Analysis of DFSA Anti-collision Protocols in passive RFID environments

M.V. Bueno-Delgado[1], J. Vales-Alonso[1], F.J. Gonzalez-Castaño[2]
(1) Communications and Technology Information Department, Polytechnic University of Cartagena, Spain.
(2) Telematics Engineering Department, University of Vigo, Spain.
E-mail: {mvictoria.bueno, javier.vales}@upct.es, javier@det.uvigo.es

***Abstract*—Frame Slotted Aloha (FSA) protocols are promising anti-collision protocols for passive RFID systems. They aim at decreasing the time to detect all the tags in range (identification delay). In FSA, the maximum identification rate (average number of tags identified per slot) is achieved when the number of contending tags matches the cycle length (number of slots in a frame). Therefore, the reader should ideally know the actual number of competing tags. However, in RFID scenarios this figure varies randomly, and the reader has to *guess* the number of contenders somehow. This paper analyzes the most relevant anti-collision algorithms; taking into account the limitations imposed by the world-wide *de-facto* standard EPCglobal Class-1 Gen-2 for passive RFID systems.**

***Index Terms*— Anti-collision algorithms, DFSA, EPCglobal Class-1 Gen-2, RFID.**

I. INTRODUCTION

Frame Slotted Aloha (FSA) protocols are promising anti-collision protocols for passive RFID systems [1]. Time is subdivided into frames (also called cycles), which are respectively subdivided into slots. In the identification process, the reader announces the length of the upcoming cycle (number of slots in the next frame) and every unidentified tag in range selects (at random) a slot to transmit its identifier. In conventional (static) FSA, the frame length is the same for every cycle. However, this is clearly inefficient: if the tags outnumber the slots, collisions will be frequent. Otherwise, there will be many empty slots. Both situations are undesirable. The optimal criterion is *maximizing the throughput*, that is, the average number of identified tags per slot. This way, the overall identification time of a population of tags is minimized as well. The best FSA performance results when there are as many competing tags as slots in the frame, yielding a maximum throughput of $e^{-1} \approx 0.36$ [1].

Therefore, the reader *must adjust the frame length in each cycle* according to the number of competing tags. Since FSA protocols always work with a static frame length, Dynamic FSA (DFSA) protocols have been suggested [1]. In DFSA, the reader can modify the frame length at each cycle, or restart it, following some criterion. However, the number of tags that enters into the coverage area usually varies randomly, thus the number of tags that compete in each cycle is unknown, and frame length adjustment is not trivial. The reader has to *guess* the number of nodes in competition by means of some estimation procedure. Until now, several works have addressed this issue.

In this paper we describe and analyze the most relevant dynamic-adjustment frame length anti-collision proposals. We point out some faulty formulae which may lead to incorrect algorithm behavior. Indeed, we have conducted an independent analysis of the protocols, assuming that a standard EPCglobal Class-1 Gen-2 RFID system [2] adopts them. The main consequence is that the frame length must be a power of 2, i.e. not any natural number. We compute the optimal frame length as a function of the number of competing tags, and we employ that value to configure the algorithms.

The rest of this paper is organized as follows: section II describes related work, section III provides an overview of the EPC Global Class-1 Gen-2 standard. Section IV introduces a classification of DFSA protocols. Section V describes the most important DFSA proposals. Section VI analyzes them. Finally, section VII concludes the paper.

II. RELATED WORK

Anti-collision algorithms for passive RFID systems are commonly divided into two groups: tree-based and Aloha-based probabilistic algorithms. The former group organizes the set of tag identities as a binary searching tree. Although these algorithms are simple and their computational cost is low, identification delay is unacceptable for large identification spaces. Therefore, these protocols are only attractive for specific applications, *e.g.* access control systems. However, they have been widely studied [3-6]. On the other hand, Aloha based probabilistic algorithms are categorized into *id-slot* and *bit-slot* ones. In the former, the tags send their complete identification, whereas in the latter they only transmit a segment. The current standards are based on *id-slot* algorithms and all the proposals analyzed in this paper belong to this class. In this context, there are several papers on the most relevant probabilistic algorithms (most of them based on DFSA), also comparing their performance versus deterministic algorithms. For example, in [7] the authors compare tree-based algorithms with one bit-slot and two id-slot algorithms. However, these proposals are currently of minor interest since they are not EPCglobal compliant. In [8], only the average identification time and optimal frame size of probabilistic protocols are studied. But, again, as in [7], the protocols do not fulfill the current standard. The work in [9] surveys anti-collision protocols and multiple reader coexistence. However, only a table comparing them is shown, and the authors do not impose current standard constraints. In [10] the authors simulate the throughput and estimation error of some DFSA protocols. However, they do not establish a clear classification, nor they adopt EPCglobal constraints.

