Traffic Differentiation in Dense Collision-free WLANs

Luis Sanabria-Russo and 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 ACs, providing a way of prioritising 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 many nodes with multiple ACs, extending the network capacity without starving low priority ACs, as experienced in EDCA. 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, including coexistence.

Index terms— Wireless LAN, Multiaccess Communication, Collision-Free, QoS, EDCA

1 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 other wireless technologies, 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 random backoff period.

WiFi's increasing adoption coupled with the envisioned multi-media, real-time and bandwidth-hungry future use cases push the need for mechanisms to prioritise traffic in order to ensure Quality of Service (QoS) in dense scenarios with many nodes [2,3] ; 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 [4]), builds over DCF in order to provide this kind of traffic differentiation.

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

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) [5, 6] 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 dense WLAN networks [5]. 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 [7], 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 [5, 6, 8–10], this work is the first to incorporate traffic differentiation into the protocol with all of its extensions. We provide the following contributions:

- Adaptation of CSMA/ECA to support multiple queues for traffic differentiation.
- Introduction of the Smart Backoff mechanism for eliminating Virtual Collisions (VC).
- First simulation results on throughput, collisions and delay for CSMA/ECA with four ACs.
- Evaluation of the coexistence and backwards compatibility with EDCA.

Results show that CSMA/ECA is able to provide collision-free traffic differentiation under saturated traffic conditions for a greater number of contenders. 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 greater than the observed in a EDCA-only network.

An overview of the traffic differentiation techniques in EDCA will be provided in Section 2. Then, we will present CSMA/ECA and its ability to provide traffic differentiation in Section 3. The performance evaluation is shown in Section 4, while we draw our conclusions in Section 5.

2 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.

2.1 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

Table 1: AC relative priorities and mapping from 802.1D user priorities (extracted from [11])

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 2: Deafult EDCA parameters (extracted from [11])

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)

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 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 [11]. Traffic types, declared by the IEEE 802.1D standard [12] are then mapped to the four ACs in EDCA (MAC bridging). The mapping is shown in Table 1.

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 $AIFS[\text{AC}] = SIFS^3 + \sigma_e(AIFSN[\text{AC}]-1)$ 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 2 shows the default CW, AIFSN and TXOP values specified for EDCA.

As it can be observed in Table 2, 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, on average these ACs

[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.

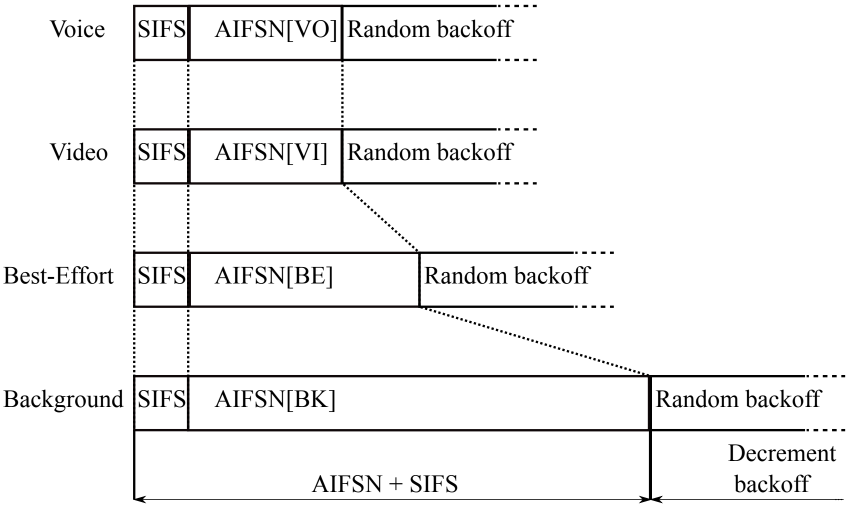

Figure 1: AIFSN values for each AC. From lowest (BK) to highest priority (VO)

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.

2.2 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 saturated 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 [7] and Section 4.2).

Therefore, great efforts have been directed towards parameter adjustments in EDCA [13–15]. 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 all ACs are required to have effective throughput, like in [14]. Further, by randomising the AIFS values it is possible to increase the channel utilisation in EDCA [15]. 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 for crowded scenarios.

