Towards More Efficient Cardinality Estimation for Large-Scale RFID Systems

Yuanqing Zheng, *Student Member, IEEE*, and Mo Li, *Member, IEEE, ACM*

***Abstract*—Radio frequency identification (RFID) cardinality estimation with an accuracy guarantee is of practical importance in various large-scale RFID applications. This paper proposes a fast RFID cardinality estimation protocol, named Zero-One Estimator (ZOE). ZOE only requires 1-bit response from the RFID tags per estimation round. More importantly, ZOE rapidly converges to optimal parameter configurations and achieves higher estimation efficiency compared to existing protocols. ZOE guarantees arbitrary accuracy requirement without imposing heavy computation and memory overhead at RFID tags except the routine operations of C1G2 standard. ZOE also provides reliable cardinality estimation with unreliable channels due to the robust protocol design. We prototype ZOE using the USRP software defined radio and the Intel WISP tags. We extensively evaluate the performance of ZOE compared to existing protocols, which demonstrates encouraging results in terms of estimation accuracy, time efficiency, as well as robustness over a large range of tag population.**

***Index Terms*—Cardinality estimation, radio frequency identification (RFID) systems.**

I. Introduction

RECENTLY, radio frequency identification (RFID) systems [9] have received significant interest from both academia and industry. A large-scale RFID system usually consists of multiple RFID readers and a huge amount of RFID tags [24]. An RFID tag is capable of storing its unique ID as well as some other information and wirelessly transmitting them back to readers. By verifying the unique IDs of RFID tags attached to physical objects, RFID readers are able to identify and itemize the objects. Due to small form factor and low cost of RFID tags, RFID systems provide us a scalable and economic way for managing massive objects in a variety of applications including inventory management [8], [15], [26], logistics [27], [29], object tracking [18], [20], etc.

This paper studies the fundamental problem of estimating the number of tags in large-scale RFID systems. Fast estimating of the cardinality of RFID tags, accordingly the number of labeled items, is of primary importance to many applications [24]. For instance, estimating the number of conference attendees with RFID badges allows us to track the movement and distribution of attendees in different conference rooms. A warehouse manager may benefit from a quick estimate of products in stock. A fast cardinality estimation scheme also serves as primary inputs to various RFID protocols. For instance, Aloha-based RFID identification protocols achieve near-optimal performance if the contention frame size can be set according to the number of contending tags [7], [12], [14]. Many missing tag monitoring protocols can also build upon accurate estimation results [15], [26].

To this end, probabilistic estimation approaches have been proposed to efficiently estimate the number of RFID tags. Some recent approaches achieve $\mathcal{O}(\log n)$ estimation efficiency to the number of RFID tags n [11], [19]. One most recent protocol, Probabilistic Estimation Tree (PET), achieves $\mathcal{O}(\log \log n)$ time efficiency for each estimation round [28]. Nevertheless, existing protocols require many independent estimation rounds to achieve high accuracy. For instance, PET takes several seconds to achieve an accurate estimation. As a basic component that may be frequently invoked by many applications, an estimation protocol can easily become the bottleneck that limits the overall performance of large-scale RFID systems. Further improving the time efficiency of each estimation round will significantly benefit the entire cardinality estimation process, meet the stringent time requirement of many real-time applications, and support larger-scale RFID systems.

While pursuing the estimation efficiency at the optimum, we are also aiming at reducing the computation and memory overhead of resource-constrained RFID tags. Most existing probabilistic approaches require generating a large volume of random numbers or alternatively prestoring them at RFID tags, which lead to heavy computation and storage burden for RFID tags. We aim at shifting such overhead from resource-constrained RFID tags to powerful RFID readers. Besides, most existing works assume a reliable wireless channel between the RFID reader and tags, which is contradicting with the fact that the wireless channel is mostly error-prone.

This paper presents Zero-One Estimator (ZOE): a fast RFID cardinality estimation protocol with guaranteed accuracy requirement. ZOE first configures the system parameters and converges to optimal settings with a bisection search with negligible overhead. With the optimized parameter settings, ZOE estimates the RFID cardinality with only 1-bit feedback from tags at each round providing extremely high estimation efficiency. We further consider the unreliable channels and propose Error Estimation and Adjustment (EEA) algorithm to adjust estimation results according to error rates.

We implement a prototype system using the Universal Software Radio Peripheral (USRP) [4] and the Intel Wireless Identification and Sensing Platform (WISP) [23]. We implement the ZOE reader functionality with USRP Software

Manuscript received March 05, 2013; revised August 29, 2013; accepted October 11, 2013; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Chen. This work was supported by the NTU Nanyang Assistant Professorship (NAP) Grant M4080738.020.

The authors are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: yuanqing1@ntu.edu.sg; limo@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2013.2288352

Defined Radio (SDR) that interrogates programmable WISP tags. The ZOE protocol only requires slight updates to the EPCglobal Class 1 Generation 2 (C1G2) standard. We also evaluate ZOE with extensive simulations in large-scale RFID systems.

The rest of this paper is organized as follows. We briefly review the related work in Section II. In Section III, we introduce our system model and describe the problem of cardinality estimation. We give a detailed description on ZOE in Section IV. We present the implementation of ZOE in Section V. We evaluate ZOE with extensive simulations in Section VI. Section VII concludes this paper.

II. Related Work

RFID identification protocols aim at collecting the tag IDs of all RFIDs in the interrogation area [7], [14], [21], [25], [30]. The identification protocols based on collision arbitration can generally be classified into two categories: Aloha-based protocols [14], [21] and Tree-based protocols [7], [30]. In small-scale RFID systems, we may directly apply the identification protocols to count the exact tag cardinality. Nevertheless, such a method becomes infeasible due to the low efficiency of identification protocols at scale. Rather than identifying all the tags, probabilistic estimation protocols specifically tailored for large-scale RFID systems estimate the number of tags to meet customized accuracy requirement.

Kodialam and Nandagopal present Unified Simple Estimator (USE) and Unified Probabilistic Estimator (UPE) [12]. Those schemes are vulnerable to multiple counting problems when multiple RFID readers are deployed to cover the interrogation region. Besides, the protocols require a cardinality upper bound known in advance. Kodialam *et al.* [13] propose Enhanced Zero-Based (EZB) estimator to estimate a relatively large number of tags. Shahzad *et al.* [22] propose Average Run-based Tag (ART) estimation protocol to improve estimation efficiency. Such approaches require $\mathcal{O}(n)$ time-slots for each estimation round to the total number of RFID tags n.

Recent probabilistic estimation approaches achieve $\mathcal{O}(\log n)$ estimation efficiency. Han *et al.* [11] propose the First Non-Empty slot Based (FNEB) estimator with binary search method to pinpoint the first nonempty slot. Qian *et al.* [19] propose the Lottery Frame (LoF)-based estimator, which is a replicate-insensitive estimation protocol. Both approaches require the tags to cooperate with the reader by generating a large volume of random numbers and respond accordingly. One most recent protocol, PET-based estimator [28], advances the estimation efficiency and achieves $\mathcal{O}(\log \log n)$ efficiency to the tag cardinality n. Li *et al.* [16], [17] design energy-efficient estimation algorithms to save energy for active RFID tags powered by batteries. Unlike prior probabilistic estimation approaches, [16] and [17] adopt the maximum likelihood estimation and propose several approaches to avoid collisions to save power.

III. System Model and the Problem

A. System Model

We consider a large-scale RFID system consisting of three major components: a large volume of RFID tags, several RFID readers, and a back-end server connecting the readers. Multiple RFID readers are normally deployed to ensure a full coverage of large-scale RFID systems. The back-end server coordinates the RFID readers and initiates the cardinality estimation process. The RFID readers relay the commands received from the back-end server and broadcast to tags, and later report the tags' responses back to the server. For ease of description, we first focus on the communication between the RFID tags and one RFID reader covering all the tags. We discuss how to coordinate multiple RFID readers in Section IV-E.2. The RFID system may use lightweight passive RFID tags or more powerful active ones.

In this paper, we exclusively focus on the RFID systems operating in the 900-MHz ultra-high frequency band. We assume that the RFID system works on a frame-slotted Aloha model. RFID readers initiate interrogation by sending operation codes and specifying PHY/MAC parameters. When energized by the continuous waves from reader, each tag backscatters a message or keeps silent. Such a communication model has been widely adopted in many RFID systems compliant with the *de facto* EPCglobal C1G2 standard [2].

