Traffic Differentiation in Dense Collision-free WLANs

Luis Sanabria-Russo, Boris Bellalta
Department of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona, Spain
{luis.sanabria,boris.bellalta}@upf.edu

***Abstract*—The ability to perform traffic differentiation is a promising feature of the current Medium Access Control (MAC) in Wireless Local Area Networks (WLANs). The Enhaced Distributed Channel Access (EDCA) MAC protocol for WLANs proposes up to four Access Categories (AC) that can be mapped to different traffic priorities. High priority ACs are allowed to transmit more often than low priority AC, providing a way of prioritizing delay sensitive traffic like voice calls or video streaming. Further, EDCA also considers the intricacies related to the management of multiple queues, virtual collisions and priority. Nevertheless, EDCA falls short in efficiency when performing in dense WLAN scenarios. Its collision-prone contention mechanism degrades the overall throughput to the point of starving low priority ACs.**

Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) is a compatible MAC protocol for WLANs which is also capable of providing traffic differentiation. Contrary to EDCA, CSMA/ECA uses a contention mechanism with a deterministic backoff technique which is capable of constructing collision-free schedules for each AC, avoiding the throughput starvation seen in low priority EDCA ACs. This work analyses traffic differentiation with CSMA/ECA by describing the mechanisms used to construct collision-free schedules with multiple queues. Furthermore, it introduces a way to eliminate Virtual Collisions (VC), which also contribute to the throughput degradation in EDCA WLANs. Finally, it provides simulation results that show CSMA/ECA outperforming EDCA in different commonly-found networking scenarios.

***Index Terms*—Wireless LAN, Multiaccess Communication, Collision-Free, QoS, EDCA.**

I. Introduction

Wireless Local Area Networks (WLANs or WiFi networks [1]) are a popular solution for wireless connectivity. Ranging from computers to wearable devices, it has widespread adoption. Unlike Ethernet, the medium in WLANs is shared. Every user having a packet to transmit must join a contention for the channel, whose winners will gain access and attempt a transmission. The Distributed Coordination Function (DCF) is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)[1], it coordinates access to the wireless channel by deferring each contender's transmission for a fixed period of time when the channel goes idle and then for a random backoff period.

WiFi's increasing adoption pushed the need for mechanisms to prioritise traffic in order to ensure Quality of Service (QoS); i.e., to provide advantageous conditions for throughput or delay sensitive applications like video calls, streaming or gaming. The Enhanced Distributed Channel Access (EDCA) (specified in IEEE 802.11e protocol [2]), builds over DCF in order to provide this kind of traffic differentiation.

EDCA proposes up to four queues or Access Categories (ACs), each one working as an instance of the DCF. Essentially, at the Medium Access Control (MAC) layer EDCA allows the highest priority AC to access the channel more often than lower priority ACs.

Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) [3], [4] is capable of achieving greater throughput than CSMA/CA mostly due to its ability to make a more efficient use of the channel, particularly in networks with many contenders [3]. Instead of deferring each contender's transmission for a random backoff period (as in CSMA/CA), CSMA/ECA instructs contenders to use a deterministic backoff after successful transmissions. In doing so, contenders that successfully transmitted in the past will do so again without colliding with other successful contenders in future cycles. By constructing collision-free schedules, CSMA/ECA is able to avoid the throughput starvation of low priority ACs [5], hence yielding greater throughput than EDCA.

Analogous to EDCA, CSMA/ECA can be extrapolated to handle multiple queues, thus providing a way to implement EDCA-compatible collision-free traffic differentiation in WLANs.

Although several studies analyse the performance or extend CSMA/ECA to support many contenders [3], [4], [6]–[8], this work is the first to incorporate traffic differentiation into the protocol with all of its extensions. We provide the following contributions:

- First simulation results on throughput, collisions and delay for CSMA/ECA with four ACs.
- Introduction of the Smart Backoff mechanism for eliminating Virtual Collisions (VC).
- Coexistence and backwards compatibility with EDCA.

Results show that CSMA/ECA is able to provide collision-free traffic differentiation under saturated traffic conditions. Moreover, because it makes a more efficient use of the channel time, CSMA/ECA outperforms EDCA in all tested scenarios. Equally important, results show that CSMA/ECA is able to coexist with EDCA. In fact, the overall throughput of a mixed network is grater than the observed in a EDCA-only network.

[1] DCF and CSMA/CA will be used interchangeably throughput the rest of the text.

An overview of the traffic differentiation techniques in EDCA will be provided in Section II. Then, we will present CSMA/ECA and its ability to provide traffic differentiation in Section III. The performance evaluation is shown in Section IV, while we draw our conclusions in Section V.

II. Related Work

Each node with a packet to transmit must join a contention for the channel. In CSMA/CA, nodes are deferred for a fixed period of idle-channel time and then for a random backoff period before attempting transmission. Because it only considers a single kind of traffic, the default contention parameters are the same for all contenders.

In this section we present the traffic differentiation capabilities of EDCA as well as other enhancements proposed by the research community.

A. Enhanced Distributed Channel Access

EDCA provides traffic differentiation by defining three parameters for each of the four ACs. First, by adjusting the Transmission Opportunity (TXOP) an AC may transmit several packets without repeating the contention for the channel, thus achieving greater throughput than other ACs. The other two parameters are related to the contention process, namely the Contention Window (CW$_{\min}$ and CW$_{\max}$, for minimum and maximum respectively) and the Arbitration Inter-Frame Spacing (AIFS). The contention windows limit the random backoff period, while the AIFS defines the fixed waiting period when the channel is idle. ACs with low contention windows and short AIFS will access the channel quicker, that is, have higher priority.

WLANs time is slotted. That is, it is composed of tiny empty slots of fixed duration σ_e, collisions and successful slots (which contain collisions or a successful transmission. Their duration denoted by σ_c and σ_s, respectively)[2]. DCF instructs backlogged stations to wait for a fixed period of time and then for a random number of empty slots (random backoff period) before attempting transmission.

EDCA extends directly from DCF. In its place, EDCA declares up to four Access Categories (AC), each one an instance of DCF with different contention parameters that allow a statistical prioritisation among them [9]. Traffic types, declared by the IEEE 802.1D standard [10] are then mapped to the four ACs in EDCA (MAC bridging). The mapping is shown in Table I.

Every backlogged AC joins the contention for the channel by drawing a random backoff, $B[\text{AC}] \leftarrow \mathcal{U}[0, CW_{\min}[\text{AC}] - 1]$; where $CW_{\min}[\text{AC}]$ is the minimum contention window for said AC. Each AC waits for a fixed $AIFS[\text{AC}] = SIFS^3 + \sigma_e(AIFSN[\text{AC}])$ period of inactivity in the channel and then starts decrementing its random backoff. Each passing empty slot decrements $B[\text{AC}]$ in one. When the backoff counter expires ($B[\text{AC}] = 0$), the AC will attempt transmission. Table II shows the default CW, AIFSN and TXOP values specified for EDCA.

