

Cardinality Estimation for Large-scale RFID Systems

Chen Qian, *Student Member, IEEE*, Hoilun Ngan, *Student Member, IEEE*,
 Yunhao Liu, *Senior Member, IEEE*, Lionel M. Ni, *Fellow, IEEE*,

Abstract—Counting the number of RFID tags (cardinality) is a fundamental problem for large-scale RFID systems. Not only does it satisfy some real application requirements, it also acts as an important aid for RFID identification. Due to the extremely long processing time, slotted ALOHA-based or tree-based arbitration protocols are often impractical for many applications, because tags are usually attached to moving objects and they may have left the reader's interrogation region before being counted. Recently, estimation schemes have been proposed to count the approximate number of tags. Most of them, however, suffer from two scalability problems: time inefficiency and multiple-reading. Without resolving these problems, large-scale RFID systems cannot easily apply the estimation scheme as well as the corresponding identification. In this paper, we present the Lottery Frame (LoF) estimation scheme, which can achieve high accuracy, low latency, and scalability. LoF estimates the tag numbers by utilizing the collision information. We show the significant advantages, e.g., high accuracy, short processing time and low overhead, of the proposed LoF scheme through analysis and simulations.

Index Terms—RFID Systems, Collision Resolution, Tag Estimation, ALOHA Networks.

1 INTRODUCTION

RADIO Frequency Identification (RFID) technology [1] has been widely used for many applications, such as localization [2] [3], objects tracking [4], activity monitoring [5], access control and security [6]. An RFID system typically consists of three components: readers, tags and the middleware software [7]. RFID readers with antennas are devices used to read or write data from/to RFID tags. RFID tags are labeled in designated objects where each tag has a small size of memory to store its unique serial number (ID) as well as other information. The simple structure and cheap price offer promising advantages for the applications of large-volume objects in a mobile environment.

Cardinality estimation, i.e., counting the approximate number of tags in a given region is one of the most important tasks in large-scale RFID systems. An estimation scheme can be applied to applications in which the user wants to know the population information of objects having the same type of identities, e.g., an intelligent transportation systems (ITS) [31] that tracks the population distribution of metropolitan vehicles, an indoor stadium system that monitors visitors or a factory that stores one kind of product. Another example scenario is major conferences such as COMDEX, E3 Expo, etc. that typically attract tens of thousands of participants

[34]. The organizers are interested in various statistics, such as how many people visit a particular booth on which day. It is also an important basic function that helps to accomplish other complicated operations such as categorization [19], key assignment and updating [24]. More important, existing works show that fast and reliable estimation evidently improves the efficiency of tag identification [15] [16] [25] [26]. Estimation algorithms can be used to adjust the contention window and set an optimal frame size in slotted ALOHA-based identification (it is known that a frame with n slots is the most efficient setting, where n is the cardinality [27]). The efficiency and scalability (up to tens of thousand tags) of estimation algorithms are concerned by recent studies on this topic [16] [18] [34].

Intuitively, an RFID system can wait until all tags in the region successfully report to it, and then compute the cardinality by identifying them. Many identification schemes can be used for this approach [9] [10] [11] [12] [13], falling into two categories: *Slotted ALOHA schemes* [11] and *Tree-traversal or Binary Splitting schemes* [12] [13]. The most significant shortcoming of identification is the long processing latency. Hence, existing schemes based on identifying individual tags are impractical for large-scale RFID systems, especially when tags are attached to mobile objects. In this case, a tag may have left the reader's range before being identified. Therefore, in order to resolve the long latency issue and meet the real-time requirement, estimation schemes without identification are suggested [16] [17] [18].

In a simple estimation scheme, an ALOHA frame with fixed time slots is only able to estimate tag cardinalities within a restricted range. For example, a frame with ten slots can estimate tens of tags. If the tag number exceeds

• Chen Qian is with the Department of Computer Sciences, University of Texas, Austin, TX 78741. This work was partially done when this author was with Hong Kong University of Science and Technology.

E-mail: cqian@cs.utexas.edu

• Hoilun Ngan, Yunhao Liu and Lionel M. Ni are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong.

E-mail: {cpeglun, liu, ni}@cse.ust.hk

100, the estimators will fail. We call such problem *the limited operating range*. Limited operating range will cause the *time inefficiency* for estimation. Existing estimators [16] [34] need to dynamically adjust the system load factor (the ratio of the cardinality to the frame size) to fit the operating range, which increases the time cost of the estimation.

Besides the time inefficiency problem, another limitation of existing estimation schemes has to be addressed. Most of those estimations are designed for a single reader. Due to the terrain and limited interrogation region of readers, however, large-scale RFID deployments often need multiple readers [28], in which several readers are placed to cover the entire region of interest. Multi-reader RFID systems, being efficient and effective, suffer from the so-called *multiple-reading* problem. That happens when a number of tags stay in the overlapping interrogation region and respond to multiple readers simultaneously.

The key reason that causes these problems is RFID tag collision. When a collision happens, the reader cannot get any information. The resource (time and energy) that is consumed in that collision is completely wasted. In identification-based schemes, the reader should ask tags to re-transmit. For estimation-based schemes, if the tag number is much larger than the frame length (time slot number), almost every slot is collision. In that case, the limited operating range problem happens, and the estimation definitely fails.

In this work, we propose the Lottery Frame (LoF) protocol, a non-arbitration based scheme to estimate the cardinality of tags. In LoF, we arrange the collision slots in an ordered pattern. LoF then extracts the tag cardinality information from the special pattern. Therefore in LoF, tag collisions are no longer considered as waste of time and energy.

LoF is a novel approach to compute tag cardinality in a very short time. Two main advantages of LoF are high accuracy and fast processing speed. The three problems of existing counting/estimation mentioned above, e.g., long latency, limited operating range and multiple-reading, are solved by LoF. Our design can be easily implemented in current RFID systems without particular assumptions about the number and placement of readers and tags.

The rest of this paper is organized as follows. In Section 2, we describe the background and motivation of our estimation protocol. Section 3 presents our baseline protocol, and Section 4 describes the detailed protocol design of LoF. We propose three techniques that can bring remarkable performance improvement to LoF in Section 5, and then give some discussions about LoF protocol in Section 6. We present the performance evaluation in Section 7. Finally we conclude this work in Section 8.

2 RELATED WORK AND CURRENT PROBLEMS

In this section, we list the two important performance metrics for tag estimation: processing time and accuracy.

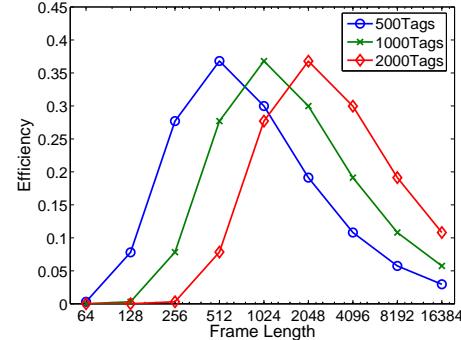


Fig. 1. The efficiency of slotted ALOHA identification

We also discuss the two problems that exist in the previous RFID tag estimation protocols [16] [17] and will affect the two performance metrics, namely time inefficiency and multiple-reading.

In our system model, we assume that there exists a separate estimation phase besides the identification phase in RFID systems. Therefore those systems can act as a tag counter without the loss of the function of tag identification.

2.1 The Background of RFID Estimation Schemes

The essential job of an RFID system is to identify tags in its interrogation region. Since RFID readers and tags usually operate on a same channel, simultaneous transmissions lead to collisions in the link layer. Protocols for arbitrating tag-to-tag collisions are usually categorized to two types, namely slotted ALOHA and tree-traversal. We describe the importance of an estimation scheme by addressing its role in efficient identification protocols.

In the slotted ALOHA protocol, given an ALOHA frame of fixed number of time slots, each tag randomly picks up a slot based on a uniform probability distribution and responds to the reader in that slot about their identities. The reader then detects the idle slots (no response), success slots (with a single response), and collided slots (with multiple responses). Only the tags that transmit to slots without collision can be recognized. The reader will then send out another query, asking successful tags to keep silent and collided tags to randomly respond again in the next round. The process continues until all tags are identified. Tree-traversal recursively splits tags into multiple subsets until each subset only includes one tag response. The advantage of tree-traversal is that the process is deterministic and the latency is predictable. Its disadvantage is, however, spending too much time and energy cost in sending queries.

Choosing the frame length (number of time slots) for slotted ALOHA is challenging. If the number of time slots is much smaller than that of tags, say 10 slots versus 100 tags, the collision happens in almost every slot, which results in success slots being very rare. If the number of time slots is much larger than that of tags, say 10000 slots

versus 100 tags, too many slots will be idle. Both cases affect the efficiency of slotted ALOHA protocol. It has been proved that a frame with n slots is the most efficient setting where n is the tag cardinality [27]. We plot the efficiency of slotted ALOHA varying in frame size in Fig. 1. It is clear that the best performance happens when the frame size is equal to n . A frame size that is close to n can also achieve high efficiency. The control overhead of tree-traversal can also be reduced by estimation [25] [26]. Thus cardinality estimation acts as an important aid for tag identification. Many existing works show that combined estimation-identification protocols are much faster and energy-efficient than simple slotted-ALOHA or tree-traversal [15] [16] [38] [26]. One pioneer work proposed by Simplot et al. [37] that employs estimation for efficient RFID tag anti-collision scheme is implemented in EPCGlobal Gen 2.