978-1-4244-4649-0/09/$25.00 ©2009 IEEE

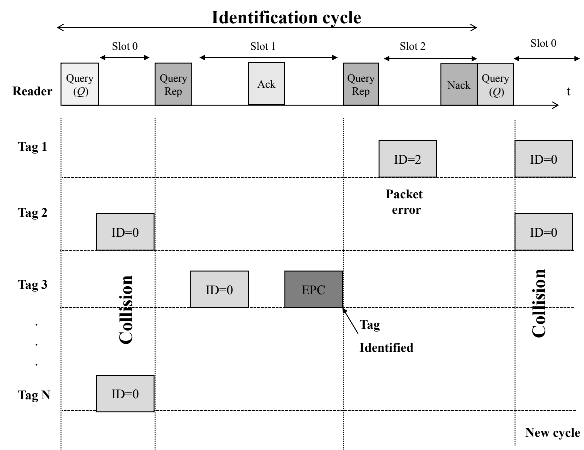

Figure 1. Identification cycle with EPCglobal Class-1 Gen-2

III. Overview of EPCglobal Class-1 Gen-2 Standard

This paper focuses on DFSA algorithms for the worldwide standard EPCglobal Class-1 Gen-2 RFID UHF (860–960 MHz) [2]. In the resting-state (no ongoing identification process), the reader monitors the environment to detect new tags, using continuous *Broadcast* packet transmissions. Tags in range reply immediately, and in case several tags answer simultaneously, a collision takes place. When the reader detects the collision it starts a new identification cycle, *i.e.*, it allocates a new frame which is in turn subdivided into slots, following the FSA scheme (see Fig. 1). An identification cycle starts when the reader transmits a *Query* packet, which includes a field of four bits with value $Q \in [0,\ldots,15]$, stating that the incoming frame length is 2^Q slots. Tags receive this packet and generate a random number r in the interval $[0, 2^Q-1]$. The r value represents the slot within the frame where each tag has randomly decided to send its identifier. In the frame, the beginning of each slot is controlled by the reader via transmission of a *QueryRep* packet, except in the first slot which starts immediately after the *Query* packet. Tags use the r value as a counter, which is decreased upon reception of a new *QueryRep* packet. When the counter r reaches 0, the tag transmits an identifier (ID), which corresponds to the random value r initially calculated for contention. Note that it must be also equal to the slot number in the frame. After transmitting the ID, three possibilities may arise:

(i) If several tags select the same slot, a collision occurs. The reader detects it and reacts starting a new slot with a *QueryRep* packet (see slot 0 in figure 1). Involved tags update their counter to $r=2^Q-1$. That means that they will not contend again until the next frame.

(ii) If the reader receives an *ID* correctly, which matches the current slot number, the reader responds with an *Ack* packet. Although all tags receive the packet, only the winner answers with a *Data* packet (including the tag EPC code [2]). If the reader receives the *Data* packet correctly, it answers with a *QueryRep* packet, thus starting a new slot. Besides, the winner tag quits the identification process (slot 1 in figure 1). However, if the reader does not receive a correct *Data* packet within a given time interval, 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 cycle (slot 2 in figure 1). After this, the reader also sends a new *QueryRep* packet to begin the new slot.

(iii) If the reader does not receive any packet before a given deadline, it is assumed that the slot is empty, and the reader starts a new one sending a new *QueryRep* packet.

This procedure continues slot-by-slot until the identification cycle finishes. Then, the reader sends a new *Query* packet to start a new cycle. Unidentified tags compete again in the new cycle, choosing a new random r value. Eventually, all tags are identified and the procedure ends. This happens when all the slots are empty in a frame. Notice also that slots duration clearly depends on the events that take place in each slot. However, the whole identification time in seconds is directly proportional to the slots required, and comparisons among protocols can be directly done in terms of logical units (slots).

The aforementioned EPCglobal (*static*) collision avoidance performs poorly due to fixed Q values. However, any DFSA strategy can be efficiently adapted to EPCglobal Class-1 Gen-2. Since the reader handles the extra protocol complexity, the tags can use standard control packets (*Query, QueryRep*, etc.). The integration of dynamic procedures is seamless.

IV. A Classification For DFSA Protocols

Establishing a clear classification of all DFSA protocols is not straightforward. We can consider different perspectives:

A. How is the protocol operation modified?

Frame structure can be adjusted in two ways: (i) controlling the number of slots in the frame with Q, *i.e.*, customizing *Query* packet or (ii) resetting the identification cycle at any time, by sending a new *Query* packet with the same or a new Q value. Throughout this paper, we denote them as *Q-control* and *Q-reset* operation, respectively.

B. Which data is used to compute length or reset the frame?

DFSA readers can monitor different variables to take operative actions. In a given cycle, it is possible to extract three variables: the number of slots filled with exactly one transmission (henceforth, *id*), the number of empty slots (e), and the number of slots with collision (c). Let us denote k as the frame length. Notice that it is possible to relate the previous variables since $k = 2^Q = id + e + c$. Therefore, two variables in the set $\{id, e, c\}$ give full information about a cycle. Indeed, information of several cycles may also be used, in this case we will label the variable corresponding to the i-*th* cycle with an i subscript (k_i, Q_i, id_i, e_i, c_i).

To summarize, we endorse a classification according to the monitored set: SF-SP, or Single-Frame, Single-Parameter (*id*, e, or c); SF-FP, or Single-Frame, Full-Parameter (two in $\{id, e, c\}$); MF-SP, or Multi-Frame, Single-Parameter; and MF-FP, or Multi-Frame, Full-Parameter.

Clearly, the performance of the algorithm will be directly related to the cardinality of this set and the quality of the information. Single-frame makes sense in case of continuous tag flow, since frame information quality degrades with age.

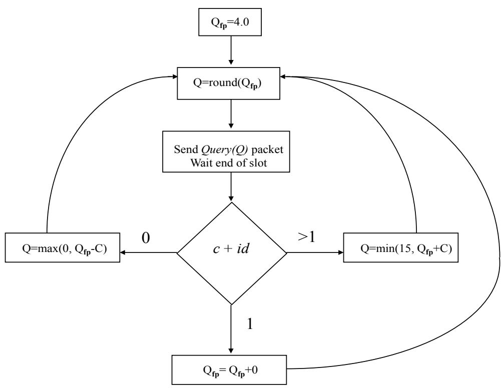

Figure 2. Dynamic-EPCglobal Class-1 Gen-2

Instead, multi-frame is advisable if new tag populations do not appear until the previous set has been completely identified. The reader may also enforce this behavior. For instance, if a conveyor belt carries the tags, the reader can control it.

C. How is frame length selected?

Every time a new *Query* packet is sent, the DFSA protocol computes its *Q* parameter. Depending on how this value is selected we classify protocol operation into two categories: (i) Direct and (ii) Indirect *Q* selection. In the former, some heuristic returns the value of *Q*. Many of these heuristics also decide if the current frame must be reset or not, *i.e.* they operate using *Q-reset*. In the latter, the algorithm first estimates the number of tags that competed in the previous cycle ($\hat{n}$). The expected number of competitors in the next cycle will be ($\hat{n}$-*id*). Then, *Q* is selected as a function of $\hat{n}$. There are different ways of calculating $\hat{n}$: heuristically, from Bayesian inference, from maximum likelihood (ML) estimation, etc.

