Cardinality Estimation for Large-scale RFID Systems

Chen Qian, Hoilun Ngan, and Yunhao Liu
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
{cqian, cpeglun, liu}@cse.ust.hk

***Abstract*—Counting or estimating the number of tags is crucial for large-scale RFID systems. The use of multiple readers was recently proposed to improve the efficiency and effectiveness in reading RFID tags. Due to the long processing time, tag identification based counting schemes are often impractical, especially when tags are attached to moving objects. The existing estimation based schemes, on the other hand, suffer from the multiple-reading problem. To address this issue, we propose the Lottery Frame (LoF) scheme, a replicate-insensitive estimation protocol, that is able to eliminate multiple-readings. We show the high accuracy, short processing time and low overhead of the proposed LoF scheme through analysis and simulations.**

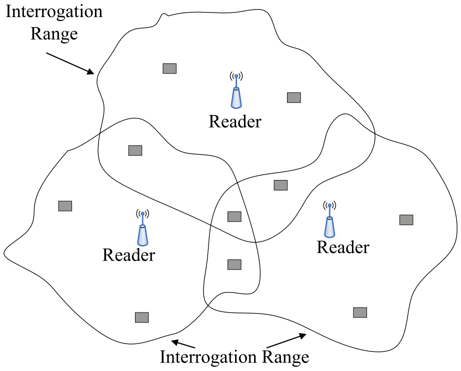


Fig. 1. An example of multi-reader RFID systems

I. Introduction

Radio Frequency Identification (RFID) technology [1] has been widely used for many applications, such as localization [2] [3] and objects tracking [4], activity monitoring [5], and 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 to RFID tags. RFID tags are labeled in designated objects where each tag has a small size of memory to store its unique ID as well as other information.

Counting the number of tags in a given region is one of the most important tasks in large-scale RFID systems. Intuitively, we can identify all tags in the region [8] and then compute the cardinality. As many tags may respond to a reader simultaneously, the collision among tags becomes a critical factor affecting the reading accuracy. Many anti-collision schemes have been proposed [9] [10] [11] [12] [13], falling into two categories: *Slotted ALOHA schemes* [11] and *Tree-based* or *Binary Splitting schemes* [12] [13]. The most significant shortcoming of such schemes is the long processing latency. As analyzed in [13], using tree-based schemes to identify 3000 tags will need over 1000 seconds. For slotted ALOHA schemes, the time is even longer [9]. Hence, existing schemes based on the identification of individual tags are impractical for large-scale RFID systems, especially when tags are attached to mobile objects. In this case, a tag may leave the reader's range before being identified.

In order to address the long latency issue, estimation schemes without identification are proposed [14] [15]. Most, if not all, of those approaches 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 [16], 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. As illustrated in Fig. 1, a number of tags stay in the overlapping interrogation region and respond to multiple readers simultaneously. Let $r_1, r_2, ..., r_m$ denote the readings of m readers. Some tags are counted by multiple readers. To estimate the total number of tags, the existing schemes can be generalized to use either $\text{MAX}(r_1, r_2, ..., r_m)$ or $\text{SUM}(r_1, r_2, ..., r_m)$ as the estimator. For example, in Fig. 1, existing schemes report either the MAX, which is 5, or the SUM, 15, while the actual number of tags is 10.

Moreover, most of previous estimation schemes use dynamic incremental slotted ALOHA frames [14] [15]. They modify their frame size based on the feedback from readers. Consequently, many communication rounds are needed to finalize an optimal length of the frame, which is linearly proportional to the size of the tag set. For example, to estimate thousands of tags, readers should create frames with hundreds of slots [14]. Thus, the processing time will be extremely long in large-scale RFID systems.

In this work, we propose the Lottery Frame (LoF) protocol, a non-identification based scheme to estimate the cardinality of tags. The advantages of LoF include: (1) replicate-insensitive with a high accuracy; (2) good scalability with short processing time and light communication overhead; and (3) no need of anti-collision schemes. The 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 II, we describe the replicate-insensitive estimation problem. Section III presents our basic ideas, and Section IV describes the detailed protocol design. We present the performance evaluation in Section V. We discuss related work in Section VI and conclude this work in Section VII.