Estimation schemes can be applied directly to applications in which the user wants to know the population information of objects that have the same type of identities, e.g., an intelligent transportation system (ITS) [31] that tracks the population distribution of metropolitan vehicles for traffic control. Estimation is also an important basic function which helps to accomplish other complicated operations such as categorization [19] and key assignment and updating [24]. Kodialam and Nandagopal [16] proposed one of the earliest work in RFID estimation. A follow-up work, Enhanced Zero-based Estimator (EZB) [34] makes the estimation based on the number of empty slots. Recently, Han et al. [21] present a novel estimation algorithm based on the first non-empty slot of the ALOHA frame. Sheng et al. [22] develop efficient schemes for continuous scanning operations in both spatial and temporal domains. Li et al. [23] study the energy efficiency problem in RFID estimation and design several energy-efficient algorithms.

Note that although estimation can help identification, in this work we do not consider the identification problem. We only focus on the estimation problem, i.e., how to count the number of tags accurately and efficiently.

There are two more types of collisions in multi-reader RFID systems other than tag-to-tag collision: reader-to-tag, and reader-to-reader. In this paper our estimation protocol runs on top of (thus is independent from) the link-layer and employs the tag-to-tag collisions to obtain a better estimation result, instead of resolving the collision problems. Thus we make the assumption that the underlying link-layer protocol is well designed to avoid reader-to-reader and reader-to-tag collisions. For example, to mitigate reader-to-reader and reader-to-tag collisions, the system may assign readers multiple channels and does not allow the interfering readers transmitting at the same time [29], apply TDMA [28], or use carrier-sensing to develop a CSMA-like protocol [30]. We also assume that the control among multiple readers is perfect [39]. We only focus on the computing problem in this paper.

2.2 Time Inefficiency

Most basic RFID estimation algorithms use the frame-slotted ALOHA model. The estimation is computed based on the ratio of the number of idle slots and the number of total slots. Relying on the idle ratio, this kind of estimators suffer the limited operating range problem. We show that the limited operating range problem occurs when $n \gg l$, where n is the cardinality and l is the slot number in a frame.

Lemma 2.1: Let V_0 denote the number of idle slots. If n, l are relatively large, the expectation of V_0 follows

$$E(V_0) = le^{-n/l} \quad (1)$$

Proof: The number of replies in the k -th time slot, $F[k]$, will become 0 if no tag responses at that time. We know that every tag has $1/l$ probability to respond at the time slot k . Therefore,

$$Pr(F[k] = 0) = (1 - \frac{1}{l})^n = e^{-n/l}$$

Since the events are independent, we obtain,

$$E(V_0) = \sum_{k=0}^{l-1} Pr(F[k] = 0) = le^{-n/l}$$

Hence, if $n = 10l$,

$$E(V_0) = le^{-n/l} = 0.00454\% \times l.$$

Almost no slots are idle. Since the estimator relies on the idle ratio, the estimation definitely fails. To guarantee that at least one slot is idle, we must have,

$$E(V_0) = le^{-n/l} \geq 1 \iff n \leq l \ln l$$

Therefore, $l \ln l$ is the upper bound of tag number n that can be successfully computed by the estimator using a frame with l slots. The estimation range is restricted. We can find that other estimators that rely on collision ratio or readable ratio also have this problem.

Limited operating range brings two serious disadvantages in time efficiency. First, if n is very large, l should also be a large number (in most cases, $l > 0.1n$) to obtain a successful estimation. The latency is still too long to be ideal. More significant, the user of the protocols has to know in advance approximately how many tags are under estimation. Otherwise, the user is not able to set the proper frame length whose operating range satisfies the cardinality of tags. It leads the system to a dilemma, because obtaining the approximate tag cardinality is just the goal of estimation!

One of the earliest RFID estimation protocols was proposed by Kodialam and Nandagopal [16]. Both PZE [16] and its enhanced version EZB [34] use a persistent probability p to overcome the limited range problem. Every tag only transmits with probability p . Hence fewer slots are collided. The value of p is computed by $p = \min(1, 1.59/\rho)$, where ρ is the load factor defined as the

ratio of tag cardinality to the frame size. PZE and EZB need to dynamically adjust ρ to fit the operating range, hence increase the time cost. Moreover, one of the most important applications of estimation is to help frame adjusting in identification algorithms. If the load factor ρ has been known, we are already able to adjust the frame size.

2.3 Multiple-reading

In order to improve coverage, many RFID systems deploy multiple readers with overlapping interrogation regions [28], to guarantee that most tags are able to access at least one reader, even when the wireless links are unreliable and dynamically changing. Suppose there are n tags in a certain region and m readers can fully cover that region. The estimation results of readers are $\tilde{n}_1, \tilde{n}_2, \dots, \tilde{n}_m$. Obviously, we have

$$\text{MAX}(\tilde{n}_1, \tilde{n}_2, \dots, \tilde{n}_m) \leq n \leq \text{SUM}(\tilde{n}_1, \tilde{n}_2, \dots, \tilde{n}_m) \quad (2)$$

By employing the existing estimators like USE and UPE [16], each reader can obtain an estimation result of the tags in its vicinity. Nevertheless, it is only possible to compute MAX and SUM to estimate the entire cardinality. In many application, both MAX and SUM are unable to present an accurate estimate. We name the problem “multiple-reading” because people use SUM as the result, in which a same tag may be counted/estimated multiple times. Those tags are called *replicates*.

The solution for the multiple-reading problem is part of the preliminary of this work [18]. Most recent works in RFID estimation also proposed techniques to estimate overlapping tags in multi-reader situations [34] [20] [21].

2.4 Our solution

In following sections, we will introduce our efficient estimation scheme, LoF, which eliminates the multiple-reading by hashing and OR operation. Our LoF protocol overcomes time inefficiency problem and is able to compute the tag cardinality in a very short time. Furthermore, it can quickly compute the load factor ρ and help PZE and EZB to obtain a better performance, as we will show in Section 7.

In the design of LoF, we assume the entire region of interest is covered by multiple readers. We also assume every tag in that region can access at least one reader. Besides, each tag can store a number of hash values, which will be explained in Section 5. We assume that there is no transmission loss between tags and readers. The removal of these assumptions has impact on the accuracy of the result, but does not overthrow its correctness.

3 A BASELINE ALGORITHM

This section describes a baseline protocol, which includes the technique to resolve the multiple-reading problem. The idea of this technique was presented by Kodialam *et al.* [34] as the EZB estimator, and the preliminary version

of this paper [18], as the LPE estimator. The difference is that, tags in EZB use a seed value and a random function to determine the slot selection, while LPE asks tags to store hash values for slot selection. LPE requires extra storage for hash values. EZB requires both extra storage and computation cost to run the random function. In this paper we use LPE to refer this technique.

Suppose each reader r_j constructs an ALOHA frame F_j with l time slots, and then broadcasts the length l to probe tags nearby. When a tag t_i receives the probe message, it applies a particular hash function $H(key)$ to its ID i . The hash values of $H(key)$ are uniformly distributed ranging from 0 to h . After obtaining the result $H(i)$, t_i normalizes $H(i)$ to $[0, l - 1]$ and denotes the normalized value as k . Then it picks the k th slot in the frame to respond.

Consider a reader r_j keeping a bitmap BM_j , where the bit $BM_j[k]$ is corresponding to the time slot k in F_j , where $k = 0, 1, 2, \dots$. If the reader r_j hears no response (idle) in the time slot k , it sets $BM_j[k]$ as 0. If the reader hears one tag’s response or multiple responses (collision), the bit $BM_j[k]$ will be set as 1. Let V_0 denote the number of bits with value 0, and V_1 denote the number of bits with value 1. In single-reader scenarios, supposing the tag number is n and the frame length is l , similar with Lemma 2.1 we have,

Lemma 3.1: If n, l are relatively large, the expectation of V_0 and V_1 follow

$$\begin{aligned} E(V_0) &= le^{-n/l} \\ E(V_1) &= l(1 - e^{-n/l}) \end{aligned} \quad (3)$$

The proof can be found in the conference version of this paper [18]. Replacing n and $E(V_0)$ by their representations in terms of observed variables \tilde{n} and V_0 , we get the estimator of LPE,

Estimator 1 (LPE). \tilde{n} is an estimator of the tag number n , where

$$\tilde{n} = -l \ln(V_0/l) \quad (4)$$

The property of hashing is suitable for eliminating the multi-reading, since the datum with the same value will have the same hash value. Applying this estimator to multi-reader RFID systems, each of the readers does not compute V_0, V_1 individually. Instead, every reader reports its bitmap to the central server. After receiving bitmaps from all readers, the server applies logical OR to those bitmaps and obtain a merged bitmap. Then the server calculates the estimator \tilde{n} referring to the merged bitmap.

