Fast Tracking the Population of Key Tags in Large-scale Anonymous RFID Systems

Xiulong Liu Xin Xie Keqiu Li Bin Xiao Jie Wu Heng Qi Dawei Lu

***Abstract*—In large-scale RFID-enabled applications, we sometimes only pay attention to a small set of key tags, instead of all. This paper studies the problem of key tag population tracking, which aims at estimating how many key tags in a given set exist in the current RFID system and how many of them are absent. Previous work is slow to solve this problem due to the serious interference replies from a large number of ordinary (i.e., non-key) tags. However, time-efficiency is a crucial metric to the studied key tag tracking problem. In this paper, we propose a singleton slot-based estimator, which is time-efficient because the RFID reader only needs to observe the status change of expected singleton slots corresponding to key tags instead of the whole time frame. In practice, the ratio of key tags to all current tags is small because "key" members are usually rare. As a result, even when the whole time frame is long, the number of expected singleton slots is limited and the running of our protocol is very fast. To obtain good scalability in large-scale RFID systems, we exploit the sampling idea in the estimation process. Rigorous theoretical analysis shows that the proposed protocol can provide guaranteed estimation accuracy to end users. Extensive simulation results demonstrate that our scheme outperforms the prior protocols by significantly reducing the time cost.**

***Index Terms*—Key RFID Tags, Cardinality Estimation, Population Tracking, Time-efficiency.**

I. Introduction

A. Background and Problem Formulation

RADIO Frequency Identification (RFID) is a form of wireless technology that can identify and track tags attached to objects or even humans. Compared with traditional bar-code technology, RFID has a variety of advantages: RFID readers do not require a direct line of sight to probe tags; RFID tags can be read at a relatively long distance; RFID readers can read tags at a very fast speed of nearly 100 tags per second. Owning to these attractive properties, RFID technology has promising prospects in various applications such as supply chain management [2], access control [3], localization [4], and object tracking [5], etc.

In a large-scale RFID system containing thousands of tags, we may only care about a small subset of tags instead of all tags. For example, consider a multi-tenant warehouse as illustrated in Fig. 1, where each tenant possesses a set of items, and only knows the tag IDs of his own items, but does not know the tag IDs of other tenants [6]. Tenant A may want to monitor the stock of his items (*i.e.*, how many of his items are present in the system), so as to dynamically determine the stock replenishment. As multiple tenants share a common RFID reader, to avoid disturbing other tenant's operations, each tenant just has a short time window to do such an inventorying operation. A straightforward solution is to use the RFID reader to perform a comprehensive and precise tag identification operation. Clearly, identifying each tag in the warehouse can obviously tell us how many items of tenant A are present in the warehouse. However, the operation of precise tag identification is critically time-consuming, because it needs to identify each tag one by one in the large tag population. Moreover, what tenant A wants to know is just the number of his items that are present in (or absent from) the warehouse. For such a purpose of stock monitoring, exact identification is even not necessary, because the tagged objects are of the same except for the tag IDs. Hence, for time-efficiency, tenant A may prefer an approximate estimation protocol to a time-consuming tag identification protocol. Tenant A refers to his tags as key tags, and the tags belonging to other tenants as ordinary tags, and desires to quickly estimate the number of present key tags and the number of absent key tags. When the estimated number of remaining key tags are below a certain threshold, the tenant performs a stock replenishment.

This problem is formulated as follows. We use $S_K = \{x_1, x_2, ..., x_k\}$ represent the set of k key tags that belong to a user, which is known by this user in advance. We denote the current set of tags in the system as $S_C = \{y_1, y_2, ..., y_c\}$, which is not known in advance because it is not easy to get particularly in dynamic RFID systems (the tagged objects

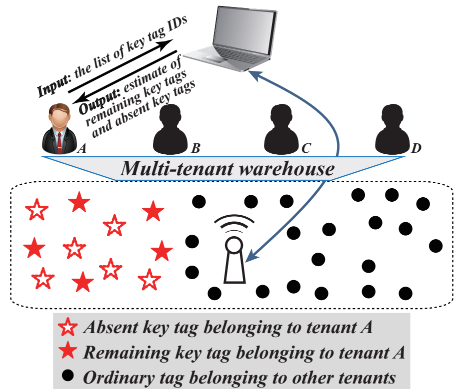


Fig. 1. Exemplifying the problem of key tag population tracking.

Xiulong Liu, Xin Xie, Heng Qi and Keqiu Li are with the School of Computer Science and Technology, Dalian University of Technology, China. (e-mail: {xiulongliudut,xiexindut}@gmail.com; {hengqi,keqiu}@dlut.edu.cn)

Bin Xiao is with Department of Computing, The Hong Kong Polytechnic University, Hong Kong. (e-mail: csbxiao@comp.polyu.edu.hk)

Jie Wu is with Department of Computer and Information Sciences, Temple University, USA. (e-mail: jiewu@temple.edu)

Dawei Lu is with the School of Mathematical Sciences, Dalian University of Technology, China. (e-mail: dwlu@dlut.edu.cn)

Corresponding Authors: Heng Qi and Keqiu Li

A preliminary version [1] of the paper has appeared in IEEE ICNP 2014.

or humans frequently move in or out). Generally, there are three types of tags. (1) *Absent key tag set* S_A: the tags that belong to the tenant but are not present in the current system. $S_A = S_K - S_C$ and its cardinality $|S_A|$ is denoted as a. (2) *Remaining key tag set* S_R: the present tags that belong to this tenant. $S_R = S_K \cap S_C$ and $|S_R|$ is denoted as r. (3) *Ordinary tag set* S_O: the present tags that belong to other tenants. $S_O = S_C - S_K$ and $|S_O|$ is denoted as o. We use S_U to represent the universal tag set. $S_U = S_K \cup S_C$ and $|S_U|$ is denoted as u. We use a factor d called dynamic degree to indicate the ratio of absent key tags to remaining key tags, *i.e.*, $d = \frac{a}{r}$. The relationship of the set sizes are as below. $k = a + r$, $u = a + r + o$, and a, r, o are natural numbers.

The key tag population tracking problem is to fast report an estimate $\hat{r}$ for r and an estimate $\hat{a}$ for a. The reported $\hat{r}$ and $\hat{a}$ should satisfy $P\{|\hat{r} - r| \leq \alpha \times r\} \geq \beta$ and $P\{|\hat{a} - a| \leq \alpha \times a\} \geq \beta$, respectively. Here, $\alpha \in (0, 1]$ is the confidence interval and $\beta \in [0, 1)$ is the required reliability. (α, β) reflects the required estimation accuracy, and is specified by the end users. For simplicity, we refer to $\hat{r}$ (or $\hat{a}$) as an (α, β) estimate of r (or a).

Someone may think that the estimation of remaining key tags and the estimation of absent key tags are two equivalent problems, because we can directly get $\hat{a}$ by calculating $k - \hat{r}$, where k is known in advance. In other words, we only need to estimate the cardinality of either r or a. However, this notion is incorrect. The reasons are as exemplified in the following. We first get an estimate $\hat{r}$ with a relative error of α, thus its absolute error is $r \times \alpha$. If we get $\hat{a}$ by directly calculating $k - \hat{r}$, obviously, the absolute error of $\hat{a}$ is also $r \times \alpha$. Then, the relative error is $\frac{r \times \alpha}{a}$, which is larger than the required value α when $r > a$. Therefore, these two types of estimation problems are not equivalent, and it is not trivial to study the estimation of r and a, respectively.

B. Prior Art and Limitation

In the following, we review the closely related work and point out their limitations when addressing the problem of key tag population tracking.

1) Tag Identification Protocols: A straightforward solution is to identify all tags in S_C. Once we know S_C, we can get the precise r by comparing S_K and S_C. A great deal of excellent tag identification protocols [7], [8] have been proposed. However, the time cost of tag identification protocols is inherently proportional to the number of tags. In an RFID-enabled warehouse that contains tens of thousands of tags, the identification time could be very long, and thus may affect the running of other RFID protocols. Hence, we sometimes prefer a fast probabilistic method even though it sacrifices some accuracy. For example, in inventory management, we only need to know the approximate number of remaining items instead of the exact number to determine the item replenishment. Moreover, tag identification protocols may cause the privacy concern, because the transmitted tag IDs as plaintext in the air may be eavesdropped by the malicious reader [9].

2) Missing Tag Identification Protocols: The existing missing tag identification schemes were proposed with a common assumption that all the tag IDs are known in advance. However, this paper considers an RFID system that also contains ordinary tags whose IDs are not known previously. The existing missing tag identification protocols cannot solely address the problem concerned by this paper, because the reader will incorrectly assert that a missing tag is present, if ordinary tag(s) respond exactly at the moment when this missing tag should respond. A solution is to execute the unknown tag identification protocols to deactivate the unknown ordinary tags before invoking the missing tag identification protocols. However, we observed from the simulation results that this solution is still time-consuming.

3) Tag Search Protocols: The literature [5], [10] exploited lightweight Bloom filter technique to search the exact tags in S_R. Although the methods in [5], [10] have got great improvement over the pure identification protocols, they are still of low time-efficiency to solve the problem of key tag population tracking. The intuitive reason is that searching exact tags consumes more time than just reporting tag cardinality.

4) Cardinality Estimation Protocols: The main technical challenge to key tag population tracking is that serious interference replies from a large number of ordinary (non-key) tags will affect the estimation of key tag cardinality. However, most existing estimation protocols [9], [11]–[20] cannot distinguish the key members from the ordinary members. They can only estimate the cardinality of S_C, which is the sum of the numbers of remaining key tags and ordinary tags. The problem formulated in literature [21], which focuses on dynamic RFID estimation, is fundamentally the same as this paper. However, the ZDE protocol proposed in [21] is not efficient to solve the problem of key tag population tracking because it requires the reader to observe the whole time frame even when the number of key tags is small.

C. The Proposed Approach

We first propose a *Basic Key tag Tracking (B-KT)* protocol based on the framed slotted aloha mechanism. The reader queries the tags by issuing a time frame that contains f slots. Each tag in S_C randomly chooses a slot to respond with the checksum of its ID. There are three types of slots in the frame: empty slot, in which no tag responds; singleton slot, in which only one tag responds; collision slot, in which two or more tags respond. Then, we can get a vector $\mathbb{C}[0..f-1]$, in which 0 represents empty; 1 represents singleton; 2 represents collision. Similarly, we can get another vector $\mathbb{K}[0..f-1]$ by virtually executing aloha protocol on the key tag set S_K. We compare these two vectors to estimate the number of remaining key tags and the number of absent key tags. Let N_{10} represent the number of slots that satisfy $\mathbb{K}[\cdot] = 1 \bigwedge \mathbb{C}[\cdot] = 0$; N_{11} represent the number of slots that satisfy $\mathbb{K}[\cdot] = 1 \bigwedge \mathbb{C}[\cdot] = 1$ and the received checksum is the same as that of the expected tag. Intuitively, the more absent key tags are, the larger the value N_{10} is; the more remaining key tags are, the larger the value N_{11} is. This paper proposes the unbiased estimators using the observed values N_{10} and N_{11} to track key tag population, *i.e.*, estimating the number of absent key tags and the number of remaining key tags. Then, we use the sampling idea to improve the scalability of B-KT, and thus propose the *Sampling-based Key tag Tracking (S-KT)* protocol.

D. Main Contributions

This paper thoroughly studies the problem of key tag population tracking and mainly makes the following contributions.

• We first propose the *Basic Key tag Tracking (B-KT)* protocol. The advantage of B-KT over prior art is that the reader only needs to observe the expected singleton slots corresponding to key tags instead of the whole time frame. The expected empty/collision slots are not used, and are directly skipped for saving time. To be more scalable, we exploit the sampling idea on B-KT to propose the *Sampling-based Key tag Tracking (S-KT)* protocol.

