Generalized Energy-Efficient Algorithms for the RFID Estimation Problem

Tao Li, Samuel S. Wu, Shigang Chen, and Mark C. K. Yang

***Abstract*—Radio frequency identification (RFID) has been gaining popularity for inventory control, object tracking, and supply-chain management in warehouses, retail stores, hospitals, etc. Periodically and automatically estimating the number of RFID tags deployed in a large area has many important applications in inventory management and theft detection. Prior works focus on designing time-efficient algorithms that can estimate tens of thousands of tags in seconds. We observe that for an RFID reader to access tags in a large area, active tags are likely to be used due to their longer operational ranges. These tags are battery-powered and use their own energy for information transmission. However, recharging batteries for tens of thousands of tags is laborious. Hence, conserving energy for active tags becomes critical. Some prior works have studied how to reduce energy expenditure of an RFID reader when it reads tag IDs. We study how to reduce the amount of energy consumed by active tags during the process of estimating the number of tags in a system. We design two energy-efficient probabilistic estimation algorithms that iteratively refine a control parameter to optimize the information carried in transmissions from tags, such that both the number and the size of transmissions are reduced. These algorithms can also take time efficiency into consideration. By tuning a contention probability parameter ω, the new algorithms can make tradeoff between energy cost and estimation time.**

***Index Terms*— Radio frequency identification (RFID) tags.**

I. Introduction

RADIO frequency identification (RFID) technology has been widely used in various commercial applications, including inventory control, object tracking, and supply-chain management. RFID tags (each storing a unique ID) are attached to merchandizes at retail stores, equipment at hospitals, or goods at warehouses, allowing an authenticated RFID reader to quickly access properties of each individual item or collect statistical information about a large group of items.

This paper focuses on an RFID-enabled function that is very useful in inventory management. Imagine a large warehouse with thousands of laptops, cell phones, electronics, apparel, bags, or furniture pieces. A national retail survey showed that administration error, vendor fraud and employee theft caused about 20 billion dollars lost a year [1]. Hence, it is desirable to have a quick way of counting the number of items in the warehouse or in each section of the warehouse. To timely detect theft or management errors, such counting may be performed frequently.

If each item is attached with an RFID tag, the counting problem can be solved by an RFID reader that receives the IDs transmitted (or backscattered) from the tags [2]. However, reading the actual tag IDs can be time-consuming because so many of them have to be delivered in the same low-rate channel, and collisions caused by simultaneous transmissions by different tags make the matter worse. To address this problem, Kodialam and Nandagopal [3], [4] showed that reading time can be greatly reduced through probabilistic methods that *estimate the number of tags with an accuracy that can be arbitrarily set*. This is called the *RFID estimation problem*. The follow-up work by Qian *et al.* [5] significantly reduces estimation time when compared to [3]. It can be shown that even for applications that require reading the actual tag IDs, estimating the number of tags as a preprocessing step will help make the main procedure of reading tag IDs much more efficient [3]. Another advantage of estimating the number of tags without reading the IDs is that it ensures anonymity of the tags, which may be useful in privacy-sensitive scenarios involving RFID-enhanced passports or driver's licences, where counting the number of people present is needed but revealing their identities is not necessary.

Is time efficiency the only performance metric for the estimation problem in large-scale RFID systems that use active tags? We argue that energy cost is also an important issue that must be carefully dealt with. For any application that requires an RFID reader to access tags in a large area, it is likely that battery-powered *active tags* will be used. *Passive tags* harvest energy from radio signal of a reader and use such a minute amount of energy to deliver information back to the reader. Their typical reading range is only several meters, which does not fit well with the big warehouse scenario. Active tags use their own power to transmit. A longer reading range can be achieved by transmitting at higher power. They are also richer in resources for implementing advanced functions. Their price becomes less of a concern if they are used for expensive merchandise or reused many times as goods moving in and out of the warehouse. However, active tags also have a problem. They are powered by batteries. Recharging batteries for tens of thousands of tags is a laborious operation, considering that tagged products may be stacked up, making tags not easily accessible. To prolong the lifetime of tags and reduce the frequency of battery recharge, all functions that involve large-scale transmission by many tags should be made

Manuscript received April 07, 2010; revised December 02, 2010 and May 18, 2011; accepted February 14, 2012; approved by IEEE/ACM Transactions on Networking Editor M. Kodialam. Date of publication April 20, 2012; date of current version December 13, 2012. This work was supported in part by the US National Science Foundation under Grant CPS-00079686.

T. Li and S. Chen are with the Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL 32608 USA (e-mail: tali@cise.ufl.edu; sgchen@cise.ufl.edu).

S. S. Wu is with the Department of Epidemiology and Health Policy Research and the Department of Statistics, University of Florida, Gainesville, FL 32611 USA.

M. C. K. Yang, deceased, was with the Department of Statistics, University of Florida, Gainesville, FL 32611 USA.

Digital Object Identifier 10.1109/TNET.2012.2192448

1063-6692/$31.00 © 2012 IEEE

energy-efficient. Prior works focus on energy-efficient anti-collision protocols that minimize energy consumption of a mobile reader [6], [7] when the reader collects tag IDs. To the best of our knowledge, this paper is the first to study energy-efficient solutions for the estimation problem in large-scale RFID systems that use active tags.

Our paper has four major contributions. First, we observe that there exists an asymmetry in energy cost. Solving the RFID estimation problem incurs energy cost both at the RFID reader and at active tags. The asymmetry is that energy cost at tags should be minimized while energy cost at the reader is relatively less of a concern because the reader's battery can be replaced easily or it may be powered by an external source. To exploit this asymmetry, our new algorithms follow a common framework that trades more energy cost at the reader for less cost at the tags. The reader will continuously refine and broadcast a control parameter called *contention probability*, which optimizes the amount of information the reader can extract from transmissions by tags. This in turn reduces the number of transmissions by tags that are necessary to achieve a certain estimation accuracy.

Second, the design of our estimation algorithms is based on the maximum likelihood estimation method (MLE) that is different from the probabilistic counting methods [8] used by [3], [4]. Our estimation algorithms optimize their performance by iteratively applying MLE with continuously refined parameters. These new algorithms not only require fewer transmissions by tags, but also minimize the size of each transmission. The number of transmissions made by tags in our best algorithm is less than one fourth achieved by the state-of-the-art algorithms. In terms of the total number of bits transmitted by tags, it is more than an order of magnitude smaller.

Third, we formally analyze the confidence intervals of estimations made by our new algorithms and establish the termination conditions for any given accuracy requirement. We perform extensive simulations to demonstrate that the measured results match well with the analytical results and that the new algorithms perform far better in terms of energy saving than the best existing algorithms.

Fourth, our algorithms are generalized with a tunable parameter ω, specifying the contention probability that tags use to decide whether they will transmit. By modifying this parameter, the generalized algorithms can make a tradeoff between energy cost and estimation time (i.e., the time it takes to complete the process of estimating the number of tags). Even though our main goal is to reduce energy cost, the ability for performance tradeoff makes our algorithms more adaptable in practical settings that are sensitive not only to energy cost but also to estimation time.

The rest of this paper is organized as follows. Section II discusses the related work. Section III defines the problem to be solved and the system model. Sections IV and V propose two energy-efficient algorithms for the RFID estimation problem. Section VI evaluates the algorithms through simulations. Section VII draws the conclusion.

II. Related Work

Most existing work focuses on how to efficiently read the tag IDs. Collision occurs when multiple tags transmit their IDs in the same time-slot. Collision arbitration protocols mainly fall into two categories: the framed ALOHA-based protocols [7], [9]–[11] and the tree-based protocols [6], [12]–[15]. In the former category, each polling request carries a frame length, and every tag individually chooses a slot in the frame to transmit its ID. The process repeats until all tags successfully transmit their IDs to the RFID reader. In the latter category, a reader first sends out an ID prefix string. The tags whose ID matches the string will respond. If a collision happens, the reader will append a "0" or "1" to the prefix string and send out the new string. This process repeats until only one tag responds. Essentially the approach traverses a binary tree with the tag IDs being the leaf nodes.

Instead of identifying individual RFID tags, Floerkemeier [16], [17] studies the problem of estimating the cardinality of a tag set based on the number of empty slots. The proposed scheme employs a Bayesian probability estimation to achieve fast estimation. The scheme is similar to hash-based estimators , [18], and the difference is discussed in [4]. In Kodialam and Nandagopal's approach [3], information from tags is collected by an RFID reader in a series of time frames. Each frame consists of a number of slots, and the tags probabilistically respond in those slots. Using the probabilistic counting methods, the reader estimates the number of tags based on the number of empty slots or the number of collision slots in each frame. Their best estimator is called the Unified Probabilistic Estimator (UPE). A follow-up work by the same authors proposes the Enhanced Zero-Based Estimator (EZB) [4], which makes its estimation based on the number of empty slots. The focus of the above estimators is to reduce the time it takes a reader to complete the estimation process. Because their goal is not conserving energy for active tags, their design is not geared toward reducing the number of transmissions made by the tags.

