Multiframe Maximum-Likelihood Tag Estimation for RFID Anticollision Protocols

Javier Vales-Alonso, *Member, IEEE*, Victoria Bueno-Delgado, *Member, IEEE*, Esteban Egea-Lopez, Francisco J. Gonzalez-Castaño, and Juan Alcaraz

***Abstract*—Automatic identification based on radio frequency identification (RFID) is progressively being introduced into industrial environments, enabling new applications and processes. In the context of communications, RFID rely mostly on Frame Slotted Aloha (FSA) anticollision protocols. Their goal is to reduce the time required to detect all the tags within range (identification time). Using FSA, the maximum identification rate is achieved when the number of contending tags equals the number of contention slots available in the frame. Therefore, the reader must estimate the number of contenders and allocate that number of slots for the next frame. This paper introduces the new MFML-DFSA anticollision protocol. It estimates the number of contenders by means of a maximum-likelihood estimator, which uses the statistical information from several frames (multiframe estimation) to improve the accuracy of the estimate. Based on this expected number of tags, the algorithm determines the best frame length for the next reading frame, taking into account the constraints of the EPCglobal Class-1 Gen-2 standard. The MFML-DFSA algorithm is compared with previous proposals and found to outperform these in terms of (lower) average identification time and computational cost, which makes it suitable for implementation in commercial RFID readers.**

***Index Terms*—Anticollision protocol, dynamic frame slotted aloha (DFSA), EPCglobal, maximum-likelihood estimation, radio frequency identification (RFID).**

I. Introduction

RADIO-FREQUENCY IDENTIFICATION (RFID) is enabling a paradigm shift in key areas of manufacturing and process automation. The goal of RFID is to allow the identification of products, objects, or people nearby by means of radio-frequency (RF) links [1]. The communication takes place between small and inexpensive devices called *tags*, which are attached to the items to be tracked, and *readers*, which collect and manage information about those items. Most RFID systems are passive, that is, the tags are battery-less and are solely powered by the RF signals of the readers. This work focuses exclusively on this kind of RFID systems. Passive tags are intended to identify people, animals, pallets, or other objects. Readers continuously transmit RF signals asking tags to identify themselves and thereby defining checking areas. Typical reading ranges are a few meters in the best conditions. When tags cross these checking areas they are powered by the reader signals and send their stored information back to the reader, thereby identifying the objects to which they are attached. Compared to other identification technologies like barcodes, RFID permits automatic identification without human intervention and without the need for a line-of-sight between the reader and the tags. RFID is used in a wide range of industrial fields, such as traceability management [2], supply chains [3], indoor positioning [4], and so forth. Besides, active research in various key areas such as physical design [5], communication protocols [6], security [7], and middleware development [8] is underway to improve RFID performance and reduce deployment and operation costs, which is essential for industrial use.

The identification process involves communications between the reader and the tags and takes place in a shared wireless channel. Basically, the reader *interrogates* tags nearby by sending a *Query* packet (the exact format of this packet depends on the particular identification procedure). Tags are energized by the reader's signal and respond to this request with their identification. When several tags answer simultaneously, a collision occurs, and the information cannot be retrieved. Therefore, an anticollision mechanism is required when multiple tags are in range. In addition, the extreme simplicity of the tags places considerable constraints on the design of collision-solving methods, whose intelligence must rely almost exclusively on the reader.

Anticollision algorithms for passive RFID systems can be classified into two groups: **tree-based** protocols and **Aloha-based** protocols.

In tree-based anticollision protocols (e.g., [9]), the reader consecutively splits the tag set into disjoint subsets until, eventually, a set has a single tag whose identification can be obtained without collisions. This procedure is repeated until the identity of all tags is retrieved. As a result, identification time may be too long. These protocols are attractive for specific applications such as access control systems. They are mainly used in low-frequency (LF) and high-frequency (HF) RFID implementations.

Aloha-based protocols, also called probabilistic or random access protocols, are the most prevalent in the UHF band. They are designed for situations in which the reader does not know exactly how many tags will cross its checking area. The most common Aloha RFID protocol is Frame Slotted-Aloha (FSA), a variation of Slotted-Aloha. As in Slotted-Aloha, time is divided into time units called slots. However, in FSA, slots are

Manuscript received June 15, 2010; revised December 07, 2010, February 11, 2011; accepted March 16, 2011. This work was supported by project CALM TEC2010-21405-C02, funded by the Spanish Ministerio de Innovación y Ciencia, also supported through European regional development funds (ERDF). It has been developed within the framework of "Programa de Ayudas a Grupos de Excelencia de la Región de Murcia", funded by Fundación Seneca, Agencia de Ciencia y Tecnología de la Región de Murcia (Plan Regional de Ciencia y Tecnología 2007/2010). Paper no. TII-10-06-0136.

J. Vales-Alonso, V. Bueno-Delgado, E. Egea-Lopez, and J. Alcaraz are with the Department of Information Technologies and Communications, Technical University of Cartagena (UPCT), E-30203 Cartagena, Spain (e-mail: javier.vales@upct.es).

F. J. Gonzalez-Castaño is with the Departament of Telematics Engineering, University of Vigo, 36310 Vigo, Spain.

Digital Object Identifier 10.1109/TII.2011.2158831

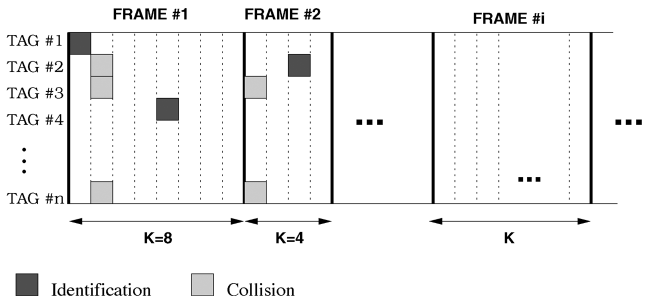

Fig. 1. FRSA operation.

subject to a superstructure called a "frame." The reader starts the identification process with an identification frame by sending a *Query* packet with information about the frame length (K slots) to the tags. The frame length is kept unchanged during the whole identification process. At each frame, each unidentified tag selects a slot at random from among the K slots to send its identifier to the reader. FSA achieves reasonably good performance at the cost of requiring a central node (the reader) to manage slot and frame synchronization. FSA has been implemented in many commercial products and has been standardized in the ISO/IEC 18000-6C [10], ISO/IEC 18000-7 [11], and EPCGlobal Class-1 Gen-2 (EPC-C1G2) standards [12].

When tags outnumber available slots, identification time increases considerably due to frequent collisions. On the other hand, if the slots outnumber the tags, many slots will be empty in the frame, which also leads to long identification times. Dynamic FSA (DFSA) protocols were conceived to address this problem. They are similar to FSA but the number of slots per frame is variable. In other words, parameter K may change from frame to frame in the *Query* packet to adjust the frame length. Fig. 1 shows an example of a generic DFSA protocol, where after many empty slot occurrences in the first frame the reader reduces the frame length for the second frame to decrease the identification time for the remaining tags.

DFSA operation is optimal in terms of reading throughput (rate of identified tags per slot) when the frame length equals the number of contenders [13]. Moreover, note that maximizing reading throughput is equivalent to minimizing identification time, that is, the time required to identify all the tags in a population. The challenge is to minimize identification time by selecting the best frame length for each frame, which will depend on the number of contending tags. Therefore, the reader should ideally know the actual number of competing tags and allocate that number of slots to the next frame. However, the number of contenders is unknown and must be estimated somehow. Some simple DFSA algorithms (see Section III) use heuristics to select the value of K directly. However, the selection is usually performed in two steps (we call this operation *indirect selection*). First, the reader estimates the number of tags ($\hat{n}_i$) that competed in the previous frame i. Thus, the expected number of contenders in the next frame is ($\hat{n}_i - s_i$), where s_i denotes the number of tags successfully identified in the previous frame. The frame length in the next frame (K_{i+1}) is selected as a function of $\hat{n}_i - s_i$. There are different ways to compute $\hat{n}_i$, mainly heuristics, Minimum Squared Error (MSE) estimation, and Maximum-Likelihood (ML) estimation.