In the Bayesian inference case, the goal is to compute the *a-posteriori* distribution of the number of competing tags. In fact, this distribution can be upgraded cycle-by-cycle to obtain a more precise estimation (for multi-frame information sets). *Q* can be selected as the value that maximizes the expected identification rate for the next cycle, or even as the value that minimizes some loss function defined over the *n* domain.

In the maximum likelihood (ML) estimation, the probability of observing a given sample set is computed as a function of *n*. Let $\hat{n}$ achieve its maximum. The *Q* value that maximizes the expected identification rate is also selected.

V. ANALYSIS OF DFSA PROTOCOLS

Table 1 shows a classification of the main DFSA proposals, according to the classification in the previous section. We consider four major groups of protocols, each one stemming from a different root or operative foundation: *dynamic EPC protocols*, *indirect Q heuristics*, *error minimization estimators* and *maximum likelihood estimators.*

A. Dynamic EPC protocols family

EPCglobal Class-1 Gen-2 [2] proposes the *dynamic* EPC mechanism as an alternative to the *static* scheme in section III. It adjusts the frame-length slot-by-slot (*Q-reset* operation), following the heuristic shown in figure 2. When the slot ends the reader checks if it was empty, successful or with collision. Accordingly, the reader increases, decreases or keeps floating point variable Q_{fp} unmodified. Then, the nearest integer to Q_{fp} becomes *Q*, and a new *Query* packet with that value is sent. Variable $C \in (0.1, 0.5)$, which can be updated to improve performance, controls the *dynamic* EPC mechanism slot-by-slot. The standard does not specify the selection of *C.* It only recommends using high values if the previous *Q* value was low and vice versa. This lack of definition has lead to many different alternatives.

In [11] the authors propose the Q^{+-} *algorithm*. Since the *Query* packet length (22 bits) exceeds the *QueryRep* packet length (4 bits), *Query* packets are only sent if the calculated *Q* value differs from the previous one. Otherwise, a *QueryRep* packet is transmitted. Besides, the *C* variable is replaced by two new variables: C_i and C_c.

The former is used when the reader detects an empty slot. The latter is applied when the reader detects a collision. The authors propose computational methods to obtain them regardless if the number of competing tags is known (or it can be estimated), or not. However, they neither propose a way to determine the number of tags, nor suggest how to estimate it. Indeed, the authors only compare Q^{+-} *algorithm* with the static EPC algorithm, instead of the dynamic one. Thus, their results are not conclusive.

Reference [12] describes the *optimum-C* protocol, which calculates the optimum *C* versus the previous *Q* value. The authors simulate a passive RFID system for $Q \in [0,\ldots,15]$, and, for each *Q,* for $C \in [0.1,\ldots,0.5]$, *only* in 0.1 steps. Finally, they give the value that achieves a best identification delay. As in [11], the *optimum-C* protocol is compared with *static* EPC and the protocol in [13], instead of *dynamic* EPC.

The *Slot-Count-Selection (SCS)* algorithm in [14] improves the *dynamic* EPC. As in [11], there are two control parameters (C_1 and C_2) instead of a single *C*. They are calculated slot-by-slot as a function of other parameters that mainly depend on *reader-to-tag* (R-T) and *tag-to-reader* (T-R) data rates. It is suggested to set $C_2 \in [0.1, 1]$ and C_1=0.1, because they claim, greatly improves performance in comparison to *Dynamic* EPC. They assume that R-T is set to 64 kbps and T-R is constant for each simulation, in the range 16 kbps ≤ T-R ≤ 128 kbps. However, this assumption is not correct, since the R-T value set affects the final T-R. In particular, with R-T= 64 kbps, T-R must be 170 kbps ≤ T-R ≤ 640 kbps. The correct value of T-R indeed depends on the modulation and data encoding assumed [2], but authors do not compute T-R this way. Following the standard specifications, C_1 and C_2 should be computed assuming T-R and R-T data rates of 40 and 80 kbps, respectively. Moreover, in *Dynamic* EPC tests they set *C* as *C*=3, but the standard recommends establishing variable values according to the C value, in order to get a better response. Therefore, results in [14] have to be reanalyzed.

B. Indirect Q heuristics family

Indirect *Q* heuristics compute $\hat{n}$ by means of some heuristic, and then adjust the frame length (*k*) to the best throughput. They only use information of the last frame. The first DFSA indirect *Q* heuristic [15] selected the number of competing nodes (in our case tags) in the i-*th* frame as:

$$\hat{n}_i = 2.39 \cdot c_i \tag{1}$$

and established the next frame length to:

$$k_{i+1} = round(\hat{n}_i - id_i) \tag{2}$$

Eq. (1) is computed assuming that the number of competing nodes follows a Poisson distribution of mean 1. Evidently, this does not hold in general. Actually, the frame length from eq. (2) does not satisfy the power-of-two constraint.

Recent research in [16-19] has also adopted the same heuristic to compute $\hat{n}$, stating that frame length must be selected according to the standard constraints. However, none of these works indicate how to compute the optimal *Q* value.

Another heuristic, the *Lower Bound* estimation, defines:

$$\hat{n}_i = id_i + 2 \cdot c_i \tag{3}$$

This trivial (every collision involves at least two tags) and inaccurate lower bound has deserved some attention [20].