The Lottery-Frame scheme (LoF) [5] by Qian *et al.* employs a geometric distribution-based scheme to determine to which slot in a time frame each tag will respond. It significantly reduces the estimation time when compared to UPE. However, every tag must respond in each of the time frames, resulting in large energy cost when active tags use their own power to transmit. The First Non-Empty slots Based algorithm (FNEB) [19] uses the slot number of the first reply from tags in a frame to count RFID tags in both static and dynamic environments.

Also related is a novel security protocol proposed by Tan *et al.* to monitor the event of missing tags in the presence of dishonest RFID readers [20]. In order to prevent a dishonest reader from replaying previously collected information, they maintain a timer in the server and periodically update the system clock. Li *et al.* [21] design a series of efficient protocols that employ novel techniques to identify missing tags in large-scale RFID systems.

None of the above estimators are designed with energy conservation in mind. In the following, we will present our energy-efficient estimators.

III. Problem Definition And System Model

A. RFID Estimation Problem

The problem is to design efficient algorithms to estimate the number of RFID tags in a deployment area without actually reading the ID of each tag. Let N be the actual number of tags and $\hat{N}$ be the estimate. The estimation accuracy is specified by a confidence interval with two parameters: a probability value α and an error bound β, both in the range of $(0, 1)$. The requirement is that the probability for $N/\hat{N}$ to fall in the interval $[1-\beta, 1+\beta]$ should be at least α, i.e.,

$$\text{Prob}\left\{(1-\beta)\hat{N} \leq N \leq (1+\beta)\hat{N}\right\} \geq \alpha.$$

Our goal is to reduce the energy overhead incurred to the tags during the estimation process that achieves the above accuracy. Prior works on the RFID estimation problem focus on time efficiency, which is the amount of time an RFID reader spends in estimating the number of tags in the system. Our work focuses on energy efficiency, which is the amount of energy the tags spend during estimation process.

B. Active Tags

The type of active RFID systems considered in this paper is applicable to a large deployment area that is hundreds of feet or more across. Passive tags are beyond the scope of this paper. If they were used, one would have to take the RFID reader and move around the whole area, collecting tag information once every few feet. Active tags allow a reader to collect information from one location.

Tagged goods (such as apparel) may stack in piles, and there may be obstacles, such as racks filled with merchandise, between a tag and the reader. We expect active tags are designed to transmit with significant power that is high enough to ensure reliable information delivery in such a demanding environment. Hence, energy cost due to the tags' transmissions is the main concern in our algorithm design; it increases at least in the square of the maximum distance to be covered by the RFID system. Energy consumption that powers a tag's circuit for computing and receiving information is not affected by long distance and obstacles. Our new estimators are designed for RFID systems where power consumption by tags is dominated by transmission events due to long distances that the systems need to cover. Energy consumed by the RFID reader is less of a concern. We assume the reader transmits at sufficiently high power.

C. Communication Protocol

We use the following communication protocol between a reader and tags. The reader first synchronizes the clocks of the tags and then performs a sequence of pollings. Clock synchronization only needs to happen at the beginning of the protocol execution. RFID systems operate in low-rate wireless channels. Our new estimators only take a few seconds to complete. Clock drift should not be a major issue in a low-rate channel within such a short period of time.

In each polling, the reader sends out a request, which is followed by a slotted time frame during which the tags respond. The polling request from the reader carries a *contention probability* $0 < p \leq 1$ and a frame size f. Each tag will participate in the current polling with probability p. If it decides to participate, it will pick a slot uniformly at random from the frame and transmit a bit string (called *response*) in that slot. The format of the response depends on the application. If the tag decides to not participate, it will keep silent. In our solutions, p will be set in the order of $1/N$.

If we know a lower bound $N_{\min}$ of N, the contention probability can be implemented efficiently to conserve energy. For example, a company's inventory of certain goods may be in the thousands and never before reduced below a certain number, or the company has a policy on the minimum inventory, or the RFID estimation becomes unnecessary when the number of tags is below a threshold. In these cases, we will have a lower bound $N_{\min}$, which can be much smaller than N. If we know such a value of $N_{\min}$, we can implement a contention probability p without requiring all tags to participate in the contention process. Since only a small number of tags actually participate in contention, energy cost is reduced. The implementation is described as follows. At the beginning of a polling, each tag makes a probabilistic decision: It goes to a standby mode for the current polling with probability $1-(1/N_{\min})$ and wakes up until the next polling starts, or it stays awake to receive the polling request with probability $1/N_{\min}$ and then decides to respond with probability $\min\{p \times N_{\min}, 1\}$. For example, if $N = 10\,000$ and $N_{\min} = 1000$, then only 10 tags stay awake in each polling. In Section IV-E, another energy-reduction method, called request-less pollings, will be proposed to eliminate most polling requests.

In the above communication protocol, the reader's request may include an optional prefix, and only tags that satisfy the prefix will participate in the polling. For example, suppose all tags deployed in one section of a warehouse carry the 96-bit GEN2 IDs that begin with "000" in the Serial Number field. In order to estimate the number of tags in this section, the request carries a predicate testing whether the first three bits of a tag's Serial Number is "000."

D. Empty/Singleton/Collision Slots

A slot is said to be *empty* if no tag responds (transmits) in the slot. It is called a *singleton slot* if exactly one tag responds. It is a *collision slot* if more than one tag responds. A singleton or collision slot is also called a *nonempty slot*. The Philips I-Code system [22] requires a slot length of 10 bits in order to distinguish singleton slots from collision slots. On the contrary, one bit is enough if we only need to distinguish empty slots from nonempty slots—"0" means empty and "1" means nonempty. Hence, the response will be much shorter (or consume much less energy) if an algorithm only needs to know empty/nonempty slots instead of all three types of slots as required by [3].

In order to prolong the lifetime of tags, there are two ways to reduce their energy consumption: reducing the size of each response and reducing the number of responses. We will design algorithms that require only the knowledge of empty/nonempty slots and employ statistical methods to minimize the amount of transmission needed from the tags.

IV. Generalized Maximum Likelihood Estimation Algorithm

Our first estimator for the number of RFID tags is called the *generalized maximum likelihood estimation* (GMLE) algorithm. It fully utilizes the information from all pollings in order to minimize the number of pollings it needs to meet the accuracy requirement.

A. Overview

GMLE uses the polling protocol described in Section III-C. The frame size f is fixed to be one slot. The RFID reader adjusts the contention probability for each polling. Let p_i be the contention probability of the ith polling. GMLE only records whether the sole slot in each polling is empty or nonempty. Based on this information, it refines the estimate $\hat{N}$ until the accuracy requirement is met. Let z_i be the slot state of the ith polling. When at least one tag responds, the slot is nonempty and $z_i = 1$. When no tag responds, it is empty and $z_i = 0$. The sequence of z_i, $i \geq 1$, forms the *response vector*.

At the ith polling, each tag has a probability p_i to transmit and, if any tag transmits, z_i will be one. Hence,

$$Prob\{z_i = 1\} = 1 - (1 - p_i)^N \approx 1 - e^{-Np_i}, \tag{1}$$

where N is the the actual number of tags.

If the contention probabilities of the pollings are picked too small, the response vector will contain mostly zeros. If the contention probabilities are picked too large, the response vector will contain mostly ones. Both cases do not provide sufficient statistical information for accurate estimation. As will be discussed shortly, our analysis shows that the optimal contention probability for minimizing the number of pollings is $p_i = 1.594/N$. The problem is that we do not know N (which is the quantity we want to estimate).

In order to determine p_i, GMLE consists of an *initialization phase* and an *iterative phase*. The former quickly produces a coarse estimation of N. The latter refines the contention probability and generates the estimation result.

B. Initialization Phase

We want to pick a small value for the initial contention probability p_1 at the first polling. The expected number of responding tags is Np_1. If p_1 is picked too large, a lot of tags will respond, which is wasteful because one response or many responses produce the same information—a nonempty slot. Suppose we know an upper bound $N_{\max}$ of N. This information is often available in practice. For example, we know $N_{\max}$ is 10 000 if the warehouse is designed to hold no more than 10 000 microwaves (each tagged with an RFID), or the company's inventory policy requires that in-store microwaves should not exceed 10 000, or the warehouse only has 10 000 RFID tags in use. $N_{\max}$ can be much bigger than N. We pick $p_1 = 1/N_{\max}$ such that the expected number of responding tags is no more than one. If $z_1 = 0$, we multiply the contention probability by a constant $C(> 1)$, i.e., $p_2 = p_1 \times C$ for the second polling. We continue multiplying the contention probability by C after each polling until a nonempty slot is observed. When that happens (say, at the lth polling), we have a coarse estimation of N to be $1/p_l$. Then, we move to the next phase. When C is relatively large, the initialization phase only takes a few pollings to complete due to the exponential increase of the contention probability.