• We theoretically investigate the accuracy of the proposed estimators. Rigorous analysis is proposed to ensure that our estimators can always satisfy the accuracy requirement specified by the users. The involved parameters are also optimized to minimize the execution time required by our protocols.

• We conduct extensive simulations to evaluate the performance of S-KT in a large-scale RFID system. Simulation results show that S-KT significantly outperforms the state-of-the art protocols in terms of time-efficiency.

The remainder of this paper is organized as follows. We propose B-KT, S-KT, and the theoretical analysis in Sections II and III, respectively. In Section IV, extensive simulation experiments are conducted to evaluate the performance of the proposed protocol. A comprehensive review of the related work is given in Section V. Section VI concludes the paper.

II. Basic Key tag Tracking Protocol

In this section, we first present the MAC layer communication mechanism used by our B-KT protocol. Then, we propose the novel singleton slot-based estimator $\hat{r}$ to estimate the cardinality of *remaining* key tags. To guarantee the required (α, β) accuracy, we propose rigorous theoretical analysis to configure the involved parameters. On the other hand, we also propose the estimator $\hat{a}$ to estimate the cardinality of *absent* key tags. Corresponding theoretical analysis is presented to guarantee the accuracy of $\hat{a}$. Finally, we analyze the time-efficiency of our B-KT protocol. The main notations used in this paper are summarized in Table I.

TABLE I
Main notations used in the paper

Notations	Descriptions
S_A / S_R / S_O	abs. key tag set / rem. key tag set / ord. tag set.
a / r / o	cardinality of S_A / S_R / S_O.
S_K / S_C / S_U	key / current / universal tag set.
k / c / u	cardinality of S_K / S_C / S_U.
u_{max}	upper bound on u.
d	dynamic degree, given by $\frac{a}{r}$.
d_{max} / d_{min}	upper / lower bound on d.
$\hat{r}$ / $\hat{a}$	estimate of r / a.
α / β	required confidence interval / required reliability.
f	size of sub-frame.
$\aleph$	# of sub-frames.
fc	frame counter.
sc	slot counter.
$H(\cdot)$	uniform hashing function.
R	random number.
$E(\cdot)$ / $D(\cdot)$	expectation / variance.
$\mathbb{K}[\cdot]$	vector got by virtually executing Aloha on S_K.
$\mathbb{C}[\cdot]$	vector got by executing Aloha on S_C.
$\hat{u}_i$	estimate of u after the i^{th} frame.
u_{max_i}	tighter upper bound on u after the i^{th} frame.
$\hat{d}_i$	estimate of d after the i^{th} frame.
d_{max_i} / d_{min_i}	tighter upper / lower bound on d after the i^{th} frame.
Z_β	percentile of β.

A. Overview of MAC Layer Communication

The MAC layer communication mechanism used by our B-KT protocol is a variant of the Framed Slotted Aloha mechanism. Because of the instability of long-frame Aloha mechanism [8], [9], [22], the frame size is typically no more than 512. For simplicity, f is fixed to 512 throughout this paper. However, when a large number of tags participate in the short frame of 512 slots, almost all slots become collisions. Such a frame full of collisions is usually useless. To solve this dilemma, we propose to sequentially initializes $\aleph$ time frames to "load" a large number of tags, where each frame still contains $f = 512$ slots.

Specifically, the reader initializes an arbitrary frame with *frame counter* $fc \in [0, \aleph - 1]$ by broadcasting a request $\langle fc, \aleph, f, R \rangle$, in which R is a random number. Each tag calculates $H(ID, R) \mod \aleph$ to determine if it will participate in the current frame. If $H(ID, R) \mod \aleph$ is equal to the current *frame counter* fc, it will participate in the current frame. Note that, R does not change among all the sub-frames, and thus each tag will pseudo-randomly determine *one and only one* sub-frame to participate. The tags participating in the current frame will pick the sc^{th} slot, where $sc = H(ID, R) \mod f$. Each tag responds the 10-bit checksum [23] of its ID in the picked slot. Even though $\aleph$ short sub-frames (containing $f = 512$ slots) are issued one by one, they can be logically interpreted as a *long* time frame that consists of $\aleph f$ slots. And all tags participate in such a long logical time frame.

B. Estimating the Remaining Key Tag Population

1) Overview of the Protocol Design: Since we know the key tag set S_K, we could *virtually* execute the above MAC layer mechanism on S_K. As illustrated in Fig. 2, we could get the slot status vector $\mathbb{K}[\cdot]$ corresponding to S_K. As aforementioned, $\aleph$ sub-frames are logically treated as a long frame containing $\aleph f$ slots. A key tag is mapped to the location of $H(ID, R) \mod \aleph f$ whose result follows a uniform distribution within $[0, \aleph f - 1]$, where ID is its 96-bit ID and R is a random number. In $\mathbb{K}[\cdot]$, 0 means no key tag is mapped to this location; 1 indicates only one key tag is mapped to this location; 2 represents two or more key tags are mapped to this location. These three types of slots are called *expected empty slots*, *expected singleton slots*, and *expected collision slots*, respectively. On the other hand, we use the same parameters to *actually* execute the MAC layer mechanism on the current tag set S_C. Thus, we could obtain another slot status vector $\mathbb{C}[\cdot]$. Specifically, the reader

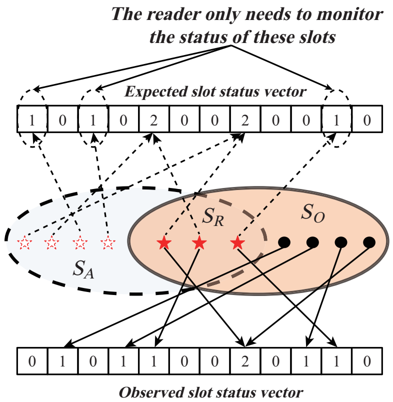


Fig. 2. The basic principle of our singleton slot-based estimators.

sequentially observes the $\aleph$ separate sub-frames (each contains f slots). If the sc^{th} slot in the fc^{th} sub-frame is an empty (singleton or collision) slot, $\mathbb{C}[(fc-1)\times f+sc]$ is set to 0 (1 or 2). Because the current tag set S_C is different from the key tag set S_K in reality, vector $\mathbb{C}[\cdot]$ may be different from $\mathbb{K}[\cdot]$. This paper proposes to use the status change of expected singleton slots to track the population of remaining key tags.

2) Proposing the Estimators $\hat{r}$: In our scheme, the reader needs to observe the expected singleton slots and record the numbers of the following two types of special slot pairs.

- $N_{1,0}$ is the number of slot pairs that satisfy: $\mathbb{K}[z]=1 \wedge \mathbb{C}[z]=0$. Here, z is the slot index in the vector.
- $N_{1,1}$ is the number of slot pairs that satisfy: $\mathbb{K}[z]=1 \wedge \mathbb{C}[z]=1 \wedge$ the received checksum is the same as that of the expected tag.

In the following, we theoretically present how to use the observed $N_{1,0}$ and $N_{1,1}$ to obtain an unbiased estimator for remaining key tags. First, we analyze the probabilistic properties behind the variables $N_{1,0}$ and $N_{1,1}$. Essentially, $N_{1,0}$ is equal to the number of absent tags in S_M that exclusively occupy slots. For an arbitrary absent tag, the probability that its picked slot is not selected by any other tags is denoted as $p_{1,0}$, which can be given as follows:

$$p_{1,0}=\left(1-\frac{1}{\aleph f}\right)^{u-1}\approx e^{-\frac{u}{\aleph f}} \tag{1}$$

Since $\aleph f$ is normally large, the above expression can be simplified to $e^{-\frac{u}{\aleph f}}$. $N_{1,0}$ follows $Bernoulli(m,p_{1,0})$ distribution. And thus, the expectation $E(N_{1,0})$ and variance $D(N_{1,0})$ of $N_{1,0}$ are given as follows:

$$E(N_{1,0})=a\times p_{1,0}=ae^{-\frac{u}{\aleph f}} \tag{2}$$

$$\begin{aligned}D(N_{1,0})&=a\times p_{1,0}\times(1-p_{1,0})\\&=ae^{-\frac{u}{\aleph f}}\left(1-e^{-\frac{u}{\aleph f}}\right)\end{aligned} \tag{3}$$

Then, let us consider the variable $N_{1,1}$. Two and only two cases are possible to contribute to $N_{1,1}$.

Case 1: If a remaining key tag *exclusively* occupies a slot within the frame, $N_{1,1}$ will be increased by 1. We denote the number of this type of slot pairs as $N^*_{1,1}$. For an arbitrary remaining key tag, the probability that it exclusively occupies a slot is denoted as $p^*_{1,1}$, which can be given as follows:

$$p^*_{1,1}=\left(1-\frac{1}{\aleph f}\right)^{u-1}\approx e^{-\frac{u}{\aleph f}} \tag{4}$$

Since $N^*_{1,1}$ follows $Bernoulli(r,p^*_{1,1})$ distribution, the expectation $E(N^*_{1,1})$ and variance $D(N^*_{1,1})$ of $N^*_{1,1}$ are given below:

$$E(N^*_{1,1})=r\times p^*_{1,1}=re^{-\frac{u}{\aleph f}} \tag{5}$$

$$\begin{aligned}D(N^*_{1,1})&=r\times p^*_{1,1}\times(1-p^*_{1,1})\\&=re^{-\frac{u}{\aleph f}}\left(1-e^{-\frac{u}{\aleph f}}\right)\end{aligned} \tag{6}$$

Case 2: If exactly an absent key tag as well as an ordinary tag pick a common slot, and their checksums are coincidentally the same, $N_{1,1}$ will be also increased by 1. And the number of this type of slot pairs is denoted as $N^{**}_{1,1}$. For an arbitrary absent key tag, the probability that it shares a common slot with only one ordinary tag and their 10-bit checksums are the same is denoted as $p^{**}_{1,1}$. We reasonably assume two arbitrary tags have the same 10-bit checksum with the probability $\frac{1}{2^{10}}$. Thus, $p^{**}_{1,1}$ can be given as follows:

$$p^{**}_{1,1}=\binom{o}{1}\times\frac{1}{\aleph f}\times\left(1-\frac{1}{\aleph f}\right)^{u-2}\times\frac{1}{2^{10}}\approx\frac{oe^{-\frac{u}{\aleph f}}}{2^{10}\aleph f} \tag{7}$$

Since $N^{**}_{1,1}$ follows the distribution of $Bernoulli(a,p^{**}_{1,1})$, the expectation $E(N^{**}_{1,1})$ and variance $D(N^{**}_{1,1})$ are given below:

$$E(N^{**}_{1,1})=a\times p^{**}_{1,1}\quad=\frac{aoe^{-\frac{u}{\aleph f}}}{2^{10}\aleph f} \tag{8}$$

$$\begin{aligned}D(N^{**}_{1,1})=&a\times p^{**}_{1,1}\times(1-p^{**}_{1,1})\\=&\frac{aoe^{-\frac{u}{\aleph f}}}{2^{10}\aleph f}\left(1-\frac{oe^{-\frac{u}{\aleph f}}}{2^{10}\aleph f}\right)\end{aligned} \tag{9}$$

Since $N_{1,1}$ consists of two parts: $N^*_{1,1}$ and $N^{**}_{1,1}$, $N_{1,1}=N^*_{1,1}+N^{**}_{1,1}$. The variables $N^*_{1,1}$ and $N^{**}_{1,1}$ are considered to be independent to each other, because $\aleph f$ is very large. Therefore, we have $E(N_{1,1})=E(N^*_{1,1})+E(N^{**}_{1,1})$; and $D(N_{1,1})=D(N^a_{1,1})+D(N^{**}_{1,1})$. Comparing $E(N^*_{1,1})$ in Eq. (5) and $E(N^{**}_{1,1})$ in Eq. (8), we find that $E(N^{**}_{1,1})$ is so minor that it can be ignored. Similarly, compared with $D(N^*_{1,1})$, $D(N^{**}_{1,1})$ is also be negligible. We then have:

$$E(N_{1,1})\approx E(N^*_{1,1})=re^{-\frac{u}{\aleph f}} \tag{10}$$

$$D(N_{1,1})\approx D(N^*_{1,1})=re^{-\frac{u}{\aleph f}}\left(1-e^{-\frac{u}{\aleph f}}\right) \tag{11}$$

According to Eqs. (2) and (10), we have:

$$r=\frac{k}{\frac{E(N_{1,0})}{E(N_{1,1})}+1} \tag{12}$$

By substituting $N_{1,0}$ for $E(N_{1,0})$ and $N_{1,1}$ for $E(N_{1,1})$ in Eq. (12), we get the estimator $\hat{r}$ as follows:

$$\hat{r}=\frac{k}{\frac{N_{1,0}}{N_{1,1}}+1} \tag{13}$$

Eq. (13) infers that we can use the observed values of $N_{1,0}$ and $N_{1,1}$ to approximately calculate the number of remaining tags.

