

Toward More Rigorous and Practical Cardinality Estimation for Large-Scale RFID Systems

Wei Gong, *Member, IEEE*, Jiangchuan Liu, *Fellow, IEEE*, Kebin Liu, *Member, IEEE*,
and Yunhao Liu, *Fellow, IEEE, ACM*

Abstract—Cardinality estimation is one of the fundamental problems in large-scale radio frequency identification systems. While many efforts have been made to achieve faster approximate counting, the accuracy of estimates itself has not received enough attention. Specifically, most state-of-the-art schemes share a two-phase paradigm implicitly or explicitly, which needs a rough estimate first and then refines it to a final estimate meeting the desired accuracy; we observe that the final estimate can largely deviate from the expectation due to the skewed rough estimate, i.e., the accuracy of final estimates is not rigorously bounded. This negative impact is hidden because former solutions either assume perfect rough estimates or rough estimates that can be produced by uniform random data or perfect hash functions that can turn any data into uniform random data. Unfortunately, both of them are hard to meet in practice. To address the above issues, we propose a novel scheme, namely, “rigorous and practical cardinality (RPC)” estimation. RPC adopts the two-phase paradigm, in which the rough estimate is derived in the first phase using pairwise-independent hashing. In the second phase, we employ t -wise-independent hashing to reinforce the rough estimate to meet arbitrary accuracy requirements. We validate the effectiveness and performance of RPC through theoretical analysis and extensive simulations. The results show that the RPC can meet the desired accuracy all the time with diverse practical settings while previous designs fail with non-uniform data.

Index Terms—RFID tags, cardinality estimation, pairwise independent hashing, t -wise independent hashing.

I. INTRODUCTION

ESTIMATING the cardinality of tags is of great importance in many RFID applications, e.g., warehouse management, tag identification, and privacy sensitive RFID systems. Imagine a huge warehouse of large retailer like Wal-Mart, thousands of mobile phones, ipods, and other office supplies are intensively piled [1]. It is tempting to quickly

and accurately estimate the number of those tagged objects for daily or weekly inventory reports, instead of laborious and unreliable humanly counting. Important applications also exist in other scenarios, such as counting the number of tourists or conference attendees with RFID tickets/cards. Furthermore, most of the RFID identification schemes [2]–[4] require an accurate estimate of tag population to set the optimal frame size. In some privacy sensitive scenarios, the exposure of unique identification information on tags, such as driver licenses and e-passports [5], can put important personal privacy at risk. Therefore, a scheme that can use the non-identifiable information from tags to compute the cardinality is necessary.

While exact counting based methods prove to be not scalable with respect to the rapidly growing number of tags [6], researchers try to do approximate counting in different ways [7]–[13]. The most recent state-of-the-art work is proposed by Zhou *et al.* [12] in which they derive the theoretical lower bound for RFID cardinality estimation, $\mathcal{O}((\log \log n + \frac{\varepsilon^{-2}}{\log \frac{1}{\varepsilon}}) \log \delta^{-1})$, where n is the upper bound of the cardinality of tags, ε and δ are user-specified thresholds for the relative error and error probability of estimates. They further prove that the two-phase paradigm is the best way to achieve near-optimal solutions, and most of the performance gains in prior works should be attributed to following this paradigm implicitly or explicitly. The core of two-phase designs is to obtain a rough estimate first and then refine it to the desired accuracy.

While time-efficiency has been greatly improved in prior methods, the accuracy of estimator has not been well investigated yet. After deep diving into current solutions, we observe that the final estimate can largely deviate from the expectation with the skewed rough estimate. This negative impact is largely hidden because former methods either assume perfect rough estimates [9], [14] or rough estimates [11], [12] that can be produced by uniform random data or perfect hash functions that can turn any data into uniform random data. Unfortunately, both of them are hard to meet in practice. Therefore, an intriguing question comes up: whether we can design a practical two-phase scheme that is able to make the final estimate rigorously bounded by using constructible and simple hash functions, irrespective of data distribution?

Our answer is positive; in this work we propose a new mechanism, Rigorous and Practical Cardinality (RPC) estimation, by using the universal hashing. The RPC adopts the

Manuscript received November 7, 2016; revised June 9, 2016 and September 27, 2016; accepted October 31, 2016; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. Hou. This work was supported in part by a Mitacs Accelerate Internship, in part by the Canada Technology Demonstration Program, in part by a Canada NSERC Discovery Grant, in part by the NSERC E.W.R. Steacie Memorial Fellowship, and in part by the Project NSFC under Grant 61472268, Grant 61303196, and Grant 61472218.

W. Gong and J. Liu are with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: gongweig@sfu.ca; jcliu@sfu.ca).

K. Liu and Y. Liu are with the Tsinghua National Laboratory for Information Science and Technology, School of Software, Tsinghua University, Beijing 100084, China, (e-mail: kebin@greenorbs.com; yunhao@greenorbs.com).

Digital Object Identifier 10.1109/TNET.2016.2634551

two-phase paradigm. In particular, given the required relative error ε and error probability δ for the final estimate, the RPC performs a rough estimation using the pairwise independent hashing in the first phase. We prove that the accuracy of this rough estimate is constant-factor bounded. In the second phase, the RPC employs the t -wise¹ independent hashing to refine the rough estimate using multiple single-slot trials. Through detailed analysis, we show that the RPC is able to get an estimate that meets the relative error ε with probability at least $\frac{11}{20}$ in a single two-phase round. After this, a Monte Carlo algorithm is introduced to boost the error probability from $\frac{11}{20}$ to δ using $\mathcal{O}(\log \delta^{-1})$ two-phase rounds. Finally, the RPC achieves $\mathcal{O}((\log \log n + \varepsilon^{-2}) \log \delta^{-1})$ estimation efficiency. This efficiency is near-optimal and is within a small $\mathcal{O}(\log \frac{1}{\varepsilon})$ factor from the theoretical lower bound [12]. Through detailed analysis and comprehensive simulations, we show that the RPC is practical, scalable, and reliable. More importantly, its accuracy is rigorously guaranteed regardless of data distribution.

We view this work as an essential step towards practical solutions of RFID estimation in large-scale as it eliminates an underpinning assumption of perfect hash functions in former schemes. We hope this can fuel more community interests and future work to design better estimation schemes along this line.

II. PRELIMINARIES

A. Problem and Assumption

An RFID system typically consists of several RFID readers and a number of tags. Each tag is attached with unique identification information (tagID) and can perform simple computation as well as communication by backscattering the reader's RF signals. Consider there are N tags in the interested area. The aim of approximating the cardinality of tags is to acquire the quantity of tags in the interested region while meeting specified accuracy requirements. Generally, accuracy requirements contain two essential parameters, the target relative error, ε , and the target error probability, δ . Given an approximated result \hat{N} , then the actual relative error is derived as $\frac{|\hat{N}-N|}{N}$. We define that an (ε, δ) approximation scheme for N is a probabilistic process that, given any $0 < \varepsilon < 1$ and $0 < \delta < 1$, the result estimate \hat{N} is within the relative error ε with probability at least $1 - \delta$. This definition can be formally defined as

$$\Pr[|\hat{N} - N| \leq \varepsilon N] \geq 1 - \delta.$$

For example, if the exact quantity of tags is 1000, the user-specified relative error ε is 0.01 and the target error probability δ is 0.01, then the output estimates of an (ε, δ) scheme should be between 990 to 1010 with the probability no less than 0.99. Table I summarizes the main notations used across this paper.

B. Communication Model and Tags

Following EPC Class 1 Generation 2 (C1G2) standard [15], we assume a frame-slotted ALOHA model in RFID systems.

¹ t is a parameter depicting the strength of independence and will be formally introduced in IV-A.

TABLE I
MAIN NOTATIONS

Symbols	Descriptions
N	exact cardinality of tags
\hat{N}	estimated cardinality of tags
S	a tag set
$[D]$	tagID domain $\{1, \dots, D\}$
C	a constant-factor estimate
Z_{tagID_i}	number of trailing zeros for hash $tagID_i$
k	constant factor for approximation
f	# of independent rounds
t	#-wise independent hash function
m	# of t -wise independent hash functions
h_c	ideal hash function (truly random)
q	the probability that #1 bin is non-empty in h_c
λ	bound for the gap between q and \hat{q}
h^t	a t -wise independent hash function
\mathcal{H}^m	a subset of size m of t -wise independent hash family
p	the probability that #1 bin is non-empty in \mathcal{H}^m
n	upper bound for the number of tags in a system

We adopt the *Reader Talks First* mode, which is widely used in many applications [8], [9]. In this model, the reader first initializes communication and then wait for tags' responses in each slot. If there is no response in this slot, the slot is called an *empty-slot*. Otherwise, it would be called a *non-empty-slot*. In theory, the reader needs only one bit to encode this simple response: "1" for busy signals and "0" for idle states. Furthermore, in some situations the reader may need to distinguish the *singleton-slot* that receives response from only one tag from the *collision-slot* that contains responses from more than one tag; a long-bit response thus can be used to discern these two types of non-empty-slot. In the design and evaluation of our RPC scheme, we only need to distinguish the empty-slot from the non-empty-slot. Generally there are two types of tags: (1) active tags that often have their own rechargeable batteries and thus have a reading distance between 150 to 300 feet; (2) passive tags that capture energy in the reader's RF signals and have a reading range less than 20 feet.

