Fast Tag Searching Protocol for Large-Scale RFID Systems

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

***Abstract*—Fast searching a particular subset in a large number of products attached with radio frequency identification (RFID) tags is of practical importance for a variety of applications, but not yet thoroughly investigated. Since the cardinality of the products can be extremely large, collecting the tag information directly from each of those tags could be highly inefficient. To address the tag searching efficiency in large-scale RFID systems, this paper proposes several algorithms to meet the stringent delay requirement in developing fast tag searching protocols. We formally formulate the tag searching problem in large-scale RFID systems. We propose utilizing compact approximators to efficiently aggregate a large volume of RFID tag information and exchange such information with a two-phase approximation protocol. By estimating the intersection of two compact approximators, the proposed two-phase compact approximator-based tag searching protocol significantly reduces the searching time compared to all possible solutions we can directly borrow from existing studies. We further introduce a scalable cardinality range estimation method that provides inexpensive input for our tag searching protocol. We conduct comprehensive simulations to validate our design. The results demonstrate that the proposed tag searching protocol is highly efficient in terms of both time efficiency and transmission overhead, leading to good applicability and scalability for large-scale RFID systems.**

***Index Terms*—Approximate protocol, radio frequency identification (RFID), tag searching.**

I. Introduction

RADIO frequency identification (RFID) technology [7] is becoming ubiquitously available in a variety of applications, including inventory management [25], transportation and logistics [6], [12], [21], object identification and tracking [19], [31], authentication and security [16], [17], [24], [28], [32], etc. Searching a particular set of RFID tags in a large-scale RFID system is of practical importance for those applications. For example, in inventory management, there is usually a need of taking stock according to a list of items. Formally, the problem of tag searching can be defined as follows: Given a set of wanted RFID tags, we wish to know which among them, if any, currently exist within the interrogation zone of RFID readers (as Fig. 1 depicts, we want to know which tags within set $\mathbf{X}$ exist in tag set $\mathbf{Y}$ in the zone). While many active efforts have been put in studying RFID systems and significant advancement in recent years has been achieved, surprisingly, the problem of searching with a large number of tags is yet underinvestigated by the research community.

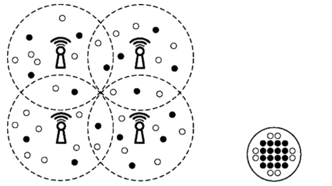


Fig. 1. RFID tag searching. Four RFID readers cover the interrogation zone $\mathbf{Y}$. Twenty-four tags are wanted $\mathbf{X}$, among which 16 tags (black dots) are indeed in the zone and the others (white dots) are absent.

Many existing works concentrate on the RFID tag identification problem [11], [18], [26], [34], which aims to identify each of a large number of RFID tags as quickly as possible. The major research issue in tag identification problem is to design efficient algorithms that resolve the tag collision problem since multiple RFID tags may contend the spatially and temporally shared communication channel. As a matter of fact, RFID identification schemes can be directly borrowed to address the tag search problem in small-scale RFID systems, i.e., when the number of RFID tags is small, we are able to collect all tag IDs within the interrogation zone and compute the intersection with a given set of wanted tags. Such solutions, however, bear the common communication collision problems among RFID tags, and in particular the long data collection process renders it inappropriate for applications with stringent delay requirement. Due to tag–tag collisions, in slotted Aloha-based identification protocols, each tag attempts 2.72 times on average to successfully deliver its ID to the reader even with ideal frame sizing [11]. As a result, the efficiency of identification protocol is very low. According to the RFID standard ISO-18000, the average identification throughput is about 100 tags per second [28]. In large-scale RFID systems, the number of RFID tags could be huge, e.g., a typical harbor port inventory management system concerns hundreds of containers, each of which may contain hundreds of thousands of products. Collecting such a vast volume of RFID tag IDs often fails to meet a stringent delay requirement. With the wider usage of RFID systems in large distribution centers, in the very near future we may face the more challenging issues due to larger number RFID tags that call for highly efficient RFID protocol designs.

By painstakingly collecting a high volume of RFID tag IDs, such identification-based approaches are far from fast and efficient tag searching. The current inadequacy motivates us to design an efficient tag searching protocol without the expensive tag ID identification phase so as to meet real-time application demands. We observe that many real applications can tolerate and are robust to certain error rate within a sufficiently small

Manuscript received January 02, 2012; revised June 12, 2012; accepted August 05, 2012; approved by IEEE/ACM Transactions on Networking Editor S. Ramasubramanian. Date of publication August 22, 2012; date of current version June 12, 2013. This work was supported by the Nanyang Technological University SCE SUG under Grant M4080103.020 and the NAP under Grant M4080738.020.

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

Digital Object Identifier 10.1109/TNET.2012.2212454

range. For example, the inventory manager may want to search for a subset of products with manufacturing flaws and such a practical application can tolerate small false positive errors, e.g., provided that all flawed products can be found, it is acceptable even if a small number of good products are wrongly identified as flawed. It is thus desirable to leverage such characteristics and seek dramatic efficiency improvement as long as the error rate can be preserved within the tolerable range.

In order to reduce the overhead of tag searching in large-scale RFID systems, we propose utilizing compact approximators to efficiently aggregate a large volume of RFID tag information and transmit the compact approximators instead of directly broadcasting or collecting tag IDs. In this paper, we present the Compact Approximator-based Tag Searching (CATS) protocol. Through two-phase compact approximator transmissions back and forth between the reader and tags, CATS efficiently finds the intersection of $\mathbf{X}$ and $\mathbf{Y}$. By transmitting the succinctly encoded sets instead of raw data sets between RFID tags and readers, the communication cost involved in the tag searching protocol is significantly reduced. In this paper, we choose the Bloom filter as a vehicle to illustrate the principle of our idea. Many variants of the Bloom filter can be used as well for the specific design purpose. Nevertheless, due to the large number of RFID tags and the heavy channel competition, how to optimize approximator utility and reduce the communication cost remains nontrivial. Constrained by computation and storage resource of RFID tags, achieving the optimal protocol performance brings challenging problems as well.

In this paper, we propose a time-efficient protocol for tag searching in large-scale RFID systems. The objective of the protocol is to: 1) examine whether any wanted tags exist in the reader interrogation region; and 2) if there are any, report the IDs of those tags. We also develop a lightweight tag cardinality range estimation algorithm for input of the tag searching protocol. We seek to reduce the communication cost and time delay involved in the tag searching process with a two-phase compact approximator exchange approach. The wanted tag set is first aggregated and broadcast to the RFID tags, and the tag feedbacks are once again aggregated and revealed to the readers. With optimal parameter settings for such a two-phase approximator exchange process, we are able to significantly reduce the transmission overhead according to particular accuracy requirement.

Our contributions can be summarized as follows. We formally introduce the tag searching problem in large-scale RFID systems. To the best of our knowledge, this is the first attempt aiming to address such a practically important and yet underinvestigated problem. We propose a baseline protocol and then, on top of it, propose a much more efficient two-phase compact approximator-based tag searching protocol that significantly improves the communication and time efficiency in tag searching with guaranteed error rate. The extensive simulation results show that compared to the baseline protocol, the proposed tag searching protocol reduces the communication time by around 79% and reduces even more compared to the tag identification-based protocols.

The rest of this paper is organized as follows. We present related work in Section II. In Section III, we formally introduce the tag searching problem and describe the design goal and requirements. We give detailed description on the tag searching protocol design and analysis in Section IV. In Section V, we conduct extensive simulations to evaluate our proposed protocols. Finally, we conclude this work in Section VI.

II. Related Work

One closely related problem is RFID identification, which aims at collecting the tag IDs of all RFIDs in the interrogation region [11], [29]. Existing RFID identification protocols generally fall into two categories: ALOHA-based [13], [23] and Tree-based [4], [18] protocols. In ALOHA-based identification protocols, the reader initiates the communication between the tags by broadcasting the query request. Upon receiving such a request, each tag randomly chooses a time-slot and transmits its ID in the corresponding slot. If the time-slot happens to be a singleton slot (i.e., the time-slot has been selected by only one tag), then the tag can be successfully identified and will keep silent in the rest of identification process. If the time-slot that the tag selected turns out to be a collision slot (i.e., more than one tag select the time-slot), then the tag cannot be identified, and therefore the tag retransmits its ID. In Tree-based identification protocols, the reader interrogates tags and detects whether any transmission collision occurs. If there are any collisions, the reader splits the tag set into two subsets and queries the subsets with fewer tags. The reader continues to split the tag set until all the tags can be successfully identified. While the RFID identification protocols can be directly borrowed to address the tag search problem, the communication overhead and time delay with large-scale RFID systems could be excessively high.

Instead of identifying all RFID tags, protocols for identifying missing tags monitor a set of tags and detect the missing-tag event [32]. The protocols periodically monitor RFID tags and report whether any tags are missing with a detection probability. In [25], the monitoring protocol sends an alarm with high probability once the number of missing tags exceeds a user-defined threshold. In [14], Li *et al.* propose a missing-tag detection protocol that can detect the missing-tag event with certainty as well as report which tags are missing. Zhang *et al.* significantly reduces the missing tag detection time by more efficiently scheduling and utilizing the multiple readers. Opposite to their purpose of finding missing tags, in this paper we focus on searching a set of tags in the interrogation area.

