Temporally or Spatially Dispersed Joint RFID Estimation Using Snapshots of Variable Lengths

Qingjun Xiao
Key Lab of Computer Network and Information
Integration (Southeast University)
Ministry of Education, China
csqjxiao@seu.edu.cn

Min Chen Shigang Chen Yian Zhou
Department of Computer and Information
Science and Engineering
University of Florida, Gainesville, FL, USA
{min, sgchen, yian}@cise.ufl.edu

ABSTRACT

Radio-frequency identification (RFID) technology has been widely used in applications such as inventory control, object tracking, supply chain management. An important research is to estimate the number of tags in a certain area covered by readers. This paper extends the research in both temporal and spatial dimensions to provide much richer information for monitoring the dynamics of distributed RFID systems. More specifically, we are interested in estimating the joint properties of any two snapshots taken at arbitrary locations and arbitrary times in a system. With many practical applications, there is however little prior work on this problem. We propose a joint RFID estimation protocol based on a simple yet versatile snapshot construction. Given the snapshots of any two tag sets, although their sizes may be very different, we design a way to combine their information and more importantly derive formulas to extract the joint properties of the two tag sets from the combined information, with an accuracy that can be arbitrarily set. Through formal analysis, we determine the optimal system parameters that minimize the execution time of taking snapshots, under the constraints of a given accuracy requirement. Our simulation results show that the proposed protocol can reduce the execution time by multifold when comparing with the best alternative approach in the literature.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: [Distributed Systems]; C.4 [**Performance of Systems**]: [Measurement techniques]; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Performance, Algorithm

Keywords

RFID; Cardinality Estimation; Random Hashing

MobiHoc'15, June 22–25, 2015, Hangzhou, China.
Copyright © 2015 ACM 978-1-4503-3489-1/15/06 ...$15.00.
http://dx.doi.org/10.1145/2746285.2746289.

1. INTRODUCTION

Radio-frequency identification (RFID) technology has been widely used in various commercial applications, including inventory control, object tracking, supply chain management, auto-payment, etc [13, 7, 3, 9, 10]. RFID tags (each carrying a unique identifier) are attached to merchandise at retail stores, equipments at hospitals, or goods at warehouses, allowing an RFID reader to quickly identify products, access properties of each individual item, or collect statistical information about a group of items.

An important system function is called *RFID estimation* [5, 4, 12, 8, 22, 15, 16, 21, 2], which is to estimate the number of tags in a certain area covered by readers. This basic function can be used to monitor the inventory level in a warehouse, the sales in a retail store, and even the popularity of attractions in tourism [2]. It can also serve as a pre-processing step to make other functions (such as tag identification [18, 11, 6]) more efficient. RFID estimation takes much less time to perform than a full system scan that collects all tag IDs. This makes it valuable since RFID systems operate at low wireless rates and the execution time has been the key performance metric in system design. Moreover, it does not identify any tags, which avoids the privacy issue, particularly in scenarios where the party performing the operation (such as warehouse or port authority) does not own the tagged items.

Motivation: This paper extends RFID estimation in both temporal and spatial dimensions to provide much richer information about the dynamics of a distributed RFID system. We use two applications to explain the problems of temporally dispersed RFID estimation and spatially dispersed estimation, respectively. Consider the application of monitoring the inventory dynamics of a warehouse over time. We are interested in the amount of goods moving in (i.e., the number of new tags) and the amount moving out (i.e., the number of departure tags) between *any two* reference time points, without identifying the tag IDs, where the reference time points may be evenly spaced by time intervals of a certain length. The problem cannot be easily solved simply by estimating the number of tags in the warehouse after each time interval by traditional approaches [5, 4, 12, 8, 22, 15, 16, 21, 2]. For instance, if the number of tags at time 1 is estimated to be 1000 and so does the number at time 2, we will not be able to know whether no new tag has moved in or 1000 new tags have moved in while all old tags have moved out. To handle this problem, we need to take *snapshots* with more detailed information about existing tags, such that from any two snapshots taken at different times, we will be able to estimate the joint properties of the corresponding two tag sets, such as their union, intersection and difference, which provide information for stocking dynamics about product inflow and outflow.

In the second application, consider the supply chain management in a large logistics network. As tagged products are shipped from location (component factory, assembly line, warehouse, port, or other storage/retail facility) to location, if each location takes periodic snapshots of its tag set and keeps a series of such snapshots over time, we will be able to make queries for joint estimation between *any two* snapshots, which may be taken from different locations or from the same location at different times. Such joint estimation, when performed over time across the network, gives a picture about how goods flow through the logistics network. For one application, this information can help diagnose erratic shipments by identifying unexpected volumes that move over supply chains with significant deviation from a pre-established business plan — it has been reported that, due to such logistic errors, more than 65% of the inventory records did not match the physical inventory [14]. Without any automatic tools, we will have to resort to manual inventory check to discover the errors, which is laborious, expensive and slow, especially when such inventory task needs to be performed at daily basis.

Moreover, comparing with traditional RFID estimation [5, 4, 12, 8, 22, 15, 16, 21, 2] (which were designed to operate at a single time and a single place), the ability to jointly consider any two temporally or spatially dispersed snapshots will enable us to expand the applications mentioned earlier, for example, by providing more detailed information about changes in inventory and sales, by monitoring the flows of tourists moving from place to place in a theme park, or by serving as a pre-processing step to make some sophisticated functions such as continuous tag monitoring [17] more efficient.

Problem, Challenge and Prior Art: From the above applications, we abstract the problem of *joint RFID estimation*, which is to estimate the joint properties of two arbitrary sets of tags at different times or different locations in a large distributed RFID system. The joint properties include the cardinalities of the union, intersection and difference of the two sets.

The key challenge is that when a snapshot is taken for one tag set, we do not know which other set (at different time/location) the joint estimation will be made with. In fact, the snapshot can be paired with any other snapshot taken in the past or future in the system.

There is little prior work on this practically interesting problem. Directly related is the differential estimation method (DiffEstm) [20], which focuses on the difference between two sets and adopts a different problem model. It uniformly sets the sizes of all snapshots based on the worst-case situation so that any two can be paired. This is very inefficient because the tag sets in a system can have very diverse sizes. For example, in the previous logistics network application, a warehouse may sometimes be almost empty, while carrying tens of thousands of items at other times. Suppose the largest set the system can handle is 50,000. Even if a tag set at a certain time is down to hundreds, the size of its snapshot will still have to be set according to 50,000 in [20].

Our Contributions: First, we propose a new solution for the generalized joint RFID estimation problem based on a simple yet versatile snapshot construction. It takes only one pass of communication between a reader and tags to construct the snapshot of a given tag set. The size of the snapshot is roughly proportional to the size of the tag set, instead of being fixed to a large worst-case value. Given the snapshots of any two tag sets, although their sizes may be very different, we propose a way to combine their information and more importantly derive formulas to extract the joint properties of the two sets from the combined information.

Second, we formally analyze the means and variances of the estimated properties computed from the formulas. We show that the formulas produce asymptotically unbiased results and they estimate the joint properties with an absolute (probabilistic) error bound that can be set arbitrarily. We also derive the formulas for determining the optimal system parameters that minimize the execution time of taking snapshots, under the constraints of a given accuracy requirement for joint estimation.

Third, we perform extensive simulations to complement the theoretical analysis. The results show that by allowing the snapshots to have variable sizes, the new solution significantly outperforms the existing method. For example, under the same accuracy requirement, the new solution achieves about 240% improvement in execution time when comparing with DiffEstm.

2. PROBLEM DEFINITION

Consider a large distributed RFID system such as a supply-chain network, consisting of multiple locations, where tagged objects are shipped from location to location. At any time and any location, there is a set of tags. Consider two arbitrary sets of tags, N and N', at different locations or at the same location but different times. We study the joint properties of the two sets, including their intersection, union and difference.