C. Iterative Phase

This phase iteratively refines the estimation result after each polling and terminates when the specified accuracy requirement is met. Let $\hat{N}_i$ be the estimated number of tags after the ith polling. To compute $\hat{N}_i$, the reader performs three tasks at the ith polling. First, it sets the contention probability as follows before sending out the polling request:

$$p_i = \frac{\omega}{\hat{N}_{i-1}} \tag{2}$$

where $\hat{N}_{i-1}$ is the estimate after the previous polling and ω is a system parameter, which will be extensively analyzed in Section IV-C.1. Second, based on the received z_i and the history information, the reader finds the new estimate of N that maximizes the following likelihood function:

$$L_i = \prod_{j=1}^{i} (1 - p_j)^{N(1-z_j)} \left(1 - (1 - p_j)^N\right)^{z_j} \tag{3}$$

where $(1 - p_j)^{N(1-z_j)}(1 - (1 - p_j)^N)^{z_j}$ is the probability for the observed state z_j of the jth polling to occur. Namely, we want to find

$$\hat{N}_i = \arg\max_N \{L_i\}. \tag{4}$$

Third, after computing $\hat{N}_i$, the reader has to determine if the confidence interval of the new estimate meets the requirement. In the following, we show how the above tasks can be achieved.

1) Compute the value of $\hat{N}_i$: We compute the new estimate of N that maximizes (3). Since the maxima is not affected by monotone transformations, we use logarithm to turn the right side of the equation from product to summation:

$$\ln(L_i) = \sum_{j=1}^{i} \left[N(1-z_j)\ln(1-p_j) + z_j \ln\left(1-(1-p_j)^N\right)\right].$$

To find the maxima, we differentiate both sides:

$$\frac{\partial \ln(L_i)}{\partial N} = \sum_{j=1}^{i} \left[(1-z_j)\ln(1-p_j) - z_j \frac{(1-p_j)^N \ln(1-p_j)}{1-(1-p_j)^N}\right]. \tag{5}$$

We then set the right side to zero and solve the equation for the new estimate $\hat{N}_i$. Note that the derivative is a monotone function of N, we can numerically obtain $\hat{N}_i$ through bisection search.

2) Termination Condition: Using the $\delta-$method [23], we show in the Appendix that when i is large, $\hat{N}_i$ approximately follows the Gaussian distribution

$$\text{Norm}\left(N, \frac{\left(1-(1-p_i)^N\right)}{i(1-p_i)^N \ln^2(1-p_i)}\right).$$

The variance of $\hat{N}_i$ is

$$\text{Var}(\hat{N}_i) \approx \frac{1-(1-p_i)^N}{i(1-p_i)^N \ln^2(1-p_i)}. \tag{6}$$

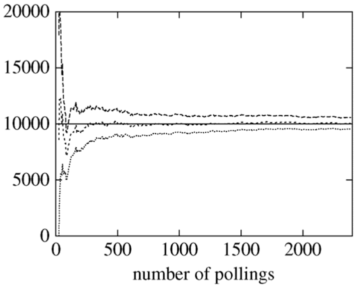


Fig. 1. Middle curve shows the estimated number of tags with respect to the number of pollings. The upper and lower curves show the confidence interval. The straight line shows the true number of tags.

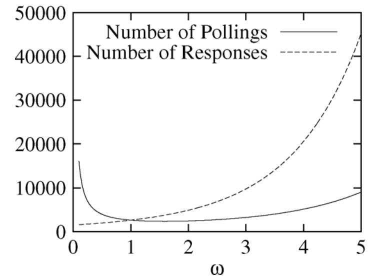


Fig. 2. Solid line shows the number of pollings with respect to ω when $\alpha = 95\%$ and $\beta = 5\%$. Dotted line shows the number of responses with respect to ω for the same parameter settings.

When N is large and p_i is small, we can approximate $(1-p_i)^N$ as e^{-Np_i} and $\ln(1-p_i)$ as p_i. The above variance becomes

$$\mathrm{Var}(\hat{N}_i) \approx \frac{e^{Np_i}-1}{ip_i^2}. \tag{7}$$

Hence, the confidence interval of N is

$$\hat{N}_i \pm Z_\alpha \cdot \sqrt{\frac{e^{\hat{N}_i p_i}-1}{ip_i^2}} \tag{8}$$

where Z_α is the α percentile for the standard Gaussian distribution. For example, when $\alpha = 95\%$, $Z_\alpha = 1.96$. Because N is undetermined, we use $\hat{N}_i$ as an approximation when computing the standard deviation in (8).

The termination condition for GMLE is therefore

$$Z_\alpha \cdot \sqrt{\frac{e^{\hat{N}_i p_i}-1}{ip_i^2}} \le \hat{N}_i \cdot \beta \tag{9}$$

where β is the error bound. The above inequality can be rewritten as

$$\sqrt{i} \ge \frac{Z_\alpha \sqrt{e^{\hat{N}_i p_i}-1}}{\hat{N}_i p_i \beta}. \tag{10}$$

When i is large, the estimation changes little from one polling to the next. Hence, $p_i = \omega/\hat{N}_{i-1} \approx \omega/\hat{N}_i$. We have

$$i \ge \frac{Z_\alpha^2 \cdot (e^\omega - 1)}{\omega^2 \beta^2}. \tag{11}$$

Hence, if ω is determined, we can theoretically compute the approximate number of pollings that is required in order to meet the accuracy requirement. For example, if $\alpha = 95\%$, $\beta = 5\%$, and $\omega = 1.594$ (which is the optimal value to be given shortly), 2372 pollings will be required. Note that (11) is independent with the actual number of tags, N. Hence, our approach has perfect scalability.

Fig. 1 shows the simulation result of GMLE when $N = 10\,000$, $\alpha = 95\%$, $\beta = 5\%$, and $\omega = 1.594$. The simulation setup can be found in Section VI. The middle curve is the estimated number of tags, $\hat{N}_i$, with respect to the number of pollings. It converges to the true value N represented by the central straight line. The upper and lower curves represent the 95% confidence interval, which shrinks as the number of pollings increases.

D. Determine the Value of ω

We demonstrate the impact of the value ω on two performance metrics: the *number of pollings* and the *number of tag responses* (i.e., the number of tag transmissions). The former measures the estimation time since each polling takes an equal amount of time for request/response exchange. The latter measures the energy cost because each response corresponds to one tag making one transmission in a slot.

1) Number of Pollings: According to (11), the number of pollings for meeting the accuracy requirement is $(Z_\alpha^2 \cdot (e^\omega - 1)/\omega^2\beta^2)$. To find its minimum value, we differentiate it with respect to ω and let the result be zero. Solving the equation, we have $\omega = 1.594$. Hence, the optimal value of p_i that minimizes the number of pollings is

$$p_i = \frac{1.594}{\hat{N}_{i-1}}. \tag{12}$$

2) Number of Responses: We count the total number of responses during the estimation process. After a small number of pollings, the estimation will closely approximate N (see Fig. 1). Hence, the expected number of responses for each polling is $Np_i \approx N_{i-1}p_i = \omega$. After $(Z_\alpha^2 \cdot (e^\omega - 1)/\omega^2\beta^2)$ pollings are made, the total number of responses is roughly

$$\frac{Z_\alpha^2 \cdot (e^\omega - 1)}{\omega^2\beta^2}\omega = \frac{Z_\alpha^2 \cdot (e^\omega - 1)}{\omega\beta^2}. \tag{13}$$

Our simulation results in Section VI demonstrate that the approximation in the above count is reasonably accurate. It is an increasing function with respect to ω, which means that a larger value of ω will lead to a larger number of responses. We give the intuition as follows: A larger ω means a larger contention probability and thus more collisions. Two or more responses in a collision slot produce the same amount of information as one response in a singleton slot (see further explanation in Section IV-F). In other words, in order to generate the necessary amount of information for meeting the accuracy requirement, more responses must be needed if there are more collisions.

3) Summary: In Fig. 2, we plot the number of pollings and the number of responses with respect to the value of ω. The number of pollings is minimized at $\omega = 1.594$. When ω is smaller than 1.594, its value controls the performance tradeoff between the two metrics. When we decrease ω, the energy cost (i.e., the number of responses) drops at the expenses of the estimation time (i.e., the number of pollings). Our further simulations in Section VI show that even at $\omega = 1.594$, the energy cost of GMLE is far below those of the existing protocols.

E. Request-Less Pollings

We observe that, after a number of pollings, the value of p_i will stay in a very small range and does not change much. It becomes unnecessary for the RFID reader to transmit it at each polling. Hence, we improve GMLE as follows: If the percentage change in p_i during a certain number M_1 of consecutive pollings is below a small threshold, the reader will broadcast a polling request, carrying the latest value of p_i, a flag indicating that it will no longer transmit polling requests for a certain number M_2 of slots, and the value of M_2. Without receiving further polling requests, the tags will respond with the same contention probability in the subsequent M_2 slots. This is called the *request-less pollings*. After M_2 slots, the reader will recalculate the contention probability, broadcast another polling request, carrying the new probability value, a flag, and M_2. This process repeats until the termination condition in (9) is met. With the threshold being 10%, $M_1 = 10$, and $M_2 = 50$, our simulation results show that the performance difference caused by request-less pollings is negligibly small even though the contention probability during request-less pollings may be slightly off the value set by (2). Request-less pollings can also be applied to the algorithm in Section V.

