# Fast and Reliable Unknown Tag Detection in Large-Scale RFID Systems

Wei Gong*, Jiangchuan Liu*, and Zhe Yang†*
{gongweig, jcliu}@sfu.ca, zyang@nwpu.edu.cn

*School of Computing Science, Simon Fraser University, Canada
†Northwestern Polytechnical University, China

## ABSTRACT

One of the most important applications of Radio Frequency Identification (RFID) technology is to detect unknown tags brought by new tagged items moved in, misplacement, or counterfeit tags. While unknown tag identification is able to pinpoint all the unknown tags, probabilistic unknown tag detection is preferred in large-scale RFID systems that need to be frequently checked up, e.g., real-time inventory monitoring. Nonetheless, we find that the efficiency of most previous works is not well optimized due to the transmission of unhelpful data. In this paper, we propose a fast and reliable method for probabilistic unknown tag detection, White Paper (WP) protocol. The key novelty of WP is to build a composite message data structure that consists of all the informative data from several independent detection synopses, i.e., excluding the useless data from communication. Hence, this design allows us to optimize the detection and communication efficiency at the same time. In particular, the compact detection synopsis is designed and tuned to minimize the failure probability for detection and the detection message is compositely constructed to reduce the transmission overhead, achieving the optimal detection and communication efficiency, respectively. We implement a prototype system using USRP software-defined radio and WISP tags to show the feasibility of this design. We also conduct extensive simulations and comparisons to show that WP achieves more than 2x performance gain compared to the state-of-the-art protocols.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; C.2.1 [**Network Architecture and Design**]: Wireless Communication

## Keywords

RFID, unknown tag detection, energy-efficient

## 1. INTRODUCTION

Over the past decade, Radio Frequency Identification (RFID) technology has witnessed an unprecedented growth in practical applications. It has several distinct advantages. First, RFID tags are so small that they can be embedded in almost everything to give a unique ID. Second, the inexpensiveness of tag makes the large-scale use for almost anything that costs more than $1 possible. Third, tags are able to be read wirelessly, from a few inches to several feet. Fourth, it enables batch operations over thousands of tags at a time, while other methods, e.g., barcode, can only deal with objects sequentially.

This paper focuses on the problem of detecting unknown tags in large-scale RFID systems. Accurate and fast unknown tag detection is very important to many applications [1]. For example, in RFID-enabled inventory control, it needs to detect unknown-tag events due to new commodities moved in or item misplacement [2]. When processing a large number of tagged items at a mail service center, unknown tag detection can help efficiently verify a batch of tags [3]. Moreover, unknown tag detection is needed as a filter module in unknown tag identification [4] and missing tag detection [5].

While unknown tag identification can exactly pinpoint all the unknown tags in a batch, it is not always necessary to collect information for all of them. Instead, knowing whether there is any unknown tag with desired accuracy and probability will be adequate in many applications, where it is almost impossible to achieve acceptable identification efficiency due to the overwhelmingly large volume of objects, e.g., RFID-enabled cross-border cargo inspection [6]. A recent study regarding harbor cargo also states that if the sampling ratio of containers is up to 10%, the whole harbor would be paralyzed [7]. Thus, improving the efficiency of unknown tag detection will significantly benefit a lot of large-scale RFID-enabled systems, especially for those with stringent time requirements.

Towards this end, several probabilistic unknown tag detection schemes have been proposed to find unknown tags in a batch. SEBA [3] proposes to use single-echoes to fast pinpoint unexpected responses from unknown tags. Bianchi et al. [8] further improve this by introducing a bloom filter like data structure. A recent scheme [1] proposes new filtering techniques based on the bloom filter. By carefully reviewing those methods, we observe that the efficiency of existing methods is not very well optimized due to the transmission of unhelpful data for detection. For instance, collision slots that contribute nothing in the detection of unknown tags

*MobiHoc'16, July 04-08, 2016, Paderborn, Germany*
© 2016 ACM. ISBN 978-1-4503-4184-4/16/07...$15.00
DOI: http://dx.doi.org/10.1145/2942358.2942368

are still included in communication messages [1, 3, 8], losing great opportunities of transmission optimization.

In this paper, we propose a fast and reliable protocol for probabilistic unknown tag detection, White Paper (WP) protocol, *where the communication message is composed of (almost) all zero slots*. In WP, we design a new compact detection synopsis and tune its parameter for optimal detection efficiency, i.e., we minimize the failure probability of detection for a given frame length. Meanwhile, based on all the informative slots of detection synopses we construct a novel composite message data structure to significantly reduce transmission overhead. Hence, we are able to optimize the detection efficiency and communication efficiency independently, resulting in high detection efficiency with minimal transmission overhead. Various fundamental energy-time tradeoffs in probabilistic unknown tag detection are also presented in our analytical framework.

We demonstrate the effectiveness of the proposed protocol through a prototype system using USRP software defined radio [9] and WISP tags [10]. Comparisons are done with extensive simulations to examine the performance in large-scale settings. Our results demonstrate that we achieve more than 2x performance gain compared to the state-of-the-art protocols.

# 2. PRELIMINARIES

## 2.1 Problem Formulation

Suppose we have a known tag set $\mathcal{S} = \{x_1, x_2, x_3, ...\}$, and a to-be-tested tag set $\mathcal{T} = \{y_1, y_2, y_3, ...\}$. The carnalities of $\mathcal{S}$ and $\mathcal{T}$ are $N$ and $n$, respectively. While $N$ is a priori, $n$ is not. The goal of probabilistic unknown tag detection is to find whether there is any unknown tag in $\mathcal{T}$ with the knowledge of $\mathcal{S}$. In practice, the two basic requirements for probabilistic unknown tag detection are i) if all the tags in $\mathcal{T}$ are known, the detection result should be negative for sure, ii) if there is at least one unknown tag in $\mathcal{T}$, the detection result should be declared as positive with high probability, i.e., a little detection failure is allowed. Towards this end, we define two parameters: $\varepsilon$, the detection failure probability, and $m$, the tolerable maximum number of unknown tags in a batch. Thus an $(\varepsilon, m)$ detection scheme should be able to detect an unknown-tag event with probability at least $1 - \varepsilon$ if the number of unknown tags in $\mathcal{T}$ is greater than or equal to $m$. Intuitively, we want $\varepsilon$ to get closer to 0 and $m$ to get closer to 1. Although we do not assume any relationship between $N$ and $n$, it is worth noting that the unknown tag detection problem becomes even more challenging when $N \gg n$.

## 2.2 System Model

A typical RFID system consists of three parts: tags, a reader[1], and a back-end server. Tags may either be read-only, having assigned unique identification information, or may be read/write, where additional data can be stored into the memory on board by the user. The back-end server usually stores all the tags' information and performs various management operations. Generally, we assume that the reader is securely connected to the back-end server through a high-speed channel. Therefore, we denote the reader and back-end server by the reader for simplicity, if not specified.

