Computer Communications 110 (2017) 120–132

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

BFCE: A Constant-time cardinality estimator for large-scale RFID systems

Binbin Li[a,b,*], Yuan He[c,**], Wenyuan Liu[b,d,**]

[a] School of Economics and Management, YanShan University, China
[b] School of Information Science and Engineering, YanShan University, China
[c] School of Software, TNLIST, Tsinghua University, China
[d] The Key Laboratory for Software Engineering of HeBei Province, China

ARTICLE INFO

Article history:
Received 29 July 2016
Revised 31 December 2016
Accepted 20 June 2017
Available online 24 June 2017

Keywords:
RFID
Cardinality estimation
Bloom filter

ABSTRACT

Cardinality estimation is of primary importance in many RFID systems and applications. To ensure the time efficiency of estimation, numerous probability-based approaches have been proposed, most of which aim at meeting arbitrary accuracy requirement and follow a similar way of only minimizing the number of required time slots from tags to reader. In this paper, we propose BFCE, a constant-time Bloom Filter based Cardinality Estimator, for particular scenarios where slightly low accuracy and high efficiency are pursued. Comparing with existing approaches, BFCE only needs constant number of bit-slots, and the overall communication overhead is also significantly cut down, as the reader only needs to broadcast a constant number of messages for parameter setting. Results from extensive simulations under various tagIDs distributions shows that BFCE is accurate and highly efficient. In terms of the overall execution time, BFCE is 30 times faster than ZOE and 2 times faster than SRC in average, the two state-of-the-arts estimation approaches.

1. Introduction

Radio Frequency IDentification (RFID) systems have been becoming important platforms for a variety of applications, such as access control [1–3], object identification [4], inventory management [5], transportation and logistics [6], localization [7–9], and tracking [10–13]. Researches on RFID have received wide interests from both industrial and academical communities.

As mentioned in many literatures, the problem of cardinality estimation is of primary importance in many RFID systems and applications. For example, there is usually a need of monitoring the real-time inventory in inventory management systems. Because it is infeasible to get the exact count of all tags in a very short time, most existing works follow the way of probabilistic estimation, such as Probabilistic Estimating Tree (PET) protocol [14], Zero-One Estimator (ZOE) [15], Simple RFID Counting (SRC) protocol [16] and Arbitrarily Accurate Approximation (A^3) protocol [17]. Taking time efficiency as the first-place performance metric, the state-of-the-art approaches can make the estimation in

$$\mathcal{O}\left(\log\log n + \frac{1}{\varepsilon^2}\right) \quad (1)$$

time slots, where n denotes the actual cardinality of tags and ε denotes the confidence interval of estimation requirement.

Two important facts, however, are often neglected. First, most existing works aim at getting estimators which can meet arbitrary accuracy requirement. While for particular applications, such as the entrance of a port, lorries loaded with various RFID-attached commodities pass the entrance one by one. To enhance the unloading efficiency inside the port, it is important to get the tags cardinality as faster as possible, so that the manager can allocate suitable manpower and equipments for the unloading quickly according to the cardinality. Although both accuracy and efficiency should be guaranteed, the estimation efficiency rather than the accuracy takes the first place in this scenarios.

Second, most exiting works simply take the number of required time slots as the efficiency metric, while the number of slots for estimation does not necessarily determine the total time of cardinality estimation. The overall temporal overhead of every communication between reader and tags, is usually a more impacting factor of the estimation efficiency. Taking ZOE [15] as an example, each frame in ZOE only contains one slot, and ZOE totally requires

* Corresponding author at: School of Economics and Management, YanShan University, China.
** Corresponding authors at: School of Information Science and Engineering, YanShan University and School of Software, TNLIST, Tsinghua University, China.
E-mail addresses: ysulbb@gmail.com (B. Li), he@greenorbs.com (Y. He), wyliu@ysu.edu.cn (W. Liu).

http://dx.doi.org/10.1016/j.comcom.2017.06.006
0140-3664/

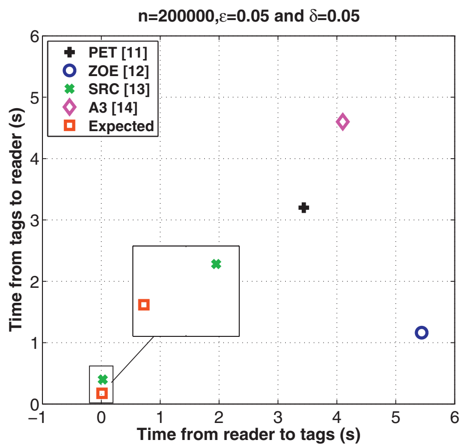


Fig. 1. The design space of RFID cardinality estimation.

at least m slots to get the final estimation result. Note that in ZOE, the reader needs to broadcast a 32-bits random seed for each slot, so that each tag can determine whether or not to participate in the current slot. The temporal overhead of communication from reader to tags (m·32), rather than the overhead from tags to reader (m· 1), is accounted as the major component in the overall execution time of ZOE.

Based on the above discussion, one may immediately raise a crucial but open problem: For a large-scale RFID system and a slightly low accuracy requirement, is there any way to estimate its cardinality in constant time and simultaneously achieve the desired estimation accuracy? Fig. 1 presents the design space of RFID cardinality estimation, comparing our expected result with the existing works.

In this paper, we propose a constant-time Bloom Filter based Cardinality Estimator (BFCE) to estimate the tag cardinality of large-scale RFID systems, when slightly low accuracy and high efficiency estimation requirement are pursued. BFCE consists of two phases, i.e. a rough estimation phase to get a lower bound of cardinality and an accurate estimation phase to get the exact result. In each phase, BFCE lets all the tags construct a w-bits Bloom Filter vector **B** in a distributed manner, using k independent hash functions and a persistence probability p. Intuitively, the number of 0s or 1s in **B** must have relationship with the tag cardinality. By modeling the relationship between the tag cardinality n and the ratio of 0s or 1s in **B**, BFCE can estimate the tag cardinality with very low communication overhead. Specifically, the first phase of BFCE uses a specific persistence probability p_s to get a rough lower bound estimation of n (denoted by $\widehat{n_{low}}$), while in the second phase, BFCE employs the optimal persistence probability p_o with $\widehat{n_{low}}$, so that the final estimation result can meet the accuracy requirement.

Our contributions can be summarized as follows:

- We are the first which take the overall execution time rather than the number of slots as the objective in the RFID cardinality estimation problem. Aim at this objective, we propose a constant-time estimation scheme BFCE for large-scale RFID systems when slightly low accuracy and high efficiency estimation requirement are pursued.
- BFCE only requires the reader to transmit constant number parameters twice in the whole estimation process, and can get the final expected estimation result within constant 1024+8192 bit-slots in just one round, so that the overall execution time of BFCE is constant.
- We validate the proposed protocol through theoretical analysis and conduct extensive simulations under different settings and tagIDs distributions to verify the effectiveness and performance of BFCE. The results demonstrate the advantages of BFCE in terms of both time efficiency and estimation accuracy.

A conference version of this work can be found in [18]. We show much more details in this paper, such as the overview of BFCE, the details of Parameter Settings and the reason why setting w=8192 is sufficient etc. Besides, we also take the channel error into consideration, and propose an extended BFCE which can achieve desired estimation accuracy under unreliable wireless channels.

The rest of the paper is organized as follows. We discuss related works in Section 2. In Section 3, we present the system model and introduce basic concepts of tag cardinality estimation. In Section 4 we elaborate on the design and analysis of BFCE. Then we extend BFCE to robust with unreliable channel and give some discussions in Section 5. Section 6 presents extensive simulations to evaluate the performance of BFCE, and comparison results with recent related works. We conclude this work in Section 7.

2. Related work

As the number of tags may be up to hundreds of thousands and there are always collisions at the reader side, it is infeasible to identify all the tags one by one for the purpose of cardinality estimation. A series of probabilistic approaches have been proposed to achieve the approximate tag cardinality efficiently.

M. Kodialam et al. propose the first cardinality estimation scheme unified probabilistic estimator (UPE) in [19], which needs to distinguish the slots to empty, single or collision slots, and utilizes the number of empty or collision slots in the frame to get the estimation. In [20], M. Kodialam et al. propose another enhanced zero-based estimator (EZB), which takes the average number of zeros in the frame as clues for estimation. In [21], C. Qian et al. propose lottery frame estimation scheme (LOF), which employs the geometric distribution hash functions to itemize all tags in order, so as to make estimation quickly. H. Han et al. propose another tag estimation scheme called first non empty based estimator (FNEB) [22], which is based on the size of the first run of 0s in the frame. With the goal of minimizing power consumption of active tags, T. Li et al. [23] propose an estimation scheme called maximum likelihood estimator (MLE) for active tags. In [24], V. Shah-Mansouri et al. propose a multi-reader tag estimation scheme, but it is based on an unrealistic assumption that any tag covered by multiple readers only replies to one among them. Shahzad et al. [25] propose average run based tag estimation (ART), which uses the average run size of 1s to estimate the tag cardinality.

On the basis of a probabilistic estimating tree (PET), Zheng et al. [14] further improve the estimation efficiency to $O(\log\log n)$ time slots. In [15], Zheng et al. propose another efficient estimate scheme zero-one estimator (ZOE), which also only needs $O(\log\log n)$ time slots. B. Chen et al. ,[16] establish strong lower bounds for both the single-set and multiple-set problems. They also design new simple RFID counting (SRC) which is more time-efficient than existing schemes. In [17], W. Gong et al. propose a new mechanism, arbitrarily accurate approximation (A^3) protocol, to reliably estimate the number of tags with arbitrary accuracy requirement.

