Unknown-Target Information Collection in Sensor-Enabled RFID Systems

Hao Yue, *Student Member, IEEE*, Chi Zhang, *Member, IEEE*, Miao Pan, *Member, IEEE*, Yuguang Fang, *Fellow, IEEE, Member, ACM*, and Shigang Chen, *Senior Member, IEEE, Member, ACM*

***Abstract*—Sensor-enabled radio frequency identification (RFID) technology has generated a lot of interest from industries lately. Integrated with miniaturized sensors, RFID tags can provide not only the IDs, but also valuable real-time information about the state of the objects or their surrounding environment, which can benefit many practical applications, such as warehouse management and inventory control. In this paper, we study the problem of designing efficient protocols for a reader to collect sensor-produced information from unknown target tags in an RFID system with minimum execution time. Different from information collection with all target tags known *a priori*, in the scenarios we consider, the reader has to first find out the target tags in order to read information from them, which makes traditional information collection protocols not efficient any more. We design a Bloom-filter-based information collection protocol (BIC) to address this challenging problem. A Bloom filter is constructed for the reader to efficiently determine the target tags, which significantly reduces the communication and time overhead. We also introduce the allocation vectors to coordinate the transmissions from different tags and minimize collision during information collection. Extensive simulation results demonstrate that our protocol is highly efficient in terms of execution time, and it performs much better than other solutions.**

***Index Terms*—Information collection, radio frequency identification (RFID) systems, time-efficient.**

I. Introduction

RADIO frequency identification (RFID) technologies have been increasingly used in various applications, such as supply chain management, inventory control, and object tracking [1]. An RFID system typically consists of one or several readers and numerous tags. Each tag has a unique ID and is attached to a physical object. The readers communicate with tags wirelessly to recognize or track the corresponding objects. Compared to the traditional barcode systems, RFID systems have many advantages, such as long operational distance and fast identification. It is expected that RFID technologies will be more and more ubiquitously available in the near future.

In the literature, most existing research on RFID technologies concentrates on the design of ID-collection protocols that read the IDs from a large number of tags [2]–[10]. In recent years, some research interest has been shifted to new functionalities of RFID systems, such as cardinality estimation [11]–[17], missing tag detection [18], [19], and tag searching [20].

Recently, sensor-enabled RFID technology has generated a lot of interest from industries [21]. Integrated with miniaturized sensors, an RFID tag can not only provide its ID, but also report real-time information about the state of the object or the conditions of the surrounding environment [22]–[24]. More importantly, the identification function of RFID systems facilitates the connection of the reported information with the specific object, which can benefit many practical applications. For example, consider a large chilled food storage facility where sensor-enabled RFID tags are attached to food items. A collection of readers are installed and periodically read the sensor-produced temperature information from 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.

We study the information collection problem in sensor-enabled RFID systems. The goal is to design efficient protocols for a reader to collect sensor information from tags within the reader's interrogation region, which are called *target tags*. Previous work has addressed a closely related problem where the set of target tags is known *a priori* to the reader [25], [26]. We consider practically common scenarios where such information is unavailable. For example, consider a large warehouse where each item is attached with a sensor-enabled RFID tag and multiple readers are installed to ensure the full coverage of the system. The set of tags in the whole warehouse can be easily monitored through a checkin/checkout procedure at the entrance. However, inside the warehouse, as the tagged items are moved around, the subset of tags covered by each reader changes over time. In other words, even though we know the set of all tags, we do not know which are the target tags for each reader at each moment. In another example, a worker carries a mobile reader and walks around in a warehouse to read information from the sensor-enabled RFID tags attached to different items. Suppose the exact placement of tags is not constantly profiled. As the mobile reader may operate at an arbitrary location, it does not know beforehand which (target) tags will be accessible during the operation. While the prior work assumes that a

Manuscript received February 23, 2012; revised November 30, 2012; May 25, 2013; and June 10, 2013; accepted June 11, 2013; approved by IEEE/ACM Transactions on Networking Editor G. Bianchi. Date of publication August 07, 2013; date of current version August 14, 2014. This work was supported in part by the U.S. National Science Foundation under Grants CNS-0916391 and CPS-0931969. The work of C. Zhang was supported in part by the National Natural Science Foundation of China under Grant 61202140.