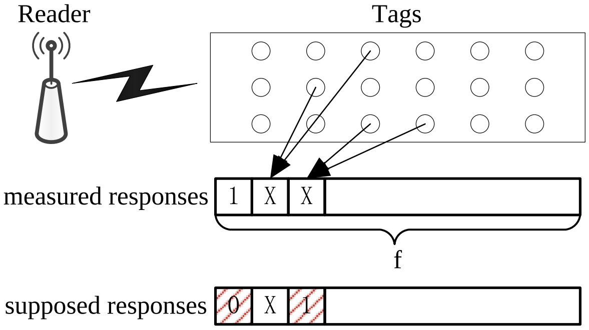


**Figure 1: The ALOHA model and basic detection scheme.**

We assume that the communication between the reader and tags follows the ALOHA model, which is widely used in EPC Global C1G2 standard [11] and many other RFID protocols [3, 12, 13, 14]. As shown in Figure 1, the reader first broadcasts to tags a probing message, which contains the frame size $f$ and the random seed value $r$. When each tag has received this probing message, it uses preloaded hash functions $H$ to compute its own reply slot number as $sn = H(f, r, ID)$, where ID is the unique identification information. Afterward, the reader issues a slot-start command to all the tags. Then each tag checks whether its supposed reply slot number is equal to the current slot number. If so, it responds instantly. Otherwise, the current slot number increases by 1. According to the number of responses in a single slot, we classify slots into three types: a *zero slot* means no response is in that slot; a *singleton slot* denotes that only one tag's reply is in the slot; a *collision slot* means there are at least two tags' responses in the slot. We also use *non-zero slot* to denote both singleton slot and collision slot. For unknown tag detection, when the ALOHA frame completes, the reader is able to compose the measured responses of size $f$. Meanwhile, since the back-end server contains all the information of tags (hash functions and unique ID), the reader can virtually construct the supposed responses as if all the tags in $\mathcal{S}$ are present. Therefore, the server can perform the detection by comparing the measured responses to the supposed responses slot by slot. There are two conditions where the server declares there indeed exist unknown tag(s): i) a supposed zero slot turns out to be a singleton or collision slot, e.g., the first position of the supposed responses in Figure 1; ii) a supposed singleton slot turns out to be a collision slot, e.g., the third position in the supposed responses. In summary, *only unknown tags would cause this "add-up" effect on responses.* We use responses and synopsis/synopses interchangeably in this paper, because the use of "responses" is from the perspective of communication and the use of "synopsis/synopses" is from the perspective of data structure.

According to the parameters of Philips I-Code [15], if we need to distinguish a zero slot from a non-zero slot, the tag only needs to transmit a short response that costs 0.4 ms, denoted as $T_s$; if we want to distinguish a zero slot from a singleton slot and a collision slot, a long response that is 0.8 ms is required, denoted as $T_l$. Moreover, if a slot is used to transmit the ID (typically 96 bits) of the tag, it costs 2.4 ms, denoted as $T_{tag}$. We prefer to use short responses than long

[1] Multiple-reader cases are discussed later.

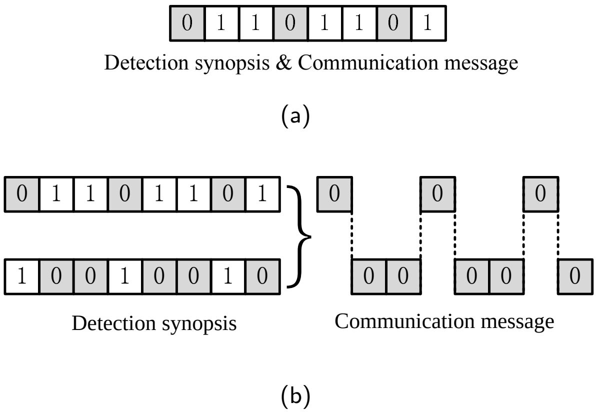


**Figure 2: (a) The detection synopsis and communication message are the same; (b) The detection synopsis and communication message are separated (0 denotes zero slot and 1 denotes non-zero slot).**

responses in terms of time efficiency. In other words, we only distinguish zero slots from non-zero slots in our scheme. We also employ the participation probability for each tag in a frame, denoted as $p$. For example, if $p = 0.25$, it means this tag would engage in this frame with 25% probability.

# 3. FAST PROBABILISTIC UNKNOWN TAG DETECTION

In this section, we first present our motivation and then describe our basic idea for fast unknown tag detection. Later we consolidate it with detailed communication protocols for both the reader and tag sides. Corresponding analysis and practical design issues are discussed as well.

## 3.1 Motivation

Here, we examine previous work by using SEBA (SEBA-2) as a case study [3], in which it only distinguishes zero slots from non-zero slots. Although the following analysis is based on SEBA, it can also apply to most existing schemes. As aforementioned, we know that zero slots are important in unknown tag detection. If an unknown tag responds in a supposed zero slot, it would cause the actual slot to be a non-zero slot, indicating that an unknown-tag event is detected. Thus, we argue that the transmission of non-zero slots in supposed responses is a waste of time because those slots contribute nothing for detection. Actually the amount of this waste is significant: about 50% of the total communication time is wasted in SEBA as later elaborated in section 3.3. As shown in Figure 2a, the grey-colored zero slots stand for useful transmission.

Therefore, it motivates us to treat communication messages and detection synopses differently by transmitting useful slots (zero slots) as many as possible, greatly improving detection and communication efficiency. Let's see an example. Suppose that the detection success probability of a SEBA synopsis is 80% and we are going to detect a single unknown tag. As shown in Figure 2b, if we have 2 independent SEBA synopses in advance, then the detection success probability of using 2 synopses together is $1-(1-80\%)^2 = 96\%$. If we have even more independent synopses, e.g., $l$, it is easy to see that $1-(1-80\%)^k$ could fast approach to 100% as $l$ increases. Later we will show how to achieve efficient communication by encoding several independent synopses into a composite message. We also consider multiple responses from each tag in a single frame, which can largely improve the detection efficiency of a synopsis.

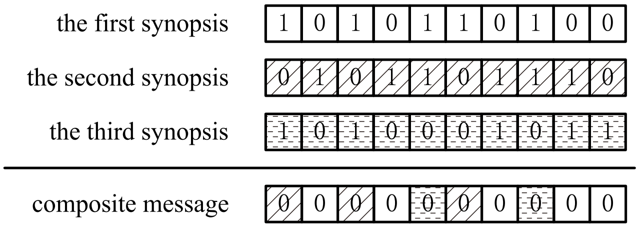


**Figure 3: The construction of a sample composite message.**

## 3.2 Basic Ideas

The basic idea of our scheme is to build a composite message data structure that consists of all the informative data from several independent detection synopses, excluding the useless data from communication.