In [24], Tan *et al.* propose a flexible protocol to authenticate a single RFID tag. In [28], Yang *et al.* propose an efficient identification-free batch authentication for large-scale RFID systems, which provides a provable probabilistic guarantee that the percentage of potential counterfeit tags is below a threshold. However, those protocols are not suitable for efficiently searching tags in large-scale RFID systems. There are currently no works that are capable of extending authentication protocols to support tag searching in a scalable manner.

Rather than identifying the RFID tags, the RFID cardinality estimation protocols count the number of distinct tags [10], [20], [33], [37], which may serve as useful inputs for tag identification or searching. When the approximate tag number is known, RFID identification protocols can achieve best performance with near-optimal settings. The proposed tag searching protocol may also need to estimate RFID cardinality in the reader interrogation zone to calibrate protocol parameters. Such cardinality estimation approaches aiming at high accurate estimation results, however, often consume excessive transmission time, and the marginal accuracy gains decrease substantially.

III. System Model and Problem Specification

A. System Model

In our model, the RFID system consists of three components: a back-end server, a number of RFID readers, and a large number of RFID tags. The back-end server coordinates the RFID readers and has powerful computation capability. The RFID readers, connected to the back-end server via high-speed networks, transmit commands of the back-end server and later report responses back to the server. When multiple readers are synchronized, we may logically consider them as a whole. In this paper, for the sake of simplicity, we regard the multiple readers as a single coordinated logical reader. For the tags, the RFID system may use battery-powered active ones that have larger transmission range, or use lightweight passive ones that are energized by radio waves transmitted by the readers and communicate with the reader by backscattering the RF carrier according to the application requirement.

In particular, the current passive RFID tags are required to implement a set of mandatory commands, e.g., the inventory commands, according to the EPC global Class-1 Gen-2 standard [1]. It provides flexibility for manufacturers to implement customized commands. In conformance with the standard, one may extend the functionalities of RFID systems and include some value-added features. Such a flexible design motivates researchers to explore and design new features for RFID tags for practical needs with currently available components (e.g., random number generator [1], [9], lightweight hash function [30], etc). We assume that the random number generators meet the randomness criteria of the EPC standard, and the lightweight hash functions satisfy the uniform distribution requirement. Considering recent development of passive tags (e.g., WISP tags [30]), we envision that similar functions will be implemented at more lightweight passive tags.

The underlying RFID system is assumed to work on a slotted MAC model. Each tag waits for the reader's command in each round of communication, which is known as the Reader Talks First mode. Each tag contains a unique 96-bit ID according to the standard setting in the EPC global Class-1 Gen-2 standard [1].

For each communication frame, a reader initiates communication first by sending commands and the parameters to tags, e.g., selecting the tags to participate in the frame and configuring the number of slots in the frame size. If no tag transmits signals in the slot, the slot is called an *empty slot*. If one or more tags transmit signals in one slot, the slot is called a *nonempty slot*. The transmitted message can be successfully decoded if a single tag responds, and messages get corrupted when multiple tags respond in the same slot. Nevertheless, if an RFID reader only needs to determine whether one slot is chosen by any tags, we can let each tag transmit a one-bit short response in the selected slot. If the reader captures any responses in the slot, it means the slot has been selected by tag(s) and the nonempty slot can be encoded as "1." If the reader senses no response in a slot, the empty slot can be encoded as "0."

The tag-to-reader $(\mathrm{T} \Rightarrow \mathrm{R})$ transmission rate and the reader-to-tag $(\mathrm{R} \Rightarrow \mathrm{T})$ transmission rate are not necessarily symmetric depending on the physical implementation and the interrogation environment [3], [22], [27]. As specified in the EPC global Class-1 Gen-2 standard, the $\mathrm{T} \Rightarrow \mathrm{R}$ data rate is either 40–640 kb/s (FM0 encoding format) or 5–320 kb/s (Miller-modulated subcarrier encoding format), while $\mathrm{R} \Rightarrow \mathrm{T}$ data rate is normally 26.7–128 kb/s [1].

TABLE I
Key Notations

Symbols	Descriptions
$\mathbf{X}$	The set of the wanted tags
$\mathbf{Y}$	The set of tags in the interrogation zone
$\mathbf{Y}_C$	The set of candidate tags
$\mathbf{Y}_{\sim C}$	The set of non-candidate tags
$\mathbf{X} \cap \mathbf{Y}$	Intersection of $\mathbf{X}$ and $\mathbf{Y}$
$\mid \cdot \mid$	Cardinality of the set
$BF(\cdot)$	The Bloom filter for the set
$\mathbf{A} \cap BF(\mathbf{B})$	The subset of $\mathbf{A}$ that can pass $BF(\mathbf{B})$
αT_b	Reader to tag per-bit transmission time
βT_b	Tag to reader per-bit transmission time
$h(\cdot)$	A uniform hash function

B. Problem Specification

As Fig. 1 depicts, we consider a large-scale RFID system with $\mathbf{Y} = \{y_1, y_2, \ldots\}$ representing all the RFID tags in the interrogation zone covered by readers, and $\mathbf{X} = \{x_1, x_2, \ldots\}$ representing the wanted RFID tags we are interested in. *The problem of searching for RFID tags is to find the intersection* $\mathbf{X} \cap \mathbf{Y}$. Each tag $x \in \mathbf{X}$ is called a *wanted* tag. The wanted tags $\mathbf{X}$ are not necessarily in the interrogation area, i.e., the intersection can be an empty set. In other instances, all the wanted tags may be covered by readers, i.e., $\mathbf{X} \subseteq \mathbf{Y}$. We do not restrict the spatial distribution of $\mathbf{Y}$. We denote by $| \cdot |$ the cardinality of the set. For example, as depicted in Fig. 1, many tags are covered by four readers in the interrogation zone $\mathbf{Y}$. There are $|\mathbf{X}| = 24$ wanted tags, among which 16 tags (black dots) are indeed in the interrogation zone and the other 8 tags (white dots) are not. For a tag $y \in \mathbf{Y}$, with *a priori* knowledge during the searching process, if $\Pr\{y \in \mathbf{X}\} \neq 0$, the tag is considered as a *candidate* tag, and if $\Pr\{y \in \mathbf{X}\} = 0$, the tag becomes a *noncandidate* tag. The candidate tags form a subset $\mathbf{Y}_C$, and the noncandidate tags form the other subset $\mathbf{Y}_{\sim C}$. Obviously, for any time instance, the subset of candidate tags and the subset of noncandidate tags are by definition mutually exclusive and complementary, i.e., $\mathbf{Y}_C \cap \mathbf{Y}_{\sim C} = \vee$ and $\mathbf{Y}_C \cup \mathbf{Y}_{\sim C} = \mathbf{Y}$. As we accumulate more knowledge during the tag searching process, some candidate tags may eventually become noncandidate tags. Table I summarizes key notations used across this paper.

In practical situations, many applications tolerate a small false rate, as long as the false rate is sufficiently small. For most applications, in order to make the searching protocol scalable, it seems natural to trade the accuracy within a tolerable range for significant efficiency improvement. Given a false rate requirement, an ideal tag searching protocol is expected to compute the intersection $\mathbf{X} \cap \mathbf{Y}$ at minimum time and communication cost with a false rate smaller than the requirement. As a matter of fact, since the low-cost lightweight RFID tags are inherently error prone, the impractical pursuit of perfect $\mathbf{X} \cap \mathbf{Y}$ calculation may lead to excessively high overhead in realistic implementations. In spite of the search accuracy and efficiency, in many applications the anonymity of the RFID tags should be protected for privacy issues. Revealing identity information to the public might raise security and privacy concerns.

To meet above constraints and requirements, the goal of this paper is to propose a fast RFID tag searching protocol that is able to efficiently calculate the intersection $\mathbf{X}\cap\mathbf{Y}$ within a guaranteed false rate. To this end, the protocol should avoid the explicit identification of tag IDs, prevent heavy communication collisions between massive RFID tags, and reduce the transmitted bits of information.

IV. Tag Searching Protocols

In this section, we introduce several approaches to develop efficient tag searching protocols for large-scale RFID systems. We start with a warmup baseline protocol that prevents collecting all tag IDs from the interrogation zone. Motivated by the limitations of the baseline protocol, we propose the two-phase compact approximator-based tag searching protocol that overcomes the drawbacks of the motivating baseline protocol and significantly reduces the transmission cost. In addition, we propose a lightweight cardinality range estimation approach for providing rough cardinality as input to the two-phase tag searching protocol.

A. Baseline Protocol: A Warmup Solution

According to the aforementioned tag identification algorithms, we know that directly collecting tag IDs from all tags in set $\mathbf{Y}$ is highly inefficient because the transmission amount is high and the tag–tag collisions are heavy. The quantity of the tags in the interrogation zone can scale up to millions in some large-scale systems, and identifying such an amount of tags is impractical. Since we are only concerned with the set of wanted tags $\mathbf{X}$ rather than all the tags in set $\mathbf{Y}$, an obvious optimization is that, instead of identifying the tag set $\mathbf{Y}$ in interrogation zone, we let the RFID reader broadcast the IDs of tags in set $\mathbf{X}$ one by one and wait for the responses from tags in set $\mathbf{Y}$. Upon receiving the broadcasted IDs, each tag compares it to its own ID and replies immediately by sending a short response if the broadcasted ID matches its own ID, or keeps silent otherwise. For each ID, we can reserve a one-bit slot for identifying tags' binary response, i.e., "1" when tag response is received, or "0" otherwise. Instead of "pulling" IDs from set $\mathbf{Y}$, by "pushing" the tag IDs from set $\mathbf{X}$, the baseline protocol avoids collecting a large amount of tag IDs as well as the heavy tag–tag collisions during the process. Since tag IDs are 96 bits long and we need a binary response from each tag, the expected execution time of the baseline protocol is approximately

$$T_{\text{Base}} = |\mathbf{X}| \times (96 \times \alpha \times T_b + \beta \times T_b) \tag{1}$$

where $\alpha \times T_b$ denotes the per bit transmission time from readers to tags, and $\beta \times T_b$ denotes the per bit transmission time from tags to readers.

