A Time-efficient Information Collection Protocol for Large-scale RFID Systems

Hao Yue[†], Chi Zhang[†], Miao Pan[†], Yuguang Fang[†] and Shigang Chen[§]
[†]Department of Electrical and Computer Engineering, University of Florida
[§]Department of Computer & Information Science & Engineering, University of Florida
Email: {hyue@, zhangchi@, miaopan@, fang@ece., sgchen@cise.}ufl.edu

***Abstract*—Sensor-enabled RFID technology has generated a lot of interest from industries lately. Integrated with miniaturized sensors, RFID tags could provide not only the IDs but also valuable real-time information about the state of the corresponding objects or the surrounding environment, which is beneficial to many practical applications, such as warehouse management and inventory control. In this paper, we study the problem on how to design efficient protocols to collect such sensor information from numerous tags in a large-scale RFID system with a number of readers deployed. Different from information collection in the small RFID system covered by only one reader, in the multi-reader scenario, each reader has to first find out which tags located in its interrogation region in order to read information from them. We start with two categories of warm-up solutions that are directly extended from the existing information collection protocols for single-reader RFID systems, and show that all of them do not work well for the multi-reader information collection problem due to their inefficiency of identifying the interrogated tags. Then, we propose a novel solution, called the Bloom filter based Information Collection protocol (BIC). In BIC, the interrogated tag identification can be efficiently achieved with a distributively constructed Bloom filter, which significantly reduces the communication overhead and thus the protocol execution time. Extensive simulations show that BIC performs better than all the warm-up solutions and its execution time is within 3 times of the lower bound.**

I. Introduction

Radio Frequency IDentification (RFID) systems [1] have been deployed for varieties of applications, such as warehouse management, inventory control and object tracking. An RFID system typically consists of one or several readers and a large number of tags. Each tag has a unique identification (ID) number and is attached to a physical object. The readers can recognize or track the object by communicating with the corresponding tag. Practically, the operational distance of RFID tags is very limited. For the widely used passive tags, it is only several feet. Even for the active tags which enjoy much richer on-tag resources, the reading range is just on the order of 100 feet [2]. Therefore, in large-scale RFID deployments, such as a warehouse or a supermarket, we always need to install multiple readers to ensure the coverage of the entire region.

Recently, sensor-enabled RFID technology has generated a lot of interest from industries [2]. Integrated with miniaturized sensors, each tag could not only provide its ID, but also report some real-time information about the conditions of the surrounding environment or the state of the object [3]–[5]. More importantly, the identification function of RFID systems facilitates the connection of the reported information with the specific object, which is desirable and beneficial to many practical applications. For example, consider a large chilled food storage facility, where sensor-enabled RFID tags are attached to the food items. A collection of readers are deployed and periodically read the sensor-produced temperature information from the tags. If abnormal temperature readings are discovered, the readers can effectively identify the corresponding items and alert the workers to carry out an inspection on them, which helps ensure the quality of the food.

Apparently, how to efficiently collect such information is a critical problem for the sensor-enabled RFID systems. In this paper, we study the problem in the multi-reader environment. Time efficiency is the most important performance criterion for the solutions to this information collection problem. The reason is that an RFID system always has many functions to be frequently performed, such as counting the number of objects placed in a certain area, identifying the missing products and so on. If the execution time is long, the information collection operation may interfere with other scheduled routine tasks of the RFID system, which makes it impossible to be executed. Moreover, short execution time could enable periodic information collection and thus achieve nearly real-time monitoring.

Much existing research on RFID technologies concentrates on the design of ID-collection protocols to read the IDs from a large number of tags with minimum execution time. In recent years, some research interest has been shifted to new functions of RFID systems, such as cardinality estimation and missing tag detection. Similar problems on information collection have been investigated previously by Chen et al. [6] and Qiao et al. [7]. However, the schemes they proposed target at single-reader RFID systems and hardly work well when directly applied to multi-reader RFID deployments (See Section II and IV for more details). To the best of our knowledge, this paper is the first work to comprehensively study the problem on time-efficient information collection protocol design for multi-reader RFID systems.

In this paper, we first examine two categories of solutions to the multi-reader information collection problem, simply called warm-up solutions, which are directly extended from the existing information collection protocols for single-reader RFID systems. We show that all of them are not efficient in

This work was partially supported by the U.S. National Science Foundation under Grants CNS-0916391 and CPS-0931969.

terms of the execution time due to the significant overhead for a reader to identify the tags located in its interrogation region. To reduce the overhead on such identification, we propose a novel solution, called the Bloom filter based Information Collection protocol (BIC), in which a Bloom filter is distributively constructed and transmitted to the reader for efficiently identifying the interrogated tags. Then, the reader uses a hash function to allocate a unique time slot to each interrogated tag for sensor information transmission and minimizes collisions. In this way, the protocol execution time is drastically reduced. Through extensive simulations and performance analysis, we demonstrate that BIC outperforms all the warm-up solutions and the execution time is within 3 times of the lower bound.

The rest of this paper is organized as follows. The related work is reviewed in Section II. Section III introduces the system model, formally defines the problem and gives a lower bound on the protocol execution time as performance benchmark. Warm-up solutions are described and analyzed in Section IV. In Section V, we propose our scheme BIC and elaborate the design of each component. Some practical issues about the implementation of BIC are discussed in Section VI. We conduct simulations and evaluate the performance of BIC in Section VII. Finally, we draw the concluding remarks in Section VIII.

II. Related Work

In the literature, a large body of research has been conducted on various issues in RFID systems. Much prior work concentrates on the design of ID-collection protocols, which read the IDs from all the tags in a single-reader RFID system. The existing ID-collection protocols can be classified into three broad categories: ALOHA-based [8]–[12], tree-based [13], [14] and hybrid [15]. In [16], Yang et al. investigate the ID-collection problem in a multi-reader environment, in which not only the tag transmission collisions but also the reader transmission collisions are considered.

Recently, some research interest has been shifted to new functions of RFID systems. In [17]–[21], a number of novel estimators are designed for fast and accurately estimating the number of distinct tags placed in a given region. Li et al. in [22] and Zhang et al. in [23] address the problem of exactly identifying the IDs of the missing tags. The security and privacy issues of RFID systems are discussed in [24]–[26].