II. Problem Description

In order to improve coverage, many RFID systems deploy multiple readers with overlapping interrogation regions [16], so as 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 $r_1, r_2, ..., r_m$. Obviously, we have

$$MAX(r_1, r_2, ..., r_m) \leq n \leq SUM(r_1, r_2, ..., r_m) \quad (1)$$

By employing the existing estimators like USE and UPE [14], each reader can obtain an estimation result of the tags in its vicinity. Nevertheless, it is only possible to compute MAX$(r_1, r_2, ..., r_m)$ and SUM$(r_1, r_2, ..., r_m)$ to estimate the entire cardinality. In some cases, these two values are feasible estimators of n. For instance, if the target area is so small that each reader is able to cover the whole area, the MAX estimator is very close to n. Also, if the readers are deployed sparsely and the overlapping regions are rare, the SUM estimator is also acceptable. The problem in SUM estimator is that tags in the overlapping regions are read by multiple readers so that they are counted for multiple times. In other words, SUM estimator contains replicate information. The replicates make SUM larger than the exact value of n like the case shown in Fig. 1.

There are mainly two types of *multiple-reading*. The type in the example of Fig. 1 is called *spatial multiple-reading*, which means each tag is possibly recorded by multiple readers. The other type observed by us is called *temporal multiple-reading*, in which a tag is counted by the same reader for multiple times. For example, in Intelligent Transportation Systems (ITS) [17], a car passing by a reader twice will be recorded as two vehicles in the database. Temporal multiple-reading also happens in signal-reader context. In this work, we only consider spatial multiple-reading. Temporal multiple-reading can be analyzed similarly.

Besides accuracy, other important metrics of RFID tag estimation algorithms include processing time and communication overhead, as fast speed is the major motivation of using estimation schemes to replace identification algorithms.

In following parts, we will introduce our replicate-insensitive estimation scheme LoF, which eliminates the replicates by hashing and OR operation. Furthermore, it remarkably saves the processing time and communication overhead compared with previous estimation schemes. 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 stores a number of hash values. The removal of these assumptions has impact on the accuracy of the result, but does not overthrow its correctness. There are no further assumptions about the number and placement of readers or tags.

III. Basic Principles for LoF Algorithm

In slotted ALOHA identification schemes, the reader creates an ALOHA frame with several time slots, and then add the frame length (number of time slots) into the interrogation message sent to tags nearby. Each tag randomly picks a time slot to transmit back. Since only the tags that transmit to slots without collision can be recognized, the reader has to keep sending requests until each tag is identified at least once.

Due to the possible collisions, the identification process needs more than one transmission round. Nevertheless, the collision information can help to estimate the cardinality, even though most tags are not identified. For example, if the frame length is 10, and 8 slots are idle (with no response), it is likely that the number of tags is no more than 3. Conversely, if nine of them are collision slots, the cardinality is mostly much larger. This is the basic idea of our design, called Linear Probabilistic Estimation (LPE), which shares some features with Linear Probabilistic Counting [18].

Suppose each reader r_j construct an ALOHA frame F_j with l time slots, and then broadcast 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 and range 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 kth 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 $k = 0, 1, 2....$ If the reader r_j hears no response (idle) in the time slot k, it sets $BM_j[k]$ as 0. Obviously, 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, we have the following result.

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

$$E(V_0) = le^{-n/l} \tag{2}$$

$$E(V_1) = l(1 - e^{-n/l}) \tag{3}$$

Proof: For the kth bit in the bitmap, it will become 0 as if no tags respond in the time slot k. We know that every tag has $1/l$ probability to respond to the time slot k. Therefore,

