).

Algorithm 3 shows an implementation of CSMA/ECA$_{Hys+FS}$, while an example of CSMA/ECA$_{Hys+FS}$ with four contenders is shown in Figure 4. In the figure the first outline indicates a collision between STA-3 and STA-4, which will provoke an increment on both station's backoff stage ($k' = k + 1$). Once STA-4's random backoff expires, CSMA/ECA$_{Hys+FS}$ instructs the station to transmit $2^{k'}$ packets,
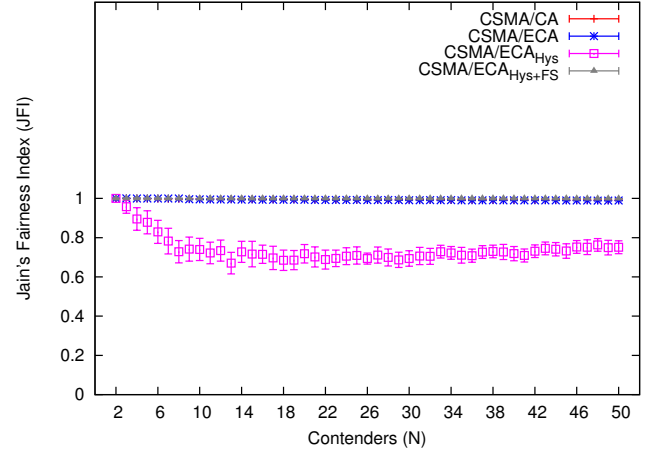
Algorithm 3: CSMA/ECA$_{Hys+FS}$

1 **while** *the device is on* **do**
2    $r \leftarrow 0$; $k \leftarrow 0$;
3    $b \leftarrow \mathcal{U}[0, 2^k CW_{\min} - 1]$;
4   **while** *there is a packet to transmit* **do**
5   **repeat**
6   **while** $B > 0$ **do**
7   wait 1 slot;
8          $B \leftarrow B - 1$;
9   Attempt transmission of 2^k packets;
10   **if** *collision* **then**
11          $r \leftarrow r + 1$;
12          $k \leftarrow \min(k + 1, m)$;
13          $B \leftarrow \mathcal{U}[0, 2^k CW_{\min} - 1]$;
14   **until** *(r = R) or (success)*;
15      $r \leftarrow 0$;
16   **if** *success* **then**
17        $B_d \leftarrow (2^k CW_{\min})/2 - 1$;
18        $B \leftarrow B_d$;
19   **else**
20   Discard packet;
21        $B \leftarrow \mathcal{U}[0, 2^k CW_{\min} - 1]$;
22   Wait until there is a packet to transmit;

and then pick a deterministic backoff, $B_d = CW(k')/2$. The same behavior is followed by STA 3.

With Hysteresis and Fair Share, CSMA/ECA$_{Hys+FS}$ is able to achieve greater throughput than CSMA/CA and for many more contenders, as shown in Figure 5 extracted from [15]. In the figure, the *CSMA/ECA with Hysteresis and Fair Share* curve shows a greater throughput because collisions are eliminated and Fair Share allows nodes to send 2^k packets upon each transmission.

B. CSMA/ECA$_{Hys+FS}$ vs. Maximum aggregation in CSMA/CA

Fair Share is an aggregation mechanism that coupled with the collision-free schedule built by CSMA/ECA$_{Hys}$ is able to

Fig. 4. CSMA/ECA$_{\text{Hys+FS}}$ example in saturation ($CW_{\min} = 16$)

Fig. 5. Throughput comparison [15]

Fig. 6. Throughput comparison with CSMA/CA$_{\text{MaxAg}}$: even through at low number of contenders CSMA/CA$_{\text{MaxAg}}$ achieves greater throughput, collisions eventually degrade the throughput below CSMA/ECA$_{\text{Hys+FS}}$'s when the number of contenders increases past $N = 10$

provide greater throughput than CSMA/CA. Figure 6 shows the achieved aggregated throughput for CSMA/ECA$_{\text{Hys+FS}}$ and CSMA/CA with nodes sending 2^m packets in each attempt. This CSMA/CA behavior will be called CSMA/CA$_{\text{MaxAg}}$ from here forth.

