Joint Property Estimation for Multiple RFID Tag Sets Using Snapshots of Variable Lengths

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

Shigang Chen Min Chen
Department of Computer & Information Science & Engineering
University of Florida, Gainesville, FL 32611, USA
{sgchen, min}@cise.ufl.edu

ABSTRACT

Radio-frequency identification (RFID) technology has been widely adopted by real-world industries. This paper presents a new application for distributively deployed RFID systems, wherein a user chooses multiple tag sets at will from different spatial or temporal domains, and then connects them by set operators (union, intersection and relative complement) to form a set expression. The user is allowed to query for the cardinality of an arbitrary set expression, which is called the *joint property* of multiple sets. We focus on the problem of estimating the joint property with bounded error, which has many potential applications. One of them is to allow users to check the number of tags in an arbitrary tag flow passing through a distributed RFID system. For this problem, we propose a solution with a novel design that supports versatile snapshot construction: Given the snapshots of multiple tag sets, although their lengths may be very different, our formulas can estimate their joint properties, with an accuracy that can be arbitrarily set. For the proposed estimator, we formally analyze its bias and variance, and also the optimal settings of protocol parameters to minimize the time cost of taking a snapshot of a tag set. The simulation results show that, under predefined accuracy requirement, our solution can reduce time cost by multiple folds as compared with existing works named DiffEstm and CCF, which require all tag sets must be encoded into snapshots with an equal length.

Categories and Subject Descriptors

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

Keywords

RFID; Cardinality Estimation; Random Hashing

1. INTRODUCTION

Over the past decade, radio-frequency identification (RFID) technology has been widely used by industries such as warehouse management, logistical control, and asset tracking in malls, hospitals or highways [5]. RFID systems can be conceptually divided into two parts: RFID tags (each carrying an unique ID) which are attached to physical objects, and RFID readers, which are deployed at places of interest to sense the existence of tags, quickly retrieve the tag IDs, or gather the statistical information about a group ot tags.

An important fundamental functionality of RFID system is called *cardinality estimation*, which is to count the number of tags in a physical region [3–5, 7, 14, 15, 18–20]. This function can be used to monitor the inventory level of a warehouse, the sales in a retail store or the popularity of a theme park. Counting the number of tags takes much less time than a full system scan that collects all tag IDs. This is an important feature since RFID systems communicate via low-rate wireless channels and the execution time cost is the key performance metric in system design. In addition to its direct utility, tag estimation can work as a pre-processing step that improves the efficiency of tag identification process [6, 13]. More importantly, since it does not collect any tag IDs, the anonymity of tags can be preserved, particularly in scenarios where the party performing the operation (such as warehouse or port authority) does not own the tagged items.

Motivation. The previous work mainly focuses on estimating the cardinality of a tag set within the radio range of a single reader [3–5, 7, 14, 15, 19, 20], or estimating the union of tag sets near multiple readers [3, 14]. This paper studies the cardinality estimation problem in a much more generalized scenario: Multiple tag sets can be captured by a distributed multi-reader system at different spatial or temporal domains. As requested by system users, these tag sets may constitute an arbitrary set expression using the operations of union ($\cup$), intersection ($\cap$) and relative complement ($\setminus$). We will estimate the cardinality of this user-desired set expression, which is called a *joint property* of multiple sets.

We use two applications to better illustrate the usefulness of this joint property estimation problem. Just imagine we are managing a large logistics network, where tagged products are shipped from one location (factory, warehouse, port, or storage/retail facility) to another. Assume the reader deployed at each location takes periodic snapshots of its local set of tags and keeps a series of such snapshots over time. When the end users want to know the quantity of goods flowing from one location to the other, we are able to address the query by estimating the cardinality of *intersection between two snapshots* from different locations. Furthermore, a more complicated user query could be the quantity of goods traversing a routing path comprised of multiple locations. We can address the query by computing the cardinality of *the intersection among two or more tag set snapshots*.

In the second application, imagine that we are monitoring a warehouse for its inventory dynamics over time. We want

MobiHoc'16, July 04-08, 2016, Paderborn, Germany

DOI: http://dx.doi.org/10.1145/2942358.2942377

to know the amount of goods entering (i.e., the number of new tags) the warehouse, and the amount leaving (i.e., the number of departed tags) between *any two* reference time points. Suppose the warehouse has been deployed with an RFID reader system to take periodical snapshots about the existing tags. Then, a solution for this problem could be examining the difference between two snapshots taken at different time points, which tells the information about product inflow and outflow within the time interval. However, a key challenge is that a large warehouse inevitably needs more than one readers to achieve full coverage, and each reader can take a snapshot only about its local set of tags. Note that the snapshots taken by different readers may not be of an identical length. When a user queries for the dynamics of such a warehouse, he or she actually wants to know how the union of multiple tag sets (scanned by readers at different locations) fluctuates over time. Such a query will produce a complex set expression — to compute the cardinality of *the set difference of two union tag sets at different time points.*

Problem. From the above two applications, we can abstract the problem of joint property estimation. It is to estimate the cardinality of an arbitrary set expression that involves multiple tag sets (whose number is denoted by k) existing in different temporal or spatial domains. The protocol designed to scan each tag set must be time-efficient, and its absolute estimation error must be kept within a predefined bound at a probability above a given threshold.

There are very limited prior studies on this problem. The differential estimator (DiffEstm) [18] and the joint RFID estimation protocol (JREP) [17] can estimate the set expression involving only two tag sets. Although the composite counting framework (CCF) [8] provides a generalized estimator for an arbitrary expression over multiple sets, it is designed based on a different, relative error model, resulting in large execution time, with unbounded worst-case time complexity. Moreover, as multiple RFID readers are deployed at different places to scan their surroundings periodically, the tag sets they encounter may differ significantly in sizes. The biggest tag set can be many times larger than the average-sized tag sets. Both the prior solutions DiffEstm and CCF encode each tag set into a data structure whose length is determined by the size of the largest possible set (or even the union of several largest tag sets for union estimation), which causes unnecessarily long protocol execution time.

Our Contributions. First, for the generalized joint property estimation problem, we propose a solution with a novel design that supports versatile snapshot construction. It adopts a two-phase protocol design between a reader and its nearby tags to construct a snapshot of the tag set. The length of the snapshot could be (but not necessary) proportional to the size of the tag set, instead of being fixed to a large worst-case value. Given the snapshots of any k tag sets, although their lengths may be very different, we have derived closed-form formulas to estimate the joint properties of the k sets.

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

Third, we perform extensive simulations, and the results show that, by allowing the snapshots to have variable lengths, the new solution significantly outperforms DiffEstm [18] and CCF [8], both of which assume a fixed length for snapshots. Under the same accuracy requirement, our new solution can reduce time cost by over 80% as compared with the previous.

2. JOINT PROPERTY ESTIMATION

In this section, we formally define the research problem of joint property estimation for multiple RFID tag sets.

Definition of Joint Properties. Suppose a distributed RFID system, where tagged objects are moved from one location to another. We use $S_1, S_2, \ldots, S_k$ to denote the tag sets captured by RFID systems at different locations or time points. They can form an arbitrary set expression as connected by the union ($\cup$), intersection ($\cap$) and relative complement ($\setminus$) operations. The cardinality of such an expression is called a joint property of the k tag sets.

A major difficulty is that the number of possible set expressions is really huge. In order to tame the high complexity, we start from a small group of special expressions. We divide the union of all k sets $S_1 \cup S_2 \ldots \cup S_k$ into subsets that are mutually disjoint. They are called *elementary subsets*, and the number of elementary subsets is merely $2^k - 1$.

As an example, in Fig. 1, we illustrate the Venn diagram of three RFID tag sets S_1, S_2, S_3, and we divide their union into $2^3 - 1 = 7$ elementary subsets. Each subset is denoted by $N_{b_3b_2b_1}$, where $b_3b_2b_1$ is a binary ranging from 001 to 111 that indicates whether the subset is included by S_3, S_2 or S_1. For instance, the subset N_{110} is included by S_3 and S_2, but excluded by S_1. In Fig. 1, N_{000} is a special case that corresponds to the tags not included by any sets S_1, S_2 or S_3.

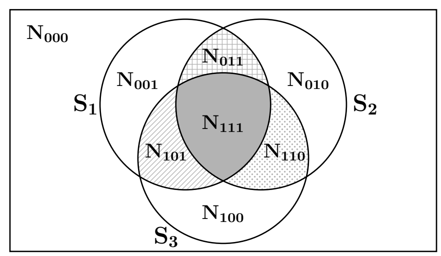

Figure 1: Venn diagram of three tag sets S_1, S_2, S_3, and illustration of elementary subsets $N_{b_3b_2b_1}$.

We formalize the concept of elementary subset $N_{b_k \ldots b_2b_1}$ as

$$N_{b_k \ldots b_2b_1} = \big(\bigcap\nolimits_{b_i \neq 0}^{1 \leq i \leq k} S_i\big) \setminus \big(\bigcup\nolimits_{b_i = 0}^{1 \leq i \leq k} S_i\big), \tag{1}$$

where the bit b_i indicates whether the elementary subset is included or excluded by the ith set S_i. It is equivalent to

$$N_{b_k \ldots b_2b_1} = \big(\bigcap\nolimits_{b_i \neq 0}^{1 \leq i \leq k} S_i\big) \cap \big(\bigcap\nolimits_{b_i = 0}^{1 \leq i \leq k} {S_i}^c\big), \tag{2}$$

if applying the rule of relative complement $A \setminus B = A \cap B^c$ to equation (1), where B^c is the absolute complement of B.