In practical large-scale RFID systems, the number of tags can scale to millions, while the number of wanted tags is usually much smaller, i.e., $|\mathbf{X}| \ll |\mathbf{Y}|$. Therefore, the baseline protocol significantly reduces the searching time. Even if $|\mathbf{X}|$ approaches $|\mathbf{Y}|$, since the baseline protocol inherently avoids tag–tag collisions, it still significantly outperforms the identification protocols in large-scale tag searching applications.

Although the baseline protocol demonstrates a promising performance improvement, it suffers several limitations that motivate our study for a more efficient and secure tag searching protocol. First, though $|\mathbf{X}|$ is probably much smaller than $|\mathbf{Y}|$, $|\mathbf{X}|$ can still be a large number for large-scale RFID systems where there are many wanted items. As the searching time increases proportionally with $|\mathbf{X}|$, the baseline protocol may still fail to meet stringent delay requirement. It is yet significant to further improve the searching efficiency. Second, the baseline protocol requires that all tags participate during the entire tag searching process, which results in unnecessary power consumptions on both reader and tag ends. Third, in the baseline protocol, unique tag IDs are explicitly transmitted and acknowledged in the air. As the explicit transmission may lead to potential privacy leaks, many works avoid such broadcasting behaviors [35], [36], [28], [37].

B. Compact Approximator-Based Tag Searching Protocol

Based on the baseline protocol, we propose a two-phase compact approximator-based tag searching protocol to further reduce transmission overhead and searching time. In particular, we transmit the compact approximators instead of explicit tag IDs to reduce transmission amount otherwise involved in broadcasting or collecting those IDs. One well-known compact approximator, Bloom filter, is capable of encoding itemized information in a hashed Boolean vector. In this paper, we use the Bloom filter as a representative approximator to carry aggregated tag ID information. One can choose a variety of compact approximators to aggregate the tag sets. Reference [2] surveys various existing techniques using such approximators.

The compact approximator-based tag searching protocol consists of two phases. In the first phase, we significantly reduce the number of candidate tags in $\mathbf{Y}_C$ by filtering the candidate tags using a Bloom filter produced with the wanted tags in $\mathbf{X}$. In the second phase, we estimate the intersection $\mathbf{X} \cap \mathbf{Y}$ by filtering the wanted tags with a virtual Bloom filter produced with the responses from the candidate tags.

1) Preliminary: Compact approximators are capable of succinctly representing a large volume of information. By transmitting the concisely encoded sets instead of the raw data sets, the communication cost involved can be significantly reduced. In this paper, we choose the Bloom filter as a vehicle to demonstrate how the compact approximator can be used to develop efficient tag searching protocol.

The Bloom filter representing a set $\mathbf{A} = \{a_1, a_2, \ldots, a_M\}$ of M elements comprises of a Boolean vector of L bits and K independent hash functions $h_i(\,\cdot\,), 1 \leq i \leq K$. Each hash function $h_i(\,\cdot\,)$ maps an element $a \in \mathbf{A} \subseteq \mathbf{\Omega}$ to a bit $h_i(a) \in \{1, 2, \ldots, L\}$, where $\mathbf{\Omega}$ represents the universal set. Initially, the L-bit array is set to "0." For each element $a \in \mathbf{A}$, the bits $h_i(a)$ are set to "1," $1 \leq i \leq K$. In order to determine whether a given element $b \in \mathbf{\Omega}$ belongs to $\mathbf{A}$, we compute K hash functions $h_i(b)$, $1 \leq i \leq K$. If all $h_i(b)$ bits on the vector have been set to "1," we assert that $b \in \mathbf{A}$, and otherwise $b \notin \mathbf{A}$. Generally, membership testing using Bloom filter has no false negatives [2], [8]. Nevertheless, it may produce false positives, i.e., an element might be misclassified to be within the set while it is not. Many practical applications tolerate such false positives, as long as the rate is sufficiently small.

Given the assumption that the K hash functions are independently and identically distributed (i.i.d.), and can uniformly map M elements into the range $\{1, 2, \ldots, L\}$, the probability of a false positive can be calculated in a straightforward way. Let P denote the probability that a particular bit remains "0." Then,

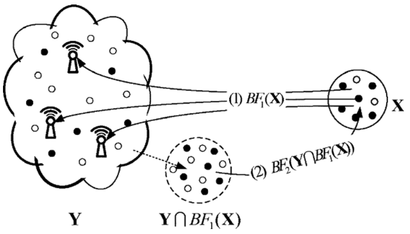

(1) $BF_1(\mathbf{X})$
X
(2) $BF_2(\mathbf{Y} \cap BF_1(\mathbf{X}))$
Y
$\mathbf{Y} \cap BF_1(\mathbf{X})$

Fig. 2. Two-phase filtering: (1) Transmission of the compact approximator $\mathrm{BF}_1(\mathbf{X})$ from back-end server to the interrogation zone. The readers broadcast $\mathrm{BF}_1(\mathbf{X})$ to the tags. The tags that pass $\mathrm{BF}_1(\mathbf{X})$ remain in candidate tag set $\mathbf{Y}_C = \mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})$. (2) The candidate tags $\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})$ respond in the second-phase communication. The responses forms $\mathrm{BF}_2(\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X}))$ for server examination.

$P = (1 - \frac{1}{L})^{M \times K} \approx e^{-M \times K/L}$ as $M \times K$ bits are independently selected by hash functions with probability $\frac{1}{L}$ for each bit. Therefore, the probability that a specific bit is set to "1" is $1 - P$. A false positive occurs when for an element $b \notin \mathbf{A}$, all $h_i(b)$ bits $(1 \le i \le K)$ are set to "1" due to the hash results of other elements. We denote the false positive rate as P_{FP}, and we have

$$P_{\mathrm{FP}} = (1 - P)^K \approx (1 - e^{-M \times K/L})^K. \tag{2}$$

It is easy to see that the minimum value of $P_{\mathrm{FP}} = 0.6185^{\frac{L}{M}}$ when the number of hash functions is $K = \frac{L}{M} \times \ln 2$ given the number of wanted tags M and the size of Bloom filter L.

In practical applications, the number of wanted tags M is usually known *a priori*. On the other hand, the size of Bloom filter vector L should be carefully selected. Since the false positive rate P_{FP} monotonically decreases with the increase of L, a larger size of Bloom filter vector guarantees a lower false positive rate. A larger L, however, in our tag searching problem may result in a larger volume of transmission in broadcasting the Bloom filter vector from the readers to RFID tags. The number of hash functions, K, determines the computation intensity on RFID tags since each tag should perform K hash functions with its constrained computation resources and storage capacities. As $K = \lfloor \frac{L}{M} \times \ln 2 \rfloor$ is proportional to the vector size L, choosing appropriate Bloom filter vector size L is critical to the protocol performance, helping to achieve better accuracy and efficiency tradeoffs. In later analysis, we assume all parameters including L and K are real numbers. In practical implementation, one can first compute the parameters, and then round them into integers.

There are many lightweight hash functions instantly available in the literature. To simplify the circuit design of passive RFID tags, one may preload random numbers on the memory chip at each tag during manufacturing. One may adopt the efficient design that organizes the random bits into a logical ring, described in [14]. Since each tag only needs a small number of random numbers (see Section V), the memory overhead is very small and within the current storage capacity of (even passive) RFID tags [1].

2) Two-Phase Compact Approximator-Based Tag Searching Protocol: We propose a Compact Approximator-based Tag Searching (CATS) protocol. CATS consists of two phases. Referring to Fig. 2, we introduce the CATS protocol in this section.

In the first phase, we reduce the candidate tags using a Bloom filter. In particular, the back-end server constructs a Bloom filter vector by mapping the wanted tags in set $\mathbf{X}$ into an L_1-bit array using K_1 hash functions with random seed S_1. The RFID readers broadcast the L_1-bit Bloom filter vector, K_1, and S_1 to all RFID tags in interrogation zone. Here, we denote the Bloom filter vector as $\mathrm{BF}_1(\mathbf{X})$. When receiving $\mathrm{BF}_1(\mathbf{X})$, K_1, and S_1, each tag $y \in \mathbf{Y}$ locally performs K_1 hash functions with random seed S_1, which are identical to those used to build $\mathrm{BF}_1(\mathbf{X})$, and checks whether the bits $h_i(x)$ of $\mathrm{BF}_1(\mathbf{X})$ are 1's for $1 \le i \le K$. If all the K_1 bits in $\mathrm{BF}_1(\mathbf{X})$ are 1's, then we say the tag y passes $\mathrm{BF}_1(\mathbf{X})$. We denote by $\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})$ the set of tags that pass the test of $\mathrm{BF}_1(\mathbf{X})$, which is the current candidate set $\mathbf{Y}_C$.