Let $n = |N|$, $n' = |N'|$, $u = |N \cup N'|$, $m = |N \cap N'|$, $d = |N - N'|$, and $d' = |N' - N|$. With loss of generality, we assume N is a larger set than N', and hence $n \geq n'$. The *joint RFID estimation problem* is to provide estimations $\hat{u}$, $\hat{m}$, $\hat{d}$, and $\hat{d}'$ for u, m, d and d' respectively, such that the following pre-defined accuracy requirements are met:

$$Prob\{\hat{u} - k \leq u \leq \hat{u} + k\} \geq \alpha \tag{1}$$

$$Prob\{\hat{d} - k \leq d \leq \hat{d} + k\} \geq \alpha \tag{2}$$

$$Prob\{\hat{m} - k \leq m \leq \hat{m} + k\} \geq \alpha \tag{3}$$

$$Prob\{\hat{d}' - k \leq d' \leq \hat{d}' + k\} \geq \alpha, \tag{4}$$

where α is a probability value, and k is a probabilistic error bound. For example, if $\alpha = 95\%$ and $k = 100$, it requires that the absolute error of each estimation should be within k for a probability of no less than 95%.

An alternative way of specifying the estimation accuracy is based on a relative error $\epsilon \in (0, 1)$.

$$Prob\{\hat{u}(1-\epsilon) \leq u \leq \hat{u}(1+\epsilon)\} \geq \alpha \tag{5}$$

$$Prob\{\hat{d}(1-\epsilon) \leq d \leq \hat{d}(1+\epsilon)\} \geq \alpha \tag{6}$$

$$Prob\{\hat{m}(1-\epsilon) \leq m \leq \hat{m}(1+\epsilon)\} \geq \alpha \tag{7}$$

$$Prob\{\hat{d}'(1-\epsilon) \leq d' \leq \hat{d}'(1+\epsilon)\} \geq \alpha, \tag{8}$$

where the relative errors $\frac{\hat{u}-u}{\hat{u}}$, $\frac{\hat{d}-d}{\hat{d}}$, $\frac{\hat{m}-m}{\hat{m}}$ and $\frac{\hat{d}'-d'}{\hat{d}'}$ are bounded by $\pm\epsilon$ at a probability of at least α.

We do not adopt this model because it is very expensive or even impossible to achieve as the values of m, d and d' can be very small (down to zero). Consider $m = 0$. In this case, we will have to make sure that $\hat{m} = m = 0$ in order for (7) to be met, which means *precise measurement* of the empty intersection, i.e., $Var(\hat{m}) = 0$. Because $\hat{m}$ is derived from the two snapshots, these snapshots cannot carry any positive variance in their information based on which $\hat{m}$ is computed; note that the snapshots are independent due to $N \cup N' = \emptyset$. Recording precise information (such as IDs of all tags) is very expensive; all existing RFID estimation methods collect imprecise information from tags with non-zero variance to save time for information collection.

One critical problem is that at the time when the snapshot is taken for any tag set, we do not know which other snapshot it will be paired with for joint estimation. Because it is possible to pair with another snapshot with no common tag, we will have to make the snapshot precise (thus expensive) due to the requirement (7).

Finally, it arguably makes more sense to specify absolute error bound in some practical scenarios. Consider the logistics network application. Suppose we want to monitor the volume of products flowing from a number of factories to a number of assembly plants. Further suppose the volume from a particular factory to a particular plant may range from zero to ten of thousands in each pair of snapshots. To get a rough idea about the volume, we may specify the accuracy requirement as an absolute error bound of ±50 items with 95% confidence. If the actual volume is 10, even though the relative error will be large, it does not change the fact that the estimated volume remains very small, giving correct assessment. On the contrary, if we specify a relative error of 1% and the actual volume is 10, it will require the estimation to have an absolute error of ±0.1 item, which is not only expensive to achieve but also unnecessary. Note that in this example we estimate small intersection from snapshots of two large sets, not estimating the cardinality of one tag set from its snapshot (e.g, bitmap) as the traditional RFID estimation does.

3. PRIOR WORK

3.1 Differential Estimation

DiffEstm [20] gives a relative error model similar to (6)-(8) but does not prove that its estimation results meet those requirements. In fact, DiffEstm cannot always meet the relative error bound because it has positive variance in its snapshots, whereas the relative error model requires snapshots to carry precise information as we have argued previously.

We give a simplified description of DiffEstm's snapshot construction: A reader makes a request $(f, p, ...)$ to tags in its coverage area. After a tag receives the request, with a probability p, it will transmit in a slot randomly selected from an ALOHA frame of size f. The reader will turn the time frame into a bitmap snapshot of length f, with each busy slot being 1 and each idle slot being 0. In the original paper, the request carries a frame size F and a parameter f. Each tag transmits in a randomly chosen slot, and the reader only listens to the first f slots. This approach is equivalent to a frame size of f with a sampling probability $p = \frac{f}{F}$.

Figure 1 illustrates how DiffEstm works. After two bitmap snapshots (on the top of the figure) are taken for two tag sets, they are bitwise-ORed to produce a combined bitmap (at the bottom). The difference and intersection of the two sets will then be derived from the information in the three bitmaps, which must all contain a sufficient number of zeros to ensure estimation accuracy [20].

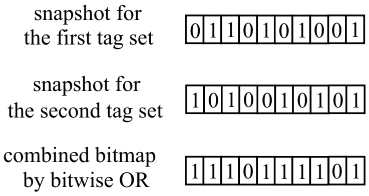
snapshot for the first tag set
0 1 1 0 1 0 1 0 0 1
snapshot for the second tag set
1 0 1 0 0 1 0 1 0 1
combined bitmap by bitwise OR
1 1 1 0 1 1 1 1 0 1

Figure 1: DiffEstm estimates the difference and intersection between two tag sets by combining their bitmap snapshots, which must have the same length and the same sampling probability.

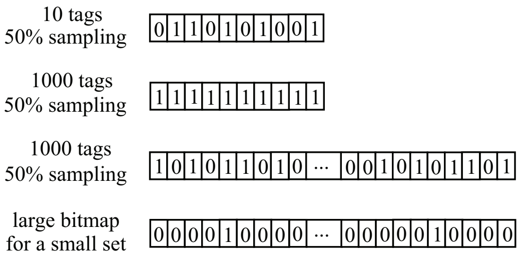

Figure 2: Large snapshots for small tag sets.

In order to support bitwise-OR, DiffEstm requires that all snapshots must have the same size and use a common sampling probability. For any small tag set, if the common sampling probability is very small, too few or even no tag will be sampled for the snapshot construction. Hence, the sampling probability will have to be reasonably large, as illustrated by the top bitmap in Figure 2, where 10 tags are recorded with 50% sampling probability. However, for a large set, a significant sampling probability will cause all bits to be set as ones (the second bitmap in the figure), unless the bitmap size is sufficiently large (the third bitmap in the figure). Now because the same large size has to be applied to all snapshots, it becomes a great waste for small tag sets (the fourth bitmap). Since each bit takes one time slot to get, a large bitmap size means a long time for taking a snapshot, even for a very small tag set.

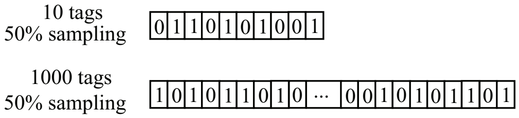

Figure 3: Snapshots of variable sizes.

Naturally, it is desirable to let each snapshot have a different size, depending on the size of the tag set it records, as illustrated in Figure 3. This will require us to develop new methods of combining two snapshots (or two bitmaps) with variable sizes. The real difficulty is not at how to combine two bitmaps per se; there are simple ways to combine them. The real difficulty comes after the combination — how to perform analysis on the information combined from non-uniform snapshots, how to use that information for joint estimation, and most importantly how to ensure the accuracy requirements in (1)-(4). These are the tasks that have not been investigated in the literature.

