Fast Collection of Data in Sensor-augmented RFID Networks

Xin Xie[*][†], Xiulong Liu[*], Weilian Xue[‡], Keqiu Li[*], Bin Xiao[†], Heng Qi[*]
[*]School of Computer Science and Technology, Dalian University of Technology, Dalian, China
[†]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[‡]School of Management, Liaoning Normal University, Dalian, China

***Abstract*—This paper studies the problem of data collection in sensor-augmented RFID networks: how to quickly obtain the error-bounded data from sensor-augmented RFID tags. Existing data collection protocols require all tags to transmit their sensor data to the reader, which incurs a significant transmission overhead in the star-shaped RFID network. To greatly reduce the transmission overhead, this paper proposes a new Sampling-based Information Collection (SIC). By exploring the correlation of sensor data, SIC estimates an error bound based on some randomly-sampled data and the user-defined error threshold. The data within the error bound has no need to be transmitted to the reader, thereby reducing the transmission overhead. We address two challenges to minimize the execution time of SIC, including how to optimize the sample size and frame size. We conduct extensive simulations to evaluate the performance of SIC and compare it with three major related work. The results demonstrate SIC is 1 to 10 times faster than the state-of-the-art solutions.**

I. Introduction

Sensor-integrated Radio Frequency Identification (RFID) tag technology enhances the ability of tags for providing the sensor data to the reader. This feature benefits a lot of applications where more detailed conditions of the products are required. For example, Sensor-integrated tags can be used to monitor the temperature of high-risk foods where bacteria may multiply if the food is stored at the wrong temperature. By monitoring the food's temperature along with the time dimension, we can analyze whether the food is polluted by the bacteria.

With the rapid growth of RFID deployment, efficient data collection from the massive amount of tags is attracting more attention. Nowadays, there are two types of data collection protocols: universal-set collection [1], [2] and certain-set collection [3]. Both approaches have their merits. Universal-set collection protocols return the data of all the tags but are comparatively slower. Certain-set collection protocols [3] are faster but only return the data of some user-defined tags. At the core of these protocols are resolving collisions among tag responses. The commonly used techniques are multiple-hashing [1], Bloom filter [2] and some variants of Bloom filter [2].

There are two fundamental limitations of existing protocols. The first limitation is that existing protocols require complex on-tag computations such as calculating specialized hash functions and parsing a long bit vector, which increases the price of the tag and is far from the Gen2 standard [4]. The second limitation is that universal data collection still occupies the channel too long and blocks other time-sensitive operations such as missing tag identification. The fundamental reason is too many tags need to transmit their data to the reader through a low-rate channel.

A. Problem Statement & Proposed Approach

This paper addresses the problem of error-bounded data collection in RFID systems. It can be formally defined as follows: Let $I = \{i_1, \cdots, i_{N_I}\}$ represent the set of IDs of the integrated tags and $X_I = \{x_1^I, \cdots, x_{N_I}^I\}$ be the sensor data of integrated tags. Knowing exactly IDs in I, our objective is to design an efficient data collection protocol using which a reader should quickly obtain all the sensor data X_I with the error threshold ϵ, which means the data obtained by the reader, $\widehat{X}_I = \{\widehat{x}_1^I, \cdots, \widehat{x}_{N_I}^I\}$, should meet the following requirements: $|\widehat{x}_j^I - x_j^I| < \epsilon, \forall i_j \in I$. The granularity provided by such error-bounded data is more than sufficient, especially considering that the sensors are rarely 100% accurate.

Error-bounded data collection problems have been widely studied in the wireless sensor network literature [5], [6]. Existing solutions usually utilize temporal, spatial or data correlation to predict the sensor data. However, we cannot apply these solutions in RFID network due to the extremely simple tag architecture. As the cost is the barrier for promoting RFID, a tag should be as simple as possible. The prediction models, that are required by the existing solutions, are too complex to be implemented on tags. Besides, tags cannot communicate among themselves, which also invalidates most of existing solutions.

In this paper, we propose a Sampling-based Information Collection (SIC) protocol. Based on some randomly-sampled data, SIC can estimate an error bound within which all the data has no need to be transmitted to the reader and can be approximated by a common value. Intuitively, to achieve this goal, SIC needs to choose an appropriate estimation model, which is the central theme of this paper. Besides, to reduce the modification to current devices, SIC uses the frame slotted ALOHA protocol specified in the Gen2 standard [4] as its MAC layer communication protocol. It consists of five steps as follows: First, the reader initializes a time frame, during which each tag randomly chooses a slot to transmit to the reader. The time frame is terminated until the reader obtains the satisfactory number of sampled data. Second, using these data samples, the reader is able to estimate the error bound based on the appropriate estimation model. Third, the reader inform tags of the estimated error bound, and the tags within the error bound are deactivated. Fourth, the reader initializes another time frame to collect data out of the error bound. Fifth, the reader uses the approximation value to replace the data of tags within the error bound.

There are two key challenges in our work. The first challenge is the degree to which the precision of the estimated error bound. An accurate error bound can minimize the number of tags out of it, thereby reducing the overhead of outlier data transmission. On the other hand, the accurate error bound is obtained at the cost of a large sample size, which means a large overhead of sample data collection. We explore a trade-off in this regard and solve an optimization problem of minimizing the total transmission time. The second challenge is to estimate the number of data out of the error bound, which is required to optimize the frame size for minimizing the execution time. To address this challenge, we apply a light-weight estimation algorithm based on the sampled data without bringing extra communication overhead.

B. Our Contribution

Our major contributions can be summarized as follows:

- We propose a Sampling-based Information Collection (SIC) protocol, which significantly compresses the transmission overhead by collecting the error-bounded data of the tag.
- We present a deep analysis on the optimization of core system parameters, including sample size and frame size, thereby minimizing the execution time of SIC.
- We evaluate the proposed protocol and compare them with several universal-set collection protocols, including Gen2, MIC [1] and BIC [2]. The simulation results demonstrate that SIC is 1 to 10 times faster than the state-of-the-art protocols.

The rest of the paper is organized as follows. Section II reviews the related work. Section III describes the system model and background knowledge. Section IV presents the detailed design of the SIC protocol. Section V introduces the estimation model of SIC. Section VI investigates the parameter optimization to minimize the execution time of SIC. Section VII evaluates the proposed protocol. Finally, Section VIII concludes this paper.