The practical communication channel is mostly error-prone depending on various factors including transmission power, interrogation distance, antenna gain, interference, etc. [6]. Due to the channel attenuation, even if there are some tags transmitting back responses, the reader may fail to detect them in practice. We call such missing detection errors as false negatives. On the other hand, the reader may falsely detect a busy channel due to the interferences, even when no response is transmitted. We call such errors as false positives.

B. Problem Description

The objective of this work is to efficiently and accurately estimate the tag cardinality. To meet stringent real-time requirement, the estimation protocol should compute an accurate tag cardinality in an efficient manner. Since the cardinality of RFID tags could range from several hundreds to even tens of thousands (e.g., a typical port inventory application may concern hundreds of containers, each of which may contain tens of thousands of RFID-labeled products), we need to design a scalable and efficient estimation approach.

Consistent with the existing approaches [11], [19], [28], the accuracy requirement is presented by (ε, δ)-approximation. We denote by $\hat{n}$ the estimated number of the tag cardinality while the actual number is n. Given the accuracy requirement of (ε, δ)-approximation, we expect an estimation result $\hat{n}$, which satisfies $\Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$. For example, when the actual tag cardinality is 10 000, $(\varepsilon = 5\%, \delta = 1\%)$-approximation expects an estimation result within the interval [9500, 10 500] with a probability of 99% and above.

Although many recent protocols have been proposed to reduce the estimation time, it still takes several seconds to accurately estimate the number of tags. For instance, PET [28] takes roughly 9 s to achieve the accuracy requirement of ($\varepsilon = 5\%$, $\delta = 1\%$)-approximation. Further improving the time efficiency will benefit many RFID protocols relying on accurate estimation results.

We abstract the time efficiency with the total time-slots used to estimate the cardinality. Most recent approaches only need

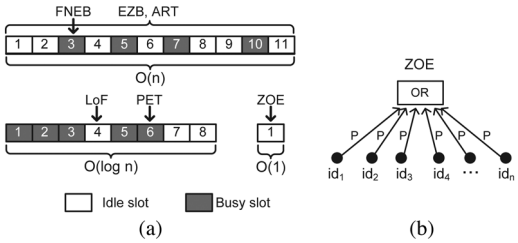


Fig. 1. Illustrative comparison of ZOE with conventional schemes of EZB, ART, FNEB, LoF, and PET. (a) Interested slots in a frame. (b) One aggregated slot.

to distinguish an idle slot from a busy slot [11], [19], [28]. The smaller number of time-slots means the shorter communication time and thus the higher time efficiency, and vice versa. Meanwhile, we seek to reduce the computation and memory burden at the RFID tags to facilitate the use of low-cost but resource-constrained passive tags rather than expensive active ones.

While most recent estimation protocols assume zero error rates in the underlying wireless channel, we target at practical unreliable channels. When the bit error rate is high, it becomes very difficult to derive accurate cardinality estimation. Nevertheless, when the channel is in mild conditions, we expect that the estimation protocol computes a reasonably accurate estimation result.

IV. Estimation Protocol

In this section, we first discuss the design principle of ZOE protocol. We then consolidate the essential idea with a cardinality estimation protocol, which provides $\mathcal{O}(1)$ time efficiency for each estimation round.

A. Principle

The existing approaches take advantage of the frame-slotted Aloha protocol to estimate the number of tags. The RFID reader normally needs to examine each time-slot in the frame. Fig. 1(a) presents illustrative examples of most recent protocols, e.g., EZB [13], ART [22], FNEB [11], LoF [19], and PET [28]. In EZB, each tag randomly selects a slot from $\mathcal{O}(n)$ slots with uniform distribution functions to send a response in each frame. If the frame size is fixed, then the more busy slots there are, the larger the tag population would be. EZB measures all $\mathcal{O}(n)$ slots to estimate tag cardinality. ART also measures $\mathcal{O}(n)$ slots in each estimation round and uses the average run size of identical responses as an estimator. ART outperforms EZB because ART has significantly smaller variance. FNEB notices that the first busy slot (herein, slot 3) indicates the tag population (i.e., the smaller the first busy slot is, the larger the tag population would be). Leveraging the monotonic feature, FNEB locates the first busy slot that takes $\mathcal{O}(\log n)$ time-slots in each frame. LoF and PET reduce the frame size to $\mathcal{O}(\log n)$ by letting each tag select a slot with geometric distribution functions, such that approximately $1/2^i$ of tags respond in the ith slot. PET achieves $\mathcal{O}(\log \log n)$ time efficiency leveraging a probabilistic binary tree structure. The estimation results of such protocols may statistically vary for each estimation round. For instance, the first busy slot in FNEB might dramatically deviate from the expected slot. Existing protocols thus need many independent estimation rounds to derive averages to accurately estimate the actual number of tags.

To improve the estimation efficiency, we propose the ZOE protocol in which each frame only contains one slot. In particular, all the responses from tags aggregate at a single slot, leading to either an idle slot if no tag responds or a busy slot otherwise as illustrated in Fig. 1(b). Suppose there exist n RFID tags, and each tag responds with the probability of P, and keeps silent with the probability of $1 - P$. Intuitively, the more tags there are, the higher probability that the reader observes a busy slot, and vice versa. We can thus measure the ratio of busy (idle) slots and infer the tag population.

Unlike conventional approaches where only small portion of tags participate in each time-slot, in ZOE, the responses from all the tags aggregate in the single time-slot, which allows ZOE to make extensive use of each time-slot. MLE [16], [17] shares a similar design principle with ZOE in that each tag probabilistically sends 1-bit response in each time-slot. The total time overhead involved in the overall estimation process is the number of estimation rounds multiplied by the overhead per estimation round. Although one may reduce the overhead per estimation round, it requires careful optimization to reduce the overall overhead and achieve higher time efficiency. By intelligently setting the system parameters, ZOE only needs comparable number of estimation rounds with conventional approaches, meaning that we reduce the overhead in each estimation round to $\mathcal{O}(1)$ while keeping the number of estimation rounds similar to those in prior schemes. In the following, we present detailed theoretical analysis and parameter optimization.

B. Zero-One Estimator Protocol

We describe the detailed ZOE protocol in this section.

1) Tag: When probed by a reader in the cardinality estimation process, each tag independently computes a random number with a uniform distribution hash function $\mathcal{H}(\mathrm{id}, s)$, where s denotes a random seed. For simplicity, we omit the notation of s for the hash functions. We denote by $\mathcal{H}_B(\mathrm{id})$ the binary representation of $\mathcal{H}(\mathrm{id})$. We also denote by $\mathcal{R}(\mathrm{id})$ the index of the right-most zero bit in $\mathcal{H}_B(\mathrm{id})$ as follows:

$$\mathcal{R}(\mathrm{id}) = \min\{i | \mathcal{H}_B(\mathrm{id})[i] = 0\}. \quad (1)$$

If $\mathcal{R}(\mathrm{id}_i) \geq \theta$, the tag responds to the reader where θ is a threshold received from the reader; if $\mathcal{R}(\mathrm{id}_i) < \theta$, the tag keeps silent.

Let the random variable of $\mathcal{R}(\mathrm{id}_i)$ be R_i, $1 \leq i \leq n$, then we have the probability

$$\Pr(R_i) = p^{R_i}(1-p)$$

where p denotes the probability that a bit of $\mathcal{H}_B(\mathrm{id})$ turns out to be "1." Typically, we assume $p = 0.5$, i.e., the hash function is a uniform distribution hash function. Then, we have $p = 1 - p = 0.5$, and

$$\Pr(R_i) = p^{R_i+1} = \frac{1}{2^{R_i+1}}.$$

The probability that a tag keeps silent given a threshold θ is

$$\Pr(R_i < \theta) = \sum_{k=0}^{\theta-1} \Pr(k) = \sum_{k=0}^{\theta-1} \frac{1}{2^{k+1}} = 1 - \frac{1}{2^\theta}.$$

On the other hand, a tag will respond to the reader with the probability of $1 - \Pr(R_i < \theta) = 1/2^\theta$.