H. Yue and Y. Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: hyue@ufl.edu; fang@ece.ufl.edu).

C. Zhang is with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: chizhang@ustc.edu.cn).

M. Pan is with the Department of Computer Science, Texas Southern University, Houston, TX 77004 USA (e-mail: panm@tsu.edu).

S. Chen is with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: sgchen@cise.ufl.edu).

Digital Object Identifier 10.1109/TNET.2013.2272761

reader always has the knowledge of all target tags in its interrogation region [25], [26], this assumption does not hold in the above scenarios, where a reader must first determine the target tags located in its interrogation region before collecting information from them. As we will demonstrate through analysis and simulation, these prior solutions (as well as the ID-collection protocols) do not work efficiently when they are adapted for information collection with unknown target tags.

In this paper, we propose a time-efficient protocol for unknown-target information collection in large-scale RFID systems, called the Bloom-filter-based Information Collection protocol (BIC), which is designed to be performed by each reader individually for the collection of information from tags in its interrogation region. We first examine two broad categories of warm-up solutions, which are designed based on existing ID-collection protocols and information collection protocols. We show that these solutions are not efficient in terms of execution time for unknown-target information collection due to their significant communication overhead for the reader to find out the target tags. In order to overcome this drawback and reduce the execution time, a Bloom filter that represents the set of target tags is constructed by our protocol and transmitted to the reader. It significantly improves the communication and time efficiencies for target tag identification. In addition, we introduce allocation vectors to schedule the information transmissions from different tags and minimize collision. Extensive simulations show that our protocol outperforms all the warm-up solutions, and its execution time is always within 2 times of the unachievable universal lower bound.

The rest of this paper is organized as follows. The related work is reviewed in Section II. In Section III, we introduce the system model and define the problem. Warm-up solutions are discussed in Section IV. In Section V, we propose our scheme BIC and elaborate the design of each component. We conduct simulations and evaluate the performance of BIC in Section VI. Finally, we draw the concluding remarks in Section VII.

II. Related Work

In the literature, extensive research effort has been devoted to the design of ID-collection protocols, which aim to read the IDs from all the tags in a single-reader RFID system with minimum execution time [2]. Based on the mechanism used for resolving tag transmission collisions occurred during ID collection, existing ID-collection protocols can be classified into two broad categories: ALOHA-based [3]–[7] and tree-based [8]–[10]. Waldeop *et al.* [27], Zhou *et al.* [28], and Yang *et al.* [29] address the ID-collection problem in multi-reader RFID systems, where the reader–tag and the reader–reader transmission collisions are considered. In [30], Xie *et al.* propose an efficient approach to identify moving tags.

Recently, some research interest has been shifted to new functions of RFID systems. In [11]–[17], 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 [18] and Zhang *et al.* in [19] address the problem of exactly identifying the IDs of the missing tags. In [20], Zheng and Li propose several algorithms to achieve efficient tag searching. The security and privacy issues of RFID systems are discussed in [31]–[35].

The studies that are most related to our work are [25] and [26]. In [25], 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 all the tags in a single-reader RFID system. In [26], 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 with minimum energy consumption. However, all these solutions assume the reader has already known the set of tags from which it will collect information, and cannot effectively address the unknown-target information collection problem where such knowledge is unavailable to the reader, as we will show in Section IV.

