A dynamic framed ALOHA scheme for batch resolution in practical CSMA-based wireless networks

Andrea Zanella
Dep. of Information Engineering,
University of Padova (Italy)
email: zanella@dei.unipd.it

***Abstract*—The batch resolution problem consists in arbitrating the channel access of a group of nodes in a wireless network in order to collect a *single* packet from each node in the shortest time. Most of existing solutions are based on the *immediate feedback* assumption and typically neglect or underestimate the actual time cost of feedback that can instead be significant in common wireless standards. In this paper, we propose and analyze ABRADE, which is a dynamic framed ALOHA scheme for conflict resolution in practical CSMA-based wireless networks. The core of ABRADE, in fact, is the dynamic adaption of the framed ALOHA parameters to the cardinality of the residual batch, in order to strike a balance between the control message overhead and the fraction of successful transmissions per frame. The parameters optimization is based on a dynamic programming argument that takes into account the time occupancy of successful, collided and idle slots, as well as the time cost of control messages. Compared against classical batch resolution algorithms in practical scenarios, the proposed solution yields up to 10% of throughput gain.**

***Index Terms*—ad hoc networks; sensor networks; node discovery; identification; batch resolution; conflict resolution; immediate feedback; deferred feedback; framed aloha, rfid**

I. Introduction

Emerging communication scenarios, such as Internet of things, opportunistic networking, dust networks, generally involve densely populated clouds of wireless terminals. In this context, it is common for a node to inquiry the surrounding nodes for different purposes, such as counting the number of neighbors, collecting data, updating connectivity information and so on. The probe message may solicit reply from a potentially large set of nodes that compete to transmit their message over the shared wireless channel. When multiple nodes transmit simultaneously, their radio signals will destructively interfere with one another at the receiver, resulting in a so-called packets collision (or conflict).

The *batch resolution problem*, also known in literature as *collision* or *conflict* resolution problem, consists in managing the channel access of a group of nodes, which form the *batch* (or *conflict set*), in such a way that each node successfully delivers *one* message to the inquirer. When a packet is successfully decoded by the inquirer, the source node is said to be *resolved* and leaves the batch. A *Batch Resolution Algorithm* (BRA) is a distributed algorithm that arbitrates the channel access in order to resolve all the nodes in the initial batch in the shortest time.

This work was partially supported by "Fondazione Cassa di Risparmio Padova e Rovigo" under the project "A large scale wireless sensor network for pervasive city-wide ambient intelligence."

The batch resolution problem resembles, in some aspects, the medium access control (MAC) problem in that both consist in managing the access of a group of nodes to a shared channel. However, it shall be noted that MAC protocols generally look at the channel contention as a steady-state phenomenon. Conversely, BRAs address scenarios where contention has a bursty nature.

The design and analysis of efficient BRAs have been deeply studied in the literature, in particular for non-CSMA slotted channels. Landmarks in this area are the *splitting-tree* algorithms devised by Hayes [1] and Capetanakis [2], whose interest has been recently reawaken in relation with the RFID tags [3]. An efficient collision resolution scheme for Poisson arrivals is the Clipped Modified Binary Tree (CMBT) algorithm, proposed by Gallager *et al.* in [4] with the name FCFS, and successively improved by Mosely and Humblet in [5]. The basic idea behind all these algorithms consists in splitting the initial conflict set in smaller subsets and, then, resolving one subset at a time in a recursive way. The various schemes differs in the policy used to split the conflict set in subsets. A comprehensive overview of these collision-resolution algorithms can be found in [6].

In pure-slotted systems, slots duration is constant, irrespective of whether they are idle, collided or successful. Conversely, in carrier-sense multiple access (CSMA) networks, idle slots are significantly shorter than busy (collided or successful) slots, because nodes can reveal the presence of signals on the channel by performing carrier-sensing. Accordingly, BRAs for CSMA channels split the initial conflict set in smaller parts, in order to increase the probability of empty slots while reducing that of collided slots. This principle has been applied, for instance, to ALOHA and FCFS algorithms, as discussed in [7].