2) Reader: A reader initiates cardinality estimation by sending a random seed and the threshold θ to the tags and waits for the responses from the tags. In the case that $\mathcal{R}(\mathrm{id}_i) < \theta, \forall i \in \{1, 2, \ldots, n\}$, the reader observes no reply from the tags. Therefore, with the assumption of independent identical distribution (i.i.d) for R_i, the probability that there is no reply from the tags (i.e., the channel is idle) is as follows:

$$\Pr(\text{idle}) = [\Pr(R_i < \theta)]^n = \left(1 - \frac{1}{2^\theta}\right)^n \approx e^{-n/2^\theta} = e^{-\lambda}$$

where the load factor $\lambda = n/2^\theta$, and $\lambda > 0$.

The probability that there is a reply (no matter a singleton reply from one tag or replies from multiple tags) is

$$\Pr(\text{reply}) = 1 - \Pr(\text{idle}) \approx 1 - e^{-n/2^\theta} = 1 - e^{-\lambda}.$$

We define a random variable X that takes value 1 with probability $\Pr(\text{idle}) \approx e^{-n/2^\theta} = e^{-\lambda}$ and value 0 with probability $\Pr(\text{reply}) \approx 1 - e^{-n/2^\theta} = 1 - e^{-\lambda}$. Then, we have

$$\Pr(X = 1) \approx e^{-\lambda} \quad \Pr(X = 0) \approx 1 - e^{-\lambda}.$$

Obviously, the random variable X follows the Bernoulli distribution. Therefore, the expectation and the standard deviation of X are as follows:

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

The maximum standard deviation of X is

$$\sigma(X)_{\max} = 0.5, \qquad \text{when } e^{-\lambda} = 0.5.$$

We define the random process $\bar{X} = (1/m)\sum_{i=1}^{m} X_i$ as the average of m independent observations, where X_i denotes the ith observation of random variable X. We assume the trials of $X_i (1 \le i \le m)$ are i.i.d, then we have $E(\bar{X}) = E(X)$ and $\sigma(\bar{X}) = \sigma(X)/\sqrt{m}$.

According to the law of large numbers [10], when m is large we have

$$\bar{X} = E(\bar{X}) = E(X) = e^{-n/2^\theta} = e^{-\lambda}. \tag{2}$$

According to (2), we can estimate the load factor as follows:

$$\hat{\lambda} = -\ln \bar{X}$$

where $\hat{\lambda}$ denotes the estimation of λ.

The observation of $\bar{X}$ can thus be used to estimate the tag cardinality $\hat{n}$ as follows:

$$\hat{n} = -2^\theta \ln \bar{X}. \tag{3}$$

Since the result may vary slightly because of the estimation variance, we seek a guaranteed cardinality estimation result, e.g., $Pr\{|\hat{n} - n| \le \varepsilon n\} \ge 1 - \delta$. The estimation accuracy requirement can be represented as follows:

$$\Pr\{|\hat{n} - n| \le \varepsilon n\} = \Pr\left\{e^{-\lambda(1+\varepsilon)} \le \bar{X} \le e^{-\lambda(1-\varepsilon)}\right\}.$$

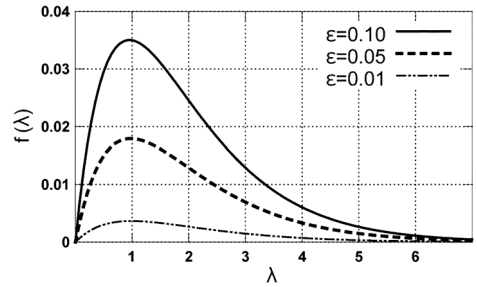


Fig. 2. $f(\lambda)$ obtains the maximum value at $\lambda \approx 1$.

We define a random variable $Y = \bar{X} - \mu/\sigma$, where $\mu = E(\bar{X}) = e^{-\lambda}$, and $\sigma = \sigma(\bar{X}) = \sigma(X)/\sqrt{m}$. By the central limit theorem [10], we know Y is asymptotically standard normal distribution.

Given a particular error probability δ, we can find a constant c that satisfies

$$1 - \delta = \Pr\{-c \le Y \le c\} = \mathrm{erf}\left(\frac{c}{\sqrt{2}}\right)$$

where erf is the Gaussian error function [10]. Therefore, we can guarantee the accuracy requirement $\Pr\{|\hat{n} - n| \le \varepsilon n\} \ge 1 - \delta$ if we have the following conditions:

$$\frac{e^{-\lambda(1+\varepsilon)} - e^{-\lambda}}{\sigma} \le -c \quad \text{and} \quad \frac{e^{-\lambda(1-\varepsilon)} - e^{-\lambda}}{\sigma} \ge c. \tag{4}$$

According to (4), we have

$$m \ge \max\left\{\left[\frac{c\sigma(X)_{\max}}{e^{-\lambda}(1 - e^{-\varepsilon\lambda})}\right]^2, \left[\frac{c\sigma(X)_{\max}}{e^{-\lambda}(e^{\varepsilon\lambda} - 1)}\right]^2\right\} \ge \left[\frac{c\sigma(X)_{\max}}{e^{-\lambda}(1 - e^{-\varepsilon\lambda})}\right]^2. \tag{5}$$

Therefore, with such m estimation frames, ZOE can guarantee the accuracy requirement of $\Pr\{|\hat{n} - n| \le \varepsilon n\} \ge 1 - \delta$. In (5), we see that m depends on $\lambda = n/2^\theta$ indicating that the threshold θ may influence the estimation efficiency. In the following, we discuss how to set the threshold to optimize performance of ZOE.

C. Parameter Setting

Before we perform the estimation process, we need to set the threshold θ that directly influences the behaviors of the tags and the estimation efficiency. If θ is too big, the reader will consistently observe idle slots, i.e., $\bar{X} = 1$; if θ is too small, the reader will observe busy slots in almost every time-slots, i.e., $\bar{X} = 0$, with high probability. In either situation, it consumes extra processing time to meet an accuracy requirement. As a matter of fact, if we look at the lower bound of the estimation round m measured in (5), since $\lambda = n/2^\theta$, the lower bound depends on the tag cardinality, which is not known in advance. We denote by $f(\lambda) = e^{-\lambda}(1 - e^{-\varepsilon\lambda}) \approx e^{-\lambda}\varepsilon\lambda$, the denominator of $c\sigma(X)_{\max}/e^{-\lambda}(1 - e^{-\varepsilon\lambda})$ in (5). To reduce the number of estimation rounds, we maximize the denominator $f(\lambda)$ since the numerator $c\sigma(X)_{\max}$ is constant given an accuracy requirement. Fig. 2 plots $f(\lambda)$ against λ for different ε. We observe that $f(\lambda)$ reaches the maximum value at $\lambda \approx 1$.

Algorithm 1: Threshold setting algorithm

1: $low \leftarrow 0, high \leftarrow 32$
2: **while** $low < high$ **do**
3: $mid \leftarrow (low + high)/2$
4: $\theta \leftarrow mid$, Compute $\bar{X}$ with Algorithm 2
5: **if** $\bar{X} >= (e^{-2} + e^{-1})/2$ **and** $\bar{X} <= (e^{-0.5} + e^{-1})/2$ **then**
6: $\theta \leftarrow mid$; **break**;
7: **end if**
8: **if** $\bar{X} > ((e^{-0.5} + e^{-1})/2)$ **then**
9: $high \leftarrow mid$
10: **else**
11: $low \leftarrow mid$
12: **end if**
13: **end while**
14: **return** θ

We compute the first order derivative of $f(\lambda) \approx e^{-\lambda}\varepsilon\lambda$

$$\frac{df(\lambda)}{d\lambda} = \varepsilon e^{-\lambda}(1-\lambda). \tag{6}$$

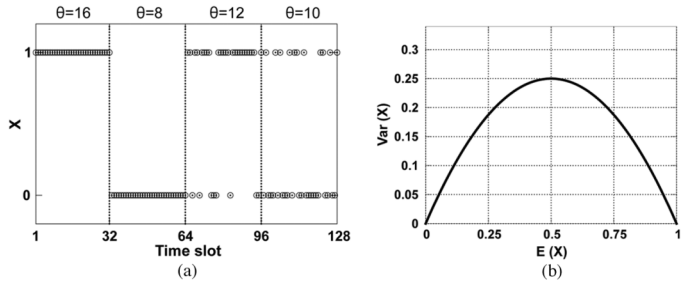
According to (6), the first order derivative vanishes at $\lambda \approx 1$, and we have $\varepsilon e^{-\lambda} > 0$. Therefore, the lower bound $m_{\min}$ is achieved at $\lambda \approx 1$, i.e., when $\bar{X} = e^{-\lambda} \approx e^{-1}$.