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

Hence we obtain,

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

Since $V_0 + V_1 = l$,

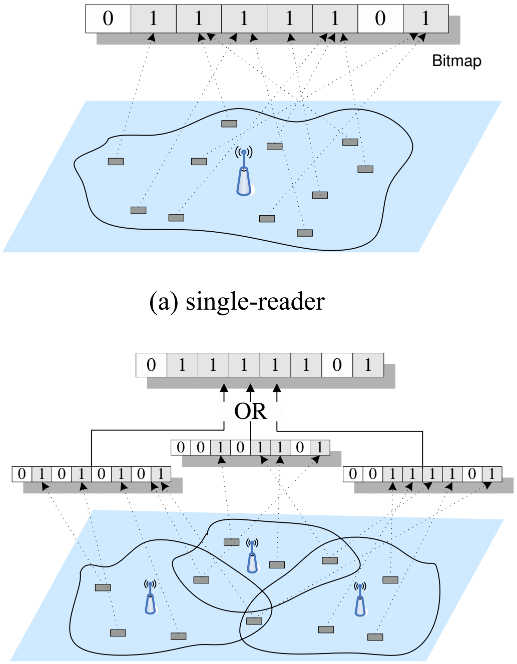
$$E(V_1) = l - E(V_0) = l(1 - e^{-n/l})$$

□

Replacing n and $E(V_0)$ by their representations in terms of observed variables $\tilde{n}$ and V_0, we get,

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

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



(a) single-reader

(b) multi-reader

Fig. 2. Examples of LPE

Figure 2(a) gives an example of Estimator 1 in single-reader scenarios. Every tag hashes itself to a time slot of a frame with size $l = 8$. After hearing the whole frame, the reader knows that two slots are with no transmission, three slots are with one transmission and the other three slots are with collision. Therefore, $V_0 = 2$ and $V_1 = 6$. We may estimate the number of tags based on formula (4).

$$\tilde{n} = -l \log(V_0/l) = -8 \times \log(2/8) = 11.0904$$

The estimated value of the tag number is 11.0904, which is very close to the exact number 11.

The property of hashing is suitable for eliminating replications, 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, a reader reports its bitmap to the central server. When 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. An example of this process in multi-reader RFID systems is illustrated in Fig. 2(b).

We claim that Estimator 1 is a replicate-insensitive estimation scheme in multi-reader scenarios and prove the correctness as follows.

Lemma 3.2: Suppose two tag sets S_1 and S_2 are in the vicinities of two readers r_1 and r_2 respectively. S_1 and S_2 share common members, i.e., $|S_1 \cap S_2| > 0$. The estimator of the merged bitmap equals to the estimation result of tag set $S_1 \cup S_2$.

Proof: Suppose the bitmaps generated by r_1 and r_2 are BM_1 and BM_2. The server obtain a third bitmap $BM_3 = BM_1$ OR BM_2.We further assume there was a super RFID reader which can cover the tag set $S_1 \cup S_2$, and the bitmap of it is BM_S. In order to prove Lemma 2, we have to show that $BM_3 = BM_S$.

For any bit $BM_3[k]$ in BM_3,

$$\begin{aligned} BM_3[k] = 1 &\iff BM_1[k] = 1 \vee BM_2[k] = 1 \\ &\iff \exists t_i \in S_1, H(i) = k \\ &\quad\ \vee \exists t_i \in S_2, H(i) = k \\ &\iff \exists t_i \in S_1 \cap S_2, H(i) = k \\ &\iff BM_S[k] = 1 \end{aligned}$$

Similarly, $BM_3[k] = 0 \iff BM_S[k] = 0$.
Therefore, $BM_3 = BM_S$ holds. □

We further have,

Lemma 3.3: Suppose tag sets $S_1, S_2, ..., S_m$ are in the vicinities of m readers $r_1, r_2, ..., r_m$ respectively. They share common members. The estimator $\tilde{n}$ of the merged bitmap equals to the estimation result of tag set $S_1 \cup S_2 \cup ... \cup S_m$.

The proof of Lemma 3.3 is similar to that of Lemma 3.2.

Theorem 3.4: Estimator 1 is a replicate-insensitive estimation in multi-reader scenarios.

Proof: As stated by Lemma 3.3, the estimator $\tilde{n}$ equals to the estimation of tag set $S_1 \cup S_2 \cup ... \cup S_m$. Since every ID in tag set $S_1 \cup S_2 \cup ... \cup S_m$ is distinct, Estimator 1 eliminates all the replicates. □

Before the idea can be implemented, however, some issues must be addressed. Note that the assumption we give for $H(key)$ to be uniformly distributed does not always hold in real circumstances. Since normally the hash values will have an uneven distribution depending on the tag set under estimation. Therefore, we provide the standard error for Estimator 1 by theoretical analysis. Here the standard error means the standard deviation of the ratio $\tilde{n}/n$.