Besides, there are also several literatures whose essence are similar with cardinality estimation. W. Gong et al. [1,2] propose two fine-grained batch authentication protocols informative Counting (INC) and wise Counting (WIC), which can authenticate a batch of tags with accurate estimates of the number of counterfeits and genuines. H. Liu et al. [26] first introduce the RFID composite counting problem, which aims at counting the tags in arbitrary set expression and propose a generic composite counting framework (CCF) that provides estimates for any set expression with desired

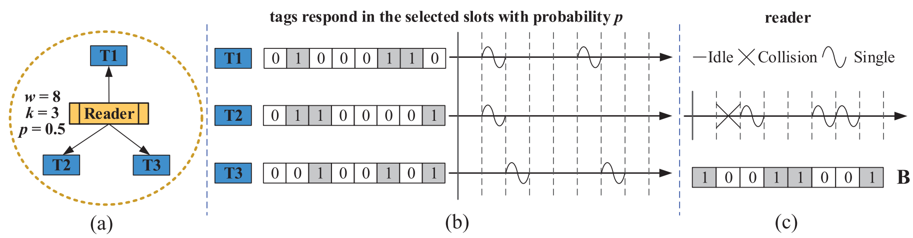


Fig. 2. A simple example of BFCE. (a) The reader first broadcasts w, k, p and random seeds **R**. (b) Each tag responds in the k selected bit-slots with probability p. (c) The reader senses the physical channel in all the w bit-slots, so as to get a vector **B** and does the estimation with **B**.

accuracy. Q. Xiao et al. propose two joint RFID estimation protocol (JREP [27] and M-JREP [28]) to estimate the joint property with bounded error for distributed RFID systems.

Limitations of the existing approaches mainly lie in the following aspects. First and foremost, it does not guarantee to minimize the overall time for estimation, when one simply takes the number of required time slots as indicator of time efficiency. Second, some existing works require prior knowledge of the rough magnitude of cardinality, so that they can reasonably tune the parameter settings for accurate estimation. Last but not least, the accuracy of existing estimators largely depends on the number of repeated rounds, and can not get the accurate estimation within a controllable length of time. The above problems either hurt the efficiency and accuracy of cardinality estimation, or limit the usability of those schemes in practical scenarios. Such facts motivate our work in this paper.

3. Preliminaries

3.1. System model

Consider a large-scale RFID systems which consists of a large volume of tags, one or multiple readers and a back-end server. And we only consider the cases of large-scale tags, e.g. there are more than 1000 tags in the RFID systems, as it is easy and fast to get the exact count of tags by using traditional identification protocols [29,30] when the cardinality is small. Each tag is assigned an unique identification (tagID), with capacity of simple computations and communications with reader through RF signal. We assume that all the tags are within the interrogation zone [31–34]of readers, and all the readers are connected to the back-end server via Ethernet. So the communication overhead between the readers and back-end server can be ignored. If more than one multiple readers are deployed in the interrogation zone, the back-end server can coordinate and synchronize all the readers, so that all the readers can behave consistently, including simultaneously send out command, receive tags' response etc. In other words, we can logically consider these readers as one reader [15].

The communication model between the reader and tags is reader-talks-first and time-slotted, which follows the EPCglobal C1G2 standard [35] To enhance the efficiency of RFID systems, a number of parallel protocols [30,36] have been proposed in recent years. In those parallel protocols, tags are allowed to transmit short information (such as 1bit) in the same slot, the reader only needs to sense the physical channel and distinguish the slots to busy or idle. If there is a busy channel, the reader gets one bit '1'. Otherwise, it gets one bit '0'. For presentation clarity, we call such a slot as bit-slot.

We also adopt the bit-slot mode in BFCE. As described in Fig. 2, the reader initializes the communication by sending out a request message (e.g., estimate), together with a series of parameters, including the length of Bloom Filter w, the number of hash functions k, k random seeds **R** and a persistence probability p. Once receiving the estimation request, each tag uses k independent hash functions to randomly pick k bit-slots, and responds with probability p in each selected bit-slot. The reader then only needs to sense the physical channel and does the estimation with **B**, which represents the status of all the w bit-slots. The clocks of tags are synchronized by the reader's signal.

3.2. Problem description

Consistent with existing approaches, we use two parameters as accuracy requirements of the estimation result: relative error ε, and error probability δ. As we take the time efficiency as the first performance metric, and allow the accuracy requirement to be slightly low, so we assume that both ε and δ are large than 0.05, i.e. $\varepsilon \geq 0.05$ and $\delta \geq 0.05$. Let n be the actual number of tags, we expect an estimation result $\hat{n}$, which satisfies

$$Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta. \tag{2}$$

For instance, if there are actually 10,000 tags in the whole system, with ($\varepsilon = 10\%$, $\delta = 10\%$) approximation, an accurate estimation approach is expected to output the result within the interval [9000, 11000] with a probability of 90% or above.

The goal of cardinality estimation is obtain the approximate number of distinct tags in the region in a fast and accurate manner. First of all, the estimation accuracy must be guaranteed. Second, since the interrogation zone of a reader can be up to 30 feet. The number of tags in the system may easily exceed tens of thousands. Hence the estimation scheme should be time-efficient and scalable as much as possible. Third, different from most existing approaches, which simply abstract the time efficiency with the number of total time slots, we take the stringent time-efficiency, i.e. the overall temporal overhead between reader and tags, as the goal. So the temporal overhead should be minimized. Moreover, passive tags are instantly energized by the reader to carry out extremely limited computations. The estimator should be lightweight enough, so as to support a variety of applications using passive tags.

4. Bloom filter based cardinality estimation

In this section, we introduce the detail of BFCE. Table 1 summarizes the symbols used across this paper.

4.1. Overview

To get an accurate estimation of tag cardinality in the reader's interrogation zone, the reader first tries to construct a bitmap which can well reflect the actual cardinality, and then does the estimation with this bitmap. Specifically, the reader first constructs a w-bit Bloom Filter vector **B**. All the bits in **B** are initialized to be 0s. Then the reader sends out the estimation command, together with several parameters, such as the length of bloom filter vector w, k random seeds R ($k = |\mathbf{R}|$), and a persistence probability p. The reader then waits for the response from tags in the following w bit-slots.

Once receiving the estimation command, each tag randomly selects k different bit-slots with k independent hash functions, whose

Table 1
Symbols used in the paper.

Symbols	Descriptions
n	Actual number of tags
$\hat{n}$	Final estimated number of tags
$\widehat{n_r}$	A rough estimation of n
$\widehat{n_{low}}$	A rough lower bound of n
ε	Confidence interval
δ	Error probability
$H(\cdot)$	Uniform hash functions
$\mathbf{R}$	Random seeds set and $\|\mathbf{R}\| = k$
$\mathbf{B}$	Bloom Filter vector at reader side
w	Length of vector $\mathbf{B}$
k	Number of hash functions
p	Persistence probability
p_s	Specific persistence probability
p_o	Optimal persistence probability
X	A bernoulli random variable
X_i	The observation of ith slot in $\mathbf{B}$
$\overline{\rho}$	The ratio of 1s in $\mathbf{B}$
c	A constant coefficient used for rough estimation
d	A constant determined by δ
q	Channel error rate

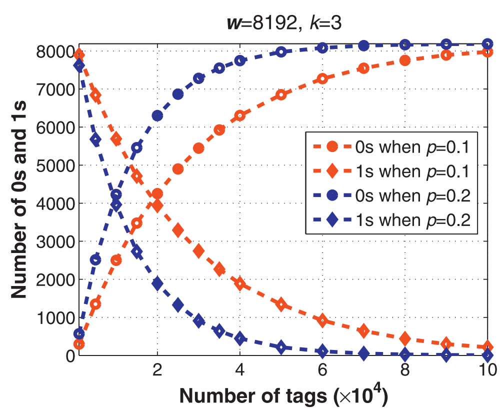


Fig. 3. The feasibility of BFCE.

value ranges in [1,w] and follows an uniform distribution. Then the tag transmits a short signal (e.g. 1 bit) with a probability p in each selected bit-slot.

For the arbitrary ith slot, the reader only needs to sense the physical channel. If the channel is idle, it means there is no tag participating in this bit-slot. Correspondingly $\mathbf{B}(i)$ is set to 1. And if the channel is busy, which indicates that at least one tag transmits in this bit-slot, so $\mathbf{B}(i)$ is set to 0. After w bit-slots, the reader can get a w-bits vector $\mathbf{B}$, which is filled with 0s and 1s.