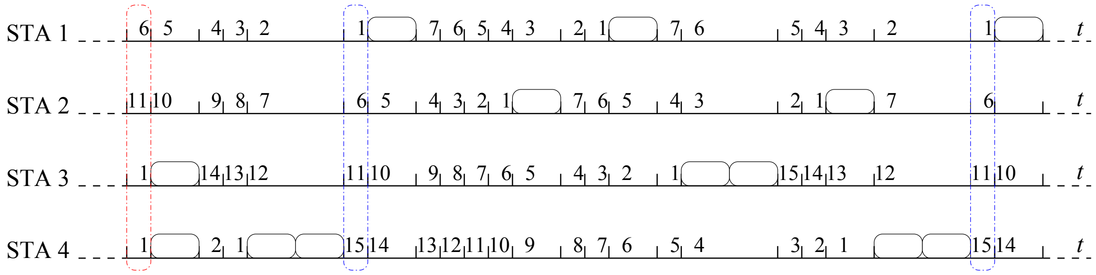

Figure 2: An example of the temporal evolution of CSMA/ECA in saturation

3 Traffic Differentiation with CSMA/ECA

Carrier Sense Multiple Access with Enhanced Collision Avoidance [5, 6] 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.

When a packet arrives at an empty MAC queue, 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 [6].

3.1 Supporting many more contenders with Hysteresis, Fair Share and the Schedule Reset mechanism

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 [5]. 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$_{\mathrm{Hys+FS}}$ respectively.

```
1  while the device is on do
2      $r \leftarrow 0$; $k \leftarrow 0$;
3      $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\mathrm{min}} - 1]$;
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                 $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\mathrm{min}} - 1]$;
14         until (r = R) or (success);
15         $r \leftarrow 0$;
16         $k \leftarrow 0$;
17         if success then
18             $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\mathrm{min}} - 1]$;
19         else
20             Discard packet;
21             $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\mathrm{min}} - 1]$;
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      $r \leftarrow 0$ ; $k \leftarrow 0$ ; $k_c \leftarrow k$;
3      $b \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\mathrm{min}} - 1]$;
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                 $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\mathrm{min}} - 1]$;
14         until (r = R) or (success);
15         $r \leftarrow 0$;
16         if success then
17             $B_d \leftarrow \lceil 2^k \mathrm{CW}_{\mathrm{min}}/2 \rceil - 1$;
18             $B \leftarrow B_d$;
19         else
20             Discard $2^{k_c}$ packets;
21             $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\mathrm{min}} - 1]$;
22         $k_c \leftarrow k$;
23     Wait until there is a packet to transmit;
```

Algorithm 2: CSMA/ECA$_{\mathrm{Hys+FS}}$: 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

The main difference between Algorithm 1 and Algorithm 2, and therefore between CSMA/CA and CSMA/ECA$_{\mathrm{Hys+FS}}$ 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.

CSMA/ECA$_{\mathrm{Hys+FS}}$ 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 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.

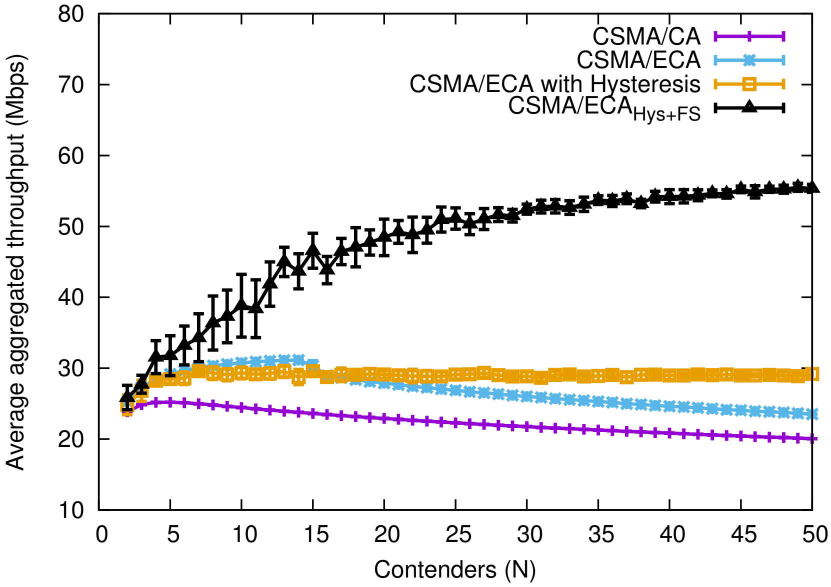

Figure 3: Average aggregate throughput ($CW_{\min} = 32$) [5]