From Lemma 3.1, we know that

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

Hence,

$$Var(V_0) = E(V_0^2) - E(V_1^2) = le^{-n/l}(1-(1+n/l)e^{-n/l})$$

Then we obtain the standard error,

$$Std(\tilde{n}/n) = \sqrt{Var(\tilde{n}/n)} = \frac{\sqrt{m}(e^{-n/l} - n/l - 1)^{1/2}}{n}$$

Table I provides the frame length l needed for estimation the number of tags with a standard error less than 10%.

TABLE I
FRAME LENGTH NEEDED FOR STANDARD ERROR OF 10%

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

As shown in Table I, the optimal frame length increases with the cardinality of tag set. As the total number of tags is unknown, the length will be initially set to be a small value. If the readers realize that too many collisions happen in the frame, they should extend the frame, e.g., double the length, then retransmit. Such process has to repeat in order to get an optimal frame size for estimation. The same problem happens in existing estimation algorithms such as USE and UPE [14]. As analyzed by the authors, to get an accurate estimation of 500 tags by UPE, the length of frame needed is 180. Using USE, the number even increases to about one thousand. Both of retransmissions and the large frame size enlarge the processing time and communication overhead.

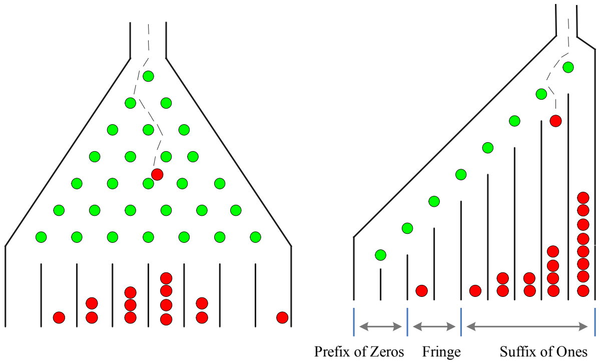
In our design, LoF employs the hash functions with geometric distribution rather than uniform distribution. Taking the advantage of this kind of hash functions, LoF has fixed and short frame length so as to reduce the latency and communication overhead in finalizing an optimal frame length.

IV. The Design of LoF Algorithm

In this section, we provide our replicate-insensitive estimation algorithm *Lottery Frame* (LoF), which combines LPE with geometric distribution, thus saving the processing time and communication overhead compared with USE, UPE and LPE.

A. Geometric Distribution and GD Galton Board

By modifying the famous statistical device Galton Board [20], we build a machine *Geometric Distribution Galton Board* (GD-Galton Board) to simulate the geometric distribution. For easy to understand, let us see the example in Fig. 3(a), where we drop balls into Galton board. Each time when a ball hits one of the nails, it can



(a) The Galton Borad (b) The GD-Galton Borad

Fig. 3. Galton Board and GD-Galton Board

```
INPUT
the number of hashed value num
OUTPUT
The selected slot number k

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
```

(a) the algorithm running on tags

```
INPUT
The number of hash functions m
OUTPUT
The set of bitmaps $BM_1$, $BM_2$, …, $BM_m$.

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 solt i
            set $BM_j[i] = 0$;
        else
            set $BM_j[i] = 1$;
        end if
    end for
end for
report $BM_1$, $BM_2$, …, $BM_m$ to the server;
```

(b) the algorithm running on readers

Fig. 4. Pseudocode of LoF

bounce right or left with 50% probability respectively. If the number of balls is sufficiently large, the heights of the ball heaps will approximately be a normal distribution. If we revise the structure of the device as shown in Fig. 3(b), by hitting a nail, a ball has 50% probability to drop into the right side slot, and 50% probability to bounce left. As a result, if the number of balls is sufficiently large, the heights of the ball heaps will follow geometric distribution, i.e., $1/2^t$ of the balls are in the tth slot. Imagine the ball heaps as a bitmap, we can point out three parts: the suffix of ones, the prefix of zeros and the fringe consists zeros and ones. Statistically, more balls drop, more left the fringe will be. Therefore, the position of fringe can be used to estimate the number of balls.