Based on the observation that non-zero slots contributes little for detection, our scheme aims to improve communication efficiency by changing non-zero slots into zero slots. Let us see a toy example in Figure 3, where 0 denotes a zero slot and 1 denotes a non-zero slot. Assume that we have three detection synopses. First, we construct the composite message based on the first synopsis, in which all the zero slots are kept in the composite message. Then by scanning the second synopsis, the composite message continues to 'absorb' useful (zero) slots. The rule of absorption is that the slot of current synopsis is zero and the corresponding slots in all the previous synopses are non-zero. For the third synopsis, we also apply this rule for combination. Finally, we get a composite message full of zero slots, nowhere for unknown tags to hide: *it is easy to spot stains (unexpected responses caused by unknown tags) on a white paper (a composite message).*

## 3.3 White Paper Protocol

In this part, we turn our basic idea into detailed protocols for both the reader and tag. We list the main notations in Table 1.

As we know that, the construction of a composite message relies on several virtual synopses. Hence, each slot in a composite message is actually an *index* of synopses, indicating which synopsis this slot comes from. We use an index vector to denote a sequence of such indexes.

The protocol consists of four major steps: index vector generation, index vector transmission, response measurement, and unknown tag detection, as shown in Algorithm 1 and 2.

**Phase one - index vector generation:** As shown in Figure 1, the supposed responses can be virtually generated since the reader knows all the information of tags in $\mathcal{S}$. The key difference is that, in WP each tag replies at $k$ different slots based on $k$ independent hash functions in a frame. The rule of constructing the composite message is first come first serve, i.e., it sequentially selects zero slots from supposed

**Table 1: Main Notations**

| Symbols | Descriptions |
|---|---|
| $N$ | number of tags in $\mathcal{S}$ |
| $n$ | number of tags in $\mathcal{T}$ |
| $\varepsilon$ | required detection failure probability |
| $m$ | tolerable maximum # of unknown tags |
| $f$ | frame size |
| $H$ | s hash function stored in tags |
| $l$ | number of virtual supposed synopses |
| $k$ | number of responses for each tag |
| $p$ | participation probability for each tag |
| $p_0$ | probability of being a zero-slot |
| $p_h$ | hidden probability for an unknown tag |
| $ci$ | random seed index |
| $csn$ | current slot number |
| $SR$ | supposed responses |
| $MR$ | measured responses |

**Algorithm 1** The WP protocol for tags

```
Receive the frame start command and the index vector $IV$.
Receive the frame size $f$, the participation probability $p$, and the random seeds $s_0, s_1, ..., s_{l-1}$.
Choose to participate in this frame or sleep based on the probability $p$.
If not participate, sleep until another frame starts.
Compute reply slot numbers $sn[i][j] = H(f, ID, s_i, k)$ where $(0 \leq i \leq l-1, 0 \leq j \leq k-1)$
Initialize the current slot number $csn \leftarrow 0$ and current random seed index $ci \leftarrow 0$.
while TRUE do
    wait-for-slot-start().
    $ci \leftarrow IV[csn]$.
    for $i = 0$ to $k-1$ do
        if $csn == sn[ci][i]$ then
            Respond instantly and break.
        end if
    end for
    $csn \leftarrow csn + 1$.
end while
```

synopses one by one. Note that as we change non-zero slots into zero slots in our best efforts, if there are still some non-zero slots after combing all the synopses, we just keep such remaining non-zero slots from the first synopsis.

**Phase two - index vector transmission:** As there are $l$ supposed responses, the size of each element in an index vector is $\lceil\log(l+1)\rceil$-bit. For instance, if $l = 3$, then each element is 2-bit long. The whole index vector may not be able to fit into a single transmission if the size of synopses is large. Hence, we can divide it into pieces and each piece includes $\lfloor\frac{96}{\lceil\log l+1\rceil}\rfloor$ indexes. Using this division, the reader starts the frame and transmits all the pieces of an Index Vector ($IV$) using $T_{tag}$ slots. At the same time, when the tag receives the frame start command, it will expect an $IV$ piece by piece.

**Phase three - response measurement:** The reader continues to broadcast several parameters to tags, including $f$, the frame size, $p$, the participation probability, and $s_0, s_1, ..., s_{l-1}$, the random seeds. Upon receiving those parameters, the tag first decides whether to participate in this frame according to $p$. If it does not participate in this frame, it will sleep until another frame starts. If it chooses to join in, it needs to compute reply slot numbers using the hash function $H$. The tag generates $k$ supposed reply slot numbers based on the different random seeds. In each time slot, the reader issues a slot start command and waits for responses. At the tag side, if any one of supposed reply slot numbers for random seed indexes is equal to the current slot number, the tag responds instantly. Otherwise, it keeps silent. When $f$ time slots are finished, the reader obtains the Measured Responses ($MR$).

**Phase four - unknown detection:** The detection process for the reader is relatively easy. First, the reader compares the Measured Responses ($MR$) to the composite Supposed Responses ($SR$) slot by slot. If any one slot in the measured responses is non-zero and its corresponding slot in the supposed responses is zero, the reader shall report a positive result, indicating there exist unknown tags in the batch. Otherwise, all the tags in the batch ($\mathcal{T}$) are deemed known since the result is negative. Note that this result may contain false negatives (detection failure), but no false positives.

## 3.4 Protocol Analysis

Now, we seek to optimize detection and communication efficiency separately.

**Detection efficiency optimization:** From the supposed synopsis generation process that is in the phase one of WP, we know that each tag selects $k$ slots in a frame. Besides, each tag chooses to participate in the frame based on the probability $p$. Hence, the probability $p_0$ that one slot in a supposed synopsis is still zero after $N$ tags' responses is

$$p_0 = (1 - p\frac{1}{f})^{kN} \approx e^{-\frac{pkN}{f}}. \tag{1}$$

Meanwhile, for an unknown tag, if all the $k$ slots it chooses are non-zero, it would be hidden in this synopsis. We can calculate this hidden probability, $p_h$, as

$$p_h = 1 - p + p(1-p_0)^k \approx 1 - p + p(1 - e^{-\frac{pkN}{f}})^k. \tag{2}$$

In order to maximize the detection efficiency of a synopsis, we needs to minimize the above hidden probability with respect to $k$ given the fixed frame length $f$. To do so, we first rewrite $(1 - e^{-\frac{pkN}{f}})^k = e^{k\ln(1-e^{-\frac{pkN}{f}})} = e^q$ where $q = k\ln(1 - e^{-\frac{pkN}{f}})$. It is easy to see that minimizing $p_h$ is equal to minimizing $q$, thus we can obtain its partial derivative as

$$\frac{dq}{dk} = \ln(1 - e^{-\frac{pkN}{f}}) + \frac{kNe^{-\frac{pkN}{f}}}{f(1 - e^{-\frac{pkN}{f}})}. \tag{3}$$

If let this derivative to be 0, we get when $k = \frac{f}{pN}\ln 2$, $p_h$ achieves its global minimum $1 - p + p(\frac{1}{2})^k$ [2].

