Exploiting an Optimal Delay-Collision Tradeoff in CSMA-Based High-Dense Wireless Systems

Exploiting an Optimal Delay-Collision tradeoff in CSMA based high-dense Wireless Systems

Nicola Cordeschi, *Member, IEEE,* Floriano De Rango, *Member, IEEE,* and Mauro Tropea, *Member, IEEE*

***Abstract*—A novel carrier sense multiple access strategy with collision avoidance (CSMA/CA) balancing contention probability and channel access time is proposed. The approach can be applied to any context where the computational simplicity of the MAC must be preferred to the complexity of the channel access strategy. Our MAC, called Delay-Collision CSMA (DC-CSMA), is a slotted nonpersistent CSMA/CA with nonuniform contention probability distribution, designed to reduce at the same time latency of contenders and preserve a high successful access probability. An utility function aiming at equalizing the effects of these two performance metrics is introduced, and the related theoretical properties and optimal distribution are derived. DC-CSMA is insensitive to the number of contenders and very robust with respect to contention window size, packet length, and impairments such as frame synchronization errors and hidden terminals, and it does not require any adaptive tuning to optimize its performance. Current technologies such as WSN, RFID, IoT devices can benefit from such a simple access technique. The numerical evaluation has been led out considering latency, successful probability and throughput, and DC-CSMA has been compared with other classical strategies such as CSMA with uniformly distributed contention probability, CSMA/p^* and Sift distribution.**

***Index Terms*—CSMA/CA, collision probability, WSN, mMTC, constrained-wireless systems, IoT, channel access delay, utility function, optimization problem.**

I. Introduction and related work

In the last years, many medium access control (MAC) strategies have been proposed for different technologies as WSN, Internet of Things (IoT), RFID systems or any other device with some limitations in computation, delay, throughput and so on [1]. A lot of interest has been given to scalable MAC approaches able to maintain good performance also in extreme cases where the number of nodes dramatically increases, or where the node density estimation is not precise due to many reasons such as node mobility, link failures, heavy energy-saving sleeping schedule policies etcetera [2], [3]. Considering the heterogeneity of the arising technologies such as 5G, WLAN, IEEE 802.15, 802.11ax, 802.11ah etc., simple MAC strategies with guaranteed theoretical bounds and scalability must be exploited to adapt to the specific operative conditions and constraints. In this respect, carrier sense multiple access with collision avoidance (CSMA/CA) is among the most prevalent techniques due to their intrinsic capability to cope with the high level of dynamism and heterogeneity of modern wireless networks in a very simple low cost way. Quite interesting contributions based on CSMA/CA techniques have been proposed, applied to WSN or RFID systems [4], [5], or integrated into vehicular and industrial delay-sensitive control networks, monitoring applications, IoT over WiFi and other relevant current technologies [6]–[9]. The works [8]–[13] develop analytical and simulation models for assessing the performance of IEEE 802.X CSMA/CA families and dimension their parameters, analyzing the impact of number of backoffs, congestion window size, backoff exponent, mainly by means of Markov Chain (MC) or recursive throughput analysis. On the whole, the analysis carried out by [8]–[13] reveals that the binary exponential backoff (BEB) collision avoidance feature is the root cause of quality of service (QoS) flaws of IEEE 802.11, 802.14, 802.16 families. Collided stations are inherently not fairly treated, this producing a significant delay variation among nodes. The regenerative model of the Restricted Access Window (RAW) slot in IEEE 802.11ah is a first step to handle this disequilibrium [8], aiming at adapting Wi-Fi networks to the emerging IoT and reduce contention when up to thousands of IoT devices operate in the same area. However, some inherent drawbacks of the standard are emphasized in [9]. Real-time wireless control and monitoring applications usually have varying degrees of timeliness requirements, and the basic IEEE 802.11 MAC mechanism cannot support them, most of the existing research on enhancing IEEE 802.11ah RAW only aiming at improving the throughput or the energy efficiency without considering explicitly the delay. As authors claim in [8], when the backoff function is *regenerated* for all the stations *without* any freezing, the contention changes significantly and this is not properly taken into account by the research community. Due to above discussed strength of regeneration in IoT contexts, in this paper we explicitly model a backoff strategy with Contention Window (CW) regeneration.

[6], [7], [14], [15] propose some modifications to the scheme implemented by the IEEE 802.X families, by adjusting the initial backoff window size or controlling the amount of traffic entering the network, some of them also comparing their approach with nonuniform distributions. On the other hand, since CW sizes equal for all nodes naturally results in good short-term fairness which lower the delay, [4], [5], [16]–[20] propose protocols where contending nodes do not perform the BEB algorithm after collisions, but rather use or dynamically converge to similar values of their contention windows. Even in this case, it is crucial to set the CW size carefully. While

N. Cordeschi is with 5G & 6G Innovation Centre, University of Surrey, Guildford, GU27XH, United Kingdom (e-mail: n.cordeschi@surrey.ac.uk). F. De Rango and M. Tropea are with the Department of Informatics, Modeling, Electronics and Systems, University of Calabria, Italy, Rende, 87040 Italy (e-mail: f.derango@dimes.unical.it, m.tropea@dimes.unical.it). F. De Rango is also with Consorzio Nazionale Italiano Telecomunicazioni (CNIT).

Corresponding author: Nicola Cordeschi, n.cordeschi@surrey.ac.uk

outperforming the traditional BEB of IEEE 802.11, both the performance and the optimized CW size of these approaches are still too sensitive to the number n of nodes, making them useful for limited communication range and low deployment densities, but not in massive high-density scenarios. We aim at proposing an approach strongly robust with respect to this important matter.

To tackle these problems [4], [5], [18]–[20] develop slotted fixed window CSMA protocols with uniform and nonuniform attempt probability distributions. Throughput and delay performance of these approaches are almost independent of the number of contenders, achieving simplicity and good throughput under both low and high contention, so fulfilling their target to be eligible for WSN, IoT, RFID. However, they do not adequately consider two major aspects. First, since they use constant CW sizes, their setting is still critical: even though they overcome the dependence of the CW on the number of active nodes, they are still quite sensitive to the CW size. They do not properly assess or control the sensitivity of performance to the choice of CW, challenging when other system parameters (like, for example, packet length) change, or in presence of non-idealities like estimation errors or hidden terminals. We deeply investigate this point. Second, to the best of our knowledge, until now nonuniform distributions with CW regeneration have mainly focused on the optimization of the access successful probability, i.e., just handling the part of delay due to collision. However, as we will show, this is not always appropriate.

With this in mind, the focus of this paper is to exploit the right tradeoff between the head-of-line *channel access delay* and *collision probability*, and extend the range of possible CSMA/CA families. We propose a novel optimized nonpersistent strategy called Delay-Collision CSMA (DC-CSMA), whose target is to try to be less conservative in comparison with other strategies and achieve a balanced tradeoff between these two opposed metrics. We compare DC-CSMA with well known nonpersistent policies with uniformly distributed contention probabilities (U-CSMA), the CSMA optimized to maximize the frame successful probability (CSMA/p^*), and Sift, a very robust policy under node estimation errors.

The paper is organized as follow: Section II presents the problem statement and a detailed analysis of the main involved system parameters, while DC-CSMA and its theoretical optimality properties are derived in Section III. Latency and normalized system throughput performance evaluation for clique saturated networks are presented in Section IV. Section V discusses CW size optimality and complexity issues, while Section VI presents performance evaluations in presence of hidden terminals and other non-idealities. Finally, Section VII concludes the paper with a final discussion.

II. Controlling the attempt probability: system model and problem setup

Let n be the number of contenders (nodes, sensors, stations, etc.) that simultaneously attempt the transmission in the current frame. We assume that all nodes are synchronized and start a frame at the same time, and we reserve the term *slot* to indicate the unit time in the contention window (CW). That is, we model a slotted CSMA/CA system for a massive clique network in saturated traffic conditions, where each node always has packets to transmit, and, furthermore, each node can sense and is affected by each other transmission that is taking place in the clique[1]. Each node independently picks a slot in the CW according to its own probability distribution (for example, the standard uniform distribution). Let K denote the contention window size, i.e., the number of contention slots in the CW, and let p_i denote the probability for a contender to choose slot i to attempt the transmission. If a node finds the selected slot already busy, occupied by a transmission in progress, it waits until the channel is idle and tries again by picking a new slot. We denote by $p = \{p_i\}_{i=1..K}$ the nodes' attempt probability distribution on slots. The probability $P_s(i)$ of a successful transmission in the slot i is the probability that a node chooses slot i, while all other nodes do not select any slot from 1 to i, and is given by:

$$P_s(i) = np_i \left(1 - \sum_{l=1}^{i} p_l\right)^{n-1}. \tag{1}$$

Hence, the total probability of success P_s over the contention window is given by the sum of the probabilities (1):

$$P_s = \sum_{i=1}^{K} P_s(i). \tag{2}$$

Tay et al. [4] propose a nonpersistent carrier sense multiple access scheme named CSMA/p^*, with p^* being the probability distribution that maximizes the probability of success P_s in (2), also discussing some assumptions under which p^* minimizes latency in event-based workload. In order to explore other possible directions to maximize performance and expand the set of probability distributions of potential interest, let us call a distribution p *asymptotically optimal* in probability (a.o.p.) if $\lim_{K\to\infty} P_s(K) = 1$[2]. In this sense, p^* in [4] is an a.o.p. distribution. In fact, by maximizing (2) for each finite value of K, p^* is the distribution with the highest rate of convergence of P_s to 1 (for example, in a saturated network with $n = 200$ contending nodes, the probability of success P_s^* provided by CSMA/p^* is already equal to 0.94 for $K = 32$ contention slots).

However, in the general case, a policy maximizing (2) (or, more weakly, any a.o.p. distribution) is not guaranteed to also obtain optimized performance either in terms of latency or throughput. This is occasioned by the fact that P_s is just an intermediate term in the performance metric of a CSMA scheme, although fundamental. In addition to it, a

[1] We will relax these assumptions in Section VI.