The studies that are most related to our work are [6] and [7]. In [6], Chen et al. design two protocols, called Single-hash Information Collection protocol (SIC) and Multi-hash Information Collection protocol (MIC), to read sensor-produced data from a large number of tags with optimal execution time. In [7], Qiao et al. investigate the information collection problem from the aspect of energy efficiency. The Tag-Ordering Polling Protocol (TOP) and the enhanced version are proposed for a reader to collect sensor information from a subset of tags in the system with minimum energy consumption. As we mentioned in Section I, their schemes are primarily designed for single-reader RFID systems, and assume the reader has already known the IDs of the tags from which it collects information. But in a large-scale RFID system with multiple readers, due to the limited interrogation region and mobility of the tags, such knowledge is not available for each RFID reader. Therefore, their protocols cannot effectively address the information collection problem in multi-reader deployments. As far as we know, how to design time-efficient information collection protocols for multi-reader RFID systems is still under exploration.

III. Preliminaries

A. System Model

Consider a large-scale RFID system with multiple readers and numerous tags. Every tag has a unique ID and is integrated with a sensor to monitor some physical parameters. The readers are statically deployed. Each of them is associated with an interrogation region, within which the reader can communicate with the tags. In this paper, these tags are also called the *interrogated tags* of the reader.

We assume that all the readers are connected to a database that stores the IDs of all the tags present in the system. The knowledge of the IDs can be obtained either by regularly updating the database when objects are moved into or out of the system or by executing an ID-collection protocol like [16]. Considering that the objects with RFID tags may proactively or passively move around in the system, the distribution of the tags will change over time and we make a practical assumption that each reader does not know which tags are located in its interrogation region. Since the running time of the information collection protocols is relatively small, the distribution of the tags is likely to be stable during the protocol execution. Even if there are some tags entering into or departing from the interrogation region of a reader during information collection, the reader can simply ignore them and start to read information from the updated set of interrogated tags in the next execution of the protocol.

Communications between RFID readers and tags are time-slotted, which follow the *Reader-Talks-First* protocol [27]: A reader initializes each round of communication by issuing a request message in a time slot and then several tags respond during a subsequent slotted time frame. The clocks of the tags are well synchronized via the signal received from the reader. Data synchronization is also an important problem for RFID systems [28], [29], but it is outside the scope of this paper.

B. Problem Definition

The problem is to design protocols for a reader to periodically collect sensor information from the interrogated tags with minimum execution time in a multi-reader RFID system. In the rest of the paper, it is also termed as the multi-reader information collection problem for simplicity. The solutions to this problem are called multi-reader information collection protocols. Note that the information referred here includes not only the set of sensor readings but also the mapping from each reading to the tag where the reading takes place so that the sensor data can be accurately associated with the corresponding object.

The multi-reader information collection problem is quite different from the one studied before in the single-reader

environment [6], [7]: In RFID systems where only one reader is deployed, all the tags will transmit information to the same reader. But in a large-scale RFID system with multiple readers installed, the tags located at different geographical positions may communicate with different readers. Due to the limited area of the interrogation region and the mobility of the tags, although each reader could attain the IDs of all the tags from the database, it still has no knowledge about which tags are present in its interrogation region and will report the sensor information to it. Thus, the reader needs to first identify the interrogated tags in order to collect information from them, which makes the multi-reader information collection problem much more complicated and challenging to be addressed.

In this paper, as the first step, we do not consider the interference among the adjacent readers, and only concentrate on information collection between one reader and its interrogated tags. Many scheduling strategies [16], [30], [31] have been proposed to deal with the reader transmission collisions, which can be easily integrated into our protocols to achieve efficient information collection among multiple readers.

C. Performance Lower Bound

Let $\mathcal{M}$ denote the set of tags in the RFID system and $m = |\mathcal{M}|$. Let $\mathcal{N}$ denote the set of tags located within the interrogation region of a reader, and $n = |\mathcal{N}|$. Obviously, we have $\mathcal{N} \subseteq \mathcal{M}$ and $n \leq m$. The ratio of n to m is represented by ρ, i.e., $\rho = n/m$. Assume that all the sensor information contains the same number of bits, which is denoted by l. Let τ_{inf} be the length of a time slot for a tag to transmit the information. Then, a lower bound on the execution time of any protocol for the reader to collect information from all the interrogated tags in $\mathcal{N}$ is $n \times \tau_{inf}$. This lower bound can never be achieved because it takes additional time for the reader to transmit control messages in order to identify the interrogated tags and coordinate their transmissions against collisions. However, it offers a benchmark to evaluate the performance of the feasible solutions.

IV. Warm-up Solutions

In this section, we analyze the solutions that are directly extended from the existing information collection protocols for the single-reader RFID systems, which are simply called warm-up solutions. We start with three typical protocols, and extend them into two categories of multi-reader information collection protocols. We show that all of them cannot efficiently solve the multi-reader information collection problem and discuss the reasons. The analysis of these warm-up solutions provides some insights into the design of the information collection protocols for the multi-reader RFID systems, which motivates the novel scheme we propose in the next section.

A. Polling Based Information Collection Protocol

The first basic protocol is called the *Polling based Information Collection protocol* (PIC). PIC consists of multiple rounds. In each round, the reader broadcasts an ID, and waits for the response from the corresponding tag. Each interrogated tag keeps listening to the communication channel until its own ID is received. Then, the tag transmits its information to the reader and does not participate in the remaining rounds. Since the reader has no knowledge about which tags located in its interrogation region, it must send out all the IDs in $\mathcal{M}$ to find out the interrogated tags and collect information from them. If it does not detect any response for a certain period of time after broadcasting an ID, which indicates that the corresponding tag is not in the interrogation region, the reader will immediately terminate the current round and start a new one.

From the above description, the total execution time of PIC can be calculated as $m \times t_{id} + (m - n) \times \tau_{det} + n \times \tau_{inf}$. Here, t_{id} represents the length of a time slot which allows the transmission of a tag ID from the reader and τ_{det} is the minimum required detection time for the reader to determine the existence of a transmission on the communication channel.

B. Framed Slotted ALOHA Based Information Collection Protocol

The ID-collection protocols can be used for information collection if each tag piggybacks the sensor information when it transmits the ID to the reader. We take the *framed slotted ALOHA based Information Collection protocol* (AIC) for example. In AIC, the reader first sends out a request message to all the interrogated tags, which specifies the number of time slots contained in the following frame. Each tag individually and randomly selects one slot to transmit both its ID and the sensor information to the reader. If there is only one tag replying in a time slot, the reader can successfully receive the information. If multiple tags respond simultaneously, a collision may occur at the reader. In that case, the involved tags will be acknowledged to restart during the next time frame. The similar process repeats until all the tags in $\mathcal{N}$ report their information to the reader.

It has been proved that the execution time of AIC for the reader to collect information from n tags is about $e \times n \times \tau$ [6], [32], where e is the natural constant and τ is the length of a time slot during which a tag is able to transmit both the ID and the sensor information to the reader.