II. Related Work

Reading data from RFID tags is the most fundamental problem in RFID research. For example, in various of applications, the reader is required to collect 96-bit IDs from tags for identification, authentication and inventory verification purposes. The prior research can be classified into two categories: Aloha based [7]–[9] and Tree based [10], [11]. It is also the basic of other specific protocols designed for certain goals, such as missing tag detection [12], localization [13]–[15], tag search [16], [17], inventory management [18]–[20] and object tracking [21]–[23].

Recently, collecting more general, non-ID information from RFID tags attracts much attention as the development of senor-integrated tags. Chen *et al.* proposed a Multiple Hash information Collection protocol (MIC) in [1]. MIC significantly improves transmission efficiency by applying multiple hash functions to resolve the tag collisions in the frame slotted Aloha protcol. To handle the information collection problem in multiple reader scenario, Zhang *et al.* [2] proposed a Bloom filter-based protocol (BIC). BIC can fast identify the tags in every region of the reader by using Bloom filter. Then, each of them is able to fast read the information of tags in its own region by leveraging a well-designed anti-collision technique. Instead of collecting information from all tags, Yan *et al.* [3] proposed a Tag-ordering protocol (TOP), which aims at collecting information from a specific group of tags.

TABLE I
KEY NOTATIONS.

Notations	Descriptions
$I/M/O/U$	the set of integrated tags; sample tags; outlier tags; ordinary tags
N_*	the size of set $*$, $*$ can be $I/M/O/U$
$\lvert\cdot\rvert/\widehat{\lvert\cdot\rvert}$	set size; the estimated set size;
$N(\cdot)/U(\cdot)$	normal distribution; uniform distribution
ϵ	threshold of tolerable error
T_1/T_2	two kinds of waiting time in Gen2 protocol
T_M/T_O	time for sampled data/ outlier data collection
$p/\widehat{p}$	the proportion of ordinary tags/ the estimated p
X_I	integrated information set $X_N = \{x_1^I, \cdots x_{N_I}^I\}$
X_M	sample information set $X_M = \{x_1^M, \cdots x_{N_M}^M\}$
X_O	outlier information set $X_O = \{x_1^O, \cdots x_{N_O}^O\}$
X_U	ordinary information set $X_U = \{x_1^U, \cdots x_{N_U}^U\}$
Q	parameter controls the length of time frame $f = 2^Q$
Q_1/Q_2	the optimal Q in the sample/outlier collection step
$\hat{r}$	approximation of data fills in error bound
E	maximum error of 95% confidence interval of $\hat{p}$

III. System Model

A. Model

Assume a single reader is deployed in the RFID system. Equipped with multiple antennas, the reader has the ability to cover the whole monitoring area. The reader is connected to a host that has a database storing all the IDs of tags in the system. Each tag integrates with a sensor for measuring some physical parameters of the surrounding environment. The reader remotely powers up a population of tags and applies the Gen2 protocol to read their identification and information. Table I lists the symbols used in this paper.

B. Gen2 Anti-collision Protocol

As our solution adopts Gen2 protocol for anti-collisions, the process of Gen2 is detailed in this subsection. As shown in Fig. 1, the protocol begins with a `Query` command, which is issued by the reader to start a frame of f slot. The frame size f is determined by a integer Q (range from 0 to 15) embedded in the `Query` command. Receiving this command, each tag generates a randomly number range from 0 to $2^Q - 1$ to store on the slot counter. The tag whose slot counter equals to zero, respond to the reader immediately. It backscatters a

16-bit random number ($RN16$) within T_1 time, otherwise, this slot is skipped by the reader. Once received the $RN16$, the reader issues an ACK command embedded with the received $RN16$ within T_2 time for acknowledgement. If the $RN16$ is lost due to channel error or collision, the reader respond a NAK within T_2 time. Once received ACK containing its $RN16$, the tag responds its identification (96-bit EPC ID), along with the control information PC and the error detection code CRC-16 within T_1 time.

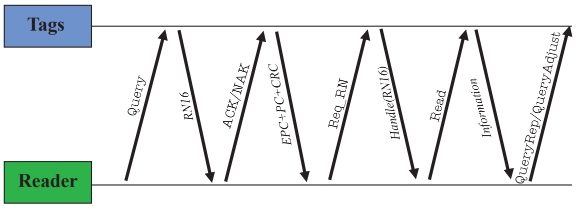

Fig. 1. Handshaking between the reader and the tag in Gen2 protocol.

To enquire on tag data, the reader issues a Req_RN command within T_2 time after receiving tag's ID. The $RN16$ is also embedded in this command. Once received this command, the tag with the same $RN16$ sends a 16-bit $handle$ to the reader, within T_1 time. Then, the reader issues a Read command embedded with the received $handle$, within T_2 time. When receiving the Read command, the tag with the same $handle$ responds its data within T_1 time. At this point, a successful information collection transaction cycle in a slot is done, the reader issues a QueryRep command within T_1 time that instructs tags to decrement their slot counter by 1 to start the new slot transaction cycle to repeat the above process.

During the time frame, the value of Q is updated according to the number of responses in each slot. The reader holds an integer Q and a float Q' in its memory, which is all set to the initial value Q_0. Let $\triangle$ be the adjusted step length defined by the user. If the reader receives multiple responses in the current slot, it updates the Q' by calculating $Q' = Q' + \triangle$; if the reader receives no response in the current slot, it updates Q' by $Q' = Q' - \triangle$; otherwise Q' remains unchanged. Once $\lfloor Q' \rfloor \neq Q$, Q is updated to $\lfloor Q' \rfloor$, and the reader issues QueryAdjust command that instructs tags to recompute the slot counter based on the updated Q.

Our protocol builds on Gen2 protocol because it is reliable and is able to handle transmission errors such as packet loss and bit-error, which are inevitable in practice due to white noise or path loss. Besides, Gen2 is supported by off-the-shelf RFID devices, which makes our protocol can be applied to the current devices with slightly hardware modification.