III. Preliminaries

A. System Model

Consider an RFID system consisting of a back-end server, one or several readers, and numerous tags. The tags can be either lightweight passive ones that are energized by the radio wave transmitted from the readers, or battery-powered active (or semi-passive) ones that have longer communication range. Each tag carries a unique ID and is integrated with one or more sensors to monitor some physical parameters. We assume that the area covered by the RFID system is large, and one mobile reader or multiple static readers are used to ensure the full coverage. Each reader covers a certain region, within which it can communicate with the tags. This region is called the *interrogation region*, and such tags are called the *target tags* of the reader. Since the objects with RFID tags may be moved 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. The RFID readers are connected to the back-end server via a high-speed wired or wireless network. We assume that the back-end server stores the IDs of all the tags present in the whole system. Such information can be obtained either by regularly updating the database when objects are moved into or out of the system, or in case of errors such as database damage, executing an ID-collection protocol such as [29].

Communications between the readers and the tags follow the *Reader-Talks-First* protocol [36]: A reader first issues a request message to initiate the communication, and then several tags respond during a number of slots in a following time frame. For each time-slot in the frame, if no tag responds, it is called an empty slot; otherwise, it is called a nonempty slot. The message transmitted in a nonempty slot can be successfully received by the reader only if a single tag responds. When multiple tags respond in the same slot, there is a transmission collision, and the reader cannot correctly decode the messages. The reader's signal will synchronize the clocks of the tags.

B. Problem Definition

Let $\mathcal{N}$ denote the set of all the tags in the RFID system and $\mathcal{N}_r$ denote the set of tags within the interrogation region of a reader r. Let $n = |\mathcal{N}|$ and $n_r = |\mathcal{N}_r|$, where $|\cdot|$ stands for the cardinality of a set. We define ρ_r as the ratio of n_r to n, i.e., $\rho_r = n_r/n$. The unknown-target information collection problem is to design a protocol for reader r to collect sensor information from the target tags in $\mathcal{N}_r$ with minimum execution time, where $\mathcal{N}_r$ is unknown to the reader r. Note that the sensor information includes not only sensor readings, but also the mapping from each reading to a tag where the reading takes place so that the sensor data can be accurately associated with the corresponding object. As we will see in the simulation results, the running time of an information collection protocol is relatively small. Hence, we assume that the distribution of the tags is stable during the protocol execution. If there are some tags entering into or departing from the interrogation region of reader r during information collection, the reader may simply ignore them this round and read information from the updated set of target tags in the next round of scheduled protocol execution.

IV. Warm-Up Solutions

In this section, we examine two broad classes of candidate solutions to the unknown-target information collection problem. One class is derived from the existing ID-collection protocols. The other class is derived from the protocols originally designed for traditional information collection where the set of target tags is known to the reader. We demonstrate the inefficiency of these schemes and also discuss the inherent reasons, which motivate the novel scheme we propose in Section V.

A. ID-Collection-Based Information Collection Protocols

The ID-collection protocols can be borrowed here for information collection, i.e., each target tag piggybacks the sensor information when transmitting its ID to the reader, which are therefore referred to as *ID-collection-based information collection protocols* (IDPS). Based on the anti-collision mechanism used to resolve the transmission collisions among different tags, the IDPS can be further classified into two broad categories: ALOHA-based and tree-based.

The *ALOHA-based Information Collection protocols* (AIC) work as follows: The reader broadcasts a request message to all the target tags, which specifies the number of slots contained in the subsequent time frame. Each tag individually and randomly selects one slot to transmit both its ID and the sensor information to the reader. If there is a collision in a time-slot due to multiple responses, the involved tags will be acknowledged to restart in the next frame. The similar process repeats until all the target tags report their information to the reader. Different from AIC, the *Tree-based Information Collection protocols* (TIC) resolve transmission collisions by splitting the set of involved tags into two subsets with tag IDs or random numbers. The splitting procedure will continue until each set contains only one tag. In this way, the target tags are organized into a tree structure. The reader walks through the tree and collects the IDs and sensor information from all the tags.

In IDPS, since each target tag will transmit its ID to the reader, the reader r could attain the set $\mathcal{N}_r$ without $\mathcal{N}$. Let τ denote the length of a time-slot during which a tag is able to transmit both the ID and the information to a reader. We can observe that it takes a reader at least τ to collect the information from one tag with IDPS. Therefore, the lower bound on the execution time of IDPS is equal to $n_r \times \tau$, which is the aggregation time for all the target tags in $\mathcal{N}_r$ to report their IDs and information to the reader r.