C. Multi-hash Information Collection Protocol

In [6], Chen et al. present the *Single-hash* and *Multi-hash Information Collection protocol* (SIC and MIC) for the single-reader RFID systems, which could also be adopted to address the multi-reader information collection problem. Here, we only consider MIC because it is the enhanced version of SIC and has better performance.

MIC is executed phase by phase. In every phase, the reader uses s hash functions to map the tags to a number of time slots in a frame. Only the slots that have a one-to-one mapping to the tags are assigned by the reader for information transmission. Other slots are just wasted to avoid collisions. The slot assignment is broadcasted by the reader at the beginning of each phase, according to which the interrogated tags sequentially report their information during the following slotted time frame. Similar to PIC, since the reader does not know the interrogated tag set $\mathcal{N}$, it has to assign one time slot to every tag in $\mathcal{M}$. If the time slot allocated to a tag turns

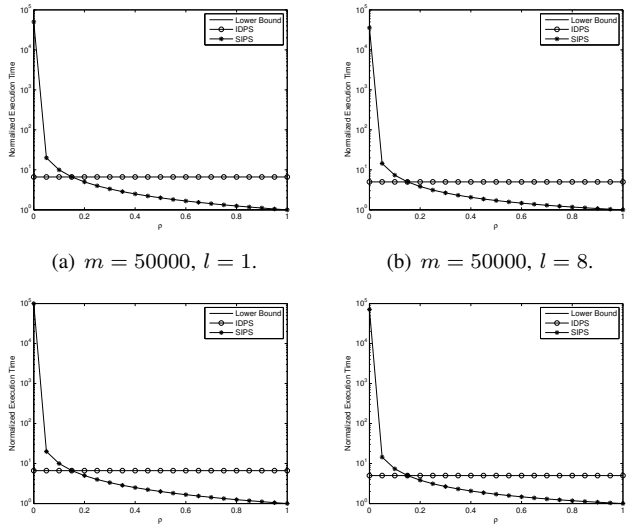
out to be empty, the tag is believed not in the interrogation region. The protocol will terminate after all the tags in $\mathcal{M}$ are assigned time slots to transmit their information.

The expected execution time of MIC can be expressed as $\frac{m}{32P_s} \times t_{id} + \frac{m}{P_s} \times \tau_{inf}$, where P_s denotes the probability that a tag is assigned one time slot to transmit its information in each phase if s hash functions are used [6]. In practice, $s = 7$ is sufficient. In this case, we have $P_7 = 86.1\%$ and the execution time is about $0.036 \times m \times t_{id} + 1.16 \times m \times \tau_{inf}$.

D. Performance Analysis

Various multi-reader information collection protocols can be designed based on the existing schemes for single-reader RFID systems. According to the way the reader identifies the interrogated tags, all of them can be classified into two categories as follows: The first category is *ID-collection based protocols* (IDPS), in which each interrogated tag piggybacks the information to its ID and transmits them together to the reader. When the reader successfully receives a piece of information, it also has the ID of the tag where the information is generated. Therefore, in IDPS the reader is able to identify the interrogated tags without the set $\mathcal{M}$. Obviously, the lower bound on the execution time of IDPS is equal to $n \times \tau$, which is the aggregation time for all the interrogated tags to report their IDs and information to the reader.

The second category is called *sequential identification based protocols* (SIPS), in which the reader finds out the interrogated tag set $\mathcal{N}$ by sequentially examining the existence of each tag in $\mathcal{M}$ within the interrogation region. PIC and MIC are two examples of SIPS. When SIPS are executed, every interrogated tag needs at least τ_{inf} to show its presence to the reader and report the information. For one tag in $\mathcal{M} \backslash \mathcal{N}$, it takes the reader at least τ_{det} to verify its absence. Thus, the lower bound on the execution time of SIPS is $(m - n) \times \tau_{det} + n \times \tau_{inf}$.



(a) $m = 50000$, $l = 1$. (b) $m = 50000$, $l = 8$.

(c) $m = 100000$, $l = 1$. (d) $m = 100000$, $l = 8$.

Fig. 1. Performance comparison of different multi-reader information collection protocols.

To evaluate the performance of the two types of multi-reader information collection protocols, we compare the lower bounds on the execution time of IDPS and SIPS with the lower bound $n \times \tau_{inf}$. From the results shown in Fig. 1, two observations can be made: First, the lower bound on the execution time of SIPS approaches the lower bound $n \times \tau_{inf}$ when ρ is close to 1. However, the performance of SIPS degrades quickly as ρ decreases to 0. Especially, when $\rho < 0.05$, the execution time of SIPS could be more than 10 times of the lower bound. Second, there is a wide constant gap between the execution time lower bound of IDPS and the lower bound $n \times \tau_{inf}$, which can be as large as seven times shown in Fig. 1(a) and Fig. 1(c). These observations indicate that *all the protocols which belong to these two categories cannot efficiently solve the multi-reader information collection problem*. There still exists large potential space for performance improvement.

E. Insights

As we discussed in Section III, a multi-reader information collection protocol should achieve both the interrogated tag identification and the sensor information collection. Hence, the total execution time of any multi-reader information collection protocol can be divided into two parts: the time for the reader to find out all the interrogated tags in $\mathcal{N}$ and the time for the tags to report their information to the reader. Both IDPS and SIPS are not efficient due to the significant time overhead on identifying the interrogated tags, which are at least $(m-n) \times \tau_{det}$ and $n \times \tau_{id}$, respectively. Here, τ_{id} denotes the length of a time slot for a tag to transmit its ID. Therefore, in order to minimize the overall protocol execution time, we need to explore new technologies for the reader to efficiently identify the interrogated tags.

V. Bloom Filter Based Information Collection Protocol

Bloom filter is a simple space-efficient probabilistic data structure for representing a set and supporting membership queries [33], [34]. Hence, if the set $\mathcal{N}$ can be transmitted to the reader in the form of a Bloom filter, the overhead for interrogated tag identification could be significantly reduced and thus the execution time of the information collection protocol. But the challenge is how to construct the Bloom filter and transmit the filtered data to the reader in the case that neither the reader nor the interrogated tags know $\mathcal{N}$. In this section, we propose the *Bloom filter based Information Collection protocol* (BIC), which takes advantage of the synchronized physical layer transmissions to distributively construct the desired Bloom filter and efficiently identify the interrogated tags.

A. Interrogated Tag Identification

To identify the interrogated tags with a Bloom filter, the reader first broadcasts a request message, which contains two parameters w and k. Here, w is the size of the Bloom filter and k is the number of independent hash functions used to construct the Bloom filter. How to choose the values of w and k will be explained later.