3.2 RFID Estimation and Union Estimation

There is a rich set of literature that estimate the cardinality n of a single tag set [5, 12, 4, 8, 22, 15, 16, 21, 2], typically giving an estimate $\hat{n}$ with a relative error model of

$$Prob\{\hat{n}(1-\epsilon) \leq n \leq \hat{n}(1+\epsilon)\} \geq \alpha, \qquad (9)$$

which is different from the model of joint estimation where intersection and difference between two sets are estimated. The execution time is a function of the relative error bound ϵ. For example, the time of LOF is $\mathcal{O}(\frac{1}{\epsilon^2}\log n)$ [12], that of PET is $\mathcal{O}(\frac{1}{\epsilon^2}\log\log n)$ [22], and that of ZOE is $\mathcal{O}(\frac{1}{\epsilon^2} + \log\log n)$ [21]. The recent work of two-phase simple RFID counting (SRC) [2] has the best performance to date.

When the tag set cannot be covered by a single reader, multiple readers will be needed, each covering a subset. Many of the RFID estimation solutions can be easily extended for estimating the union size of the subsets. Among them, SRC_M [2] performs the best, achieving a reduction in execution time by up to 300%, when comparing with others. SRC_M also uses bitmaps. For each subset, it create multiple bitmaps, each under a different sampling probability. It then identifies the best sampling probability and combines the bitmaps of that probability from different subsets by bitwise OR. The combined bitmap records all tags in the union and can thus be used to estimate the union cardinality with the method from [5].

What makes SRC_M efficient is that as it scans one subset after another, it leverages the information learned from the previous subsets to reduce the number of bitmaps (different sampling probabilities) it needs for each subsequent subset. However, this method cannot be extrapolated to temporally/spatiall dispersed joint estimation where we do not know which tag sets will be jointly estimated beforehand and thus cannot leverage one set's information to help reduce the overhead for the other.

If we nevertheless want to apply SRC_M to joint estimation, we may use a common sampling probability that is optimized for the worst-case scenario such that an error bound for the union estimation will always be met. In this case, SRC_M will become DiffEstm except that the former considers only union while the latter also addresses difference and intersection (which are more difficult to estimate and analyze).

4. JOINT RFID ESTIMATION PROTOCOL

This section presents our joint RFID estimation protocol (JREP). Our protocol consists of two components: an online encoding component for measuring the information of each tag set and storing it in a bitmap called *snapshot*, and an offline data analysis component for estimating the joint properties of two arbitrary sets such as intersection/union/difference cardinalities, using their snapshots. We use an asymmetric design to push most complexity to the offline component, while keeping the online component as efficient as possible.

4.1 Online Encoding

Consider a snapshot taken at an arbitrary location and an arbitrary time in a large RFID system of multiple locations. Let N be the set of tags existing at the location and the time when the snapshot is taken, and n be the number of tags in N. The reader that performs the snapshot will first get a rough estimation for the value of n by using an existing cardinality estimation protocol [8, 12, 22]. It determines a frame size f for the snapshot as follows:

$$f = \min_{p\in(0,1]}\{2^{\lceil\log_2(\frac{np}{\omega})\rceil}\}, \qquad (10)$$

where p is a sampling probability and

$$\omega = -\frac{3}{4} + \frac{\sqrt{3}}{4}\sqrt{4p\Big(\frac{k^2}{n_{\max}{Z_\alpha}^2} + 2\Big) - 5}.$$

We use Z_α to denote the $\frac{1+\alpha}{2}$ percentile for the unit normal distribution. For example, when $\alpha = 95\%$, $Z_\alpha = 1.96$. Later we will formally derive the above formulas that minimize the time overhead of online encoding and the storage overhead of the snapshot in the worst case, under the constraints of (1)-(4). Let p^* be the sampling probability that minimizes (10). The value of p^* only depends on $n_{\max}$, k, and α. Hence, it is pre-determined for a system once these parameters are set.

The encoding process is described as follows: The reader broadcasts an encoding request with parameters f and p^*. Upon receipt of the request, each tag decides with probability p^* whether it will participate in the encoding. If it does, it selects a slot uniformly at random and transmit a pulse during that slot. The reader monitors the status of each slot — with the detection of a pulse, it records the slot as a bit '1'; otherwise, it records a bit '0'. After the frame, the reader has a bitmap of ones and zeros, which will be stored and used later for joint estimation. We call this bitmap as a snapshot of the tag set N.

The implementation of sampling may be done as follows: Let M be a large integer. The reader broadcasts an integer $\lfloor p^*M\rfloor$ instead of a floating number p^*. A tag computes a pseudo-random value $R(ID)$, where ID is the tag's identifier and R is a pseudo-random function (required by the C1G2 standard). The tag is sampled if $R(ID) \bmod M < \lfloor p^*M\rfloor$. The slot selection also leverages the random function R. The tag computes $R(ID\,|\,r) \bmod f$, where r is a randomly-chosen constant pre-configured with all tags, to make the values of $R(ID\,|\,r)$ and $R(ID)$ independent of each other. With these implementations, we have the following property established.

Property 1. *Consider an arbitrary common tag in any two sets N and N', whose frame sizes are f and f' respectively, with $f \geq f'$. A tag is either sampled for encoding in both frames or neither. $\forall j \in [0, f)$, if the tag is sampled and does not select the $(j \bmod f')$th slot in the frame of f', then it will not select the jth slot in the frame of f.*

Proof. It is easy to see that the tag will be either sampled for both frames or neither, because the same pseudo-random value $R(ID)$ is used for sampling.

Suppose the tag does not select the $(j \bmod f')$th slot in the frame of f'. That is, $R(ID\,|\,r) \neq j \mod f'$. Because both f and f' are the powers of two and $f \geq f'$, f' must be able to divide f. Hence, $R(ID\,|\,r) \neq j \mod f$, which means the jth slot in frame of f is not selected. □

4.2 Offline Joint RFID Estimation

Given two arbitrary snapshots, B and B', which may be taken at the same location but different times or at different locations, we give the formulas for estimating their difference, intersection and union.

4.2.1 Expanded OR

Let f and f' be the sizes of B and B', respectively. Without losing generality, suppose $f \geq f'$. According to (10), we know that both f and f' are the powers of two. The reason for them to be powers of two is to support the following operation that combines the information from the snapshots of two arbitrary tag sets for joint estimation.

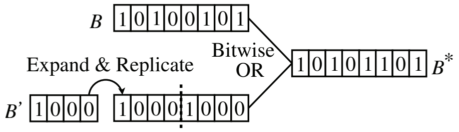

Figure 4: Expanded OR of two bitmaps B and B'.

We introduce an auxiliary bitmap, which is called the *expanded OR* between B and B', and is denoted as B^*. The expanded OR, which has been illustrated in Fig. 4, is defined as follows: We know that f is a multiple of f'. If $f \neq f'$, we will replicate B' for $\frac{f}{f'}$ times, such that it is expanded to the same length of B. We then perform bitwise OR between the two bitmaps, and the resulting B^* is f bits long.

The operation of expanded OR may appear to be trivial, but the implication of replicating the information of one bitmap when combining with another bitmap is not obvious at all. It requires rigorous analysis for its impact on estimation accuracy as the technique was never used in RFID estimation before.