3) Investigating the Accuracy of Estimator $\hat{r}$: Due to probabilistic variance, the reported $\hat{r}$ may differ from the actual value r. It is not trivial to study the accuracy of our estimator. We present the following theorem to rigorously prove that proposed estimator $\hat{r}$ is unbiased. Moreover, its variance is formally given to measure the estimation accuracy.

Theorem 1. *When the frame size $\aleph f$ is large enough, $\hat{r}$ in Eq. (13) is an unbiased estimator of r. That is, $E(\hat{r}) = r$. And the variance of the estimator $\hat{r}$ is as follows:*

$$D(\hat{r}) = \frac{ar}{k}\left(e^{\frac{u}{\aleph f}} - 1\right) \tag{14}$$

Proof: $\hat{r} = \frac{k}{\frac{N_{1,0}}{N_{1,1}}+1}$ in Eq. (13) can be treated as a function of $N_{1,0}$ and $N_{1,1}$. Thus, it is denoted as $g(N_{1,0}, N_{1,1})$. Inspired by [9], we leverage Taylor series expansion [24] to get the expectation and variance of r. In what follows, we present the Taylor series expansion of function $g(N_{1,0}, N_{1,1})$ around (θ_1, θ_2), where $\theta_1 = E(N_{1,0})$, $\theta_2 = E(N_{1,1})$.

$$g(N_{1,0}, N_{1,1}) \approx g(\theta_1, \theta_2) + \left[(N_{1,0}-\theta_1)\frac{\partial g}{\partial N_{1,0}} + (N_{1,1}-\theta_2)\frac{\partial g}{\partial N_{1,1}}\right] + \frac{1}{2}\left[(N_{1,0}-\theta_1)^2\frac{\partial^2 g}{\partial N_{1,0}^2} + 2(N_{1,0}-\theta_1)(N_{1,1}-\theta_2)\frac{\partial^2 g}{\partial N_{1,0}\partial N_{1,1}} + (N_{1,1}-\theta_2)^2\frac{\partial^2 g}{\partial N_{1,1}^2}\right]$$

Taking the expectation of both sides, we have:

$$E[g(N_{1,0}, N_{1,1})] = g(\theta_1, \theta_2) + \frac{1}{2}[D(N_{1,0})\frac{\partial^2 g}{\partial N_{1,0}^2} + D(N_{1,1})\frac{\partial^2 g}{\partial N_{1,1}^2}]$$

In the above equation, $N_{1,0}$ and $N_{1,1}$ are independent to each other when considering $\aleph f$ is large enough. Thus, $Cov(N_{1,0}, N_{1,1})$ is simplified to 0. As required in Eq. (15), the second-order partial derivatives of function $g(N_{1,0}, N_{1,1})$ are as follows.

$$\begin{cases} \dfrac{\partial^2 g(N_{1,0}, N_{1,1})}{\partial N_{1,0}^2} \,|_{N_{1,1}=\theta_2}^{N_{1,0}=\theta_1} = \dfrac{2k\theta_2}{(\theta_1+\theta_2)^3} \\ \dfrac{\partial^2 g(N_{1,0}, N_{1,1})}{\partial N_{1,1}^2} \,|_{N_{1,1}=\theta_2}^{N_{1,0}=\theta_1} = \dfrac{-2k\theta_1}{(\theta_1+\theta_2)^3} \end{cases}$$

Putting the above values into Eq. (15), and replacing θ_1 by $E(N_{1,0})$, θ_2 by $E(N_{1,1})$, we then have:

$$E\left[g(N_{1,0}, N_{1,1})\right] = g[E(N_{1,0}), E(N_{1,1})] + N\left[\frac{E(N_{1,1})D(N_{1,0}) - E(N_{1,0})D(N_{1,1})}{[E(N_{1,0}) + E(N1,1)]^3}\right]$$

Combining the expectations and variances of $N_{1,0}$ and $N_{1,1}$ in Eqs. (2), (3), (10), (11) into the above equation, we have:

$$E(\hat{r}) = E[g(N_{1,0}, N_{1,1})] = g[E(N_{1,0}), E(N_{1,1})] = \frac{k}{\frac{E(N_{1,0})}{E(N_{1,1})}+1} = r$$

The above equation indicates that $\hat{r}$ is an unbiased estimator of r. The variance $D(\hat{r})$ of $\hat{r}$ is calculated as follows:

$$D(\hat{r}) = E[\hat{r} - E(\hat{r})]^2 = E[g(N_{1,0}, N_{1,1}) - r]^2 \tag{15}$$

We use the first-order Taylor series expansion of $g(N_{1,0}, N_{1,1})$ to substitute it in Eq. (15). Thus, we have:

$$D(\hat{r}) = D(N_{1,0})\left(\frac{\partial g}{\partial N_{1,0}}\right)^2 + D(N_{1,1})\left(\frac{\partial g}{\partial N_{1,1}}\right)^2 \tag{16}$$

As required in Eq. (16), the first-order partial derivatives of function $g(N_{1,0}, N_{1,1})$ are calculated as follows.

$$\begin{cases} \dfrac{\partial g(N_{1,0}, N_{1,1})}{\partial N_{1,0}} \,|_{N_{1,1}=\theta_2}^{N_{1,0}=\theta_1} = \dfrac{-k\theta_2}{(\theta_1+\theta_2)^2} \\ \dfrac{\partial g(N_{1,0}, N_{1,1})}{\partial N_{1,1}} \,|_{N_{1,1}=\theta_2}^{N_{1,0}=\theta_1} = \dfrac{k\theta_1}{(\theta_1+\theta_2)^2} \end{cases}$$

Putting the above values into Eq. (16) and replacing θ_1 by $E(N_{1,0})$, θ_2 by $E(N_{1,1})$, we then have:

$$D(\hat{r}) = \frac{k^2\left[E^2(N_{1,1})D(N_{1,0}) + E^2(N_{1,0})D(N_{1,1})\right]}{[E(N_{1,0}) + E(N_{1,1})]^4} \tag{17}$$

Combining the expectations and variances of $N_{1,0}$ and $N_{1,1}$ into Eq. (17), we get the equation in Eq. (14). ■

For large-scale RFID systems, the proposed estimator and its variance obviously hold on. To verify the correctness of the proposed estimator and its variance in small-scale RFID systems, two new sets of simulations are conducted. The simulation results in Fig. 3 show that the theoretical values of the proposed estimator and its variance coincide well with their simulation values with various parameters. Hence, the theoretical analysis proposed above still holds on for small-scale RFID systems.

After investigating the expectation and variance of the proposed estimator $\hat{r}$, one may ask: *How many sub-frames are adequate to ensure that B-KT meets the required (α, β) accuracy when estimating the remaining key tag population?.* We propose the following theorem to give the answer.

Theorem 2. *If the number $\aleph$ of sub-frames is not less than $u/[f\ln(\frac{k\alpha^2}{dZ_\beta^2}+1)]$, the estimation result $\hat{r}$ will meet the predefined accuracy (α, β), that is, $P\{|\hat{r} - r| \le \alpha \cdot r\} \ge \beta$.*

Proof: According to the central limit theorem [25], we have that $W = \frac{\hat{r}-E(\hat{r})}{\sqrt{D(\hat{r})}}$ satisfies the standard normal distribution. We can find a percentile Z_β of β such that $P\{-Z_\beta \le W \le Z_\beta\} \ge \beta$. For example, if $\beta = 95\%$ then $Z_\beta = 1.96$. The required estimation accuracy can be rewritten as follows:

$$P\{|\hat{r} - r| \le \alpha \cdot r\} = P\{(1-\alpha)r \le \hat{r} \le (1+\alpha)r\} = P\left\{\frac{(1-\alpha)r - E(\hat{r})}{\sqrt{D(\hat{r})}} \le \frac{\hat{r} - E(\hat{r})}{\sqrt{D(\hat{r})}} \le \frac{(1+\alpha)r - E(\hat{r})}{\sqrt{D(\hat{r})}}\right\} \tag{18}$$

According to Eq. (18), if the following inequalities are simultaneously satisfied:

$$\begin{cases} \dfrac{(1-\alpha)r - E(\hat{r})}{\sqrt{D(\hat{r})}} \le -Z_\beta \\ \dfrac{(1+\alpha)r - E(\hat{r})}{\sqrt{D(\hat{r})}} \ge Z_\beta, \end{cases}$$

we can guarantee $P\{|\hat{r} - r| \le \alpha \cdot r\} \ge \beta$ [16]. Substituting $E(\hat{r}) = r$ and $D(\hat{r}) = \frac{ar}{k}(e^{\frac{u}{\aleph f}} - 1)$ into the above inequalities and solving them, we have $\aleph \ge u/[f\ln(\frac{k\alpha^2}{dZ_\beta^2}+1)]$. ■

In Theorem 2, we have presented how to set $\aleph$ thereby providing an (α, β) estimator $\hat{r}$. However, u and d is not known in prior. We observe that the frame number $\aleph$ shown in Theorem 2 is a *monotonically increasing* function with respect to u and d. Therefore, we could use $u = u_{max}$ and $d = d_{max}$

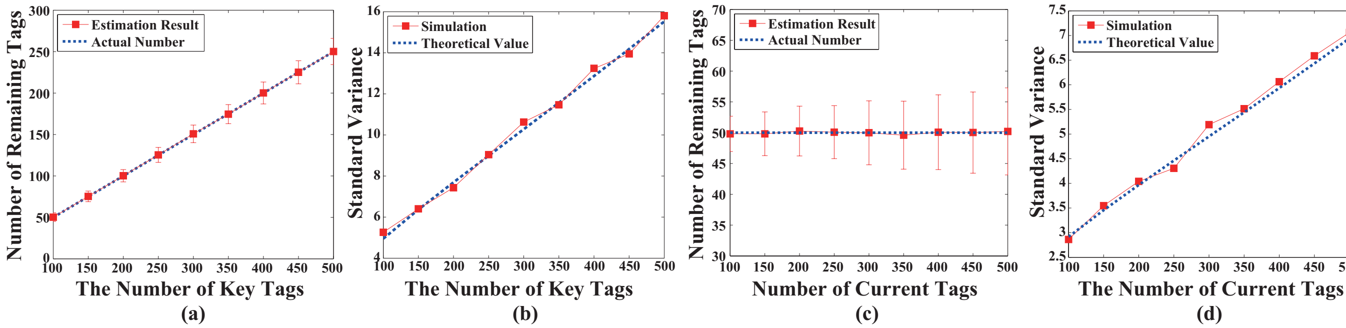


Fig. 3. Verifying the Correctness of Theoretical Analysis in Small-scale RFID Systems. Frame size $f = 512$, sub-frame number $\aleph = 1$, sampling probability $p = 1$. (a) $\hat{r}$ vs. r when varying k. (b) $Var(\hat{r})$: Simulations vs. Theoretical Values when varying k. (c) $\hat{r}$ vs. r when varying c. (d) $Var(\hat{r})$: Simulations vs. Theoretical Values when varying c.

to calculate $\aleph_r$ such that the (α, β) accuracy can be always satisfied for any actual u and d. Here, u_{max}, d_{max}, and d_{min} represent the extreme values of u and d. The extreme values of u and d can be set according to the users' experience. For example, an inventory manager usually knows the approximate popularity of a set of items. Accordingly, he can predict how many items can be sold at most, *i.e.*, the maximum value of a. Then, the max value of d, *i.e.*, d_{max}, can be set to $\frac{a_{max}}{k-a_{max}}$.