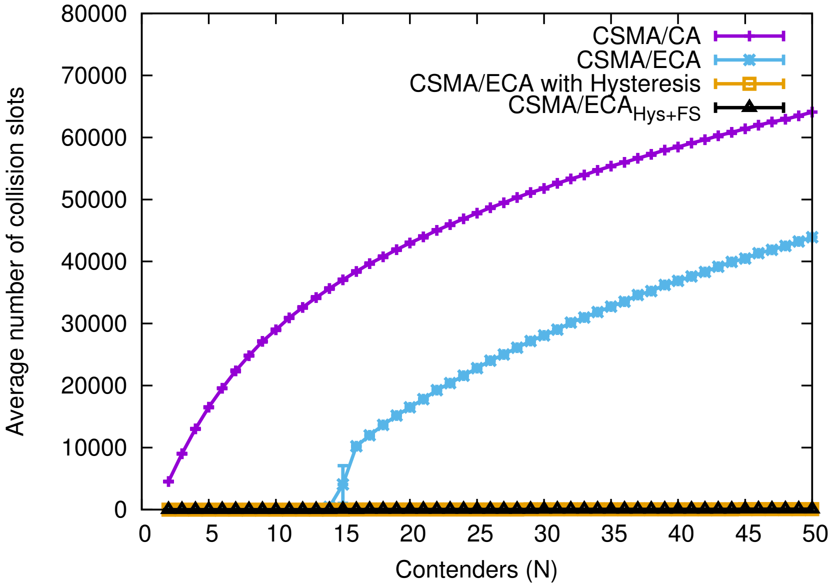

Figure 4: Average aggregate number of collision slots ($CW_{\min} = 32$) [5]

The *Schedule Reset* mechanism for CSMA/ECA$_{\text{Hys+FS}}$ introduced in [5] 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 $B_{\text{d}} = 31$ 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 $B_{\text{d}} = 31$. Starting from the left, the current $B_{\text{d}} = 0$ 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

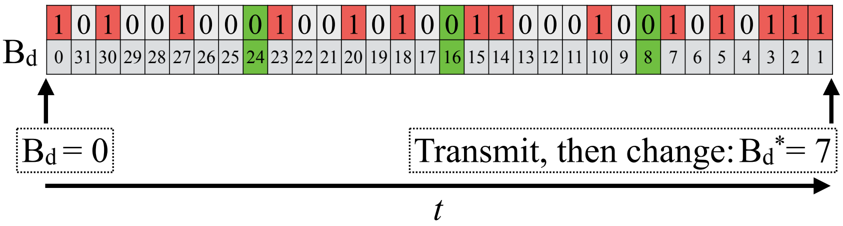

Figure 5: Example of the Schedule Reset mechanism (extracted from [5])

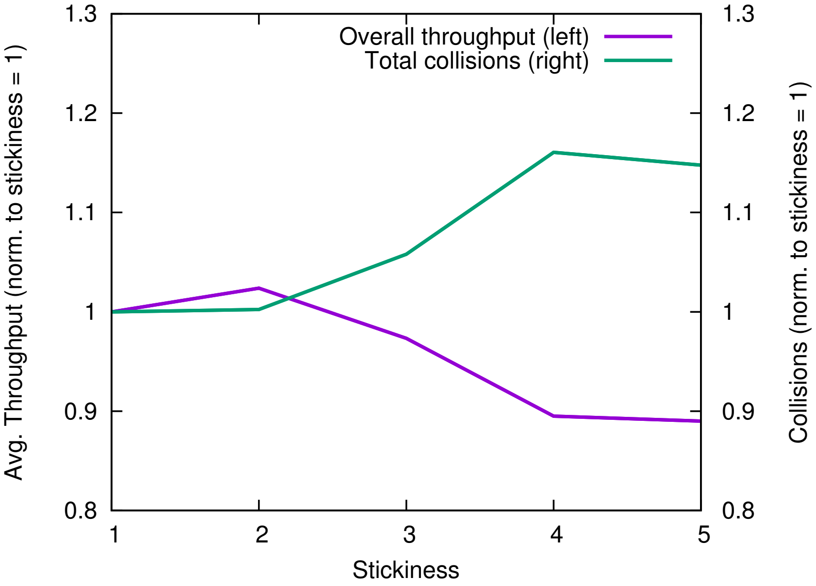

Figure 6: Average throughput and collisions with different levels of stickiness for CSMA/ECA$_{\mathrm{QoS}}$ in non-saturation conditions (as explained in Section 4.1)

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$_{\mathrm{Hys+FS}}$ by filling a bitmap b of size $B_\mathrm{d}+1$. Each bit t, $t \in \{0, \dots, B_\mathrm{d}\}$ 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 of all users' transmissions, therefore no additional collisions are introduced as a consequence of the schedule change. This implies a value of $\gamma = 2^{m-1}CW_{\min}/B_\mathrm{d}$. 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 after an effective schedule change has proven to be suitable for noisy scenarios [5], and the one used to provide the simulation results in Section 4. Stickiness is not a new concept to CSMA/ECA [9]. It simply instructs the contender to *stick* to the deterministic backoff even in the event of *stickiness* number of failed transmissions. This