For a shorter notation, we replace $N_{b_k \ldots b_2b_1}$ by N_x, where x is a decimal that is equal to the binary value $b_k \ldots b_2b_1$. Hence, the definition of elementary subset in (2) becomes

$$N_x = \big(\bigcap\nolimits_{2^{i-1} \wedge x \neq 0}^{1 \leq i \leq k} S_i\big) \cap \big(\bigcap\nolimits_{2^{i-1} \wedge x = 0}^{1 \leq i \leq k} {S_i}^c\big), \tag{3}$$

where $2^{i-1} \wedge x$ extracts the ith bit from x by bitwise AND $\wedge$. There are two boundary cases: $N_{2^k-1} = S_1 \cap S_2 \ldots \cap S_k$ is the intersection of all sets, and $N_0 = {S_1}^c \cap {S_2}^c \ldots \cap {S_k}^c = (S_1 \cup S_2 \ldots \cup S_k)^c$ is the complement of the union of all sets.

For an arbitrary elementary subset N_x with $1 \leq x < 2^k$, we denote its cardinality by n_x, and call it a *joint property* of the k tag sets $S_1, S_2, \ldots, S_k$. If the cardinalities n_x ($1 \leq x < 2^k$) of all elementary subsets are known, we can derive the cardinality of an arbitrary set expression by summing up the cardinalities of elementary subsets it includes. This is because any set expression can be rewritten as the union of several elementary subsets. As an example, let the queried tag set be $S_3 \cap (S_2 \cup S_1)$. It is equal to $S_3 \cap \big((S_2{}^c \cap S_1) \cup (S_2 \cap S_1{}^c) \cup (S_2 \cap S_1)\big)$. By applying the distributive law, it becomes $(S_3 \cap S_2{}^c \cap S_1) \cup (S_3 \cap S_2 \cap S_1{}^c) \cup (S_3 \cap S_2 \cap S_1)$. By the definition of N_x in (3), it equals $N_5 \cup N_6 \cup N_7$. Hence, the queried cardinality $|S_1 \cap (S_2 \cup S_3)|$ is equal to $n_5 + n_6 + n_7$.

The cardinality of such a set expression is called a *composite joint property*. A special case of composite joint property is the cardinality of $N_0{}^c$, which is equal to the union of all elementary subsets N_x, $1 \leq x < 2^k$. We denote such a union cardinality as n_0^c, which is equal to $\sum_{1 \leq x < 2^k} n_x$.

In a word, we emphasize on deriving the $2^k - 1$ elementary joint properties n_x, and the composite joint properties, whose total number is huge, are left to a secondary position.

Probabilistic Estimation. In many practical applications, it often does not require to know the exact value of the joint property n_x, and an approximated value $\hat{n_x}$ with desired accuracy is adequate. In the following problem definition, we require the absolute estimation error $\hat{n_x} - n_x$ to be bounded by a predefined range $\pm\theta$ at a probability of at least $1 - \delta$, which is called the (θ, δ) model. Besides n_x, we also consider to keep the absolute estimation error of n_0^c bounded by $\pm\theta$, which can act as a representative of composite joint property.

Definition 1 (Joint Property Estimation Problem). *For joint properties n_0^c or n_x ($1 \leq x < 2^k$), the joint estimation problem is to find an algorithm for generating estimations $\hat{n_0^c}$ and $\hat{n_x}$. They should satisfy the accuracy constraint:*

$$\begin{aligned} Prob\{n_0^c - \theta \leq \hat{n_0^c} \leq n_0^c + \theta\} &\geq 1 - \delta \\ Prob\{n_x - \theta \leq \hat{n_x} \leq n_x + \theta\} &\geq 1 - \delta, \end{aligned} \tag{4}$$

where $n_0^c \pm \theta$ and $n_x \pm \theta$ are the confidence interval of estimations $\hat{n_0^c}$ and $\hat{n_x}$, respectively, and $1 - \delta$ is the confidence level.

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

$$\begin{aligned} Prob\{n_0^c\,(1 - \epsilon) \leq \hat{n_0^c} \leq n_0^c\,(1 + \epsilon)\} &\geq 1 - \delta \\ Prob\{n_x(1 - \epsilon) \leq \hat{n_x} \leq n_x\,(1 + \epsilon)\} &\geq 1 - \delta. \end{aligned} \tag{5}$$

According to this model, the probabilities for the relative errors $\frac{\hat{n_0^c} - n_0^c}{n_0^c}$ and $\frac{\hat{n_x} - n_x}{n_x}$ to fall into the range $\pm\epsilon$ are at least $1 - \delta$.

This relative error model has been adopted by the previous work [8, 18], with a time complexity of $O(\frac{1}{\epsilon^2 J} \ln \frac{1}{\delta})$, where J is *Jaccard similarity* — the ratio of the intersection size of all tag sets to the union size of all sets. However, $\frac{1}{J}$ could be very large, since the intersection size can be small or even zero while the union size is very large, which may be the routine case in practice, instead of being rare. For the applications in the introduction, there can be two warehouses with few products moved between them, or there can be times when few products are moved in or out of a warehouse. In both cases, $\frac{1}{J}$ is very large or even tends to infinite.

In conclusion, the relative error model, as a remanent from the earlier literature on cardinality estimation of a single tag set, is no longer suitable for joint property estimation of multiple sets. Therefore, this paper adopts the absolute error model in (4). The previous protocols named DiffEstm [18] and CCF [8] are not designed under this model.

3. ALOHA-BASED RFID PROTOCOL

This section introduces a standardized RFID communication protocol based on slotted ALOHA, which can be used to take a snapshot of a tag set without collecting any tag IDs.

3.1 ALOHA Communication Protocol

A reader communicates with the tags in its radio range, using the following slotted ALOHA protocol, which is partially compliant with the EPCglobal RFID standard [1].

Initially, the reader broadcasts a `Query` command to start an ALOHA frame with m time slots and using R as random seed. Upon hearing the command, a tag selects a time slot in a pseudorandom fashion by a hash function $H(id \oplus R)$ mod m, where m is the frame length, id is the tag ID, and $\oplus$ is bitwise XOR that mixes tag id and random seed R.

Then, the reader transmits the m time slots one by one. At the boundary of each two adjacent slots, it broadcasts a `QueryRep` command to terminate the current slot and start the next. As the slot index grows, the tags whose generated hash values are consistent with the current slot index will send out their responses.

By executing the above framed-slotted ALOHA protocol, from the perspective of the reader, the responses of all tags distribute uniformly in the frame with m time slots. Furthermore, a sampling mechanism can be incorporated into the ALOHA frame as a field in the frame header. Due to the sampling, only p fraction ($0 < p \leq 1$) of tags respond in the frame and the rest of them just keep silent.

3.2 Empty/Busy Time Slots

A time slot is said to be *empty* if it contains no tag replies, or *busy* if there are at least one tags responses. Busy slots can be further classified into *singleton* slots (containing exactly one tag response) and *collision* slots (containing at least two tag responses). According to the EPCglobal RFID standard [1], each tag response needs to contain a 16-bits random number in order to differentiate singleton slots from collision ones (or 96 bits for reporting a tag ID). By contrast, if it only needs to distinguish empty slots from busy ones, transmitting merely one bit is sufficient for indicating the presence of a tag in the slot, which may reduce by multiple folds the average time cost per slot. Symbolically, we use "0" to denote empty state and "1" to denote busy state.

We will design protocols that only require the RFID reader to differentiate between empty slots and busy ones. From the perspective of reader, an ALOHA frame is equivalent to an array of bits recording the 0/1 states of the sequence of time slots, which is called a *bitmap* that encodes the current set of tags [5]. Afterwards, the three terms "tag set snapshot", "frame" and "bitmap" will be used interchangeably.

We assume that the RFID reader at each location invokes the ALOHA protocol periodically to take a snapshot of the set of tags currently in its radio range. For the tag sets S_1, S_2, …, S_k in different spatial and temporal domains, let B_1, B_2, …, B_k be the bitmaps that encode them, respectively.

4. A BASELINE SOLUTION

In this section, for implementing the joint property estimation, we present a baseline solution that uses the bitmaps $B_1, B_2, \ldots, B_k$. This protocol implicitly assumes that all these bitmaps must have an equal length and use a common sampling probability, because it needs to apply the bitwise OR operation to any subset of these bitmaps.

This baseline protocol is doomed to have poor time efficiency, which is explained as follows. For a small tag set, if the sampling probability is very small, too few or even no tag will be sampled for the snapshot construction. Hence, the sampling probability has to be reasonably large, as depicted by the first bitmap in Fig. 2, where 20 tags are recorded with 40% sampling probability. However, for a large set, a significant sampling probability will cause all bits to be set as ones (as illustrated by the second bitmap in Fig. 2), unless the bitmap length is sufficiently large (see the third bitmap of Fig. 2). Now because the same large length has to be applied to all bitmaps, it becomes a great waste for small tag sets (the fourth bitmap of Fig. 2). Since each bit takes one time slot to determine, a large bitmap length implies a long time for taking a snapshot, even for a very small tag set.

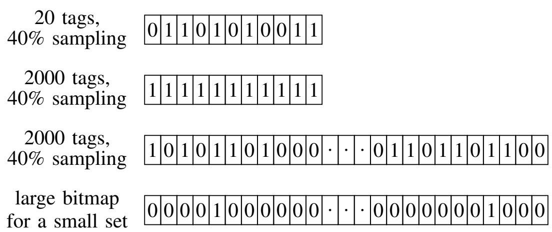

Figure 2: Inefficiency problem of baseline protocol when handling small tag sets and large tag sets.

