Collision-free WLANs: From Concepts to Working Protocols

Luis Sanabria-Russo

TESI DOCTORAL UPF / ANY 2016

DIRECTORS DE LA TESI

Boris Bellalta, Jaume Barcelo, Miquel Oliver
Department of Information and Communication Technologies (DTIC)

Universitat Pompeu Fabra Barcelona

To my family.

Acknowledgments

My appreciation is extended to the projects and institutions that supported this research, such as the Spanish government through CISNETS (TEC2012-32354), and by the Catalan Government through SGR 2014-1173. Specially UPF's DTIC for its constant support and kind disposition.

To Jaume, Boris, and Francesco, for showing me the thrill derived from well conducted experimental discovery. I am hooked now.

Abstract

Using a deterministic backoff after successful transmissions as a distributed contention mechanism for access to the channel can draw benefits from scheduled operation, like the elimination of simultaneous transmissions, and Quality of Service (QoS). Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) is a compatible protocol that implements such technique, outperforming the current Medium Access Control (MAC) protocol in WiFi. This work describes several extensions to CSMA/ECA that allows it to support many more contenders in a collision-free schedule in very diverse scenarios, while evenly distributing the available throughput among users. Further, it introduces CSMA/ECA with multiple queues for providing traffic differentiation. Lastly, it presents progressive evolutions of an off-the-shelf hardware implementation of CSMA/ECA using open source tools.

Resum

L'ús d'un backoff determinista després de transmissions exitoses com a mecanisme de contenció distribuït per l'accés al canal pot extreure beneficis de l'operació planificada, com l'eliminació de les transmissions simultànies, i Qualitat de Servei (QoS). Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) és un protocol compatible que implementa aquesta tècnica, superant l'actual protocol Medium Access Control (MAC) en Wi-Fi. Aquest treball descriu diverses extensions de CSMA/CA que li permet suportar molts més contendents sense col·lisions en molt diversos escenaris, mentre distribueix equitativament el throughput entre els usuaris. A més, s'introdueix CSMA / ECA amb múltiples cues per proporcionar diferenciació de trànsit. Finalment, es presenten evolucions progressives d'una implementació de maquinari off-the-shelf de CSMA / ECA utilitzant eines de codi obert.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Wireless Local Area Networks (WLANs and/or WiFi) are almost ubiquitous. Many things can be said about the reasons of its popularity, but few arguments can undermine the incredible impact standardisation has had on its success.

The evolution of the WiFi standard, commonly referred to as the IEEE 802.11 set of protocols, responds to numerous requirements. Some are needed to ensure backwards compatibility, like employing the same, completely decentralised channel access mechanism. Others are feature demands from applications, like increased throughput, or traffic differentiation for time sensitive flows (e.g., video streaming). Furthermore, given its widespread adoption and increasing popularity, crowded WLANs are now much more common. Therefore, the ability to handle many clients in crowded scenarios is expected to be a key feature of future WiFi.

IEEE 802.11 defines modulation/coding schemes, time spacing between transmissions, channel frequency and bandwidth, frame format, channel access, backwards compatibility, and many more details needed to effectively function as a WLAN. The last approved amendment to the Physical (PHY) and Medium Access Control (MAC) protocols, called IEEE 802.11ac operates exclusively at the 5 GHz band. It proposes higher modulation and coding schemes, frame aggregation, better spectrum efficiency, and wider WiFi channels to achieve linear increases in throughput.

Furthermore, 802.11ac inherits some of the technologies introduced in the previous amendment, 802.11n, like Quality of Service (QoS) mechanisms for prioritising access using queues or Access Categories (AC).

The first wave of 802.11ac products was released in early 2013, and by the end of 2014 an IEEE Task Group (TG) was formed to draft a next amendment to the WiFi standard, called 802.11ax [32]. Dubbed Higher Efficiency WLAN (HEW), this new version of the standard introduces transmission techniques to increase the spectrum efficiency, like uplink/downlink transmission to multiple clients at the same time using Orthogonal Frequency-Division Multiple Access (OFDMA), and multiple antenna data transmissions techniques, and dynamic adjustments of the Clear Channel Assessment (CCA) detection thresholds to minimise the effects of interfering networks. HEW also aims at incorporating some of 802.11ac features into the 2.4 GHz band, like higher modulation/coding schemes. These new technologies propose improvements in real world performance as experienced by the user, targeting the challenges of crowded WLAN scenarios [20].

Moving at a faster pace, the research community has noticed that despite all the proposed enhancements, every standard amendment keeps inheriting the same Distributed Coordination Function (DCF) to orchestrate transmissions, mostly for backwards compatibility reasons. DCF uses a Binary Exponential Backoff (BEB) algorithm to randomise a client's transmission attempt, and is part of the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol defined by IEEE 802.11. Although praised for its simplicity, adaptability to different scenarios and widespread adoption, DCF is not capable of completely avoiding collision events, which significantly degrade performance.

Recent analysis for 802.11ac shows how the throughput decreases as the number of active clients in the WLAN grows [16]. The impact of collisions is specially relevant in WiFi scenarios with a large number of clients, where a significant percentage of the channel time is wasted as a consequence. This, not just negatively affects performance, but also misuses the available spectrum, undermining the potential of the aforementioned technologies incorporated into the amendment.

Collisions are mostly a consequence of the random channel access orchestrated at the MAC layer by DCF. It instructs clients willing to transmit to pick a random backoff counter between zero and a Contention Window (CW) minus one i.e., $[0, \mathrm{CW}-1]$. Consequently, each node decrements its own backoff counter by one if the channel is observed free for a predefined period of time. Once the backoff expires (reaches zero) the client will attempt transmission. A transmission is considered successful upon the reception of an Acknowledgement (Ack) frame from the receiver. If so, the current CW is reset to its minimum value, $\mathrm{CW}_{\min}$. If no Ack is received a collision is assumed. A retransmission is scheduled after doubling the current CW and drawing a new random backoff. Doubling the CW is part of the BEB algorithm, and it effectively reduces the collision probability of the retransmission attempt by decreasing the likelihood of two or more clients drawing the same random backoff counter.

It might be evident that granting access to the channel using a random backoff produces collision events. This is because there is nothing preventing two or more clients from drawing the same random backoff and attempting a transmission at the same time. Furthermore, as the CW is reset after a successful transmission, a successful client's next transmission attempt is more likely to suffer a collision.

Despite the many enhancements, WiFi still employs a random backoff to orchestrate channel access. This caps the maximum performance of the network by wasting network resources recovering from collisions, contributing to the MAC bottleneck issue [51].

Motivated by the growing use of time sensitive applications, a polling-based channel access mechanism was integrated into the 802.11 standard. This extension to DCF is called the Point Coordination Function (PCF). PCF uses the Access Point (AP) as a centralised coordinator to poll clients for transmissions using a Round-Robin scheduler. Although contention-free, PCF has not gotten widespread adoption nor is it available on most of the current hardware.

Among many proposals to enhance or replace the MAC protocol, some show that by obtaining additional information from the network, like the number of active clients (or contenders) the CW values can be dynamically adjusted, so the collision probability can be kept low [12, 45]. These approaches yield encouraging results, nevertheless the performance evaluation under varying traffic conditions provides only an approximation to the average number of contenders, yielding sub-optimal results. Furthermore, other means of obtaining the current number of contenders may introduce a centralised entity to an otherwise completely decentralised channel access protocol [8].

Other more disruptive proposals aim at using OFDMA to translate the backoff procedure to the frequency domain [66]. Nevertheless, because the choice of OFDM subcarriers is made using a random backoff, like in DCF there is still the possibility of collisions. Furthermore, as OFDMA is only supported by recent standard amendments, backwards compatibility is compromised. Transitioning to a Time Division Multiple Access (TDMA)-like schedule by using a deterministic (instead of random) backoff value after successful transmissions promises collision-free operation without requiring a centralised entity, or compromising backwards compatibility [7, 19, 29, 42]. The idea is based on the fact that every client successfully transmits at a different moment in time. In a network with N clients in contention for the channel, a consecutive collision-free transmission can be achieved if all successful transmitters pick the same backoff value. A collision-free schedule is then achieved when no collisions are observed between the clients' transmissions.

Learning-BEB or Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) [7] is a fully decentralised and collision-free MAC for WLANs. It differs from CSMA/CA in that it uses a deterministic backoff, $B_{\mathrm{d}} = \lceil \mathrm{CW}_{\min}/2 \rceil - 1$ after successful transmissions, where $\mathrm{CW}_{\min}$ is the minimum contention window of typical value $\mathrm{CW}_{\min} = 16$. By doing so, contenders that successfully transmitted on schedule n, will transmit without colliding with other successful nodes in future cycles. Even-though it does not require knowledge of the current number of contenders (N), when $N > B_{\mathrm{d}}$ collisions reappear and the throughput approximates to CSMA/CA's.

ZeroCollision MAC (ZCMAC) [42] builds collision-free schedules by instructing clients to reserve a transmission slot from a predefined virtual schedule of M transmission-slots in length. Learning-MAC (L-MAC) [19] defines a learning parameter that determines whether or not to repeat the choice of backoff counter, only based on results from previous transmission attempts. Both ZCMAC and L-MAC require the construction of a virtual schedule, which optimal length is equal to the number of active clients in the WLAN. If said virtual schedule is shorter than the number of active clients, collisions reappear. Conversely, an overestimation will increase the time between successful transmissions of a client. Nevertheless, obtaining the exact number of contenders might not be possible on a per-schedule basis. Consequently, an adaptive schedule length mechanism is proposed [19].

Without requiring information about the number of contenders, [29] proposes a mechanism for adjusting the value of the deterministic backoff, and therefore changing the length of the collision-free schedule. Nevertheless, because the decision making is not based on exactly what the client observes from the channel, a schedule reduction may actually introduce new collisions (the same happens to adaptive ZCMAC and L-MAC). Moreover, in situations when clients have different schedule lengths, the available throughput would not be evenly distributed among all the clients (some clients having larger schedules than others), compromising what is known to be WiFi's throughput fairness.

Despite the limitations of the aforementioned proposals, all of them provide better performance than DCF on specific simulation scenarios. Nevertheless, incorporation into the standard is not easy task, requiring many years for its validation. On top of this, experimental evaluation using real hardware prototypes may be challenging, mainly because implementations require expensive hardware and/or use complicated software tools that lack appropriate MAC layer abstractions. Furthermore, despite being equipped with a microprocessor capable of executing the set of instructions that drive the MAC operation, WiFi cards manufacturers close-sourced the implementation code, preventing the experimental evaluation of new MAC protocols, and consequently slowing down innovation [14].

This work introduces extensions to Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) [7] that allows it to built and adapt collision-free schedules with many clients. A combination of schedule length adaptation and a frame aggregation technique ensures an even distribution of the available throughput among nodes, for single queue and QoS scenarios with multiple CSMA/ECA queues. Furthermore, this work provides the first experimental results from a CSMA/ECA implementation on commercial hardware using open-source tools on a real testbed.

Objectives

The main objective of this thesis is to propose a set of mechanisms that will allow CSMA/ECA to handle many contenders in a collision-free schedule. In order to achieve this goal the following summarises the secondary objectives:

- Provide a clear overview of collision-free channel access protocols in WLANs.
- Propose a mechanism to allow the construction of throughput-fair collision-free schedules in a completely decentralised manner using CSMA/ECA.
- Propose and evaluate a mechanism for adapting the schedule length in CSMA/ECA.
- Evaluate the effects of different traffic sources, channel errors, and crowded scenarios over CSMA/ECA with simulations.
- Provide collision-free traffic differentiation using multiple CSMA/ECA queues.
- Prototype and evaluate CSMA/ECA experimentally using commercial hardware.
- Serve as a base for supporting context-specific wireless MAC protocols for WLANs.

Contributions

The contributions of this work cover different aspects. First, our proposal uses the same contention parameters as the current standard implementation, so it is backwards compatible. Furthermore, CSMA/ECA does not require a centralised entity in order to achieve collision-free schedules or outperform DCF, specially in crowded WLANs.

Second, the mechanism that allows CSMA/ECA to adapt the schedule length does not create additional collisions. Moreover, as the required information is easily available to each client, it can be easily incorporated on to other similar protocols, like [29].

Thirdly, experimental results from a commercial hardware implementation using open-source tools show that CSMA/ECA outperforms the current MAC standard, specially in crowded scenarios. But more importantly, it suggests that taylor-made MAC protocols may tackle specific problems derived from a dynamic network context, like changing application-level requirements. In the end, it is argued that the ability to dynamically change the MAC protocol can solve many compatibility issues, make a more efficient use of the network resources, and provide enhanced QoS mechanisms on future WLANs [14].

The following summarises the contributions of this thesis:

- Introduces Hysteresis and FairShare for the first time. Establishing throughput fairness for even a large number of contenders.

- Shows first simulation results of CSMA/ECA with Hysteresis and FairShare (CSMA/ECA$_{\text{Hys+FS}}$), under scenarios with different packet error rate and traffic conditions.

- Introduces the Schedule Reset mechanism for reducing the periods between successful transmissions during collision-free operation.

- Performs an evaluation of CSMA/ECA$_{\text{Hys+FS}}$ with multiple EDCA-like queues (called CSMA/ECA$_{\text{QoS}}$) and introduces a mechanism to eliminate virtual collisions.

- Experimental evaluation of the first implementation on commercial hardware of CSMA/ECA (with Hysteresis and the Schedule Reset mechanism, too).

Beyond the encouraging results derived from this work, CSMA/ECA with all of its extensions could be challenged by overlooked scenarios, or may not satisfy requirements from specific applications. As mentioned in [14], one protocol can hardly fit all uses. With this in mind, it is thought the future of MAC protocol development is likely to be software-based. WNICs manufacturers as well as other incumbents in the industry should envision solutions in the form of platforms, where research can be made and tested. Innovation in turn would come from the set of tools provided to the MAC Protocol Designer to develop better solutions. Even-though this work covers a single MAC protocol proposal (as well as several extensions to it), it also dwells into the process of commercial hardware implementation using open source tools. Suggesting that solving WiFi's MAC issues related to compatibility, scalability and QoS, among others, is possible, but only if all entities involved in the process are up for customisation.

Chapter 2

RELATED WORK

2.1 Distributed channel access through contention

Time in WLANs is divided into tiny empty slots of fixed length σ_e, collisions, and successful slots of length σ_c and σ_s, respectively. Collision and successful slots contain collisions or successful transmissions, making them several orders of magnitude larger than empty slots ($\sigma_e \ll \min(\sigma_s, \sigma_c)$). One of the effects of collisions is the degradation of the network performance by wasting channel time on collisions slots.

Recent advances in the WLANs Physical layer (PHY) [28, 51] push the research community towards the development of MAC protocols able to take advantage of a much faster PHY. By reducing the time spent in collisions nodes are able to transmit more often, which in turn translates to an increase in the network throughput. Further, the upcoming MAC protocols for WLANs should lean towards a fully decentralised approach in order to avoid injecting extra control traffic that may reduce the data throughput.

The current MAC protocol for WiFi implements CSMA/CA. It does so through DCF, deferring each user's transmission for a random back-off period when the channel is sensed idle and then for a fixed period of DIFS[1] before immediately accessing the channel. When a packet ar-

[1]The DCF Inter-Frame Space (DIFS) is defined in [3]. It is equal to 34μs for 802.11

rives at the MAC queue, a contender generates a random backoff, $B \leftarrow \mathcal{U}[0, \text{CW}_{\text{min}} - 1]$. After listening to the channel for a duration of SIFS[2], each empty slot decrements B by one. Once the counter expires ($B = 0$) the user listens the channel for an additional DIFS period, if detected free the transmissions is performed. Otherwise, the contender keeps listening to the channel until it is detected free for a period of DIFS.

A transmitter expects a confirmation from the receiver in the form of Acknowledgement (ACK) frames. If an ACK is received for a given transmission, it is considered a success. Consequently, the node resets its Current Contention Window, $\text{CW}_{\text{curr}} \leftarrow \text{CW}_{\text{min}}$, before drawing another random backoff counter[3]. On the other hand, the absence of an ACK is treated as a collision. The node proceeds to double its current contention window incrementing its backoff stage $k \in [0, \cdots, m]$ by one, $\text{CW}_{\text{curr}} \leftarrow \min(2^k \text{CW}_{\text{min}}, \text{CW}_{\text{max}})$, before drawing a random backoff $B \leftarrow \mathcal{U}[0, \text{CW}_{\text{curr}} - 1]$ for the retransmission attempt. This process is illustrated in Algorithm 1, where the node starts by setting the retransmissions counter and backoff stage to zero ($r \in [0, R]$ and $k \in [0, m]$ respectively, with m the maximum backoff stage, and $R = m + 1$ the retransmission limit. The typical value for m is 5). Lines 10 and 13 highlight the handling of a collision.

n/ac/ax in the 5GHz band.

[2]The Short Inter-Frame Space (SIFS) [3] is equal to 16μs for 802.11 n/ac/ax in the 5GHz band.