TABLE I: AC relative priorities and mapping from 802.1D user priorities (extracted from [9])

Priority	**802.1D User priority**	**AC**	**Designation**
Lowest	1	BK	Background
	2		
	0	BE	Best Effort
	3		
	4	VI	Video
	5		
Highest	6	VO	Voice
	7		

TABLE II: Deafult EDCA parameters (extracted from [9])

AC	**CW$_{\min}$**	**CW$_{\max}$**	**m**	**AIFSN**	**TXOP limit**
BK	32	1024	5	8	0 (only one MSDU)
BE	32	1024	5	4	0 (only one MSDU)
VI	16	32	1	3	3.008 ms
VO	8	16	1	3	1.504 ms
Legacy	16	1024	5	3	0 (only one MSDU)

As it can be observed in Table II, ACs BK and BE may only send one MAC Service Data Unit (MSDU) upon each attempt. Whereas VI and VO can allocate the channel for longer periods. The TXOP parameter offers resource fairness rather than throughput fairness, that is, all ACs of the same category will receive close to the same average channel time regardless of its data rate. Furthermore, because the CW and AIFSN values for VI and VO are smaller than the others's, on average these ACs will access the channel quicker; thus providing priority in the contention. The relative AIFSN lengths for each AC are shown in Fig. 1.

While being effective in providing traffic differentiation and priority, in principle EDCA is unable to eliminate collisions. For instance, two ACs from different contenders may draw the same random backoff and consequently attempt transmission in the same time slot, causing a collision. Collisions force a retransmission of the frame by repeating a contention after doubling the current CW. Furthermore, if two or more ACs within a node experience a backoff expiration at the same instant, a Virtual Collision (VC) will occur. VC are resolved by granting the channel to the highest priority AC, while doubling the CW for the lower priority ACs; just as it is done in the event of a real collision.

It follows directly from above that collisions waste channel time and thus contribute to the throughput degradation in WLANs. Moreover, the probability of collision increases as more contenders join the network, each one having four ACs attempting to gain access to the channel.

B. Other contributions

Because ACs in EDCA perform contention independently of the others, each AC mimics a station. This explains why the collision probability in EDCA is higher than in single-AC DCF networks with the same number of nodes. Furthermore, the contention parameters in EDCA work better in scenarios with low number of contenders, but often cause starvation of low priority ACs in crowded scenarios (see [5] and Section IV-B).

[2] Empty slots are much shorter than collision or successful slots, that is, $\sigma_e \ll \min(\sigma_c, \sigma_s)$.

[3] The Short Inter-Frame Space (SIFS) is defined in [1]. It is equal to 10 or 16μs for 802.11 n and ac respectively.

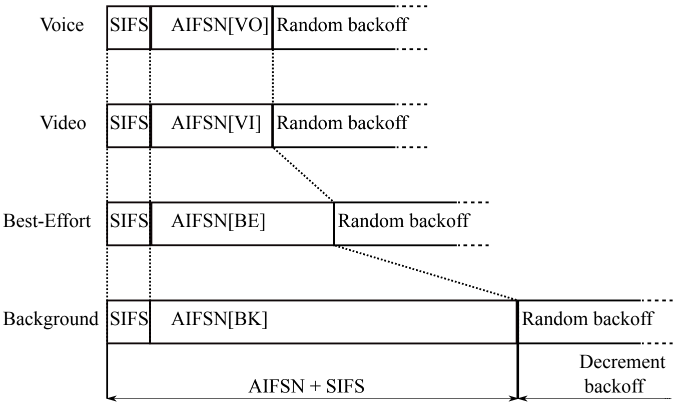


Fig. 1: AIFSN values for each AC. From lowest (BK) to highest priority (VO).

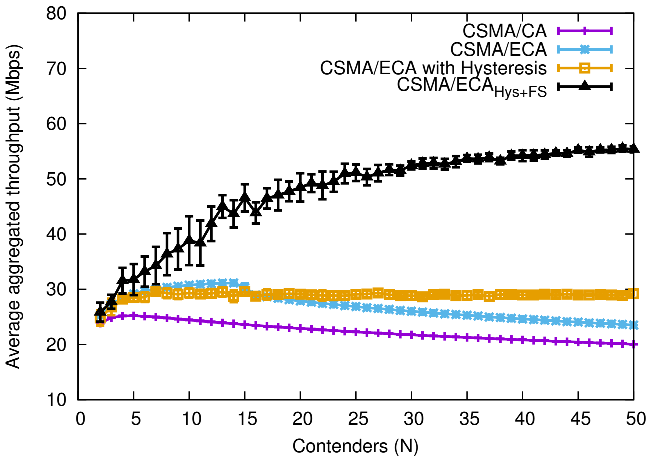


Fig. 3: Average aggregated throughput ($CW_{\min} = 32$) [3]

Therefore, great efforts have been directed towards parameter adjustments in EDCA [11]–[13]. For example, by dynamically adjusting the AIFS for each AC it is possible to maintain traffic differentiation while avoiding the starvation of low priority ACs. This is especially relevant in WLANs where medical-grade QoS is desired [12]. Further, by randomising the AIFS values it is possible to increase the channel utilisation in EDCA [13]. Nevertheless, all these approaches either require information about the number of nodes participating in the contention, or inject additional traffic to the network, which may be unsuitable in crowded scenarios.

III. Traffic Differentiation with CSMA/ECA

Carrier Sense Multiple Access with Enhanced Collision Avoidance [3], [4] is able to build collision-free schedules by using a deterministic backoff after successful transmissions. That is, if all saturated contenders are able to perform a successful transmission and then pick a deterministic backoff, they will not collide among each other in future transmissions.

To attempt a transmission, stations generate a random backoff $B \leftarrow \mathcal{U}[0, CW_{\min} - 1]$, just as in DCF. Every passing empty slot decrements B in one. When $B = 0$, the station will attempt a transmission. If the transmission fails, the node will increment its backoff stage $k \in [0, m]$ in one (where m is the maximum backoff stage of typical value $m = 5$) and use another random backoff $B \leftarrow \mathcal{U}[0, CW(k) - 1]$; where $CW(k) = 2^k CW_{\min}$ is the contention window at backoff stage k. Otherwise, the successful station will then pick a deterministic backoff for its next transmission, $B_{\text{d}} \leftarrow \lceil CW_{\min}/2 \rceil - 1$. This value of B_{d} is roughly equal to the expectation of a random backoff at the same backoff stage ($k = 0$, in this case), making it fair and compatible with CSMA/CA stations [4].

CSMA/ECA is also capable of allocating many contenders in a collision-free schedule by not reseting the backoff stage k after a successful transmission, as opposed to CSMA/CA. That is, a node at backoff stage k would select $B_{\text{d}} \leftarrow \lceil CW(k)/2 \rceil - 1$ as its deterministic backoff after a successful transmission. This extension to CSMA/ECA is called Hysteresis.

Hysteresis forces some contenders to have larger schedules than others, resulting in an unfair distribution of the network resources. This effect can be compensated by allowing nodes at backoff stage k to transmit 2^k packets upon each transmission attempt. We call this extension Fair Share and it ensures an even distribution of the available throughput among contenders.