B. Sequential Identification-Based Information Collection Protocols

We now consider the existing protocols designed for information collection where the set of target tags is known to the reader. A basic protocol is called the *Polling-based Information Collection protocol* (PIC) [25]. The reader broadcasts the IDs one after another and waits for the response from the corresponding tag. Each target tag keeps listening to the communication channel until its own ID is received. Then, it transmits the information to the reader and remains silent thereafter. For unknown-target information collection, since the reader does not have prior knowledge about $\mathcal{N}_r$, it must broadcast all the IDs in $\mathcal{N}$ to determine the target tags. Another protocol is called the MIC [25], which removes the transmissions of tag IDs in PIC. MIC consists of multiple phases. In each phase, the reader uses several hash functions to map the tags to different 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 transmissions, and others are wasted to avoid collisions. Similar to PIC, when MIC is used for unknown-target information collection, since the reader r does not know its target tag set $\mathcal{N}_r$, it has to assign one time-slot to each tag in $\mathcal{N}$. If the time-slot allocated to a tag turns out to be empty, the tag is believed not in the interrogation region of reader r.

One common characteristic of the protocols described above is that the reader sequentially examines all the tags in $\mathcal{N}$ to find out the target tag set $\mathcal{N}_r$. Therefore, these protocols are also referred to as *sequential identification-based information collection protocols* (SIPS). Let τ_{inf} be the length of a time-slot for a tag to transmit the information to a reader. When SIPS are executed, each target tag needs at least τ_{inf} to show its presence and report the information to the reader. In addition, for every tag in $\mathcal{N} \backslash \mathcal{N}_r$, it takes a reader at least τ_{det} to verify its absence, where τ_{det} is the minimum required detection time for a reader to determine the existence of a transmission on the communication channel. Therefore, the lower bound on the execution time of SIPS is $(n - n_r) \times \tau_{\text{det}} + n_r \times \tau_{\text{inf}}$.

C. Lower Bound for all Unknown-Target Information Collection Protocols

A lower bound on the execution time of *any* protocol for unknown-target information collection is $n_r \times \tau_{\text{inf}}$, which is the aggregated time for all the target tags to transmit their information to the reader r. Note that this lower bound can never be achieved because the reader has to transmit additional control messages to find out the target tags and coordinate their transmissions against collisions. We use the lower bound as a benchmark to evaluate the performance of different protocols for unknown-target information collection.

D. Performance Analysis

So far, we have derived a lower bound for execution time of IDPS, a lower bound for execution time of SIPS, and a universal

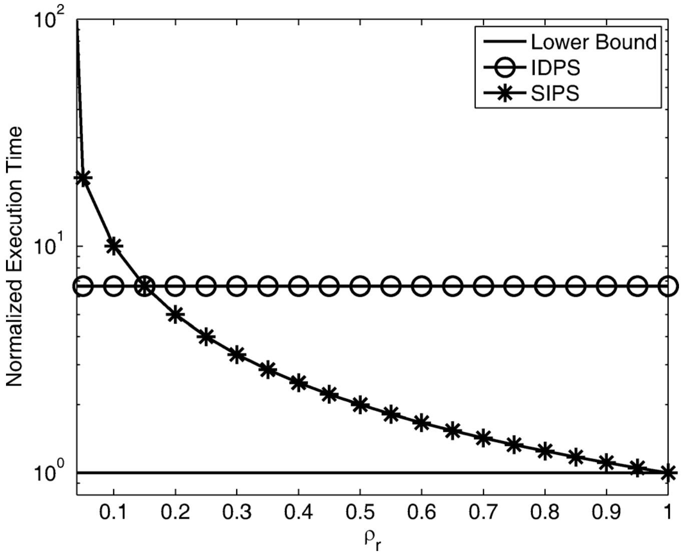


Fig. 1. Performance comparison of different information collection protocols.

lower bound for all unknown-target information collection protocols. In this section, we will show that the first two lower bounds—which represent the state of the art—are much higher than the third universal lower bound. This result indicates that there is potentially much room for improvement. Indeed, our new protocol significantly outperforms IDPS and SIPS, and its execution time is close to the universal lower bound.