[2] Using the second derivative test, we know it is a minimum instead of a maximum, since its second derivative value at

**Algorithm 2** The WP protocol for the reader

```
//Phase one - index vector construction.
Generate $l$ random seeds $s_0, s_1, ..., s_{l-1}$ and corresponding supposed responses $SR_0, SR_1, ..., SR_{l-1}$.
Initialize the index vector $IV[i] \leftarrow 0 (0 \le i \le f-1)$.
Initialize the combined supposed responses $SR \leftarrow SR_0$.
for $i = 1$ to $l-1$ do
  for $j = 0$ to $f-1$ do
    if $IV[j] == 0$ and $SR_i[j] == 0$ then
      $IV[j] \leftarrow i$, $SR[j] \leftarrow 0$.
    end if
  end for
end for
//Phase two - index vector transmission.
Divide the $IV$ into pieces and each piece contains $\lfloor \frac{96}{\lceil \log l+1 \rceil} \rfloor$ indexes.
Issue a frame start command and transmit the $IV$ piece by piece.
//Phase three - response measurement
Broadcast the frame size $f$, the participation probability $p$, and the random seeds $s_0, s_1, ..., s_{l-1}$.
Initialize the measured responses $MR[i] \leftarrow 0 (0 \le i \le f-1)$.
for $i = 0$ to $f-1$ do
  Issue slot-start command.
  wait-for-tags-response().
  if there is any response in this slot then
    $MR[i] \leftarrow 1$.
  end if
end for
//Phase four - unknown detection
for $i = 0$ to $f-1$ do
  if $MR[j] == 1$ and $SR[j] == 0$ then
    Report a positive result and return.
  end if
end for
Report a negative result and return.
```

Let us see an illustrative example showing the impact of different $k$ on the hidden probability (detection failure probability). In Figure 4, our settings are: the size of $\mathcal{T}$ is 1000; the tolerable number of unknown tags is only 1; and the participation probability is 1. By varying the size of frame, we compare different $k(1, ..., 5)$ to the theoretical optimal $k(\frac{f}{N}\ln 2)$. We observe that SEBA ($k = 1$) achieves optimal when hidden probability is above 0.382, which is quite inefficient for detection. In particular, when frame size is 5000, $k = 3$ achieves optimal hidden probability 0.09, whereas its of SEBA is only 0.18. Note that for SEBA, when $p_h$ achieves its minimum, $p_0 \approx e^{-\frac{kN}{f}} = \frac{1}{2}$. It means that about 50% of the total slots in a frame are non-zero slots, which are of no use in unknown tag detection, leading to unnecessary and wasteful transmission. This further makes necessary the idea of constructing composite messages to improve communication efficiency.

**Communication efficiency optimization:** As aforementioned, we know that the composite message is constructed using $l$ supposed synopses. In order to achieve the user-specified requirements for $(\varepsilon, m)$, we should obtain the

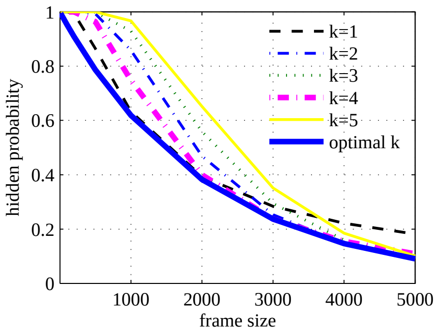


**Figure 4: Different frame size vs hidden probability when $N = 1000, m = 1, p = 1$.**

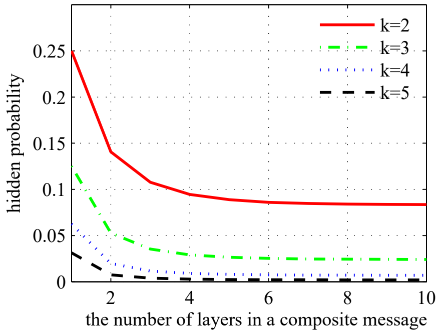


**Figure 5: Different number of layers in a composite message vs hidden probability when $m = 1, p = 1$, and $\frac{f}{N} = \frac{k}{\ln 2}$.**

hidden probability of $m$ unknown tags in a composite message, denoted by $\alpha_h$. To do so, we first calculate the hidden probability of a single unknown tag in a composite message, denoted as $\beta_h$. It is obvious that $\alpha_h = \beta_h^m$. Virtually we can divide a final composite message into $l$ layers, each of which only contains the zero slots from $i$-th $(0 \le i \le l-1)$ synopsis. Let $w_i$ be the probability of an original slot in the $i$-th synopsis to be chosen into the $i$-th layer of the composite message, and $\gamma_i$ be the hidden probability of the $i$-th layer of the composite message. Iteratively, according to the criteria that a slot is chosen into the composite message only if it is zero in $i$-th layer and all of corresponding positions in former layers are non-zero slots, we can have

$$w_i = (1-p_0)^i p_0, \gamma_i = 1-p+p(1-w_i)^k, (0 \le i \le l-1). \quad (4)$$

After $l$ iterations, we have

$$\beta_h = \prod_{i=0}^{l-1} \gamma_i. \quad (5)$$

It is easy check that when $i = 0$, the results $w_0 = p_0$ and $\beta_h = \gamma_0$, which are consistent with the former analysis. Therefore, in order to fulfill the requirements of $(\varepsilon, m)$, the following equation should be satisfied

$$\varepsilon \ge \alpha_h = \beta_h^m = (\prod_{i=0}^{l-1}(1-p+p(1-w_i)^k))^m. \quad (6)$$

We observe that it follows the law of diminishing marginal returns regarding the number of layers $(l)$ in the composite

point $k = \frac{f}{pN}\ln 2$ is greater than 0.

message. As shown in Figure 5, we set $m = 1, p = 1$ and $\frac{f}{N} = \frac{k}{\ln 2}$. By varying $l$, we observe similar trends for $k \in [2, 5]$. Note that the case where $l$ is above 5 is not shown, because the return of increasing $l$ is less than 0.001, which is negligible. Therefore in the following we use $l = 5$ as default unless otherwise specified[3].

## 3.5 WP with Multiple Readers

To work with multiple readers, we adapt the approach proposed in [12][13] which control all the readers through a central server. Therefore, before the phase one the central server shall disseminate parameters (e.g., seed value $s_0$,$s_1$,...) with the same value across all the readers. Then each reader should construct supposed responses based on the seed values from the control sever, rather than on its own. That is to say, all the readers share the same seed values in corresponding frames. Due to this design, before the phase four the central server needs to combine measured responses across the readers by logically OR-ing them slot by slot.

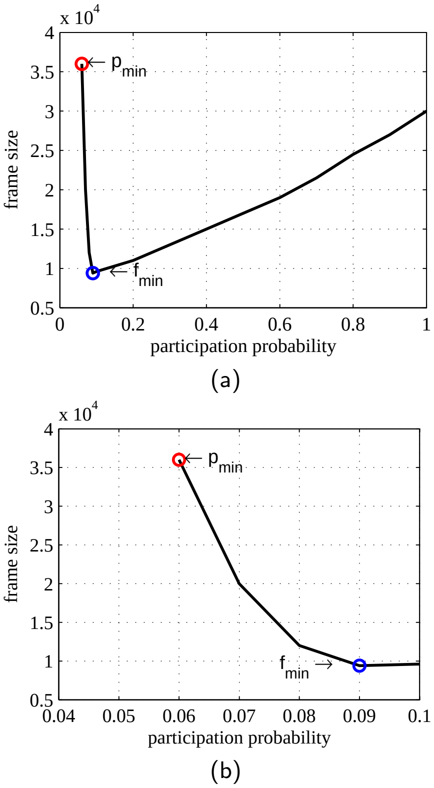
Another important aspect about multiple readers is additional reader-reader collisions. Fortunately, WP is compatible with state-of-the-art reader scheduling algorithms, which takes care of these collisions nicely, e.g., [16].