Figure 2 shows a temporal example of CSMA/ECA with its extensions, namely, Hysteresis and Fair Share [3]. The *STA-#* labels denote the saturated contenders. The horizontal lines represent time, where the numbers are the number of empty slots left until the expiration of the backoff counter. In the figure, the first outline indicates a collision which forces STA-3 and STA-4 to increase their backoff stages ($k \leftarrow \min(k + 1, m)$) and use a random backoff, $B \leftarrow \mathcal{U}[0, CW(k) - 1]$. When STA-4's backoff expires, it will then transmit 2^k packets. Because the transmission was successful it uses $B_{\text{d}} = \lceil CW(k)/2 \rceil - 1$ as its deterministic backoff for the next transmission. Notice that once all stations transmit successfully collisions are eliminated.

CSMA/ECA with Hysteresis and Fair Share (CSMA/ECA$_{\text{Hys+FS}}$) is able to outperform CSMA/CA, as shown in the simulation results presented in Figures 3 and 4. For the sake of clarity, Algorithm 1 and Algorithm 2 exemplify CSMA/CA and CSMA/ECA$_{\text{Hys+FS}}$ respectively.

The main difference between Algorithm 1 and Algorithm 2, and therefore between CSMA/CA and CSMA/ECA$_{\text{Hys+FS}}$ is that the latter uses a deterministic backoff after a successful transmission. This difference is highlighted at line 18 and line 17 of Algorithms 1 and 2, respectively.

A. Schedule Reset mechanism

CSMA/ECA$_{\text{Hys+FS}}$ instructs nodes not to reset their backoff stage after a successful transmission. This is done in order to increase the cycle length and provide a collision-free schedule for many contenders, which is desirable in dense scenarios.

Having a big deterministic backoff increases the time between successful transmissions. Furthermore, if not operating in a scenario with many nodes the empty slots between transmissions are not longer negligible and degrade the overall throughput of the system. For instance, if nodes withdraw

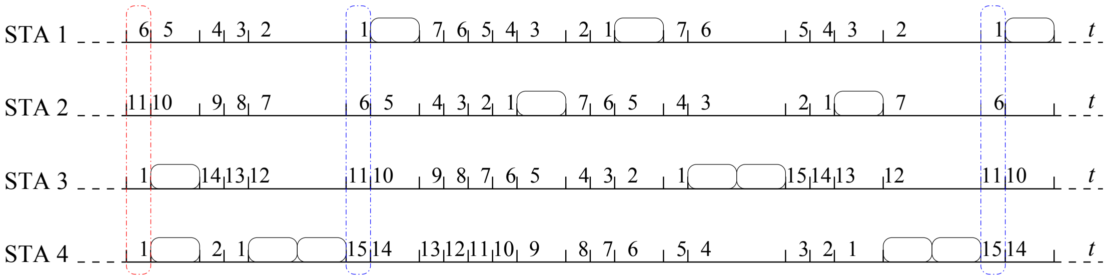

STA 1
6 5 4 3 2 1 7 6 5 4 3 2 1 7 6 5 4 3 2 1 t
STA 2
11 10 9 8 7 6 5 4 3 2 1 7 6 5 4 3 2 1 7 6 t
STA 3
1 14 13 12 11 10 9 8 7 6 5 4 3 2 1 15 14 13 12 11 10 t
STA 4
1 2 1 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 15 14 t

Fig. 2: An example of the temporal evolution of CSMA/ECA in saturation

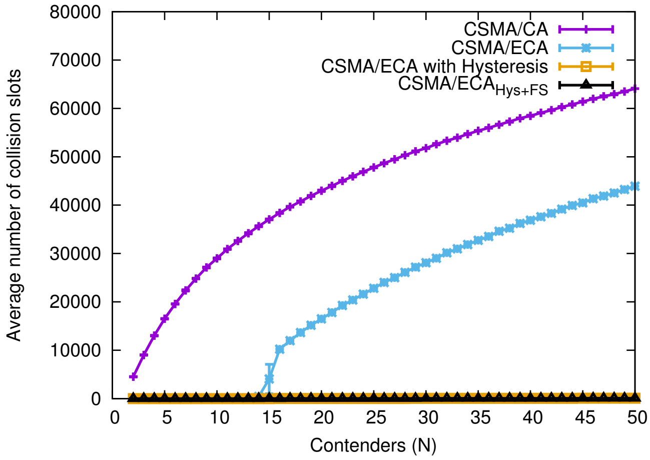


Fig. 4: Average aggregated number of collision slots ($CW_{\min} = 32$) [3]

```
while the device is on do
    $r \leftarrow 0$; $k \leftarrow 0$;
    $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
    while there is a packet to transmit do
        repeat
            while $B > 0$ do
                wait 1 slot;
                $B \leftarrow B - 1$;
            Attempt transmission of 1 packet;
            if collision then
                $r \leftarrow r + 1$;
                $k \leftarrow \min(k + 1, m)$;
                $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
        until $(r = R)$ or (success);
        $r \leftarrow 0$;
        $k \leftarrow 0$;
        if success then
            $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
        else
            Discard packet;
            $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
    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.

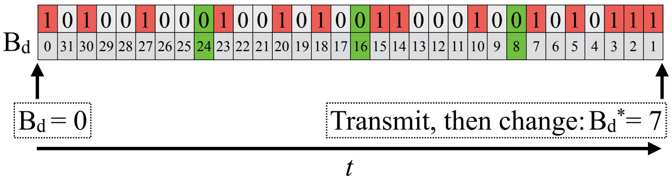


Fig. 5: Example of the Schedule Reset mechanism (extracted from [3])

from the contention their previously used slots will now be empty. Contenders should be aware of this issue and pursue opportunities to reduce their deterministic backoff without sacrificing too much in collisions.

The *Schedule Reset* mechanism for CSMA/ECA$_{\text{Hys+FS}}$ introduced in [3] consists on finding the smallest collision-free schedule between a contender's transmissions and then change the node's deterministic backoff to fit in that schedule. Take a contender with a $B_{\mathrm{d}} = 31$ as an example. By listening to the slots between its transmissions, we are able to determine the availability of smaller (and possibly) collision-free schedules.

Figure 5 shows the slots between the transmissions of a contender with $B_{\mathrm{d}} = 31$. Starting from the left, the current $B_{\mathrm{d}} = 0$ means that the slot will be filled with the contender's own transmission. Each following slot containing either a transmission or a collision is identified with the number one, while empty slots are marked with a zero. Notice that the highlighted empty slots appear every eight slots, suggesting that a schedule reduction from $B_{\mathrm{d}} = 31$ to $B_{\mathrm{d}}^* = 7$ is possible. The schedule change is performed after the contender's next successful transmission.

Schedule Reset (SR) is implemented in CSMA/ECA$_{\text{Hys+FS}}$ by filling a bitmap b of size $B_{\mathrm{d}}{+}1$. Each bit t, $t \in \{0, \ldots, B_{\mathrm{d}}\}$ in the bitmap is the result of a bitwise OR operation between its current value, $b[t]$ and the state of the slot; which equals to one when busy or zero when idle. After γ consecutive successful transmissions (sxTx), the bitmap is evaluated. If a change of schedule is possible, it is executed just after the next successful transmission.

It is possible to configure Schedule Reset in two modes, namely *conservative* and *aggressive*. These modes relate to the number of consecutive transmissions needed to evaluate the bitmap, that is, γ. A conservative SR contains the information