III. SOURCES OF ESTIMATION INACCURACY—TWO CASE STUDIES

Generally, the two-phase design of cardinality estimation in RFID includes the first phase that aims to get a rough estimate and the second phase that refines the rough estimate to arbitrary user-specified accuracy [10], [12]. While a long line of research has been done on improving time-efficiency, the estimation accuracy of prior methods is not well investigated yet. By carefully examining prior methods, we find that the accuracy of final estimates could be seriously affected by skewed rough estimates. In the following, we present empirical findings and then deduce the reasons for our observations.

A. How Does Rough Estimates Affect Final Estimates?

We focus on two recent solutions, ART [11] and SRC [12], which use the two-phase design explicitly. Our experiments

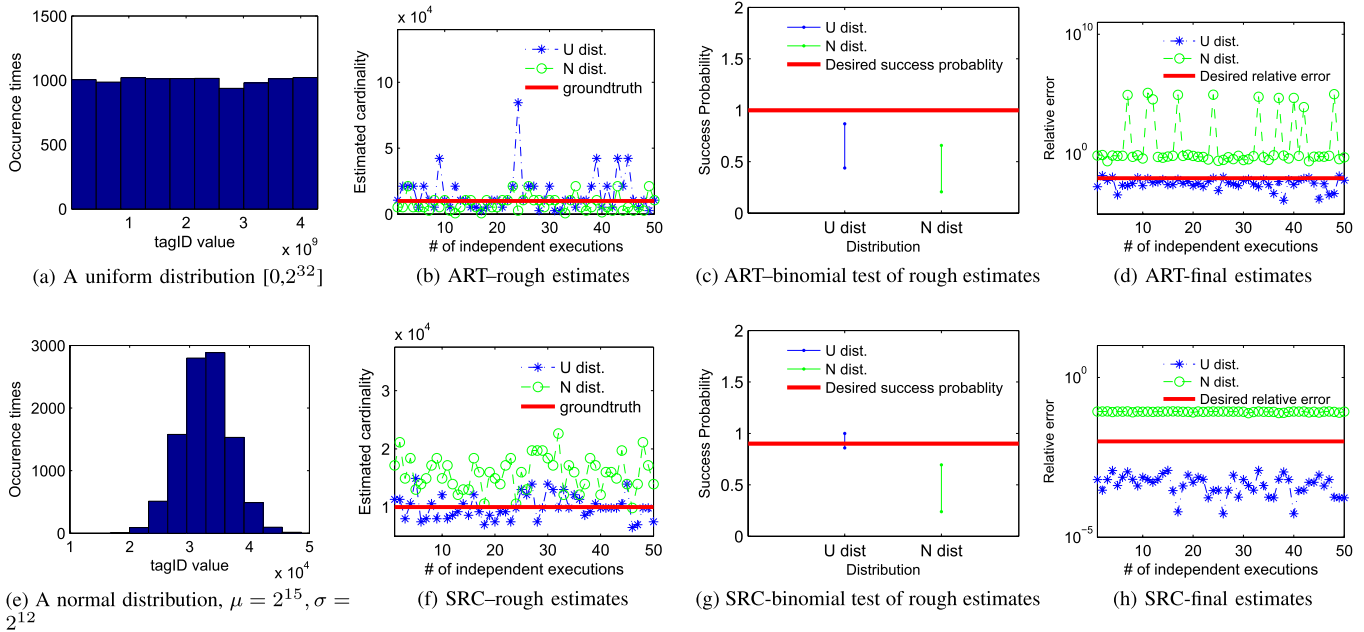


Fig. 1. A detailed investigation of two state-of-the-art schemes ART [11] and SRC [12] with uniformly distributed (a) and normally distributed data (e). We see that the quality of ART’s rough estimates under the uniform distribution is not that desirable in (c), leading to the final estimates, some of which meet desired accuracy and the others do not as in (d). With the normal distribution, the quality of ART’s rough estimates is far from ideal, therefore it is not surprising the final estimates are largely skewed. Similar trends can be observed for SRC in (g) and (h). The major difference is that the quality of SRC’s rough estimates in the uniform distribution meets its target, so the corresponding final estimates fit the desired accuracy well, but it still fails under the normal distribution.

with other schemes share similar observations. For brevity, their results are not included here. For both algorithms, we conduct experiments² based on the data following a uniform distribution and a normal distribution, respectively,³ as shown in Figure 1a and Figure 1e.

Let’s examine ART first. In the first phase of ART, it tries to obtain an upper bound of the cardinality of tags, t_m , as a rough estimate, which means it needs the rough estimate to be always greater than the actual cardinality. As shown in Figure 1b, for both distributions, not all the rough estimates are above the ground truth. The results of the normal distribution are even worse than that of the uniform distribution. To further study the quality of rough estimates, we conduct a Binomial test in which the event is defined as whether a rough estimate achieves the upper bound for the actual cardinality. In Figure 1c, we report the estimated probability intervals of the Binomial test with 99.99% confidence intervals. We see that neither the estimated probability intervals in the uniform distribution nor those in the normal distribution has intersections with the desired probability or beyond. But the estimated probability intervals

in the uniform distribution is closer to the desired probability compared to those in the normal distribution, which means the quality of rough estimates in the uniform distribution is better. We depict the final results of ART in Figure 1d, which shows a clear relationship between rough estimates and final estimates. ART fails to achieve the desired relative error in the normal distribution, whereas it achieves the desired accuracy for most of the executions in the uniform distribution. Actually this finding goes well with the report of [12], in which it says ART “will actually achieve a relative error that is somewhat larger than the target ε ”.

Next, we check how SRC goes in the same settings. For SRC, it tries to obtain a rough estimate that should be within the relative error 0.5 and the success probability 0.9. In Figure 1f and Figure 1g, we see that in the uniform distribution, SRC successfully gets the rough estimates as it expects, whereas in the normal distribution, the quality of the rough estimates is far from adequate since the estimated probability intervals is way far from the desired 0.9, even with the confidence intervals 99.99%. In Figure 1h, it is not surprising to see that the final estimates in the uniform distribution perfectly achieve the target relative error. But due to the bad quality of rough estimates, SRC fails to meet the desired accuracy in the normal distribution.

B. What Are the Reasons for Skewed Estimates?

For ART, it needs an upper bound of cardinality for the second phase estimation. Therefore when the rough estimate fails to be an upper bound, the final estimate is definitely affected. Although ART designers already try to make their

²Our experiment settings are as follows. Following C1G2 [15], the size of identification information for each tag (tagID) is 96 bits. The ground truth for the cardinality is 10,000. The user-specified relative error $\varepsilon = 0.01$ and error probability $\delta = 0.01\%$, which is sufficiently low to let us focus on the relative error. Each algorithm takes 50 independent executions. Since here we focus on accuracy, not time-efficiency, we let each algorithm use enough time as long as it needs. Since neither of them specifies the type of hash function in their papers, we use the pairwise independent hashing as an alternative, which has guaranteed uniformity in theory and is widely used in Bloom Filter.

³We also tested various data under different distributions for prior schemes. The results are also skewed. The normal distribution is just a representative.

TABLE II
A COMPARISON OF MAJOR EXISTING ESTIMATION SCHEMES

	Complexity	Uniform hash implementation	Rough estimate	Quality of Rough estimate
EZB [16]	$\mathcal{O}(\frac{1}{\epsilon^2} \log n)$	Not specified	Choose partitioned ranges in advance	Depends on user input
FNEB [14]	$\mathcal{O}(\frac{1}{\epsilon^2} \log n)$	Not specified	estimated	Not investigated
LOF [7]	$\mathcal{O}(\frac{1}{\epsilon^2} \log n)$	Not specified	N/A	N/A
PET [8]	$\mathcal{O}(\frac{1}{\epsilon^2} \log \log n)$	Not specified	N/A	N/A
ART [11]	$\mathcal{O}(\frac{1}{\epsilon^2} + \log n)$	Not specified	estimated	Not investigated
ZOE [9]	$\mathcal{O}(\frac{1}{\epsilon^2} + \log \log n)$	Not specified	estimated	Not investigated
SRC [12]	$\mathcal{O}(\frac{1}{\epsilon^2} + \log \log n)$	Not specified	estimated	Investigated but might fail sometimes
RPC	$\mathcal{O}(\frac{1}{\epsilon^2} + \log \log n)$	Specified, universal hashing	estimated	Investigated and rigorous bounded

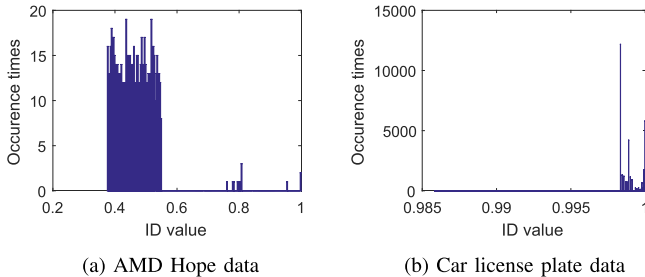


Fig. 2. (a) the histogram of AMD RFID data [17]. (b) the histogram of sampled RFID car license plate data in a city [18]. Both cases show that a sampled group of tagIDs is not guaranteed to be uniformly distributed. Note that tagID values are normalized in both figures.