Although CSMA/CA$_{\text{MaxAg}}$ performs maximum aggregation, collisions degrade the aggregated throughput as more contenders attempt transmission. On the other hand, CSMA/ECA$_{\text{Hys+FS}}$ is able to build a collision-free schedule

and takes advantage of the aggregation provided by Fair Share.

C. Clock drift issue in descentralized collision-free MAC protocols

CSMA/ECA relies on stations being able to correctly count empty slots and consequently attempt transmissions in the appropriate slot according to the backoff timer. Failure to do so may be caused by clock imperfections inside the Wireless Network Interface Cards (WNIC), which is commonly referred to as *clock drift*. As pointed out in [21], clock drift is a common issue that degrades the throughput in distributed collision-free MAC protocols like the ones reviewed in Sect. II.

While miscounting empty slots have no significant effect on CSMA/CA throughput [21], it has a direct impact on CSMA/ECA. In a collision-free schedule with saturated CSMA/ECA contenders, a station miscounting empty slots will *drift* to a possibly busy slot, collide and force a reconvergence (if possible) to a collision-free schedule (see Sect. V-A2).

D. Backwards compatibilty and coexistence

CSMA/ECA$_{\text{Hys+FS}}$ springs from a modification to CSMA/CA's backoff mechanism. It keeps the range of values CSMA/CA nodes use to draw a random backoff (i.e., use the same $CW_{\min}$ and $CW_{\max}$), allowing CSMA/ECA$_{\text{Hys+FS}}$ contenders to coexist with CSMA/CA nodes in the same network. Further, the selection of CSMA/ECA$_{\text{Hys+FS}}$'s deterministic backoff, B_d, is the expected value for the current backoff stage k ($B_d := \lceil E[0, CW(k)] \rceil$); which ensures fairness among contenders. An overview of the attained throughput for different proportions of CSMA/ECA$_{\text{Hys+FS}}$ and CSMA/CA nodes is presented in Section V-C.

IV. SIMULATION

This section provides the simulation parameters for testing CSMA/ECA$_{\text{Hys+FS}}$ under two different traffic conditions, namely saturated and non-saturated. Further, the simulation of the clock drift effect, and the coexistence with CSMA/CA are also subjects to be addressed in this section.

A. Scenario details

Results are obtained by running multiple simulations over a modified version of the COST [22] simulator, available

TABLE I
PHY AND MAC PARAMETERS FOR THE SIMULATIONS

PHY	
Parameter	Value
PHY rate	65 Mbps
Empty slot	16 μs
DIFS	34 μs
SIFS	9 μs
MAC	
Parameter	Value
Maximum backoff stage (m)	5
Minium Contention Window ($CW_{\min}$)	16
Maximum retransmission attempts	6
Packet size (Bytes)	1024
MAC queue size (Packets)	1000

Fig. 7. Throughput under saturated conditions

at [23]. PHY and MAC parameters are detailed in Table I. The following assumptions were made:

- Unspecified parameters follow the IEEE 802.11n standard.
- All nodes are in communication range.
- There are no external interferences or channel errors.
- Collisions take as much channel time as successful transmissions ($\sigma_s = \sigma_c$).

If not mentioned otherwise, results are derived from 100 simulations of 100 seconds in length. Figures show 95% confidence intervals.

B. Saturated and Non-saturated stations

A saturated station always has packets in its MAC queue. This is modeled by setting the packet arrival rate to the MAC queue (Δ_{PAR}) to a value greater than the achievable throughput. To ensure saturation, stations are set to fill their MAC queue at $\Delta_{PAR} = 65$ Mbps, which is purposefully greater than the capacity of the channel.

To evaluate the performance under non-saturated conditions, stations need to be able to empty their MAC queues. To do so, the packet arrival rate to the MAC queue is set to $\Delta_{PAR} = 1$ Mbps. These values of ΔPAR have proven to produce the desired effects.

C. Performance under clock drift

Clock drift is simulated by setting a drift probability, p_{cd}. Each station has a probability of $p_{cd}/2$ of miscounting one slot more, and $p_{cd}/2$ of miscounting one slot less. This approach follows the one proposed by Gong et. al in [21].

