

ZOE: Fast Cardinality Estimation for Large-Scale RFID Systems

Yuanqing Zheng, Mo Li

School of Computer Engineering, Nanyang Technological University, Singapore
{yuanqing1, limo}@ntu.edu.sg

Abstract—Estimating the RFID cardinality with accuracy guarantee is an important task in large-scale RFID systems. This paper proposes a fast RFID cardinality estimation scheme. The proposed Zero-One Estimator (ZOE) protocol rapidly converges to optimal parameter settings and achieves high estimation efficiency. ZOE significantly improves the cardinality estimation efficiency, achieving 3x performance gain compared with existing protocols. Meanwhile, ZOE guarantees arbitrary accuracy requirement without imposing computation and memory overhead at RFID tags. Due to the simplicity and robustness, the ZOE protocol provides reliable cardinality estimation even over noisy channel. We implement a prototype system using the USRP software defined radio and Intel WISP RFID tags. We also evaluate the performance of ZOE with extensive simulations. The evaluation of ZOE shows encouraging results in terms of estimation accuracy, time efficiency, as well as robustness.

I. INTRODUCTION

Radio Frequency Identification (RFID) systems [9] have recently received significant interests from both industry and academia. A large-scale RFID system usually consists of multiple RFID readers and a huge amount of RFID tags [23]. 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, 14, 17, 22, 25], logistics [26, 28], object tracking [15, 18], etc.

Fast estimating the cardinality of RFID tags, accordingly the number of labeled items, is of primary importance in many applications, e.g., estimating the number of conference attendees with RFID badges. The estimated RFID cardinality with guaranteed accuracy may also serve as primary inputs to upper-layer RFID protocols. For instance, Aloha-based RFID identification protocols can achieve near-optimal performance if the frame size is set according to the number of tags.

In order to achieve efficient RFID cardinality estimation, probabilistic estimation approaches have been proposed which aim to estimate the approximate number of RFID tags. Some recent approaches achieve $\mathcal{O}(\log n)$ estimation efficiency to the number of RFID tags n [11, 16]. One most recent protocol, Probabilistic Estimation Tree (PET), achieves $\mathcal{O}(\log \log n)$ time efficiency for each estimation round [27]. Nevertheless, existing protocols require many independent estimation rounds so as to achieve high accuracy on the estimation results.

Further improving the time efficiency of each estimation round will significantly benefit the entire cardinality estimation process, meet the stringent time requirement of many realtime applications, and support larger scale of RFID systems. While pursuing the estimation efficiency at the optimum, we are also aiming at reducing the processing overhead of resource-constrained RFID tags. Most existing probabilistic approaches require generating large volume of random numbers or alternatively pre-storing them at RFID tags, which leads to heavy computation and storage burden for RFID tags. We aim at shifting such processing overhead from resource-constrained RFID tags to the much more 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.

In this paper, we present Zero-One Estimator (ZOE): a simple yet fast RFID cardinality estimation protocol with guaranteed accuracy requirement. ZOE first tunes the system parameters and converges to optimal settings with a bisection search with small overhead. With the optimized parameter settings, ZOE estimates the RFID cardinality with only one bit feedback from tags at each round providing extremely high estimation efficiency. Through simply XORing the original random number stored at each RFID tag with a common random number generated by RFID readers, ZOE shifts the burden of generating randomness from lightweight tags to relatively powerful readers. We further take the noisy wireless channel into consideration and propose Error Estimation and Adjustment (EEA) algorithm to adjust estimation results according to the practical communication error rates.

We implement a prototype system using the Universal Software Radio Peripheral (USRP) [4] in concert with the Intel Wireless Identification and Sensing Platform (WISP) [21]. We implement the ZOE reader functionality with USRP Software 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 to study its performance in large-scale RFID systems.

The rest of this paper is organized as follows. We first 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 detailed description on ZOE in Section IV. We present the implementation of ZOE in Section V. We evaluate the performance of ZOE with extensive simulations in Section VI. Section VII concludes this paper.

II. RELATED WORK

One closely related problem is RFID identification which aims at collecting the tag IDs of all RFIDs in the interrogation area [7, 19, 20, 24, 29]. The identification protocols based on collision arbitration can generally be classified into two categories: Ahola-based protocols [20] and Tree-based protocols [7, 29]. 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. Rather than identifying all the tags, probabilistic estimation protocols estimate the number of tags which meets the 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.* propose Enhanced Zero-Based (EZB) estimator to estimate relatively large number of tags [13].