upper bound big enough, they do not characterize the error probability of rough estimates and further do not take this error probability into account for computing the final error probability. Other schemes, e.g., [9], [14], also share this drawback, since they all need a perfect rough estimate for the second phase to ensure the quality of final estimates. So the first reason for skewed final estimates is that *the quality of rough estimates in most existing methods might be lower than desirable, i.e., they do not realize how accurate the rough estimate should be to make the final estimate accuracy-guaranteed.*

Probably SRC is the closest one addressing the above limitation since SRC explicitly requires its rough estimate to be within the relative error 0.5 and the success probability 0.9 by invoking LOF [7] 10 times in the first phase. However, LOF may fail to achieve the desired accuracy as SRC needs. First, the estimate of LOF might be skewed in the non-uniform distribution which comes from its pre-stored hash values on tags. We use a toy example to illustrate this. Suppose we have 1,000 tags of values $(tagID_1, tagID_2, \dots, tagID_{1000})$, which are uniformly distributed in the range of $[0, 2^{128} - 1]$.⁴ Then we take out first 50 tags $(tagID_1, tagID_2, \dots, tagID_{50})$. Obviously, these sampled tags are not uniformly distributed in $[0, 2^{128} - 1]$ as expected in the algorithm of LoF. We also present two realworld RFID datasets in Figure 2, which shows a group of sampled RFID data is not sure to be uniformly distributed. As there are so many reasons accounting for such non-uniform distribution such as sampling process and characteristics of

data, it is hard to enumerate all of them. Grouping is one of the major causes. As some bits of tagID are for group IDs (or for multi-layer grouping), it can make the tagID distribution even more complicated. For more details regarding to RFID grouping problems, please refer to [19]. To summarize, samples from uniform distribution are not guaranteed to be uniformly distributed.

As one may wonder that whether increasing the trials of LoF in the first phase would mitigate the skewed estimates. We further investigate this aspect with diverse settings. Unfortunately, we have tried and found that more trials in the first phase do not help in improving the skewed rough estimates and thus the final estimates. The main reason is that the underpinning assumption that the data is uniform random or can be made uniform random by perfect hash functions is still invalid in the first and second phase. From a theoretical point of view, this negative effect comes from that *a deterministic hash function cannot offer any guarantee in the distribution of hash values in presence of adversaries since the adversaries can even choose pre-hash values that have exactly the same hashes* [20], [21]. In summary, without employing practical hash functions and properly characterizing such hash functions, the accuracy of estimators is hard to be rigorously bounded.⁵

C. How to Design Accuracy-Guaranteed Estimators?

It is worth noting that we reveal the above limitations of prior solutions by introducing different data distributions. Actually, it just comes from one of many practical perspectives, not exhaustive. More specifically, we want to emphasize that for practical systems, *both the rough estimate and the final estimate needed to be rigorously bounded using practical hash functions.* Table II compares RPC with major existing RFID estimation schemes.⁶

IV. RIGOROUS AND PRACTICAL CARDINALITY ESTIMATION

A. Universal Hashing

The basic idea of universal hashing is to pick up the hash function randomly from a large family of hash functions,

⁵We also investigate other well-known hash functions in RFID counting, e.g., murmur3, lookup3. Unfortunately, we find that they cannot meet the requirement all the time either.

⁶We omit δ here, since repeating the algorithm for $\mathcal{O}(\log \frac{1}{\delta})$ times can meet the target δ by using Monte Carlo randomized algorithms.

⁴128 comes from the length of MD5 digest.

therefore the randomness in choosing the hash function can be used to ensure a guarantee on the uniform random distribution of hash values, which fits our design goal quite well. Note that although the universal hashing is widely used in many hash related applications, e.g., linear probing [22], bloom filter [23]. We are the first to bring this to RFID estimation.

We give a brief introduction to t -universal hashing which is also called t -wise independent hashing. For more details, please refer to the seminal work [20]. Assume we want to map keys from some universe D into Y bins, a family of hash functions $\mathcal{H} = h^t : D \rightarrow [Y]$ is t -wise independent where $[Y] = \{0, 1, \dots, Y - 1\}$, if for any t distinct keys $(x_1, x_2, \dots, x_t) \in D^t$ and any t hash values $(y_1, y_2, \dots, y_t) \in [Y]^t$, we have

$$\Pr\left[\bigwedge_{j=1}^t (h^t(x_j) = y_j)\right] = \prod_{j=1}^t \Pr[h^t(x_j) = y_j]. \quad (1)$$

If $t = 2$, we also call it the pairwise independent hash function⁷ and one typical form is

$$h(x) = ax + b \pmod{pr},$$

where pr is a prime, a, b are random integers modulo pr with $a \neq 0$. Based on this pairwise independence design, it is easy to extend it to t -wise independence. A formal definition is as follows.

Let pr be a prime and $t \geq 2$ be an integer. Then $\mathbb{Z}_{pr} = \{0, 1, \dots, pr - 1\}$ is a field with operations of addition and multiplication mod pr . The hash function $h^t : \mathbb{Z}_{pr}^k \rightarrow \mathbb{Z}_{pr}^k$ is t -wise independent hash function given by

$$h^t(x) = \sum_{i=0}^{t-1} a_i x^i \pmod{pr}. \quad (2)$$

The parameter t can be tuned according to different independence requirements. Intuitively, the larger t is, the closer it is to the truly random hashing.

B. Basic Design




































Our RPC adopts a two-phase design, in which the first phase is to obtain a rough estimate using loglog-counting [24] and pairwise independent hash functions, and the second phase is able to provide a finer estimate using “balls and bins” and t -wise independent hash functions. The basic ideas are briefly presented as follows.

First-Phase: Let S be a set of N tags, $\{tagID_1, \dots, tagID_N\}$. For simplicity, we assume that there is a pairwise independent hash function $h : [0, D] \rightarrow [0, 2^w - 1]$. Let $\lceil w = \log N \rceil$, i.e., $2^w \geq N$. We use Z_{tagID_i} to denote the number of trailing zeros (rightmost zeros) in the binary form of $h(tagID_i)$ and use Z^{max} to denote the maximum trailing zeros of hash values for all i in S .

The basic algorithm approximates the cardinality of S as

$$C = 2^{Z^{max}}. \quad (3)$$

⁷Note that pairwise independence does not imply mutual independence.

		Tag ₁	Tag ₂	Tag ₃	Tag ₄	Tag ₅
	Reader					
↓	h_1					
	h_2					
	h_3					
	h_4					
	h_5					
	h_6					



 response ($h_i=0$)
 silence ($h_i \neq 0$)

Fig. 3. An illustrative example for the RPC.

For example, we assume that a tag set $S = \{2, 4, 6, 8\}$, $w = 2$, and the hash values are $\{0, 1, 2, 3\}$. According to that $h(2) = 0 = (00)_2$, we can obtain $Z_2 = 2$. Likewise, $Z_4 = 0, Z_6 = 0, Z_8 = 1$, hence $Z^{max} = 2$. Finally the rough estimate is given by $C = 2^{Z^{max}} = 4$.

Second-Phase: After we obtain a rough estimate C . By introducing a “balls and bins” approach, we randomly hash N balls into C bins and use the observed probability of the first bin being non-empty to estimate the final estimate as,

$$\hat{N} = \frac{\ln(1 - \hat{q})}{\ln(1 - \frac{1}{C})}. \quad (4)$$

Figure 3 shows an example to demonstrate the RPC’s workflow. Suppose we have 5 tags in total and a rough estimate $C = 8$. Then the reader starts probing tags using t -wise independent hash functions (h_1, \dots, h_6) . After each probing, the reader just needs to record the status of the first slot. The probe result is recorded as busy if there is at least one response from tags, e.g., h_1, h_4, h_6 . Otherwise it is marked as empty, e.g., h_2, h_3, h_5 . Therefore we can estimate the probability that the first slot is non-empty as $3/6$. Together with the rough estimate C , we get a final estimate of N as $\hat{N} = \frac{\ln(1-3/6)}{\ln(1-1/8)} \approx 5$.

Next, we are going to detailed examine our two-phase protocols. In particular, we will prove that the result of the first phase is a constant-factor approximation, i.e., $C = \Theta(N)$, and the second phase can refine C to $(1 \pm \epsilon)$ approximation. Based on this single round two-phase estimation, we further quantify how many rounds are needed to boost the success probability to $1 - \delta$.

C. Constant-Factor Approximation

In this subsection, we are going to show that the output C is off by N at most a constant factor. Note that our first-phase estimation is based on loglog-counting [24] and the major difference is we use pairwise independent hashing, instead of perfect hashing. The two important properties of the pairwise independent function, h , are that: first, for every fixed $tagID_i$, $h(tagID_i)$ is uniformly distributed over $[0, 2^w - 1]$; second, this mapping is pairwise independent.

Definition 1: Let r be an integer between 0 and w . And k is a positive integer, r_1 is the smallest r such that $2^r > kN$, and r_2 is the smallest r such that $2^r \geq \frac{N}{k}$.

Lemma 1: $\Pr[Z_{tagID_i} \geq r] = 2^{-r}$.