Fig. 1 shows the execution time comparison of different information collection protocols, where the lower bounds of IDPS and SIPS are normalized with respect to the universal lower bound $n_r \times \tau_{\text{inf}}$, and the information has 1 bit. Two observations can be made from Fig. 1. First, the lower bound on the execution time of SIPS approaches the universal lower bound when ρ_r is close to 1. However, its performance degrades quickly as ρ_r decreases to 0. Especially, when $\rho_r < 0.1$, the execution time of SIPS could be more than 10 times of the universal lower bound. Second, there is a wide constant gap between the execution time of IDPS and the universal lower bound, which can be as large as 7 times as shown in Fig. 1. Note that the lower bounds on the execution time of IDPS and SIPS are the best performance that the information collection protocols falling into these two classes can possibly achieve. Therefore, neither of them are efficient enough for unknown-target information collection.

During the information collection, since the reader r does not know which tags are located within its interrogation region, it has to first determine the target tags before collecting information from them. Hence, the overall execution time of any unknown-target information collection protocol can be divided into two parts: the time for the reader to find out all the target tags and the time for the tags to report their information to the reader. IDPS and SIPS are not efficient for unknown-target information collection due to the significant time overhead incurred for determining the target tags, which are at least $n_r \times \tau_{\text{id}}$ and $(n - n_r) \times \tau_{\text{det}}$, respectively. Here, τ_{id} denotes the length of a time-slot for a tag to transmit its ID to the reader. Therefore, in order to minimize the overall execution time for information collection, we need to explore new technologies for the reader to efficiently determine the target 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 [37]. Hence, if the set $\mathcal{N}_r$ can be transmitted to the reader in the form of a Bloom filter, the communication overhead for target tag identification could be drastically reduced and thus the overall time for information collection. Following this idea, we propose a *Bloom-filter-based Information Collection protocol* (BIC), by which a Bloom filter is distributively constructed for the reader to efficiently determine the target tags. In addition, we introduce the allocation vectors to schedule the transmissions from different tags in order to reduce collision. By using these mechanisms, BIC significantly improves time efficiency for unknown-target information collection as compared to the warm-up solutions.

A. Protocol Description

To determine the target tags with a Bloom filter, the reader r first broadcasts a request message, which contains two parameters w_r and k_r. Here, w_r is the size of the Bloom filter, and k_r is the number of independent hash functions used to construct the Bloom filter. How to choose the values of w_r and k_r will be explained later. Let $h_1, h_2, \ldots, h_{k_r}$ denote the k_r hash functions, each with range $\{0, 1, \ldots, w_r - 1\}$. Upon receiving the request message, every target tag in $\mathcal{N}_r$ generates an array of w_r bits, all of which are initialized to 0. With the k_r hash functions, the tag pseudo-randomly maps its unique ID to k_r bits at positions $h_1(\text{ID}), h_2(\text{ID}), \ldots, h_{k_r}(\text{ID})$ in the array and sets them to 1. The resulting array is called a *Bloom filter basis*. All the target tags transmit their respective Bloom filter basis simultaneously. At the physical layer, a binary "0" corresponds to an idle carrier, where no signal is detected in the channel; a binary "1" corresponds to a busy carrier [18], where a transmission signal is detected in the channel. For each bit received at 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_r-bit array $\mathcal{B}_r$, which turns out to be the Bloom filter constructed based on $\mathcal{N}_r$.