## 3.6 Hardware Requirements

The WP protocol requires programmability for both the reader and tags. The programmability of the reader is easy to achieve since both the software radio defined reader [9] and commercial readers, e.g., ThingMagic M6e 4-port UHF RFID reader [17], are able to support user-defined commands. While being unable to be supported by off-the-shelf C1G2 tags, WP protocol can be implemented on programmable tags, such as WISP [10] or OpenBeacon [18]. Besides the available commands defined by C1G2 (e.g., **QUERY**, **READ**), we need to implement two user-defined commands. The first command is to transmit the index vector to tags, and the second is to measure the response in the slot after probing. The details are given in section 5.

# 4. ENERGY AND TIME TRADEOFFS

The energy cost of tags is another important issue we should carefully cope with. For example, in a large RFID-enabled warehouse, active tags are usually used to label commodities. Since active tags are battery-powered, recharging batteries for thousands of tags is really a heavy work process, and even in some cases the tags are not easily accessible, e.g., tagged commodities may be intensively piled. Here, we mainly focus on the energy consumption caused by wireless transmission, we use the participation probability of tags in a frame, $p$, to depict the energy consumption of the tags in the detection. The smaller $p$ is, the fewer tags to transmit responses and thus the less energy consumed. As most of the time is spent on the frame, we use the frame size $f$ to represent the time cost of WP. Therefore, we strive to achieve energy-time tradeoffs in probabilistic unknown tag detection. One typical problem is how to minimize the communication time under predefined energy-constraints. The other problem is how to minimize the energy consumption in a limited period of time. In both cases, predefined $\varepsilon$ and $m$ requirements should be satisfied at the same time.



(a)

(b)

**Figure 6: (a) Participation probability vs frame size when $N = 100,000, k = 1, m = 50$, and $\epsilon = 0.05$. (b) Zoom view of (a) for p$\in [p_{min}, \mathcal{F}(f_{min})]$**

Intuitively, one may want both $f$ and $p$ to be as small as possible. However, their choices must satisfy equation 6, which means we cannot minimize both of them at the same time, providing opportunities to make energy-time tradeoffs. Without loss of generality, we can define two functions

$$\mathcal{F}(f) = p, \quad \mathcal{G}(p) = \mathcal{F}^{-1}(p) = f. \tag{7}$$

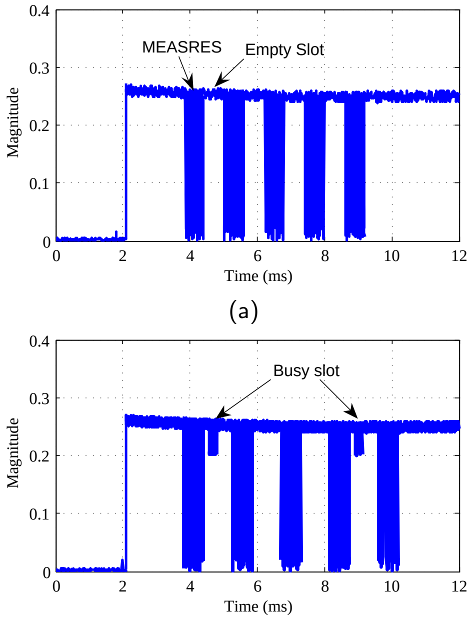
That is to say, given system parameters $(N, k, l)$ and user-specified parameters $(\varepsilon, m)$, $\mathcal{G}(p) = f$ can find the minimum $f$ that satisfies $\varepsilon \geq \alpha_h$. $\mathcal{F}$ is the inverse function of $\mathcal{G}$.

If we set $N = 100,000, k = 1, m = 50$, and $\epsilon = 0.05$, we can plot the curve of $\mathcal{G}$ with varying $p$, as in Figure 6. This energy-time curve measures energy cost as $n * p$ tags participating in the frame. $\mathcal{G}(p)$ denotes the corresponding optimal frame size. The two distinct points of $p_{min}$ and $f_{min}$ need some explanations.

**Finding the minimum $p$.** It is obvious that the participation probability $p$ cannot be arbitrarily small, since $\varepsilon \geq \alpha_h$ may not hold when $p$ it too small. Therefore, there is a minimum participation probability $p_{min}$ that satisfies the user-specified $\varepsilon$. Using $\mathcal{G}$, it is easy to obtain the $p_{min}$ through a bisection search. According to the settings in Figure 6, $p_{min}$ is found to be 0.06.

**Finding the minimum $f$.** Similarly, it is easy to derive that between $p_{min}$ to 1, there must be a minimum frame size $f_{min}$ that makes $\varepsilon \geq \alpha_h$ hold. By a bisection search, we can find $f_{min}$ as 9460 under the settings in Figure 6. Note

[3] For system that may require an extremely low error on the detection failure probability, e.g., $10^{-5}$, a larger $l$ should be employed.



(a)

(b)

**Figure 7: (a) The communication between the reader and 3 known WISP tags. Each slot after the MEASRES command is empty. (b) The communication between the reader and 5 WISP tags (3 known tags and 2 unknown tags). Busy slots are found after the first and the fourth command, respectively, which means unknown tags are detected.**

that for different parameter settings, the curve of $\mathcal{G}$ may be different but the process of finding $p_{min}$ and $f_{min}$ are the same.

**Constrained optimization problems.** There are two closely related constrained optimization problems: energy-constrained least time problem and time-constrained least energy problem. The energy-constrained least time problem always takes the maximum number of tags to participate in the frame, $n_u$, as input. Therefore, we just set the $\frac{n_u}{N}$ as the maximum participation probability, then a bisection search in the range $[p_{min}, \max(\frac{n_u}{N}, \mathcal{F}(f_{min}))]$ would give the optimal. On the other hand, the time-constrained least energy problem often takes the upper bound of frame size $f_u$ as a constraint. After carefully reviewing the energy-time curve in Figure 6, we observe that all the solutions should be in the range $[p_{min}, \mathcal{F}(f_{min})]$. Since if we choose $p > \mathcal{F}(f_{min})$, both the energy cost and time cost are increased. Thus, we set $f = f_u$ and do a bisection search in the range $[p_{min}, \mathcal{F}(f_{min})]$ to find the optimal $p$.

## 5. IMPLEMENTATION

Our prototype of WP is based on USRP software defined radio and programmable WISP tags.