C. Estimating the Absent Key Tag Population

In the above section, we have presented how to estimate the remaining key tag population r. Using the knowledge of $k = a + r$, it is easy deduce the estimator $\hat{a}$ for absent key tag population as follows.

$$\hat{a} = k - \hat{r} = \frac{k}{\frac{N_{1,1}}{N_{1,0}} + 1} \tag{19}$$

However, as elaborated by the example in Section I, even if the estimator $\hat{r}$ satisfies the (α, β) accuracy, the deduced estimator $\hat{a}$ may not meet the (α, β) accuracy. Hence, it is not trivial to investigate how to configure the involved parameters to make the estimator $\hat{a}$ satisfy (α, β) accuracy. In the following, we first study the probabilistic properties of $\hat{a}$. In the following theorem, we give the expectation and variance of the absent key tag estimator $\hat{a}$.

Theorem 3. *When the frame size $\aleph f$ is large enough, $\hat{a}$ in Eq. (19) is approximately an unbiased estimator of a. That is, $E(\hat{a}) = a$. And the variance of the estimator a is as follows:*

$$D(\hat{a}) = \frac{ar}{k}\left(e^{\frac{u}{\aleph f}} - 1\right) \tag{20}$$

Proof: Since $\hat{a} = k - \hat{r}$, we have $E(\hat{a}) = E(k - \hat{r}) = k - E(\hat{r})$ and $D(\hat{a}) = D(k - \hat{r}) = D(\hat{r})$. In Theorem 1, we have proved that $E(\hat{r}) = r$ and $D(\hat{r}) = \frac{ar}{k}(e^{\frac{u}{\aleph f}} - 1)$. Therefore, we have $E(\hat{a}) = k - r = a$ and $D(\hat{a}) = \frac{ar}{k}(e^{\frac{u}{\aleph f}} - 1)$. ∎

Again, we should also investigate how to configure the number of sub-frames $\aleph$ to guarantee the required (α, β) accuracy of the new estimator $\hat{a}$ for absent key tag population.

Theorem 4. *If the number $\aleph$ of sub-frames is not less than $u/[f\ln(\frac{kd\alpha^2}{Z_\beta^2} + 1)]$, the estimation result $\hat{a}$ will satisfy the required accuracy (α, β), that is, $P\{|\hat{a} - a| \leq \alpha \cdot a\} \geq \beta$.*

Proof: The proof is similar with Theorem 2. ∎

Theorem 4 presents how to configure $\aleph$ to produce an (α, β) estimate of a. Again, we still need the values of u and d to calculate the minimum frame number, which however are not known in prior. We find that the minimum $\aleph$ for estimating r is a monotonically increasing function against u while a decreasing function against d. Therefore, we initially use $u = u_{max}$ and $d = d_{min}$ to calculate $\aleph$ such that the (α, β) accuracy can be always satisfied for any actual u and d.

D. Filtering out the Expected Empty/Collision Slots

It is easy to find that both of our two estimators, *i.e.*, $\hat{r}$ in Eq. (13) and $\hat{a}$ in Eq. (19), only require the reader to monitor the status of *expected singleton slots*. And thus, the expected empty slots as well as the expected collision slots are not used at all, and their execution wastes a large amount of time. Exploiting the methods used in [26] [27], the expected empty slots and collision slots can be directly filtered without execution. The reader construct a lightweight bitmap whose length is equal to the sub-frame size. In the bitmap, 1 indicates the expected singleton slots that need to be executed and 0 represents the expected empty/collision slots. The reader broadcasts the constructed bitmap to all tags. Each tag checks whether the picked bit in the received bitmap is 1 or not. If the picked bit is 1, it will respond according to the order of 1 in the bitmap. For example, a tag will respond in the 5^{th} slot if it chooses the 5^{th} 1 in the bitmap. In contrary, a tag will keep silent if it finds the picked bit in the bitmap is 0. As a result, only the expected singleton slots, which account for just a small ratio in the long time frame, are executed.

E. Execution Time of B-KT

In this section, we will investigate the execution time of the proposed B-KT. Generally, the estimation process of B-KT includes three phases: (1) *Transmission of Initial Parameters.* For an arbitrary sub-frame, a tag slot t_{tag}, which can support the transmission of a 96-bit ID, is adequate to broadcast the initialization parameters $\langle fc, \aleph, f, R\rangle$; (2) *Transmission of f-bit Bitmap.* The bitmap is divided into 96-bit segments to to be transmitted in $\lceil\frac{f}{96}\rceil$ tag slots; (3) *Execution of the Expected Singleton Slots.* An arbitrary slot in a sub-frame has

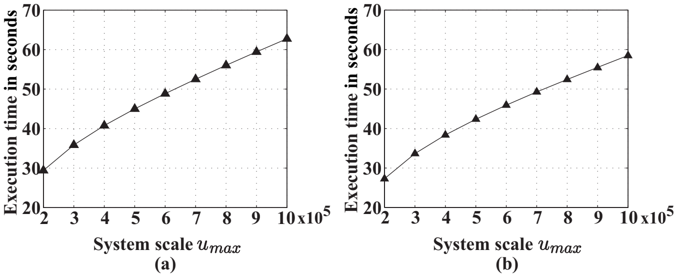


Fig. 4. Execution time of B-KT *vs.* u_{max}: $\alpha = 5\%$, $\beta = 95\%$, and k is fixed to $50,000$; $d \in [\frac{1}{6}, 8]$; u_{max} varies from $200,000$ to $1,000,000$. (a) Estimating r; (b) Estimating a.

the following probability to be an expected singleton slot.

$$p_{1,*} = \binom{k}{1} \times \frac{1}{\aleph} \times \frac{1}{f} \times \left(1 - \frac{1}{\aleph} \times \frac{1}{f}\right)^{k-1} \approx \frac{ke^{-\frac{k}{\aleph f}}}{\aleph f} \tag{21}$$

And then, the number of expected singleton slots that need to be executed in this sub-frame is $f \times p_{1,*} = \frac{k}{\aleph} e^{-\frac{k}{\aleph f}}$. Combining the above three parts of time, the time cost of this sub-frame is $t_{tag} + \lceil \frac{f}{96} \rceil t_{tag} + \frac{k}{\aleph} e^{-\frac{k}{\aleph f}} t_{long}$. Here, t_{long} is the length of the slot that supports the transmission of a 10-bit ID checksum. For $\aleph$ sub-frames in total, the whole execution time of B-KT protocol denoted as T_B is given as follows:

$$T_B = \aleph \times \left(t_{tag} + \left\lceil \frac{f}{96} \right\rceil t_{tag} + \frac{k}{\aleph} e^{-\frac{k}{\aleph f}} t_{long} \right) \tag{22}$$

III. Sampling-based Key tag Tracking Protocol

In this section, we first point out that the scalability of our B-KT protocol is not good enough. The limitation of B-KT motivates us to use the sampling idea [28] [29] to propose an enhanced protocol, named *Sampling-based Key tag Tracking (S-KT)*. We then propose theoretical analysis to guarantee the (α, β) accuracy of our S-KT. The numerical results demonstrate that S-KT performs much better than B-KT. However, time-efficiency of S-KT still has a large room to be improved. Finally, we propose an effective method to bridge the gap between the actual performance of S-KT and its ideal case.

A. Motivation of Using Sampling

One of the most important requirements of a tag estimation scheme is *scalability*, i.e., the estimation time needs to be scalable to large population sizes [9]. However, the numerical results in Fig. 4 reveal that the execution time of B-KT increases sharply with the increase of u_{max}. The underlying reason is that we have to guarantee the ratio of slot pairs $\langle 1, 0 \rangle$ and $\langle 1, 1 \rangle$ in a frame at a certain level to guarantee the estimators meeting (α, β) accuracy. Therefore, the sub-frame number should be significantly large when the universal tag set S_U contains a large number of tags. To achieve a better scalability, we introduce the sampling idea [28] [29] and let the tags participate in the estimation process with a probability p. Then, we can execute our B-KT protocol on small sample tag sets S'_K and S'_C. A tiny sampling method suitable for the RFID devices can be found in [28].

B. Estimating the Remaining Key Tag Population

1) Sampling-based Estimator $\hat{r}'$: We still use the observations of $N_{1,0}$ and $N_{1,1}$ to estimate the remaining key tag population. To differentiate the new analytical procedures from those in the last section, we introduce two new notations but with the same physical meaning as the original ones, $N'_{1,0}$ and $N'_{1,1}$. Similar to the analysis in Section II, the expectations and variances of $N'_{1,0}$ and $N'_{1,1}$ are given as follows:

$$\begin{aligned} &E(N'_{1,0}) = mpe^{-\frac{up}{\aleph f}} \text{ and } D(N'_{1,0}) = m(pe^{-\frac{up}{\aleph f}})(1 - pe^{-\frac{up}{\aleph f}}) \\ &E(N'_{1,1}) = rpe^{-\frac{up}{\aleph f}} \text{ and } D(N'_{1,1}) = r(pe^{-\frac{up}{\aleph f}})(1 - pe^{-\frac{up}{\aleph f}}) \end{aligned} \tag{23}$$

According to Eq. (23), the sampling-based estimator for remaining key tag population can be given as follows:

$$\hat{r}' = \frac{k}{\frac{N'_{1,0}}{N'_{1,1}} + 1} \tag{24}$$

2) Investigating the Accuracy of Estimator $\hat{r}'$: It is not trivial to investigate the probabilistic property of the new sampling-based estimator. Therefore, we propose Theorem 5 to give the expectation and variance of the new estimator $\hat{r}'$.

Theorem 5. *When the frame size $\aleph f$ is large enough, $\hat{r}'$ is an unbiased estimator of r. That is, $E(\hat{r}') = r$. And the variance of the estimator $\hat{r}'$ is as follows:*

$$D(\hat{r}') = \frac{ar}{k} \left(\frac{1}{p} e^{\frac{up}{\aleph f}} - 1 \right) \tag{25}$$

Proof: Using equations in Eqs. (23) and (24), this theorem can be similarly deduced from proof of Theorem 1. ∎

Then, we propose the following Theorem to investigate how many sub-frames are adequate to ensure that S-KT can meet the required (α, β) accuracy when estimating the remaining key tag population.

Theorem 6. *With a fixed sampling probability $p \in (\frac{Z_\beta^2 d}{k\alpha^2 + Z_\beta^2 d}, 1]$, if the number $\aleph$ of sub-frames is not less than $up/\{f \ln[(\frac{k\alpha^2}{Z_\beta^2 d} + 1)p]\}$, the estimation result $\hat{r}'$ will meet the predefined accuracy (α, β), that is, $P\{|\hat{r}' - r| \le \alpha \cdot r\} \ge \beta$.*

Proof: This can be deduced from proof of Theorem 2. Please note that, not all $p \in (0, 1]$ can be used. If p is too small, the denominator $f \ln[(\frac{k\alpha^2}{Z_\beta^2 d} + 1)p]$ will become negative. By solving $\ln[(\frac{k\alpha^2}{Z_\beta^2 d} + 1)p] > 0$, we get the ranges of the sampling probability p as $(\frac{Z_\beta^2 d}{k\alpha^2 + Z_\beta^2 d}, 1]$. ∎

Clearly, the expression of $\aleph$ proposed in Theorem 6 is still an increasing function against u and d. Thus, $\aleph$ should be calculated by $u = u_{max}$ and $d = d_{max}$ so as to accommodate any actual u and d.