Let N be the original tag set that is encoded by bitmap B. The size of N is denoted as n. Let X_j, $0 \leq j < f$, be the event that the jth bit in B remains zero after the n tags are randomly sampled and encoded into this bitmap.

$$Prob\{X_j\} = (1 - \frac{p}{f})^n \tag{11}$$

Let V be a random variable for the fraction of bits in B that are zeros (We can also measure an instance value of V from the snapshot B. This instance value will be used in the estimator derived later). We have

$$V = \frac{1}{f}\sum_{i=0}^{f-1} 1_{X_j},$$

where 1_{X_j} be the indicator variable of X_j, whose value is 1 when the event X_j occurs and 0 otherwise. Clearly, $E(1_{X_j}) = Prob\{X_j\}$. Hence,

$$E(V) = \frac{1}{f}\sum_{i=0}^{f-1} E(1_{X_j}) = \frac{1}{f}\sum_{i=0}^{f-1} Prob\{X_j\} = (1-\frac{p}{f})^n. \tag{12}$$

Let N' be the tag set encoded by B', n' be the set size, and Y_j, $0 \leq j < f'$, be the event that the jth bit in B' is zero.

$$Prob\{Y_j\} = (1 - \frac{p}{f'})^{n'} \tag{13}$$

The above equation is also true for any bit in the expanded B' (for producing B^*). Let Z_j, $0 \leq j < f$, be the event that the jth bit in B^* is zero. Since this bit is OR of the jth bit in B and the (j mod f')th bit in B', the event Z_j happens when both events X_j and $Y_{j \bmod f'}$ happens. Hence, we have

$$\begin{aligned} Prob\{Z_j\} &= Prob\{X_j \wedge Y_{j \bmod f'}\} \\ &= Prob\{X_j \,|\, Y_{j \bmod f'}\} \cdot Prob\{Y_{j \bmod f'}\} \\ &= (1 - \frac{p}{f})^d (1 - \frac{p}{f'})^{n'}, \end{aligned} \tag{14}$$

where $Prob\{Y_{j \bmod f'}\} = (1 - \frac{p}{f'})^{n'}$ is from Eq. (13), and we have $Prob\{X_j \,|\, Y_{j \bmod f'}\} = (1 - \frac{p}{f})^d$, because the condition $Y_{j \bmod f'}$, according to Property 1, suggests that all common tags of N and N' won't select the jth slot in the frame of f, and consequently only the d tags in $N - N'$ may select this slot.

Following a similar process of deriving (12), we have

$$E(V') = (1 - \frac{p}{f'})^{n'} \tag{15}$$

$$E(V^*) = (1 - \frac{p}{f})^d (1 - \frac{p}{f'})^{n'}, \tag{16}$$

where $V' = \frac{1}{f'}\sum 1_{Y_j}$ and $V^* = \frac{1}{f}\sum 1_{Z_j}$.

In the following, we give the estimators for the joint properties of N and N', including d, d', m, and u.

4.2.2 Estimator of Set Difference $d = |N - N'|$

Replacing the first term $(1-\frac{p}{f'})^{n'}$ in (16) by $E(V')$, we have

$$E(V^*) = E(V')(1 - \frac{p}{f})^d,$$

$$d = \frac{\ln E(V^*) - \ln E(V')}{\ln(1 - \frac{p}{f})}. \tag{17}$$

Replacing $E(V^*)$ and $E(V')$ by their instance values V^* and V' measured from B^* and B', we obtain the estimator $\hat{d}$.

$$\hat{d} = \frac{\ln V^* - \ln V'}{\ln(1 - \frac{p}{f})} \tag{18}$$

Such replacement with instance values produces asymptotically unbiased results, which originates from the central limit theorem and the multivariate δ-method [1], as explained below. Since V' (or V^*) is the arithmetic mean of a large number of independent random variables $V' = \frac{1}{f'}\sum 1_{Y_j}$ (or $V^* = \frac{1}{f}\sum 1_{Z_j}$), according to the central limit theorem, it approximates a Gaussian distribution, and its variance is inversely proportional to the number of random variables f' (or f). Further consider that $\hat{d}$ in (18) is a function of V' and V^* with continuous first partial derivatives. We can apply the multivariate δ-method, and conclude that the estimation $\hat{d}$ approximates a Gaussian distribution, whose expected value is $E(\hat{d}) \approx \frac{\ln E(V^*) - \ln E(V')}{\ln(1-\frac{p}{f})}$. Combined with (17), we have $E(\hat{d})$ equal to d asymptotically, when f and f' are large enough. Later, we will derive the mean of $\hat{d}$ with better accuracy.

Below we use a similar approach to derive the estimators of d', m and u.

4.2.3 Estimator of Set Difference $d' = |N' - N|$

Based on the definitions of d and d', we know that $d = n + d' - n'$, where $n + d'$ is the number of tags in $|N \cup N'|$. Applying it to (16), we have

$$\begin{aligned} E(V^*) &= (1 - \frac{p}{f})^{n+d'-n'} (1 - \frac{p}{f'})^{n'} \\ &= E(V)(1 - \frac{p}{f})^{d'} E(V') \,/\, (1 - \frac{p}{f})^{n'} \\ &= E(V)(1 - \frac{p}{f})^{d'} E(V') \,/\, E(V')^{\ln(1-\frac{p}{f})/\ln(1-\frac{p}{f'})}. \end{aligned}$$

Solving the above equation for d', we have

$$d' = \frac{\ln E(V^*) - \ln E(V) - \ln E(V')}{\ln(1 - \frac{p}{f})} + \frac{\ln E(V')}{\ln(1 - \frac{p}{f'})}.$$

Similar to the previous estimator, we can substitute the expected values $E(V)$, $E(V')$ and $E(V^*)$ by their instance values V, V' and V^* measured from B, B' and B^*, and have an asymptotically unbiased estimator $\hat{d'}$ of approximately Gaussian distribution for the intersection cardinality d'.

$$\hat{d'} = \frac{\ln V^* - \ln V - \ln V'}{\ln(1-\frac{p}{f})} + \frac{\ln V'}{\ln(1-\frac{p}{f'})} \tag{19}$$

4.2.4 Estimator of Set Intersection $m = |N \cap N'|$

We rewrite the expected value of V^* in (16) as

$$E(V^*) = (1-\frac{p}{f'})^{n'}(1-\frac{p}{f})^{n-m} = E(V')E(V)(1-\frac{p}{f})^{-m}$$

$$m = \frac{\ln E(V) + \ln E(V') - \ln E(V^*)}{\ln(1-\frac{p}{f})}.$$

Substitute the expected values $E(V)$, $E(V')$ and $E(V^*)$ by their instance values V, V' and V^*, we have an asymptotically unbiased estimator $\hat{m}$ of approximately Gaussian distribution for the intersection cardinality m.

$$\hat{m} = \frac{\ln V + \ln V' - \ln V^*}{\ln(1-\frac{p}{f})} \tag{20}$$

4.2.5 Estimator of Set Union $u = |N \cup N'|$

Multiplying both sides of Eq. (16) with $(1-\frac{p}{f})^{n'}$, we have

$$E(V^*)(1-\frac{p}{f})^{n'} = (1-\frac{p}{f})^{d+n'}E(V')$$

$$E(V^*)E(V')^{\ln(1-\frac{p}{f})/\ln(1-\frac{p}{f'})} = (1-\frac{p}{f})^{u}E(V')$$

$$u = \frac{\ln E(V^*) - \ln E(V')}{\ln(1-\frac{p}{f})} + \frac{\ln E(V')}{\ln(1-\frac{p}{f'})}.$$

Substitute the expected values $E(V')$ and $E(V^*)$ by their instance values V' and V^*, we have an estimator $\hat{u}$ of approximately Gaussian distribution for the union cardinality.

$$\hat{u} = \frac{\ln V^* - \ln V'}{\ln(1-\frac{p}{f})} + \frac{\ln V'}{\ln(1-\frac{p}{f'})} \tag{21}$$

4.2.6 Traditional Estimator of $n = |N|$ and $n' = |N'|$

We estimate n and n' based on the classical work in [19]:

$$\hat{n} = \frac{\ln V}{\ln(1-\frac{p}{f})} \qquad \hat{n'} = \frac{\ln V'}{\ln(1-\frac{p}{f'})}, \tag{22}$$

where $\hat{n}$ and $\hat{n'}$ denote the estimated values.

4.2.7 Reduction to DiffEstm [20] and PZE [5]

It is interesting to see that when we set $f = f'$, the estimators (18), (19) and (20) are reduced to the estimators of DiffEstm. If we further set the sets identical, i.e., $N = N'$, the union estimator (21) becomes the PZE estimator in [5], similar to those in (22) for a single set. Hence, DiffEstm and PZE are special cases of our estimators. Note that PZE repeats a small frame with a small sampling probability many times to reduce estimation variance. Here we use a larger bitmap (with a larger sampling probability) to reduce variance. The two approaches are equivalent [2].

