IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 18, 2018, accepted September 7, 2018, date of publication September 17, 2018, date of current version October 17, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2870691

Anti-Collision Adaptations of BLE Active Scanning for Dense IoT Tracking Applications

ÁNGELA HERNÁNDEZ-SOLANA[1], DAVID PÉREZ-DÍAZ-DE-CERIO[2], ANTONIO VALDOVINOS[1], AND JOSE LUIS VALENZUELA[2]

[1]Aragón Institute for Engineering Research, University of Zaragoza, 50018 Zaragoza, Spain
[2]Signal Theory and Communications Department, Universitat Politècnica de Catalunya, 08860 Castelldefels, Spain

Corresponding author: Ángela Hernández-Solana (anhersol@unizar.es)

This work was supported in part by the MINECO/FEDER under Projects TEC2014-58341-C4-2-R and TEC2014-60258-C2-2-R and in part by the Gobierno de Aragón/FEDER (Grupo de investigación T31_17R).

ABSTRACT Bluetooth low energy (BLE) is one of most promising technologies to enable the Internet-of-Things (IoT) paradigm. The BLE neighbor discovery process (NDP) based on active scanning may be the core of multiple IoT applications in which a large and varying number of users/devices/tags must be detected in a short period of time. Minimizing the discovery latency and maximizing the number of devices that can be discovered in a limited time are challenging issues due to collisions between frames sent by advertisers and scanners. The mechanism for resolution of collisions between scanners has a great impact on the achieved performance, but backoff in NDP has been poorly studied so far. This paper includes a detailed analysis of backoff in NDP, identifies and studies the factors involved in the process, reveals the limitations and problems presented by the algorithm suggested by the specifications and proposes simple and practical adaptations on scanner functionality. They are easily compatible with the current definitions of the standard, which together with a new proposal for the backoff scheme, may significantly improve the discovery latencies and, thus, the probability of discovering a large number of devices in high density scenarios.

INDEX TERMS Internet of Things (IoT), BLE, backoff algorithm, neighbor discovery process, discovery latency.

I. INTRODUCTION

Bluetooth Low Energy (BLE) was first introduced in version 4.0 of the Bluetooth Core Specification to eliminate the pairing and simplify the complex Bluetooth discovery processes, while still supporting short data exchanges. In addition to lower cost, fast neighbor discovery process (NDP), included in the specification [1], [2], and the periodic sleep during connections, allow to BLE a significant reduction of energy consumption. These characteristics have made it to emerge as one of the most promising technologies to enable the Internet-of-Things (IoT) paradigm. Nowadays, its position is reaffirmed with Bluetooth version 5 [2], and BLE is considered an attractive option for a wide range of applications, including sport monitoring, home electronics, smarthealth, domotics, security, intelligent transportation systems, etc. [3].

All communications in BLE networks must involve neighbor discovery process. NDP is required the first time a BLE device needs to create a connection or exchange information with its neighbors. But, NDP can be, by itself, the core of multiple IoT applications. Among the wide range of them, we focus our interest on applications that involve the discovery of a large number of users/devices/tags in a short period of time, such as race tracking for sport events, access control, cattle control, goods traceability, etc. That is, dense IoT tracking applications, inside an ecosystem where the presence of external devices interfering our application or service may also be possible.

In the simplest form of NDP, devices which apply it operate in passive scanning mode. This means that devices to be discovered (advertisers) periodically send short advertising messages in broadcast mode. Devices in passive scanning (scanners), simply listen to advertising messages, but they do not respond in any way. The scheme is simple and it has been shown [4] to be a reliable alternative to discover a high number of devices on very short periods of time, with no additional data exchange. Nevertheless, there is no direct interaction between the advertiser and the scanners.

NDP based on active scanning can be used when reception acknowledgment or more data exchange are needed. In active scanning, the advertisers continuously and sequentially send packets named ADV PDUs, through each of the three advertising channels defined in the specifications. These messages, listened by the scanners, trigger a three-way handshake. All the scanning devices receiving an active advertising message respond with a unicast scan request message (SCAN_REQ) to the advertiser (on the same channel where it has been received) and the advertiser finishes the data exchange with a scan response message (SCAN_RSP) with additional payload information. The scanners shall run a backoff procedure to minimize collisions of SCAN_REQ PDUs from multiple scanners.

The discovery probabilities and latencies, especially when a large number of tags (advertisers) have to be discovered in short periods of time, are severely affected by the characteristics of the backoff mechanism implementation [5]. Its implementation is mandatory, but after v5.0 [2] the standard only proposes the backoff mechanism defined in v4.2 [1] as an example of such a procedure, which has been proved to be affected by many limitations: 1) High probability of unnecessary backoff activation [5]. The backoff applies unnecessarily many times, and even when only one scanner is present, due to the lack of efficient discrimination between the collisions that really involve two or more scanners and erroneous receptions due to fading or collisions that involve only one scanner and one or several tags. This fact occurs constantly in dense IoT scenarios and may result in large discovery latencies. 2) Unfairness performance. There are some scanners that monopolize the access to the channel, completing the discovery process, while many others have poor chances of completing their discovery processes. This is clearly undesired when the scanners do not cooperate with each other. This unfairness, together with the throughput inefficiency, is connected with limitations of the random backoff proposed in the specifications and its suggested adjustment rules.

Since v5.0, the standard only requires that devices implement some backoff algorithm that allows to share the medium responsibly. This means that backoff rules could be totally different between manufactures. If that is the case, being these implementations proprietary, they would be often unknown. Concerning to the state of the art, although the backoff mechanism implementation has a great impact in the BLE device discovery process, the topic has been poorly studied so far. The almost totality of the works concerning to NDP (most of them focused on developing analytical models of NDP, or on evaluating NDP for different parameter setting using extensive simulations) assume the standard backoff implementation or do not even consider it [6]–[8]. Only very few works [9]–[12] have addressed specifically the impact of this topic. Contrary to the collision resolution schemes applied in many wireless networks, particularly in IEEE 802.11, where this issue is one of the most important to control any transmission along the air interface, in this case, the backoff only applies to collision resolution between scanners. That is, it is defined completely disaggregated to any possible strategy to reduce the probability of collision between advertisers and scanners or between advertisers transmissions. Moreover, in the CSMA/CA MAC used in IEEE 802.11, transmissions are asynchronous. Meanwhile, in NDP, backoff is defined to solve collisions in a process where the scanners requests occur at a predefined time relative to the reception of the advertising packets. Thus, collisions are affected by a synchronous scheme. All these facts introduce peculiarities that justify a particularized study of the issue in BLE.

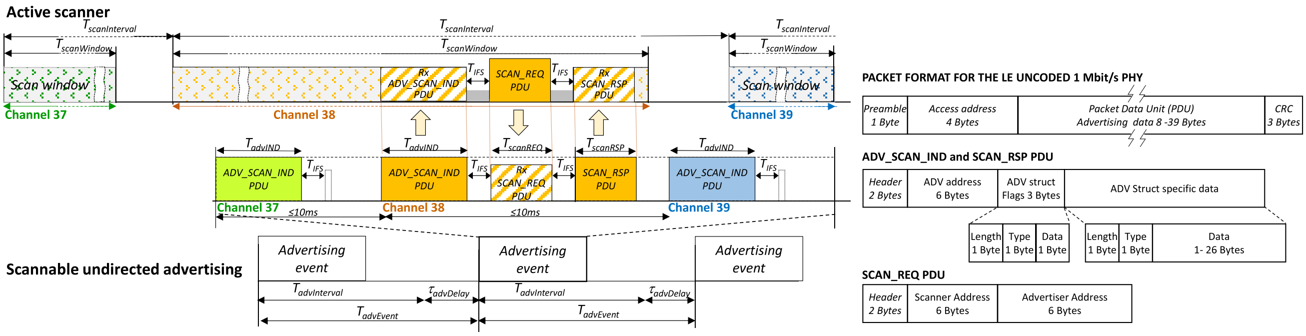
The aim of this paper is to provide a more detailed analysis of backoff in NDP. To better identify and understand the impact of the factors involved in the process, we include in the analysis a detailed characterization of practical limitations of real chipsets. This allows us to do a more critical analysis of the potential backoff implementations. Finally, without excluding further improvements, this work proposes simple and practical adaptations on scanner's functionality, easily compatible with the definitions of the current standards, which together with a new proposal for the backoff scheme, may significantly improve the discovery latencies and, thus, the probability of discovering a large number of devices in scenarios where a number of scanners are often colliding persistently on the same channel frequency.

The rest of the paper is structured as follows. Section II shortly describes the main characteristics of the BLE standard and the active scanning process. This section includes a description of usual non-idealities existing in the manufactured chipset scanner operations which impact on discovery latencies, particularly in IoT applications where a dense number of tags are present. Section III summarizes the main drawbacks of the backoff scheme and their effects on BLE discovery capacities. Section IV reviews related work concerning with the state of the art of backoff mechanisms and discuss their potential applicability on BLE. The details of the proposed mechanism are given in Section V and Section VI includes the performance evaluation of the proposal. Finally, section VII concludes the paper.

II. BLE ACTIVE SCANNING FUNDAMENTS REVIEW

This section reviews the basis of BLE technology, particularly those aspects related to the device/neighbor discovery process. BLE operates in the unlicensed 2.4 GHz ISM band using 40 physical channels separated 2 MHz. Unlike classical Bluetooth, and in order to balance contention and delay, BLE restricts NDP transmissions to three special channels (37, 38, and 39), called advertising channels, while the rest are data channels.

The basic communication mechanism to discover devices in BLE is based on advertisement events (see Fig. 1). In order to be discovered, a device is configured in advertising mode (from now on, named advertiser). Under this state, the advertiser broadcasts advertising messages (ADV PDUs) in sequence over each of the three advertising channels (index=37, 38, and 39) although a mask can be applied to select any combination of these three channels. The advertising event is repeated after $T_{advEvent}$, a period of time that is composed by the sum of a fixed part ($T_{advInterval}$),



(a) (b)

FIGURE 1. **Scannable undirected advertising event. (a) Example of scannable undirected advertising event. (b) Packet formats.**

which shall be an integer multiple of 0.625 ms in the range from 20 ms to 10.24 s, and a random time ($\tau_{advDelay}$), with a value in the range of 0 ms to 10 ms. On the other side, devices configured in active scanning mode (from now on, named scanner), listen for advertisement PDUs in order to discover their neighbors. Scanners listen in one of the three advertisement channels during a configurable time period called scan window ($T_{scanWindow}$). After a scan interval ($T_{scanInterval}$), a time period at least as long as $T_{scanWindow}$, the scanner switches to the next advertisement channel in a round-robin fashion. The standard specifies *passive* and *active scanning* modes, but our interest focus on *active scanning* and, among the several advertisement modes, on *scannable undirected advertising events* (the advertiser can use either ADV_SCAN_IND or ADV_EXT_IND PDUs, but this work only considers ADV_SCAN_IND PDU, with duration T_{advIND}).

With *scannable undirected advertising events,* when an ADV_SCAN_IND packet is received by a scanner in active scanning mode and this advertiser is allowed by its scanner filter policy, it shall respond with a *scan request* message (SCAN_REQ PDU) in the same frequency to request additional information from the advertiser and, then, listen for a scan response PDU (SCAN_RSP PDU). A SCAN_REQ PDU (with a fixed duration $T_{scanREQ} = 176\ \mu s$) is sent exactly 150 μs (T_{IFS}) after the successful reception of the ADV_SCAN_IND. If the advertiser receives a SCAN_REQ PDU that contains its device address from a scanner allowed by its advertising filter policy, it shall reply with a SCAN_RSP PDU (with duration $T_{scanRSP}$) in the same advertising channel index, a T_{IFS} later. Once the SCAN_RSP PDU is sent, or if the advertising filter policy prohibits processing the SCAN_REQ, the advertiser shall either move to the next used primary channel index to send another ADV_SCAN_IND PDU (also if a SCAN_REQ PDU has not been received), or close the advertising event. The time interval between the beginnings of two consecutive ADV_SCAN_IND PDUs, within an advertisement event, shall be less than 10 ms. A scanner shall continue to respond to the same advertiser until it has successfully received the SCAN_RSP PDU, and may either respond to or ignore subsequent scannable PDUs from the same advertiser.