D. Coexistance with CSMA/CA

To test the performance of CSMA/CA and CSMA/ECA$_{Hys+FS}$ stations in the same network, simulations are set with a CSMA/CA node density of 1/4, 1/2 and 3/4 of the total.

V. RESULTS

A. Saturated nodes

In CSMA/CA, saturated nodes coupled with a big number of contenders will normally be related to an increase in the collision probability. This effect is in part the result of resetting the backoff stage after a successful transmission and the generation of a random backoff. On the other hand, this scenario provides an advantageous condition to CSMA/ECA$_{Hys+FS}$ nodes. In saturation, CSMA/ECA$_{Hys+FS}$ nodes build a collision-free schedule and stick to their deterministic backoff as long as they transmit successfully, effectively eliminating collisions.

This section aims at overviewing the throughput of CSMA/CA and CSMA/ECA$_{Hys+FS}$ in saturation, as well as the collision probability, the average delay and the effect of clock drift over the throughput.

1) Throughput: CSMA/ECA$_{Hys+FS}$ nodes are able to build a collision-free schedule, thus the increase in the throughput seen in Figure 7. As mentioned in Section III-A, Hysteresis allows the allocation of more contenders in a collision-free schedule, while Fair Share ensures an even distribution of the available throughput. Whereas CSMA/CA throughput keeps decreasing due to an augmented number of collisions as the number of nodes increases (see Figure 8). Further, Figure 9 shows the fraction of collision slots for CSMA/ECA$_{Hys+FS}$ and CSMA/CA. In the figure, is appreciated how the fraction of collision slots keeps decreasing once CSMA/ECA$_{Hys+FS}$ reaches the collision-free schedule.

2) Effect of clock drift over the achieved throughput in saturation: Figure 10 shows the network aggregated throughput with 16 saturated stations and an increasing clock drift probability.

In Figure 10, a station has a clock drift probability equal to p_{cd}. Each station has a probability of $p_{cd}/2$ of miscounting one slot more, and $p_{cd}/2$ of miscounting one slot less. Because CSMA/CA is based on a random backoff, miscounting slots has no significant effect on the throughput. For the CSMA/ECA curve, it is possible to appreciate a slight decrease of the throughput as p_{cd} increases, caused by collisions due to the drift.

The CSMA/ECA$_{Hys+FS}$ curve in Figure 10 shows instead an increase of the aggregated throughput as p_{cd} grows. The increase in collisions make CSMA/ECA$_{Hys+FS}$ contenders to increment their backoff stage ($k' = k + 1$) and aggregate packets for transmissions according to Fair Share (sending $2^{k'}$
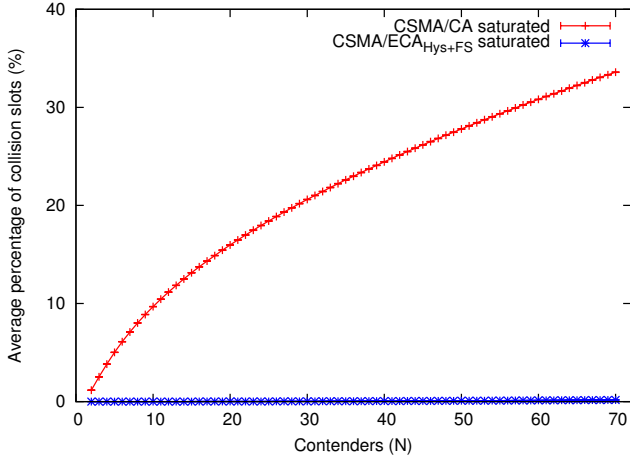
Fig. 8. Average percentage of collision slots: the fraction of time slots containing collisions.
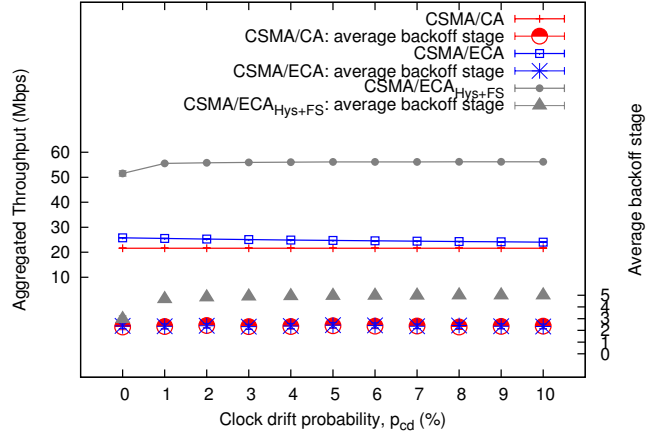