The most fundamental difference in (18), (19), (20) and (21) from the prior art is the generalized semantics of $V*$, which is the fraction of bits that are zeros in the bitmap B^* combined from two bitmaps of variable sizes, f and f', where each bit in the smaller bitmap has to be used multiple times in order to enable bitwise OR. However, it is not intuitive why this multiple use of the same bits will work in estimation until we formally analyze the estimation accuracy under such a maneuver of combining information from non-uniform bitmaps.

5. ANALYSIS

In this section, we analyze the accuracy of the joint estimations $\hat{d}$, $\hat{d'}$, $\hat{m}$ and $\hat{u}$. To derive them, we will need the mean and variance of $\hat{n}$ ($\hat{n'}$), which can be found in [19]:

$$E(\hat{n}) \approx n + \frac{1}{2p}(e^{\omega} - \omega p - 1) \tag{23}$$

$$Var(\hat{n}) \approx \frac{f}{p^2}(e^{\omega} - \omega p - 1), \tag{24}$$

where $\omega = \frac{pn}{f}$ is called the *load factor* of bitmap B;

$$E(\hat{n'}) \approx n' + \frac{1}{2p}(e^{\omega'} - \omega' p - 1) \tag{25}$$

$$Var(\hat{n'}) \approx \frac{f'}{p^2}(e^{\omega'} - \omega' p - 1), \tag{26}$$

where $\omega' = \frac{pn'}{f'}$ is the load factor of bitmap B'.

5.1 Mean and Variance of $\hat{d}$

We first derive $E(\hat{d})$ and $Var(\hat{d})$. From (18) and (22), we rewrite the formula for $\hat{d}$ as

$$\hat{d} = \frac{\ln V^*}{\ln(1-\frac{p}{f})} - \hat{n'} \cdot \frac{\ln(1-\frac{p}{f'})}{\ln(1-\frac{p}{f})}. \tag{27}$$

The statistical properties of $\hat{n'}$ are known. The values of p, f, f' are also known. The first term is denoted as $\hat{n^*}$:

$$\hat{n^*} = \frac{\ln V^*}{\ln(1-\frac{p}{f})}. \tag{28}$$

We have derived the mean and variance of $\hat{n^*}$. Due to space limitation, we omit the process, which will be reported in a longer journal version.

$$E(\hat{n^*}) \approx n^* + \frac{1}{2p}(e^{\omega^*} - \omega^* p - 1)$$

$$Var(\hat{n^*}) \approx \frac{f}{p^2}(e^{\omega^*} - \omega^* p - 1),$$

where

$$n^* = d + n'\frac{f}{f'}, \qquad \omega^* = \frac{pd}{f} + \frac{pn'}{f'}, \tag{29}$$

and ω^* is the load factor of B^*. When $f = f'$, we see that (28) and (21) are equivalent. In this case, $\hat{n^*} = \hat{u}$, and B^* can be regarded as an encoding of the union of the two sets.

Based on (27) and (29), using the fact that $\ln(1-x) \approx -x$ when x is sufficiently small, the expected value of $\hat{d}$ is

$$E(\hat{d}) = E\Big(\hat{n^*} - \hat{n'} \cdot \frac{\ln(1-\frac{p}{f'})}{\ln(1-\frac{p}{f})}\Big) \approx E(\hat{n^*} - \frac{f}{f'}\hat{n'})$$
$$= d + \frac{1}{2p}(e^{\omega^*} - \omega^* p - 1) - \frac{f}{f'}\frac{1}{2p}(e^{\omega'} - \omega' p - 1).$$

Recall from the previous section that all proposed estimators are asymptotically unbiased.

The variance of $\hat{d}$ is derived as follows.

$$Var(\hat{d}) \approx Var(\hat{n^*} - \tfrac{f}{f'}\hat{n'})$$
$$= Var(\hat{n^*}) + \tfrac{f^2}{f'^2}Var(\hat{n'}) - 2\tfrac{f}{f'}Cov(\hat{n^*}, \hat{n'})$$

We have derived that $Cov(\hat{n^*}, \hat{n'}) \approx \frac{f}{f'} Var(n')$; the process is omitted due to space limitation. Hence,

$$Var(\hat{d}) \approx Var(\hat{n^*}) - \frac{f^2}{f'^2} Var(\hat{n'}). \tag{30}$$

The above equation implies that $Var(\hat{d})$ is smaller than $Var(\hat{n^*})$. Intuitively, because $\hat{d}$ is calculated from $\hat{n^*}$ and $\hat{n'}$, the estimation $\hat{d}$ should contain the estimation error of both $\hat{n^*}$ and $\hat{n'}$. However, our analysis shows that the two estimates $\hat{n^*}$ and $\hat{n'}$ positively correlate with each other, i.e., $Cov(\hat{n^*}, \hat{n'}) > 0$, which causes the variance of $\hat{d}$ smaller.

5.2 Mean and Variance of $\hat{u}$

From (21), (22), and (28), by the fact that $\ln(1-x) \approx -x$ when x is sufficiently small, we have $\hat{u} \approx \hat{n^*} - \frac{f-f'}{f'}\hat{n'}$. Hence,

$$\begin{aligned} E(\hat{u}) &\approx E(\hat{n^*}) - \frac{f-f'}{f'}E(\hat{n'}) \\ &= u + \frac{1}{2p}(e^{\omega^*} - \omega^* p - 1) - \frac{f-f'}{f'}\frac{1}{2p}(e^{\omega'} - \omega' p - 1) \end{aligned}$$

$$Var(\hat{u}) \approx Var(\hat{n^*}) + \frac{(f-f')^2}{f'^2}Var(\hat{n'}) - 2\frac{f-f'}{f'}Cov(\hat{n^*}, \hat{n'}).$$

Since $Cov(\hat{n^*}, \hat{n'}) \approx \frac{f}{f'}Var(n')$, we have

$$Var(\hat{u}) \approx Var(\hat{n^*}) - \frac{f^2 - f'^2}{f'^2}Var(\hat{n'}). \tag{31}$$

5.3 Mean and Variance of $\hat{m}$

From (20), (22), (28), we have $\hat{m} \approx \hat{n} + \frac{f}{f'}\hat{n'} - \hat{n^*}$. Hence,

$$\begin{aligned} E(\hat{m}) \approx& E(\hat{n}) + \frac{f}{f'}E(\hat{n'}) - E(\hat{n^*}) \approx m + \frac{1}{2p}(e^{\omega} - \omega p - 1) \\ &+ \frac{f}{f'}\frac{1}{2p}(e^{\omega'} - \omega' p - 1) - \frac{1}{2p}(e^{\omega^*} - \omega^* p - 1) \\ Var(\hat{m}) \approx& Var(\hat{n}) + \frac{f^2}{f'^2}Var(\hat{n'}) + Var(\hat{n^*}) \\ &+ 2\frac{f}{f'}Cov(\hat{n}, \hat{n'}) - 2Cov(\hat{n^*}, \hat{n}) - 2\frac{f}{f'}Cov(\hat{n^*}, \hat{n'}). \end{aligned}$$

We have derived that $Cov(\hat{n}, \hat{n'}) = \frac{f}{p^2}(e^{\frac{mp}{f}} - \frac{m}{f}p^2 - 1)$ and that we have derived that $Cov(\hat{n^*}, \hat{n}) \approx Var(\hat{n}) + (\frac{f}{f'} - 1)\frac{f}{p^2}(e^{\frac{mp}{f}} - \frac{m}{f}p^2 - 1)$. Also using $Cov(\hat{n^*}, \hat{n'}) \approx \frac{f}{f'}Var(n')$, we have

$$\begin{aligned} Var(\hat{m}) &\approx Var(\hat{n}) + \frac{f^2}{f'^2}Var(\hat{n'}) + Var(\hat{n^*}) \\ &+ 2\frac{f}{p^2}(e^{\frac{mp}{f}} - \frac{m}{f}p^2 - 1) - 2Var(\hat{n}) - 2\frac{f^2}{f'^2}Var(\hat{n'}) \\ =& Var(\hat{n^*}) - Var(\hat{n}) - \frac{f^2}{f'^2}Var(\hat{n'}) + 2\frac{f}{p^2}(e^{\frac{mp}{f}} - \frac{m}{f}p^2 - 1). \end{aligned} \tag{32}$$