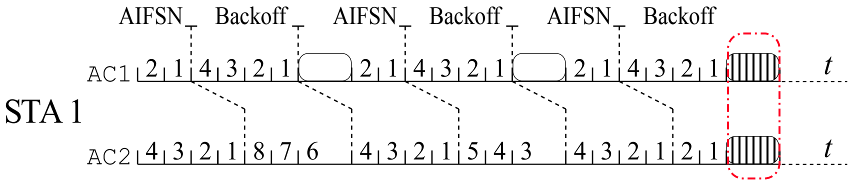

Figure 7: Example temporal evolution of CSMA/ECA$_{\text{Hys+FS}}$ with two ACs using AIFS resulting in a virtual collision. Only considering AIFSN values of 2 and 4, B_{d} of 4 and 8 for AC1 and AC2 respectively

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. Simulations results presented in Section 4 imply an increase from `stickiness` $= 2$ to `stickiness` $= 3$ after a successful Schedule Reset.

3.2 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 [5].

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 2.

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 3 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, $B_{\text{d}} = \frac{2^{k[\text{AC2}]} CW_{\min}[\text{AC2}]}{2} - 1 = 31$.

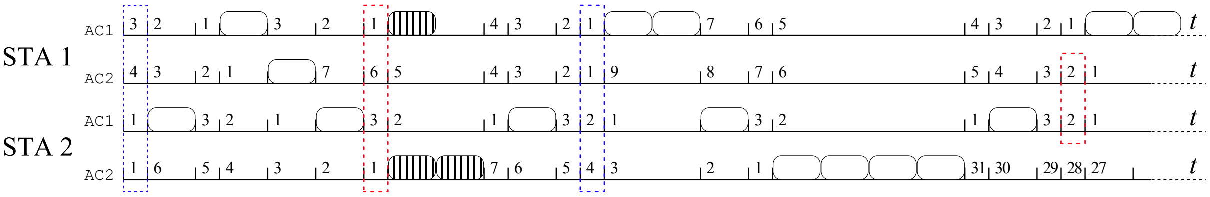

Figure 8: An example of the temporal evolution of CSMA/ECA$_{\text{QoS}}$ in saturation (CW_{min}[AC1,AC2] $=$ $[8, 16]$; m[AC1,AC2] $= [5, 5]$)

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.

3.3 Collisions and Virtual Collisions-free operation using Smart Backoff

Consider a complete schedule of length $C = 2^m CW_{\text{min}}$, 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 3). 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 $2^{m[\text{VO}]} CW_{\text{min}}[\text{VO}]/2 = 8$ contenders, where m[VO] is the maximum backoff stage of the AC with the smallest CW_{max}. Even-though Table 3 provides CSMA/ECA$_{\text{QoS}}$'s default contention parameters, these will be adjusted in Section 4.2 in order to support more contenders while in saturation.

VCs in CSMA/ECA$_{\text{QoS}}$ force lower priority 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[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 but do not consider the deterministic backoff selected by successful ACs. 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$_{\text{QoS}}$.