```
while the device is on do
    $r \leftarrow 0$ ; $k \leftarrow 0$ ; $k_c \leftarrow k$;
    $b \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
    while there is a packet to transmit do
        repeat
            while $B > 0$ do
                wait 1 slot;
                $B \leftarrow B - 1$;
            Attempt transmission of $2^k$ packets;
            if collision then
                $r \leftarrow r + 1$;
                $k \leftarrow \min(k + 1, m)$;
                $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
        until ($r = R$) or (success);
        $r \leftarrow 0$;
        if success then
            $B_d \leftarrow \lceil 2^k \mathrm{CW}_{\min}/2 \rceil - 1$;
            $B \leftarrow B_d$;
        else
            Discard $2^{k_c}$ packets;
            $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
        $k_c \leftarrow k$;
    Wait until there is a packet to transmit;
```

Algorithm 2: CSMA/ECA$_{\text{Hys+FS}}$: k_c refers to the contention backoff stage, that is, the backoff stage with which a contention for transmission is started. After R retransmission attempts, Fair Share instructs the node to drop 2^{k_c} packets

of all users' transmissions, therefore no additional collisions are introduced as a consequence of the schedule change. This implies a value of $\gamma = 2^{m-1} CW_{\min}/B_{\text{d}}$. On the other hand, setting $\gamma = 1$ triggers a bitmap evaluation after just two consecutive transmissions, rendering this choice of γ the aggressive mode.

Aggressive Schedule Reset coupled with an increase in the Stickiness has proven to be suitable for noisy scenarios [3]. Stickiness is not a new concept to CSMA/ECA [7]. It simply instructs the contender to *stick* to the deterministic backoff even in the event of *stickiness* number of failed transmissions. This allows for a faster convergence towards a collision-free schedule. CSMA/ECA$_{\text{QoS}}$ with a default level of stickiness equal to 2 has proven to provide the better combination of high throughput and low collisions, as shown in Figure 6.

B. *Incorporating multiple ACs into CSMA/ECA*

As shown before, CSMA/ECA and its extensions (referred to as CSMA/ECA$_{\text{Hys+FS}}$ from this point forward) are able to construct a collision-free schedule under saturated conditions, outperforming CSMA/CA. Furthermore, CSMA/ECA$_{\text{Hys+FS}}$ uses the same default contention parameters used by CSMA/CA, so the compatibility is maintained [3].

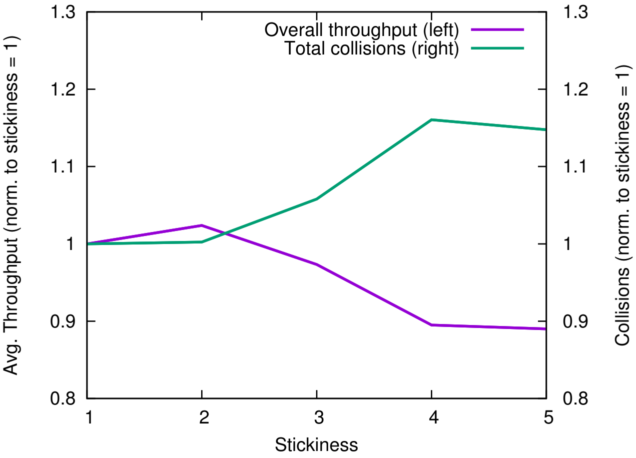


Fig. 6: Average throughput and collisions with different levels of stickiness for CSMA/ECA$_{\text{QoS}}$ in non-saturation conditions (as explained in Section IV-A)

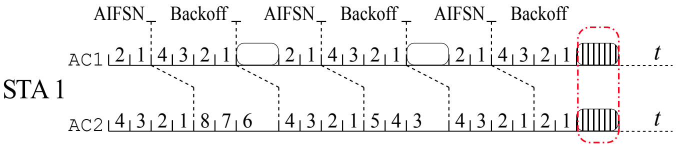


Fig. 7: Example temporal evolution of CSMA/ECA$_{\text{Hys+FS}}$ with two ACs using AIFS resulting in a virtual collision. Only considering AIFSN values of 2 and 4, B_{d} of 4 and 8 for AC1 and AC2 respectively

Providing priority is to ensure a more frequent access to some ACs over others. In CSMA/ECA$_{\text{Hys+FS}}$ this is only subject to the deterministic backoff. That is, an AC using a shorter B_{d} would in turn access the channel more often than those using a larger one. To maintain compatibility with EDCA, CSMA/ECA$_{\text{Hys+FS}}$ considers the same four ACs and CW values shown in Table II.

Nevertheless, AIFS and TXOP are not fit for multiple CSMA/ECA$_{\text{Hys+FS}}$ queues. For instance, AIFS values are not required since differentiation is only provided by the deterministic backoff. The incorporation of different AIFS for each category would trigger Virtual Collisions that in turn may disrupt an existent collision-free schedule. Figure 7 shows a VC in CSMA/ECA$_{\text{Hys+FS}}$ (indicated by the outline) consequence of using AIFS during a collision-free schedule.

TXOP in EDCA ensures that all traffic from the same category receives on average the same channel time rather than the same throughput. In contrast, CSMA/ECA$_{\text{Hys+FS}}$'s goal through Fair Share is to provide close to equal average throughput to same-priority ACs. We leave the elaboration of an adaptive Fair Share which ensures resource fairness rather than throughput fairness as a future research topic.

As EDCA extends DCF into four ACs, similarly there is an instance of CSMA/ECA$_{\text{Hys+FS}}$ for each AC. We will refer to CSMA/ECA$_{\text{Hys+FS}}$ with multiple ACs as CSMA/ECA$_{\text{QoS}}$ from here forward. Table III shows the CW as well as the lowest and largest B_{d} for ACs in CSMA/ECA$_{\text{QoS}}$.

Figure 8 shows an example of CSMA/ECA$_{\text{QoS}}$ with two contenders and two ACs; where AC1 has higher priority than AC2. In the figure, the first outline indicates a VC between AC1 and AC2 from STA-2. VC in CSMA/ECA$_{\text{QoS}}$ are handled just as in EDCA, that is, the AC with the highest priority is granted access to the channel, while the other ACs involved in the VC double their contention windows and use a random backoff for the next transmission. Consequently, AC1 from STA-2 successfully transmits and then uses $B_{\text{d}} = \frac{2^0 CW_{\min}[\text{AC1}]}{2} - 1 = 3$.

Still on Figure 8, the second outline indicates a collision between STA-2's AC2 and AC1 from STA-1. At this moment in time STA-2 AC2's backoff stage has been increased in two occasions ($k[\text{AC2}] = 2$). When said AC2 is able to transmit, it sends $2^{k[\text{AC2}]}$ packets according to Fair Share. Then, it uses a deterministic backoff, $B_{\text{d}} = \frac{2^{k[\text{AC2}]} CW_{\min}[\text{AC2}]}{2} - 1 = 31$.

The third outline in Figure 8 indicates an VC in STA-1, which is resolved allowing AC1 and deferring AC2's transmission using a random backoff with a doubled CW. A future collision between STA-2's AC1 and AC2 from STA-1 is highlighted by the last outline.

C. Collisions and Virtual Collisions-free operation using Smart Backoff

Consider a complete schedule of length $C = 2^m CW_{\min}$, and $m = 5$. With CSMA/ECA$_{\text{Hys+FS}}$ and a single AC it is possible to allocate a collision-free transmission slot for up to $C/2 = 512$ contenders (the highest $B_{\text{d}} + 1$ for AC Legacy in Table III). Nevertheless, with CSMA/ECA$_{\text{QoS}}$ and all ACs in saturation each contender mimics the behaviour of four saturated nodes (one for each AC). This means that the total number of supported collision-free contenders will be reduced in order to provide a transmission slot for all the ACs in the network. If all the ACs are in saturation, i.e., have a packet to transmit, CSMA/ECA$_{\text{QoS}}$ can provide collision-free operation for up to $2^{m[\text{VO}]} CW_{\min}[\text{VO}]/2 = 8$ contenders, where $m[\text{VO}]$ is the maximum backoff stage of the AC with the smallest $CW_{\max}$. Even-though Table III provides CSMA/ECA$_{\text{QoS}}$'s default contention parameters, these will be adjusted in Section IV-B in order to support more contenders while in saturation.

VCs in CSMA/ECA$_{\text{QoS}}$ force ACs to defer their transmissions using a random backoff. Therefore, VCs can disrupt any existent collision-free schedule in CSMA/ECA$_{\text{QoS}}$, wasting channel time recovering from collisions and degrading the overall throughput. Given that all AC's backoff counters are known to the contender, there is nothing preventing it from using this information to avoid future VCs.

CSMA/ECA$_{\text{QoS}}$ eliminates VCs by picking a $B[\text{AC}]$ that is not equal to any of the other AC's counters. This is exemplified in Figure 9. The figure shows a contender with four ACs and $B_{\text{d}}[4] = [4, 8, 16, 16]$, for AC1 to AC4 respectively. AC1 effectively transmits, but AC3 suffers from a VC. It then selects a random backoff, $B[\text{AC3}] = 24$, which is different from any other AC's counters. Then, to eliminate future VCs in CSMA/ECA$_{\text{QoS}}$ the successes of other ACs's transmissions should also be taken into account. This is achieved by selecting a number whose absolute difference with each of the other AC's counters is not a multiple of each comparison's smallest deterministic backoff. Algorithm 3 shows the process of selecting what is referred to as a *Smart Backoff* in CSMA/ECA$_{\text{QoS}}$. It shows four ACs, although it can used to eliminate VCs with as many ACs as needed. Smart Backoff is used instead of a random backoff in CSMA/ECA$_{\text{QoS}}$.