Figure 2(a) gives an example of Estimator 1 in single-reader scenarios (to the ease of understanding and drawing, we do not use very large values of n and l). An example of LPE in multi-reader RFID systems is illustrated in Fig. 2(b). Comparing with Fig. 2(a), all readers in Fig. 2(b) cooperate together like a “super reader” that can cover the entire region without generating replicates.

Lemma 3.2: Suppose tag sets S_1, S_2, \dots, S_m are in the vicinities of m readers r_1, r_2, \dots, r_m respectively. They

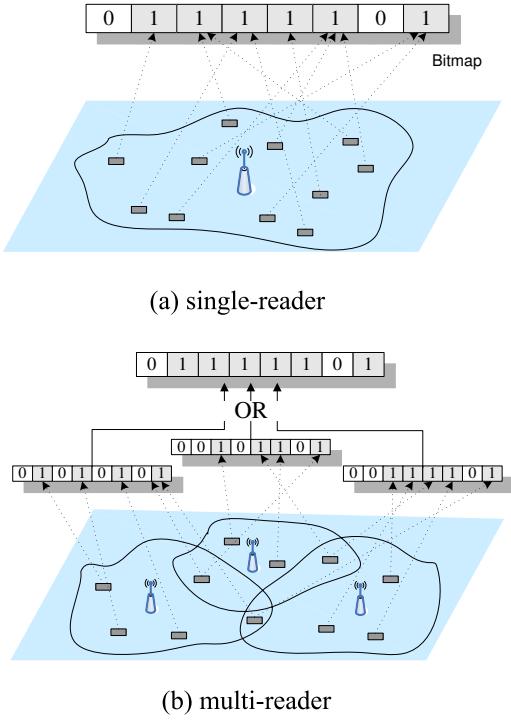


Fig. 2. Examples of LPE

share common members. The estimator \hat{n} of the merged bitmap equals to the estimation result of tag set $S_1 \cup S_2 \cup \dots \cup S_m$.

Thus we obtain,

Theorem 3.3: Estimator 1 (LPE) is a *replicate-insensitive* estimation, which eliminates the multiple reading, in multi-reader scenarios.

The proofs are quite straightforward and can be found in the preliminary version of this paper [18]. Note that the estimator definitely contains error, but the error is not brought by multiple-reading. For example, if all readers have a same bit pattern, say 11101101, the merged bitmap is also 11101101. This is a bad estimation result, as each reader's result is equal to that of taking them together. However, this result is still replicate-insensitive. Even if there was a "Super reader" which can cover the entire region, the resulting bitmap is still 11101101. The error is produced by the particular estimation algorithm, not replications.

LPE has two obvious disadvantages. First, the latency is still too long to be ideal. Table 1 provides the frame length l needed for LPE versus the number of tags. Moreover, it still has a limited operating range as analyzed in Section 2.2. In other words, if we do not know the tag cardinality, how could we set the frame length by referring Table 1? Thus we only use LPE (and EZB) as baseline algorithms. Further techniques are needed to improve it.

TABLE 1
Frame length needed for LPE

N	100	500	1000	5000	10000	50000
Frame Length	80	172	268	948	1709	6909

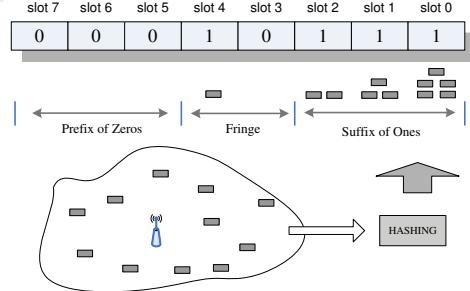


Fig. 3. An example of LoF estimation

4 THE DESIGN OF LOF PROTOCOL

In this section, we present our estimation protocol *Lottery Frame* (LoF), which combines LPE with geometric distribution, thus providing the scalability while saving the processing time and communication overhead compared with USE, UPE and LPE.

4.1 The General Protocol

Our estimation protocol LoF is developed based on the probabilistic bitmap counting techniques proposed for database processing [33] [35]. We are inspired by these techniques and use ALOHA frames as bitmaps for estimation. We adapt a spatial concept (bitmap) to a temporal one (time slots) in system design.

Every RFID tag can be considered as a lottery ticket, and the ticket number is the tag ID. To determine which kind of prize the ticket wins, the tag ID is hashed by a geometric distributed hash function $H()$ to a ALOHA slot, i.e., an ID has $1/2^t$ probability to be in the $(t - 1)$ th slot (the slots start from the right and ranks from 0). The class of prize is determined by the order number of slots. Hence the higher the class of prize is, the harder a tag can get it. Imagine the whole ALOHA frame as a bitmap. After hashing all tag IDs, we can point out three parts: the suffix of ones, the prefix of zeros and the fringe consists zeros and ones, like Fig. 3. Statistically, more tags reply, more left the fringe will be. The cardinality of tags can be estimated based on the position of fringe.

4.1.1 Tag

When probed by a reader in the estimation process, the tag applies the hash function to its ID and responds in a time slot according to the result. The simplest hash function with geometric distribution is,

$H(ID) =$ the position of least-significant (right-most) bit of zero in binary representation of ID

For example, $H(010100)=0$ and $H(001011)=2$. Apparently, 50% of the IDs are hashed to slot 0, because the

INPUT the number of hashed value num OUTPUT The selected slot number k	INPUT The number of hash functions m OUTPUT The set of bitmaps BM_1, BM_2, \dots, BM_m .
PROCEDURE while TRUE wait_for_messages(); if there is a probe message from a reader transmit a short message in time slot $k = H_{num}(ID)$; end if end while	PROCEDURE wait_for_message_from_server(); for $j=1$ to m broadcast a request to tags; for $i = 0$ to $l-1$ wait_for_responses(); if there are no responses in time slot i set $BM_j[i] = 0$; else set $BM_j[i] = 1$; end if end for end for report BM_1, BM_2, \dots, BM_m to the server;

(a) the algorithm running on tags

(b) the algorithm running on readers

Fig. 4. Pseudocode of LoF algorithm

least-significant bit (bit 0) has 50% probability to be zero. Also $1/2^t$ of the IDs are hashed to the slot $t - 1$. Here we just use this geometric distributed hash as an example, in real applications we may choose other geometric distributed hash functions as explained in Section 5.1. All geometric distributed hashes share the similar property and can be employed in LoF.

To make the implementation convenient, we just write the value $H(ID)$ onto tags during production. *LoF only requires tags storing hash values, instead of hash functions.* The penalty is only a little extra memory to store $H(ID)$ as the string of $H(ID)$ is much shorter than that of ID. Since LoF may use multiple hash functions to increase the accuracy as we will discuss in the next part, a tag using LoF has to attach multiple values in its memory, and selects one of them upon the request of the reader. Tags respond readers by a short message without any identifiable information.

Here we only use the least-significant bit of zero on the tag ID as an example for the geometric distributed hash functions. In reality, tags within an area might not be randomly and uniformly distributed. Hence in Section 5.1 we propose an advanced method to compute hash values that is independent from particular tag ID distributions.

The distributed LoF algorithm for tags is formally described in Fig. 4(a).

4.1.2 Reader

In LoF, readers use a slotted ALOHA model. Each reader also generates a bitmap at each round of communication. The positions of the bitmap correspond to time slots of the ALOHA frame. After hearing the tag responses, every reader should report its bitmap to a specific server. We illustrate our distributed algorithm for readers in Fig. 4(b), where the parameter m is used in case we need do multiple estimations as in Section 5.1. For this step we can let $m = 1$.

4.1.3 Server

We should expect, if there are n tags, approximately $1/2^t$ of the responses are in time slot $t - 1$. Thus, by merging the bitmaps by OR operation, the k th bit in

bitmap $BM[k]$ will be zero if $k \gg \log_2 n$, or be one if $k \ll \log_2 n$. The fringe consists zeros and ones for the k whose value is near $\log_2 n$.

In theory, the relationship between the geometric hashing result and the cardinality n is given by [33] and [35]. Based on their results, we have the following Lemma,

Lemma 4.1: Suppose R is the position of the right-most zero in BM , i.e. $R = \min\{i | BM[i] = 0\}$. The expected value of R satisfies

$$E(R) = \log_2(\varphi n) + P(\log_2 n) + o(1) \quad (5)$$

where the constant $\varphi = 0.775351\dots$ and $P(u)$ is a periodic and continuous functions of u with period 1 and amplitude bounded by 10^{-5} .

Lemma 4.1 was proven in [33]. Omitting the term $P(\log_2 n) + o(1)$, and replacing n and $E(R)$ by their representations in terms of observed variables \tilde{n} and R , we obtain,

Estimator 2. \tilde{n} is an estimator of the tag number n , where

$$\tilde{n} = 1/\varphi \times 2^R = 1.2897 \times 2^R \quad (6)$$

The following theorem shows that using geometric distribution hash functions, LoF has the advantage of fixed and short frame length.

Theorem 4.2: A frame with $\log_2 N$ slots is sufficient for the estimation protocol using geometric distribution hash functions, where N is the number of all the same series tags in production.