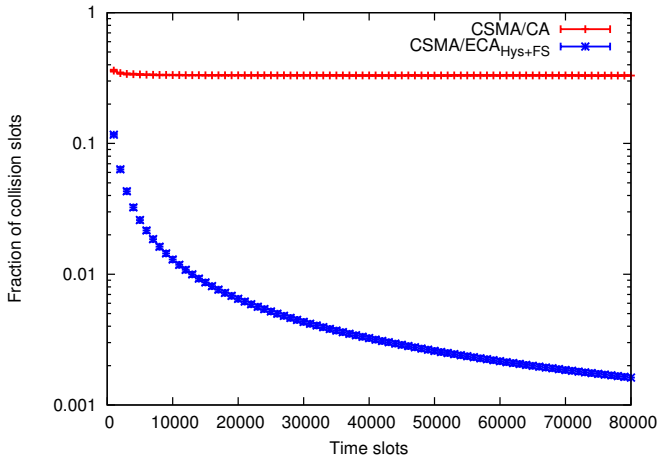Fig. 10. Throughput when increasing the clock drift probability

Fig. 9. Evolution of the fraction of collision slots in a scenario with 70 saturated stations.
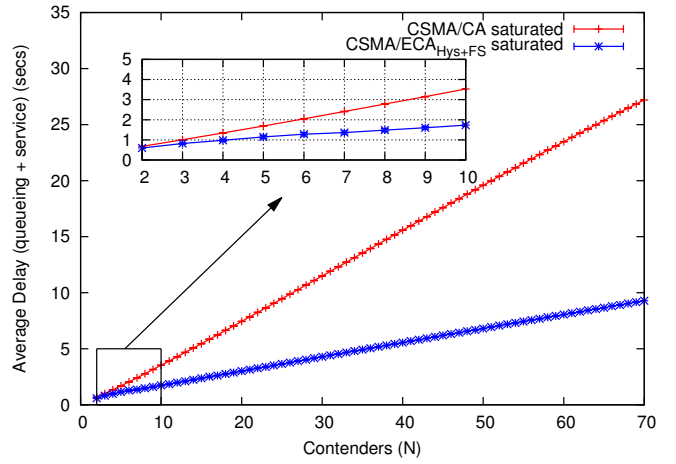

Fig. 11. Average system delay: aggregating the average delay of all stations

packets in each attempt), effectively increasing the throughput.

As it also can be appreciated in the figure, the average backoff stage for CSMA/ECA$_{Hys+FS}$ contenders increases rapidly to its maximum value ($m = 5$), reducing the effect of clock drift over CSMA/ECA$_{Hys+FS}$ nodes given that their transmissions would now be separated by more slots.

3) Delay: It is related to the elapsed time between the arrival of a packet to the MAC queue up until an acknowledgment is received for such packet(s).

In Figure 11, CSMA/ECA$_{Hys+FS}$ shows a lower delay due to the aggregation performed via Hysteresis and Fair Share, mainly because these aggregated transmissions are acknowledged by a single ACK. It is important to highlight that because nodes are in saturation, the arrival time to the MAC queue is considered the same for all the aggregated packets.

B. Non-saturated nodes

Emptying the MAC queue in CSMA/ECA$_{Hys+FS}$ means that nodes will reset their backoff stage and pick a random backoff when a new packet arrives at the queue; breaking

the collision-free schedule (if any) for CSMA/ECA$_{Hys+FS}$ contenders. The followings show the impact over throughput and delay when using CSMA/CA and CSMA/ECA$_{Hys+FS}$ in non-saturated conditions.