After the reader attains the Bloom filter $\mathcal{B}_r$, it tests all the IDs in $\mathcal{N}$ to determine the target tag set $\mathcal{N}_r$. Specifically, for each ID in $\mathcal{N}$, the bits at positions $h_1(\text{ID}), h_2(\text{ID}), \ldots, h_{k_r}(\text{ID})$ in $\mathcal{B}_r$ are examined. If any of them is 0, the corresponding tag is certainly not in $\mathcal{N}_r$. Otherwise, the tag is considered to be included in $\mathcal{N}_r$. Finally, the reader r can extract a new set from $\mathcal{B}_r$, which is denoted as $\tilde{\mathcal{N}}_r$. According to the property of Bloom filter, false negatives are impossible, which means any tag in $\mathcal{N}_r$ will be identified into $\tilde{\mathcal{N}}_r$. However, false positives may occur with a certain probability. In the case of false positives, a tag that is in fact not included in $\mathcal{N}_r$ is identified into $\tilde{\mathcal{N}}_r$ because all the bits it is mapped to in $\mathcal{B}_r$ are set to 1 by other IDs in $\mathcal{N}_r$. Therefore, the set $\tilde{\mathcal{N}}_r$ satisfies $\mathcal{N}_r \subseteq \tilde{\mathcal{N}}_r$. The tags in $\tilde{\mathcal{N}}_r \backslash \mathcal{N}_r$ are called the *false positive identified tags*.

For illustrative purpose, we take a simple example to show the procedures of the target 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.,

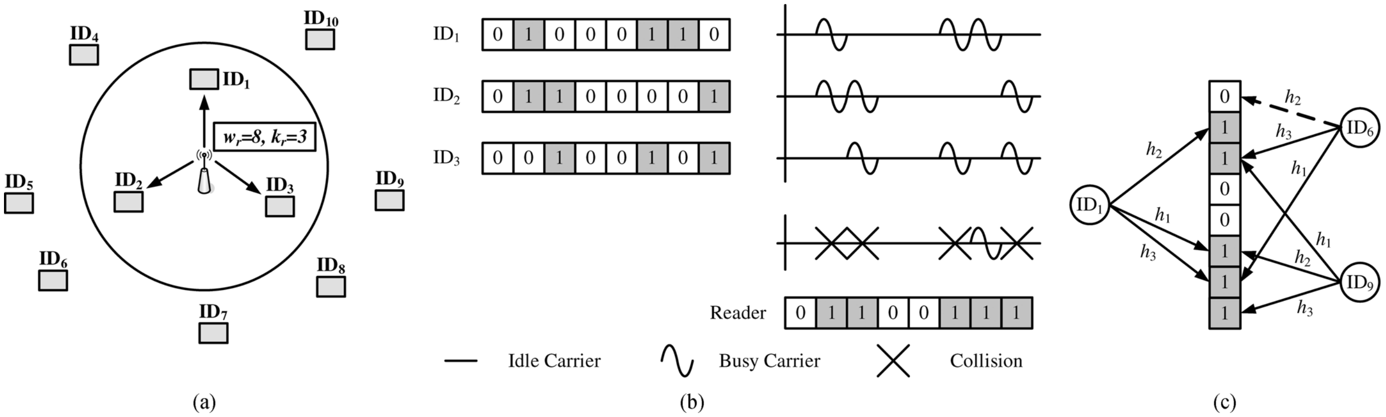


Fig. 2. Simple example to illustrate the procedures of the target tag identification with a Bloom filter. (a) Broadcast transmission of the parameters. (b) Bloom filter construction. (c) Extraction of the target tag set.