The existing literature, in general, does not consider the time cost of feedbacks in the performance analysis of BRAs. In practical systems, however, each transmission (included feedbacks) takes a minimum amount of time for nodes to perform basic operations, such as RX/TX switching, signal detection and synchronization, and so on. This time may represent a significant part of the overall transmission time of a packet, in particular when the transmission rate is large. For instance, with reference to the IEEE 802.11 standard, the transmission of an acknowledgment frame (ACK) at the basic rate of 6 Mbit/s amounts to approximately 20% of the

transmission time of a full-size (1500 bytes) data frame sent at 54 Mbit/s. Therefore, the time cost of feedbacks may have non-negligible impact on the performance of BRAs, when considering commercial radio technologies. This observation makes questionable whether immediate feedback is actually the best strategy for the design of BRAs.

In this paper, we propose the *Adaptive-window Batch Resolution Algorithm with Deferred feedback* (ABRADE), which waives immediate feedback to adopt a contention scheme inspired to the classical framed Aloha approach [8]. The algorithm parameters are optimized on the basis of a dynamic programming argument that takes into account the cost of probing, idle sensing, collisions and successful transmissions. Comparison with classical BRAs shows that ABRADE brings significant performance gain in practical scenarios by reducing the feedback overhead.

The rest of the paper is organized as follows. Sec. II defines the system model and describes ABRADE. The mathematical and asymptotic analysis of ABRADE performance is developed in Sec. III, whereas in Sec. IV compare by simulation ABRADE against FCFS in two practical scenarios, based on the common IEEE 802.11g and IEEE 802.15.4 radio standards. Finally, Sec. V draws conclusions and discusses possible extensions to the proposed approach.

II. Algorithm description and analysis

In this section we first introduce the reference system model considered in this work and, then, we describe the ABRADE algorithm and derive the parameters optimization scheme.

A. System model

The scenario considered in this work entails a node, called inquirer, that wishes to collect a single packet from each of n surrounding nodes, which form the batch. The batch size n is generally random. Nonetheless, the theoretical analysis of ABRADE is developed under the assumption that n is known to the inquirer. This assumption will be relaxed in the simulation analysis presented in Sec. IV.

The inquirer is in charge of controlling the execution of the BRA and providing feedback. All nodes are supposed to be within the transmission/reception range of the inquirer and capable of carrier-sensing each other transmission [9], so that there is no hidden nodes. Also, we do not consider multi-hop transmissions. For the sake of simplicity, we make the classical assumption of noiseless radio channel: transmissions are always successful, except in case of collisions. Accordingly, time axis is divided in slots of unequal duration that will be referred to as *idle*, when no nodes transmit, *successful*, when a single node transmits, or *collided* when multiple nodes transmit simultaneously.

The transmission time of data packets is assumed to be constant and equal to T_{data}, inclusive of all mandatory guard times. As customary, all time measures will be normalized with respect to T_{data}, unless otherwise specified. Accordingly, the time duration of successful and collided slots is set equal to 1, whereas the time duration of an idle slot will be denoted by β, which is generally much less than 1.

B. The ABRADE algorithm

The ABRADE scheme works in successive *resolution rounds*. Transmissions in each round occurs as in the CSMA-based framed ALOHA algorithm, that is to say, each unresolved node, independently of the others, transmits its packet in a slot randomly chosen with uniform probability $q \leq 1/w$ among the w slots of the *contention frame*. Slots are marked as idle, successful or collided by the inquirer, according to their state. When the w slots of the contention frame are elapsed, the inquirer closes the round by broadcasting a *probe message* that contains an aggregate feedback field together with other control information, such as the frame size w and the transmission probability q to be used in the next round. Upon receiving the probe message, nodes check whether their previous transmission attempt was successful or not. In the first case, they immediately quit ABRADE, otherwise they keep competing in the next round, using the values of w and q carried by the last probe message.

C. Dynamic parameters optimization

The core of ABRADE consists in the dynamic adaptation of the parameters w and q to the residual multiplicity n of the batch after each resolution round. The adaptation aims at minimizing the *Batch Resolution Interval* (BRI), i.e., the average time required to resolve all the nodes in the initial batch, which is denoted by $\mathcal{T}(n)$. This is equivalent to maximizing the *batch throughput*, defined as the average number of nodes resolved in the unit time and given by

$$\lambda(n) = \frac{n}{\mathcal{T}(n)}\,. \tag{1}$$

To optimize the parameters we express the maximization problem in a recursive form that can be solved by dynamic programming [10]. Then, let s, c and i denote the number of successful, collided and idle slots, respectively, observed during a resolution round with contention frame size w, so that