Let $h_1, h_2, \ldots, h_k$ denote the k hash functions, each with range $\{0, 1, \ldots, w-1\}$. Upon receiving the request message,

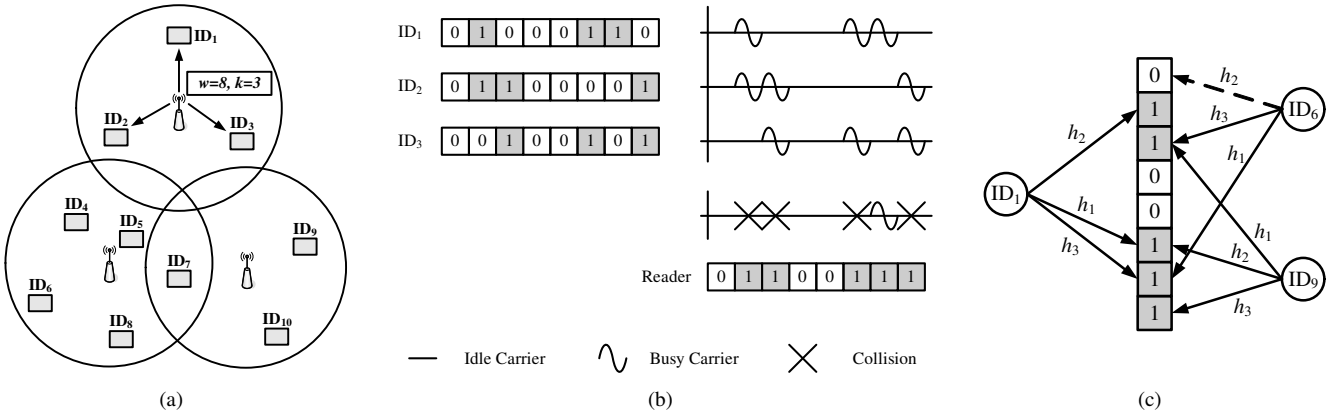


Fig. 2. A simple example to illustrate the procedures of the interrogated tag identification with a Bloom filter.

every interrogated tag in $\mathcal{N}$ generates an array of w bits, all of which are initialized to 0. With the k hash functions, the tag pseudo-randomly maps its unique ID to k bits at positions $h_1(ID), h_2(ID), \ldots, h_k(ID)$ in the array, and sets them to 1. The resulting array is called a *Bloom filter basis*.

All the interrogated tags transmit their respective Bloom filter basis simultaneously. In the physical layer, a binary '0' is represented by an idle carrier and a binary '1' is represented by a busy carrier [22]. For each bit received by the reader, if the channel is idle, the bit is set to 0. If the channel is busy, which indicates that at least one tag transmits the busy carrier for this bit, the reader sets it to 1. After the transmissions of all the Bloom filter bases, the reader can generate a new w-bit array $\mathcal{B}$, which turns out to be the Bloom filter constructed based on $\mathcal{N}$.

The reader uses the Bloom filter $\mathcal{B}$ to find out the interrogated tag set $\mathcal{N}$ from $\mathcal{M}$. For each ID in $\mathcal{M}$, the bits at positions $h_1(ID), h_2(ID), \ldots, h_k(ID)$ in $\mathcal{B}$ are examined. If any of them is 0, the corresponding tag is certainly not in $\mathcal{N}$. Otherwise, the tag is considered to be included in $\mathcal{N}$. Finally, the reader can retrieve a tag set from $\mathcal{B}$, which is denoted as $\tilde{\mathcal{N}}$. According to the property of Bloom filter, false negatives are impossible, which means any element in $\mathcal{N}$ will be identified into $\tilde{\mathcal{N}}$. But false positives may occur with a certain probability. In the case of false positives, a tag which in fact does not belong to $\mathcal{N}$ is identified into $\tilde{\mathcal{N}}$ because all the bits it is mapped to in $\mathcal{B}$ are set to 1 by other IDs in $\mathcal{N}$. Hence, the set $\tilde{\mathcal{N}}$ satisfies $\mathcal{N} \subseteq \tilde{\mathcal{N}} \subseteq \mathcal{M}$. The expected cardinality of $\tilde{\mathcal{N}}$ is equal to $n + p \times (m - n)$, where p denotes the probability of the false positives of the Bloom filter $\mathcal{B}$. The tags in $\tilde{\mathcal{N}} \backslash \mathcal{N}$ are called the *false positive identified tags*.

For illustrative purpose, we take a simple example to show the procedures of the interrogated tag identification as well as demonstrate the correctness of the Bloom filter construction. Let us examine a toy RFID system with 10 tags, i.e., $\mathcal{M} = \{ID_1, ID_2, \ldots, ID_{10}\}$. For one specific reader, the tags ID_1, ID_2 and ID_3 are located within its interrogation region, i.e., $\mathcal{N} = \{ID_1, ID_2, ID_3\}$. As shown in Fig. 2(a), to identify the interrogated tags, the reader first sends out a request message, which contains the values of the parameters w and k. Here, we assume $w = 8$ and $k = 3$. When receiving the request message, each tag in $\mathcal{N}$ individually generates an 8-bit Bloom filter basis with its ID and the three hash functions h_1, h_2 and h_3. Suppose the Bloom filter bases of the tags ID_1, ID_2 and ID_3 are 01000110, 01100001 and 00100101, respectively. The three tags concurrently transmit their Bloom filter bases to the reader. The reader interprets each bit received according to the state of the channel, which is depicted in Fig 2(b). If the channel is idle, such as the first bit, the reader sets it to 0. If the reader detects that the channel is busy, such as the second bit, it is set to 1. After receiving all the Bloom filter bases, the reader attains a bit array 01100111, which is exactly the same as the Bloom filter constructed from the set $\mathcal{N}$. Then, it checks the elements in $\mathcal{M}$ one by one to find out the set $\mathcal{N}$ with the Bloom filter. For example, in Fig. 2(c), since the bits at positions $h_1(ID_1)$, $h_2(ID_1)$ and $h_3(ID_1)$ are all equal to 1, the tag ID_1 is believed to be in $\mathcal{N}$. But the tag ID_6 is not included in the set $\mathcal{N}$ due to the fact that $h_2(ID_6) = 0$. For the tag ID_9, it is actually not in $\mathcal{N}$ but it is able to pass the test. Thus, it will be falsely identified as an interrogated tag.