[3]If the node has other packets to transmit in its MAC queue.

```
while the device is on do
    r ← 0; k ← 0;
    B ← U[0, 2^k CW_min − 1];
    while there is a packet to transmit do
        repeat
            while B > 0 do
                wait 1 slot;
                B ← B − 1;
            Attempt transmission of 1 packet;
            if collision then
                r ← r + 1;
                k ← min(k + 1, m);
                B ← U[0, 2^k CW_min − 1];
        until (r = R) or (success);
        r ← 0;
        k ← 0;
        if success then
            B ← U[0, 2^k CW_min − 1];
        else
            Discard packet;
            B ← U[0, 2^k 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 R retransmissions are reached, the packet waiting for transmission is dropped.

Collisions keep degrading the overall throughput of a WLAN as the number of contenders increases [16]. This is mainly due to channel time wasted in the failed transmission. The RTS/CTS message exchange between transmitter/received was originally intended to solve the hidden node problem in WLANs [50]. However, it also provides advantages for a large number of contenders as it reduces the collision duration, which compensates for the RTS/CTS overhead. Initially, a transmitter enters in contention in order to send a short Request to Send (RTS) message to the receiver. Consequently, the receiver performs contention to respond with a Clear to Send (CTS) message (which is received by all contenders), allocating the channel to the transmitter. Upon reception of the CTS message, the transmitter is granted contention free access to the channel. Collisions using the RTS/CTS mechanism are shorter that using Basic Access (BA) (in which collisions are normally assumed to occupy as much channel time as a successful transmission), given the short size of RTS and CTS packets.

2.2 Fine tuning contention parameters

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 [9, 32]; 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 for providing this kind of traffic differentiation.

2.2.1 Enhanced Distributed Channel Access

EDCA proposes up to four queues or Access Categories (ACs), each one working as an instance of the DCF with different contention parameters that allow a statistical prioritisation among them [50]. Traffic types, declared by the IEEE 802.1D standard [1] are then mapped to the four ACs in EDCA (MAC bridging). The mapping is shown in Table 2.1. Essentially, at the Medium Access Control (MAC) layer EDCA allows the higher priority ACs to access the channel more often.

It 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 ($\mathrm{CW}_{\min}$ and $\mathrm{CW}_{\max}$, 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.

Every backlogged AC joins the contention for the channel by drawing a random backoff, $B[\mathrm{AC}] \leftarrow \mathcal{U}[0, \mathrm{CW}_{\min}[\mathrm{AC}] - 1]$; where $\mathrm{CW}_{\min}[\mathrm{AC}]$ is the minimum contention window for said AC. Each AC waits for a fixed $\mathrm{AIFS}[\mathrm{AC}] = \mathrm{SIFS} + \sigma_e(\mathrm{AIFSN}[\mathrm{AC}] - 1)$ period of inactivity in the channel and then starts decrementing its random backoff. Each passing empty slot decrements $B[\mathrm{AC}]$ in one. When the backoff counter expires ($B[\mathrm{AC}] = 0$), the AC will attempt transmission. A successful transmission is declared upon the reception of an Acknowledgement (ACK) frame from the receiver, a collision is assumed otherwise.

EDCA instructs the successful AC to reset its current contention window ($\mathrm{CW}_{\mathrm{curr}}[\mathrm{AC}]$) to $\mathrm{CW}_{\min}[\mathrm{AC}]$, while failed transmissions force a retransmission attempt after doubling the current contention window, that is, $\mathrm{CW}_{\mathrm{curr}}[\mathrm{AC}] \leftarrow \min(2 * \mathrm{CW}_{\mathrm{curr}}[\mathrm{AC}], \mathrm{CW}_{\max}[\mathrm{AC}])$. Table 2.2 shows the default CW, AIFSN and TXOP values specified for EDCA.

As it can be observed in Table 2.2, 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 will access the channel quicker; thus providing priority in the contention.

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. 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 $CW_{curr}[AC]$ for the lower priority ACs; just as it is done in the event of a real collision.

Table 2.1: AC relative priorities and mapping from 802.1D user priorities

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 2.2: Deafult EDCA parameters

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)

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.

2.2.2 EDCA enhanced

Because ACs in EDCA perform contention independently of the others, each AC mimics a DCF station. This explains why the collision probability in EDCA is higher than in 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 [43]).

Great efforts have been directed towards parameter adjustments in EDCA, mostly to ensure QoS for high priority ACs while maintaining low delay and losses [33, 48, 69]. 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 [69]. Further, by randomising the AIFS values it is possible to increase the channel utilisation in EDCA [48].

MAC parameter adjustment algorithms work as functions that select future values for contention or transmission parameters in each AC. Most consider changing the contention windows, mainly because these were the only contention parameters in DCF. Nevertheless, adjustment of AIFS, and/or TXOP are also possible. These can be classified as [15]:

- Static or Adaptive: static algorithms define contention parameters for all ACs, which remain unchanged throughout the operation. An adaptive algorithm selects the best contention parameters for each AC depending on the detected flows. They also react to network congestion variations.

- Measurement or Model based: measurement-based algorithms divide time in periods, say Δt. By observing different metrics, e.g.: AC queue size, or collision rate during Δt seconds, the algorithm estimates better MAC parameters to increase QoS in high priority ACs. On the other hand, model-based algorithms update MAC parameters every time a new flow is observed. These approaches can be combined, for instance, using Call Admission Control (CAC) coupled with a monitoring period of Δt. Such combination may accept or reject flows, and announce new MAC parameters according to the measured metrics, like [10].

- Centralised or Distributed: a key characteristic of EDCA, and DCF before it, is its distributed nature. That is, EDCA defines a static, measurement-based algorithm that reacts to network conditions. MAC parameter computation in distributed algorithms is performed at each node, independently. Centralised algorithms, additionally, make use of information obtained by a centralised entity, like the Access Point (AP). Distributed algorithms do not need additional control messages to adjust MAC parameters, as opposed to centralised ones.

Another example of adaptive, distributed and measurement-based algorithms for WLANs is proposed in [53]. It follows EDCA rules for updating the CW after failed transmissions. Nevertheless, after a successful transmission the $CW_{curr}[AC]$ is *slowly* reset to $CW_{min}[AC]$ by computing a Multiplication Factor (MF), which itself depends on the ratio between failed transmissions and transmission attempts. This *Slow Reset* of the $CW_{curr}[AC]$ reduces the collision probability of the immediate attempt after a successful transmission. In [39], distributed TXOP adaptation is combined with a CAC. Called Enhanced TXOP (ETXOP), this algorithm estimates the network congestion using the number of times a station's backoff counter is frozen, and then adjusts TXOP sizes so the application's requirements for each AC are met. Combined with a distributed model-based algorithm, namely a CAC which handles flows coming from applications at each node, EXTOP ensures that only flows with a guaranteed QoS are accepted for contention.

Centralised algorithms may take advantage of an AP's point of view of the network and of its ability to transmit MAC parameter updates in beacon frames. In [10], a centralised CAC algorithm distinguishes between VoIP and TCP flows, and at the same time between uplink and downlink traffic. Measuring each flow type, required bandwidth, and average frame length the CAC reacts to each new flow request, adjusting CW, AIFS or

TXOP values to comply with defined VoIP requirements, like delay and frame loss. The CAC handles the flows differently depending on its characteristics:

- Downlink TCP flow: if the number of existing downlink TCP flows is below an estimated threshold, the flow is admitted.
- Uplink TCP flow: if the number of existing uplink TCP flows is below an estimated threshold, the flow is admitted. Otherwise, other CW_{min} values are proposed via a Beacon frame so the newly arrived flow can be admitted.
- Downlink VoIP flow: if the number of packets in the queue for other downlink VoIP flows is below a threshold, the new flow is accepted and the threshold updated.
- Uplink VoIP flow: the flow is admitted if the grade of service of existing flows is not affected. On the positive case, other CW_{min} and AIFS values are proposed to admit the newly arrived flow.
- If no other parameter update is feasible, the flow is rejected.

Algorithms may be combined, or focus on iteration in order to provide advantageous conditions for high priority traffic. Nevertheless, as proposals deviate too much from the IEEE 802.11 MAC standard, the chances of being accepted as an amendment decreases [51, 73]. Moreover, performance evaluations should implement updated audio and video source models, using specifications of widely-used codecs in order to mimic realistic scenarios [46, 75].

The way traffic differentiation is defined in IEEE 802.11e is through a static, completely distributed, and measurement-based algorithm, that is, EDCA. As its goal is to provide QoS to high priority ACs, low priority traffic is often deferred to the point of throughput starvation. Additionally, EDCA's random backoff mechanism is prone to an elevated number of real and virtual collisions, widening the effects of throughput starvation to higher priority ACs[4].

2.3 Avoiding collisions

Performing time slot reservation for each transmission is a well known technique for achieving a high throughput and maintaining Quality of Service (QoS) in TDMA-like schemes [22]. Trying to achieve a TDMA-like operation in WLANs by modifying DCF's random backoff procedure has been shown to provide similar benefits [29]. The following are MAC protocols for WLANs, decentralised and capable of attaining greater throughput than DCF by constructing collision-free schedules using reservation techniques. A survey of collision-free MAC protocols for WLANs is presented in [19].

2.3.1 Zero Collision MAC

Zero Collision MAC (ZCMAC) [42] achieves a zero collision schedule for WLANs in a fully decentralised way. It does so by allowing contenders to reserve one empty slot from a predefined virtual schedule of M-slots in length. Backlogged stations pick a slot in the virtual cycle to attempt transmission. If two or more stations picked the same slot in the cycle, their transmissions will eventually collide. This forces the involved contenders to randomly and uniformly select other empty slot from those detected empty in the previous cycle plus the slot where they collided. When all N stations reserve a different slot, a collision-free schedule is achieved.

ZCMAC is able to outperform CSMA/CA under different scenarios. Nevertheless, given that the length of ZCMAC's virtual cycle has to be predefined without actual knowledge of the real number of contenders in

[4]Throughput starvation is first observed in AC[BK], and then in AC[BE] as the number of contenders increases.

the deployment, the protocol is unable to provide a collision-free schedule when $N > M$. Furthermore, if M is overestimated ($M \gg N$), the fixed-width empty slots between each contender's successful transmission are no longer negligible and contribute to the degradation of the network performance. Additionally, ZCMAC nodes require common knowledge of where the virtual schedule starts/ends. This is not considered in CSMA/CA and constitutes an obstacle towards standardisation.

2.3.2 Learning-MAC

Learning-MAC (L-MAC) [19] is another MAC protocol able to build a collision-free schedule for many contenders. It does so defining a *learning strength* parameter, $\beta \in (0, 1)$. Each contender starts by picking a slot s for transmission of the schedule n of length C at random with uniform probability. After a contender picks slot $s(n)$, its selection in the next schedule, $s(n + 1)$, will be conditioned by the result of the current attempt. Equations (2.1) and (2.2) extracted from [19] show the probability of selecting the same slot $s(n)$ in cycle $n + 1$.

$$\left.\begin{aligned} p_{s(n)}(n+1) &= 1, \\ p_j(n+1) &= 0, \end{aligned}\right\} \quad \textit{Success} \tag{2.1}$$

$$\left.\begin{aligned} p_{s(n)}(n+1) &= \beta p_{s(n)}(n), \\ p_j(n+1) &= \beta p_j(n) + \frac{1-\beta}{C-1}, \end{aligned}\right\} \quad \textit{Collision} \tag{2.2}$$

for all $j \neq s(n),\ j \in \{1, \ldots, C\}$. That is, if a station successfully transmitted in $s(n)$, it will pick the same slot on the next schedule with probability one. Otherwise, it follows Equation (2.2).

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

L-MAC is able to achieve higher throughput than CSMA/CA with a very fast convergence speed. Nevertheless, the choice of β suppose a previous knowledge of the number of empty slots ($C - N$, where N is the number of contenders), which is not easily available to CSMA/CA. It may require a centralised entity [8], or provide only approximate values depending on the traffic conditions [12, 45].

Further extensions to L-MAC introduced an *Adaptive* schedule length in order to increase the number of supported contenders in a collision-free schedule. This adaptive schedule length is doubled or halved depending on the presence of collisions or many empty slots per schedule, respectively. The effects of reducing the schedule length may provoke a reconvergence phase which can result in short-term fairness issues. Moreover, L-MAC nodes also require common knowledge of the start/end of the schedule.

The aforementioned reservation-like protocols, namely, L-MAC and ZCMAC could be adapted to support traffic differentiation by using multiple schedules. Semi-Random Backoff (SRB) [29] is able to build collision-free schedules using a deterministic backoff after successful transmissions. Further, SRB proposes traffic differentiation using different deterministic backoffs for each AC. Nevertheless, frame aggregation techniques are not considered, leading to throughput unfairness issues. Moreover, results for non-saturated scenarios do not follow realistic traffic sources for voice and video, providing inaccurate modelling of nodes' behaviour regarding the arrival or withdrawal from contention. Finally, as backwards compatibility is considered a key aspect of WiFi's popularity, evaluations should include evaluations of mixed scenarios using accurate models for traffic sources.

2.4 The issues with ubiquity

WLAN's channel is shared, so in order to maximise the available throughput there has to be as much concurrent transmissions as possible.

As IEEE 802.11 amendments[5] propose new functionalities, backwards compatibility mechanisms are enforced to support legacy stations[6]. For

[5]IEEE 802.11 n/ac/ax.

[6]Stations using IEEE 802.11g or older specifications a refered to as legacy.

instance, all stations in the Basic Service Set (BSS) need to agree which primary channel number they use, bandwidth to operate, as well as being informed about the capabilities supported by the BSS, like frame aggregation or protection mechanisms for ensuring the correct operation of legacy stations. Similarly, Clear Channel Assessment (CCA) carrier sensing threshold values are defined by the standard. These are usually advertised by the Access Point (AP).

2.4.1 Very crowded scenarios

The IEEE 802.11ax High Efficiency WLAN (HEW) Task Group (TGax) [32] envisions what can now be considered typical crowded scenarios, like dense residential buildings, offices, or open places like stadiums [31]. There are three possible conditions in these scenarios:

- Many nodes associated with a single AP: where the high number of nodes increases the probability of collisions.
- Overlapping Basic Service Sets (OBSS): when multiple BSSs operate in the same WiFi channel.
- A combination of the above.

OBSS imply that transmissions from neighbouring WLANs affect the local contention for the channel. That is, if a node associated with AP i, n_i, is in contention for transmission, it should determine the state of the channel. If sensed idle, the node will continue the contention process and eventually attempt transmission. Evidently if other node, associated with neighbouring AP j, say n_j, is transmitting and in communication range, n_i will sense the channel as busy, even though it is associated to a different AP. n_i will then defer its transmission.

2.4.2 Carrier sensing thresholds

CSMA/CA performs two types of carrier sensing for attempting transmissions: Physical Carrier Sensing (PCS), and Virtual Carrier Sensing

(VCS). The PCS is done at the Physical Layer (PHY) via the CCA, that is, if after the backoff period the channel energy during a period of DIFS is observed below an Energy Detection (ED), the node will attempt transmission immediately, considering the channel as empty. On the other hand, if the channel is busy the node will keep listening to the channel until it is sensed free for DIFS and then transmits. The other carrier sensing mechanism, VCS, uses control frames to broadcast information about the transmission duration (RTS/CTS mechanism frames [50]). This way other listening stations are aware of ongoing transmissions.

There are several works studying the effects of physical carrier sense sensitivity over the throughput [72]. Further, it is posible to achieve optimal adjustment of the ED threshold so the throughput is maximised in WLANs [4].

The TGax, in charge of pushing the 802.11 ax amendment has proposed several scenarios and functionalities to be considered for simulation purposes [31]. Furthermore, it focuses on PCS and dynamic ED threshold adaptation to increase spacial reuse [32], specifically the Dynamic Sensitivity Control (DSC) [4, 5, 23]. Several studies show that by dynamically adjusting the ED threshold it is possible to reduce the effects of neighbouring WLANs that produce throughput degradation in crowded scenarios.

Dynamic Sensitivity Control (DSC)

Dynamically fine tuning the CCA thresholds can leverage the effects of neighbouring transmissions. Figure 2.1 shows what is defined as a node's Interference Range (I_R) and Communication Range (C_R). C_R is shorter than I_R. This is because the threshold associated to C_R, referred to as C_T, is purposefully higher than the ED threshold[7], this means that signals detected above ED will trigger the CCA mechanism, forcing the receiver to stop the backoff process (if any). Nevertheless, only received signals whose energy is greater than C_S are successfully decoded.

[7] C_T has a typical value of -62dB in 802.11ax for transmissions without OFDM hearders, while the ED threshold is set to -82dB.

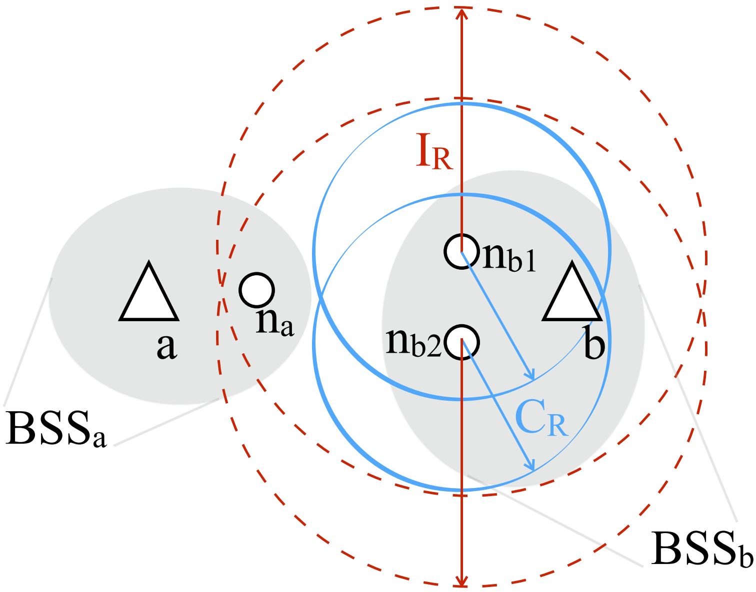

Figure 2.1: Graphic representation of an Overlapping Basic Service Set (OBSS)

The well studied exposed and hidden node problems [50] are closely related to the ability of a node to correctly determine the state of the channel. The former is produced when a node is starved due to neighbouring transmissions. This can be exemplified using Figure 2.1. In the figure, node n_a is associated with AP a[8] of BSS$_a$. As it is located within nodes n_{b1} and n_{b2}'s I$_\text{R}$, transmissions from any node belonging to BSS$_b$ will trigger n_a's CCA mechanism unnecessarily. The hidden node problem is related to disruptions of ongoing transmissions caused by nodes of the same BSS or other OBSS, producing failures. These interruptions are often the result of a node's inability to detect ongoing transmissions[9].

DSC is a proposed algorithm for 802.11ax that seeks to reduce the im-

[8]Represented with a triangle in the figure.

[9]Caused by distance or obstacles among transmitters. In sum, the hidden node does not sense any active signal within its I$_\text{R}$.

pact of neighbouring transmissions in OBSSs by dynamically adjusting the ED threshold. Again, Figure 2.1 can be used to exemplify DSC's motivation. For instance, n_a and any node from BSS$_b$ can actually perform transmissions at the same, given that their respective $\mathrm{I_R}$ do not include any node from a different BSS.

To achieve a dynamic adaptation of the sensitivity[10], DSC measures the signal strength of received AP Beacons (R dBm), and defines a margin M dBm and an Upper Limit L dBm. The ED threshold is then updated after every received Beacon following Equations 2.3 and 2.4. [23] uses $L = -40$ and $M = 20$ as an example, showing that is possible to confine the $\mathrm{I_R}$ only to direct neighbours. Furthermore, DSC achieves a considerable reduction on the number OBSS in the TGax residential building scenario (as the Scenario HEW shown in figure 2.2), a priority use case for HEW. In the figure, the Scenario HEW represents a common residential building following TGax scenarios [31]. The height from floor to ceiling is $F = 3$ meters, and $L = 10$ meters. APs (represented by the triangle) are randomly placed inside each apartment $q \in [1, \cdots, A]$ (of area equals to L^2) and fixed at $z = 1.5$ meters from the floor.

$$i = \min(\mathrm{ED}, L) \tag{2.3}$$

$$\mathrm{ED} = i - M \tag{2.4}$$

Alternatives

There seems to be consensus among the research community about the benefits of DSC. Nevertheless, *when* and *how* the threshold is to adapted is still an open issue. For instance, which initial values of ED, L and M to use, or when to apply Equations 2.3 and 2.4. In [24] it is shown that fine calibration at the edges of each BSS is needed in order to maximise the throughput. Similarly, [40] argue that there are still opportunities to find

[10]In this case, of the ED threshold.

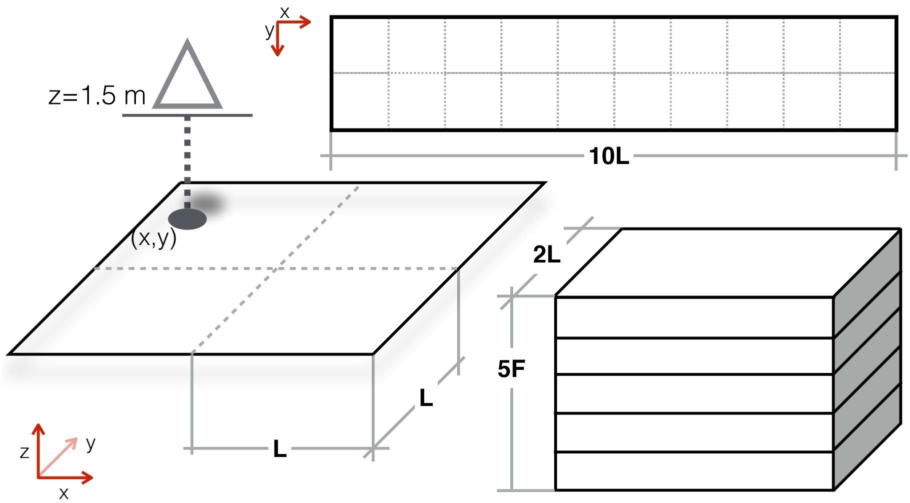

Figure 2.2: Scenario HEW: a typical dense residential building

better sensitivity adjustment mechanisms, such as coordination among neighbouring APs.

Transmit Power Control (TPC), efficient channel allocation, and DSC are combined in order to confine each nodes' I_R more efficiently, increasing spatial reuse [26]. Nevertheless, results do not encourage or reveal any incentive for using TPC in stations. Other techniques use bits in the PHY header to identify each BSS, this is called Colouring. Results from tests of TPC, DSC and Color [25] reveal that:

- As the BSS ID is injected into the PHY header, colouring allows nodes to abort the reception procedure of a frame from a different BSS. Falling back to an idle state.
- Colouring does not show better performance than DCS in the residential building scenario [34].
- As AP transmissions must be detected by all nodes in the BSS, specific considerations are required for a TPC method for the AP.

- There are compatibility issues with Colouring and TPC, meaning that all STAs and APs should be compatible. Otherwise may be negatively affected.

This is a hot research topic for HEWs. Fortunately, TGax has provided all the necessary guidelines and suggestions to the research community, so it may propose new mechanisms over an agreed upon frame of reference [31, 32].

Chapter 3

PART I: CARRIER SENSE MULTIPLE ACCESS WITH ENHANCED COLLISION AVOIDANCE

CSMA/ECA [7] is a fully decentralised and collision-free MAC for WLANs. It differs from CSMA/CA in that it uses a deterministic backoff, $B_{\mathrm{d}} = \lceil \mathrm{CW}_{\min}/2 \rceil - 1$ after successful transmissions, where $\mathrm{CW}_{\min}$ is the minimum contention window of typical value $\mathrm{CW}_{\min} = 16$. By doing so, contenders that successfully transmitted on schedule n, will transmit without colliding with other successful nodes in future cycles [29].

Collisions are handled as in CSMA/CA, which is described in Algorithm 1. Algorithm 2 provides an explanation of CSMA/ECA's deterministic backoff mechanism, which main difference with CSMA/CA (and therefore with Algorithm 1) relies on the selection of a deterministic backoff after a successful transmission (compare Line 18 in Algorithm 1 with Line 18 in Algorithm 2). Figure 3.1 shows an example of CSMA/ECA dynamics with four contenders.

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**
      $B_d \leftarrow \lceil 2^k \mathrm{CW}_{\min}/2 \rceil - 1$;
      $B \leftarrow B_d$;
    else
      Discard packet;
      $B \leftarrow \mathcal{U}[0, 2^k \mathrm{CW}_{\min} - 1]$;
  Wait until there is a packet to transmit;

Algorithm 2: CSMA/ECA

In Figure 3.1, the *STA-#* labels represent stations willing to transmit. The horizontal lines represent a time axis with each number indicating the amount of empty slots left for the backoff to expire. Stations willing to transmit begin the contention for the channel by waiting a random backoff, B. The first outline highlights the fact that stations STA-3 and STA-4 will eventually collide because they have selected the same B. After

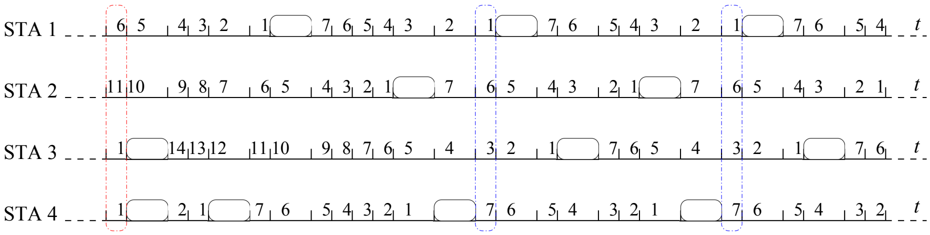

Figure 3.1: An example of the temporal evolution of CSMA/ECA in saturation

recomputing the random backoff, STA-4's attempt results in a successful transmission, which instructs the node to use a deterministic backoff, $B_{\text{d}} = 7$ in this case. By doing so, all successful STAs will not collide among each other in future cycles.

Collision slots being orders of magnitude larger than empty slots degrade the network performance. When CSMA/ECA builds the collision-free schedule all contenders are able to successfully transmit more often, increasing the aggregate throughput beyond CSMA/CA's. Figure 3.2, shows the average aggregate throughput of CSMA/ECA, CSMA/CA and collision-free protocolos presented in Chapter 2.3, namely ZCMAC and L-MAC [19].

Referring to Figure 3.2, CSMA/ECA is able to achieve an aggregate throughput that goes beyond CSMA/CA up until the number of contenders is greater than $B_{\text{d}} + 1$ ($N = 8$ in the case of the figure). Beyond this point, the network will have a mixed behaviour relating to backoff mechanisms: some nodes will successfully transmit and use a deterministic backoff while others will collide due to the lack of empty slots and return to a random backoff. As more contenders join the network, CSMA/ECA performance will approximate to CSMA/CA's. This effect is also observed in ZCMAC and L-MAC. It happens because the virtual cycle used by these protocols (M for ZCMAC and C for L-MAC) is lower than the number of contenders, N[1].

[1]L-MAC and ZCMAC curves in Figure 3.2 appear to yield the same aggregate throughput. This is because these protocols do not consider an increase in the back-

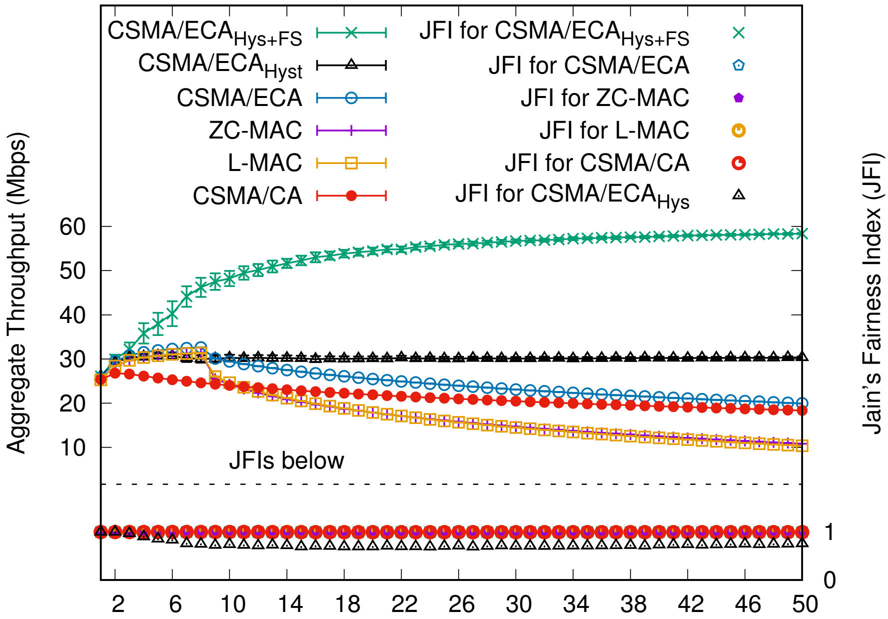

Figure 3.2: CSMA/ECA example in saturation: throughput

The *JFI* curves in Figure 3.2 show the Jain's Fairness Index [36] for all tested protocols. Showing a JFI equal to one suggests that the available throughput is shared evenly among all stations.

3.1 Supporting many more contenders

As was mentioned before, CSMA/ECA is only able to build a collision-free schedule if the number of contenders N, is less or equal to $B_{\mathrm{d}} + 1$. When $N > B_{\mathrm{d}} + 1$, collisions reappear.

To be able to attain a collision-free schedule even when the number

---

off stage after a failed transmission, augmenting the collision probability beyond CSMA/CA's. An adaptation of the schedule length leverages this issue, which is preseted in [19] as *Adaptive* L-MAC and L-ZC.

of contenders exceeds $B_\mathrm{d} + 1$, we introduce *Hysteresis*. Hysteresis is a property of the protocol that instructs nodes not to reset their backoff stage (k) after successful transmissions, but to use a deterministic backoff $B_\mathrm{d} = \lceil \mathrm{CW}(k)/2 \rceil - 1$, where $\mathrm{CW}(k) = 2^k\mathrm{CW}_\mathrm{min}$. This measure allows the adaptation of the schedule length, admitting many more contenders in a collision-free schedule. This idea of a schedule is significantly different from the virtual schedule required by the protocols described in Chapter 2.3, that is, CSMA/ECA with Hysteresis does not require a previous knowledge of the number of contenders, the result of previous transmissions or the start/end of the schedule, easing its implementation in real hardware.

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

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

The idea of allowing the transmission of more packets to stations that transmit less often was initially proposed by Fang et al. in [19]. It was later adapted to $\mathrm{CSMA/ECA_{Hys}}$ and named Fair Share in [60]. Figure 3.2 shows the JFI for CSMA/CA as well as for $\mathrm{CSMA/ECA_{Hys+FS}}$.

In Figure 3.2, the only curve deviating from JFI = 1 is $\mathrm{CSMA/ECA_{Hys}}$, suggesting an uneven partition of the channel access time among contenders (which is fixed with Fair Share).

Algorithm 3 describes $\mathrm{CSMA/ECA_{Hys+FS}}$, additionally Table 3.1 provides a short list of notations used throughout the text, while a short example of $\mathrm{CSMA/ECA_{Hys+FS}}$ with four contenders is shown in Figure 3.3. In the figure the first outline indicates a collision between STA-3 and STA-4, which will provoke an increment on both stations' backoff stage ($k \leftarrow k + 1$). Once STA-4's random backoff expires, $\mathrm{CSMA/ECA_{Hys+FS}}$

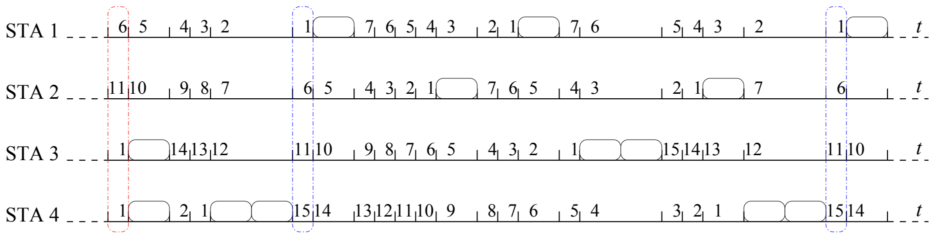

Figure 3.3: An example of the temporal evolution of CSMA/ECA$_{\text{Hys+FS}}$ in saturation

Table 3.1: CSMA/ECA Notation

Notation	Description
k	Backoff stage
m	Maximum backoff stage
B	Random backoff
B_{d}	Deterministic backoff
CW_{min}	Minimum Contention Window
CSMA/ECA$_{\text{Hys}}$	CSMA/ECA with Hysteresis
CSMA/ECA$_{\text{Hys+FS}}$	CSMA/ECA with Hysteresis and Fair Share
CSMA/ECA$_{\text{Hys+MaxAg}}$	CSMA/ECA with Hysteresis and Maximum Aggregation
CSMA/CA$_{\text{FS}}$	CSMA/CA with Fair Share
CSMA/ECA$_{\text{MaxAg}}$	CSMA/CA with Maximum Aggregation

instructs the station to transmit 2^k packets, and then use a deterministic backoff, $B_{\text{d}} = \lceil \text{CW}(k)/2 \rceil - 1$. The same procedure is followed by STA-3.