$\mathcal{N} = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_{10}\}$. For the reader r, we assume that the tags ID_1, ID_2, and ID_3 are located within its interrogation region, i.e., $\mathcal{N}_r = \{\mathrm{ID}_1, \mathrm{ID}_2, \mathrm{ID}_3\}$. As shown in Fig. 2(a), to determine the target tags, the reader r first sends out a request message, which contains the values of the parameters w_r and k_r. Here, we assume $w_r = 8$ and $k_r = 3$. When receiving the request message, each tag in $\mathcal{N}_r$ 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. Then, 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). After receiving all the Bloom filter bases, the reader attains a bit array 01100111, which is exactly the same as the Bloom filter constructed based on the set $\mathcal{N}_r$. Then, the reader checks the elements in $\mathcal{N}$ one by one to find out the set $\mathcal{N}_r$ with the Bloom filter. For example, as shown in Fig. 2(c), since the bits at positions $h_1(\mathrm{ID}_1)$, $h_2(\mathrm{ID}_1)$, and $h_3(\mathrm{ID}_1)$ are all equal to 1, the tag ID_1 is believed to be in $\mathcal{N}_r$. However, the tag ID_6 is not included in the set $\mathcal{N}_r$ due to the fact that $h_2(\mathrm{ID}_6) = 0$. For the tag ID_9, it is actually not in $\mathcal{N}_r$, but it is able to pass the test. Thus, it will be falsely identified as a target tag.

When the reader r obtains the set $\tilde{\mathcal{N}}_r$, it could start to collect information from the target tags. Information collection consists of several rounds. Each round begins with a request message broadcast from the reader, followed by a slotted time frame during which some tags are scheduled to report their information to the reader. The reader uses a so-called *allocation vector* to coordinate the tags' transmissions, which is denoted as $\mathcal{V}_r$. The length of the allocation vector $\mathcal{V}_r$ exploited in each round is equal to the number of the tags in $\tilde{\mathcal{N}}_r$ from which the reader has not yet collected the sensor information. In the rest of the paper, these tags are referred to as *uncollected tags* for simplicity. The reader picks a random number π and uses a hash function h to map the ID of each uncollected tag to a bit in $\mathcal{V}_r$, which is called the *indicator bit* of the uncollected tag. For each bit in $\mathcal{V}_r$, if there is only one uncollected tag mapped to it, the bit is set to 1, which means the tag is allowed to respond its information to the reader during one of the time-slots in the following frame. Otherwise, if no or several uncollected tags are mapped to the same bit, the bit is 0.

At the beginning of a round, the reader r first broadcasts a request message to all the tags within its interrogation region, which contains the random number π and the allocation vector $\mathcal{V}_r$. If the allocation vector $\mathcal{V}_r$ 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 VI-A). When receiving the request message, each uncollected tag inputs its ID and π into the same hash function exploited by the reader and obtains the position of its indicator bit in the allocation vector $\mathcal{V}_r$. Then, it examines the corresponding bit. If its value is 0, the tag will delay the information transmission to the next round to avoid potential transmission collisions. If the bit is 1, the tag then calculates how many 1's appear before its indicator bit in $\mathcal{V}_r$. 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 1's preceding its indicator bit, the tag should be the $(i+1)$th responder in the current round to report its information. Then, during the following time frame, it will transmit the information in the $(i+1)$th slot without collision.

If one time-slot allocated to a tag in $\tilde{\mathcal{N}}_r$ for information transmission turns out to be empty, the tag is a false positive identified one, and the reader will delete the corresponding ID from $\tilde{\mathcal{N}}_r$. In this way, at the end of the information collection, the reader will successfully remove all the false positive identified tags from $\tilde{\mathcal{N}}_r$ and exactly obtain the target tag set $\mathcal{N}_r$.

B. Parameter Determination

Next, we show how to determine the values of the parameters w_r and k_r of the Bloom filter. The total execution time of information collection is the sum of the time for the reader to find out its target tags and the time for the tags to transmit their information to the reader. During the target tag identification, the reader r broadcasts one request message, which is followed by the concurrent transmissions of the Bloom filter bases from all the target tags. The time for a target tag to transmit its w_r-bit Bloom filter basis can be calculated as $w_r \times \tau_{\mathrm{bit}}$, where τ_{bit}