Intuitively, the number of 0s or 1s in $\mathbf{B}$ is associated with all the parameters as aforementioned. Without distinguishing the difference of hash functions and the random seeds, the number of 0s or 1s is only associated with the number of tags n, the length of bloom filter vector w, the number of hash functions k, and the response probability p. Fig. 3 shows the interrelation between n and the numbers of 0s and 1s in $\mathbf{B}$, when we fix w=8192, k=3 and set p=0.1, p=0.2, respectively. From the figure, we can see that there is linear relationship between the number of tags and the number of 0s or 1s in the vector $\mathbf{B}$. Intuitively it is feasible to accurately estimate the number of tags $\hat{n}$ according to the w-bits bitmap $\mathbf{B}$, k and p. However, there is still an important problem we have to confront. To get a estimation which can meet the accuracy requirement, we should tune all the parameters including w, k, and p to be optimal. Aiming at a constant-time estimator, we can empirically set w and k to be constant (see our analysis later), how to tuning the parameter p will still be a great challenge because we do not have any prior knowledge about the actual tag cardinality. Similar with previous approaches like SRC [16], A^3[17], we also employ the two-phase approach in BFCE as shown in Fig. 4. The first phase, namely the rough lower bound estimation phase, BFCE tries to get a rough lower bound of the cardinality with a specific persistence probability p_s. With this rough lower bound, BFCE then gets the approximate optimal persistence probability p_o in the final accurate estimation phase, so as to guarantee the final estimation result $\hat{n}$ be an (ε, δ) estimation of n. The concrete design will be introduced in the rest of this section.

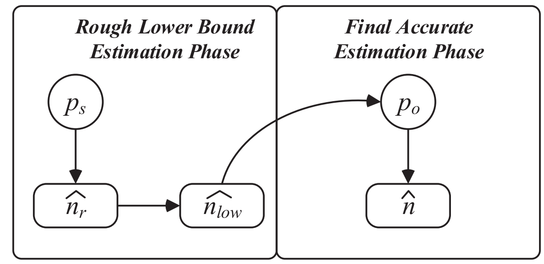


Fig. 4. The overview of BFCE.

4.2. Generic algorithm

Assuming that all hash functions follow uniform distribution, the probability of the arbitrary ith bit in $\mathbf{B}$ being 0 or 1 can be calculated using Theorem 1.

Theorem 1. *Let n be the actual tag cardinality, w be the length of Bloom Filter vector, k be the number of hash functions and p be the persistence probability (i.e., the probability for a tag participates in each selected bit-slot), then,*

$$Pr\{\mathbf{B}(i) = 1 | i \in [1, w]\} = e^{-\lambda}, \tag{3}$$

$$Pr\{\mathbf{B}(i) = 0 | i \in [1, w]\} = 1 - e^{-\lambda}, \tag{4}$$

where $\lambda = \frac{kpn}{w}$.

Proof. Because all hash functions follow uniform distribution in the range [1, w], the probability that a hash function of arbitrary tag being the ith bit-slot in $\mathbf{B}$ is

$$Pr\{H(\cdot) = i | i \in [1, w]\} = \frac{1}{w}. \tag{5}$$

With the persistence probability p, the probability for the tag responds in the ith bit-slot is $\frac{p}{w}$, and the probability for the tag does not respond in ith slot is $1 - \frac{p}{w}$.

In $\mathbf{B}$, the probability of the arbitrary bit i (where $i \in [1, w]$) being 1 is

$$Pr\{\mathbf{B}(i) = 1\} = \left(1 - \frac{p}{w}\right)^{kn}, \tag{6}$$

which means all the kn hash functions of n tags have not selected the ith bit-slot.

Using the approximation of

$$\lim_{x \to \infty} \left(1 - \frac{1}{x}\right)^{x} = e^{-1}, \tag{7}$$

the above equation can be simplified as

$$Pr\{\mathbf{B}(i) = 1\} = \left(1 - \frac{p}{w}\right)^{kn} \approx e^{-\frac{kpn}{w}} = e^{-\lambda}, \tag{8}$$

where $\lambda = \frac{kpn}{w}$.

Correspondingly, the probability of $\mathbf{B}(i)$ being 0, which means more than one tag respond in the ith bit-slot, can be calculated by

$$Pr\{\mathbf{B}(i)=0\}=1-Pr\{\mathbf{B}(i)=1\}\approx 1-e^{-\lambda}. \tag{9}$$

□

We define a random variable X which takes value 1 with probability $Pr\{\mathbf{B}(i)=1\}\approx e^{-\lambda}$ and takes value 0 with probability $Pr\{\mathbf{B}(i)=0\}\approx 1-e^{-\lambda}$. Then we have

$$Pr\{X=1\}=e^{-\lambda}, Pr\{X=0\}=1-e^{-\lambda}. \tag{10}$$

It is not hard to get that the random variable X follows the Bernoulli distribution. Therefore, the expectation and the standard deviation of X are as follows:

$$E(X)=e^{-\lambda}, \sigma(X)=\sqrt{Var(X)}=\sqrt{e^{-\lambda}(1-e^{-\lambda})}. \tag{11}$$

Theorem 2. *Let $\overline{\rho}=\frac{1}{w}\sum_1^w X(i)$ be the average of w independent observations, where $X(i)$ denotes the ith observation of random variable X. Then the tag cardinality can be calculated by*

$$\hat{n}=-\frac{w\ln\overline{\rho}}{kp}. \tag{12}$$

Proof. Assuming that all the trials of X_i $(1\le i\le w)$ are independent as [15] does, we have $E(\overline{\rho})=E(X)$ and $\sigma(\overline{\rho})=\frac{\sigma(X)}{\sqrt{w}}$.

According to the law of large numbers, when w is large enough we have

$$\overline{\rho}=E(\overline{\rho})=E(X)=e^{-\lambda}. \tag{13}$$

So we can estimate λ as follows:

$$\hat{\lambda}=-\ln\overline{\rho}. \tag{14}$$

The observation of $\overline{\rho}$ can be used to estimate the tag cardinality as follows:

$$\hat{n}=-\frac{w\ln\overline{\rho}}{kp}. \tag{15}$$

□

From Eq. (12), we can get that w, k, p and $\overline{\rho}$ all influence the estimation accuracy of $\hat{n}$. Particularly, the estimator will not work when $\overline{\rho}=0$ or $\overline{\rho}=1$, which means that all bits in vector $\mathbf{B}$ are identical (0s or 1s). They are the two exceptions we should avoid.

4.3. Parameter settings

The parameter k in Eq. (12) whose value denotes the number of hash functions is introduced to cope with the various distributions of tagIDs. It can not be too small. A small k will lead to a great variance of $\hat{n}$ because of the pseudo-random of hash functions. On the other hand, it is also time-consuming for tags to get k random numbers if k is too large. The communication overhead between reader and tags will also be increased, as the reader needs to broadcast more random seeds to all the tags. Taking all these factors into consideration, we empirically set $k=3$ in BFCE for a reasonable tradeoff between cost and accuracy.

When it comes to w, similar situations occur. A large w will cause both the exception of all 1s in $\mathbf{B}$ and high temporal overhead for BFCE, while a small w will also cause another exception of all 0s in $\mathbf{B}$. Besides, we should also take the scalability of BFCE into account when determining the value of w. We define $\gamma=-\frac{\ln\overline{\rho}}{kp}$, where k is set to 3. As both p and $\overline{\rho}$ vary in the range (0,1), we can get the variation of γ with different p and $\overline{\rho}$, as depicted in Fig. 5. Then we can find $0.000326\le\gamma\le 2365.9$. According to Eq. (12), we get $0.000326\cdot w\le\hat{n}\le 2365.9\cdot w$. That is to say, the value of w actually bounds the scalability of the estimator. In our work, to achieve a constant-time estimator, we set w=8192, which is scalable enough for most RFID systems. Under this setting, the

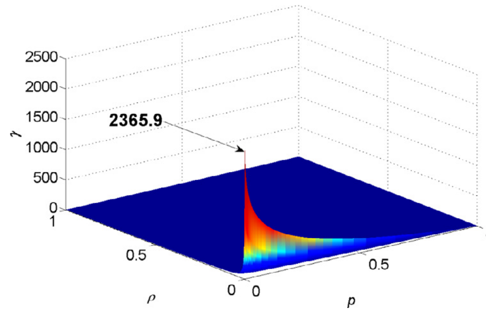


Fig. 5. The variation of $\gamma=-\frac{\ln\overline{\rho}}{3p}$, when both persistence probability p and the ratio of 1s in **B** ρ are varied in the range (0,1).

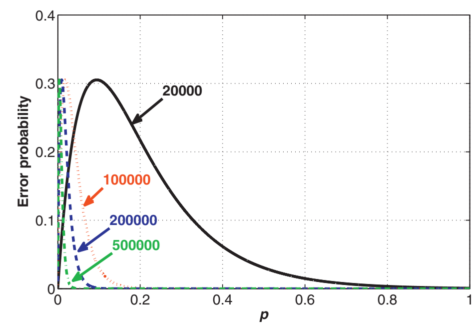


Fig. 6. The error probability of $\overline{\rho}$ with different p and n after 8192 trials.

maximum cardinality that the estimator can estimate exceeds 19 millions, which is sufficient for almost all kinds of application scenarios.

The reasons why w is set at 8192 is two fold. First, the cardinality of tags in a practical scenario is not infinitely large. Setting w to an appropriate value enables one to simultaneously obtain good scalability of the estimator and sufficient capacity to accommodate all the tags in a reader's interrogation zone. Second, the computation overhead of hash functions should also be considered. Setting w=8192 will also greatly reduce this overhead as we show in next section.

Since Bloom Filter is a compact storage of information, one may have doubt on whether w=8192 is enough to get the desired estimation accuracy. According to Chebyshev's inequality

$$Pr\{|\overline{\rho}-E(X)|\ge\varphi\}\le\frac{D(X)}{\varphi^2}, \tag{16}$$

Fig. 6 plots the error probability of $\overline{\rho}$ with different p and n after 8192 trials when φ is set at 0.01. We find that regardless of the value of n, there are always optimal p_o that guarantees the error probability is close to 0. The existing schemes like SRC [16] guarantees the accuracy by fixing the persistence probability and tuning the frame size. Different from this manner, we try to tune the persistence probability of BFCE, so that w=8192 trials are sufficient to guarantee the equation $\overline{\rho}=E(X)$ holds.