Recent probabilistic estimation approaches achieve $\mathcal{O}(\log n)$ estimation efficiency to the total number of RFID tags n . Han *et al.* propose the First Non-Empty slot Based (FNEB) estimator with binary search method [11]. Qian *et al.* [16] 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 large volume of random numbers and respond accordingly. One most recent protocol, Probabilistic Estimating Tree (PET) based estimator [27], advances the estimation efficiency and achieves $\mathcal{O}(\log \log n)$ processing time efficiency to the tag cardinality n . Such approaches only consider an ideal wireless channel which is free of transmission error.

III. SYSTEM MODEL AND PROBLEM DESCRIPTION

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 backend server connecting the readers. Multiple RFID readers are normally deployed to ensure a full coverage in large-scale RFID systems. The backend server coordinates the RFID readers and initiates the cardinality estimation process. The RFID readers relay the commands received from the backend server and broadcast to tags, and later report the tags' responses back to the server. When coordinated and synchronized, the multiple readers can logically be considered as one RFID reader. The RFID system may use lightweight passive RFID tags or powerful active ones.

RFID systems may operate over a wide variety of frequency bands (e.g., 13.56/433/900MHz). We exclusively focus on the RFID systems operating in the 900MHz ultra-high frequency band. We assume that the RFID system works on 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 according to the reader's command. 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 realtime requirement, the estimation protocol should compute an accurate tag cardinality in an efficient manner. Since the cardinality of RFID tags could easily grow up to tens of thousands (e.g., a typical port inventory application concerns hundreds of containers, each of which contains a large number of products), we need to design a scalable and efficient estimation approach that can guarantee the customized accuracy requirement.

Consistent with the existing approaches [11, 16, 27], 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 $\Pr\{|\hat{n} - n| \leq \epsilon n\} \geq 1 - \delta$. For example, when the actual tag cardinality is 10000, $(\epsilon = 5\%, \delta = 1\%)$ -approximation expects an estimation result within the interval [9500, 10500] with a probability of 99% and above.

We abstract the time efficiency with the total time slots used to estimate the cardinality. Most recent approaches only need to distinguish an idle slot from a busy slot [11, 16, 27]. The smaller number of time slots means the shorter processing 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 channel conditions with false negatives and false positives. 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. Table I summarizes the key notations used across this paper.

TABLE I
KEY NOTATIONS

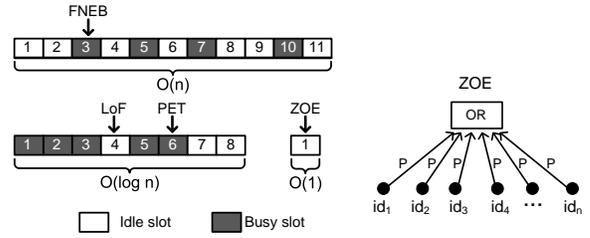
Symbols	Descriptions
n	Actual number of tags
\hat{n}	Estimated number of tags
m	Estimation rounds
$\mathcal{H}(\cdot)$	A uniform hash function
$\mathcal{H}_B(\cdot)$	Binary representation of $\mathcal{H}(\cdot)$
$\mathcal{R}(\cdot)$	Position of right-most zero in $\mathcal{H}_B(\cdot)$
θ	A threshold affecting the behavior of tags
λ	Load factor $n/2^\theta$
q	Channel error rate

A. Principle

The conventional approaches take advantage of the frame-slotted Aloha protocol to estimate the number of tags. The RFID reader normally needs to examine the state of each time slot in the frame. Figure 1(a) presents illustrative examples of most recent protocols, e.g., FNEB [11], LoF [16], and PET [27]. In FNEB, each tag randomly selects a slot from $\mathcal{O}(n)$ slots with uniform distribution functions to send a response in each estimation round. If the frame size is fixed, then 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 by examining $\mathcal{O}(\log n)$ slots. 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 nearly $1/2^i$ of tags respond in the i th 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 one estimation round might dramatically deviate from the expected value. Existing protocols thus need many independent estimation rounds to derive an average to accurately approximate 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 Figure 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 with a priori knowledge of P .