[2] In fact, for each policy, the attempt probability distribution $p = p(K)$ is a *family* of distributions, depending on the content window size K and, possibly, other system parameters (e.g., current and/or maximum number of contenders the system can handle, etc.) and the same dependence reflects in performance metrics like $P_s(K)$ in (2). Unless needed, throughout the paper we do not indicate explicitly this dependence to avoid cumbersome notation.

key performance parameter for a random access scheme is the expected successful slot number

$$\overline{\ell}_s = \sum_{i=1}^{K} iP_s(i). \tag{3}$$

If on the one hand it is important to maximize P_s, on the other is essential to keep $\overline{\ell}_s$ as low as possible. Table I shows the probability of success, the expected successful slot number and the head-of-line latency, i.e. the latency of first transmission (defined as the expected delay for successful transmission in multiples of slot time) for CSMA/p^* and the standard nonpersistent uniform distribution (U-CSMA), in a low-density scenario with 10 contending nodes and a packet length equal to 40 slot-times. It turns out that, as expected, by increasing the contention window size K both policies improve their probability of success, but the expected successful slot number $\overline{\ell}_s$ of CSMA/p^* always settles around $33\% - 35\%$ of the CW, causing a significant drop in latency and a high sensitivity of CSMA/p^* to the value of K. This is due to the fact that for the CSMA/p^* scheme $P_s(i)$ in (1) *linearly* decrease over time slots, spreading too much its successful probability in order to maximize it, whereas the U-CSMA one exponentially decreases so that concentrates in the first few slots. As an effect (see Table I), CSMA/p^* is not guaranteed to always have better performance than the uniform one, at least in low-density scenarios.

TABLE I: Probability of success, expected successful slot number and latency for increasing values of CW size

K (n=10)	16	32	64	128
P_s (CSMA/p^*)	0.90	0.95	0.97	0.99
P_s (U-CSMA)	0.72	0.85	0.92	0.96
$\overline{\ell}_s$ (CSMA/p^*)	5.57	10.96	21.68	43.08
$\overline{\ell}_s$ (U-CSMA)	1.40	2.88	5.80	11.63
L (CSMA/p^*)	11.13	14.61	24.30	45.14
L (U-CSMA)	18.61	11.04	10.16	14.24

On the other hand, as illustrated in Table II, the probability of success of CSMA/p^* is quite stable and insensitive to the number n of contenders, whereas uniform distribution one dramatically drops for increasing node density. This is

TABLE II: Successful probability for increasing values of contending nodes

n (K=16)	5	10	15	30	60
P_s (CSMA/p^*)	0.91	0.90	0.90	0.90	0.89
P_s (U-CSMA)	0.85	0.72	0.60	0.33	0.08

due to the fact that for for U-CSMA we have [4] $P_s \approx (n/K)/(e^{n/K} - 1)$, showing as U-CSMA performance are highly sensitive to the ratio n/K, preventing its effective applicability to a really high node-density scenario.

Motivated by the above analysis, a challenging problem is to search for a CSMA scheme able to maximize P_s in (2) *by controlling* the way the probability of success is spread among slots in (1) in order to keep $\overline{\ell}_s$ low and best capture qualities of both CSMA/p^* and U-CSMA policies. To proceed in this direction, in the next subsection we provide a mathematical system model for the latency of schemes with CW regeneration and investigate the relationship among the performance parameters introduced above and the way they differently affect latency in different application scenarios, presenting our proposal.

A. Latency performance of CSMA schemes with CW regeneration

Let T_s and T_p denote the duration of a contention slot and a packet transmission, respectively. We define the latency L as the expected head-of-line access delay for a successful transmission. Let τ be the the propagation delay plus the time for carrier sensing and transceiver switching from receive to data send. If a transmission starts at slot i, with probability given by (1), the latency is $(i-1)T_s+\tau$. Since collisions only occur at the beginning of a packet transmission (in other slots they can be sensed by the other transmitting nodes), the time wasted due to collisions is the same as the transmission time of a packet, and the latency is $(i-1)T_s + \tau + T_p + L$. With regard to the attempt distribution p, as in [21] we consider the more general weaker distribution constraint:

$$\sum_{i=1}^{K} p_i \leq 1, \tag{4}$$

(i.e., we assume that each node can decide to *contend* in a frame or remain *idle* skipping to the next frame, with a possibly non-zero probability p_0 of not choosing any slot). In such a case, if all contenders skip the frame, the latency is $KT_s + L$. Altogether, assuming the propagation delay plus sensing plus switching time proportional to the slot time ($\tau \approx T_s$)[3], the latency is:

$$L = \sum_{i=1}^{K} \left[P_s(i)\,(iT_s) + P_c(i)\,(iT_s + T_p + L)\right] + P_\varnothing\,(KT_s + L)$$

where $P_\varnothing$ is the possibly non-zero probability for the CW to remain idle, and $P_c(i)$ is the probability of collision in slot i. Rearranging terms, latency can be expressed as

$$L = \left(\frac{\overline{\ell}_s + \overline{\ell}_c + P_\varnothing K}{P_s}\right) T_s + \left(\frac{(1-P_\varnothing)}{P_s} - 1\right) T_p, \tag{5}$$

with $\overline{\ell}_c = \sum_{i=1}^{K} iP_c(i)$ being the expected slot number when the collision begins. To give an insight into the diversified way system parameters affect latency and generally system performance, we examine two interesting limit cases.

1) High packet length/time slot ratio ($T_p/T_s \to \infty$): we can see as in the limiting case of sufficiently large T_p with respect to T_s, latency is well approximated by the latter term of the sum (5), that, for distributions p forced to choose a slot in the CW (like in the standard uniform distribution or CSMA/p^*, where $\sum_{i=1}^{K} p_i = 1$), leads to:

$$L \approx \left(\frac{1}{P_s} - 1\right) T_p, \tag{6}$$

[3]This is not a model limiting assumption. Depending on the considered application, the opposite case of τ negligible compared to T_s ($\tau \ll T_s$) leads to a latency L' equal to $L' = L - \frac{(1-P_\varnothing)}{P_s} T_s$, with L being the one in (5), with a difference $L' - L$ of the order of the time slot.

confirming that CSMA/p^*, by maximizing P_s, has indeed in this asymptotic case the smallest latency.

However, from one hand, as detailed in standard documents [22], [23] data packets are typically very small in sensor and control networks and, from the other hand, due to the progress in PHY layer in the last decade (increased speed, bandwidth, multiple spatial flows [24]) even in more general wireless networks it is becoming more and more mandatory to also examine the following second opposite limiting case.

2) Small packet length/time slot ratio ($T_p/T_s \to 1$)*:* When the packet-length is of the order of the time-slot duration, $T_p \approx T_s$, from (5) we have: $L \approx \left(\overline{\ell}_s/P_s\right) T_s + \left(\overline{\ell}_c + (K-1)P_\varnothing + (1-P_s)\right) T_s/P_s$. Let $P_c = \sum_{i=1}^{K} P_c(i)$ be the (total) frame collision probability. Being $\overline{\ell}_c \leq K P_c$, for the latency's second term we can write the following upper bound:

$$\left(\frac{\overline{\ell}_c + (K-1)P_\varnothing + (1-P_s)}{P_s}\right) \leq (K+1)\left(\frac{1}{P_s} - 1\right).$$

From Table I, $K = 32$, for the CSMA/p^* scheme we can see as $(K+1)\,(1/P_s - 1) = 1.74$, whereas $\overline{\ell}_s/P_s = 11.54$, one order of magnitude higher, so that:

$$L \approx \overline{\ell'}_s T_s, \tag{7}$$

with $\overline{\ell'}_s \triangleq \left(\overline{\ell}_s/P_s\right)$ being the conditional successful slot number (i.e., the expected successful slot number if there is a success in the contention window). Hence, for small values of the packet-length T_p, comparable to T_s (actually, at least up to two order of magnitude higher than T_s, as we will show in Section IV), the probability of success is not the best parameter to be optimized to improve performance.

We also refer to (6) and (7) as the contributions to the latency related to the average time wasted in collisions (*wasted* time) or waiting for a transmission (*waiting* time), respectively. As seen, (6) and (7) control latency each in a specific situation.

Finally, in the general case, regardless of the relationship between T_p and T_s, from (5) we have $L \geq \overline{\ell'}_s T_s$ for any distribution p, this lower bound becoming tight for any a.o.p. distribution for increasing values of K. Therefore, in order to develop schemes able to behave well in terms of probability of success (that is, a.o.p. distributions) *and*, at the same time, scalable with regard to the choice of K, it is important to turn the attention on the effect of the CW size on the expected successful slot number, and search for strategies able to keep $\overline{\ell'}_s$ low and as insensitive as possible to the choice of K, preserving the possibility to *i)* set K to fulfill the desired successful probability *without* having to pay the drawback of a high expected successful slot number that would impair latency (and throughput), and *ii)* be scalable and possibly compliant with legacy (so to be, for example, usefully integrated in already working BEB strategies like in the CSMA/CA IEEE 802.11 standard family, or any other optimized scheme able to adequately control K, like [6], [7], [16]).

To move in this direction and take account of the tradeoff between these two performance metrics, P_s and $\overline{\ell}_s$, we propose to maximize the following utility function:

$$U_K(P_s, \overline{\ell}_s) = P_s - \frac{\overline{\ell}_s}{K+1}. \tag{8}$$

DC-CSMA will be the optimal nonpersistent strategy related to it. Before proceeding further, let us stress here that we will develop DC-CSMA under the general weaker constraint (4), and that throughout the paper we compare it with an improved version of CSMA/p^*, optimized over K slots and subject to the same constraint (4). To avoid any ambiguity, we will call CSMA/p^+ the policy originally proposed in [4], i.e. the one optimized over $K-1$ slots and satisfying the stricter constraint: $\sum_{i=1}^{K} p_i = 1$.