Algorithm 1 regulates the behavior of the RFID reader. Depending on which phase the estimator is in, the reader either gets a specific persistence probability for rough estimation of the lower bound of cardinality, or calculates the approximate optimal persistence probability p for accurate estimation (line 4). The reader initiates the estimation process by sending out w, R and p (line 5). After that, the reader senses the channel and records the status into the vector $\mathbf{B}$ (line 6–12). The ratio of 1s $\overline{\rho}$ in $\mathbf{B}$ is calculated after all the w bit-slots end (line 13). Finally, the estimation of tag cardinality is calculated using Eq. (12) (line 14).

Each tag performs simple tasks as regulated in Algorithm 2. In each estimation phase, when receiving an estimation command,

Algorithm 1 BFCE algorithm for reader.

1: $\mathbf{B} = 0$ //8192bits
2: $w \leftarrow 8192$
3: $k = 3$
4: Get p for rough or accurate estimation
5: Initiate the estimation, broadcast w, k, $\mathbf{R}$ and p
6: **for** $i \leftarrow 1$ to w **do**
7: **if** there is no response in the slot **then**
8: $\mathbf{B}(i) \leftarrow 1$
9: **else**
10: $\mathbf{B}(i) \leftarrow 0$
11: **end if**
12: **end for**
13: $\overline{\rho} \leftarrow \frac{1}{w}\sum_1^w \mathbf{B}(i)$
14: **return** $\hat{n} \leftarrow -\frac{w \ln \overline{\rho}}{kp}$

Algorithm 2 BFCE algorithm for RFID tag.

1: Receive w, k, $\mathbf{R}$, p
2: $\mathbf{S} \leftarrow H_j(tagID, \mathbf{R}_j, w),\ j = 1 \cdots k$
3: **for** $i \leftarrow 1$ to w **do**
4: $\mathbf{S} \leftarrow \mathbf{S} - 1$
5: **if** any s in $\mathbf{S}$ equals 0 **then**
6: Respond with a probability of p immediately
7: **else**
8: Keep silent
10: **end if**
11: **end for**

the tag computes the selected S bit-slots with k different hash functions (line 2). If any s in S equals 0, the tag sends a response with a probability of p. Otherwise, it keeps silent (line 3–11).

4.4. Rough lower bound estimation phase

Before performing the final (ε, δ) estimation, we first try to get a rough lower bound of tag cardinality (denoted by $\widehat{n_{low}}$ and $\widehat{n_{low}} \le n$), see the reason in the next section. Nevertheless, since we do not have any prior knowledge, we turn to get a rough estimation (denoted by $\widehat{n_r}$) of n firstly. According to Eq. (12), as long as the ratio $\overline{\rho} \ne 0$ and $\overline{\rho} \ne 1$, we can get an estimation of n. We set a specific persistence probability, e.g. $p_s = \frac{2^3}{2^{10}}$, and observe the received Xs in the coming 32 bit-slots. If all the 32 slots are idle slots, which means there are no response in all slots, we adjust the response probability p_s to $p_s + \frac{2}{2^{10}}$. On the contrary, if all the 32 bit-slots are busy slots, which indicates the probability p_s is too large for the current cardinality, we reduce it to $p_s - \frac{1}{2^{10}}$. This procedure is immediately terminated once both idle and busy slots appear in the 32 bit-slots.

Through several tests, we can get a valid persistence probability p_s quickly. With this p_s, BFCE starts a new round to get $\widehat{n_r}$ according to Algorithm 1. As we only expect a rough lower bound of n, rather than the actual n, we can terminate the estimation at any time (e.g. after 1024 bit-slots). The feasibility of using only 1024 trials of Xs to get the rough estimation is that we assume all the hash functions follow uniform distribution, So the $E(\overline{\rho})$ of 1024 trials theoretically equals to the $E(\overline{\rho})$ of 8192 trials. That is to say, the $\overline{\rho}$ of 1024 bit-slots could approximately represents the $\overline{\rho}$ of 8192 slots. However, there may be difference between $\widehat{n_r}$ and n. So we take $c * \widehat{n_r}$ as the rough lower bound $\widehat{n_{low}}$, where the value of c ranges in (0, 1) and we will validate the influence of different c in Section 4.

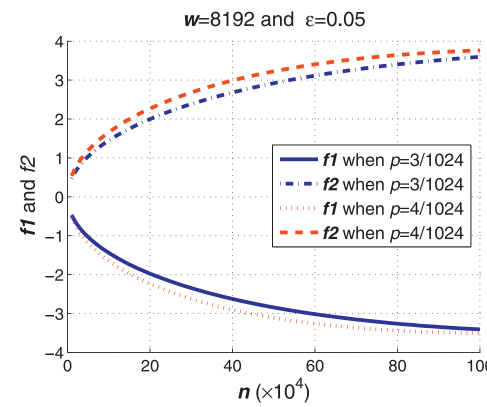


Fig. 7. The monotonicity of f_1 and f_2 when the persistence probability p is small.

4.5. Final accurate estimation phase

Different from previous literatures, which require numerous rounds to get an approximate $\overline{\rho}$, we tune the value of p to get an accurate estimation of $\overline{\rho}$ in just one round, and then get the cardinality estimation result that meets the accuracy requirement $Pr\{|\hat{n} - n| \le \varepsilon n\} \ge 1 - \delta$. Next, We will show how to get the approximate optimal p_o with the rough lower bound estimation $\widehat{n_{low}}$, which we have got in the last section.

Theorem 3. *Given the accuracy requirement of (ε, δ), $\hat{n}$ is an (ε, δ) estimation of n if*

$$f_1 \le -d \text{ and } f_2 \ge d \tag{17}$$

hold, where $f_1 = \frac{e^{-\lambda(1+\varepsilon)} - e^{-\lambda}}{\frac{\sigma(X)}{\sqrt{w}}}$, $f_2 = \frac{e^{-\lambda(1-\varepsilon)} - e^{-\lambda}}{\frac{\sigma(X)}{\sqrt{w}}}$, *and* $d = \sqrt{2} erfinv(1 - \delta)$.

Proof. Because $\lambda = \frac{kpn}{w}$, according to Eq. (12), the estimation accuracy requirement can be represented by

$$Pr\{e^{-\lambda(1+\varepsilon)} \le \overline{\rho} \le e^{-\lambda(1-\varepsilon)}\} \ge 1 - \delta. \tag{18}$$

Based on the fact that the variance of $\overline{\rho}$ is reduced if the experiment is repeated for many times (e.g. w=8192 times), we define a random variable $Y = \frac{\overline{\rho} - \mu}{\sigma}$, where $\mu = E(\overline{\rho}) = e^{-\lambda}$, and $\sigma = \sigma(\overline{\rho}) = \frac{\sigma(X)}{\sqrt{w}}$. Thus, Eq. (18) becomes

$$Pr\{f_1 \le Y \le f_2\} \ge 1 - \delta. \tag{19}$$

By the *central limit theorem*, we know Y is approximately a standard normal random variable. Given a particular error probability δ, we can find a constant d that satisfies

$$Pr\{-d \le Y \le d\} = 1 - \delta. \tag{20}$$

Combining Eqs. (19) and (20), one can guarantee the accuracy requirement $Pr\{|\hat{n} - n| \le \varepsilon n\} \ge 1 - \delta$ if the following conditions are satisfied:

$$f_1 \le -d \text{ and } f_2 \ge d. \tag{21}$$

□

As a large p will cause the exception of all 0s in $\mathbf{B}$, especially when the cardinality is large, BFCE takes the minimal p that satisfies Eq. (17) as the optimal p_o, so we can guarantee that $\hat{n}$ is an (ε, δ) estimation of n. Therefore, the optimal p_o is usually small (e.g. $p = \frac{3}{2^{10}}$), especially when n is large. This can also be verified by Fig. 6. However, it is impossible to get the optimal p_o by solving Eq. (17), as the actual value of n is unknown. To get the optimal value for p_o, we first take both f_1 and f_2 as functions of n, and then analyse their monotonicity. As shown in Fig. 7, when p is small, given $w = 8192$, $k = 3$ and the confidence interval $\varepsilon = 0.05$,

f_1 and f_2 are monotonically decreasing and increasing functions of n, respectively. So we can approximately take f_1 as a monotonically decreasing function of n, and f_2 as a monotonically increasing function of n. Hence, we have the following Theorem.

Theorem 4. *Let $\widehat{n_{low}}$ be a rough lower bound estimation of n, which has been obtained in the previous rough estimation phase, i.e., $\widehat{n_{low}} \leq n$. let p_o be the minimal probability that satisfies*

$$f_1(\widehat{n_{low}}) \leq -d \text{ and } f_2(\widehat{n_{low}}) \geq d, \tag{22}$$

Eq. (17)holds when using this p_o.