F. Information Loss Due to Collision

GMLE has a frame size of one slot. It obtains only binary information at each polling. No matter how many tags respond, the information that the reader receives is always the same, i.e., $z_i = 1$, which implies information loss when two or more tags decide to transmit at a polling. Let us compare two scenarios. In one scenario, only one tag responds at a polling. In the other, two tags respond. These two scenarios generate the same information but the energy cost of the second scenario is twice of the first. To address this issue, we design another algorithm that reduces the probability of collision and, moreover, compensate the impact of collision in its computation.

V. Enhanced Generalized Maximum Likelihood Estimation Algorithm

The *enhanced generalized maximum likelihood estimation* (EGMLE) algorithm is our second estimator for the number of RFID tags. It also utilizes history information from previous pollings and uses the maximum likelihood method to estimate the number of tags. However, instead of only obtaining binary information, it computes the number of responses in each polling. Because more information can be extracted, it is able to achieve much better energy efficiency than GMLE.

A. Overview

EGMLE uses the same polling protocol as GMLE does, except that its frame size f is larger than one in order to reduce the probability of collision. The result of the ith polling, x_i, is no longer a binary value. Instead, it is an estimate of the number of tags that respond during the polling.

EGMLE takes two steps to solve the collision problem. First, it increases the frame size f such that the tags that decide to respond at a polling are likely to respond at different slots in the frame. We pick values for p_i and f such that the collision probability is very small. Second, we compensate the remaining impact of collision in our computation.

EGMLE also consists of an *initialization phase* and an *iterative phase*. The initialization phase of EGMLE is the same as the initialization phase of GMLE, except that when the RFID reader obtains the first nonzero result x_l at the lth polling with a contention probability p_l, it computes a coarse estimation of N as (x_l/p_l). Then, it moves to the next phase, described as follows.

B. Iterative Phase

This phase iteratively refines the estimation after each polling and terminates when the specified accuracy requirement is met. The reader performs four tasks during the ith polling. First, it computes the contention probability before sending out the polling request

$$p_i = \frac{\omega}{\hat{N}_{i-1}} \tag{14}$$

where $\hat{N}_{i-1}$ is the estimate after the previous polling and ω is one by default. As we will show in Section V-C, performance tradeoff can be made by choosing other values for ω.

Second, the reader computes the number of responses x_i in the current frame.

Third, based on the received x_i and the history information, the reader computes the new estimate of N that maximizes the following likelihood function:

$$L_i = \prod_{j=l+1}^{i} \left[\frac{1}{\sqrt{2\pi N p_j (1-p_j)}} \cdot e^{-\frac{((1+\varepsilon)x_j - Np_j)^2}{2Np_j(1-p_j)}} \right] \tag{15}$$

where ε is introduced to compensate for collision and the iterative phase begins from the $(l+1)$th polling. The above formula and the value of ε will be derived shortly. The new estimate is

$$\hat{N}_i = \arg\max_N \{L_i\}. \tag{16}$$

Fourth, after computing $\hat{N}_i$, the reader determines if the estimate meets the accuracy requirement. In the following, we give the details of the above tasks.

1) Compute the Number of Responses: At the ith polling, the reader measures the number of nonempty slots in the frame, denoted as x_i, which is an integer in the range of $[0 \ldots f]$. Due to possible collision, the actual number of responses, denoted as x_i^*, can be greater. Let $x_i^* = (1+\varepsilon)x_i$. The value of ε is determined as follows.

Since each tag independently decides to respond with probability p_i, x_i^* follows a binomial distribution, $\mathrm{Bino}(N, p_i)$, i.e.,

$$\mathrm{Prob}\,\{x_i^* = k\} = \binom{N}{k} p_i^k (1-p_i)^{N-k}. \tag{17}$$

Suppose ω takes the default value, 1. When i is large, N_{i-1} approximates N and thus $p_i \approx 1/N$. If N is sufficiently large, $\mathrm{Prob}\{x_i^* = 2\} \approx 0.1839$, $\mathrm{Prob}\{x_i^* = 3\} \approx 0.0613$,

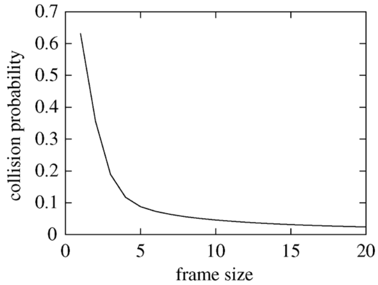


Fig. 3. Collision probability with respect to the frame size f.

$\text{Prob}\{x_i^* = 4\} \approx 0.0153$, and the probability decreases exponentially with respect to k. $\text{Prob}\{x_i^* > 4\}$ is only about 0.0037.

Next, we compute the probability for collision to happen at the ith polling, which is denoted as $\text{Prob}_i\{\text{collision}\}$.

$$\begin{aligned}
&\text{Prob}_i\{\text{collision}\} \\
&= \sum_{k=2}^{N} \text{Prob}_i \left\{\text{collision}|x_i^* = k\right\} \times \text{Prob}\left\{x_i^* = k\right\} \\
&= \sum_{k=2}^{f} \left(1 - \frac{P(f,k)}{f^k}\right) \times \text{Prob}\left\{x_i^* = k\right\} \\
&\quad + \sum_{k=f+1}^{N} 1 \times \text{Prob}\left\{x_i^* = k\right\}
\end{aligned}$$

where $P(f,k) = (f!/(f-k)!)$ is the permutation function. Fig. 3 shows the collision probability $\text{Prob}_i\{\text{collision}\}$ with respect to f. It diminishes quickly as f increases. When $f = 10$ (which is what we use in the simulations), $\text{Prob}_i\{\text{collision}\}$ is just 0.046. With such a small probability, the chance for more than two tags involved in a collision or more than one collision at a polling is exceedingly small and thus ignored. Therefore, to approximate x_i^*, we multiply x_i by 1.046 to compensate the impact of collision. Namely, $\varepsilon = 0.046$.

2) Compute the Value of $\hat{N}_i$: Recall that the iterative phase starts at the $(l+1)$th polling. After the ith polling, the reader has collected the values of $x_j, l < j \leq i$. By our previous analysis, we know that $x_i^* = (1+\varepsilon)x_i$ and it follows a binomial distribution $\text{Bino}(N, p_j)$. When N is large enough, the binomial distribution can be closely approximated by a Gaussian distribution $\text{Norm}(\mu_j, \sigma_j)$ with parameters $\mu_j = Np_j$ and $\sigma_j = \sqrt{Np_j(1-p_j)}$. Namely

$$x_j^* \approx (1+\varepsilon)x_j \sim \text{Norm}\left(Np_j, Np_j(1-p_j)\right). \tag{18}$$

Hence, the probability for the *measured number of responses*, $(1+\varepsilon)x_j$, to occur under this distribution is $(1/\sqrt{2\pi Np_j(1-p_j)}) \cdot \exp[-(((1+\varepsilon)x_j - Np_j)^2/2Np_j(1-p_j))]$. The likelihood function for all measured numbers of responses in the pollings, $(1+\varepsilon)x_j, l < j \leq i$, to occur is

$$L_i = \prod_{j=l+1}^{i} \left[\frac{1}{\sqrt{2\pi Np_j(1-p_j)}} \cdot e^{-\frac{((1+\varepsilon)x_j - Np_j)^2}{2Np_j(1-p_j)}}\right]. \tag{19}$$

Our goal is to find the value $\hat{N}_i$ that maximizes the likelihood function. We first take logarithm on both sides of (19).

$$\ln(L_i) = \sum_{j=l+1}^{i} \left[\ln \frac{1}{\sqrt{2\pi Np_j(1-p_j)}} - \frac{((1+\varepsilon)x_j - Np_j)^2}{2Np_j(1-p_j)}\right]. \tag{20}$$

We then differentiate both sides.

$$\begin{aligned}
\frac{\partial \ln(L_i)}{\partial N} &= \sum_{j=l+1}^{i} \left[-\frac{1}{2N} + \frac{(1+\varepsilon)^2 x_j^2 - (Np_j)^2}{2N^2 p_j(1-p_j)}\right] \\
&= \sum_{j=l+1}^{i} \frac{(1+\varepsilon)^2 x_j^2 - (Np_j)^2}{2N^2 p_j(1-p_j)} - \frac{i-l}{2N}.
\end{aligned} \tag{21}$$

Finally, we set the right side to be zero and numerically compute the value of $\hat{N}_i$.