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, thereby achieving 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 conventional approaches. Towards the more efficient



(a) Estimation frame and interested slot (b) One aggregated slot
Fig. 1. An illustrative comparison of ZOE with conventional schemes of FNEB, LoF and PET.

cardinality estimation, we present detailed theoretical analysis and consolidate the aforementioned design principle with a prototype system in the following.

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 $\mathcal{H}(\text{id}, s)$, where s denotes a random seed. For simplicity, we omit the notation of s for the hash functions. We denote by $\mathcal{H}_B(\text{id})$ the binary representation of $\mathcal{H}(\text{id})$. We also denote by $\mathcal{R}(\text{id})$ the index of the right-most zero bit in $\mathcal{H}_B(\text{id})$ as follows

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

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

Let the random variable of $\mathcal{R}(\text{id}_i)$ 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 $\mathcal{H}_B(\text{id})$ 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 $1 - \Pr(R_i < \theta) = \frac{1}{2^\theta}$.

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 $\mathcal{R}(\text{id}_i) < \theta$, $\forall i \in \{1, 2, \dots, n\}$, 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

Algorithm 1 ZOE algorithm for RFID readers

```
1:  $m \leftarrow \lceil \frac{c\sigma(X)_{\max}}{e^{-\lambda}(1-e^{-\varepsilon\lambda})} \rceil^2$ 
2: Initiate the estimation, broadcast  $\theta$ 
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:  $\bar{X} \leftarrow \frac{1}{m} \sum_{i=1}^m X_i$ 
12: return  $\hat{n} \leftarrow -2^\theta \ln \bar{X}$ 
```

Algorithm 2 ZOE algorithm for each RFID tag

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

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 which 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

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

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

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

The maximum standard deviation of X is

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

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

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

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

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

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

Since the result may vary slightly because of the estimation variance, we seek a guaranteed cardinality estimation result, e.g., $\Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$. The estimation accuracy

requirement can be represented as follows

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

We define a random variable $Y = \frac{\bar{X} - \mu}{\sigma}$, where $\mu = E(\bar{X}) = e^{-\lambda}$, and $\sigma = \sigma(\bar{X}) = \frac{\sigma(X)}{\sqrt{m}}$. 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

$$1 - \delta = \Pr\{-c \leq Y \leq c\} = \text{erf}\left(\frac{c}{\sqrt{2}}\right),$$

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

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

According to (4), we have

$$\begin{aligned} m &\geq \max\left\{\left[\frac{c\sigma(X)_{\max}}{e^{-\lambda}(1 - e^{-\varepsilon\lambda})}\right]^2, \left[\frac{c\sigma(X)_{\max}}{e^{-\lambda}(e^{\varepsilon\lambda} - 1)}\right]^2\right\} \\ &\geq \left[\frac{c\sigma(X)_{\max}}{e^{-\lambda}(1 - e^{-\varepsilon\lambda})}\right]^2. \end{aligned} \quad (5)$$

Therefore, with such m estimation frames, ZOE can guarantee the accuracy requirement of $\Pr\{|\hat{n} - n| \leq \varepsilon n\} \geq 1 - \delta$. Algorithm 1 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 (line 3-4), and records the tags' responses (line 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).

Each tag performs simple tasks as regulated in Algorithm 2. In each estimation round, when receiving a random seed s , the tag computes the random number $\mathcal{R}(\text{id})$ according to (1). The tag keeps silent or responds to the reader according to $\mathcal{R}(\text{id})$ and the threshold θ . If $\mathcal{R}(\text{id}) \geq \theta$ the tag sends a response, and otherwise keeps silent (line 2-9).

Yet one problem remains. 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 guarantee the optimal performance of ZOE.