This observation motivates us to adapt the threshold θ according to the observation of a short sequence of the tags' responses such that $\bar{X}$ becomes close to e^{-1}. When the reader observes too many idle slots, i.e., $\bar{X} \gg e^{-1}$, it decreases the threshold to increase the probability that tags would send responses; when the reader observes almost all the busy slots, i.e., $\bar{X} \ll e^{-1}$, it increases the threshold to decrease the response probability.

The expected value of $\bar{X}$ is monotonically nondecreasing against the threshold. We exploit such a monotonic feature to fast converge to an optimal threshold. We can reach a suitable θ that provides us $\bar{X} \approx e^{-1}$ with bisection search. Since we know the target average of $\bar{X}$, i.e., $e^{-1} \approx 0.37$, we can terminate the bisection search when the intermediate value of $\bar{X}$ becomes very close to $e^{-1} \approx 0.37$. In particular, we adapt θ and terminate the bisection process when the intermediate value $\bar{X}(\theta_{\text{int}})$ reaches the interval $[(e^{-2} + e^{-1})/2, (e^{-0.5} + e^{-1})/2]$ and use θ_{int} as the threshold.

Algorithm 1 presents the threshold setting process using bisection search method. The threshold is set to be the average of low and $high$. The low end and $high$ end are adjusted according to $\bar{X}$ (lines 8–12). Finally, the two ends converge, and the average is used as the threshold θ (line 2). When the intermediate value $\bar{X}$ becomes close to the target average e^{-1}, the bisection process terminates, and $\theta = mid$ is used as the threshold (lines 5–7).

Fig. 3(a) depicts an example of setting the threshold. In the experiment, the actual tag cardinality is 1024, and thus the optimal threshold is $\theta = \log_2 1024 = 10$. We repeat a small number of trials in each bisection step to derive $\bar{X}$. In Fig. 3(a), we see that the experiment consists of four steps (i.e., $\theta = 16$, 8, 12, and 10), and the number of trials is set to be an empirical number of 32 (we will elaborate why 32 is sufficient shortly). At the first step (1–32), we start with the threshold $\theta = 16$. The reader observes 32 consecutive idle slots denoted by "1"s in Fig. 3(a). Since $\bar{X} \gg e^{-1}$, we adjust the parameter by decreasing θ at the second step (33–64), and we repeat again 32 trials with the threshold $\theta = 8$. The reader observes 32 straight busy slots denoted by "0"s. At the third step (65–96), the threshold is tuned to be $(8 + 16)/2 = 12$. In this case, the reader observes both "1"s and "0"s ($\bar{X} = 24/32 = 0.75 > e^{-1}$). At the final step (97–128), we run the estimation with $\theta = (8 + 12)/2 = 10$, and the reader observes mixed "1"s and "0"s ($\bar{X} = 11/32$). Since $\bar{X} = 11/32 \approx 0.34$ at the final step is quite close to $e^{-1} \approx 0.37$, we set the threshold θ to be 10.



(a) (b)

Fig. 3. Parameter setting process: (a) fast convergence to the optimal threshold value with bisection search method; (b) when $E(X) \to 0$ or $E(X) \to 1$, the variance is very small.

Here, we elaborate why the empirical number of 32 trials is sufficient for the threshold setting. Fig. 3(b) plots the variance of X against the expectation of X. We find that when the expectation $E(X) \to 0$ or $E(X) \to 1$, the variance $\text{Var}(X) \to 0$, indicating that when θ is either too big or too small, $\bar{X}$ shall be relatively stable around $E(X)$ to tell the scale of n. Therefore, it is safe to roughly and rapidly estimate the scale of tag cardinality and set the threshold accordingly with a small number of runs. This is the reason why we can use a small sequence of 32 slots to calculate the optimal θ.

The parameter setting process involves several bisection steps to determine a threshold. This small amount of overhead after further reduction by early termination becomes almost negligible (about 3% of the total estimation overhead). Therefore, we can first tune the threshold θ and converge to an optimal parameter setting at a very small cost. Using this optimal threshold, we can estimate the accurate cardinality with minimal number of estimation rounds achieving higher overall efficiency.

D. ZOE Algorithms

Algorithm 2 regulates the behavior of the RFID reader. The reader calculates the estimation rounds m according to (5) given an accuracy requirement (line 1). The reader initiates the estimation process by energizing the tags and sending the threshold θ (line 2). The reader generates random seeds and broadcasts them (lines 3 and 4), and records the tags' responses (lines 5–9). The average $\bar{X}$ is thus calculated based on the m estimation rounds (line 11). Finally, the estimated tag cardinality is computed according to (3) (line 12).

Algorithm 3 regulates the behavior of each tag. In each estimation round, when receiving a random seed s, the tag computes

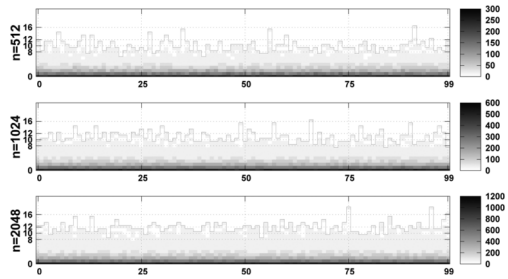


Fig. 4. Distribution of random numbers in 100 estimation rounds. The skyline depicts $\mathcal{R}(\mathrm{id}_i)_{\max}$.

Algorithm 2: ZOE algorithm for RFID readers

1: $m \leftarrow [c\sigma(X)_{\max}/e^{-\lambda}(1-e^{-\varepsilon\lambda})]^2$
2: Initiate the estimation, broadcast θ
3: **for** $i \leftarrow 1$ to m **do**
4: Generate a random seed s and broadcast it
5: **if** there is no response in the slot **then**
6: $X_i \leftarrow 1$
7: **else**
8: $X_i \leftarrow 0$
9: **end if**
10: **end for**
11: $\bar{X} \leftarrow (1/m)\sum_{i=1}^{m} X_i$
12: **return** $\hat{n} \leftarrow -2^{\theta}\ln \bar{X}$

Algorithm 3: ZOE algorithm for each RFID tag

1: Receive the threshold θ
2: **while** TRUE **do**
3: Receive the random seed s; Compute $\mathcal{R}(\mathrm{id})$
4: **if** $\mathcal{R}(\mathrm{id}) \geq \theta$ **then**
5: Respond immediately
6: **else**
7: Keep silent
8: **end if**
9: **end while**

the random number $\mathcal{R}(\mathrm{id})$ according to (1). The tag keeps silent or responds to the reader according to $\mathcal{R}(\mathrm{id})$ and the threshold θ. If $\mathcal{R}(\mathrm{id}) \geq \theta$ the tag sends a response, and otherwise keeps silent (lines 2–9).

Fig. 4 gives an illustrative example of the estimation process. The subfigures depict the three cases where the total numbers of RFID tags are 512, 1024, and 2048, respectively. The x-axis represents 100 independent estimation rounds, and the y-axis represents the random numbers. The gray level intensity of grid (x, y) indicates the amount of tags that generate the random number of y at the xth estimation round. The skyline thus indicates the maximum values $\mathcal{R}(\mathrm{id}_i)_{\max}$. At each slot, if the skyline is higher than the threshold θ, the reader will receive responses from some tags (i.e., $X = 0$), and if the skyline is lower than the threshold θ, the reader will observe an idle slot (i.e., $X = 1$). The average $\bar{X}$ of the 1-bit responses then can be utilized to estimate the total number of tags.

E. Discussion

1) Reliable Estimation With Unreliable Channel: Most existing protocols study the cardinality estimation assuming a reliable communication channel, while the wireless channel is error-prone depending on various conditions (e.g., interrogation power, communication distance, etc.). The recent protocols fail to capture the actual cardinality under unreliable channels even with the knowledge of error rates. For instance, the false detection of response signal might tamper the monotonic feature of response signal along the estimation path in PET protocol [28] and substantially degrades estimation accuracy. LoF [19] also significantly relies on the channel condition, and the estimation accuracy decreases dramatically even with a small error rate.