```
while the device is on do
    r ← 0 ; k ← 0 ; k_c ← k;
    b ← U[0, 2^k CW_min − 1];
    while there is a packet to transmit do
        repeat
            while B > 0 do
                wait 1 slot;
                B ← B − 1;
            Attempt transmission of 2^k packets;
            if collision then
                r ← r + 1;
                k ← min(k + 1, m);
                B ← U[0, 2^k CW_min − 1];
        until (r = R) or (success);
        r ← 0;
        if success then
            B_d ← ⌈2^k CW_min/2⌉ − 1;
            B ← B_d;
        else
            Discard 2^{k_c} packets;
            B ← U[0, 2^k CW_min − 1];
        k_c ← k;
    Wait until there is a packet to transmit;
```

Algorithm 3: 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

With Hysteresis and Fair Share, CSMA/ECA$_{\text{Hys+FS}}$ is able to achieve greater throughput than CSMA/CA and for many more contenders, as shown also in Figure 3.2. In the figure, the CSMA/ECA$_{\text{Hys+FS}}$ curve shows a greater throughput because collisions are eliminated and Fair Share allows nodes to send 2^k packets upon each transmission. This throughput increase is the result of aggregation via Fair Share. It carries the negative effect of raising the average time between successful transmissions, which may affect delay-sensitive traffic, like gaming or live video/voice/tv streaming.

Channel errors, hidden nodes, or unsaturated traffic conditions will disrupt any collision-free schedule, generate collisions and push all stations' backoff stages to the maximum value very quickly (contributing to the delay increase). This issue is leveraged with the Schedule Reset mechanism, introduced in Chapter 3.1.6.

3.1.1 The effects of aggregation

Fair Share is an A-MPDU aggregation mechanism [68] that coupled with the collision-free schedule built by CSMA/ECA$_{\text{Hys}}$ is able to provide short-term fairness. However, it also improves the throughput since the aggregation process makes the packet transmission more efficient by reducing overheads. Furthermore, besides the current backoff stage k, the level of aggregation provided by Fair Share depends also on the buffer occupancy.

The downside of Fair Share is that it may increase the time between two consecutive transmissions from the same node, which may affect negatively delay-sensitive applications such as gaming or high definition real-time video. An example of the duration of a transmission, $T(l_k)$, is provided by Equation (3.3) in the following Section 3.1.2.

In scenarios where short-term fairness and the time between consecutive transmissions are not relevant, Fair Share can be replaced by *Maximum Aggregation* (MaxAg), which will significantly improve the system throughput. In Maximum Aggregation all nodes aggregate as many packets as possible at every transmission, i.e., they send 2^m packets in each attempt.

Figure 3.4 shows the aggregate throughput for CSMA/CA using the Fair Share mechanism (CSMA/CA$_{\text{FS}}$), CSMA/CA$_{\text{MaxAg}}$, CSMA/ECA$_{\text{Hys+FS}}$, and CSMA/ECA$_{\text{Hys+MaxAg}}$. JFIs are shown in the bottom half of Figure 3.4. Although CSMA/CA$_{\text{FS}}$ and CSMA/CA$_{\text{MaxAg}}$ perform aggrega-

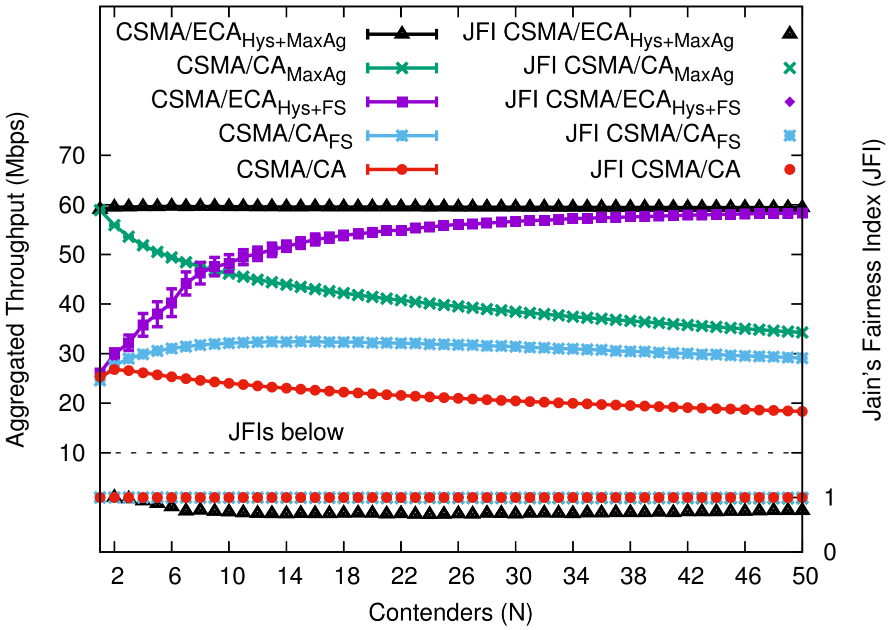

Figure 3.4: Throughput comparison with CSMA/CA_{MaxAg}

tion, collisions degrade the aggregate throughput as more contenders attempt transmission. On the other hand, CSMA/ECA_{Hys+FS} is able to build a collision-free schedule and takes advantage of the aggregation provided by Fair Share, which opposed to just using CSMA/$ECA_{Hys+MaxAg}$, it is fair.

To summarise the effects of using aggregation:

- It increases the aggregate throughput: because nodes are able to send multiple packets in each attempt, the system throughput is increased. Moreover, Fair Share compensates those nodes at higher backoff stages to ensure throughput fairness.

- Maximum aggregation supposes the deactivation of the Fair Share mechanism: performing maximum aggregation upon each transmission attempt is equivalent to having different schedule lengths and not compensating nodes at higher backoff stages. Although the aggregate throughput increases, this results in an uneven distribution of the channel time among contenders, which renders it unfair.

- Longer periods between transmission attempts: given that each transmission takes longer, the time between transmission attempts also increases. This may specially affect delay-sensitive applications.

Since we consider that fairness and a short inter-transmission time are even more important than raw throughput for the next generation of WLANs, we keep CSMA/ECA$_{\text{Hys+FS}}$ as the reference protocol.

3.1.2 Throughput bounds of CSMA/ECA$_{\text{Hys+FS}}$

They correspond to the maximum and minimum achievable throughput without the possibility of collisions using Hysteresis and Fair Share. The ideal network using CSMA/ECA$_{\text{Hys+FS}}$ employs the minimum schedule length that guarantees a collision-free operation. That is, with a schedule length of $C = 2^k B_{\text{d}} + 1$, where $k = \lceil log_2(N/(B_{\text{d}} + 1)) \rceil$. Using this minimum schedule length, N nodes will be at the same backoff stage if $N \leq B_{\text{d}} + 1$. Otherwise, $h = N - (C - N)$ nodes would occupy the k-th backoff stage and the other $N - h$ nodes the $(k-1)$-th one. The system throughput is computed as follows:

$$S = \begin{cases} hs_k(l_k) + (N-h)s_{k-1}(l_{k-1}), & \text{if } N > B_{\text{d}} + 1 \\ Ns_k(l_k), & \text{otherwise} \end{cases} \tag{3.1}$$

where $s_k(l_k)$ and $s_{k-1}(l_{k-1})$ are the throughput achieved by the nodes at the k-th and $(k-1)$-th backoff stages sending l_k and l_{k-1} packets respectively. These are given by:

$$s_k(l_k) = \frac{l_k L}{hT(l_k) + 2(N-h)T(l_{k-1}) + \sigma_e(C-N)} \tag{3.2a}$$

$$s_{k-1}(l_{k-1}) = \frac{l_{k-1} L}{(N-h)T(l_{k-1}) + k\frac{T(l_k)}{2}} \tag{3.2b}$$

where L is the data payload, $T(l_k)$ and $T(l_{k-1})$ are the duration of the transmission of l_k and l_{k-1} packets, respectively; σ_e is the duration of an empty slot. Additionally, $T(l_k)$ derives from Equation (3.3):

$$T(l_k) = \left(T_{\text{PHY}} + \left\lceil \frac{\text{SF} + l_k(\text{MD} + \text{MH} + L) + \text{TB}}{L_{\text{DBPS}}} \right\rceil T_{\text{sym}}\right) + \text{SIFS} + \left(T_{\text{PHY}} + \left\lceil \frac{\text{SF} + L_{\text{BA}} + \text{TB}}{L_{\text{DBPS}}} \right\rceil T_{\text{sym}}\right) + \text{DIFS} + \sigma_e \quad (3.3)$$

where $T_{\text{PHY}} = 32\ \mu$s is the duration of the PHY-layer preamble and headers, $T_{\text{sym}} = 4\ \mu$s is the duration of an OFDM (Orthogonal Frequency Division Multiplexing) symbol. SF is the *Service Field* (16 bits), MD is the *MPDU Delimiter* (32 bits), MH is the *MAC header* (288 bits), TB is the number of *Tail Bits* (6 bits), L_{BA} is the *Block-ACK* length (256 bits) and $L_{\text{DBPS}} = 256$ is the number of bits in each OFDM symbol.

The *Lower-bound* is then derived from considering the operation of an ideal CSMA/$\text{ECA}_{\text{Hys+FS}}$ network. Nodes use the minimum backoff stage possible and aggregate proportionally, thus yielding the minimum throughput achievable by a CSMA/$\text{ECA}_{\text{Hys+FS}}$ network. It is computed following Equation (3.1) with $l_k = 2^k$ and $l_{k-1} = 2^{k-1}$.

The *Upper-bound* is obtained from considering the operation of a network using CSMA/$\text{ECA}_{\text{Hys+FS}}$, but forcing nodes to use the maximum backoff stage for determining the cycle length and the level of aggregation. It is also computed using Equation (3.1) but considering that all nodes are in the maximum backoff stage ($k = m$) and therefore $l_k = 2^m$.

The maximum throughput achievable is the result of deactivating the Fair Share rules by forcing nodes to use maximum aggregation regardless of their backoff stage. This is called *Maximum Aggregation (Hys+MaxAg)* in Figure 3.5. It can be derived from Equation (3.1) considering $l_k = 2^m$ and $l_{k-1} = 2^m$.

It is interesting to see in Figure 3.5 how collisions force colliding contenders using CSMA/$\text{ECA}_{\text{Hys+FS}}$ to increase their backoff stage and aggregate more with Fair Share. This explains why the CSMA/$\text{ECA}_{\text{Hys+FS}}$

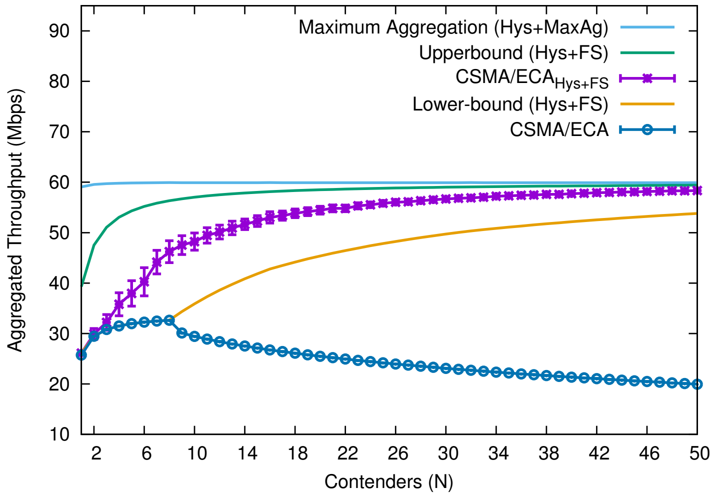

Figure 3.5: Upper and Lower throughput bounds for CSMA/ECA$_{\text{Hys+FS}}$

curve separates itself from the *Lower-bound* at a very low number of contenders.

Although using maximum aggregation in Figure 3.5 increases the throughput, it carries the effect of unevenly distributing the available bandwidth among contenders, as mentioned in Chapter 3.1.1.

The tools for deriving these bounds are available as MATLAB functions in [62].

3.1.3 Clock drift issue in descentralized collision-free MAC protocols

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

MAC protocols like the ones reviewed in Chapter 2.3.

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

3.1.4 Channel errors

Failed transmissions due to channel errors are handled as collisions. Therefore, collision-free periods under this type of channel model are frequently interrupted. In order to accelerate the convergence to collision-free schedules in presence of channel errors $\text{CSMA/ECA}_{\text{Hys+FS}}$ instructs nodes to *stick* to the current deterministic backoff even after *stickiness* number of consecutive failed transmissions. Stickiness was introduced to CSMA/ECA in [8], where it is described and evaluated. It allows for a faster convergence towards a collision-free schedule, especially when performing under heavy channel errors.

Failed transmissions due to channel errors also means that a few moments of operation under a noisy channel can increase the contenders' deterministic backoff to its maximum value. Such event carries the undesired effects of increasing the time between successful transmissions and reducing the overall system throughput due to fewer collision-free periods of operation.

3.1.5 Hidden terminals

One key characteristic of IEEE 802.11 devices is that their carrier sense range is at least two times greater than their data range [18]. In this situation, the impact of hidden nodes can be considered to be low. This is because a given transmission could only be interfered by other transmissions from very distant nodes with energy levels not higher than the noise floor, which using modern radios can be easily solved by the capture effect. However, in specific deployments, where obstacles also play an important role on the propagation effects, hidden nodes may appear, causing asynchronous packet collisions [38].

In the case of CSMA/ECA$_{\text{Hys+FS}}$ WLANs, the same negative effects as in IEEE 802.11 WLANs are expected. In addition, collisions with hidden terminals will cause nodes to leave any collision-free schedule, as channel errors do, thus continuously disrupting any attempt of collision-free operation.

3.1.6 Schedule Reset mechanism

CSMA/ECA$_{\text{Hys+FS}}$ instructs nodes to keep their current backoff stage after a successful transmission (resetting it to zero only if the node empties its MAC queue, see Line 2 in Algorithm 3). 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.

Nevertheless, having a big deterministic backoff increases the time between successful transmissions. If not operating in a scenario with many nodes, the empty slots between transmissions are no longer negligible and degrade the overall throughput of the system. For instance, if nodes withdraw from the contention (i.e.: empty their MAC queue, or move to other network) their previously used slots will now be empty. On the other hand, in scenarios with channel errors contenders rapidly end up with the largest deterministic backoff, remaining there until the MAC queue empties. Nodes should be aware of this issue and pursue opportunities to reduce their deterministic backoff, B_{d} without sacrificing too much in collisions.

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

Figure 3.6 shows the slots between the transmissions of a contender

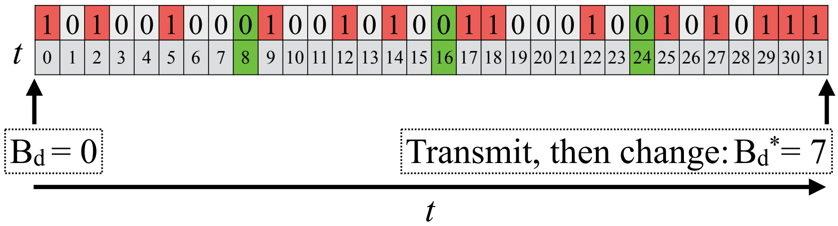

Figure 3.6: Example of the Schedule Reset mechanism

with $B_{\text{d}} = 31$ and $\text{CW}_{\text{min}} = 16$[2]. Starting from the left, the current $B_{\text{d}} = 0$ means that the first 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 $B_{\text{d}} = 31$ to $B^*_{\text{d}} = 7$ is possible; where B^*_{d} is the new deterministic backoff assigned by Schedule Reset[3]. The schedule change is performed after the contender's next successful transmission.

A conservative schedule reduction

Schedule Reset is described in Algorithm 4. Each contender starts filling a bitmap b of size $B_{\text{d}} + 1$ with the information observed in each passing slot during $\gamma = \lceil C/B_{\text{d}} \rceil$ consecutive successful transmissions (sxTx in Algorithm 4); where $C = \lceil 2^m \text{CW}_{\text{min}}/2 \rceil - 1$ is the biggest deterministic backoff, and γ is referred to as the Schedule Reset *threshold* from this point forward.

[2] The lower row shows values of t for a bitmap $b[t]$ of size $|b[t]| = B_{\text{d}} + 1 = 32$.

[3] $B_{\text{d}} = 2^0 CW_{\text{min}}/2 - 1 = 7$. So the change is made to the backoff stage, k. In this case from 2, to $k = 0$. This means that any new schedule must also be a power of two minus one.