C. Parameter setting

Before we perform the estimation process, we need to set the threshold θ which 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 $\frac{c\sigma(X)_{\max}}{e^{-\lambda}(1 - e^{-\varepsilon\lambda})}$ in (5). To reduce the number

Algorithm 3 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 1
5:   if  $\bar{X} \geq \frac{e^{-2} + e^{-1}}{2}$  and  $\bar{X} \leq \frac{e^{-0.5} + e^{-1}}{2}$  then
6:      $\theta \leftarrow mid$ ; break;
7:   end if
8:   if  $\bar{X} > \frac{e^{-0.5} + e^{-1}}{2}$  then
9:      $high \leftarrow mid$ 
10:  else
11:     $low \leftarrow mid$ 
12:  end if
13: end while
14: return  $\theta$ 

```

of estimation rounds, we maximize the denominator $f(\lambda)$ since the numerator $c\sigma(X)_{\max}$ is constant given an accuracy requirement. We compute the first order derivative of $f(\lambda) \approx e^{-\lambda}\varepsilon\lambda$ as follows,

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

According to (6), we find 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 decrements 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 increments the threshold to decrease the probability that tags would send responses.

The expected value of \bar{X} is monotonically non-decreasing 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 3 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} (line 8-12). \bar{X} is computed according to Algorithm 1 with the parameters $\theta = mid$ (line 4). Finally, the two ends converge, and the average is used as the threshold θ (line 2). When the intermediate value \bar{X} becomes very close to the target average e^{-1} , the parameter setting process terminates and $\theta = mid$ is used as the threshold (line 5-7).

Figure 2(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 Figure 2(a), we see that the experiment consists of 4 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

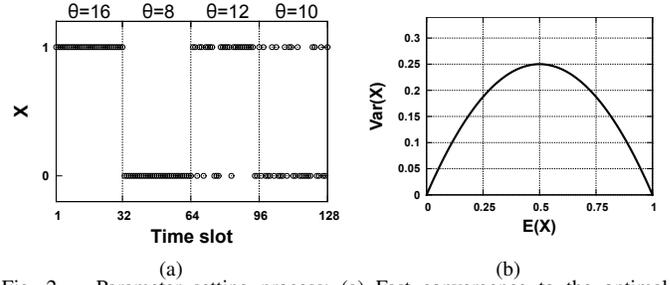


Fig. 2. Parameter setting process: (a) Fast convergence to the optimal threshold value with bisection search method; (b) When $E(X) \rightarrow 0$ or $E(X) \rightarrow 1$, the variance is very small.

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 Figure 2(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 $\theta = (8 + 12)/2 = 10$, in which case the reader also 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.

Here, we elaborate why the empirical number of 32 trials is sufficient for the threshold setting. Figure 2(b) plots the variance of X against the expectation of X . We find that when the expectation $E(X) \rightarrow 0$ or $E(X) \rightarrow 1$, the variance $Var(X) \rightarrow 0$, 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 converges to an optimal parameter setting at a very small cost. Using such an optimal threshold, we can estimate the accurate cardinality with minimal number of estimation rounds achieving higher overall estimation efficiency.

D. Discussion

1) *Reliable estimation with unreliable channel*: Most existing protocols study the cardinality estimation with reliable communication channel, while the wireless channel is error-prone depending on various conditions. The recent protocols fail to capture the actual cardinality over noisy 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 [27], and substantially degrades estimation accuracy. LoF [16] also relies on channel qualities, and estimation accuracy decreases dramatically even with a small error rate.

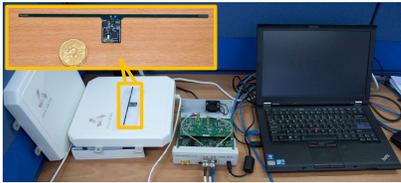


Fig. 3. Testbed: 2 circular antennas are mounted to the USRP N210 software defined radio. The USRP N210 is connected via GigE to a laptop which acts as an RFID reader. The reader interrogates WISP RFID tags.