In the estimation method in [21] (*C-ratio*), as a simplification, the authors assume a binomial distribution of the number of tags that select each slot (with success probability *p=1/k*). They define the collision ratio (*C-ratio*) as the ratio of the number of slots with collision to the frame size. Since the collision event is the complementary of the empty slot or single-tag ones, the *C-ratio* can be calculated as:

$$C_{ratio} \stackrel{\Delta}{=} \frac{c_i}{k_i} = 1 - \left(1 - \frac{1}{k_i}\right)^{n_i} \left(1 + \frac{n_i}{k_i - 1}\right) \tag{4}$$

The second part of eq. (4) is evaluated for each $n_i \geq 2 \cdot c_i + id_i$ and $\hat{n}$ is the value that gives a closest approximation to the *C-ratio*. However, this formula is undefined when the numbers of collision slots and frame slots coincide, $c_i = k_i$. This may occur for short frames and many competing tags. Despite this, other authors have considered this estimation [20]. The estimation method in [22] also assumes independent binomial distribution of the tags in each slot.

$$\frac{c_i}{k_i} = \sum_{j=2}^{n_i} \binom{n_i}{j} \left(\frac{1}{k_i}\right)^j \left(1 - \frac{1}{k_i}\right)^{n_i - 1} \tag{5}$$

Unlike the previous proposals, the simple $\hat{n}$ estimator in [23] is a SF-FP estimator (Eq. 6).

$$n_i = (k_i - 1)\frac{id_i}{e_i} \tag{6}$$

Nevertheless, this estimator has shortcomings. If $id_i = 0$, it selects $\hat{n}_i = 0$, but it must be $\hat{n}_i \geq 2 \cdot c_i$. For $e_i = 0$, the value of $\hat{n}_i$ is undefined. In this case, the authors suggest applying an upper bound to estimate the number of tags.

Finally, we must remark that these heuristics do not provide any procedure to select the optimal *Q* value for the next frame.

C. Error minimization estimator

Vogt proposes in [13] a SF-FP procedure based on Minimum Squared Error (MSE) estimation. It minimizes the Euclidean norm of the vector difference between actual frame statistics and expected values. Eq. (7) assumes *n* tags in the access procedure. Let $\hat{n}$ be the value of *n* that minimizes the error.

$$\hat{n}_i = \arg\min_n \left\{ \left\| \begin{pmatrix} e_i \\ id_i \\ c_i \end{pmatrix} - \begin{pmatrix} E\{\mathbf{e} \mid n_i\} \\ E\{\mathbf{id} \mid n_i\} \\ E\{\mathbf{c} \mid n_i\} \end{pmatrix} \right\| \right\} \tag{7}$$

Where random variables **e, id** and **c** represent the number of empty, successful and collision slots respectively. Their expected values are computed assuming independent binomial distribution of the tags in each slot (eqs. (8), (9) and (10)):

$$E\{\mathbf{e}|n_i\} = k_i \left(1 + \frac{1}{k_i}\right)^{n_i} \tag{8}$$

$$E\{\mathbf{id}|n_i\} = k_i n_i \left(\frac{1}{k_i}\right)\left(1 + \frac{1}{k_i}\right)^{n_i - 1} \tag{9}$$

$$E\{\mathbf{c}|n_i\} = k_i - E\{\mathbf{e}|n_i\} - E\{\mathbf{id}|n_i\} \tag{10}$$

	Protocol	Operation	Data Set	Q selection	$\hat{n}$ computation
Dynamic EPC Protocols family	**Dynamic EPC [2]**	*Q-reset*	*SF-FP*	*Direct*	---
	Q^{+-} [11]	*Q-reset*	*SF-FP*	*Direct*	----
	Optimum-C [12]	*Q-reset*	*SF-FP*	*Direct*	----
	SCS [14]	*Q-reset*	*SF-FP*	*Direct*	----
In*d*irect Q Heuristic family	**Schoute [15]**	*Q-control*	*SF-SP*	*Indirect, Q=N*	*Heuristic*
	Lower Bound [13]	*Q-control*	*SF-SP*	*Indirect, Q=N*	*Heuristic*
	C-ratio [21]	*Q-control*	*SF-SP*	*Indirect, Q=N*	*Heuristic*
	Wang [22]	*Q-control*	*SF-SP*	*Indirect-Q*	*Heuristic*
	Chen-1 [23]	*Q-control*	*SF-FP*	*Indirect, Q=N*	*Heuristic*
Error Minimization Estimators	**MSE-Vogt [13]**	*Q-control*	*SF-FP*	*Indirect, Q=N*	*MSE*
	MSE-SbS [25]	*Q-reset*	*MF-FP*	*Indirect Q*	*MSE*
Maximum Likelihood estimators	**Chen-2 [27]**	*Q-control*	*SF-FP*	*Indirect, Q=N*	*ML*
	SbS [26]	*Q-reset*	*MF-FP*	*Indirect, Q=N*	*ML*
	Chen-3 [30]	*Q-control*	*SF-FP*	*Indirect, Q=N*	*ML*
	Floerker [31]	*Q-control*	*MF-FP*	*Indirect, Q=N*	*ML*

Table I. Classification of D-FSA protocols

Let $\hat{n}$ be the value of n that minimizes eq. (7). A limitation in [13] is that the performance of the algorithm is only compared against a heuristic in the same work, which is compatible with the I-code system [24], but it has not been evaluated for the EPCglobal Class-1 Gen-2 standard.

In [25] the authors propose MSE-SbS (*Minimum Squared Error Slot-by-Slot*) to modify the estimator of [13] to improve the response. They propose a *Q-reset* operation based on the error function in eq. (7). However, they do not specify when the reader must reset the cycle to get the best performance. The statistical information of several previous cycles allows us to improve the estimation (MF-FP operation), by means of:

$$\hat{n}_i = \arg\min_n \left\{ \left\| \begin{pmatrix} e_i \\ id_i \\ c_i \end{pmatrix} - \begin{pmatrix} E\{\mathbf{e} \mid n_i\} \\ E\{\mathbf{id} \mid n_i\} \\ E\{\mathbf{c} \mid n_i\} \end{pmatrix} \right\| + \sum_{p=1}^{i-1} \left\| \begin{pmatrix} e_p \\ id_p \\ c_p \end{pmatrix} - \begin{pmatrix} E\{\mathbf{e} \mid n_p\} \\ E\{\mathbf{id} \mid n_p\} \\ E\{\mathbf{c} \mid n_p\} \end{pmatrix} \right\| \right\} \tag{11}$$

That is, minimizing the cumulative error function. Although this is proposed as an enhancement to [13], it is compared instead against the heuristic in [15] and the maximum likelihood estimator of the same authors [26].