Remark (*Some insights about the choice (8)*)*:* The utility function (8) is a special case of the more general one: $U_K(P_s, \overline{\ell}_s, \theta_K) = P_s - \overline{\ell}_s/\theta_K$, identifying the Pareto tradeoff between the two metrics P_s and $\overline{\ell}_s$ that control the wasted time (6) and the waiting time (7), respectively, namely the two contrasting targets that is essential to optimize in order to control latency (5). From this perspective, both CSMA/p^* and DC-CSMA can be incorporated in this more general common mathematical framework as special cases. We explicitly point out that the proof reported in the Appendix leading to the optimal DC-CSMA expression generalizes to any possible choice of θ_K, thus allowing to control the whole Pareto family and switch between DC-CSMA and CSMA/p^* in a smooth manner. In this sense, our analysis generalizes the one in [4] that led to CSMA/p^*, and contains it as a particular case. Based on these observations, some justification is needed for our specific choice $\theta_K = K+1$. In this regard, for any given K, when θ_K is too high, U reduces to P_s and to the CSMA/p^* strategy, only controlling the wasted time (6) due to collisions. On the contrary, when θ_K is too small ($\theta_K \leq 1$), the optimal solution becomes the null one (i.e., obtaining a success is not considered valuable enough compared to the time needed to have it, and the best choice is to refrain completely from transmission). Being for a generic distribution $\overline{\ell}_s$ bounded above by K, it can be shown that to avoid trivial solutions and keep U bounded in the valuable interval $[0, 1]$ without limiting the distribution feasibility space we must have $\theta_K > K$. Thus, by setting $\theta_K = (K+1)$ and rewriting the utility in terms of $\overline{\ell'}_s$ as $U = P_s\left(1 - \overline{\ell'}_s/\theta_K\right)$, we obtain the following straightforward lower and upper bounds on P_s and $\overline{\ell'}_s$, respectively:

$$P_s \geq \overline{U}; \qquad \overline{\ell'}_s \leq (K+1)(1-\overline{U}), \tag{9}$$

showing as (8) is able to bound both the wasted time and the waiting time through the unique number $\overline{U}$, irrespective of the specific protocol dependent scenario, data unit format and any other details of data transmission. From (9) it is immediate to recognize that greater values of θ_K would provide less tight bounds for the conditional successful slot number (and the corresponding waiting time (7))[4].

In the following Section III we derive the optimal distribution for (8) and its properties. Based on the above *Remark*,

[4] θ_K might also be a real number in $]K, K+1[$, but this is just a mathematical sophistication and does not affect the analysis.

we stress that the optimality results we detail in the following do not extend to a general θ_K, but are explicitly related to our optimized choice in (8).

III. The optimal CSMA distribution for collision-delay tradeoff

We now validate the choice (8) by deriving some of its optimality properties. We begin with a result related to the probability of success P_s.

Proposition 1 (*Asymptotic successful probability optimality*)*:* Let p the probability distribution that maximizes (8). p is asymptotically optimal in probability (a.o.p), i.e., its probability of success approaches 1 for increasing values of the contention window size K.

Proof: The proof is by contradiction. Let $p(K)$ be the distribution that maximizes (8) for a contention window size of K slots, and let $P_s(p(K))$ denote the successful probability achieved by such a distribution. For a possibly different size K^*, let $p^*(K^*)$ and $P_s(p^*(K^*))$ be the CSMA/p^* distribution maximizing (2) and the corresponding optimal successful probability, respectively. Since $\lim_{K^*\to\infty} P_s(p^*(K^*)) = 1$ [4], to show that $p(K)$ is a.o.p. as well, it is sufficient to prove that for any K^* and any $f \in]0,1[$ there always exists a value K such that

$$P_s(p(K)) \geq f P_s(p^*(K^*)).$$

To this end, note that $p^*(K^*)$ is also feasible for the CW with $K > K^*$ slots[5]. Hence, being by definition $p(K)$ the optimal distribution for $U_K(\cdot)$ in (8), we can write:

$$U_K(p^*(K^*)) \leq U_K(p(K)). \tag{10}$$

From the definition (8) we have $U_K(p(K)) \leq P_s(p(K))$. Let us assume by contradiction that there exist a K^* and a $f \in]0,1[$ such that $P_s(p(K)) < f P_s(p^*(K^*))$ for *any* $K > K^*$. Being $\overline{\ell}_s(p^*(K^*))$ the expected successful slot number achieved by $p^*(K^*)$, from the last two inequalities and (10) we derive:

$$(1-f)P_s(p^*(K^*)) < \frac{\overline{\ell}_s(p^*(K^*))}{K+1}, \tag{11}$$

getting a contradiction, since the right term of (11) tends to 0 as K goes to infinity, whereas the left term is fixed and strictly positive. This shows that 1 is a limit point for $P_s(p(K))$. Being U_K monotone and bounded in the interval $[0,1]$, it is easy to see that it is the only limit point and the successful probability converges: $\lim_{K\to\infty} P_s(p(K)) = 1$. □

Essentially, *Proposition 1* ensures that by increasing the number of slots in the CW is always possible to achieve the desired successful probability. As previously highlighted, this result cannot be obtained simply through the examination of the utility function expression (8), i.e., U_K does not converges to P_s for increasing values of K. As a consequence, despite achieving the same optimality property in terms of probability of success P_s, the distribution p maximizing (8) does not retain the same poor CSMA/p^*-like behavior in terms of $\overline{\ell}_s$. In this regard, we can state the following optimality result.

Proposition 2 (*Successful slot number increase*)*:* For the distribution p, the expected successful slot number $\overline{\ell}_s$ only increases *sublinearly* with the contention window size K, i.e.,

$$\lim_{K\to\infty} \frac{\overline{\ell}_s(K)}{K} = 0. \tag{12}$$

Proof: From *Proposition 1*, for any $\varepsilon > 0$ there exists $\overline{K}$ such that for any pair (K, K^*), with $K > K^* > \overline{K}$, we have: $|P_s(p(K)) - P_s(p(K^*))| < \varepsilon/2$. Furthermore, for such a $K^* > \overline{K}$ we can always choose $K > K^*$ so that: $\overline{\ell}_s(K^*)/(K+1) < \varepsilon/2$. From the monotonicity of $U_K(\cdot)$ we have $U_K(p(K^*)) \leq U_K(p(K)))$, that can be re-arranged as:

$$\frac{\overline{\ell}_s(K)}{K+1} \leq |P_s(p(K)) - P_s(p(K^*))| + \frac{\overline{\ell}_s(K^*)}{K+1} < \varepsilon,$$

completing the proof. □

In Section IV we will evaluate the combined effect of these optimality results on latency and throughput performance. We now proceed to define the optimal probability distribution p and discuss some implementation aspects. We defer the proof of its optimality to the Appendix.

Definition (The optimal distribution for (8)): Let p'_i be the probability of choosing slot i conditioned on not choosing any slot before i (i.e., the conditional attempt probability), and let $V \triangleq (K+1)U_K$ a renormalized version of the utility function U_K in (8). Let r_i be the "optimal reward" measured at slot i, defined as the optimal value of the renormalized utility function if the policy decided not to attempt transmission in the first $(i-1)$ slots, i.e., it is the maximum utility that can be reached by holding off the first transmission attempt until slot i[6].

The conditional probability distribution p'_i and the optimal reward r_i can be calculated via the following backward recursion, for $i = K, \ldots, 1$:

$$p'_i = \frac{(K-i+1) - r_{i+1}}{n(K-i+1) - r_{i+1}} \tag{13}$$

$$r_i = (K-i+1)\left(1-p'_i\right)^{n-1} \tag{14}$$

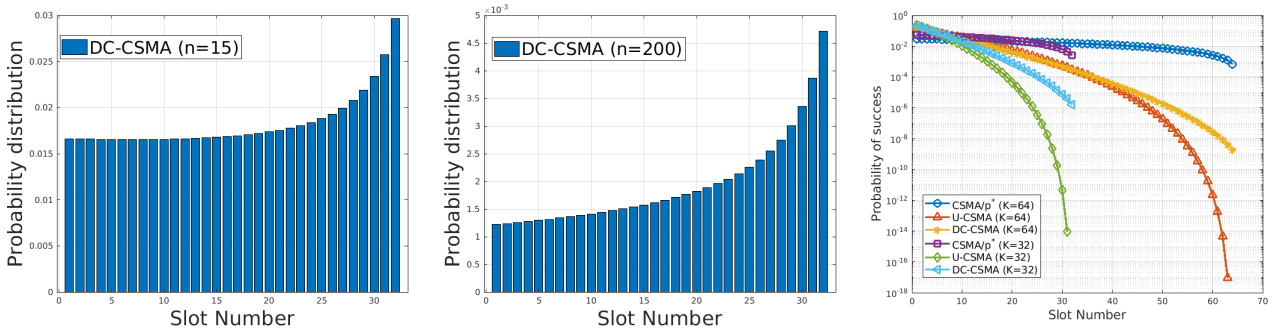
with the starting value r_{K+1} being the reward for keeping the entire contention window idle, clearly equal to zero ($r_{K+1} = 0$). Finally, the optimal nonpersistent probability distribution p is given by

$$p_i = \left(\frac{(K-i+1) - r_{i+1}}{n(K-i+1) - r_{i+1}}\right)\left(1 - \sum_{l=1}^{i-1} p_l\right) \tag{15}$$

for $i = 1, \ldots, K$, that is, it can be calculated as a function of the conditional probability as $p_i = p'_i(1 - \sum_{l=1}^{i-1} p_l)$ via a forward recursion. □

[5]Each distribution feasible for K^* can trivially be used for a contention window size $K > K^*$, by setting its slot attempt probabilities in the interval $[K^*+1, K]$ to 0.

[6]The optimal distribution p is obviously not affected by the renormalization and the optimum rewards in terms of the original U_K in (8) are equal to $r_i/(K+1)$. We renormalized in order to keep expressions (13), (14) in a more compact form and better show the intuition behind the policy.



(a) Contention probability distribution (b) Contention probability distribution (c) Probability of success

Fig. 1: (a)-(b) DC-CSMA's contention probability distribution over slots for two values of n and $K = 32$. (c) Distribution of the probability of success over slots (1) for different values of contention window size K and $n = 10$ contenders.

Fig.1 shows the DC-CSMA attempt probability distribution over slots for $K = 32$ and $n = 15, 200$ (low and medium-high node density scenario). Unlike CSMA/p^* [4], for DC-CSMA the distribution is no longer guaranteed to be strictly increasing, its shape depending on the number n of contenders. Specifically, in order to handle both P_s and $\overline{\ell}_s$ at the same time, distribution p tends to spread more fairly the attempt probabilities over slots, especially in less collision-affected scenarios (i.e., low values of n), where it does not sacrifice much the first slots of CW, aiming at keeping the expected successful slot number as low as possible.