IV. Protocol Description

In this section, we present a detailed description of the five steps of the proposed Sampling-based Information Collection (SIC). Fig.2 gives a high overview of SIC. The following assumptions are made: a) The reader and the host is connected with a high speed link, being regarded as a whole. b) The reader has limited resource to carry out simple computation. At each execution turn, the steps that are done by the reader are as follows:

1) Use Gen2 protocol detailed in Section III to collect sampled data X_M from randomly picked N_M tags. The reader first initializes a time frame of N_I slots, where N_I is the number of integrated tags. Each tag responds to the reader in a slot whose index is the least significant Q bit of the random number generator RNG on tag, where Q is a integer computed by $Q = \lfloor \log_2(N_I) \rfloor$. Only the responses in the singleton slot (*i.e.*, the slot without collisions) can be successfully received by the reader. When the number of received samples reaches N_M, the reader terminates the Gen2 execution. The sample tags M are deactivated and do not respond to the reader in the following step.
2) Compute the error bound based on the data samples X_M and error threshold ϵ. Let $\widehat{r}$ be the approximation of the data fills in the error bound. The error bound is represented by $[\widehat{r} - \epsilon, \widehat{r} + \epsilon]$. $\widehat{r}$ is computed based on the estimation model, which is detailed in the next section.
3) Tell tags whether they are within the error bound $[\widehat{r} - \epsilon, \widehat{r} + \epsilon]$ by broadcasting a Select command embedded with the upper-bound $\widehat{r} + \epsilon$ lower-bound $\widehat{r} - \epsilon$. The tag whose data within the error bound is called ordinary tag which is represented by U and the tag whose data out of the error bound is called outlier tag, which is represented by O. The ordinary tags in U are deactivated and do not respond to the reader in the following step.
4) Collect data of outlier tags O. The reader initializes a time frame of N_O slots, where N_O is a estimated number of outlier tags, which is computed based on the data samples. The frame is terminated until all the tags have been deactivated.
5) Complement the data of tags in U. As the reader knows the IDs of integrated tags I, sample tags M and outlier tags O after the above four steps, it also get the IDs of ordinary tags U based on the equation $U = I - M - O$. The data of these tags is approximated with $\widehat{r}$ because they are within the error bound.

The actions followed by the tags are relatively sample, each tag only needs to respond to the reader in the first, third and fourth steps, the detailed process are as follows:

1) Report its information to the reader (identical to Step 1 above) compliance with Gen2 protocol.
2) Determine whether it is within the error bound by checking two criteria: 1) $x_j^I < \widehat{r} - \epsilon$ and 2) $x_j^I > \widehat{r} + \epsilon$, where x_j^I is the sensor data of tag i_j. If a tag meets none of the above criteria, it is an ordinary tag, which can be approximated by $\widehat{r}$ according to the user's requirements. Therefore, these tags can be deactivated. (identical to Step 3 above)
3) Report its data to the reader if it is out of the error bound (identical to Step 4 above) compliance with Gen2 protocol.

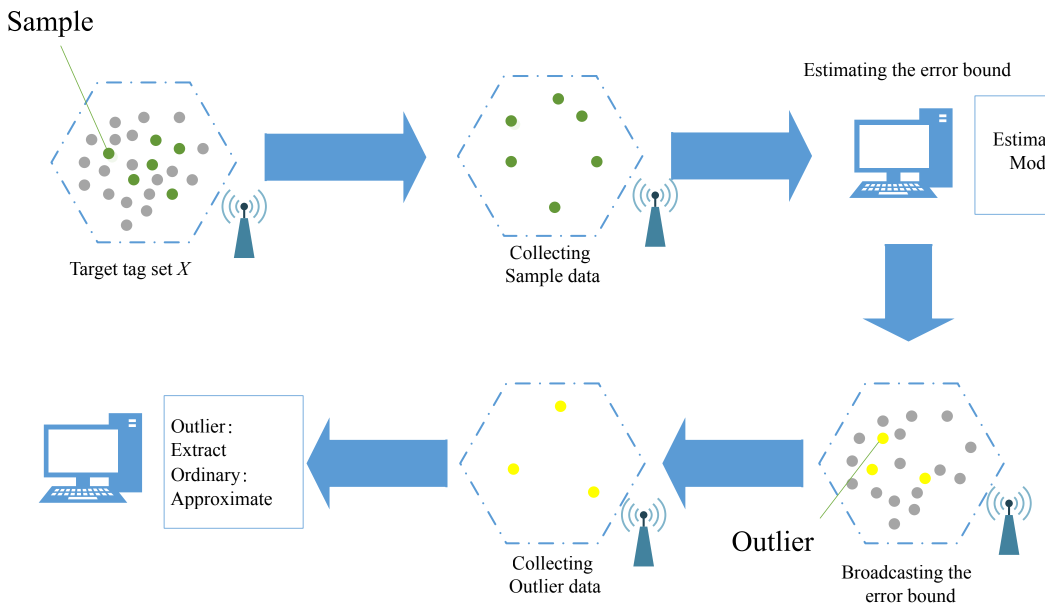

Fig. 2. Workflow of SIC, which consists of five step that are sequentially executed

V. Estimation Model

Our solution to compress the data transmission is based on the insight that if we can bear error-bounded data, we can estimate a error bound $[\widehat{r}-\epsilon, \widehat{r}+\epsilon]$ that covers the maximum number of data values, and use $\widehat{r}$ to replace their exactly value on the reader side, thereby reducing the data transmission of these tags. The estimated error bound is computed based on the sampled data X_M, which is collected in the first step of our protocol.

A. Estimation Algorithm

To maximum the number of tags within the error bound, we adopt a proportion estimation model to derivate the optimal $[\widehat{r}-\epsilon, \widehat{r}+\epsilon]$ that maximum the proportion of data lies in it. The detailed process is shown in Algorithm 1. Assume the data samples $X_M = \{x_1^M, ..., x_{N_M}^M\}$ are sorted in order of increasing. At first, we set the lower-bound to x_1^M, and use *count* to record the number of samples within the error bound $[x_1^M, x_1^M + 2\epsilon]$. If *count* is larger than *max*, we update *max* with *count* and record lower-bound lb as x_1^M. Then, we sequentially set the lower-bound of the interval to $x_2^M, ..., x_{N_m - max}^M$, and update $max = count$ and $lb = x_i^M$ if the number of tags within $[x_i^M, x_i^M + 2\epsilon]$ $(2 \le i \le N_m - max)$ is larger than the current max. Finally, the algorithm return the final lower-bound lb, upper-bound ub and the estimated proportion $\widehat{p}$.