D. Maximum Likelihood (ML) estimators

The main idea behind this group of estimators is to compute the conditional probability of some observed event (or set of events) assuming that n nodes are undertaking the identification process, and selecting the n that maximizes such probability. The main issue of these algorithms is the exact formulation of this conditional probability, and the extra computational cost, which may render them unusable.

Reference [27] proposed a ML algorithm derived from the occupancy problem described in [28]. It is a *Q-control* SF-SP mechanism. When a cycle ends, the reader calculates the probability to find e_i slots empty and selects $\hat{n}$ as follows:

$$\hat{n}_i = \arg\max_{n_i \geq id_i + 2\cdot c_i} \{P(k_i, e_i \mid n_i)\}, \text{ being}$$

$$P(k_i, e_i \mid n_i) = \frac{(-1)^{e_i} k_i!}{e_i! k_i^{n_i}} \sum_{j=e_i}^{k_i} (-1)^j \frac{(k_i - j)^{n_i}}{(j - e_i)!(k_i - j)!} \tag{12}$$

Notice that this is an exact computation, unlike in previous groups of protocols, which assumed independent identically distributed (iid) binomial or Poisson distributions of tags in each slot. As a drawback, its performance has been only tested against the heuristic in [13] and the static EPC algorithm [2].

In [29] the authors presented an algorithm similar to the one proposed in [27]. In addition, in [29] the authors point out computational unsuitability for large values of k_i and $n_{i,}$, and propose the heuristic estimator of eq. (12) as an alternative. Nevertheless, this heuristic shows an error when e_i=0, because in the numerator appears *log(0)*, which is not defined.

$$\hat{n}_i = \frac{\log\left(\frac{e_i}{k_i}\right)}{\log\left(1 - \frac{1}{k_i}\right)} \tag{13}$$

Reference [26] is a step-forward of the one in [25] proposing the SbS (*Slot-by-Slot*) ML estimator that uses the number of empty slots as well as the number of identified tags. The authors propose this algorithm as a *Q-reset* mechanism. However, they do not provide any cycle restart decision rule. The probability formula in [26] can be reduced to:

$$P(k_i, e_i, id_i \mid n_i) = \frac{k_i! n_i!}{e_i! id_i! k_i^{n_i}} \sum_{z=0}^{\min\{c_i, (n_i - id_i)\}} \frac{(-1)^z}{z!} \cdot \left(\sum_{j=0}^{\min\{c_i - z, (n_i - id_i)\}} \frac{(-1)^j}{j!} \frac{(c_i - z - j)^{n_i - id_i - j}}{(c_i - z - j)!(n_i - id_i - j)!} \right) \tag{14}$$

However, the original (as the reduced) formula is erroneous: it returns negative probabilities in some cases (*e.g.* c_i=1, id_i =1 e_i=1, k_i=4, for $n_i \geq id_i + 2\cdot c_i$), and we have tested that in other configurations where results are positive, they do not coincide with the simulated results.

In [30] the author models the probability of event $\{id, e, c\}$ as a multinomial distribution problem (an approximation to the actual probability). From eqs. (8), (9) and (10) the probabilities of empty, successful and collision slot are $p_0 = E\{\mathbf{e} \mid n_i\}$, $p_1 = E\{\mathbf{id} \mid n_i\}$ and $p_{\geq 2} = E\{\mathbf{c} \mid n_i\}$, respectively.

$$P(k_i, e_i, id_{i,} c_i \mid n_i) = \frac{k_i!}{e_i! id_i! c_i!} p_0^{e_i} p_1^{id_i} p_{\geq 2}^{c_i} \tag{15}$$

As in [27], the author only compares this estimator with the heuristics proposed in [13] and [15].

The estimator in [31] uses statistical information of several frames (it is the only MF-FP estimator proposed so far) to update the tags probability distribution according to a Bayesian methodology. That is, from the *a-priori* tag probability distribution (distribution at the end of cycle *i*-1) the *a-posteriori* distribution is derived (distribution at the end of cycle *i*), according to expression (16).

$$P(n_i \mid cycles_{1:i}) = \alpha \cdot P(n_i \mid cycles_{1:(i-1)}) \cdot P(k_i, id_i, e_i, c_i \mid n_i) \tag{16}$$

Where α is a normalizing constant, whose value is not indicated by the authors. In this framework, at the end of each frame, the reader extracts as $\hat{n}$ the mode of the *a-posteriori* distribution. However, in the first iteration, since the *a-priori* distribution is not available, authors directly assume that the likelihood is the *a-posteriori* distribution. This is, indeed, false, since likelihood functions are not mass probability functions (the sums of probabilities are not equal to one).

We have obtained simplified versions of that expression [31], but the final form is unnecessarily complex for this survey (it is available in [31]). Besides, as in previous works, the comparison only considers some heuristics [13, 15].

VI. Performance Evaluation

In this section we compare the performance of the previously discussed DFSA algorithms. As stated in sections II and V, this has not clearly -and, in our opinion, fairly- been achieved so far.

A. Evaluation rules

We have selected conditions and parameters representing an actual system implementation:

- The physical configuration parameters correspond to a commercial Alien 8800 [32] system, which is a widely used EPCglobal Class-1 Gen-2 RFID equipment. We have validated our simulation results with laboratory test beds based on this system [33].
- The EPCglobal Class-1 Gen-2 frame length constraint (power of two) holds. For each number of competing tags *N*, there is an optimal *Q* that maximizes the throughput. Conversely, for each *Q* there is a (closed) set of *N* values, for which the expected performance will be optimal. Table II summarizes the values of *Q* (see the appendix for their calculation). These values have not been evaluated in any previous work. When any of the algorithms under evaluation has to select the next frame size we just pick the correct value from this table. Notice that this does not add complexity to the algorithms.
- In our simulation, exactly *N* tags enter the reader range. The identification proceeds until all tags are detected.
- In each simulation the number of slots (*S*) is measured. For each *N* the same setup is simulated (as many independent runs) until a confidence interval of 10% of the mean values of *S* with a confidence degree of 90%.