Proof. Because f_1, f_2 are monotonically decreasing and increasing functions of n, respectively, we have

$$f_1(n) \leq f_1(\widehat{n_{low}}) \text{ and } f_2(n) \geq f_2(\widehat{n_{low}}). \tag{23}$$

Combining Eqs. (22) and (23), Eq. (17) holds. □

Based on the above analysis, we get the approximate optimal p_o according to Eq. (22), via brute-force calculation (e.g. from $\frac{1}{2^{10}}$ to $\frac{2^{10}-1}{2^{10}}$) with priori knowledge of $\widehat{n_{low}}$, which has been obtained in the first rough estimation phase. One may have the doubt that is there must be one or more p_o. In fact, through our previous experiments, we find that there is always such p_o, as long as the accuracy requirement is not higher than our assumption. A higher accuracy requirement (both ε and δ are close to 0) will cause the failure of getting p_o. Without this p_o, BFCE has to using a specified probability, so that the estimation accuracy requirement cannot be guaranteed. Taken together, we take the minimal p_o that satisfies Eq. (22) as the approximate optimal persistence probability, since p_o is usually small. With this p_o, we can guarantee the estimation result calculated with Eq. (12) is an (ε, δ) estimation.

4.6. Overhead analysis

As we mention in the previous sections, BFCE finishes estimation in just one round, which consists of two phases, namely a rough lower bound estimation phase and an accurate estimation phase. In each phase, the reader only needs to transmit constant number of parameters, and then senses the physical channel to get a bloom vector **B** within 8192 bit-slots. In the rough estimation phase, as we only expect a rough lower bound of the tag cardinality, we may deliberately terminate the phase in just 1024 bit-slots. Hence the temporal overhead of this phase, denoted by t_1, can be calculated by

$$t_1 = (l_w + l_k + k \cdot l_R + l_p) \cdot t_{r\to t} + t_{int} + 1024 * t_{t\to r}, \tag{24}$$

where l_w, l_k, l_R, and l_p are the length of w, k, random seeds, and p, respectively $t_{r\to t}$ is the time for the reader to transmit 1-bit information, t_{int} is the time interval between two consecutive transmissions from the reader to tags or vice versa, and $t_{t\to r}$ is the time for the tags to transmit 1-bit information. Since both w=8192 and k=3 are constant, we can preload them to tags and need not transmit them at runtime. Therefore, t_1 can be simplified as

$$t_1 = (3 \cdot l_R + l_p) \cdot t_{r\to t} + t_{int} + 1024 \cdot t_{t\to r}. \tag{25}$$

Similarly, the temporal overhead of the accurate estimation phase, denoted by t_2, is calculated by

$$t_2 = t_{int} + (3 \cdot l_R + l_p) \cdot t_{r\to t} + t_{int} + 8192 \cdot t_{t\to r}. \tag{26}$$

Based on the above analysis, the overall temporal overhead of BFCE (denoted by t) is the summation of t_1 and t_2, which is

$$t = (6 * l_R + 2 * l_p) \cdot t_{r\to t} + 3 \cdot t_{int} + 9216 \cdot t_{t\to r}. \tag{27}$$

According to the EPCglobal C1G2 standard [35], the time for a reader to transmit one-bit information is 37.76 µs. The time interval is 302 µs. The time for tags to transmit one-bit information is 18.88 µs, namely $t_{r\to t}$=37.76 µs, t_{int}=302 µs and $t_{t\to r}$=18.88 µs. If we restrict the lengths of both the random seeds l_R and the persistence probability l_p to be 32 bits, the overall temporal overhead of BFCE is 0.1847 s. It means that BFCE can rapidly get the final accurate estimation within constant time less than 0.19 s.

5. Discussion

5.1. Estimation with unreliable channel

All the above design and analysis of BFCE is based on the assumption of a reliable and perfect communication channel between reader and tags. However, there are always interference from various wireless source, it is irrational . Most existing protocols can not get the final estimation which satisfies the accuracy requirement over an unreliable channel, the estimation accuracy may decrease dramatically even with a small error rate of the channel. We assume that the error rate of communication channel is time-invariant during the whole short estimation process, and we also assume the false negative rate and false positive rate are both q for simplicity. The value of q can be obtained after a simple test, namely let all the tags respond a specific vector simultaneously, the reader then examines the received vector. Then we extend BFCE to eBFCE, so that it is fault tolerance with unreliable channel.

We denote by $\hat{\rho}_{error}$ the average of w independent observations with error rate of q. Then we have

$$E(\hat{\rho}_{error}) = E(\overline{\rho} - q\overline{\rho} + q(1-\overline{\rho})) = E(\overline{\rho} - q(2\overline{\rho} - 1)). \tag{28}$$

According to Eq. (28), we can compute $E(\overline{\rho})$ as

$$E(\overline{\rho}) = \frac{E(\hat{\rho}_{error}) - q}{1-2q}. \tag{29}$$

So we can extend Eq. (12) and estimate the cardinality as follows:

$$\hat{n}_{error} = -\frac{w}{kp} \ln \frac{E(\hat{\rho}_{error}) - q}{1-2q}. \tag{30}$$

From Eq. (30), we can get that the reliable channel condition is the special case when q=0.

5.2. Implementation of the hash functions

In BFCE, all the tags are required to select $k = 3$ bit-slots using hash functions and respond in the selected slots with a probability of p. Instead of storing many hash functions on resource-constrained tags, a 32-bits random number (denoted by RN in the binary form) is prestored on each tag, prior to the RFID system deployment. To implement the hash functions, the reader generates three uniformly distributed random seeds (denoted by $RS[i]$ in the binary form, where $i \in 1, 2, 3$) at the very start of each phase and broadcasts them to all the tags. When receiving the random seeds, each tag computes the three hash values by

$$H(id) = bitget(RN \oplus RS(i), 13:1), \tag{31}$$

where $\oplus$ denotes the bitwise XOR operation and *bitget* is a function to get the lowest 13 bits of the XOR results. Such a simple method only requires the tags to perform lightweight bitwise XOR computation and *bitget* operations to get the hash values.

5.3. Setting the persistence probability

Then, most existing works implement the persistence probability p by virtually extending frame size for $\frac{1}{p}$ times, i.e., the reader announces a frame size of w/p and terminates the frame after the first w slots. This scheme seems not usable in BFCE, because the value of p is usually small. The size of virtual vector after being extended will be large and slows down the hash function related computations. Instead, we let the reader broadcast the numerator of p (denoted by p_n) rather than the actual p. On receiving p_n, each tag randomly selects several bits (e.g. 10 bits) from the pre-stored random number. If the selected value (in the decimal form) turns out to be smaller than p_n-1, the tag will respond in current bit-slot. Otherwise, the tag will keep silent. In this way, we also get a lightweight p-persistence. All the conclusions and theorems proved before still hold under this setting.

6. Performance evaluation

We conduct extensive simulations under various tagID distributions to evaluate the performance of BFCE. First, we assess the estimation accuracy of BFCE with varied cardinalities of tags under different settings. We then compare BFCE with two state-of-the-arts approaches, ZOE [15] and SRC [16], in terms of estimate accuracy and time efficiency.

6.1. Setup and metrics

We generate three tag ID sets following different distributions as the input data for experiments. As shown in Fig. 8, the first set (denoted by T1) follows uniform distribution between 1 and 10^{15}. The second tag sets(denoted by T2) follows an approximate normal distribution. The third tag sets (denoted by T3) follows normal distribution.

Instead of repeating many rounds of estimation and taking the average as the final outputs, we just take the result of one round estimation as the final result. We adopt a relative metric to evaluate the accuracy, namely

$$\text{Accuracy} = \frac{|\hat{n} - n|}{n}, \tag{32}$$

where $\hat{n}$ denotes the estimation result and n refers to the actual number of tags. A good estimator is expected to return an estimation result close to the actual value. The closer it is to 0, the higher the estimation accuracy is.

To evaluate the time efficiency of different estimators, we take the overall execution time of estimators as the second metric, the execution time of BFCE is the overall time of every communication between reader and tags. According to EPCglobal C1G2 standard [35], any two consecutive transmission from the reader to tags or vice versa are separated by a waiting time of 302 μs. The transmission rate from the reader to tags is 26.5 kb/s. It takes 37.76 μs to transmit 1 bit. Assuming that the length of a random seed is 32 bits, it totally takes 1510 μs for the reader to broadcast a 32-bits random seed. The rate from a tag to the reader is 53 kb/s, it takes 18.88 μs for a tag to transmit 1 bit. So the time for tags to transmit l bits signal is approximately 18.88 l+302 μs.

6.2. Performance under different settings

We first examine the accuracy of BFCE with different parameters settings under all the three tag ID distributions. Fig. 9(a) presents the different estimation accuracy to get (0.05,0.05) estimation. The results with different actual cardinality n under all the three distributions are shown together. Recall that c is the constant coefficient used in the rough estimation phase. From this figure, we can see that the accuracies are very close to 0 even the actual tag cardinality goes up to 1 million, and always can meet the desired accuracy requirement in all cases. This group of experiments reveal that different tagID distributions have little impact on the estimation accuracy.

Then we fix the actual tag cardinality n=500,000, and evaluate the estimation accuracy with different ε and δ. Fig. 9(b) plots the accuracy when ε is varied from 0.05 to 0.3 and other parameters are fixed. Whatever ε is, BFCE always achieves estimation accuracy below 0.04, which is far better than the required ε. We see similar results when δ is varied from 0.05 to 0.3 under all the T1, T2 and T3 distributions as shown in Fig. 9(c).