According with the specification, the scanner shall run a backoff procedure to minimize collisions of SCAN_REQ PDUs from multiple scanners. Since version 5.0 of BLE specifications the implementation of the backoff algorithm is out of standardization, but the mechanism defined in BLE v4.2 remains suggested as an example of possible implementation. This means that most of real BLE chipsets implement this algorithm. The backoff procedure defined in v4.2 uses two parameters, *backoffCount* and *upperLimit*. Upon entering the scanning state, the *upperLimit* and *backoffCount* are set to one. On every received ADV_SCAN_IND PDU that is allowed by the scanner filter policy, the *backoffCount* is reduced by one (until zero) and the SCAN_REQ PDU is only sent by the scanner when *backoffCount* becomes 0. If after sending a SCAN_REQ PDU a valid SCAN_RSP PDU is not received at the Link Layer of the scanner, it is considered a failure; otherwise it is considered a success. The *upperLimit* value is doubled every two consecutive failures (until it reaches the value of 256), and halved (until it reaches the value of 1) after every two consecutive successes. Moreover, after every success or failure, the link layer selects a pseudo-random integer value for *backoffCount,* between one and *upperLimit* inclusive.

Real Chipset Implications: Regardless of the specified above, after analyzing a wide number of BLE chipset manufacturers (including Cambridge Silicon Radio, Nordic Semiconductor, Bluegiga, Broadcom, and Texas Instruments), we have verified that all of them present non-idealities, that should not be obviated when the performance of NDP procedures (latency, probability of detection, power consumption, etc.) is evaluated. Actually, the impact is very significant as we demonstrate in [4] and [5], regardless the discovery mode (passive or active), whether or not the backoff is applied or the type of backoff. These non-idealities do not appear associated with the backoff itself, but its knowledge is important to understand what happens when backoff applies and assess the feasibility of introducing some proposal to reduce the collisions.

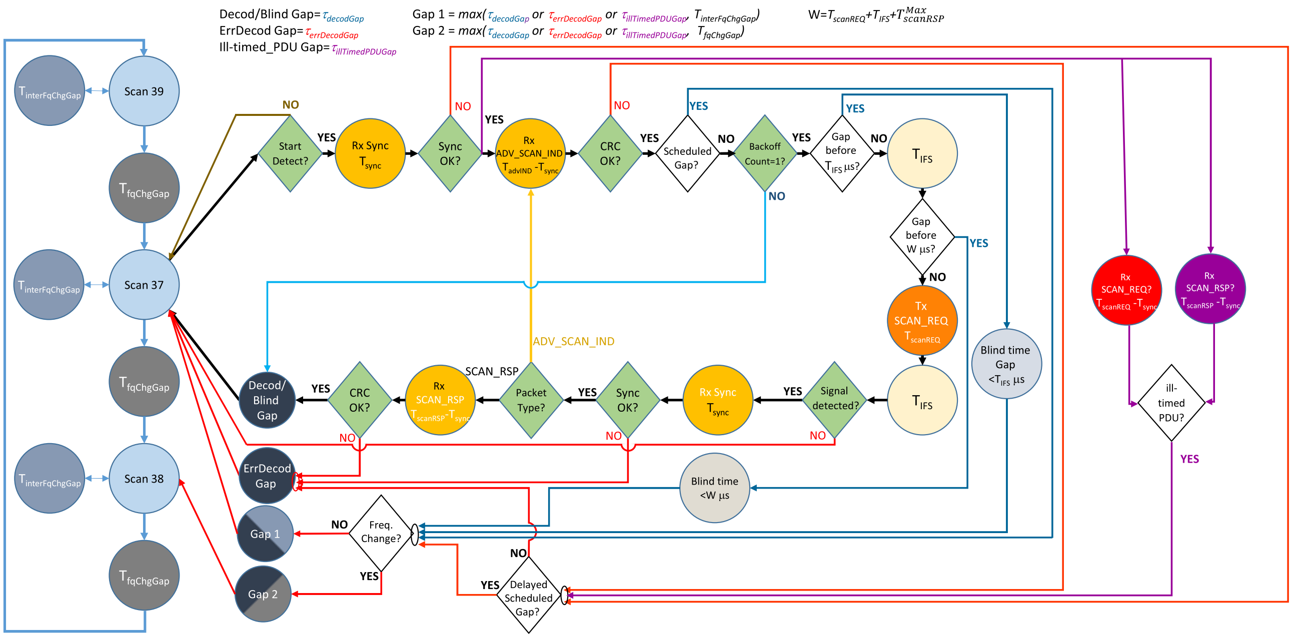


FIGURE 2. Example of scanner state diagram for a real device. Scannable undirected advertising event.

Fig. 2 shows an example of how the state diagram of a real scanner device should be modified (see [5] for further details). With slight variations, all the analyzed real chipsets present non-idealities and some reception characteristics, which can be summarized in:

- Even when the scanning devices are configured in continuous scan mode ($T_{scanInterval} = T_{scanWindow}$), all the scanning devices present undesired and unexpected pauses in the scanning (blind times): 1) gaps that appear when the scanner changes the scanning frequency ($T_{fqChgGap}$) and 2) additional short pauses that appear following several periodic patterns during the $T_{scanWindow}$ interval ($T_{interFqChgGap}$). These pauses can be deterministic or random variables, whose distributions are necessary to characterize. These gaps may be postponed to meet the three steps of the message exchange process.

- The receiver always tries to process the first incoming packet while it is in the scanning state. If the process has already been initiated when another packet is received, the second packet will always be discarded.

- Scanners introduce additional blind times whenever a packet is received. We have named these times decoding gaps, but we can identify different types: 1) after a synchronization failure the scanner aborts the packet processing procedure and enters into a blind time (deterministic or random variable with mean $\tau_{errDecGap}$), 2) if synchronization is successful but CRC is erroneous ($\tau_{errDecGap}$), 3) CRC is correct (τ_{decGap}), 4) if synchronization is successful but the type of message is out of sequence in the corresponding process ($\tau_{illTimedPDUGap}$), for instance, a SCAN_REQ or SCAN_RSP received without sending a previous SCAN_REQ, and 5) if SCAN_REQ is not sent due to backoff.

III. DRAWBACKS OF BACKOFF BASED COLLISION RESOLUTION ON BLE DISCOVERY CAPACITIES

In this paper we tackle the two main problems identified in [5] concerning with the backoff scheme suggested in the standard to minimize collisions of scan request PDUs from multiple scanners.

A. INEFFICIENT ACTIVATION AND OPERATION

The backoff algorithm may be applied to avoid collisions between SCAN_REQs in scenarios with two or more scanners transmitting simultaneously in the same channel (see Event#1 in Fig. 3, where NDP is illustrated). Nevertheless, these collisions are identified exclusively by the unsuccessful reception of the expected SCAN_RSP PDU. The inaccuracy of this method leads to the occurrence of many unnecessary backoff activations. This fact occurs with a very high probability in very dense scenarios, including scenarios with only one scanner, where a backoff algorithm is not required. This leads to a severe and unnecessary degradation of the discovery latency. Unnecessary backoff activations are associated with:

- SCAN_REQ or SCAN_RSP that are not received due to collisions with PDU messages different from SCAN_REQ. For instance, in Fig. 3, the SCAN_REQ in the event #2 is not received due to a collision with an ADV_SCAN_IND PDU, sent by advertiser @#3. In the events #4 and #6 the SCAN_RSPs are not received due to collisions with other ADV_SCAN_IND PDUs. Other types of collisions could also appear as collisions between SCAN_RSP and SCAN_REQ.

- Scanners do not discriminate between a SCAN_RSP unsent and a SCAN_RSP unsuccessfully processed, even

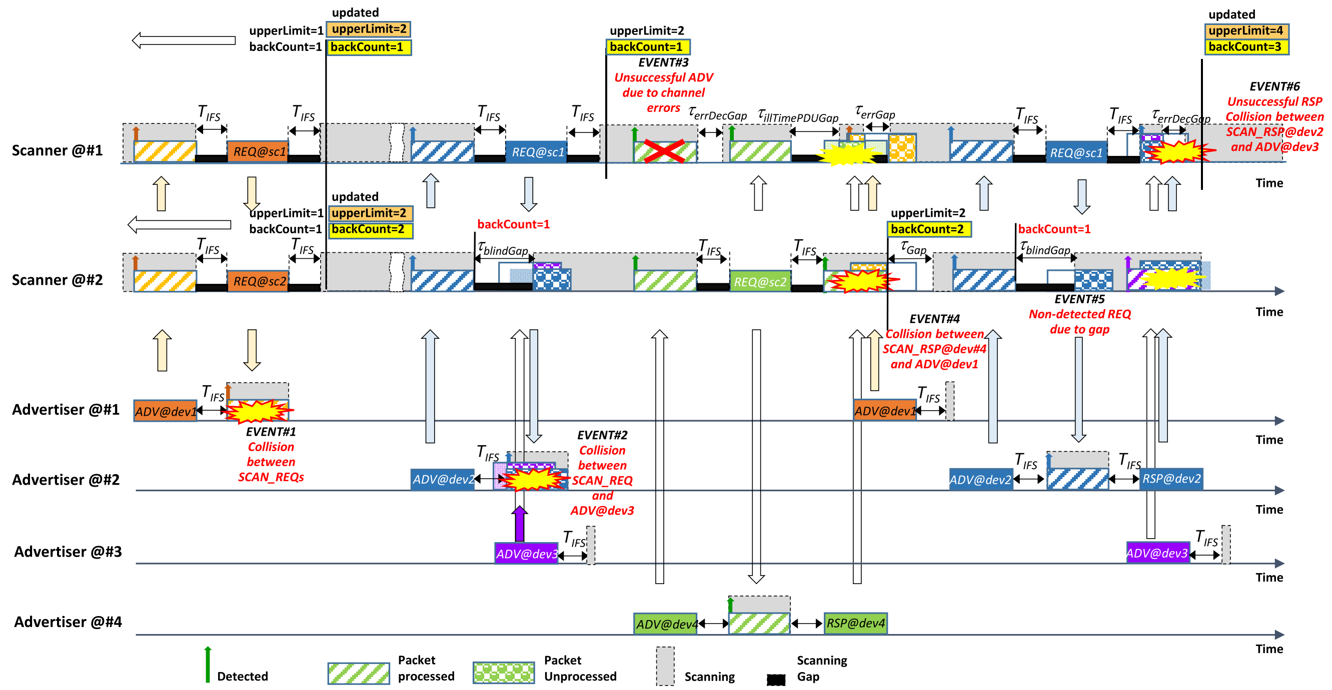


FIGURE 3. Example of control messages exchanged for NDP and backoff evolution according with real chipsets implementation. Scanner@#1 and scanner@#2 are scanning in the same frequency channel.

when it could be possible, as will be stated in section V. Seeing Fig. 3, the scanner @#2 in event #4, unlike the scanner @#1 in event #6, starts synchronizing the SCAN_RSP and it could even know that it is receiving a SCAN_RSP.

Fig. 3 also allows us to illustrate the impact of real chipset non-idealities on NDP performance. For example, both scanners are scanning the same frequency channel, but due to channel errors, scanner @#1 in event#3 is prevented to detect the ADV_SCAN_IND from the advertiser @#4, while this PDU is received by the scanner @#2. In this case, note that the scanner @#1 is capable of receiving the linked SCAN_REQ, sent by scanner @#2, although it does not process the message nor sends it to the link layer. Next, the unexpected reception of this SCAN_REQ deals on an additional gap which prevents the reception of the SCAN_RSP sent by advertiser @#4. In event #5, the scanner @#2 cannot process the SCAN_REQ sent by scanner @#1 due to the additional scanning gaps included in real chipsets.

B. UNFAIRNESS

Fairness, jointly to throughput efficiency and latency, has significant impact on the performance and application quality of service (QoS). The unfairness problem appears when some scanners can transmit a significantly larger number of SCAN_REQs than others. Due to the contention resolution process design, a lucky scanner, which successfully receives the SCAN_RSP, tends to maintain the contention window within a low value, while the other nodes continue growing their contention windows, thus reducing their chances of sending a SCAN_REQ. This results in domination of SCAN_RSP reception (finalization of the neighbor discovery process) by the successful nodes.