Indeed, some algorithms leave some parameters open or are faulty. In those cases we assume the following:

- Dynamic EPC [2]: Since the standard only recommends to set high *C* values if the last frame *Q* is low (and *vice-versa*), we set *C=0.2,* if *Q*>8 and *C=0.4,* if *Q*<8.
- Q^{+-} algorithm [11]. The computing methodology for C_c is unknown. The authors only state that C_c Є [0.1, 0.5]. Therefore, we assumed the same as in the previous assumption.
- SCS algorithm [14]: C_1 and C_2 are computed assuming T-R and R-T data rates of 40 and 80 kbps, respectively.
- The C-ratio algorithm [21] fails if C_{ratio}=1, and Chen-1 [23] is undefined under some conditions (section V.B). In those cases, we take the lower bound estimation [13].
- MSE-SbS [25] and SbS [26] do not specify how often the frame must be reset. We have assumed a cycle-by-cycle operation. In addition, "negative" likelihoods in [26] are treated as zeros (see section V.D).
- Indirect-Q algorithm frame size is selected as in table II.

The previous mechanisms were evaluated with a discrete-event Matlab simulator. An optimal algorithm, with full knowledge of the tags in each cycle, was the reference. The initial value of *Q* was 4 in all cases, following the standard.

B.Evaluation results

Figures 3, 4, 5, and 6 show the mean identification time versus the initial tag population size (*N*), up to 500, for the four groups of protocols introduced in section V. We observe in all cases that the more initial tags, the larger the differences between the identification time of all proposals and the optimum. Besides, we have also computed the effect of non-optimal *Q* selection. All algorithms have been tested under the original assumption of selecting 2^Q as the closest value to $\hat{n}$. This renders 5-10% performance degradation in Heuristics, MSE and ML algorithms. Notice that the dynamic EPC protocol family is not affected by this issue, since *Q* is directly computed.

In short, the best algorithm proposals (considering large *N* values and optimal *Q* selection) are: Optimum-C [12], Lower-bound [13], MSE-SbS [25], and Chen-3 [30]. Figure 7 compares them. The ML estimator (Chen-3) achieves the closest approach to the optimal algorithm for large *N* values. Notice that this protocol, in fact, uses an approximation to compute the likelihood. Therefore, exact computation may lead to a better response. In addition, the mixed ML-Bayesian strategy in [31] has an excellent performance for lower values of *N*, but it degrades as the initial population grows.

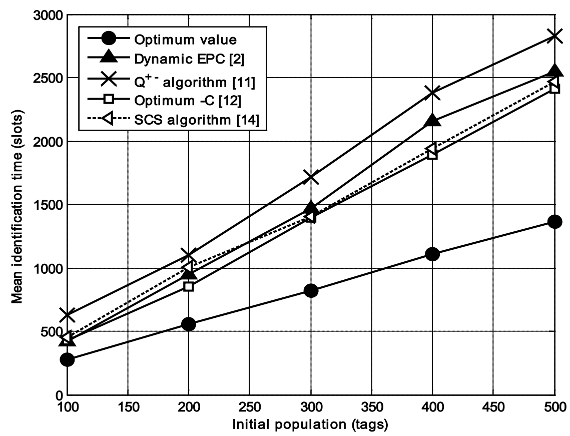

Figure 3. Dynamic EPC family. Mean Identification time *versus* initial tag population

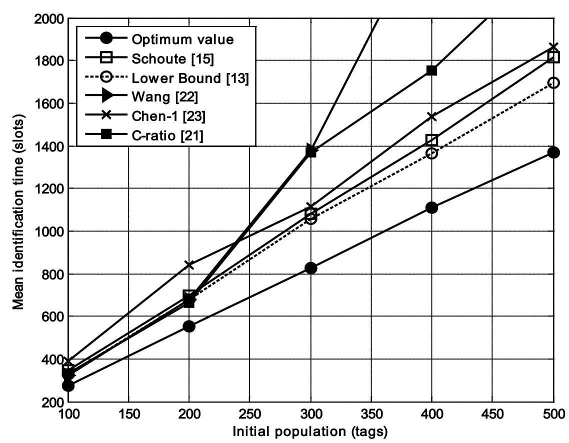

Figure 4. Heuristic estimators. Mean identification time *versus* initial tag population

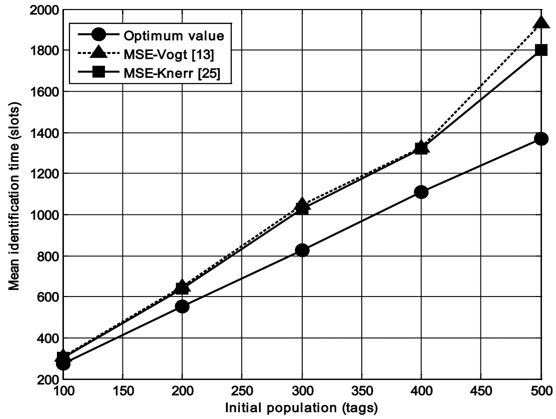

Figure 5. MSE estimators. Mean identification time *versus* initial tag population

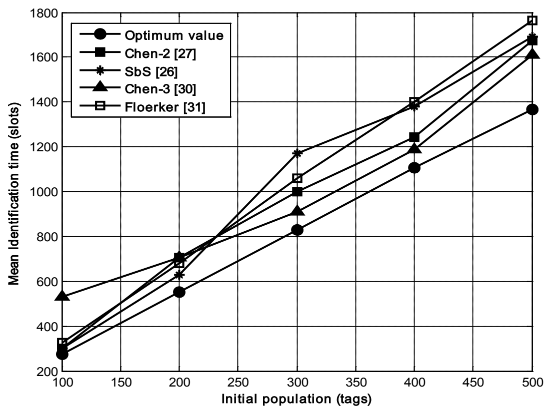

Figure 6. ML estimators. Mean Identification time *versus* initial tag population