Therefore, the baseline protocol will waste execution time, if the tag sets it handles dramatically differ in sizes. But still we need it to compare with ours when conducting simulation studies. So we describe its estimation equations as follows.

Union Cardinality Estimation. For the frames $B_1, B_2, \ldots,$ B_k, an important property is that, if a tag is sampled to reply, it will respond in the same time slot among all frames. This is because the tag uses the same hash function $H(id \oplus R) \bmod m$ for slot selection in different frames, where m is the common length of all frames and R is the random seed.

Out of the k bitmaps, we can arbitrarily select c bitmaps and combine them by bitwise OR, which is denoted as $B_{i_1} \vee B_{i_2} \ldots \vee B_{i_c}$ with $1 \le i_1 < i_2 \ldots < i_c \le k$. Because a tag responds at the same slot in all these frames, when calculating the bitwise OR, its duplicated responses in different frames will be filtered automatically. Therefore, the OR of c bitmaps $B_{i_1} \vee B_{i_2} \ldots \vee B_{i_c}$ is equivalent to the bitmap encoding of the union of c tag sets $S_{i_1} \cup S_{i_2} \ldots \cup S_{i_c}$ [5].

For the cardinality of such a union set, a good estimator is to use the fraction of zero bits in the corresponding OR bitmap [5]. Specifically, let z be the fraction of zero bits in an OR bitmap. Then, the corresponding union cardinality can be estimated as $-m\log(z)/p$, where m is the number of bits in the OR bitmap, and p is the sampling probability.

Joint Property Estimation. Since the cardinality of any union set is known, we can easily derive the joint property $n_{2^k-1} = |S_1 \cap S_2 \ldots \cap S_k|$, which is the cardinality of the intersection of all the k tag sets. According to the well-known *principle of inclusion and exclusion*, n_{2^k-1} is equal to

$$\begin{aligned} |S_1 \cap S_2 \ldots \cap S_k| = & \sum_{1 \le i_1 \le k} |S_{i_1}| - \sum_{1 \le i_1 < i_2 \le k} |S_{i_1} \cup S_{i_2}| \\ & + \sum_{1 \le i_1 < i_2 < i_3 \le k} |S_{i_1} \cup S_{i_2} \cup S_{i_3}| + \ldots \\ & + (-1)^{k-1} |S_1 \cup S_2 \ldots \cup S_k|, \end{aligned} \tag{6}$$

where the union cardinalities $|S_{i_1}|$, $|S_{i_1} \cup S_{i_2}|$, $|S_{i_1} \cup S_{i_2} \cup S_{i_3}|$, $\ldots$, $|S_1 \cup S_2 \ldots \cup S_k|$ all have unbiased estimators as stated.

Interestingly, (6) can be extended to estimating any joint property n_x with $1 \le x < 2^k$. By the definition of N_x in (3), $n_x = \big|\bigcap_{2^{i-1} \wedge x \neq 0}^{1 \le i \le k} S_i\big| - \big|\big(\bigcap_{2^{i-1} \wedge x \neq 0}^{1 \le i \le k} S_i\big) \cap \big(\bigcup_{2^{i-1} \wedge x = 0}^{1 \le i \le k} S_i\big)\big|$. The first term $|\cap_{2^{i-1} \wedge x \neq 0}^{1 \le i \le k} S_i|$ is the intersection of multiple tag sets and hence can be estimated using (6). The second term is the intersection of $\bigcap_{2^{i-1} \wedge x \neq 0}^{1 \le i \le k} S_i$ and $\bigcup_{2^{i-1} \wedge x = 0}^{1 \le i \le k} S_i$, where the latter can be treated as a single tag set encoded into the OR bitmap $\bigvee_{2^{i-1} \wedge x = 0}^{1 \le i \le k} B_i$. Hence, the second term can be regarded as the intersection of multiple sets with a union set, and hence can also be estimated by (6).

Because this baseline protocol depends on the INClusion-EXClusion principle in (6) for estimating the intersection of multiple sets from their unions, we call it INC-EXC for short.

5. ADAPTIVE ESTIMATION PROTOCOL

We presents our Joint RFID Estimation Protocol named M-JREP to derive joint properties of Multiple tag sets, even when they are encoded into bitmaps of different lengths.

Naturally, it is desirable to let each bitmap have a different length, depending on the cardinality of the tag set it encodes. This inspires us to develop a new algorithm of combining k bitmaps with variable lengths. The real difficulty is not at how to combine k bitmaps; there are simple ways to combine them. The real difficulty comes after the combination — how to perform analysis on the information combined from non-uniformly sized snapshots, how to use that information for joint property estimation, and most importantly, how to ensure the satisfaction of accuracy requirements in (4). These are the tasks that have not been fulfilled in the literature.

Our M-JREP protocol is comprised of two components: an online encoding component for compressing each tag set into a bitmap, and an offline analysis component for estimating the joint properties of multiple tag sets, using their bitmap-based snapshots. Whenever a bitmap encoding of a tag set is collected by the online phase, the RFID reader offloads it immediately to a central server for long-term storage and for offline processing. Such an asymmetric design will push most complexity to the offline component at the server side, while keeping the online component at the side of reader and tags (for raw data collection) as efficient as possible. We will describe the online component in the first subsection, and then the offline component in the subsequent subsection.

5.1 Online Encoding of a Tag Set

We use a two-phase protocol to encode tag set S_i into a bitmap with length proportional to the set size s_i. The first phase generates an estimation for the number of tags s_i with coarse accuracy, and the second phase uses the coarse estimation $\hat{s_i}$ to configure ALOHA frame length and re-scan the tag set. It in fact is a common practice for RFID researchers to use such a two-phase protocol for estimating the cardinality of a single tag set [3], which will be explained later in section 9 for related works. In following, we describe the two-phase protocol with more details.

Firstly, the RFID reader that covers the tag set S_i invokes an existing protocol to generate a coarse estimation of the set size s_i. Because only a single tag set is handled in this phase, the estimation accuracy is specified by the (ϵ, δ) model in (5), in which the probability for the relative estimation error to fall within the range $\pm\epsilon$ is at least $1 - \delta$. Since the accuracy is rather coarse, we often configure $\epsilon = 20\%$ and $\delta = 5\%$. To attain the goal, many existing protocols can be applied, such as LoF [14], GMLE [7], PET [20] and ZOE [19]. In order to attain the pre-defined estimation accuracy of tag set size, the needed number of time slots for scanning the tag set is $\mathcal{O}(\frac{1}{\epsilon^2}\log(s_{\max})) \cdot \log(\frac{1}{\delta})$ for LoF, or $\mathcal{O}(\frac{1}{\epsilon^2}\log\log(s_{\max})) \cdot \log(\frac{1}{\delta})$ for PET, where $s_{\max}$ is an upper bound for the size of any tag set, according to a recent survey study [3]. It is clear that the time expense is proportional to the *logarithm* or even the *log-logarithm* of the tag set size. Hence, the time cost of the first phase is negligibly small when its accuracy requirement is coarse. For instance, when the relative estimation error ϵ is 20% and δ is 5%, the time cost of the LoF algorithm [14] is $32\log(s_{\max})$.

Secondly, because the size of tag set has been coarsely estimated by the first phase as $\hat{s_i}$, the reader can re-scan the tag set by an ALOHA frame B_i whose length m_i is *linearly* proportional to $\hat{s_i}$. Then, the frame length m_i satisfies

$$m_i = \min_{p\in(0,1]} \{2^{\lceil \log_2(\frac{\hat{s_i}}{\rho})\rceil}\}, \tag{7}$$

where ρ is the load factor of the frame. Note that the frame length m_i is explicitly configured to an integral power of two. Then, for any two frames, the length of the longer frame is always integral multiple of the length of the shorter frame.

Equation (7) can be approximated as the primary time cost of the online component for tag set encoding. Later in section 7.1, we will prove that an optimized value of the load factor ρ that minimizes (7) while satisfying the pre-defined accuracy constraint of joint property estimation in (4) is

$$\rho = \frac{1}{2p\,k_{\max}}\left(-3+\sqrt{3}\sqrt{8p\left(\frac{\theta^2}{k_{\max}s_{\max}Z_\delta{}^2}+1\right)-5}\right), \tag{8}$$

where θ is the bound of absolute estimation error, Z_δ is the $1-\frac{\delta}{2}$ quantile of standard Gaussian distribution (e.g., $Z_{0.05}\approx 1.96$), $s_{\max}$ is the upper bound of the cardinality of a tag set, $k_{\max}$ is the largest number of tag sets that may involve in any user query, and p is the sampling probability. Since $s_{\max}$, $k_{\max}$, θ and Z_δ are all fixed values, (8) can be regarded a function of only one variable p. Let p^* be the optimal sampling probability that maximizes (8), which in turn will minimize the time cost of encoding a tag set in (7). It is clear that the value of p^* only depends on $s_{\max}$, $k_{\max}$, θ and δ. Hence, p^* is pre-determined for a system once these parameters are set, and the best ρ is also pre-determined.

5.2 Offline Estimation of Joint Properties of Multiple Tag Sets

In this subsection, we present an offline analysis algorithm that derives the joint properties n_0^c and n_x, $1\le x<2^k$, of k tag sets, using the bitmaps $B_1, B_2, \ldots, B_k$. Without loss of generality, we assume the bitmaps are sorted by their lengths in non-descending order that satisfies $m_1 \le m_2 \le \ldots \le m_k$.