Proof: Clearly, the number of tags currently under estimation follows $n < N$. We know that $1/2^t$ of the n tags response to the time slot $t - 1$. Let $t = \log_2 N$. We have,

$$n \times \frac{1}{2^{\log_2 N}} < n \times \frac{1}{2^{\log_2 n}} = 1$$

which implies the time slot $\log_2 N - 1$ has no responses, and those slots for which $t > \log_2 N$ are also empty. Therefore, $\log_2 N$ slots are sufficient for LoF estimation. \square

Suppose there are 50000 tags produced in total. According to Theorem 4.2, we only have to fix the length of ALOHA frames as $\log_2 50000 = 15.60$. LoF evidently saves the processing time. Moreover, there is no constraint of operation range for LoF. Using frames with fixed length of 32, LoF can estimate the number of tags up to 2^{32} . Hence, LoF is highly scalable and time-efficient.

LoF sacrifices some space on tag memory for storing special hash functions.

4.2 LoF in Multi-reader RFID Systems

Similar to LPE, LoF can also eliminate the replications by merging bitmaps from readers. Since tags use hash function to select time slots for responding, the LoF protocol is also replicate-insensitive in multi-reader RFID systems. Suppose tag sets S_1, S_2, \dots, S_m are in the vicinities of m readers r_1, r_2, \dots, r_m respectively. They share common members. The estimator \tilde{n} of the merged

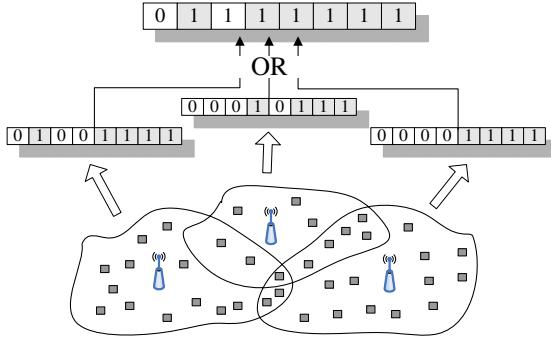


Fig. 5. An Example of LoF for Multiple Readers

LoF bitmap equals to the estimation result of tag set $S_1 \cup S_2 \cup \dots \cup S_m$. In other words, there is no difference between using multiple distributed readers and using a “super reader” that covers the entire region.

Theorem 4.3: LoF is a replicate-insensitive estimation in multi-reader scenarios.

The proof is same as that of Theorem 3.3.

All bitmaps before merging have a similar pattern, i.e., 0s in the one side and 1s in the other. However, since the estimation result is determined by the position of the right-most 0, bitmaps with a similar pattern might have highly different results. For instance, the estimate of 00010111 is $1.2897 \times 2^3 \approx 10$, but those of 00001111 and 01001111 are $1.2897 \times 2^4 \approx 21$. Merging them together to be 01011111, the result is $1.2897 \times 2^5 \approx 41$ as in Fig. 5.

5 PERFORMANCE IMPROVEMENT

Since LoF is only estimation rather than a precise counting protocol, its result must have some errors from the exact tag cardinality. If we employ the estimator as an aid to make identification protocols efficient, the estimation error might not be a big problem. Nevertheless, there are some applications that require obtaining the cardinality of objects as accurate as possible in a very short time, for which identification runs too slow. For instance, an airport or a stadium has large amounts of moving objects, and the statistical process should be fast enough to make the data valuable. In those systems we may want to trade some processing time for more accurate results, as long as the time is allowed.

In this section, we introduce three important techniques that can remarkably improve the estimation accuracy or reduce the latency of LoF, namely Multi-hash, Multi-splitting, and Sudden Victory.

5.1 Multiple Hash Functions

In [33], the authors derived the standard deviation of the geometric hashing result R . Modifying the theorem in [33] into our context, we have,

Theorem 5.1: Suppose R is the position of the right most zero in BM . The standard deviation of R satisfies

$$\sigma_n^2 = \sigma_\infty^2 + Q(\log_2 n) + o(1) \quad (7)$$

where the constant $\sigma_\infty = 1.1213\dots$ and $Q(u)$ is a periodic functions of u with mean value 0, period 1 and amplitude bounded by 10^{-5} .

Luckily, the estimator used in LoF protocol is proved to be *asymptotically unbiased* [33] [35]. In that sense, if we make several independent estimations and compute the average result, the standard deviation will be significantly reduced. In LoF protocol, we can employ a set of m independent hash functions. The readers should generate m bitmaps by all of the hash functions. Then LoF has m bitmaps and compute m positions of the right most zero R_1, R_2, \dots, R_m . Consider the average value

$$\bar{R} = (R_1 + R_2 + \dots + R_m)/m$$

The variable \bar{R} has the expectation and standard deviation that satisfy

$$E(\bar{R}) \approx \log_2(\varphi n), \sigma(\bar{R}) \approx \sigma_\infty/\sqrt{m}$$

Therefore, the improved estimator is

$$\tilde{n} = 1.2897 \times 2^{\bar{R}} = 1.2897 \times 2^{\sum_i R_i/m} \quad (8)$$

Figure 6 plots the standard error (defined in Section 7.1) in terms of the number of independent hash functions. We show both theoretical and empirical (in estimating 500, 5000 and 50000 tags) results.

Also, from Theorem 4.3 we know that there is no difference between using multiple distributed readers and using a “super reader” that covers the entire region. The estimation error of multiple hashes *does not increase* in multi-reader environments. We will show the fact by experiments.

Let α be the *error probability* and β be the *error bound* (also called confidence interval). We say LoF *achieves the accuracy requirement* if $\Pr[|\tilde{n} - n| \leq \beta n] \geq 1 - \alpha$. We show that,

Theorem 5.2: Given α and β , LoF achieves the accuracy requirement if $m \geq \max\{[\frac{-\sigma_\infty c}{\log_2(1-\beta)}]^2, [\frac{\sigma_\infty c}{\log_2(1+\beta)}]^2\}$, where c is obtained by solving $1 - \alpha = \text{erf}(\frac{c}{\sqrt{2}})$, erf is the Gaussian error function.

Proof: Let $\mu = E(\bar{R}) = \log_2(\varphi n)$ and $\sigma = \sigma(\bar{R}) = \sigma_\infty/\sqrt{m}$. From the central limit theorem, we have,

$$X = \frac{\bar{R} - \mu}{\sigma} \sim \mathcal{N}(0, 1)$$

i.e., X is a Gaussian with mean 0 and variance 1. Hence the cumulative distribution function is

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{u^2}{2}} du$$

Let a constant c satisfies,

$$1 - \alpha = \Pr[-c \leq X \leq c] = \text{erf}(\frac{c}{\sqrt{2}})$$

where erf is the Gaussian error function. For any value of α , we may get a corresponding c by solving the above equation.

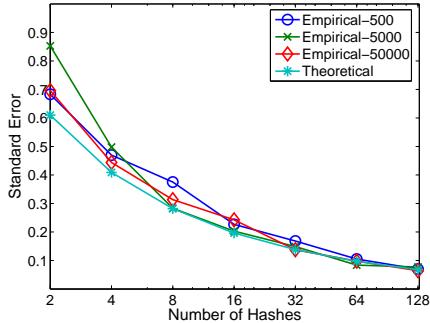


Fig. 6. Standard Error for Multiple Hash Functions

Also,

$$\begin{aligned} \Pr[|\tilde{n} - n| \leq \beta n] &= \Pr[(1 - \beta)n \leq \tilde{n} \leq (1 + \beta)n] \\ &= \Pr[(1 - \beta)n \leq \frac{1}{\varphi} 2^{\bar{R}} \leq (1 + \beta)n] \\ &= \Pr[\log_2((1 - \beta)\varphi n) \leq \bar{R} \leq \log_2((1 + \beta)\varphi n)] \end{aligned}$$

Hence, if $\frac{\log_2((1-\beta)\varphi n)-\mu}{\sigma} \leq -c$ and $\frac{\log_2((1+\beta)\varphi n)-\mu}{\sigma} \geq c$, $\Pr[|\tilde{n} - n| \leq \beta n] \geq 1 - \alpha$ is satisfied, i.e., LoF achieves the accuracy requirement.

Solving the inequalities, we get

$$m \geq \max\left\{\left[\frac{-\sigma_\infty c}{\log_2(1 - \beta)}\right]^2, \left[\frac{\sigma_\infty c}{\log_2(1 + \beta)}\right]^2\right\}$$

□

There are several approaches to find multiple geometric distributed hashes. One easy way is provided as following. In Section 4.1.1 we suggest a simple geometric distributed hash function: the position of least-significant bit of zero in binary representation of tag ID. Let us denote this hash as H' . Then we also employ a group of uniformly distributed hash functions, e.g., Message-Digest algorithm 5 (MD5) or Secure Hash Algorithm (SHA-1), denoted by $H_1, H_2, H_3, \dots, H_k$. For any $H_i \in H_1, H_2, H_3, \dots, H_k$, H_i hashes the ID to another binary representation, which can also be considered as a type of "ID". It is obvious that $H'(H_1(ID)), H'(H_2(ID)), H'(H_3(ID)), \dots, H'(H_k(ID))$ are all geometric distributed hash functions, because $H_i(ID)$'s rightmost zero also has a probability of $1/2^t$ to be in bit $t - 1$. Note that the hash values of MD5 are 128-bit, but it is not difficult converting them to the length we want.