Algorithm 1: Estimating the error bound

```
Input: Sorted Sample data $X_M = \{x_1^M, ..., x_{N_M}^M\}$ and
       the error threshold $\epsilon$
Output: The error bound $[lb, ub]$ and proportion $\widehat{p}$
$max$=0;
for $i$=1; $i \le N_m$; $i$++ do
    $count$=0;
    for $j$=$i$; $x_j^M \le x_j^M + 2\epsilon$; $j$++ do
        $count$++;
    end
    if $(count \ge max)$ then
        $lb = x_j^M$;
        $max$=$count$;
    end
    if $(max \ge N_m - i)$ then
        return $lb = x_j^M, ub = x_j^M + 2\epsilon, \widehat{p} = max/N_M$;
    end
end
return $lb = x_j^M, ub = x_j^M + 2\epsilon, \widehat{p} = max/N_M$;
```

B. Theory Analysis

The theory foundation of proportion estimation model lies in statistical sampling: given a uniform sample from a finite population, what is the proportion estimation's error bound? The approach described in the above is valid whenever the following two conditions are met:

- The sampling method is simple random sampling, and the probability of success is the same for each trial.
- The sample is sufficiently large. According to a frequently used thumb, the size is reasonable as long as $N_m \times \widehat{p} > 0.5$ and $N_m \times (1 - \widehat{p}) > 0.5$

Let p denote the proportion of ordinary tags fills in the error bound. As $N_I \gg N_M$, the sampling process in our protocol can be regarded as a Bernoulli process.

Theorem 1 *The Uniformly Minimum Variance Unbiased Es-*

timation for p is $\widehat{p}$, the sample proportion $\widehat{p} = \frac{max}{N_M}$.

Proof: Let $f(max; p, N_m)$ be the probability of obtaining max ordinary tags in N_m sampled data, it can be expressed as:

$$f(max; p, N_m) = \binom{N_m}{max} p^k (1-p)^{N_m - max} \tag{1}$$

The expectation value of $\widehat{p}$ is therefore given by

$$\begin{aligned} E[\widehat{p}] &= \sum_{k=0}^{m} p \times f(max; p, N_m) \\ &= (1-p)^{N_m} \times (\frac{1}{1-p})^{N_m} \times p = p \end{aligned} \tag{2}$$

Therefore, $\widehat{p}$ is an unbiased estimator of p. The variance of $\widehat{p}$ can be represented by $Var(\widehat{p}) = p(1-p)/N_M$, which equals to the Cramr-Rao lower bound for the variance of unbiased estimators of p. Hence, $\widehat{p}$ is the Uniformly Minimum Variance Unbiased Estimation (UMVUE) of p. ■

VI. Parameters Analysis

Recall from the previous section that the major overhead of SIC is the data transmission between reader and tags. To minimize the execution time, we may reduce the number of responding tags or improve the efficiency of the time frame, which can be achieved by optimizing sample size and time frame, respectively.

A. Optimization of Sample Size

The sample size N_m is a key parameter that determines the number of responding tags and affects the execution time. If N_M is too large, the estimated error bound is accurate but incurs a large overhead of collecting sampled data. On the contrary, if N_M is too small, the estimated error bound is unreliable, which may cover less ordinary tags and incur a large overhead of collecting outlier tags. Essentially, the sample size N_M trades off between the time costs of sample collection and outlier collection. To optimize N_M, we first calculates the time for collection sample tags T_M and outlier tags T_O, respectively. Then, we formulate and solve a constraint optimization problem with minimizing $T_M + T_O$.

1) Overhead of Sampling: Consider that the transmission overhead is always proportional to the number of collected tags, the key challenge is translate into minimize the number of sample tags and the outlier tags. The approximation of estimated $\widehat{p}$ is usually justified by the central limit theorem. The expression is:

$$\widehat{p} \pm z_{1-\alpha} \sqrt{\frac{\widehat{p}(1-\widehat{p})}{N_M}}, \tag{3}$$

where $z_{1-\alpha}$ is the $1 - \frac{1}{2}\alpha$ quantile of a standard normal distribution, α is the error quantile ranges from 0 to 1. When $\alpha = 5\%$, $z_{1-\alpha} = 1.96$, and the error of the estimator $\widehat{p}$ is $E = 1.96\sqrt{\frac{\widehat{p}(1-\widehat{p})}{N_M}}$ with 95% confidence level. Given an a specific E, we can derivate the required sample size as follows:

$$N_M = 1.96^2 \times \frac{\widehat{p}(1-\widehat{p})}{E^2} \tag{4}$$

Therefore, the execution time of the sampling can be expressed as $T_M = N_M \times t'$, where t' represents the average overhead for collecting data from a tag.

2) Overhead of Collecting Outlier: If the estimated proportion $\widehat{p}$ is exactly same with the actual p, the number of outlier tags can be expressed as $N_o = \widehat{p} \times (N_I - N_M)$. However, if the sample size is to small, the error of $\widehat{p}$ can be extremely large because an rough estimated error bound $[\widehat{r} - \epsilon, \widehat{r} - \epsilon]$ covers data of fewer tags. The maximum number of outlier tags caused by inaccurate estimation is:

$$N_O = (1 - p + E)\,(N_I - N_M) \tag{5}$$

The execution time can be expressed as $T_O = N_O \times t'$

3) Joint Optimization: The total execution time is the sum of T_M + T_O, which can be expressed as follows:

$$\begin{aligned} T_M + T_O &= [N_M + (1-p+E)\,(N_I - N_M)] \times t' \\ &= [N_M(p-E) + N_I\,(1-p+E)] \times t' \end{aligned} \tag{6}$$

Let $f(E)$ be the $(T_M + T_O)/t$, we can get the following equation by substituting (4) into the (6) and replacing p with its approximation $\widehat{p}$:

$$f(E) = \frac{1.96^2 \widehat{p}(1-\widehat{p})(\widehat{p} - E)}{E^2} + N_I(1 - \widehat{p} + E) \tag{7}$$

The derivative of $f(E)$ with respect to E is:

$$\frac{\partial f(E)}{\partial E} = \frac{-192\,\widehat{p}^2 + 192\,\widehat{p}^3 + \left(96\,\widehat{p} - 96\,\widehat{p}^2\right)\,E + 25\,E^3\,N_I}{25\,E^3} \tag{8}$$