$$w = i + s + c\,. \tag{2}$$

Denoting by y the time duration of the round, we have

$$y = s + c + i\beta + \beta_p = \beta_p + w - i(1-\beta) \tag{3}$$

where β_p denotes the transmission time of the probe packet. Hence, denoting by $\tau(n)$ the overall time required to resolve a batch of size n , we can write the following recursive form

$$\tau(n) = y + \tau(n-s)\,, \tag{4}$$

where $\tau(0) = 0$. Clearly, $\tau(n)$ is a random variable, whose expectation equals the BRI for a batch of size n, i.e.,

$$\mathrm{E}\left[\tau(n)\right] = \mathcal{T}(n)\,.$$

Thus, taking the expectation of both sides of (4) and applying the total probability theorem with respect to s we obtain

$$\begin{aligned}\mathcal{T}(n) &= \mathrm{E}\left[y\right] + \sum_{r=0}^{\min\{w,n\}} \mathcal{T}(n-r)p_s(r) \\ &= \frac{\mathrm{E}\left[y\right] + \sum_{r=1}^{\min\{w,n\}} \mathcal{T}(n-r)p_s(r)}{1-p_s(0)}\,,\end{aligned} \tag{5}$$

where $p_s(r)$ is the probability of exactly r successful slots in the resolution round. By basic combinatorial analysis [11], we obtain

$$p_s(r) = \sum_{m=r}^{n} \binom{n}{m} (wq)^m (1-wq)^{n-m} \times \sum_{j=r}^{\min\{w,m\}} \binom{w}{j}\binom{j}{r}\binom{m}{j} j!(-1)^{j-r} \frac{(w-j)^{m-j}}{w^m} \tag{6}$$

for $r \leq \min\{w, n\}$ and zero otherwise.[1] Moreover, from (3) we have

$$\begin{aligned} \mathrm{E}[y] &= \beta_p + w - \mathrm{E}[i](1-\beta) \\ &= \beta_p + w - w(1-q)^n(1-\beta). \end{aligned} \tag{7}$$

Finally, using (7) and (6) into (5) we obtain the following recursive expression

$$\mathcal{T}(n) = \frac{\beta_p + w - w(1-q)^n(1-\beta) + \sum_{r=1}^{\min\{w,n\}} p_s(r)\mathcal{T}(n-r)}{1-p_s(0)}, \tag{8}$$

where we did not expanded the term $p_s(r)$ for readability reasons. Now, let $\mathcal{T}^\star(k)$ denote the minimum BRI for a batch of size k that can be obtained by optimal setting ABRADE parameters during the entire resolution phase. Hence, from (8) we can write

$$\mathcal{T}^\star(n) = \min_{\substack{w,q \\ wq\leq 1}} \left\{ \frac{\beta_p + w - w(1-q)^n(1-\beta) + \sum_{r=1}^{\min\{w,n\}} p_s(r)\mathcal{T}^\star(n-r)}{1-p_s(0)} \right\} \tag{9}$$

which can be solved recursively, starting from $k = 1$ for which we have $\mathcal{T}^\star(1) = 1 + \beta_p$ for $w = 1$ and $q = 1$. The values of w and q that attain the minimum for each k are denoted by $w_n^\star$ and $q_n^\star$, respectively. Similarly, the throughput obtained by replacing (9) into (1) is denoted by $\lambda^\star(n)$.

III. Performance Analysis of ABRADE

Unfortunately, a direct inspection of (9) does not reveal much of the relation between $w_n^\star$ and $q_n^\star$ and the system parameters n, β and β_p. Nonetheless, plotting the solution of (9) when varying n for different values of β we obtained the curves in Fig. 1. The value of $q_n^\star$ is not reported, being always equal to $q_n^\star = 1/w_n^\star$. We observe that, as natural, both $\mathcal{T}^\star(n)$ and $w_n^\star$ grows almost linearly with n.

To better appreciate this relation, we inspect the throughput curves reported in Fig. 2 as a function of β, for different values of n. We first note that the throughput increases with n. In fact, provided that the contention parameters are adapted to the batch size, the overhead due to the control traffic (i.e., the probe message) becomes progressively less relevant as the size of the batch grows. Second, we note that the throughput curves rapidly converge toward a superior limit $\Lambda = \lim_{n\to\infty} \lambda^\star(n)$ (dashed curve in Fig. 2). To determine

[1] The presence of binomials in (6) can raise numerical stability problems. However, as suggested in [**?**], it is possible to derive $p_s(r)$ in a recursive manner that overcomes the numerical stability issues.