For tag IDs with a special distribution instead of the uniformly random, using "the position of least-significant bit of zero", i.e., H' , may affect the accuracy of LoF. However, by applying the above technique, tag IDs with any distributions will be re-distributed to uniform by MD5 or SHA-1. We show an empirical result in Fig. 7. There are 10000 tags whose original ID distribution is far from uniform. Hence by applying H' , the result is not geometric distributed (marked by circles). However, using both MD5 and H' , the result is very close to a perfect geometric distribution (marked by crosses).

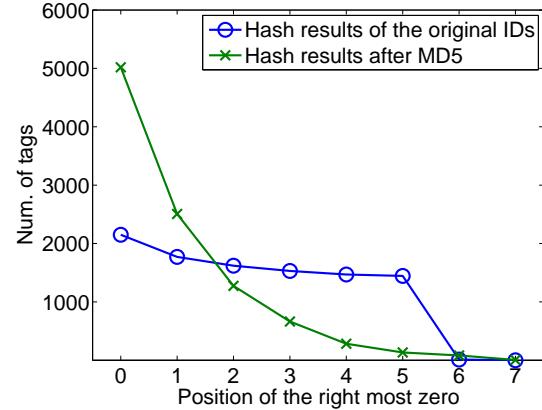


Fig. 7. The influence of a special ID distribution will be eliminated by MD5

Each hash value is within the range $[0, \log_2 N - 1]$. The storage cost for each hash value is the bit-length of the maximum value $\log_2 N - 1$, i.e., $\log_2(\log_2 N - 1)$. If each tag stores m hash values, the cost is $m \log_2(\log_2 N - 1)$. Note that even the tag ID requires at least $\log_2 N$ storage. Storing extra hash values does not increase the cost significantly.

Multi-hash is a tradeoff between time/energy efficiency and estimation accuracy. Nevertheless, as LoF reduces the resource cost from $O(n)$ to around $O(\log n)$, Multi-hash is still much more efficient than UPE and identification-based schemes.

5.2 Multi-splitting

Multi-splitting repeats the estimation in another way. Using LoF, the whole tag set is split into multiple subsets. Each subset contains tags that reply to the same collision slot. We also have a estimation value \tilde{n} . If more accurate result is needed. We further split those subsets by recursively applying LoF. Let $n_{20}, n_{21}, n_{22}, \dots$ be the estimated cardinality for subset 0, 1, 2, ... in the second-time splitting respectively. We then obtain a two-splitting estimation result by

$$\tilde{n}_2 = \text{Num}_{\text{ReadableSlot}} + \sum n_{2i} \quad (9)$$

The recursively calling of LoF stops when $|\tilde{n}_i - \tilde{n}_{i-1}| < T$, where T is an accuracy requirement threshold given by the user. T could be either a constant number, or a percentage value that represents the difference between \tilde{n}_i and \tilde{n}_{i-1} .

In the example of Fig. 8, slot 0, 1 and 2 are collisions slots. Each of them corresponds to one tag subset. Slot 5 only has one response and therefore the tag in slot 5 can be successfully counted. The other slots are idle. Tag subsets in slot 0, 1 and 2 will be further processed, which is called level-2 splitting. The reader can obtain the knowledge of an approximated size \tilde{n} of each subset. By Estimator 2, we have

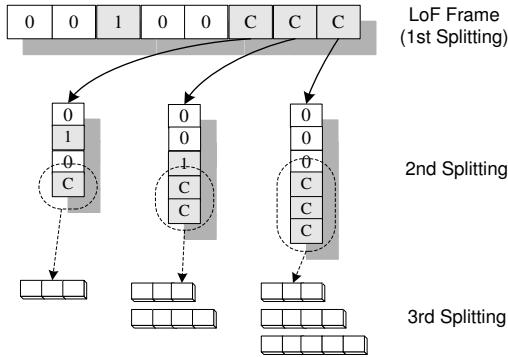


Fig. 8. Example of Multi-splitting

Estimator 3. If the i th slot is a collision slot, the estimated cardinality of tags that reply in this slot, i.e., the cardinality of s_i , is

$$\tilde{n}(i) = 1.2897 \times 2^{R-i} \quad (10)$$

Based on these values, the further splitting will be processed with shorter ALOHA frames. Multi-splitting continues until the condition $|\tilde{n}_i - \tilde{n}_{i-1}| < T$ is satisfied. As the example in Fig. 8, the splitting stops at the third time. Finally, the sum of these subsets derives the total number of tags in the original set, \tilde{n}_3 .

We will compare the performance of Multi-hash and Multi-splitting in the evaluation section. We will find that the performance of Multi-splitting is not as good as that of Multi-hash. However, Multi-splitting has its unique importance. If we recursively apply Multi-splitting until every slot only contains one tag response, the protocol is actually an identification protocol as every tag is identified. Therefore Multi-splitting is a unified protocol that can be used for all of estimation, precise counting and identification, by just controlling the recursion level. Moreover, since the approximate cardinality of every collided slot is known, such identification is much more efficient than simple slotted ALOHA [25]. Other estimators like UPE, EZB and LoF with Multi-hash do not have this feature. They all set the estimation as an independent phase from the identification.

Similar to Multi-hash, Multi-splitting also increases the storage cost on each tag. The hash values used on different levels should be different. Since no tags belong to more than one subset on a same level, all subsets on a same level can use the hash values produced by a same hash function. Each tag select a hash value from its memory by the current splitting level L specified in the reader query. Therefore, the maximum hash values stored in each tag is just the maximum value of L . We analyze the approximate maximum value of L as follows. Suppose the maximum tag number is N . In the level-1 splitting, the largest subset contains about $N/2$ tags. Hence in about $\log_2 N$ -th level, the largest subset includes only 1 tag. The number of hash values stored by each tag is thus $\log_2 N$. In fact, there is no need to do

the splitting until the subset only contains 1 tag. Practical RFID system may control the deepest splitting level by a constant K , e.g., $K = 16$. In this case, each tag only need to store 16 hash values. Similar to the analysis in Section 5.1, the extra storage cost for Multi-splitting is $K \log_2(\log_2 N - 1)$.

Detailed analysis for the time and energy efficiency of Multi-splitting can be found in [25].

5.3 Sudden Victory

According to the Sudden Victory Rule in soccer and golf games, a game may end as soon as one player is ahead of the others under some circumstances, e.g., in extra time. In LoF protocol, since the estimator only depends on R , the position of the right-most zero in the bitmap. The reader listens to the slots from right to left. It can stop listening as soon as it hears an idle slot, which represents the rightmost zero in the bitmap. Then it broadcasts a query to tell other tags to stop replying. Therefore, most ALOHA frames are not necessarily to be completed. Another alternative to implement this scheme is simply let the reader shut down the electromagnetic field without sending any messages. Such *Sudden Victory Rule* can potentially shorten the processing time. No matter how long the frame is initially set, the listening always stops around the $\log_2 n$ -th time slot, where n is the current cardinality of tags under estimating. Sudden Victory rule reduces the processing time of each estimation from $\log_2 N$ (as proved in Theorem 4.2) to around $\log_2 n$.

Applying Sudden Victory rule, the readers need to pay more cost to probe the "Stop" message. For Multi-hash scheme, however, a reader can combine the "Stop" message and the "Start" message of the next independent estimation, so that the extra cost can be reduced to minimum. The time and energy cost for the "Stop" signal is different for various RFID systems. It is possible that the "Stop" signal takes significantly longer time than each time slot. In such case, the RFID system should not apply the Sudden Victory.

Multi-hash and Multi-splitting are exclusive. Nevertheless, Sudden Victory rule is orthogonal to both of them. In this paper, we use Multi-hash plus Sudden Victory as our main strategy.

6 DISCUSSION

The key reason that causes the long latency and limited operating range problems is RFID tag collision. When multiple tags respond a reader simultaneously (or within a very short time interval), a collision happens. Since the reader cannot recognize any data from those tags, no information can be extracted. Therefore the resource (time and energy) that is consumed in the collision is completely wasted.

LoF compresses the number of collision slots to $\log n$ size. By employing geometric distributed hash function, the first several slots include the most tags. If $t > \log_2 N$, slot t is definitely idle. This compression allows the

estimation processing to finish in a very short time period so that LoF can satisfy the real-time requirements for most RFID applications. Secondly, LoF arranges the collision slots by their tag numbers in a descending order from right to left. Each time the tag number approximately decreases by half. When detecting an idle slot, LoF can estimate that the slot on its right hand may not have too many tag responses, probably 2-4. In this way, the approximate number of tags in each collision can be obtained. LoF then extracts the tag cardinality information from these collisions and the position of right-most zero. Therefore in LoF, tag collisions are no longer considered as waste of time and energy.