Proof: In the above lemma, $Z_{tagID_i} \geq r$ means that hash value $h(tagID_i)$ is between 0 and $2^{w-r} - 1$. Since the hash value $h(tagID_i)$ is uniformly distributed in range of $[0, 2^w - 1]$, we can get

$$\Pr[Z_{tagID_i} \geq r] = \frac{2^{w-r}}{2^w} = 2^{-r}. \quad (5)$$

□

Definition 2: Given any specific r , for each $tagID_i \in S$, we define

$$x_i(r) = \begin{cases} 1 & \text{if } Z_{tagID_i} \geq r \\ 0 & \text{if } Z_{tagID_i} < r \end{cases}$$

and $X(r) = \sum_{tagID_i \in S} x_i(r)$.

By Lemma 1, we know that $x_{tagID_i}(r)$ takes 1 with the probability 2^{-r} , hence the expectation is given by

$$\mathbf{E}[x_i(r)] = 2^{-r}. \quad (6)$$

Also, the corresponding variance is given by

$$\mathbf{Var}[x_i(r)] = 2^{-r}(1 - 2^{-r}). \quad (7)$$

Lemma 2: $\Pr[X(r_1) > 0] < \frac{1}{k}$.

Proof: By the definition of r_1 and (6),

$$\mathbf{E}[X(r_1)] = \sum_{tagID_i \in S} \mathbf{E}[x_{tagID_i}(r_1)] = N \cdot 2^{-r_1} < \frac{1}{k}.$$

Therefore, by the Markov inequality, we have

$$\Pr[X(r_1) > 0] = \Pr[X(r_1) \geq 1] \leq \mathbf{E}[X(r_1)] < \frac{1}{k}. \quad \square$$

Lemma 3: $\Pr[X(r_2) = 0] < \frac{2}{k}$.

Proof: Likewise, we can obtain

$$\mathbf{E}[X(r_2)] = N2^{-r_2}.$$

Since $X(r_2)$ is the sum of pairwise independent variables and each of which has a variance $2^{-r_2}(1 - 2^{-r_2})$, the variance of $X(r_2)$ can be given by

$$\begin{aligned} \mathbf{Var}[X(r_2)] &= \mathbf{Var}\left[\sum_{i=1}^N x_i(r_2)\right] \\ &= \sum_{i=1}^N \sum_{j=1}^N \mathbf{Cov}(x_i(r_2), x_j(r_2)) \\ &= \sum_{i=1}^N \mathbf{Var}[x_i(r_2)] \\ &\quad + 2 \sum_{1 \leq i < j \leq N} \mathbf{Cov}(x_i(r_2), x_j(r_2)), \end{aligned}$$

where $\mathbf{Cov}()$ denotes covariance. Note that the last equality comes from the fact $\mathbf{Cov}(a, a) = \mathbf{Var}[a]$. By pairwise independence, $\mathbf{Cov}(x_i(r_2), x_j(r_2)) = 0$ if $i \neq j$. Thus, we can obtain

$$\begin{aligned} \mathbf{Var}[X(r_2)] &= \sum_{i=1}^N \mathbf{Var}[x_i(r_2)] = N \mathbf{Var}[x_i(r_2)] \\ &= N \cdot 2^{-r_2} \cdot (1 - 2^{-r_2}) < N2^{-r_2}. \end{aligned}$$

Further, by the Chebyshev inequality, we know

$$\begin{aligned} \Pr[X(r_2) = 0] &= \Pr[E[X(r_2)] - X(r_2) = \mathbf{E}[X(r_2)]] \\ &\leq \Pr[|E[X(r_2)] - X(r_2)| = \mathbf{E}[X(r_2)]] \\ &\leq \Pr[|E[X(r_2)] - X(r_2)| \geq \mathbf{E}[X(r_2)]] \\ &\leq \frac{\mathbf{Var}[X(r_2)]}{(\mathbf{E}[X(r_2)])^2} \\ &< \frac{N2^{-r_2}}{(N2^{-r_2})^2} \\ &= \frac{2^{r_2}}{N}. \end{aligned}$$

By the definition of r_2 , we know that $2^{r_2} < 2 \cdot \frac{N}{k}$. Otherwise, r_2 cannot be the smallest r satisfying $2^r \geq \frac{N}{k}$. Combining this and the above inequality proves that $\Pr[X(r_2) = 0] < \frac{2}{k}$. □

Theorem 1 (Constant-Factor Approximation Bound): For any $k > 3$, $\Pr[\frac{1}{k} \leq \frac{C}{N} \leq k] \geq 1 - \frac{3}{k}$.

Proof: First, we show that if $X(r_1) = 0$ and $X(r_2) \neq 0$, the above theorem is correct. If $X(r_1) = 0$, it means that there is no $tagID_i \in S$ that can give $Z_{tagID_i} \geq r_1$, and thus $Z^{max} < r_1$. Likewise, if $X(r_2) \neq 0$, it means that there is at least one $tagID_i \in S$ that can satisfy $Z_{tagID_i} \geq r_2$ and thus $Z^{max} \geq r_2$. Also, according to the definition of r_1, r_2 , and Z^{max} , we can derive that if $r_2 \leq Z^{max} < r_1$, the above theorem is correct.

By lemma 2 and lemma 3, we know $X(r_1) \geq 1$ can happen with the probability at most $\frac{1}{k}$, whereas $X(r_2) = 0$ can happen with the probability at most $\frac{2}{k}$, thus the union bound of two events happening is at most $\frac{3}{k}$. Therefore, the probability of having ' $X(r_1) = 0$ and $X(r_2) \neq 0$ ' is at least $1 - \frac{3}{k}$. □

As shown in Theorem 1, the coarse result C is indeed probabilistically bounded by a interval and is associated with a non-negligible probability. The constant-approximation here means the C is constantly deviate from the real N in probabilistic nature and this approximation factor is modeled as k , which can be *any* integer greater than 3. Note that the parameter k is to depict the probability distribution of $\frac{C}{N}$, i.e., in some case, the estimated C might be quite skewed, but its probability distribution still follows Theorem 1. For example, if let k be 100, Theorem 1 says, the probability of $\frac{1}{100} \leq \frac{C}{N} \leq 100$ is at least $1 - \frac{3}{100} = 0.97$, i.e., the probability of $\frac{C}{N} < \frac{1}{100}$ or $\frac{C}{N} > 100$ is less than $1 - 0.97 = 0.03$. More specifically, $\hat{C}/N = 10,000$ is still possible but its happening probability shall follow Theorem 1. Later, we shall include such non-negligible probability for skewed \hat{C} into the second phase using the union bound, which is detailed in the proof of Theorem 2.

D. Refining Rough Estimate to the Desired Accuracy

From the former sub-section, we obtain a constant-factor estimate $C = \Theta(N)$. By introducing a ‘‘balls and bins’’ approach, we are going to refine this rough estimate to any desired accuracy ε , i.e., pushing $\Theta(N)$ to $(1 \pm \varepsilon)N$. The key intuition is that when randomly hashing N balls into C bins, the probability that the specific one bin (such as the first bin) is empty, is highly concentrated about its expectation. Thus we form this expectation as a function of N and then

inverting such function provides a good estimate of N with high probability.

If we randomly hash N balls into C bins, then a truly random hash function of this process is given by $h_c : [D] \rightarrow [C]$, where D is the universe of input data (balls). Thus, the probability that the first bin is non-empty should be

$$q = \Pr[h_c^{-1}(0) \cap S \neq \emptyset] = 1 - (1 - \frac{1}{C})^N. \quad (8)$$

Then the following lemma shows that if we can obtain a good estimate \hat{q} that is close enough to the real q , then inverting equation 8 can produce an ε approximation of N .

Lemma 4: Let $k > 3$ and $\varepsilon > 0$. Suppose C and N are such that $\frac{1}{2k} \leq \frac{N}{C} \leq \frac{1}{2}$. Then if $|q - \hat{q}| \leq \lambda = \min(\frac{1}{e} - \frac{1}{3}, \frac{\varepsilon}{6k})$, an estimate \hat{N} , defined as

$$\hat{N} = \frac{\ln(1 - \hat{q})}{\ln(1 - \frac{1}{C})} \quad (9)$$

satisfies $|\hat{N} - N| \leq \varepsilon N$

Proof: See Appendix A for proof. \square

According to lemma 4, we know that approximating q is a good way to estimate the cardinality of tags. However, the ideal hash function h_c are not known to be constructible efficiently. Therefore we choose to employ t -wise independent hash functions to generate a desired approximation of q .

Specifically, let \mathcal{H} be a family of t -wise independent hash functions from $[D]$ into $[C]$, and $p = \Pr_{h \in \mathcal{H}}[h^{-1}(0) \cap S \neq \emptyset]$. Next we will show if t is large enough, then p can be arbitrarily close to q .

Lemma 5: Let t be $\lceil \frac{\log \frac{2}{\varepsilon}}{\log 5} \rceil$, then $|p - q| \leq \frac{\lambda}{2}$ where $\lambda = \min(\frac{1}{e} - \frac{1}{3}, \frac{\varepsilon}{6k})$.

Proof: See Appendix B for proof. \square

From the above lemma, we show that p can be at most $\frac{\lambda}{2}$ far from q ; therefore if we can obtain an estimate \hat{p} of p satisfying $|p - \hat{p}| \leq \frac{\lambda}{2}$, then $|q - \hat{p}| \leq \lambda$ can hold. Hence, we shall examine how to obtain a good estimate, \hat{p} , that is arbitrarily close to p .