In fact, from Fig.1 we can see as DC-CSMA quite approaches a flat distribution for $n = 15$, at least for the first 24 slots. On the other hand, in more deeply competition affected scenarios (and, consequently, more exposed to collision), where a probability of success P_s high enough is harder to achieve, the delay-collision distribution lowers down the attempt probability on first slots and progressively increases it. In this case (Fig.1, $n = 200$), DC-CSMA and CSMA/p^* distributions have a more similar trend, both increasing slowly in the first slots and more rapidly in the last few ones. Nevertheless, despite the similar behavior, even in the high-dense scenario they have different shapes and the delay-collision distribution p does not reduce to a scaled-up version of p^*.

As mentioned above, throughout the paper we set CSMA/p^* to effectively work on K slots instead of $K - 1$, in order to ensure a fair comparison and not penalizing it especially for low values of CW size, neither in probability of success nor in latency (i.e., we removed the stricter constraint $p^*_K = 1 - p^*_1 - \ldots - p^*_{K-1}$)[7]. Table III shows the expected number $\overline{n}_t$ of transmission attempts over the CW for some values of n and K, that is the average number of nodes out of the total number n that decide *not to skip* to the next frame according to what allowed by (4), but pick up a slot in the current window. It turns out that the number of nodes expected to attempt transmission under DC-CSMA is approximately twice or three times higher than the one under CSMA/p^*[8]. Hence, p is more aggressive that p^*, especially for high values of K and a large number of contenders. Fig.1(c) compares the resulting successful probability distribution (1), illustrating as DC-CSMA is somehow intermediate between CSMA/p^* and U-CSMA, resulting in *i)* a *higher* successful probability in the first few slots when compared to the one provided by p^*, and *ii)* a decrease over slots stronger than the very slow one provided by CSMA/p^*, but more graceful than the abrupt sharp drop of U-CSMA. This intermediate reshaping aims at lowering down the expected successful slot number without incurring in the heavy degradation affecting (2) for the uniform distribution[9]. In this respect, the effect of such a less hesitant behavior of p in transmission attempts is a larger number of collisions and, consequently, a slight decrease in the overall successful probability (2). However, as one can see from the first plot (top) in Fig.2, this drop is rather limited and in the worst case p achieves a percentage of over 92% of the maximum possible one provided, by definition, by p^*.

TABLE III: Number of re-transmission attempts $\overline{n}_t$.

n (K=32)		15	200	n (K=64)		15	200
$\overline{n}_t$	(CSMA/p^*)	4.96	5.83	$\overline{n}_t$	(CSMA/p^*)	5.80	7.10
$\overline{n}_t$	(DC-CSMA)	8.78	12.41	$\overline{n}_t$	(DC-CSMA)	11.01	18.38

On the other hand, at the cost of the slight decrease of P_s, the reshaping of $P_s(i)$ allows to strongly reduce the expected successful slot number $\overline{\ell}_s$, making it asymptotically vanishing when compared with the overall CW size K. This claim is supported by Fig.2 (bottom), that plots the fraction of the expected successful slot number compared to the number of slots in the contention window. The graph shows that the delay-collision distribution p is able to always keep $\overline{\ell}_s$ bounded

[7]Both policies could as well as have been compared under this constraint active, without any change in the hierarchy of the relative optimality range. However, as we will show in Section VI, this would deeply affect performance in presence of hidden terminals and frame synchronization errors.

[8]More precisely, 1.7–2.8 for the values in Table III, and up to 3.3 times for a contention window size $K = 128$ and a number of contenders equal to $n = 1000$.

[9]In Fig.1(c) we plotted the probabilities for a low-dense node scenario ($n = 10$) for display purposes, in order to also show in the same plot the successful probabilities of the uniform distribution. As already indicated, for increasing values of n performance of standard nonpersistent CSMA dramatically drops, and it would not be possible to appreciate in the same plot the differences between p and p^*, that are basically independent from n and largely insensitive to its value.

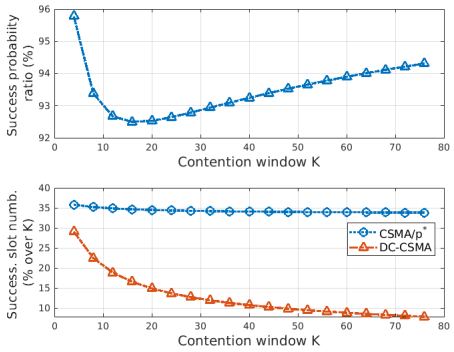


Fig. 2: (top) Probability of success of DC-CSMA as a percentage of the optimal one provided by CSMA/p^*. (bottom) Expected successful slot number of both policies as a percentage of K.

to the first few slots, *regardless* of K. Intuitively, this means that DC-CSMA is more stable and scalable with respect to the contention window size, and can adjust it to accommodate the desired successful probability P_s without incurring in increasing delay. On the contrary, Fig.2 (bottom) shows as for CSMA/p^* the expected successful slot number is always located around 33%–34% of the CW size.

IV. Latency and normalized system throughput performance

In this section our MAC strategy DC-CSMA has been compared with the standard nonpersistent CSMA where the contention probability is uniformly distributed (U-CSMA), and the nonuniform CSMA/p^*, optimal in successful probability, proposed by Tay *et al.* [4]. Performance have been evaluated considering the following metrics: probability of success P_s, expected successful slot number $\overline{\ell}_s$, access delay in terms of latency L and normalized system throughput S. We have tested the range of applicability of each policy under different number of contenders, contention window size, packet length ratio T_p/T_s. With regards to throughput, let

$$\overline{T}_s \triangleq \sum_{i=1}^{K} P_s(i)\left[iT_s + T_p\right], \qquad \overline{T}_c \triangleq \sum_{i=1}^{K} P_c(i)\left[iT_s + T_p\right]$$

be the expected lengths of successful and collided transmission, respectively. As in [19], we express the normalized system throughput as the fraction of time the channel is used to successfully transmit. Therefore, being the expected frame length $\overline{T}_f$ equal to $\overline{T}_s + \overline{T}_c + P_\varnothing(KT_s)$, the normalized saturation throughput can be obtained through a simple generalization of the model proposed in [19], and rearranged as:

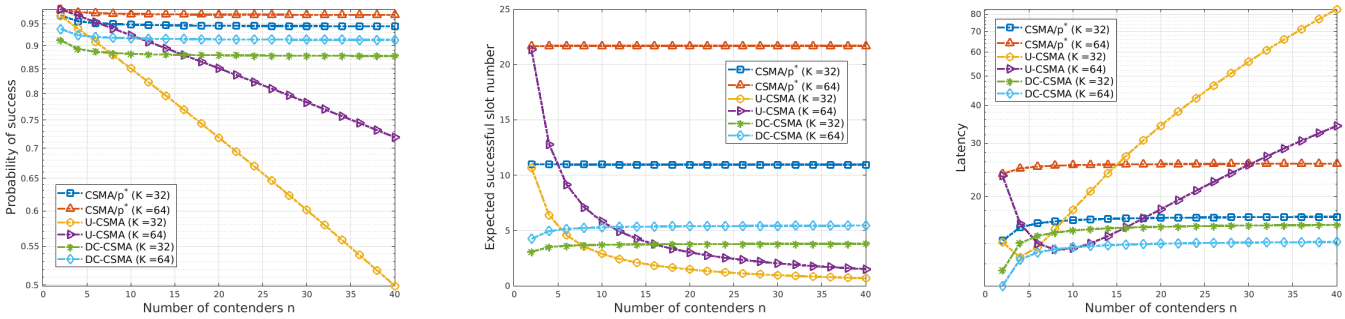
$$S = \frac{P_s T_p}{\overline{T}_f} = \frac{P_s T_p}{\left(\overline{\ell}_s + \overline{\ell}_c + P_\varnothing K\right) T_s + (1 - P_\varnothing)\, T_p} \qquad (16)$$

Probability of success and expected successful slot number for DC-CSMA, U-CSMA and CSMA/p^* are presented in Fig.3(a) and Fig.3(b). DC-CSMA and CSMA/p^* performance are almost flat as the number of contenders increases, showing as they are able to properly adapt to the actual node density condition, and the same almost flat behavior maintains out of the shown range (up to tens of thousands and more contending nodes). On the contrary, U-CSMA quickly degrades its performance when the number of contenders increases. Being designed to maximize this specific metrics, as expected CSMA/p^* reaches the best performance in terms of successful probability, while our proposal remains really close to the optimum, regardless of n. However, from Fig.3(b) some interesting results can be noticed. The increase in successful probability provided by CSMA/p^* when CW size moves from $K = 32$ to $K = 64$ is strongly paid in terms of expected successful slot number, up to 4 times (and more) higher than the more stable one provided by DC-CSMA, less sensitive to K than CSMA/p^*, making it able to exploit the CW size to increase P_s without degrading the access time to the channel.

A. Latency performance

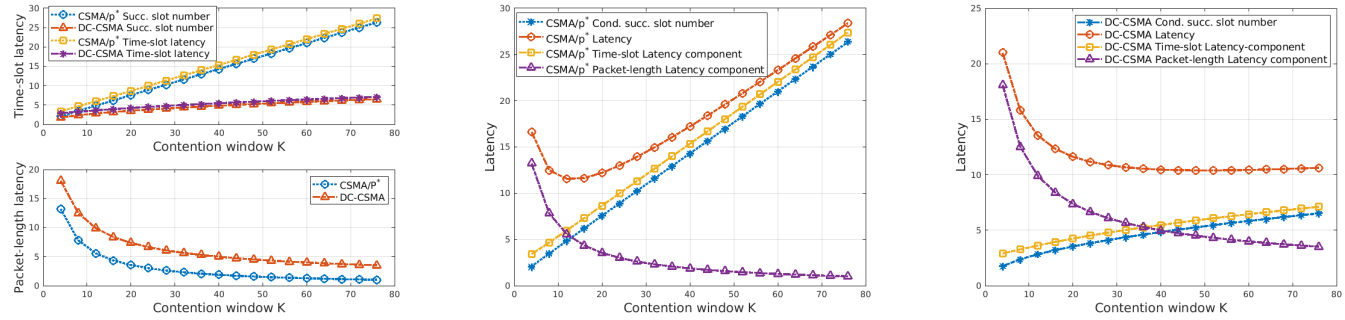
The combined effect of the above discussed tradeoff is illustrated in Fig.3(c), that shows the way latency is affected for a packet length T_p equal to 80 slot-times. Some conclusions may be drawn. First, CSMA/p^* presents an higher latency and, what is more, it degrades its performance as the contention window size K increases, showing as its strategy is affected by the contention window size. Second, DC-CSMA is able to maintain low the delay in data transmission and, moreover, its overall performance improve by increasing the CW size, as a result of the balanced tradeoff among P_s and $\overline{\ell}_s$ depicted in Fig.3(a) and Fig.3(b). Third, U-CSMA, instead, degrades significantly its performance when the number of contenders increases. This testifies a low scalability of the classical uniform distribution. However, it is possible to see as U-CSMA outperforms CSMA/p^* for a low number of nodes, indicating as CSMA/p^* fits high node-density scenario, but can be suboptimal otherwise. DC-CSMA performs well regardless of n.