Due to the structure of tag responses, it becomes very challenging and complicated for existing protocols to adjust the estimation and compute an accurate result even when error rates can be measured. In contrast, since the ZOE protocol relies solely on 0/1 responses from tags and does not assume any (e.g., monotonic in [11] and [28]) patterns of the tag responses, it is inherently more robust over unreliable channels. Before the estimation process, we can measure the communication error rate between tags and readers. For example, the reader requests the tags to transmit zero-one alternating responses and compares them to the received messages to measure the error rate, assuming that the error rate is relatively stable during the short period of estimation process.

We denote the false negative rate and false positive rate to be p and q, respectively. We propose the EEA algorithm to adjust the estimation results according to the error rates.

We denote by $\bar{X}_{\mathrm{Error}}$ the average value of m independent observations with the error rates of p and q. Then, we have

$$E(\bar{X}_{\mathrm{Error}}) = E\left[\bar{X}(1-p-q)+p\right]. \tag{7}$$

According to (7), we compute $E(\bar{X})$ as follows,

$$E(\bar{X}) = \frac{E(\bar{X}_{\mathrm{Error}})-p}{1-p-q}.$$

We extend (3) and estimate the tag cardinality as follows:

$$\hat{n}_{\mathrm{Error}} = -2^{\theta}\ln\left(\frac{\bar{X}_{\mathrm{Error}}-p}{1-p-q}\right). \tag{8}$$

From (8), we find that the ideal channel condition is equivalent to the special case where $p = q = 0$. Over a totally random channel (i.e., $p = q = 0.5$), the errors completely overwhelm the measurement and estimation. Nevertheless, we can successfully compensate the communication error, if $p, q \in (0, 0.5)$.

2) Coordinating Multiple RFID Readers: In practical RFID systems, multiple RFID readers are typically used to cover a large interrogation area. Deploying more readers reduces the number of tags in the coverage of each reader, which mitigates the tag-to-tag contention. Besides, the readers without overlapping coverage can interrogate tags in parallel without reader-to-reader interference [22], [26].

Commodity RFID readers are typically connected to a powerful back-end server via high-speed Gigabit Ethernet links. As such, slot-level time synchronization among multiple readers can be easily achieved. Thus, many prior works view multiple readers coordinated by the server as a single reader [22], [28].

Some prior estimation schemes, however, suffer multiple counting problems when a tag located in the overlapped area reports to multiple readers [12], [19], [22]. In ZOE, each reader relays the commands of back-end server in such a way that a tag in the overlapped area receives the same set of parameters (e.g., random seed s, threshold θ, etc.) in each estimation round possibly from multiple readers. We note that whether a tag would respond only depends on the system parameters and its unique ID, meaning that in each synchronized slot, a tag either responds to all readers or consistently keeps silent. Therefore, the back-end server may merge tag responses by applying the logical OR on all the responses observed by multiple readers, and obtain the same result as if using a single reader [22], [28].

3) Reducing the Overhead at RFID Tags: In the basic algorithm, each tag needs to generate a random number at each estimation round. Generating random numbers, however, requires a fair amount of computation at the RFID tags. Existing approaches propose to preload multiple random numbers into tags for multiple estimation rounds and use a new random number in each estimation round, which incurs extra memory overhead. At resource-constrained RFID tags, either method in providing the randomness is far from satisfactory. Instead of using new random numbers at different estimating rounds, ZOE only preloads one 32-bit random number into each RFID tag denoted as $\mathcal{H}_B(\mathrm{id})$. In each estimation round, the back-end server generates a uniformly distributed random number denoted as R_B and broadcasts it to tags. Receiving R_B, each tag computes $\mathcal{R}(\mathrm{id}) = \min\{i|[\mathcal{H}_B(\mathrm{id}) \oplus R_B][i] = 0\}$, where $\oplus$ denotes the bitwise XOR, and participates in the estimation round with $\mathcal{R}(\mathrm{id})$. Such a method only requires the tags to perform a lightweight bitwise XOR function. The memory overhead for each tag can thus be reduced to one 32-bit random number. By this modification, we may expect nearly independent trials, and the above analysis of the ZOE protocol still holds.

V. Implementation

We implement a prototype system to validate ZOE using USRP SDR and WISP tags. Commercial off-the-shelf RFID readers only provide limited high-level APIs for developers [1]. The combination of the USRP and the WISP platform provides us full programmability to both RFID reader and tags. We implement the ZOE reader using USRP N210 software defined radio based on the GNURadio platform and the Gen2 RFID projects [3] to interrogate WISP RFID tags. The ZOE reader uses the USRP RFX900 daughterboard that operates in the 900-MHz band [4]. Due to hardware constraints, we do not implement channel hopping using USRP N210 [2]. We connect the daughterboard to Alien ALR-8696-C circular polarized antennas with the antenna gain of 8.5 dBic [1]. The typical power output of an RFX900 daughterboard is only 23 dBm (200 mW), far less than 30 dBm (1 W) of a commercial RFID reader. We bypass the SAW filter of RFX900 daughterboard to increase the transmission power in the ISM band. We connect the USRP N210 via Gigabit Ethernet to a laptop equipped with a qual-core 2.67-GHz processor and 2.9 GB memory running Ubuntu 10.10. Physical-layer responses of WISP tags are transferred to and processed at the laptop. Fig. 5 shows the testbed.

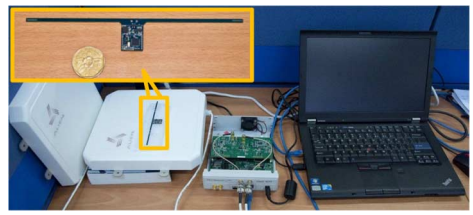


Fig. 5. Testbed: Two circular antennas are mounted to the USRP N210 software defined radio. The USRP N210 is connected via GigE to a laptop, which acts as an RFID reader. The reader interrogates WISP RFID tags.

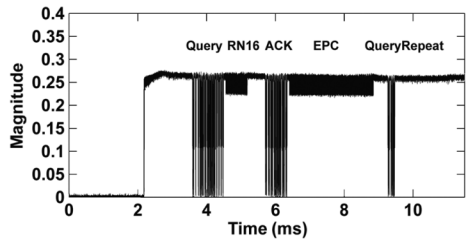


Fig. 6. Communication between reader and tag in the inventory communication. The `Query` command is sent by the reader at around 4 ms followed by the reply of RN16 from a tag. The `ACK` is sent at around 6 ms followed by the EPC code from the tag. The `QueryRepeat` is sent to query other tags.

We implement the ZOE tag using the programmable WISP tags based on the WISP4.1 hardware and firmware. The WISP tag mainly consists of an RFID circuitry and an ultra-low-power MSP430 microcontroller. The RFID circuitry is used to harvest power and respond radio signals. The WISP4.1 firmware written in C and assembly codes has partially implemented the EPCglobal C1G2 protocol [2], [5]. We extend the EPCglobal C1G2 protocol with the functionality of ZOE cardinality estimation. The implementation of the ZOE protocol only requires a slight extension to the EPCglobal C1G2 protocol.

In EPCglobal C1G2 standard [2], the RFID reader initiates each communication round between an RFID reader and tags. The reader transmits an operation code (e.g., `Query`, `Write`, `Select`, `ACK`, etc.) indicating the expected operation of tags, the backscatter bit rate, and tag encoding schemes (e.g., FM0 or Miller) [2]. Fig. 6 shows the communication between a reader and a tag in the inventory communication round where the downlink uses pulse interval encoding at 40 kHz and uplink uses Miller-4 encoding at 250 kHz. The reader initiates the communication by sending a `Query` command to the tag. When receiving a command, each tag responds according to the operation code. As the operation code is `Query` in this case, the tag transmits a 16-bit random number (RN16) back to the reader and waits for `ACK` following the EPCglobal C1G2 standard [2]. Once the reader correctly `ACK`s the RN16, the tag responds with the EPC code as depicted in Fig. 6.