Table 3: Deafult CSMA/ECA$_{\text{QoS}}$ contention parameters

AC	$\mathbf{CW}_{\min}$	$\mathbf{CW}_{\max}$	**m**	**lowest** $B_{\mathbf{d}}$	**highest** $B_{\mathbf{d}}$
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

```
$AC := 4$; // 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$_{\text{QoS}}$

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

4 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 [16], which is available via [17]. 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.

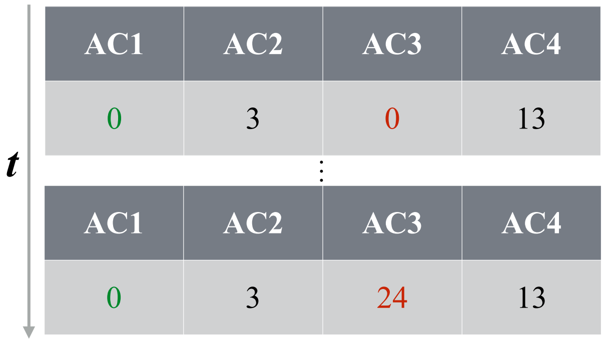

Figure 9: Four AC backoff counters in a CSMA/ECA$_{\text{QoS}}$ node with $B_{\text{d}}[4] = [4, 8, 16, 16]$. After a VC or collision, the new random backoff counter must not be equal to any other AC's counter

Table 4: 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

- All nodes can be assumed to be in communication range with each other.
- Not using Request-to-Send (RTS) or Clear-to-Send (CTS) messages.
- Aggressive Schedule Reset is used, with $\gamma = 1$.

Additionally, Table 4 provides information about relevant PHY and MAC parameters used in the simulator.

Apart from the assumptions presented above, the following 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.

4.1 Simulation parameters

4.1.1 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 empties 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.

Table 5: Updated CSMA/ECA$_{\text{QoS}}$ contention parameters for the simulations

AC	**CW**$_{\text{min}}$	**CW**$_{\text{max}}$	**m**	**lowest** $B_{\mathbf{d}}$	**highest** $B_{\mathbf{d}}$
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

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 aggregate throughput in order to induce periods of inactivity upon each node. After extensive tests, $\Delta\text{P} = 2\text{Mbps}$ was chosen.

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 $\Delta\text{P}_{\text{VO}} = 0.15\Delta\text{P}$.

4.1.2 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.

4.1.3 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 $\Delta\text{P} = 2\text{Mbps}$.

4.2 Simulation Results

4.2.1 System throughput and collisions

Saturation: Figure 10 gathers the simulation results for average aggregate throughput and collisions in a saturated network. Figure 10-a shows the results referring to EDCA. 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] and AC[BE] are not able to achieve 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] starts

[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 1.

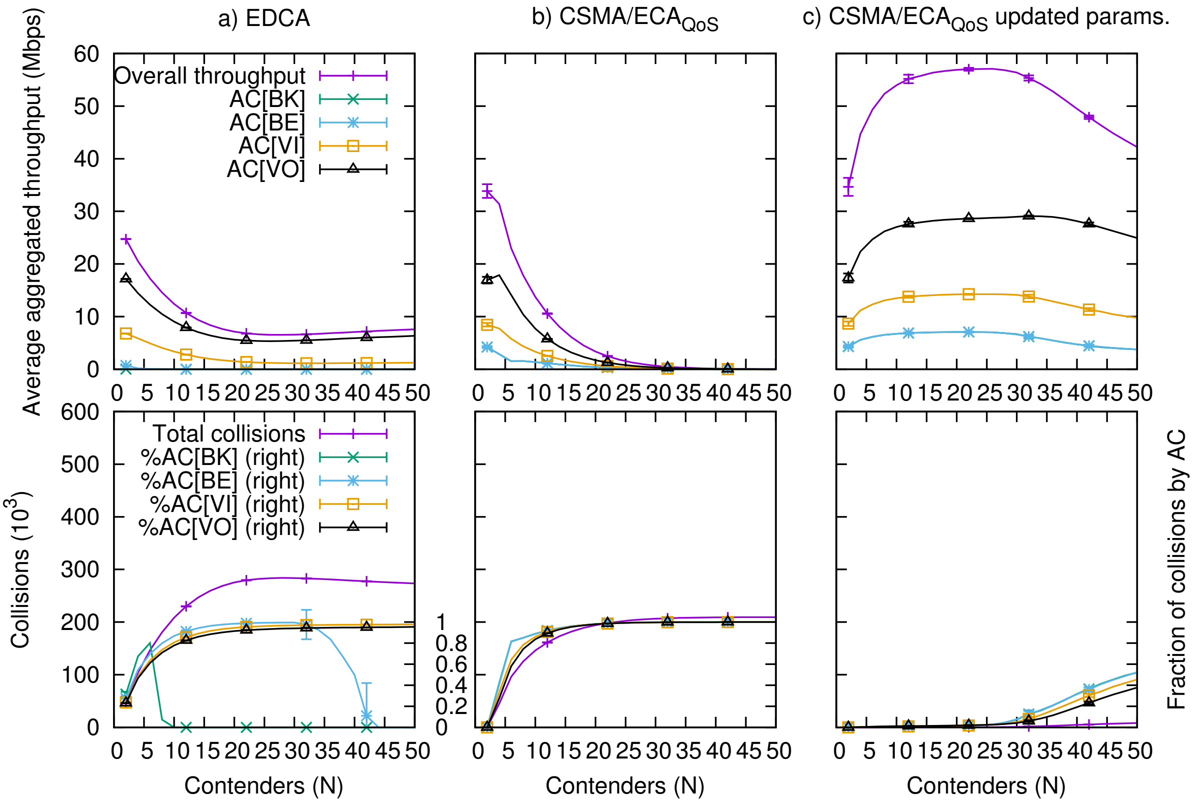

Figure 10: Average aggregate Throughput and Collisions for a) EDCA, b) CSMA/ECA$_{\mathrm{QoS}}$ and c) CSMA/ECA$_{\mathrm{QoS}}$ with parameters from Table 5 in saturation