To further validate the stability of BFCE, we run the BFCE for 100 rounds when n=500,000, ε=0.05 and δ=0.05. Fig. 10 presents the cumulative distributions of the estimation results in T1, T2 and T3, respectively. According to the simulation results, we find

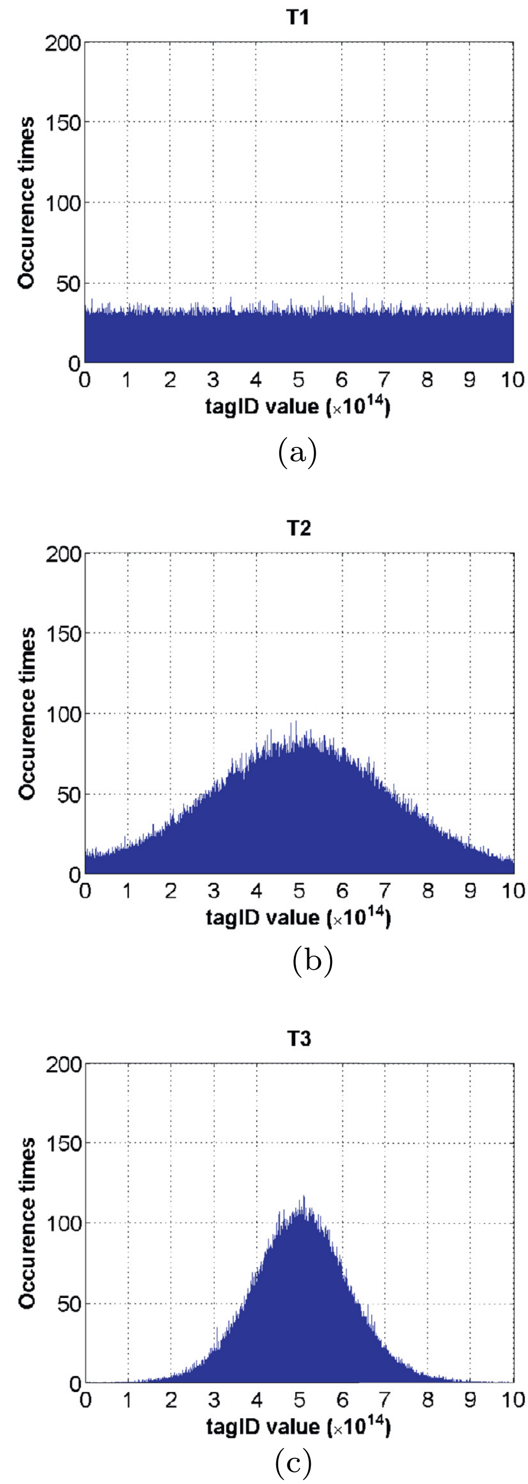


Fig. 8. Three tagIDs sets used in the simulation under different distributions.

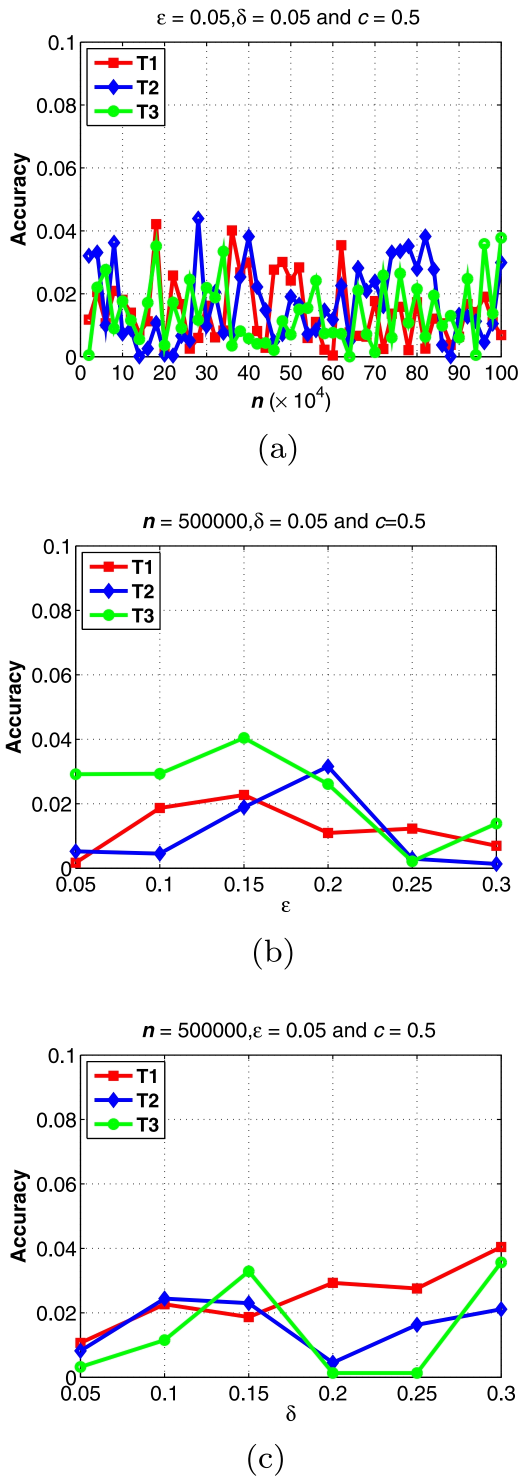


Fig. 9. Estimation accuracy with different n, ε and δ under different tagIDs distributions.

that the estimation results of BFCE are tightly concentrated around the actual cardinality under all the three tagID distributions. It means that BFCE offers more accurate estimation after multiple runs. Compared with previous approaches which need to be executed hundreds of repeated rounds, we can achieve an extremely accurate estimation in no more than 100 rounds.

Then, we validate the impact of the coefficient c (used in the rough estimation phase) on the estimation accuracy. Fig. 11 presents the simulation results when c is varied from 0.1 to 0.9. We can see that there are no large oscillations under all the three distributions whatever c is. Actually, the rough estimation $\widehat{n_r}$ in the first phase of BFCE is already close to the actual cardinality, so there is a high probability that we can guarantee the condition of $\widehat{n_{low}} \le n$ even when c is large. According to Theorem 4, as long as $\widehat{n_{low}} \le n$, the approximate optimal persistence probability p_o calculated by Eq. (22) is close to the expected value, and results in very small variation of the final accurate estimation. Finally, we also verify the robustness of extended BFCE (eBFCE) when the estimation encounters unreliable channel. Fig. 12 plots the accuracy comparison between basic BFCE and eBFCE when the channel error rate q is set to be 10% (both the false negative rate and false positive rate are 10% as aforementioned) under different tagID distributions. The figure suggests that the estimation accuracy of basic BFCE degrades greatly under noisy and unreliable channel, and can not meet the accuracy requirement in most time. While in eBFCE, which takes the channel error into consideration, the estimation accuracy remains reliable in most case, even with 10% channel error.

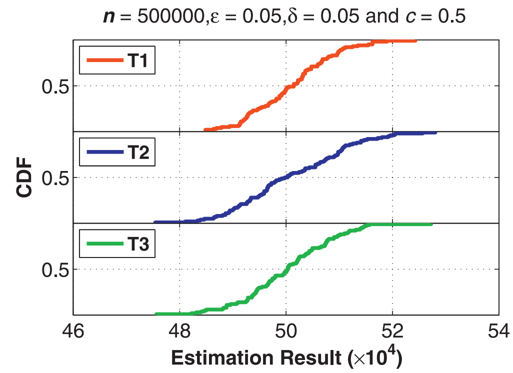


Fig. 10. Cumulative distributions of BFCE under different tagIDs distributions.

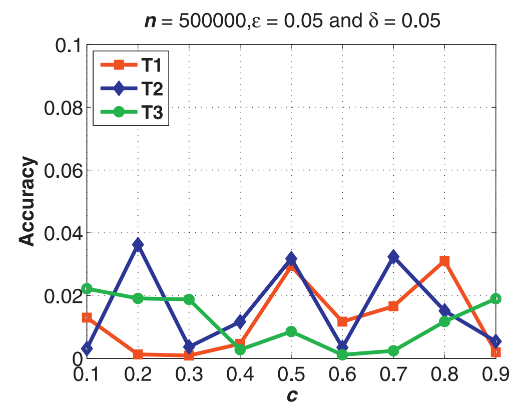


Fig. 11. Estimation accuracy with different c under different tagID distributions.