We assume that the error rate is time-invariant during the short period of estimation process. We consider the false negative rate and false positive rate are both q for simplicity (i.e., the chances of missing a tag response and triggering a false detection are equally likely). One may easily extend it to the cases of asymmetric false positive/negative rates. We propose Error Estimation and Adjustment (EEA) algorithm to adjust the estimation results according to error rates.

We denote by \bar{X}_{Error} the average value of m independent observations with the error rate q . Then we have

$$E(\bar{X}_{\text{Error}}) = E(\bar{X} - q(2\bar{X} - 1)). \quad (7)$$

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

$$E(\bar{X}) = \frac{E(\bar{X}_{\text{Error}}) - q}{1 - 2q}.$$

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

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

From (8), we find that the ideal channel condition is the special case where $q = 0$. When $q = 0.5$, the communication becomes totally random, i.e., the channel noise completely overwhelms the measurement. Nevertheless, we can successfully compensate the communication error, if $0 < q < 0.5$.

2) *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 random numbers into tags, which requires extra memory. 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, a 32-bit random number is stored at each RFID tag during tag manufacturing and used for all estimation rounds. Many off-the-shelf methods are available for manufacturers to preload the random numbers to tags [27]. The backend server generates a uniformly distributed random number at each estimation round and broadcasts it to RFID tags. We denote the random number in the binary form as R_B . Receiving the random number, each tag computes $\mathcal{R}(\text{id}) = \min\{i | [\mathcal{H}_B(\text{id}) \oplus R_B][i] = 0\}$, where \oplus denotes the bitwise XOR operation, and participates in the estimation round with $\mathcal{R}(\text{id})$. Such a method only requires the tags to perform a lightweight bitwise XOR function. 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 RFID manufacturers

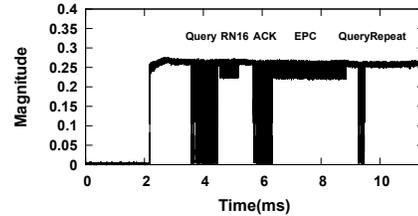


Fig. 4. The communication between reader and tag in the inventory communication. The `Query` command is sent by the reader at around 4ms followed by the reply of RN16 from a tag. The ACK is sent at around 6ms followed by the EPC code from the tag. The `QueryRepeat` is sent to query other tags.

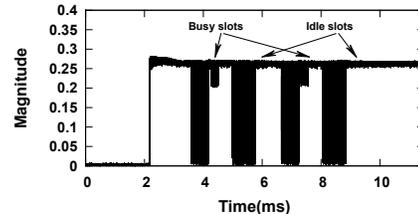


Fig. 5. The communication between reader and tag in the ZOE protocol. The first `Count` command is sent at around 4ms followed by a busy slot. The second `Count` command is followed by an empty slot.

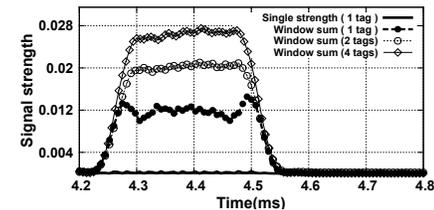


Fig. 6. The tag response detection using moving window summation of signal strength. When multiple tags respond simultaneously, the aggregated signal strength increases.

only provide limited interfaces to developers. 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 based on the GNURadio platform and the Gen2 RFID projects [3] to interrogate WISP RFID tags. The ZOE reader uses the USRP RFX900 daughterboard which operates in the 900MHz band [4]. We connect the daughterboard to Alien ALR-8696-C circular polarized antennas with the antenna gain of 8.5dBic [1]. The typical power output of RFX900 daughterboard is only 23dBm (200mW), far less than 30dBm (1W) of commercial RFID reader. We connect the USRP N210 via Gigabit Ethernet to a laptop equipped with a qual-core 2.67GHz processor and 2.9GB memory running Ubuntu 10.10. Figure 3 shows the testbed.

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 16-bit MSP430 microcontroller. The RFID circuitry is used to harvest power and backscatter radio signals. Current WISP firmware 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.

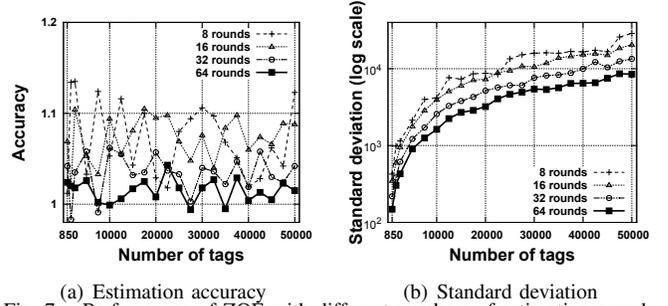
In EPCglobal C1G2 standard [2], the RFID reader initiates