C. Estimating the Absent Key Tag Population

1) Sampling-based Estimator $\hat{a}'$: In Section III-B1, we have given the sampling-based estimator for remaining key tag population. Again, using the knowledge of $k = a + r$, we get the sampling-based estimator for absent key tag population below.

$$\hat{a}' = k - \hat{r}' = \frac{k}{\frac{N'_{1,1}}{N'_{1,0}} + 1} \tag{26}$$

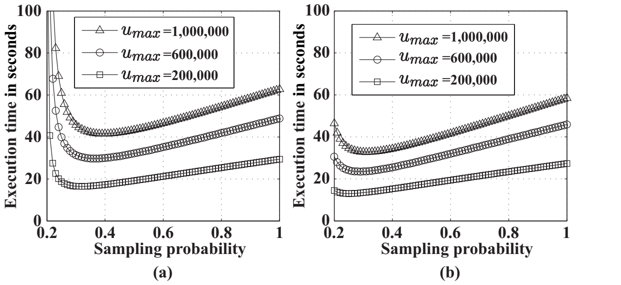


Fig. 5. Execution time of S-KT *vs.* p. $\alpha = 5\%$, $\beta = 95\%$, and k is fixed to $50,000$, $d \in [\frac{1}{6}, 8]$, p varies from 0.2 to 1. (a) Estimating r; (b) Estimating a.

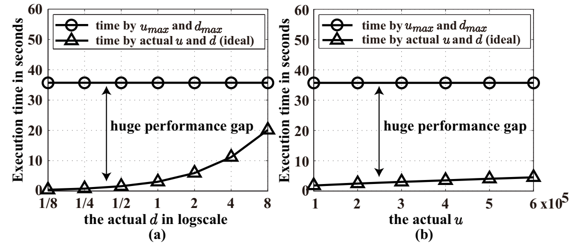


Fig. 6. Performance gap when estimating r: time by u_{max} and d_{max} *vs.* time by actual u and d. $k = 50,000$, $\alpha = 5\%$, $\beta = 95\%$, $u_{max} = 800,000$, $d_{max} = 8$. (a) $u = 300,000$, the actual d varies from $\frac{1}{8}$ to 8 in logscale; (b) $d = 1$, the actual u varies from $100,000$ to $600,000$.

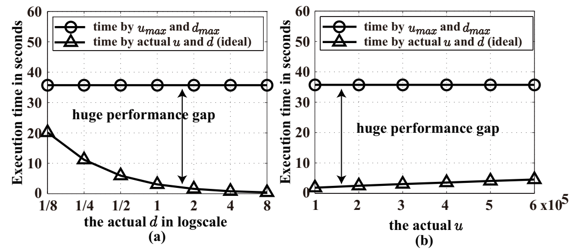


Fig. 7. Performance gap when estimating a: time by u_{max} and d_{min} *vs.* time by actual u and d. $k = 50,000$, $\alpha = 5\%$, $\beta = 95\%$, $u_{max} = 800,000$, $d_{min} = \frac{1}{8}$. (a) $u = 300,000$, the actual d varies from $\frac{1}{8}$ to 8 in logscale; (b) $d = 1$, the actual u varies from $100,000$ to $600,000$.

2) Investigating the Accuracy of Estimator $\hat{a}'$: In the following, we propose Theorem 7 to prove that our sampling-based estimator $\hat{a}'$ is still unbiased. Moreover, we give the estimator variance to quantify the accuracy of $\hat{a}'$.

Theorem 7. *When the frame size $\aleph f$ is large enough, $\hat{a}'$ is an unbiased estimator of a. That is, $E(\hat{a}') = a$. And the variances of the estimator $\hat{a}'$ is as follows:*

$$D(\hat{a}') = \frac{ar}{k}\left(\frac{1}{p}e^{\frac{up}{\aleph f}} - 1\right) \tag{27}$$

Proof: Based on Eqs. (23) and (27), this theorem can be similarly deduced from proof of Theorem 1. ∎

It is also not trivial to study how many frames are adequate to generate an accurate estimator for absent key tag population that satisfies (α, β) accuracy.

Theorem 8. *With a fixed sampling probability $p \in (\frac{Z_\beta^2}{kd\alpha^2 + Z_\beta^2}, 1]$, if the number $\aleph$ of sub-frames is not less than $up/\{f\ln[(\frac{kd\alpha^2}{Z_\beta^2} + 1)p]\}$, the estimation result $\hat{a}'$ will meet the predefined accuracy (α, β), that is, $P\{|\hat{a}' - a| \leq \alpha \cdot a\} \geq \beta$.*

Proof: This proof is similar with that of Theorem 6. ∎

Theorem 8 indicates that the minimum sub-frame number $\aleph$ is an increasing function with respect to u but a decreasing function against d. Hence, $\aleph$ should be calculated by $u = u_{max}$ and $d = d_{min}$ so as to accommodate any actual u and d.

D. Execution Time of S-KT

Similar with the analysis in Section II-E, the whole execution time of S-KT, denoted as T_S, is given as follows:

$$T_S = \aleph t_{tag} + \aleph\left\lceil\frac{f}{96}\right\rceil t_{tag} + kpe^{-\frac{kp}{\aleph f}} t_{long} \tag{28}$$

As illustrated in Fig. 5, the configuration of sampling probability significantly affects the performance of S-KT when either estimating the remaining key tag population or estimating the absent key tag population. We can use an exhaustive searching method to find the optimal sampling probability p, which occurs offline before the running of our S-KT protocol.

E. Bridging the Gap between Actual Situation and Ideal One

1) Motivation: Recall that we have to use the extreme values (i.e., u_{max}, d_{max} or d_{min}) of them to calculate the minimum sub-frame number $\aleph$ because the actual u and d are not known in prior. However, the numerical results illustrated in Figs. 6 and 7 reveal that there is a huge *performance gap* between the actual execution time and the *ideal* execution time. The ideal execution time is to calculate the frame number by using the actual u and d. An interesting question is: *How to make the performance of S-KT approach its ideal case (i.e., assuming u and d are known in prior)?* To answer this question, this section proposes an *early termination* tactic to bridge the performance gap.

At the very beginning, we configure the parameters p and $\aleph$ based on u_{max} and d_{max} when estimating r (or u_{max} and d_{min} when estimating a). After an arbitrary sub-frame $i \in [0, \aleph - 1]$, we leverage the observation of the first $i + 1$ sub-frames that have already been executed to give tighter bounds u_{max_i} on u and d_{max_i} (or d_{min_i}) on d. Based on new u_{max_i} and d_{max_i} (or d_{min_i}), the backend server determines if the current estimation result of r (or a) has already met the required (α, β) accuracy. If so, the reader will terminate the execution right now, otherwise, the next sub-frame will be executed.

2) Giving the Tighter Bounds on u and d: According to Eq. (23), we first leverage the observed $N_{1,0}'^i$ and $N_{1,1}'^i$ after the i^{th} sub-frame to approximate u and d as follows.

$$\hat{u}_i = -\frac{\aleph f}{p}\ln\left(\frac{N_{1,0}'^i + N_{1,1}'^i}{kp_i}\right) \quad \text{and} \quad \hat{d}_i = \frac{N_{1,0}'^i}{N_{1,1}'^i}, \tag{29}$$

where the actual sampling probability p_i is equal to $\frac{(i+1)p}{\aleph}$. Similar with Section II-B3, we get the expectation and vari-

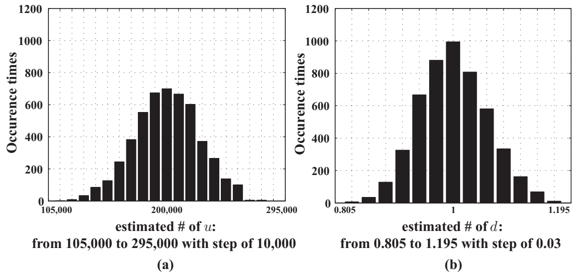


Fig. 8. Investigating the distribution of $\hat{u}_i$ and $\hat{d}_i$. $t = 50,000$, $u = 200,000$, $d = 1$. (a) Recording the estimate $\hat{u}$ after 50^{th} sub-frame with $5,000$ independent trials; (b) Recording the estimate $\hat{d}$ after 50^{th} sub-frame with $5,000$ independent trials.

ance of $\hat{u}_i$ and $\hat{d}_i$ as follows, respectively.

$$
\begin{aligned}
&E(\hat{u}_i) = u \quad \text{and} \quad D(\hat{u}_i) = \frac{\aleph^2 f^2}{kp^2}\left(\frac{e^{\frac{up}{\aleph f}}}{p_i} - 1\right) \\
&E(\hat{d}_i) = d \quad \text{and} \quad D(\hat{d}_i) = \frac{d(d+1)^2}{k} \times \left(\frac{e^{\frac{up}{\aleph f}}}{p_i} - 1\right)
\end{aligned}
\tag{30}
$$

The *three-sigma rule* [30] indicates that: if a variable V follows the normal distribution, then it can differ from its expectation $E(V)$ by a quantity exceeding $3\sqrt{D(V)}$ with a probability no more than 0.3%. The simulation results in Fig. 8 reveal that both $\hat{u}_i$ and $\hat{d}_i$ approximately follow the normal distribution. Hence, we have the following inequalities.

$$
\begin{aligned}
&P[E(\hat{u}_i) - 3\sqrt{D(\hat{u}_i)} < \hat{u}_i < E(\hat{u}_i) + 3\sqrt{D(\hat{u}_i)}] > 99.7\% \\
&P[E(\hat{d}_i) - 3\sqrt{D(\hat{d}_i)} < \hat{d}_i < E(d_i) + 3\sqrt{D(\hat{d}_i)}] > 99.7\%
\end{aligned}
\tag{31}
$$

Then, we can get the new bounds on u and d as follows.

$$
\begin{aligned}
u_{max\_i} &= \hat{u}_i + \frac{3\aleph f}{p}\sqrt{\frac{1}{k}\left(\frac{e^{\frac{\hat{u}_i p}{\aleph f}}}{p_i} - 1\right)} \\
d_{max\_i} &= \hat{d}_i + 3(\hat{d}_i + 1)\sqrt{\frac{\hat{d}_i}{k}\left(\frac{e^{\frac{\hat{u}_i p}{\aleph f}}}{p_i} - 1\right)} \\
d_{min\_i} &= \hat{d}_i - 3(\hat{d}_i + 1)\sqrt{\frac{\hat{d}_i}{t}\left(\frac{e^{\frac{\hat{u}_i p}{k f}}}{p_i} - 1\right)},
\end{aligned}
\tag{32}
$$

where $\hat{u}_i$ and $\hat{d}_i$ are the estimation results got from Eq. (29).

3) Termination Condition When Estimating r: When estimating the remaining key tag population, it is easy to deduce from Theorem 6 that: if the following two conditions are satisfied simultaneously, the estimate $\hat{r}$ can satisfy the required (α, β) accuracy. Then, the estimation process terminates.

$$
\begin{aligned}
&p_i > \frac{Z_\beta^2 d_{max\_i}}{k\alpha^2 + Z_\beta^2 d_{max\_i}} \\
&(i+1) \geq u_{max\_i} p_i \Big/ \left\{ f \ln\left[\left(\frac{k\alpha^2}{Z_\beta^2 d_{max\_i}} + 1\right) p_i\right]\right\}
\end{aligned}
\tag{33}
$$

4) Termination Condition When Estimating a: Similarly, when estimating the absent key tag population, we can deduce from Theorem 8 that: if the following two conditions are satisfied simultaneously, the estimate $\hat{a}$ can satisfy the required (α, β) accuracy. Then, the estimation process terminates.

$$
\begin{aligned}
&p_i > \frac{Z_\beta^2}{k d_{min\_i}\alpha^2 + Z_\beta^2} \\
&(i+1) \geq u_{max\_i} p_i \Big/ \left\{ f \ln\left[\left(\frac{k d_{min\_i}\alpha^2}{Z_\beta^2} + 1\right) p_i\right]\right\}
\end{aligned}
\tag{34}
$$

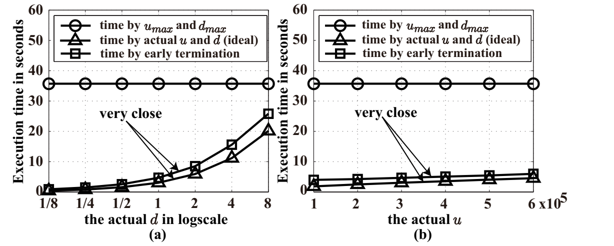


Fig. 9. Performance improvement benefiting from early termination when estimating r: $k = 50,000$, $\alpha = 5\%$, $\beta = 95\%$, $u_{max} = 800,000$, $d_{max} = 8$. (a) $u = 300,000$, the actual d varies from $\frac{1}{8}$ to 8 in logscale; (b) $d = 1$, the actual u varies from $100,000$ to $600,000$.