Let $g(E)$ be the numerator of the equation above, the equation $g(E) = 0$ has at least one solution E among the real number. Let a, b, c, d be the coefficients of $g(E)$, we have $a = 25N_I$, $b = 0$ $c = 96\widehat{p} - 96\widehat{p}^2$ and $d = -192\,\widehat{p}^2 + 192\,\widehat{p}^3$. To distinguish the number of roots, we adopts the following discriminant:

$$\Delta = 18abcd - 4b^3d + b^2c^2 - 4ac^3 - 27a^2d^2 \tag{9}$$

Substituting a,b,c,d into (9), we have:

$$\begin{aligned} \Delta &= -4ac^3 - 27a^2d^2 \\ &= 100N_I(96\widehat{p}^2 - 96\widehat{p})^3 - 16875N_I^2(-192\widehat{p}^2 + 192\widehat{p}^3)^2 \\ &= 96^2 \times 100N_I \times (\widehat{p}^2 - \widehat{p})^2 \times [(96\widehat{p}^2 - 96\widehat{p}) - 675\widehat{p}^2 N_I] \end{aligned} \tag{10}$$

As $\widehat{p}^2 - \widehat{p} < 0$, we have $\Delta < 0$, which means $g(E)$ has only one real root. Let $A = \frac{96\,\widehat{p}^3 - 96\,\widehat{p}^2}{25\,N_I}$ and $B = \frac{32\,(\widehat{p}-1)\,\widehat{p}\,\sqrt{\widehat{p}\,(32 - 32\,\widehat{p} + 225\,\widehat{p}\,N_I)/N_I}}{125\,N_I}$, we can derive the root as follows:

$$E = (A - B)^{1/3} + \frac{32\widehat{p}^2 - 32\widehat{p}}{25N_I(A-B)^{1/3}} \tag{11}$$

The derivative of $g(E)$ is:

$$\frac{\partial g(E)}{\partial E} = 75E^2N_I - 96\widehat{p}^2 + 96\widehat{p} > 0 \tag{12}$$

As $g(E)$ is monotonically increasing according to (12), we derive that $g(E) < 0$ when $E < (A-B)^{1/3} + \frac{32\widehat{p}^2 - 32\widehat{p}}{25N_I(A-B)^{1/3}}$ and

$g(E) > 0$ when $E > (A-B)^{1/3} + \frac{32\widehat{p}^2-32\widehat{p}}{25N_I(A-B)^{1/3}}$. Therefore $f(E)$ is minimized when $E = (A-B)^{1/3} + \frac{32\widehat{p}^2-32\widehat{p}}{25N_I(A-B)^{1/3}}$

As E and $\widehat{p}$ are unknown parameters before collecting sampled data. Our protocol has to dynamic compute their value during the sample collection step. The data is processed in pipelined execution, estimating the proportion $\widehat{p}$, calculating the optimal E and updating the sample size after each 10 sampled data is collected. The collection process is terminated until the number of collected sample tags exceed the latest N_m

B. Optimization of Q

In addition to the optimization of sample size. How to set the appropriately frame size is a remaining problem for minimizing the execution time of SIC. Many of prior work has proved that the efficiency of the time frame is maximized when the frame size equals to the number of responding tags [24]. As the frame size is controlled by a integer Q in Gen2 protocol, to coincidence with it, in this subsection, we present how to set the optimal Q in each collection round.

In the step of collecting sampled data, the frame parameter Q_1 is set to $\lfloor \log_2 N_I \rfloor$ because all the tags in I are ready to respond to the reader. In the step of collecting outlier tags, as only the tags out of the error bound respond to the reader, the frame parameter Q_2 is set to $\lfloor \log_2 N_O \rfloor$. Recall from the previous subsection, N_O is not known in advance but can be estimated based on the sampled data X_M. Based on (5), we have

$$\widehat{Q}_2 = \lfloor \log_2 (1-\widehat{p}+E)(N_I - N_M) \rfloor \tag{13}$$

To evaluate the accuracy of estimator $\widehat{Q}_2$, we calculate the variance of it. Assume the sampled data are randomly picked, each picking can be regarded as independent. Recall that the variance of $\widehat{p}$ can be expressed as $p(1-p)/N_M$, which achieves the maximum value when $p = 0.5$. Therefore, the variance of $\widehat{p}$, $Var(\widehat{p}) \leq \frac{1}{4N_M}$. Then, we can derivate the variance of estimated N_O, $Var(\widehat{N}_O) \leq \frac{(N_I-N_M)^2}{4N_M}$. Using the Taylor series to approximate the moments of the transformed random variable, we can get the following relationship:

$$\begin{aligned} Var(\widehat{Q}_2) &\approx \left[\frac{\partial \log_2(N_O)}{\partial N_O}\right] \times Var(\widehat{N}_O) \\ &\approx \frac{Var(\widehat{N}_O)}{(\ln 2 \times N_O)^2} \approx \frac{1}{2N_M(1-\widehat{p})^2} \end{aligned} \tag{14}$$

C. Impact of Channel Errors

In the real-world environment, the communication channel is error-prone. The white noise may corrupt the message exchanged between the reader and tags, e.g., 0 becomes 1 or 1 becomes 0, which is called bit error. More seriously, some message are even not detected at all due to the path loss. Most of the literature focuses on minimizing the transmission bits or execution time of the protocol. They usually adopt a time efficient data structure, e.g., Bloom filter which is broadcast to the tags. The data structure is required to be correctly received by all tags in the system. However, it is a stringent requirement due to the unavoidable channel errors.

Our protocol inherits the error handing mechanism of Gen2 [4], which packages a CRC(Cyclic Redundancy Code) along with the transmission data. The receiver detects the bit error by verifying the CRC of received message. If it fails the verification of CRC, the whole message is been dropped. The receiver considers it receives a invalid command and follow the action detailed in Gen2 standard [4]. In most cases, if the receiver is a tag, it resets a slots and waits for the subsequent commands; if the receiver is a reader, it terminates the current slot and starts a new slot. Similarly, if the receiver do not receive the message after a period, it takes the similar actions as receiving a invalid command.

Dropping the error message protects the correctness of received messages, but incurs a lot of retransmissions, which prolongs the execution time of our protocol. The increased time is from the following two aspects: First, the path loss and bit errors lead to the infinitive a part of singleton slots, the reader has to start new slots to communicate with tags in these slots which increases the execution time. Second, some ordinary tags may miss the command for classifying outlier tags. These ordinary tags cannot be deactivated and increase the number of tags to be collected, thereby increasing the execution time.