1) Throughput: In Figure 12, the aggregated throughput increases linearly for the *CSMA/CA non-saturated* curve until saturation is reached at around 22 nodes, where the throughput begins to degrade. The *CSMA/ECA$_{Hys+FS}$ non-saturated* curve has a similar behavior, entering saturation at around 60 nodes. Further, at around 30 nodes we see an increase in the average backoff stage for CSMA/ECA$_{Hys+FS}$ contenders which suggests an increment in dropped packets due to retransmission attempts. This effect is shown in Figure 13, where at around 30 nodes CSMA/ECA$_{Hys+FS}$ contenders start colliding and dropping packets.

After 40 contenders, the MAC queue of CSMA/ECA$_{Hys+FS}$ nodes starts to fill, as appreciated in Figure 14; gradually building a collision-free schedule due to CSMA/ECA$_{Hys+FS}$'s deterministic backoff after successful transmissions.

2) Delay: In Figure 15, a rapid increase in the delay for CSMA/CA nodes is appreciated at the saturation point (around 20 contenders). For the CSMA/ECA$_{Hys+FS}$ nodes the
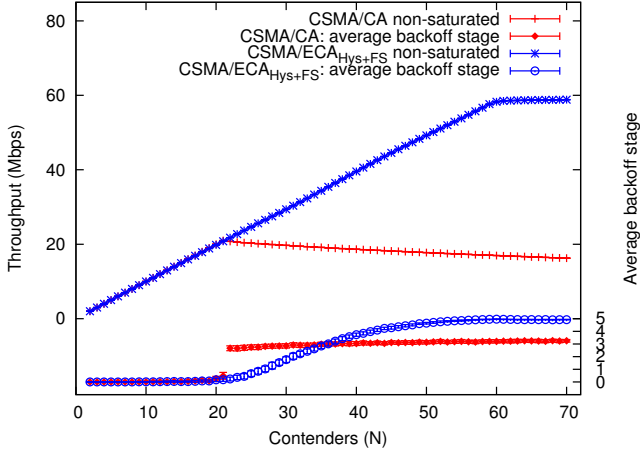
Fig. 12.   Throughput in non-saturation conditions
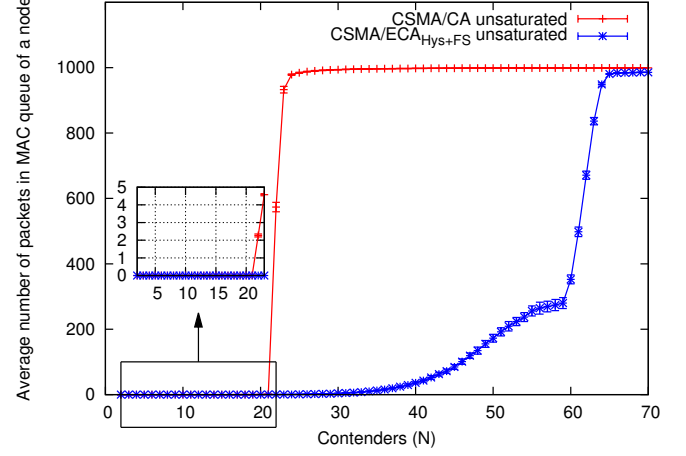

Fig. 14.   Number of packets in the MAC queue of a node
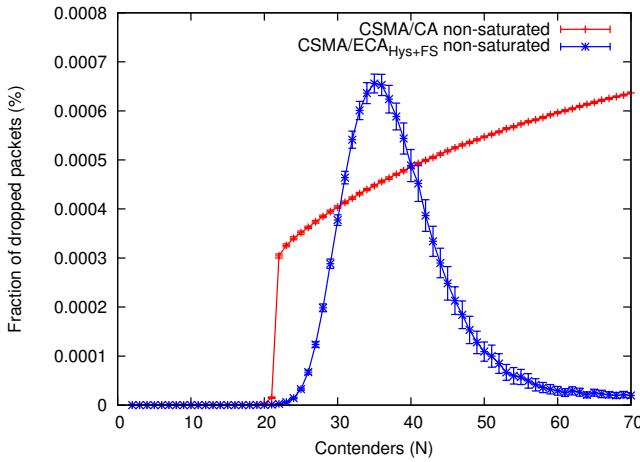