B. The General Protocol

In the rules of lotteries, it is very easy for a player to win a small prize, while winning a higher class of prize needs really good luck. Good lucks are rare, so we are able to estimate the number of tickets a player bought based on the prizes he wins. Suppose a lottery has several classes of prizes. If the player wins quite a lot different high classes of prizes, we can guess that he bought a great number of tickets.

Inspired by such life experiences, we design our replicate-insensitive estimation algorithm LoF. We design a lottery-style game using GD-Galton board. 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 drops a ball into GD-Galton board. The more left slot the ball stands in, the higher prize it wins. Here the GD-Galton board actually plays the role as a geometric distribution hash function $H()$. We estimate the cardinality based on the position of fringe in GD-Galton board.

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 slots according to the result, i.e., the prize it wins. 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 least-significant bit (bit 0) has 50% probability to be zero. Also, it can be easily proved that $1/2^t$ of the IDs are hashed to the slot $t-1$.

To make the implementation convenient, we just write the value $H(ID)$ onto the tags in production. 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. Finally, tags respond readers by a short message without any identifiable information.

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

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. For LoF, readers should know the number of hash functions used, but the length of frame is fixed. We illustrate our distributed algorithm for readers in Fig. 4(b).

3) Server: We should expect, if there are n tags, approximately $n/2$ of the responses are in time slot 0, approximately $n/4$ of the responses are in time slot 1and $1/2^t$ of the responses are in time slot $t-1$. Thus, by merging the bitmaps by OR operation, the kth 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 $log_2 n$.

By modifying a theorem suggested by Flajolet and Martin [19], we have the following Lemma which offers a relationship between the position of fringe in bitmap BM and the total number of tags n.

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

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

where the constant $\varphi = 0.775351...$ 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 [19]. 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

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

Formula (6) also justifies the statement we suggested that the fringe consists zeros and ones for the bit k whose value is near $\log_2 n$. Moreover, since tags use hash function to select time slots for responding, the LoF algorithm is also replicate-insensitive in multi-reader RFID systems.

Theorem 4.2: LoF is a replicate-insensitive estimation in multi-reader scenarios.

The proof is same as that of Theorem 3.4.

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

Theorem 4.3: A frame with $\log_2 N$ slots is sufficient for the estimation algorithm 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. □

Suppose there are totally 50000 tags produced. According to Theorem 3, we only have to fix the length of ALOHA frames as $\log_2 50000 = 15.60$. Compared with hundreds and thousands time slots needed for USE, UPE and Estimator 1, LoF evidently saves the processing time. Moreover, there is no constraint of operation range for LoF. Using frames with fixed length of 16, LoF can estimate the number of tags up to 50000. Hence, LoF is highly scalable.

C. Multiple Hash Functions

Based on the analysis in [19], we may also derive the standard deviation of the geometric distributed bitmap counting as the following theorem.

Theorem 4.4: Suppose R is the position of the right most zero in BM. The standard deviation of R satisfies

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

where the constant $\sigma = 1.1213...$ and $Q(u)$ is a periodic functions of u with mean value 0 and period 1.

An error with approximately 1.12 bit might be acceptable for some applications, e.g., to realize the size of the tag set for tag identification. It is however too high for many other applications. Luckily, the estimator used in LoF algorithm is an unbiased estimator. In that sense, if we make several independent estimations and compute the average result, the standard deviation will be significantly reduced. In LoF algorithm, 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 $R_1, R_2, ..., R_m$. Consider the average value

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

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

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

Table II lists the standard error in terms of the number of independent hash functions.

TABLE II
STANDARD ERROR FOR MULTIPLE HASH FUNCTIONS

M	1	2	4	8	16	32	128
Std Error	117%	73%	47%	32%	20%	15%	7%

D. Comparison with other Methods

We summarize LoF in applicability, processing time, communication overhead, space and computation complexity. In multi-reader scenarios, LoF can support replication-insensitive estimation. Assume the number of readers is constant. According to Theorem 4.3, for each communication round the length of slotted ALOHA frame is $O(\log n)$. Employing multiple hash functions, the communication round will be a constant number based on the accuracy desired. Consequently for the entire LoF algorithm using multiple hash functions, the processing time is also $O(\log n)$. Readers and the server only have to store bitmaps with the same length as frames. Thus the space cost is also $O(\log n)$. The complexity for computing the final result is $O(\log n)$, because LoF should scan each bitmap to find the right most zero. For the fairness of comparison, we multiply the number of hash functions, k, to each value. Table III compares LoF with other four RFID tag counting approaches: identification (Iden), USE, UPE [14] and LPE.