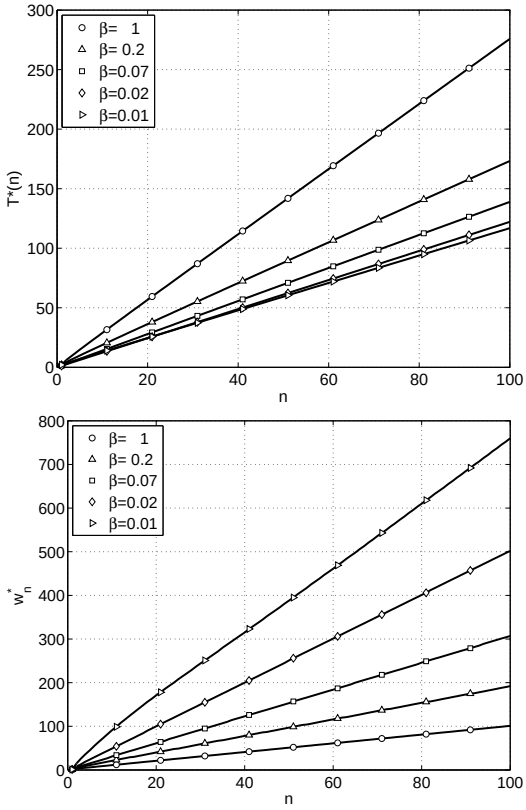

Figure 1. Minimum batch resolution interval $\mathcal{T}^\star(n)$ (above) and optimal contention window size $w_n^\star$ (below) as a function of the batch size n, for different values of β and with $\beta_p = 1$.

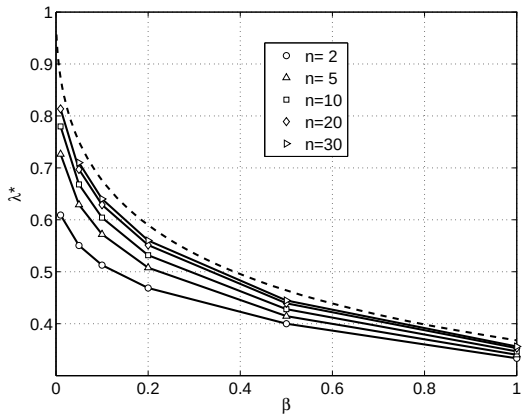

Figure 2. Throughput Λ vs β, for different values of the batch size n and $\beta_p = 1$.

an analytical expression of Λ, we assume $\mathcal{T}^\star(k) \simeq k\Lambda$ for sufficiently large k. From (5) we thus obtain

$$\begin{aligned} n\Lambda &\simeq \mathrm{E}[y] + \sum_{s=0}^{\min\{w_n^\star,n\}} (n-r)\Lambda\, p_s \\ &\simeq \mathrm{E}[y] + n\Lambda - \Lambda\,\mathrm{E}[s], \end{aligned} \tag{10}$$

from which

$$\Lambda \simeq \frac{\mathrm{E}[s]}{\mathrm{E}[y]} = \frac{n w_n^\star q_n^\star (1-q_n^\star)^{n-1}}{\beta_p + w_n^\star - w_n^\star(1-q_n^\star)^n(1-\beta)}. \tag{11}$$

We now introduce

$$\mu_n = n q_n^\star = \frac{n}{w_n^\star}, \tag{12}$$

which corresponds to the mean number of transmissions per slot, with optimal parameters setting. Using (12) in (11) we get

$$\Lambda \simeq \frac{\mu_n\left(1-\frac{\mu_n}{n}\right)^{n-1}}{\frac{\beta_p}{w_n^\star}+1-\left(1-\frac{\mu_n}{n}\right)^n(1-\beta)} \tag{13}$$

and letting $n \to \infty$ we finally obtain

$$\Lambda = \frac{\mu_\infty \exp(-\mu_\infty)}{1+b_p-(1-\beta)\exp(-\mu_\infty)} \tag{14}$$