We implement the ZOE protocol by following the conventional reader-initiated approach. We first add the `Count` command into the command set of the standard. To estimate the tag cardinality, the reader initiates counting procedure by sending a `Count` command along with other parameters (θ, R_B, encoding scheme, etc.). In the case that the operation code is `Count`, the tag computes $\mathcal{R}(\mathrm{id}) = \min\{i|[\mathcal{H}_B(\mathrm{id}) \oplus R_B][i] = 0\}$. If $\mathcal{R}(\mathrm{id}) \geq \theta$, the tag transmits a short response according to the encoding scheme, and keeps silent otherwise. Fig. 7 shows the

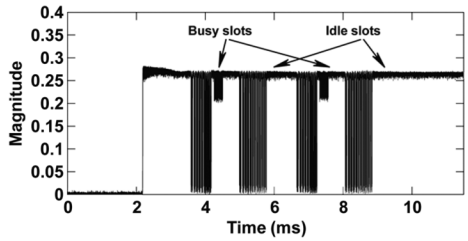


Fig. 7. Communication between reader and tag in the ZOE protocol. The first `Count` command is sent at around 4 ms followed by a busy slot. The second `Count` command is followed by an empty slot.

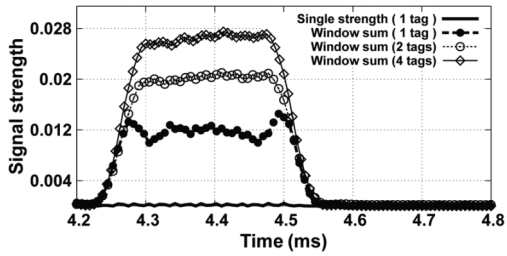


Fig. 8. Tag response detection using moving window summation of signal strength. When multiple tags respond simultaneously, the aggregated signal strength increases.

communication between the reader and the tag in four counting rounds, where the operation code is `Count` with $\theta = 1$, varying R_B, and the Miller-4 encoding scheme. In Fig. 7, we can see that two short responses follow the first and the third `Count` commands at around 4 and 7 ms, respectively; while no response follows the second and the fourth `Count` commands. One may notice that the first `Count` command takes a slightly longer time than the second `Count` command. The reason is that RFID reader uses the pulse interval encoding scheme, in which bit-1 takes twice the transmission time of bit-0. As the reader generates different R_B for each `Count` command, the transmission time varies slightly across the commands.

To send a short response, a tag simply transmits a single tone (at 250 kHz) that allows robust response detection at readers. We first feed the signals into a bandpass filter with center frequency of 250 kHz to remove most background noise. We use the standard moving window summation (with window width of 64) to smoothen out any sudden changes due to noises in the band. If the signal strength exceeds the mean plus three standard deviations (i.e., 99.7% confidence level), we say the channel is busy, and idle otherwise. Fig. 8 shows the signal strength around the frequency band of 250 kHz and the moving window summation of the tag response following the first `Count` command approximately between 4.25 and 4.5 ms. We observe a big jump of moving window summation during the tag response period (4.25–4.5 ms), while the sum is small and flat when no tag response is transmitted (e.g., after 4.6 ms). As shown in Fig. 8, when multiple tags respond simultaneously using on–off keying, the aggregated signal strength still provides valid indications of tag responses.

Although ZOE can run in real time on the USRP N210 in concert with the WISP RFID tags, in small-scale RFID systems, it is sufficient to collect tag IDs and derive an accurate tag number. In the following, we focus on the large-scale simulations to compare ZOE to the existing cardinality estimation schemes. This is for two reasons. First, partially due to the complexity of existing cardinality estimation schemes, such approaches have not yet been successfully implemented on programmable RFID tags. Second, we want to compare the schemes in various complex settings, such as error-free and error-prone channel conditions, and varying number of tags. Besides, programming, debugging, and testing a large number of programmable RFID tags still remain challenging.

VI. Evaluation

We conduct extensive simulations under various scenarios to study the performance of the ZOE protocol. We first investigate the estimation accuracy and the corresponding processing cost of ZOE. We then compare ZOE to the most recent approaches FNEB [11], LoF [19], PET [28], and ART [22] in terms of the time efficiency, as well as computation and memory overhead at tags. We further investigate the estimation performance of different protocols over noisy channels.

A. Simulation Setting and Performance Metrics

We first focus on the ideal communication channel (i.e., no transmission error occurs between RFID tags and RFID readers), and the reader is capable of correctly detecting the responses from tags. After that, we evaluate the robustness and reliability of the estimation protocols with unreliable channel conditions. For all simulation instances, we repeat 300 runs and report the average if not explicitly specified otherwise.

The estimation accuracy is one of the most important metrics for an estimator. Consistent with existing works, we use the same accuracy metric as studied in [11], [28]

$$\text{Accuracy} = E\left(\frac{\hat{n}}{n}\right)$$

where $\hat{n}$ denotes the estimation result and n refers to the actual number of tags. This metric evaluates the estimation accuracy and bias. An ideal estimator is expected to return an estimation result close to the actual value. The closer it is to 1, the higher the estimation accuracy is.

We use the standard deviation to measure the estimation precision

$$\sigma = \sqrt{E\left[(\hat{n} - n)^2\right]}$$

where the operator $E[.]$ denotes the average of all runs. A high standard deviation indicates the estimation results spread out, whereas a low standard deviation means the estimation results concentrate. Therefore, we expect an ideal estimator with a low standard deviation.

Given the accuracy requirement of (ε, δ)-approximation, we examine the estimating time that it takes to meet the requirement. Since the data rate varies depending on various factors (e.g., PHY/MAC implementations and time-varying channel conditions, etc.), same as the benchmark approaches, we abstract the estimation time with the number of total time-slots that each protocol consumes for fair comparison.

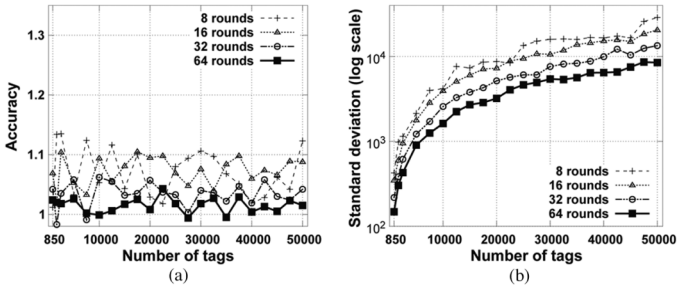


Fig. 9. Performance of ZOE with different numbers of estimating rounds. (a) Estimation accuracy. (b) Standard deviation.

Finally, another metric we consider is the computation and memory overhead at RFID tags. We measure the overhead by comparing the quantity of random numbers generated or stored at RFID tag side.

B. Proposed Protocol Investigation

We demonstrate that the ZOE protocol provides tunable estimation accuracy at the cost of processing time. Fig. 9(a) depicts different estimation accuracies while different numbers of estimation rounds are applied. The threshold θ is set at the optimal value for all cases. The figure suggests that one can improve the estimation accuracy by running additional rounds of estimation. By repeating 64 rounds of estimation, ZOE already achieves the accuracy very close to 1 regardless of the actual tag cardinality, which suggests that the tag cardinality has little impact on the estimation accuracy.

Fig. 9(b) illustrates the standard deviation that indicates the precision of the estimator. The figure suggests that one can reduce the standard deviation and thus improve the estimation accuracy by performing extra estimation rounds. With 64 estimation rounds, ZOE achieves standard deviation less than 20% of total RFID tag number, i.e., it achieves less than 0.2 of normalized standard deviation.

We investigate the estimation accuracy for different communication error rates. The estimation round m is fixed at 64 in all experiments. In the cases that $m \neq 64$, the simulation results suggest similar trends. Fig. 10 plots the estimation accuracy with and without the EEA algorithm, respectively. As shown in Fig. 10, the estimation accuracy of the basic ZOE protocol degrades dramatically with the increase of the communication error, whereas the estimation accuracy with EEA remains reliable with various error rates. That is because EEA takes into the consideration the communication error and incorporates such information into the estimation.

C. Performance Comparison

In the following, we compare the performance of ZOE to the representative estimation protocols, FNEB, LoF, and PET. As the existing approaches do not tolerate communication errors, we first focus on the performance comparison with the ideal channel.