Now, we show how to determine the values of parameters w and k. The false positive probability of the Bloom filter constructed in BIC can be represented as follows:

$$p = \left[1 - \left(1 - \frac{1}{w}\right)^{kn}\right]^k \approx \left(1 - e^{-\frac{kn}{w}}\right)^k. \tag{1}$$

Given the number of interrogated tags n and the probability of false positives p, the length of the Bloom filter w can be calculated as

$$w = -\frac{n \times \ln p}{(\ln 2)^2}, \tag{2}$$

and the optimal value of k is

$$k = \frac{w}{n} \ln 2. \tag{3}$$

B. Information Collection

After the reader attains the set $\tilde{\mathcal{N}}$, it can start to collect information from the interrogated tags. Information collection consists of several rounds. Each round begins with a request

message sent out from the reader, followed by a slotted time frame during which some tags are scheduled to report their information. The reader uses a so-called *allocation vector* to coordinate the tags' transmissions, which is denoted as $\mathcal{V}$. The length of the allocation vector $\mathcal{V}$ exploited in each round is equal to the number of the tags in $\tilde{\mathcal{N}}$ from which the reader has not yet received the sensor information. In the rest of the paper, these tags are called *uncollected tags* for simplicity. The reader picks a random number r and uses a hash function h to map the ID of every uncollected tag to a bit in $\mathcal{V}$, which is called the *indicator bit* of the uncollected tag. For each bit in $\mathcal{V}$, if there is only one uncollected tag mapped to it, the bit is 1, which means that the tag is allowed to respond its information to the reader during one of the time slots in the following frame. Otherwise, if several uncollected tags are mapped to the same bit, the bit is 0. These tags will keep silent in this round to avoid potential transmission collisions.

At the beginning of a round, the reader first broadcasts the request message to all the tags within its interrogation region, which contains the random number r and the allocation vector $\mathcal{V}$. If the allocation vector $\mathcal{V}$ is too long, the reader could divide it into 96-bit segments and transmit each one of them in a time slot of length t_{id} (See Section VII-A).

When receiving the request message, every uncollected tag inputs its ID and r into the same hash function exploited by the reader, and obtains the position of its indicator bit in the allocation vector $\mathcal{V}$. Then, it examines the corresponding bit. If its value is 0, the tag will delay the information transmission to the next round. If the bit is 1, the tag then calculates how many 1s appear before its indicator bit in $\mathcal{V}$. Since each bit of value 1 in the allocation vector represents a tag that is scheduled to transmit the sensor information to the reader in the following time frame, if there are i 1s preceding its indicator bit, the tag should be the $(i+1)$th responder of the current round to report its information. Then, during the subsequent slotted time frame, it will transmit the information in the $(i+1)$th time slot without collisions.

If a time slot allocated to a tag in $\tilde{\mathcal{N}}$ to report information turns out to be empty, the tag is a false positive identified one and the reader will delete the corresponding ID from $\tilde{\mathcal{N}}$. In this way, at the end of the information collection, the reader will obtain the interrogated tag set $\mathcal{N}$ after it removes all the false positive identified tags from $\tilde{\mathcal{N}}$.

C. Execution Time Analysis

The overall execution time of BIC is equal to the sum of the time taken for the reader to identify the interrogated tags and collect information from them. During the interrogated tag identification, the reader broadcasts one request message, which is followed by the concurrent transmissions of the Bloom filter bases from all the interrogated tags. The time for an interrogated tag to transmit its w-bit Bloom filter basis can be calculated as $w \times \tau_{bit}$, where τ_{bit} is the time for a tag to transmit one bit. From Eqn. (2), it could also be expressed as $-\frac{n \times \ln p}{(\ln 2)^2} \times \tau_{bit}$. Since the request message is very short, its transmission time is not considered. Therefore, the time for identifying the interrogated tags is about $-\frac{n \ln p}{(\ln 2)^2} \times \tau_{bit}$.

The execution time for information collection consists of two parts: the time for the reader to transmit the request messages and the time for the slotted frames. Similar to [6], we can prove that the expected number of indicator bits for each tag is e. Also recall that the expected cardinality of the set $\tilde{\mathcal{N}}$ is $n+p\times(m-n)$. Thus, the total number of bits in all allocation vectors is expected to be $e \times \big[n+p\times(m-n)\big]$, and the expected time for the reader to broadcast all the allocation vectors is about $\frac{e\times[n+p\times(m-n)]}{96} \times t_{id}$. The rest of the request message excluding the allocation vector is very small and the transmission time can be ignored. Since each tag in $\tilde{\mathcal{N}}$ is allocated a unique time slot to report its information to the reader, the total number of time slots in all the frames should be equal to $n + p \times (m-n)$. Then, the overall frame time in all rounds is $\big[n + p \times (m-n)\big] \times \tau_{inf}$. Therefore, the total execution time for information collection is equal to $\frac{e\times[n+p\times(m-n)]}{96} \times t_{id} + \big[n + p \times (m-n)\big] \times \tau_{inf}$.

Based on the above analysis, the expected execution time of BIC, denoted as T, can be computed as follows:

$$T = -\frac{n \times \ln p}{(\ln 2)^2} \times \tau_{bit} + \left(\frac{e \times t_{id}}{96} + \tau_{inf}\right) \times \big[n+p\times(m-n)\big]. \tag{4}$$

VI. Implementation Issues

A. Cardinality Estimation

From Eqn. (2) and (3), to determine the parameters w and k of the Bloom filter, the reader must know the number of the interrogated tags n, which may not be available in some application scenarios. In that case, we need to estimate the cardinality of the interrogated tag population at the beginning of BIC. In the literature, many estimation algorithms [17]–[21] have been designed to quickly and accurately estimate the cardinality of a tag set for RFID systems, which can be used to obtain the approximated value of n for BIC. Since the running time of the estimation algorithms is always negligible compared to the time for identifying the interrogated tags and collecting information from them in BIC, we do not take it into consideration when deriving the expression of the execution time T in Eqn. (4).

B. Hash Function

The issue on how to implement hash functions on RFID tags has been discussed in [6], [22]. Basically, the hash values are derived from a string of pre-stored random bits in each tag, which are generated by an offline random number generator with the ID of the tag as seed. The string is divided into multiple segments. Each segment contains the same number of bits and forms a logical ring. In BIC, when an interrogated tag receives the parameters w and k from the reader, it takes the first k rings to construct the Bloom filter basis. Specifically, the value of $h_i(ID)$ is calculated as the number represented by the bits in the ith ring modulo the length of the Bloom filter w. During information collection, the hash result for the position of the indicator bit can be simply computed as the number represented by the bits after the rth bit in the entire string modulo the frame size.