Besides, to estimate $\hat{n}_i$, DFSA readers use different information. In a given frame i, it is possible to monitor three variables: the number of slots filled with a single transmission (s_i), the number of empty slots (e_i), and the number of slots with collision (c_i). Note that $K_i = s_i + e_i + c_i$. Since the readers always know the frame length (K_i), two variables out of $\{s_i, e_i, c_i\}$ give full information about the events in the frame (*full-frame information estimation*). However, many DFSA algorithms only use one of these variables, either c_i or s_i. Indeed, information from several frames, $i, i-1, i-2, \ldots$ can be used (*multiframe estimation*). Multiframe (MF) estimators usually provide an estimate $\hat{n}$ of the initial number of tags in the identification process.

The accuracy of the estimator will be directly related to the information available. Single-frame estimation makes sense for a continuous flow (of tags that continuously enter and leave the coverage area), since in this case frame information quality decreases with time. In contrast, MF is advisable if new tag populations do not appear until the previous set has been completely identified. Note that the reader may also enforce this behavior. For instance, if a conveyor belt is carrying the tags, the reader may stop it when necessary.

Different DFSA algorithms have been proposed to optimize the identification process based on the previous concepts. The most relevant ones have been studied in depth in a previous study by our group [14]. The shortcomings identified in that work (see Section III) motivated us to propose a new DFSA algorithm that improves the estimation of $\hat{n}$ and provides an optimal criterion to select K. Besides, in this paper, we explicitly address the computational feasibility of our algorithm.

The new estimator is a MF DFSA algorithm based on a ML estimation of $\hat{n}$, which we call MFML-DFSA. The implementation is addressed in the context of the EPCglobal Class 1 Gen 2 (EPC-C1G2) [12] standard (see Section II). Thus, frame length cannot be an arbitrary natural number, but a number in the set $\{K = 2^Q : Q = 1, \ldots, 15\}$. We must remark that while previous work simply selected the frame length as the nearest value in that set to the estimated number of contenders, we demonstrate (in Section IV-A) that this assignment is suboptimal and explicitly compute the optimal Q value as a function of the expected contenders. Therefore, MFML-DFSA may be directly adopted by current EPC-C1G2 RFID reader devices, *without modifications at the tag side*. The results (Sections V and VI) show that MFML-DFSA achieves shorter identification times than previous DFSA proposals.

The rest of this paper is organized as follows. Section II describes the EPCglobal Class-1 Gen-2 standard, currently used in UHF RFID passive systems. Section III discusses the state-of-the-art in DFSA algorithms. The MFML-DFSA algorithm is thoroughly explained in Section IV. Section V discusses its implementation, particularly its computational feasibility. Section VI evaluates the performance of MFML-DFSA and the main DFSA alternatives. Finally, Section VII concludes this paper.

II. EPCGLOBAL CLASS-1 GEN-2

EPCglobal, an industry-oriented organization, has developed the Electronic Product Code (EPC) standard EPCglobal Class 1 Gen 2 (EPC-C1G2) [12]. EPC-C1G2 proposes an anticollision mechanism for passive RFID systems based on a variation

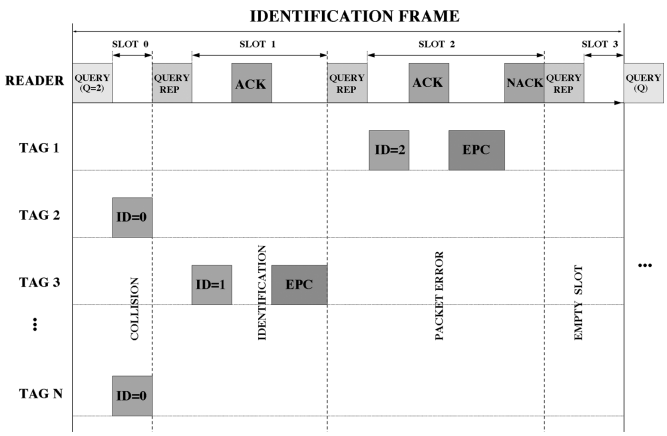

Fig. 2. EPCglobal Class-1 Gen-2: Fixed frame length procedure.

of FSA. Fig. 2 illustrates the operation of this mechanism. In the resting state (no ongoing identification process), the reader monitors the environment to detect new tags, with continuous *Broadcast* packet transmissions. Tags within range reply immediately. If several tags answer simultaneously, a collision occurs. When the reader detects the collision it starts a new identification process following an FSA schema. Moreover, EPC-C1G2 allows two identification procedures.

- *Fixed frame length procedure* (fixed EPC-C1G2): all identification frames have the same number of slots.
- *Variable frame length procedure* (variable EPC-C1G2): the reader can change the number of slots per frame dynamically (DFSA operation).

A. Fixed Frame Length Procedure

An identification frame starts when the reader transmits a *Query* packet, including a four-bit field with parameter $Q \in [0, \ldots, 15]$, stating that the length of the frame will be $K = 2^Q$ slots. Tags within range receive this packet and generate a random number r in the interval $[0, 2^Q - 1]$. This number represents the slot within the frame where the tag has randomly decided to send its identification number $\mathrm{ID} = r$. The reader controls the beginning of each slot by transmitting a *QueryRep* packet, except in the first slot (slot #0), which starts automatically immediately after the *Query* packet. The tags use r to set a counter, which is decreased upon reception of a new *QueryRep*. When the counter reaches 0, the tag transmits its identifier ID, which corresponds to the random value that was initially calculated. Note that it must also be equal to the slot number in the frame. After the ID is transmitted, four situations are possible.

- If several tags select the same slot, a collision occurs, and the reader is unable to decode an ID packet. The reader reacts by starting a new slot with a *QueryRep* packet (see slot #0 in Fig. 2). Affected tags update their counter to $r = 2^Q - 1$. That means that they will not contend again until the next frame.
- If a single tag responds but the reader cannot decode its ID packet, then, after a bounded time, the reader considers that the slot has expired and starts a new slot by sending a *QueryRep* packet. In this case, the tag also updates its counter to $r = 2^Q - 1$ when *QueryRep* is received.
- If the reader correctly receives the ID that matches the current slot number, it responds with an *Ack* packet. Although all tags receive the packet, only the winner answers with a *Data* packet (including its EPC code, which is its actual identification code). If the reader receives the *Data* packet correctly, it answers with a *QueryRep* packet, thus starting a new slot. Besides, the winning tag quits from the identification process (see slot #1 in Fig. 2). However, if the reader does not receive a correct *Data* packet within a bounded time, it considers that the slot has expired and sends a *Nack* packet. Again, only the involved tag updates its counter value to $r = 2^Q - 1$. Thus, this tag will not contend again in this identification frame (see slot #2 in Fig. 2). After this, the reader also sends a new *QueryRep* packet to begin the new slot.
- If the reader does not receive a packet before a given deadline, it is assumed that the slot is empty (see slot #3 in Fig. 2), and the reader starts a new one by sending a new *QueryRep* packet.

This procedure continues until the identification frame finishes. Then, the reader sends a new *Query* packet to start a new frame. Unidentified tags compete again in the new frame, selecting a new random r value. Eventually, all tags are identified and the procedure ends. This takes place when all the slots are empty in a frame.