Although all these bitmaps are assumed by equation (7) to have the same load factor ρ, our offline algorithm to describe later can in fact work well if each bitmap B_i has its own load factor ρ_i. We will prove in Section 6 that, only when $\rho_1 = \rho_2 = \ldots = \rho_k = \rho$, can we minimize the protocol time cost.

Expanded Bitwise OR. We introduce two bitwise operations which will be used later. In the binary representation of a value y, let $lo(y)$ be the location of the lowest-order 1-bit, and let $hi(y)$ be the location of the highest-order 1-bit. For example, if the binary representation of y is 1010001, then we have $lo(y)=1$ and $hi(y)=7$. A boundary case is that y is equal to zero. In this case, we define $hi(y)=0$ and $lo(y)=0$.

For the bitmaps, we introduce an auxiliary bitmap called *expanded OR* to combine them, which is noted by $OR_{b_k\ldots b_2b_1}$.

$$OR_{b_k\ldots b_2b_1} = \bigvee_{b_i\neq 0}^{1\le i\le k} Expand(B_i,\ m_{hi(b_k\ldots b_2b_1)})$$

The subscript $b_k\ldots b_2b_1$ indicates for each bitmap whether it is involved in the expanded OR: If b_i is one, then the bitmap B_i is involved. Among the chosen bitmaps, the length of the longest bitmap is $m_{hi(b_k\ldots b_2b_1)}$, because $m_1\le m_2\ldots\le m_k$ and $hi(b_k\ldots b_2b_1)$ is the position of the highest 1-bit in binary $b_k\ldots b_2b_1$. The function $Expand(B_i,\ m_{hi(b_k\ldots b_2b_1)})$ increases the length of B_i to the largest bitmap length $m_{hi(b_k\ldots b_2b_1)}$ by self replication, such that all bitmaps after expansion have an equal length and can be combined by bitwise OR $\bigvee$.

Figure 3 illustrates an example of applying expanded OR to three bitmaps B_1, B_2 and B_3. Among them, B_3 is the longest. We replicate B_1 for one time and B_2 for three times, such that after expansion, all bitmaps are of the same length and the bitwise OR operation can be used to combine them.

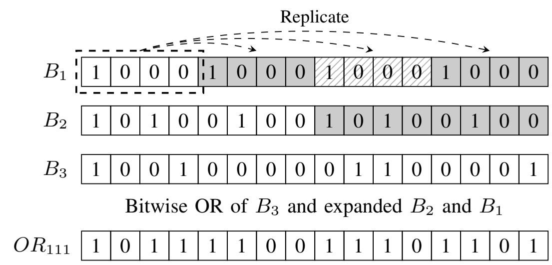

Figure 3: Expanded OR of three bitmaps B_1, B_2, B_3, whose lengths are 4, 8, 16, respectively.

For simplicity, we replace $OR_{b_k\ldots b_2b_1}$ by a shorter notation OR_y, where y is a decimal equal to $b_k\ldots b_2b_1$. Therefore,

$$OR_y = \bigvee_{2^{i-1}\wedge y\neq 0}^{1\le i\le k} Expand(B_i,\ m_{hi(y)}). \tag{9}$$

It is always feasible to expand the bitmap B_i to have the same length with the longest bitmap $B_{hi(y)}$, because both their lengths m_i and $m_{hi(y)}$ are the powers of two by equation (7), and the ratio $m_{hi(y)}/m_i$ is definitely an integer.

Expected Zero Fraction of $\mathbf{OR_y}$. We analyze the expected fraction of zero bits in the bitmap OR_y. Let us focus on just one bit in OR_y, and we have the following property.

PROPERTY 1 (PROBABILITY OF A TAG ASSIGNING A BIT). *Considering an arbitrary bit in bitmap OR_y and an arbitrary tag from elementary subset N_x, when the bitwise AND of x and y is non-zero, the probability of the tag assigning the bit to one is $\frac{p}{m_{lo(x\wedge y)}}$; when $x\wedge y$ is zero, the probability is zero.*

PROOF. Please check out the Appendix A. □

According to Property 1, a tag has the chance to assign a bit of OR_y to one, only when it is in a subset N_x that satisfies $x\wedge y\neq 0$. Because in such a subset N_x, any tag assigns the jth bit of OR_y at a probability of $\frac{p}{m_{lo(x\wedge y)}}$, the probability that all tags in N_x do not assign this bit is $(1-\frac{p}{m_{lo(x\wedge y)}})^{n_x}$.

Let $X_y^{(j)}$ be the event that the jth bit in OR_y remains zero. The occurrence of the event needs all tags in any N_x with $x\wedge y\neq 0$ do not assign this bit. Thus, its probability is

$$Prob\{X_y^{(j)}\} = \prod_{1\le x<2^k}^{x\wedge y\neq 0}\left(1-\frac{p}{m_{lo(x\wedge y)}}\right)^{n_x}. \tag{10}$$

Let z_y be the fraction of bits in OR_y that remains zeros:

$$z_y = \frac{1}{m_{hi(y)}}\sum_{0\le j<m_{hi(y)}} 1_{X_y^{(j)}}, \tag{11}$$

where $m_{hi(y)}$ is the number of bits in bitmap OR_y (see (9)), and $1_{X_y^{(j)}}$ is the indicator function of the event $X_y^{(j)}$, whose value is one if the event occurs and is zero otherwise. Since the zero fraction z_y is the arithmetic mean of a large number of independent variables, by the central limit theorem, z_y approximates a Gaussian distribution. Its expected value is

$$E(z_y) = E\left(\frac{1}{m_{hi(y)}}\sum 1_{X_y^{(j)}}\right) = \frac{1}{m_{hi(y)}}\sum_{0\le j<m_{hi(y)}} E(1_{X_y^{(j)}}).$$

Clearly, $E(1_{X_y^{(j)}}) = Prob\{X_y^{(j)}\}$. Hence, applying (10),

$$\begin{aligned} E(z_y) &= \tfrac{1}{m_{hi(y)}} \textstyle\sum_{0 \le j < m_{hi(y)}} Prob\{X_y^{(j)}\} \\ &= Prob\{X_y^{(j)}\} = \textstyle\prod_{1 \le x < 2^k}^{x \wedge y \ne 0} \big(1 - \tfrac{p}{m_{lo(x \wedge y)}}\big)^{n_x}. \end{aligned}$$

Applying the approximation $(1 - \frac{p}{m})^n \approx (1 - \frac{p}{m'})^{\frac{m'}{m} n}$ which works when m and m' are both large, we have

$$\begin{aligned} E(z_y) &\approx \textstyle\prod_{1 \le x < 2^k}^{x \wedge y \ne 0} \big(1 - \tfrac{p}{m_{hi(y)}}\big)^{\frac{m_{hi(y)}}{m_{lo(x \wedge y)}} \cdot n_x} \\ &= \big(1 - \tfrac{p}{m_{hi(y)}}\big)^{\sum_{1 \le x < 2^k}^{x \wedge y \ne 0} \frac{m_{hi(y)}}{m_{lo(x \wedge y)}} \cdot n_x}. \end{aligned}$$

Using the sign function sgn (which equals to 1, 0 or -1 when its input parameter is positive, zero or negative), we have

$$E(z_y) \approx \big(1 - \tfrac{p}{m_{hi(y)}}\big)^{\sum_{1 \le x < 2^k} \mathrm{sgn}(x \wedge y) \cdot \frac{m_{hi(y)}}{m_{lo(x \wedge y)}} \cdot n_x}. \quad (12)$$

Estimator of Joint Property n_x. Using the fraction of zero bits in OR_y with $1 \le y < 2^k$, we are able to estimate each joint property n_x with $1 \le x < 2^k$. We will show later that this essentially is a *fully determined linear system*, which puts $2^k - 1$ constraints over $2^k - 1$ unknown variables.

By (11), we know that the variance of z_y is inversely proportional to the number of observations $m_{hi(y)}$. Hence, when the number of slots $m_{hi(y)}$ in the frame OR_y is sufficiently large, we can approximate $E(z_y)$ by z_y. Then, (12) becomes

$$z_y \approx \big(1 - \tfrac{p}{m_{hi(y)}}\big)^{\sum_{1 \le x < 2^k} \mathrm{sgn}(x \wedge y) \cdot \frac{m_{hi(y)}}{m_{lo(x \wedge y)}} \cdot n_x}.$$

We will prove later that such an approximation indeed produces unbiased estimators. Taking the logarithm of both sides,

$$\log(z_y) / \log\big(1 - \tfrac{p}{m_{hi(y)}}\big) \approx \textstyle\sum_{1 \le x < 2^k} \mathrm{sgn}(x \wedge y) \cdot \tfrac{m_{hi(y)}}{m_{lo(x \wedge y)}} \cdot n_x.$$

Applying the approximation $\log(1 - \frac{p}{m}) \approx -\frac{p}{m}$ for large m,

$$-\tfrac{m_{hi(y)}}{p} \log(z_y) \approx \textstyle\sum_{1 \le x < 2^k} \mathrm{sgn}(x \wedge y) \cdot \tfrac{m_{hi(y)}}{m_{lo(x \wedge y)}} \cdot n_x.$$