Unfair characteristic creates unequal competition between nodes, which is clearly undesired when the scanners do not cooperate with each other. For example, when some scanners do not belong to our system. On the contrary, in a planned application, where scanners could cooperate with each other, we might think that this characteristic may not necessarily be negative. The reason is that the lucky scanner may monopolize the discovery of all devices in the area, requiring a shorter delay than if everyone competes in equality. Nevertheless, even in this case, unfairness is negative in some situations. The problem appears when advertisers that have good propagation conditions only with scanners with large contention windows enter in the coverage area. As a result, the discovery latency may be unnecessarily high.

IV. RELATED WORK ABOUT BACKOFF

Even though the backoff mechanism implementation has a great impact in the device discovery process of BLE, its effects have been poorly studied so far. In fact, apart from the mechanism proposed in the standard, the proposals concerning to the backoff implementation are limited to a few works [9]–[12]. Probably, no attention has been paid because its use has been considered exceptional. However, the use of BLE in IoT applications, and the application of backoff in dense scenarios, changes this assumption.

In [9], a randomization of the frequency scanning sequence of each scanner is proposed, so that if two scanners coincide in the scan frequency and collide their SCAN_REQ PDUs, the probability of collision in the subsequent transmission decreases by following different sequences in the frequencies that they scan. The proposal can be useful. Nevertheless, in addition to the need for scanner synchronization, the problem is that for environments where a large number of scanning devices is expected further optimization mechanisms are required to solve inevitable contentions of multiple devices on the same frequency channel. In order to combat persistent collisions on the same frequency channel, in [10] Kim and Han propose an algorithm that eliminates the fixed synchronization of 150 μs existing in the standard between the ADV_SCAN_IND, SCAN_REQ and SCAN_RSP packets, and introduce a random response time for the sending of the SCAN_REQ PDU by the scanner. The limitation of that proposal for practical implementation is that it implies a change on the timing relationship between the subsequent ADV_SCAN_IND, SCAN_REQ and SCAN_RSP PDUs, not compatible with the current versions of the BLE standard. These changes do not only affect the scanner but also the advertisers. Knowing that scanners must sent the SCAN_REQ PDU on the same channel as the ADV_SCAN_IND PDU has been received, the proposal does not explicitly consider timing control to allow the advertisers to send ADV_SCAN_IND PDUs on different frequency channels. Besides, the analyses performed in [10] are limited to a network where only one device acts as advertiser and other N devices are working as scanners. In [11], Yang and Tseng also propose that scanners respond in different times to minimize collisions. Actually, the work proposes a wait-slot scheme to distribute SCAN_REQ PDUs in different times when the scanners respond upon receiving an ADV_SCAN_IND PDU. But, in addition, it proposes a two-way communication (SCAN_RSP is not required) instead of the standard active scanning process.

The study carried out in [12] does not really introduce a proposal concerning to the backoff procedure, but, pointing out some limitations associated to it, the authors suggest some modifications in order to improve the discovering capacities. During the backoff state, different scanning devices could successfully request the data. The proposal of [12] is based on an opportunistic listening of advertiser responses, instead of the scan responses being ignored by those scanning devices that are prevented to send their request due to the backoff state. This proposal will improve the performance, but does not resolve the main drawbacks connected with the backoff procedure implementation.

Since backoff in BLE is a not sufficiently studied subject, other backoff procedures should be further investigated in depth. In particular, the objective is to study other proposals that can be applied, but keeping the basis and the timing of the active scanning procedure, and, eventually, supported by the *upperLimit* and *backoffCount* parameters.

Backoff schemes have been addressed and are a subject of interest in many research works for different wireless technologies. The scheme proposed in BLE specifications is one of the possible variants derived from the well-known Binary Exponential Backoff (BEB) algorithm. BEB algorithm [13], employed in IEEE 802.11 DCF (Distributed Coordination Function), is probably the most prevalent and widely used random backoff policy. In BEB the value of the contention window (CW) is doubled (bounded by a CW_{max}) after each transmission fails and it is reset to the initial contention window CW_{min} after every successful transmission. Despite its simplicity and good behavior in many cases, it suffers from an unfairness problem and low throughput under high traffic load. Thus, classified under the category of fixed-CW (CW_{min} and CW_{max} are set statically) several variants have been proposed to improve the fairness or throughput of BEB. Essentially, they differ on how CW is increased or decreased after every successful or unsuccessful transmission. The Multiplicative Increase Linear Decrease (MILD) algorithm has been proposed in [14]. MILD increases CW by a factor of 1.5 rather than doubling after each unsuccessful transmission and CW is decremented linearly (by 1), instead of resetting to the minimum, after a successful transmission. In addition, MILD assumes that each node, upon overhearing any successful transmission in the local area, copies the CW used into its local area. This means that each transmitted packet must include a copy of the CW it applies. This increases the header size of packets but can greatly improve the fairness because all the contending nodes converge to use similar values. However, this policy has its shortcomings. MILD does not perform well when the network load is light because it takes quite a long time to recover from the backoff caused by occasional collisions. Furthermore, when the number of competing nodes changes sharply from high to low, MILD cannot adjust its CW fast enough due to its ‘‘linear decrease’’ mechanism. Linear/Multiplicative Increase and Linear Decrease (LMILD) in [15] or Sensing Backoff Algorithm (SBA) [16] do not employ the CW copy mechanism of MILD, but nodes overhear any transmission on the channel. In LMILD/SBA colliding nodes increase their CW by multiplying it by a factor m_c. After successful transmissions (experienced or overheard), a node decreases its CW linearly in a l_s quantity and, after any overheard collision, it increases its CW by l_c units, being the goal to obtain the optimal step to increase and decrease the CW parameter [16]. LMILD improves the throughput of MILD, but does not lead to fairness and thus, it loses potential fairness improvements of MILD. On the other hand, misdetection or false positive problems may affect its performance.

To overcome the drawbacks associated to linear decrease options in MILD/LMILD, in the Exponential Increase Exponential Decrease (EIED) algorithm proposed in [17] and [18], the CW size is increased by a factor r_I and decreased by a factor r_D, on each collision and successful transmission, respectively. In a similar way, in Double Increment Double Decrement (DIDD) [19], CW is doubled when packets collide and halved on a successful packet transmission. The GDCF (Gentle DCF) algorithm proposed in [20] for DCF, applies a more conservative policy than DIDD, by halving the CW after c consecutive successful transmissions (c needs to be optimized). This assumption is made to reduce the collision probability, especially when the number of competing nodes is large. The goal remains being setting the *c* value for different traffic scenarios. However, the scheme proposed for DCF reduces CW to 1 if the channel is idle.

The backoff procedure suggested by the BLE specification, in addition, also applies a conservative policy for unsuccessful transmissions. In this case, by imposing a number of consecutive errors (two) to double CW, the mechanism provides an extra of robustness in front of false collision estimation between scanners. As we refer in section III, collisions between SCAN_REQ and ADV_SCAN_IND or SCAN_RSP, or between SCAN_RSP and ADV_SCAN_IND should not increase the CW. By imposing consecutive errors, the mechanism improves a probabilistic discrimination between these collisions or fading channels errors, and scanner collisions. To unnecessarily increase CW, assuming erroneously that all packet losses are due to collisions in fading wireless channels, results in an inefficient performance [21].

Nevertheless, this scheme, like BEB and almost all the referred schemes (except MILD), suffers from a fairness problem. These schemes reset or decrement the contention window of a successful sender, while other devices maintain or increase their upper contention windows, and thus reduce their chances of accessing the channel. This results in channel access domination of the successful nodes. In addition, the selection of CW_{min} and CW_{max} has a significant impact on the performance. Improper selection increases collisions and unfairness. Since these algorithms do not implicitly consider dynamic traffic loads changes to set the CW limits, they may suffer maladjustment and loss of throughput in those conditions. Moreover, the CW adjustment rules, based on the result of the last transmission, implies constant changes in CW, which may be considered a drawback in itself. Besides, improper CW adjustment rules are the main cause of reduced throughput.

Actually, applying a constant-window backoff (CW) in all the contending nodes maximizes the network throughput and ensures the fairness. However, the optimal window selection depends, among others, on traffic load. Furthermore, the applicability of these schemes requires updated information about the number of nodes in the network, which is not practically applicable in a dynamic environment. Optimal CW selection is not possible, but there are many works that propose adaptive CW adjustments based on estimations (measurements) of the network condition along a period of time, regardless of the last transmission result [22]–[24]. Many proposals are based on estimating the number of active nodes and/or traffic load by observing the channel status (number of slots times that were observed to be busy or channel state probabilities -*collision, success and idle*- for the total backoff period), and then they exchange channel state information and establish a certain coordination between the nodes.

In BLE, any potential scheme that assumes overhearing the channel needs to be applied carefully, given that collisions between ADV_SCAN_IND PDUs do not concern the number of potential collisions between SCAN_REQ PDUs. Estimating the number of unknown scanners in this context is really difficult, especially if we consider the absence of monitoring capabilities (apart from the expected SCAN_RSP detection) included in the specification and knowing that each advertiser and each scanner can enter or leave the mutual coverage area at any time. In addition, although the number of scanners remains stable for a long time and the estimation is feasible, the scanners that compete in each channel can change along the time due to their particular and uncoordinated configuration of the scanning parameters.

Seeing that sharing CW values between competing nodes improves the performance (fairness and throughput), a CW copy mechanism, like that included in MILD, may be useful to offer fairness performance. Nevertheless, its use requires some adaptations in order to be considered, because backoff application in BLE is quite different from IEEE 802.11: active scanning in BLE does not apply channel sensing before transmission.

On the other hand, collisions between scanners in a coverage area are expected to be semi persistent. All the scanners must transmit a SCAN_REQ (only backoff prevents it) to any advertiser (allowed by the scanner filter policy) from which an ADV_SCAN_IND is received. Seeing this fact, a Semi-Random Backoff (SRB) proposal, explored for WLAN [25], [26] could be interesting. The idea is using a deterministic backoff CW after successful transmissions and a random backoff otherwise. If the deterministic backoff is constant for all the stations (in our case, scanners), the system naturally converges to a collision-free operation (like TDMA structure) in which the stations transmit in a round-robin deterministic fashion (each station/scanner has been assigned a slot/time opportunity). However, the proposal incurs in a challenging problem: how to optimally and dynamically converge to a CW size according with the number of stations. The chosen CW (optimally it should be equal to the number of stations) should not result in an inefficient use of resources or an unnecessary delay. As the ''order'' or ''slot'' will be released if a station does not transmit successfully in one or some consecutive backoff intervals (cycles), the performance of the network quickly deteriorates and may affect many transmissions. If applied to active scanning in BLE, the probability of unsuccessful transmissions could be high when the number of advertisers is high due to collisions that do not depend on the number of scanners. On the other hand, estimating the number of scanners present in the coverage area is another challenging issue. In short, it presents many of the problems mentioned above for random proposals.

Finally, analyzing it with another approach, the problem shows many similarities with tag identification in RFID, where many tags may respond simultaneously to a reader's request. Applying it to our BLE scenario, the scanner would correspond to the tag and the advertiser to the reader. In that context, tree-based protocols are the basis of many collision resolution algorithms [27]–[29]. Binary tree is one of the most representative of these algorithms. The problem is that this approach is difficult to apply to BLE because every scanner needs to recognize (or having feedback from an advertiser) three states: idle (no transmission is attempted by any other scanner), readable (one transmission is recognized, that is, SCAN_REQ has been successfully received by the advertiser) and collision (more than one scanner have attempted to transmit). Recognizing these states without the advertiser feedback is hard and will always be imprecise, because it must be derived from SCAN_RSP detection. Including feedback information on the ADV_SCAN_IND or SCAN_RSP messages implies new overhead information, even though reserved bits on the PDU header could be used. However, the main drawback is that, instead of an ideal scenario, where only one reader or a limited number of neighbor readers are present, the BLE context we are interested in, actually implies a high number of advertisers (readers) that may collide and affect these communications. Even when only one reader is considered, when the number of tags is large, the process may suffer from consecutive collisions that increase the detection delay. Therefore, this type of approach has been currently excluded.

Seeing the limitations of all the backoff procedures reviewed above, our goal is to configure a simple and heuristic proposal, adapted to the specific characteristics of the protocol used in BLE.

V. ADAPTATIONS OF BLE ACTIVE SCANNING FOR DENSE IoT TRACKING APPLICATIONS

Considering the backoff drawbacks in BLE, we propose several adaptations into the scanning devices and a new backoff rule to improve discovery capacities.