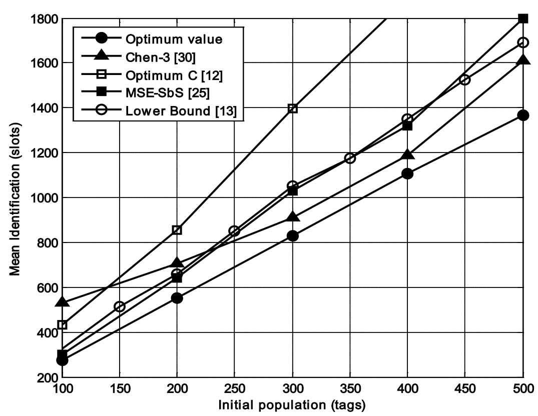

Figure 7. Comparison of the best algorithms proposals. Mean identification time *versus* initial tag population

VII. Conclusions and Open Issues

We achieve relevant results regarding current DFSA proposals: classification and direct comparison of the most important algorithms and optimal frame length selection. We demonstrate that maximum-likelihood algorithms achieve the best performance, in terms of mean identification delay. We conclude that Chen-3 [30] is appropriate for populations over 300 tags, and Lower-bound [13] or MSE-SbS [25] are otherwise preferable (see Fig. 7). To conclude we present some interesting open issues:

- MSE and ML algorithms achieve notably good performance despite of some non-realistic assumptions (*e.g.* Chen-3 approximations) or faulty formulas [21, 26]. Therefore, there may be some margin for improvement.
- Multi-frame proposals are scarce. Increasing the statistical information may lead to better estimators.
- Algorithm performance often depends on parameter selection, but this matter is open in many cases.
- *Q*-control proposals depend on collected statistical information, which, in the first cycle -after the resting state- is not available. The standard suggests setting Q_1=4. However, we believe that Q_1 can be tuned according to additional criteria (*e.g.*, the largest expected tag population) to improve performance.
- Algorithm performance also depends on the computation time for estimating the number of tags. Therefore, the computational order of each proposal can be analyzed.

Acknowledgment

This work has been funded by grants DEP2006-56158-C03-03/EQUI and TEC2007-67966-01/02/TCM (CON-PARTE-1/2) and "Programa de Ayudas a Grupos de Excelencia de la Region de Murcia, Fundacion Seneca", Spain.

References

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

[2] "EPC Radio-Frequency Identify protocol for communications at 868-960 MHz,Version 1.0.9: EPCglobal Standard Specification", 2004. Available on-line: http://www.epcglobalinc.org/standards

[3] Shih, D-H., Sun, P-L, Yen, D.C., Huang, S-M, "Taxonomy and survey of RFID anti-collision protocols". Computer and Communications, vol. 29, pp. 2150-2166, 2006.

[4] Hush, D.R, Wood, C., "Analysis of Tree Algorithms for RFID Arbitration". In Proc. of International Symposium on Information Theory, pp. 107-114, Cambridge, Massachusetts, USA, 1998.

[5] Abraham, C., Ahuja, V., Ghosh A.K., Pakanati, P., "Inventory Management using Passive RFID Tags: A survey", Department of Computer Science, The University of Texas at Dallas, Richardson, Texas, USA, pp. 1-16, 2002.

[6] Myung, J., Lee, W., "Adaptive Splitting Protocols for RFID Tag Collision Arbitration", *MobiHoc'06*, Florence, Italy, pp. 203-213, 2006.

[7] Quan, C-H., Hong, W-K., Kim, H-C., "Performance Analysis of Tag Anti-collision Algorithms for RFID Systems", EUC Workshops 2006, LNCS, vol. 4097, pp. 382-391, 2006.

[8] Burdet, L.A., "RFID Multiple Access Methods", Technical Report ETH Zurich, 2004.

[9] Tang, Z., He, Y., "Research of Multi-access and Anti-collision Protocols in RFID Systems". In Proc. of IEEE Int.ernational Workshop on Anti-counterfeiting, Security and Identification. pp. 377-380, 2007.

[10] Wang, Z., Liu, D., Zhou, X., Tan, X., Wang, J., Min, H., "Anti-collision Scheme Analysis of RFID System". Auto-ID Labs, 2007.

[11] Lee, D., Kim, K., Lee, W., "Q+- Algorithm: An Enhanced RFID Tag Collision Arbitration Algorithm". In Proc. of Ubiquitous Intelligence and Computing. LNCS, vol. 4611, pp. 23-32, 2007.

[12] Joe, I., Lee, J., "A Novel Anti-collision Algorithm with Optimal Frame Size for RFID System". In Proc. of 5th International Conference on Software Engineering Research, Management and Applications, pp. 424-428, 2007.

[13] Vogt, H., "Efficient Object Identification with Passive RFID Tags". In Proc. of International Conference on Pervasive Computing, LNCS, vol. 2414, pp. 98-113, 2002.