If we define the measurement of the number of tags in OR_y as

$$\hat{u_y} = -\tfrac{m_{hi(y)}}{p} \log(z_y), \quad (13)$$

the above equation can be simplified as

$$\textstyle\sum_{1 \le x < 2^k} \mathrm{sgn}(x \wedge y) \cdot \tfrac{m_{hi(y)}}{m_{lo(x \wedge y)}} \cdot n_x \approx \hat{u_y}. \quad (14)$$

Applying (13) to each OR_y bitmap, we can collect a vector of measurements $\hat{\mathbf{u}} = \big[\hat{u_1}, \hat{u_2}, \ldots, \hat{u_y}, \ldots, \widehat{u_{2^k-1}}\big]^T$. If putting together the sizes of all elementary subsets, we can obtain the vector of unknowns: $\mathbf{n} = \big[n_1, n_2, \ldots, n_x, \ldots, n_{2^k-1}\big]^{\mathrm{T}}$. With $\hat{\mathbf{u}}$ and $\mathbf{n}$ properly defined, (14) can be rewritten as

$$\mathbf{M}\,\mathbf{n} \;\approx\; \hat{\mathbf{u}}, \quad (15)$$

where $\mathbf{M}$ is the coefficient matrix defined in (16). Its element uses y as row index and x as column index, $1 \le x, y < 2^k$.

$$\mathbf{M} = \left[\mathrm{sgn}(x \wedge y) \cdot \frac{m_{hi(y)}}{m_{lo(x \wedge y)}}\right] \quad (16)$$

For elements of $\mathbf{M}$, if $x \wedge y$ is zero, then $\mathrm{sgn}(x \wedge y)$ is zero, and the expression $\mathrm{sgn}(x \wedge y) \cdot \frac{m_{hi(y)}}{m_{lo(x \wedge y)}}$ is treated as zero.

In Appendix B of the extended version [12], we prove that the coefficient matrix $\mathbf{M}$ is non-singular and provide a recursive formula for calculating $\mathbf{M}^{-1}$. Hence, we can solve the equation system and obtain an estimator of joint properties.

$$\hat{\mathbf{n}} \;=\; \mathbf{M}^{-1}\,\hat{\mathbf{u}} \quad (17)$$

Estimator of Composite Joint Property n_0^c. Among the numerous composite joint properties, the largest and the most important property is n_0^c — the number of tags in the union of all the sets. We estimate it as $\hat{n_0^c} = \sum_{1 \le x < 2^k} \hat{n_x}$, the sum of all elements in the vector $\hat{\mathbf{n}}$. After simplification,

$$\begin{aligned} \hat{n_0^c} = \widehat{u_{2^k-1}} &- \frac{m_2 - m_1}{m_1}\hat{u_1} - \frac{m_3 - m_2}{m_2}\hat{u_3} \ldots \\ &- \frac{m_{j+1} - m_j}{m_j}\widehat{u_{2^j-1}} \ldots - \frac{m_k - m_{k-1}}{m_{k-1}}\widehat{u_{2^{k-1}-1}}. \quad (18) \end{aligned}$$

6. THEORETICAL ANALYSIS

In this section, we analyze the bias and variance of the M-JREP estimators. It is easy to prove that the estimators $\hat{n_x}$ in (17) and $\hat{n_0^c}$ in (18) are asymptotically unbiased, due to the rigid process by which they are derived. We place the detailed proof of asymptotic unbiasedness in Appendix C of the extended version [12]. In following, we focus on analyzing their variances, which determine their estimation errors.

6.1 Probabilistic Distribution of Measurements z_y

Because the zero fraction z_y of bitmap OR_y is the input of M-JREP estimator, we need to first analyze its mean and variance. By (11), the zero ratio z_y is the arithmetic mean of independent variables. When the number of variables $m_{hi(y)}$ is large, according to the central limit theorem, z_y approximates a Gaussian distribution. The mean value of z_y is given in (12), and can be further simplified as $E(z_y) \approx e^{-p\,\omega_y}$, where ω_y is defined below and its physical meaning is the load factor of OR_y.

$$\omega_y = \textstyle\sum_{1 \le x < 2^k} \mathrm{sgn(x \wedge y)} \cdot \tfrac{n_x}{m_{lo(x \wedge y)}} \quad (19)$$

Note that the symbol ω_y is different from ρ_i, which later will be used to denote the load factor of the frame B_i.

We derive in Appendix D [12] that the covariance of zero ratio z_{y_1} of OR_{y_1} and zero ratio z_{y_2} of OR_{y_2} is approximately

$$Cov(z_{y_1}, z_{y_2}) \approx \frac{e^{-p\omega_{y_1 \vee y_2}} - (1 + p^2 \omega^*_{y_1, y_2})\, e^{-p(\omega_{y_1} + \omega_{y_2})}}{\min(m_{hi(y_1)}, m_{hi(y_2)})}, \quad (20)$$

where $\omega^*_{y_1,y_2}$ is density of common tags of OR_{y_1} and OR_{y_2}.

$$\omega^*_{y_1,y_2} = \min(m_{hi(y_1)}, m_{hi(y_2)}) \textstyle\sum_{x \wedge y_2 \ne 0}^{x \wedge y_1 \ne 0} \tfrac{n_x}{m_{lo(x \wedge y_1)} m_{lo(x \wedge y_2)}} \quad (21)$$

6.2 Variance of Cardinality Measurements $\hat{u_y}$

We have defined the measurement of the number of tags $\hat{u_y}$ in the OR bitmap OR_y in (13). We will analyze the covariance of cardinality measurement $\hat{u_{y_1}}$ of bitmap OR_{y_1} and cardinality measurement $\hat{u_{y_2}}$ of bitmap OR_{y_2}.

$$Cov(\hat{u_{y_1}}, \hat{u_{y_2}}) = Cov\big(-\tfrac{m_{hi(y_1)}}{p} \log(z_{y_1}), -\tfrac{m_{hi(y_2)}}{p} \log(z_{y_2})\big)$$

Proved in Appendix E [12], the above equation approximates

$$Cov(\hat{u_{y_1}}, \hat{u_{y_2}}) \approx \tfrac{m_{hi(y_1)} m_{hi(y_2)}}{p^2\, e^{-p\,(\omega_{y_1} + \omega_{y_2})}}\, Cov(z_{y_1}, z_{y_2}).$$

By substituting $Cov(z_{y_1}, z_{y_2})$ with its approximation in (20),

$$\begin{aligned} Cov(\hat{u_{y_1}}, \hat{u_{y_2}}) \approx \frac{1}{p^2} &\max(m_{hi(y_1)}, m_{hi(y_2)}) \cdot \\ &\big(e^{p\,(\omega_{y_1} + \omega_{y_2} - \omega_{y_1 \vee y_2})} - (1 + p^2 \omega^*_{y_1,y_2})\big), \quad (22) \end{aligned}$$

where ω_y can be found in (19), and $\omega^*_{y_1,y_2}$ is defined in (21). When $y_1 = y_2 = y$, the above covariance becomes $Var(\hat{u_y})$.

$$Var(\hat{u_y}) \approx \tfrac{1}{p^2} m_{hi(y)} \big(e^{p\,\omega_y} - \big(1 + p^2 \sum_{x \wedge y \ne 0} \tfrac{m_{hi(y)}}{m_{lo(x \wedge y)}^{\;2}} n_x\big)\big) \quad (23)$$

6.3 Estimation Variance of Joint Property

The covariance $Cov(\hat{n_{x_1}}, \hat{n_{x_2}})$ of any two joint property estimations $\hat{n_{x_1}}$ and $\hat{n_{x_2}}$, $1 \le x_1, x_2 < 2^k$, can form a matrix $Cov(\hat{\mathbf{n}}, \hat{\mathbf{n}})$. According to the proof in Appendix F of [12],

$$Cov(\hat{\mathbf{n}}, \hat{\mathbf{n}}) = \mathbf{M}^{-1}\, Cov(\hat{\mathbf{u}}, \hat{\mathbf{u}})\, (\mathbf{M}^{-1})^T, \tag{24}$$

where $Cov(\hat{\mathbf{u}}, \hat{\mathbf{u}})$ is a covariance matrix whose entry is in (22).