As collisions also provide cardinality information, tags are not required to re-transmit even though they are in collisions. Thus LoF further reduces the time and energy cost. In Section 5, however, we introduce some techniques that may ask tags to re-transmit to improve the estimation accuracy. We will show that with those techniques, the processing speed of LoF is still much faster than existing schemes.

One limitation of LoF is that it requires additional memory storage and a customized production or pre-processing process. They might bring extra cost to today's RFID applications. However, we expect two potential ways to overcome this problem and help LoF to be widely applied. First is that the production of customized RFID chips becomes simple enough. Second is that the computing on RFID tags becomes powerful enough to support geometric distributed hash functions or use sequential Bernoulli trials to simulate the geometric distribution. By [36], current RFID tags are already capable to run efficient Bernoulli trials.

7 PERFORMANCE EVALUATION

In this section, LoF is evaluated through comprehensive simulations. First we describe the simulation setup, and address the schemes and performance metrics we evaluate. We then provide the simulation results for both single-reader and multi-reader scenarios.

7.1 Simulation Methodologies and Performance Metrics

We built a *packet-level* simulator, where tags are mobile and can move out of readers' interrogation range. The interrogation range of each reader is set to be circular with the same radius. We assume that there is no transmission loss between tags and readers. Readers are capable to detect if it is idle, a single reply or collision in any frame slot. In case there are multiple readers, at most one reader is operating at any time to avoid reader-reader collision. Each tag is engineered to have globally unique IDs and a set of pre-computed 20-bit hash values, because we assume the maximum tag number is 2^{20} . The cost of LoF does not depend on the ID length. In our simulation setting, LoF can support any ID length equal to or smaller than 20 bits. The geometric distributed

hash functions can map IDs with any length to one of the 20 ALOHA slots. Note that the actual cost of each estimation is usually much less than 20 slots because of the Sudden Victory rule. We assume that there is no information about the system load factor (the ratio of the tag number to the ALOHA frame size), or the tag population range.

TABLE 2
Simulation Setup

	Single-reader	Multi-reader
Schemes evaluated	LoF and EZB	LoF, UPE and EZB
Metrics	accuracy and latency	accuracy and latency
Tag size	128 - 65536	128 - 65536
Number of readers	1	4
Terrain	N/A	100 x 150 rectangular
Interrogation range	infinite	50-80
Repeating times	100	100

We compare the estimation schemes in two scenarios: single-reader and multi-reader environments. For the single-reader environment, we contrast LoF with EZB [34] and UPE [16]. Unlike UPE, EZB supports estimation in multi-reader environments. Also EZB is asymptotically unbiased, i.e., with more independent experiments the accuracy can be improved. We also evaluate LoF and EZB for different number of independent experiments in both scenarios. Finally we will compare the performance of Multi-hash and Multi-splitting. Table 2 shows the setup of our simulator. The simulation takes 100 runs with the same parameters, and we report the average.

We test the a wide range of RFID cardinalities (128-65536) to evaluate the scalability of our algorithm. In addition, we want to be consistent to the experiments in the EZB paper [34]. In the EZB paper, the simulation scenario is a major conference such as COMDEX, E3 Expo, etc. that typically attract tens of thousands of participants. The cardinality range used by its simulation is from about 100 to 50392.

The estimation accuracy is illustrated in terms of the standard deviation σ and standard error e . Suppose the actual cardinality is n and the estimated value is \tilde{n} . The *standard deviation* σ , commonly used in statistics, is defined as,

$$\sigma = \sqrt{E[|\tilde{n} - n|^2]}$$

where the operator E denotes the average or expected value. To the convenience of the comparison, we additionally define *standard error* e that scales the value of σ into a small range:

$$e = \frac{\sigma}{n}$$

Note that this error metric is stricter than the one used in the preliminary version [18], i.e., the average absolute difference over the cardinality. Ideally, the error should

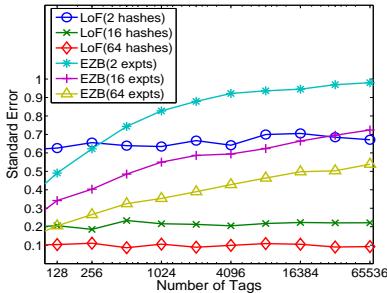


Fig. 9. Accuracy comparison in single-reader environment: LoF vs. EZB

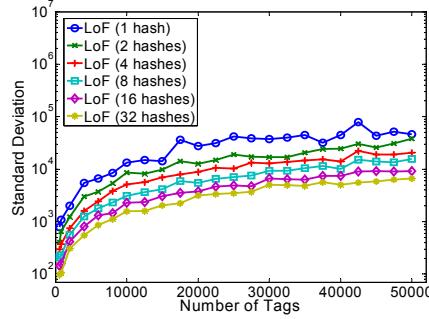


Fig. 10. Standard deviation for LoF with multiple Hashes

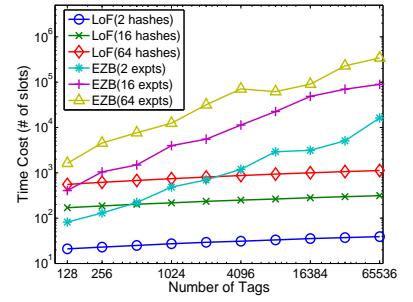


Fig. 11. Time cost comparison for single reader: LoF and EZB

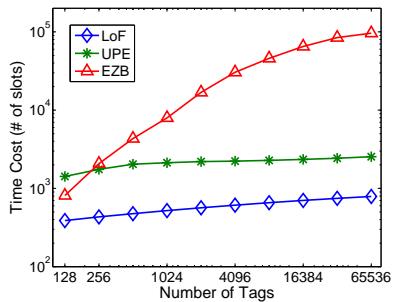


Fig. 12. Time cost comparison for LoF, UPE, and EZB for $\alpha = 1\%$, $\beta = 5\%$

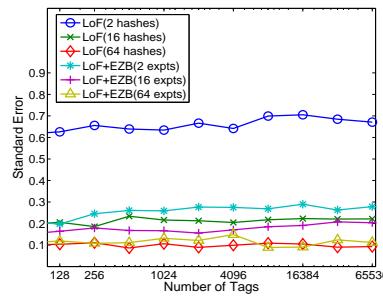


Fig. 13. Accuracy comparison: LoF vs. LoF-aided EZB

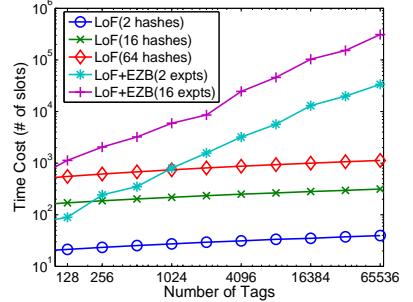


Fig. 14. Time cost comparison: LoF and LoF-aided EZB

be 0. The closer the error to 0, the better the estimation is.

Another metric in our concern is the time cost. Tag cardinality estimation requires several transmission rounds among readers and tags. For each round, readers take a number of time slots according to frame length. For convenience, we assume the time cost of one reader query is equal to four times of that of a time slot. The exact time cost for real RFID systems depends on the particular system setting. We abstract the estimation time as the total number of frame slots plus four times the number of reader queries during the whole process. In the figures, we will use “the number of slots” to represent the time cost, although it actually includes the cost for queries. The reader queries include the starting query of the ALOHA frame and the “Stop” query of the sudden victory rule.

7.2 Single-reader Scenario

We plot the estimation accuracy of LoF and EZB in terms of the standard error in Figure 9. Both of them are tested with 2, 16, 64 independent estimates. With more hash functions used in LoF, the error reduces. Employing 16 hash functions, the error is down to less than 0.2. The change of actual tag number has no significant effect to

LoF. In other words, the 20-slot frame is suitable for a wide range of tag numbers. The standard error for EZB decreases when the number of experiments is increased, because EZB is also asymptotically unbiased. However, the standard error is relatively high for large number of tags. On the other hand, LoF curves are relatively flat.

Figure 10 shows the standard deviation of estimation accuracy of LoF. When more hash functions are used in the estimation, the standard deviation of the estimation drops.

Figure 11 presents the time costs of LoF and EZB, in terms of the number of total time slots and queries. LoF employs Sudden Victory rule. Since the time requirement of LoF is only increased on a logarithmic scale, the total cost rises slowly. On the other hand, EZB’s time cost increases linearly with the tag cardinality. We show the time cost with tag numbers varying from 128 to 65536 in logarithmic scale. It is clear that when the tag number is over 256, LoF with 16 hashes performs better than other techniques. When the number is over 1000, LoF with 64 hashes also outperforms EZB.