In order to better understand why DC-CSMA outperforms CSMA/p^*, for $n = 500$ and $T_p = 40$ slot-times, Fig.4(a) shows the time-slot dependent latency component and packet-length dependent latency component associated to the delay, that is, the first and second addend of (5), respectively. Fig.4(a) confirms that $\overline{\ell'}_s$ is the dominant component of the time-slot latency component for both DC-CSMA and CSMA/p^*, corroborating our choice (8). Note that the main component affecting the overall CSMA/p^* delay is the *time-slot latency*. In fact, in order to increase the frame successful probability CSMA/p^* applies a very conservative strategy incurring in many idle slots that degrade performance. On the contrary, DC-CSMA approximately equalizes the two latency components for a large range of CW sizes (as an example, for $K = 64$ the CSMA/p^* time-slot latency component is around 4 times higher than the packet-length one, the ratio becoming more and more unbalanced for increasing K, while for DC-CSMA it stabilizes at levels around 1 for $K \geq 30$). The impact on the overall latency is illustrated in Fig.4(b) and Fig.4(c). All delay components of CSMA/p^* are presented in Fig.4(b): time slot-latency and conditional success slot number severely affect the performance for increasing K. In fact, CSMA/p^* is



(a) Probability of success (b) Expected successful slot number (c) Latency

Fig. 3: (a) Probability of success and (b) Expected successful slot number vs increasing number of contenders under different contention window size K. (c) Corresponding Latency evaluation for $T_p = 80$ slot-times.



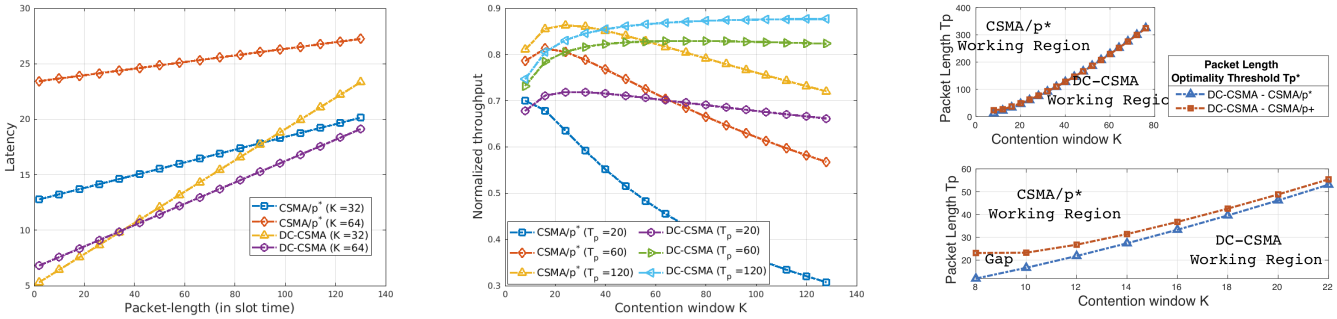
(a) Latency components (b) CSMA/p^* Latency (c) DC-CSMA Latency

Fig. 4: (a) Time-slot dependent and packet-length dependent latency for DC-CSMA and CSMA/p^*, under different contention window size K, with $n = 500$ and $T_p = 40$ slot-times. (b)-(c) Corresponding CSMA/p^* and DC-CSMA total Latency.

too reliant on the proper dimensioning of window size K, not explicitly taken into account by its optimization procedure and, consequently, resulting in performance too much dependent on its setting. In many contexts where different CW sizes could be used such as IEEE 802.11, IEEE 802.15.4 or other technologies that try to reduce the collision probability changing in real time the contention window [16], this can create a few problems if some importance needs to be given also to the channel access delay among all contenders. On the other hand, DC-CSMA, via its flat stable behavior (Fig.4(c)), is able to better preserve its performance over different CW sizes. Hence, the shaping of the contention probability can be more independent of its value, extending its possible application in situations where a programmable MAC could scale the CW without complicating too much the MAC protocols. In other terms, DC-CSMA maintains the advantage of CSMA/p^*, that is, the simplicity to apply in many context such as WSN, IoT or RFID, with the further strength to also exploit the dimension of the delay in a more stable way.

B. Normalized system throughput and packet-length/time-slot ratio dependence

The extension of the operative range of DC-CSMA strategy has been further investigated in Fig.5, where performance are compared under different packet lengths T_p, measured in multiple of slot-times, for $n = 200$ contenders. From Fig.5(a), we see as for a fixed CW size DC-CSMA has a quite extended range of optimality. It is interesting to observe as, for a very high packet length, CSMA/p^* recovers and can perform better than DC-CSMA. This is due to the second component of the latency, as shown in the previous graphs, that can become more significant in comparison with the time-slot one. This confirms the asymptotic analysis of Section II-A about the packet-length/slot-time ratio; with regard to this, interestingly, we can provide now a quantitative indication of what *small* or *high* packet-length/slot-time ratio means in order to choose the best performing probability distribution for a CSMA scheme. While CSMA/p^* asymptotically outperforms DC-CSMA (i.e., for $T_p/T_s \to \infty$), it is possible to see how for a wide operative range of packet lengths the performance of DC-CSMA can be better. We see as at least up to $T_p \approx 120$ slot-times, DC-CSMA should be preferred. Moreover, if the CW size of DC-CSMA is high or can be scaled up, the DC-CSMA is always more performing. Concerning the normalized throughput shown in Fig.5(b), the same analysis applies and it is possible to see the (actually, quite limited) range of values of K where CSMA/p^* outperforms DC-CSMA on the basis of the packet length. However, note as the approximately flat behavior of the DC-



(a) DC-CSMA and CSMA/p^* latency (b) DC-CSMA and CSMA/p^* Normalized Throughput (c) Packet length optimality threshold

Fig. 5: (a) Latency vs packet length. (b) Normalized throughput vs contention window size K ($n = 200$ contenders). (c) Packet length optimality threshold T_p^*.

CSMA throughput makes it quite insensitive to possible real-time changes of packet lengths or packet length differentiation among nodes, depending on the application. As an example, we can see as with the setting $K = 64$ DC-CSMA throughput is strongly robust for any T_p in the considered range $[20, 120]$. More specifically, the throughput obviously depends on the packet length T_p, but *does not* degrade with respect to the maximum one achievable for that T_p through a customized tuning, while the CSMA/p^* setting of CW should be adjusted to any change of T_p (which is hard in practice) if it does not want to lose up to $8\% - 10\%$ of its own maximum throughput. We anticipate that this issue will have a more deep negative impact on performance of CSMA/p^* (and its heuristic Sift) under non-ideal network conditions (see Section VI). This testifies that under very massive contention such as in IoT, RFID or 5G contexts, where it is possible to use large contention windows, our proposal could be a good compromise.

Before concluding this section, Fig.5(c) summarizes which protocol works better depending on the situation, making possible a practical dimensioning concerning which strategy to use. In fact, Fig.5(c) shows the packet length threshold T_p^* that separates the DC-CSMA and CSMA/p^* optimal working regions as a function of K. For any given K, for packet lengths T_p sufficiently high the wasted time (6) due to the expected number of collisions dominates the latency L and CSMA/p^* is to be preferred. On the contrary, for packet lengths under the threshold T_p^*, collisions are more tolerated because of the smaller size of the transmitted packets and the waiting time (7) affects performance more, allowing DC-CSMA to achieves better performance. Note from Fig.5(c) that the threshold T_p^* splitting the two optimal working regions fast increases with the CW size. In fact, it is a *super-linear* increase (e.g., for $K = 32$ it is $T_p^*/K = 2.85$, whereas for $K = 64$ it is $T_p^*/K = 3.92$, indicating that in these operative conditions DC-CSMA should be preferred to CSMA/p^* whenever the packet lengths T_p are lower than 91 or 251, respectively, allowing the profitable working region of DC-CSMA to quickly saturate the space for increasing values of CW size. As a final example, for $K = 128$, just beyond the plotted range, it reaches the value $T_p^* \approx 700$. Due to the above considerations, CSMA/p^* (and, as we show in the following, its heuristic Sift) is forced to keep K quite *static* and *low*, indeed limited to the range $[16, 32]$. In Fig.5(c) we also plotted the optimality threshold related to CSMA/p^+. As it could be expected, under ideal conditions CSMA/p^+ suffers a penalty gap compared to its improved version CSMA/p^* only for relatively low values of K (Fig.5(c) (bottom)), while the graphs rapidly overlap for increasing size (Fig.5(c) (top)). In Section VI we will see that this will be no longer the case in presence of hidden terminals or frame synchronization errors.

Summarizing the results achieved so far, the above analysis showed that: *i)* CSMA/p^* tends to work better with low CW sizes, whereas DC-CSMA with higher ones; *ii)* in order to guarantee a probability of success of the order of 0.95 (or above) CSMA/p^* and DC-CSMA should use contention windows at least of the order of 32 and 64, respectively. In any case, for $K \geq 64$ they can be considered practically equivalent in terms of probability of success and wasted time due to collisions, regardless of the number n of contenders. On the other hand, with regard to the average waiting time needed to start a successful transmission, it is evident the performance gap to the benefit of DC-CSMA (basically, due to the result established in *Proposition 2*). This is the reason why DC-CSMA generally outperforms CSMA/p^*.

V. CW SIZE ADAPTATION AND COMPLEXITY

In this section we prove a theoretical result about CW size optimality, and discuss some practical aspects related to CW size selection and complexity.