```
1  if sxTx == 1 then
2      Initialising;
3      if B == 0 then
4          b[B_d + 1] = {0};
5          t = 0;
6  if (sxTx > 0) && (sxTx ≤ γ) then
7      Filling the bitmap;
8      b[t]||σ_i(t);
9      t++;
10     if t == B_d then
11         t = 0;
12 if sxTx == γ then
13     Analising;
14     sxTx = 0;
15     for (j = 0; j < k; j++) do
16         y = ⌈2^j CW_min/2⌉;
17         isItPossible = 0;
18         for (m = y; m < B_d + 1; m+ = y) do
19             isItPossible + = b[m];
20         if isItPossible == 0 then
21             change = 1;
22             newStage = m;
23             break;
24 if change == 1 then
25     Making the change;
26     k = newStage;
27     change = 0;
```

Algorithm 4: Schedule Reset Mechanism for CSMA/ECA$_{\text{Hys+FS}}$. Every consecutive successful transmission increases the variable SxTx by one, while a collision resets it to zero. The algorithm is called a *reset* when all smaller deterministic backoff are tested. On the other hand, when j in Line 15 is initialised to $j = k - 1$, it is called a schedule *halving*

Each bit t, $t \in \{0, \ldots, B_{\text{d}}\}$ in the bitmap is the result of a bitwise OR operation between its current value and the state of the corresponding slot. This is shown in Line 8 in Algorithm 4, where $\sigma_i(t)$ is a function that evaluates to 0 if the slot corresponding to $b[t]$ is empty in iteration i, $i \in \{1, \ldots, \gamma\}$; or 1 otherwise. After γ iterations, the bitmap is evaluated (Line 12 in Algorithm 4).

Schedule Reset tests all possible deterministic backoffs that are smaller than the current B_{d} (Lines 15-19 in Algorithm 4), starting with the smallest one. If the corresponding bits in the bitmap are registered as empty, the process is stopped and the change to the new deterministic backoff is made (Lines 20-27 in Algorithm 4). Otherwise, the process is restarted after the next successful transmission.

Using Figure 3.6 as an example, we can see that $\sum_t b[t] = 0$, for $t \in \{8, 16, 24\}$. This means that the transmission slots corresponding to a deterministic backoff, $B_{\text{d}}^* = 7$ are free, therefore a change of schedule is possible. In case of suffering a collision immediately after applying the schedule reduction, the node reverses the changes made by Schedule Reset before handling the collision.

Aggressive approach to face channel errors

The default γ ensures that the bitmap registers all the transmission slots in the network (assuming saturated traffic and a perfect channel), providing enough information for performing the schedule reduction without disrupting collision-free schedules.

Nevertheless, this γ can be rendered too conservative and unnecessary because nodes may randomly leave the collision-free schedule due to channel errors, or when their queues are emptied. Therefore, this condition produces Schedule Reset bitmaps that do not contain enough information for successfully avoiding collisions after the schedule reduction. This is also true for any $\gamma^* > \gamma$.

This effect is leveraged by setting $\gamma = 1$, meaning that the bitmap is filled with the information of slots between only two consecutive transmissions. This measure increases the frequency of schedule reset attempts. Furthermore, incrementing the node's stickiness after a successful reduction of the schedule allows for a faster convergence towards a collision-free schedule, especially when performing under heavy channel errors.

3.1.7 Backwards compatibilty and coexistence

CSMA/ECA$_{\text{Hys+FS}}$ springs from a modification to CSMA/CA's backoff mechanism. It uses CSMA/CA's contention parameters (i.e., use the same $\text{CW}_{\min}$ and $\text{CW}_{\max}$), allowing CSMA/ECA$_{\text{Hys+FS}}$ contenders to coexist with CSMA/CA nodes in the same network. Further, the selection of CSMA/ECA$_{\text{Hys+FS}}$'s deterministic backoff, B_{d}, is the expected value for the current backoff stage k ($B_{\text{d}} := \lceil E[0, CW(k) - 1] \rceil$) [60], which ensures fairness among contenders.

3.2 Simulation setup description

This section provides a description of the simulation parameters for testing CSMA/ECA$_{\text{Hys+FS}}$ under two different traffic conditions, namely saturated and non-saturated. We also provide details on how channel errors are modelled and what are its effects over the transmissions. Further, the simulation of the clock drift effect, Schedule Reset parameters, and the coexistence with CSMA/CA are also subjects to be addressed in this section.

3.2.1 Scenario details

Results are obtained by executing a simulator developed from scratch using the COST simulation libraries [76], available at [54]. PHY and MAC parameters are detailed in Table 3.2. Some assumptions were made in order to test the performance at the MAC layer:

- Unspecified parameters follow the IEEE 802.11n (2.4 GHz) amendment.

Table 3.2: 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 backoff stage (m)	5
Minimum Contention Window ($\mathrm{CW}_{\min}$)	16
Maximum retransmission attempts	6
Data payload (Bytes)	1024
MAC queue size (Packets)	1000

- All nodes are in communication range.
- No Request-to-Send (RTS) or Clear-to-Send (CTS) messages are used.
- Collisions take as much channel time as successful transmissions.

The aforementioned assumptions ensure that the simulation results are just effects of the MAC behaviour. If not mentioned otherwise, results are derived from 20 simulations of 100 seconds in length, each one with a different seed. Figures also show the standard deviation.

3.2.2 Saturated and Non-saturated stations

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

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

3.2.3 Performance under clock drift

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

3.2.4 Channel errors

Channel errors are modelled by assigning a probability of a packet being corrupted by the channel, $p_e > 0$. That is, in a single packet transmission there is probability p_e that the transmission will not be acknowledged. If the transmission is an A-MPDU (like in the case of CSMA/ECA$_{\text{Hys+FS}}$), p_e will affect each MPDU individually and independently. Therefore, an A-MPDU transmission will be considered a complete failure only if all frames in the aggregation are affected by the channel error probability.

All results are shown with stickiness equal to one. That is, after a collision, CSMA/ECA$_{\text{Hys+FS}}$ nodes will use a random backoff. Nevertheless, in the following section curves described as *dynStick* correspond to contenders using an aggressive configuration of Schedule Reset with dynamic stickiness, temporarily increasing the stickiness value to two after a successful reduction of the schedule[4]. This increase is done in order to converge faster to collision-free schedules when operating with channel errors.

[4]Therefore using a random backoff after two consecutive collisions.

3.2.5 Coexistence with CSMA/CA

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

3.2.6 Applying Schedule Reset

A set of results under saturated conditions and channel errors applying Schedule Reset are provided. Some of the results are generated with a $\gamma = 1$, or *aggressive* Schedule Reset (indicated as *aggr.* in the figures). These settings provide the highest throughput under the tested conditions, and also in the real hardware implementations such as the one shown in Chapter 6.

3.3 Simulation results

3.3.1 Saturated nodes

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

The following overview the throughput in saturation of CSMA/CA and CSMA/ECA$_{\text{Hys+FS}}$, as well as the collision probability, the average time between successful transmissions, and the effect of clock drift over the throughput.

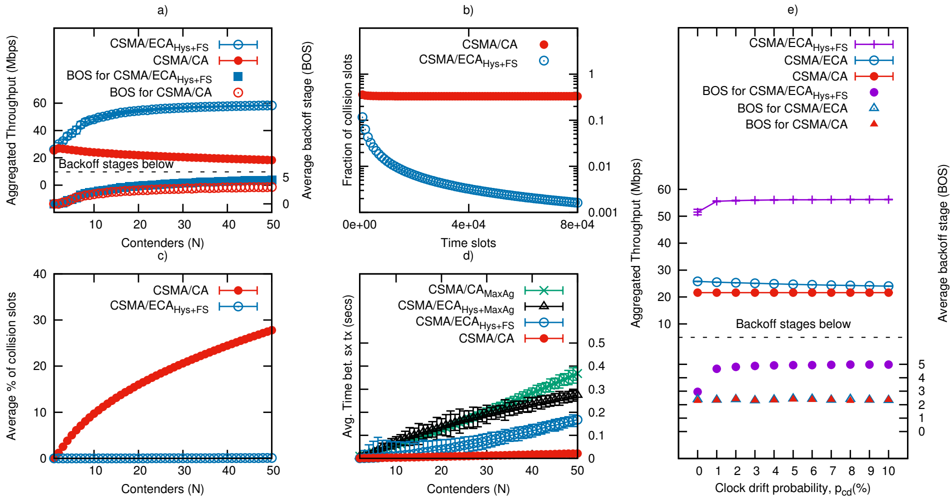

Figure 3.7: Simulation results under saturated traffic

Throughput

CSMA/ECA$_{\text{Hys+FS}}$ nodes are able to build a collision-free schedule, use the channel more efficiently, and experience a throughput increase. Figure 3.7 shows: a) Throughput under saturated conditions; b) Evolution of the fraction of collision slots in a scenario with 50 saturated stations; c) Average percentage of collision slots: fraction of time slots containing collisions; d) Average time between successful transmissions (sx tx), e) Throughput when increasing the clock drift probability.

In Figure 3.7a, hysteresis allows the allocation of more contenders by increasing the length of the collision-free schedule, while Fair Share ensures an even distribution of the available throughput. This is reflected by the average backoff stage, which value increases with the number of contenders. In contrast, CSMA/CA throughput keeps decreasing due to an augmented number of collisions as the number of nodes increases (see Figure 3.7c). Further, Figure 3.7b shows the fraction of collision slots for CSMA/ECA$_{\text{Hys+FS}}$ and CSMA/CA as simulation time passes. In the figure, it is possible to see how the fraction of collision slots keeps decreasing once CSMA/ECA$_{\text{Hys+FS}}$ reaches collision-free operation.

Effect of clock drift over the achieved throughput in saturation

Figure 3.7e shows the network aggregate throughput with 16 saturated stations and an increasing clock drift probability.

In Figure 3.7e, a station has a clock drift probability equal to p_{cd}. Each station has a probability of $p_{cd}/2$ of miscounting one slot more, and $p_{cd}/2$ of miscounting one slot less. Because CSMA/CA is based on a random backoff, miscounting slots has no significant effect on the throughput. For the CSMA/ECA curve, it is possible to observe a slight decrease of the throughput as p_{cd} increases, caused by collisions due to the drift.

The CSMA/ECA$_{\text{Hys+FS}}$ curve in Figure 3.7e shows an increase of the aggregate throughput as p_{cd} grows. Collisions make CSMA/ECA$_{\text{Hys+FS}}$ contenders to increment their backoff stage and aggregate packets for transmissions according to Fair Share, effectively increasing the throughput.

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

Time between successful transmissions

It is related to the elapsed time between the contention for transmission and the reception of an ACK.

In Figure 3.7d, we see an increased average time between successful transmissions in all tests with maximum aggregation, like, CSMA/CA$_{\text{MaxAg}}$ and CSMA/ECA$_{\text{Hys+MaxAg}}$. This is due to the multiple packets that are sent in each attempt. CSMA/CA$_{\text{MaxAg}}$, though, has an increased value due to collisions also taking longer channel time.

Although CSMA/ECA$_{\text{Hys+FS}}$ periods between successful transmissions are large due to Fair Share, it has a lower metric when compared with the maximum aggregation curves in Figure 3.7d.

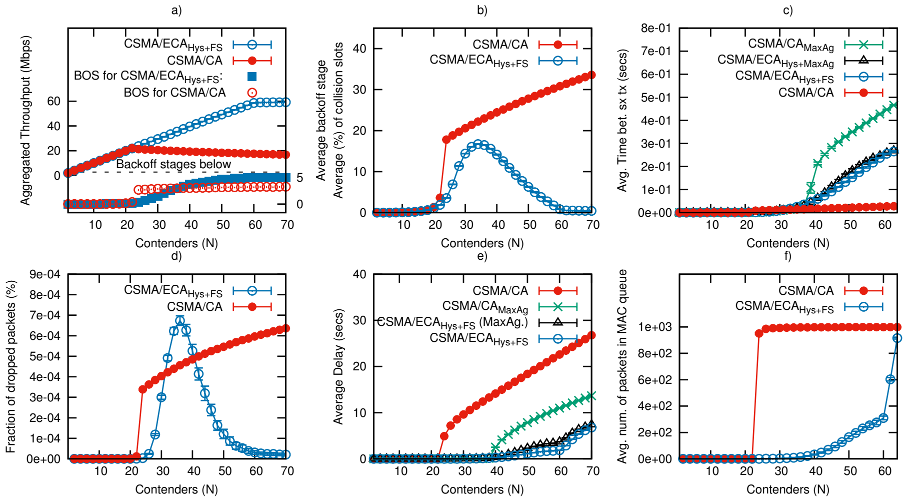

Figure 3.8: Simulation results under non-saturated traffic

3.3.2 Non-saturated nodes

Emptying the MAC queue in CSMA/ECA$_{\text{Hys+FS}}$ means that nodes will reset their backoff stage to zero and use a random backoff when a new packet arrives at the queue, breaking any collision-free operation. The following shows the impact over throughput, delay and time between successful transmissions when using CSMA/CA and CSMA/ECA$_{\text{Hys+FS}}$ in non-saturated conditions.

Throughput

Figure 3.8 shows simulation results under non-saturated traffic: a) Throughput; b) Average percentage of collision slots: the fraction of time slots containing collisions; c) Average time between successful transmissions; d) Average fraction of dropped packets; e) Average system delay; f) Average number of packets in the MAC queue of a node.

In Figure 3.8a, the aggregate throughput increases linearly for the

CSMA/CA curve until saturation is reached at around 22 nodes, where the throughput begins to degrade. The CSMA/ECA$_{\text{Hys+FS}}$ curve has a similar behaviour, entering saturation at around 60 nodes. Further, at around 30 nodes we see an increase in the average backoff stage for CSMA/ECA$_{\text{Hys+FS}}$ contenders which suggests an increment in collisions. This effect is shown in Figure 3.8b and Figure 3.8d, where at around 35 nodes CSMA/ECA$_{\text{Hys+FS}}$ contenders start colliding and dropping packets.

As indicated by Figure 3.8b, when $20 < N \leq 35$ CSMA/ECA$_{\text{Hys+FS}}$ nodes suffer from an increasing number of collisions. This is due to nodes emptying their MAC queue quicker due to Fair Share, as shown in Figure 3.8f and re-entering the contention with a random backoff every time a new packet arrives.

This increase in collisions may also cause the dropping of packets due to reaching the maximum retransmission limit. As CSMA/ECA$_{\text{Hys+FS}}$ drops more packets after reaching such limit (see Line 20 in Algorithm 3), it shows a higher fraction of dropped packets in Figure 3.8d.

Beyond 35 contenders, the MAC queue of CSMA/ECA$_{\text{Hys+FS}}$ nodes starts to fill up, gradually allowing longer periods of collision-free operation due to CSMA/ECA$_{\text{Hys+FS}}$ nodes getting saturated. This allows CSMA/ECA$_{\text{Hys+FS}}$ to outperform CSMA/CA.

Delay

This metric refers to the elapsed time between the injection of a packet into the station's MAC queue and the reception of an ACK for such packet.

In Figure 3.8e, a rapid increase in the delay for CSMA/CA nodes is observed at the saturation point (around 20 contenders), whereas delay is still low for CSMA/ECA$_{\text{Hys+FS}}$.

Further, with CSMA/ECA$_{\text{Hys+FS}}$ the percentage of blocked packets from the MAC queue[5] is lower than CSMA/CA or CSMA/CA$_{\text{MaxAg}}$ (see

[5]When the MAC queue is full, packets comming from the uppers are discarded (or blocked) at the entrance of the buffer.

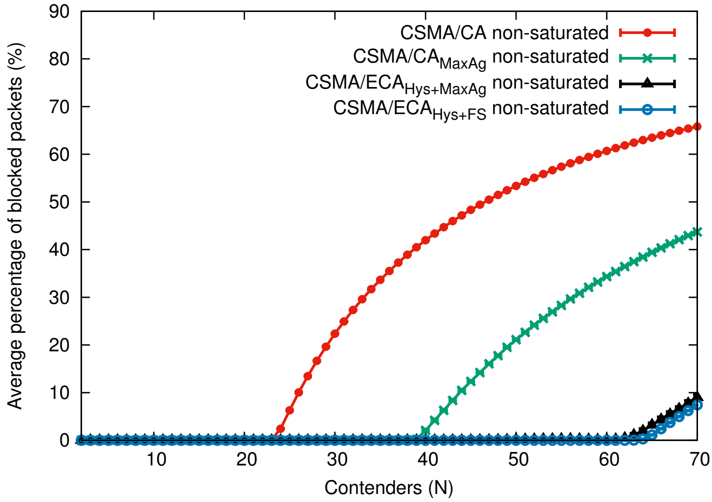

Figure 3.9: Average fraction of blocked packets

Figure 3.9). This is due to the construction of collision-free schedules which ensure that large A-MPDU transmissions do not suffer from collisions.

As CSMA/ECA$_{Hys+FS}$ nodes get saturated, the delay increases due to longer queueing and contention time (see the number of packets in the MAC queue for CSMA/ECA$_{Hys+FS}$ nodes in Figure 3.8f and how it is related to the increase in delay shown in Figure 3.8e).

Figure 3.8c shows the average time between successful transmissions. It is possible to see from the figure that when CSMA/ECA$_{Hys+FS}$ approaches the saturation point the average time between successful transmissions increases, resembling Figure 3.7d.

Chapter 4

PART II: COEXISTENCE AND OVERLAPPING BASIC SERVICE SETS

CSMA/ECA is thought to be an evolution of CSMA/CA given its similarities and the ability to coexist with the latter. This chapter provides simulations results for a setup of different proportions of CSMA/CA nodes in a network where there are also CSMA/ECA$_{\text{Hys+FS}}$ contenders, that is: 1/4, 1/2 and 3/4 of the total nodes run CSMA/CA, while the rest uses CSMA/ECA$_{\text{Hys+FS}}$. This network configuration will be referred to as *mixed network setup* from here on. Later, we provide simulation results for different configurations of CSMA/ECA under Overlapping Basic Service Sets (OBSS) scenarios, including the typical residential building scenario proposed by the IEEE 802.11ax High Efficiency WLANs (HEW) Task Group (TGax) [31, 32].

4.1 Getting along with the old

Figure 4.1 gathers the coexistence scenario results, showing: a) Network throughput when composed by various proportions of CSMA/CA and CSMA/ECA$_{\text{Hys+FS}}$ nodes in saturation; b) Average percentage of collision

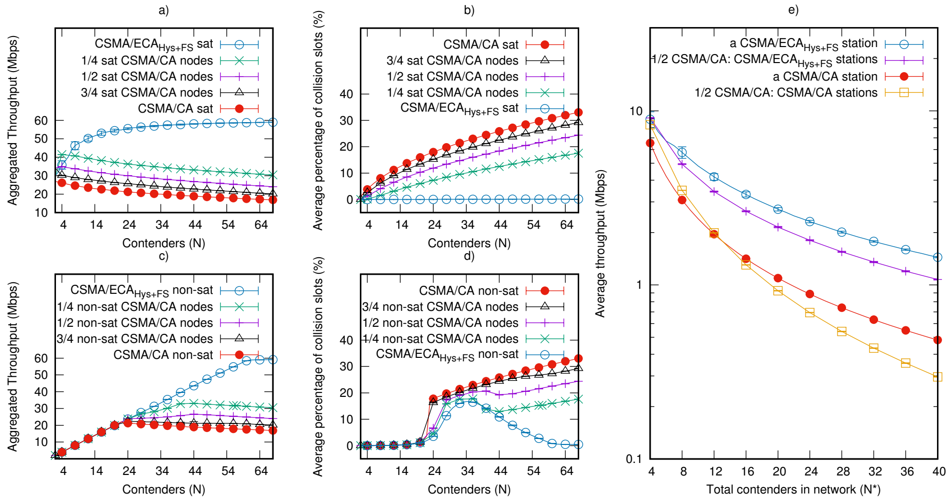

Figure 4.1: Coexistence results

slots for the tested saturated mixed network setups proportions; c) Network throughput when composed by various proportions of non-saturated CSMA/CA and CSMA/ECA$_{\text{Hys+FS}}$ nodes; d) Average percentage of collision slots for the tested mixed network setups proportions in non-saturated conditions; e) Average station throughput per MAC protocol in a saturated mixed network

Figure 4.1a shows the aggregate network throughput for different proportions of CSMA/CA nodes in a mixed network setup. In the figure it is observed how the mixed network setup curves lay between the CSMA/CA and CSMA/ECA$_{\text{Hys+FS}}$ curves. As the proportion of CSMA/CA nodes decreases, the throughput increases as the result of a lower probability of collision, as can be seen in Figure 4.1b. A similar behaviour is observed when testing the same proportion of nodes under non-saturated conditions. Figure 4.1c and Figure 4.1d show the average aggregate throughput and fraction of collisions slots in a non-saturated mixed network setup.

As shown in Figure 4.1a and Figure 4.1c, at a lower proportion of CSMA/CA nodes $(1/4)$ the average aggregate throughput is the greatest among the tested mixed network setups. This is because collisions trigger Hysteresis and Fair Share in CSMA/ECA$_{\text{Hys+FS}}$ nodes, lowering the number of times these nodes enter in a contention and reducing the overall collision probability when compared to an only CSMA/CA network (see Figure 4.1b, Figure 4.1d and Figure 4.1e). As the proportion of CSMA/CA nodes increases, the network throughput approximates to CSMA/CA.

Finally, Figure 4.1e shows the average throughput for both kind of stations in a saturated mixed setup (half the contenders using CSMA/CA). It is possible to see that for a low total number contenders ($N^* \leq 12$) CSMA/CA stations attain greater throughput than in a CSMA/CA-only network. Again, this is because in the mixed network setup the other $N^*/2$ contenders with CSMA/ECA$_{\text{Hys+FS}}$ use a deterministic backoff, leaving many empty slots between transmissions.

Still referring to Figure 4.1e, for $N^* > 12$ periods of collision-free operation and Fair Share aggregation allows CSMA/ECA$_{\text{Hys+FS}}$ to allocate the channel for longer periods than CSMA/CA nodes, which throughput degrades even more than in CSMA/CA-only. These results suggest that a switch to CSMA/ECA$_{\text{Hys+FS}}$ can be beneficial for networks with high number of contenders.

4.2 Channel errors and Schedule Reset

Figure 4.2 gathers the simulation results using the Schedule Reset Mechanism (see Chapter 3.1.6). Curves called "aggr" refer to values of $\gamma = 1$, while "halv" refers to a schedule halving (see Algorithm 4). *dynStick* refers to curves where the node's stickiness is temporarily increased to two after a successful reduction of the schedule. It shows: a) Average throughput for a saturated network; b) Average throughput for a non-saturated network; c) Time between successful transmissions for a saturated network; d) Time between successful transmissions for a non-saturated network.

Figure 4.2a shows the average aggregate throughput of a saturated network with a channel error probability, $p_e = 0.1$. We can see that

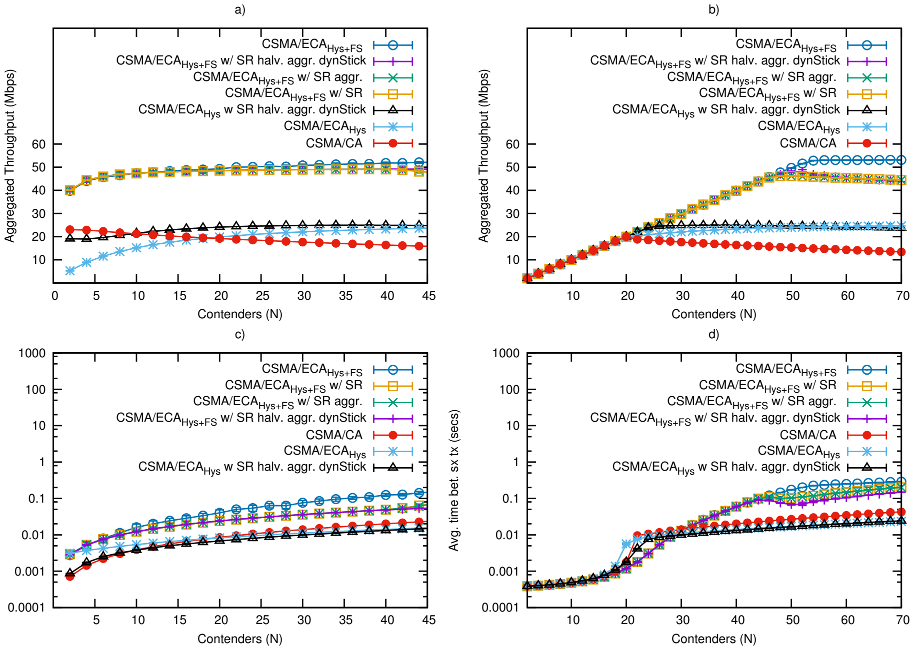

Figure 4.2: Performance results with Schedule Reset

the highest throughput, corresponding to CSMA/ECA$_{Hys+FS}$ is lower than the one observed in a perfect channel (Figure 3.7a). Furthermore, curves with Schedule Reset (SR) seem to have lower aggregate throughput. This is because the backoff stage, and therefore the level of aggregation done by Fair Share, is repeatedly reduced.

Still assessing Figure 4.2a and focusing on the SR curves, we can see that both aggressive schedule halving with dynamic stickiness (*SR aggr. halv. dynStick*) curves (with and without Fair Share) have higher aggregate throughput than the rest. This Schedule Reset configuration reduces the time between successful transmissions (see Figure 4.2c) and maintains collision-free operation for longer periods of time thanks to the increase in the node's stickiness.

Figure 4.2c shows the average time between successful transmissions for the same network setup. Even-though CSMA/ECA$_{\text{Hys+FS}}$ still has a higher metric than CSMA/CA due to the aggregation done by Fair Share, Schedule Reset is able of reducing this metric by almost 43%.

Figure 4.2b and Figure 4.2d show the aggregate throughput and time between successful transmissions, respectively for a non-saturated network with the same $p_e = 0.1$. Since under non-saturated conditions CSMA/ECA$_{\text{Hys+FS}}$ nodes reset their schedule every time they empty their MAC queues, Schedule Reset's benefits are not relevant. Nevertheless, a reduction in the time between successful transmissions is observed for Schedule Reset curves in Figure 4.2d.

4.3 Very crowded environments

First, we do a performance evaluation of CSMA/CA, CSMA/ECA and CSMA/ECA$_{\text{Hyst+SR}}$ in a single AP scenario using NS-3 for the first time (the implementation of CSMA/ECA is described in Section 4.4 below). Then, to test the effects of neighbouring WLANs, we define three different scenarios:

- **Scenario A:** a linear array of A number of APs, with N nodes forming a circle around each AP $i \in [1, \cdots, A]$. APs are separated by Δ_x metres, and each node $j \in [0, \cdots, 2\pi]$ associated with AP i, that is, node $n_{i,j}$, is at δ meters from i. Neighbouring nodes[1] are separated by δ_n. Each node has a Communication Range C$_{\text{R}}$, and an Interference Range I$_{\text{R}}$. Transmissions from nodes within I$_{\text{R}}$ will trigger the carrier sense mechanism, freezing the backoff counter. Nevertheless, only received frames from nodes within C$_{\text{R}}$ are effectively decoded. Figure 4.3 shows and example Scenario A.
  We also test a *control* Scenario A, with no losses suffered inside I$_{\text{R}}$ $=$ C$_{\text{R}}$ $= 2\delta$. Otherwise, Scenario A uses a Log-distance propagation loss model with loss-exponent (details provided below), as

[1]In the Scenario A case with $N = 4$ nodes, only $n_{i,0}$ and $n_{k,\pi}$ are considered neighbouring nodes, where $|i-k| = 1;\ (i,k) \in [1, \ldots, A]$.

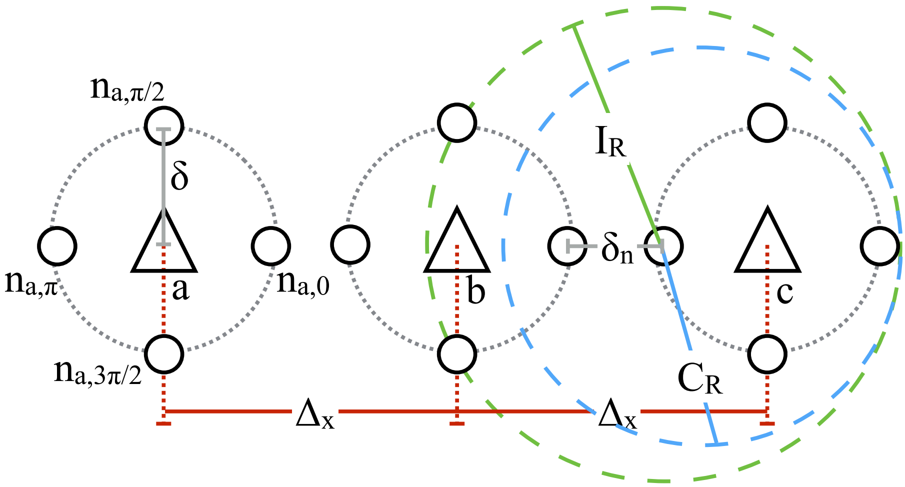

Figure 4.3: Scenario A example

proposed by IEEE 802.11ax High Efficiency WLAN (HEW) Task Group (TGax) [31].

- **Scenario B:** AP $i \in [1, \cdots, A]$ is arranged as in Scenario A, but each node $j \in [0, \cdots, N]$ is randomly placed at $p_{ij}(x, y)$, where $x, y \in [-\delta, \delta]$ are plane[2] coordinates, which is centered in the AP i.

- **Scenario HEW:** follows the simulation scenario 1 suggested by TGax [31], also called the residential building scenario. Figure 2.2 provides details regarding the dimensions and placement of APs.

Regarding Scenario HEW, the TGax proposes different criteria for evaluating IEEE 802.11ax WLAN. Scenario HEW follows TGax's guidelines in the sense that:

- $L = 10$ and $F = 3$ are the side of a square room $q \in [1, \cdots, A]$ and its height in meters, respectively (see Figure 2.2).

[2]It is possible to simulate 3D scenarios by adding an additional coordinate.

- The AP $i \in [1, \cdots, A]$ is randomly placed inside room q at a fixed heigh of $z = 1.5$m.

- Nodes also are randomly placed inside room q at a fixed heigh of $z = 1.5$m, and associated with AP i. We identify a node $j \in [0, \ldots, N-1]$ as n_{ij}. $N = 10$ per AP.

- Walls and floors impose propagation losses to the signal. Our model uses the same propagation loss model and loss-exponents. (4.1) (based on the one proposed in [31]) shows the path loss $p_l(x)$ (dB), where x is the known distance to the transmitter, f_c is the operating frequency (see Table 4.1), $(x > 5)$ evaluates to 1 when the condition is met or returns 0 otherwise. Finally, W represent the aggregate number of walls traversed to reach the receiver, while Z is the number of floors traversed until the signal arrives at the receiver. All the multi-AP scenarios use this same propagation loss model (only Scenario HEW considers a building, Z and W in (4.1) are set to zero otherwise).

$$p_l(x) = 40.05 + 20\ \mathrm{Log}_{10}\left(\frac{f_c}{5(10^9)}\right) + 20\ \mathrm{Log}_{10}(\min(x, 5)) + (x > 5)35\ \mathrm{Log}_{10}\left(\frac{x}{5}\right) + 17Z + 12W \quad (4.1)$$

Test are performed using saturated sources at each node, keeping the MAC queue filled at all times. This means that nodes always have a packet to transmit. Additionally, simulations follow MAC and PHY specifications from the IEEE 802.11n standard, using a 20 MHz channel in the 5 GHz band. The rate of the stations is fixed. Details about the CCA and Energy Detection (ED) thresholds, as well as other MAC and PHY details are shown in Table 4.1.

Table 4.1: PHY, MAC, CCA, and ED parameters used in the simulations

PHY	
Parameter	**Value**
PHY rate	72.2 Mbps
MCS	HtMcs7
Channel Width	20 MHz
Operating Frequency f_c	5.24 GHz
Channel Number	48
Empty slot	9 μs
DIFS	34 μs
SIFS	16 μs
MAC	
$\mathrm{CW}_{\mathrm{min}}$	16
$\mathrm{CW}_{\mathrm{max}}$	1024
Maximum retransmission attempts	7
Default Packet size (Bytes)	1470
Channel and Tx/Rx properties	
cca1Threshold (CCA)	-62 dBm
edThreshold (ED)	-82 dBm
Tx power	15 dBm
Per wall losses	12 dB
Per floor losses	17 dB

4.4 Leveraging NS-3 to simulate reality

The backoff mechanism controlled by the `EdcaTxopN` class is modified to react differently to the effective reception of an ACK. That is, instead of following CSMA/CA backoff mechanism, nodes are reconfigured to follow CSMA/ECA$_{\text{Hyst+SR}}$ upon the call to the `EdcaTxopN::GotAck` method[3].

The Schedule Reset mechanism required simple modifications to the

[3]Which happens everytime a successful transmission is acknoledged by the receiver.

`DcfManager` class. It creates a bitmap according to the current deterministic backoff, and updates it following the channel conditions at each decrementing slot.

Position of the nodes

The scenarios defined in Section 4.3 are implemented using different mobility and propagation loss models provided by NS-3. Specifically, each node's position is fixed during simulation time, and the signal is attenuated according to (4.1) with the help of piggybacked Log-Distance propagation models [67, 70].

The design and configuration of a building is made simple by the `Building` class in NS-3 [49]. It provides several sub-classes and methods for specifying size, materials and attenuation properties using the `HybridBuildingsPropagationLossModel` class on top of the `ThreeLogDistancePropagationLossModel`.

Our implementation was made using the NS-3 [52] network simulator, and can be accessed via [65]. A tutorial on how to use CSMA/ECA MAC for WiFi in NS-3 is provided by [64].

We proceed to do a series of performance evaluations using the aforementioned scenarios, modifying its characteristics in order to understand the behaviour of each MAC protocol under different conditions. If not specified otherwise, results are derived from five iterations of a twenty five second NS-3 simulation with different seeds.

4.4.1 Single AP

Figure 4.4 shows: a) average aggregate throughput and b) average fraction of failed transmissions under saturated traffic conditions. The scenario of this test supposes perfect communication among all nodes. Results for DCF, CSMA/ECA, and an implementation of CSMA/ECA with Hysteresis and conservative Schedule Reset with dynamic stickiness, namely CSMA/ECA$_{\text{Hyst+SR}}$, are presented.

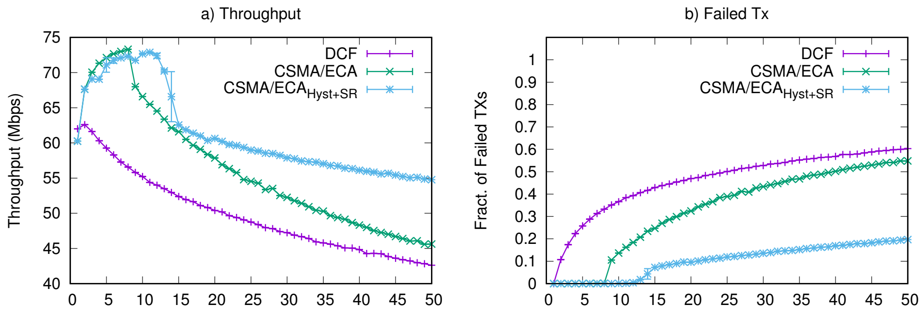

Figure 4.4: Single AP performance using NS-3

Results show how DCF's throughput degrades as the number of contenders increases. This is due to the channel time wasted recovering from collisions. On the other hand, when $N \leq B_{\mathrm{d}}$, CSMA/ECA is able to reach collision-free operation. Further, applying Hysteresis allows CSMA/ECA$_{\text{Hyst+SR}}$ to increase the size of the collision-free schedule augmenting the overall throughput for a greater number of contenders. Schedule Resets seeks opportunities to reduce the size of the deterministic backoff to prevent large periods between successful transmissions.

4.4.2 OBSS simulation results

A control Scenario A is shown on the left of Figure 4.5, where the legend is located at the bottom right corner of the figure. Here, $\mathrm{C_R} = \mathrm{I_R} = 2\delta$ m, so the effect of neighbouring nodes' transmissions can be delimited with precision. This Scenario A configuration implies that the transmissions from a border node $n_{a,0}$ will trigger neighbouring node $n_{b,\pi}$ and AP b's CCA mechanism, deferring their transmissions. This effect is caused by all neighbouring nodes.

On the other side, at the right of Figure 4.5 $\mathrm{C_R}$ and $\mathrm{I_R}$ ranges depend on the received signal power, that is, are subject to the CCA and ED

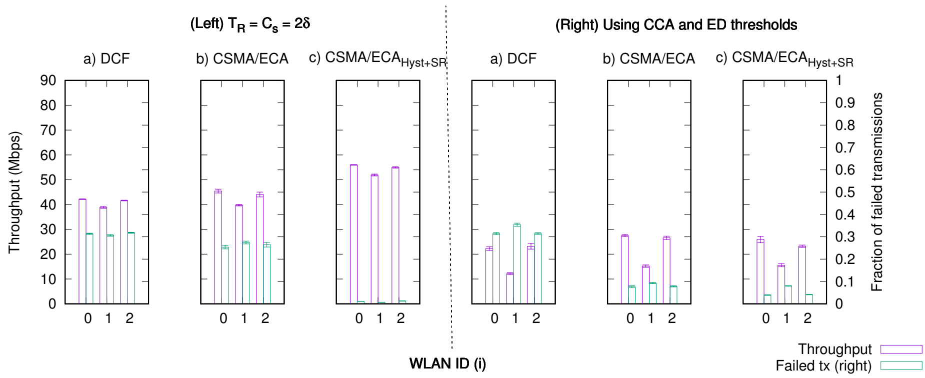

Figure 4.5: Control Scenario A, and Scenario A

thresholds, thus affected by propagation losses. Results in the figure are derived with $N = 4$, $\Delta_x = 15$ m, $\delta = \frac{1}{3}\Delta_x$ m, and $\delta_n = \delta$.

Figure 4.5 (left) shows how the throughput is degraded in the middle WLAN-1. This is caused by the transmissions of nodes from adjacent networks. For instance, when a neighbouring node from WLAN-0, $n_{0,0}$ transmits, $n_{1,\pi}$ and AP-1 detect the channel as busy. Now, suppose that $n_{0,0}$ and $n_{2,\pi}$ transmit at the same time. This condition will completely prevent WLAN-1 nodes from transmitting successfully during the aforementioned nodes' transmissions, bringing periods of inactivity that contribute to the observed throughput degradation. Figure 4.6 clearly shows that neighbouring nodes are the most negatively affected, supporting our assumptions over the control Scenario A.

Still focusing on Figure 4.5 (left), CSMA/ECA shows higher fraction of failures than CSMA/ECA$_{\text{Hys+SR}}$. This is because Hysteresis allows larger schedules, avoiding collisions more efficiently. DCF nodes on the other hand waste channel time recovering from collisions.

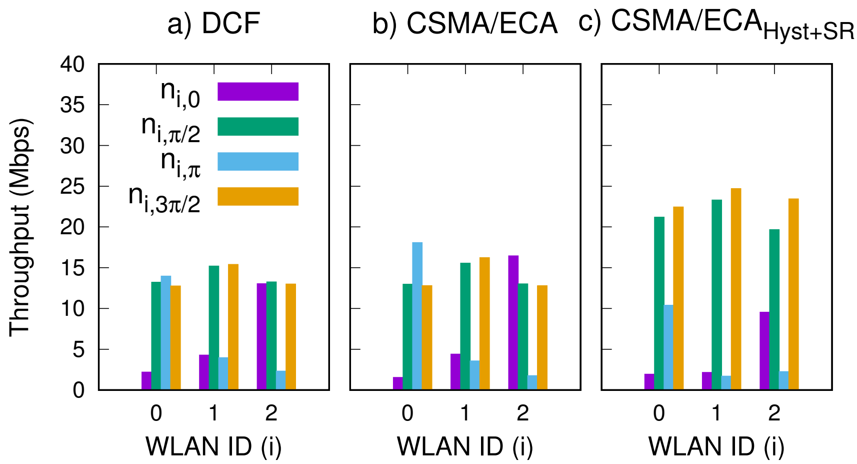

Figure 4.6: Throughput per station for the control Scenario A

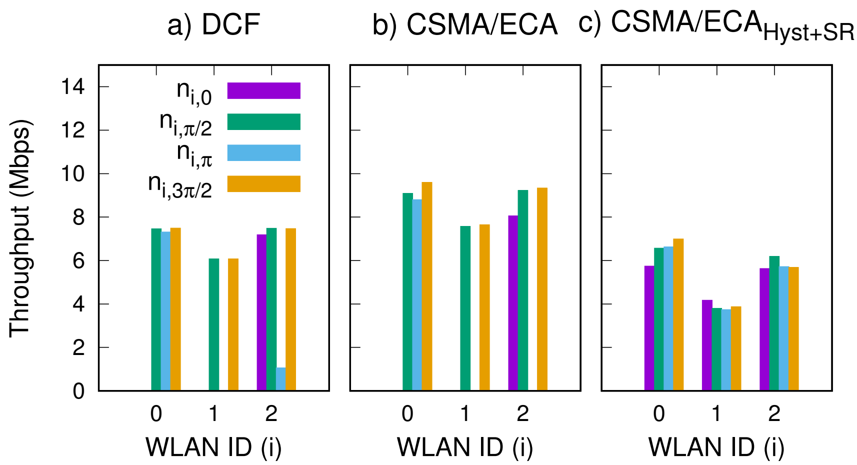

Figure 4.7: Throughput per station for Scenario A

Using the default CCA and ED thresholds increases T_{R} and C_{s} when compared to the control Scenario A. Therefore, for an example node n_{ij} there will be more transmissions affecting the CCA and contention mechanisms. As mentioned in [37], there are now more nodes in the *contention domain*, increasing the collision probability. Looking at the right side of Figure 4.5, it shows a higher number of failed transmissions in DCF, coupled with a considerable overall throughput degradation. CSMA/ECA$_{\text{Hyst+SR}}$ on the other hand, shows less failures, mainly due to a better collision avoidance mechanism.

As CSMA/ECA$_{\text{Hyst+SR}}$ nodes rapidly reach large deterministic backoffs, they are able to produce collision-free schedules with enough empty slots, leveraging the effects of neighbouring nodes' transmissions. On the other hand, CSMA/ECA still shows higher fraction of losses, but the periods of scheduled collision-free operation prevent further increase. Finally, DCF is gravely affected by the higher collision probability.

Figure 4.7 shows the throughput per station. Results from both Figure 4.5 (right) and Figure 4.7 suggest this is a very collision-prone scenario, and CSMA/ECA$_{\text{Hyst+SR}}$ is able to avoid starvation, distributing the available throughput of each WLAN more efficiently than DCF, or for that matter CSMA/ECA, too.

Scenario B and many more users

This section presents results for Scenario B with $N = 20$ and $A = 10$. That is, node $i \in [1, \cdots, N]$ are randomly placed around AP $i \in [1, \cdots, A]$ at $p_{ij}(x, y)$, where $x, y \in [-\delta, \delta]$, and elevated $z = 1.5$m from the floor. As before, $\Delta_x = 15$ m, and $\delta = \frac{1}{3}\Delta_x$ m.

Figure 4.8 shows that CSMA/ECA$_{\text{Hyst+SR}}$ is more efficient at reducing the fraction of failed transmissions. Additionally, Figures 4.9a-c show the JFI [36] for all WLANs using one of the three tested protocols: DCF, CSMA/ECA and CSMA/ECA$_{\text{Hyst+SR}}$.

Results indicate that CSMA/ECA$_{\text{Hyst+SR}}$ not only is able to increase the fairness among contenders of the same WLAN, but also provides an aggregate throughput increase as a consequence. This can be observed in Figure 4.9d.

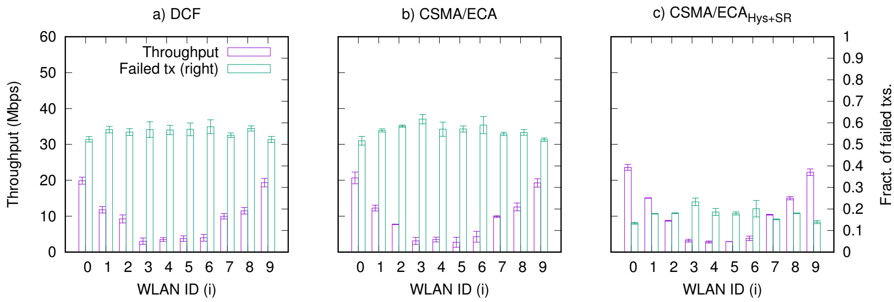

Figure 4.8: Scenario B detailed simulation results

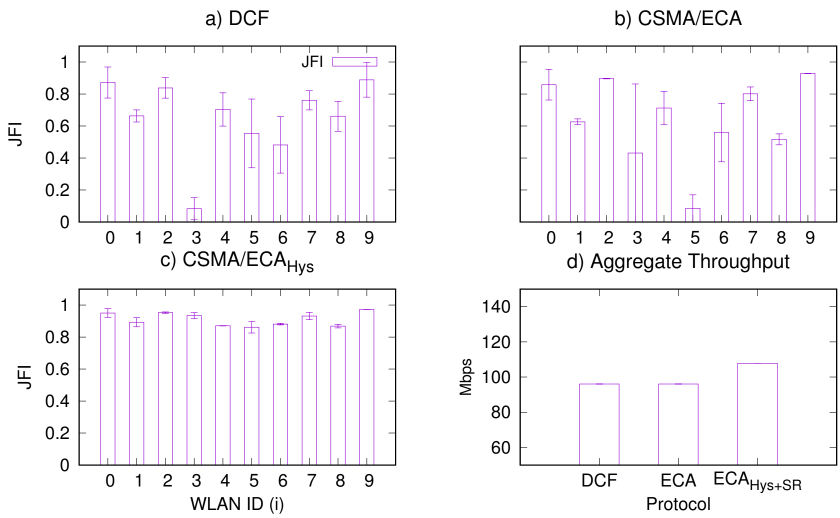

Figure 4.9: Overall results for Scenario B

4.4.3 Scenario HEW: residential building

Figure 4.10 gathers the aggregate results for the Scenario HEW or residential building simulations, with $N = 10$ nodes per WiFi. Figure 4.10a shows the aggregate throughput per floor in the example building of Figure 2.2. As expected, the bottom and top floors have a smaller contention

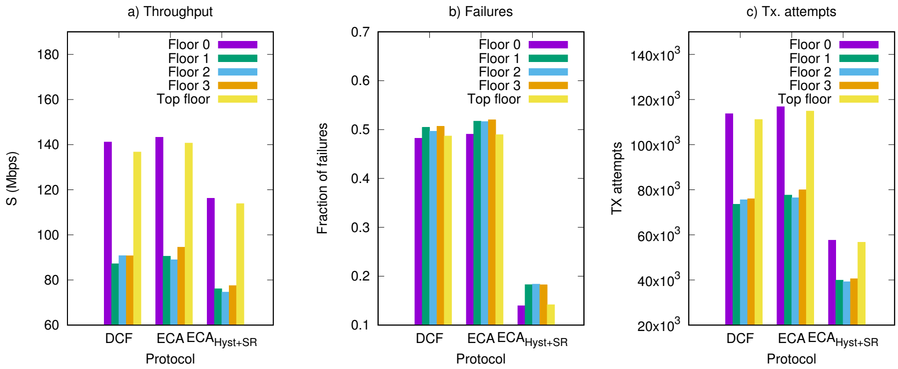

Figure 4.10: Aggregate results for the Scenario HEW simulations

domain, showing higher throughput due to a lower fraction of losses in Figure 4.10b. CSMA/ECA$_{\text{Hyst+SR}}$ (ECA$_{\text{Hyst+SR}}$ in Figure 4.10) stations are unable to reduce the deterministic backoff, ending with a big period between successful transmissions which translates in a lower overall throughput. Nevertheless, Figure 4.10b shows that it is very effective at reducing failures.

Figure 4.11 shows overall metrics of throughput (S), JFI, failures, and transmission attempts. Results show higher throughput for CSMA/ECA over DCF, despite having around 50% of transmissions resulting in failure. The use of a deterministic backoff after successful transmissions creates periods of schedule-like transmissions (as in [41, 61]), increasing the number of successful transmissions.

Attempting to increase the aggressiveness of CSMA/ECA$_{\text{Hyst+SR}}$ may enhance the aggregate throughput for this protocol. In Figure 4.12 we show different configurations of CSMA/ECA$_{\text{Hyst+SR}}$, namely: Hysteresis only (*Hyst*), Schedule Reset as in previous tests (*Hyst+SR*), SR with a reduced $\text{CW}_{\max} \leftarrow 255$ (*Hyst+SR$_R$*), and a configuration of SR denoted as aggressive[4] (*Hyst+SR aggr.*). Even-though *Hyst+SR$_R$* shows higher throughput, it increases the fraction of failed transmissions. The aggres-

[4] It simply means $\gamma = 1$ and a *halving* of the schedule.

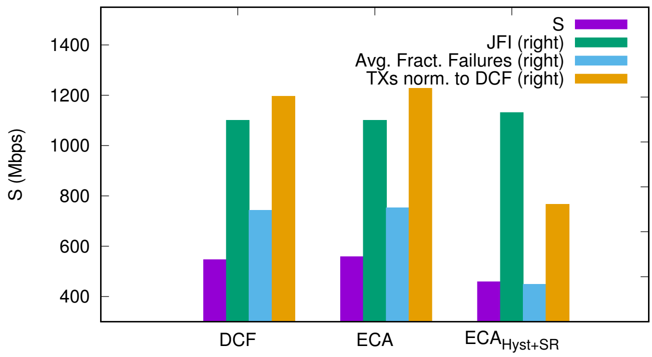

Figure 4.11: Overall metrics for Scenario HEW

sive schedule halving represented by *Hyst+SR aggr.* in Figure 4.12 is not able to reduce the schedule to lower values, despite just analysing the bitmap after 2 consecutive transmissions ($\gamma = 1$). We select *Hyst+SR* as the reference protocol because is the configuration that provides better tradeoff between overall aggregate throughput and fraction of failures.

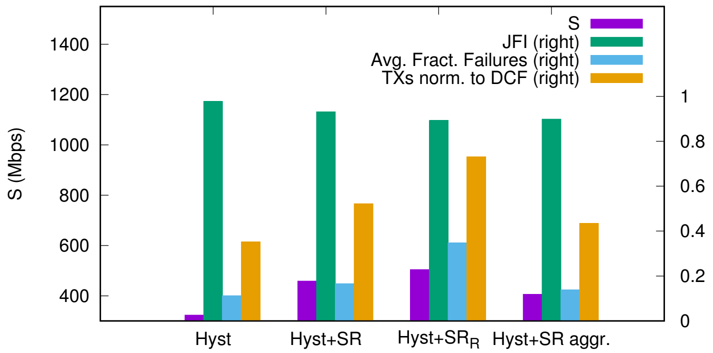

Figure 4.12: Comparison among different CSMA/ECA$_{\text{Hyst+SR}}$ configurations

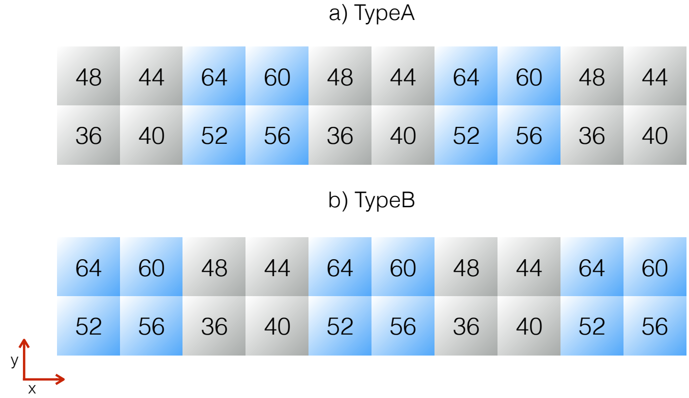

Figure 4.13: Non-overlapping WiFi channel allocations for a Scenario HEW test, $C = 8$ channels

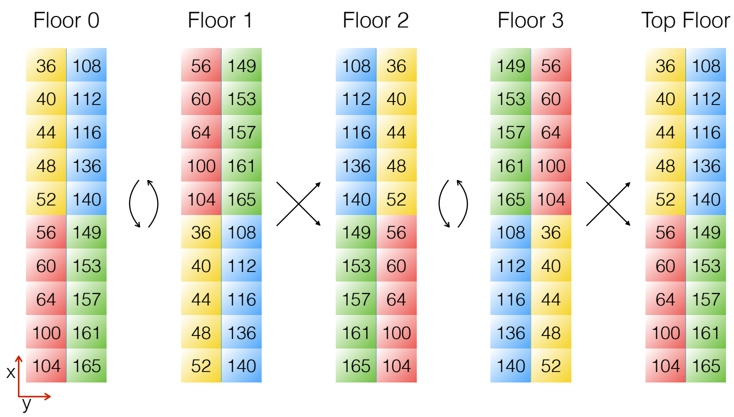

Figure 4.14: Non-overlapping WiFi channel allocations for a Scenario HEW test, $C = 20$ channels

4.4.4 Softening the conditions using different channels

Having a very big contention domain increases the percentage of failed transmissions, nevertheless, CSMA/ECA$_{\text{Hyst+SR}}$ is able to leverage this

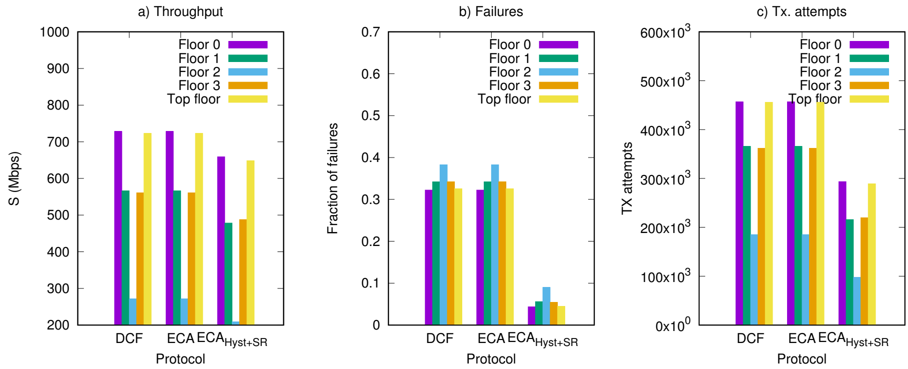

Figure 4.15: Scenario HEW with efficient channel allocation results, $C = 8$ channels

issue using a deterministic backoff and Hysteresis. Despite being outperformed by CSMA/ECA in the residential building scenario, results from Scenario B in Section 4.4.2 show that CSMA/ECA$_{\text{Hyst+SR}}$ provides a considerable reduction of failures while increasing the overall throughput and fairness. This can be beneficial for applications where low losses and fairness are preferred. Furthermore, as less transmission attempts are performed CSMA/ECA$_{\text{Hyst+SR}}$ may constitute an advantage for energy constrained applications.

The following presents simulation results using Scenario HEW in saturation with a different WiFi channel for each room of the building. First, a distribution using only $C = 8$ non-overlapping WiFi channels in the IEEE 802.11n 5GHz band (shown in Figure 4.13), and then a more efficient distribution using $C = 20$ (see Figure 4.14).

1. Figure 4.13 shows the two types of WiFi channel assignments for each floor of the building in Figure 2.2 using $C = 8$ non-overlapping channels. Floors 0, 2 and 4 use TypeA, while floors 1 and 3 use TypeB.

   Figure 4.15, shows the results obtained when allocating different

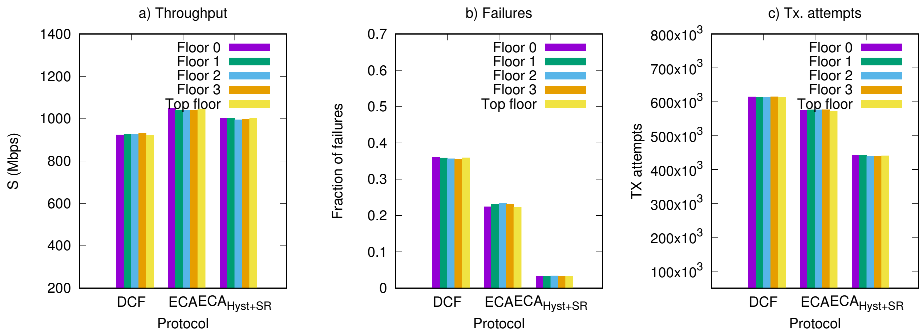

Figure 4.16: Scenario HEW with efficient channel allocation results, $C = 20$ channels

channels for each WLAN. Interestingly, the effect over the middle floor is easily observed, revealing higher failures and lower throughput than the rest. CSMA/ECA and DCF show similar performance, as $N > B_{\mathrm{d}}$ this is expected. On the other hand, nodes using CSMA/ECA$_{\mathrm{Hyst+SR}}$ are unable to reduce the schedule length any further, ending with big periods between successful transmissions that translate into lower throughput. Nevertheless, the efficient collision avoidance mechanisms use by this protocol reduces the fraction of failures and transmissions attempts considerably.

2. Then, we proceed to an even more efficient allocation of the available non-overlapping channels using $C = 20$. The channel distribution is shown in Figure 4.14, while results are presented in Figure 4.16.

   As each WLAN contention domain is effectively reduced by a more efficient distribution of the non-overlapping channels, a fairness increase is evidenced. Further, the middle floor (Floor 2) is not specially affected.

   In this scenario CSMA/ECA$_{\mathrm{Hyst+SR}}$ is still unable to outperform CSMA/ECA. Nevertheless, the same benefits in terms of failed

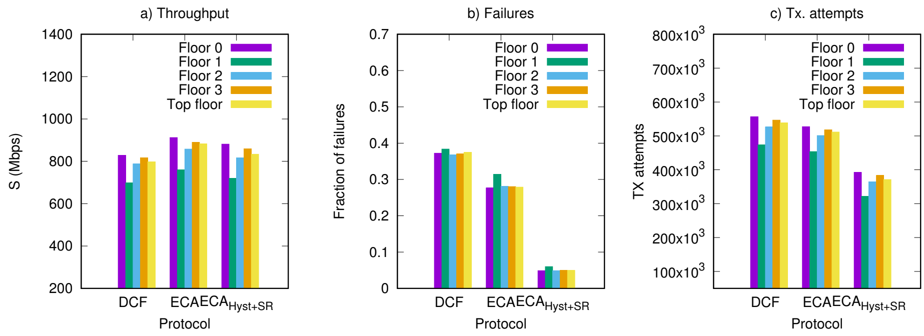

Figure 4.17: Scenario HEW with random channel allocation results, $C = 20$ channels

transmissions and transmission attempts are observed.

3. Figure 4.17 shows simulation results for $C = 20$, but the allocation is made at random.

   This scenario, al-though not ideal, can be considered closer to reality. Figure 4.17 shows higher throughput for CSMA/ECA, but the difference with CSMA/ECA$_{\text{Hyst+SR}}$ is very small. As the latter still shows lower failures and transmission attempts, it is considered as the overall best in this scenario.

Overview

As scenarios become more crowded, the interaction among Overlapping BSS (OBSS) becomes of significant importance for optimising the MAC throughput. The Clear Channel Assessment (CCA) mechanism for determining the state of the channel (busy or empty) relies of MAC-specific[5] thresholds, which can change considerably the size of the contention domain of a node.

DCF, as it is based on a random backoff technique, is gravely affected by dense scenarios. Degrading the overall throughput as the number of

[5] That is, IEEE 802.11-specific

contenders increases. CSMA/ECA on the other hand, uses a deterministic backoff after successful transmissions technique, which coupled with extensions like Hysteresis and Schedule Reset allows CSMA/ECA$_{\text{Hyst+SR}}$ to construct collision-free schedules for more contenders.

We tested these three protocols under different scenarios, ranging from single AP, to a complex residential building following TGax specifications for IEEE 802.11ax. Overall, the deterministic backoff technique increases the number of successful transmissions in all of the tested scenarios, outperforming DCF. Furthermore, Hysteresis and Schedule Reset keep CSMA/ECA$_{\text{Hyst+SR}}$'s failures way lower than the other tested protocols. As this is achieved using longer collision-free schedules, less transmissions attempts are performed, consequently providing a potential reduction in the overall energy consumption.

It is observed that some attributes are beneficial in certain scenarios. For instance, low contention scenarios can draw benefit from the aggressiveness provided by DCF's random backoff mechanism, which leverages the time wasted recovering from failed transmissions. Whereas high contention conditions, like crowded single-AP or multi-AP scenarios may benefit from a deterministic backoff after successful transmissions technique, such as the used by CSMA/ECA and CSMA/ECA$_{\text{Hyst+SR}}$. Results from this work evidence the importance of being able to determine the network conditions, and calls for mechanisms able to adapt the MAC protocol accordingly in order to draw benefits.

Different research directions could be derived from this work:

- Multi-rate scenarios, which are affected by propagation loss models.

- Dynamic ED and CCA threshold adaptation using Dynamic Sensitivity Control [72].

- Schedule Reset and its relation to the sensitivity thresholds, as SR's decisions are based on what is observed in the channel.

- Big data analytics. As scenarios get bigger and complex, better data manipulation techniques should be used to interpret what is really happening in the scenario.

As all our implementation is open source and freely available at [64, 65], we encourage other researchers to learn from our experience and start developing tests using complex mobility and propagation models as the ones used in Scenario HEW.

Chapter 5

TRAFFIC DIFFERENTIATION USING CSMA/ECA

5.1 CSMA/ECA$_{\text{QoS}}$

CSMA/ECA and its extensions are able to construct collision-free schedules under saturated conditions, outperforming CSMA/CA. Furthermore, CSMA/ECA uses the same default contention parameters as CSMA/CA, so the compatibility is maintained [57].

Providing priority is to ensure a more frequent access to some ACs over others. In CSMA/ECA 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 considers the same four ACs.

Nevertheless, AIFS and TXOP are not fit for multiple CSMA/ECA 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 with real collisions. Figure 5.1 shows a VC in CSMA/ECA with two queues[1] (indicated by the outline)

[1]Only considering AIFSN values of 2 and 4, B_{d} of 4 and 8 for AC1 and AC2 respectively

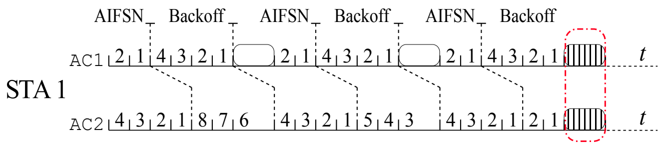

Figure 5.1: Example temporal evolution of CSMA/ECA with two ACs using AIFS resulting in a virtual collision

Table 5.1: CSMA/ECA$_{\text{QoS}}$ contention parameters

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

consequence of using AIFS during a collision-free schedule. As the lower priority AC proceeds to select a random backoff, its next transmission may disrupt any ongoing collision-free operation.

TXOP in EDCA ensures that all traffic from the same category receives on average the same channel time. In contrast, CSMA/ECA's goal through Fair Share is to provide close to equal average throughput to same-priority ACs. The combination of Fair Share and Schedule Reset provides throughput fairness through aggregation. Further, it attempts to evenly distribute the channel time among AC increasing the frequency of transmissions, permanently seeking opportunities to reduce the schedule.

As EDCA extends DCF into four ACs, similarly we define an instance of CSMA/ECA for each AC. We will refer to CSMA/ECA with multiple ACs as CSMA/ECA$_{\text{QoS}}$ from here forward. Table 5.1 shows the CW, lowest and largest B_{d}, and maximum backoff stage m.

Figure 5.2 shows an example of CSMA/ECA$_{\text{QoS}}$ with two contenders and two ACs; where AC1 has higher priority than AC2[2]. In the figure, the

[2] CW_{min}[AC1,AC2] $= [8, 16]$; m[AC1,AC2] $= [5, 5]$

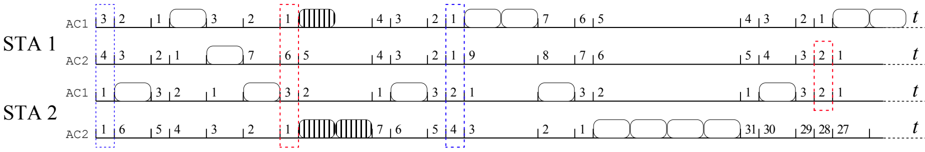

Figure 5.2: An example of the temporal evolution of CSMA/ECA$_{\text{QoS}}$ in saturation

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 5.2, 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 5.2 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.

5.2 Collisions and Virtual Collisions-free operation using Smart Backoff

Consider a complete schedule of length $C = 2^m \text{CW}_{\min}$, and $m = 5$. With CSMA/ECA and a single AC 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 5.1). Nevertheless, with CSMA/ECA$_{\text{QoS}}$ and all ACs in saturation i.e., have a packet to transmit, each contender mimics the behaviour of four saturated CSMA/ECA 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, CSMA/ECA$_{\text{QoS}}$ can provide collision-free operation for up to $2^{(m[\text{VO}]-3)}\text{CW}_{\min}[VO] = 32$ contenders, where $m[\text{VO}]$ is the maximum backoff stage of the AC with the smallest $\text{CW}_{\max}$, that is AC[VO] in Table 5.1[3].

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 $B[\text{AC}]$ that is not equal to any of the other AC's counters. 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 5 decribes 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$_{\text{QoS}}$, regardless of the aggregation mechanism used.

What results from Algorithm 5 is a Smart Backoff counter that will not cause a VC on the next transmission attempts.

[3]The maximum number of collision-free contenders in saturation is reduced when using the Schedule Reset Mechanism. This is due to the reduction of the average backoff stage of AC[VO], $k[VO] \leq m[VO]$.