The adaptations can be applied both to the backoff mechanism suggested by the standard and to the proposed backoff mechanism, we have named Aware to Context Conservative Multiplicative Increase and Decrease (AwCMID). Realize that the proposals focus on reducing collisions of scanners operating in the same channel, without excluding additional policies aimed to synchronize the scanner operation along the three channels.

Use of physical layer information to discriminate between unsent SCAN_RSP (with a certain probability, due to scan request collisions) and unsuccessful receptions of SCAN_RSP.

One cause of the discovery capacity efficiency decrease is that if the SCAN_RSP PDU was not successfully received at the scanning, the backoff algorithm always assumes that the error is due to scanner collisions. However, as we refer in section III, when a large number of tags or advertising devices coexist under the scanner coverage, the probability that a requested SCAN_RSP is not received at the scanner due to collisions of the SCAN_RSP with ADV_SCAN_IND from neighbor advertisers may be significantly high and even more relevant than the absence of SCAN_RSP due to SCAN_REQ collisions between scanning devices. For instance, in a scenario with N=120 advertisers, $T_{advEvent} = 105$ ms, $T_{IFS} = 150$ μs, $T_{advIND} = 176$ μs, and $T_{scanRSP} = 128$ μs, the scan response collision probability with ADV_SCAN_IND transmissions of other devices (even in ideal channel conditions) is around 27%. This probability is calculated according with expression (1).

$$P^{col}_{NDScanRSP} = 1 - \left(1 - \frac{\min(T_{IFS}, T_{advIND}) + T_{scanRSP}}{T_{advEvent}}\right)^{N-1} \quad (1)$$

That is, since the backoff algorithm assumes that all unsuccessful receptions are due to collisions between scanners, the *upperLimit* of the backoff procedure can be unnecessarily increased (doubled) with high probability even though the cause might be different. This leads to a significant inefficiency. Discrimination is critical as long as the number of advertising devices increases and packet losses due to channel errors become more significant. The goal of our proposal is to discriminate between unsent scan responses and unsuccessful receptions. From experimental measurements with a large number of different chipsets we have verified that scanning devices could potentially estimate the cause of an unsuccessful SCAN_RSP detection. While a device is in a scanning state, once it starts detecting energy on the channel, it begins a packet processing procedure. In fact, the receiver always tries to process the first incoming packet and if subsequent packets are received when the process has already been initiated, they will always be discarded. In this case, if exactly after T_{IFS} μs the scanner successfully synchronizes a preamble, the device can deduce a potential SCAN_RSP reception even though the associated PDU is erroneous or only partially detected. Scan responses colliding with packets that start their transmissions on the T_{IFS} interval cannot be detected. Thus, even they have been sent, we cannot discriminate this fact. But, a scan response that has collided with a PDU whose transmission starts after the preamble time or is corrupted due to channel errors, can be identified. To be more conservative, according with the model included in section II, we have imposed that we only identify the scan response if the collision occurs after the sum of the preamble and access address times. The probability that an ADV PDU is exactly sent at T_{IFS} μs is really low. Thus, a false detection of an erroneous ADV_SCAN_IND PDU as a potential SCAN_RSP PDU is negligible.

Modified Backoff Algorithm to improve fairness. As shown in section III, all the proposed collision resolution algorithms are heuristic. Considering the practical limitations of efficiently implementing any proposal that leads to an ideal deterministic ordering of SCAN_REQ transmissions (that is, borrowing the idea of semi-random backoff), we opt for applying a random backoff, we have named AwCMID.

As discussed in section IV, estimating the number of unknown scanners in a dynamic environment with a dense concentration of advertisers would be really difficult, especially without introducing relevant modifications to the active discovery process defined in BLE. As the simplicity is a main requirement, between fixed-CW and adaptive-CW adjustment (referred to CW_{max} and CW_{min} setting), we adopt the first approach, being the maximum *upperLimit*(CW_{max}) set to 256 and the minimum (CW_{min}) to 1.

We cannot forget that the goal of the IoT application scenario is to detect (receive the SCAN_RSP response) the advertising devices in the shortest period of time. The main efficiency parameter is the time required to discover all the devices (based on their SCAN_RSP detection) in the coverage area. Or, if preferred, the probability that all the devices present in the scanner coverage area are detected within a limited time interval (window of opportunity or dwell time). In addition, it is interesting to obtain the percentage of devices detected in this time interval. As stated before, the optimum would be that the values of CW_{max} and CW_{min} were predetermined based on the number of scanners. We exclude this option due to the variability in the number of scanners and the difficulty of discriminating the exact number of scanners based on collisions when a dense advertiser scenario is considered. As long as the *upperLimit* grows (up to 256), the SCAN_REQ collisions decrease, but, if *upperLimit* grows unnecessarily, the efficiency in terms of discovery delay greatly degrades. Thus, the main approach of the proposed method is to modify the *upperLimit* adjustment rule.

The performance of the scenario is limited by collisions with ADV_SCAN_INDs. In absence of recognizable neighbor scanners, we maintain the mechanism proposed in the standard. By imposing two consecutive errors, the mechanism reduces the probability of growing the CW due to collisions between SCAN_REQ and ADV_SCAN_IND or SCAN_RSP, or between SCAN_RSP and ADV_SCAN_IND. On the other hand, halving the CW allows a fast recovery from the backoff caused by occasional collisions or collisions with ADV_SCAN_IND/SCAN_RSP when the number of neighbor scanners is low or even none. However, by imposing two consecutive successful transmissions, a balance is sought with scenarios where the number of scanners is high. This prevents that an occasional successful transmission decreases too fast the *upperLimit*, increasing the number of collisions.

However, in order to obtain a better performance when multiples scanners compete, we apply an *upperLimit* copy mechanism. The idea is that scanners can know the *upperLimit* values used by the other scanners with which they compete, in a similar way of that proposed in MILD, but with some adaptations. It provides two advantages: 1) thanks to CW overhearing fairness can be greatly improved and 2) the *upperLimit* is expected to be maintained in lower values (lower delays can be achieved).

The proposal requires some modifications of the scanner and the advertisers:

1) Each scanner needs to include in its transmitted SCAN_REQ PDU the *upperLimit* value that it has used in the last backoff cycle. New overhead is not required when specifications 4.2 or 5 are applied. The PDU Header contains 4 bits reserved for future uses (b_{4-5} and b_{14-15}). The length is enough to map the exponent used in the *upperLimit* adjustment (2^w w=0,. . ., 8).
2) The advertiser that successfully receives the SCAN_REQ and responds with a SCAN_RSP includes in its SCAN_RSP PDU header a copy of the *upperLimit* received in the REQ.
3) Scanners in backoff state are able to opportunistically monitor SCAN_RSP packets sent in response to the SCAN_REQ of another scanner (no matter the scanner could have sent the SCAN_REQ but backoff prevented it, or the scanner has not detected the ADV_SCAN_IND) In this case, the scanner only learns the *upperLimit* if the packet has been successfully received. This is a conservative assumption.
4) We assume that although all the SCAN_REQ PDUs include an *upperLimit* copy, usually a scanner that has detected an ADV_SCAN_IND and goes into the backoff will not be able to hear the neighbor SCAN_REQ PDU because the decoding time is greater than 150 μs. On the other hand, even when the reception of the SCAN_REQ is possible (for instance, as in the event#3 of Fig. 3), the overhearing of this frame does not provide enough information about the result of the reception on the advertiser. Thus, we only consider the *upperLimit* copy included in the SCAN_RSP PDUs.

The new *upperLimit* adjustment rule, implemented by a scanner, after success or failure of receiving the SCAN_RSP PDU (requested by it), is based on the following variables: *reqRSP(n-1)* is the result (successful or unsuccessful reception) of the previous requested SCAN_RSP (the (n-1)th event), and *upperLimit(n-1)* is the value the scanner had applied in the subsequent *backoffCount* update. In addition, *reqRSP(n)* is the result of the current reception and *upperLimit(n)* the value to be updated. *OLupperLimit* is the *upperLimit* values obtained opportunistically from non-requested advertisers (*oppRSP*), between *reqRSP(n-1)* and *reqRSP(n)*. *NCsucc* and *NCcol* are the number of consecutive success and collisions, respectively. The new adjustment rule is defined as follows:

Case 1: Between *reqRSP(n-1)* and *reqRSP(n)*, the scanner listens to at least one *oppRSP*. The scanner will always take the *OLupperLimit* from the latest *oppRSP*. **Then**:

If *reqRSP(n)* and *reqRSP(n-1)* were successful **then**
upperLimit(n) ←
←max(2, min(0.5·*upperLimit(n-1), OLupperLimit*))
Reset counter *NCsucc* to *0*.
else if *reqRSP(n)* is successful and *reqRSP(n-1)* was unsuccessful (counting that the *oppRSP* and the *reqRSP(n)* are already two consecutive successes) **then**
if (*OLupperLimit*≤*upperLimit(n-1)*)
upperLimit(n) ←
←max(2, min(0.5·*upperLimit(n-1), OLupperLimit*))
Reset counter *NCsucc* to *0*.
else
upperLimit(n) ←*upperLimit(n-1)*
NCsucc ←*1*

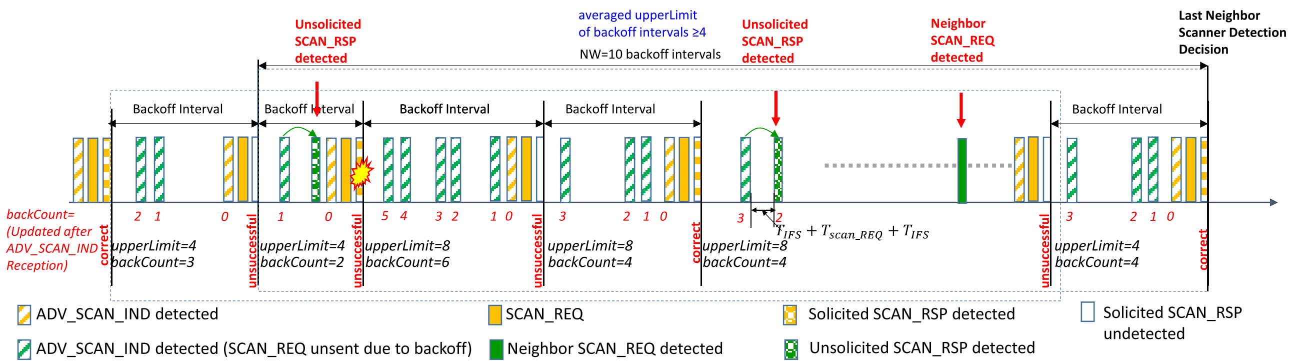


FIGURE 4. Monitoring scheme to detect the absence/presence of neighbor scanners.

end if
else if *reqRSP(n)* is unsuccessful (no matter *reqRSP(n-1)* was success or unsuccessful) **then**
upperLimit(n) ← max(*2, OLupperLimit*)
Reset counter *NCcol* to *0*.
end if

Case 2: If no *oppRSP* has been listened between *reqRSP(n-1)* and *reqRSP(n)*, the mechanism rules as the BLE specification. **Then**:
Upon two consecutive collisions (*NCcol*==2) **then**
upperLimit(n) ← min(*2·upperLimit(n-1)*,*256*)
Reset counter *NCcol* to *0*.
Upon two consecutive successes (*NCsucc==2*) **then**
upperLimit(n) ← max(1, *0.5·upperLimit(n-1)*)
Reset counter *NCsucc* to *0*.

Note that in all the cases if *upperLimit(n)* is updated, *NCsucc* or *NCcol* are reset to 0.

Opportunistic RSP listening. Same as suggested in [12], opportunistic listening of scan responses can be included in all the backoff algorithms and, as expected, it will always improve the efficiency. As we explain in section II, according with the standard specification the scanning devices in backoff will ignore scan response messages sent in response to other scanning devices scan requests. The improvement adopts the proposal made in [12] in order to extend the capabilities of the scanning devices to accept non-requested response messages if the scanner devices have entered in backoff mode for that advertiser. That is, the capabilities of a scanner that has successfully processed a ADV_SCAN_IND, but does not send its corresponding request due to the backoff process may be extended so that the link layer does not ignore these responses. In fact, from experimental measurements of current BLE chipsets we have verified that not requested responses are completely processed (decoding gaps are included after CRC verification and not only after the synchronization phase verification). Thus, the extension in order to move the content to upper layers is straightforward. The content of all other scan responses that could not be associated to a deliberate decision of not sending a scan request, is ignored by the link layer and is not sent to the highest layer.