Since the Bloom filter has no false negatives, the tags that cannot pass the test of $\mathrm{BF}_1(\mathbf{X})$, denoted as $\mathbf{Y} - \mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})$, may directly classify themselves into noncandidate set $\mathbf{Y}_{\sim C}$. The noncandidate tags will keep silent and do not participate in later tag searching operations. On the other hand, the tags in $\mathbf{Y}_C$ will stay alert and participate in the following phases of CATS. We emphasize that because of the false positive problem of the Bloom filter, there are a few tags $y \notin \mathbf{X} \cap \mathbf{Y}$ that may pass the test of $\mathrm{BF}_1(\mathbf{X})$ in the first phase with a probability of P_{FP1}.

The expected number of the candidate tags after the first phase $\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})$ is

$$\begin{aligned} E(|\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})|) &= |\mathbf{Y} - \mathbf{X} \cap \mathbf{Y}| \times P_{\mathrm{FP1}} + |\mathbf{X} \cap \mathbf{Y}| \\ &\le |\mathbf{Y}| \times P_{\mathrm{FP1}} + |\mathbf{X} \cap \mathbf{Y}|. \end{aligned}$$

The false positive rate is

$$P_{\mathrm{FP1\,min}} = 0.6185^{\frac{L_1}{|\mathbf{X}|}} = \phi^{\frac{L_1}{|\mathbf{X}|}}. \tag{3}$$

In a later description, we define the constant $\phi = 0.6185$. Since P_{FP1} exponentially decreases with the increase of L_1, we can reduce the false positive rate P_{FP1} at the cost of additional transmission of a longer $\mathrm{BF}_1(\mathbf{X})$. The transmission time of $\mathrm{BF}_1(\mathbf{X})$ is

$$T_1 = L_1 \times \alpha \times T_b. \tag{4}$$

For the purpose of clarity, we ignore the transmission cost of the configuration parameters including K_1, S_1, and L_1, which normally take several bytes to encode.

After filtering $\mathbf{Y}$ with $\mathrm{BF}_1(\mathbf{X})$, the cardinality of candidate tags $|\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})|$ will become much smaller than the original cardinality $|\mathbf{Y}|$, in the cases that $|\mathbf{X} \cap \mathbf{Y}| < |\mathbf{Y}|$ and $P_{\mathrm{FP1\,min}}$ is small. At the current stage, each of the RFID tag IDs in the candidate set $\mathbf{Y}_C$ is preserved locally on the tag chip, and $\mathbf{X} \cap \mathbf{Y}$ still remains unknown to the reader. Such a set is not adequately accurate, yet explicitly letting tags within the set send their IDs back to the readers may result in heavy collisions in a large-scale RFID system.

In the second phase, the RFID readers broadcast the parameters K_2, L_2, and a new random seed S_2 to the RFID tags and initiate another round of filtering. Upon receiving the configuration parameters, each candidate tag $y \in \mathbf{Y}_C = \mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})$ calculates and selects K_2 slots at the indexes $h_i(y), (1 \le i \le K_2)$ in a frame of L_2 length. In a later process, each candidate tag transmits a short response at each of the K_2 corresponding slots. Such a process is similar with the first phase of CATS, however, with the Bloom filter coded by the responses from the tags. Fig. 3 depicts such a process. The candidate tag $y_1 \in \mathbf{Y}_C$ picks K_2 time-slots according to the hash functions and sends a short response in each of the time-slots. Other tags select the slots based on their own hash functions and response as well. In this way, a Bloom filter is formed in the air by such responses falling

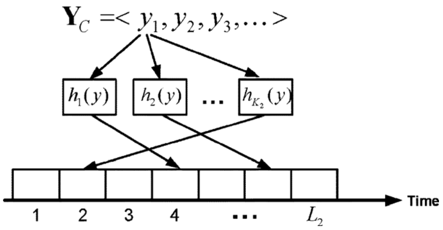


Fig. 3. Bloom filter formed in the air. Each of the candidate tags randomly selects K_2 time-slots from the L_2 slots using hash functions and sends short responses in each of the K_2 time-slots.

into different slots in the frame. As in [14], [37], and [42], we assume that the responses from tags are synchronized by the reader's interrogation. In the frame, there are two different types of slots: empty slots and nonempty slots. In particular, according to the responses from the candidate tags, the reader encodes an L_2-bit vector as follows: If the ith slot is an empty slot, the reader set the ith bit of the vector to be "0," otherwise "1." A virtual Bloom filter is thus constructed based on the responses from each of the remaining candidate tags in $\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})$. We denote the L_2-bit long vector as $\mathrm{BF}_2(\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X}))$. Note that $\mathrm{BF}_2(\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X}))$ is not managed by any RFID tags but the back-end server beyond the readers.

With the knowledge of $\mathrm{BF}_2(\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X}))$, the back-end server performs membership testing using tag IDs from the wanted tag set $\mathbf{X}$ to determine the intersection $\mathbf{X} \cap \mathbf{Y}$. The tag IDs that pass the $\mathrm{BF}_2(\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X}))$ test are considered to be within the set $\mathbf{X} \cap \mathrm{BF}_2(\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X}))$.

According to the characteristics of the Bloom filter, for arbitrary sets $\mathbf{A}$ and $\mathbf{B}$, we have

$$\begin{gathered}\mathbf{A} \cap \mathbf{B} \subseteq \mathbf{A} \cap \mathrm{BF}(\mathbf{B}) \\ |\mathbf{A} \cap \mathbf{B}| \leq |\mathbf{A} \cap \mathrm{BF}(\mathbf{B})|\end{gathered} \tag{5}$$

We can infer from (5)

$$\begin{gathered}\mathbf{X} \cap \mathbf{Y} \subseteq \mathbf{X} \cap \mathrm{BF}_2(\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})) \\ |\mathbf{X} \cap \mathbf{Y}| \leq |\mathbf{X} \cap \mathrm{BF}_2(\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X}))|\end{gathered} \tag{6}$$

In the CATS protocol, we use $\mathbf{X} \cap \mathrm{BF}_2(\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X}))$ to approximate the intersection $\mathbf{X} \cap \mathbf{Y}$.

C. Joint Optimization

Although CATS guarantees that any wanted tag $x \in \mathbf{X} \cap \mathbf{Y}$ can be correctly classified, due to false positive property of the Bloom filter, there might be unwanted tags misclassified as in $\mathbf{X} \cap \mathbf{Y}$.

The false positive probability after the second phase filtering is

$$P_{\mathrm{FP2}} \approx (1 - e^{-|\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})| \times K_2 / L_2})^{K_2}. \tag{7}$$

With the optimal setting $K_2 = \frac{L_2}{|\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})|}$, the minimum false probability is

$$P_{\mathrm{FP2\,min}} = \phi^{L_2 / |\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})|}. \tag{8}$$

Therefore, the expected number of tags that are misclassified is $|\mathbf{X} - \mathbf{X} \cap \mathbf{Y}| \times P_{\mathrm{FP2}}$. Given a tolerable false positive rate P_{REQ}, the CATS is able to guarantee $P_{\mathrm{FP2}} \leq P_{\mathrm{REQ}}$ with joint optimization on L_1 and L_2.

The transmission time of the frame with L_2 response slots in the second phase is

$$T_2 = L_2 \times \beta \times T_b. \tag{9}$$

Similar to the first phase, we ignore the transmission cost of configuration parameters including K_2, S_2, and L_2.

According to (4) and (9), the total transmission time in CATS is

$$T_t = T_1 + T_2 = T_b \times (\alpha L_1 + \beta L_2) = T_b \times L_t \tag{10}$$

where $L_t = \alpha L_1 + \beta L_2$ abstracts the total bits transmitted in CATS.

According to (5), (7), and (8), we have

$$\begin{aligned}P_{\mathrm{FP2}} &= \phi^{L_2/|\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})|} \leq \phi^{L_2/(|\mathbf{Y}| P_{\mathrm{FP1}} + |\mathbf{X} \cap \mathbf{Y}|)} \\ &\leq \phi^{L_2/(|\mathbf{Y}| \phi^{L_1/|\mathbf{X}|} + |\mathbf{X} \cap \mathbf{Y}|)} \leq \phi^{L_2/(|\mathbf{Y}| \phi^{L_1/|\mathbf{X}|} + |\mathbf{X}|)}.\end{aligned} \tag{11}$$

To maximize the protocol efficiency, we expect to meet the false rate requirement of P_{REQ} with a minimum total transmission time. Therefore, the problem can be modeled as the following optimization problem:

$$\begin{aligned}\text{Minimize}: &\quad L_t = \alpha L_1 + \beta L_2 \\ \text{Subject to}: &\quad \phi^{L_2/(|\mathbf{Y}| \phi^{L_1/|\mathbf{X}|} + |\mathbf{X}|)} \leq P_{\mathrm{REQ}}.\end{aligned}$$

We solve such a problem with the Lagrange multiplier technique. We define

$$\Lambda(L_1, L_2, \lambda) = \alpha L_1 + \beta L_2 + \lambda(\phi^{L_2/(|\mathbf{Y}| \phi^{L_1/|\mathbf{X}|} + |\mathbf{X}|)} - P_{\mathrm{REQ}})$$

and solve $\nabla_{L_1, L_2, \lambda} \Lambda(L_1, L_2, \lambda) = 0$.