D. The Accuracy of SIC

A major concern of SIC is that it improves the time-efficiency by sacrificing the accuracy of collected data. However, such concern is unnecessary due to the following two reasons: First, SIC indeed sacrifices the precision of some data (*i.e.*, data in X_U). However, these data is most common in the system, which generally refers to the normal condition, containing litter valuable information for objects managcing and tracking. Therefore, this approximation is acceptable in most cases. On the contrary, data of outlier tags X_O, that is generally regarded important and valuable, can be collect without losing cprecision.

Second, the accuracy of data measured by sensor may not be high due to the internal error. As the large-scale RFID system requires a mass of tags, which makes the user extremely sensitive to the tag price, the tags are most likely to be integrated with low-price sensor that provides limited precision. Therefore, the sensor data is inherently approximation of the exactly value.

VII. Evaluation

In this section, we implement SIC in python and evaluate its performance by simulations. A large number of simulations were carried out to evaluate the performance of SIC. At first, we generated a series of data sets that follow different data distribution to show its effect on the execution time of SIC. Then, we varied the error threshold ϵ to test the performance of SIC. Next, we varied E, which determine the sample size to show the importance of optimization. Finally, the fourth simulation set the packet loss rate to be fixed at 0%, 1.25%, 2.5%, 5%, 10%, 20%. For each error rate, the corresponding execution time was recorded. Besides, we also implement three

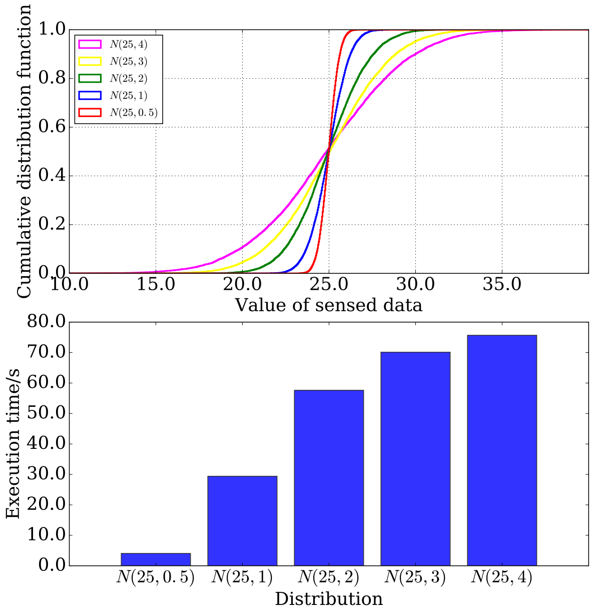

Fig. 3. The performance of SIC over data set with different normal distribution

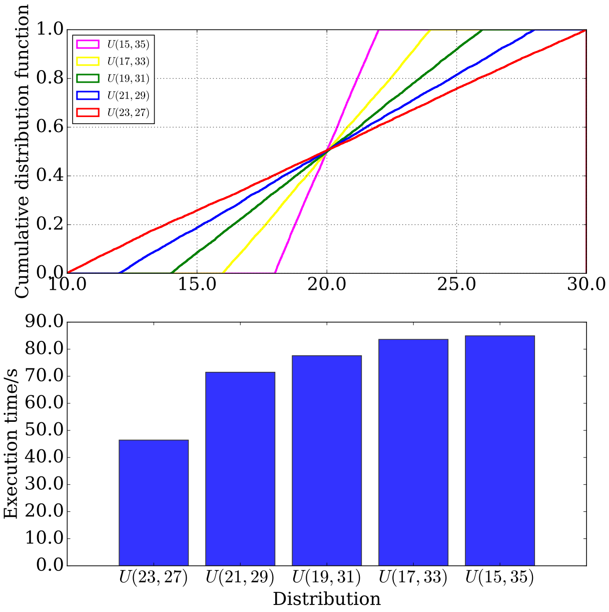

Fig. 4. The performance of SIC over data set with different uniform distribution

prior protocols in python, namely Gen2 [4], MIC [1] and BIC [2], and compared their performance with SIC side by side.

A. Simulation Settings

In our simulation, the communication parameter settings follow the specification of the Gen2 standard [4]. We assume the length of the information ready to be collected is 16 bits. The transmission rate between the reader and tag is equivalent, both $40kb/s$, namely, it costs $25us$ to transmit one bit. Let T_{pari} be the backscatter-link pulse-repetition interval, the waiting time between reader transmission and tag response, and the waiting time between tag transmission and reader response are $T_1 = 10T_{pari}$ and $T_2 = 3T_{pari}$, respectively. Because $T_{pari} \approx 25us$, we have $T_1 = 250us$, and $T_2 = 75us$. Without specific introduction, the results in the following tables and figures are the average time of 100 turns.

B. Effects of Data Distributions

Data distribution is the major factor that determines the effectiveness of SIC. In this subsection, we evaluate the performance of SIC over data sets with different distribution. First of all, as shown in Fig. 3, we generate five data sets whose values follows normal distribution $N(25, 0.5)$, $N(25, 1)$, $N(25, 2)$, $N(25, 3)$ and $N(25, 4)$, respectively. Each data set has the same size: $10K$. Each value represents the sensor data of a tag. We run SIC ($\epsilon = 1$) 100 times based on these data, and the execution time is shown in Fig. 3. Obviously, SIC has a better performance when the data distribution has a smaller standard error. For example, when data set follows the distribution $N(25, 1)$, SIC takes less than $30s$ to collect all data from tags. By comparison, when data set follows the distribution $N(25, 4)$, the time cost increases to more than $75s$ This is because the former data set has a small standard error, which means the data is concentrated in a narrow range. As a result, a fixed length error bound is able to cover more tags, thereby reducing the execution time of collecting outlier tags.

Besides, we also test the performance of SIC over data sets follows uniform distributions $U(20, 25)$, $U(18, 27)$ and $U(15, 30)$. The data set has the same size:10K and is shown in Fig. 4. The execution time of the SIC is shown in Fig. 4. It illustrates that execution time of SIC is longer with a narrower range of uniform distribution. The underlying reason is the same as the last experiment.