Given the same estimation accuracy requirement of ($\varepsilon = 5\%$, $\delta = 1\%$)-approximation, we compare the time-slots that each estimation protocol takes to achieve the accuracy requirement to varied number of tags ranging from 850 to 50 000 in Fig. 11. For the proposed ZOE protocol, the entire estimation process

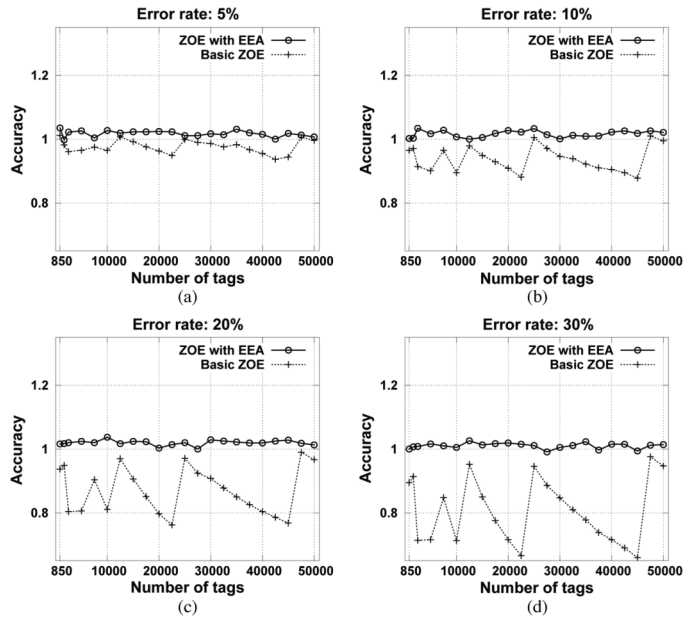


Fig. 10. Evaluation of estimation accuracy under different error rates: (a) error rate $= 5\%$; (b) error rate $= 10\%$; (c) error rate $= 20\%$; (d) error rate $= 30\%$.

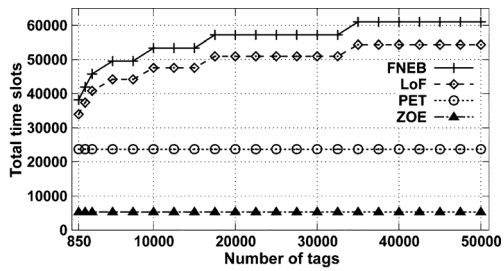


Fig. 11. Performance comparison to estimate different tag cardinality with the same accuracy requirement of ($\varepsilon = 5\%$, $\delta = 1\%$)-approximation.

consists of the time-slots to select a suitable threshold and m time-slots to improve the accuracy. We add the time-slots for the two stages and present the sum. For other benchmark protocols, we assume that they have the upper bound of tag population (e.g., $\lceil \log_2(n) \rceil$) as prior knowledge so they can optimize their settings. In the figure, we see that ZOE takes substantially less time compared to benchmark schemes. In particular, ZOE only needs approximately 22% of time-slots that PET needs to achieve the same accuracy requirement. Compared to FNEB and LoF, ZOE yields higher performance gain when the RFID network further scales up. The results suggest that while benchmarks need more time-slots to estimate more tags, the tag cardinality has little impact on estimation efficiency for ZOE. In the following, we compare the estimating time-slots that each protocol takes to achieve different accuracy requirements to the fixed tag cardinality of 50 000.

We first keep the error probability $\delta = 1\%$ fixed and vary the confidence interval from 5% to 20%. Fig. 12(a) plots the total time-slots needed by each protocol. Then, we keep the confidence interval $\varepsilon = 5\%$ fixed and vary the error probability δ from 1% to 15%, and the simulation results are presented in Fig. 12(b). According to the simulation results, ZOE only consumes about 31% processing time of PET to provide the same

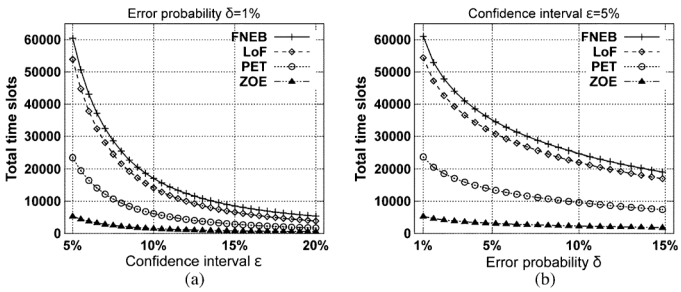


Fig. 12. Performance comparison: (a) protocol performance with different confidence interval ε, and the same error probability $\delta = 1\%$; (b) protocol performance with different error probability δ and the same confidence interval $\varepsilon = 5\%$.

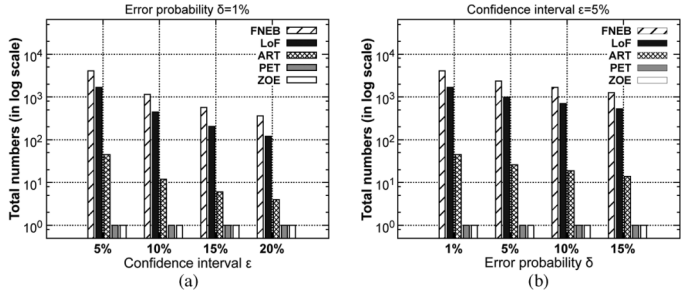


Fig. 14. Memory overhead in storing the random numbers: (a) with different confidence interval ε, and the same error probability $\delta = 1\%$; (b) with different error probability δ, and the same confidence interval $\varepsilon = 5\%$.

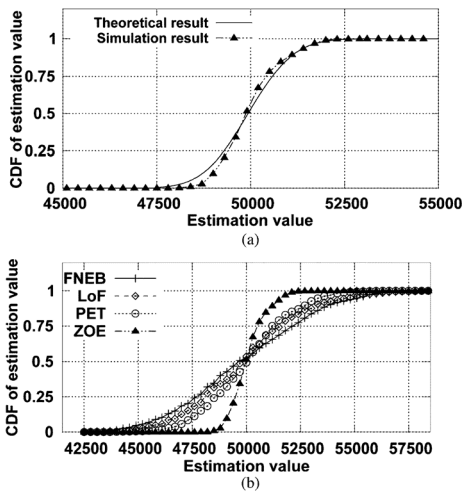


Fig. 13. Cumulative distribution of estimation results: (a) theoretical performance versus simulation results of ZOE; (b) cumulative distribution comparison of different protocols.

TABLE I
COMPARISON OF TIME-SLOTS NEEDED TO ACHIEVE ($\varepsilon = 5\%$, $\delta = 1\%$)-APPROXIMATION WITH VARYING ESTIMATED UPPER BOUNDS. THE ACTUAL NUMBER OF TAGS IS 50 000

Upper bound	30000	35000	40000	45000	50000
ART	13900	11500	9900	7900	6400
ZOE	5300	5300	5300	5300	5300

estimation accuracy, which translates to more than 3× performance improvement in terms of time efficiency and even more compared to LoF and FNEB. We can infer from the simulation results that provided the same amount of processing time, the estimation accuracy of ZOE should be more accurate. As the ZOE protocol features $\mathcal{O}(1)$ estimation efficiency, the time efficiency improvement of ZOE will be more remarkable over existing protocols when the RFID network scales up.

Fig. 13(a) presents the simulated performance against the analytical performance of the proposed ZOE protocol given the accuracy requirement of $\varepsilon = 5\%$ and $\delta = 1\%$ with the actual tag cardinality of 50 000. In Fig. 13(a), we observe that the simulation results match the analytical performance. Almost all the estimated values fall into the 5% interval [47 500, 52 500]. We examine the small portion of estimated numbers $\hat{n} \notin [47\,500, 52\,500]$ and find them very close to the expected range.

We provide PET, FNEB, and LoF the same amount of time-slots to estimate the actual tag cardinality of 50 000 and present the distributions in Fig. 13(b). According to the simulation results, we find that the estimation results of ZOE are much more concentrated about the actual cardinality. Moreover, the number of outliers is much smaller than those of PET, FNEB, and LoF. In particular, with the same processing time that 99% estimation results fall into the confidence interval [47 500, 52 500] in ZOE, the existing approaches can only guarantee less than 80% results within such an interval.