We obtain the optimal settings for L_1 and L_2 for the optimization problem as follows:

$$\begin{aligned}L_1 &= |\mathbf{X}| \log_{\phi} \left(-\frac{\alpha |\mathbf{X}|}{\beta |\mathbf{Y}| \ln P_{\mathrm{REQ}}} \right) \\ L_2 &= \frac{|\mathbf{X}|}{\ln \phi} \left(\ln P_{\mathrm{REQ}} - \frac{\alpha}{\beta} \right).\end{aligned} \tag{12}$$

This way, we can guarantee the false positive probability of P_{REQ} with the total transmission of L_1-bit R $\Rightarrow$ T and L_2-bit T $\Rightarrow$ R transmissions.

According to (12), one may notice that CATS requires the cardinality $|\mathbf{X}|$ of the wanted tags as well as the cardinality $|\mathbf{Y}|$ of the tags in the interrogation zone as inputs to set the optimal frame size L_1 and L_2. While in most cases $|\mathbf{X}|$ is already known in advance to the back-end server, in practice, we may have only rough estimation on the cardinality $|\mathbf{Y}|$ of the RFID tags in interrogation zone.

Therefore, we investigate the sensitivity and robustness of the protocol optimality to the variance of $|\mathbf{Y}|$. If the optimal frame size setting is very sensitive to the variance on $|\mathbf{Y}|$, tuning the frame sizes with an inaccurate set cardinality might result in huge performance degradation.

From (12), we find that $|\mathbf{Y}|$ directly influences the setting of frame size L_1 in the first-phase of CATS. We compute the first order derivative as follows:

$$\frac{dL_1}{d|\mathbf{Y}|} = -\frac{|\mathbf{X}|}{|\mathbf{Y}| \ln \phi} \approx \frac{2.08 |\mathbf{X}|}{|\mathbf{Y}|}. \tag{13}$$

From (13), we notice that the fraction $\frac{|\mathbf{X}|}{|\mathbf{Y}|}$ determines the first-order derivative. Since $|\mathbf{Y}|$ tends to be much bigger than $|\mathbf{X}|$

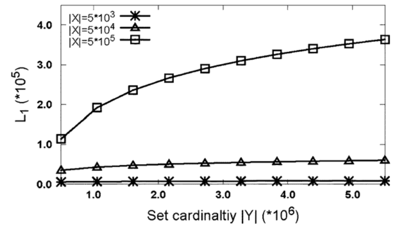


Fig. 4. Sensitivity analysis: L_1 against $|\mathbf{Y}|$.

in most scenarios, the first-order derivative can be very small, meaning that L_1 is not sensitive to the variance of $|\mathbf{Y}|$.

Fig. 4 plots the optimal setting of frame size L_1 given $|\mathbf{X}| = 5 \times 10^3$, 5×10^4, and 5×10^5, and $|\mathbf{Y}|$ varying from 1×10^6 to 5×10^6. When $\frac{|\mathbf{X}|}{|\mathbf{Y}|}$ is small, the derivative of L_1 is very small. On the other hand, Fig. 4 suggests that by slightly increasing L_1, we are able to accommodate a much larger tag set $\mathbf{Y}$.

In Section IV-D, we will later show that it is communication economic to use a slightly increased Bloom filter length L_1 so as to guarantee P_{REQ} with high probability.

D. Cardinality Range Estimation

In some application scenarios, even a rough estimation of $|\mathbf{Y}|$ is not available, and thus we have to estimate the cardinality of the tag population. There are many existing tag cardinality estimation algorithms. Those algorithms, however, need a relatively long processing time to derive accurate estimation results [20], [10], [33], [37]. The marginal accuracy improvement decreases dramatically when one pursues higher accuracy estimation results. This observation motivates a rough estimation of $|\mathbf{Y}|$ range rather than an accurate estimation with excessive communication overhead.

Instead of pursuing an accurate cardinality estimation result at excessive communication cost, since the parameters are not sensitive to the estimation error of the tag cardinality, we propose a lightweight cardinality range estimation component to provide cardinality input for CATS. Aiming at a rough cardinality range estimation, the proposed approach tries to maximize the marginal gain of estimation accuracy. During the process, there are several estimation rounds. In each round, the reader collects the 1-bit slot containing empty or nonempty responses from tags in order to roughly estimate the size of tag set.

At the beginning, the reader broadcasts a threshold u and the number of estimation rounds n to the tags and monitors the communication channel for the responses from tags in the following n slots. When receiving the threshold u and the number of estimation rounds n, each tag independently computes a binary random vector with a uniform distribution hash function $h_B(\text{ID})$. We denote by $R(\text{ID})$ the position of right-most zero of $h_B(\text{ID})$. For example, assuming that

$$h_B(\text{ID}_1) = 0100\underbrace{11}_{2}, h_B(\text{ID}_2) = 100\underbrace{111}_{3}, \tag{14}$$

$R(\text{ID}_1) = 2$ and $R(\text{ID}_2) = 3$, respectively. Obviously, the random number $R(\text{ID})$ follows geometric distribution with probability $\Pr\{R(\text{ID}) = k\} = \frac{1}{2^{k+1}}$.

We denote by R_{ij}, $(1 \leq i \leq |\mathbf{Y}|, 1 \leq j \leq n)$ the random number of tag i in the jth estimation round. In the estimation round j, if $R_{ij} > u$, then tag i responds to the reader by sending a 1-bit short response; otherwise the tag keeps silent. As a result, n consecutive empty or nonempty slots will be sensed by the reader.

The probability that each tag keeps silent in the jth estimation round is

$$\Pr(R_{ij} < u) = \sum_{k=0}^{u-1} \Pr(k) = \sum_{k=0}^{u-1} \frac{1}{2^{k+1}} = 1 - \frac{1}{2^u}. \tag{15}$$

In the case that all $R_{ij} < u$, $i \in \{1, 2, \ldots, |\mathbf{Y}|\}$, the reader observes no signal from the tags, i.e., the channel is idle. Therefore, with the assumption of i.i.d. for R_{ij}, the probability that the reader observes an idle channel in the jth estimation round is as follows:

$$\Pr(\text{idle}) = [\Pr(R_i < u)]^{|\mathbf{Y}|} = \left(1 - \frac{1}{2^u}\right)^{|\mathbf{Y}|}$$
$$\approx e^{-|\mathbf{Y}|/2^u} = e^{-\rho}$$

where we define $\rho = \frac{|\mathbf{Y}|}{2^u}$.

We define a Bernoulli random variable X that takes value 1 with probability $\Pr(\text{idle}) \approx e^{-\rho}$ and value 0 with probability $1 - \Pr(\text{idle}) \approx 1 - e^{-\rho}$, so we have

$$\Pr(X = 1) \approx e^{-\rho} \quad \Pr(X = 0) \approx 1 - e^{-\rho}. \tag{16}$$

Therefore, according to (16), the expected value and the variance of X are

$$E(X) = e^{-\rho} \quad \sigma^2(X) = e^{-\rho}(1 - e^{-\rho}). \tag{17}$$

The maximum variance of X is $\sigma^2(X)_{\text{MAX}} = 0.25$, when $e^{-\rho} = 0.5$.

We define the average value of n measurements as $\bar{X} = \frac{1}{n}\sum_{j=1}^{n} X_j$. Then, the expectation and the variance of $\bar{X}$ are

$$E(\bar{X}) = E(X) = e^{-\rho} \quad \sigma^2(\bar{X}) = \frac{\sigma^2(X)}{n} \leq \frac{0.25}{n}. \tag{18}$$

According to (18), the observation of $\bar{X}$ can be used to estimate the set cardinality $|\hat{\mathbf{Y}}|$ as follows:

$$|\hat{\mathbf{Y}}| = -2^u \ln \bar{X}. \tag{19}$$

The challenge in such a cardinality estimation approach arises, however, when either $\Pr(\text{idle}) \approx e^{-\rho} \to 0$ or 1. Without observing an adequate number of distinct channel states, the estimation accuracy on $|\hat{\mathbf{Y}}|$ would be poor [5]. This motivates us to adjust the threshold u so as to quickly adapt to the cardinality range. The RFID reader adaptively calibrates u according to the tags' responses and progressively narrows down the estimation range, e.g., if we observe very few idle states (i.e., $\bar{X} \to 0$), we infer that $|\mathbf{Y}| \gg 2^u$ and increment u; if we observe idle channel in almost all rounds (i.e., $\bar{X} \to 1$), we infer that $|\mathbf{Y}| \ll 2^u$ and decrement u.

The expected value of $\bar{X}$ is nondecreasing with the increase of u. With the monotonic feature, we can speed up the convergence of u and narrow down the estimating range with a bisection search. Fig. 5 plots an example of the cardinality range estimation process with bisection search. The cardinality under investigation is $|\mathbf{Y}| = 2^{10} = 1024$. For each step of estimation, we repeat $n = 32$ rounds to derive $\bar{X}$. At the first step, we estimate with $u = \frac{0+32}{2} = 16$. The reader observes consecutive 1's (which leads to $\bar{X} = 1$), therefore we infer $|\mathbf{Y}| \ll 2^u$. At the second step, we perform estimation with

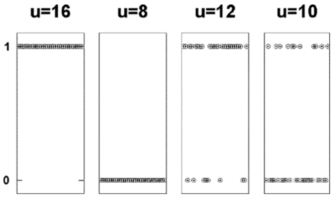


Fig. 5. Bisection search for fast estimation range convergence.