Definition 3: Let $\mathcal{H}^m = \{h_1, \dots, h_m\}$ be a subset of a family \mathcal{H} of t -wise independent hash functions from $[D]$ into $[C]$. Then for each $h_i \in \mathcal{H}^m$ we define a variable

$$x_{h_i}(\mathcal{H}^m) = \begin{cases} 1 & \text{if } h_i^{-1}(0) \cap S \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

and the estimate of p is given by

$$\hat{p} = X(\mathcal{H}^m) = \frac{1}{m} \sum_{h_i \in \mathcal{H}^m} x_{h_i}(\mathcal{H}^m) = \frac{1}{m} |\{i | h_i^{-1}(0) \cap S \neq \emptyset\}|.$$

Lemma 6: Let m be $\lceil -\frac{72k^2}{\varepsilon^2} \ln \frac{1}{42} \rceil$, then $\Pr[|X(\mathcal{H}^m) - p| > \frac{\lambda}{2}] \leq \frac{1}{21}$

Proof: See Appendix C for proof. \square

Now we know that if we get a family of t -wise independent hash functions of size m , the estimate \hat{p} can be close to p within $\frac{\lambda}{2}$. Hence, we put all the above lemmas together, an ε estimation is given by the following theorem.

Theorem 2 (Epsilon Approximation Bound): Let k be 7, t be $\lceil \frac{\log \frac{2}{\varepsilon}}{\log 5} \rceil$, and the size of subset \mathcal{H}^m be $\lceil -\frac{3528}{\varepsilon^2} \ln \frac{1}{42} \rceil$, then $\Pr[|\hat{N} - N| \leq \varepsilon N] \geq \frac{11}{21}$.

Proof: By theorem 1, let $k = 7$, the constant estimation in the first phase gives an error probability of $\frac{3}{7}$ at most, as $\frac{N}{k} \leq$

$C' \leq kN \Rightarrow 2k \cdot \frac{N}{k} \leq 2k \cdot C' \leq 2k \cdot kN \Rightarrow 2N \leq C \leq 2k^2 N$ with the probability of $1 - \frac{3}{7}$. Combing lemma 5 and lemma 6, we know that the estimate \hat{p} of q gives the error probability of at most $\frac{1}{21}$, since $|p - q| \leq \frac{\lambda}{2}$ holds when $t = \lceil \frac{\log \frac{2}{\varepsilon}}{\log 5} \rceil$ and $|\hat{p} - p| \leq \frac{\lambda}{2}$ with the probability at least $\frac{20}{21}$. Therefore, the union bound that the probability of as least one of the two events happening is at most $\frac{3}{7} + \frac{1}{21} = \frac{10}{21}$. This is sufficient to establish theorem 2. \square

E. Boosting Success Probability

The theorem 2 shows that an ε -estimate \hat{N} can be given with the probability at least $\frac{11}{21}$. But this success probability does not seem very impressive. To meet the requirement of some high standard applications, it may need to be able to succeed with a probability arbitrarily close to 1, i.e., δ can be arbitrarily close to 0.

We independently select f hash subsets \mathcal{H}^{m_i} ($1 \leq i \leq f$) from a family \mathcal{H} of t -wise independent hash functions. Let \hat{N}_i be the estimate for each subset \mathcal{H}^{m_i} . Then we use \hat{N} to denote the median of $\hat{N}_1, \dots, \hat{N}_f$. Thus, we can define random variables as

$$x(\mathcal{H}^{m_i}) = \begin{cases} 0 & \text{if } |\hat{N}_i - N| \leq \varepsilon N \\ 1 & \text{otherwise} \end{cases}$$

and $X = \sum_{i=1}^f x(\mathcal{H}^{m_i})$.

Theorem 3 (Delta Approximation Bound): For any δ between 0 and 1, there is an $f = \mathcal{O}(\log \delta^{-1})$ ensuring that $\Pr[|\hat{N} - N| \leq \varepsilon N] \geq 1 - \delta$.

Proof: From theorem 2, we know that $x(\mathcal{H}^{m_i})$ takes 1 with the probability at most $\alpha = \frac{10}{21}$. So we can assume that $\mathbf{E}[x(\mathcal{H}^{m_i})] = \alpha < \frac{1}{2}$ and $\mathbf{E}[X] = f\alpha$. If X is less than $\frac{f}{2}$, we can see that $|\hat{N}_i - N| \leq \varepsilon N$ definitely holds since \hat{N} is the median of $\hat{N}_1, \dots, \hat{N}_f$. Thus, if the event $X \geq \frac{f}{2}$ happens with the probability at most δ , the argument in the above theorem is correct. Towards this, by the Chernoff bound, we have

$$\begin{aligned} \Pr[X \geq \frac{f}{2}] &= \Pr[X - \mathbf{E}[X] \geq \frac{f}{2} - \mathbf{E}[X]] \\ &\leq \Pr[|X - \mathbf{E}[X]| \geq \frac{f}{2} - \mathbf{E}[X]] \\ &= \Pr[|X - \mathbf{E}[X]| \geq \frac{f}{2} - f\alpha] \\ &= \Pr[|X - \mathbf{E}[X]| \geq \frac{1-\alpha}{2} \cdot f\alpha] \\ &\leq 2e^{-\frac{(\frac{1-\alpha}{2})^2}{3\alpha^2} \cdot f\alpha} \leq \delta. \end{aligned}$$

Therefore, if we set $f = \lceil \frac{3\alpha^2}{(\frac{1-\alpha}{2})^2} \ln \frac{2}{\delta} \rceil = \lceil 1200 \ln \frac{2}{\delta} \rceil = \mathcal{O}(\log \delta^{-1})$, we can make $\Pr[X \geq \frac{f}{2}] \leq \delta$, and then the complement event $X < \frac{f}{2}$ happens with the probability at least $1 - \delta$. \square

F. Complexity Analysis

From the before, in the first phase, the time slots needed are $\mathcal{O}(\log \log n)$. In the second phase, it needs $\mathcal{O}(\varepsilon^{-2})$ time slots. By theorem 3, it also requires $\mathcal{O}(\log \delta^{-1})$ independent estimation rounds. Therefore, the total time complexity is

$\mathcal{O}((\log \log n + \varepsilon^{-2}) \log \delta^{-1})$, which is nearly constant for a given (ε, δ) . More importantly, compared with previous approaches our final result is a rigorously bounded (ε, δ) estimate. Note that this efficiency is only within a small $\mathcal{O}(\log \frac{1}{\varepsilon})$ factor from the theoretical lower bound in [12].

V. IMPLEMENTATION ISSUES

Hardware Requirement: The RPC algorithms require the programmability on both readers and tags. For readers, the programmability is easy to achieve since both software radio defined readers [25] and commercial-off-the-shelf readers, are able to support user-defined commands. For tags, while being unable to be supported by off-the-shelf C1G2 tags, the RPC can be implemented by programmable passive and active tags, such as WISP or OpenBeacon [26].

Thanks to advances of hash function designs for ultra low-power devices including passive tags [27], the hardware implementation of many complicated hash functions, e.g., AES-128, SHA-256, and universal hashing, become easier. In our case, we can employ the Weighted NH-Polynomial with Reduction (WH) method in [27]. WH exploits the same register to hold the hash of previously processed blocks, which obviates the need for extra temporary registers and results in the perfect serialization. In particular, under 0.13 μm logic process, the total power consumption of WH is 11.6 μW at 500 KHz, of which the dynamic power consumption is 2.26 μW and the leakage power consumption is 9.4 μW . For a passive tag that consumes on average 600 μA at 1.8 v [28], this power consumption is fairly acceptable as it only amounts to 1.07% of the total power consumption of a passive tag.

Another point worth noting is that due to the limited power supply of the daughterboard of USRP (e.g., only 200mW for the RFX900 daughterboard), the reading range is limited to tens of centimeters, making the test of a large number of tags infeasible. A possible solution is to use an external RF amplifier to increase the power of transmitted signals. Two important things deserve careful attention for the above solution. i) Legality. A radio-related certificate is required in most countries to get and operate amplifiers. ii) Safety. Significant RF power needs to be treated with utmost respect to ensure the operator's safety.

Programmability & t-Wise Independent Hash Function: While we realize that the required programmability may affect large-scale applicability due to the cost, we believe as more and more realworld applications and new programmable and configurable RFID architectures are coming out [29], the cost of programmable tags shall decrease dramatically in the near future. Besides, the theoretically achievable bound provided in this paper can be a useful guide for designing efficient network protocols in many backscatter networks, as the cardinality of tags is such a fundamental parameter.

C1G2 Compatibility: As we show that the RPC requires slight updates to the C1G2 protocol. Actually, most existing solutions are not fully compliant with the C1G2, such as [7], [8], [9], and [12]. There are many reasons for this. One important reason is that the C1G2 was designed purely for the identification purpose many years ago, exposing quite

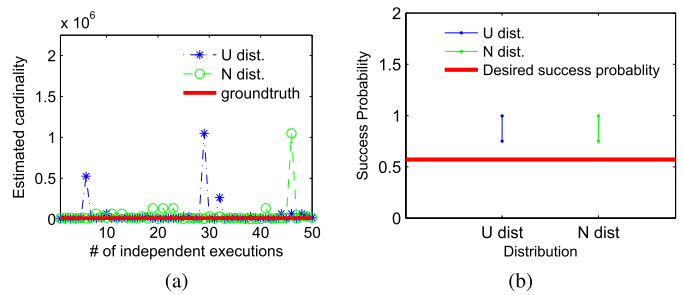


Fig. 4. Quality of rough estimates in RPC. (a) rough estimates. (b) Binomial test of rough estimates.

limited configuration space for other applications, including RFID estimation, missing-tag applications. Another reason is each slot in the C1G2 usually takes at least 16 bits, which is quite inefficient as compared to the single-bit slot used in the RPC and many other state-of-the-art schemes [9], [12].