B. Variable Frame Length Procedure

To mitigate the poor efficiency of EPC-C1G2 with fixed frame length, the standard proposes a variable frame length procedure, named variable EPC-C1G2 or variable EPC. Although EPC-C1G2 is seamlessly compatible with DFSA protocols that dynamically select Q in every new frame, the variable EPC method operates by resetting the frame whenever a slot ends. Note that in EPC-C1G2 the frame structure can also be adjusted by resetting the identification frame by sending a new *Query* packet with the same Q value or a new one instead of the next *QueryRep* packet when a slot ends. In this paper, we denote this slot-by-slot operation as *Q-slot*, and the normal frame-by-frame operation as *Q-frame* operation. The variable EPC procedure operates in a *Q-slot* fashion, selecting the new Q value at the end of each slot by means of the algorithm depicted in Fig. 3: when a slot ends, the reader checks if it was empty, successful or affected by a collision. Accordingly, the reader updates a floating point variable Q_{fp}. Then, the nearest integer to Q_{fp} is selected as Q, and a new *Query* packet with that value is sent. Variable C in (0.1, 0.5) (see Fig. 3) controls the selection of the frame length. It can be tuned dynamically to improve performance. However, the standard leaves the selection of C values open.

III. Related Work

The most relevant DFSA algorithms have been studied in [14]. In this section, we briefly review that study, together with the conclusions drawn. We also emphasize the key differences between MFML-DFSA and other DFSA algorithms. Table I shows the characteristics of the most remarkable DFSA proposals according to the criteria described in the previous sections. The characteristics of the MFML-DFSA proposal are summarized as well in Table I. We consider four major groups of protocols, each stemming from a different operative root: variable EPCglobal protocols, indirect heuristics, error

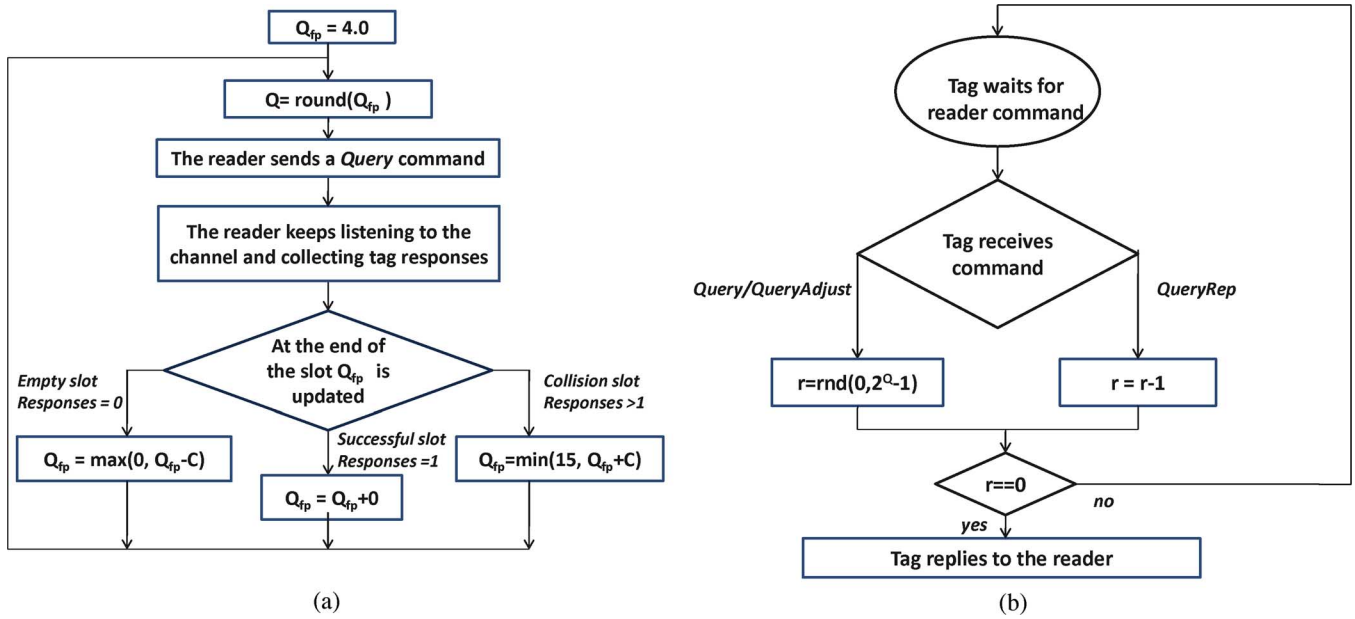

Fig. 3. EPCglobal Class-1 Gen-2, variable frame-length procedure. (a) Reader protocol. (b) Tag protocol.

TABLE I
COMPARISON OF DFSA PROTOCOLS

	Protocol	Operation	Multi-Frame Estimation	Full-Frame Information	Q Selection	$\hat{n}$ Computation
Dynamic EPC protocol family	Dynamic EPC [12]	Q-slot	No	–	Heuristic	–
	Q^{+-} [15]	Q-slot	No	–	Heuristic	–
	Optimum-C [16]	Q-slot	No	–	Heuristic	–
	SCS [17]	Q-slot	No	–	Heuristic	–
Indirect heuristics	Schoute [18]	Q-frame	No	No	Indirect	Heuristic
	Lower Bound [19]	Q-frame	No	No	Indirect	Heuristic
	Wang [20]	Q-frame	No	No	Indirect	Heuristic
	C-ratio [21]	Q-frame	No	No	Indirect	Heuristic
	Chen-1 [22]	Q-frame	No	Yes	Indirect	Heuristic
Error minimization estimators	MSE-Vogt [19]	Q-frame	No	Yes	Indirect	MSE
	MSE-SbS [23]	Q-slot	Yes	Yes	Indirect	MSE
Maximum likelihood estimators	Chen-2 [24]	Q-frame	No	Yes	Indirect	ML
	SbS [6]	Q-slot	Yes	Yes	Indirect	ML
	Chen-3 [27]	Q-frame	No	Yes	Indirect	ML
	Floerker [28]	Q-frame	Yes	Yes	Indirect	ML
	MFML-DFSA	Q-frame	Yes	Yes	Indirect	ML

minimization estimators, and maximum likelihood estimators. We review them all, focusing on those based on ML estimators, since they achieve the best performance as noted in [14]. Section VI compares them with our approach.

A. Variable EPCglobal Protocol Family

As described in Section II, EPC-C1G2 proposes a variable frame-length mechanism as an alternative to the fixed frame-length schema. It adjusts the frame length slot-by-slot (*Q-slot* operation) following the heuristic shown in Fig. 3. In this schema, the value of parameter C affects the computation of Q_{fp} and thus the value of Q. Since the value of C is open in the standard (as stated in the previous section), different alternatives to set it dynamically have been proposed, such as the Q^{+-} algorithm [15], the *Optimum-C* protocol [16], and the *Slot-Count-Selection* algorithm [17]. The performance of these algorithms is limited, as discussed in depth in [14].

B. Indirect Heuristics

Indirect heuristics estimate the number of contenders by means of oversimplified formulas, and then adjust the frame length (K) to the nearest power of two. Most proposals of this type only use information from the last frame (see Table I). As we stated in [14], their performance is poor compared with statistical estimation methods. Some examples of this family are the work by Schoute [18], the *Lower Bound* estimation [19] or the procedures proposed by Wang [20], Cha [21], and Chen [22].

C. Error Minimization Estimators

In [19], Vogt proposed a procedure based on MSE estimation, which minimizes the Euclidean norm of the vector difference between actual frame statistics and their expected values. The statistics considered are the number of empty, successful, and collision slots. However, the expected values are computed assuming a simplification of independent binomial distributions of the tags in each slot, giving rise to inaccurate results. In [23], the authors introduce another MSE estimator procedure. They propose a *Q-slot* operation based on the error function in [19], but extended to several frames.