A. CW size adaptation

In Fig.5(c) we compared DC-CSMA and CSMA/p^* in terms of packet length optimality intervals *for each* assigned CW size K. An interesting issue is to allow each policy to search for its *best* CW size K^{opt} for each specific application scenario (in terms of traffic, topology, channel), and *then* compare. We explicitly point out that for the general distribution case we are investigating this is a nontrivial problem even for a clique saturated network and, to the best of our knowledge, generally

unsolved. We expect this problem to be generally intractable due to the integer nonlinear dependence on K, and remains out of the scope of the paper. Nevertheless, this open issue allows to highlight some distinctive features of the family of strategies we consider. About the results shown so far, it could remain unclear if the advantage of DC-CSMA might disappear with different settings of network parameters. This is not the case. We can establish the following result.

Proposition 3 (*CW optimality*)*:* For any packet length size T_p and related CSMA/p^* (whose CW size K^* is possibly tuned to achieve the *best* performance among all CSMA/p^* for that specific situation), there exists an appropriate setting of DC-CSMA that guarantees the *same* successful probability of CSMA/p^* (i.e, the same wasting time due to collisions), with a *lower* expected successful slot number (i.e., a lower waiting time due to idle slots).

Proof: Let K^* be the CW size used by CSMA/p^*, possibly optimized. Let K be the CW size such that DC-CSMA obtains the same frame successful probability as CSMA/p^*, i.e.:

$$P_s(p(K)) = P_s(p^*(K^*)).$$

By leveraging *Proposition 1*, such a K clearly exists. Being CSMA/p^* the optimum in probability for any fixed CW, it must be $K > K^*$. Since each distribution feasible for K^* can trivially be used for a contention window size $K > K^*$ by setting to 0 the attempt probabilities in the interval $[K^* + 1, K]$, p^* can be applied to K as well. Therefore we can write: $U_K(p^*(K^*)) < U_K(p(K))$, where the inequality is strict since the distribution p in (13)–(15) optimal for (8) is unique by construction. Finally, from this last inequality and definition (8) we directly obtain:

$$\overline{\ell}_s(p(K)) < \overline{\ell}_s(p^*(K^*)).$$

□

We recall that the optimized setting of the DC-CSMA CW size able to achieve the highest performance, whose existence is ensured by *Proposition 3*, is generally unknown. For both policies, a precise CW adaptation would add a second-level optimization structure that would *i)* increase the computational complexity and *ii)* introduce transient intervals with a consequent increase of convergence-time and latency.

Nevertheless, *Proposition 3* makes clear that Fig.5(c) should be used to select between DC-CSMA and CSMA/p^* *only when* K is fixed, like in practical low cost implementations as detailed in Section V-B. On the other hand, whenever it is possible to choose the optimized K or converge to it, DC-CSMA always obtains better results. In this regard, dynamic adaptation of the CW size like, for example, modifications of the approach proposed in [16] could be investigated. However, as seen in Section IV and more deeply investigated in Section VI, due to the robustness of DC-CSMA with respect to the used CW size in a wide range of conditions, the dynamic search of such an optimal K value can thus be avoided. Fixed choices of K result in almost optimal performance, without any additional cost neither in terms of complexity nor in terms of convergence time. In fact, in addition to the advantages investigated above, one of the strengths of DC-CSMA is its application/traffic *independence*, in contrast with traditional uniform approaches that totally rely on the CW size adaptation and convergence to a steady-state solution.

B. Complexity analysis and practical implementation

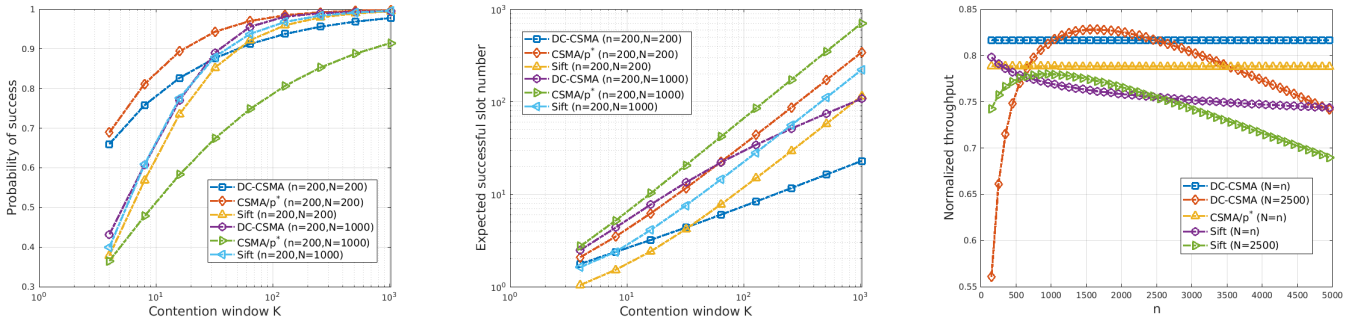
About complexity, the only required computational effort is the computation of the optimal DC-CSMA probability distribution p through the backward and forward recursion (13)-(15), quite trivial and of the order of $O(K)$. Therefore, DC-CSMA and CSMA/p^* retain the same computational complexity. This is one of the reasons that led us to choose (8) as utility function, together with its independence from traffic patterns. Specifically, the low complexity is due to the closed form expressions (13)-(14) connecting the probability distribution to the optimal rewards r_i of the utility function (8). This allows to avoid having to resort to fully numerical optimization and more complex dynamic programming approaches that would indeed increase complexity and impair a practical applicability (also introducing convergence time issues). In fact, by leveraging (9), the utility function (8) can control P_s and $\overline{\ell}'s$ *without* directly addressing the more cumbersome traffic-dependent latency and throughput expressions (5), (16). By taking a further look at (13)-(14) we can also recognize that it is not necessary to re-calculate them every time K changes (e.g, increases), the new distribution being a shifted version of the previous one, obtained by adding the conditional probabilities of the initial slots in the updated CW, while the others shift. Moreover, depending on the application, or in case of strongly limited capabilities of devices, a further possibility could be to avoid any run-time calculation and just communicate to the device (or store in it) the sample distribution for a K_{max} big enough to cover the need of the user/application, while devices with some computational capabilities could calculate it. In this regard, in Section VI-A we will verify the strong robustness of DC-CSMA against node estimation errors. This allows devices to calculate the optimal distribution one-shot or update it only on a periodic basis.

VI. Node estimation errors, hidden terminals and frame synchronization errors

In this section we further compare performance in presence of some non-idealities that impair real systems with more general network topologies, like node estimation errors, frame synchronization errors and hidden terminals. We anticipate that the performance gap will increase, precisely because of the greater robustness of DC-CSMA. We also compared performance with Sift [5], [7], [18], [19], a very robust policy under node estimation errors.

A. Robustness against node estimation errors

In order to evaluate the robustness of the proposed policy to the estimation error on number of active nodes, we compared performance when the number n of contenders is different from the estimated value N used by distributions. In Fig.6(a) and Fig.6(b) we show the probability of success



(a) Probability of success (b) Expected successful slot number (c) Normalized throughput

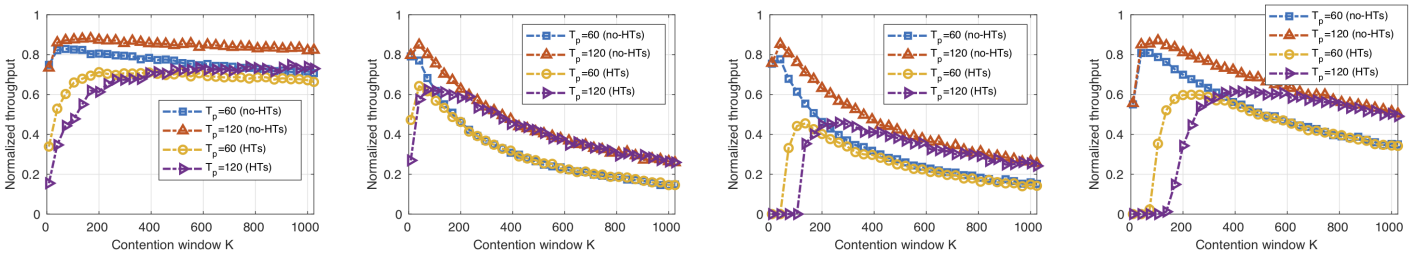
Fig. 6: (a)-(b) Probability of success and Expected successful slot number vs contention window size K, under node estimation error. (c) Normalized throughput vs number of nodes n under node estimation error for $T_p = 60$ slot-times and $K = 32$.

and the expected successful slot number when $n = 200$ while N is a five times higher overestimate of n, i.e. it is a quite rough estimate possibly set to the maximum number of active contenders in the systems. We compared DC-CSMA with CSMA/p^* and Sift, a robust distribution under node estimation errors whose shaping parameter α is set to [5], [18] $\alpha = N^{-1/(K-1)}$. As a benchmark, for the three policies we also show the perfect node estimation case when N coincides with n. Some interesting conclusions can be drawn by the examination of these pictures. First, the distribution who maximizes the successful probability, CSMA/p^*, is also the most sensitive to the node estimation errors; its successful probability drops sharply under an inaccurate estimate. On the other hand, DC-CSMA closely follows CSMA/p^* under perfect node estimation, and, furthermore, it is quite robust with respect to node overestimation. Indeed, despite the high estimation error, its success probability does not drop under the condition that it uses a CW size K greater than approximately 30. The same is true for the Sift distribution, that confirms itself as strongly stable under node overestimation, from a successful probability point of view. This might lead to the wrong idea to consider the Sift independent from n, provided that N is chosen as large as possible. However, from Fig.6(b) it can be seen as the Sift expected successful slot number is sensitive both to N and K. More in detail, Sift $\overline{\ell}_s$ retains the same linear dependence on K than CSMA/p^* (they have the same slope in the log-log scale of Fig.6(b)), while, on the contrary, DC-CSMA is strongly scalable with respect to the CW size, even under strongly inaccurate node estimation (both DC-CSMA graphs exhibit the same slope in Fig.6(b), corresponding to a sublinear increase). As a consequence, DC-CSMA can properly balance the opposite effects of successful probability and expected successful slot number on overall performance. This is confirmed by Fig.6(c), where we compared the normalized throughput for $T_p = 60$ slot-times and $K = 32$, i.e., we chose a CW size where CSMA/p^* and Sift still well behave. Even in this setting we can see as DC-CSMA outperforms both policies. About the node estimation error, for both DC-CSMA and Sift it can be recognized as the throughput penalty is more severe when N underestimates n (for example, when $N = 2500$ and $n = 5000$ both DC-CSMA and Sift loose about $7\% - 9\%$ with respect to their own perfect node estimation benchmark throughput, respectively), whereas both are strongly robust under overestimation. Only when the number of nodes is very low compared to the estimated one (approximately, from 7 to 10 times lower than N), DC-CSMA performance starts to drop and Sift is to be preferred.