to starve, 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 aggregate throughput and collisions of a saturated CSMA/ECA$_{\mathrm{QoS}}$ network using the default contention parameters (see Table 3). 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. However at low number ($N \leq 8$), CSMA/ECA$_{\mathrm{QoS}}$ is able to reach collision-free operation, achieving higher throughput than EDCA (Figure 10-b).

To achieve collision-free operation for many more contenders, the maximum backoff stage for AC[VI] and AC[VO] in CSMA/ECA$_{\mathrm{QoS}}$ are increased to match the other ACs, as shown in Table 5. Figure 10-c provides throughput and collisions results with these new contention parameters. The figure shows higher throughput for a wider number of contenders, 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 updated contention parameters. It is relevant to highlight that unlike EDCA, CSMA/ECA$_{\mathrm{QoS}}$ is able to accommodate transmissions for every AC when there is a high number of contenders (e.g., $N \geq 40$). In the same manner, Figure 10-c also shows a clear reduction in the number of collisions when using CSMA/ECA$_{\mathrm{QoS}}$.

The updated parameters for CSMA/ECA$_{\mathrm{QoS}}$ (Table 5) make a tradeoff between throughput and aggressiveness. Even-though allowing CSMA/ECA$_{\mathrm{QoS}}$'s deterministic backoffs to grow past EDCA's default parameters increases the time between successful transmissions (as shown in the following Figure 11), it prevents the starvation of low priority ACs and provide a greater overall throughput due to

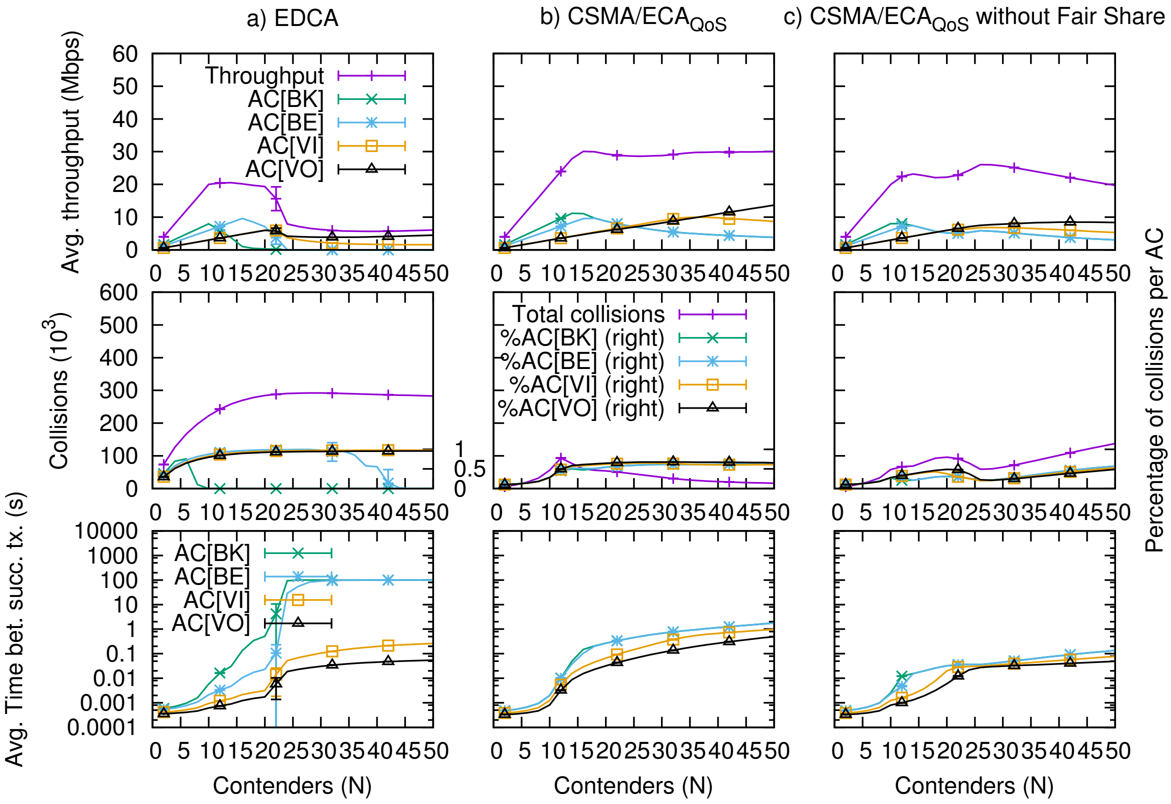

Figure 11: Average aggregate Throughput, Collisions and Time Between Successful Transmissions for a) EDCA and b) CSMA/ECA$_{QoS}$ and c) CSMA/ECA$_{QoS}$ network (without Fair Share) in non-saturation

Table 6: Saturation points in non-saturation scenarios (N)

Protocol	BK	BE	VI	VO
EDCA	10	16	22	22
CSMA/ECAQoS	14	16	36	50
CSMA/ECAQoS w/o FS	10	16	28	28

a reduction in the number of collisions and the aggregation performed by Fair Share.

Non-Saturation: In the non-saturation scenario nodes in the network have $\Delta P = 2$Mbps. Moreover, the shares of ΔP assigned to each AC are defined in Section 4.1. Results are derived from simulations performed with $p_e = 0.1$.

Figure 11 shows the (from top to bottom row) average aggregate throughput, collisions and time between successful transmissions for (from left to right column) a) EDCA, b) CSMA/ECA$_{QoS}$ and c) CSMA/ECA$_{QoS}$ without Fair Share (both CSMA/ECA$_{QoS}$ results using the contention parameters presented in Table 5). 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 ACs get completely saturated. At this point, both the throughput and collision figures for EDCA start to resemble the ones presented in the saturation case. Table 6 shows the saturation points for each protocol.