In following, we analyze the estimation variance of the composite joint property n_0^c. Equation (18) is rewritten as

$$\hat{n_0^c} = \hat{u_1} + (-\tfrac{m_2}{m_1}\hat{u_1} + \hat{u_3}) + (-\tfrac{m_3}{m_2}\hat{u_3} + \hat{u_7}) \ldots + (-\tfrac{m_j}{m_{j-1}} \cdot \widehat{u_{2^{j-1}-1}} + \widehat{u_{2^j-1}}) \ldots + (-\tfrac{m_k}{m_{k-1}}\widehat{u_{2^{k-1}-1}} + \widehat{u_{2^k-1}}). \tag{25}$$

To simplify this equation, we define $\hat{d_j}$ as

$$\hat{d_j} = -\tfrac{m_j}{m_{j-1}}\widehat{u_{2^{j-1}-1}} + \widehat{u_{2^j-1}}, \quad \text{with } 2 \le j \le k. \tag{26}$$

We have proved in Appendix C of [12] that $\hat{d_j}$ is an unbiased estimation of $d_j = |S_j \setminus (S_1 \cup S_2 \ldots \cup S_{j-1})|$. Applying (26) to (25), we have

$$\hat{n_0^c} \;=\; \hat{u_1} + \textstyle\sum_{2 \le j \le k} \hat{d_j}. \tag{27}$$

Then, $Var(\hat{n_0^c}) = Var(\hat{u_1} + \sum_{2 \le j \le k} \hat{d_j})$. In Appendix G of the extended version [12], we prove that $Cov(\hat{d_i}, \hat{d_j}) \approx 0$ for any i, j values with $2 \le i < j$, and $Cov(\hat{u_1}, \hat{d_i}) \approx 0$. Hence,

$$Var(\hat{n_0^c}) \approx Var(\hat{u_1}) + \textstyle\sum_{2 \le j \le k} Var(\hat{d_j}), \tag{28}$$

where $Var(\hat{u_y})$ is given in (23). By definition of $\hat{d_j}$ in (26),

$$Var(\hat{d_j}) = \tfrac{{m_j}^2}{{m_{j-1}}^2} Var(\widehat{u_{2^{j-1}-1}}) + Var(\widehat{u_{2^j-1}}) - 2\tfrac{m_j}{m_{j-1}} \cdot Cov(\widehat{u_{2^{j-1}-1}}, \widehat{u_{2^j-1}}).$$

By (22), $Cov(\widehat{u_{2^{j-1}-1}}, \widehat{u_{2^j-1}}) \approx \frac{m_j}{m_{j-1}} Var(\widehat{u_{2^{j-1}-1}})$. Then,

$$Var(\hat{d_j}) \approx -\tfrac{{m_j}^2}{{m_{j-1}}^2} Var(\widehat{u_{2^{j-1}-1}}) + Var(\widehat{u_{2^j-1}}). \tag{29}$$

Applying the above equation of $Var(\hat{d_j})$ to (28), we have

$$\begin{aligned} Var(\hat{n_0^c}) &\approx Var(\hat{u_1}) + \sum_{2 \le j \le k} \big(Var(\widehat{u_{2^j-1}}) - \tfrac{{m_j}^2}{{m_{j-1}}^2} Var(\widehat{u_{2^{j-1}-1}})\big) \\ &= Var(\widehat{u_{2^k-1}}) - \textstyle\sum_{1 \le j < k} \tfrac{{m_{j+1}}^2 - {m_j}^2}{{m_j}^2} Var(\widehat{u_{2^j-1}}). \end{aligned} \tag{30}$$

7. PROTOCOL PARAMETERS

In this section, we optimize the parameters of M-JREP protocol, under the accuracy constraints of joint property estimation in (4). There are many system parameters. But the size of largest tag sets $s_{\max}$ is a phenomenon of physical world, and is beyond the control of RFID system. The accuracy model (θ, δ) and the number of tag sets k totally depends on the demand of users. Hence, only two parameters are controllable and should be optimized, i.e., the load factor ρ_i of the frame B_i and the common sampling probability p.

In the first subsection, we investigate the appropriate configuration for the load factor ρ_i for frame B_i. In the second subsection, we study how to optimize the sampling probability p to minimize the execution time (or the size of frame B_i).

7.1 Configuration of Load Factors

Equation (4) requires that the probability for the absolute estimation errors of joint properties n_0^c and n_x to fall within $\pm\theta$ is at least $1-\delta$. We proved before that both $\hat{n_x}$ and $\hat{n_0}$ are asymptotically unbiased estimations and they approximate Gaussian distributions. Hence, Eq. (4) can be translated to

$$Var(\hat{n_0^c}) \le (\theta \,/\, Z_\delta)^2 \text{ and } Var(\hat{n_x}) \le (\theta \,/\, Z_\delta)^2, \tag{31}$$

where Z_δ is $1-\frac{\delta}{2}$ quantile of standard Gaussian distribution.

Property 2 (Variance Upper Bounds). *The tight upper bound of $Var(\hat{n_x})$, $1 \le x < 2^k$, and $Var(\hat{n_0^c})$ is $Var(\widehat{u_{2^k-1}})$.*

$$Var(\hat{n_x}), Var(\hat{d_j}) \le Var(\hat{n_0^c}) \le Var(\widehat{u_{2^k-1}}) \tag{32}$$

Meanwhile, $Var(\widehat{u_{2^k-1}})$ is tightly upper bounded by

$$Var(\widehat{u_{2^k-1}}) \le \tfrac{1}{p^2}\tfrac{s_k}{\rho_k}\big(e^{p\sum_{1 \le i \le k}\rho_i} - (1 + p^2 \textstyle\sum_{1 \le i \le k}\rho_i)\big). \tag{33}$$

Proof. From (30), $Var(\hat{n_0^c}) \le Var(\widehat{u_{2^k-1}})$. For proof of other parts, see Appendix H of the extended version [12]. □

By the above property, $Var(\hat{n_0^c})$ and $Var(\hat{n_x})$ are tightly upper bounded by (33). Hence, the two constraints in (31) can be tightened as

$$\tfrac{1}{p^2}\tfrac{s_k}{\rho_k}\big(e^{p\sum_{1 \le i \le k}\rho_i} - (1 + p^2 \textstyle\sum_{1 \le i \le k}\rho_i)\big) \le (\theta \,/\, Z_\delta)^2, \tag{34}$$

which guarantees that (31) is satisfied even in the worst case.

In our system design, we shall configure $\rho = \rho_1 = \rho_2 \ldots = \rho_k$ as a system-wide optimal load factor, which will be explained at the end of this subsection. Then, (34) becomes

$$\tfrac{1}{p^2}\tfrac{s_k}{\rho}\big(e^{pk\rho} - (1 + p^2 k\rho)\big) \le (\theta \,/\, Z_\delta)^2. \tag{35}$$

In this subsection, we focus on the optimization of the load factor ρ, and keep the sampling probability p temporally fixed, whose optimization is postponed to the next subsection. Equation (35) has no explicit solution for ρ due to the existence of exponential term $e^{pk\rho}$. Hence, we apply the Taylor series $e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \mathcal{O}(x^4)$ to (35). Then,

$$\tfrac{1}{p^2}\tfrac{s_k}{\rho}\big(1 + pk\rho + \tfrac{(pk\rho)^2}{2} + \tfrac{(pk\rho)^3}{6} - (1 + p^2 k\rho)\big) \le \tfrac{\theta^2}{{Z_\delta}^2}$$

$$\rho \le \tfrac{1}{2pk}\big(-3 + \sqrt{3}\sqrt{8p(\tfrac{1}{ks_k}\tfrac{\theta^2}{{Z_\delta}^2} + 1) - 5}\big).$$

Assume the size of any tag set s_i is at most $s_{\max}$, and the number of tag sets k involved in any query is at most $k_{\max}$. In the worst case $s_k = s_{\max}$ and $k = k_{\max}$, we must ensure

$$\rho \le \tfrac{1}{2pk_{\max}}\big(-3 + \sqrt{3}\sqrt{8p(\tfrac{1}{k_{\max}s_{\max}}\tfrac{\theta^2}{{Z_\delta}^2} + 1) - 5}\big). \tag{36}$$

Since $\rho = \rho_i = \frac{s_i}{m_i}$, it is inversely proportional to the frame length m_i, which measures the protocol time cost for encoding the tag set S_i. Thus, we configure the target load factor as large as possible under the constraint, and obtain Eq. (8).

We justify our choice of setting $\rho = \rho_1 = \rho_2 \ldots = \rho_k$. The left side of (34) is an increasing function in each ρ_i. If we allow these load factors to be unequal and still set their values to be as small as possible, then some of them will be greater than the right side of (36) and others will be small. Because $S_1, S_2, \ldots, S_k$ 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 (36) and some other will have smaller load factors. Let $S_1', S_2', \ldots, S_k'$ be the k tag sets with load factors greater than the right side of (36). We should be able to perform joint estimation on any k encoded tag sets without violating the accuracy requirement. However, if we perform joint estimation on $S_1', S_2', \ldots, S_k'$, because their load factors are larger than (36), the constraint of (34) will not hold.

7.2 Optimization of Sampling Probability p

Because $\rho = \rho_i = \frac{s_i}{m_i}$, we have $m_i = \frac{s_i}{\rho}$. Recall that the value of m_i must be a power of two to support the expanded OR of multiple ALOHA frames. Hence, $m_i = 2^{\lceil \log_2(\frac{s_i}{\rho}) \rceil}$. We want to choose the optimal sampling probability that minimizes the protocol execution time by keeping the frame length m_i as small as possible. Hence, we have the formula for the frame length defined by section 5-A as (7) and quoted here: $m_i = \min_{p\in(0,1]}\{2^{\lceil \log_2(\frac{\hat{s_i}}{\rho})\rceil}\}$, where the load factor ρ is determined by (8), and the sampling probability p is hidden inside (8) and needs to be optimized. The optimal p^* that can minimize m_i depends on the pre-determined parameters $s_{\max}$, $k_{\max}$, θ and δ. We can numerically compute from (7) the optimal sampling probability p^* that minimizes m_i.

8. SIMULATION STUDIES

We evaluate the performance of our M-JREP protocol by simulations. The most related work include CCF (Composite Counting Framework [8]), which is based on the relative error model, and DiffEstm (Differential Estimator [18]), which is also based on the relative error model and can only handle two tag sets. Please check Section 2 for discussion on the absolute error model and the relative error model. In this section, we will compare our M-JREP protocol with CCF and INC-EXC. Note that INC-EXC protocol in Section 4 degrades to DiffEstm [18], when it handles only two tag sets.