where $b_p = \lim_{n\to\infty} \beta_p/w_n^\star$, while

$$\mu_\infty = \lim_{n\to\infty} \mu_n\,, \tag{15}$$

which is the average number of transmissions per slot that maximizes the asymptotic throughput Λ. To determine this value, we set to zero the derivative of the right-hand side of (14) with respect to μ_∞ and, after some algebra, we find the following transcendent equation

$$(1+b_p)(1-\mu_\infty) = (1-\beta)\exp(-\mu_\infty)\,, \tag{16}$$

which can be easily solved with numerical methods, such as bisection. Note that, as we will see later on, b_p is typically negligible, so that μ_∞ strictly depends on β only, spanning from 0 to 1 as β varies from 1 to 0, as shown in Fig. 3. This behavior reflects the principle according to which the channel access strategy shall be less aggressive when idle slots are much shorter than busy slots. Finally, replacing (16) into (14)

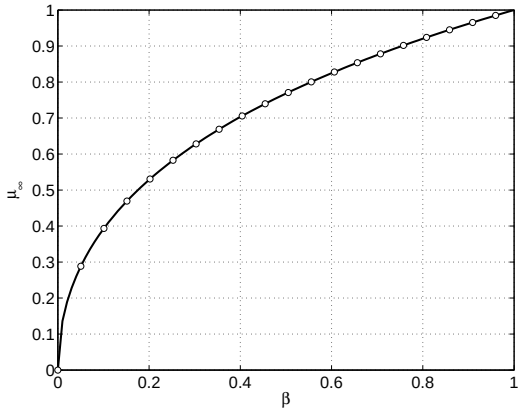

Figure 3. Asymptotic number of transmissions per slot, μ_∞, as a function of the normalized idle slot duration β ($b_p = 0$).

yields

$$\Lambda = \exp(-\mu_\infty)\,, \tag{17}$$

which is the superior limit to ABRADE throughput, represented by the dashed curve in Fig. 2.

Another interesting performance measure, here denoted by $\mathcal{E}$, is the mean number of transmissions required by a node to be resolved. To determine an estimate of $\mathcal{E}$ we assume that the success probability for a node transmission is stationary and equal to P_s. Under this hypothesis, the number of transmission attempts to successfully deliver the message to the inquirer is a geometric random variable, with mean

$$\mathcal{E} = \frac{1}{P_s}\,.$$

We then observe that, for sufficiently large values of n, the number of transmissions in a certain slot can be approximated by a Poisson-distributed random variable with mean μ_∞. Accordingly, the success probability for a certain transmission, which is equal to the probability that no other transmissions occur in the same slot, can be approximated as

$$P_s = \exp(-\mu_\infty)\,, \tag{18}$$

from which we get

$$\mathcal{E} = \exp(\mu_\infty)\,. \tag{19}$$

This performance measure is related to the energy efficiency of the protocol since packet transmission is a major source of energy consumption in wireless systems. Therefore, the larger $\mathcal{E}$, the more energy is consumed by a node to be resolved.

IV. Case study

In this section, we compare ABRADE against FCFS algorithm [7], which is the fastest known conflict resolution algorithm for Poisson-distributed batch multiplicities. FCFS is based on the immediate feedback assumption, according to which each node receives at the end of the slot a feedback that specifies whether the slot was idle, successful or collided. In practical CSMA-based networks, idle and collision feedbacks are implicitly provided by the carrier-sense mechanism. Conversely, successful feedback is explicitly carried by an acknowledgement (ACK) message, which typically consists in a short control packet without any payload field.

In order for the comparison to have practical significance, we set the system parameters according to the specifications of two popular off-the-shelf radio technologies, namely IEEE 802.11 and IEEE 802.15.4 that, for ease of notation, will be henceforth referred to as WF (WiFi) and ZB (ZigBee), respectively.

A. System parameters

The data structures and carrier-sense mechanism of WF and ZB are similar and will be described in a unified manner. Let t_{BCK} denote the reference idle (backoff) slot time. Furthermore, let T_{PCK} be the transmission time of a data packet, consisting of PHY preamble and header, and MAC header, payload and trailer. In case of unacknowledged transmissions, successive data frames have to be separated by (at least) a certain Inter Frame Space (IFS) time interval, here denoted by t_{IFS}. In case of acknowledged transmission, instead, the ACK frame of duration T_{ACK} shall be returned within a time interval $t_{ACK} < t_{IFS}$ after the completion of the data frame transmission. Moreover, an IFS shall follow any ACK before new data transmissions. If a valid ACK is not received within a certain timeout t_{TO} the transmission is assumed to be collided and a new data frame can be transmitted immediately after.