Detection of absence/presence of neighbor scanners. BLE specification defines as mandatory that the scanner runs a backoff procedure to minimize collisions of SCAN_REQ PDUs from multiple scanners. However, because the specification states using the non-detection of the scan response PDU as an indication of SCAN_REQ collisions between scanners and the parameter to control de backoff process, the backoff runs even when it is not really required (for instance, in scenarios with only one scanner). As a result, the throughput may be many times severely and unnecessarily degraded by the mandatory backoff application. This indicator will be particularly wrong in a high dense scenario, where we often have non-detections of scan responses due to:

1) Collisions of SCAN_REQ with ADV_SCAN_IND or even SCAN_RSP sent by other advertisers in the coverage area or fading errors affecting to a SCAN_REQ reception.
2) Even when the SCAN_REQ is successfully received by the required advertiser, non-detections may be due to collisions of SCAN_RSP with ADV_SCAN_IND sent by other advertisers or fading errors affecting to the SCAN_RSP.

Thus, the aim of our proposal is to limit the effects of the backoff (without deactivating the process) by implementing a monitoring procedure (measurement based) that allows the scanner to estimate the presence or absence of other scanners. The procedure shall be simple and suitable for dynamic environments where conditions may change. The proposal discards the option of estimating the number of scanners. In CSMA/CA based networks as IEEE 802.11, several authors have proposed mechanisms based on determining an almost optimal fixed CW by estimating the number of active stations. Estimations based on the channel status (*collision, success and idle* periods for the total backoff cycle) have often been suggested [23]. However, active scanning BLE operation is, by far, very different from CSMA/CA. In BLE, backoff is only used to combat almost persistent SCAN_REQ collisions coming from scanners. Meanwhile, multiple advertiser transmissions coexist simultaneously and independently, with their corresponding collision probabilities. Being difficult to obtain an estimation about the number of scanner based on channel state probabilities, the first objective is to perform a binary decision about the presence or absence of other scanners.

In order to make a decision, for the active scanning process each scanner has to monitor the presence of unsolicited packets (scan responses) at the air interface for a predetermined sliding observation window, composed by a number *NW* of backoff intervals (see Fig.4). The backoff interval is the period of time between the event in which the scanner transmits a scan request PDU after reaching its *backoffCount* parameter the value of zero, and the next opportunity in which this situation is repeated. The duration of the period is lower the higher the rate of advertisers is. As specified in section II, in order for the *backoffCount* value to be decremented by one (until it reaches zero), the scanner needs to receive an ADV_SCAN_IND that is allowed by the scanner filter policy, for which a scan request PDU has to be sent. This means that, although the scanner only sends the SCAN_REQ when *backoffCount* becomes zero, it can listen to unsolicited SCAN_RSP messages corresponding to neighbor scanners that send scan requests after the ADV_SCAN_IND reception. This is the basis of the opportunistic RSP listening defined above, but with the following variations of the monitoring process:

- The scanner counts a SCAN_RSP reception whenever it is successfully decoded at the receiver, no matter the scanner could have sent a scan request (if it was not in backoff state) or not (if it has not received the ADV_SCAN_IND PDU).
- If the scanner has not sent the SCAN_REQ due to the backoff state, and if exactly $T_{IFS} + T_{scanREQ} + T_{IFS}$ μs after receiving an ADV_SCAN_IND the scanner starts successfully synchronizing a preamble, the scanner device counts a potential unsolicited SCAN_RSP reception, even if the associated PDU is erroneous or only partially detected.
- The scanner may optionally be programmed in order to count any SCAN_REQ PDU reception successfully decoded at the receiver, although it was ignored by the link layer. In this case, note that SCAN_REQ PDU detection depends on the chipset. For instance, in the analyzed real chipset a SCAN_REQ PDU cannot be detected after an ADV_SCAN_IND PDU successful detection due to the decoding gap interval.

In a backoff interval, even though other scanner devices exist, the probability of not detecting any unsolicited scan response or a scan request can be very high when the number of advertisers is high or the channel is unfavorable. However, the probability of detecting nothing during *NW=10* consecutive intervals is drastically reduced even in very dense scenarios. Seeing that, our proposal sets that the minimum time required to detect the absence of neighbor scanners is *NW=10* consecutive intervals. The time interval may be higher than *NW=10* because the scanner only takes a decision if in the last backoff interval it detects a solicited SCAN_RSP and if during the last *NW=10* consecutive intervals no unsolicited SCAN_RSP or neighbor SCAN_REQ have been detected. In this case, the *upperLimit* is reset to 1 and the number of backoff intervals monitored is reset to 0. The backoff procedure continues to be updated according to the general guidelines. That is, the decision does not remain over time, since the proposal tries to favor a rapid response from the backoff mechanism in case the distribution of competing scanners changes in a short time interval.

In order to combat some troubles if the scheme is applied with an unfair backoff procedure, we need to include some additional considerations. The aim is to reduce the probability that a scanner that monopolizes the SCAN_REQ transmission opportunities thinks that (with a high probability) it is alone. To reduce this probability, we force that the decision at the end of a *NW=10* backoff interval window can only be carried out when the averaged size of the backoff intervals in this window is equal or higher than an average *upperLimit* of 4 (see Fig.4).

VI. PERFORMANCE

In this section we present the comparison of the performance results obtained with the backoff mechanism suggested by the standard BLE [1] (we identify it as stdBLE ND), and the proposed backoff scheme, which we have named Aware to Context Conservative Multiplicative Increase and Decrease (AwCMID). Realize that AwCMID denomination concerns to the modified backoff rule. Improvements linked to the discrimination between unsent and unsuccessful SCAN_RSP, opportunistic SCAN_RSP listening and detection of neighbor scanners will be applied to both mechanisms. The evaluation has been performed by means of simulation, developing a simulator in C++ that fully reproduces (without any simplification) the advertising process according with the specifications, but also including the real chipset characteristics. Without losing generality about the overall conclusions of the comparison, the evaluation using real devices allows us to overcome the limitations that appear when considering an excessively idealized reproduction of the NDP described in the standard. In general, an ideal assumption implies that the probability of non-detection of the different frames involved in these scenarios is strongly underestimated, as it has been previously demonstrated in other studies [5]. Thus, the quantitative values concerning to discovery capacities would be overestimated. We consider a generic scenario, where a variable number N of advertising devices (up to 120) are needed to be discovered by a variable number of scanners (up to 8) in the shortest possible time interval. This scenario is compatible with dense IoT tracking applications (for instance a tracking application of runners during a race) where devices are under coverage of the readers for a time interval of T_{cov} s. Performance statistics are obtained by averaging up to 10,000 coverage time intervals.

All the devices (scanners and advertisers, respectively) are identically configured. Scanning devices are scanning continuously ($T_{scanWindow} = T_{scanInterval} = 500$ ms) and their $T_{scanWindow}$ are aligned in time and frequency. Actually, if a tracking application involving several scanners (up to three sharing a coverage area) is deployed, obviously, the best option is to force the scanners to be completely unsynchronized in their scanning frequency, to avoid SCAN_REQ collisions. Nevertheless, if the number of deployed scanners is higher or a number of scanners that do not cooperate with each other (or which do not belong to the tracking system) coexist in the same area, there will be competition between them. Actually, in the last scenario, the potential quantity of scanners that persistently scan on the same frequency will be significantly lower than the total number. Thus, by imposing the scanners to be aligned, we are analyzing a pessimistic scenario in a more controlled fashion in order to obtain conclusions, and a number up to eight scanners can be considered fairly high.

The scanners take into account the limitations of real devices discussed in section II. Specifically, the behavior characterization corresponds to devices named type 2 devices in [5], whose state diagram is depicted in Fig. 2. The value of the frequency change blind times ($T_{fqChgGap}$) is equal to 16.05 ms. The intermediate scanning gaps ($T_{interFqChgGap}$), distributed along a $T_{scanWindow}$ according a specific pattern, are equal to 300 μs, being $T_{gapInt1} = 16.82$ ms and $T_{gapInt2} = 4.3$ ms the periods of time between them. The details about this behaviour and the specific gap distribution pattern are described thoroughly in [4]. In addition, τ_{decGap}, $\tau_{errDecGap}$ and $\tau_{illTimePDUGap}$ correspond to deterministic values and are equal to 194 μs, 144 μs and 168 μs, respectively. Finally, $T^{max}_{scanRSP}$ is equal to the maximum allowed duration of a SCAN_RSP (376 μs). Advertisers are configured with $T_{advInterval} = 100$ ms, $T_{advIND} = T_{scanREQ} = 176$ μs and $T_{scanRSP} = 128\mu$s. In addition, the scannable advertisement event T_{IFS} is 150 μs, the time between the beginning of two consecutive ADV_SCAN_IND is 430 μs if no SCAN_REQ is received and the time between the end of a SCAN_RSP transmission and the next ADV_SCAN_IND within the event is 210 μs.

On the other hand, results are obtained under the following assumptions: 1) all the nodes are in the range of each other; 2) propagation delays are negligible; 3) in addition to packet collisions (no capture effect is considered), frame transmissions are affected by different values of Block Error Rate (BLER), applied independently for each possible receiver device.

The metrics used in the evaluation are the percentage of advertisers whose SCAN_RSP have been received by a scanner in the time of coverage (T_{cov}) and the fairness. T_{cov} is considered with a granularity of one second. The first metric provides a measure of the discovery latency. The Jain's fairness index is the standard traditional measure of network fairness. It is defined as follows (2):

$$J = \frac{1}{N_{SC}} \frac{\left(\sum_{i=1}^{N_{SC}} X_i\right)^2}{\sum_{i=1}^{N_{SC}} (X_i)^2} \tag{2}$$

where N_{SC} is the number of scanners and X_i is the number of *SCAN_REQ* transmitted by the *i-th* scanner. The Jain's fairness index is in the interval [0, 1], where larger values indicate better fairness. Specifically, absolute fairness is achieved when J=1 and absolute unfairness is achieved when J =1/ N_{SC}. To calculate the fairness index for a given timing window size, we use a sliding window. We calculate the fairness index for each window and then, we take the Jain's index as the average of the computed fairness index for the overall valid windows. Repeating this procedure for several values of the window size yields both short and long term fairness.

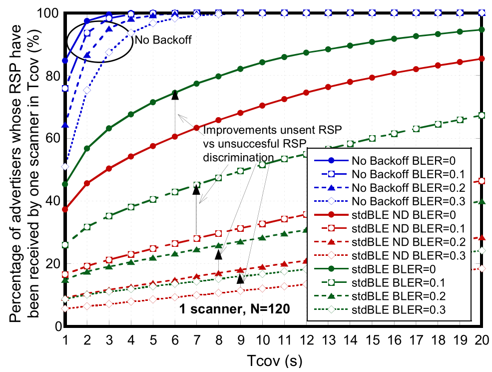


FIGURE 5. Percentage of advertisers detected by one scanner. Comparison between no backoff application, backoff suggested in the standard (named stdBLE ND) and the improvement based on discriminating between unsent and unsuccessfully receptions of SCAN_RSPs (named stdBLE).

First, we evaluate the impact of using physical layer information to discriminate between unsent SCAN_RSP (or undetected SCAN_RSP, see section V) and unsuccessfully receptions of SCAN_RSP. As an example, although the results are generalizable, Fig. 5 depicts the simulation results in a dense IoT scenario with only one scanner and N=120 advertiser devices. Under several error conditions (BLER), it compares the percentage of users from which the scanner receives at least one SCAN_RSP when: 1) no backoff is applied (this is an ideal assumption because in a real scenario the scanner cannot assume the knowledge of the absence of other uncontrolled scanners, and the application of a backoff mechanism is mandatory), 2) the backoff scheme is implemented according with the standard specification (named stdBLE ND, that is with no discrimination) and 3) the proposed modification (named stdBLE). Fig. 5 allows to illustrate how backoff (stdBLE ND) applies unnecessarily, due to collisions of SCAN_REQ and SCAN_RSP with ADV_SCAN_IND and transmission errors, which leads to a severe and unnecessary degradation of the discovery latency (one can compare stdBLE ND and no backoff). Once the degradation is assumed due to the mandatory backoff, the proposal of discrimination (see stdBLE) always allows to improve the efficiency in a very significant way, regardless the absolute quantification of the improvement achieved when a variable number of scanners and advertiser are involved. For instance, in this case, when BLER=0, the scanner has received a SCAN_RSP from 80% of devices in less than 8 s when the discrimination is applied, whereas the discovery latency for the same percentage grows up to 16 s when the standard backoff is applied. Concerning to AwCMID, realize that it works like stdBLE when only one scanner is present and in fact, same as stdBLE, AwCMID can be supported considering both discrimination and non-discrimination. Since that discrimination always improves the performance in both stdBLE and AwCMID, from now on it will be included by default in the rest of the analysis.