C. Channel Error

Channel error is another important implementation issue, which may corrupt the data exchange between the reader and the interrogated tags and disturb the normal execution of BIC. For example, a number of bits in the Bloom filter bases may be corrupted during transmissions, which makes false negatives of the Bloom filter also possible. In the case of false negatives, some interrogated tags in $\mathcal{N}$ will not be identified into $\tilde{\mathcal{N}}$ and hence will not be intentionally allocated a time slot by the reader. We call them the *false negative identified tags*. Finally, these tags either have no chance to report their information to the reader, or respond in the time slots that are allocated to other tags and cause collisions. The false negative problem can be easily solved by adding another phase at the end of BIC, in which the reader executes an ID-collection based protocol to read sensor information from the false negative identified tags.

During information collection, to guarantee the correctness of the received information from the interrogated tags, we include 16-bit checksum to the information for error detection. Each segment of 96 bits in the allocation vector also carries 16-bit checksum. In BIC, an interrogated tag that is allowed to report the information in the current round determines its allocated time slot based on all the bits preceding its indicator bit in the allocation vector, which is vulnerable to the channel error because even one bit flip may lead to a wrong decision. To reduce the negative impact of the channel error on the transmission order determination, we add a header into each segment, which records the total number of 1s in the previous segments. When an interrogated tag correctly receives the segment that contains its indicator bit, it could compute its transmission order from the value in the header and the number of 1s appearing before its indicator bit in the current segment, no matter the previous segments are corrupted or not. If the tag finds that the segment containing its indicator bit is corrupted, it will not participate in the remaining rounds to avoid potential transmission collisions. The tag can report the information to the reader during the execution of the ID-collection protocol at the end of BIC.

D. Applications

Generally, BIC can be applied to the cases where the reader lacks the knowledge about the subset of tags under query for information collection, which are not limited to the multi-reader scenario we focus on in this paper. For example, consider a single-reader RFID system with a large number of battery-powered active tags deployed. Each tag is equipped with a sensor, which monitors the residual energy level of the battery in the tag. If the amount of the residual energy is below a certain threshold, the tag will report a piece of information to the reader for battery replacement. Otherwise, it keeps silent to save energy. At any time, which tag needs battery replacement is unknown. Therefore, when the reader initializes to collect information from all the tags, it does not know which tag will respond. In that case, our BIC can help the reader identify the tags with low residual energy and collect information from them in a short time.

In another example, a worker carries a mobile reader and walks around a warehouse to read information from the sensor-augmented tags. Due to the limited operational distance of the tags, the entire region covered by the RFID system is divided into several subregions. Each time the walker stays in one subregion and collects information from the interrogated tags with the reader. Assume the reader has the ID of all the tags in the RFID system but it has no knowledge about the location of each tag, which is true in most applications. Therefore, the reader does not know from which tags it will read the sensor information in every subregion. The BIC could also be used in this scenario to achieve time-efficient information collection.

VII. Performance Evaluation

In this section, we evaluate the performance of BIC. We compare the execution time of BIC with PIC, AIC, MIC as well as the lower bounds on the execution time of IDPS and SIPS, which demonstrates the efficiency of BIC for information collection in multi-reader RFID systems.

A. Simulation Setting

The simulation setting is based on the Philips I-Code specification [27] and the Gen2 standard [35]. Each tag ID is 96 bits long, which contains a 16-bit CRC code. Any two consecutive transmissions are separated by a time interval of $302\mu s$. The transmission rate of the reader is 26.5Kb/s. Thus, the time for the reader to transmit an ID or a segment of allocation vectors is $3927\mu s$ with a time interval included, i.e., $t_{id} = 3927\mu s$. The transmission rate of a tag is 53Kb/s, which is different from that of the reader. It takes $18.88\mu s$ for a tag to transmit one bit, i.e., $\tau_{bit} = 18.88\mu s$. The value of τ_{inf} is calculated as the sum of a time interval and the information transmission time that equals to $18.88\mu s$ multiplied by the length of the information l. For example, if the sensor information is 8 bits long, τ_{inf} is equal to $452\mu s$. Recall that τ_{det} and τ represent the minimum required channel detection time and the length of a time slot for a tag to transmit both the ID and information, respectively. Similarly, we have $\tau_{det} = 321\mu s$ [22] and $\tau = (18.88 \times l + 2114)\mu s$.

In our experiments, the false positive probability of the Bloom filter is set to 1×10^{-4}. Under the same simulation setting, we take the average values of 100 simulation runs as results.

B. Execution Time Comparison under Different Number of Tags m

We first evaluate the performance of BIC under different values of m, which varies from 10000 to 100000. For each value of m, we assume $n = 0.1m$ so that the ratio ρ remains a constant during the set of simulations.

Table I illustrates the execution time of different multi-reader information collection protocols when the information is 1 bit. It is shown that BIC outperforms all the other protocols. For example, when $m = 50000$ and $n = 5000$, it takes the reader about 212.4 seconds to collect all the information with PIC, which is about 133 times of the lower bound, 1.6 seconds. AIC works better than PIC, and it just

needs 29.0 seconds. MIC further reduces the execution time to 25.8 seconds. Our BIC has the minimum execution time of 4.0 seconds, which is only 2.5 times of the lower bound.

TABLE I
EXECUTION TIME COMPARISON (IN SECONDS) WHEN THE SENSOR INFORMATION IS 1 BIT.

(m, n)	PIC	AIC	MIC	IDPS	SIPS	BIC	LB
(10000, 1000)	42.5	5.8	5.2	2.1	3.2	0.8	0.3
(20000, 2000)	85.0	11.6	10.3	4.3	6.4	1.6	0.6
(30000, 3000)	127.4	17.4	15.5	6.4	9.6	2.4	1.0
(40000, 4000)	170.0	23.2	20.6	8.5	12.8	3.2	1.3
(50000, 5000)	212.4	29.0	25.8	10.7	16.0	4.0	1.6
(75000, 7500)	318.6	43.5	38.7	16.0	24.1	6.0	2.4
(100000, 10000)	424.8	57.9	51.5	21.3	32.1	8.0	3.2

TABLE II
EXECUTION TIME COMPARISON (IN SECONDS) WHEN THE SENSOR INFORMATION IS 8 BITS.