```
AC := 4; // number of Access Categories
CW_min[AC]; // CW_min for all ACs
B_d[AC]; // B_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 ∈ [1, AC]
//
while (F ≠ 1) or (Cb ≠ 1) do
    B[i] ← U[0, 2^{k[i]} CW_min[i]];
    for (j = 1; j ≤ AC; j++) do
        if (j ≠ i) then
            F[j] ← |B[i] − B[j]| mod [min(B_d[i], B_d[j])];
            if (F[j] ≠ 0) then
                F[j] ← 1;
            if (B[i] ≠ B[j]) then
                Cb[j] ← 1;
            else
                Cb[j] ← 0;
return (B[i]);
```

Algorithm 5: Smart Backoff: eliminating Virtual Collisions in CSMA/ECA$_{\mathrm{QoS}}$

5.3 Simulation setup description

In order to test the traffic differentiation in CSMA/ECA$_{\mathrm{QoS}}$ and its capability of outperform EDCA in terms of number of supported delay-sensitive flows and aggregate throughput, we have used a customised version of the COST simulator [76], which is available via [63]. If not expressed otherwise, each point in the presented figures is obtained from averaging twenty executions of duration equal to forty seconds. Further considerations:

- PHY/MAC headers, and other unspecified parameters follow the IEEE 802.11ax (5 GHz) standard [30].
- All nodes can be assumed to be in communication range with each other.
- Transmission of several frames per attempt supposes AMPDU aggregation with compressed Block ACK [50].
- RTS/CTS mechanism is used, as transmitting multiple frames in a TXOP requires a protection mechanism in EDCA [2].
- Smart Backoff is used in CSMA/ECA$_{\text{QoS}}$.
- Aggressive Schedule Reset is used, with $\gamma = 1$.
- Dynamic Stickiness defines a maximum `stickiness = 2`.
- CSMA/ECA$_{\text{QoS}}$ AC[BK] does not use Schedule Reset in order to provide differentiation with AC[BE].

Additionally, Table 5.2 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 generators, channel conditions and overall scenarios to be evaluated. Then, simulation results for achieved throughput, number of collisions and time between successful transmissions are presented.

5.3.1 Updated traffic sources

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

Table 5.2: PHY and MAC parameters for CSMA/ECA$_{\text{QoS}}$ simulations

PHY	
Parameter	**Value**
PHY rate	65 Mbps
Channel Width	20 MHz
Number of Streams	1
OFDM bits/symbol	6
Coding rate	3/4
Empty slot	9 μs
DIFS	34 μs
SIFS	16 μs
MAC	
Parameter	**Value**
Maximum retransmission attempts	7
MAC queue size (Packets)	1000
CSMA/ECA$_{\text{QoS}}$	
Parameter	**Value**
Schedule Reset mode	aggressive ($\gamma = 1$)
Dynamic stickiness	on
Smart Backoff	on

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 performing a VoIP call, its MAC queue will be empty while silence is detected by the codec.

Non-saturation scenarios play an important part on the performance evaluation, specially because both EDCA and CSMA/ECA$_{\text{QoS}}$ reset their respective $\text{CW}_{\text{curr}}[AC] \leftarrow \text{CW}_{\min}[AC]$ when the queue for an specific AC is detected empty, which continuously resets collision-free schedules. Details of the traffic sources for the non-saturated scenario are provided below as well as in Table 5.3.

Table 5.3: Traffic sources detail

1) AC[VO]	
Parameter	**Value**
On duration	3.110 s
Off duration	3.2727 s
Rate	15.2 kbps
Payload	38 B
2) AC[VI]	
PSNR	43.5 dB, best
GOP size	16
GOP	*IBBBPBBBPBBBPBBB*
Average I size	5658 B
Average P size	1634 B
Average B size	348 B
Frame size standard deviation	2 times the average
Average Rate	300 kbps
3) AC[BE] and AC[BK]	
Rate	65 Mbps
Payload	1470 B

- AC[VO] source: we emulate a voice codec with silence detection. That is, when the energy of a voice signal is below a threshold during a determined number of sampled packets, the source stops injecting voice packets into the MAC queue. The Internet Low Bit Rate Codec (iLBC) [6] is a robust codec designed for IP networks. It features smooth speech quality degradation in case of frame losses, making it suitable for VoIP. It is modelled as an On/Off source, other parameters are shown in Table 5.3. A Constant Bit Rate (CBR) traffic source is active during the On phase. It follows a geometric distribution of talkspurts and silence intervals. Durations follow *Geom-APD-W0* settings in [46].

- AC[VI] source: follows the characteristics of the H.264/Advanced

Video Coding (or H.264/AVC) [75]. Its improved compression tools makes it ideal for high quality video streaming. Video source modelling greatly depends on the video source, that is, action films after packetised produce very different frames than a static interview. This results in rate variability. As also tested in [75], an example Group of Images (GOP) representative of an action movie source is selected[4]. A GOP is composed of I, P and B frames, used to represent past, present and future in a video stream. For a given image quality (PSNR) and size (in pixels by pixels), Table 5.3 shows the average and standard deviation of the I, P and B frame sizes, alongside other video source characteristics.

- AC[BE] and AC[BK] sources: queues are saturated in all scenarios.

As the goal of the *saturated* scenario evaluation is to compare the efficiency of the contention mechanisms used by EDCA and CSMA/ECA$_{\text{QoS}}$, all ACs use circular MAC queues, which are filled at startup with 1470B frames.

5.3.2 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 MPDU not being acknowledged, p_e. It affects every MPDU independently. That is, for every transmission we draw a number from a random variable $X \sim \mathcal{U}[0, 1]$, if the number drawn is lower than p_e the frame will not be acknowledged. In the case of MDPU aggregation (AMPDU), it is considered a failed transmission only if all MPDUs in the AMPDU are independently affected by p_e. A value of $p_e = 0.1$ has been selected for the simulation of the non-saturated scenario, but a comparison with different values for p_e is also provided. The saturation scenario is tested with a perfect channel.

[4]Due to its higher rate variability.

5.4 Simulation results

This section presents results using both aggregation strategies, that is, Fair Share and TXOP[AC]. These are referred to as CSMA/ECA$_{\text{QoS+FS}}$ and CSMA/ECA$_{\text{QoS+TXOP}}$, respectively. The latter means that upon wining access to the channel an ACs will transmit without contention for as long as indicated by TXOP[AC] in Table 2.2. Any kind of frame aggregation is only performed on high priority ACs, that is, AC[VO] and AC[VI]. Furthermore, to provide differentiation between AC[BE] and AC[BK], Schedule Reset is turned off for AC[BK]. This means that $CW_{\text{curr}}[BK]$ is only reduced when reaching the retransmission limit or when the queue for this AC is detected empty. In both cases it is reset to $CW_{\text{min}}[BK]$.

We first evaluate the performance of CSMA/ECA$_{\text{QoS+FS}}$, taking special interest to the throughput, failures, fairness and average delay in both traffic conditions. Table 5.2 shows the default CSMA/ECA$_{\text{QoS}}$ settings regarding Schedule Reset, Smart Backoff, and stickiness. Then, we study EDCA and compare the results against CSMA/ECA$_{\text{QoS+FS}}$, including a mixed network scenario. Next, we replace Fair Share in CSMA/ECA$_{\text{QoS}}$ with TXOP rules to provide a just comparison with EDCA. We identify this case as CSMA/ECA$_{\text{QoS+TXOP}}$. Finally, we propose a discussion about the results.

5.4.1 CSMA/ECA$_{\text{QoS+FS}}$

Figure 5.3 shows as columns : a) Average aggregate throughput, b) failed transmissions, and c) Jain's Fairness Index [36][5] for CSMA/ECA$_{\text{QoS+FS}}$ in saturation. All frame sizes are equal to 1470B. The bottom row focus on the non-saturation scenario. It shows the same metrics except for the latter, which shows d) the average queueing delay (queue + contention).

As shown in the figure, CSMA/ECA$_{\text{QoS+FS}}$ is able to keep a steady overall throughput for a large number of contenders in saturated condi-

[5]The JFI is an indicator of fairness regarding the ditribution of the available throughput in a system. As the throughput in WLANs is to be equally distributed among contenders, a JFI$= 1$ is expected.

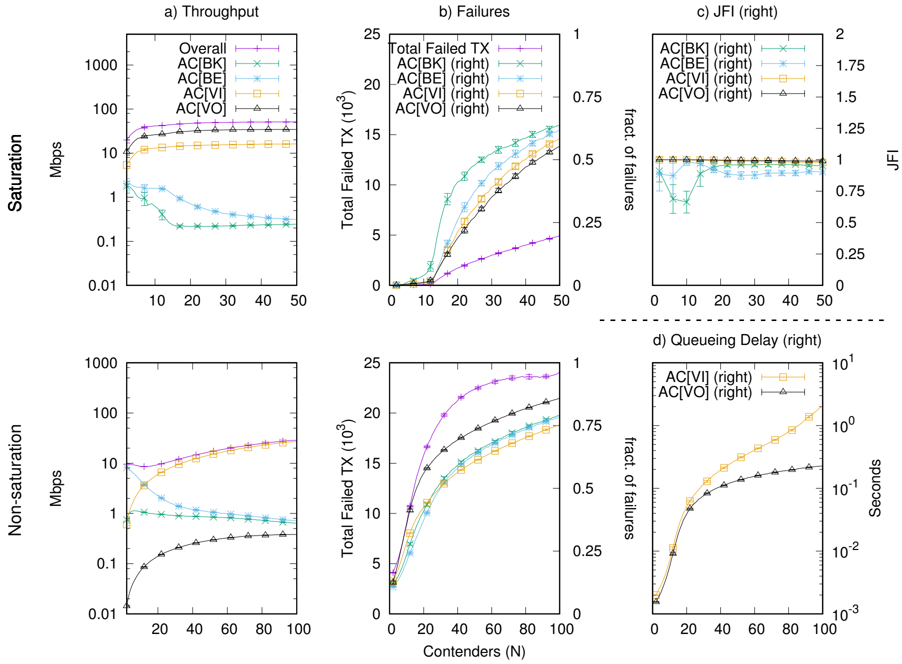

Figure 5.3: Combined results for CSMA/ECA$_{\text{QoS+FS}}$

tions. Moreover, as ACs aggregate frames proportionally to its current schedule length, throughput fairness is achieved for high priority ACs. Collision-free operation is reached for $N \leq 12$, as shown in Figure 5.3-b. This is lower than the maximum of $N = 32$ mentioned in Chapter 5.2 and is a consequence of Schedule Reset's $\gamma = 1$. For $N \leq 12$, SR often fails to encounter further reduction opportunities, often succeeding keeping ACs with shorter schedules than the maximum. At higher $N > 12$, the aggressiveness of SR due to $\gamma = 1$ leads to schedule reductions that cause collisions.

Figure 5.4 provides a set of comparisons for: a) different configurations of Schedule Reset fixing the number of contenders to $N = 8$.

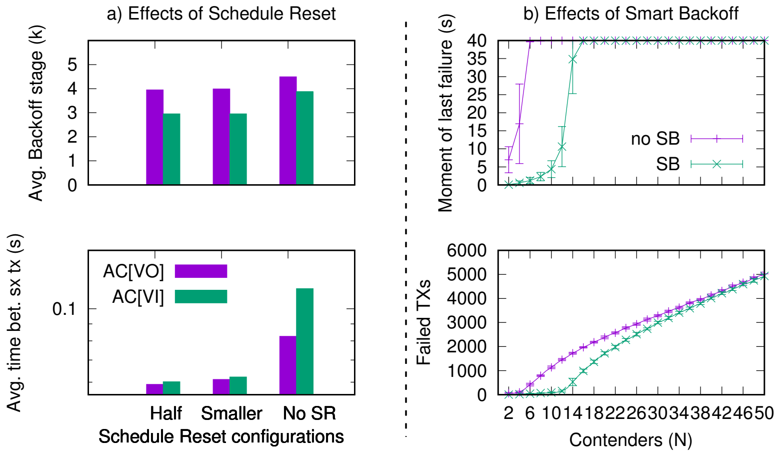

Figure 5.4: Different Schedule Reset configurations

These are: *Half*: SR only attempts to halve the current deterministic backoff; *Smaller*: changes to smaller backoffs are allowed; *no SR*: not using Schedule Reset. Then: b) the effect of Smart Backoff over the convergence time of CSMA/ECA$_{\text{QoS+FS}}$, as well as the fraction of failed transmissions in saturated conditions.

As shown in Figure 5.4-a, the difference between selecting *Half* the current schedule and looking to reduce it to the *Smaller* available length are not significant in terms of average final backoff stage. Nevertheless, a reduction is observed when compared against not using SR. Looking at the average time between successful transmissions, the *Half* configuration provides better results given that a drastic reduction of the schedule increases the collision probability when using $\gamma = 1$. As this value of γ is required in order to increase the reduction attempts in non-saturated conditions with $p_e > 0$, the *Half* configuration is used. That is, Schedule Reset will evaluate the bitmap and only perform a reduction to half the current deterministic backoff.

Smart Backoff prevents virtual collisions and consequent disruption of collision-free schedules. As shown in Figure 5.4-b, collision-free oper-

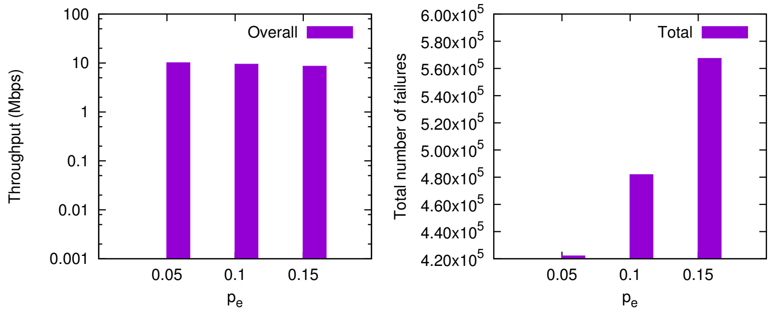

Figure 5.5: Average throughput and Failed transmissions for different levels of p_e in non-saturation.

ation is only achieved with SB, and for $N \leq 14$ during simulation time.[6].

In non-saturation, CSMA/ECA$_{\text{QoS+FS}}$ in Figure 5.3 is able to construct collision-free schedules for short periods of time that allow AC[VO] and AC[VI] to saturate at a much higher number of contenders. Further, as shown in Figure 5.3-d the queueing delay of the highest priority AC[VO] is lower than other ACs. The value of $p_e = 0.1$ is selected because it produces a moderate increase in the total number of failures observed in Figure 5.5, where a range of $p_e > 0$ with a fixed $N = 1$ are tested. As nodes are supposed to be in communication range among each other, we avoid using higher p_e values.

5.4.2 EDCA comparison and coexistence

Figure 5.6 gathers the simulation results in a saturated network using Basic Access (BA) and RTS/CTS. It shows the average aggregate Throughput and Collisions for a) EDCA, and b) CSMA/ECA$_{\text{QoS+FS}}$ in saturation. All frame sizes are equal to 1470B.

EDCA with RTS/CTS (Figure 5.6-a bottom) shows higher throughput than using BA. This is an effect of wasting less time recovering from

[6]Being the average backoff stage for AC[VO], $k[\text{VO}] = 4$, as mentioned in Chapter 5.2 collision free operation is possible for up to $2^{(k[\text{VO}]-3)}CW_{\min} = 16$ nodes.

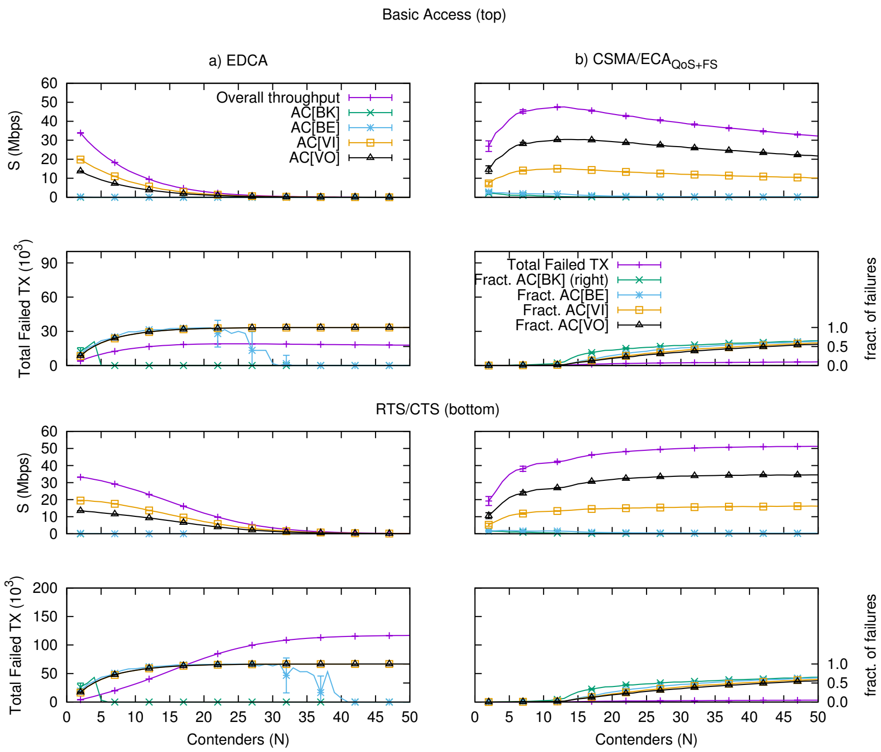

Figure 5.6: Average aggregate Throughput and Collisions for a) EDCA, and b) CSMA/ECA$_{\text{QoS+FS}}$ in saturation

collisions. Moreover, as more time is made available for transmission attempts, RTS/CTS produces a higher total number of failed transmissions, but keeps the same fraction of failures as in BA. RTS/CTS also loosens the starvation of low priority ACs. As indicated by the fraction of failures, the starvation of EDCA AC[BE] occurs at a higher $N = 42$, against $N = 32$ observed using BA. Given that RTS/CTS is required by the IEEE 802.11e standard when performing frame aggregation, further results do not consider BA.

The efficiency of eliminating collisions with CSMA/ECA$_{\text{QoS+FS}}$ is clearly evident at high number of contenders. Conversely, EDCA's throughput

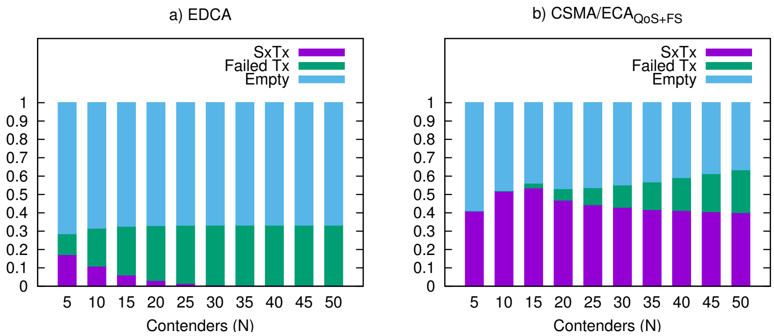

Figure 5.7: Fraction of slots during a saturated simulation with a growing number of contenders

decreases very rapidly, mostly because of an extremely high fraction of failures. Figure 5.7 shows the percentage of empty, successful and failure slots observed during a simulation in saturated conditions.

Even though it clearly outperforms EDCA for high number of contenders, CSMA/ECA$_{\text{QoS+FS}}$ shows lower overall throughput for $N \leq 5$ in Figure 5.6-b. This is due to Fair Share, which aggregates according to the current backoff stage[7]. As collisions are quickly eliminated with Smart Backoff, the level of aggregation produced by Fair Share is often lower than TXOP[AC], hence the lower throughput.

Turning to the non-saturation scenario, Figure 5.8 shows the average aggregate throughput, fraction of failures, and time between successful transmissions as rows $i = (1, 2, 3)$, using labels $j = (a, b, c, d)$ to identify each AC as a column. Subfigures are referred as Figure 5.8.$i.j$. Legend is located at the bottom right corner of the figure

In Figure 5.8.1.a and 5.8.1.b, EDCA AC[VO] and AC[VI] achieve less throughput, mainly because they get saturated at lower N. On the other hand, CSMA/ECA$_{\text{QoS+FS}}$ AC[VI] saturates with a considerable larger N[8]. On the other hand, AC[BE] in Figure 5.8.1.c shows a slightly higher

[7]That is, $2^{k[AC]}$ frames in an AMPDU.

[8]The average number of aggregated frames using Fair Share is greater than TXOP[VI], thus emptying AC[VI] queue quicker.

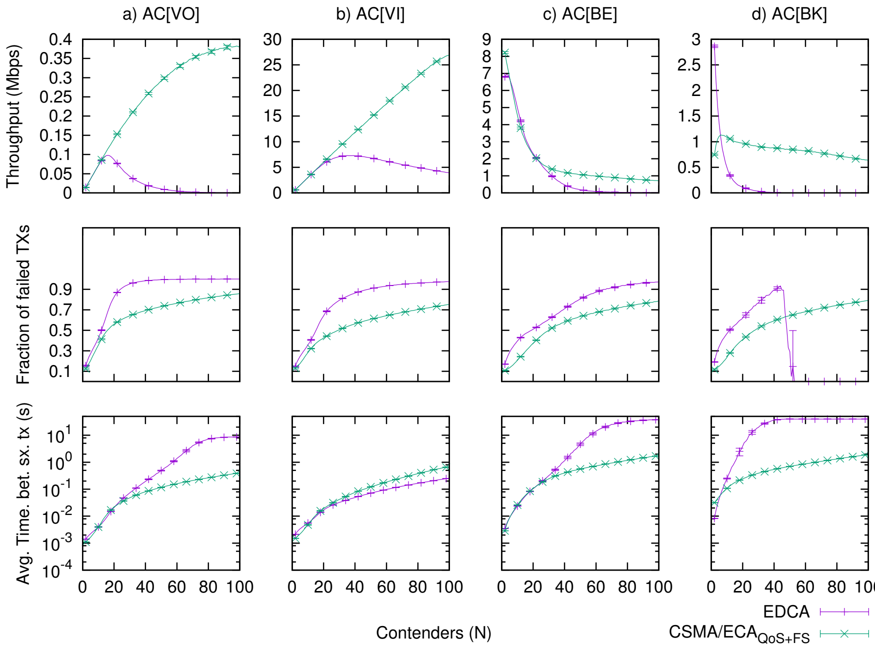

Figure 5.8: Comparison among protocols per AC in the non-saturation scenario

throughput in EDCA for $6 < N \leq 16$. This is attributed to the aggressiveness of EDCA's random backoff. Nevertheless, for $N > 16$ CSMA/ECA$_{\text{QoS+FS}}$ AC[BE] maintains a steady throughput for an increased number of contenders. Further, CSMA/ECA$_{\text{QoS+FS}}$ AC[BK] outperforms EDCA's for $N > 5$ (big deterministic backoffs and the lack of Schedule Reset in AC[BK] account for the lower throughput for $N \leq 5$).

A big part of CSMA/ECA$_{\text{QoS+FS}}$ throughput enhancement is consequence of a better collision avoidance. This is supported by the reduced fraction of failures shown in Figure 5.8.2. Furthermore, the lower fraction of failures observed are the result of the higher saturation point of AC[VO] and AC[VI] due to Fair Share (see Figure 5.8.1.a and 5.8.1.b).

CSMA/ECA$_{\text{QoS+FS}}$ AC[VO] in Figure 5.8.3.a has a lower average time between successful transmission, and for a larger number of contenders than EDCA. Looking at AC[VI] in Figure 5.8.3.b, CSMA/ECA$_{\text{QoS+FS}}$ shows a higher metric when $N > 16$. This is partly due to the duration of successful transmissions of low priority ACs, which are brought to starvation by EDCA as N increases.

Figure 5.8.3.c shows higher time between successful transmissions for EDCA AC[BE]. Conversely, this same metric is slightly higher for CSMA/ECA$_{\text{QoS+FS}}$ AC[BK] at $N \leq 5$, as shown in Figure 5.8.3.d. As AC[BK] does not use Schedule Reset, the big deterministic backoffs used are responsible for longer periods between successful transmissions. Nevertheless, this effect is reversed for higher N.

Mixed Scenario

The following results are extracted from simulations performed with a network setup composed of two types of nodes: 50% EDCA and 50% CSMA/ECA$_{\text{QoS+FS}}$. It uses the non-saturation scenario settings with $p_e = 0.1$. Figure 5.9 shows per node average aggregate throughput, fraction of failed transmissions, and time between successful transmissions as rows $i = (1, 2, 3)$, using labels $j = (a, b, c, d)$ to identify each AC as a column. Subfigures are referred as Figure 5.9.$i.j$. Curves from pure EDCA and CSMA/ECA$_{\text{QoS+FS}}$ networks are also presented. Legend is located at the bottom right corner of the figure.