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]. Figure 4 shows the communication between a reader and a tag in the inventory communication round where the downlink uses pulse interval encoding at 40kHz and uplink uses Miller-4 encoding at 250kHz. 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 ACKs the RN16, the tag responds with the EPC code as depicted in Figure 4.

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 $\mathcal{R}(\text{id}) = \min\{i | [\mathcal{H}_B(\text{id}) \oplus R_B][i] = 0\}$. If $\mathcal{R}(\text{id}) \geq \theta$, the tag transmits a short response according to the encoding scheme, and keeps silent otherwise. Figure 5 shows the communication between the reader and the tag in 4 counting rounds, where the operation code is Count with $\theta = 1$, varying R_B , and the Miller-4 encoding scheme. In Figure 5, we can see that two short responses follow the first and the third Count commands at around 4ms and 7ms, respectively; while no response follows the second and the fourth Count commands. One may notice that the the first Count command takes 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 of 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 in Figure 5.

To send a short response, a tag simply transmits a single tone (at 250kHz) which allows simple yet robust detection at readers. We first feed the signals into a bandpass filter with center frequency of 250kHz to remove most background noise. We use the standard moving window summation (with 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. Figure 6 shows the signal strength around the frequency band of 250kHz and moving window summation of the tag response following the first Count command approximately between 4.25ms and 4.5ms. We observe a big jump of moving window summation during the tag response period (4.25-4.5ms), while the sum is small and flat when no tag response is transmitted (e.g., after 4.6ms). As shown in Figure 6, when multiple tags respond simultaneously using on-off keying, the aggregated signal strength still provides valid indications of tag responses with moving window summation.

Although ZOE can run in realtime on the USRP N210



(a) Estimation accuracy (b) Standard deviation
Fig. 7. Performance of ZOE with different numbers of estimating rounds.

in concert with the WISP RFID tags, we turn to the large-scale simulations to compare ZOE with the existing cardinality estimation schemes. This is for two reasons. First, partially due to the complexity of existing cardinality estimation schemes (e.g., FNEB, LoF, and PET), 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 with the most recent approaches FNEB, LoF, and PET 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 LoF and PET.

$$\text{Accuracy} = E(\hat{n}/n),$$

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

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

where the operator $E[\cdot]$ 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

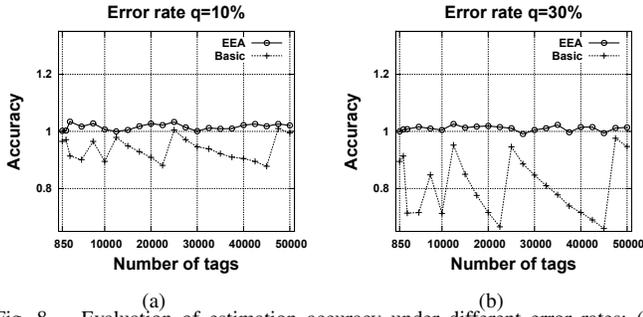


Fig. 8. Evaluation of estimation accuracy under different error rates: (a) $q = 10\%$; (b) $q = 30\%$.

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 transmission rate varies depending on various factors (e.g., PHY/MAC layer implementations and 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.

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. Figure 7(a) depicts different estimation accuracy while different number 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 variation of tag cardinality has little impact on the estimation accuracy.

Figure 7(b) illustrates the standard deviation which 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 varying from 5% to 30%. The estimation round m is fixed at 64 in all experiments. In the cases that $m \neq 64$, the simulation results suggest similar trends.