In Figure 11-b at $N \leq 14$, contenders join and withdraw from the contention more frequently. This causes a disruption of any existing collision-free schedule, which is represented by the increase in collisions at $N = 14$. Once ACs get saturated, more periods of collision-free operation are achieved, accounting for the subsequent 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 3.1).

4.2.2 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 EDCA provides a fairly low metric for AC[VO] and AC[VI] as a consequence of starving low priority ACs. 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 for AC[VI] and AC[VO]. This is partly because of the packet aggregation performed by Fair Share, which makes transmissions longer. However, low priority AC do not starve as they are able to perform successful transmissions even for a high number of contenders. 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 is reduced to EDCA levels as shown in Figure 11-c, while the throughput is still higher.

Table 6 shows the saturation points for each protocol. That is, the number of nodes at which the network gets saturated. As it can be seen in the table, CSMA/ECAQoS saturates at higher number of contenders. This translates in an enhanced ability to provide higher throughput for a wider number of users than EDCA.

4.2.3 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 aggregate 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. The Fair Share mechanism is deactivated in order to reduce the time between successful transmissions.

As shown in Figure 12-b, CSMA/ECA$_{\text{QoS}}$ nodes achieve greater throughput. Even-though there are periods of collision-free operation, Figure 12-c shows that the time between successful transmissions can be further reduced by not using 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 (at lower N). 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 an EDCA-only network, shown in Figure 11-a.

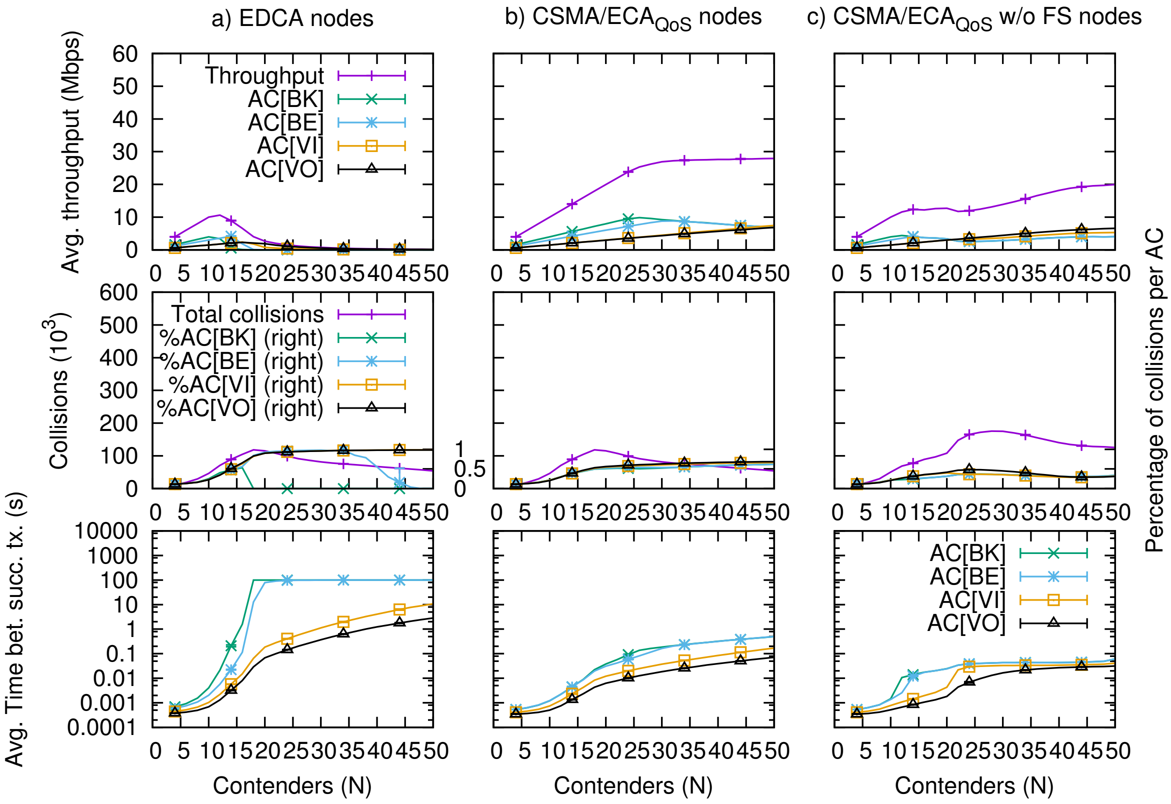

Figure 12: Average aggregate throughput, Collisions and time between successful transmissions for a) EDCA nodes and b) CSMA/ECA$_{\mathrm{QoS}}$ nodes and c) CSMA/ECA$_{\mathrm{QoS}}$ network (without Fair Share) nodes in a 50%-50% mixed non-saturated network

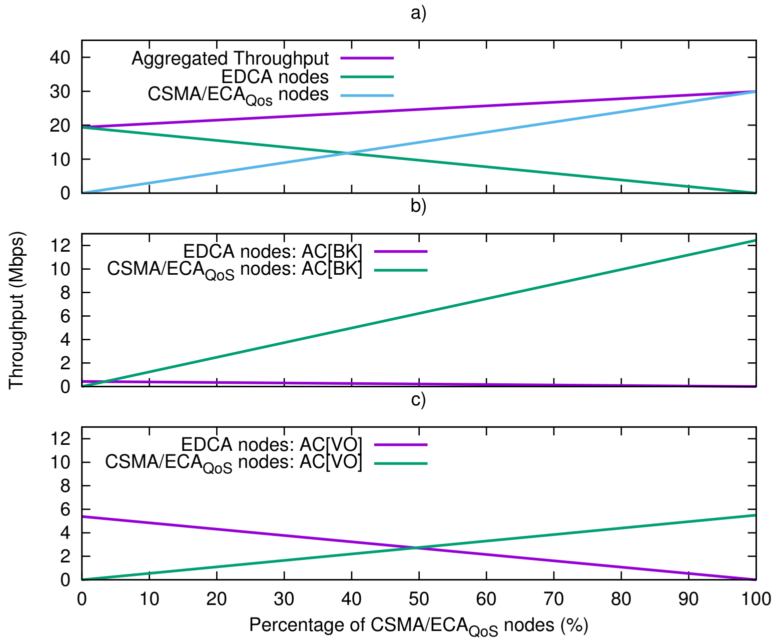

Figure 13: Throughput for a mixed network setup with 20 contenders in non-saturation: a) Overall throughput, b) Throughput of AC[BK], c) Throughput of AC[VO]

EDCA takes a much more aggressive approach towards providing traffic differentiation, mostly due to its tight contention parameters and AIFS. On the other hand, even at low number of contenders EDCA's tight contention parameters provoke an increase in the number of collisions. Furthermore, it starves low priority ACs (like BK and BE); increasing the delay consequently (see Figure 12-a).