TABLE III
COMPARISON OF COUNTING/ESTIMATION SCHEMES

	Repli.-insen.	**Frame Length**	**Process Time**	**Space**	**Comp.**
Iden	YES	$O(n)$	$O(n^2)$	$O(n \log n)$	$O(n \log n)$
USE	NO	$O(n)$	$O(n \log n)$	$O(n)$	$O(n)$
UPE	NO	$O(n)/p$	$O(n \log n)$	$O(n)$	$O(n)$
LPE	YES	$O(n)$	$O(n \log n)$	$O(n)$	$O(n)$
LoF	YES	$O(k \log n)$	$O(k \log n)$	$O(k \log n)$	$O(k \log n)$

V. 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.

A. Simulation Methodologies and Performance Metrics

In our simulator, 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 there is idle, single reply or collision in any frame slot. In case there are multiple readers, at most one reader is operating at any time. Each tag is engineered to have globally unique IDs and a set of pre-computed 32-bit hash values. In LoF, they are able to pick the right hash values for frame slot selection.

We compare the estimation schemes in two scenarios: single-reader environment and multi-reader environment. For the single-reader environment, we contrast LoF with UPE, the main strategy in [14]. UPE does not support estimation in multi-reader environments, so we compare LoF with the two statistical values, MAX and SUM. Each reading in MAX and SUM is generated by UPE. Since the accuracy of LoF increases by adding more hash functions, we also evaluate the LoF protocol for different number of hash functions in both scenarios. Table IV shows the setup of our simulator. The simulation takes 100 runs with the same parameters, and we report the average.

TABLE IV
SIMULATION SETUP

	Single-reader	**Multi-reader**
Schemes evaluated	LoF and UPE	LoF, MAX and SUM
Metrics	accuracy and latency	accuracy and latency
Tag size	50-50000	5000 and 50000
Number of readers	1	4
Terrain	N/A	100 x 150 rectangular
Interrogation range	infinite	50-70
Repeating times	100	100

The estimation accuracy is illustrated in terms of error and normalized variance. Suppose the actual cardinality is n and the estimated value is $\tilde{n}$. Error and normalized variance are defined as

$$Error = \frac{|\tilde{n} - n|}{n}$$

$$NorVar = \frac{1}{100} \sum \frac{|\tilde{n} - n|}{n}$$

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

Another metric we concern is latency. Tag cardinality estimation requires several transmission rounds among