```
$AC := 4$; // number of Access Categories
$CW_{\min}[AC]$; // $CW_{\min}$ for all ACs
$B_{\text{d}}[AC]$; // $B_{\text{d}}$ for all ACs
$B[AC]$; // current $B$ from all ACs
$k[AC]$; // current backoff stage
$F[AC] := \{0\}$;
$Cb[AC] := \{0\}$;
//
// looking for a suitable $B[i]$; $i \in [1, AC]$
//
while $(F \neq 1)$ or $(Cb \neq 1)$ do
    $B[i] \leftarrow \mathcal{U}[0, 2^{k[i]} CW_{\min}[i]]$;
    for $(j = 1; j \leq AC; j++)$ do
        if $(j \neq i)$ then
            $F[j] \leftarrow |B[i] - B[j]| \bmod [\min(B_{\text{d}}[i], B_{\text{d}}[j])]$;
            if $(F[j] \neq 0)$ then
                $F[j] \leftarrow 1$;
            if $(B[i] \neq B[j])$ then
                $Cb[j] \leftarrow 1$;
            else
                $Cb[j] \leftarrow 0$;
return $(B[i])$;
```

Algorithm 3: Smart Backoff: eliminating Virtual Collisions in CSMA/ECA$_{\text{QoS}}$

TABLE III: Deafult CSMA/ECA$_{\text{QoS}}$ contention parameters

AC	CW$_{\min}$	CW$_{\max}$	m	lowest B_{d}	highest B_{d}
BK	32	1024	5	15	511
BE	32	1024	5	15	511
VI	16	32	1	7	15
VO	8	16	1	3	7
Legacy	32	1024	5	15	511

What results from Algorithm 3 is a Smart Backoff counter that will not cause a VC on the next transmission attempts while in saturation.

IV. Performance Evaluation

In order to test the traffic differentiation capabilities of CSMA/ECA$_{\text{QoS}}$ we have used a customised version of the COST simulator [14], which is available via [15]. If not expressed otherwise, each point in the presented figures (including Figures 3 and 4) is obtained from averaging fifty executions of duration equal to one hundred seconds. Further considerations:

- Unspecified parameters follow the IEEE 802.11n (2.4 GHz) standard.
- All nodes can be assumed to be in communication range with each other.

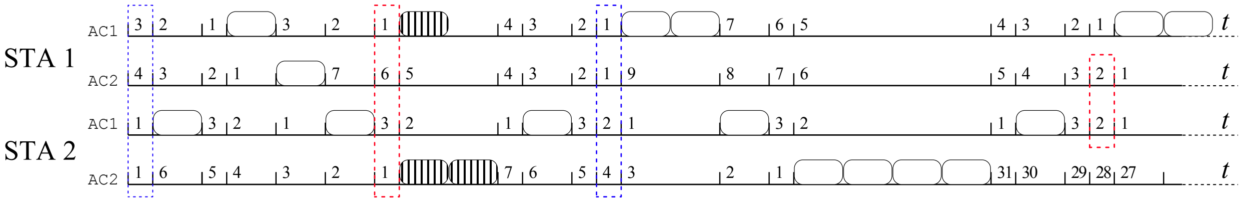
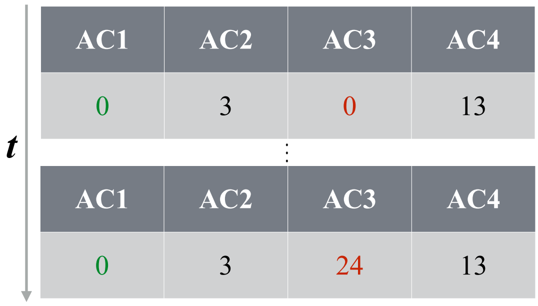

STA 1
AC1 3 2 1 3 2 1 4 3 2 1 7 6 5 4 3 2 1 t
AC2 4 3 2 1 7 6 5 4 3 2 1 9 8 7 6 5 4 3 2 1 t
STA 2
AC1 1 3 2 1 3 2 1 3 2 1 3 2 1 3 2 1 t
AC2 1 6 5 4 3 2 1 7 6 5 4 3 2 1 31 30 29 28 27 t

Fig. 8: An example of the temporal evolution of CSMA/ECA$_{\text{QoS}}$ in saturation ($CW_{\min}$[AC1,AC2] $= [8, 16]$; m[AC1,AC2] $= [5, 5]$)



AC1	AC2	AC3	AC4
0	3	0	13

AC1	AC2	AC3	AC4
0	3	24	13

Fig. 9: Four AC backoff counters in a CSMA/ECA$_{\text{QoS}}$ node with $B_{\text{d}}[4] = [4, 8, 16, 16]$. After a VC or collision, the new random backoff counter must not be equal to any other AC's counter

TABLE IV: PHY and MAC parameters for the simulations

PHY	
Parameter	**Value**
PHY rate	65 Mbps
Empty slot	9 μs
DIFS	28 μs
SIFS	10 μs
MAC	
Parameter	**Value**
Maximum retransmission attempts	7
Packet size (Bytes)	1024
MAC queue size (Packets)	1000

TABLE V: Updated CSMA/ECA$_{\text{QoS}}$ contention parameters for the simulations

AC	CW$_{\text{min}}$	CW$_{\text{max}}$	m	lowest B_{d}	highest B_{d}
BK	32	1024	5	15	511
BE	32	1024	5	15	511
VI	16	512	5	7	255
VO	8	256	5	3	127
Legacy	32	1024	5	15	511

- Not using Request-to-Send (RTS) or Clear-to-Send (CTS) messages.
- Collisions are assumed to take as much channel time as successful transmissions.

Additionally, Table IV provides information about relevant PHY and MAC parameters used in the simulator.

Apart from the assumptions presented above, the followings provide details about the traffic source generator, channel conditions and overall scenarios to be evaluated. Then, simulation results for achieved throughput, number of collisions and time between successful transmissions are presented.

A. *Simulation parameters*

1) Traffic conditions: There are two main scenarios regarding traffic generation in a node. The *saturated* traffic condition refers to a node that always has a packet for transmission in its MAC queue. On the other hand, a *non-saturated* node can also empty its MAC queue and withdraw from the channel contention. These states do not fall far from reality, for instance, a node might be in saturation while it is performing a file transfer. But if instead the node is only uploading a short file, does nothing for a period of time and then continues to perform a voice call over the network, its traffic will be considered to be non-saturated over that period of time.

In our simulator we mimic this behaviour by defining a packet arrival rate (ΔP). To simulate a saturated network, for instance, we set ΔP in every node to the channel capacity. If non-saturated traffic is the goal, ΔP will be set to a value way below the system maximum aggregated throughput in order to induce periods of inactivity upon each node. After extensive tests, $\Delta\text{P} = 2\text{Mbps}$ has proved to provide satisfactory results.