As defined in Section 5.1, let the error probability $\alpha = 1\%$ and the error bound $\beta = 5\%$. We compare the time cost of LoF, UPE and EZB in Fig. 12. LoF

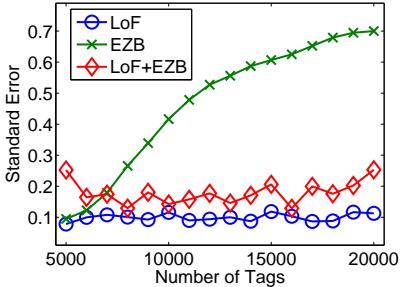


Fig. 15. Best effort under fixed time budget (1000 slots)

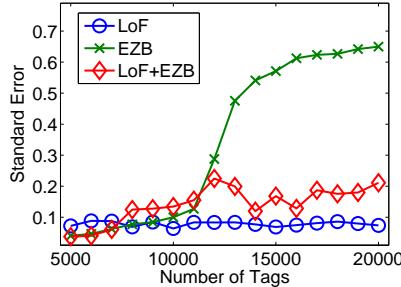


Fig. 16. Best effort under fixed time budget (1500 slots)

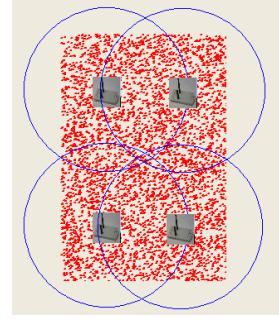


Fig. 17. Simulation environment

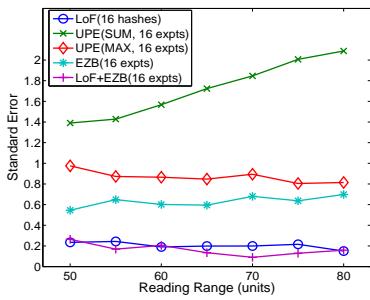


Fig. 18. Accuracy comparison in multi-reader environment

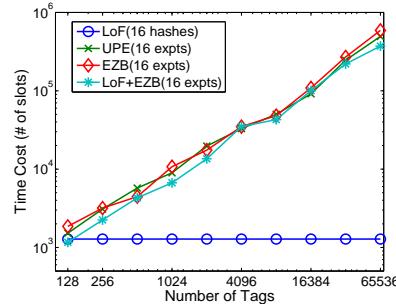


Fig. 19. Time cost comparison in multi-reader environment

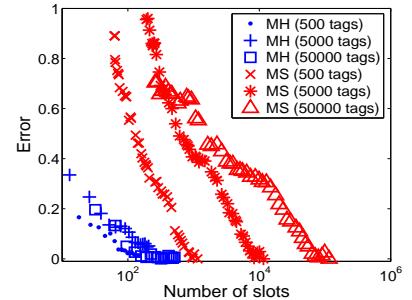


Fig. 20. Time cost comparison between multi-hash and multi-splitting

requires the least time among all protocols. Since UPE is not asymptotically unbiased, and cannot be run many times independently to achieve high accuracy, we do not compare UPE in Fig. 9-11. Note that UPE needs to recognize whether a slot is an idle, success or collision. However, LoF and EZB only need to distinguish an idle slot from a non-idle one, therefore the time duration of every slot of LoF and EZB is much shorter than that of UPE [21].

Compared with UPE and EZB, LoF provides likewise good estimation accuracy with less expense on time cost. In addition, users have no need to change the frame length in LoF.

7.2.1 LoF-aided EZB

The standard error of EZB in Fig. 9 is high when the cardinality is large. Our LoF estimator can help EZB to quickly predict the load factor ρ . We conduct a series of experiments as follows: the reader first uses LoF to estimate the cardinality, and then uses the estimate to compute the load factor ρ for EZB. As suggested by [34], the persistence probability p is computed by

$$p = \min(1, 1.59/\rho)$$

The accuracy of such LoF-aided EZB method is shown in Fig. 13. With more reliable load factors, LoF-aided EZB performs much better than simple EZB. Further more, the curves of LoF-aided EZB are more flat than EZB's, which indicates that LoF-aided EZB is also scalable.

Figures 14 presents the time cost comparison for LoF and LoF-aided EZB. We may conclude that for small tag cardinality (around 100), LoF-aided EZB has less time cost and similar accuracy. For large tag set (> 1000), multi-hashed LoF is more desirable.

7.2.2 Best Effort

Here we evaluate the capability of these estimators in such a manner: given a fixed budget of time slots (1000 and 1500), we wish to estimate the tag cardinality (ranging from 2000-20000) as accurate as each estimator can do.

LoF tries multiple hashes until the budget is used up. For example, if each estimation takes about 20 slots (including the reader queries), then LoF will try about 50 different hashes for the 1000-slot budget. It then computes the result by (8). EZB is executed multiple times, until the budget runs over. LoF-aided EZB first employs LoF to estimate the load factor, then determines the time to run EZB.

The results are plotted in Fig. 15 and 16. When the tag number is close to the budget, all of the three methods have good accuracy. After the cardinality growing out of the operating range of EZB, the error of EZB rises very quickly. The error of LoF-aided EZB has a slight growth when the cardinality becomes large. On the other hand, the performance of LoF almost keeps still. Comparing the two figures, the operating range of EZB depends on the time budget, but those of LoF and LoF-aided EZB depend less. The performance of LoF is stable.

7.3 Multi-reader scenario

We then consider multiple readers that have overlapping interrogation regions. Different number of tags is randomly distributed in a 100 units \times 150 units rectangular terrain. The terrain is covered by four readers, which are located near the corners. The reading range is varying between 50 and 70 units. The simulation model is shown in Fig. 17.

Figure 18 provides the accuracy of LoF, MAX, SUM, EZB and LoF-aided EZB in multi-reader environment, where each reading in MAX and SUM is generated by UPE. It indicates that MAX and SUM estimators perform poorly in cardinality estimation. These estimations are affected by multiple factors including but not limited to the number of readers, the interrogation range and reader deployment. Estimations by MAX are usually well-below the actual cardinality while those by SUM are always multiple times of the actual number of tags. As shown in Fig. 18, when the interrogation range grows, the error of MAX becomes lower and SUM goes further beyond accurate, because longer interrogation range will cause more overlapping areas. The cardinality estimation from merged bitmaps of every LoF reader provides the highest accuracy. EZB supports multi-reader environments. It does not perform well in Fig. 18 only because of the same reason in Fig. 9, i.e., no information about the tag number. With the help of LoF, EZB can obtain very accurate results in multi-reader context.

Figure 19 shows the frame slot requirements for these estimators for multiple readers. To compute the overall time costs, we may assume that when a reader sends out its query, other readers will keep silent until the probing reader finishes listening. This is the simplest TDMA scheduling for the MAC layer of RFID readers. Thus we sum up the LoF frame length from each of the four readers. In real world environment, the processing latency can be further reduced by some advanced techniques [28] [30]. Again LoF achieves the best time efficiency.

7.4 Multiple-hashing vs. Multiple-splitting

This subsection evaluates the performance of Multi-hash and Multi-splitting schemes. As stated in previous section, Multi-hash and Multi-splitting can be viewed as

horizontal and vertical iterative LoF processes respectively. In Multi-splitting, the splitting process is continued until the condition $|\tilde{n}_i - \tilde{n}_{i-1}| < T$ is satisfied. The threshold T is an application parameter which affects the performance metrics – error and latency. The smaller the value of the threshold T , the more the iteration is. A smaller threshold results in less error but longer latency. The threshold can be defined as number of tags or fraction of tags depending on the application requirements. In our simulation, we have chosen the later definition.

Figure 20 shows the performance of Multi-hash and Multi-splitting by varying the number of hashes (for Multi-hash) and threshold (for Multi-splitting) for different number of tags in the terrain. In Multi-splitting, the reader requires more slots for larger estimation tag set in order to achieving the same level of accuracy. The probability of satisfying the condition $|\tilde{n}_i - \tilde{n}_{i-1}| < T$ decreases with the increasing number of the tags in the estimation set. This leads to more iteration to trim down the size of the estimation set and hence increases the number of slots for cardinality estimation.

Multi-hash always achieves better performance than Multi-splitting does. Multi-hash also has the advantage of better performance predictability over Multiple-splitting. For a predefined performance requirement, choosing the suitable number of hashes is easier than selecting the proper threshold T .

8 CONCLUSION

Counting the number of tags is a crucial task in large-scale RFID systems. In this work, we propose LoF, an accurate and efficient estimation protocol that can resolve the time inefficiency and multiple-reading problems. Hence it can be applied for large-scale RFID systems. It estimates the tag cardinality by extracting information from the collisions. Our theoretical analysis and simulation results show that LoF can achieve accurate estimation in both single-reader and multi-reader environments, and significantly reduce the time cost.

We are also planning to explore more issues that are practical in multi-reader RFID systems, such as object tracking and tag authentication. Recent study on finding popular tag categories uses cardinality estimation as a basic tool [19]. With our method, the efficiency and accuracy of such algorithms can definitely be highly improved. We will also study how LoF cooperate with the identification and precise counting protocols [25]. Indeed, LoF design can be easily applied to any network protocol that runs on top of an ALOHA-based link layer. It can also benefit other fields, e.g., wireless sensor networks, cellular networks and vehicular networks.