6.3. Comparison

We then compare the performance of BFCE with two typical state-of-the-art schemes, ZOE [15] and SRC [16]. Note that ZOE requires a rough estimation of n as input to get the final accurate estimation, we slightly modify ZOE, and add a rough estimation phase to ZOE. For simplicity, we invoke LOF [21] and run it for 10 rounds. We then use LOFs output as the rough estimation input of ZOE. To achieve an (ε, δ) estimation with SRC where δ is smaller than 0.2, we repeat the second phase of SRC for m rounds, where

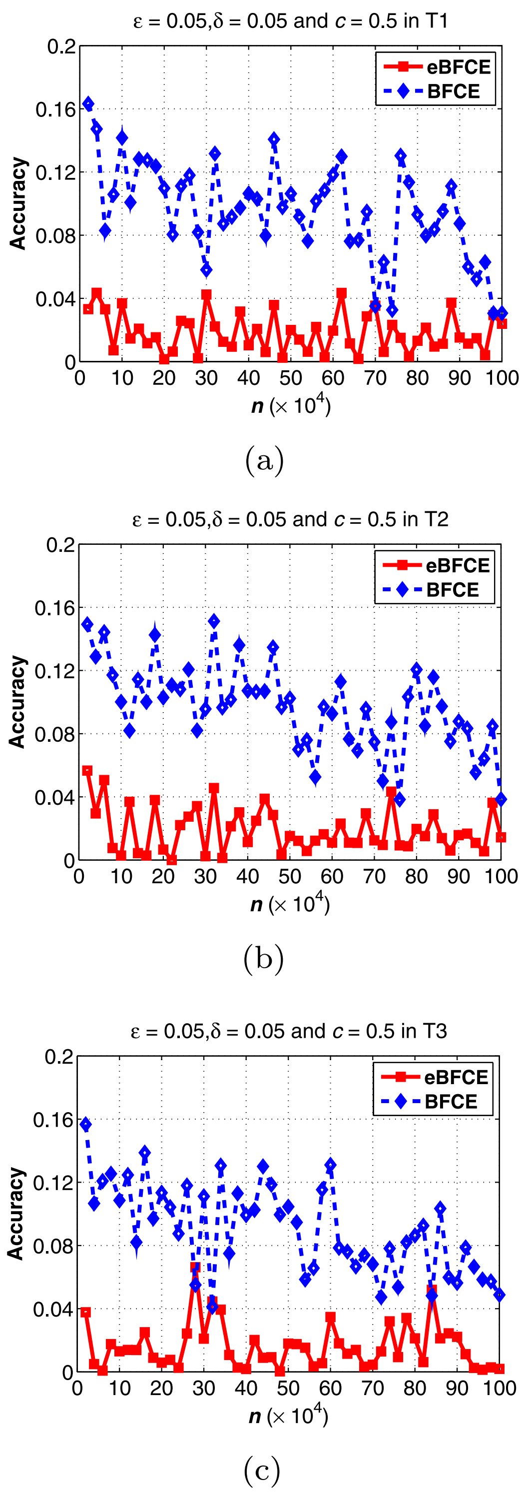


Fig. 12. Estimation accuracy of eBFCE and BFCE with channel error rate 10% under different tagIDs distributions.

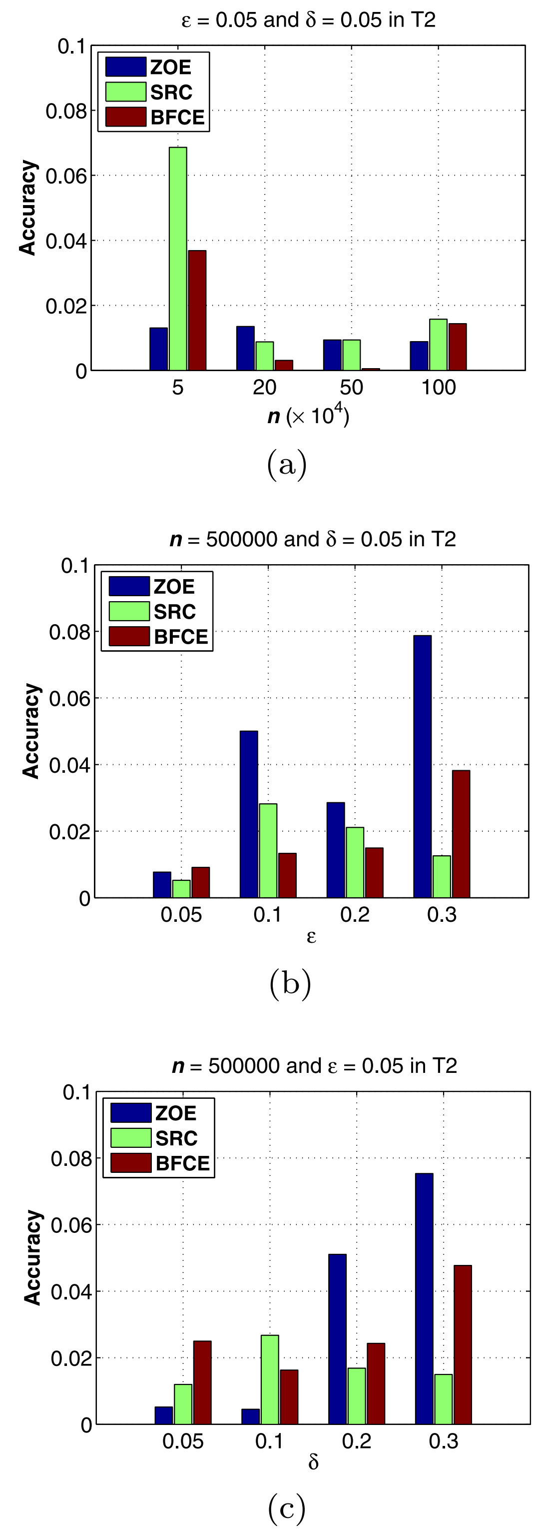


Fig. 13. The comparison of accuracy with different n, ε and δ in one of tagIDs set T2.

m is the smallest integer that satisfies

$$\sum_{i=(m+1)/2}^{m} \binom{m}{i} \cdot 0.8^{i} \cdot 0.2^{m-i} \geq 1-\delta. \tag{33}$$

we conduct performance comparison with all the three tagID distributions. Due to the page limit, Fig. 13 and Fig. 14 present the accuracy and execution time in only one distribution (T2). As shown in Fig. 13, both ZOE and SRC can achieve the desired estimation in almost all the case except several exceptions. Specifically, when n=50,000, the accuracy requirement is set to ε=0.05 and δ=0.05, SRC gets a final estimation 53,430, and the accuracy is about 0.68. Given n=500,000, ε=0.05 and δ=0.3, ZOE outputs an estimation result 537,656, which also exceeds the desired confidence interval. The reason for the exceptions of ZOE and SRC is as follows. The estimation results of ZOE and SRC largely depends on the accuracy of

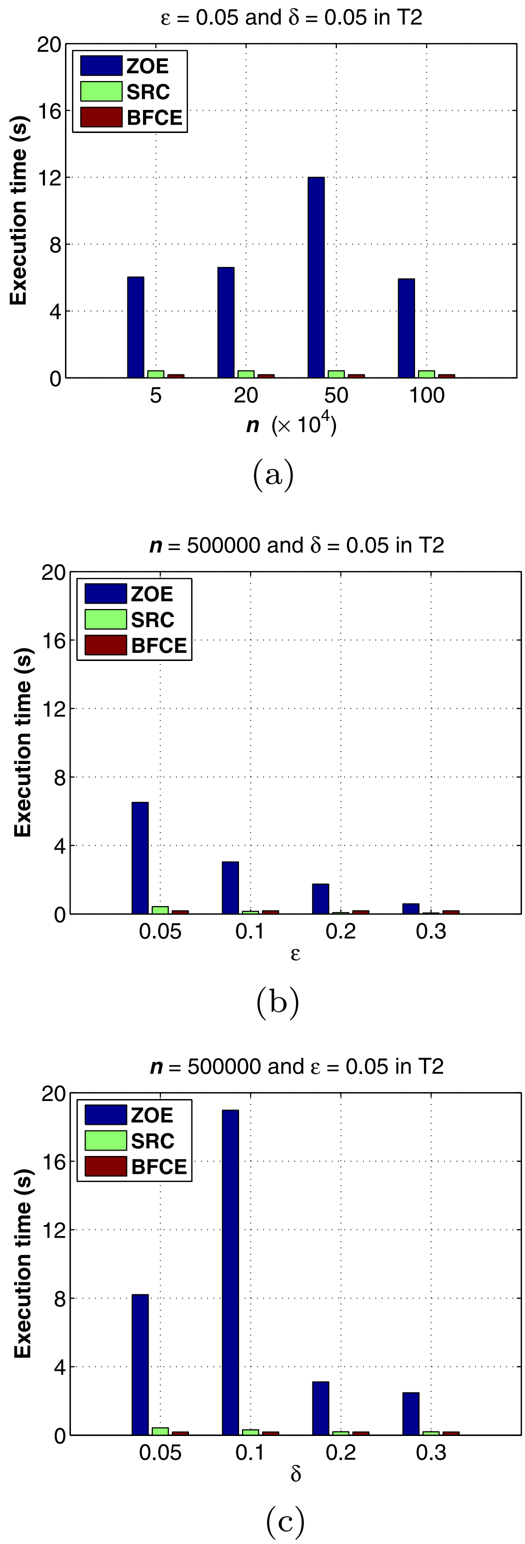


Fig. 14. The comparison of overall execution time with different n, ε and δ in one of tagIDs set T2.

rough estimation, namely the output results of the first estimation phase in ZOE and SRC. In contrast, BFCE always can achieve the desired accuracy in all the cases in only one round, because BFCE's final estimation is only concerned with the rough lower bound of cardinality, rather than an exact value of roughly estimated cardinality.

In Fig. 14, we examine the overall execution time of BFCE, compared with that of ZOE and SRC with different parameters settings in the distribution T2. We can see from the figures that the execution time of ZOE is usually large, about several seconds in all the cases, and even goes up to 18 s in the worst case. There are two reasons for the poor performance of ZOE. First, ZOE needs to continually broadcast 32-bits random seeds for each slot, so the communication time from the reader to tags accounts for the major portion of execution time. Second, the number of required slots of ZOE has great relationship with the output of the rough estimation phase. An estimation that fairly deviates from the actual cardinality will lead to a sharp growth of the required time slots. Although the overall execution time of SRC is much shorter than ZOE's, there are still apparent variance because the execution time of SRC also has relationship with the accuracy of rough estimation. In comparison, BFCE always gets the desired estimation in a constant time, within just 0.19 s, which is 30 times faster than ZOE, and 2 times faster than SRC in average.