The performance in non-saturation for EDCA and CSMA/ECA$_{\text{QoS}}$ will be greatly dependent on the share of ΔP each AC gets, that is, the percentage of generated traffic that is going to be assigned to each AC. These shares, denoted $\Delta\text{P}_{\text{AC}}$ are distributed in the following way: $\Delta\text{P}_{\text{BK}} = 0.4\Delta\text{P}$, $\Delta\text{P}_{\text{BE}} = 0.3\Delta\text{P}$, $\Delta\text{P}_{\text{VI}} = 0.15\Delta\text{P}$ and $\Delta\text{P}_{\text{VO}} = 0.15\Delta\text{P}$. These values comply with the desired scenario.

2) Channel errors: The inability to receive an ACK frame is handled as a collision, both in EDCA and CSMA/ECA$_{\text{QoS}}$. This could happen due to channel imperfections preventing the receiver from decoding the transmissions. In order to simulate the effects of channel errors over the MAC protocol, we define the likelihood of a packet not being acknowledged, P_e. It affects every packet independently. That is, for every packet being transmitted we draw a number from a random variable $X \sim \mathcal{U}[0, 1]$, if the number drawn is lower than P_e the packet will not be acknowledged. In the case of an aggregation of several packets, it is considered a failed transmission only if all packets in the aggregation are affected negatively by P_e. A value of $P_e = 0.1$ has been selected for the simulations.

3) Scenarios to test: In this work we propose two main scenarios: saturation and non-saturation. The non-saturation scenario will be composed of nodes with $\Delta\text{P} = 2\text{Mbps}$.

B. *Simulation Results*

1) System throughput and collisions:

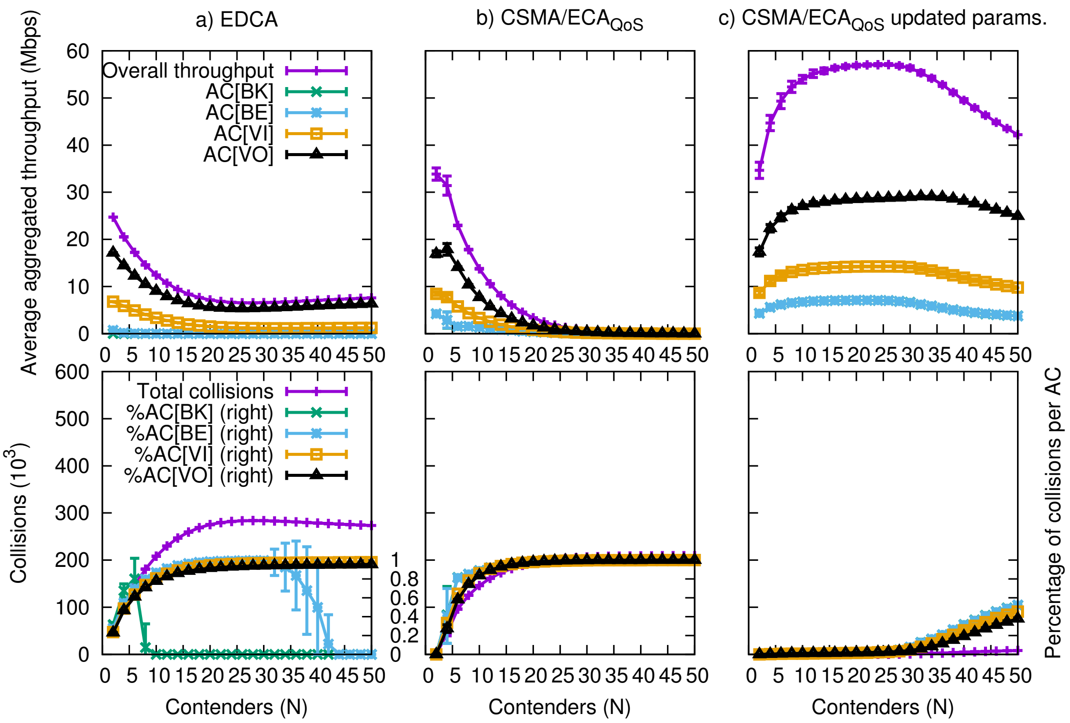


Fig. 10: Average Aggregated Throughput and Collisions for a) EDCA, b) CSMA/ECA$_{\text{QoS}}$ and c) CSMA/ECA$_{\text{QoS}}$ in saturation with parameters from Table V

- Saturation: Figure 10 gathers the simulation results for average aggregated throughput and collisions in a saturated network. Referring to EDCA results as Figure 10-a, it shows that the AC with the highest priority, AC[VO][4], achieves greater throughput than the others. This is a direct consequence of effective traffic differentiation in EDCA. The lower row of Figure 10 shows the total number of collisions as well as the average percentage of collisions suffered by each AC.

  Figure 10-a also shows that AC[BK] is not able to provide significant throughput for a high number of contenders. This is because its transmission attempts are being deferred in order to serve the high priority ACs. Furthermore, EDCA collisions show that at around 30 contenders AC[BE] also starts to get its transmission attempts deferred, easing the contention process for high priority ACs. This is the main reason of the overall throughput increase seen in Figure 10-a at around 30 nodes.

  Figure 10-b shows the average aggregated throughput and collisions of a saturated CSMA/ECA$_{\text{QoS}}$ network using the default contention parameters (see Table III). These settings do not allow the construction of collision-free schedules for many contenders. This is evidenced by the high number of collisions at around 20 contenders. Moreover, the figure shows CSMA/ECA$_{\text{QoS}}$ throughput decreasing faster than EDCA.

  To allow the construction of collision-free schedules, the contention parameters for CSMA/ECA$_{\text{QoS}}$ are modified to the ones shown in Table V. Figure 10-c provides throughput and collisions results with these new contention parameters. The figure shows a higher and more stable throughput, as opposed to EDCA, where it decreases rapidly as contenders join the network. Further, collisions start to increase at around 32 contenders, which is the maximum number of saturated collision-free contenders supported with the current contention parameters. It is relevant to highlight that unlike EDCA, CSMA/ECA$_{\text{QoS}}$ is able to accommodate transmissions for every AC when there is a high number of contenders (e.g.: $N \geq 40$).

  Figure 10-c shows a substantial increase in the throughput achieved by each AC with CSMA/ECA$_{\text{QoS}}$, as well as the overall aggregated throughput. In the same manner, is also shows a clear reduction in the number of collisions when using CSMA/ECA$_{\text{QoS}}$.