3) Termination Condition: The *fisher information*[1] $\mathcal{I}(\hat{N}_i)$ of L_i is defined as follows:

$$\mathcal{I}(\hat{N}_i) = -E\left[\frac{\partial^2 \ln(L_i)}{\partial N^2}\right]. \tag{22}$$

According to (21), we have

$$\begin{aligned}
\mathcal{I}(\hat{N}_i) &= E\left[\sum_{j=l+1}^{i} \frac{(1+\varepsilon)^2 x_j^2}{N^3 p_j(1-p_j)} - \frac{i-l}{2N^2}\right] \\
&= \sum_{j=l+1}^{i} \frac{(Np_j)^2 + Np_j(1-p_j)}{N^3 p_j(1-p_j)} - \frac{i-l}{2N^2} \qquad (23) \\
&= \sum_{j=l+1}^{i} \frac{p_j}{N(1-p_j)} + \frac{i-l}{2N^2}. \qquad (24)
\end{aligned}$$

Above, we have applied $E((1+\varepsilon)^2 x_j^2) = (Np_j)^2 + Np_j(1-p_j)$ in (23) because $(1+\varepsilon)x_j \sim \text{Norm}(Np_j, Np_j(1-p_j))$ and $E(x^2) = (E(x))^2 + \text{Var}(x)$.

Following the classical theory for MLE, when i is sufficiently large, the distribution of $\hat{N}_i$ is approximated by

$$\text{Norm}\left(N, \frac{1}{\mathcal{I}(\hat{N}_i)}\right). \tag{25}$$

Hence, the confidence interval is

$$\hat{N}_i \pm Z_\alpha \cdot \sqrt{\frac{1}{\mathcal{I}(\hat{N}_i)}}. \tag{26}$$

Note that we use $\hat{N}_i$ as an approximation for N in the computation when necessary since N is unknown. The termination condition for EGMLE to achieve the required accurary is

$$Z_\alpha \cdot \sqrt{\frac{1}{\mathcal{I}(\hat{N}_i)}} \leq \hat{N}_i \cdot \beta. \tag{27}$$

[1]The fisher information [24] is a way of measuring the amount of information that an observable random variable x carries about an unknown parameter θ upon which the likelihood function of θ, $L(\theta) = f(x;\theta)$, depends.

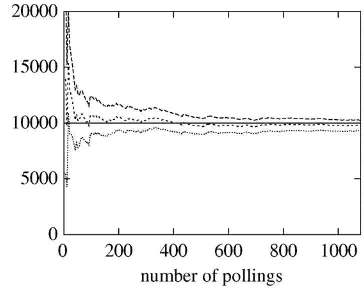


Fig. 4. Middle curve shows the estimated number of tags with respect to the number of pollings. Upper and lower curves show the confidence interval. Straight line shows the true number of tags.

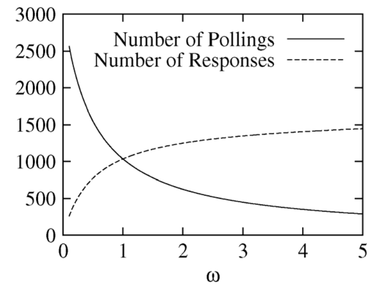


Fig. 5. Solid line shows the number of pollings with respect to ω when $\alpha = 95\%$ and $\beta = 5\%$. Dotted line shows the number of responses with respect to ω for the same parameter settings.

Fig. 4 shows the simulation result of EGMLE when $N = 10\,000$, $\alpha = 95\%$, $\beta = 5\%$, and $\omega = 1$. The middle curve is the value of $\hat{N}_i$, which converges to the value of N represented by the central straight line. The upper and lower curves represent the 95% confidence interval, which shrinks as the number of pollings increases. The algorithm terminates after 1081 pollings.

C. Performance Tradeoff

According to (14), the contention probability is proportional to ω. We study how the value of ω controls the tradeoff between the estimation time and the energy cost, which are measured by the number of pollings and the number of responses, respectively.

1) Number of Pollings: Since the MLE approach provides statistically consistent estimate, when i is large, (24) can be approximated as follows:

$$\begin{aligned}\mathcal{I}(\hat{N}_i) &= \sum_{j=l+1}^{i} \frac{p_j}{N(1-p_j)} + \frac{i-l}{2N^2}\\ &\approx \left(\frac{p_i}{N(1-p_i)} + \frac{1}{2N^2}\right)\cdot(i-l)\\ &\approx \frac{2Np_i+1}{2N^2}\cdot(i-l) \qquad (28)\end{aligned}$$

where $p_i \ll 1$. According to (27), we have

$$\mathcal{I}(\hat{N}_i) \geq \left(\frac{Z_\alpha}{\hat{N}_i\cdot\beta}\right)^2. \qquad (29)$$

Equations (28) and (29) give us the following inequality:

$$\begin{aligned}\frac{2Np_i+1}{2N^2}\cdot(i-l) &\geq \left(\frac{Z_\alpha}{\hat{N}_i\cdot\beta}\right)^2\\ i &\geq \frac{2Z_\alpha^2}{(2\omega+1)\beta^2} \qquad (30)\end{aligned}$$

where $\hat{N}_i \approx N$ and $l \ll i$. Hence, the number of pollings it takes to achieve the accuracy requirement is $2Z_\alpha^2/(2\omega+1)\beta^2$.

The solid line in Fig. 5 shows the number of pollings with respect to ω when $\alpha = 95\%$ and $\beta = 5\%$. It is a decreasing function in ω. The reason is that a larger ω results in more responses (and thus more information) in each polling. Consequently, a less number of pollings is needed to achieve a certain accuracy requirement.

2) Number of Responses: When i is large, the expected number of responses for each polling is $Np_i \approx N_{i-1}p_i = \omega$. After $2Z_\alpha^2/(2\omega+1)\beta^2$ pollings are made, the total number of responses is roughly

$$\frac{Z_\alpha^2\cdot(e^\omega-1)}{\omega^2\beta^2}\omega = \frac{Z_\alpha^2\cdot(e^\omega-1)}{\omega\beta^2}. \qquad (31)$$

The dotted line in Fig. 5 shows the number of responses with respect to ω when $\alpha = 95\%$ and $\beta = 5\%$. It is an increasing function in ω, which means that a larger value of ω will lead to a larger number of responses.

3) Summary: Fig. 5 demonstrates the performance tradeoff under different values of ω. As we decrease ω, EGMLE achieves better energy efficiency by requiring a fewer number of responses, at the expense of time efficiency by requiring a larger number of pollings.

VI. Simulation Result

We evaluate the performance of GMLE and EGMLE by simulations. In order to demonstrate the performance tradeoff between energy cost and estimation time, we choose two different contention probability parameters for each of the two algorithms. We use $\omega = 0.5$ and 1.594 for GMLE, i.e., $p_i = 0.5/\hat{N_{i-1}}$ and $1.594/\hat{N_{i-1}}$. Note that 1.594 is the optimal value of ω for time efficiency in GMLE. We denote the corresponding variants of the algorithm as GMLE(0.5) and GMLE(1.594).

For EGMLE, Fig. 5 shows that the number of pollings and the number of responses are both monotonic functions with respect to ω, which means there is no optimal ω for either energy efficiency or time efficiency. We choose $\omega = 0.5$ and 1.0 for EGMLE, i.e., $p_i = 0.5/\hat{N_{i-1}}$ and $1.0/\hat{N_{i-1}}$. The corresponding variants of the algorithm are denoted as EGMLE(0.5) and EGMLE(1.0). Section V-B shows how to compute the compensation parameter ε for EGMLE(1.0), which is 0.046. Following the same steps, we obtain $\varepsilon = 0.012$ for EGMLE(0.5). We compare the proposed algorithms to the state-of-the-art algorithms in the related work. They are the UPE [3] and the EZB estimator [4]. The original UPE, denoted as UPE-O, is very energy-inefficient because its contention probability begins from 100%, and thus all tags will respond. We modify it (denoted as UPE-M) to begin from a small initial contention probability $1/N_{\max}$ and keep the remaining part of UPE-O. This section shows the performance of both UPE-O and UPE-M. We run each simulation 100 times and average the outcomes.