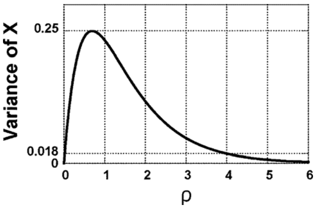


Fig. 6. Tag response variance with $\rho = \frac{|\mathbf{Y}|}{2^u}$.

$u = \frac{0+16}{2} = 8$. In such a case, the reader observes consecutive 0's ($\bar{X} = 0$), indicating $|\mathbf{Y}| \gg 2^u$. At the third step, we estimate with $u = \frac{8+16}{2} = 12$. The reader observes mixed 0's and 1's ($\bar{X} = \frac{24}{32} = 0.75$), and we can compute the estimated cardinality by $|\hat{\mathbf{Y}}| = -2^u \ln \bar{X} = -2^{12} \times \ln 0.75 \approx 1178$. At the final step, we run the estimation with $u = \frac{8+12}{2} = 10$, with the estimation result of $|\hat{\mathbf{Y}}| = -2^{10} \times \ln(\frac{11}{32}) \approx 1093$. By such means, we rapidly converge the estimated cardinality range toward the real value.

Fig. 6 plots the variance of X against ρ. We find that when $\rho = \frac{|\mathbf{Y}|}{2^u} \to 0$ or $\rho = \frac{|\mathbf{Y}|}{2^u} > 4$, the variance of X becomes very small, which implies that though *inaccurate* to directly compute $|\hat{\mathbf{Y}}|$, $\bar{X}$ is relatively *stable* to tell the scale of $|\hat{\mathbf{Y}}|$ in such a case. Mapped back to the above example in Fig. 5, this is the main reason why at the first and the second steps, we can rapidly and confidently narrow down the estimation range with only a small number of estimation rounds (32 rounds of all 0's or all 1's).

Since the result may still vary slightly because of the estimation variance, we seek a guaranteed cardinality estimation confidence range, i.e., $\Pr\{|\mathbf{Y}|(1-\varepsilon) \le |\hat{\mathbf{Y}}| \le |\mathbf{Y}|(1+\varepsilon)\} \ge 1 - \frac{1}{k^2}$. We can rewrite the estimation range requirement as follows:

$$\begin{aligned}\Pr\{|\mathbf{Y}|(1-\varepsilon) \le |\hat{\mathbf{Y}}| \le |\mathbf{Y}|(1+\varepsilon)\} \\ = \Pr\left\{e^{-\frac{|\mathbf{Y}|(1+\varepsilon)}{2^u}} \le \bar{X} \le e^{-\frac{|\mathbf{Y}|(1-\varepsilon)}{2^u}}\right\}. \quad (20)\end{aligned}$$

According to Chebyshev's inequality, we have

$$\Pr\left\{e^{-\frac{|\mathbf{Y}|}{2^u}} - \frac{k}{2\sqrt{n}} \le \bar{X} \le e^{-\frac{|\mathbf{Y}|}{2^u}} + \frac{k}{2\sqrt{n}}\right\} \ge 1 - \frac{1}{k^2}. \quad (21)$$

Combining (20) and (21), we compute the minimum estimation round n^* which can guarantee the estimation requirement $\Pr\{|\mathbf{Y}|(1-\varepsilon) \le |\hat{\mathbf{Y}}| \le |\mathbf{Y}|(1+\varepsilon)\} \ge 1 - \frac{1}{k^2}$.

The estimation result influences the first-phase parameter setting in the CATS protocol, such as L_1 and K_1. Since the CATS protocol can accommodate a larger number of tags by slightly increasing L_1, it is communication-economic to enhance the robustness at a small communication cost. Assume that the estimation value is $|\hat{\mathbf{Y}}|$ and the length of the Bloom filter in the first phase is $L_1(|\hat{\mathbf{Y}}|)$, then to accommodate $|\hat{\mathbf{Y}}|(1+\varepsilon)$ tags, according to (12), we can slightly increase L_1 by $|\mathbf{X}| \log_\phi \frac{1}{(1+\varepsilon)}$. In the later evaluation part, we show that $\varepsilon = 10\%$ already gives us quite good performance.

E. Discussion

1) Further Optimization Before the Second-Phase Filtering: We jointly optimize the Bloom filter sizes L_1 and L_2 and preset the frame lengths. As a matter of fact, after the filtering, we have another chance to estimate the cardinality of short-listed candidate tags $|\mathbf{Y}_C| = |\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})|$ and further tune the second-phase parameters L_2 and K_2 accordingly

$$\begin{aligned}L_2 &= |\mathbf{Y} \cap BF_1(\mathbf{X})| \times \log_\phi P_{\mathrm{REQ}} \\ K_2 &= \frac{L_2}{|\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})|} \times \ln 2 \quad (22)\end{aligned}$$

This provides one more chance to optimize the frame size L_2 in order to achieve the false positive requirement with even lower communication cost. In the scenario that $|\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})|$ is very small and execution time involved in collecting the IDs is within the delay requirement, the tag searching protocol may even directly collect the IDs from those tags instead of performing the second filtering phase.

In this paper, we particularly focus on the large-scale RFID system. The reason is when the number of RFID tags is small, one may directly use the baseline protocol or resort to anti-collision identification protocols to collect all the tags in the interrogation area. The identification time may not be a compelling issue in small-scale systems. Since the tag cardinality estimation protocols can approximate the tag population, if the tag population in the interrogation area turns out very small, one may use the baseline protocol or the identification-based approach. For instance, one may compare T_t to T_{Base} to determine whether the baseline protocol or the CATS protocol should be used for tag searching.

2) Multiple Readers and Mobile Tags: Since the propagation range of both RFID readers and RFID tags are limited, many large-scale applications deploy multiple readers to enhance the coverage for a large number of tags in the interrogation region. In such scenarios, duplicate readings of the same object are very common. In CATS, however, the back-end server aggregates the 1-bit responses from all readers. Even if a tag is located in the overlapped interrogation region and its response is overheard by multiple readers, its impact on the back-end aggregation is equivalent to a single response. Therefore, the CATS protocol well handles the multiple-reader scenario with the duplicate-insensitive nature in tag responses. When RFID tags are attached to mobile objects and move within the interrogation region of multiple readers, the response from the same tag will finally converge at the back-end server even if it might go through multiple readers. Such a scenario is equivalent to that of the multiple readers and thus can be correctly handled by the CATS protocol.

3) Computation Overhead: In CATS, the back-end server needs to encode the large number of tag IDs into the Bloom filter with a number of hash functions and perform membership testing based on the responses from tags in the second phase. As we compute the Bloom filter with powerful servers and the computation can easily be parallelized, the computation time involved in encoding and membership testing is in the subsecond level. Moreover, many efficient variants of the compact approximators (e.g., counting Bloom filter) can be used to further reduce the computation overhead at the servers. The RFID tags are required to perform a certain amount of computation as well, but the computation overhead is very limited, i.e., each tag only needs to compute a small number of hash values and responds according to the reader's interrogation (e.g., about 10 hash values; cf. Section V). The computation at each tag is performed in an independent and distributed manner, and it takes several milliseconds to compute a hash value at RFID tags [1]. Therefore, the computation time involved in hash function computation at each tag is negligible. Considering the most recent development at RFID tags [30], we may expect that the hash functions will be available at more lightweight passive tags.

4) Energy Consumption: Many RFID systems deploy multiple readers to cover the interrogation region, and each reader draws substantial energy to communicate with tags (e.g., it sends radio waves to energize passive tags and broadcasts tag IDs). Therefore, energy conservation must be taken into consideration of the protocol design [38]–[41]. While the baseline protocol in this paper requires that all tags participate in the entire tag searching process, the CATS protocol can significantly reduce the energy consumption as the readers and tags transmit the succinctly encoded sets instead of raw tag IDs. Moreover, since only the candidate tags are required to stay alert and participate in the second phase, the noncandidate tags can keep silent and thus save a substantial amount of energy.

5) Anonymity: In some applications where the tag ID carries private information about the associated item, explicitly transmitting such information might lead to privacy leakage. CATS resists such privacy threats because each RFID tag does not explicitly broadcast its ID. Instead, each tag responds to the reader's query according to implicit hash results. In addition, the tag does not reveal the hash result directly to the public. At each response slot, a number of tags reply to the reader, and their responses cumulate. Neither the readers nor any overhearing entities can distinguish the exact set of tags that respond at a collision slot. Consequently, the hash values of RFID tags are well preserved from any eavesdroppers.

V. Evaluation

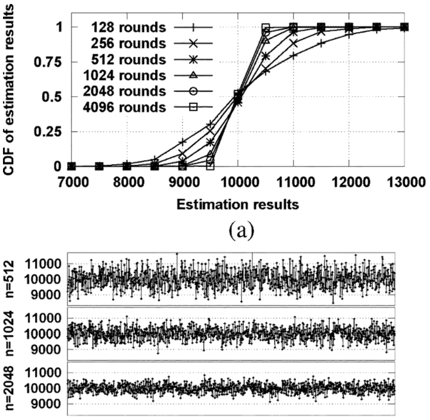
A. Simulation Setting and Performance Metrics

In the simulations, we focus on the large-scale RFID systems where the tag cardinality is big and $|\mathbf{X}| < |\mathbf{Y}|$. We assume that there is no transmission loss between RFID tags and the reader. In each frame, the reader initiates the communication by sending commands to the tags and waits for tag's response. The RFID reader is capable of detecting and distinguishing empty slots from nonempty slots. All presented results are obtained by averaging over 150 runs.