D. ML Estimators

The idea behind this group of estimators is to compute the conditional probability of an observed event (or set of events), and selecting as $\hat{n}_i$ (or $\hat{n}$ in case of MF estimators) the value that maximizes this probability. The main problems with these algorithms are the exact formulation of this conditional probability and their computational cost, which may render them unusable.

In [24], the authors proposed an ML algorithm derived from the occupancy problem described in [25]. When a frame i ends, the RFID reader computes the probability of finding e_i empty slots when the frame length is K_i and selects $\hat{n}_i$ as follows:

$$\hat{n}_i = \underset{n_i \geq s_i + 2c_i}{\operatorname{argmax}} \{P(K_i, e_i, n_i)\} \tag{1}$$

$$\Pr(K_i, e_i, n_i) = \frac{(-1)^{e_i} K_i!}{e_i! K_i^{n_i}} \sum_{j=e_i}^{K_i} \frac{(-1)^j (K_i - j)^{n_i}}{(j - e_i)!(K_i - j)!} \tag{2}$$

Note that this is an exact computation, unlike in Vogt's work [19], which assumed independent identically distributed (iid) binomial distributions of tags in each slot.

In [26], the authors presented an algorithm similar to the one in [24]. In addition, in [26], the authors remark that (2) is unfeasible for large values of K_i and n_i and propose the following heuristic estimator as an alternative:

$$\hat{n}_i = \frac{\log\left(\frac{e_i}{K_i}\right)}{\log\left(1 - \frac{1}{K_i}\right)}. \tag{3}$$

Nevertheless, this heuristic is erroneous when $e_i = 0$, because the term $\log(0)$ appears in the numerator.

The Slot-by-Slot (SbS) ML estimator in [6] uses the number of empty slots and identified tags. The authors propose this algorithm as a *Q-slot* mechanism. The probability formula in [6] can be reduced to

$$\begin{aligned}\Pr(K_i, e_i, s_i, n_i) = {} & \frac{K_i! n_i!}{e_i! s_i! K_i^{n_i}} \sum_{z=0}^{\min\{c_i, n_i - s_i\}} \frac{(-1)^z}{z!} \\ & \times \sum_{j=0}^{\min\{c_i - z_i, n_i - s_i\}} \frac{(K_i - 1)^j}{j!} \\ & \times \frac{(c_i - z - j)^{n_i - s_i - j}}{j!\,(c_i - z - j)!\,(n_i - s_i - j)!}\end{aligned} \tag{4}$$

However, the original formula (like the reduced one) is erroneous. It returns negative probabilities in some cases (e.g., $c_i = 1$, $s_i = 1$, $e_i = 1$, $K_i = 4$, for $n_i \geq s_i + 2 \cdot c_i$).

In [27], the author models the probability of event $\{s_i, e_i, c_i\}$ as a multinomial distribution problem (note that this is an approximation of the actual probability). The probabilities of empty, successful, and collision slots are denoted as p_0, p_1, and $p_{\geq 2}$. These probabilities are computed in [27] obtaining

$$\Pr(K_i, e_i, s_i, c_i) = \frac{K_i!}{e_i! s_i! c_i!} p_0^{e_i} p_1^{s_i} p_{\geq 2}^{c_i} \tag{5}$$

for $n_i \geq s_i + 2 \cdot c_i$.

Finally, the estimator in [28] uses statistical information from several frames (in fact, it is the only MF *Q-frame* schema so far that uses full information from each frame to estimate the number of contenders) to update the initial tag probability distribution according to expression (6)

$$\begin{aligned}&\Pr(n_i \,|\, \text{frames}_{1:i}) \\ &\qquad = \alpha \Pr(n_i \,|\, \text{frames}_{1:i-1}) \Pr(K_i, e_i, s_i, c_i \,|\, n)\end{aligned} \tag{6}$$

where α is a normalizing constant whose value is not defined. $\Pr(n_i \,|\, \text{frames}_{1:i})$ denotes the *a-posteriori* probability distribution of the number of contenders (n_i) after the ith frame, whereas $\Pr(n_i \,|\, \text{frames}_{1:i-1})$ denotes the *a-priori* distribution before that frame. The formulation of $\Pr(K_i, e_i, s_i, c_i \,|\, n)$ is given in [28]. In this framework, at the end of each frame i, the reader extracts $\hat{n}_i$ as the mode of the *a-posteriori* distribution. However, in the first iteration, since the *a priori* distribution is not available, the authors assume directly that the likelihood is the *a posteriori* distribution.

This last proposal has similarities to ours since it is also a MF full-information proposal. However, the computational cost of their estimator is higher, which notably increases the identification time of the algorithm (see Section VI).

IV. MFML-DFSA Algorithm

In this section, we describe our MFML-DFSA algorithm to compute the optimal frame length that maximizes the throughput. Let us call n the initial number of tags to identify. In our model, we assume that all tags remain in the identification area at least until their identifiers are correctly received, and that new tags do not enter the identification area during the reading process. The goal is to identify the n tags in the shortest time (equivalently, slots) possible. The identification process requires a series of consecutive reading frames $(i = 1, 2, \ldots)$ for all tags to be identified. At the end of frame i, the reader knows the number of identified tags (s_i), the number of slots with collisions (c_i) and the number of empty slots (e_i). Then, at the end of frame i, MFML-DFSA proceeds as follows.

- $\hat{n}$, the most likely number of tags at the beginning of the identification process, is computed by means of the ML estimator, as a function of the set $\{(K_j, s_j, c_j, e_j); j = 1, \ldots, \mathrm{i}\}$ (see Section IV-A).
- The most likely number of tags that will compete in the next frame $(i + 1)$ is $\widetilde{n_{i+1}} = \hat{n} - \sum_{j=1}^{i} s_j$ (the estimated total number of tags minus those already identified).
- Then, $K_{i+1} = 2^{Q_{i+1}}$ is accordingly selected to maximize the expected throughput at frame $i + 1$ (see Section IV-B).

A. $\hat{n}$ Computation

Let $P(N, K, s, c, e)$ be the probability of obtaining a sample of s slots filled with exactly one reply, c slots with a collision and e empty slots, if N tags compete for identification in an arbitrary frame of K slots. This probability is computed in Appendix A, yielding

$$\begin{aligned}&P(N, K, s, c, e) \\ &\quad = \frac{K!}{s!c!e!} \binom{K + N - 1}{N}^{-1} \binom{N - s - c - 1}{N - s - 2c} \\ &\quad = \frac{K!(K-1)!}{s!c!e!(c-1)!} \frac{\prod_{a=N-s-2c-1}^{N-s-c-1} a}{\prod_{b=N+1}^{N+K-1} b}.\end{aligned} \tag{7}$$

Equation (7) can be computed for $N \geq s + 2c$, since N is at least the sum of the tags identified plus the colliding ones (at least two per collision). Let us remark that the previous formula is exact, unlike the simplifications in previous work discussed in Section III.