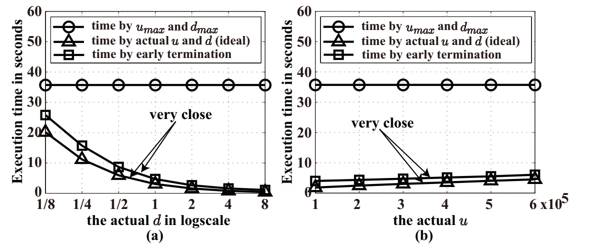


Fig. 10. Performance improvement benefiting from early termination when estimating a: $k = 50,000$, $\alpha = 5\%$, $\beta = 95\%$, $u_{max} = 800,000$, $d_{min} = \frac{1}{8}$. (a) $u = 300,000$, the actual d varies from $\frac{1}{8}$ to 8 in logscale; (b) $d = 1$, the actual u varies from $100,000$ to $600,000$.

The simulation results in Figs. 9 and 10 reveal that the proposed early termination tactic can well bridge the performance gap discussed above, and thus makes the performance of S-KT very close to the ideal case.

5) Discussion on the Early Termination Method: Since the actual values of u and d are unknown in advance, we have to use their extreme values (u_{max}, d_{min}, and d_{max}) to calculate the required number of frames. After several frames, we could obtain tighter bounds on these variables by using 3-sigma rule. For example, $u_{max} = \hat{u} + 3\sqrt{Var(\hat{u})}$ and $d_{min} = \hat{d} - 3\sqrt{Var(\hat{d})}$. Fig. 11(a) infers that, the variances of both $\hat{d}$ and $\hat{u}$ decrease as the number of executed frames increases. Then, the calculated upper/lower bounds on d and u will be closer to their actual values. As a result, the number of frames calculated by the upper/lower bounds on $\hat{u}$ and $\hat{d}$ approaches the ideal frame number, which is shown in Fig. 11(b). When the calculated number of frames is larger than the number of executed frames, the estimation process will be terminated. We use ideal frame number to represent the number of frames that is calculated by the actual values of u and d. Fig. 11(b) reveals that the calculated number of frames is always larger than the ideal frame number because of using the 3-sigma method. Even though the early termination cannot completely bridge the gap between the calculated frame

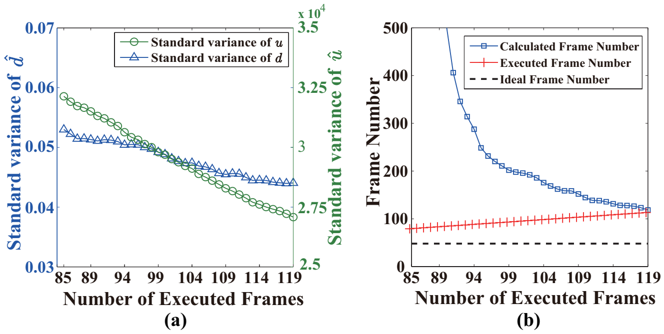


Fig. 11. (a) Variances of $\hat{u}$ and $\hat{d}$ vs. number of executed frames; (b) The number of frames calculated by upper/lower bounds on $\hat{d}$ and $\hat{u}$ vs. the number of executed frames. $k = 50,000$, $\alpha = 5\%$, $\beta = 95\%$, $u = 300,000$ $d = 1$, $u_{max} = 800,000$, $d_{max} = 8$.

number and the ideal frame number, this type of gap has been significantly reduced.

IV. Performance Evaluation

A. Simulation Settings

The simulators were implemented via MATLAB. We first evaluate the time-efficiency of our S-KT scheme by comparing with the state-of-the-art protocol, *i.e.*, ZDE [21], which is a dedicated protocol that aims at estimating the remaining/absent key tags. Besides that, we also compare with two tag search protocols, CATS [5], ITSP [10], and two tag identification protocols, EDFSA [31], Tree Hopping (TH) [8]. Note that, CATS, ITSP, EDFSA, and TH can tell which exactly tags are remaining or absent. It seems not fair to compare S-KT with them. But for a comprehensive comparison with prior schemes, we still use them as the benchmark protocols. In the multi-reader scenarios, the adjacent readers may conflict with each other due to the reader-reader collision [32]. Without loss of generality, like many excellent RFID literature [5], [10], [21], [22], this paper also focuses on the single-reader scenario. The transmission rate between the reader and tags is asymmetric. The uplink rate is $53Kb/s$ (*i.e.*, it takes $18.8us$ to transmit 1-bit data from a tag to a reader), while the downlink rate is $26.5Kb/s$ (*i.e.*, it takes $37.7us$ to transmit 1-bit data from a reader to a tag). Between any two consecutive data transmissions, there is a waiting time $\tau_w = 302us$ [9]. The ambient interferences, *e.g.*, metal, water, white noise, multipath, do affect the RF communication. But due to space limitation, like many high level journal/conference literature [8], [9], [33], [34], we did not discuss the issue of ambient interferences in this paper. And we will pay more attention to these practical issues in our future work. We conduct experiments to evaluate the actual estimation reliability of S-KT. Each simulation is independently repeated 500 times and we report the average results.

B. Performance When Estimating the Remaining Key Tags

In this section, we conduct simulations to investigate the impact of various parameters, including c, k, d, α, β, on the time-efficiency and actual reliability of S-KT when estimating the number of remaining key tags.

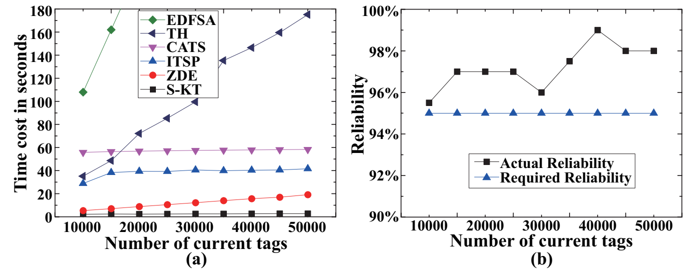


Fig. 12. Impact of c when estimating the number of remaining key tags. (a) Time cost of each protocol vs. c; (b) Actual reliability of S-KT vs. c.

1) Impact of the Number of Current Tag Number c: S-KT is consistently faster than all existing protocols and can satisfy the required reliability with varying number of current tags c. Fig. 12(a)(b) are plotted using $k = 20,000$, $d = 1$, $u_{max} = 100,000$, $d_{min} = 1/8$, $d_{max} = 8$, $\alpha = 5\%$, $\beta = 95\%$, c varies from $10,000$ to $50,000$. We observed from Fig. 12(a) that S-KT has the good scalability against the number of current tags. The underlying reason is presented below. S-KT only needs to execute the singleton slots of key tags, hence, the number of slots that need to be executed in a frame has nothing to do with the number of current tags. However, the ordinary tags indeed interfere with the estimation process of S-KT, which will increase the number of required frames for achieving a certain estimation accuracy (α, β). Hence, the time cost of S-KT slightly increases as the number of current tags increases. The time cost of two tag identification protocols, *i.e.*, EDFSA and TH, increases linearly with respect to c because they inherently need to identify each current tag. The time cost of ZDE also increases significantly against c because it needs to observe not only the slots of key tags but also a huge number of slots of ordinary tags. All in all, S-KT is consistently faster than the other protocols. For example, when $c = 10,000$, the time cost of EDFSA, TH, ITSP, CATS, and ZDE is 108s, 35s, 28.8s, 55.7s, and 5.4s, respectively. While the time cost of our S-KT is just 2.3s, which is 15.2x faster than the fastest tag identification protocol (*i.e.*, TH), 12.5x faster than the fastest tag search protocol (*i.e.*, ITSP), and 2.3x faster than the state-of-the-art ZDE protocol. Note that, the speedup of S-KT over ZDE will be more significant when the number of current tags c increases. Compared with the tag identification protocols, even though our S-KT protocol sacrifices some accuracy due to the probabilistic nature, it achieves nearly 15x speedup. In some scenarios, *e.g.*, inventory management, we may only need to know the approximate number of remaining items to determine the item replenishment. In this case, our probabilistic method is preferred due to its time-efficiency.

Moreover, the results in Fig. 12(b) demonstrate that S-KT can satisfy the required reliability with varying c.

2) Impact of the Number of Key Tag Number k: S-KT is consistently faster than all existing protocols and can satisfy the required reliability with varying number of key tags k. Fig. 13(a)(b) are plotted using $c = 20,000$, $d = 1$, $d_{min} = 1/8$, $d_{max} = 8$, $\alpha = 5\%$, $\beta = 95\%$, k varies from varies from $4,000$ to $20,000$. We made several observations from Fig. 13(a), which are presented as below. The execution time of ITSP and CATS increases linearly with respect to the

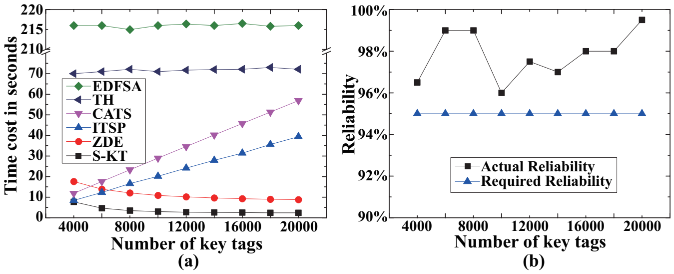


Fig. 13. Impact of k when estimating the number of remaining tags. (a) Time cost of each protocol vs. k; (b) Actual reliability of S-KT vs. k.

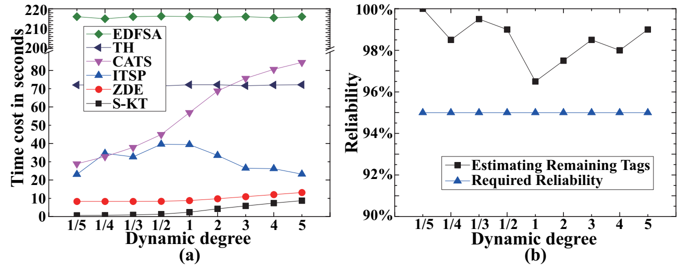


Fig. 14. Impact of d when estimating the number of remaining tags. (a) Time cost of each protocol vs. d; (b) Actual reliability of S-KT vs. d.