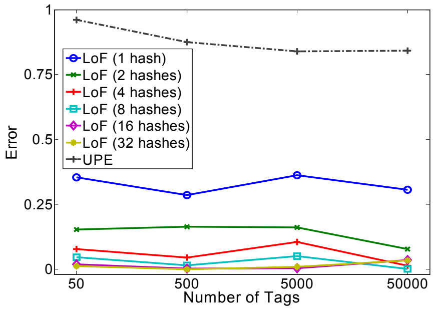


Fig. 5: Accuracy Comparison in Single-reader Context

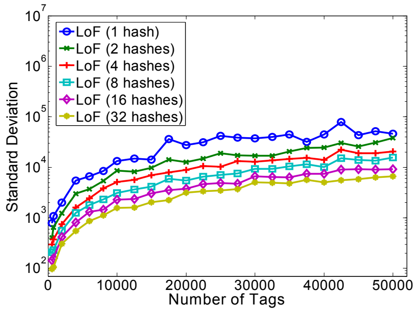


Fig. 6: Standard Deviation for LoF with Multiple Hashes

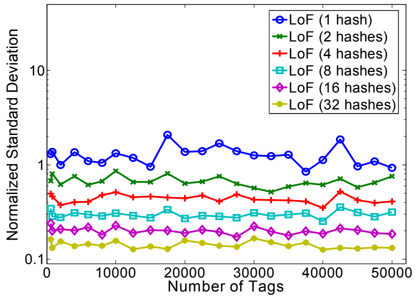


Fig. 7: Normalized Standard Deviation for LoF with Multiple Hashes

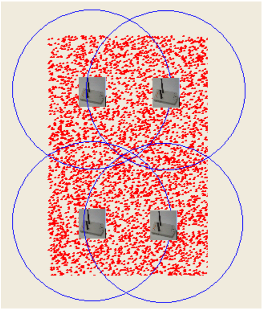


Fig. 8: Simulation Model

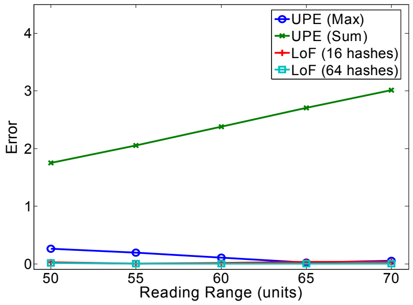


Fig. 9: Accuracy Comparison in Multi-reader Context (5000 tags)

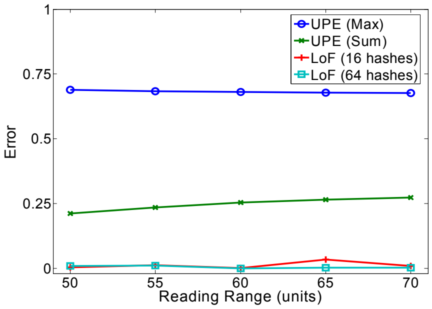


Fig. 10: Accuracy Comparison in Multi-reader Context (50000 tags)

readers and tags. For each round, readers take a number of time slots according to frame length. We abstract the estimation time as the total number of frame slots of the whole process.

B. Single-reader Scenario

Figure 5 plots the estimation accuracy of LoF and PUE. With more hash functions used in cardinality estimation, the error reduces. Employing 16 hash functions, the error is down to less than 0.04, i.e. 4% deviation from the actual cardinality. The change of actual tag size has no significant effect to LoF. In other words, the same frame length is suitable for a wide range of tag size. For UPE, the error is always greater than 0.80.

We are also interested in the accuracy distribution in each simulation. Figure 6 and 7 show the standard deviation and normalized standard deviation of estimation accuracy of LoF. When more hash functions are used in the estimation, the standard deviation of the estimation drops. Employing 16 hash functions, the normalized standard deviation is not more than 0.30. The number of tags affects little on the variation as illustrated in Fig. 7.

LoF employs fixed frame length. Therefore, independent on the actual tag cardinality, the total number of time slots required is equal to the frame length multiplied by the number of hash functions, i.e., $32m$ in our simulation as shown in Table V. On the other

TABLE V
REQUIRED TIME SLOT NUMBER FOR LOF

Number of Hashes	1	2	4	8	16	32
Time Slots	32	64	128	256	512	1024

TABLE VI
REQUIRED TIME SLOT NUMBER FOR UPE

Number of Tags	50	500	5000	10000	30000	50000
Time Slots	62	461	3215	4570	11045	23291

hand, Table VI shows that the frame slot requirement of UPE depends on the actual tag cardinality, which is relatively long. Besides, without prior knowledge of the approximate cardinality order, UPE suffers from huge expense in finalizing an optimal length.

Compared with UPE, LoF provides tunable estimation accuracy with expense on estimation time and frame slots requirements. Also, users are capable to predict the number of time slots required.

C. *Multi-reader scenario*

We then consider multiple readers which have overlapping interrogation regions. Different number of tags (5000 and 50000) is randomly distributed in a 100 units x 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. 8.

Figure 9 and 10 provide the accuracy of LoF, MAX and SUM in multi-reader environment, where each reading in MAX and SUM is generated by UPE. They indicate that MAX and SUM estimators perform poorly in cardinality estimation no matter which single reader algorithm is used. 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. 9, 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. When only 16 hash functions are used, the estimation error is already less than 4%. The estimation error is within 2% when 64 hash functions are used. Comparing Fig. 9 and Fig. 10, we again find that the number of tags has no significant impact on accuracy of LoF.