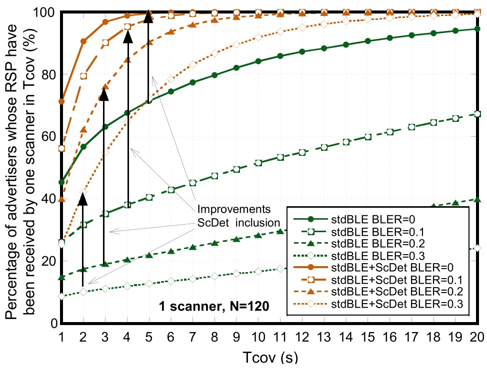


FIGURE 6. Percentage of advertisers detected by one scanner. Comparison between stdBLE and stdBLE applying neighbor scanner detection.

As a complement, in the same scenario (only one scanner is considered), Fig. 6 depicts the improvement achieved when the scanner applies the measurement based method, described in section V, to estimate and decide in a binary way the presence or not of neighbor scanners (denoted as stdBLE+ScDet). Obviously, the ideal can never be reached. The backoff is working and the *upperLimit* grows many times. Actually, the design requires that *upperLimit* grows at least until average values of 4 to make a decision and, as a result, the latency is greater than in the case of not applying backoff. However, the improvement is certainly very significant. In this case, for BLER=0, the scanner has successfully received the SCAN_RSP from 90% of devices in less than 2 seconds. Fig. 6 illustrates a demanding scenario. When the number of devices or, more generally, the overall rate of ADV_SCAN_IND is low, non-detection probabilities of SCAN_REQ (and SCAN_RSP) due to collisions decrease. In this case, the *upperLimit* is maintained in lower values, and the correction achieved due to neighbor detection also has a large impact, but it is less than in dense scenarios.

In Fig. 7, we compare stdBLE and AwCMID in terms of percentage of advertisers whose SCAN_RSP have been received individually by each scanner according with the coverage time (T_{cov}). Comparison is performed for a variable number of devices (from 40 to 120) and scanners (2 to 8), under moderate channel error conditions BLER=0.1. Realize that when two or more scanners are considered, the depicted percentage is obtained by averaging the percentages obtained individually by all the scanners in each realization. Thus, the fairness distribution needs to be considered simultaneously, in order to properly evaluate these results.

Starting with the comparison between stdBLE and AwCMID (see Fig. 7.1a and Fig. 7.3a), the proposed backoff rule (AwCMID) clearly exceeds stdBLE performance. In AwCMID, as logically expected, higher percentages are achieved as the number of scanners and advertisers decreases, due to lower collision probabilities between scanners and with the advertiser transmissions. Nevertheless, stdBLE needs a more careful analysis, considering fairness. Fig. 8.b illustrates the fairness index of stdBLE in the same scenario. The fairness index of AwCMID has not been depicted because it overcomes the value 0.99 in all the cases. That is, all the scanners achieve similar results. Seeing Fig. 8.b, fairness in stdBLE grows as the number of advertisers increases and for high BLER values. When BLER=0.1 and N=40 devices need to be detected, the Jain's indexes are 0.52, 0.28, 0.20 and 0.16 for N_{SC} = 2, 4, 6 and 8 scanners, respectively. These Jain's indexes are close to the theoretical values associated to a completely unfair index. That is, near to $1/N_{SC}$ (0.5, 0.25, 0.16, 0.125, respectively). This means that for N=40 one of the scanners monopolizes the access to the channel in each coverage period (T_{cov}) keeping low *upperLimit* values, whereas the other nodes continue growing their *upperLimit* towards the highest values. In addition, the SCAN_RSP reception is not only challenged by the little number of transmission opportunities: after a SCAN_REQ transmission, the reception of a successful SCAN_RSP is also conditioned by the collisions between SCAN_REQ, SCAN_RSP and ADV_SCAN_IND. This makes the percentage of advertisers discovered by these unfortunate scanners very low. All these facts imply that the percentages depicted in Fig. 7.1a for N=40 (resulting from the averaging of all the scanners) are low. When the number of advertising devices increases (N=80, and N=120), fairness grows. However, the mean of the percentages of advertisers that can be discovered by the scanners continues to be low. All the scanners are affected by the growth of the backoff window due to the multiple types of collisions in the scenario, in addition to the transmission errors. See that the *upperLimit* grows when the number of advertising devices increases and, as a result, the discovery latencies also rise. This justifies the best performance achieved for $N = 80$ versus $N = 120$. Fig. 9.a depicts the mean *upperLimit* achieved by the group of scanners in this scenario (realize that, same as Fig. 7.1a, fairness needs to be taken into account in order to obtain conclusions about the absolute values). Let's note that the results obtained with stdBLE ND have not been included in Fig. 7. Nevertheless, realize that the evolution of the Jain's index towards a more fairness

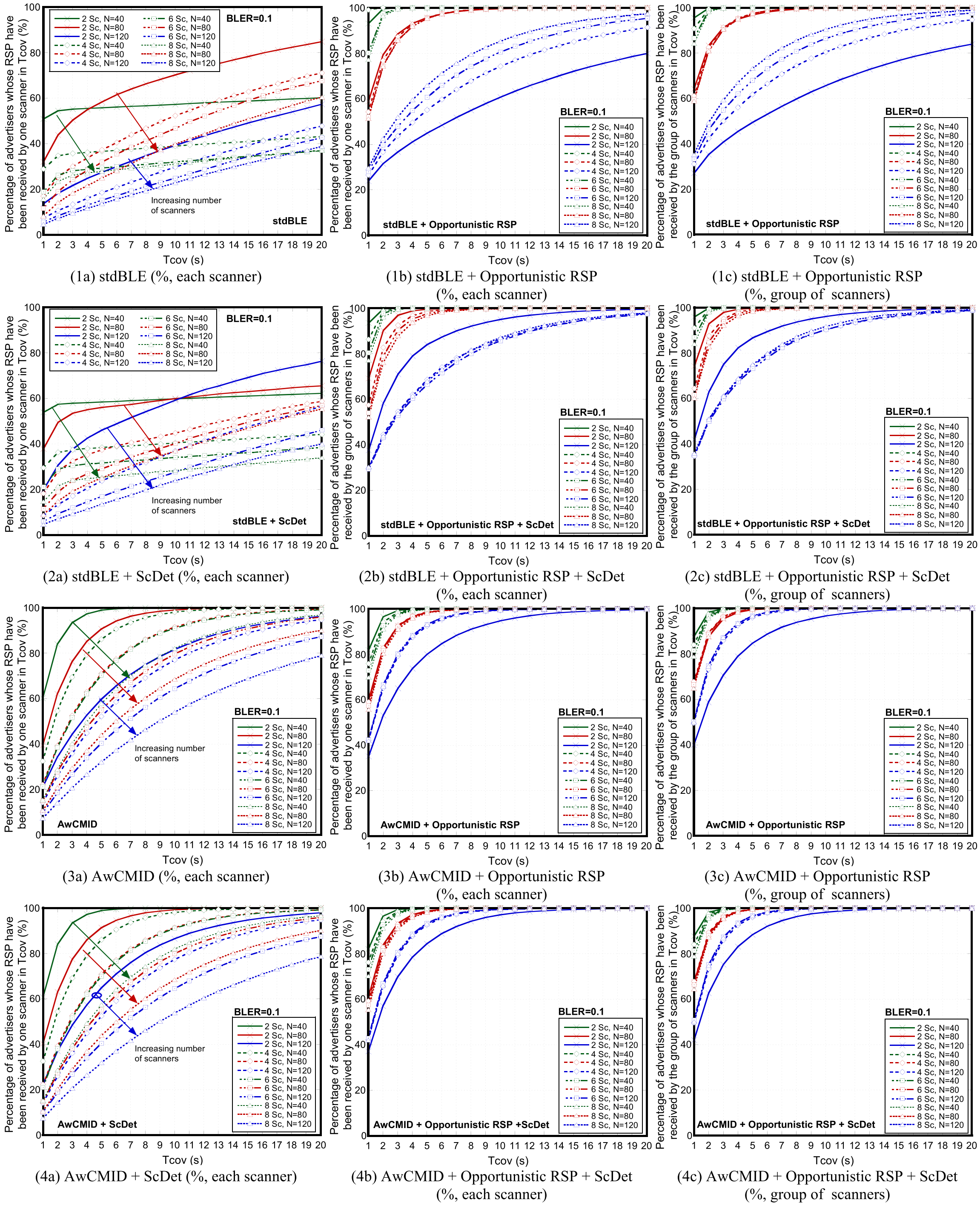
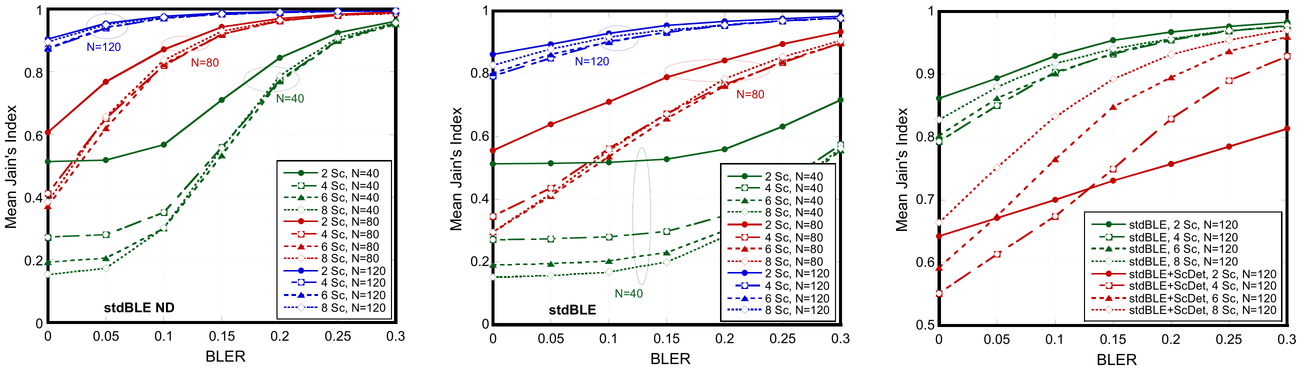


FIGURE 7. **Performance comparison between stdBLE and AwCMID, varying the number of devices and scanners for BLER=0.1. From top to bottom, row 1: AwCMID, row 2: AwCMID applying neighbor detection, row 3: stdBLE and row 4: stdBLE applying neighbor detection. From the left to right: column (a) percentage of advertisers whose RSP have been received individually by each scanner in T_{COV}, (b) same as (a), percentage achieved by one scanner when it applies opportunistic RSP listening and (c) percentage achieved by the group of scanners when applying opportunistic RSP listening.**

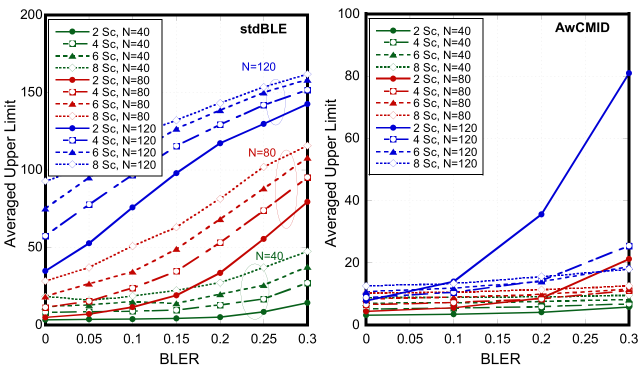
behavior is faster and more pronounced than in stdBLE, when increasing BLER and N (see Fig. 8.a). This is because the lack of discrimination between the types of errors that impact the SCAN_RSP reception is even more pronounced.