TABLE I
NUMBER OF RESPONSES WHEN $\alpha = 90\%$, $\beta = 9\%$

N	Total number of responses						
	GMLE(0.5)	GMLE(1.594)	EGMLE(0.5)	EGMLE(1.0)	UPE-O	UPE-M	EZB
5000	432S	767 S	172 S	225 S	6345 L	709 L	4342 S
10000	414S	832 S	180 S	231 S	11986 L	899 L	8683 S
20000	402S	844 S	186 S	213 S	22895 L	977 L	17366 S

TABLE II
NUMBER OF RESPONSES WHEN $\alpha = 90\%$, $\beta = 6\%$

N	Total number of responses						
	GMLE(0.5)	GMLE(1.594)	EGMLE(0.5)	EGMLE(1.0)	UPE-O	UPE-M	EZB
5000	1041 S	1855 S	402 S	523 S	7144 L	1811 L	7236 S
10000	1153 S	1924 S	414 S	519 S	12645 L	1687 L	14472 S
20000	1015 S	1797 S	375 S	503 S	23808 L	1814 L	28944 S

TABLE III
NUMBER OF RESPONSES WHEN $\alpha = 90\%$, $\beta = 3\%$

N	Total number of responses						
	GMLE(0.5)	GMLE(1.594)	EGMLE(0.5)	EGMLE(1.0)	UPE-O	UPE-M	EZB
5000	3927S	7341 S	1499 S	2037 S	12664 L	6426 L	27497 S
10000	3760S	7339 S	1489 S	2059 S	18023 L	6581 L	54993 S
20000	3783S	7350 S	1543 S	2002 S	28708 L	6993 L	109987 S

TABLE IV
NUMBER OF RESPONSES WHEN $\alpha = 95\%$, $\beta = 9\%$

N	Total number of responses						
	GMLE(0.5)	GMLE(1.594)	EGMLE(0.5)	EGMLE(1.0)	UPE-O	UPE-M	EZB
5000	603S	1112 S	258 S	330 S	6715 L	1073 L	4342 S
10000	669S	1120 S	247 S	304 S	12062 L	961 L	8683 S
20000	680S	1197 S	262 S	320 S	23345 L	1136 L	17366 S

TABLE V
NUMBER OF RESPONSES WHEN $\alpha = 95\%$, $\beta = 6\%$

N	Total number of responses						
	GMLE(0.5)	GMLE(1.594)	EGMLE(0.5)	EGMLE(1.0)	UPE-O	UPE-M	EZB
5000	1340 S	2515 S	581 S	736 S	7712 L	2598 L	10130 S
10000	1354 S	2511 S	596 S	736 S	13477 L	2318 L	20261 S
20000	1381 S	2630 S	555 S	749 S	24631 L	2510 L	40521 S

In the initialization phase of our algorithms, let $N_{\max} = 1\,000\,000$ and $C = 2$. The frame size in EGMLE(0.5) and EGMLE(1.0) is 10 slots. The parameters for UPE and EZB are chosen based on the original papers whenever possible. All algorithms except for UPE need only to identify empty and nonempty slots. To set a nonempty slot apart from an empty slot, a tag only needs to respond with a short bit string (one bit) to make the channel busy. UPE has to identify empty, singleton, and collision slots. To set a singleton slot apart from a collision slot, many more bits (10 used by UPE) are necessary [25]. For example, CRC may be used to detect collision.

The energy cost of an algorithm depends on: 1) the number of responses that all tags transmit before the algorithm terminates; and 2) the size of each response. We use "S" to mean that the response is a short bit string (in the empty/non-empty case), and "L" to mean a long bit string (in the empty/singleton/collision case).

We do not include the simulation results for LoF [5] because its energy cost is much higher than others. Its number of responses transmitted by the tags is kN, where k is the number of frames used in the estimation process.

A. Number of Responses

The first simulation studies the number of responses in each algorithm with respect to N, α, and β. Table I shows the number of responses with respect to N when $\alpha = 90\%$ and $\beta = 9\%$. The proposed algorithms require fewer responses than UPE and EZB. As predicted, UPE-O is energy-inefficient; UPE-M works much better. The best algorithm is EGMLE(0.5), whose number of responses is about one fifth of what UPE-M requires and one ninetieth of what EZB requires when N is 20 000. Moreover, each response in UPE is much longer.

GMLE(0.5) has a smaller energy cost than GMLE(1.594). For example, $N = 10\,000$, the ratio between the number of responses by GMLE(1.594) and that by GMLE(0.5) is 2.01, which is close to the theoretically computed ratio of 1.90 in Fig. 2. Similarly, EGMLE(0.5) is more energy-efficient than EGMLE(1.0). When $N = 10\,000$, the ratio between the number of responses by GMLE(1.594) and that by GMLE(0.5) is 1.28, which is also close to the theoretical value of 1.34 in Fig. 5.

We vary α from 90% to 95% and to 99%, and vary β from 9% to 6% and to 3%. Tables II–IX show similar comparison under

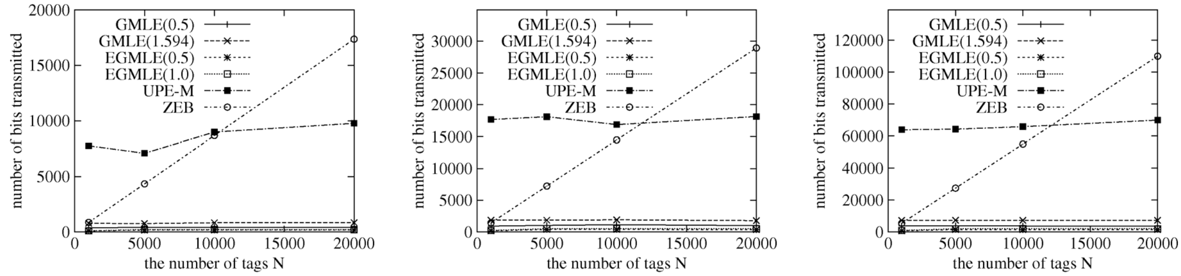


Fig. 6. Numbers of bits transmitted when $\alpha = 90\%$, $\beta =$ 9%, 6%, and 3%.

TABLE VI
NUMBER OF RESPONSES WHEN $\alpha = 95\%$, $\beta = 3\%$

N	Total number of responses						
	GMLE(0.5)	GMLE(1.594)	EGMLE(0.5)	EGMLE(1.0)	UPE-O	UPE-M	EZB
5000	5687 S	10493 S	2181 S	2915 S	14678 L	8858 L	39074 S
10000	5673 S	10286 S	2267 S	2924 S	20845 L	9364 L	78148 S
20000	5588 S	10637 S	2217 S	2990 S	32339 L	9683 L	156297 S

TABLE VII
NUMBER OF RESPONSES WHEN $\alpha = 99\%$, $\beta = 9\%$

N	Total number of responses						
	GMLE(0.5)	GMLE(1.594)	EGMLE(0.5)	EGMLE(1.0)	UPE-O	UPE-M	EZB
5000	1040 S	2162 S	427 S	453 S	7240 L	1726 L	7236 S
10000	1071 S	2135 S	416 S	529 S	12842 L	1906 L	14472 S
20000	1017 S	1916 S	439 S	573 S	23982 L	1819 L	28944 S

TABLE VIII
NUMBER OF RESPONSES WHEN $\alpha = 99\%$, $\beta = 6\%$

N	Total number of responses						
	GMLE(0.5)	GMLE(1.594)	EGMLE(0.5)	EGMLE(1.0)	UPE-O	UPE-M	EZB
5000	2527 S	4785 S	965 S	1269 S	9679 L	4311 L	17366 S
10000	2527 S	4637 S	973 S	1248 S	15336 L	4130 L	34733 S
20000	2440 S	4580 S	991 S	1293 S	26128 L	4044 L	69465 S

TABLE IX
NUMBER OF RESPONSES WHEN $\alpha = 99\%$, $\beta = 3\%$

N	Total number of responses						
	GMLE(0.5)	GMLE(1.594)	EGMLE(0.5)	EGMLE(1.0)	UPE-O	UPE-M	EZB
5000	9693 S	18690 S	3818 S	4993 S	21823 L	16705 L	65124 S
10000	9606 S	18223 S	3791 S	4998 S	27667 L	15882 L	130247 S
20000	9385 S	17735 S	3847 S	5027 S	38935 L	16471 L	260495 S

different values of α and β values. In all cases, the number of responses increases when α increases or β decreases, and except for EZB, the number does not vary much with respect to N, meaning that all algorithms except for EZB achieve good scalability. The ratio between the numbers for different algorithms appears to be quite stable under different parameter settings.

B. Total Number of Bits Transmitted

The second simulation evaluates the energy cost of the algorithms. As mentioned before, one bit is enough to separate empty/nonempty slot. Hence, the response of GMLE, EGMLE, and EZB is one bit long. A response in UPE-M is 10 bits long [3]. We compare the total number of bits transmitted by all tags before each algorithm terminates. We omit the results for UPE-O, which are much worse than the results of UPE-M. Fig. 6 shows the simulation results with respect to N when $\alpha = 90\%$, $\beta =$ 9%, 6% and 3%. For example, when $\alpha = 90\%$, $\beta = 3\%$, and $N = 20\,000$, the ratio between the number of bits transmitted by UPE-M (EZB) and that by our best estimator EGMLE(0.5) is 45.32 (71.28). Figs. 7 and 8 show the comparison under different β values when $\alpha = 95\%$ and 99%, respectively. Their results are similar to Fig. 6. It should be noted that the number of bits transmitted is not an accurate measurement of the energy cost because it ignores the energy spent to power up the radio and synchronize with the reader. However, combining the number of bits and the number of transmissions (in Section VI-A) still gives a good idea on how energy-efficient each algorithm is.