C. Effects of Error Threshold ϵ

SIC is the only protocol that provides data with varying degree of precision guaranteeing. We assume there are $5K$ tags whose data value follows a normal distribution $N(25, 1)$, we vary the error threshold of the sensor data to test the performance of the SIC. The results are shown in Fig. 5. We find that the execution time is decreased with the increases of ϵ. Specifically, SIC costs more than $120s$ to collect data when $\epsilon = 0.25$ but a remarkable reduction of execution time is found when $\epsilon = 4$, where SIC only takes $8s$ to collect all the data. The above results reveals that SIC has a better performance when we can bear a larger ϵ. This is not doubt, because the larger error tolerance always means more tags can be approximated by the estimated value and fewer outlier tags need to be collected. An extreme setting is $\epsilon = 0$, which means the user rejects any approximation. Then, SIC works as a traditional Gen2 and brings none benefits.

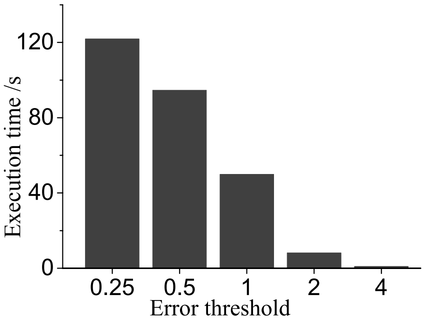

Fig. 5. Execution time of SIC with different ϵ.

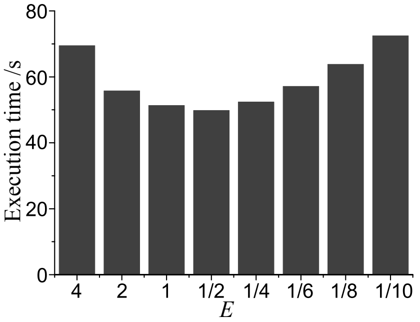

Fig. 6. Execution time of SIC with different E.

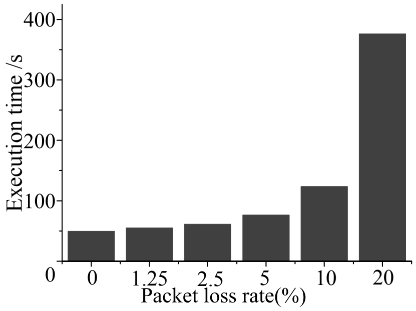

Fig. 7. Execution time of SIC with different packet loss rate.

TABLE II
THE STATISTICS OF EXECUTION TIME (SECONDS) ON DIFFERENT PARAMETERS SETTING

E	Mean	Standard deviation	Min	Maximum
4	69.49	21.98	48.29	141.51
2	55.77	10.66	48.35	110.54
1	51.34	3.60	48.40	67.89
$1/2$	49.84	0.84	49.08	54.54
$1/4$	52.38	0.36	51.99	53.38
$1/6$	57.10	0.17	56.75	57.38
$1/8$	63.81	0.15	63.59	64.10
$1/10$	72.48	0.14	72.24	72.74

D. Effects of proportion error E

Recall from Section VI, E significantly affects the execution time of collecting sampled data and outlier tags. In this subsection, we assume there are $5K$ tags whose data value follows the distribution of $N(25,1)$ and the required error threshold $\epsilon = 1$. We vary E from 4 to $1/10$. Referring to Fig. 6, we note that the execution time is first decreased and then increased with respect to E. This is because a larger E (*e.g.*, $E = 4$) results in inaccurate estimated value, which covers fewer tags and increases the execution time. On the other hand, if E is too small (*e.g.*, $E = 1/8$), the required sample size significantly increases, more samples need to be collected, which also increases the execution time. Therefore, we need to choose an appropriate E to minimize the execution time as we discussed in Section VI.

Table II gives a more detailed comparison of execution time of SIC with different E. It is obvious that with the decreases of E, the standard deviation of execution time is reduced, which means the stability of our protocol is increased. This is because the estimation model returns the error bound with a less deviation. Meanwhile, the minimum execution time increases with respect to E. This is because even if the sample size is small (*i.e.*, E is large), it might return an good estimation of the error bound with a small probability, in this case, SIC can be extremely small. But it is unwise to set a large E because the average execution time is significantly increases.

E. Impact of Unreliable Channel

As we analyzed in Section IV, the transmission errors prolong the execution time of SIC. In this experiment, we set $n = 1K$, $\epsilon = 1$, $E = 0.5$ and the data distribution is set to $N(25,1)$. The packet loss rate varies between 0% and 20%. As shown in Fig. 7, the execution time increases with the increases of packet loss rate. We find moderate increase in the execution time when the packet loss rate is less than 5%, but the growth rate significantly increases when the packet loss rate is over 10%. Compared with the 0% setting, the execution time is doubled when the packet loss rate is equal to 10% and is septupled when the packet loss rate is equal to 20%. This is because the probability of successful transmission is significantly reduced with the increases of the packet loss rate. Thus, the tags have to retransmit so many times, which leads to remarkable increase in the execution time.

F. Protocol Comparison with Prior Work

Tables III and IV compare the execution time of SIC with recent whole-set collection protocols include Gen2 [4], MIC [1] and BIC [2], where SIC ($\epsilon = 1$) and SIC ($\epsilon = 2$) means the SIC with different required error threshold. We first assume the measured data follows the $N(15,1)$ distribution and the number of tags varies from $1K$ to $9K$. As shown in Table III, SIC significantly outperforms the state-of-the-art BIC. For example, the execution time of SIC ($\epsilon = 1$) is only 57.3% of the time need by BIC, when the number of tags equals to $5K$. SIC ($\epsilon = 1$) further reduces the execution time, costing only 9.4% of the time needed by BIC. Although the proposed SIC uses traditional Gen2 protocol to collect data from tags, it still outperforms MIC and BIC because SIC only needs to gather information from a part of tags that consist of sample tags and outlier tags. It is little surprise that SIC ($\epsilon = 1$) outperforms SIC ($\epsilon = 1$) because the latter one can bear a large error threshold, which reduces the number of outlier tags to be collected.