TABLE VII
REQUIRED TIME SLOT NUMBER FOR LOF

Number of hashes	1	4	16	32	64
Time Slots	128	512	2048	4096	8192

TABLE VIII
REQUIRED TIME SLOT NUMBER FOR UPE

	5000 tags			50000 tags		
Range	50	60	70	50	60	70
Time Slots	3600	3600	3630	33291	50370	71250

Table VII and Table VIII show the frame slot requirements for LoF and PUE scheme. The results are similar to the single reader case.

VI. RELATED WORK

To improve the coverage in large-scale RFID systems, multiple readers are often deployed in the region of interest. In a recent work [16], the author analyzed three types of collisions in multi-reader environment: tag-tag collision, reader-tag collision and reader-reader collision. The work focuses on mitigating reader-tag and reader-reader collisions. In our work, the term collision only refers to tag-tag collision. We assume no collisions in the other two types.

The existing RFID identification algorithms can be classified into two main categories based on which anti-collision scheme they use. The first type of RFID identification algorithms uses *Aloha-based anti-collision schemes* [11]. By employing such schemes, the reader creates a frame with a certain number of time slots, and then adds the frame length into the interrogation message sent to tags nearby. Each tag who receives the message randomly picks a time slot to transmit back. Since collisions in a same slot possibly happen and only the tags which transmit to slots without collision can be recognized, the reader has to keep sending requests until each of the tags is identified at least once. The other type of identification algorithms perform anti-collision by *Tree-based* or *Binary Splitting schemes* [12] [13]. Using Tree-based schemes, in each round, reader splits the set of tags into two subsets and labeled them by binary numbers. The reader repeats such process until each subset has only one tag. Thus the reader is able to identify all tags. The common drawback of these two types is the relatively long identification time.

To speed-up the identification protocols, Floerkemeier [15] suggests estimating the cardinality of tags based on the number of idle slots in current frame. A Bayesian probability estimation is used to maximize throughput. The most recent RFID tag estimation algorithms Unified Simple Estimator (USE) and Unified Probabilistic Estimator (UPE) are proposed by Kodialam and Nandagopal [14]. In USE, the authors proposed three estimators and analyzed their operating range and accuracy. Moreover, UPE employs the probabilistic framed ALOHA protocol to further mitigate the limitation that the operating range of a frame is dependent on the frame size. Nevertheless, as analyzed in section III-B, such a limitation still exists. Besides, USE and UPE are vulnerable to multiple-reading because they support no schemes that can eliminate the replicate information.

VII. Conclusions

Counting the number of tags is a crucial task in large-scale RFID systems. In this work, we propose LoF, a replicate-insensitive estimation protocol that can eliminate multiple-readings. 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 counting latency and communication overhead. We are also planning to explore more practical issues in multi-reader RFID systems, such as efficient identification, object tracking, as well as authentication. Indeed, this design can be easily extended to other fields, e.g., wireless sensor networks.

VIII. Acknowledgement

This work is supported in part by the Hong Kong RGC grants HKUST6183/06E, HKUST6169/07E, and HKBU 1/05C, the National High Technology Research and Development Program of China (863 Program) under grant No. 2007AA01Z180, the National Basic Research Program of China (973 Program) under grant No. 2006CB303000, NSFC Key Project grant No. 60533110, and the HKUST Nansha Research Fund NRC06/07.EG01.

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] G. Avoine and P. Oechslin, "A Scalable and Provably Secure Hash-Based RFID Protocol," in Proceeding of *PerCom Workshops*, 2005.

[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] 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.

[15] C. Floerkemeier, "Transmission control scheme for fast RFID object identification," in Proceeding of *PerCom Workshops*, 2006.

[16] 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.

[17] A. D. Joseph, A. R. Beresford, J. Bacon, et al, "Intelligent Transportation Systems," *IEEE Pervasive Computing*, vol. 5, pp. 63-67, 2006.

[18] 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.

[19] P. Flajolet and G. N. Martin, "Probabilistic Counting Algorithms for Data Base Applications," *Journal of Computer and System Science*, vol. 31, 1985.

[20] V. V. Kozlov and M. Y. Mitrofanova, "Galton Board," *Regular Chaotic Dynamics*, vol. 8, pp. 431-439, 2002.