VI. EVALUATION

We evaluate the performance of the RPC under extensive simulations. First, we study the estimation accuracy of the RPC. Then we compare the RPC with three state-of-the-art methods, ZOE, ART, and SRC with data under different distributions.

A. Setup and Metrics

We use the settings in Figure 1 as default, unless otherwise specified. We assume the communication between tags and the reader is reliable. By default, we take 400 runs and report the average. Besides the accuracy requirement of relative error $\varepsilon = |\frac{N-\hat{N}}{N}|$ and error probability δ , we also include two other metrics, standard deviation, $\sigma = \sqrt{E[(\hat{N} - N)^2]}$, and normalized standard deviation, $\sigma_n = \frac{\sigma}{E[N]}$.

B. RPC Investigation

Quality of Rough Estimates: As shown in Figure 4a, some rough estimates are good and some are not for both distribution. But note that the RPC only requires the rough estimate C to satisfy

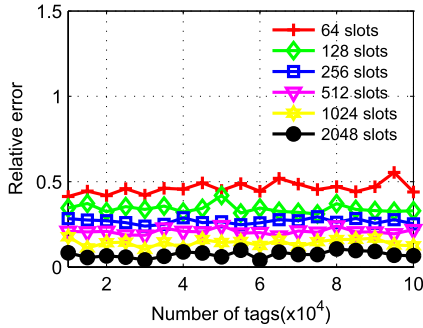
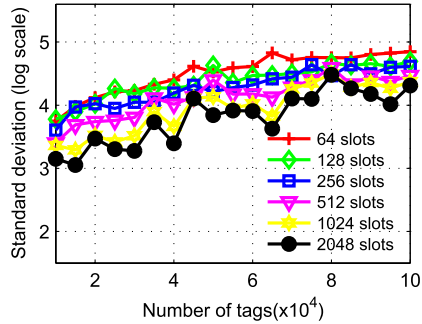
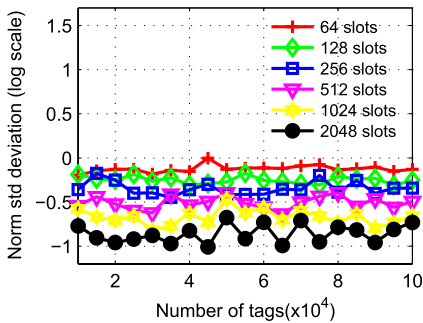
$$\Pr\left[\frac{1}{k} \leq \frac{C}{N} \leq k\right] \geq 1 - \frac{3}{k}.$$

In the general protocol implementation, we set $k = 7$. So the RPC's rough estimates should follow

$$\Pr\left[\frac{1}{7} \leq \frac{C}{N} \leq 7\right] \geq \frac{4}{7}.$$

To test whether rough estimates fulfill the above requirement, we conduct a Binomial test and report the estimated probability intervals. As shown in Figure 4b, we observe that both the rough estimates from the uniform and normal distribution meet the goal. This also agrees with our analysis that the RPC can provide rigorously bounded rough estimates with any data distribution.

RPC With Different Frame Sizes: Next, we study how the RPC performs with different frame sizes. In Figure 5,

Fig. 5. Relative error of estimate \hat{N} Fig. 6. Standard deviation of estimate \hat{N} Fig. 7. Normalized std deviation of estimate \hat{N}

we can see that as the frame size increases, the relative error is getting smaller. In particular, with only 256 time slots, the RPC maintains the relative errors around 0.25. The relative error is reduced to around 0.06 when 2048 time slots are used. Figure 5 also shows that the relative errors are insensitive to the number of tags. In other words, the RPC can obtain accurate estimates in near-constant time for any size of tags, without any priori about the actual number of tags. We examine standard deviations and normalize standard deviations in Figure 6 and 7, respectively. Figure 6 demonstrates that the larger frame size effectively diminishes the standard deviation of estimates. As illustrated in Figure 7, we again see that the number of tags has little influence on the normalized standard deviation. In particular, using 512 time slots, the normalize standard deviations are mostly between 0.2 to 0.3.

C. Performance Comparison

We compare the RPC with three state-of-the-art schemes, ZOE,⁸ ART, and SRC, in terms of actual relative error. We synthesize 10,000 tagIDs from four typical distributions: a uniform distribution in range $[0, 2^{32}]$; a normal distribution with $\mu = 2^{15}$, $\sigma = 2^{12}$; a poisson distribution with $\lambda' = 10^7$; an exponential distribution with $\lambda' = 10^5$. Note that the types and parameters of distributions are just representatives of different data, which is by no means exhaustive. Each of four methods is executed for 50 independent times. As shown in Figure 8, we can see that with the uniform distribution, all methods behave well except several outliers from ZOE and ART, which are due to not rigorously bounded rough estimates.⁹ With other three non-uniform distributions, ZOE, ART, and SRC fail to meet the desired relative error. As the analysis in section III, it is mainly because the largely skewed rough estimates in the non-uniform distributions seriously affect the accuracy of final estimates. On the contrary, the RPC accomplishes its goal in those non-uniform distributions. The reason is that differing from prior schemes, the RPC do not assume uniform random data or perfect hash functions that can make any data into uniform random data. We can conclude that the rigorously bounded rough estimate and final estimate make the RPC achieve its goal and insensitive to data distributions.

Also, we compare the time efficiency of the RPC with other schemes. Specifically, we examine this comparison in two cases: (1) the execution time of different schemes with the same predefined estimation accuracy ε ; (2) the actual estimation accuracy of different schemes with the same execution time under various distributions. For case 1, we vary target relative errors at 0.01, 0.03, and 0.05, and fix the error probability at 0.01. The results are shown in Figure 9a. First, we observe that as the ε requirement increases, the execution time decreases, which shows tradeoffs between execution time and accuracy requirement. Second, we observe that the RPC takes more time than the others schemes. This is exactly as we expected since the RPC achieves tradeoffs between time efficiency and estimation accuracy. In particular, the RPC scarifies its time-efficiency for guaranteed accuracy, i.e., the number of time slots needed in the RPC is more than that of state-of-the-art schemes due to the additional overhead brought by the universal hashing. Note that although the former schemes might have better time efficiency than the RPC, *such performance is achieved at the cost of accuracy*, which is proved by aforementioned experiments and theoretical analysis since the perfect hash assumption does not exist in practice. For case 2, we fix the execution time at 50 s, targeted ε at 10% and compare those schemes with three different distributions in Figure 8. The results are shown in Figure 9b. Under the uniform distribution, the relative errors of the RPC, SRC, and ZOE meet the requirement while ART achieves 12% which is a bit over the target 10%. However, for the normal and poisson distributions, only the RPC achieves its goal and other schemes

⁸Since ZOE is the advanced version of LOF and PET, we omit LOF and PET here for brevity. Comparisons with LOF and PET can be found in [10].

⁹This point is also confirmed in work [12].