(m, n)	PIC	AIC	MIC	IDPS	SIPS	BIC	LB
(10000, 1000)	42.6	6.1	6.7	2.3	3.3	0.9	0.5
(20000, 2000)	85.2	12.3	13.4	4.5	6.7	1.9	0.9
(30000, 3000)	127.8	18.5	20.1	6.8	10.0	2.8	1.4
(40000, 4000)	170.4	24.7	26.8	9.1	13.4	3.7	1.8
(50000, 5000)	213.1	30.8	33.4	11.3	16.7	4.6	2.3
(75000, 7500)	319.6	46.1	50.2	17.0	25.1	7.0	3.4
(100000, 10000)	426.1	61.6	66.9	22.7	33.4	9.3	4.5

TABLE III
EXECUTION TIME COMPARISON (IN SECONDS) WHEN THE SENSOR INFORMATION IS 16 BITS.

(m, n)	PIC	AIC	MIC	IDPS	SIPS	BIC	LB
(10000, 1000)	42.8	6.6	8.4	2.4	3.5	1.1	0.6
(20000, 2000)	85.5	13.1	16.9	4.8	7.0	2.2	1.2
(30000, 3000)	128.3	19.7	25.3	7.2	10.5	3.2	1.8
(40000, 4000)	171.0	26.3	33.8	9.7	14.0	4.3	2.4
(50000, 5000)	213.8	32.8	42.2	12.1	17.5	5.4	3.0
(75000, 7500)	320.7	49.3	63.3	18.1	26.2	8.1	4.5
(100000, 10000)	427.6	65.7	84.4	24.2	34.9	10.8	6.0

Table II and III show the experiment results when the length of the information is 8 bits and 16 bits, respectively. We observe that BIC still achieves the highest time efficiency among all the protocols. For example, in the case that $m = 30000$, $n = 3000$ and the information is 8 bits long, the execution time of BIC is 2.8 seconds, the execution time of MIC is 20.1 seconds, the execution time of AIC is 18.5 seconds and the execution time of PIC is 127.8 seconds. In the case that $m = 75000$, $n = 7500$ and the information is 16 bits long, the execution time of BIC is 8.1 seconds, the execution time of MIC is 63.3 seconds, the execution time of AIC is 49.3 seconds and the execution time of PIC is 320.7 seconds.

C. Execution Time Comparison under Different Ratio ρ

In this subsection, we compare the performance of different multi-reader information collection protocols with respect to the ratio ρ. We set $m = 100000$ and make n change from 2000 to 25000. Table IV, V and VI illustrate the execution time comparison with different information length.

From the results shown in Table IV, V and VI, two observations can be made. First, the execution time of SIPS is not affected by the ratio ρ, which only depends on the number of tags in the whole system and the size of sensor information. For other protocols, the execution time increases as ρ increases. Second, compared to PIC, AIC and MIC, BIC achieves the best performance. For instance, in Table IV where each piece of information contains 1 bit, when $m = 100000$ and $n = 3000$, the execution time of BIC is only 2.4 seconds, which is 14% of the time required by AIC, 5% of the time required by MIC and 0.6% of the time required by PIC. In Table V where each piece of information contains 8 bits, when $m = 100000$ and $n = 7500$, the execution time of BIC is only 7.0 seconds, which is 15% of the time required by AIC, 10% of the time required by MIC and 2% of the time required by PIC. In Table VI where each piece of information contains 16 bits, when $m = 100000$ and $n = 25000$, the execution time of BIC is only 26.9 seconds, which is 16% of the time required by AIC, 32% of the time required by MIC and 6% of the time required by PIC.

TABLE IV
EXECUTION TIME COMPARISON (IN SECONDS) WHEN THE SENSOR INFORMATION IS 1 BIT.

(m, n)	PIC	AIC	MIC	IDPS	SIPS	BIC	LB
(100000, 2000)	424.8	11.6	51.5	4.3	32.1	1.6	0.6
(100000, 3000)	424.8	17.4	51.5	6.4	32.1	2.4	1.0
(100000, 5000)	424.8	29.0	51.5	10.7	32.1	4.0	1.6
(100000, 7500)	424.8	43.4	51.5	16.0	32.1	6.0	2.4
(100000, 10000)	424.8	58.0	51.5	21.3	32.1	8.0	3.2
(100000, 25000)	424.8	144.9	51.5	53.3	32.1	20.0	8.0

TABLE V
EXECUTION TIME COMPARISON (IN SECONDS) WHEN THE SENSOR INFORMATION IS 8 BITS.

(m, n)	PIC	AIC	MIC	IDPS	SIPS	BIC	LB
(100000, 2000)	425.0	12.4	66.9	4.5	32.4	1.9	0.9
(100000, 3000)	425.2	18.4	66.9	6.8	32.5	2.8	1.4
(100000, 5000)	425.4	30.8	66.9	11.3	32.7	4.6	2.3
(100000, 7500)	425.8	46.2	66.9	17.0	33.1	7.0	3.4
(100000, 10000)	426.1	61.6	66.9	22.7	33.4	9.3	4.5
(100000, 25000)	428.1	154.1	66.9	56.6	35.4	23.2	11.3

TABLE VI
EXECUTION TIME COMPARISON (IN SECONDS) WHEN THE SENSOR INFORMATION IS 16 BITS.

(m, n)	PIC	AIC	MIC	IDPS	SIPS	BIC	LB
(100000, 2000)	425.4	13.1	84.5	4.8	32.7	2.2	1.2
(100000, 3000)	425.6	19.6	84.5	7.2	32.9	3.2	1.8
(100000, 5000)	426.2	32.9	84.5	12.1	33.5	5.4	3.0
(100000, 7500)	426.9	49.3	84.5	18.1	34.2	8.1	4.5
(100000, 10000)	427.6	65.7	84.5	24.2	34.9	10.8	6.0
(100000, 25000)	431.9	164.2	84.5	60.4	39.2	26.9	15.1

VIII. CONCLUSION

This paper studies the problem of efficiently collecting sensor information from the tags in a large-scale RFID system

where a number of readers are deployed. Different from the information collection problem in single-reader RFID deployments, in the multi-reader environment, each reader needs to first find out all the interrogated tags before reading information from them. Warm-up solutions that are directly extended from existing single-reader information collection protocols are not efficient in terms of the execution time due to high overhead for identifying the interrogated tags. We design a Bloom filter based information collection protocol (BIC). A Bloom filter representing the interrogated tag set is distributively constructed and transmitted to the reader, which significantly reduces the interrogated tag identification overhead and improves the performance of the multi-reader information collection protocol. Extensive simulations are conducted to demonstrate the time efficiency of BIC. The results show that BIC outperforms all the warm-up solutions and the execution time is within 3 times of the lower bound.

REFERENCES

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-field Communication, Third Edition*. John Wiley & Sons, 2010.