With reference to the system model defined in Sec. II, the system parameters can thus be set as follows

$$\begin{aligned} &T_{data} = T_{PCK} + t_{IFS}\,; \quad \beta = \frac{t_{BCK}}{T_{data}}\,; \\ &\beta_i = 0\,; \quad \beta_s = \frac{T_{ACK}+t_{ack}}{T_{data}}\,; \quad \beta_c = \frac{t_{TO}-t_{IFS}}{T_{data}}\,; \end{aligned} \tag{20}$$

where β_i, β_s and β_c are the normalized feedback time for idle, successful, and collided slots, respectively.

The probe message can be realized by using a data frame whose payload carries the binary representation of the contention frame size $w^\star$ for the upcoming resolution round and the aggregate feedback field, which may consist of a bit-mask in one-to-one correspondence with the slots in the contention frame, in such a way that bits 1 denote successful slots, whereas bits 0 indicate collided and idle slots.[2] Note that, for large frame sizes, the aggregate ACK field may not fit in the payload of a single MAC frame. In this case, it will be needed to send multiple consecutive probe messages. Making a worst-case assumption, we then approximate the length of the probe message for a contention frame of w slots as

$$\beta_p = \lceil w b_p \rceil \,, \quad \text{with} \quad b_p \simeq 1/L_{\max}$$

where $L_{\max}$ is the maximum payload size of a MAC frame.

The values of all the system parameters for the WF and ZB scenarios are reported in Tab. I.

B. Batch multiplicity

The FCFS algorithm is designed to resolve conflicts among messages that arrive according to a Poisson process and it makes use of the message arrival instants to arbitrate the channel access. Therefore, the *plain* application of FCFS algorithm in case of batch arrivals yields poor performance [12]. However, the optimality of the algorithm can be easily re-established for batch arrivals, provided that the batch size is Poisson-distributed with known mean $N = \mathrm{E}[n]$. In this case, as explained in [13], each node in the batch may generate at random its *virtual arrival instant* in the time interval $[0, N/\lambda_{\max}]$, where $\lambda_{\max}$ is the maximal throughput achievable by the FCFS algorithm [7]. FCFS algorithm can then be applied by considering the virtual arrival epochs in place of the actual message arrival times. Clearly, any distribution of the batch size other than Poisson will yet result in some performance loss. Therefore, for fair comparison, we only consider batches with Poisson-distributed multiplicity n.

C. Batch size estimator

As mentioned, ABRADE parameters have been optimized under the assumption of perfect knowledge of the batch size n at each resolution round. Relaxing this assumption will yield a certain performance loss and, moreover, will require to couple ABRADE with a batch size estimation (BSE) module that feeds ABRADE with dynamic estimates $\hat{n}$ of the actual batch size n. Although interesting by its own (see, for instance, [14]), the analysis of efficient estimators is out of the scope of this paper. Instead, we consider an extremely simple BSE that works as follows. The initial estimate is set to

$$\hat{n} = N \,,$$

and, after each resolution round, it is updated as

$$\hat{n} = \max\{\hat{n}_o, 2c + s\} \,, \tag{21}$$

where $\hat{n}_o$ is the previous estimate, whereas s and c, as usual, denote the number of successful and collided slots in the resolution round. The residual bath size is then obtained by subtracting to $\hat{n}$ the overall number of nodes resolved since the beginning of the resolution process.

[2]The parameter $q^\star$ is always assumed to be equal to $1/w^\star$, so that it is not included in the probe message.

Table I
SYSTEM PARAMETERS SETTING IN WF AND ZB SCENARIOS.

	T_{data} [ms]	β	β_i	β_s	β_c	$L_{\max}$ [bits]
WF	0.399	0.0225	0	0.1319	0.1319	18496
ZB	4.896	0.0654	0	0.1111	0.0458	944

D. Simulation results

We simulated both ABRADE and FCFS algorithms by considering the system parameters in Tab. I. The mean batch size N has been varied from 1 to 1500. For each value of N, we run 20000 independent instances of both the algorithms. Graphs report the 99% confidence interval for each point, though error bars are tight-fitting the curve and mostly covered by marks.