We mainly consider the searching efficiency given a tolerable false positive. Since both $\mathrm{T} \Rightarrow \mathrm{R}$ and $\mathrm{R} \Rightarrow \mathrm{T}$ transmission rates vary depending on various factors, we assume both $\mathrm{T} \Rightarrow \mathrm{R}$ and $\mathrm{R} \Rightarrow \mathrm{T}$ data rates to be 40 kb/s, i.e., the transmission time for each bit equals 25 μs [1]. The total transmission time reflects the protocol efficiency. The protocol with short transmission time will be able to scale up with more RFID tags.

We also concern whether the tag searching algorithm can guarantee the tolerable false positive specified by users. We use the false positive fraction as the accuracy indicator, denoted as

$$\mathrm{fraction}_{\mathrm{FP}} = \frac{|\{x | x \in \mathbf{X} - \mathbf{X} \cap \mathbf{Y}, x \in \mathbf{X} \cap \mathrm{BF}_2\}|}{|\mathbf{X} - \mathbf{X} \cap \mathbf{Y}|}$$



(a)

(b)

Fig. 7. Cardinality range estimation with $|\mathbf{Y}| = 10\,000$. (a) Cumulative distribution of estimation results. (b) Estimation results with estimation rounds $n = 512$, 1024, and 2048.

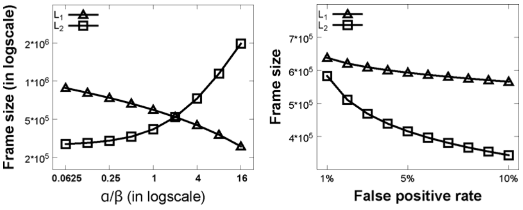
where $X \cap \mathrm{BF}_2$ denotes the set of the tags that pass the two-phase membership test. Given a false positive requirement P_{REQ}, the protocol is expected to guarantee $\mathrm{fraction}_{\mathrm{FP}} \leq P_{\mathrm{REQ}}$.

B. Protocol Investigation

1) Cardinality Range Estimation: We first investigate the cardinality range estimation algorithm and demonstrate its effectiveness. In the simulation, we set the tag cardinality in the interrogation region to be 10 000. Fig. 7(a) illustrates the cumulative distribution of the estimated values for the tag cardinality with different rounds of estimation. Fig. 7(b) plots 1000 estimation results for each of the 512, 1024, and 2048 estimation rounds, respectively. We observe that the cardinality estimation protocol provides tunable estimation accuracy in terms of estimation time, i.e., the more estimation rounds we run, the more accurate estimation results will be. The marginal accuracy improvement, however, decreases when we reach 1024 rounds. Moreover, since the CATS algorithm is not sensitive to the tag cardinality in the interrogation region, we prefer rough estimation rather than expensive high accuracy estimation. The empirical results show that 1024 estimation rounds suffice to get accurate cardinality range input for CATS. As depicted in Fig. 7(b), with 1024 estimation rounds, we observe that: 1) most estimation results, 99.3%, are within the confidence range [9000, 11000]; and 2) for the small portion, 0.7%, of the estimation results off the confidence range are still very close to the confidence range. Therefore, in later experiments, we estimate the tag cardinality range with 1024 rounds of estimation and set tolerable cardinality range estimation error to be $\varepsilon = 10\%$.

2) CATS Investigation: In this section, we investigate the CATS protocol performance. The cardinality of tag set $\mathbf{Y}$ is either known in advance or estimated with the previous cardinality range estimation approach.

Fig. 8 plots the optimal frame sizes L_1 and L_2 under different settings of $\mathrm{R} \Rightarrow \mathrm{T}$ and $\mathrm{T} \Rightarrow \mathrm{R}$ data rates, as well as different false positive requirements. In Fig. 8(a), we fix $P_{\mathrm{REQ}} = 5\%$ and vary $\frac{\alpha}{\beta}$ from 0.0625 to 16. When $\frac{\alpha}{\beta}$ is small, meaning that $\mathrm{R} \Rightarrow \mathrm{T}$ per bit transmission time is smaller than that of $\mathrm{T} \Rightarrow \mathrm{R}$, CATS prefers a longer L_1 and a shorter L_2. When $\frac{\alpha}{\beta}$ is big, CATS prefers a shorter L_1 and a longer L_2 so as to shift more communications to the second phase. In Fig. 8(b), we fix $\frac{\alpha}{\beta} = 1$



(a) (b)

Fig. 8. (a) $\frac{\alpha}{\beta}$ varies from $\frac{1}{16}$ to 16, and the same false positive rate $P_{\mathrm{REQ}} = 5\%$. (b) P_{REQ} varies from 1% to 10%, and the symmetric data rates $\frac{\alpha}{\beta} = 1$.

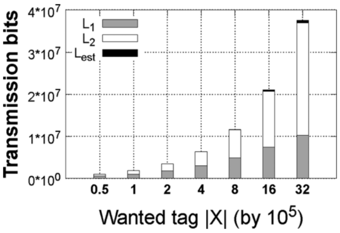


Fig. 9. Transmission overhead: L_1, L_2, and L_{est}.

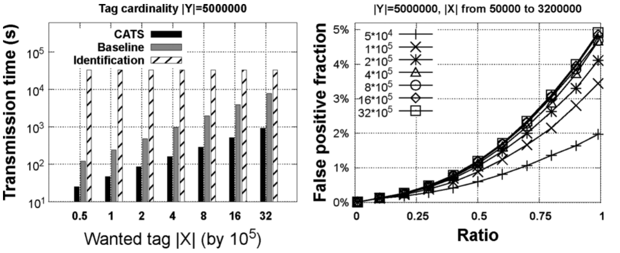
TABLE II
PARAMETER SETTING ($\frac{\alpha}{\beta} = 1$, $P_{\mathrm{REQ}} = 0.05$, AND $|\mathbf{Y}| = 5\,000\,000$)

$\lvert\mathbf{X}\rvert(10^5)$	$K_{(1,2)}$	L_1	L_2	L_{est}	L_{sum}
0.5	(8,4)	593429	415825	11099	1020353
1	(7,4)	1042590	831650	21014	1895254
2	(6,4)	1796645	1663301	40844	3500790
4	(5,4)	3016219	3326602	80504	6423325
8	(4,4)	4878294	6653204	159824	11691322
16	(3,4)	7448301	13306408	318464	21073173
32	(2,4)	10280027	26612817	635744	37528588

and vary P_{REQ} from 1% to 10%. A larger tolerable false positive rate trades for shorter frame sizes and thus shorter transmission time.

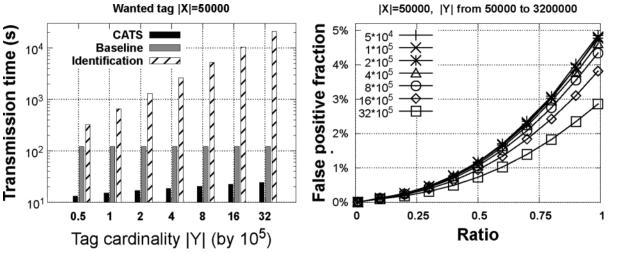
For simplicity, in the following experiments we assume symmetric $\mathrm{T} \Rightarrow \mathrm{R}$ and $\mathrm{R} \Rightarrow \mathrm{T}$ data rates (i.e., $\alpha = \beta$), and a required false positive rate $P_{\mathrm{REQ}} = 5\%$. In Table II, we show the optimal parameter settings given $|\mathbf{X}|$ varying from 50 000 to 3 200 000 and $|\mathbf{Y}| = 5\,000\,000$. $K_{(1,2)}$ represents the hash functions that each tag will take in the first- and second-phase Bloom filter membership testing and inserting. L_1 and L_2 are the optimized Bloom filter sizes in the two phases. L_{est}, representing the cardinality range estimation overhead, consists of $\log_2 32 = 5$ bisection steps (32 rounds for each step) to converge to an estimation range, and 1024 rounds to sharpen the accuracy (with $\varepsilon = 10\%$). We count the extra $|\mathbf{X}| \log_{\phi} \frac{1}{(1+\varepsilon)}$ in L_{est} as well. L_{sum} denotes total bit transmission involved in the entire CATS protocol, i.e., $L_{\mathrm{sum}} = L_1 + L_2 + L_{\mathrm{est}}$. According to Table II, we find L_{est} counts for less than 2% of the total bit transmission L_{sum}. L_{sum} has already taken the cardinality range estimation overhead into consideration and is used to abstract the overall transmission overhead of CATS. Fig. 9 plots the transmission overhead involved in different phases. According to Fig. 9, we can explicitly find that while effective in estimating the cardinality range, the overhead of cardinality range estimation L_{est} only counts for a very small portion of total transmission time L_{sum}.

Fig. 10(a) compares the total transmission time of tag searching with CATS, the baseline protocol, and an ideal ALOHA-based identification protocol with the optimal efficiency of 36.8% [11]. The expected transmission time of the baseline protocol is almost 4.81 to 8.41 times that of CATS.