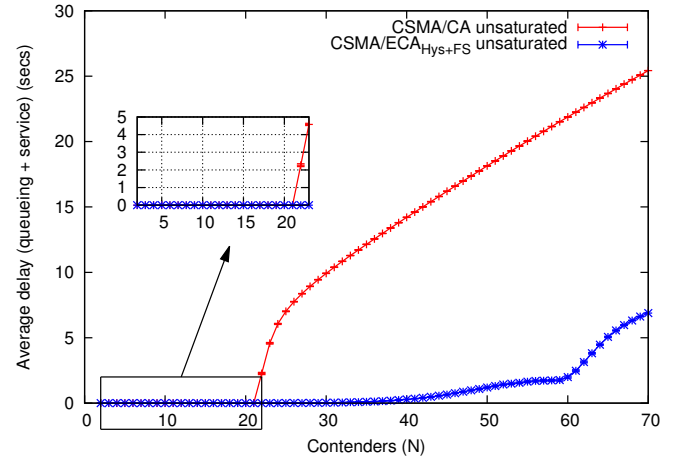Fig. 13.   Number of droppped packets due to reaching the retransmissions threshold

Fig. 15.   Average system delay: aggregating the average delay of all stations

delay is still low at CSMA/CA saturation point even-though we see an increase in the number of dropped packets (see Figure 13). This is due to Hysteresis and Fair Share, which reduces the average delay given that more packets are transmitted and acknowledged with a single ACK. As CSMA/ECA$_{Hys+FS}$ nodes get saturated, the delay increases due to longer queueing and contention time (see the number of packets in the MAC queue for CSMA/ECA$_{Hys+FS}$ nodes in Figure 14 and how it is related to the increase in delay shown in Figure 15).

C. Coexistence with CSMA/CA

CSMA/ECA is thought to be an evolution of CSMA/CA given its similarities and the ability to coexists with the latter. This section provides simulations results for a setup of different proportions of CSMA/CA nodes in a network where there are also CSMA/ECA$_{Hys+FS}$ contenders, that is: 1/4, 1/2 and 3/4 of the total nodes run CSMA/CA, while the rest uses CSMA/ECA$_{Hys+FS}$. This network configuration will be referred to as *mixed network setup* from here on.

1) Throughput: Figure 16 shows the network throughput for different proportions of CSMA/CA nodes in a mixed network setup.

In the figure it is appreciated how the mixed network setups curves lay between the CSMA/CA and CSMA/ECA$_{Hys+FS}$ curves. As the proportion of CSMA/CA nodes decreases, the throughput increases as the result of a lower probability of collision, as can be seen in Figure 17.

As shown in Figure 16, at a lower proportion of CSMA/CA nodes (1/4) the average aggregated throughput is the greatest among the tested mixed network setups. This is because collisions trigger Hysteresis and Fair Share in CSMA/ECA$_{Hys+FS}$ nodes, lowering the number of times these nodes enter in a contention and reducing the overall collision probability when compared to an only CSMA/CA network (see Figure 17). As the proportion of CSMA/CA nodes increases, the network throughput approximates to CSMA/CA.