B. *Hidden terminals and frame synchronization errors.*

The most severe non-ideality affecting performance of nonuniform backoff strategies with CW regeneration is the loss of synchronization among frames. A major cause of such a loss is the presence of hidden terminals (HTs). In fact, devices naturally abort their backoff as soon as they sense a transmission on the channel, and trigger the start of a new contention period over K slots when they perceive the channel continuously idle for a distributed inter-frame space (DIFS). The presence of HTs destroys this natural coordination and synchronization among devices that relies on sensing. When this happens, performance abruptly collapses and reduces to the one provided by standard uniform distributions (see [19] for an empirical analysis of the phenomenon, through simulations and testbeds). The reason is the loss of hierarchy among slots and the equalization of the access probabilities. This adds up to the performance degradation that hidden terminals would naturally produce anyway. In Fig.7 we show the throughput performance degradation (for Sift, CSMA/p^+, CSMA/p^* and DC-CSMA) under harsh conditions when $n = 200$ nodes share a common channel and, in addition to saturated traffic and massive scenario, for each node three other fully saturated nodes inside the network act as HTs sharing the same receiver, with packet lengths T_p equal to 60 and 120 slots, respectively. We tested up to $K = 1024$, the maximum CW size (CW_{max}) used in the IEEE 802.11 standard; this wider range also allows to detect the turning point in DC-CSMA's performance, that due to its strong robustness requires a higher CW range to be noticed.

Table IV summarizes the main results, indicating: *i)* the percent Optimality Loss Rate (%OLR) each strategy suffers in presence of HTs compared to its ideal case, obtained by comparing the peak performance each policy achieves with HTs compared to the peak performance the same policy achieves



(a) DC-CSMA (b) CSMA/p^* (c) CSMA/p^+ (d) Sift

Fig. 7: Normalized throughput performance vs contention window size K, for $T_p = 60$ and $T_p = 120$ slot-times, with hidden terminals (HTs) and without hidden terminals (no-HTs).

without HTs; *ii)* its 10%-Robustness Range (10%RR), i.e. the extension of the average CW size range (up to 1024, in the tested interval) in which performance does not fall below 10% of the policy's optimal one achieved with the best possible choice of K; *iii)* the Static Stability Range (SSR), namely, the extension of the intersection between the two 10%RRs concerning the HTs and no-HTs cases. SSR provides a measure of the possibility for a policy to conservatively dimension K regardless of a prior knowledge of the environment and HTs presence (hard to achieve in practice and time-dependent), so to optimally work in both cases without sacrificing performance. The wider is SSR, the less sensitive is the policy to a choice of K not tuned for the specific situation.

TABLE IV: Performance metrics in presence of HTs. The first block refers to $T_p = 60$ slot-times, the second one to $T_p = 120$ slot-times.

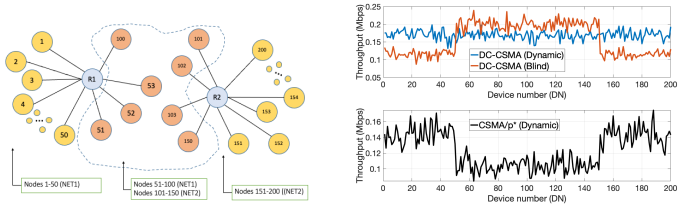
	DC-CSMA	CSMA/p^*	CSMA/p^+	Sift
%OLR	14%	23%	43%	24%
10%RR (no-HTs)	980	90	90	109
10%RR (HTs)	918	70	170	237
SSR	896	27	0	0
%OLR	18%	28%	44%	25%
10%RR (no-HTs)	993	180	180	228
10%RR (HTs)	823	92	250	472
SSR	721	13	0	11

We spend some words to interpret the obtained results and draw some conclusions. First, what discussed above with regard to hidden terminals and frame synchronization explains the performance drop of CSMA/p^+ and Sift in Fig.7, in agreement with [19]. Second, the 40% peak performance improvement of CSMA/p^* over CSMA/p^+ is due to its weaker constraint (4) as compared to the equality one of CSMA/p^+ and Sift. Specifically, although this weaker constraint leads to fairly negligible differences under ideal conditions (see Fig.5(c)), it is crucial in presence of non-idealities. In fact, as seen in Table III and related discussion about the expected number $\overline{n}_t$ of transmission attempts, the optimized exploitation of (4) acts as a natural distributed admission control scheme that strongly limits the flooding phenomena that can affect contention when synchronization or sensing fail.

Finally, regardless of the robustness results, we explicitly point out that even comparing DC-CSMA to the best possible CSMA/p^*, we obtain an improvement of 15% in presence of HTs, that increases up to 56% compared to CSMA/p^+. Compared to Section IV, this gap increases because of the greater degree of freedom of DC-CSMA in choosing the CW size. More in detail, while under perfect sensing both strategies can normally guarantee high probability of success for relatively low values of K (see Fig.3(a)), this is not true when ideal conditions no longer apply. CW size adaptation is also a natural instrument to spread contention in order to contrast the HTs problem, as traditional BEB schemes indicate. Due to the above analysis CSMA/p^* and Sift cannot explore this possibility.

As a final test, we consider the two star subnetworks NET1 and NET2 in Fig.8(a), each consisting of 100 nodes, each node labeled with its device number DN, with DN $\in \{1, \ldots, 100\}$ for NET1 and DN $\in \{101, \ldots, 200\}$ for NET2. Each yellow node (DN $\in \{1, \ldots, 50\}$ for NET1 and DN $\in \{151, \ldots, 200\}$ for NET2) is out of the interference range of the other subnetwork, just belonging to the clique of 100 nodes related its own subnetwork. On the other hand, the red nodes (DN $\in \{51, \ldots, 150\}$) are within the sensing and interference range of each other, and also interfere with the receiver of the other subnetwork. As a consequence, each red node *i)* belongs to one of two cliques of 150 nodes (namely, the 100 nodes within the same subnetwork plus the 50 red nodes of the adjacent one), therefore experiencing a more crowed channel, and *ii)* acts as HT for the far yellow nodes of the other subnetwork. Thus, the per node access situation of the entire network is strongly unbalanced. For an IoT application whose WiFi protocol parameters are reported in [25, Table I], Fig.8(b)(top) plots the throughput performance (in Mbps) achieved by each node, averaged over 9 (s) of network operation. The protocol parameters correspond to a payload PL of 8 (Kbit), a total packet duration equal to $T_p = 205,7$ (μs) (i.e., 23 slot-times in multiples of the slot duration $T_s = 9$ (μs)) and a CW size $K = 64$. In the blue graph (Dynamic), nodes are topology-aware and can adapt their optimal distribution by taking account of the exact per-node clique dimension. The graph confirms that DC-CSMA is able to perfectly equalize throughput among devices, even under the strongly unbalanced depicted scenario. Conversely, in the red graph (Blind) nodes simply use the static distribution dimensioned for $n = 100$ and $K = 64$, and are totally *unaware* of the presence of the other subnetwork. In such a blind situation throughput

differently distributes among nodes, and HTs impair yellow nodes' performance. Nevertheless, we stress that the total network throughput just falls from 33.9 (Mbps) (Dynamic) to 31.6 (Mbps) (Blind), with only a 7% decrease, confirming DC-CSMA's throughput robustness even under strong node estimation errors and HTs. Finally, Fig.8(b)(bottom) shows the per-node throughput performance of the topology aware optimized CSMA/p^*. It can be recognized that the traffic cannot be perfectly balanced even when nodes are aware of clique-sizes. This is due to an excessive conservative behavior of CSMA/p^* that over-limits the nodes in the most crowded cliques (red nodes). Besides the unfairness, we measured a total 27% network throughput fall compared to DC-CSMA.



(a) Network topology (b) Throughput

Fig. 8: Throughput performance after 9 (s), for $n = 200$ nodes (100 each star subnetwork), $K = 64$.

VII. Discussion and Conclusions

Being a basic contention-based channel access, DC-CSMA is perfectly suitable to be integrated with other techniques to provide service-level differentiation (like [19], or in combination with other approaches described in it) or to pursue specific targets, as done for CSMA/p^* and Sift for example in a vehicular context [7]. A further topic of investigation is to use OFDMA on top of DC-CSMA, and optimize their integration. We believe that the flexibility of DC-CSMA makes it a good candidate in these contexts as well. This is left as future work.

In this work, a novel nonpersistent CSMA called DC-CSMA able to balance the frame successful probability and the expected success slot number has been proposed. It is a good compromise between the opposed qualities of U-CSMA and CSMA/p^*: it slightly lowers the frame successful probability compared to the maximal one provided by CSMA/p^*, but offers in return a very low expected successful slot number and latency independently of the CW size. Moreover, DC-CSMA outperforms CSMA/p^* and Sift for a broad range of packet lengths, while retaining the robustness of Sift to node estimation errors. Furthermore, the performance gap increases in presence of hidden terminals. These features make it suitable for high-dense delay-sensitive technologies such as WSN, RFID, IoT devices, that can benefit from access techniques robust with respect to the access parameters that do not require time-wasting tuning procedures.