**Setup:** We implement the Software-Defined RFID reader (SDReader) using a USRP N210 and the Gen 2 RFID Tools [9]. This SDReader works in 900MHz band based on an RFX900 daughterboard which is connected to Alien circular polarized antennas. Then we connect the SDReader to a laptop via the built-in Ethernet port on the USRP N210. The operating system is Ubuntu 14.04.2 LTS (32-bit).

**Table 2: The relative energy and time cost of WP($k = 7, l = 5$) with $\mathcal{F}^{-1}(f_{min})$ and $p_{min}$ to SEBA, when $\varepsilon = 0.01$ and $N = 100,000$.**

| | $\mathcal{F}^{-1}(f_{min})$ | | $p_{min}$ | |
|---|---|---|---|---|
| | relative energy cost | relative time cost | relative energy cost | relative time cost |
| m=10 | 10.2% | 9.8% | 9.6% | 98.4% |
| m=50 | 6.3% | 4.3% | 5.9% | 146.7% |
| m=100 | 4.2% | 2.1% | 3.8% | 342.9% |

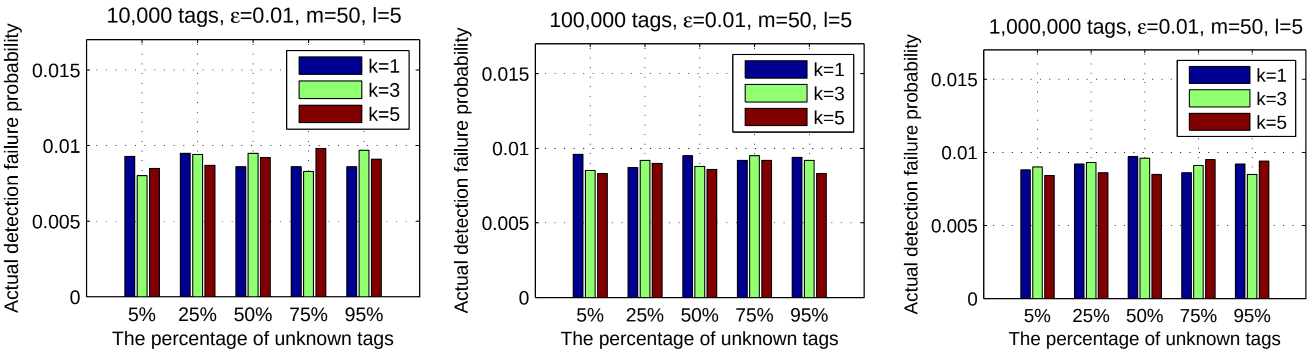
The tag implementation is based on WISP 4.1 hardware (4.1DL) [10]. WISP 4.1 is equipped with an ultra-low power MSP430 micro-controller which is able to do basic computations. Since the C1G2 protocol is partially built in WISP 4.1 firmware, we just need to extend it with the functions of WP.

**Protocol implementation:** In line with the reader-initiated approach of C1G2, we add two commands into the command set: **TRANSIV** that is used to transmit an index vector, and **MEASRES** that can start the slot and measure the responses from tags. Since the major procedures are already described in section 3, we just detail the core part in the phase three here. To measure the responses from tags, the reader sends out a **MEASRES** command along with other parameters, e.g., the participation probability $p$ and random seeds. When the WISP tag has received the **MEASRES** command, it starts computing the reply slot numbers $sn[i][j] = H(f, ID, s_i, k)$. If any of $sn[i][j]$ is equal to the current slot number $csn$, the tag responds instantly. Otherwise keep silent. To respond, the WISP tag just simply transmits a single tone at 250kHz, which has proven enough for robust detection [19].

**Detecting unknown tags:** We prototype an unknown tag detection system which includes 5 WISP tags and 1 SDReader. Among 5 WISP tags, three of them are known and the other two are unknown. The communication is shown in Figure 7a where all 3 known tags are present. All the slots after **MEASRES** command are empty since there is no unknown tag. Then we put 2 unknown tags into the field. The responses measured are shown in Figure 7b. We find two short responses after the first and the fourth **MEASRES** command, indicating an unknown-tag event being detected. Although our WP prototype works well in real-time, we turn to large-scale simulations for more detailed examinations and comparisons with state-of-the-art methods. There are two reasons for this. First, the large-scale field experiment is still hard for the USRP and WISP platform in terms of programming, debugging, and testing [19]. For example, the operating range of the SDReader is quite limited since the power output is only 200mW due to the limitation of RFX900 daughterboards, far less than commercial readers. Second, we would like to compare with prior schemes in various settings, e.g., the size of frames and unknown ratios.

## 6. EVALUATION

We perform extensive simulations to evaluate the perfor-



(a) (b) (c)

**Figure 8: When $\varepsilon = 0.01, m = 50, l = 5$, the percentage of unknowns vs actual detection failure probability. (a) 10,000 tags; (b) 100,000 tags; (c) 1,000,000 tags.**

mance of WP and compare it with the state-of-the-art unknown detection schemes: SEBA [3], SEBA+ [8], and SBF [1].

## 6.1 Simulation Setup

Our simulation parameters are set according to the Philips I-Code system [15], in which $T_s = 0.4$ ms and $T_{tag} = 2.4$ ms, including the waiting time. The $T_s$ slot is used to transmit tag responses. The broadcast data, including the random seeds, the frame size, the participation probability, and the index vector, are transmitted using multiple $T_{tag}$ slots. The time cost of both downlink and uplink is measured as

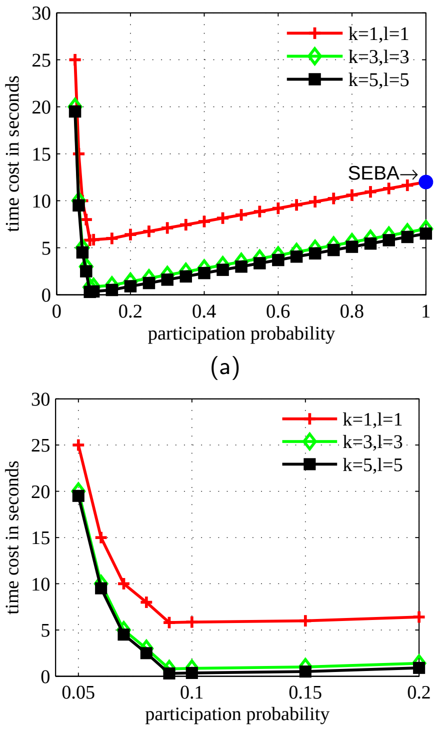
$$timecost = |\frac{sizeof(broadcastdata)}{96}|T_{tag} + fT_s.$$

The energy cost is depicted by the participation probability $p$ and the number of tags participated in the frame together, which is $np$.

## 6.2 WP Investigation

**Detection failure probability:** First, we examine the actual detection failure probability of WP, which is an important metric in our scheme. We fix $p = 1, \epsilon = 0.01, m = 50$, and $l = 5$. As shown in Figure 8a, by varying the percentage of unknown tags from 5% to 95%, the actual detection failure probabilities are always below the predefined $\epsilon = 0.01$ for different $k$. This result shows that WP can effectively detect the unknown-tag event with the desired requirements. Similar results can be found in both Figure 8b and 8c. Those two subfigures further suggest that our WP is able to detect unknown tags in different sizes from 10,000 to 1,000,000.