We will consider two performance metrics. First, given the same accuracy requirement defined in (4), we compare the execution times of all the three protocols. For M-JREP, its execution time is measured as the number of time slots it takes the reader to encode a tag set into a bitmap, including the frame length m_i and other slots needed to give an initial rough estimation of tag set size s_i. We adopt GMLE [7] to generate an initial estimate with a 95% confidence interval of ±20% error. The time cost of GMLE hence is approximately $1.544 \cdot {Z_{0.05}}^2/0.2^2 \approx 148$ slots [7].

Second, when the protocols are subject to the same average execution time, we compare their probabilities of meeting a given error bound $\pm\theta$. 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. When presenting simulation results, we only show the probability of successfully bounding the estimation error of union cardinality n_0^c, and omit the bounding probability of n_x, since by Property 2, $Var(\hat{n_x}) \le Var(\hat{n_0^c})$ and the bounding probability of n_x is always larger than n_0^c.

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 $s_{\max} = 50,000$. The average cardinality of a tag set is $s_{\text{avg}} = 10,000$, 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 Gaussian distribution $\mathcal{N}(10000, 2000^2)$ truncated by the range $(0, s_{\max}]$. For the accuracy requirement, we configure $\delta = 5\%$ and $\theta = 800$ by default. We will perform simulation studies with other values of system parameters δ, θ, $s_{\max}$ and s_{avg} as well.

8.1 Protocol Execution Time to Achieve the Same Estimation Accuracy

We compare the average time cost of the three protocols, which are forced to meet the same accuracy constraint. A critical parameter that affects protocol performance is $k_{\max}$, the largest number of tag sets involved in a query. We perform simulations with $k_{\max}$ set to the value 2, 4, 6 or 8.

Before simulation studies, we explain how to theoretically configure the parameters of M-JREP, i.e., compute the value of load factor ρ from (8) and the optimal sampling probability p^* from (7). For different $k_{\max}$ values, we show the corresponding ρ and p^* in Table 1. We find that in most circumstances with reasonable high accuracy, the optimal sampling probability p^* should be configured close to one in order to avoid sampling error.

Table 1: Parameter Settings for M-JREP Protocol

Number of Sets $k_{\max}$	2	4	6	8
Optimal Sampling Probability p^*	1	1	1	1
Theoretical Value of Load factor ρ	0.86	0.28	0.14	0.09
Empirical Value of Load factor ρ	1.39	0.68	0.35	0.13

The theoretical values of ρ are set conservatively (on the third row of Table 1) to guarantee that the accuracy constraint is satisfied even in the worst case. Alternatively, their values can be set empirically through simulations for normal situations. In our simulation, we first compute the initial value of ρ from (8) and then perform bi-section search to increase it as large as possible such that the resulting value of m_i will still satisfy the accuracy requirement. Consequently, on the last row of the above table, when $k_{\max}$ equals 2, 4, 6 or 8, the load factor ρ is empirically configured to 1.39, 0.68, 0.35 or 0.13, respectively. It shows that, to support the user queries that involve more tag sets, the load factor ρ must decrease, which is consistent with equation (8).

In the second row of Table 2, we present the average execution time of M-JREP in simulations (using the empirical parameters in Table 1) . When INC-EXC and CCF realize the same estimation accuracy, their execution times are shown in the third and fourth rows of Table 2, respectively. Because they are not designed for absolute error bound, there is no formula to compute their frame length or the number of hash values stored. With $s_{max} = 50,000$, we use exhaustive search by simulation to find their minimum time cost that can meet the error bound. Table 2 shows that the frame length used by INC-EXC is at least 500% larger than that of M-JREP, and the time cost of CCF is even higher. This is because EXC-INC (or CCF) has to adopt a large frame length (or store a large amount of tag hash values) to tolerate the worst case of estimating joint properties for $k_{\max}$ tag sets whose cardinalities range between 45,000 and 50,000. This expensive time cost is fixed even when encoding small tag sets whose average size is only about 10,000. The time costs of CCF could get even worse than the results shown in the last row of Table 2, if it is applied to another worst scenario of estimating the intersection of multiple sets, which is empty.

Table 2: Comparison of Time Cost Among Protocols

Number of Sets $k_{\max}$	2	4	6	8
Time Cost of M-JREP	10,274	21,072	40,234	77,044
Time Cost of INC-EXC	50,920	124,725	230,112	384,616
Time Cost of CCF for Union	42,244	168,976	380,196	675,904

To give a straightforward impression on the time costs of M-JREP, INC-EXC and CCF, we configure $k_{\max}$ to 4 and show their comparison results in Fig. 4, where the horizontal axis is the size of a tag set, which varies from 100 to 50,000, and the vertical axis is the number of time slots needed (or hash values stored for CCF) to take a snapshot of the tag set. Due to the nature of their designs, INC-EXC uses a constant frame length of 124,725 slots, and CCF uses constant time cost of recording 168,976 hash values. The frame length of M-JREP is variable. It is small when the tag set is small. For example, for a set of 10,000 tags, the number of time slots needed by M-JREP is $2^{\lceil \log_2(10{,}000/0.68)\rceil} + 148 = 16,532$, only 13% of what is needed by INC-EXC. The average time cost of M-JREP is 21,072 shown by the solid horizontal line.

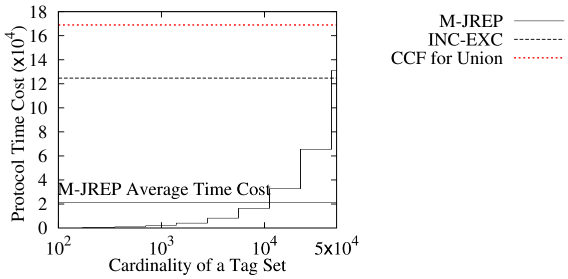

Figure 4: Protocol execution time under the parameter settings: $k_{\max} = 4$, $s_{\max} = 50000$, $\theta = 800$, $\delta = 5\%$.

8.2 Estimation Accuracy under the Same Execution Time

We compare the estimation accuracy of INC-EXC and M-JREP, when giving them the same execution time, which is configured by the second row of Table 2. Here, we omit the results of CCF for space, which are worse than INC-EXC.

When presenting simulation results, a difficulty is that the estimation accuracy is strongly affected by the sizes of tag sets involved and their ways of overlapping. It is impossible to present the simulation results of all the cases, and hence we focus on only two of them. The first is an extreme case that deals with $k_{\max}$ large sets (from 45,000 to 50,000) which are slightly overlapped. The second case can be regarded as a normal case that handles two large sets and $k_{\max} - 2$ small sets whose sizes randomly distribute between 0 and 5,000.

The simulation results of the extreme case are shown in Table 3, and the results of the normal case is in Table 4. In both tables, our M-JREP protocol performs well, because its probability of bounding absolute estimation error with θ is always above $1 - \delta = 95\%$. In contrast, the accuracy of INC-EXC is non-satisfactory for the normal case in Table 4, and severely degrades when handling the extreme case in Table 3.

Table 3: Accuracy when Handling $k_{\max}$ Large Sets

Number of Sets $k_{\max}$	2	4	6	8
Bounding Probability of M-JREP	95%	95%	95%	95%
Bounding Probability of INC-EXC	7.8%	9.6%	18.4%	28.8%

Table 4: Accuracy Comparison when Handling Two Large Tag Sets and $k_{\max} - 2$ Small Tag Sets

Number of Sets $k_{\max}$	2	4	6	8
Bounding Probability of M-JREP	95%	96%	96.6%	99.6%
Bounding Probability of INC-EXC	7.8%	50.4%	86.6%	99%

9. RELATED WORK

Much existing RFID work concentrates on how to collect efficiently the IDs of a group of tags, which is called *tag identification*. Since the tags communicate with a reader through wireless medium, inevitably collisions will happen when multiple tags respond to the same reader simultaneously. Collision arbitration protocols mainly fall into two categories, i.e., tree-based protocols [13], and slotted ALOHA protocols [6]. The de-facto RFID standard, EPCglobal C1G2, is a variant of the slotted ALOHA protocol [1].

Another branch of RFID research investigates how to accurately estimate the cardinality of a tag set at low time cost without any ID collection. To minimize the time cost, a plethora of protocols have been developed, including unified probabilistic estimator [5], lottery frame protocol [14], generalized maximum likelihood estimation [7], first non-empty slot based estimation [4], probabilistic estimating tree [20], average run based tag estimation [15], and zero-one estimator [19]. An important recent study has proposed a two-phase protocol named SRC_s [3], which uses the first phase to swiftly make a rough estimation of the tag cardinality, and the second phase based on ALOHA frame for achieving better accuracy. Our M-JREP also adopts a two-phase protocol for efficiently encoding a tag set into a bitmap.

A recent trend is to extend the tag counting problem from a single set to multiple sets. Some researchers focus on two tag sets scanned by a reader at different time points, and estimate the cardinalities of their intersection/differences [9–11, 16, 18]. Such information can help detect missing tags (which exist in the previous tag set, but no longer in the current set), remaining tags (existing in both sets), and new tags (opposite to missing tags). Another work named CCF is able to estimate the cardinality of an arbitrary set expression [8]. It assumes that each tag set is encoded into a sketch named *k-min hash values* [2] and the configured value of k must be the same for the sketches of all tag sets.