Appendix

Without loss of generality we consider in the proof the renormalized version $V = (K+1)U_K$ of the utility function (8). Let

$$P_{\varnothing}(i) \triangleq \prod_{l=1}^{i} (1-p'_l)^n , \tag{17}$$

for $i = 1, \ldots, K$ be the probability that CW has remained idle until the end of slot i, expressed as a function of the conditional attempt probabilities p'_i. We assume, by definition, $P_{\varnothing}(0) = 1$ (i.e., the probability that the frame begins already occupied by an ongoing transmission is zero by definition). It is easy to see that the utility function V can be re-arranged as $V = \sum_{i=1}^{K} V_i$, where the i-th addend

$$V_i = n\,(K-i+1)\,P_{\varnothing}(i-1)p'_i\,(1-p'_i)^{n-1} , \tag{18}$$

is the value of the utility function related to a success at slot i. Let us now denote by $V^{(i)} \triangleq \sum_{l=i}^{K} V_l$ the *forward* utility function related to slot i, that is, the sum of the last $K-i+1$ terms of the utility function. Since maximum occur at an interior point, as it can be easily verified, and the first $(i-1)$ addends V_l (18) for $l < i$ do not depend on p'_i, it must necessarily be: $\frac{\partial}{\partial p'_i} V \equiv \frac{\partial}{\partial p'_i} V^{(i)} = 0$ for $i = 1, \ldots, K$. In order to derive an easy to handle expression, from (18) the forward utility function $V^{(i)}$ related to slot i can be re-arranged as

$$V^{(i)} = P_{\varnothing}(i-1)r_i \tag{19}$$

where

$$r_i = \sum_{l=i}^{K} \left(n\,(K-l+1)\,p'_l\,(1-p'_l)^{n-1} \cdot \prod_{m=i}^{l-1} (1-p'_m)^n \right)$$

is the value of the utility function *conditioned* on idle in the first $(i-1)$ slots, i.e., what we called the reward for refraining the transmission until slot i. As expected, r_i only depends on the forward conditional probabilities p'_l, for $l \geq i$.

Hence, being from (17) $P_{\varnothing}(i) = P_{\varnothing}(i-1)\,(1-p'_i)^n$, from (18) and (19) we can derive the following recursive expression:

$$V^{(i)} = V_i + V^{(i+1)} = V_i + P_{\varnothing}(i)r_{i+1} =$$
$$= P_{\varnothing}\,(i-1)(1-p'_i)^{n-1} \cdot \Big[(n(K-i+1) - r_{i+1})\,p'_i + r_{i+1} \Big] \tag{20}$$

for $i = 1, \ldots, K$, with $r_{K+1} = 0$, showing as the forward utility function related to slot i depends on *past* actions via the idle probability $P_{\varnothing}(i-1)$, *future* actions via the expected future reward r_{i+1}, and the *present* action via the probability p'_i to attempt transmission at the current slot. Since $P_{\varnothing}(i-1)$ and r_{i+1} do not depend on p'_i, the derivative of (20) with respect to p'_i is

$$\frac{\partial V^{(i)}}{\partial p'_i} = \Big(n(K-i+1) - r_{i+1}\Big) P_{\varnothing}(i-1)(1-p'_i)^{n-1}$$
$$- \Big[\Big(n(K-i+1) - r_{i+1}\Big)p'_i + r_{i+1} \Big] P_{\varnothing}(i-1)(n-1)(1-p'_i)^{n-2}$$
$$= nP_{\varnothing}(i-1)(1-p'_i)^{n-2}$$
$$\times \Big[\Big((K-i+1) - r_{i+1}\Big) - \Big(n(K-i+1) - r_{i+1}\Big)p'_i \Big], \tag{21}$$

that set to 0 allows to derive (13). By examining the sign of (21) it can be verified that (13) is a maximum. Finally, by introducing (13) into (20) we get (14), while the optimal

nonpersistent distribution (15) can be obtained via the standard relationship between conditional and non-conditional probabilities $p_i = p'_i \left(1 - \sum_{l=1}^{i-1} p_l\right)$.

REFERENCES

[1] A. Kumar, M. Zhao, K.-J. Wong, Y. L. Guan, and P. H. J. Chong, "A comprehensive study of iot and wsn mac protocols: Research issues, challenges and opportunities," *IEEE Access*, vol. 6, pp. 76 228–76 262, 2018.

[2] B. Ismaiel, M. Abolhasan, D. Smith, W. Ni, and D. Franklin, "Scalable mac protocol for d2d communication for future 5g networks," in *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2017, pp. 542–547.

[3] M. A. Rahman, A. T. Asyhari, I. F. Kurniawan, M. J. Ali, M. M. Rahman, and M. Karim, "A scalable hybrid mac strategy for traffic-differentiated iot-enabled intra-vehicular networks," *Computer Communications*, 2020.

[4] Y. Tay, K. Jamieson, and H. Balakrishnan, "Collision-minimizing csma and its applications to wireless sensor networks," *IEEE Journal on selected areas in Communications*, vol. 22, no. 6, pp. 1048–1057, 2004.

[5] M. V. Bueno-Delgado, R. Ferrero, F. Gandino, P. Pavon-Marino, and M. Rebaudengo, "A geometric distribution reader anti-collision protocol for rfid dense reader environments," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 296–306, 2013.

[6] P. K. Sahoo and J.-P. Sheu, "Design and analysis of collision free mac for wireless sensor networks with or without data retransmission," *Journal of Network and Computer Applications*, vol. 80, pp. 10–21, 2017.

[7] C. García-Costa, E. Egea-López, and J. García-Haro, "Evaluation of mac contention techniques for efficient geo-routing in vehicular networks," *Ad Hoc Networks*, vol. 37, pp. 44–62, 2016.

[8] E. Khorov, A. Krotov, A. Lyakhov, R. Yusupov, M. Condoluci, M. Dohler, and I. Akyildiz, "Enabling the internet of things with wi-fi halow—performance evaluation of the restricted access window," *IEEE Access*, vol. 7, pp. 127 402–127 415, 2019.

[9] Y. Cheng, D. Yang, H. Zhou, and H. Wang, "Adopting ieee 802.11 mac for industrial delay-sensitive wireless control and monitoring applications: A survey," *Computer Networks*, vol. 157, pp. 41–67, 2019.

[10] F. Shu and T. Sakurai, "A new analytical model for the ieee 802.15. 4 csma-ca protocol," *Computer networks*, vol. 55, no. 11, pp. 2576–2591, 2011.

[11] S. Moulik, S. Misra, and C. Chakraborty, "Performance evaluation and delay-power trade-off analysis of zigbee protocol," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 404–416, 2018.

[12] M. Michalopoulou and P. Mähönen, "A mean field analysis of csma/ca throughput," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2093–2104, 2016.

[13] S. Rashwand and J. MišIć, "Effects of access phases lengths on performance of ieee 802.15. 6 csma/ca," *Computer Networks*, vol. 56, no. 12, pp. 2832–2846, 2012.

[14] Y.-W. Kuo and J.-H. Huang, "A csma-based mac protocol for wlans with automatic synchronization capability to provide hard quality of service guarantees," *Computer Networks*, vol. 127, pp. 31–42, 2017.

[15] J. Choi, S. Byeon, S. Choi, and K. B. Lee, "Activity probability-based performance analysis and contention control for ieee 802.11 wlans," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 1802–1814, 2016.

[16] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, 2005, pp. 121–132.

[17] I. Demirkol and C. Ersoy, "Energy and delay optimized contention for wireless sensor networks," *Computer networks*, vol. 53, no. 12, pp. 2106–2119, 2009.

[18] K. Jamieson, H. Balakrishnan, and Y. C. Tay, "Sift: A mac protocol for event-driven wireless sensor networks," *Wireless Sensor Networks*, vol. 3868, pp. 260–275, 2006.

[19] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-mac: a hybrid mac for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 511–524, 2008.

[20] M. V. B. Delgado and P. P. Mariño, "Using non-uniform probability distribution p* to improve identification performance in dense rfid reader environments," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, 2013, pp. 468–471.

[21] C. Joo and N. B. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1481–1493, 2009.

[22] "Ieee std 802.15.4tm-2015," pp. 1–708, 2015.

[23] "Bluetooth specification version 5.0 — vol 0, part a," pp. 1–2822, Dec. 2016.

[24] "Part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-TM-2016*, pp. 1–3504, 2016.

[25] J. Ho and M. Jo, "Offloading wireless energy harvesting for iot devices on unlicensed bands," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3663–3675, 2019.

Nicola Cordeschi received the five-year Laurea degree (summa cum laude) in communication engineering and the PhD degree in information and communication engineering from the Sapienza University of Rome in 2004 and 2008, respectively. He was Fellow Researcher with the DIET Dept., Sapienza University of Rome for ten years, and Assistant Professor from 2009 to 2016. Then he worked with DIMES Department, University of Calabria. Since 2021, he is with 6G Innovation Centre, University of Surrey, UK. He is a recipient of two Best Paper Awards and has authored or coauthored over 80 papers, many of which published in premier network journals and conferences including IEEE TMC, TVT, TCOM, IEEE/ACM TON, and more. His research interests include adaptive wireless communications, cognitive radio access, medium access control, multi-antenna systems, energy-efficiency, resource management in vehicular communications, grid/cloud computing, distributed computing, cross-layer optimization, mathematical optimization, game theory.

Floriano De Rango received the graduated degree in computer science in October 2000, and the Ph.D. degree in electronics and telecommunications engineering in January 2005, both from the University of Calabria, Cosenza, Italy. From January 2000 to October 2000, he was with the Telecom Research LAB C.S.E.L.T. in Turin with a Scholarship. From March 2004 to November 2004, he was a Visiting Researcher with the University of California at Los Angeles, Los Angeles, CA, USA. He is currently an Associate Professor with DIMES Department, University of Calabria. He has authored or coauthored more than 220 papers among international journals and conferences. His interests include Internet of Things, IP QoS architectures, adaptive wireless networks, mobile ad hoc networks, and pervasive computing, security, vehicular ad hoc networks, and green networks. In 2009, he founded a Spin-off company (Spintel s.r.l.) in the smart control tools design field for energy efficiency for smart building and cities and Internet of Things (IoT) that won some awards in international competitions such as the Intel Business Challenge Award. His company has been ranked in the first 20 Innovative Start-ups in Europe (IBCE 2013). He was a recipient of the Young Researcher Award in 2007.

Mauro Tropea received the master's degree in April 2003 and the Ph.D. degree in January 2009 in computer science engineering with the University of Calabria, Cosenza, Italy. From May 2008 to October 2008, he was a visiting researcher with the Telecommunication Department in ESA ESTEC Noordwijk, The Netherlands. Since 2003, he has been with the group of telecommunications with the DIMES Department. Since September 2018, he was a Postdoctral with the DIMES Department. His research interests include satellite communications, QoS architectures, bio-inspired algorithms, VANETs, FANETs, hierarchical networks, and multicasting.