Figure 5.9.1.a shows EDCA AC[VO] getting saturated at around the same number of contenders as in the non-saturated scenario of Figure 5.8.1.a ($N = 14$). Similarly, as the total number of contenders (N') increases, the throughput of EDCA AC[VO] is degraded even more. CSMA/ECA$_{\text{QoS+FS}}$ nodes are able to avoid collisions more efficiently, resulting in an increased number of successful transmissions for even more users. EDCA AC[VI] in Figure 5.9.1.b shows the same saturation point as in Figure 5.8.1.b.

Still referring to the average throughput, CSMA/ECA$_{\text{QoS+FS}}$ nodes's AC[BE] and AC[BK] in Figure 5.9.1.c and 5.9.1.d show lower throughput than EDCA's for $N'_{\text{be}} < 16$ and $N'_{\text{bk}} < 10$, respectively. This is also observed in Figures 5.8.1.c and 5.8.1.d. Again, this is because the

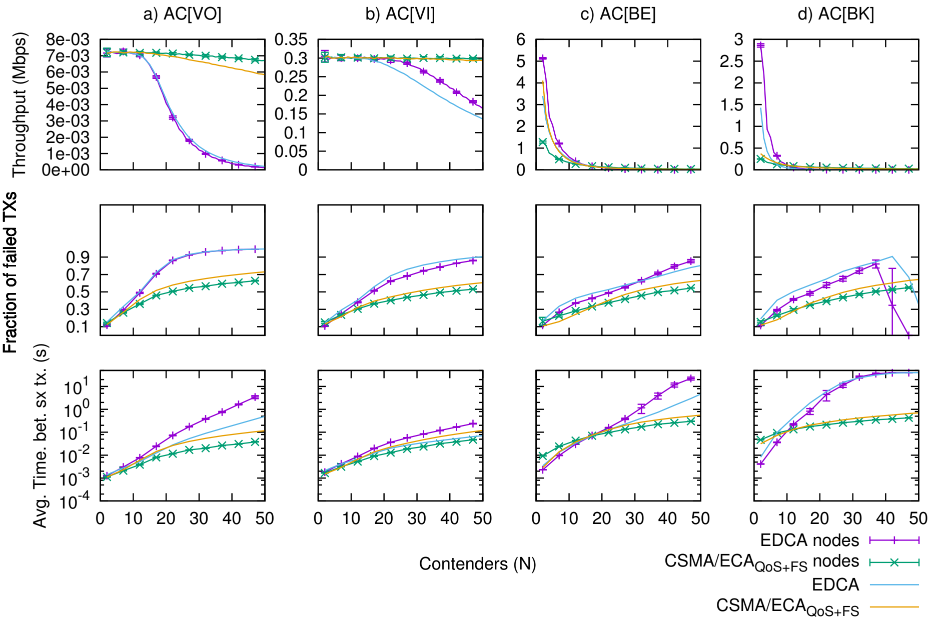

Figure 5.9: Comparison among protocols per AC in the Mixed Scenario in non-saturation

average deterministic backoff used by CSMA/ECA$_{QoS+FS}$ AC[BE] and AC[BK] at this number of contenders increases the time between successful transmissions beyond EDCA's. This effect can be seen in Figure 5.9.3.c and 5.9.3.d.

Short periods of collision-free operation are achieved among successful ACs using CSMA/ECA$_{QoS+FS}$ due to the deterministic backoff after successful transmissions. This reservation-like[9], instead of random contention mechanism is less aggressive, reducing the number of transmission attempts. Nevertheless, it considerably increases efficiency by eliminating collisions.

[9]From the point of view of each AC.

Figure 5.9.3 shows the average time between successful transmissions. EDCA AC[VI] and AC[VO] are negatively affected by nodes using CSMA/ECA$_{QoS+FS}$. In fact, both ACs's metrics are always higher than CSMA/ECA$_{QoS+FS}$'s (Figure 5.9.3.a and 5.9.3.b). This is mainly due to CSMA/ECA$_{QoS+FS}$ AC[BE] and AC[BK] transmissions, which are normally starved in crowded EDCA networks.

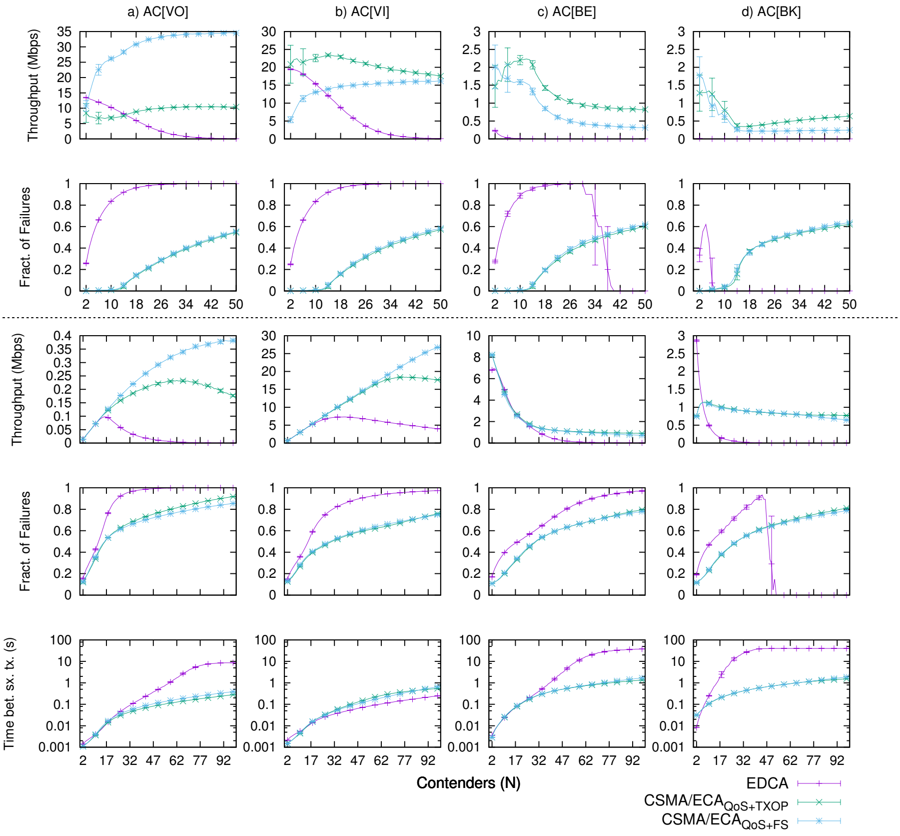

Figure 5.10: Comparison using saturated and non-saturated sources: EDCA, CSMA/ECA$_{QoS+TXOP}$, and CSMA/ECA$_{QoS+FS}$

5.4.3 CSMA/ECA$_{\text{QoS+TXOP}}$

Fair Share aggregates up to 32 frames in an AMPDU, nevertheless, the variable-size video frames proposed for the non-saturation scenario often sum up to more than the maximum TXOP limit defined for EDCA (see Table 2.2). Conversely, EDCA aggregates more packets at lower number of nodes. As Fair Share performs aggregation according to the AC's schedule length, at $N \leq 12$ CSMA/ECA$_{\text{QoS+FS}}$ ACs reach collision-free operation with short schedules.

To provide a just comparison with EDCA, Fair Share is adjusted. That is, AC[VO] and AC[VI] are instructed to always transmit as long as TXOP[AC], as in EDCA. Figure 5.10 shows the average aggregate throughput (S) and failed transmissions for EDCA and CSMA/ECA$_{\text{QoS+TXOP}}$ in saturation (top). The bottom of the figure shows the same metrics and the average time between successful transmissions in non-saturation. Columns show the different metrics per AC.

The elimination of collisions with CSMA/ECA$_{\text{QoS+TXOP}}$ in saturation results in an uneven distribution of the channel resources among contenders for $N \leq 10$, showing high variability and throughput unfairness. This was originally expected and solved with Fair Share (see Figure 5.3-c and [57]), but as transmissions are limited by TXOP[AC], ACs with larger schedules are not compensated aggregating more. Instead, CSMA/ECA$_{\text{QoS+TXOP}}$ ACs pursue opportunities to leverage this issue attempting reductions of the deterministic backoff using Schedule Reset. As the number of contender increases ($N > 10$), collisions push all CSMA/ECA$_{\text{QoS+TXOP}}$ ACs to their largest deterministic backoff, establishing throughput fairness among same category ACs.

CSMA/ECA$_{\text{QoS+TXOP}}$ ACs rapidly converge to a collision-free operation with Smart Backoff. Results suggest that most of the time high priority ACs, like AC[VO] converge with larger schedules than other low priority ACs. This constitutes a priority inversion in terms of throughput, causing the high variability observed in the first row of Figure 5.10 for $N \leq 10$.

Looking at the bottom of Figure 5.10, CSMA/ECA$_{\text{QoS+TXOP}}$ clearly

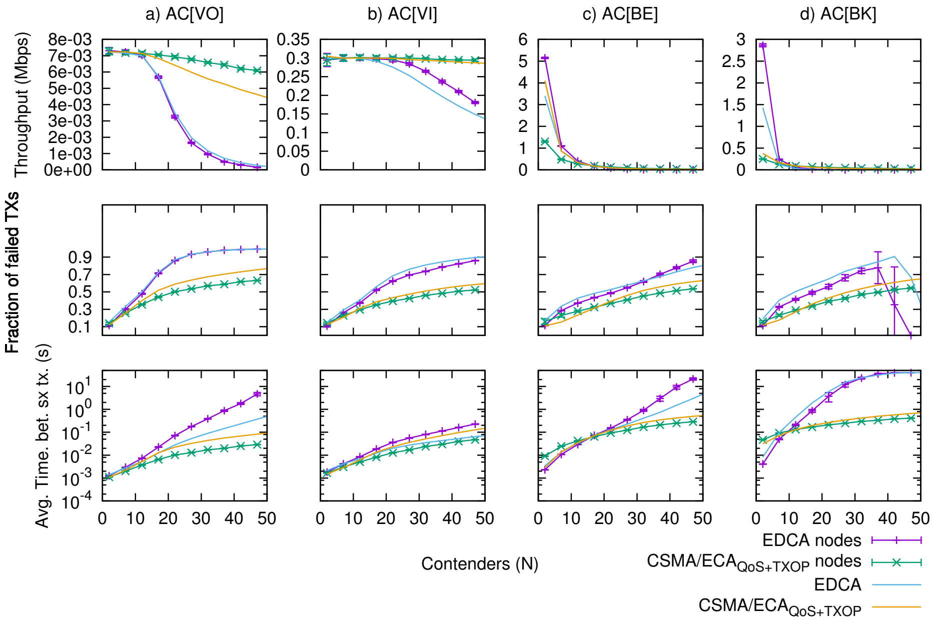

Figure 5.11: Comparison using CSMA/ECA$_{\text{QoS+TXOP}}$ in the Mixed Scenario in non-saturation

outperforms EDCA in the non-saturation scenario, besides, its average time between successful transmissions is practically equal to the one observed in Figure 5.8.3.

As Figure 5.9 in Chapter 5.4.2, the new Figure 5.11 shows a Mixed Scenario where 50% of nodes use EDCA, while the other 50% of nodes use CSMA/ECA$_{\text{QoS+TXOP}}$. The figure shows that the interaction among nodes with different protocols is pretty much the same as when using CSMA/ECA$_{\text{QoS+FS}}$.

Figure 5.12 shows a comparison between Fair Share and TXOP in CSMA/ECA$_{\text{QoS}}$. Column a) shows the saturation scenario, presenting throughput and JFI. b) presents the non-saturation scenario results, namely throughput and average time between successful transmissions. Results

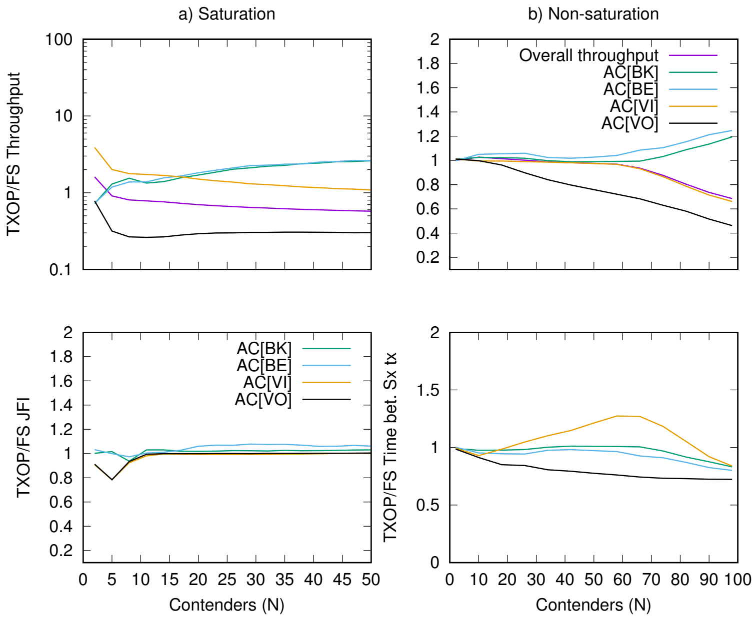

Figure 5.12: Comparison between using TXOP and Fair Share with CSMA/ECA$_{\text{QoS}}$

are normalised to CSMA/ECA$_{\text{QoS+FS}}$ values.

The throughput unfairness resulting from using TXOP is clearly appreciable in the saturated scenario for $N \leq 10$. Nevertheless, higher throughput is observed in CSMA/ECA$_{\text{QoS+TXOP}}$ AC[VI] due to the shorter TXOP[VO]. Figure 5.12-a suggests that the TXOP[VO] limitation allows low priority ACs to achieve higher throughput. This effect is also observed in the non-saturation scenario, referred by Figure 5.12-b. As nodes approach saturation, the shorter TXOP[VO] transmissions produce an overall reduction in the time between successful transmissions of other ACs.

5.4.4 Discussion

After the analysis, it is clear that the number of contenders, channel and traffic conditions play a main role in the performance of both MAC protocols.

A perfect channel, RTS/CTS, and low number of contenders are ideal conditions for EDCA in saturation. Nevertheless, CSMA/ECA$_{\text{QoS+TXOP}}$ ACs converge into collision-free schedules with different lengths. Despite Schedule Reset's efforts to reduce the schedule length, ACs rapidly reach collision-free operation and no further reduction is possible without introducing new collisions, producing the throughput oscillations observed in Figure 5.10 at $N \leq 10$. This issue is normally solved with Fair Share. Interestingly, using TXOP aggregation instead of Fair Share produces higher throughput for low number of nodes (despite the irregular throughput distribution), and as TXOP[VO] transmissions are shorter the overall delay of lower priority AC's transmissions is reduced when compared against Fair Share.

As scenarios become crowded, CSMA/ECA$_{\text{QoS+TXOP}}$ consistently outperforms EDCA (see Figure 5.10). Further, it shows lower fraction of failed transmissions for a considerably higher number of contenders. Failed transmissions and non-saturated sources keep changing the structure of Schedule Reset's bitmap, providing more opportunities to reduce the deterministic backoff. Finally, a priority inversion is observed at the bottom row of Figure 5.10, where EDCA AC[VI] shows lower average time between successful transmissions than AC[VO], which is almost starved due to the tight contention parameters.

CSMA/ECA$_{\text{QoS+TXOP}}$ results suggest it is better than EDCA for crowded scenarios, specially if:

- Traffic differentiation is to be ensured for high number of contenders.
- Transmissions from low priority ACs are not to be starved.
- To prevent AC priority inversions.

From the point of view of EDCA nodes in the mixed scenario, the deterministic backoff used by the other 50% of CSMA/ECA$_{\text{QoS+FS}}$ nodes during collision-free periods produce an increase in the number of empty slots. More empty slots imply lower probability of collisions. This means that sharing the network with CSMA/ECA$_{\text{QoS}}$ nodes reduces the collision probability for EDCA nodes. Therefore, the number of successful transmissions from low priority ACs is expected to be higher than in the EDCA-only scenario, increasing the time between successful transmissions of high priority ACs (as shown in Figures 5.9.3.a, 5.9.3.b, and Figures 5.11.3.a, 5.11.3.b).

From the point of view of CSMA/ECA$_{\text{QoS+FS}}$ nodes, the saturation point for AC[VO] and AC[VI] is moved to around $N' = 18$, matching EDCA's. Now being saturated, CSMA/ECA$_{\text{QoS+FS}}$ ACs are able to operate without collisions for a number of consecutive transmissions before colliding. This results in a reduction of the time between successful transmissions, coupled with a higher throughput when compared against the non-saturated homogeneous network scenario. The latter still being non-saturated at the same $N = N'$.

Chapter 6

FROM CONCEPTS TO WORKING PROTOCOLS

Even-though the IEEE 802.11 set of WLAN standards define the procedures to guarantee effective communication among hosts, the implementation part is the task of manufacturers. This means that *how* the standard is implemented vary from vendor to vendor and explains why firmware is closely related to the underlying hardware. Current efforts both from the industry and the open source community (as in the case of MadWiFi [71] driver and OpenFWWF [27,73] firmware), created the opportunity to prototype and test new MAC protocols proposals on cheap commodity hardware.

Several manufacturers embraced the Soft-MAC [47] approach for interconnecting their WiFi Network Interface Cards (NICs) with general purpose systems. A dedicated CPU on the NIC controls the radio circuitry and pulls complete 802.11 frames prepared by the main Operating System (OS) kernel from an interconnecting bus, e.g., a PCI bus, and schedules their transmission in real time. Thanks to this approach, the NIC offloads the time-critical actions related to the channel access, while the main kernel controls all other functionalities. The CPU on the NIC runs the MAC algorithm by executing a software (the firmware from here on) that reacts to transmission/reception history and drives the evolution

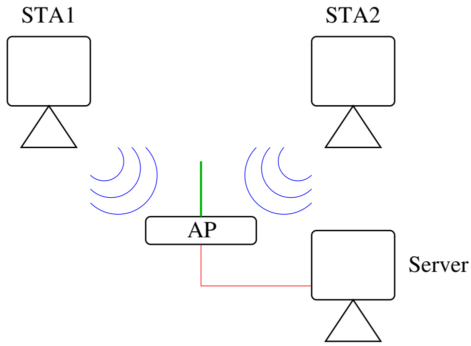

Figure 6.1: Initial testing scenario setup

of the Contention Window and the Backoff Counter. By replacing the firmware, one can deeply customise the MAC or even switch to a different one, e.g., Time Division Multiple Access (TDMA) [73], instead of CSMA/CA.

6.1 Departing from DCF

One of the main advantages of CSMA/ECA in terms of implementation is that it does not deviate too much from the current MAC. This allows the use of open firmware that already contains the base code for CSMA/CA to be modified towards CSMA/ECA. This is achieved modifying some functions of the open sourced OpenFWWF firmware.

An initial prototype of CSMA/ECA [7] instructs nodes to pick a deterministic backoff, $B_\mathrm{d} \leftarrow \lceil \mathrm{CW}_\mathrm{min}/2 \rceil - 1$ after successful transmissions. The firmware was modified to execute a procedure which is very similar to the proposed protocol in saturated conditions. The open source OpenFWWF firmware is used in combination with the `b43` Linux wireless card driver, which in turn is supported by a limited set of Broadcom cards [44].

A simple testing scenario was built in order to check whether the modifications performed matched the expected CSMA/ECA behaviour [41]. This was composed of two Ubuntu 8.10 PCs with Broadcom BCM4318 cards running OpenFWWF firmware as WLAN STAtions (STA): STA1

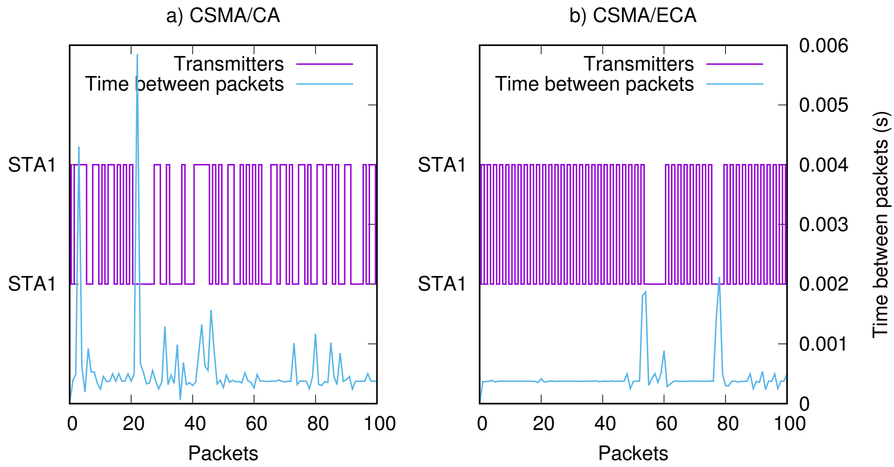

Figure 6.2: a) CSMA/CA and b) CSMA/ECA transmission turns between STA1 and STA2

with Intel Pentium 4 3 GHz and 768 MB of RAM; and STA2 with Intel Core 2 Quad 2.66 GHz and 3 GB of RAM, both connected to a Linksys WAG354G Access Point (AP). To make performance tests, Iperf [74] tool generates 1470B UDP datagrams at 65 Mbps from both STAs to a Server wired to the AP using Ethernet, effectively saturating each STA. At the Server, Wireshark [17] captures all packets from the STAs. Figure 6.1 provides an overview of the testing scenario.

The aim of the initial test is to reveal evidence of the deterministic backoff counter. Figure 6.2 show a random set of a hundred server-received packets from STAs running a) CSMA/CA and b) CSMA/ECA. CSMA/CA's randomised backoff mechanism can be appreciated in Figure 6.2a, where the "Transmitters" line shows how the transmitter of a given packet could be either of the contending stations. Whereas in CSMA/ECA (Figure 6.2b), transmitters almost alternate transmissions.

From the "Time between packets" curve, we can see that the average time between two consecutive packets (seen from the channel perspective) is greater for CSMA/CA nodes. This is due to the random backoff mechanism, which is prone to collisions and extends the period between successful transmissions. CSMA/ECA nodes on the other hand are able to construct collision-free schedules, lowering the average period between successful transmissions.

6.2 Scheduling transmissions very precisely with Collision-Free MAC (CF-MAC)

Previous experimental studies, like [41, 56, 59], show that collision-free operation with OpenFWWF and CSMA/ECA can be achieved only for high values of the deterministic backoff. In particular, when using short deterministic backoff values, like 8, 16 or 32 slots, stations fail to maintain a collision-free operation for the length of the experiments, whether due to lack of time precision or misinterpretation of the state of the channel before transmission (caused by an imperfect Clear Channel Assessment (CCA) mechanism).

In order to ensure precision in the scheduling mechanism a more accurate set of instructions is implemented at firmware level [61]. These modifications make use of a continuous timer to schedule transmissions instead of a backoff based on discrete slots. Further, possible problems with the CCA mechanism are avoided by sensing the channel for a period equivalent to only two empty slots before the scheduled transmission.

OpenFWWF implements a simple State Machine (SM) for controlling the hardware in real time. The SM evolution is driven by a main loop that reacts to events by executing specific handlers[1]. When a packet, originally prepared by the Linux kernel, is ready in the NIC memory, handler `packet_ready` sets up the radio hardware according to the packet meta data (e.g., it fixes rate, modulation format, and power level), schedules the transmission and jumps back to the main loop. Then, the Transmission Engine (TXE) takes care of accessing the channel, i.e., it decrements the Backoff Counter (B) according to the Distributed Channel Function

[1]In the following we consider only the limited subset of events that were changed for the implementation.

(DCF) rules and it eventually starts the actual transmission. This triggers the execution of the `tx_frame_now` event that prepares the ACK time-out clock and finalises the MAC header[2]. If the ACK-frame is received or if the ACK time-out expires and the maximum number of attempts for this packet is reached, handler `update_params` resets the Contention Window to the minimum (CW_{min}), otherwise it doubles the current CW; finally loading the B counter with a fresh value.

Our implementation could not exploit the TXE engine, i.e., by loading a timer which value is proportional to a precomputed delay time (duration of a transmission plus the effective reception of the ACK). All busy channel episodes, in fact, would temporary stop the timer countdown and nodes would quickly go out-of-sync, as experimented in [41, 56, 59]. To avoid these unpredictable backoff inaccuracies we exploited another feature as in [11] that allows the firmware to start the immediate transmission of a frame, independently of the channel conditions (we call this feature the Tx_{now} instruction from this point onwards). We hence added to the main loop a single compare and jump instruction that checks if the internal clock exceeded a precomputed instant; however, instead of starting the transmission unconditionally, we check the busy channel indicator, in particular we defer until the channel has been idle for at least two slots to avoid collisions with ongoing transmissions.

[2]As transmission has already started, these actions must be completed before the physical preamble end.