Figure 8 plots the estimation accuracy with and without Error Estimation and Adjustment (EEA) algorithm, respectively. As shown in Figure 8, 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.

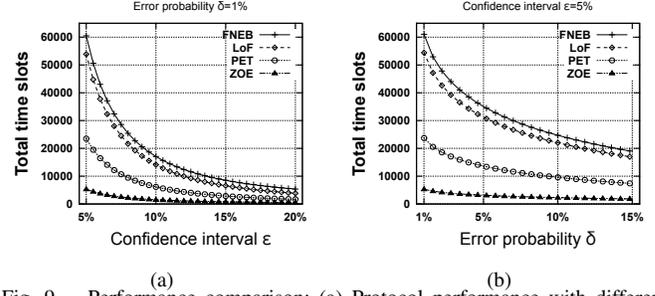


Fig. 9. Performance comparison: (a) Protocol performance with different confidence interval ϵ , and the same error probability $\delta = 1\%$; (b) Protocol performance with different error probability δ and the same confidence interval $\epsilon = 5\%$.

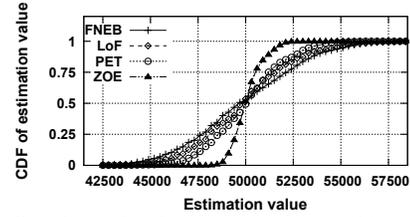


Fig. 10. Cumulative distribution comparison of different protocols.

C. Performance comparison

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

Given the same estimation accuracy requirement of (ϵ, δ) -approximation, we compare the estimating time slots that each estimation protocol needs to achieve the accuracy requirement. The actual tag cardinality is 50000. For the proposed ZOE protocol, the entire estimation process 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 the comparison with other protocols.

The recent protocols shall perform multiple estimation rounds to meet the certain estimation accuracy. We first keep the error probability $\delta = 1\%$ fixed and vary the confidence interval from 5% to 20%. Figure 9(a) plots the total time slots needed by each protocol. Then we keep the confidence interval $\epsilon = 5\%$ fixed and vary the error probability δ from 1% to 15%, and the simulation results are presented Figure 9(b). According to the simulation results, ZOE only consumes about 31% processing time of PET to provide the same estimation accuracy, which translates to more than 3x performance improvement in terms of time efficiency and even more compared with 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.

We provide PET, FNEB and LoF the same amount of time slots to estimate the actual tag cardinality of 50000, and present the distributions in Figure 10. According to the simulation results, we find that the estimation results of ZOE are much more concentrated about the actual cardinality. Besides, 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

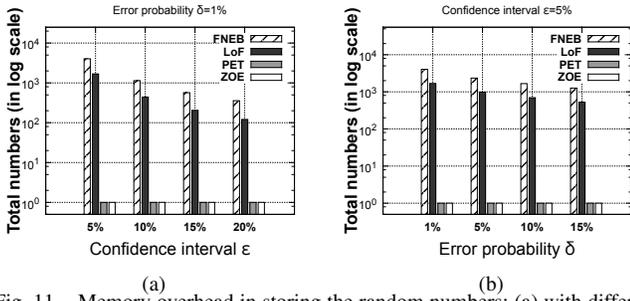


Fig. 11. Memory overhead in storing the random numbers: (a) with different confidence interval ε , and the same error probability $\delta = 1\%$; (b) with different error probability δ , and the same confidence interval $\varepsilon = 5\%$.

interval [47500, 52500] in ZOE, the existing approaches can only guarantee less than 80% results within such an interval.

We compare the computation and memory overhead at RFID tags. We examine the memory overhead and compare ZOE with recent protocols in Figure 11. We fix the error probability $\delta = 1\%$ and vary the confidence interval ε from 5% to 20% in Figure 11(a). We vary the error probability δ from 1% to 15% with fixed confidence interval $\varepsilon = 5\%$ in Figure 11(b). According to the statistics, we observe that ZOE and PET consume constant small storage, and outperforms FNEB and LoF which require much larger memory cost.

Till now we focus on the performance comparison over ideal channels. In Figure 12, we examine the estimation accuracy of ZOE compared with recent approaches with different error rates. We vary the error rate from 5% to 30%, and the actual tag cardinality is 50000. According to Figure 12, we find that the estimation accuracies of LoF and PET are significantly biased from the actual value. Though FNEB is more robust than LoF and PET, it still fails to provide an unbiased and accurate estimation. On the other hand, ZOE with EEA resists the various 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 (ZOE) which improves the estimation time efficiency in meeting arbitrary accuracy requirement. ZOE only requires one-bit response from the RFID tags per estimation round while prior works require several time slots. We also enhance the robustness of cardinality estimation over noisy channels. We implement a prototype system based on the GNURadio/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 experiment results demonstrate that ZOE outperforms the most recent cardinality estimation protocols.