TABLE III
EXECUTION TIME (SECONDS) COMPARISON WHEN THE DATA FOLLOWS NORMAL DISTRIBUTION $N(15,1)$

N_I	Cen2	MIC	BIC	SIC ($\epsilon = 1$)	SIC ($\epsilon = 2$)
1K	14.67	9.60	8.61	5.69	1.51
3K	44.57	28.79	25.82	15.16	2.85
5K	74.59	47.98	43.04	24.66	4.05
7K	103.58	67.18	60.26	34.34	5.82

Then, we test the performance of SIC with irregular distribution. The test dataset consists of 4K data follow $N(17,2)$ and 6K data follows $N(15,1)$. Table IV shows that SIC still outperforms the state-of-the-art protocol. However, SIC with $\epsilon = 1$ has very similar performance with BIC. This is because the irregular distribution values are scattered. A narrow error bound can only cover a small number of outlier tags, which narrows the gap between the execution time of SIC and BIC. However, SIC is still a better choice, because the low-layer protocol adopted by SIC is Gen2, which is more reliable and can handle transmission errors. SIC with $\epsilon = 2$ still outperforms other protocols, costing only 50.9% of the time needed by BIC when the number of tags equals to $7K$.

TABLE IV
EXECUTION TIME (SECONDS) COMPARISON WHEN THE DATA FOLLOWS IRREGULAR DISTRIBUTION

N_I	Cen2	MIC	BIC	SIC ($\epsilon = 1$)	SIC ($\epsilon = 2$)
1K	14.67	9.60	8.61	8.60	4.38
3K	44.57	28.79	25.82	24.50	13.71
5K	74.59	47.98	43.04	40.49	22.85
7K	103.58	67.18	60.26	57.19	30.69
9K	134.94	86.37	77.47	73.40	40.58

VIII. CONCLUSION

This paper makes the following three contributions. First, we formally define a new practical problem of information collection with a tolerance for a certain error. Second, we propose an Sampling-based Information Collection (SIC) protocol, which adopts a sample estimator to compress the data transmission. SIC significantly improves the time-efficiency in comparison to the state-of-the-art solutions, while being able to guarantee an arbitrary data precision. Third, we investigate how to set the sample size and frame length to optimize the execution time of SIC. Finally, extensive simulations are conducted to evaluate the performance of the proposed SIC. The results show that SIC is 1 to 10 times faster than the state-of-the-art solutions.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation for Distinguished Young Scholars of China (Grant No. 61225010); the State Key Program of National Natural Science of China(Grant No. 61432002); NSFC Grant Nos. 61272417, 61300189 and 61370199; Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20130041110019); the Fundamental Research Funds for the Central Universities (Grant. DUT15QY20) and HK PolyU 4-BCB9

REFERENCES

[1] S. Chen, M. Zhang, and B. Xiao, “Efficient information collection protocols for sensor-augmented RFID networks,” in *Proc. of IEEE INFOCOM*, 2011.

[2] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen, “A time-efficient information collection protocol for large-scale RFID systems,” in *Proc. of IEEE INFOCOM*, 2012.

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

[4] *EPC Radio-Frequency Identity Protocols Class-1 Gen-2 UHF RFID Protocol for Communications at 860MHz-960MHz, EPCglobal*, http://www.epcglobalinc.org/standards/uhfc1g2, Apr 2011.

[5] D. Tulone and S. Madden, “Paq: Time series forecasting for approximate query answering in sensor networks,” in *In Proc. of EWSN*, 2006.

[6] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, “Approximate data collection in sensor networks using probabilistic models,” in *In Proc. of ICDE*, 2006.

[7] L. Xie, B. Sheng, C. C. Tan, H. Han, Q. Li, and D. Chen, “Efficient tag identification in mobile RFID systems,” in *Proc. of IEEE INFOCOM*, 2010.

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

[9] M. Shahzad and A. X. Liu, “Every bit counts: Fast and scalable rfid estimation,” in *Proc. of ACM MOBICOM*, 2012.

[10] L. Pan and H. Wu, “Smart trend-traversal: a low delay and energy tag arbitration protocol for large RFID systems,” in *Proc. of IEEE INFOCOM*, 2009.

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

[12] W. Luo, S. Chen, T. Li, and Y. Qiao, “Probabilistic missing-tag detection and energy-time tradeoff in large-scale RFID systems,” in *Proc. of ACM Mobihoc*, 2012.

[13] L. M. Ni, D. Zhang, and M. R. Souryal, “RFID-based localization and tracking technologies,” *IEEE Wireless Communications*, vol. 18, no. 2, 2011.

[14] W. Zhu, J. Cao, Y. Xu, L. Yang, and J. Kong, “Fault-tolerant rfid reader localization based on passive RFID tags,” in *Proc. of IEEE INFOCOM*, 2012.

[15] T. Liu, L. Yang, Q. Lin, Y. Guo, and Y. Liu, “Anchor-free backscatter positioning for rfid tags with high accuracy,” in *Proc. of IEEE INFOCOM*, 2014.

[16] Y. Zheng and M. Li, “Fast tag searching protocol for large-scale RFID systems,” in *Proc. of IEEE ICNP*, 2011.

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

[18] W. Luo, Y. Qiao, and S. Chen, “An efficient protocol for RFID multigroup threshold-based classification,” in *Proc. of IEEE INFOCOM*, 2013.

[19] J. Liu, B. Xiao, K. Bu, and L. Chen, “Efficient distributed query processing in large rfid-enabled supply chains,” in *Proc. of IEEE INFOCOM*, 2014.

[20] W. Gong, K. Liu, X. Miao, Q. Ma, Z. Yang, and Y. Liu, “Informative counting: fine-grained batch authentication for large-scale rfid systems,” in *Proc. of ACM MOBIHOC*, 2013.

[21] X. Liu, K. Li, G. Min, K. Lin, B. Xiao, Y. Shen, and W. Qu, “Efficient unknown tag identification protocols in large-scale rfid systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3145–3155, 2014.

[22] L. Yang, Y. Chen, Xiang-Yang, C. Xiao, M. Li, and Y. Liu, “Tagoram:real-time tracking of mobile rfid tags to high precision using cots devices,” in *Proc. of ACM MOBICOM*, 2014.

[23] K. Bu, B. Xiao, Q. Xiao, and S. Chen, “Efficient misplaced-tag pinpointing in large rfid systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2094–2106, 2012.

[24] W. Luo, S. Chen, T. Li, and S. Chen, “Efficient missing tag detection in RFID systems,” in *Proc. of IEEE INFOCOM*, 2011.