[14] Daneshmand, M., Wang, C., Sohraby, K., "A New-Count Selection Algorithm for RFID protocol", 2nd International Conference on CHINACOM, pp. 926-930, 2007.
[15] Schoute, F.C, "Dynamic Frame Length ALOHA", IEEE Transactions on Communications, vol. 31, no. 4, pp. 565-568. 1983.
[16] Deng, X., Rong, M., Liu, T., Yuan, Y., Yu, D., "Tag Count Frame Slotted Aloha: A Novel Anti-Collision Protocol in RFID Systems. IEE VTC Spring 2008, pp. 2666-2670, Marina-Bay, Singapore, 2008.
[17] Rivest, R.L, "Network Control Bayesian Broadcast". IEEE Transactions on Information Theory, vol. IT-33, no.3, pp. 323-328, 1987.
[18] J. Cha, J. Kim, "Novel Anti-collision Algorithms for Fast Object Identification RFID System", in Proc. of the 11th Conference on Parallel and Distributed Systems, vol. 2, pp. 63-67, 2005.
[19] Tong, Q., Zou, X., Liu, D., Dai, Y. "Modeling the Anti-collision Process of RFID System by Markov Chain", In Proc of. International Conference on Wireless Communications, Networking and Mobile Computing WiCom 2007, pp. 2054-2057, 2007.
[20] Abderrazak, H., Slaheddine, B., Ridha, B.,"Performance Analysis of Aloha Protocol for RFID Communication Systems". International Journal of Computer Science and Network Security, vol. 7, no. 12, pp. 80-83, 2007.
[21] Cha, J-R., Kim, J-H, "Dynamic Frame Slotted Aloha Algorithms using Fast Tag Estimation Method for RFID System". In Proc. of. IEEE Consumer Communications and Networking Conference, vol. 2, pp. 768-772, 2006.
[22] Wang, B-S., Zhang, Q-S., Yang, D-K., Di, J-S., "Transmission Control Solutions Using Interval Estimation Method for EPC C1G2 RFID Tag Identification", In Proc. of Int. Conf. on Wireless Communications, Networking and Mobile Computing, pp. 2105-2108, 2007.
[23] Wen-Tzu Chen, "An efficient Anti-Collision Method for Tag Identification in a RFID System", IEICE Transactions on Communications, Vol. E89-B, no. 12, pp. 3386-3392, 2006.
[24] I-CODE: available on line in: http://www.semiconductor.com/resources/reports_database/view_device.asp?sinumber=8205. Last access: Nov. 2008.
[25] Knerr, B.Holzer, M., Angerer, C., Rup. M, "Slot-by-slot Minimum Squared error estimator for tags populations in FSA protocols". In Proc. of 2nd Int. EURASIP Workshop on RFID, pp. 1-13, Budapest, 2008.
[26] Knerr, B.Holzer, M., Angerer, C., Rup. M.,"Slot-by-slot Maximum Likelihood estimation of tags populations in frame slotted aloha protocols". Proc. SPECTS'08, pp. 303-308, Edinburgh, UK, 2008.
[27] Chen, W-T., Lin, G-H., "An Efficient Scheme for multiple access in a RFID System". In Proc. of International Conference on Wireless Networks, pp. 160-163, 2006.
[28] Feller, W., "An Introduction to Probability Theory and its Applications", vol. 1, 3rd edition, New York, Wiley 1970.
[29] Khandhelwal, G., Lee, K., Yener, A., "ASAP: A MAC Protocol for Dense and Time-Constrained RFID Systems". EURASIP Journal on Wireless Communications and Networking, vol. 2007.
[30] Chen, W-T., "An Accurate Tag Estimate Method for Improving the Performance of an RFID Anti-collision Algorithm Based on Dynamic Frame Length ALOHA", IEEE Trans. on Automation Science and Engineering, 2008.
[31] Floerkemeier, C., Wille, M., "Comparison of Transmission Schemes for Framed Aloha based RFID", Workshop on RFID Extended Network Deployment of Technologies and Applications, Phoenix, AZ, 2006.
[32] Alien Reader 8800, development evaluation kit: http://www.alientechnology.com/readers/index.php
[33] M. V. Bueno-Delgado, J. Vales-Alonso, E. Egea-López, J. García-Haro, "Estudio de la configuración óptima de longitud de ciclo en sistemas RFID". Proc. VII Jornadas de Ingeniería Telemática, Spain. 2008.
[34] Odlyzco, A.M., "Asymptotic Enumeration Methods", Handbook of combinatory", vol. 2, pp. 1063-1229, MIT Press, Cambridge, Massachusetts, USA, 1996.

APPENDIX

The throughput of a FSA system, $S(n, k)$, is given by equation (15) as a function of the number of available slots (k) and the number of competitors (n), and reach a maximum $S=e^{-1}$ if $n = k$. However, as previously stated, the number of slots for EPCglobal must lie in $\{2^Q: Q=0,\ldots, 15\}$.

$$S = \frac{n}{k}\left(1-\frac{1}{k}\right)^{n-1} = \frac{n}{2^Q}\left(1-\frac{1}{2^Q}\right)^{n-1} \tag{17}$$

Given a value of Q we can compute the set of values of n maximizing the expected throughput. These sets are close and have the form $[n_{min}(Q),\ldots, 2^Q, \ldots, n_{min}(Q)]$. That set is close and reaches a maximum since $S(n,k)$ is unimodal, that is, there exists an index m such that $S(1,k)\leq S(2,k) \leq \ldots\leq S(m,k)\geq S(m+1,k) \geq\ldots$

Unimodality is easy to prove since $S(n,k)$ is log-concave and it has no zeros [34]. Log-concavity condition is:

$$S(n-1,k) - S(n+1,k) \leq S^2(n,k) \tag{18}$$

Therefore,

$$\frac{\frac{n-1}{k}\left(1-\frac{1}{k}\right)^{n-2}}{\frac{n}{k}\left(1-\frac{1}{k}\right)^{n-1}} = \frac{n-1}{n\left(1-\frac{1}{k}\right)} = \frac{S(n-1,k)}{S(n,k)} \leq$$

$$\leq \frac{n}{(n-1)\left(1-\frac{1}{k}\right)} = \frac{S(n,k)}{S(n+1,k)} \tag{19}$$

Hence, unimodality is proved. In the boundaries of each set, the following condition must be satisfied for each Q value:

$$n_{max}(Q-1) = n_{min}(Q)-1 \tag{20}$$

being $n_{max}(Q-1)$ the largest natural value satisfying:

$$S\left(n_{\max}(Q-1), Q-1\right) > S\left(n_{\min}(Q), Q\right) \tag{21}$$

That is,

$$\frac{n_{\max}(Q-1)}{2^Q-1}\left(1-\frac{1}{2^{Q-1}}\right)^{n_{\max}(Q-1)-1} > \tag{22}$$

$$> \frac{n_{\max}(Q)}{2^Q}\left(1-\frac{1}{2^Q}\right)^{n_{\max}(Q)-1}$$

A sequential algorithm can easily evaluate this condition. Table II summarizes the optimal set associated with each Q.

Optimal *Q*	***n* range**
0	n=1
1	1<n≤3
2	3<n≤6
3	6<n≤11
4	11<n≤22
5	22<n≤44
6	44<n≤89
7	89<n≤177
8	177<n≤355
9	355<n≤710
10	710<n≤1420
11	1420<n≤2839
12	2839<n≤5678
13	5678<n≤11357
14	11357<n≤22713
15	n>22713

Table II. Optimal *Q*