VI. REAL HARDWARE IMPLEMENTATION AND FUTURE DIRECTIONS

The implementation of CSMA/ECA$_{Hys+FS}$ should be carried out at a firmware level due to the tight timing constrains
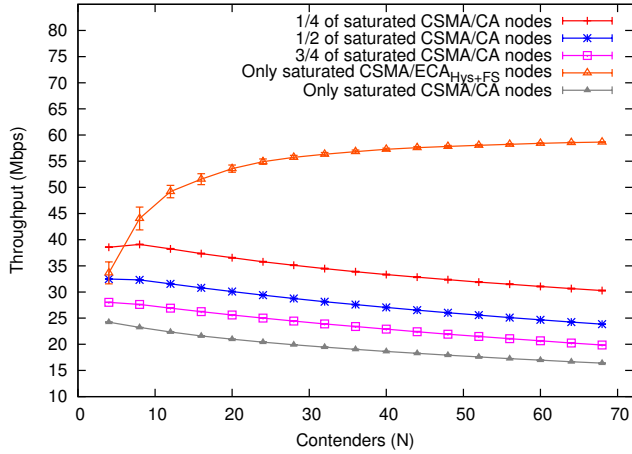
Fig. 16. Network throughput when composed by various proportions of CSMA/CA and CSMA/ECA$_{Hys+FS}$ nodes
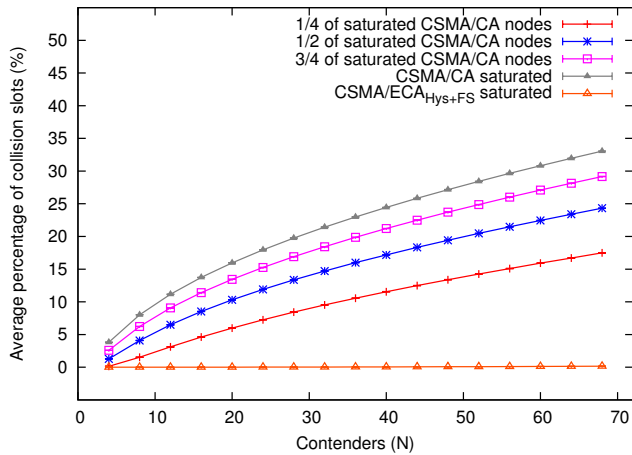

Fig. 17. Average percentage of collision slots for the tested mixed network setups proportions

related to the backoff procedure. There are several alternatives to modify the backoff operation in WNICs, either using open firmware like OpenFWWF [24] or through Field Programmable Gate Arrays (FPGA).

Although the FPGA option would in theory provide the strict timing requirements, the associated costs and difficulty are greater than the OpenFWWF alternative. OpenFWWF provides an open CSMA/CA firmware for a specific model of Broadcom WNICs' chipset, so the resulting firmware can be uploaded and tested in real commercial hardware.

By making the modifications represented in Algorithm 2, a proof-of-concept prototype of CSMA/ECA was built using comercial hardware [25]. The prototype was able to create a collision-free schedule among a reduced number of contenders using a big deterministic backoff. Nevertheless, clock drift, imprecise timing [26] and other prototyping non-idialities prevented the construction of a collision-free schedule when using a small deterministic backoff or dynamic rate adaptation algorithms, like Minstrel [27].

Further development on the prototype is being carried out in order to better prevent timing problems and incorporate Hysteresis and Fair Share.

VII. Conclusions

VIII. Acknowledgements

References

[1] "IEEE Standard for Information Technology - Telecommunications and Information exchange between systems. Local and Metropolitan Area Networks - Specific requirements," *IEEE Std 802.11TM-2012*, p. 1646, 2012.

[2] Perahia, E, "IEEE 802.11n Development: History, Process, and Technology," *Communications Magazine, IEEE*, vol. 46, no. 7, pp. 48–55, 2008.

[3] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: a media access protocol for wireless LAN's," in *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4. ACM, 1994, pp. 212–225.

[4] C. Wang, B. Li, and L. Li, "A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 4, pp. 1235–1246, 2004.

[5] F. Cali, M. Conti, E. Gregori, and P. Aleph, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," vol. 8, no. 6, pp. 785–799, 2000.

[6] A. Lopez-Toledo, T. Vercauteren, and X. Wang, "Adaptive Optimization of IEEE 802.11 DCF Based on Bayesian Estimation of the Number of Competing Terminals," vol. 5, no. 9, p. 1283, 2006.

[7] J. Barcelo, B. Bellalta, C. Cano, and M. Oliver, "Learning-BEB: Avoiding Collisions in WLAN," in *Eunice*, 2008.

[8] J. Barcelo, B. Bellalta, A. Sfairopoulou, C. Cano, and M. Oliver, "CSMA with Enhanced Collision Avoidance: a Performance Assessment," in *IEEE VTC Spring*, 2009.

[9] Y. He, R. Yuan, J. Sun, and W. Gong, "Semi-Random Backoff: Towards resource reservation for channel access in wireless LANs," in *17th IEEE International Conference on Network Protocols*. IEEE, 2009, pp. 21–30.