However, the aforementioned previous studies on multiple-set counting problem are limited from three perspectives. First, most of them are not designed to handle a general set expression, except the work in [8]. Second, all of them specifies the accuracy requirement by the relative error model. Unfortunately, when the quantity to estimate approaches zero, their time cost to meet the accuracy requirement skyrockets to infinity (see Section 2 for detailed discussion). The correct choice is to use instead the absolute error model. Third, previous work requires that all tag sets must be compressed into data structures (called snapshots) with the same length, such that multiple snapshots can be merged easily to estimate the union of multiple sets [8, 9, 16, 18]. However, these snapshots may not have an equal length, especially when the tag sets they encode dramatically differ in sizes, which is commonly seen in real-world scenarios.

A very recent work [17] addresses this third problem by allowing the bitmap-based snapshots of tag sets to have adaptively different lengths, in order to improve the protocol time efficiency. But it is still inadequate in that it only deals with the joint cardinality estimation of two tag sets. In many applications, it is required to estimate the cardinality of a general set expression that may involve an arbitrary number of tag sets. Our paper can solve this problem efficiently.

10. CONCLUSION

In this paper, we have formulated a problem called *joint property estimation*, in which the cardinality of an arbitrary set expression (involving multiple tag sets from different spatial or temporal domains) is estimated with bounded absolute error. We propose a protocol named M-JREP with a novel design that allows multiple tag sets to be encoded into bitmaps with varied lengths. It provides a new method called *expanded OR* to combine the multiple bitmaps, and it designs formulas to exploit the combined information, estimate the cardinalities of all elementary subsets, and finally calculate the cardinality of the desired set expression. We have analyzed the bias and variance of M-JREP, and also the optimal setting of its protocol parameters under predefined accuracy requirements. We have performed extensive simulation studies. The results show that our protocol can reduce the execution time by multiple folds as compared with INC-EXC and CCF protocols, which require all tag sets must be encoded into length-consistent snapshots.

Acknowledgment

This work was supported in part by Natural Science Foundation of United States under grant CNS-1409797, China National High Technology Research and Development Program under grant 2013AA013503, National Natural Science Foundation of China under grants 61472385, 61502098, 61502100, 61532013, and 61320106007, by Jiangsu Provincial Natural Science Foundation of China under grants BK20150629 and BK20150637, by Jiangsu Provincial Key Laboratory of Network and Information Security under grant BM2003201, by Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under grant 93K-9. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

11. REFERENCES

[1] EPC^{TM} radio-frequency identity protocols generation-2 UHF RFID protocol for communications at 860MHz – 960MHz v2.0.0, 2014.

[2] Z. Bar-Yossef, T. S. Jayram, R. Kumar, D. Sivakumar, and L. Trevisan. Counting distinct elements in a data stream. *Proc. of RANDOM*, 2002.

[3] B. Chen, Z. Zhou, and H. Yu. Understanding RFID counting protocols. *Proc. of ACM MOBICOM*, 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. *MobiQuitous*, 2005.

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

[8] H. Liu, W. Gong, L. Chen, W. He, K. Liu, and Y. Liu. Generic composite counting in RFID systems. *Proc. of IEEE ICDCS*, 2014.

[9] H. Liu, W. Gong, X. Miao, K. Liu, and W. He. Towards adaptive continuous scanning in large-scale RFID systems. *Proc. of IEEE INFOCOM*, 2014.

[10] X. Liu, B. Xiao, S. Zhang, and K. Bu. Unknown tag identification in large RFID systems: An efficient and complete solution. *IEEE Transactions on Parallel and Distributed Systems*, 26(6):1775–1788, 2015.

[11] X. Liu, S. Zhang, B. Xiao, and K. Bu. Flexible and time-efficient tag scanning with handheld readers. *IEEE Transactions on Mobile Computing*, 15(4):840–852, 2016.

[12] ——. Extended version online of the submited paper. https://www.dropbox.com/s/bac8ogtyjgs85sp/mobihoc16-rfid.pdf, 2016.

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

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

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

[16] C. Tan, B. Sheng, and Q. Li. How to monitor for missing RFID tags. *Proc. of IEEE ICDCS*, 2008.

[17] Q. Xiao, S. Chen, M. Chen, and Y. Zhou. Temporally or spatially dispersed joint RFID estimation using snapshots of variable lengths. *Proc. of ACM MOBIHOC*, 2015.

[18] Q. Xiao, B. Xiao, and S. Chen. Differential estimation in dynamic RFID systems. *Proc. of IEEE INFOCOM (mini-conference)*, 2013.

[19] Y. Zheng and M. Li. ZOE: Fast cardinality estimation for large-scale RFID systems. *Proc. of IEEE INFOCOM*, 2013.

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

APPENDIX

A. PROOF OF PROPERTY 1

We firstly define a few notations. In the binary format of x, the series of one-bits from low end to high is at positions

$$\ell(x,1),\ \ell(x,2),\ \ldots,\ \ell(x,bc(x)), \tag{37}$$

where $\ell(x,i)$ is the location of the ith one-bit in x, and $bc(x)$ is the number of one-bits in x (or called the bit count of x). For simplicity, we denote

- the location of the lowest-order 1-bit $\ell(x,1)$ by $lo(x)$, and
- the location of the highest-order 1-bit $\ell(x,bc(x))$ by $hi(x)$.

By the definition of elementary subset N_x in (3), the one-bits in binary format of x decide which tag sets will include N_x, i.e., $S_{\ell(x,1)}, S_{\ell(x,2)}, \ldots, S_{\ell(x,bc(x))}$. Since the tag set S_i are encoded by the ALOHA frames B_i, the one-bits in x also decide which frames may receive responses from tags in N_x.

$$B_{\ell(x,1)},\ B_{\ell(x,2)},\ \ldots,\ B_{\ell(x,bc(x))} \tag{38}$$

For this list of frames, the following properties establishes.

Property 3. *For an arbitrary tag from the elementary subset N_x, it may respond in (or be encoded by) the frames $B_{\ell(x,1)}, B_{\ell(x,2)}, \ldots, B_{\ell(x,bc(x))}$. The tag will be sampled to respond either in all these frames or in none of them, since the sampling process is performed in a pseudorandom fashion.*

Property 4. *Assume a tag in N_x is sampled to respond, and its list of encoding frames in (38) includes $B_{\ell(x,i)}$ and $B_{\ell(x,i')}$, where the former frame is no longer than the latter $m_{\ell(x,i)} \leq m_{\ell(x,i')}$. For an arbitrary slot number j, if the tag does not pick the $(j \bmod m_{\ell(x,i)})$th slot in frame $B_{\ell(x,i)}$, it will neither select the $(j \bmod m_{\ell(x,i')})$th slot in $B_{\ell(x,i')}$.*

Proof. Suppose a tag id is sampled and does not select the $(j \bmod m_{\ell(x,i)})$th slot in frame $B_{\ell(x,i)}$, i.e., $H(id \oplus R) \neq j \bmod m_{\ell(x,i)}$. Since both $m_{\ell(x,i)}$ and $m_{\ell(x,i')}$ are the powers of two and $m_{\ell(x,i)} \leq m_{\ell(x,i')}$, the former is able to divide the latter. Thus, $H(id \oplus R) \neq j \mod m_{\ell(x,i')}$, implying that the jth slot in $B_{\ell(x,i')}$ is not selected by the tag. □

Property 5. *Among the list of frames in (38), if in the first frame $B_{\ell(x,1)}$, a tag in subset N_x does not select the $(j \bmod m_{\ell(x,1)})$th slot, then in any subsequent frame $B_{\ell(x,i)}$ with $i > 1$, the tag neither selects the $(j \bmod m_{\ell(x,i)})$th slot.*

Consider an arbitrary tag id in elementary subset N_x. As mentioned in (38), the frames that may receive responses of the tag id are $B_{\ell(x,1)}, B_{\ell(x,2)}, \ldots, B_{\ell(x,bc(x))}$. Further consider the bitmap OR_y, which is the expanded OR of bitmaps $B_{\ell(y,1)}, B_{\ell(y,2)}, \ldots, B_{\ell(y,bc(y))}$ as defined in equation (9). Among these selected bitmaps as marked by y, the bitmaps that may receive the response of tag id in the subset N_x are

$$B_{\ell(x\wedge y,1)}, B_{\ell(x\wedge y,2)}, \ldots, B_{\ell(x\wedge y,i)}, \ldots, B_{\ell(x\wedge y,bc(x\wedge y))}. \tag{39}$$

According to (9), the jth bit of bitmap OR_y is the OR of the $(j \bmod m_{\ell(y,i)})$th bit in bitmap $B_{\ell(y,i)}$ with the index i ranging from 1 to $bc(y)$. But the tag in subset N_x only appears in the list of bitmaps in (39). Hence, we only need to analyze the probability for the tag to assign the $(j \bmod m_{\ell(x\wedge y,i)})$th bit in bitmap $B_{\ell(x\wedge y,i)}$ with $i \in [1, bc(x \wedge y)]$.

As explained in Property 5, if the tag id does not select the $(j \bmod m_{\ell(x\wedge y,1)})$th slot in $B_{\ell(x\wedge y,1)}$, then it neither selects the $(j \bmod m_{\ell_{x\wedge y}(i)})$th slot in $B_{\ell(x\wedge y,i)}$ for any i value. Hence, the probability that the tag id picks the jth slot in OR_y equals the probability that the tag assigns the $(j \bmod m_{\ell(x\wedge y,1)})$th bit in $B_{\ell(x\wedge y,1)}$, i.e., $\frac{p}{m_{\ell_{x\wedge y}(1)}} = \frac{p}{m_{lo(x\wedge y)}}$, where $lo(x\wedge y)$ and $\ell(x\wedge y,1)$ are the location of lowest 1-bit in $x\wedge y$.