ACKNOWLEDGMENT

This research was supported in part by: Hong Kong RGC Grant HKUST617710, China NSFC Grants 60933011 and 60933012, the National Basic Research Program of

China (973 Program) under Grant No. 2006CB303000, the National Science and Technology Major Project of China under Grant No. 2009ZX03006-001, the Science and Technology Planning Project of Guangdong Province, China under Grant No. 2009A080207002, NSFC/RGC Joint Research Scheme N_HKUST 602/08, the National Basic Research Program of China (973 Program) under grant No. 2011CB302705, and National Science Foundation grant CNS-0830939.

REFERENCES

- [1] K. Finkenzeller, *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*. John Wiley & Sons, 2000.
- [2] L. M. Ni, Y. Liu, Y. C. Lau, and A. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," in Proceeding of the first IEEE International Conference in Pervasive Computing and Communications (PerCom), 2003.
- [3] A. Ranganathan, J. Al-Muhtadi, S. Chetan, R. Campbell, and M. D. Mickunas, "MiddleWhere: A Middleware for Location Awareness in Ubiquitous Computing Applications," in Proceeding of the 5th International Middleware Conference (Middleware), 2004.
- [4] D. Zhang, J. Ma, Q. Chen, and L. M. Ni, "An RF-based System for Tracking Transceiver-free Objects," in Proceeding of the Fifth IEEE International Conference in Pervasive Computing and Communications (PerCom), 2007.
- [5] Y. Liu, L. Chen, J. Pei, Q. Chen, and Y. Zhao, "Mining Frequent Trajectory Patterns for Activity Monitoring Using Radio Frequency Tag Arrays," in Proceeding of the Fifth IEEE International Conference in Pervasive Computing and Communications (PerCom), 2007.
- [6] Shlomi Dolev, Marina Kopeetsky, "Secure Communication for RFIDs Proactive Information Security Within Computational Security." *SSS* 2006
- [7] R. Want, "An Introduction to RFID Technology," *IEEE Pervasive Computing*, vol. 5, pp. 25- 33, 2005.
- [8] I. Chlamtac, C. Petrioli, and J. Redi, "Energy-conserving access protocols for identification networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 7, pp. 51-59, 1999.
- [9] J. Lee, T. Kwon, Y. Choi, S. K. Das, and K.-a. Kim, "Analysis of RFID anti-collision algorithms using smart antennas," in Proceeding of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys), 2004.
- [10] V. Namboodiri and L. Gao, "Energy-Aware Tag Anti-Collision Protocols for RFID Systems," in Proceeding of the Fifth IEEE International Conference in Pervasive Computing and Communications (PerCom), 2007.
- [11] L. G. Roberts, "Aloha Packet System with and without Slots and Capture," *ACM SIGCOMM Computer Communication Review*, vol. 5, pp. 28-42, 1975.
- [12] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. on Information Theory*, vol. IT-25, pp. 505-515, 1979.
- [13] J. Myung and W. Lee, "Adaptive Splitting Protocols for RFID Tag Collision Arbitration," in Proceeding of International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2006.
- [14] J. Myung, W. Lee, J. Srivastava and T. Shih, "Tag-Splitting: Adaptive Collision Arbitration Protocols for RFID Tag Identification," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, No. 6, 2007.
- [15] S. S. Lam, *Packet Switching in a Multi-Access Broadcast Channel with Application to Satellite Communication in a Computer Network*, Ph. D. Dissertation, Computer Science Department, University of California at Los Angeles, March 1974.
- [16] M. Kodialam and T. Nandagopal, "Fast and Reliable Estimation Schemes in RFID Systems," in Proceeding of the Twelfth Annual International Conference on Mobile Computing and Networking (Mobicom), 2006.
- [17] C. Floerkemeier, "Transmission control scheme for fast RFID object identification," in Proceeding of PerCom Workshops, 2006.
- [18] C. Qian, H.-L. Ngan, Y. Liu, "Cardinality Estimation for Large-scale RFID Systems", in Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communication (IEEE PerCom' 08).
- [19] B. Sheng, C. C. Tan, Q. Li and W. Mao, "Finding Popular Categories for RFID Tags," in *ACM MobiHoc*, 2008.
- [20] V. Shah-Mansouri and V. W.S. Wong, "Anonymous Cardinality Estimation in RFID Systems with Multiple Readers," in Proc. of *IEEE Globecom*, 2009
- [21] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID Tags Efficiently and Anonymously," in *IEEE INFOCOM*, 2010.
- [22] B. Sheng, Q. Li, and W. Mao, "Efficient Continuous Scanning in RFID Systems," in *IEEE INFOCOM*, 2010.
- [23] T. Li, S. Wu, S. Chen, and M. Yang, "Energy Efficient Algorithms for the RFID Estimation Problem," in *IEEE Infocom*, 2010.
- [24] L. Lu, J. Han, L. Hu, Y. Liu, and L. M. Ni, "Dynamic Key-Updating: Privacy-Preserving Authentication for RFID Systems," (PDF), *IEEE PerCom* 2007.
- [25] C. Qian, *Efficient Cardinality Counting for Large-scale RFID Systems*, Master's thesis, Hong Kong University of Science and Technology, July, 2008.
<http://www.cs.utexas.edu/~cqian/thesis.pdf>
- [26] G. Maselli, C. Petrioli, C. Vicari, "Dynamic Tag Estimation for Optimizing Tree Slotted Aloha in RFID Networks," *ACM MSWIM* 2008.
- [27] F. F. Kuo, "The ALOHA system." *ACM Computer Communication Review*, 1995.
- [28] Z. Zhou, H. Gupta, S. R. Das, and X. Zhu, "Slotted Scheduled Tag Access in Multi-Reader RFID Systems," in Proceeding of the fifteenth IEEE International Conference on Network Protocols (ICNP), 2007.
- [29] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: An Anticollision Algorithm for the Reader Collision Problem," in *IEEE International Conference on Communications*, vol. 2, 2002.
- [30] S. Jain, S. R. Das, "Collision Avoidance in a Dense RFID Network," in *ACM WiNTECH*, 2006.
- [31] A. D. Joseph, A. R. Beresford, J. Bacon, et al, "Intelligent Transportation Systems," *IEEE Pervasive Computing*, vol. 5, pp. 63-67, 2006.
- [32] K.-Y. Whang, B. T. Vander-Zanden, and H. M. Taylor, "A Linear-Time Probabilistic Counting Algorithm for Database Applications," *ACM Transactions on Database Systems*, vol. 15, pp. 208-229, 1990.
- [33] P. Flajolet and G. N. Martin, "Probabilistic Counting Algorithms for Data Base Applications," *Journal of Computer and System Science*, vol. 31, 1985.
- [34] M. Kodialam, T. Nandagopal, W. C. Lau, "Anonymous Tracking using RFID tags," in *IEEE Infocom*, 2007.
- [35] M. Durand and P. Flajolet, "Loglog Counting of Large Cardinalities," in *ESA*, 2003.
- [36] D. Holcomb, W. Burleson, K. Fu, "Power-up SRAM State as an Identifying Fingerprint and Source of True Random Numbers," to appear at *IEEE Transactions on Computers*.
- [37] D. Simplot, M. Latteux, and R. Kalinowski, "An Adaptive Anti-Collision Protocol for Smart Labels," *LIFL/Gemplus contribution to Joint ISO/IETC 18000-3 Work Group*, 2001.
- [38] C. Qian, Y. Liu, H. Ngan, L. M. Ni, "ASAP: Scalable Identification and Counting for Contactless RFID Systems," to appear at *IEEE ICDCS*, 2010.
- [39] Application Level Events (ALE) Standard, EPCglobal Board.
<http://www.epcglobalinc.org/standards/ale>



Chen Qian Chen Qian is a Ph.D. student at Department of Computer Science, the University of Texas at Austin. He received the BS degree in computer science from Nanjing University, China, in 2006, the MPhil degree in computer science and engineering from the Hong Kong University of Science and Technology, in 2008. His research interests include computer networking, distributed systems, and pervasive computing. He is a student member of IEEE and ACM.

PLACE
PHOTO
HERE

Hoilun Ngan Raymond Hoilun Ngan is a Ph.D. candidate in Computer Science and Engineering, Hong Kong University of Science and Technology. He received his BEng degree in Computer Engineering (First Class Honors) and MPhil degree in Computer Science from HKUST in 2003 and 2005 respectively.

PLACE
PHOTO
HERE

Yunhao Liu Yunhao Liu received the BS degree in automation from Tsinghua University, China, in 1995, the MS and PhD degrees in computer science and engineering from Michigan State University, in 2003 and 2004, respectively. He is now a professor at TNLIST, School of Software, Tsinghua University, as well as a faculty member with the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His research interests include wireless sensor network, peer-to-peer computing, and pervasive computing. He is a senior member of the IEEE Computer Society and an ACM Distinguished Speaker.

PLACE
PHOTO
HERE

Lionel M. Ni Lionel M. Ni is Chair Professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). He also serves as the Special Assistant to the President of HKUST and Director of the HKUST China Ministry of Education/Microsoft Research Asia IT Key Lab. A fellow of IEEE, Dr. Ni has chaired over 30 professional conferences and has received six awards for authoring outstanding papers.