After the first frame of the identification process, the probability of event $\{(K_1, s_1, c_1, e_1)\}$ if n tags contend is

$$\text{prob}\{(K_1, s_1, c_1, e_1)\} = P(n, 2^{Q_1}, s_1, c_1, e_1). \quad (8)$$

After the second frame

$$\begin{aligned}&\text{prob}\{(K_1, s_1, c_1, e_1), (K_2, s_2, c_2, e_2)\}\\ &\quad = P(n, 2^{Q_1}, s_1, c_1, e_1)\\ &\qquad \times P(n - s_1, 2^{Q_2}, s_2, c_2, e_2). \end{aligned} \quad (9)$$

Note that the identification frames are independent. Therefore, the probabilities of the observed events are independent as well. Then, after the i frames, if the initial number of tags is n, the probability of a given set of events $\{(K_j, s_j, c_j, e_j) : j = 1, \ldots, i\}$ is calculated as

$$\prod_{j=1}^{i} P\left(n - \sum_{u=1}^{j} s_{u-1}, 2^{Q_j}, s_j, c_j, e_j\right). \quad (10)$$

Let us remark that in frame j the number of tags that have not been identified yet is $n - \sum_{u=1}^{j} s_{u-1}$, being s_0 zero for consistency. Therefore, $\hat{n}$ is computed as the value that maximizes the probability given in (10), yielding its ML estimator

$$\hat{n} = \underset{n \geq \max_{j=1,\ldots,i} n_j}{\text{argmax}} \prod_{j=1}^{i} P\left(n - \sum_{u=1}^{j} s_{u-1}, 2^{Q_j}, s_j, c_j, e_j\right) \quad (11)$$

where $n_j = 2c_j + \sum_{u=1}^{j} s_u$, that is, the minimum number of tags known to have contended at frame j based on the number of slots with collisions and single responses.

B. Q Selection

The expected throughput of an FSA system is $S(N, K) = (N/K)(1 - (1/K))^{N-1}$, as shown in [13], and it reaches its maximum $S(N, K) = e^{-1}$ if $N = K$, for high N values (see also [13]). However, the number of slots per frame for EPC-C1G2 must be in $\{2^Q : Q = 0, \ldots, 15\}$. Therefore, in this case, the throughput is

$$S(N, K) = \frac{N}{K}\left(1 - \frac{1}{K}\right)^{N-1} = \frac{N}{2^Q}\left(1 - \frac{1}{2^Q}\right)^{N-1}. \quad (12)$$

For each value of Q there is a set of values of N for which Q attains maximum throughput. Fig. 4 illustrates this statement. These sets have the form $[N_{\min}(Q), \ldots, 2^Q, \ldots, N_{\max}(Q)]$, where $N_{\min}(Q)$ and $N_{\max}(Q)$ are the minimum and maximum number of tags for which Q provides the best throughput (see Fig. 4). Note that the sets are compact and always contain the point with maximal throughput ($N = K = 2^Q$), since it maximizes $S(N, K)$.

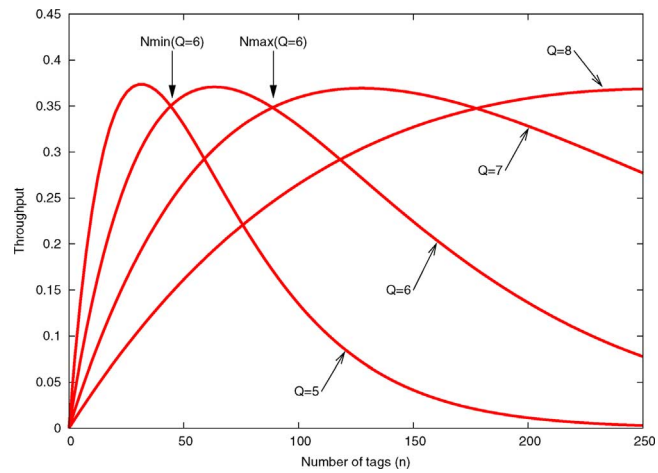

Fig. 4. Throughput versus N for different Q settings. Each curve shows the expected throughput for a given value of Q. Note how each value of Q provides the maximum throughput for a set of values of N. In the figure, the set of values of N between $N_{\min}(Q = 6)$ and $N_{\max}(Q = 6)$ achieve maximum throughput at $Q = 6$.

Our goal is to compute the borders of each set to determine where a given Q value maximizes throughput. Let us note that the maximum value of N for $Q - 1$ is one tag less than the minimum value for Q, i.e., $N_{\max}(Q - 1) = N_{\min}(Q) - 1$. Therefore, $N_{\max}(Q - 1)$ must be the largest integer fulfilling inequality (13)

$$\begin{aligned}&S(N_{\max}(Q-1), 2^{Q-1}) > S(N_{\min}(Q), 2^Q) \Rightarrow\\ &S(N_{\max}(Q-1), 2^{Q-1}) > S(N_{\max}(Q-1)+1, 2^Q).\end{aligned} \quad (13)$$

Hence

$$\begin{aligned}&\frac{N_{\max}(Q-1)}{2^{Q-1}}\left(1 - \frac{1}{2^{Q-1}}\right)^{N_{\max}(Q-1)-1}\\ &\qquad > \frac{N_{\max}(Q-1)+1}{2^Q}\left(1 - \frac{1}{2^Q}\right)^{N_{\max}(Q-1)}.\end{aligned} \quad (14)$$

In other words, given a value of Q, we find the highest value of N that makes the throughput using $Q - 1$ greater than the throughput using Q. Therefore, to calculate the optimal sets, we follow the next algorithm:

1) Set $Q = 1$ and $N = 1$. Obviously, for this particular case $N_{\min}(1) = 1$.
2) Do $N = N + 1$, and check if inequality (14) is fulfilled .
   a) If not, $N_{\min}(Q + 1) = N$ and $N_{\max}(Q) = N - 1$.
   b) Otherwise, repeat.
3) Do $Q = Q + 1$ and repeat until $Q = 16$ (computation can be accelerated[1]).

Table II summarizes the results for an arbitrary frame i. Note that selecting the nearest valid frame-length to $\widetilde{n}_i$ as K is a sub-optimal choice (e.g., if $\widetilde{n}_i = 90$, the nearest valid K is $2^6 = 64$; however, the optimal K is $2^7 = 128$).

In addition, note that there is no information available before the reading procedure starts, therefore, Q_1 must be selected from other criteria. Currently, commercial RFID readers select

[1]If after this step N is directly set to $2^Q + 1$, since $N_{\max}(Q) > 2^Q$ as stated previously

TABLE II
OPTIMAL Q_i VERSUS $\widetilde{n_i}$ RANGE

Q_i	$\widetilde{n_i}$ range
0	$\widetilde{n_i} = 1$
1	$1 < \widetilde{n_i} \leq 3$
2	$3 < \widetilde{n_i} \leq 6$
3	$6 < \widetilde{n_i} \leq 11$
4	$11 < \widetilde{n_i} \leq 22$
5	$22 < \widetilde{n_i} \leq 44$
6	$44 < \widetilde{n_i} \leq 89$
7	$89 < \widetilde{n_i} \leq 177$
8	$177 < \widetilde{n_i} \leq 355$
9	$355 < \widetilde{n_i} \leq 710$
10	$710 < \widetilde{n_i} \leq 1420$
11	$1420 < \widetilde{n_i} \leq 2839$
12	$2839 < \widetilde{n_i} \leq 5678$
13	$5678 < \widetilde{n_i} \leq 11357$
14	$11357 < \widetilde{n_i} \leq 22713$
15	$22713 < \widetilde{n_i}$

$Q_1 = 4$ by default, independently of the population of tags in their range. This work shows that better values of Q_1 can be selected if the number of contending tags in the first frame is known to lie within a certain interval (see Section VI).