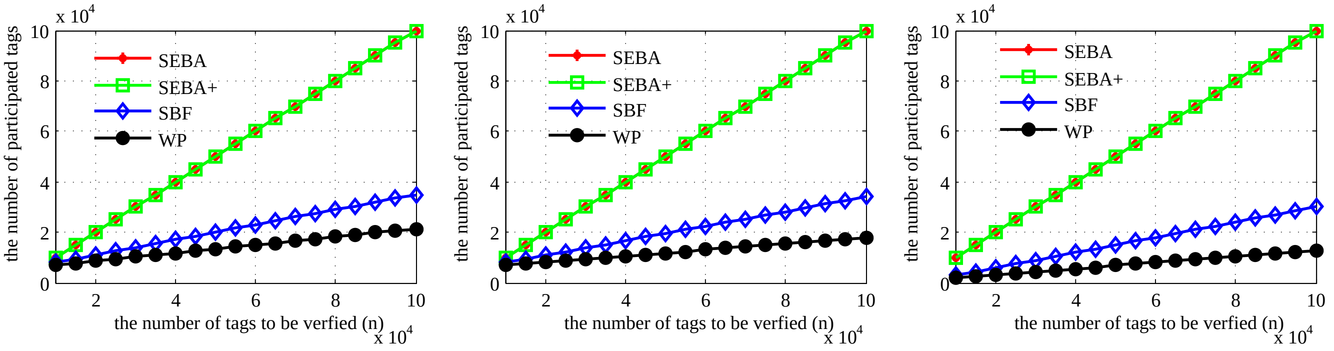
**Energy-time tradeoffs:** Then, we evaluate the energy-time tradeoff of WP with SEBA. We set $N = 100,000, m = 75$, and $\varepsilon = 0.1$. As shown in Figure 9, SEBA is at the rightmost point of curve, which is $k = 1, l = 1$, since its participation probability is 1. We make two key observations here. First, WP significantly outperforms SEBA under various parameters, although there are some additional time for transmitting more random seeds and the index vector. Besides, WP can adjust the energy cost by tuning $p$, which is not considered in SEBA. Second, there are diminishing marginal returns of increasing $k$ and $l$, since the difference between $k = 3, l = 3$ to $k = 5, l = 5$ is much less than its between $k = 3, l = 3$ to $k = 1, l = 1$. Note that these results are consistent with the former analysis, and it is not a meaningless reproduction, since the simulations embody much details of practical RFID systems, which is not covered in the theoretical analysis. The quantified results are given in Table 2, when $k = 7, l = 5, \varepsilon = 0.01, N = 100,000$. We show the results at two critical points $f_{min}$ and $p_{min}$. All those data shows that WP is indeed an efficient probabilistic unknown tag detection protocol in terms of energy cost and time cost, compared to SEBA. In particular, when $m = 100$, the energy cost of WP is only 3.8% of SEBA at



(a)

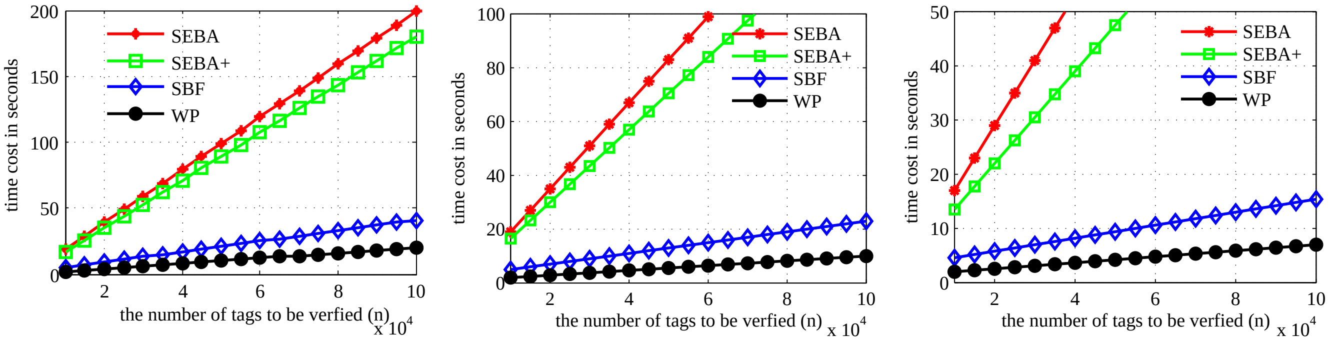
(b)

**Figure 9: (a) Participation probability vs frame size when $N = 100,000, m = 75$ and $\epsilon = 0.1$. (b) Zoom view of (a) for p$\in [0, 0.2]$**



(a) (b) (c)

**Figure 10: When $N = 100,000, \varepsilon = 0.01$, the number of tags to be verified VS the number of participated tags. (a) $m = 10$; (b) $m = 20$; (c) $m = 50$.**



(a) (b) (c)

**Figure 11: When $N = 100,000, \varepsilon = 0.01$, the number of tags to be verified VS time cost. (a) $m = 10$; (b) $m = 20$; (c) $m = 50$.**

$p_{min}$ that achieves least energy cost, and the time cost of WP is just 2.1% of SEBA at $f_{min}$ that is the point of least time cost.

## 6.3 Comparison

Here, we compare the performance of WP with SEBA [3], SEBA+ [8], and SBF (SBF-UDP) [1], under different number of tags and the tolerable minimum number of unknown tags. We set $N = 100,000, \varepsilon = 0.01$, $m = 10, 20, 50$, and $n$ ranging from 10,000 to 100,000.

**The number of tags to be verified VS the number of participated tags:** As shown in Figure 10a, WP always has the smallest energy cost among all the protocols. In particular, SEBA and SEBA+ are the same worst (overlapping) due to no energy conservation strategy built-in, i.e., the participation probability is always 1. While SBF employs a sampling probability scheme and is better than SEBA and SEBA+, it is not as good as WP since it does not eliminate the wasteful transmission completely. Similar trends can be seen in Figure 10b and 10c as well.

**The number of tags to be verified VS time cost:** As shown in Figure 11, WP significantly outperforms all the prior schemes in terms of time cost. Specifically, when $N = 100,000, n = 80,000, \varepsilon = 0.01, m = 10$, WP is as much as 9.9x, 9x, and 2.1x faster than SEBA, SEBA+, and SBF, respectively, as shown in Figure 11a. This advantage mainly comes from the compact composite message design that has no wasteful information involved in the communication. Note that in Figure 11b and 11c, the plots for SEBA and SEBA+ are out of range of the vertical axes.