Going back to the comparison between stdBLE and AwCMID, by comparing Fig. 9.a with Fig. 9.b, which illustrate the same parameter for AwCMID, we see that AwCMID tends to assign low backoff intervals. In fact, in AwCMID, as the number of scanners increases, the growth of the backoff window according to the BLER is almost negligible. That is, the AwCMID is more protected against factors as the fact that wireless transmission errors and collisions between SCAN_REQ and ADV_SCAN_IND (or SCAN_RSP and ADV_SCAN_IND) apply unnecessarily an increase in the



(a) (b) (c)

FIGURE 8. **Fairness index of backoff schemes, varying the number of devices, the number of scanners and BLER rates. From left to right (a) backoff mechanism proposed in the standard (stdBLE ND), (b) stdBLE including the improvement based on discriminate between unsent and unsuccessfully receptions of RSPs (stdBLE), and (c) comparison between stdBLE and stdBLE adding neighbor detection (stdBLE+ScDet) when N=120.**



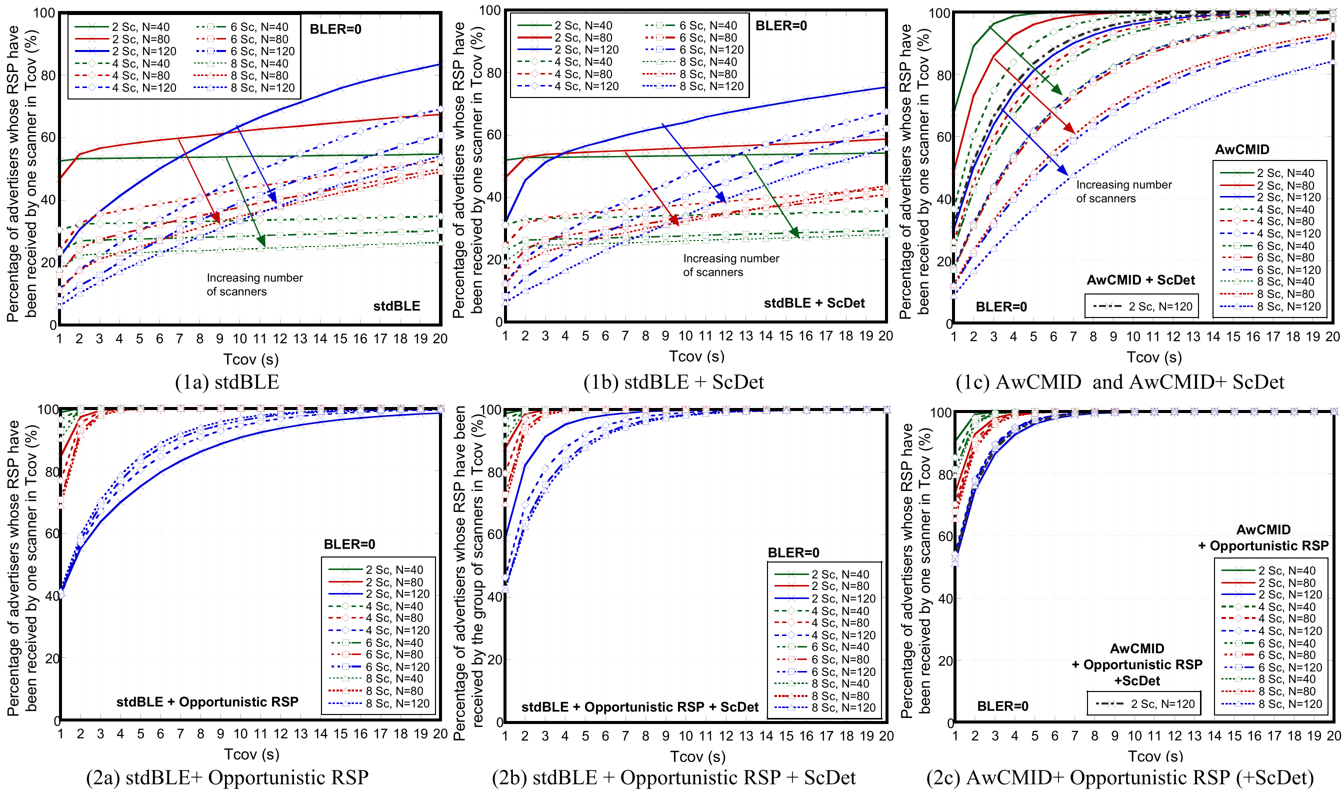
(a) (b)

FIGURE 9. **Comparison of the backoff intervals for stdBLE and AwCMID. (a) stdBLE (b) AwCMID.**

backoff window (*upperLimit*). As shown in Fig. 9.b, when $N_{SC} = 2$, the protection is lower. This fact justifies the observed growth when N=120, since in this case, the probability of collisions with transmitted advertisements is very high.

Fig. 7.2a and 7.4a allows us to compare and evaluate the impact of adding neighbor scanner detection (named ScDet) in stdBLE and AwCMID, respectively. According to the simulation conditions described at the beginning of this section (all the nodes are in the range of each other and all advertisers and scanners are configured identically), an ideal implementation of ScDet should not modify the results compared to Fig. 7.1a and 7.1b, since there are several scanners. However, ScDet is a measurement based method and there is a non-negligible probability of not detecting the presence of neighbors. Nevertheless, the performance of AwCMID and stdBLE is very different. AwCMID is much more robust. In AwCMID, mostly ever, the results obtained with ScDet (see Fig. 7.4a) do not vary compared to those obtained when no detection is implemented (see Fig. 7.3a). Only when the number of neighbors is the minimum ($N_{SC} = 2$) and the number of advertisers is high (N=120), the scheme occasionally estimates the absence of neighbors, and the *upperLimit* is reset to one. In this particular case, the error in the estimation is indirectly beneficial. It compensates for the unnecessary increases in the backoff window due to wireless transmissions errors and collisions, different from collisions between scanners, as we have discussed previously. As a result, the percentage of advertisers the scanners are able to detect in T_{cov} slightly increases. Concerning stdBLE, errors in the estimation are much more frequent, due to the unfair performance of the backoff scheme. The scanners which retain higher opportunities to transmit SCAN_REQ also have a higher probability of not detecting neighbors (their *upperLimit* are low, whereas the neighbors have a high *upperLimit*). As a result, they reset the *upperLimit* to 1, while the other scanners continue to maintain or increase the *upperLimit*. This means that the probability of detecting neighbor is even lower. At the end, stdBLE by applying ScDet becomes even more unfair, as depicted in Fig. 8.c for N=120. Fairness degradation due to ScDet is sharper as the number of scanners (N_{SC}) decreases. On the basis of the arguments that explain the results obtained in Fig. 7.1a (particularly for N = 40), now the results for N = 40, 80, 120 in Fig. 7.2a tend to equalize.

In addition, in Fig. 7 we evaluate the impact of adding opportunistic RSP listening (second column) over stdBLE and AwCMID. Note that opportunistic RSP listening only applies to SCAN_RSP successfully decoded and to SCAN_RSP receptions linked to a SCAN_REQ that has not been sent by the scanner due to backoff. Now, the results of both schemes (stdBLE and AwCMID) improve significantly and differences are reduced. Nevertheless, AwCMID continues providing better performance. In addition, advantages of AwCMID are more significant as the number of advertisers increases. Finally, the third column of Fig. 7 shows the percentage of advertisers whose SCAN_RSP have been received by the group of scanners in T_{cov}. We can see that the percentage reached by the group (applying opportunistic RSP listening) is only slightly better than that achieved by each of



(1a) stdBLE (1b) stdBLE + ScDet (1c) AwCMID and AwCMID+ ScDet

(2a) stdBLE+ Opportunistic RSP (2b) stdBLE + Opportunistic RSP + ScDet (2c) AwCMID+ Opportunistic RSP (+ScDet)

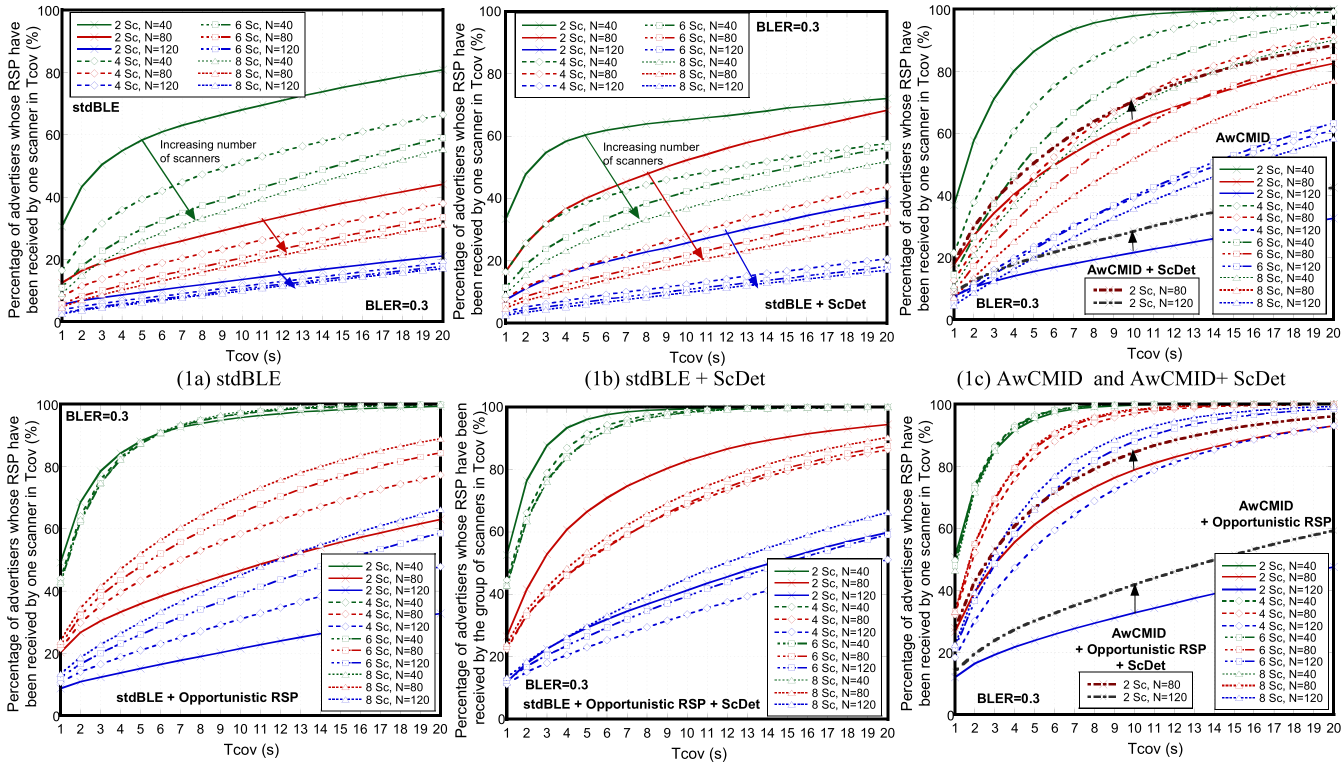
FIGURE 10. Performance comparison between stdBLE and AwCMID in terms of percentage of advertisers whose RSP have been received individually by each scanner in T_{cov}, for ideal channel BLER=0. From left to right (a) stdBLE, (b) stdBLE adding neighbor detection and (c) AwCMID with and without neighbor detection. From up to down, row (1) rules without opportunistic RSP listening and row (2) rules adding opportunistic RSP listening.

the scanners, but the behavior and comparison conclusions are similar. Actually, the cooperation between scanners is not the normal situation. As previously indicated, it is assumed that in the case of deploying an application that involves several scanners, they will be configured in such a way that the frequency scanning pattern is desynchronized. This would reduce the probability of collision and to obtain the greatest diversity possible, by processing the ADV_SCAN_IND (belonging to one advertisement event) in several scanners at the same time.