V. EFFICIENT IMPLEMENTATION

Regarding algorithm implementation feasibility, the following iterative method is proposed. Note that maximizing the probability given in (10) is equivalent to maximizing its logarithm. Moreover, the logarithm of the product of probabilities in (10) can be expressed as the sum of their logarithms. To speed computations up, the RFID reader keeps an array with predefined computations of $\sum_{u=1}^{n} \log(u)$ for $n = 1, \ldots, 2^{15} + n_{\max}$, where $n_{\max}$ denotes the maximum number of competing tags and must be tailored to the scenario. Let l_n be the nth position in this array, and let us define an array p_n with $n_{\max}$ positions that are initially set to zero, and let $n_{\min} = 1$. Then, after frame i, it is necessary to:

1) Update $n_{\min}$, $n_{\min} = \max\{n_{\min}, 2c_i + \sum_{j=1}^{i} s_j\}$.
2) Compute the logarithm of the last term in product (10) for $n = n_{\min}, \ldots, n_{\max}$. That is, compute

$$\log\left(P\left(n - \sum_{u=1}^{i} s_{u-1}, 2^{Q_i}, s_i, c_i, e_i\right)\right). \quad (15)$$

Note—see (7)—that this is the sum of a constant

$$\log\left(\frac{2^{Q_i}!(2^{Q_i}-1)!}{s_i!c_i!e_i!(c_i-1)!}\right) = l_{2^{Q_i}} + l_{2^{Q_i}-1} - l_{s_i} - l_{c_i} - l_{e_i} - l_{c_i-1}, \quad (16)$$

plus a factor that varies with n' (being $n' = n - \sum_{u=1}^{i} s_{u-1}$)

$$\log\left(\frac{\prod_{a=n'-s_i-2c_i-1}^{n'-s_i-c_i-1} a}{\prod_{b=n'+1}^{n'+2^{Q_i}-1} b}\right) = l_{n'-s_i-c_i-1} - l_{n'-s_i-2c_i-2} - l_{n'+2^{Q_i}-1} + l_{n'}. \quad (17)$$

Therefore, this step requires at most $5 + 4n_{\max}$ sums.

TABLE III
TYPICAL VALUES OF EPCGLOBAL CLASS-1 GEN-2 PARAMETERS

Parameter	Symbol	value
Electronic Product Code	EPC	96 bits
Initial Q value	Q_1	4
Reference time interval for a data-0 in teader-to-tag signalling	TARI	$12.5us$
Time interval for a data-0 in reader-to-tag signalling	DATA0	1.0 TARI
Time interval for a data-1 in reader-to-tag signalling	DATA1	1.5 TARI
Tag-to-reader calibration symbol	TRcal	64 μs
Reader-to-tag calibration symbol	RTcal	31.25 μs
Divide ratio	DR	8
Backscatter link frequency	LF	DR/Trcal
Number of subcarrier cycles per symbol in tag-to-reader direction	M	1,2,4,8
Reader-to-tag rate	Rtrate	64 Kbps
Tag-to-reader rate	Trrate	LF/M
Link pulse repetition interval	T_{pri}	1/LF
Tag-to-reader preamble	TRP	$6T_{pri}$
Tag-to-reader end of signalling	$T \rightarrow R$ EoS	$2T_{pri}$
Delimiter	Del	12.5 μs
Reader-to-tag preamble	RTP	Del + DATA0+ TRcal + RTcal
Reader-to-tag frame synchronization	RTF	RTP - RTcal
Time since reader transmission to tag response	T_1	max(RTcal,10 T_{pri})
Time since tag response to reader transmission	T_2	$5T_{pri}$
Time a reader waits after T_1 before it issues another command	T_3	$5T_{pri}$
Minimum time between reader commands	T_4	2RTcal
Query packet	Query	22 bits
QueryAdjust packet	QueryAdjust	9 bits
QueryRep packet	QueryRep	4 bits
Ack packet	Ack	18 bits
Nack packet	Nack	8 bits

3) Then, the sum of logarithmic probabilities is updated, $p_n = p_n + \log(P(n - \sum_{u=1}^{i} s_{u-1}, 2^{Q_i}, s_i, c_i, e_i))$. The estimation of the initial number of contenders corresponds to

$$\hat{n} = \underset{n \in [n_{\min}, n_{\max}]}{\operatorname{argmax}} \; p_n. \quad (18)$$

Therefore, $\widetilde{n_{i+1}}$ is computed as $\hat{n} - \sum_{j=1}^{i} s_j$. This step requires $n_{\max}$ sums and comparisons.
4) Finally, the best value of Q_{i+1} is selected from Table II as a function of $\widetilde{n_{i+1}}$.

VI. PERFORMANCE EVALUATION AND BENCHMARKING

The performance of MFML-DFSA and the main DFSA alternatives (see Section III-D) has been evaluated by means of a discrete-event simulator, developed in C++ within the OMNeT++ framework [29]. We have considered a scenario with a single passive reader and a set of tags that enter the reader coverage area and do not leave it until all the tags are successfully identified. The simulator computed the total number of slots required to identify the whole tag population. This experiment was repeated (note that each run was independent) until a confidence interval for the mean value was achieved with 95% confidence degree. The physical configuration parameters of the commercial UHF (868 MHz) Alien 8800 reader [30] (see Table III) were used. The simulator had been validated previously by means of laboratory test beds based on that reader [31]. Moreover, the

TABLE IV
COMPARISON OF COMPUTATIONAL COST

		Scenario 1		Scenario 2	
Protocol	**Order**	**FLOP/frame**	**Reading time increase (%)**	**FLOP/frame**	**Reading time increase (%)**
SbS [6]	$O(n)$	$1.3 \cdot 10^7$	$\approx 3\%$	$3.7 \cdot 10^8$	$\approx 24\%$
Chen-3 [27]	$O(n)$	$7 \cdot 10^5$	$< 1\%$	$7 \cdot 10^6$	$< 1\%$
Floerker [28]	$O(n^2)$	$4.4 \cdot 10^7$	$\approx 11\%$	$3.6 \cdot 10^9$	$> 100\%$
MFML-DFSA	$O(n)$	10^3	$< 0.1\%$	$6 \cdot 10^5$	$< 0.1\%$

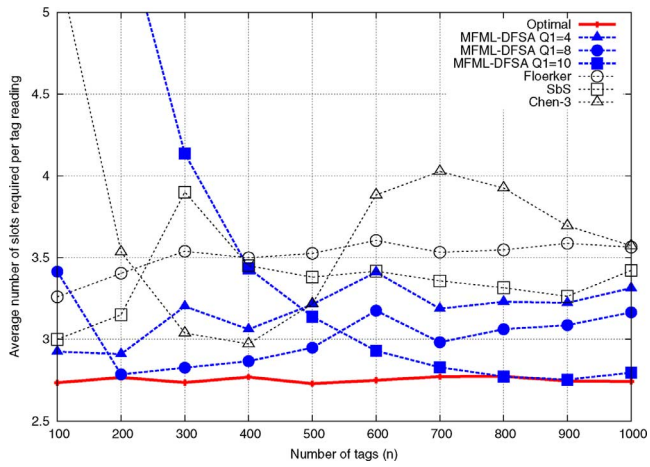

Fig. 5. Average number of slots required per tag reading versus n.

optimal Q selection criterion (Table II) was used in all cases to focus the analysis on the performance of the estimation schema.

Fig. 5 shows the performance of MFML-DFSA starting with $Q_1 = 4, 8, 10$, compared to the best DFSA algorithms ([6], [27], [28]) studied in [14]. These algorithms start with $Q_1 = 4$. The optimal Q_i from Table II is selected from the expected $\widetilde{n}_i$ in all cases. As a reference, we also depict the performance of an optimal DFSA algorithm, i.e., one with perfect knowledge of the competing tags in each frame. Our proposal outperforms previous ones in all cases when $Q_1 = 8$ if $N > 100$. Regardless of the initial Q_1 value, MFML-DFSA always performs better when $n > 500$. If $Q_1 = 4$, MFML-DFSA outperforms other ML estimators for a broad range of n values (if $n < 300$ or $n > 400$). Indeed, it can be observed that MFML-DFSA approaches the optimal bound in a range of values of n (e.g., for $700 < n < 1000$ if $Q_1 = 10$), illustrating that MFML-DFSA effectively exploits a rough knowledge about the initial number of tags (i.e., a relatively wide range).