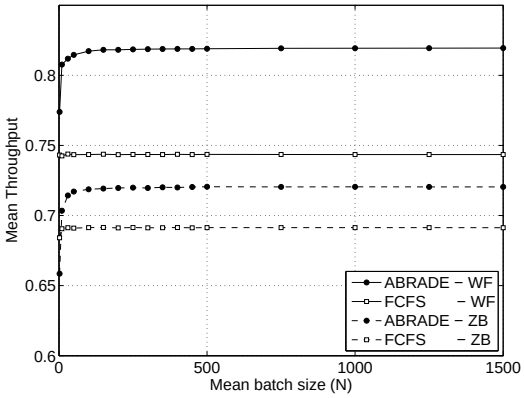

Figure 4. Mean throughput $\mathrm{E}[\lambda(n)]$ of ABRADE and FCFS as a function of the mean batch size N, in WF and ZB scenarios.

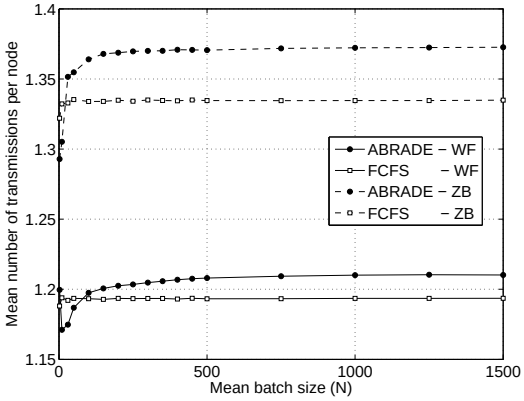

Figure 5. Mean number of transmissions per node, $\mathcal{E}$, for ABRADE and FCFS as a function of the mean batch size N, in WF and ZB scenarios.

Fig. 4 shows the mean throughput $\mathrm{E}[\lambda(n)]$ obtained by ABRADE (black marks) and FCFS (white marks) in WF (solid lines) and ZB (dashed lines) scenarios. Throughput curves initially grows rapidly with N, to bend toward the asymptotic values for $N \geq 150$. We observe that ABRADE performs better than FCFS for every value of N in both scenarios, though the performance gap is less marked for ZB scenario whose feedback costs are smaller than for WF.

In Fig. 5 we report $\mathcal{E}$ as a function of N. In this respect, the two algorithms show very similar performance, with a slight

Table II
ASYMPTOTIC PERFORMANCE MEASURES OF ABRADE IN WF AND ZB SCENARIOS: SIMULATION (AT $N = 1500$) VS THEORY.

	Λ		$\mathcal{E}$	
ABRADE	Simulation	Theory	Simulation	Theory
WF	0.81775	0.8202	1.2102	1.2192
ZB	0.71924	0.7229	1.3726	1.3832
FCFS	Simulation	Theory	Simulation	Theory
WF	0.7409	0.8388	1.1936	1.1913
ZB	0.6901	0.7598	1.3349	1.3033

advantage of FCFS over ABRADE for large batch sizes. This is due to the immediate feedback mechanism that allows FCFS to resolve more effective than ABRADE in resolving collisions. However, the cost of immediate feedback absorbs this small advantage, so that ABRADE is more efficient than FCFS in terms of batch resolution time. Note that, for very small values of n, ABRADE actually exhibits lower $\mathcal{E}$ but, on the other hand, larger $\mathcal{T}(n)$ than FCFS. The reason is that, for small batches, the cost of probe message is relevant, so that ABRADE chooses longer frames in order to reduce the mean number of transmissions per frame and, in turn, the collision probability and the number of rounds required to resolve the batch. This yields a reduction of $\mathcal{E}$.

Finally, Tab. II collects the asymptotic performance measures of ABRADE and FCFS obtained by simulation and theoretical analysis. We can see that simulation and theoretical values for ABRADE are in very agreement, thus confirming the validity of the mathematical model and the marginal impact of the simplifying assumptions considered in the analysis of Sec. III. Conversely, the simulation results for the asymptotic throughput provided by FCFS are significantly worse than the theoretical values provided in [7], which were obtained under the simplifying assumption of negligible feedback costs.

V. CONCLUSIONS

In this paper we presented ABRADE, a batch resolution algorithm, derived from the well-known framed ALOHA scheme, that dynamically adapts the length of the contention frame during the resolution of the batch in order to minimize the overall batch resolution interval. Conversely to most conflict resolution algorithms presented in the literature, ABRADE waives immediate feedback paradigm in favor of aggregate feedback, in order to reduce overhead.