(a) (b)

Fig. 10. CATS investigation with varied $|\mathbf{X}|$ from 0.5×10^5 to 32×10^5 and the fixed $|\mathbf{Y}| = 5\,000\,000$. (a) Total transmission time (in *log scale*). (b) False positive fraction.



(a) (b)

Fig. 11. CATS investigation with varied $|\mathbf{Y}|$ from 0.5×10^5 to 32×10^5 and the fixed $|\mathbf{X}| = 50\,000$. (a) Total transmission time (in *log scale*). (b) False positive fraction.

According to the results, CATS significantly outperforms the optimal performance state-of-the-art identification-based protocols can achieve. Besides the search efficiency, we are also interested in whether the false positive rate of CATS is within the requirement bound. Fig. 10(b) plots the false positive fraction of CATS with different scenarios, where we fix $|\mathbf{Y}| = 5\,000\,000$ and vary $|\mathbf{X}|$ from 50 000 to 3 200 000. We observe that, for each setting, the false positive rate increases as $\mathrm{Ratio} = \frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X}|}$ increases, and that the false positive rate also increases with $|\mathbf{Y}|$. The false positive rates of all tests remain within the tolerable bound $P_{\mathrm{REQ}} = 5\%$.

One may notice that in the case of $|\mathbf{X}| = 50\,000$, the false positive rate is below 2%, indicating that CATS performs beyond the expected false positive rate. It is mainly because we consider the worst case in tuning CATS parameters (L_1, L_2, and the like) without the knowledge of $|\mathbf{X} \cap \mathbf{Y}|$, i.e., we bound $|\mathbf{Y} \cap \mathrm{BF}_1(\mathbf{X})|$ with $|\mathbf{Y}| P_{\mathrm{FP1}} + |\mathbf{X}|$.

In Fig. 11, we examine the same metrics by varying $|\mathbf{Y}|$ from 50 000 to 3 200 000 with the fixed $|\mathbf{X}| = 50\,000$. Similarly, Fig. 11(a) shows that CATS significantly reduces transmission time compared to the other two approaches. Fig. 11(b) shows that CATS secures the false positive rate $P_{\mathrm{REQ}} = 5\%$ for all tests.

VI. CONCLUSION AND FUTURE WORK

In this paper, we study the tag searching problem in large-scale RFID systems. The solution to such a problem is of significant importance for many RFID management applications. To meet the stringent delay requirement, we propose CATS, a compact approximator-based tag searching protocol, which significantly improves the searching efficiency in comparison to the state-of-the-art approaches, while being able to secure an arbitrary required false rate. We propose a lightweight cardinality range estimation algorithm for providing cardinality input to CATS. We do extensive simulations to evaluate the performance of CATS, and the results demonstrate that CATS outperforms other possible solutions originated from existing approaches. For the future work, we are planning to implement the CATS protocol at passive RFID tags (e.g., WISP tags [30]) and validate its feasibility and efficiency under practical scenarios. We will also work on active tags and investigate the energy consumptions of the proposed tag searching protocols.

REFERENCES

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

[2] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet Math.*, vol. 1, no. 4, pp. 485–509, 2003.

[3] M. Buettner and D. Wetherall, "An empirical study of uhf RFID performance," in *Proc. ACM MobiCom*, 2008, pp. 223–234.

[4] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inf. Theory*, vol. IT-25, no. 5, pp. 505–515, Sep. 1979.

[5] B. Chen, Z. Zhou, Y. Zhao, and H. Yu, "Efficient error estimating coding: Feasibility and applications," in *Proc. ACM SIGCOMM*, 2010, pp. 3–14.

[6] S. Chen, M. Zhang, and B. Xiao, "Efficient information collection protocols for sensor-augmented RFID networks," in *Proc. IEEE INFOCOM*, 2011, pp. 3101–3109.

[7] K. Finkenzeller, *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*. New York: Wiley, 2000.

[8] D. Guo, Y. Liu, X. Li, and P. Yang, "False negative problem of counting bloom filter," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 651–664, May 2010.

[9] D. Holcomb, W. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Trans. Comput.*, vol. 58, no. 9, pp. 1198–1210, Sep. 2009.

[10] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. ACM MobiCom*, 2006, pp. 322–333.

[11] T. F. La Porta, G. Maselli, and C. Petrioli, "Anticollision protocols for single-reader RFID systems: Temporal analysis and optimization," *IEEE Trans. Mobile Comput.*, vol. 10, no. 2, pp. 267–279, Feb. 2011.

[12] C.-H. Lee and C.-W. Chung, "Efficient storage scheme and query processing for supply chain management using RFID," in *Proc. ACM SIGMOD*, 2008, pp. 291–302.

[13] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proc. IEEE MobiQuitous*, 2005, pp. 166–172.

[14] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. ACM MobiHoc*, 2010, pp. 1–10.

[15] T. Li, S. Wu, S. Chen, and M. Yang, "Energy efficient algorithms for the RFID estimation problem," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[16] L. Lu, Y. Liu, and X. Li, "Refresh: Weak privacy model for RFID systems," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[17] D. Molnar and D. Wagner, "Privacy and security in library RFID: issues, practices, and architectures," in *Proc. ACM CCS*, 2004, pp. 210–219.

[18] J. Myung and W. Lee, "Adaptive splitting protocols for RFID tag collision arbitration," in *Proc. ACM MobiHoc*, 2006, pp. 202–213.

[19] L. M. Ni, Y. Liu, Y. C. Lau, and A. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, 2004.

[20] C. Qian, H. Ngan, and Y. Liu, "Cardinality estimation for large-scale RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1441–1454, Sep. 2011.

[21] Y. Qiao, S. Chen, T. Li, and S. Chen, "Energy-efficient polling protocols in RFID systems," in *Proc. ACM MobiHoc*, 2011, Article no. 25.

[22] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, "A general model of wireless interference," in *Proc. ACM MobiCom*, 2007, pp. 171–182.

[23] L. G. Roberts, "Aloha packet system with and without slots and capture," *Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28–42, 1975.

[24] C. C. Tan, B. Sheng, and Q. Li, "Secure and serverless RFID authentication and search protocols," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1400–1407, Apr. 2008.

[25] C. C. Tan, B. Sheng, and Q. Li, "How to monitor for missing RFID tags," in *Proc. IEEE ICDCS*, 2008, pp. 295–302.

[26] S. Tang, J. Yuan, X.-Y. Li, G. Chen, Y. Liu, and J. Zhao, "RASPberry: A stable reader activation scheduling protocol in multi-reader RFID systems," in *Proc. IEEE ICNP*, 2009, pp. 304–313.

[27] X. Xu, L. Gu, J. Wang, and G. Xing, "Negotiate power and performance in the reality of RFID systems," in *Proc. IEEE PerCom*, 2010, pp. 88–97.

[28] L. Yang, J. Han, Y. Qi, and Y. Liu, "Identification-free batch authentication for RFID tags," in *Proc. IEEE ICNP*, 2010, pp. 154–163.

[29] L. Yang, J. Han, Y. Qi, C. Wang, T. Gu, and Y. Liu, "Season: Shelving interference and joint identification in large-scale RFID systems," in *Proc. IEEE INFOCOM*, 2011, pp. 3092–3100.

[30] D. Yeager, A. Sample, and J. Smith, "WISP: A passively powered UHF RFID tag with sensing and computation," in *RFID Handbook: Applications, Technology, Security, and Privacy*, S. Ahson and M. Ilyas, Eds. Boca Raton, FL: CRC Press, Mar. 2008.

[31] D. Zhang, J. Zhou, M. Guo, J. Cao, and T. Li, "TASA: Tag-free activity sensing using RFID tag arrays," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 4, pp. 558–570, 2011.

[32] R. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast identification of the missing tags in a large RFID system," in *Proc. IEEE SECON*, 2011, pp. 278–286.

[33] Y. Zheng, M. Li, and C. Qian, "PET: Probabilistic estimating tree for large-scale RFID estimation," in *Proc. IEEE ICDCS*, 2011, pp. 37–46.

[34] Z. Zhou, H. Gupta, S. R. Das, and X. Zhu, "Slotted scheduled tag access in multi-reader RFID systems," in *Proc. IEEE ICNP*, 2007, pp. 61–70.

[35] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *Proc. IEEE Symp. Security Privacy*, 2008, pp. 129–142.

[36] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats: Non-invasive security for implanted medical devices," in *Proc. ACM SIGCOMM*, 2011, pp. 2–13.

[37] H. Han, B. Sheng, C. C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID tags efficiently and anonymously," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[38] V. Namboodiri and L. Gao, "Energy-aware tag anti-collision protocols for RFID systems," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 33–59, Jan. 2010.

[39] L. Pan and H. Wu, "Smart trend-traversal protocol for RFID tag arbitration," *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3565–3569, Nov. 2011.

[40] M. Buettner, B. Greenstein, and D. Wetherall, "Dewdrop: An energy-aware runtime for computational RFID," in *Proc. USENIX NSDI*, 2011, pp. 197–210.

[41] B. Ransford, J. Sorber, and K. Fu, "Mementos: System support for long-running computation on RFID-scale devices," in *Proc. ASPLOS*, 2011, pp. 159–170.

[42] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, "Efficient and reliable low-power backscatter networks," in *Proc. ACM SIGCOMM*, 2012, pp. 61–72.

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

His research interests include distributed systems and pervasive computing.

Mr. Zheng is a student member of the Association for Computing Machinery (ACM).

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

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

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