The identification time results in Fig. 5 do not include the computation time for estimating the number of tags, selecting Q, etc. Note that these decisions must be taken at the end of each frame, before a new $Query$ packet is sent. The reader can delay the transmission of those packets (since the tags do not transmit until a packet is received). However, this may increase overall identification time. In order to provide some insight into performance in real situations, we have selected two examples where the computational cost has been calculated. The examples are as follows.

- Scenario 1. At the end of the ith frame (with $K_i = 128$), the reader has collected the following statistical information: $e_i = 48$, $c_i = 30$, and $s_i = 22$. Note that, with this configuration, $n \geq 2c_i + s_i = 82$. The algorithm iterates for every possible value of n up to a given $n_{\max}$ and the optimum is extracted. In this scenario, we assume $n_{\max} = 100$. Then, the worst case requires $(100 - 82) = 18$ iterations.
- Scenario 2. At the end of the ith frame ($K_i = 512$), the reader has collected the following statistical information: $e_i = 170$, $c_i = 192$, and $s_i = 170$. In this case, we assume $n_{\max} = 1000$. Then, $(1000 - 554) = 446$ iterations are necessary.

The following assumptions have been considered to compute the cost.

- A computational power of 1 GFLOPS (10^9 FLOating Point Operations per Second) is considered, which is representative of the performance of an average Digital Signal Processor (DSP). DSPs are common in RFID hardware such as the Alien 8800 [30].
- The MF algorithm [28] and our proposal do not require recomputing the sum from the first iteration to the last one. Instead, they only update previous computations to take into account the last frame.
- Finally, we have assumed a computational cost of 50 FLOP for power, logarithm, and exponential operations and 100 FLOP for factorial operations.

Table IV summarizes the approximate number of computations required per frame in both scenarios. Additionally, the time required to perform these computations is expressed relative to frame length, as a percentage. Despite being $O(n)$, the approach in [6] has a high computational cost, which increases total identification time. Indeed, the approach in [28] has an unacceptably high computational cost in some cases, which prevents real implementation in its current algorithmic form. MFML-DFSA has a low computational cost in both scenarios. Therefore, we can claim that the entire MFML-DFSA process is not computationally demanding and, for current commercial CPUs, extremely large ranges can be analyzed in $\sim$1 μs (note that the typical slot duration is 3 ms).

VII. CONCLUSION

The strengths and weaknesses of the DFSA algorithms studied in [14] have been reviewed. Departing from this review, we propose a new feasible MFML-DFSA algorithm that employs statistical information from several previous frames and uses a ML estimator to compute the expected number of competing tags. The results show that MFML-DFSA outperforms the current DFSA proposals, achieving better identification time for a low computational cost. Its implementation is computationally feasible and it does not require any modification to tag operation, therefore satisfying the EPC-C1G2 standard. MFML-DFSA can be directly implemented in current RFID readers allowing seamless adoption by RFID vendors. As future work, we will study the applicability of MFML-DFSA in

dense reader environments, where schedulers are mandatory to coordinate the readers [32], [33]. These schedulers may benefit from improved estimators such as MFML-DFSA.

APPENDIX A
COMPUTATION OF $P(N, K, s, c, e)$

To compute the probability $P(N, K, s, c, e)$, henceforth $P(N, K)$, we apply the technique in [34], where the authors formulate probabilistic transforms for urn models that convert the dependent random variables describing urn occupancies (slot occupancies in our case) into independent random variables. Due to the independence of random variables in the transform domain, it is simpler to compute the statistics of interest, and get the desired result afterwards by inverting the transform.

Let $P(N, K)$ be the probability of interest and $\boldsymbol{P}(\lambda, K)$ its transformation, where λ is a parameter that is only meaningful in the transform domain. Note that there is no dependence on the number of balls (tags), N, in the transform domain.

The procedure is as follows: first, the appropriate transform for a particular urn model is selected. In our case, the K urns (slots) are distinguishable and the N balls (tags) are indistinguishable, since we are only interested in the number of balls within each urn. In this case, the independent random variables $Z_1, \ldots, Z_K$ describing the occupancy of an urn in the transform domain are geometrically distributed with mean λ [34]. That is, $P(Z_i = n) = (1-\lambda)\lambda^n$. Second, the probability of interest $\boldsymbol{P}(\lambda, K)$ is computed in the transform domain. In our case, given a frame of length K, the probability of having s urns with one ball, c urns with several balls, and e empty urns is

$$\begin{aligned}\boldsymbol{P}(\lambda, K) &= \frac{K!}{s!c!e!} P(Z=1)^s P \\ &\times (Z>1)^c P(Z=0)^e \\ &= \frac{K!}{s!c!e!}((1-\lambda)\lambda)^s \lambda^{2c}(1-\lambda)^e. \end{aligned} \tag{19}$$

Finally, the inverse transform is computed as

$$\begin{aligned}P(N, K) &= \binom{K+N-1}{N}^{-1} \\ &\times [\lambda^N]\{\boldsymbol{P}(\lambda, K)/(1-\lambda)^N\}\end{aligned} \tag{20}$$

with $[\lambda^N]\{h(\lambda)\}$ denoting the coefficient of λ^N in the power series $\{h(\lambda)\}$. In our case, rewriting (19) as a power series in λ

$$\begin{aligned}P(N, K) &= \frac{K!}{s!c!e!}\binom{K+N-1}{N}^{-1} \\ &\times [\lambda^N]\left\{\sum_{l=0}^{\infty}\binom{K-s-e+l-1}{l}\lambda^{l+s+2c}\right\}\end{aligned} \tag{21}$$

and extracting the coefficient of λ^N for the appropriate N value $N = l + s + 2c$, we obtain the result in (7).

ACKNOWLEDGMENT

The authors are indebted to the anonymous referees that revised this work.

REFERENCES

[1] K. Finkenzeller, *RFID Handbook: Radio-Frequency Identification, Fundamentals and Applications*. New York: Wiley, 2000.

[2] F. Gandino, B. Montrucchio, M. Rebaudengo, and E. Sánchez, "On improving automation by integrating RFID in the traceability management of the agri-food sector," *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2357–2365, Jul. 2009.

[3] G. M. Gaukler, "Item-level RFID in a retail supply chain with stock-out-based substitution," *IEEE Trans. Ind. Informat.*, in press.

[4] S. Saab and S. Nakad, "A standalone RFID indoor-positioning system using passive tags," *IEEE Trans. Ind. Electron.*, in press.

[5] F. Dai, Y. Shi, J. Wu, and Y. Yao, "A fully integrated 900 MHz passive RFID transponder front-end with novel zero-threshold RF-DC rectifier," *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2317–2325, Jul. 2009.

[6] B. Knerr, M. Holzer, C. Angerer, and M. Rupp, "Slot-wise maximum likelihood estimation of the tag population size in FSA protocols," *IEEE Trans. Commun.*, vol. 58, no. 2, pp. 578–585, Feb. 2010.

[7] M. Henseler, M. Rossberg, and G. Schaefer, "Credential management for automatic identification solutions in supply chain management," *IEEE Trans. Ind. Informat.*, vol. 4, no. 4, pp. 303–314, Nov. 2008.

[8] S. Chalasani and R. Boppana, "Data architectures for RFID transactions," *IEEE Trans. Ind. Informat.*, vol. 3, no. 3, pp. 246–257, Aug. 2007.

[9] Y.-H. Chen, S.-J. Horng, R.-S. Run, J.-L. Lai, R.-J. Chen, W.-C. Chen, Y. Pan, and T. Takao, "A novel anti-collision algorithm in RFID systems for identifying passive tags," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 105–121, Feb. 2010.