5.4 Mean and Variance of $\hat{d'}$

From (20), (22), (28), we have $\hat{d'} \approx \hat{n^*} - \hat{n} - \frac{f-f'}{f'}\hat{n'}$. Hence, we can calculate

$$\begin{aligned} E(\hat{d'}) &\approx E(\hat{n^*}) - E(\hat{n}) - \frac{f-f'}{f'}E(\hat{n'}) \\ \approx& d' + \frac{1}{2p}(e^{\omega^*} - \omega^* p - 1) - \frac{1}{2p}(e^{\omega} - \omega p - 1) \\ &- \frac{f-f'}{f'}\frac{1}{2p}(e^{\omega'} - \omega' p - 1) \\ Var(\hat{d'}) &\approx Var(\hat{n^*}) + Var(\hat{n}) + (\frac{f-f'}{f'})^2 Var(\hat{n'}) \\ &+ 2\frac{f-f'}{f'}Cov(\hat{n}, \hat{n'}) - 2Cov(\hat{n^*}, \hat{n}) - 2\frac{f-f'}{f'}Cov(\hat{n^*}, \hat{n'}). \end{aligned}$$

The three covariances are known when we previously derive the variance of $\hat{m}$. Therefore,

$$\begin{aligned} Var(\hat{d'}) =& Var(\hat{n^*}) + Var(\hat{n}) + (\frac{f-f'}{f'})^2 Var(\hat{n'}) \\ &- 2Var(\hat{n}) - 2\frac{f-f'}{f'}\frac{f}{f'}Var(\hat{n'}) \\ =& Var(\hat{n^*}) - Var(\hat{n}) - \frac{f^2 - f'^2}{f'^2}Var(\hat{n'}). \end{aligned} \tag{33}$$

Because the estimators approximate Gaussian distributions, the accuracy requirements of (1)-(4) can be turned into a set of constraints on bounded $Var(\hat{d})$, $Var(\hat{d'})$, $Var(\hat{m})$, and $Var(\hat{u})$. The following property shows that these constraints can be turned into a single one on $Var(\hat{n^*})$, which will be used to determine the optimal system parameters. The proof of the property is omitted due to space limitation.

PROPERTY 2. *$Var(\hat{n^*})$ is an approximately tight upper bound of $Var(\hat{d})$, $Var(\hat{d'})$, $Var(\hat{m})$, and $Var(\hat{u})$.*

6. SYSTEM PARAMETERS

We have already known that the distributions of joint estimations, $\hat{d}$, $\hat{d'}$, $\hat{m}$, and $\hat{u}$, approximate Gaussian distributions. Our requirement is to bound the estimation error of each joint property by a constant k with high probability, as stated in (1)-(4). In this section, we try to set the optimal system parameters f and p to minimize the protocol execution time, subject to the accuracy requirement.

6.1 Load Factor

Consider the requirement (1) on the union cardinality estimation $\hat{u}$, which specifies a confidence interval of width $2k$ at a confidence level of α. For a Gaussian distribution with $E(\hat{u}) \approx u$, the requirement on $\hat{u}$ is translated to:

$$Z_\alpha\sqrt{Var(\hat{u})} \le k \qquad Var(\hat{u}) \le \frac{k^2}{{Z_\alpha}^2}, \tag{34}$$

where Z_α denotes the $\frac{1+\alpha}{2}$ percentile for the unit Gaussian distribution. Similar, the requirement (3)-(4) can be translated to

$$Var(\hat{m}) \le \frac{k^2}{{Z_\alpha}^2}, \quad Var(\hat{d}) \le \frac{k^2}{{Z_\alpha}^2}, \quad Var(\hat{d'}) \le \frac{k^2}{{Z_\alpha}^2}. \tag{35}$$

In order to cover all possible cases, due to Property 2, all these constraints (1)-(4) can be replaced by $Var(\hat{n^*}) \le \frac{k^2}{{Z_\alpha}^2}$.

$$\begin{aligned} \frac{f}{p^2}\left(e^{\omega^*} - \omega^* p - 1\right) \le& \frac{k^2}{{Z_\alpha}^2} \\ \frac{n}{p\omega}\Big((1-p)\omega^* + \frac{1}{2}\omega^{*2} + \frac{1}{6}\omega^{*3}\Big) \le& \frac{k^2}{{Z_\alpha}^2} \quad \text{Taylor Expansion} \end{aligned} \tag{36}$$

From (29), we have $\omega^* = \frac{pd}{f} + \frac{pn'}{f'} = \frac{pn}{f} - \frac{pm}{f} + \frac{pn'}{f'} = \omega + \omega' - \frac{pm}{f}$. If $m = 0$, then $\omega^* = \omega + \omega'$, which maximizes the left side of (36). We consider this worst-case constraint in terms of m. Hence,

$$\frac{n}{p\omega}\Big((1-p)(\omega+\omega') + \frac{1}{2}(\omega+\omega')^2 + \frac{1}{6}(\omega+\omega')^3\Big) \le \frac{k^2}{{Z_\alpha}^2}$$
$$\frac{1}{\omega}\Big((1-p)(\omega+\omega') + \frac{1}{2}(\omega+\omega')^2 + \frac{1}{6}(\omega+\omega')^3\Big) \le \frac{k^2 p}{{Z_\alpha}^2 n}.$$

To satisfy the above constraint in the worst case in terms of n (which is bounded by $n_{\max}$), we have

$$\frac{1}{\omega}\Big((1-p)(\omega+\omega') + \frac{1}{2}(\omega+\omega')^2 + \frac{1}{6}(\omega+\omega')^3\Big) \le \frac{k^2 p}{{Z_\alpha}^2 n_{\max}}. \tag{37}$$

In our system design, we shall set both w and w' for a system-wide optimal load factor. With $\omega' = \omega$, we have

$$\frac{1}{\omega}\Big((1-p)2\omega + \frac{1}{2}(2\omega)^2 + \frac{1}{6}(2\omega)^3\Big) \le \frac{k^2 p}{{Z_\alpha}^2 n_{\max}}$$

$$\omega \le -\frac{3}{4} + \frac{\sqrt{3}}{4}\sqrt{4p\Big(\frac{k^2}{n_{\max}{Z_\alpha}^2} + 2\Big) - 5}. \tag{38}$$

Because $w = \frac{np}{f}$, it is inversely proportional to the frame size f, which measures the protocol execution time when encoding the tags in B. Hence, we should set our target load factor as

$$\omega = -\frac{3}{4} + \frac{\sqrt{3}}{4}\sqrt{4p\Big(\frac{k^2}{n_{\max}{Z_\alpha}^2} + 2\Big) - 5}. \tag{39}$$

We justify our choice of setting $\omega = \omega'$ above. The left side of (37) is an increasing function in both ω and ω'. If we allow $\omega' \neq \omega$ and still set their values to be as small as possible, then one of them will be greater than the right side of (39) and the other will be smaller. Because N and N' are arbitrary tag sets under consideration, it means that some tag sets will be encoded with their load factors greater than the right side of (39) and some others will have smaller load factors. Let N_1 and N_2 be two sets with load factors greater than (39). We should be able to perform joint estimation on any two encoded sets without violating the accuracy requirement. However, if we perform joint estimation on N_1 and N_2, because their load factors are larger than (39), the constraint of (37) will not hold.