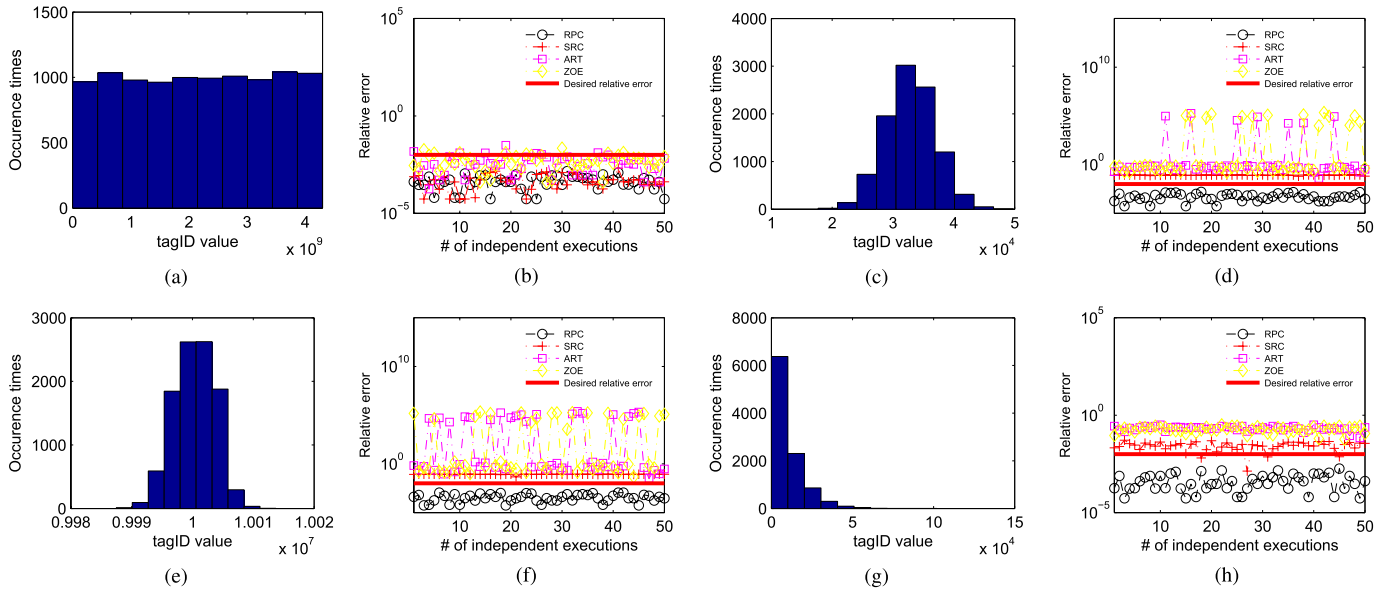


Fig. 8. Relative error investigation of three state-of-the-art schemes, ZOE, ART, and SRC with four different distributions. With the uniform distribution, RPC and SRC consistently meet the desired ε while ZOE and ART have little outliers. With other three non-uniform distributions, ZOE, ART, and SRC all fail to meet the desired accuracy. In contrast, the RPC achieves the accuracy requirement in all the distributions. (a) Uniform distribution $[0, 2^{32}]$. (b) Relative error comparisons under uniform distribution. (c) Normal distribution, $\mu = 2^{15}$, $\sigma = 2^{12}$. (d) Relative error comparisons under normal distribution. (e) Poisson distribution, $\lambda' = 10^7$. (f) Relative error comparisons under poisson distribution. (g) Exponential distribution, $\lambda' = 10^5$. (h) Relative error comparisons under exponential distribution.

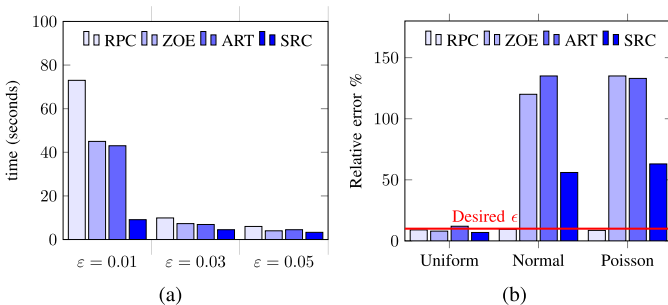


Fig. 9. $N = 10,000$. (a) Comparison of time overhead with the same ε settings. (b) Comparison of actual estimation accuracy with the same execution time (50 s).

fail because of the unbounded rough estimates and the too ideal hash assumption. The results of different execution time settings are quite similar to Figure 9b, which are not included here.

VII. RELATED WORK

A number of probabilistic approaches are designed to quickly obtain the approximated cardinality of tags. Kodialam and Nandagopal [6] first propose probabilistic schemes, Unified Simple Estimator (USE) and Unified Probabilistic Estimator (UPE). Qian *et al.* [7] proposes LOF algorithms, in which the geometric distribution hashing is used to itemize tags in order to fast acquire estimates with $\mathcal{O}(\log n)$ time slots. Zheng and Li [8] further improve the efficiency of estimation to $\mathcal{O}(\log \log n)$ by designing a Probabilistic Estimating Tree (PET). Shahzad and Liu [11] introduce Average Run based Tag estimation (ART) scheme to fast estimate the cardinality.

ZOE is proposed by Zheng and Li [9] to fast estimate the cardinality of tags using only single-slot trials. Most recent work by Zhou *et al.* [12] derives the lower bound for RFID estimation and insightfully points out that the two-phase design is the source gain of most prior methods. Although so much work has been done to efficiently solve this problem, as demonstrated in section III the accuracy itself has yet to be well investigated. In particular, final estimates can largely deviate from the expectation due to skewed rough estimates. The RPC distinguishes itself by providing rigorously bounded results using constructible hash functions and working well with any data distribution.

Recently, several other counting schemes that focus on fine-grained quantities of multiple RFID sets are proposed. A fine-grained batch authentication is introduced to provide accurate estimates of the number of counterfeits and genuines [30]. Gong *et al.* [31] build a generic framework to count tags under arbitrary set expressions. While these approaches efficiently estimate more complex tags quantities (e.g., counterfeits) of multiple tag sets, the RPC concentrates on the overall number of tags and is complementary to them.

Probabilistic counting problems are also extensively studied in data-stream algorithms. Durand and Flajolet [24] first design the well-known FM-Sketch algorithm for approximating the number of distinct elements in data stream. But they assume some ideal properties of hash functions such as the random oracle. Alon *et al.* [32] proposes to use random pairwise independent hash functions to substitute the random oracle. Bar-Yossef *et al.* [33] give three algorithms with different space-time tradeoffs for approximating the cardinality of data streams. Our two-phase solution is inspired by the work in [33], however, those algorithms can not be directly applied

in RFID systems because the model of RFID is very different from that in data streams. In fact, the RPC design, including algorithms, protocols, and the implementation, is specifically devised for RFID systems.

VIII. CONCLUSION

This paper concerns the fundamental problem of tag estimation. By observing that most prior methods fail to meet the desired accuracy due to skewed rough estimates, we propose a rigorous and practical two-phase design for approximating the cardinality and achieve $\mathcal{O}((\log \log n + \varepsilon^{-2}) \log \delta^{-1})$ time-efficiency. In contrast to prior schemes, our method works with any data distribution and uses constructible hash functions. Through analysis and experiment comparisons, we show that our design is able to meet the desired accuracy all the time while other state-of-the-art schemes might fail in some cases. We hope this work could inspire more future work to pay more attention to designing better accuracy-guaranteed schemes for large-scale RFID systems.

APPENDIX A

PROOF OF LEMMA 4

Proof: We prove this by using some well-known bounds and a little calculus. As $C \geq 2N$, hence $C \geq 2$ and $\frac{1}{C} \leq \frac{1}{2}$. Also we know that $(1-x) \geq e^{-2x}$ when $x \leq \frac{1}{2}$. Therefore

$$1 - \frac{1}{C} \geq e^{-\frac{2}{C}} \Rightarrow q = 1 - (1 - \frac{1}{C})^N \leq 1 - e^{-\frac{2N}{C}} \leq 1 - \frac{1}{e}.$$

By definition, $\lambda \leq \frac{1}{e} - \frac{1}{3}$, thus $q + \lambda \leq \frac{2}{3}$, so we can obtain

$$\frac{1}{1 - (q + \lambda)} < 3. \quad (10)$$

Meanwhile, as we know that $\ln(1-x) + x < 0$ when $x < 1$, so when $C > 1$, we can have

$$-\frac{1}{\ln(1 - \frac{1}{C})} \leq C. \quad (11)$$

The calculus we use is that for any continuous function there is $|f(x) - f(\bar{x})| \leq \varepsilon \sup_{y \in (x, \bar{x})} f'(y)$ if \bar{x} is close to x . Hence, for $f(x) = \ln(1-x)$, we know that

$$|\ln(1-x) - \ln(1-\bar{x})| \leq \frac{|x - \bar{x}|}{\max(1-x, 1-\bar{x})}. \quad (12)$$

Combing (10), (11), and (12), it gives that

$$\begin{aligned} |\hat{N} - N| &= \frac{|\ln(1-q) - \ln(1-\hat{q})|}{-\ln(1 - \frac{1}{C})} \\ &\leq C \cdot \frac{|q - \hat{q}|}{\max(1-q, 1-\hat{q})} \\ &\leq 3 \cdot 2kN \cdot \frac{\varepsilon}{6k} = \varepsilon N. \end{aligned}$$

□

APPENDIX B

PROOF OF LEMMA 5

Proof: Let $\mathcal{H}_i \subseteq \mathcal{H}$ be the subset of hash functions that map the i -th element of S into 0. As p is to count the percentage of the number of hash functions that map

some element to 0, to the number of all hash functions, so $p = \frac{|\bigcup_{i=1}^N \mathcal{H}_i|}{|\mathcal{H}|}$. By the inclusion-exclusion, we have

$$p = \sum_i Pr_{h \in \mathcal{H}}[h \in \mathcal{H}_i] - \sum_{i < j} Pr_{h \in \mathcal{H}}[h \in (\mathcal{H}_i \cap \mathcal{H}_j)] + \dots$$

Let T_l to be the l -th term in the above equation. Therefore, for any odd $t > 0$, we can get

$$\sum_{l=1}^{t-1} (-1)^{l+1} T_l \leq p \leq \sum_{l=1}^t (-1)^{l+1} T_l.$$

Since the hash functions in \mathcal{H} are t -wise independent, the probabilities of all $\binom{N}{l}$ subsets can multiple together, i.e.,

$$\sum_{l=1}^{t-1} (-1)^{l+1} \binom{N}{l} C^{-l} \leq p \leq \sum_{l=1}^t (-1)^{l+1} \binom{N}{l} C^{-l}. \quad (13)$$

At the same time, by the binomial expansion we can change the expression q into

$$q = 1 - (1 - \frac{1}{C})^N = \sum_{l=1}^N (-1)^{l+1} \binom{N}{l} C^{-l}$$

and for odd t , we have

$$\sum_{l=1}^{t-1} (-1)^{l+1} \binom{N}{l} C^{-l} \leq q \leq \sum_{l=1}^t (-1)^{l+1} \binom{N}{l} C^{-l}. \quad (14)$$