In the following, we compare against the most recent estimation protocol ART [22]. ART uses the average run size of identical responses as an estimator. ART outperforms many existing protocols in terms of time efficiency [22] because ART has significantly smaller variances. ART, however, needs the upper bound of tag numbers to set several key parameters, e.g., frame size, persistence probability, etc. When the upper bound is not available or the upper bound estimation is inaccurate, ART takes more time-slots to meet an accuracy requirement.

Table I compares the time-slots needed to achieve the same ($\varepsilon = 5\%$, $\delta = 1\%$)-approximation to varying estimated upper bounds. The actual number of tags is 50 000. According to the results, we find that ZOE can reduce the time-slots by approximately 20% compared to ART even when ART accurately estimates the upper bound of number of tags in advance. We find that ART takes more time-slots when it underestimates the tag population in parameter settings. In comparison, ZOE does not have to know the upper bound in advance.

We compare the computation and memory overhead at RFID tags. We examine the memory overhead and compare ZOE to recent protocols in Fig. 14. We fix the error probability $\delta = 1\%$ and vary the confidence interval ε from 5% to 20% in Fig. 14(a). We vary the error probability δ from 1% to 15% with fixed confidence interval $\varepsilon = 5\%$ in Fig. 14(b). We find that ZOE and PET consume constant small storage and outperform other schemes that require larger memory cost.

Until now, we focus on the performance comparison over ideal channels. In Fig. 15, we examine the estimation accuracy of ZOE compared to recent approaches with different error rates. We vary the error rate from 5% to 30%, and the actual tag cardinality is 50 000. According to Fig. 15, we find that the estimation accuracies of LoF and PET are significantly biased from the actual value. Though FNEB and ART are more robust than LoF and PET, FNEB and ART still fail to provide unbiased results. On the other hand, ZOE with EEA resists the various

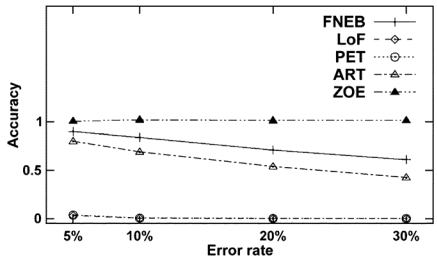


Fig. 15. Accuracy comparisons under varying error rates.

error rates and provides accurate estimation results even when the error rate reaches 30%.

VII. Conclusion

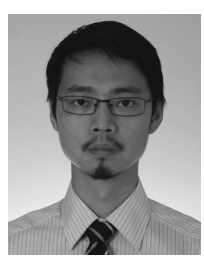
In this paper, we propose a cardinality estimation protocol based on Zero-One Estimator, which improves the estimation time efficiency in meeting arbitrary accuracy requirement. ZOE only requires 1-bit response from the RFID tags per estimation round. Moreover, ZOE rapidly converges to optimal parameter configurations and achieves high estimation efficiency. We enhance the robustness of cardinality estimation over noisy channels. We implement a prototype system based on the GNU-Radio/USRP platform in concert with the WISP RFID tags. ZOE only requires slight updates to the EPCglobal C1G2 standard. We also conduct extensive simulations to evaluate the performance of ZOE in large-scale settings. The results demonstrate that ZOE outperforms the most recent cardinality estimation protocols.

References

[1] Alien Technology, Morgan Hill, CA, USA, "Alien Technology," [Online]. Available: http://www.alientechnology.com
[2] EPCglobal, Brussels, Belgium, "Class 1 Generation 2 UHF air interface protocol standard "Gen 2"," [Online]. Available: http://www.epcglobalinc.org/standards/uhfc1g2
[3] "Gen 2 RFID tools," [Online]. Available: https://www.cgran.org/wiki/Gen2
[4] Ettus Research, Santa Clara, CA, USA, "Ettus Research," [Online]. Available: http://www.ettus.com
[5] "WISP platform," [Online]. Available: http://wisp.wikispaces.com
[6] M. Buettner and D. Wetherall, "An empirical study of UHF RFID performance," in *Proc. ACM MobiCom*, 2008, pp. 223–234.
[7] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inf. Theory*, vol. IT-25, no. 5, pp. 505–515, Sep. 1979.
[8] S. Chen, M. Zhang, and B. Xiao, "Efficient information collection protocols for sensor-augmented RFID networks," in *Proc. IEEE INFOCOM*, 2011, pp. 3101–3109.
[9] K. Finkenzeller, *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*. New York, NY, USA: Wiley, 2000.
[10] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, 3rd ed. Oxford, U.K.: Oxford Univ. Press, 2001.
[11] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID tags efficiently and anonymously," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.
[12] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. ACM MobiCom*, 2006, pp. 322–333.
[13] M. Kodialam, T. Nandagopal, and W. C. Lau, "Anonymous tracking using RFID tags," in *Proc. IEEE INFOCOM*, 2007, pp. 1217–1225.
[14] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proc. IEEE MobiQuitous*, 2005, pp. 166–172.
[15] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. ACM MobiHoc*, 2010, pp. 1–10.
[16] T. Li, S. Wu, S. Chen, and M. Yang, "Energy efficient algorithms for the RFID estimation problem," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.
[17] T. Li, S. Wu, S. Chen, and M. Yang, "Generalized energy-efficient algorithms for the RFID estimation problem," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1978–1990, Dec. 2012.
[18] L. M. Ni, Y. Liu, Y. C. Lau, and A. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, 2004.
[19] C. Qian, H. Ngan, and Y. Liu, "Cardinality estimation for large-scale RFID systems," in *Proc. IEEE PerCom*, 2008, pp. 30–39.
[20] Y. Qiao, S. Chen, T. Li, and S. Chen, "Energy-efficient polling protocols in RFID systems," in *Proc. ACM MobiHoc*, 2011, Art. no. 25.
[21] L. G. Roberts, "Aloha packet system with and without slots and capture," *Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28–42, 1975.
[22] M. Shahzad and A. X. Liu, "Every bit counts: Fast and scalable RFID estimation," in *Proc. ACM MobiCom*, 2012, pp. 365–376.
[23] J. R. Smith, A. P. Sample, P. S. Powledge, S. Roy, and A. Mamishev, "A wirelessly-powered platform for sensing and computation," in *Proc. ACM UbiComp*, 2006, pp. 495–506.
[24] R. Want, "An introduction to RFID technology," *IEEE Pervasive Comput.*, vol. 5, no. 1, pp. 25–33, Jan.–Mar. 2005.
[25] L. Yang, J. Han, Y. Qi, C. Wang, T. Gu, and Y. Liu, "Season: Shelving interference and joint identification in large-scale RFID systems," in *Proc. IEEE INFOCOM*, 2011, pp. 3092–3100.
[26] R. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast identification of the missing tags in a large RFID system," in *Proc. IEEE SECON*, 2011, pp. 278–286.
[27] Y. Zhang, L. T. Yang, and J. Chen, *RFID and Sensor Networks: Architectures, Protocols, Security and Integrations*. New York, NY, USA: Auerbach, 2010.
[28] Y. Zheng and M. Li, "PET: Probabilistic estimating tree for large-scale RFID estimation," *IEEE Trans. Mobile Comput.*, vol. 11, no. 11, pp. 1763–1774, Nov. 2012.
[29] Y. Zheng and M. Li, "Fast tag searching protocol for large-scale RFID systems," in *Proc. IEEE ICNP*, 2011, pp. 363–372.
[30] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, "Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems," in *Proc. ISLPED*, 2004, pp. 357–362.

Yuanqing Zheng (S'11) received the B.S. degree in electrical engineering and M.E. degree in communication and information system from Beijing Normal University, Beijing, China, in 2007 and 2010, respectively, and is currently pursuing the Ph.D. degree in computer engineering at Nanyang Technological University, Singapore.

His research interests include distributed systems and pervasive computing.



Mo Li (M'06) received the B.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 2004, and the Ph.D. degree in computer science and engineering from Hong Kong University of Science and Technology, Hong Kong, in 2009.

He is currently an Assistant Professor with the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include wireless sensor networking, pervasive computing, and mobile and wireless computing.

Dr. Li is a member of the Association for Computing Machinery (ACM). He won the ACM Hong Kong Chapter Prof. Francis Chin Research Award in 2009 and the Hong Kong ICT Award Best Innovation and Research Grand Award in 2007.