6.2 Frame Size and Sampling Probability

From (38) and $w = \frac{np}{f}$, we have

$$f \ge \frac{np}{-\frac{3}{4} + \frac{\sqrt{3}}{4}\sqrt{4p\big(\frac{k^2}{n_{\max}{Z_\alpha}^2} + 2\big) - 5}}. \tag{40}$$

Recall that the value of f is set to be a power of two in order to support expanded OR between the snapshots of any two tag sets. We want to choose the optimal sampling probability that minimizes the protocol execution time by keeping the frame size f as small as possible. Hence, we have the formula for the frame size as quoted in Section 4.1:

$$f \;=\; \min_{p\in(0,1]} \{2^{\lceil \log_2(\frac{np}{\omega})\rceil}\}, \tag{41}$$

where p is a sampling probability and the load factor ω is determined by Eq. (39). It requires a prior knowledge of n, the number of tags in N, which can be estimated through an existing protocol such as GMLE [8], LOF [12] and PET [22]. The optimal sampling probability p^* that minimizes f depends on the pre-determined parameters $n_{\max}$, k and α. Hence, it can be pre-computed.

7. SIMULATION RESULTS

7.1 Simulation Setting

We evaluate the performance of the proposed JREP protocol and compare it to DiffEstm [20] for joint estimation. For the union estimation, we also want to compare with SRC_M, the best protocol among those that were originally designed to estimate the cardinality of a tag set covered by multiple readers. Assuming that the optimal sampling probability is known, SRC_M becomes equivalent to DiffEstm in union estimation. See Section 3.2.

We consider two performance metrics. First, when the two protocols are subject to the same average execution time, we compare *their probabilities of meeting a given error bound*. The probability is measured as the number of joint estimations that meet the error bound divided by the total number of joint estimations performed in the simulation. In favor to DiffEstm and SRC_M, we assume that they know the optimal sampling probability that maximizes their worst-case probabilities of meeting the error bound. The original paper [20] does not give a formula for this optimal sampling probability. We obtain it through exhaustive search by simulations. Second, given an accuracy requirement as defined in (1)-(4), we compare the *execution times* of the two protocols. The execution time is measured as the number of time slots it takes the reader to encode a tag set in a snapshot bitmap, including the frame size f and other slots needed to give an initial rough estimation of n (Section 6). For JREP, we invoke GMLE [8] to generate a raw estimation of n with a 95% confidence interval of ±10% error.

The system model is a distributed RFID system of multiple locations, where each reader periodically takes a snapshot of its local set of tags, whose number ranges from 0 to 50,000, with $n_{max} = 50,000$. The average number of tags in a set is chosen to be 10,000, which reflects that the normal business flow of tagged objects is smaller than the worst-case number that the system is designed to handle. The size of each tag set will be taken from a truncated normal distribution $Norm(10000, 2000^2)$ in the range of $[0, n_{max}]$. For the accuracy requirement, we set $\alpha = 95\%$ and $k = 500$ by default. We will also perform simulation with other values of k and α.

7.2 Estimation Accuracy under Same Execution Time

We first set the parameters for JREP so that it satisfies the accuracy requirement of $\alpha = 95\%$ and $k = 500$, i.e., the difference between the estimated values ($\hat{d}$, $\hat{d'}$, $\hat{u}$, $\hat{m}$) and the true values (d, d', u, m) is bounded by 500 with 95% probability. We can compute the value of ω from (39) and then the values of f and p from (41). However, since the value of f is rounded up to the power of two to support the operation of expanded OR, these two parameters are in fact set conservatively. Alternatively, we can set their values empirically through simulations (similar to [2]). We first compute the initial value of ω from (39) and then perform bi-section search to reduce it as small as possible such that the resulting values of f and p from (41) will still satisfy the accuracy requirement. As a result, the load factor ω is 0.735.

Table 1: Probability for intersection cardinality estimation (m) by JREP ($\omega = 0.735$) to meet the bound $k = 500$

$n'(\times10^3)$ $n(\times10^3)$	[0, 5)	[5, 10)	[10, 15)	[15, 20)	[20, 25)	[25, 30)	[30, 35)	[35,40)	[40, 45)	[45, 50)
[0, 5)	1.00	—	—	—	—	—	—	—	—	—
[5, 10)	1.00	1.00	—	—	—	—	—	—	—	—
[10, 15)	1.00	1.00	1.00	—	—	—	—	—	—	—
[15, 20)	1.00	1.00	1.00	1.00	—	—	—	—	—	—
[20, 25)	1.00	1.00	1.00	1.00	1.00	—	—	—	—	—
[25, 30)	1.00	1.00	1.00	1.00	1.00	1.00	—	—	—	—
[30, 35)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	—	—	—
[35, 40)	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	—	—
[40, 45)	0.99	0.99	0.99	1.00	0.99	1.00	1.00	1.00	1.00	—
[45, 50)	**0.98**	0.99	0.99	1.00	0.99	1.00	1.00	1.00	0.99	0.99

Table 2: Probability for union cardinality estimation (u) by JREP ($\omega = 0.735$) to meet the error bound $k = 500$

$n'(\times10^3)$ $n(\times10^3)$	[0, 5)	[5, 10)	[10, 15)	[15, 20)	[20, 25)	[25, 30)	[30, 35)	[35,40)	[40, 45)	[45, 50)
[0, 5)	1.00	—	—	—	—	—	—	—	—	—
[5, 10)	1.00	1.00	—	—	—	—	—	—	—	—
[10, 15)	1.00	1.00	1.00	—	—	—	—	—	—	—
[15, 20)	1.00	1.00	1.00	1.00	—	—	—	—	—	—
[20, 25)	0.99	1.00	1.00	1.00	0.99	—	—	—	—	—
[25, 30)	1.00	1.00	1.00	1.00	0.99	1.00	—	—	—	—
[30, 35)	0.99	0.99	0.99	0.99	0.99	1.00	1.00	—	—	—
[35, 40)	0.98	0.98	0.99	0.99	0.97	1.00	0.99	0.99	—	—
[40, 45)	0.97	0.97	0.97	0.97	0.96	0.99	0.99	0.98	0.97	—
[45, 50)	**0.95**	0.96	0.97	0.98	0.96	0.99	0.97	0.98	0.97	0.96

Table 3: Probability for intersection cardinality estimation (m) by DiffEstm ($f = 18405$) to meet $k = 500$

$n'(\times10^3)$ $n(\times10^3)$	[0, 5)	[5, 10)	[10, 15)	[15, 20)	[20, 25)	[25, 30)	[30, 35)	[35,40)	[40, 45)	[45, 50)
[0, 5)	1.00	—	—	—	—	—	—	—	—	—
[5, 10)	1.00	1.00	—	—	—	—	—	—	—	—
[10, 15)	1.00	1.00	1.00	—	—	—	—	—	—	—
[15, 20)	1.00	1.00	1.00	1.00	—	—	—	—	—	—
[20, 25)	1.00	1.00	1.00	1.00	0.98	—	—	—	—	—
[25, 30)	1.00	1.00	0.99	0.98	0.97	0.94	—	—	—	—
[30, 35)	1.00	1.00	0.98	0.97	0.94	0.89	0.85	—	—	—
[35, 40)	1.00	1.00	0.96	0.96	0.91	0.89	0.82	0.76	—	—
[40, 45)	1.00	0.99	0.94	0.89	0.86	0.79	0.72	0.73	0.67	—
[45, 50)	1.00	0.96	0.92	0.88	0.82	0.77	0.74	0.68	0.61	**0.51**

Table 4: Probability for union cardinality estimation (u) by DiffEstm/SRC$_{\mathrm{M}}$ ($f = 18405$) to meet $k = 500$

$n'(\times10^3)$ $n(\times10^3)$	[0, 5)	[5, 10)	[10, 15)	[15, 20)	[20, 25)	[25, 30)	[30, 35)	[35,40)	[40, 45)	[45, 50)
[0, 5)	1.00	—	—	—	—	—	—	—	—	—
[5, 10)	1.00	1.00	—	—	—	—	—	—	—	—
[10, 15)	1.00	1.00	1.00	—	—	—	—	—	—	—
[15, 20)	1.00	1.00	1.00	0.99	—	—	—	—	—	—
[20, 25)	1.00	0.99	0.99	0.98	0.94	—	—	—	—	—
[25, 30)	0.99	0.97	0.95	0.93	0.91	0.87	—	—	—	—
[30, 35)	0.96	0.94	0.90	0.88	0.85	0.81	0.75	—	—	—
[35, 40)	0.91	0.89	0.84	0.84	0.78	0.76	0.71	0.67	—	—
[40, 45)	0.86	0.82	0.77	0.73	0.70	0.66	0.62	0.61	0.56	—
[45, 50)	0.78	0.72	0.69	0.67	0.64	0.60	0.59	0.56	0.51	**0.42**