[10] *Information Technology—Radio Frequency Identification for Item Management—Part 6: Parameters for Air Interface Communications at 860 MHz to 960 MHz*, ISO/IEC 18000-6C, 2004.

[11] *Information Technology—Radio Frequency Identification for Item Management—Part 7: Parameters for Active Air Interface Communications at 433 MHz*, ISO/IEC 18000-7:2008.

[12] EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz 960 MHz EPC Global, ver. 1.2.0, Oct. 2008.

[13] N. Abramson, "Packet switching with satellites," in *Proc. Nat. Comput. Conf.*, Jun. 1973, pp. 695–702.

[14] M. V. B. Delgado, J. V. Alonso, and F. J. González-Castaño, "Analysis of DFSA anti-collision protocols in passive RFID environments," in *Proc. 35th Int. Conf. IEEE Ind. Electron. Soc.*, Porto, Portugal, Nov. 2009, pp. 2630–2637.

[15] D. Lee, K. Kim, and W. Lee, "Q^{+-} Algorithm: An enhanced RFID tag collision arbitration algorithm," in *Proc. Ubiquitous Intell. Comput.*, Hong Kong, China, Jul. 2007, vol. 4611, Lecture Notes in Computer Science, pp. 23–32.

[16] I. Joe and J. Lee, "A novel anti-collision algorithm with optimal frame size for RFID system," in *Proc. 5th ACIS Int. Conf. Softw. Eng. Res., Manage. Appl.*, Aug. 2007, pp. 424–428.

[17] M. Daneshmand, C. Wang, and K. Sohraby, "A new-count selection algorithm for RFID protocol," in *Proc. 2nd Int. Conf. Commun. Networking in China*, Shanghai, China, Aug. 2007, pp. 926–930.

[18] F. C. Schoute, "Dynamic frame length ALOHA," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 565–568, Apr. 1983.

[19] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. Int. Conf. Pervasive Comput.*, Zurich, Switzerland, Aug. 2002, vol. 2414, Lecture Notes in Computer Science, pp. 98–113.

[20] B.-S. Wang, Q.-S. Zhang, D.-K. Yang, and J.-S. Di, "Transmission control solutions using interval estimation method for EPC C1G2 RFID tag identification," in *Proc. Int. Conf. Wireless Commun., Networking and Mobile Comput.*, New York, Oct. 2007, pp. 2105–2108.

[21] J.-R. Cha and J.-H. Kim, "Dynamic frame slotted aloha algorithms using fast tag estimation method for RFID system," in *Proc. IEEE Consumer Commun. Networking Conf.*, Las Vegas, NV, Jan. 2006, pp. 768–772.

[22] W.-T. Chen, "An efficient anti-collision method for tag identification in a RFID system," *IEICE J. Trans. Commun.*, vol. E89-B, no. 12, pp. 3386–3392, 2006.

[23] B. Knerr, M. Holzer, C. Angerer, and M. Rupp, "Slot-by-slot minimum squared error estimator for tags populations in FSA protocols," in *Proc. 2nd Int. EURASIP Workshop on RFID*, Budapest, Hungary, Jul. 2008, pp. 1–13.

[24] W.-T. Chen and L. Guan-Hung, "An efficient scheme for multiple access in a RFID system," in *Proc. Int. Conf. Wireless Networks*, Las Vegas, NV, Jun. 2006, pp. 160–163.

[25] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. New York: Wiley, 1970, vol. 1.

[26] G. Khandelwal, L. Kyounghwan, A. Yener, and S. Serbetli, "ASAP: A MAC protocol for dense and time constrained RFID systems," *EURASIP J. Wireless Commun. Networking*, vol. 9, pp. 4028–4033, 2007.

[27] W.-T. Chen, "An accurate tag estimated method for improving the performance of an RFID anti-collision algorithm based on dynamic frame length ALOHA," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 9–15, Mar. 2008.

[28] C. Floerkemeier, "Transmission control scheme for fast RFID object identification," in *Proc. 4rth IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOMW'06)*, Pisa, Italy, Mar. 2006, pp. 457–462.

[29] [Online]. Available: http://www.omnetpp.orgOMNeT++ Network Simulation Framework Online at

[30] [Online]. Available: http://www.alientechnology.com/readers/index.phpAlien Reader 8800, development evaluation kit. Online at

[31] M. V. Bueno-Delgado and J. Vales-Alonso, "On the optimal frame-length configuration on real passive RFID systems," *J. Network and Comput. Appl.*, in press.

[32] J.-B. Eom, S.-B. Yim, and T.-J. Lee, "An efficient reader anticollision algorithm in dense reader networks with mobile RFID readers," *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2326–2336, Jul. 2009.

[33] D.-Y. Kim, H.-G. Yoon, B.-J. Jang, and J.-G. Yook, "Effects of reader-to-reader interference on the UHF RFID interrogation range," *IEEE Trans. Ind. Informat.*, vol. 56, no. 7, pp. 2337–2346, Jul. 2009.

[34] O. Milenkovic and K. J. Compton, "Probabilistic transforms for combinatorial urn models," *Combinatorics, Probability and Computing*, vol. 13, no. 4–5, pp. 645–675, 2004.

Javier Vales-Alonso (M'07) received the M.Sc. degree from the University of Vigo, Vigo, Spain, in 2000 and the Ph.D. degree from the Technical University of Cartagena, Cartagena, Spain, in 2005.

Since 2002, he has been with Department of Information Technologies and Communications at Technical University of Cartagena. He is also involved with several Spanish national R&D projects related to development of ambient intelligence applications. His main research topics lie in wireless communications areas, mainly in WSN, VANET, and RFID fields.

Victoria Bueno-Delgado (M'10) received the Telematics Engineering degree, the Telecommunications Engineering degree, and the European Ph.D. degree in telecommunications from the Technical University of Cartagena, Cartagena, Spain, in 2002, 2004, and 2010, respectively.

Since 2004, she has been a Researcher at the Department of Information Technologies and Communications, Technical University of Cartagena. Since 2006, she has been an Assistant Professor at the University of Cartagena. She has published several journal and conference papers in the area of wireless communications, addressing topics like performance improvement of anticollision protocols. Her research interests include communication protocols and deployment techniques in radio frequency identification systems and wireless sensor networks.

Esteban Egea-Lopez received the Telecommunications Engineering degree from the Technical University of Valencia (UPV), Valencia, Spain, in 2000, the M.S. degree in electronics from the University of Gavle, Gavle, Sweden, in 2001, and the Ph.D. degree in telecommunications from the Technical University of Cartagena, Cartagena, Spain, in 2006.

Since 2001, he has been an Assistant Professor at the Department of Information Technologies and Communications, Polytechnic University of Cartagena. His research interest is focused on RFID, vehicular, ad hoc and wireless sensor networks.

Francisco J. Gonzalez-Castaño received the Ph.D. degree from the University of Vigo, Vigo, Spain, in 1998.

He is a Full Professor with the Department of Telematics Engineering, University of Vigo. He is also with Gradiant, Spain, as the Research Director in Networks and Applications. He leads the Information Technologies Group, University of Vigo, Spain (http://www-gti.det.uvigo.es). He holds three Spanish patents, a European patent, and a U.S. patent. He has published over 50 papers in international journals, in the fields of telecommunications and computer science, and he has participated in several relevant national and international projects.

Juan Alcaraz received the Engineering degree from the Technical University of Valencia, Valencia, Spain, in 1999 and the Ph.D. degree from the Technical University of Cartagena, Cartagena, Spain, in 2007.

After working for several telecommunication companies, he joined the Technical University of Cartagena, Cartagena, Spain, in 2004, where he currently works as an Associate Professor. He has published several journal papers in the area of wireless communications, addressing topics like vehicular networks, and radio frequency identification systems.