- Non-Saturation: In the non-saturation scenario, nodes in the network have $\Delta\text{P} = 2\text{Mbps}$. Moreover, the shares of ΔP assigned to each AC are defined in Table **??**. Results are derived from simulations performed with $P_e = 0.1$.

  Figure 11 shows the (from top to bottom row) average aggregated throughput, collisions and time between successful transmissions for (from left to right column) a) EDCA, b) CSMA/ECA$_{\text{QoS}}$ and c) CSMA/ECA$_{\text{QoS}}$ without Fair Share. In Figure 11-a, EDCA throughput keeps increasing up until one of its ACs reaches saturation due to starvation. This is the case of AC[BK] at around $N = 10$. It is only at around $N = 24$ where the rest of

[4] Starting at this point, ACs will be referred to as AC[*x*], where '*x*' is the abbreviation of the AC name, as in Table I.

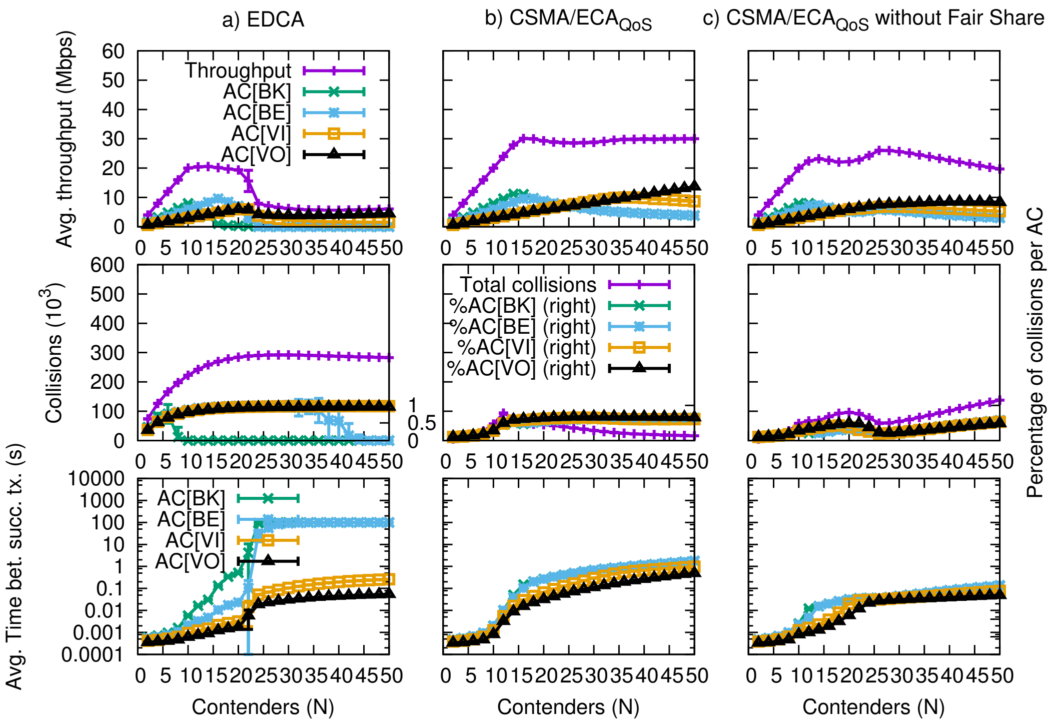


Fig. 11: Average Aggregated Throughput, Collisions and Time Between Successful Transmissions for a) EDCA and b) CSMA/ECA$_{\text{QoS}}$ and c) CSMA/ECA$_{\text{QoS}}$ network (without Fair Share) in non-saturation

ACs get completely saturated, and both figures start to resemble the ones presented in the saturation case.

In Figure 11-b at low N, contenders join and withdraw from the contention more frequently. This causes a disruption of any existing collision-free schedule, which is represented by the steep increase in collisions at $N = 14$. Once ACs get saturated, more periods of collision-free operation are achieved, accounting for the reduction in the total number of collisions

Even-though P_e induces retransmissions that degrade the performance, throughput gains and collision reductions with CSMA/ECA$_{\text{QoS}}$ are maintained. This is mainly due to the level of stickiness used and the additional increase provided by the Schedule Reset mechanism (see Section III-A).

2) Average time between successful transmissions: This metric refers to the average time between two consecutive successful transmissions of each AC. Figure 11-a shows that despite preventing transmissions from low priority ACs, AC[VO] and AC[VI] maintain a fairly low metric. This is specially relevant for time sensitive applications, like voice or video calls. Nevertheless, it implies that while high priority ACs are saturated, there is no effective throughput for other kind of traffic.

On the other hand, as shown in Figure 11-b, CSMA/ECA$_{\text{QoS}}$ takes more time between successful transmissions. This is partly because of the packet aggregation performed by Fair Share, which make transmissions longer. Furthermore, as low priority ACs are also transmitting, these also contribute to the increase in the time between successful transmissions.

Although EDCA keeps the time between successful transmission very low, it does so by almost preventing low priority ACs from accessing the channel. If Fair Share is deactivated in CSMA/ECA$_{\text{QoS}}$, the overall throughput might get negatively affected (no aggregation is performed), but the time between successful transmissions will be reduced, as shown in Figure 11-c.

3) Coexistence with EDCA: The followings are extracted from simulations performed with a network setup composed of 50% EDCA nodes and 50% CSMA/ECA$_{\text{QoS}}$ in non-saturation and $P_e = 0.1$. Figure 12 shows (from first to last row) average aggregated throughput, collisions and average time between successful transmissions for each group of nodes (from left to right column): a) EDCA nodes, b)CSMA/ECA$_{\text{QoS}}$ nodes and an additional test with c) CSMA/ECA$_{\text{QoS}}$ without Fair Share (w/o FS) nodes.

Even-though there are periods of collision-free operation, which is evidenced by the higher throughput experienced by CSMA/ECA$_{\text{QoS}}$ nodes, Figure 12-b shows a higher time between successful transmissions due to FS. Deactivating Fair Share causes a degradation in the overall throughput and an increase in the number of collisions, mostly due to reaching the saturation point faster. Nevertheless, Figure 12-c shows that CSMA/ECA$_{\text{QoS}}$ w/o FS nodes still experience higher throughput, lower collisions and similar time between successful transmissions than EDCA nodes, both in Figure 12-a and in a EDCA-only network, shown in Figure 11-a.

EDCA takes a much more aggressive approach towards providing traffic differentiation, mostly due to its tight contention