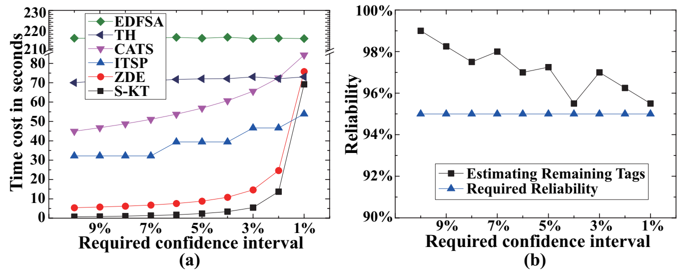


Fig. 15. Impact of α when estimating the number of remaining key tags. (a) Time cost of each protocol; (b) Actual reliability of S-KT.

number of key tags, because they require the bloom filter length to be proportional to the number of key tags for a certain filtering accuracy. The time cost of EDFSA and TH keeps stable with varying number of key tags, because they need to identify each tag in the system, regardless of the number of key tags. Unlike these protocols, the time cost of S-KT and ZDE decreases as the number of key tags increases. The underlying reason is as follows. According to Theorem 6, the number of frames required by S-KT is a monotonically decreasing function of k. The results in Fig. 13(a) reveal that the performance of the tag search protocols (*i.e.*, ITSP and CATS) is close or even better than our S-KT when the number of key tags is quite small (*e.g.*, $4,000$). On the contrary, when the number of key tags is large, our S-KT will significantly outperform all the other protocols. For example, when $k = 20,000$, S-KT runs 30x faster than the fastest tag identification protocol (*i.e.*, TH), 16.4x faster than the fastest tag search protocol (*i.e.*, ITSP), 3.7x faster than the state-of-the-art ZDE protocol. Another important observation from Fig. 13(b) is that S-KT always satisfies the required reliability regardless of the setting of k.

3) Impact of the Dynamic Degree d: S-KT is consistently faster than all existing protocols and can satisfy the required reliability with varying dynamic degree d. Fig. 14(a)(b) are plotted using $k = 20,000$, $c = 20,000$, $d_{max} = 8$, $d_{min} = 1/8$, $u_{max} = 10,0000$, d varies from $1/5$ to 5. We observed from Fig. 14(a) that the execution time of S-KT slightly increases as the value of d increases. The underlying reason is that, according to Theorem 6, the number of frames required by S-KT is a monotonically increasing function of d. The time cost of EDFSA and TH is stable because they still need to identify each tag in current tag set S_C, regardless of the dynamic degree d. Generally, S-KT is consistently faster than the other protocols with varying setting of d. For example, when $d = 1$, S-KT is 30x faster than the fastest tag identification protocol (*i.e.*, TH), 16.4x faster than the fastest tag search protocol (*i.e.*, ITSP), 3.7x faster than the state-of-the-art ZDE protocol. Again, we observed that our S-KT always satisfies the required reliability regardless of the setting of d.

4) Impact of Confidence Interval α and Reliablity β: S-KT significantly outperforms the other protocols when the required estimation accuracy is not very strict, and can always satisfy the required reliability regardless of the settings of α and β. Fig. 15(a)(b) are plotted using $k = 20,000$, $c = 20,000$, $d = 1$, $u_{max} = 10,000$, $d_{min} = 1/8$, $d_{max} = 8$, $\beta = 95\%$, α varies from 10% to 1%. The time cost of tag identification protocols (*i.e.*, EDFSA and TH) keeps stable regardless of the required accuracy, because they need to identify each tag and have 100% accuracy. Except for EDFSA and TH, the time cost of the other protocols increases as the required confidence interval α decreases. The results in Fig. 15(a) reveal that S-KT is sensitive to the required confidence interval α. The underlying reason is that α lies in the $\ln()$ part in the denominator of the number of required sub-frames. When α is very small (*e.g.*, 1%), the execution time of estimation protocols will be very large. Note that, we sometimes do not need such accurate estimation result in practice. The parameter settings of Fig. 16(a)(b) are the same as that of Fig. 15(a)(b) except for that $\alpha = 5\%$ and β varies from 90% to 99%. The execution time of EDFSA and TH is still stable. The time cost of the two tag identification protocols (*i.e.*, ITSP and CATS) is also stable against the required reliability β, because they did not consider the concept of reliability and can only ensure the confidence interval on expectation. The time cost of the two estimation protocols (*i.e.*, S-KT and ZDE) increases slightly as the required reliability β increases. An interesting observation is that the estimation protocols are not quite sensitive to the required reliability β. The underlying reason is as follows. When β varies from 90% to 99%, Z_β (*i.e.*, the percentile of β) just varies from 1.64 to 2.58. Relatively, Z_β does not vary as drastically as α. The number of required frame size $\Theta\{\frac{1}{\ln(\alpha^2/Z_\beta^2)}\}$ will be more sensitive to α than β. The results shown in Fig. 15(b) and Fig. 16(b) demonstrate that S-KT can satisfy any required confidence interval α and reliability β.

5) Variance in Execution Time: We conducted a set of simulations to investigate the variance in the execution time of the proposed S-KT protocol. The PDF and CDF curves in Fig. 17 show that the execution time of S-KT varies from 2.15s to 2.46s with an average value of 2.32s. It approximately

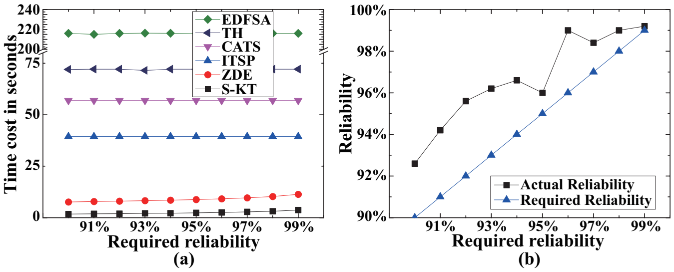


Fig. 16. Impact of β when estimating the number of remaining key tags. (a) Time cost of each protocol; (b) Actual reliability of S-KT.

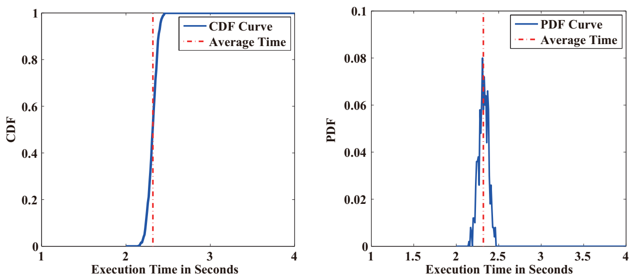


Fig. 17. Evaluating the Variance in The Execution Time of S-KT by CDF and PDF Curves. $k = 200$, $c = 400$, $d = 1$, $(\alpha, \beta) = (5\%, 95\%)$.

follows the normal distribution $Norm(2.32, 0.0033)$, which has a quite small standard variance of $\sqrt{0.0033} = 0.0574$.

C. Performance When Estimating the Absent Key Tags

In this section, we conduct simulations to evaluate the performance of our S-KT when estimating the absent key tag population a. Except for the parameter d, the impact of the other parameters on S-KT when estimating the number of absent key tags is similar with that when estimating the number of remaining key tags. Hence, we can refer to the results shown in Figs. 12(a), Fig. 13(a), Fig. 15(a), and Fig. 16(a) when considering the impact of c, k, α, β on the protocols' time-efficiency, respectively. Fig. 18(a) shows the impact of d on the protocols. Different from Fig. 14(a), the time cost of S-KT decreases as the dynamic degree d increases. The reason is that, according to Theorem 8, the number of required frames is a monotonically decreasing function of d. Here, we also compare with CU [35]+SFMTI [36]. We observed that S-KT is consistently faster than all the other protocols with various settings of d. The results in Figs. 18(b)$\sim$(f) demonstrate that S-KT can satisfy the required reliability with various parameter settings when estimating the number of absent tags.

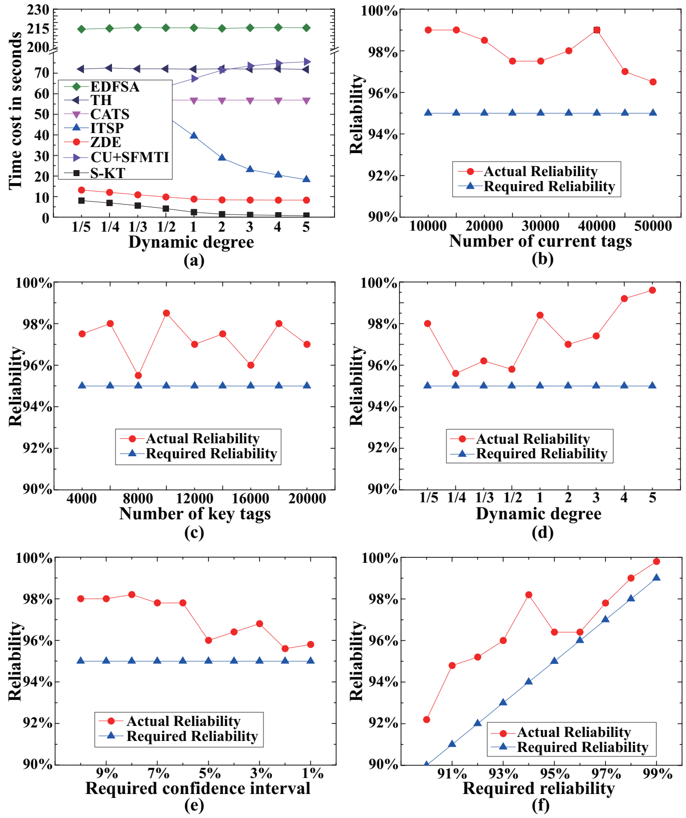


Fig. 18. Performance when estimating the number of absent tags a. (a) Time cost vs. varying dynamic degree; (b$\sim$f) Reliability vs various parameters.

V. Related Work

In RFID-enabled applications, one of the most fundamental tasks is *tag identification* that aims at identifying all the IDs of tags within the interrogation ranges of a reader. The identification protocols are generally classified into two categories: Aloha-based protocols [7], [31], [37] and Tree-based protocols [8], [38]–[40]. In ALOHA-based identification protocols, the reader queries the tags and periodically broadcasts synchronization signals to create a slotted time frame. Upon receiving such a request, each tag randomly picks a slot in the frame to relay its ID information. If a tag *exclusively* occupies a slot, its ID can be received by the reader. In contrary, if it shares a common slot with other tags, then its ID cannot be received by the reader due to the signal collision, and therefore retransmission is required [9]. The tree-based identification protocol is a recursive depth-first searching algorithm performed by the reader. Specifically, the reader organizes all IDs in a binary tree whose height is equal to the length of a tag ID. The left (right) branches of the tree is marked by '0s' ('1s'). Clearly, each leaf corresponds to a potential tag ID. The reader queries the tags by broadcasting a prefix starting from the root of the binary tree. The tags whose IDs match the queried prefix will respond their ID information. If two or more tags respond simultaneously, signal collision will occur, the reader then generates two new query prefixes by appending a '0' and a '1' to the previous query prefix. The tags will be queried by these two new prefixes successively. On the other hand, if exactly one (or none) tag responds its ID information, the reader will successfully receive the corresponding ID (or receive nothing). Then, the new nearby prefix will be queried in the next time. This process continues until all the tags have been identified [41].

Besides the exact identification, the problem of estimating the cardinality of tags has also attracted great attention from the research community. The first literature about tag estimation was proposed by Kodialam *et al.* in [13]. The proposed Unified Simple Estimator (USE) and Unified Probabilistic Estimator (UPE) perform estimation based on the number of empty slots or that of collision slots in a frame, respectively. Qian *et al.* [15] exploited the hashing with geometric distribution to estimate the cardinality of tags and thus proposed the Lottery Frame (LoF) scheme. Zheng *et al.* proposed Probabilistic Estimation Tree (PET) to provide a estimation method for the RFID systems which work based on tree-walking algorithms [17]. Shahzad *et al.* proposed the Average Run based Tag estimation (ART) by observing the average length of sequences of consecutive non-empty slots [9]. Li *et al.* proposed an estimation scheme called Maximum Likelihood Estimator (MLE) which takes the energy-efficiency into consideration [14]. These estimation schemes concentrate on approximating the cardinality of tags in a static RFID system. However, in practice, the RFID systems are usually dynamic—the tagged items or humans may frequently move in and out. The above estimation schemes can only tell you, for example, there are $10,000$ tags in the system at time T_1 and $15,000$ tags at time T_2. However, they cannot tell you how many tags are moved out and how many new ones are moved in during this period. Liu *et al.* studied the tag cardinality estimation for multi-category RFID systems [12]. For privacy reason, Liu *et al.* also studied the RFID estimation problem with the presence of blocker tag [20]. Xiao *et al.* studied the problem of tag estimation focusing on dynamic RFID systems [21]. ZDE scheme needs the reader to observe all slots in a time frame, which triggers its low time-efficiency. Gong *et al.* proposed INformative Counting (INC) to estimate the number of counterfeit tags [22].