## 7. RELATED WORK

The first probabilistic unknown tag detection scheme, SEBA, is proposed in [3]. In SEBA, the reader first builds a supposed echo sketch in the back-end server, then compares it with the measured echo sketch to detect unknown tags. Then SEBA+ [8] is introduced to improve the performance of SEBA based on the bloom filter. By further exploring the characteristics of bloom filter, Liu et al. [1] combine the standard bloom filter and a sampling process to propose the Sampling Bloom Filter (SBF) for fast unknown tag detection. Although those probabilistic schemes can effectively pinpoint unknown tag events, they still suffer from inefficient communication due to the wasteful transmission of unhelpful data.

Several unknown tag identification schemes are proposed to exactly find all the unknown tags in a batch [4, 20]. When applied in unknown tag detection applications, those schemes cost much more time and energy than probabilistic detection methods [1]. There are also a number of probabilistic solutions for many other RFID problems. Probabilistic estimation schemes are proposed to acquire the approximate size of tags in interested regions [12, 13, 19, 21, 22, 23, 24, 25, 26]. But those methods only count the number of tags and so are unable to distinguish unknown tags from known ones. Several exact identification and probabilistic detection of missing tags are introduced in [14, 27, 28]. Nevertheless, missing tag problems always assume all the information about the to-be-tested tags are known in advance, which is hard to meet in the unknown tag detection. Furthermore, they can only find missing tags, but not unknown tags.

# 8. CONCLUSION

In this paper, we have proposed a fast and reliable probabilistic unknown tag detection scheme. At its core, we have constructed the composite message data structure that includes only informative data, excluding all the unhelpful data from communication. Moreover, various energy-time tradeoffs have been achieved in our analytic framework. Through detailed analysis and experiments, we have showed that the proposed protocol can significantly outperform previous methods in terms of time and energy efficiency.

# 9. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for valuable and insightful comments. This work is supported by a Canada NSERC Discovery Grant and an NSERC E.W.R. Steacie Memorial Fellowship. The work of Z. Yang is supported in part by Project NSFC (61402372) and Project of International Cooperation of Shaanxi Province (2015KW-011).

# 10. REFERENCES

[1] Xiulong Liu, Heng Qi, Keqiu Li, Ivan Stojmenovic, Alex X Liu, Yanming Shen, Wenyu Qu, and Weilian Xue. Sampling Bloom Filter-Based Detection of Unknown RFID Tags. *IEEE Transactions on Communications*, 63(4):1432–1442, 2015.

[2] Qingjun Xiao, Bin Xiao, and Shigang Chen. Differential Estimation in Dynamic RFID Systems. In *Proc. of IEEE INFOCOM*, 2013.

[3] Lei Yang, Jinsong Han, Yong Qi, and Yunhao Liu. Identification-Free Batch Authentication for RFID Tags. In *Proc. of IEEE ICNP*, 2010.

[4] Xiulong Liu, Keqiu Li, Yanming Shen, Geyong Min, Bin Xiao, Wenyu Qu, and Hongjuan Li. A Fast Approach to Unknown Tag Identification in Large Scale RFID Systems. In *Proc. of IEEE ICCCN*, 2013.

[5] Shahzad Muhammad and Alex X. Liu. Expecting the Unexpected: Fast and Reliable Detection of Missing RFID Tags in the Wild . In *Proc. of IEEE INFOCOM*, 2015.

[6] Third business mission focuses on cargo-tracking technology. http://www.winnipegfreepress.com/business/centreport-heading-back-to-china-147708545.html.

[7] Shi Chunlin. Sino-u.s.cooperation on marine transportation security: progress and problems. 2010.

[8] Giuseppe Bianchi. Revisiting an RFID Identification-Free Batch Authentication Approach. *Communications Letters, IEEE*, 15(6):632–634, 2011.

[9] Gen 2 RFID Tools. *https://moo.cmcl.cs.cmu.edu/trac/cgran/wiki/Gen2.*

[10] WISP Platform. *http://wisp.wikispaces.com/WISPFirmware.*

[11] EPCglobal Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960MHz. 2008.

[12] Murali Kodialam and Thyaga Nandagopal. Fast and Reliable Estimation Schemes in RFID Systems. In *Proc. of ACM MOBICOM*, 2006.

[13] Muhammad Shahzad and Alex X Liu. Every Bit Counts - Fast and Scalable RFID Estimation. In *Proc. of ACM MOBICOM*, 2012.

[14] Chiu Chiang Tan, Bo Sheng, and Qun Li. How to Monitor for Missing RFID tags. In *Proc. of IEEE ICDCS*, 2008.

[15] Philips Semiconductors. Your Supplier Guide to ICODE Smart Label Solutions. 2008.

[16] Zongheng Zhou, Himanshu Gupta, Samir R Das, and Xianjin Zhu. Slotted Scheduled Tag Access in Multi-Reader RFID Systems. In *Proc. of IEEE ICNP*, 2007.

[17] ThingMagic. Mercury6e rfid reader module. *www.thingmagic.com/embedded-rfid-readers.*

[18] OpenBeacon. *http://www.openbeacon.org/.*

[19] Yuanqing Zheng and Mo Li. ZOE: Fast Cardinality Estimation for Large-Scale RFID Systems. In *Proc. of IEEE INFOCOM*, 2013.

[20] Xuan Liu, Shigeng Zhang, Kai Bu, and Bin Xiao. Complete and Fast Unknown Tag Identification in Large RFID Systems. In *Proc. of IEEE MASS*, 2012.

[21] Yuanqing Zheng and Mo Li. PET: Probabilistic Estimating Tree for Large-Scale RFID Estimation. *IEEE Transactions on Mobile Computing*, 11(11):1763–1774, 2012.

[22] Binbin Chen, Ziling Zhou, and Haifeng Yu. Understanding RFID counting protocols. In *Proc. of ACM MOBICOM*, 2013.

[23] Tao Li, Samuel Wu, Shigang Chen, and Mark Yang. Energy Efficient Algorithms for the RFID Estimation Problem. In *Proc. of IEEE INFOCOM*, 2010.

[24] Wei Gong, Kebin Liu, Xin Miao, Qiang Ma, Zheng Yang, and Yunhao Liu. Informative Counting: Fine-grained Batch Authentication for Large-scale RFID Systems. In *Proc. of ACM MobiHoc*, 2013.

[25] Wei Gong, Kebin Liu, Xin Miao, and Haoxiang Liu. Arbitrarily Accurate Approximation Scheme for Large-Scale RFID Cardinality Estimation. In *Proc. of IEEE INFOCOM*, 2014.

[26] Haoxiang Liu, Wei Gong, Lei Chen, Wenbo He, Kebin Liu, and Yunhao Liu. Generic Composite Counting in RFID Systems. In *Proc. of IEEE ICDCS*, 2014.

[27] Tao Li, Shigang Chen, and Yibei Ling. Identifying the Missing Tags in a Large RFID System. In *Proc. of ACM MOBIHOC*, 2010.

[28] Yuanqing Zheng and Mo Li. P-MTI: Physical-layer Missing Tag Identification via Compressive Sensing. In *Proc. of IEEE INFOCOM*, 2013.