Same as in Fig. 7, in Fig. 10 and Fig. 11 (even though the distribution of the subfigures is different) we analyze the results obtained for stdBLE and AwCMID for BLER=0 and BLER=0.3, respectively. Now, it has not been considered necessary to include the percentage achieved by the group of scanners (the results of the comparison with respect to individual scanner detection when applying opportunistic RSP listening are similar to those obtained in Fig. 7). On the other hand, results concerning to AwCMID (third column) include both AwCMID and AwCMID adding ScDet. Note that, in this case (same as it occurs when BLER=0.1), ScDet does not modify AwCMID performance in almost any case. Thus, we only represent the cases where changes are appreciated. That is, $N_{sc} = 2$ and N=120 when BLER=0 and N=80, 120 when BLER=0.3. In all the cases, the results for stdBLE and AwCMID are justified by the same effects discussed above. Results for BLER = 0.3 may draw attention in some cases. For AwCMID, when the number of scanners is minimum ($N_{SC} = 2$) and the number of advertisers is high, N = 80 and N = 120, the percentages decrease in relation to those obtained for a greater number of scanners, unlike what happens for BLER = 0 and BLER = 0.1. The reason is that the probability of correctly obtaining the *upperLimit* used by the only neighboring scanner is decreased due to the errors and, as a result, the averaged *upperLimit* grows (see Fig. 9.b).

On the other hand, both in AwCMID and stdBLE, opportunistic RSP listening improves the percentages. However, when BLER=0.3, the probability of opportunistic RSP listening is significantly lower as the smaller the number of scanner is. As a result, in some cases, $N_{SC} = 2$, which provides the better percentages when a scanner works individually, becomes the worst configuration when opportunistic RSP listening is applied.

Beyond this, now, the interest is focused on the quantitative comparison between stdBLE and the proposed scheme (AwCMID). AwCMID is completely fair in all the cases, being the Jain's index always higher than 0.99. AwCMID outperforms stdBLE in all cases (in the basic implementation, by applying ScDet and/or Opportunistic RSP listening), particularly as the number of advertisers and BLER increases.



(1a) stdBLE (1b) stdBLE + ScDet (1c) AwCMID and AwCMID+ ScDet

(2a) stdBLE+ Opportunistic RSP (2b) stdBLE + Opportunistic RSP + ScDet (2c) AwCMID+ Opportunistic RSP (+ScDet)

FIGURE 11. Performance comparison between stdBLE and AwCMID in terms of percentage of advertisers whose RSP have been received individually by each scanner in T_{cov}, for BLER=0.3. From left to right (a) stdBLE, (b) stdBLE adding neighbor detection (named ScDet) and (c) AwCMID with and without neighbor detection. From up to down, row (1) rules without opportunistic RSP listening and row (2) rules adding opportunistic RSP listening.

VII. CONCLUSIONS

BLE and the active discovery process itself is becoming an attractive option as a core of multiple IoT applications, particularly tracking tags in dense scenarios. The backoff mechanism implementation has a great impact on the BLE device discovery process, but the topic has been poorly studied so far. The use of backoff is mandatory, but its definition has been left out of specification in the new BLE 5.0 version, although the backoff scheme specified in BLE v4.2, remains suggested as a possible implementation. The proposed mechanism, joint to the particularities of the active scanning process specification, presents several drawbacks that have been analyzed in detail and quantified in the paper: high probability of unnecessary backoff activation, unfairness and, particularly, lack of efficiency due to poor discrimination of collisions between scanners (which are the ones the backoff mechanism pretends to avoid) and other type of transmission errors and collisions that coexist in the scenario. Without forgetting the peculiarities of BLE, in this paper we review the current collision resolution approaches in BLE and other wireless systems, and critically analyze the feasibility of adaptations to the BLE context. We propose three simple and practical improvements, easily compatible with the definitions of current standards, which clearly outperform the backoff suggested in the standard. 1) We analyze the feasibility of using receiver capabilities of the scanners to improve the discrimination between unsent SCAN_RSP and unsuccessful reception. Many times it is possible to estimate that a SCAN_RSP has been sent even though its reception is erroneous (due to BLER or collision). 2) We quantify the effects of incorporating opportunistic listening of SCAN_RSP which have not been requested. 3) We propose a measurement based mechanism to estimate the absence of neighbor scanners and to mitigate the unnecessary activation and progress of the backoff mechanism in scenarios with only one scanner. These proposals have been made after taking into account the features and practical limitations of real devices. In addition, in this paper we propose an alternative backoff rule, named AwCMID, which is feasible to be implemented with the current specification. The proposal clearly outperforms the backoff mechanism described in the standard. AwCMID solves the unfairness problem, and, besides, provides lower discovery latencies. It is more stable and robust, offering very significant advantages in scenarios with a high concentration of tags to be discovered.

REFERENCES

[1] *Bluetooth Core Specification Version 4.2*, Bluetooth SIG, Kirkland, WA, USA, Dec. 2014.

[2] *Bluetooth Core Specification Version 5.0*, Bluetooth SIG, Kirkland, WA, USA, Dec. 2016.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[4] D. Perez-Diaz-de-Cerio, A. Hernández, J. L. Valenzuela, and A. Valdovinos, "Analytical and experimental performance evaluation of BLE neighbor discovery process including non-idealities of real chipsets," *Sensors*, vol. 17, no. 3, Art. no. 499, Mar. 2017.

[5] A. Hernández-Solana, D. Perez-Diaz-de-Cerio, A. Valdovinos, and J. L. Valenzuela, "Proposal and evaluation of BLE discovery process based on new features of Bluetooth 5.0," *Sensors*, vol. 17, no. 9, Sep. 2017, Art. no. 1988.

[6] K. Cho, G. Park, W. Cho, J. Seo, and K. Han, "Performance analysis of device discovery of Bluetooth low energy (BLE) networks," *Comput. Commun.*, vol. 81, pp. 72–85, May 2016.

[7] K. Cho *et al.*, "Analysis of latency performance of bluetooth low energy (BLE) networks," *Sensors*, vol. 15, no. 1, pp. 59–78, 2015.

[8] G. Park *et al.*, "An adaptive parameter setting algorithm to enhance performance in self-organizing bluetooth low energy networks," *Wireless Pers. Commun.*, vol. 87, no. 3, pp. 953–969, Apr. 2016.

[9] L. Cutrignelli, "Channel randomization for backoff efficiency improvement ," U.S. Patent 8 583 042 B2, Nov. 12, 2013.

[10] J. Kim and K. Han, "Backoff scheme for crowded Bluetooth low energy networks," *IET Commun.*, vol. 11, no. 4, pp. 548–557, Mar. 2017.

[11] T.-T. Yang and H.-W. Tseng, "Two-way communication with wait-slot scheme for neighbor discovery process in dense Bluetooth low energy networks," in *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2017, pp. 1–7.

[12] A. F. Harris, III, V. Khanna, G. Tuncay, R. Want, and R. Kravets, "Bluetooth low energy in dense IoT environments," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 30–36, Dec. 2016.

[13] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-1997, 1997.

[14] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LAN's," in *Proc. SIGCOMM*, London, U.K., Aug./Sep. 1994, pp. 212–225.

[15] J. Deng, P. K. Varshney, and Z. J. Haas, "A new backoff algorithm for the IEEE 802.11 distributed coordination function," in *Proc. CNDS*, San Diego, CA, USA, Jan. 2004, pp. 1–6.

[16] Z. Haas and J. Deng, "On optimizing the backoff interval for random access schemes," *IEEE Trans. Commun.*, vol. 51, no. 12, pp. 2081–2090, Feb. 2003.

[17] N.-O. Song, B.-J. Kwak, J. Song, and M. E. Miller, "Enhancement of IEEE 802.11 distributed coordination function with exponential increase exponential decrease backoff algorithm," in *Proc. 57th Semiannu. IEEE VTC-Spring*, Jeju, South Korea, Apr. 2003, pp. 2775–2778.

[18] T. Panthum, S. Sittichivapark, and J. Sartthong, "Performance analysis of EIED backoff algorithm of the IEEE 802.11 MAC under fading channel errors," in *Proc. 13th JCSSE*, Khon Kaen, Thailand, Jul. 2016, pp. 1–6.

[19] P. Chatzimisios *et al.*, "A simple and effective backoff scheme for the IEEE 802.11 MAC protocol," in *Proc. CITSA*, Orlando, FL, USA, Jul. 2005, pp. 1–6.

[20] C. Wang, B. Li, and L. Li, "A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF," *IEEE Trans. Veh. Technol.*, vol. 53, no. 4, pp. 1235–1246, Jul. 2004.

[21] S. Y. Han and N. B. Abu-Ghazaleh, "On backoff in fading wireless channels," in *Proc. AD HOC-NOW*, Nice, France, Sep. 2008, pp. 251–264.

[22] T. Abdelkader and K. Naik, "A localized adaptive strategy to calculate the backoff interval in contention-based vehicular networks," *IEEE Access*, vol. 2, pp. 215–226, 2014.

[23] I. Syed and B.-H. Roh, "Adaptive backoff algorithm for contention window for dense IEEE 802.11 WLANs," *Mobile Inf. Syst.*, vol. 2016, Art. no. 8967281, May 2016.

[24] I. Syed, S. Shin, B. Roh, and M. Adnan, "Performance improvement of QoS-enabled WLANs using adaptive contention window backoff algorithm," *IEEE Syst. J.*, to be published, doi: 10.1109/JSYST.2017.2694859.

[25] J. Barcelo, B. Bellalta, C. Cano, A. Sfairopoulou, M. Oliver, and K. Verma, "Towards a collision-free WLAN: Dynamic parameter adjustment in CSMA/E2CA," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, Art. no. 708617, Mar. 2011.

[26] Y. He, X. Ma, A. V. Vasilakos, R. Yuan, and W. Gong, "Semi-random backoff: Towards resource reservation for channel access in wireless LANs," *IEEE Trans. Netw.*, vol. 21, no. 1, pp. 204–217, Feb. 2013.

[27] D. K. Klair, K.-W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 3, pp. 400–421, 4th Quart., 2010.

[28] J. Myung, W. Lee, J. Srivastava, and T. K. Shih, "Tag-splitting: Adaptive collision arbitration protocols for RFID tag identification," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 6, pp. 763–775, Jun. 2007.

[29] J. B. Eom and T. J. Lee, "Accurate tag estimation for dynamic framed-slotted ALOHA in RFID systems," *IEEE Commun. Lett.*, vol. 14, no. 1, pp. 60–62, Jan. 2010.

ÁNGELA HERNÁNDEZ-SOLANA received the degree in telecommunications engineering and the Ph.D. degree from the Universitat Politècnica de Catalunya (UPC), Spain, in 1997 and 2005, respectively. She was with UPC and the University of Zaragoza, where she has been an Associate Professor since 2010. She is a member of the Aragón Institute of Engineering Research. Her research interests include 5G/4G technologies, heterogeneous communication networks, and mission-critical communication networks, with emphasis on transmission techniques, radio resource management and quality of service, mobility management and planning, and dimensioning of mobile networks.

DAVID PÉREZ-DÍAZ-DE-CERIO received the M.Sc. degree in telecommunications engineering in 2003 the Ph.D. degree in telecommunication from the Technical University of Catalonia in 2010. In 2003, he joined the WiComTec Research Group, Signal Theory and Communications Department, Universitat Politècnica de Catalunya, as a Collaborating Lecturer. All his lectures are held at the Escola d'Enginyeria de Telecomunicació i Aeroespacial de Catalunya for bachelor's degree, second-cycle, and MAST master students. He has participated as a consultant in several local and European projects in addition to other public and private funded projects. His research interests are wireless communications systems; especially those based on the IEEE 802.X standards and their use in e-health applications and IoT.

ANTONIO VALDOVINOS received the degree in telecommunications engineering and the Ph.D. degree from the Universitat Politècnica de Catalunya (UPC), Spain, in 1990 and 1994, respectively. He was with UPC and the University of Zaragoza, where he has been a Full Professor since 2003. His research interests include 5G/4G technologies, heterogeneous communication networks and mission-critical communication networks, with emphasis on transmission techniques, radio resource management and quality of service, mobility management, and planning and dimensioning of mobile networks.

JOSE LUIS VALENZUELA received the degree in telecommunication engineering and the Ph.D. degree from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 1993 and 1997, respectively. He is currently an Associate Professor with the Signal Theory and Communications Department, UPC. After graduation, he was concerned with equalization techniques for digital systems. He has also been involved in the field of digital communications with particular emphasis on digital radio and its performance under multipath propagation conditions. His research interest is in the field of wireless sensor networks and wireless communications systems.