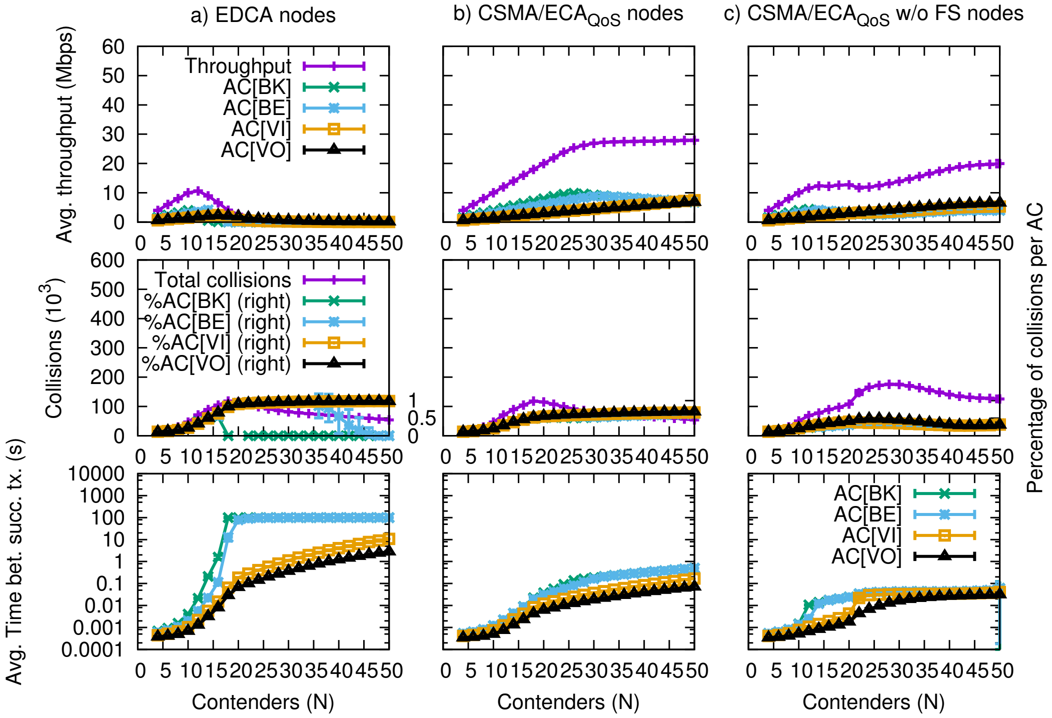


Fig. 12: Average aggregated throughput, Collisions and time between successful transmissions for a) EDCA nodes and b) CSMA/ECA$_{QoS}$ nodes and c) CSMA/ECA$_{QoS}$ network (without Fair Share) nodes in a 50%-50% mixed network

parameters and AIFS. In the mixed network of Figure 12-b, CSMA/ECA$_{QoS}$ nodes elevate the average metric due to its moderate contention, specially at low number of contenders. On the other hand, even at low number of contenders EDCA's tight contention parameters provoke an increase in the number of collisions, increasing the delay consequently (see Figure 12-a).

V. Conclusions

EDCA is able of providing effective traffic differentiation in WLANs. It does so instantiating DCF for each of its four supported MAC queues, or Access Categories (AC), and defining different contention and transmission parameters that allow an statistical differentiation among them. The proposed CSMA/ECA$_{QoS}$, is also a totally distributed MAC protocol for WiFi. It is able of providing traffic differentiation even for a high number of contenders. It does so by instructing nodes to pick a deterministic backoff counter after successful transmissions. These counters are based on the contention parameters in order to provide differentiation among the ACs.

Results highlight EDCA's difficulties to serve many contenders with multiple ACs. Its contention mechanism, being based on a random backoff, is in principle unable to eliminate collisions that degrade the overall performance of the network. Further, strict differentiation techniques like AIFS, and the additional transmission deferrals due to Virtual Collisions starve low-priority ACs in terms of throughput.

CSMA/ECA$_{QoS}$ is able to construct collision-free periods that provide an overall throughput increase, while still providing traffic differentiation. Moreover, CSMA/ECA$_{QoS}$ is able to bring traffic differentiation to crowded networks without killing the throughput of low priority ACs, as EDCA does. Further, because both protocols use similar contention parameters, CSMA/ECA$_{QoS}$ can coexist with EDCA nodes in the same network and still enjoy higher throughput and traffic differentiation.

Acknowledgment

Partially supported by the Spanish government under project CISNETS (TEC2012-32354).

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] "IEEE Standard for Information Technology - Telecommunications and Information exchange between systems. Local and Metropolitan Area Networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements," *IEEE Std 802.11eTM-2005*, 2005.

[3] L. Sanabria-Russo, J. Barcelo, B. Bellalta, and F. Gringoli, "A High Efficiency MAC Protocol for WLANs: Providing Fairness in Dense Scenarios," *arXiv preprint arXiv:1412.1395v2*, 2015.

[4] 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.

[5] A. Lindgren, A. Almquist, and O. Schelen, "Evaluation of Quality of Service Schemes for IEEE 802.11 Wireless LANs," in *26th Annual IEEE Conference on Local Computer Networks*, 2001, pp. 348–351.

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

[7] 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.

[8] L. Sanabria-Russo, J. Barcelo, A. Faridi, and B. Bellalta, "WLANs throughput improvement with CSMA/ECA," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2014, pp. 125–126.

[9] E. Perahia and R. Stacey, *"Next Generation Wireless LANs: 802.11 n and 802.11 ac"*. Cambridge University Press, 2013.

[10] "IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges," *IEEE Std 802.1dTM-2004*, 2004.

[11] A. Iera, G. Ruggeri, and D. Tripodi, "Providing Throughput Guarantees in 802.11e WLAN Through a Dynamic Priority Assignment Mechanism," *Wireless Personal Communications*, vol. 34, no. 1-2, pp. 109–125, 2005. [Online]. Available: http://dx.doi.org/10.1007/s11277-005-8729-7

[12] S. Son, K.-J. Park, and E.-C. Park, "Adaptive tuning of IEEE 802.11e EDCA for medical-grade QoS," in *Fifth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2013, pp. 650–651.

[13] T. Nilsson and J. Farooq, "A Novel MAC scheme for solving the QoS parameter adjustment problem in IEEE 802.11e EDCA," in *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2008, pp. 1–9.

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

[15] Sanabria-Russo, L. (2014) Github repository: CSMA-ECA-HEW. Webpage, accessed June 2015. [Online]. Available: https://github.com/SanabriaRusso/CSMA-ECA-HEW