[2] A. Ruhanen, M. Hanhikorpi, F. Bertuccelli, A. Colonna, W. Malik, D. Ranasinghe, T. S. Lopez, N. Yan, and M. Tavilampi, *Sensor-enabled RFID Tag Handbook*. BRIDGE, IST-2005-033546, 2008.

[3] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," *Wireless Networks*, vol. 10, no. 6, pp. 701–710, November 2004.

[4] R. Want, "Enabling Ubiquitous Sensing with RFID," *IEEE Computer*, vol. 37, no. 4, pp. 84–86, April 2004.

[5] M. Miura, S. Ito, R. Takatsuka, T. Sugihara, and S. Kunifuji, "An Empirical Study of an RFID Mat Sensor System in a Group Home," *Journal of Networks*, vol. 4, no. 2, pp. 133–139, April 2009.

[6] S. Chen, M. Zhang, and B. Xiao, "Efficient Information Collection Protocols for Sensor-augmented RFID Networks," in *Proc. of InfoCom 2011*, Shanghai, China, April 2011.

[7] Y. Qiao, S. Chen, T. Li, and S. Chen, "Energy-efficient Polling Protocols in RFID Systems," in *Proc. of MobiHoc 2011*, Las Vegas, NV, September 2011.

[8] H. Vogt, "Efficient Object Identification with Passive RFID Tags," in *Proc. of International Conference on Pervasive Computing (Pervasive) 2002*, Zurich, Switzerland, August 2002.

[9] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed ALOHA for Multiple RFID Objects Identification," *IEICE Transactions on Communications*, vol. E88-B, no. 3, pp. 991–999, March 2005.

[10] S. R. Lee, S. D. Joo, and C. W. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification," in *Proc. of MOBIQUITOUS 2005*, San Diego, CA, July 2005.

[11] J. R. Cha and J. H. Kim, "Dynamic Framed Slotted ALOHA Algorithms using Fast Tag Estimation Method for RFID System," in *Proc. of IEEE Consumer Communications and Networking Conference (CCNC) 2006*, Las Vegas, NV, January 2006.

[12] V. Sarangan, M. R. Devarapalli, and S. Radhakrishnan, "A Framework for Fast RFID Tag Reading in Static and Mobile Environments," *Computer Networks (Elsevier) Journal*, vol. 52, no. 5, pp. 1058–1073, April 2008.

[13] J. Myung and W. Lee, "Adaptive Splitting Protocols for RFID Tag Collision Arbitration," in *Proc. of ACM MobiHoc 2006*, Florence, Italy, May 2006.

[14] N. Bhandari, A. Sahoo, and S. Lyer, "Intelligent Query Tree (IQT) Protocol to Improve RFID Tag Read Efficiency," in *Proc. of International Conference in Information Technology (ICIT) 2006*, Orissa, India, December 2006.

[15] T. F. L. Porta, G. Maselli, and C. Petrioli, "Anticollision Protocols for Single-Reader RFID Systems: Temporal Analysis and Optimization," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 267–279, February 2011.

[16] 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. of InfoCom 2011*, Shanghai, China, April 2011.

[17] M. Kodialam and T. Nandagopal, "Fast and Reliable Estimation Schemes in RFID Systems," in *Proc. of ACM MobiCom 2006*, Los Angeles, CA, September 2006.

[18] M. Kodialam, T. Nandagopal, and W. Lau, "Anonymous Tracking Using RFID Tags," in *Proc. of InfoCom 2007*, Anchorage, Alaska, May 2007.

[19] T. Li, S. Wu, S. Chen, and M. Yang, "Energy Efficient Algorithms for the RFID Estinamtion Problem," in *Proc. of InfoCom 2010*, San Diego, CA, March 2010.

[20] H. Han, B. Sheng, C. Tan, Q. Li, W. Mao, and S. Lu, "Counting RFID Tags Efficiently and Anonymously," in *Proc. of InfoCom 2010*, San Diego, CA, March 2010.

[21] C. Qian, H. Ngan, and L. Hu, "Cardinality Estimation for Large-scale RFID Systems," in *Proc. of PerCom 2008*, Hong Kong, March 2008.

[22] T. Li, S. Chen, and Y. Ling, "Identifying the Missing Tags in a Large RFID System," in *Proc. of ACM MobiHoc 2010*, Chicago, IL, September 2010.

[23] R. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast Identification of the Missing Tags in a Large-scale RFID System," in *Proc. of SECON 2011*, Salt Lake City, Utah, June 2011.

[24] C. Tan, B. Sheng, and Q. Li, "How to Monitor for Missing RFID Tags," in *Proc. of ICDCS 2008*, Beijing, China, June 2008.

[25] L. Lu, J. Han, R. Xiao, and Y. Liu, "ACTION: Breaking the Privacy Barrier for RFID Systems," in *Proc. of InfoCom 2009*, Rio de Janeiro, Brazil, April 2009.

[26] L. Yang, J. Han, Y. Qi, and Y. Liu, "Identification-Free Batch Authentication for RFID Tags," in *Proc. of ICNP 2010*, Kyoto, Japan, October 2010.

[27] Philips Semiconductors, "I-CODE UID Smart Label IC Functional Specification," January, 2004. [Online]. Available: http://www.nxp.com/acrobat_download/other/identification/SL092030.pdf

[28] K. S. Leong, M. L. Ng, A. R. Grasso, and P. H. Cole, "Synchronization of RFID Readers for Dense RFID Reader Environments," in *Proc. of 2006 International Symposium on Applications and Internet Workshops*, Phoenix, AZ, January 2006.

[29] C. Angerer and M. Rupp, "Advanced synchronisation and Decoding in RFID Reader Receivers," in *Proc. of IEEE Radio and Wireless Symposium 2009*, San Diego, CA, January 2009.

[30] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: An Anticollision Algorithm for the Reader Collision Problem," in *Proc. of ICC 2003*, Anchorage, Alaska, May 2003.

[31] Z. Zhou, H. Gupta, S. R. Das, and X. Zhu, "Slotted Scheduled Tag Access in Multi-Reader RFID Systems," in *Proc. of ICNP 2007*, Beijing, China, October 2007.

[32] L. G. Roberts, "ALOHA Packet System with and without Slots and Capture," *ACM SIGCOMM Computer Communications Review*, vol. 5, no. 2, pp. 28–42, April 1975.

[33] B. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, July 1970.

[34] A. Broder and M. Mitzenmacher, "Network Applications of Bloom Filters: A Survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485–509, March 2004.

[35] EPCglobal, "EPC Radio-frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz - 960MHz," October, 2008. [Online]. Available: http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2/uhfc1g2_1_2_0-standard-20080511.pdf