Table 1 shows the probability for the intersection estimation $\hat{m}$ by JREP to meet the error bound 500. We simulate two tag sets of sizes n and n', with $n \geq n'$. The first column shows the range from which n is chosen uniformly at random. For example, the first range is $[0, 5000)$ and the last range is $[45000, 50000)$. Be ware that the numbers of in the first column have a unit of 1000. The first row shows the range from which n' is chosen. Similarly its first range is $[0, 5000)$ and the last range is $[45000, 50000)$. Because we require $n \geq n'$, the combinations of n and n' above the diagonal are invalid. Each cell in the table shows the probability of meeting the error bound when n and n' are chosen from specified ranges. For example, consider the left bottom cell inside the table, where n is chosen from $[45000, 50000)$ and n' from $[0, 5000)$. The probability of meeting the error bound is 98%, which is measured from 1,000 simulation runs, each with two tag sets N and N' arbitrarily generated and the number m of common tags randomly chosen from the range $[0, n']$.

Table 2 shows the probability for the union estimation $\hat{u}$ by JREP to meet the error bound 500. All probabilities in both tables are greater than 95%, which confirms our

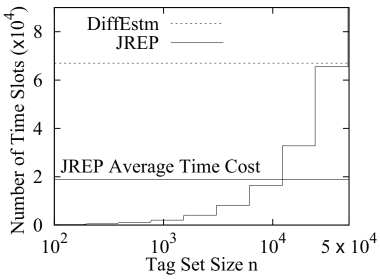

Figure 5: Time comparison with $k = 500$ and $\alpha = 95\%$

analytical results that the proposed estimators satisfy the accuracy requirements (1) and (3). The same is true for difference estimations $\hat{d}$ and $\hat{d}'$, which are not shown due to space limitation.

For JREP, the frame size depends on the size of tag set to be encoded. With an average size of 10,000 tags and $\omega = 0.735$, the average execution time of JREP is 18,405 slots in our simulation. We use the same number of slots for DiffEstm (or SRC_{M}), and the simulation results are shown in Tables 3 and 4, where the probability for m to meet the error bound 500 can be as low as 51% when both n and n' are large, and the probability for u to meet the error bound can be as low as 42%, which is much worse than what JREP can achieve at the same time cost.

7.3 Execution Time to Achieve Same Accuracy

Next, we fix the accuracy requirements with $\alpha = 95\%$ and $k = 500$, and compare the execution times of JREP and DiffEstm for taking a bitmap snapshot of a tag set. Because DiffEstm is not designed for absolute error bound, there is no formula to compute its frame size. With $n_{max} = 50,000$, we use exhaustive search by simulation to find its minimum frame size that can meet the error bound. The results are shown in Fig. 5, where the horizontal axis is the size of a tag set, which varies from 100 to 50000, and the vertical axis is the number of time slots needed to take a bitmap snapshot of the set. Due to the nature of its design, DiffEstm uses a constant frame size of 67,000 slots. The frame size of JREP is variable. It is small when the tag set is small. For example, for a set of 5000 tags, the number of time slots needed by JREP is 8,192, only 12% of what's needed by DiffEstm. The average time cost of JREP is shown by the solid horizontal line. Similar results are observed from simulations with different accuracy requirements (whose results cannot be included due to limited space).

8. CONCLUSION

This paper studies the problem of joint cardinality estimation: Given any two tag sets in a large RFID system, estimating their union cardinality, intersection cardinality, and difference cardinalities. We propose a solution called JREP that adapts its snapshot based on the size of the tag set that it records. Variable-sized snapshots are combined through expanded OR to support joint estimation. We derive a full set of estimators, analyze their accuracies, and provide formulas for setting the optimal system parameters. We demonstrate that the new solution is much more efficient than the prior art.

9. ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 61300024, Jiangsu Provincial Natural Science Foundation of China under Grant BK20130634, China Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20130092120036, and National Science Foundation of United States under grants CNS-1409797 and CNS-1115548.

10. REFERENCES

[1] G. Casella and R. L. Berger. *Statistical Inference*. 2nd Edition, Duxbury Press, 2002.

[2] B. Chen, Z. Zhou, and H. Yu. Understanding RFID counting protocols. *Proc. of ACM MOBICOM*, 2013.

[3] M. Chen, W. Luo, Z. Mo, S. Chen, and Y. Fang. An efficient tag search protocol in large-scale RFID systems. *Proc. of IEEE INFOCOM*, April 2013.

[4] H. Han, B. Sheng, C. Tan, Q. Li, W. Mao, and S. Lu. Counting RFID tags efficiently and anonymously. *Proc. of IEEE INFOCOM*, 2010.

[5] M. Kodialam and T. Nandagopal. Fast and reliable estimation schemes in RFID systems. *Proc. of ACM MOBICOM*, 2006.

[6] S.-R. Lee, S.-D. Joo, and C.-W. Lee. An enhanced dynamic framed slotted aloha algorithm for RFID tag identification. *Proc. of IEEE MOBIQUITOUS*, pages 166–174, 2005.

[7] T. Li, S. Chen, and Y. Ling. Efficient protocols for identifying the missing tags in a large RFID system. *IEEE/ACM Transactions on Networking*, 21(6), 2013.

[8] T. Li, S. Wu, S. Chen, and M. Yang. Energy efficient algorithms for the RFID estimation problem. *Proc. of INFOCOM*, 2010.

[9] W. Luo, Y. Qiao, and S. Chen. An efficient protocol for RFID multigroup threshold-based classification. *Proc. of IEEE INFOCOM*, April 2014.

[10] W. Luo, Y. Qiao, S. Chen, and T. Li. Missing-tag detection and energy-time tradeoff in large-scale RFID systems with unreliable channels. *IEEE/ACM Transactions on Networking*, 22(4):1079 – 1091, August 2014.

[11] J. Myung and W. Lee. Adaptive splitting protocols for RFID tag collision arbitration. *Proc. of ACM MOBIHOC*, 2006.

[12] C. Qian, H. Ngan, and Y. Liu. Cardinality estimation for large-scale RFID systems. *Proc. of IEEE PERCOM*, 2008.

[13] Y. Qiao, S. Chen, T. Li, and S. Chen. Energy-efficient polling protocols in RFID systems. *Proc. of ACM Mobihoc*, May 2011.

[14] Y. Rekik. Inventory inaccuracies in supply chains: How can RFID improve the performance? *Wiley Encyclopedia of Operations Research and Management Science*, 2010.

[15] V. Shah-Mansouri and V. Wong. Cardinality estimation in RFID systems with multiple readers. *IEEE Transactions on Wireless Communications*, 10(5):1458–1469, May 2011.

[16] M. Shahzad and A. X. Liu. Every bit counts: Fast and scalable RFID estimation. *Proc. of ACM MOBICOM*, 2012.

[17] B. Sheng, Q. Li, and W. Mao. Efficient continuous scanning in RFID systems. *Proc. of IEEE INFOCOM*, 2010.

[18] H. Vogt. Efficient object identification with passive RFID tags. *Proc. of IEEE PERCOM*, 2002.

[19] 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*, 15(2):208–229, June 1990.

[20] Q. Xiao, B. Xiao, and S. Chen. Differential estimation in dynamic RFID systems. *Proc. of IEEE INFOCOM*, 2013.

[21] Y. Zheng and M. Li. Zoe: Fast cardinality estimation for large-scale RFID systems. *Proc. of IEEE INFOCOM*, pages 908–916, 2013.

[22] Y. Zheng, M. Li, and C. Qian. Pet: Probabilistic estimating tree for large-scale RFID estimation. *Proc. of IEEE ICDCS*, June 2011.