7. Conclusion

In this paper, we propose a Bloom Filter based Cardinality Estimation (BFCE) scheme for tag cardinality estimation in RFID systems, when slightly low accuracy and high efficiency estimation requirement are pursued. BFCE achieves guaranteed estimation accuracy in constant time. Moreover, implementing BFCE only requires slight updates to the EPCglobal C1G2 standard [35] and fits a wide variety of application purposes. We conduct extensive simulations to evaluate the performance of BFCE under different settings. The experiment results demonstrate that BFCE outperforms state-of-the-arts schemes in terms of time efficiency and estimation accuracy.

Acknowledgment

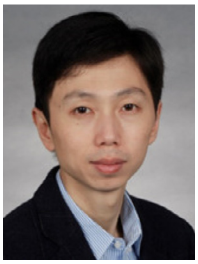
This work is supported in part by National Natural Science Foundation of China (NSFC No. 61672448), National Science Fund for Excellent Young Scientist (No. 61422207) , Natural Science Foundation of Hebei Province (Nos. F2014203062 and No. G2015203242) and Basic Research Foundation of Yanshan University(No. 16LGB016).

References

[1] W. Gong, K. Liu, X. Miao, Q. Ma, Z. Yang, Y. Liu, Informative counting: fine-grained batch authentication for large-scale RFID systems, in: Proceedings of ACM Mobihoc, 2013.
[2] W. Gong, Y. Liu, A. Nayak, C. Wang, Wise counting: fast and efficient batch authentication for large-scale RFID systems, in: Proceedings of ACM Mobihoc, 2014.
[3] M. Chen, S. Chen, RFID technologies for internet of things, Wireless Networks, Springer, 2016.
[4] H. Vogt, Efficient Object Identification With Passive RFID Tags, Springer, Berlin Heidelberg, 2002.
[5] Y. Zheng, M. Li, Fast tag searching protocol for large-scale RFID systems, IEEE/ACM Trans. Netw. 21 (3) (2013) 924–934.
[6] S. Qi, Y. Zheng, M. Li, L. Lu, Y. Liu, Collector: a secure RFID-enabled batch recall protocol, in: Proceedings of IEEE INFOCOM, 2014.
[7] L.M. Ni, Y. Liu, Y.C. Lau, A.P. Patil, Landmarc: indoor location sensing using active RFID, Wireless netw. 10 (6) (2004) 701–710.
[8] J. Wang, D. Katabi, Dude, where's my card?: RFID positioning that works with multipath and non-line of sight, in: Proceedings of the ACM SIGCOMM, 2013.
[9] T. Liu, L. Yang, Q. Lin, Y. Guo, Y. Liu, Anchor-free backscatter positioning for RFID tags with high accuracy, in: Proceedings of IEEE INFOCOM, 2014.
[10] L. Yang, J. Cao, W. Zhu, S. Tang, A hybrid method for achieving high accuracy and efficiency in object tracking using passive RFID, in: Proceedings of IEEE PerCom, 2012.
[11] J. Han, H. Ding, C. Qian, D. Ma, W. Xi, Z. Wang, L. Jiang, J. Zhiping, L. Shangguan, CBID: a customer behavior identification system using passive tags, in: Proceedings of IEEE ICNP, 2014.
[12] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, Y. Liu, Tagoram: real-time tracking of mobile RFID tags to high precision using cots devices, in: Proceedings of ACM MobiCom, 2014.

[13] J. Han, C. Qian, X. Wang, D. Ma, J. Zhao, W. Xi, Z. Jiang, Z. Wang, Twins: device-free object tracking using passive tags, IEEE/ACM Trans. Netw. (2015).
[14] Y. Zheng, M. Li, Pet: probabilistic estimating tree for large-scale RFID estimation, Mob. Comput. IEEE Trans. 11 (11) (2012) 1763–1774.
[15] Y. Zheng, M. Li, ZOE: fast cardinality estimation for large-scale RFID systems, in: Proceedings of IEEE INFOCOM, 2013.
[16] B. Chen, Z. Zhou, H. Yu, Understanding RFID counting protocols, in: Proceedings of ACM MobiCom, 2013.
[17] W. Gong, K. Liu, X. Miao, H. Liu, Arbitrarily accurate approximation scheme for large-scale RFID cardinality estimation, in: Proceedings of IEEE INFOCOM, 2014.
[18] B. Li, Y. He, W. Liu, Towards constant-time cardinality estimation for large-scale RFID systems, in: Proceedings of ICPP, 2015.
[19] M. Kodialam, T. Nandagopal, Fast and reliable estimation schemes in RFID systems, in: Proceedings of ACM MobiCom, 2006.
[20] M. Kodialam, T. Nandagopal, W.C. Lau, Anonymous tracking using RFID tags, in: Proceedings of IEEE INFOCOM, 2007.
[21] C. Qian, H. Ngan, Y. Liu, L.M. Ni, Cardinality estimation for large-scale RFID systems, Syst. IEEE Trans. Parallel Distrib. 22 (9) (2011) 1441–1454.
[22] H. Han, B. Sheng, C. Tan, Q. Li, W. Mao, S. Lu, Counting RFID tags efficiently and anonymously, in: Proceedings of IEEE INFOCOM, 2010.
[23] T. Li, S. Wu, S. Chen, M. Yang, Energy efficient algorithms for the RFID estimation problem, in: Proceedings of IEEE INFOCOM, 2010.
[24] V. Shah-Mansouri, V.W. Wong, Cardinality estimation in RFID systems with multiple readers, IEEE Trans. Wirel. Commun. 10 (5) (2011) 1458–1469.
[25] M. Shahzad, A.X. Liu, Every bit counts: fast and scalable RFID estimation, in: Proceedings of ACM MobiCom, 2012.
[26] H. Liu, W. Gong, L. Chen, W. He, K. Liu, Y. Liu, Generic composite counting in RFID systems, in: Proceedings of IEEE ICDCS, 2014.
[27] Q. Xiao, M. Chen, S. Chen, Y. Zhou, Temporally or spatially dispersed joint RFID estimation using snapshots of variable lengths, in: Proceedings of ACM Mobihoc, 2015, pp. 247–256.
[28] Q. Xiao, S. Chen, M. Chen, Joint property estimation for multiple RFID tag sets using snapshots of variable lengths, in: Proceedings of ACM Mobihoc, 2016, pp. 151–160.
[29] C. Qian, Y. Liu, H. Ngan, L.M. Ni, ASAP: scalable identification and counting for contactless RFID systems, in: Proceedings of IEEE ICDCS, 2010.
[30] L. Kong, L. He, Y. Gu, M.-Y. Wu, T. He, A parallel identification protocol for RFID systems, in: Proceedings of IEEE INFOCOM, 2014.
[31] P. Jankowski-Mihulowicz, M. Weglarski, Determination of passive and semi–passive chip parameters required for synthesis of interrogation zone in UHF RFID systems, Elektron. Elektrotech. 20 (9) (2014) 65–73.
[32] P. Jankowskimihulowicz, M. Weglarski, Interrogation zone determination in HF RFID systems with multiplexed antennas, Arch. Electr. Eng. 64 (3) (2015) 459–470.
[33] L. Wang, X. Qi, J. Xiao, K. Wu, M. Hamdi, Q. Zhang, Exploring smart pilot for wireless rate adaptation, IEEE Trans. Wireless Commun. 15 (7) (2016) 4571–4582.
[34] G. Wang, S. Zhang, K. Wu, Q. Zhang, L.M. Ni, Tim: fine-grained rate adaptation in wlans, IEEE Trans. Mob. Comput. 15 (3) (2016) 748–761.
[35] EPCglobal, EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID: specification for RFID air interface for communications at 860 mhz -960 mhz, Version 2.0.1,2015[Online]. Available: https://www.gs1.org/sites/default/files/docs/epc/Gen2_Protocol_Standard.pdf.
[36] J. Wang, H. Hassanieh, D. Katabi, P. Indyk, Efficient and reliable low-power backscatter networks, in: Proceedings of ACM SIGCOMM, 2012.

Binbin Li received both his BE and ME degree in the School of Computer and Technology, Harbin Institute of Technology, China. He is now an assistant professor in the School of Economics and Management and a Ph.D. student in the School of Information Science and Engineering of YanShan University, China. His research interests include mobile networks, and RFID.



Yuan He is an associate professor in the School of Software and TNLIST of Tsinghua University. He received his BE degree in University of Science and Technology of China, his ME degree in Institute of Software, Chinese Academy of Sciences, and his Ph.D. degree in Hong Kong University of Science and Technology. His research interests include wireless networks, Internet of Things, mobile & pervasive computing. He is a member of the IEEE and ACM.

Wenyuan Liu received the BS and MS degrees in computer science at Northeast Heavy Machinery Institute, China, and the Ph.D. degree in computer science at the Harbin Institute of Technology in 2000. Since 1996, he has been with the School of Information Science and Engineering at the Yanshan University, Qinhuangdao City, China, where he is currently a professor. He is also a special invited researcher at IoT of TNlist, Tsinghua University. His research interests include wireless sensor networks and mobile networks.