```
while the device is on do
    ret ← 0 ; k ← 0;
    B ← U[0, 2^k CW_min − 1];
    while there is a packet to transmit do
        repeat
            while B > 0 do
                wait 1 slot;
                B ← B − 1;
            Attempt transmission of 1 packet;
            if collision then
                ret ← ret + 1;
                k ← min(k + 1, m);
                B ← U[0, 2^k CW_min − 1];
        until (ret = R) or (success);
        r ← 0;
        k ← 0;
        if ret = R then
            Discard packet;
        else
            repeat
                wait T_c(N, r) μs;
                Attempt transmission of 1 packet;
            until (collision);
        B ← U[0, 2^k CW_min − 1];
    Wait until there is a packet to transmit;
```

Algorithm 6: Overview of the precise transmission packet scheduling mechanism, CF-MAC

6.2.1 Implications of using the $\mathbf{Tx_{now}}$ instruction

The modifications required to make use of the Tx_{now} instruction have the following effects:

- It supposes a modified use of the carrier sense algorithm. That is, the node only listens to the channel for a short period of at least two empty slots before attempting the transmission.

- For using the Tx_{now} instruction, the slotted-time backoff approach could not be used. Therefore, we create a timer based on real time (measured in μs).

- The use of a timer ($T_{\text{c}}(N, r)$) is subject to the number of contenders (N) and the minimum data transmission rate in the network (r, in bps). The proposed definition of $T_{\text{c}}(N, r)$ is shown in (6.1), where P is the number of bits sent for each transmission attempt, $SIFS$ is the 802.11g Short Inter-Frame Spacing and $ACK_{\texttt{rx}}(r)$ is the duration of an ACK reception.

$$T_{\text{c}}(N, r) = N \left(\frac{P}{r} + SIFS + ACK_{\texttt{rx}}(r) + \epsilon \right) \quad (6.1)$$

  So, $T_{\text{c}}(N, r)$ basically is the duration of a transmission at a certain rate (r) and the reception of an ACK plus a guard interval (ϵ), multiplied by the number of contenders in the network (N). Notice that $T_{\text{c}}(N, r)$ supposes a previous knowledge of the number of contenders, making the obtained results useful for research purposes but with little practical use in real world WLANs.

Algorithm 6 shows an example of the proposed implementation, referred to as Collision-Free MAC (CF-MAC) from hereon. Basically, stations substitute the random backoff B by the $T_{\text{c}}(N, r)$ timer after a successful transmission. Successful nodes will continue to attempt transmission after $T_{\text{c}}(N, r)$ μs until a collision is detected (this process is detailed at Line 21), after which a random backoff is drawn and the node goes

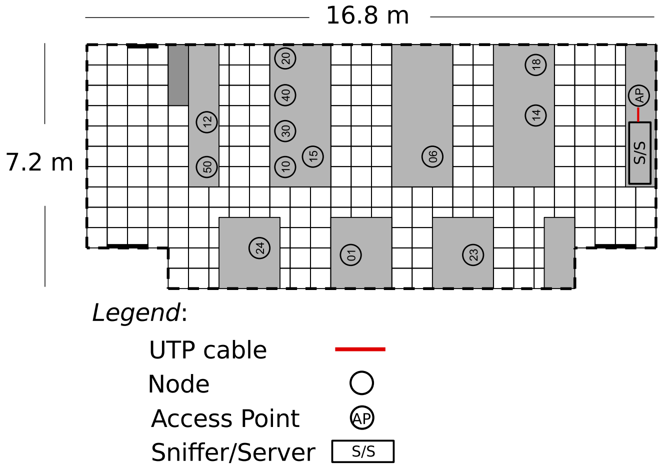

Figure 6.3: CF-MAC testbed

back to CSMA/CA operation.

6.2.2 Checking the precision

Each node used for the testing of CF-MAC is equipped with a commercial WiFi card compatible with both OpenFWWF and the `b43` driver. The modified firmware was loaded into the testing nodes which were arranged mimicking a conventional workspace environment: placed at different distances from an AP and using a free WiFi channel in order to avoid external interferences from other networks. Figure 6.3 is a graphic representation of the nodes' layout, while Table 6.1 gathers the PHY and MAC settings used.

Upon each test, the rate of each station is fixed and an iPerf [74] session is stablished with the Server/Sniffer (S/S). Each node then is saturated (is always attempting to transmit) for a period of ninety seconds. Transmissions are then captured by the S/S using TCPdump [35]. S/S is represented in Figure 6.3 as a single station, connected via Ethernet to the Access Point (AP).

Table 6.1: PHY and MAC parameters for the CF-MAC testbed

PHY	
Parameter	**Value**
PHY rate (Mbps)	$6, 11, 12, 24, 48$
Empty slot (μs)	9
DIFS (μs)	28
SIFS (μs)	10
MAC	
Parameter	**Value**
Maximum backoff stage (m)	5
Minimum Contention Window ($CW_{\min}$)	16
Maximum retransmission attempts	6
Packet size (Bytes)	1470
Duration of each test (s)	90

It is possible to derive several metrics by analysing the captured transmissions files and reading different counters setup at the firmware, like:

- Throughput per station: by looking at the log of each iPerf session, it is possible to obtain an estimation of the achieved throughput of each station. Further, by looking at the number of successfully sent packets (counted by the firmware) a measure of throughput can also be derived.

- Inter-arrival time: is the time between the transmission of two frames by the same station. This metric reflects the time invested in the contention mechanism.

- Fraction of lost frames: each time a station attempts a retransmission, the result of the previous transmission is counted as a failure. Knowing the number of failures and the total number of transmission attempts, a fraction of lost frames can be computed.

When comparing both protocols it is useful to look at the achieved

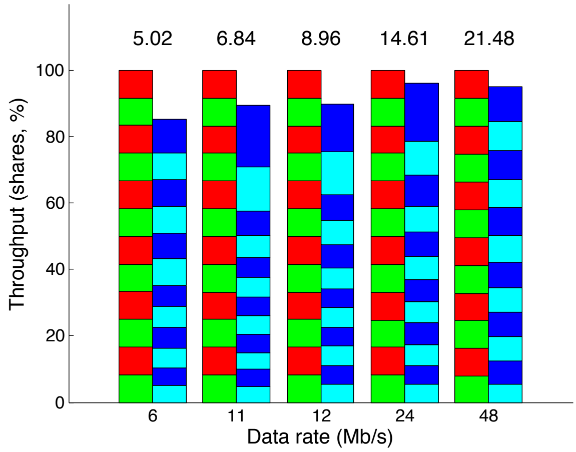

Figure 6.4: Throughput for different data rates using CF-MAC

throughput, but also at how the available bandwidth is distributed among the contenders.

Throughput

Figure 6.4 shows ten independent experiments with increasing data rates and twelve nodes each. For each x-point, two separate experiments are shown, one for a network composed of only CSMA/CA nodes (right bar) and one for nodes loaded with our prototype (left bar). Each bar is divided in boxes which represent the throughput share of a station.

Given that stations using our prototype are able to construct a collision-free schedule using the $T_{\mathrm{c}}(N, r)$ timer after a successful transmission, the channel is used more efficiently. Whereas CSMA/CA stations waste time recovering from collisions and contenting for the channel.

We can see that the CSMA/CA network achieves less cumulative throughput. Further, the throughput is not evenly distributed among the contenders, as shown in Figure 6.5. The figure shows the min/max throughput ratio for network setups with increasing number of nodes and different rates. CSMA/CA is referenced in the legend as /CA, followed by the

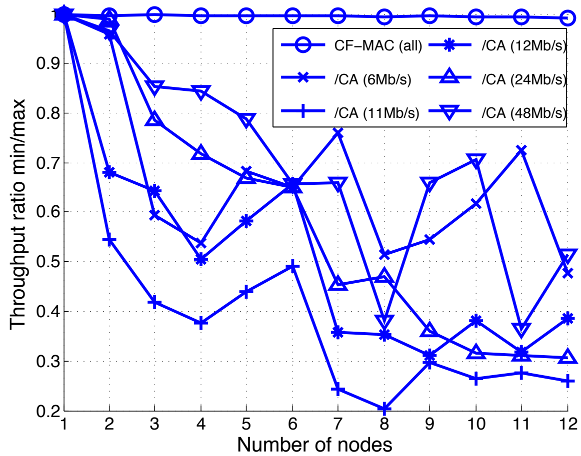

Figure 6.5: Min/Max throughput ratio using CF-MAC

rate at which the experiment was performed. With any number of nodes and all tested rates, our prototype shows that the throughput is efficiently shared among contenders, whereas different CSMA/CA network setups show an uneven distribution of the available throughput.

Inter-arrival Times

CSMA/CA nodes pick B randomly and freeze it when a transmission is being performed in the channel, which translates in a variable inter-arrival time; while CF-MAC stations schedule transmissions according to the predefined timer ($T_{\mathrm{c}}(N, r)$). This is made evident by Fig. 6.6. For each X-axis point in the figure the left boxes represent CF-MAC stations, while the right circles are CSMA/CA's; the hovering numbers represent the CF-MAC average in milliseconds. Middle points represent the average among all CSMA/CA nodes, while higher and lower circles represent the maximum and minimum respectively.

In the figure, the time between consecutive transmissions varies considerably more for CSMA/CA than for CF-MAC. This suggests that CSMA/CA

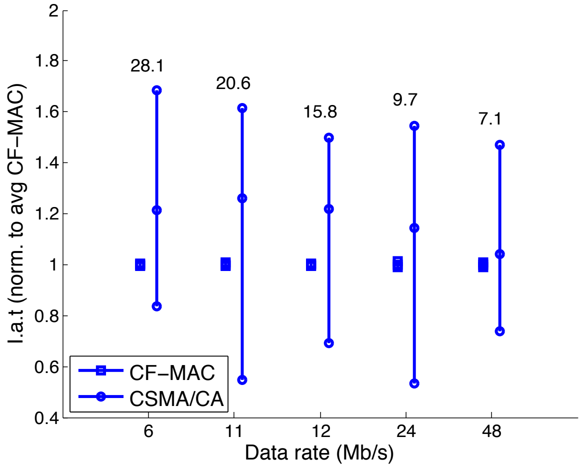

Figure 6.6: Inter-arrival Times (normalized to the average of CF-MAC)

nodes on average spend more time in contention and recovering from collisions, also contributing to the throughput degradation.

Lost Frames

The modifications made to the firmware also included the incorporation of counters, which were allocated in free segments in the card's memory. To derive a measure of the average losses per node we counted the number of failed transmissions (indicated by the lack of reception of an ACK), successful transmissions (when an ACK is received) and the number of transmission attempts. Figure 6.7 shows the average losses for CSMA/CA and CF-MAC alongside a reference curve derived from the model in [13].

CSMA/CA stations suffer from a increased number of collisions, mostly due to the randomness of the backoff mechanism; whereas CF-MAC enjoys a much more reduced number of collisions due to the implementation of the deterministic timer, $T_c(N, r)$ after successful transmissions.

In Figure 6.7, at higher rates (24, 48 Mb/s) the average losses seem to be reduced. This effect can be caused by a defective CCA mechanism on

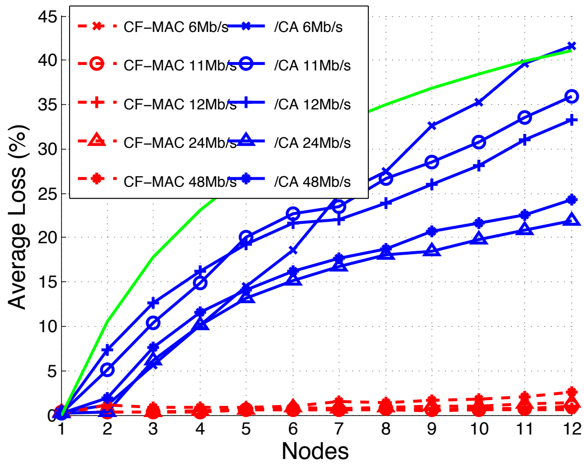

Figure 6.7: Fraction of losses of CF-MAC

the cards. Transmissions at these rates are shorter, so transmitters are less prone to make erroneous inferences about the channel state. On the other hand, stations at lower rates should listen to the channel for longer periods of time before attempting transmission, thus increasing the probability of a misinterpretation of the channel state.

Not suitable for real world implementations

CF-MAC is able to construct Collision-Free schedules by means of using a deterministic timer after successful transmissions, which allows a better use of the available channel time in WLANs.

Using a precise schedule for transmissions allows CF-MAC to greatly reduce the fraction of collisions in comparison with CSMA/CA. Further, this reduction of wasted channel time recovering from collisions or spent in contention is reflected in a better distribution of the available bandwidth among contenders. Moreover, by using a deterministic timer after successful transmissions stations greatly reduce the variability of time between transmissions attempts, making it a suitable technique for delay-sensitive communications.

CF-MAC was tested in a real testbed, using off-the-shelf hardware and a modified firmware for the wireless cards. Many real-world un-ideal conditions, like the performance of the CCA implementation or impractical assumptions like previous knowledge of the number of contenders for setting the deterministic timer prevent this specific implementation from being an adequate MAC protocol for WLANs.

Despite the above, CF-MAC shows that it is possible to reduce the number of collisions and increase the channel efficiency by attempting to construct collision-free schedules.

6.3 Implementing CSMA/ECA$_{\text{Hys}}$ and Schedule Reset in real hardware

We built on the DCF protocol implemented in OpenFWWF and adapted the firmware to create collision-free schedules as described in Algorithm 2.

The modification was straightforward: for every transmitted data frame the firmware sets an ACK-time-out alarm. Later, either at the expiration of the timer or when the acknowledgment frame is received, it executes a handler labelled `tx_contention_params_update`. It updates the contention window value `STATE_CW` and backoff counter according to the success/failure of the previous transmission attempt, just as in Algorithm 2. Implementing CSMA/ECA$_{\text{Hys}}$ required just a modification specifying that a reset of the backoff stage (or `STATE_CW` in this case) is performed only when a packet is dropped or when the MAC queue empties. Contrary to CF-MAC, CSMA/ECA$_{\text{Hys}}$ implementations do not use a predefined timer.

To incorporate stickiness into the prototype we added a `STATE` to the system that can be either `STATE_DETERMINISTIC` or `STATE_RANDOM` (related to the type of backoff being used), and a `STICKINESS` counter that we reset each time we have a success and decrement when failing. When the counter gets to zero we enter the random state. Upon a successful transmission we unconditionally enter the deterministic state and reset

Table 6.2: PHY, MAC and other parameters for the CSMA/ECA$_{\text{Hys}}$ implementation

PHY	
Parameter	**Value**
PHY rate	48 Mbps
Empty slot	9 μs
DIFS	28 μs
SIFS	10 μs
IEEE 802.11g WiFi channel	14
MAC	
Parameter	**Value**
Maximum backoff stage (m)	5
Minium Contention Window (CW_{min})	16
Maximum retransmission attempts	6
Data payload (Bytes)	1470
Schedule Reset γ	1
Schedule Reset mode	halving, dynStick
Default stickiness	1
TESTBED	
N	25
Distance between nodes and AP	8 m
Arrangement of nodes	Semicircle

the stickiness counter to `DEFAULT_STICKINESS`.

For the Schedule Reset mechanism, we added a bitmap for monitoring the state of every single slot after a successful transmission. To index the current slot in the bitmap we used the hardware register called `SPR_IFS_BKOFFDELAY`, which counts how many slots have still to come before the next transmission. The value of the register is decremented once per idle slot.

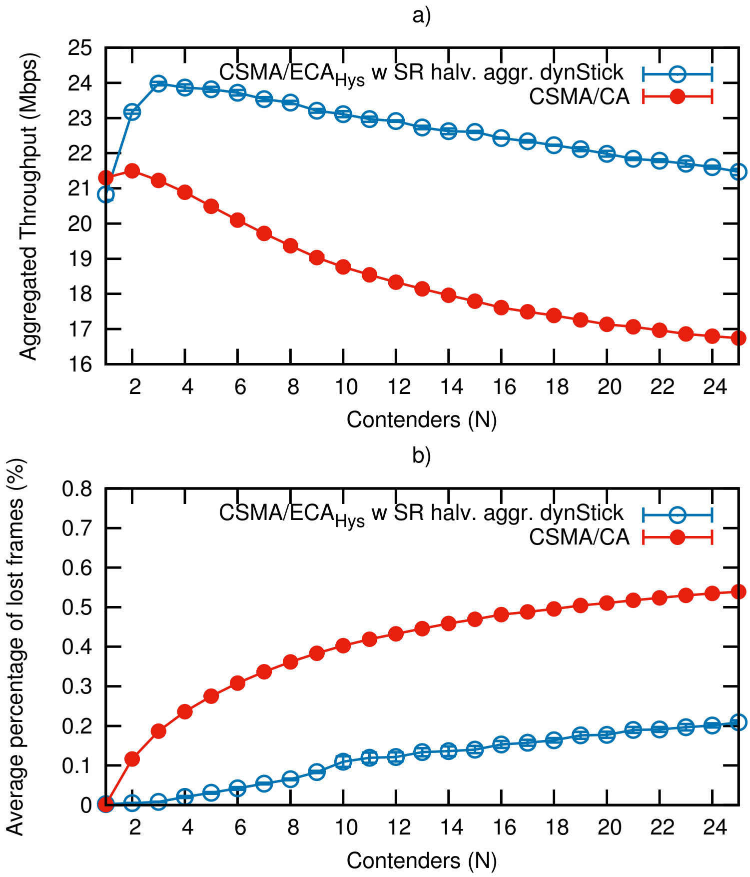

Figure 6.8: Implementation results

To avoid spurious detection, instead of continuously checking the state of the channel, we rely on the execution of the `rx_plcp` handler, which is called each time a valid PLCP is detected. When this happens, we implicitly know that the backoff counter has been frozen at least $20\mu s$ ago, which is the time between the detection of the first short trailing sequence and the complete decoding of the PLCP signal data. In any case, the slot is marked as busy.

We have the option to keep filling the bitmap for up to a defined number of consecutive successful transmissions, BITMAP_ROUND_BUILD (γ in Chapter 3.1.6) before checking if the central slot in the bitmap is available. If that is the case, the node's current schedule is halved (a schedule *halving*, following Algorithm 4) and its stickiness is incremented to DEFAULT_STICKINESS$+1$.

We performed four experiments for each tested number of contenders, starting from 1 and up to 25. For every experiment, each station establishes an iPerf [74] session and transmits saturated UDP traffic towards a central AP, which also functions as iPerf server for each flow. We used WiFi channel 14 in order to avoid interference from other networks. The aggregate throughput is derived from the iPerf logs, while the percentage of lost frames is reported by the firmware. This is known to be $(\text{tx} - \text{sx})/\text{tx}$; where tx are the number of transmission attempts, and sx are the number of acknowledged frames.

Figure 6.8a shows the average aggregate throughput, while Figure 6.8b presents the average percentage of lost frames. Details of the testbed are shown in Table 6.2.

CSMA/ECA$_{\text{Hys}}$ has greater throughput due to its ability to avoid collisions more efficiently than CSMA/CA. Further, the Schedule Reset aggressiveness prevents CSMA/ECA$_{\text{Hys}}$ nodes from increasing too much the time between successful transmissions.

Nevertheless, the implementation results reveal that there are other underlying factors that disrupt collision-free schedules. This is evidenced by the increasing percentage of lost frames followed by a throughput degradation in CSMA/ECA$_{\text{Hys}}$. As mentioned before, these factors include CCA imperfections that mistakenly interpret the channel as empty, allowing the node to transmit while other transmissions may be on course. Causing a collision.

Chapter 7

CONCLUSIONS

Coordinating access to the channel in every-day WiFi is done in a completely decentralised manner. That is, techniques to avoid errors count on local information only. Using a random backoff technique following CSMA/CA has been the de facto MAC protocol for WiFi, being relatively easy to implement by manufacturers, as well as up to the firmware and driver level in open source systems.

This decentralised nature and backwards compatibility has championed the spread of CSMA/CA. With time IEEE 802.11 amendments proposed mechanisms to circumvent the performance degradation produced by collision events. For instance, a consequence of using the RTS/CTS mechanism is a reduction in the amount of time wasted recovering from collisions, so the overall throughput can be increased in spite of larger overheads (particularly in presence of few users).

Similarly, a great amount of research work can be found regarding parameter adjustments in CSMA/CA, and distributed estimation of current number of contenders in the same WLAN. Despite the insight and improvements derived from such amendments, the core channel access procedure is still based on a random backoff. That is, even with the very high throughput provided at the PHY layer, collision events render the MAC as a bottleneck in terms of throughput, especially at high number of users per WLAN.

We have shown that using a deterministic backoff after successful transmissions is a better collision avoidance technique for WiFi. Furthermore, the resulting collision-free schedules can be adjusted so more nodes can take advantage of it. Throughput fairness issues arising from the uneven distribution of periods between transmissions are solved with an AMPDU aggregation technique.

The resulting protocol, CSMA/ECA, as it provides important improvements with high number of contenders is then tested under overlapping BSS scenarios. Where contention for the channel spans several neighbouring WLANs. Again, results show that the deterministic backoff after successful transmissions technique is better for all the tested scenarios, even those considered for the upcoming IEEE 802.11ax.

Traffic differentiation in WLANs using multiple CSMA/ECA queues provides considerable advantages in crowded scenarios, like outperforming EDCA. As the latter struggles with collisions at high number of contenders, multiple CSMA/ECA queues are specially beneficial. Nevertheless, we encountered that imposing limitations, like TXOP alters the throughput fairness of CSMA/ECA.

We identified that in order to extract a good estimate of the performance of a protocol is required to use realistic traffic source models, especially to simulate non-saturated traffic, and/or video and audio.

Simulation environments can only take us so far. With the aim of pushing CSMA/ECA virtues to the test many real hardware implementations were built. First, the plausibility of changing the time-sensitive backoff procedure in an off-the-shelf WiFi card was confirmed using open source firmware. Then, the transmission instructions were directly modified in order to accurately mimic CSMA/ECA behaviour. Again, real hardware implementations results using a 25 node testbed showed higher throughput for CSMA/ECA.

At this point, CSMA/ECA represents a completely decentralised and *backwards compatible alternative to CSMA/CA. Able to provide performance improvements in very crowded scenarios.*

Nevertheless, the efforts made towards maintaining compatibility and dealing with realistic traffic sources revealed the unsuitability of the one-fits-all approach of MAC protocols for WiFi.

The ever increasing requirements from applications, as well as the crowded scenarios envisioned for the future of WiFi challenge the very foundations of the MAC, suggesting a more flexible and reconfigurable approach to leverage current problems. This way MAC protocol designers can direct attention to specific requirements, like QoS, user load balancing, fairness or crowded scenarios.

We have noticed that many problems such as backwards compatibility and QoS can be leveraged by over-the-air MAC protocol reconfigurability.

Allowing more efficient MAC protocol design by means of open hardware, Software Defined strategies for WiFi, and Wireless MAC Processors has the potential to change the perspective from which current amendments are proposed. Promoting innovation and a more systemic view of the MAC resources, problems, requirements and limitations.

It is thought that the evolution of MAC protocols for WiFi should no longer focus of compatibility issues, nor on the principle of being completely distributed. Current demands and envisioned scenarios call for MAC protocols capable of having a systemic view of the network, being able to adapt to changing conditions and requirements. Therefore it is thought that allowing over-the-air MAC protocol reconfiguration, as well as supporting development of WiFi-specific Software Defined Network abstractions can serve as a better platform for the discovery of solutions to current challenges. Making it even easier to come up with a concept and turn it into a working protocol.

List of Publications

The contributions of this thesis have been already published, or are submitted for publication.

Journal submissions

1. **Sanabria-Russo, L.**, Barcelo, J., Bellalta, B., and Gringoli, F. (2014). A High Efficiency MAC Protocol for WLANs: Providing Fairness in Dense Scenarios. *Accepted manuscript to IEEE Transactions on Networking (ToN) journal* [57].

2. **Sanabria-Russo, L.**, and Bellalta, B. (2015). Traffic Differentiation in Dense Collision-free WLANs, *Submitted* [55].

3. **Sanabria-Russo, L.**, Bellalta, B., Gringoli, Facchi, N., and Gringoli, F. (2016). Collision-free Operation in High Density WLAN Deployments. *Submitted*.

Conference and Workshops

1. **Sanabria-Russo, L.**, Barcelo, J., Domingo, A., and Bellalta, B. (2012, November). Spectrum Sensing with USRP-E110. In International Workshop on Multiple Access Communications (pp. 79-84). Springer Berlin Heidelberg [58].

2. **Sanabria-Russo, L.**, Faridi, A., Bellalta, B., Barcelo, J., and Oliver, M. (2013, June). Future evolution of CSMA protocols for the IEEE 802.11 standard. In 2013 IEEE International Conference on Communications Workshops (ICC) (pp. 1274-1279) [60].

3. **Sanabria-Russo, L.**, Barcelo, J., and Bellalta, B. (2013, December). Prototyping Distributed Collision-Free MAC Protocols for WLANs in Real Hardware. In International Workshop on Multiple Access Communications (pp. 82-87). Springer International Publishing [56].

4. **Sanabria-Russo, L.**, Barcelo, J., Faridi, A., and Bellalta, B. (2014, April). WLANs throughput improvement with CSMA/ECA. In IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 125-126) [59].

5. **Sanabria-Russo, L.**, Gringoli, F., Barcelo, J., and Bellalta, B. (2015, June). Implementation and experimental evaluation of a collision-free MAC Protocol for WLANs. In IEEE International Conference on Communications (ICC) (pp. 1036-1042) [61].

Bibliography

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

[2] IEEE Standard for Information Technology–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 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)*, pages 1–212, Nov 2005.

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

[4] O. Acholem and B. Harvey. Throughput Performance in Multihop Networks using Adaptive Carrier Sensing Threshold. In *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)*, pages 287–291, March 2010.

[5] M. S. Afaqui, E. Garcia-Villegas, E. Lopez-Aguilera, G. Smith, and D. Camps. Evaluation of Dynamic Sensitivity Control algorithm for IEEE 802.11ax. In *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1060–1065, March 2015.

[6] Steven Andersen, Alan Duric, Henrik Astrom, Roar Hagen, W Kleijn, and Jan Linden. Internet Low Bit Rate Codec (iLBC). Technical report, 2004.

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

[8] 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.

[9] Boris Bellalta. IEEE 802.11 ax: High-Efficiency WLANs. *IEEE Wireless Communications (arXiv preprint arXiv:1501.01496)*, Accepted July 2015.

[10] Boris Bellalta, Michela Meo, and Miquel Oliver. A BEB-based Admission Control for VoIP calls in WLAN with coexisting elastic TCP flows. In *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, pages 130–141. Springer, 2006.

[11] Daniel Berger, Francesco Gringoli, Nicolò Facchi, Ivan Martinovic, and Jens Schmitt. Gaining Insight on Friendly Jamming in a Real-world IEEE 802.11 Network. In *Proceedings of the ACM Conference on Security and Privacy in Wireless & Mobile Networks*, 2014.

[12] G. Bianchi and I. Tinnirello. Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 Network. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 844–852 vol.2, March 2003.

[13] Giuseppe Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[14] Giuseppe Bianchi and Ilenia Tinnirello. One size hardly fits all: towards context-specific wireless mac protocol deployment. In *Proceedings of the 8th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, pages 1–8. ACM, 2013.

[15] Cristina Cano, Boris Bellalta, Anna Sfairopoulou, and Jaume Barceló. Tuning the EDCA parameters in WLANs with heterogeneous traffic: A flow-level analysis. *Computer Networks*, 54(13):2199–2214, 2010.

[16] Chulho Chung, Yunho Jung, and Jaeseok Kim. Saturation Throughput Analysis of IEEE 802.11 ac TXOP sharing mode. *Electronics Letters*, 51(25):2164–2166, 2015.

[17] Combs, Gerald and others. Wireshark, 2007. Available at: http://www.wireshark.org.

[18] Jing Deng, Ben Liang, and Pramod K Varshney. Tuning the carrier sensing range of IEEE 802.11 MAC. In *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, volume 5, pages 2987–2991. IEEE, 2004.

[19] Minyu Fang, David Malone, Ken R. Duffy, and Douglas J. Leith. Decentralised learning MACs for collision-free access in WLANs. *Wireless Networks*, 19:83–98, 2013.

[20] Michelle X Gong, Brian Hart, and Shiwen Mao. Advanced Wireless LAN Technologies: IEEE 802.11 ac and beyond. *GetMobile: Mobile Computing and Communications*, 18(4):48–52, 2015.

[21] Wunan Gong and David Malone. Addressing Slot Drift in Decentralized Collision Free Access Schemes for WLANs. In Boris Bellalta, Alexey Vinel, Magnus Jonsson, Jaume Barcelo, Roman Maslennikov, Periklis Chatzimisios, and David Malone, editors, *Multiple Access Communications*, volume 7642 of *Lecture Notes*

in Computer Science, pages 146–157. Springer Berlin Heidelberg, 2012.

[22] David J Goodman, Reinaldo A Valenzuela, KT Gayliard, and B Ramamurthi. Packet Reservation Multiple Access for Local Wireless Communications. *IEEE Transactions on Communications*, 37(8):885–890, 1989.

[23] Graham Smith. Dynamic Sensitivity Control for HEW. Webpage, accessed Jun 2016, 2014.

[24] Graham Smith. DSC Calibration Results with NS-3. Webpage, accessed Jun 2016, 2015.

[25] Graham Smith. Indoor Enterprise Scenarios, Color, DSC and TPC. Webpage, accessed Jun 2016, 2016.

[26] Graham Smith. TG ax Enterprise Scenario, TPC and DSC. Webpage, accessed Jun 2016, 2016.

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

[28] L. Hanzo, H. Haas, S. Imre, D. O'Brien, M. Rupp, and L. Gyongyosi. Wireless Myths, Realities, and Futures: From 3G/4G to Optical and Quantum Wireless. *Proceedings of the IEEE*, 100(Special Centennial Issue):1853–1888, 2012.

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

[30] IEEE. Proposed TGax draft specification. doc.: IEEE 802.11-16/0024r0. Technical report, IEEE, 2016.

[31] IEEE 802.11 TGax. Simulation scenarios for IEEE 802.11 HEW. https://mentor.ieee.org/802.11/dcn/14/11-14-0980-16-00ax-simulation-scenarios.docx, 2014.

[32] IEEE 802.11 TGax. Status of Project IEEE 802.11ax High Efficiency WLAN (HEW). http://www.ieee802.org/11/Reports/tgax_update.htm, 2016.

[33] Antonio Iera, Giuseppe Ruggeri, and Domenico Tripodi. Providing Throughput Guarantees in 802.11e WLAN Through a Dynamic Priority Assignment Mechanism. *Wireless Personal Communications*, 34(1-2):109–125, 2005.

[34] Takeshi Itagaki, Yuichi Morioka, Masahito Mori, Koichi Ishihara, Shoko Shinohara, and Yasuhiko Inoue. Performance Analysis of BSS Color and DSC, 2015. Available at: https://mentor.ieee.org/802.11/dcn/15/11-15-0045-00-00ax-performance-analysis-of-bss-color-and-dsc.pptx.

[35] Van Jacobson, Craig Leres, and Steven McCanne. Tcpdump public repository. *Web page at* `http://www.tcpdump.org`, 2003.

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

[37] Irfan Jamil, Laurent Cariou, and Jean-Francois Helard. Improving the capacity of future IEEE 802.11 high efficiency WLANs. In *Telecommunications (ICT), 2014 21st International Conference on*, pages 303–307. IEEE, 2014.

[38] Beakcheol Jang and Mihail L Sichitiu. IEEE 802.11 Saturation Throughput Analysis in the Presence of Hidden Terminals. *IEEE/ACM Transactions on Networking (TON)*, 20(2):557–570, 2012.

[39] Adlen Ksentini, Abdelhak Guéroui, and Mohamed Naimi. Adaptive Transmission Opportunity with Admission Control for IEEE 802.11e Networks. In *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '05, pages 234–241, New York, NY, USA, 2005. ACM.

[40] Parag Kulkarni and Fengming Cao. Taming the Densification Challenge in Next Generation Wireless LANs: An Investigation into the Use of Dynamic Sensitivity Control. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on*, pages 860–867. IEEE, 2015.

[41] L. Sanabria-Russo. Report: Prototyping Collision-Free MAC Protocols in Real Hardware. Webpage, 2013. Available at: http://luissanabria.me/written/beca-test1.pdf.

[42] Jiwoong Lee and Jean C Walrand. Design and Analysis of an asynchronous Zero Collision MAC protocol. *arXiv preprint arXiv:0806.3542*, 2008.

[43] 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*, pages 348–351, 2001.

[44] Linux Wireless. b43 and b43legacy. Webpage accessed July 2013. Available at: http://wireless.kernel.org/en/users/Drivers/b43.

[45] A. Lopez-Toledo, T. Vercauteren, and X. Wang. Adaptive Optimization of IEEE 802.11 DCF Based on Bayesian Estimation of the Number of Competing Terminals. 5(9):1283, 2006.

[46] Michael Menth, Andreas Binzenhöfer, and Stefan Mühleck. Source Models for Speech Traffic Revisited. *IEEE/ACM Transactions on Networking (TON)*, 17(4):1042–1051, 2009.

[47] Michael Neufeld, Jeff Fifield, Christian Doerr, Anmol Sheth, and Dirk Grunwald. Softmac-flexible wireless research platform. In *Proc. HotNets-IV*, 2005.

[48] 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)*, pages 1–9, June 2008.

[49] NS-3 Project. Buildings Module. https://www.nsnam.org/docs/release/3.14/models/html/buildings.html, 2011.

[50] Eldad Perahia and Robert Stacey. *"Next Generation Wireless LANs: 802.11 n and 802.11 ac"*. Cambridge University Press, 2013.

[51] Perahia, E. IEEE 802.11n Development: History, Process, and Technology. *Communications Magazine, IEEE*, 46(7):48–55, 2008.

[52] George F Riley and Thomas R Henderson. The NS-3 network simulator. In *Modeling and Tools for Network Simulation*, pages 15–34. Springer, 2010.

[53] L. Romdhani, Qiang Ni, and T. Turletti. Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-hoc Networks. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 2, pages 1373–1378 vol.2, March 2003.

[54] L. Sanabria-Russo, J Barcelo, and B Bellalta. Implementing CSMA/ECA in COST, 2015.

[55] L. Sanabria-Russo and B. Bellalta. Traffic Differentiation in Dense Collision-Free WLANs using CSMA/ECA. *arXiv preprint arXiv:1512.02062*, 2015.

[56] Luis Sanabria-Russo, Jaume Barcelo, and Boris Bellalta. Prototyping Distributed Collision-Free MAC Protocols for WLANs in Real

Hardware. In *International Workshop on Multiple Access Communications*, pages 82–87. Springer, 2013.

[57] Luis Sanabria-Russo, Jaume Barcelo, Boris Bellalta, and Francesco Gringoli. A High Efficiency MAC Protocol for WLANs: Providing Fairness in Dense Scenarios. *arXiv preprint arXiv:1412.1395v2*, 2015.

[58] Luis Sanabria-Russo, Jaume Barcelo, Albert Domingo, and Boris Bellalta. Spectrum Sensing with USRP-E110. In *International Workshop on Multiple Access Communications*, pages 79–84. Springer, 2012.

[59] Luis Sanabria-Russo, Jaume Barcelo, Azadeh Faridi, and Boris Bellalta. WLANs Throughput Improvement with CSMA/ECA. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 125–126. IEEE, 2014.

[60] Luis Sanabria-Russo, Azadeh Faridi, Boris Bellalta, Jaume Barceló, and Miquel 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.

[61] Luis Sanabria-Russo, Francesco Gringoli, Jaume Barcelo, and Boris Bellalta. Implementation and Experimental Evaluation of a Collision-Free MAC Protocol for WLANs. In *2015 IEEE International Conference on Communications (ICC)*, pages 1036–1042. IEEE, 2015.

[62] Sanabria-Russo, L. Github repository: CSMA-ECA-bounds-example. Webpage, 2014. Available at: https://github.com/SanabriaRusso/CSMA-ECA-bounds-example.

[63] Sanabria-Russo, L. Github repository: CSMA-ECA-HEW. Webpage, accessed June 2015, 2014.

[64] Sanabria-Russo, L. CSMA/ECA in NS-3: a super short tutorial. Webpage, 2016.

[65] Sanabria-Russo, L. Github repository: CSMA-ECA-NS3. Webpage, accessed June 2016, 2016.

[66] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. No time to countdown: Migrating backoff to the frequency domain. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 241–252. ACM, 2011.

[67] Shahwaiz Afaqui, M. DSC calibration results with NS-3. https://mentor.ieee.org/802.11/dcn/15/11-15-1316-03-00ax-dsc-calibration-results-with-ns-3.pptx, 2015.

[68] D. Skordoulis, Qiang Ni, Hsiao-Hwa Chen, A.P. Stephens, Changwen Liu, and A. Jamalipour. IEEE 802.11n MAC Frame Aggregation Mechanisms for Next-Generation High-Throughput WLANs. *Wireless Communications, IEEE*, 15(1):40–47, February 2008.

[69] Sunghwa Son, Kyung-Joon 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)*, pages 650–651, July 2013.

[70] Mirko Stoffers and George Riley. Comparing the NS-3 Propagation Models. In *2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 61–67. IEEE, 2012.

[71] The MADWifi Project. Multiband Atheros Driver for Wireless Fidelity. Webpage, 2013. Available at: http://madwifi-project.org/.

[72] C. Thorpe and L. Murphy. A Survey of Adaptive Carrier Sensing Mechanisms for IEEE 802.11 Wireless Networks. *IEEE Communications Surveys Tutorials*, 16(3):1266–1293, Third 2014.

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

[74] Ajay Tirumala, Feng Qin, Jon Dugan, Jim Ferguson, and Kevin Gibbs. Iperf: The TCP/UDP bandwidth measurement tool. *http://dast.nlanr.net/Projects*, 2005.

[75] Geert Van der Auwera, Prasanth T David, and Martin Reisslein. Traffic Characteristics of H.264/AVC Variable Bit Rate Video. *Communications Magazine, IEEE*, 46(11):164–174, 2008.

[76] E. Yucesan, CH. Chen, JL. Snowdon, and JM. Charnes. COST: A component-oriented discrete event simulator. In *Winter Simulation Conference*, 2002.