ABRADE performance has been analyzed, both mathematically and by simulations, and compared against FCFS, the best performing algorithm in the literature based on the immediate feedback paradigm. The comparison confirmed that, by setting the system parameters in accordance with the specifications of common radio standards, such as IEEE 802.11 (WF) and IEEE 802.15.4 (ZB), the asymptotic throughput of FCFS exhibits a loss with respect to the theoretical values derived in [7] of approximately 9% for WF and 6% for ZB. Conversely, ABRADE approaches the theoretical optimal values of FCFS, while maintaining approximately the same mean number of transmissions per node and, in turn, the same mean energy consumption.

It is worth noting that all the complexity of the parameters optimization is at the inquirer, whereas the other nodes have to perform only basic operations. Actually, the frame-length optimization may be computationally demanding, thus potentially preventing the on-line application of ABRADE. However, the computation can be performed offline and the optimal parameters can then be stored in a table at the inquirer. Alternatively, it is possible to make use of the asymptotic results by approximating $w_n^\star$ as $\lceil \mu_\infty n \rceil$. In this way, the frame length adaptation can be performed by a simple rounded multiplication.

We finally observe that ABRADE may be further ameliorated by exploiting idle-slot feedbacks that are implicitly provided by the carrier-sense mechanism without any extra time-cost. This information may be used, for instance, to slid forward the contention frame in case of runs of idle slots, in order to avoid almost-certain collisions in the last part of the frame. Furthermore, the application of ABRADE without any *a priori* knowledge of the batch size requires the design of a batch-size estimate module that is capable of providing a sufficiently accurate estimate of n to correctly drive the ABRADE optimization mechanism. Such extensions are left for future investigation.

REFERENCES

[1] J. Hayes, "An adaptive technique for local distribution," *Communications, IEEE Transactions on*, vol. 26, no. 8, pp. 1178–1186, Aug 1978.

[2] J. Capetanakis, "Tree algorithms for packet broadcast channels," *Information Theory, IEEE Transactions on*, vol. 25, no. 5, pp. 505–515, Sep 1979.

[3] P. Popovski, "Tree protocols for RFID tags with generalized arbitration spaces," in *Spread Spectrum Techniques and Applications, 2008. ISSSTA '08. IEEE 10th International Symposium on*, Aug. 2008, pp. 18–22.

[4] R. Gallager, "Conflict resolution in random access broadcast networks," in *AFOSR Workshop Commun. Theory Appl., Provincetown, MA*, September 1978, pp. 74–76.

[5] J. Mosely and P. Humblet, "A class of efficient contention resolution algorithms for multiple access channels," *Communications, IEEE Transactions on*, vol. 33, no. 2, pp. 145–151, Feb 1985.

[6] M. Molle and G. Polyzos, "Conflict resolution algorithms and their performance analysis," Department of Computer Science and Engineering, UCSD, East Lansing, Michigan, Tech. Rep. CS93-300, July 1993.

[7] D. Bertsekas and R. Gallager, *Data networks (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.

[8] S. Tasaka, "Stability and performance of the R-Aloha packet broadcast system," *IEEE Trans. Comput.*, vol. C, no. 32, pp. 717–726, Aug. 1983.

[9] K. Jamieson, B. Hull, A. K. Miu, and H. Balakrishnan, "Understanding the Real-World Performance of Carrier Sense," in *ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND)*, Philadelphia, PA, August 2005.

[10] R. Bellman, *Dynamic Programming*. Princeton University Press, Princeton, NJ. 1957, Republished 2003.

[11] G. Pierobon, A. Zanella, and A. Salloum, "Contention-TDMA protocol: Performance evaluation," *IEEE Transactions on Vehicular Technology*, vol. 51, pp. 781–788, 2002.

[12] I. Cidon and M. Sidi, "Conflict multiplicity estimation and batch resolution algorithms," *Information Theory, IEEE Transactions on*, vol. 34, no. 1, pp. 101–110, Jan 1988.

[13] P. Popovski, F. H. P. Fitzek, and R. Prasad, "A class of algorithms for collision resolution with multiplicity estimation," *Algorithmica*, vol. 49, no. 4, pp. 286–317, 2007.

[14] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2006, pp. 322–333.