[10] J. Barcelo, A. Toledo, C. Cano, and M. Oliver, "Fairness and Convergence of CSMA with Enhanced Collision Avoidance (ECA)," in *2010 IEEE International Conference on Communications (ICC)*, may 2010, pp. 1 –6.

[11] M. Fang, D. Malone, K. Duffy, and D. Leith, "Decentralised learning MACs for collision-free access in WLANs," *Wireless Networks*, vol. 19, pp. 83–98, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11276-012-0452-1

[12] K. Hui, T. Li, D. Guo, and R. Berry, "Exploiting peer-to-peer state exchange for distributed medium access control," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 2368–2372.

[13] J. Barcelo, B. Bellalta, C. Cano, A. Sfairopoulou, and M. Oliver, "Towards a Collision-Free WLAN: Dynamic Parameter Adjustment in CSMA/E2CA," in *EURASIP Journal on Wireless Communications and Networking*, 2011.

[14] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, "Wireless MAC processors: Programming MAC protocols on commodity Hardware," in *INFOCOM, 2012 Proceedings IEEE*, march 2012, pp. 1269 –1277.

[15] L. Sanabria-Russo, A. Faridi, B. Bellalta, J. Barceló, and M. Oliver, "Future Evolution of CSMA Protocols for the IEEE 802.11 Standard," in *2nd IEEE Workshop on Telecommunications Standards: From Research to Standards. Budapest, Hungary*, 2013.

[16] G. Martorell, F. Riera-Palou, G. Femenias, J. Barcelo, and B. Bellalta, "On the performance evaluation of CSMA/E2CA protocol with open loop ARF-based adaptive modulation and coding," *European Wireless. 18th European Wireless Conference*, pp. 1 –8, april 2012.

[17] L. Hanzo, H. Haas, S. Imre, D. O'Brien, M. Rupp, and L. Gyongyosi, "Wireless Myths, Realities, and Futures: From 3G/4G to Optical and Quantum Wireless," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1853–1888, 2012.

[18] J. Lee and J. C. Walrand, "Design and nalysis of an asynchronous zero collision MAC protocol," *arXiv preprint arXiv:0806.3542*, 2008.

[19] M. Fang, D. Malone, K. Duffy, and D. Leith, "Decentralised learning MACs for collision-free access in WLANs," *Wireless Networks*, pp. 1–16, 2010.

[20] R. Jain, D. Chiu, and W. Hawe, *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System*. Eastern Research Laboratory, Digital Equipment Corporation, 1984.

[21] W. Gong and D. Malone, "Addressing Slot Drift in Decentralized Collision Free Access Schemes for WLANs," in *Multiple Access Communications*, ser. Lecture Notes in Computer Science, B. Bellalta, A. Vinel, M. Jonsson, J. Barcelo, R. Maslennikov, P. Chatzimisios, and D. Malone, Eds. Springer Berlin Heidelberg, 2012, vol. 7642, pp. 146–157. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-34976-8_16

[22] E. Yücesan, C. Chen, J. Snowdon, and J. Charnes, "COST: A component-oriented discrete event simulator," in *Winter Simulation Conference*, 2002.

[23] L. Sanabria-Russo, J. Barcelo, and B. Bellalta. (2012) Implementing CSMA/ECA in COST. [Online]. Available: https://github.com/SanabriaRusso/CSMA-E2CA

[24] F. Gringoli and L. Nava. Open Firmware for WiFi Networks. Webpage. [Online]. Available: http://www.ing.unibs.it/openfwwf/

[25] L. Sanabria-Russo, J. Barcelo, and B. Bellalta, "Prototyping Distributed Collision-Free MAC Protocols for WLANs in Real Hardware," in *Multiple Access Communcations*. Springer International Publishing, 2013, pp. 82–87.

[26] G. Bianchi, A. Di Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello, "Experimental Assessment of the Backoff Behavior of Commercial IEEE 802.11 b Network Cards," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE, 2007, pp. 1181–1189.

[27] MadWiFi project. Minstrel: a rate control algorithm for mac80211. Webpage. [Online]. Available: http://bit.ly/minstrelAlgorithm