Figure 13 shows the a) overall, b) AC[BK], and c) AC[VO] throughput for a non-saturated mixed network with $p_e = 0.1$ and 20 contenders as the percentage of CSMA/ECA$_{\text{QoS}}$ nodes increases from 0 to 100%. Figure 13-a shows that the overall throughput of the system increases with the proportion of CSMA/ECA$_{\text{QoS}}$ nodes. This is mainly because CSMA/ECA$_{\text{QoS}}$ nodes do not starve low priority ACs and keep high priority ACs throughput unaffected.

AC[BK] for EDCA nodes in Figure 13-b shows little variation. As seen in Figure 11-a, at this number of contenders ($N = 20$) AC[BK] has almost no throughput due to starvation by EDCA. On the other hand, for CSMA/ECA$_{\text{QoS}}$ this low priority AC shows higher throughput even for low percentages of CSMA/ECA$_{\text{QoS}}$ nodes ($N > 10\%$).

Figure 13-c, showing AC[VO]'s throughput, is consistent with Figure 13-a, that is, it is possible to observe a throughput degradation due to the coexistence in both protocols. Nevertheless, AC[VI] throughput degrades almost at the same rate for both protocols (same slope around ±0.6).

5 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.

Results highlight EDCA's problems at serving many contenders with multiple ACs. Specifically, 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$_{\text{QoS}}$ is able to construct collision-free periods that provide an overall throughput increase, while still providing traffic differentiation for many more contenders. That is, CSMA/ECA$_{\text{QoS}}$ 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$_{\text{QoS}}$ 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 802.11 TGax, "Status of IEEE 802.11 HEW Task Group," http://www.ieee802.org/11/Reports/tgax_update.htm, 2014.

[3] B. Bellalta, "IEEE 820.11 ax: High-Efficiency WLANs," *IEEE Wireless Communications (arXiv preprint arXiv:1501.01496)*, Accepted July 2015.

[4] "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.

[5] 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.

[6] 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.

[7] 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.

[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] 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.

[10] 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.

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

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

[13] 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

[14] 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.

[15] 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.

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

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