C. Estimation Time

The third simulation compares the time it takes for each algorithm to complete the estimation of N. Based on the specification of the Philips I-Code system [22], after the required waiting times (e.g., gap between transmissions) are included, it can be calculated that an RFID reader needs 0.4 ms to detect an empty slot, 0.8 ms to detect a collision or a singleton slot, and

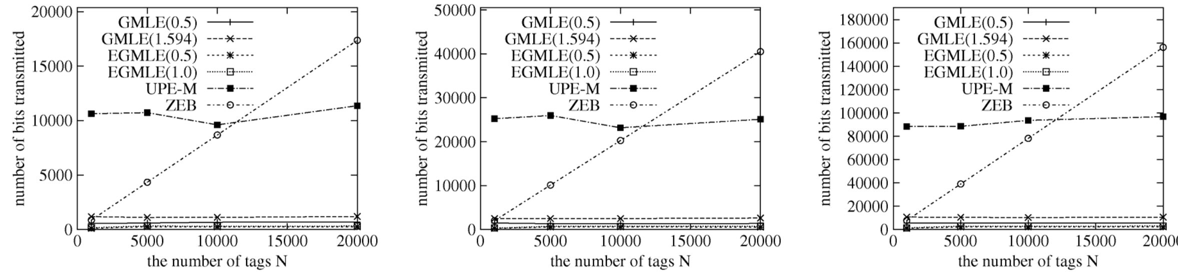


Fig. 7. Numbers of bits transmitted when $\alpha = 95\%$, $\beta =$ 9%, 6%, and 3%.

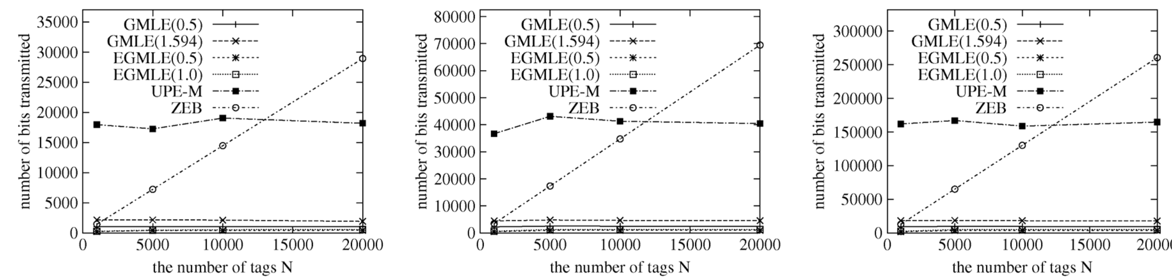


Fig. 8. Numbers of bits transmitted when $\alpha = 99\%$, $\beta = hbox9\%$, 6%, and 3%.

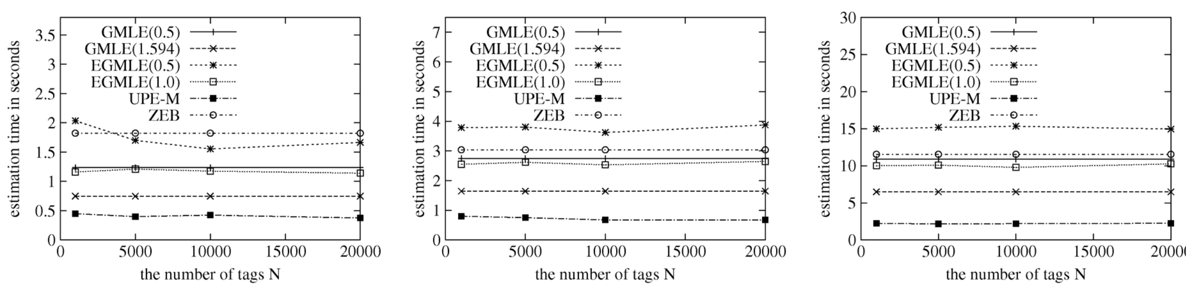


Fig. 9. Estimation times of the algorithms when $\alpha = 90\%$, $\beta =$ 9%, 6%, and 3%.

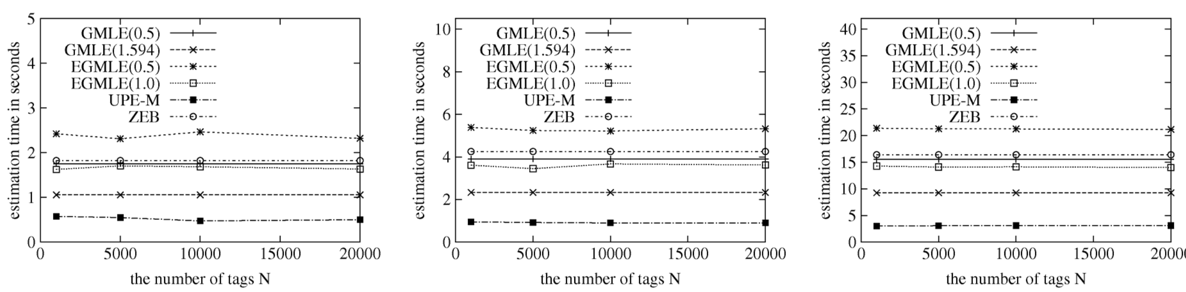


Fig. 10. Estimation times of the algorithms when $\alpha = 95\%$, $\beta =$ 9%, 6%, and 3%.

1 ms to broadcast a polling request. Hence, GMLE, EGMLE, and EZB require a slot length of 0.4 ms, while UPE-M requires a slot length of 0.8 ms. Recall that the contention probability takes the form of $\omega/\hat{N}_i$, where ω is a known constant. Thus, the reader transmits $\hat{N}_i$ instead of the actual probability value in the polling requests. If we assume $N_{\max}$ is no more than a million, then 20 bits for $\hat{N}_i$ is sufficient. GMLE has a fixed frame size of one slot. EGMLE has a fixed frame size of 10 slots. EZB and UPE-M also have predetermined frame sizes. Let $\alpha = 90\%$, $\beta =$ 9%, 6%, and 3%. The three plots in Fig. 9 show the estimation times of the algorithms with respect to the number of tags in the deployment. The times grow very slowly as the number of tags increases, which suggests the algorithms all scale well. In the first plot of Fig. 9, UPE-M takes the least amount of time, only about 0.5 s, to estimate 20 000 tags, while the other algorithms take between 0.7–2.0 s. GMLE(1.594) takes less estimation time than GMLE(0.5), and the ratio is 0.61, which is consistent with the theoretical value of 0.58 in Fig. 2. Similarly, EGMLE(1.0) takes less time than EGMLE(0.5), and the ratio is 0.68, which is also consistent with the theoretical value of 0.67 in Fig. 5. Figs. 10 and 11 show similar simulation results when $\alpha = 95\%$ and 99%, respectively. Even though the new algorithms take longer to complete, their estimation time is still small. We believe the extra time needed can be well justified for the large energy saving.

There exists a performance tradeoff between GMLE and EGMLE. In Sections VI-A and VI-B, we have examined energy cost in terms of number of responses and number of transmitted

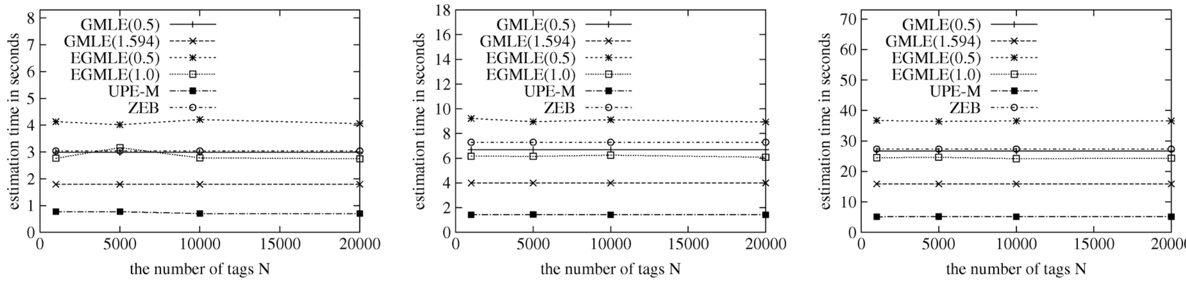


Fig. 11. Estimation times of the algorithms when $\alpha = 99\%$, $\beta =$ 9%, 6%, and 3%.

bits. EGMLE always performs better than GMLE. In this section, we compare estimation time of our two methods. GMLE performs better than EGMLE. Because this paper's focus is on energy efficiency, we regard EGMLE as our best estimator for energy saving.

VII. Conclusion

This paper proposes two probabilistic algorithms for estimating the number of RFID tags in a region. We believe the algorithms are the first of its kind that targets at prolonging the lifetime of the active RFIDs. Their energy cost is far less than the state-of-the-art algorithms in the related work. Moreover, we reveal a fundamental tradeoff between the energy cost and the estimation time. By tuning a system parameter, the algorithms can trade longer estimation time for less energy cost, or vice versa.