Since both (13) and (14) are sandwiched, we know that if t is sufficiently large, the difference between two terms q and p can be arbitrarily small. As derived by (13) and (14), the interval of width is $\binom{N}{t} C^{-t}$ and t is $\lceil \frac{\log \frac{2}{\varepsilon}}{\log 5} \rceil$, we have

$$|p - q| \leq \binom{N}{t} C^{-t} \leq (\frac{eN}{tC})^t \leq (\frac{1}{5})^t \leq \frac{\lambda}{2}.$$

□

APPENDIX C

PROOF OF LEMMA 6

Proof: By the definition, we know that $x_{h_i}(\mathcal{H}^m)$ takes 1 with the probability p . Hence, we can derive $\mathbf{E}[x_{h_i}(\mathcal{H}^m)] = p$ and $\mathbf{Var}[x_{h_i}(\mathcal{H}^m)] = p(1-p)$. As $X(\mathcal{H}^m)$ is the sum of m independent variables, we know that

$$\mathbf{E}[X(\mathcal{H}^m)] = \frac{1}{m} \sum_{h_i \in \mathcal{H}^m} \mathbf{E}[x_{h_i}] = \frac{1}{m} \cdot mp = p$$

Then by the Hoeffding's inequality [34], we obtain

$$\begin{aligned} \Pr[|X(\mathcal{H}^m) - p| > \frac{\lambda}{2}] &\leq 2e^{-2m(\frac{\lambda}{2})^2} \\ &= 2e^{-2 \cdot (-\frac{72k^2}{\varepsilon^2} \ln \frac{1}{42}) \frac{\varepsilon^2}{4}} = \frac{1}{21}. \end{aligned}$$

□

REFERENCES

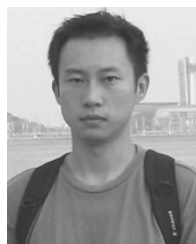
- [1] R. B. Freeman, A. O. Nakamura, L. I. Nakamura, M. Prud'homme, and A. Pyman, "Wal-Mart innovation and productivity: A viewpoint," *Can. J. Econ.*, vol. 44, no. 2, pp. 486–508, 2011.
- [2] W. Gong *et al.*, "Fast and adaptive continuous scanning in large-scale RFID systems," *IEEE/ACM Trans. Netw.*, to be published, doi: 10.1109/TNET.2016.2521333.
- [3] J. Han *et al.*, "GenePrint: Generic and accurate physical-layer identification for UHF RFID tags," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 846–858, Apr. 2016.
- [4] Y. Yin, L. Xie, J. Wu, and S. Lu, "Focus and shoot: Exploring autofocus in RFID tag identification towards a specified area," *IEEE Trans. Comput.*, vol. 65, no. 3, pp. 888–901, Mar. 2016.
- [5] A. Juels, D. Molnar, and D. Wagner, "Security and privacy issues in E-passports," in *Proc. IEEE SecureComm*, Sep. 2005, pp. 74–88.
- [6] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. ACM MOBICOM*, 2006, pp. 322–333.
- [7] C. Qian, H. Ngan, Y. Liu, and L. M. Ni, "Cardinality estimation for large-scale RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1441–1454, Sep. 2011.
- [8] Y. Zheng and M. Li, "PET: Probabilistic estimating tree for large-scale RFID estimation," *IEEE Trans. Mobile Comput.*, vol. 11, no. 11, pp. 1763–1774, Nov. 2012.
- [9] Y. Zheng and M. Li, "Towards more efficient cardinality estimation for large-scale RFID systems," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6, pp. 1886–1896, Dec. 2014.
- [10] W. Gong, K. Liu, X. Miao, and H. Liu, "Arbitrarily accurate approximation scheme for large-scale RFID cardinality estimation," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 477–485.
- [11] M. Shahzad and A. X. Liu, "Fast and accurate estimation of RFID tags," *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 241–254, Feb. 2015.
- [12] Z. Zhou, B. Chen, and H. Yu, "Understanding RFID counting protocols," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 312–327, Feb. 2016.
- [13] X. Liu *et al.*, "RFID estimation with blocker tags," *IEEE/ACM Trans. Netw.*, doi: 10.1109/TNET.2016.2595571.
- [14] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID tags efficiently and anonymously," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
- [15] *EPCglobal Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz–960 MHz*, GS1, Brussels, Belgium, 2008.
- [16] M. Kodialam, T. Nandagopal, and W. C. Lau, "Anonymous tracking using RFID tags," in *Proc. IEEE INFOCOM*, May 2007, pp. 1217–1225.
- [17] (2008). *AMD Hope RFID Data*. [Online]. Available: <https://networkdata.ics.uci.edu/data.php?id=110>
- [18] (2015). *RFID Data of Car License Plates in Nanjing City*. [Online]. Available: <http://www.datatang.com/data/47187>
- [19] J. Liu, M. Chen, B. Xiao, F. Zhu, S. Chen, and L. Chen, "Efficient RFID grouping protocols," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 3177–3190, Oct. 2016.
- [20] J. L. Carter and M. N. Wegman, "Universal classes of hash functions," in *Proc. ACM STOC*, 1977, pp. 106–112.
- [21] A. Mandal and A. Roy, "Relational Hash: Probabilistic hash for verifying relations, secure against forgery and more," in *Advances in Cryptology—CRYPTO*. Berlin, Germany: Springer, 2015, pp. 518–537.
- [22] A. Pagh, R. Pagh, and M. Ruzic, "Linear probing with constant independence," in *Proc. ACM STOC*, 2007, pp. 1–13.
- [23] A. Kirsch and M. Mitzenmacher, "Less hashing, same performance: Building a better bloom filter," in *Algorithms—ESA*. Berlin, Germany: Springer, 2006, pp. 456–467.
- [24] M. Durand and P. Flajolet, "Loglog counting of large cardinalities," in *Algorithms—ESA*. Berlin, Germany: Springer, 2003, pp. 605–617.
- [25] *Gen 2 RFID Tools*, accessed on Dec 7, 2016. [Online]. Available: <https://github.com/nikosl21/Gen2>
- [26] *OpenBeacon*, accessed on Dec 7, 2016. [Online]. Available: <http://www.openbeacon.org/>
- [27] K. Yuksel, J. P. Kaps, and B. Sunar, "Universal hash functions for emerging ultra-low-power networks," in *Proc. CNDS*, 2004.
- [28] A. P. Sample, D. J. Yeager, P. S. Powladge, and J. R. Smith, "Design of a passively-powered, programmable sensing platform for UHF RFID systems," in *Proc. IEEE RFID*, Mar. 2007, pp. 149–156.
- [29] P. Zhang, P. Hu, V. Pasikanti, and D. Ganesan, "EkhoNet: High speed ultra low-power backscatter for next generation sensors," in *Proc. ACM MobiCom*, 2014, pp. 557–568.
- [30] W. Gong, I. Stojmenovic, A. Nayak, K. Liu, and H. Liu, "Fast and scalable counterfeits estimation for large-scale RFID systems," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1052–1064, Apr. 2016.
- [31] W. Gong, H. Liu, L. Chen, K. Liu, and Y. Liu, "Fast composite counting in RFID systems," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2756–2767, Oct. 2016.
- [32] N. Alon, Y. Matias, and M. Szegedy, "The space complexity of approximating the frequency moments," in *Proc. ACM STOC*, 1996, pp. 20–29.
- [33] Z. Bar-Yossef, T. S. Jayram, R. Kumar, D. Sivakumar, and L. Trevisan, "Counting distinct elements in a data stream," in *Randomization and Approximation Techniques in Computer Science*. London, U.K.: Springer-Verlag, 2002, pp. 1–10.
- [34] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Statist. Assoc.*, vol. 58, no. 301, pp. 13–30, 1963.



Wei Gong (M'14) received the B.S. degree from the Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2003, and the M.S. and Ph.D. degrees from the School of Software and Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2007 and 2012, respectively. His research interests include RFID applications, wireless networks, and mobile computing.



Jiangchuan Liu (S'01–M'03–SM'08–F'17) received the B.Eng. degree from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2003. He was an Assistant Professor with The Chinese University of Hong Kong, Hong Kong, from 2003 to 2004. He is currently a University Professor with Simon Fraser University, Vancouver, BC, Canada. His research interests include cloud computing, peer-to-peer systems, multimedia communications, and wireless networking. He is a co-recipient of the Test of Time Paper Award of the IEEE INFOCOM 2015 (Inaugural year) and the ACM Multimedia Best Paper Award, 2012. He is an Associate Editor of the IEEE TRANSACTIONS ON BIG DATA and the IEEE TRANSACTIONS ON MULTIMEDIA, and an Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.



Kebin Liu (M'08) received the B.S. degree from the Department of Computer Science, Tongji University, and the M.S. degree from Shanghai Jiaotong University, China. He is currently pursuing the joint Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiaotong University, and Department of Computer Science and Engineering, Hong Kong University of Science and Technology, under the supervision of Dr. Y. Liu. His research interests include sensor networks and distributed systems.



Yunhao Liu (S'03–M'04–SM'06–F'15) received the B.S. degree in automation from Tsinghua University, China, in 1995, the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University in 2003 and 2004, respectively. He is currently the Chang Jiang Professor with the School of Software and Tsinghua National Laboratory for Information Science and Technology, Tsinghua University. His research interests include wireless sensor network, peer-to-peer computing, and pervasive computing. He is an ACM Fellow.