VI. Conclusion

This paper studies how to efficiently track the key tag population, *i.e.*, estimating the cardinality of remaining key tags and the cardinality of absent key tags. The estimation results are of practical importance in many application scenarios such as tracking the popularity of a set of items or their inventory to formulate a good replenishment plan. The main challenge is how to subtract the interference replies from a large number of ordinary tags. To address this problem, we first propose a *Basic Key tag Tracking (B-KT)* protocol, whose advantage is that the reader only needs to observe the expected singleton slots instead of the whole long time frame. To save time, B-KT skips the expected empty/collision slots. Based on B-KT, we exploit the sampling idea and early termination tactic to further propose the *Sampling-based Key tag Tracking (S-KT)* protocol, which possesses better time-efficiency and scalability. This paper also theoretically investigates the parameter settings to guarantee the estimation accuracy arbitrarily set by the users. Extensive simulation experiments have been conducted to evaluate the performance of the proposed S-KT protocol. The results demonstrate that our protocol outperforms related protocols by significantly reducing the execution time.

Acknowledgment

This work is supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61225010); the State Key Program of National Natural Science of China(Grant No. 61432002); NSFC Grant Nos. 61272417, 61300189 and 61370199; Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20130041110019), and the Fundamental Research Funds for the Central Universities (Grant. DUT15QY20). This work is partially supported by Tianjin Key Laboratory of Advanced Networking (TANK), Tianjin 300350, China

References

[1] X. Liu, K. Li, H. Qi, B. Xiao, and X. Xie, "Fast Counting the Key Tags in Anonymous RFID Systems," *Proc. of IEEE ICNP*, 2014.

[2] L. Shangguan, Z. Zhou, X. Zheng, L. Yang, Y. Liu, and J. Han, "ShopMiner: Mining Customer Shopping Behavior in Physical Clothing Stores with Passive RFIDs," *Proc. of ACM SenSys*, 2015.

[3] F. R. B. Nath and R. Want, "RFID Technology and Applications," *Proc. IEEE Pervasive Computing*, 2006.

[4] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu, "Relative Localization of RFID Tags using Spatial-temporal Phase Profiling," *Proc. of USENIX NSDI*, 2015.

[5] Y. Zheng and M. Li, "Fast Tag Searching Protocol for Large-Scale RFID Systems," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 924–934, 2013.

[6] S. Muhammad and A. X. Liu, "Expecting the Unexpected: Fast and Reliable Detection of Missing RFID Tags in the Wild," *Proc. of IEEE INFOCOM*, 2015.

[7] F. C. Schoute, "Dynamic Frame Length ALOHA," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 565–568, 1983.

[8] M. Shahzad and A. X. Liu, "Probabilistic optimal tree hopping for rfid identification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 796–809, 2015.

[9] ——, "Fast and accurate estimation of rfid tags," *Networking, IEEE/ACM Transactions on*, vol. 23, no. 1, pp. 241–254, 2015.

[10] 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*, 2013.

[11] M. Kodialam, T. Nandagopal, and W. C. Lau, "Anonymous Tracking using RFID tags," *Proc. of IEEE INFOCOM*, 2007.

[12] X. Liu, K. Li, A. X. Liu, S. Guo, A. L. Wang, X. Xie, and J. Wu, "Multi-category RFID Estimation," *IEEE/ACM Transactions on Networking*, in press, 2016.

[13] M. Kodialam and T. Nandagopal, "Fast and Reliable Estimation Schemes in RFID Systems," *Proc. of ACM MobiCom*, 2006.

[14] T. Li, S. Wu, S. Chen, and M. Yang, "Generalized Energy-Efficient Algorithms for the RFID Estimation Problem," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1978–1990, 2012.

[15] C. Qian, H. Ngan, Y. Liu, and L. Ni., "Cardinality Estimation for Large-scale RFID Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1441–1454, 2011.

[16] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID Tags Efficiently and Anonymously," *Proc. of IEEE INFOCOM*, 2010.

[17] Y. Zheng and M. Li, "PET: Probabilistic Estimating Tree for Large-scale RFID Estimation," *IEEE Transactions on Mobile Computing*, vol. 11, no. 11, pp. 1763–1774, 2012.

[18] ——, "ZOE: Fast Cardinality Estimation for Large-scale RFID Systems," *Proc. of IEEE INFOCOM*, 2013.

[19] B. Chen, Z. Zhou, and H. Yu, "Understanding RFID Counting Protocols," *Proc. of ACM MobiCom*, 2013.

[20] X. Liu, B. Xiao, K. Li, A. X. Liu, J. Wu, X. Xie, and H. Qi, "RFID Estimation with Blocker Tags," *IEEE/ACM Transactions on Networking*, in press, 2016.

[21] Q. Xiao, B. Xiao, and S. Chen, "Differential Estimation in Dynamic RFID Systems," *Proc. of IEEE INFOCOM*, 2013.

[22] W. Gong, K. Liu, X. Miao, Q. Ma, Z. Yang, and Y. Liu, "Informative Counting: Fine-grained Batch Authentication for Large-Scale RFID Systems," *Proc. of ACM MobiHoc*, 2013.

[23] T. Li, S. Chen, and Y. Ling, "Efficient Protocols for Identifying the Missing Tags in a Large RFID System," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1974–1987, 2013.

[24] D. E. Smith, *A Source Book in Mathematics*. Courier Dover Publications, 2012.

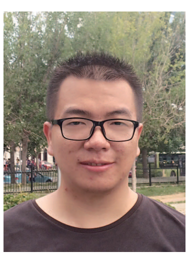
[25] J. Rice, *Mathematical Statistics and Data Analysis*. Cengage Learning, 2006.

[26] Y. Qiao, S. Chen, T. Li, and S. Chen, "Energy-efficient Polling Protocols in RFID Systems," *Proc. of ACM MobiHoc*, 2011.

[27] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen, "A Time-efficient Information Collection Protocols for Large-scale RFID Systems," *Proc. of IEEE INFOCOM*, 2012.

[28] W. Luo, S. Chen, T. Li, and S. Chen, "Efficient Missing Tag Detection in RFID Systems," *Proc. of IEEE INFOCOM*, 2011.
[29] W. Luo, S. Chen, T. Li, and Y. Qiao, "Probabilistic Missing-tag Detection and Energy-Time Tradeoff in Large-scale RFID Systems," *Proc. of ACM MobiHoc*, 2012.
[30] N. V. Smirnov, I. V. Dunin-Barkovskij, and W. Richter, *Mathematische Statistik in der Technik*. Dt. Verlag d. Wiss., 1963.
[31] S. Lee, S. Joo, and C. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification," *Proc. of IEEE MobiQuitous*, 2005.
[32] L. Yang, J. Han, C. Wang, T. Gu, and Y. Liu, "Season: Shelving Interference and Joint Identification in Large-scale RFID Systems," *Proc. of IEEE INFOCOM*, 2011.
[33] L. Xie, H. Han, Q. Li, J. Wu, and S. Lu, "Efficiently Collecting Histograms Over RFID Tags," *Proc. of IEEE INFOCOM*, 2014.
[34] L. Kong, L. He, Y. Gu, M.-Y. Wu, and T. He, "A Parallel Identification Protocol for RFID Systems," *Proc. of IEEE INFOCOM*, 2014.
[35] B. Sheng, Q. Li, and W. Mao, "Efficient Continuous Scanning in RFID systems," *Proc. of IEEE INFOCOM*, 2010.
[36] X. Liu, K. Li, G. Min, Y. Shen, A. X. Liu, and W. Qu, "Completely Pinpointing the Missing RFID Tags in a Time-Efficient Way," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 87–96, 2015.
[37] L. G. Roberts, "Aloha Packet System with and without Slots and Capture," *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.
[38] J. Myung and W. Lee, "Adaptive Splitting Protocols for RFID Tag Collision Arbitration," *Proc. of ACM MobiHoc*, 2006.
[39] N. Bhandari, A. Sahoo, and S. Iyer, "Intelligent Query Tree (IQT) Protocol to Improve RFID Tag Read Efficiency," *Proc. of IEEE ICIT*, 2006.
[40] V. Namboodiri and L. Gao, "Energy-Aware Tag Anti-Collision Protocols for RFID Systems," *Proc. of IEEE PerCom*, 2007.
[41] M. Lehtonen, F. Michahelles, and E. Fleisch, "How to Detect Cloned Tags in a Reliable Way from Incomplete RFID Traces," *Proc. of IEEE RFID*, 2009.

Xiulong Liu received the B.E. degree from the School of Software Technology, Dalian University of Technology, China, in 2010. Currently, he is a Ph.D. candidate in the School of Computer Science and Technology, Dalian University of Technology, China. He served as a Research Assistant in the Hong Kong Polytechnic University in 2014, and a Visiting Scholar in Temple University in 2015. His research interests include RFID systems and wireless sensor networks.



Xin Xie received the B.Sc. B.E degree in computer science from Dalian University of Technology, Dalian, China, in 2013. He is currently pursuing the Ph.D in Computer application technology at Dalian University of Technology. His research interests includes RFID technologies and wireless networks.

Keqiu Li received the bachelor's and master's degrees from the Department of Applied Mathematics at the Dalian University of Technology in 1994 and 1997, respectively. He received the Ph.D. degree from the Graduate School of Information Science, Japan Advanced Institute of Science and Technology in 2005. He also has two-year postdoctoral experience in the University of Tokyo, Japan. He is currently a professor in the School of Computer Science and Technology, Dalian University of Technology, China. He has published more than 100 technical papers, such as IEEE TPDS, ACM TOIT, and ACM TOMCCAP. He is an Associate Editor of IEEE TPDS and IEEE TC. He is a senior member of IEEE. His research interests include internet technology, data center networks, cloud computing and wireless networks.

Bin Xiao received the B.Sc. and M.Sc. degrees in electronics engineering from Fudan University, China, in 1997 and 2000, respectively, and the Ph.D. degree in computer science from the University of Texas at Dallas in 2003. After his Ph.D. graduation, he joined the Hong Kong Polytechnic University as an assistant professor. Currently, he is an associate professor in the Department of Computing at The Hong Kong Polytechnic University, Hong Kong. His research interests include mobile cloud computing, data management, network security, wireless sensor networks, and RFID systems. He is an associate editor for the International Journal of Parallel, Emergent and Distributed Systems. Dr. Xiao is a recipient of the Best Paper Award from IEEE/IFIP EUC 2011.

Jie Wu is the chair and a Laura H. Carnell Professor in the Department of Computer and Information Sciences at Temple University. Prior to joining Temple University, he was a program director at the National Science Foundation and Distinguished Professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr.Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Computers, IEEE Transactions on Service Computing, and Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair/chair for IEEE MASS 2006, IEEE IPDPS 2008 and IEEE ICDCS 2013, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. Currently, he is serving as general chair for ACM MobiHoc 2014. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

Heng Qi is an associate professor at the School of Computer Science and Technology, Dalian University of Technology, China. He received bachelor's degree from Hunan University in 2004 and master's degree from Dalian University of Technology in 2006. Then he received his Ph.D. degree from Dalian University of Technology in 2012. His research interests include computer network, wireless network and multimedia computing.

Dawei Lu is an associate professor at the School of Mathematical Sciences, Dalian University of Technology, China. He received the bachelor's degree from Dalian University of Technology in 2004, and the Ph.D. degree in 2009. His research interests include precise large deviation for sums of random variables, gamma function; Euler constant, and the first exit time of Brownian motion.