Appendix
Distribution and Variance of $\hat{N}_i$

Let i be a large positive integer. Consider the sequence of Bernoulli random variables, Z_j, $1 \leq j \leq i$, whose success probability is $q = 1 - (1 - p_i)^N$. Let $\hat{q} = (\sum_{j=1}^{i} Z_j)/i$, which is the estimation of the success probability q. It is known that asymptotically $\hat{q}$ follows a normal distribution

$$\hat{q} \sim \text{Norm}\left(q, \frac{q(1-q)}{i}\right). \tag{32}$$

Because the MLE approach provides statistically consistent estimate, when i is large, we can consider the contention probabilities in the later stage of the pooling process to be approximately a constant. In addition, the number of polling results before stabilization of the contention probability is limited, and their impact will diminish as i becomes large. That is, they can be ignored when the asymptotic property of $\hat{N}_i$ is considered. Hence, for the asymptotic property, we can let $p_j = p_i$, for $1 \leq j \leq i$, and (5) becomes

$$\frac{\partial \ln(L_i)}{\partial N} = \ln(1-p_i)\left[\left(i - \sum_{j=1}^{i} Z_j\right) - \frac{(1-p_i)^N}{1-(1-p_i)^N}\sum_{j=1}^{i} Z_j\right]. \tag{33}$$

Therefore, the MLE $\hat{N}_i$ that solves $(\partial \ln(L_i)/\partial N) = 0$ satisfies

$$(1-p_i)^{\hat{N}_i} = 1 - \left(\sum_{j=1}^{i} Z_j\right)/i = 1 - \hat{q}. \tag{34}$$

Hence, from (32), $(1 - p_i)^{\hat{N}}$ asymptotically follows the following normal distribution:

$$\text{Norm}\left((1-p_i)^N, \frac{\left(1-(1-p_i)^N\right)(1-p_i)^N}{i}\right). \tag{35}$$

According to the δ-method [23], if a random variable X_i satisfies

$$X_i \xrightarrow{D} \text{Norm}\left(\theta, \frac{\sigma^2}{i}\right) \tag{36}$$

where θ and σ are finite constants and $\xrightarrow{D}$ means convergence in distribution, then we must have

$$g(X_i) \xrightarrow{D} \text{Norm}\left(g(\theta), \frac{\sigma^2 \left[g'(\theta)\right]^2}{i}\right) \tag{37}$$

for any function g such that $g'(\theta)$ exists and takes a nonzero value. Based on (36) and (37), taking the logarithm of (35), we have

$$\hat{N}_i \cdot \ln(1-p_i) \sim \text{Norm}\left(N\ln(1-p_i), \frac{\left(1-(1-p_i)^N\right)}{i(1-p_i)^N}\right). \tag{38}$$

That is

$$\hat{N}_i \sim \text{Norm}\left(N, \frac{\left(1-(1-p_i)^N\right)}{i(1-p_i)^N \ln^2(1-p_i)}\right). \tag{39}$$

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] R. Hollinger and J. Davis, "National retail security survey," 2001 [Online]. Available: http://diogenesllc.com/NRSS_2001.pdf
[2] R. Want, "An introduction to RFID technology," *IEEE Pervasive Comput.*, vol. 5, no. 1, pp. 25–33, Jan. 2006, in *Proc. IEEE PerCom*.
[3] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. ACM MobiCom*, Los Angeles, CA, 2006, pp. 322–333.
[4] M. Kodialam, T. Nandagopal, and W. Lau, "Anonymous tracking using RFID tags," in *Proc. IEEE INFOCOM*, 2007, pp. 1217–1225.
[5] C. Qian, H. Ngan, and Y. Liu, "Cardinality estimation for large-scale RFID systems," in *Proc. IEEE PerCom*, 2008, pp. 30–39.
[6] V. Namboodiri and L. Gao, "Energy-Aware tag anti-collision protocols for RFID systems," in *Proc. IEEE PerCom*, 2007, pp. 23–36.

[7] D. Klair, K. Chin, and R. Raad, "On the energy consumption of pure and slotted aloha based RFID anti-collision protocols," *Comput. Commun.*, vol. 32, no. 5, pp. 961–973, 2008.

[8] K. Hwang, B. Vander-Zanden, and H. Taylor, "A linear-time probabilistic counting algorithm for database applications," *Trans. Database Syst.*, vol. 15, no. 2, pp. 208–229, Jun. 1990.

[9] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. IEEE PerCom*, 2002, pp. 98–113.

[10] J. Zhai and G. N. Wang, "An anti-collision algorithm using two-functioned estimation for RFID tags," in *Proc. ICCSA*, 2005, pp. 702–711.

[11] J. Cha and J. Kim, "Novel anti-collision algorithms for fast object identification in RFID system," in *Proc. IEEE ICPADS*, 2005, vol. 2, pp. 63–67.

[12] D. Hush and C. Wood, "Analysis of tree algorithm for RFID arbitration," in *Proc. IEEE ISIT*, 1998, p. 107.

[13] J. Myung and W. Lee, "An adaptive memoryless tag anti-collision protocol for RFID networks," in *Proc. IEEE ICC*, 2005, pp. 10–15.

[14] H. Choi, J. Cha, and J. Kim, "Fast wireless anti-collision algorithm in ubiquitous ID system," in *Proc. IEEE VTC*, Sep. 2004, vol. 6, pp. 4589–4592.

[15] L. Pan and H. Wu, "Smart trend-traversal: A low delay and energy tag arbitration protocol for large RFID systems," in *Proc. IEEE INFOCOM*, 2009, pp. 2571–2575.

[16] C. Floerkemeier, "Transmission control scheme for fast RFID object identification," in *Proc. PerCom Workshops Pervasive Wireless Netw.*, 2006, pp. 467–462.

[17] C. Floerkemeier and M. Wille, "Comparison of transmission schemes for framed ALOHA based RFID protocols," in *Proc. SAINT Workshops RFID Extended Netw. Deployment Technol. Appl.*, 2006, pp. 94–97.

[18] M. Durand and P. Flajolet, "LogLog counting of large cardinalities," in *Proc. Eur. Symp. Algor.*, 2003, pp. 605–617.

[19] H. Han, B. Sheng, C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID tags efficiently and anonymously," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[20] C. C. Tan, B. Sheng, and Q. Li, "How to monitor for missing RFID tags," in *Proc. IEEE ICDCS*, Jun. 2008, pp. 295–302.

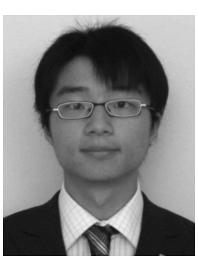
[21] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. ACM MobiHoc*, 2010, pp. 1–10.

[22] Philips Semiconductors, Eindhoven, The Netherlands, "I-CODE smart label RFID tags," Jan. 2004 [Online]. Available: http://www.nxp.com/acrobat_download/other/identification/SL092030.pdf

[23] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. Pacific Grove, CA: Duxbury, 2002.

[24] E. L. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed. New York: Springer, 1998.

[25] EPCglobal, "EPC radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz–960 MHz version 1.0.9," 2005 [Online]. Available: http://www.epcglobalinc.org/standards/uhfc1g2/uhfc1g2_1_0_9-standard-2005% 0126.pdf



Tao Li received the B.S. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2007, and is currently pursuing the Ph.D. degree in computer and information science and engineering at the University of Florida, Gainesville.

His advisor is Dr. Shigang Chen, and his research interests include network traffic measurement and RFID technologies.

Samuel S. Wu received the Ph.D. degree in statistics from Cornell University, Ithaca, NY, in 1998.

He is an Associate Professor and Interim Chair of the Department of Biostatistics, University of Florida, Gainesville. He has published on optimal sequential allocation strategy derived for missile defense system and numerous journal articles at the cutting edge of adaptive clinical trial design. He has extensive research experience on statistical modeling and simultaneous statistical inference methods, especially for sequential experiment designs.

Shigang Chen received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 1993, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana–Champaign in 1996 and 1999, respectively, all in computer science.

After graduation, he worked with Cisco Systems for three years before joining the University of Florida, Gainesville, in 2002, where he is currently an Associate Professor with the Department of Computer and Information Science and Engineering. His research interests include wireless networks, network protocols and algorithms, and distributed computing.

Dr. Chen is an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and *Computer Networks*. He served as a TPC Co-Chair for IEEE IWQoS 2009 and the Computer and Network Security Symposium of IEEE IWCCC 2006, a Vice TPC Chair for IEEE MASS 2005, a Vice General Chair for QShine 2005, a TPC Co-Chair for QShine 2004, and a TPC member for many conferences including IEEE ICNP, IEEE INFOCOM, IEEE ICDCS, IEEE ICC, IEEE GLOBECOM, etc. He received the IEEE Communications Society Best Tutorial Paper Award in 1999 and the NSF CAREER Award in 2007.

Mark C. K. Yang passed away on October 24, 2010. He received the Ph.D. degree in statistics from the University of Wisconsin—Madison.

He was a Professor of statistics with the University of Florida, Gainesville. He authored two books and over 100 peer-reviewed publications in statistical methodology and applied probability, genetics-related research, and other statistical applications.

Prof. Yang was a Fellow of the American Statistical Association.