ACKNOWLEDGMENT

We acknowledge the support from NTU Nanyang Assistant Professorship (NAP) grant M4080738.020, Microsoft research grant FY12-RES-THEME-001, and NSFC grant No. 61272456.

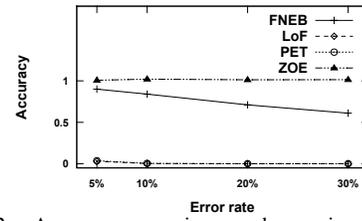


Fig. 12. Accuracy comparisons under varying error rates.

REFERENCES

- [1] "Alien Technology", <http://www.alientechnology.com>
- [2] "EPCglobal C1G2", <http://www.epcglobalinc.org/standards/uhf1g2>
- [3] "Gen 2 RFID Tools", <https://www.cgran.org/wiki/Gen2>
- [4] "Ettus Research", <http://www.ettus.com>
- [5] "WISP Platform", <http://wisp.wikispaces.com>
- [6] M. Buettner and D. Wetherall, "An Empirical Study of UHF RFID Performance", in *ACM MobiCom*, 2008.
- [7] J. I. Capetanakis, "Tree algorithms for packet broadcast channels", *IEEE Trans. on Information Theory*, vol. 25, issue 5, pp. 505-515, 1979.
- [8] S. Chen, M. Zhang, and B. Xiao, "Efficient Information Collection Protocols for Sensor-augmented RFID Networks", in *IEEE INFOCOM*, 2011.
- [9] K. Finkenzeller, "RFID Handbook: Radio-Frequency Identification Fundamentals and Applications", *John Wiley & Sons*, 2000.
- [10] G. R. Grimmett, D. R. Stirzaker, "Probability and Random Processes, 3rd edition", *Oxford University Press*, 2001.
- [11] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID Tags Efficiently and Anonymously", in *IEEE INFOCOM*, 2010.
- [12] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems", in *ACM MobiCom*, 2006.
- [13] M. Kodialam, T. Nandagopal, and W. C. Lau, "Anonymous Tracking using RFID tags", in *IEEE INFOCOM*, 2007.
- [14] T. Li, S. Chen, and Y. Ling, "Identifying the Missing Tags in a Large RFID System", in *ACM MobiHoc*, 2010.
- [15] L. M. Ni, Y. Liu, Y. C. Lau, and A. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID", *ACM Wireless Networks*, vol. 10, issue 6, pp. 701-710, 2004.
- [16] C. Qian, H. Ngan, and Y. Liu, "Cardinality Estimation for Large-scale RFID Systems", in *IEEE PerCom*, 2008.
- [17] C. Qian, Y. Liu, H.-L. Ngan, and L. M. Ni, "ASAP: Scalable Identification and Counting for Contactless RFID Systems", in *IEEE ICDCS*, 2010.
- [18] Y. Qiao, S. Chen, T. Li, and S. Chen, "Energy-efficient Polling Protocols in RFID Systems", in *ACM MobiHoc*, 2011.
- [19] T. F. La Porta, G. Maselli, and C. Petrioli, "Anticollision Protocols for Single-Reader RFID Systems: Temporal Analysis and Optimization", *IEEE Trans. on Mobile Computing*, vol. 10, issue 2, pp. 267-279, 2011.
- [20] L. G. Roberts, "Aloha Packet System with and without Slots and Capture", *ACM SIGCOMM Computer Communication Review*, vol. 5, issue 2, pp. 28-42, 1975.
- [21] J. R. Smith, A. P. Sample, P. S. Powlledge, S. Roy, and A. Mamishev, "A wirelessly-powered platform for sensing and computation", in *ACM Ubicomp*, 2006.
- [22] C. C. Tan, B. Sheng, and Q. Li, "How to Monitor for Missing RFID Tags", in *IEEE ICDCS*, 2008.
- [23] R. Want, "An Introduction to RFID Technology", *IEEE Pervasive Computing*, vol. 5, issue 1, pp. 25-33, 2005.
- [24] 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 *IEEE INFOCOM*, 2011.
- [25] R. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast Identification of the Missing Tags in a Large RFID System", in *IEEE SECON*, 2011.
- [26] Y. Zhang, L. T. Yang, and J. Chen "RFID and Sensor Networks: Architectures, Protocols, Security and Integrations", *Auerbach Publications*, 2010.
- [27] Y. Zheng, M. Li, and C. Qian, "PET: Probabilistic Estimating Tree for Large-Scale RFID Estimation", in *IEEE ICDCS*, 2011.
- [28] Y. Zheng and M. Li, "Fast Tag Searching Protocol for Large-Scale RFID Systems", in *IEEE ICNP*, 2011.
- [29] 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 *ISLPED*, 2004.