is the time for a tag to transmit one bit. Since the request message is very short, we do not take its transmission time into consideration. We also ignore the computation time for the reader to extract the target tag set from the Bloom filter. Therefore, the time for target tag identification is about $w_r \times \tau_{\text{bit}}$. The time for information collection includes the time for the reader to transmit the request messages and the time for the slotted frames. Similar to [25], we can prove that the expected number of indicator bits for each tag is e, where e is the natural constant. Let p_r denote the probability of the false positives of the Bloom filter $\mathcal{B}_r$. Then, the expected cardinality of the set $\tilde{\mathcal{N}_r}$ is $n_r + p_r \times (n - n_r)$. Hence, the total number of bits in all allocation vectors is expected to be $e \times \left[n_r + p_r \times (n - n_r)\right]$, and the expected time for the reader to broadcast all the allocation vectors is about $\frac{e \times [n_r + p_r \times (n - n_r)]}{96} \times t_{\text{id}}$. The rest of the request message is very small, and thus the transmission time can be ignored. Since each tag in $\tilde{\mathcal{N}_r}$ will be allocated a unique time-slot to report its information to the reader, the expected number of time-slots in all the frames should be equal to $n_r + p_r \times (n - n_r)$, and the overall frame time in all rounds is expected to be $\left[n_r + p_r \times (n - n_r)\right] \times \tau_{\text{inf}}$.

Based on the above analysis, the expected execution time for information collection at the reader r, which is denoted as T_r, can be calculated as follows:

$$T_r = w_r \times \tau_{\text{bit}} + \left(\frac{e \times t_{\text{id}}}{96} + \tau_{\text{inf}}\right) \times \left[n_r + p_r \times (n - n_r)\right]. \quad (1)$$

Given the number of target tags n_r and the probability of false positives p_r, the minimum length of the Bloom filter w_r is [38]

$$w_r = -\frac{n_r \times \ln p_r}{(\ln 2)^2} \quad (2)$$

when the number of hash functions $k_r = \frac{w_r}{n_r} \ln 2$. Therefore, the execution time T_r can be rewritten as

$$T_r = -\frac{n_r \times \ln p_r}{(\ln 2)^2} \times \tau_{\text{bit}} + \left(\frac{e \times t_{\text{id}}}{96} + \tau_{\text{inf}}\right) \times \left[n_r + p_r \times (n - n_r)\right]. \quad (3)$$

When the number of all the tags n, the number of target tags n_r and the information length l are determined, T_r is a function of the false positive probability p_r. Compute the first-order derivative of T_r with respect to p_r and let it be zero, i.e.,

$$\frac{dT_r}{dp_r} = -\frac{n_r \times \tau_{\text{bit}}}{(\ln 2)^2} \times \frac{1}{p_r} + \left(\frac{e \times t_{\text{id}}}{96} + \tau_{\text{inf}}\right) \times (n - n_r) = 0. \quad (4)$$

By solving (4), we obtain

$$p = \frac{\tau_{\text{bit}} \times \rho_r}{(\ln 2)^2 \left(\frac{e \times t_{\text{id}}}{96} + \tau_{\text{inf}}\right)(1 - \rho_r)}. \quad (5)$$

Note that p_r must satisfy $0 < p_r \leq 1$ and we have $p > 0$. When $0 < p \leq 1$, the optimal value of p_r is

$$p_r^o = p = \frac{\tau_{\text{bit}} \times \rho_r}{(\ln 2)^2 \left(\frac{e \times t_{\text{id}}}{96} + \tau_{\text{inf}}\right)(1 - \rho_r)}. \quad (6)$$

When $p > 1$, we have $dT_r/dp_r < 0$ for $0 < p_r \leq 1$. Hence, the optimal value of p_r is

$$p_r^o = 1. \quad (7)$$

Therefore, the optimal value of p_r can be determined as

$$p_r^o = \min\left(\frac{\tau_{\text{bit}} \times \rho_r}{(\ln 2)^2 \left(\frac{e \times t_{\text{id}}}{96} + \tau_{\text{inf}}\right)(1 - \rho_r)}, 1\right). \quad (8)$$

Then, the length of the Bloom filter is

$$w_r^o = -\frac{n_r \times \ln p_r^o}{(\ln 2)^2} \quad (9)$$

and the number of hash functions is

$$k_r^o = \frac{w_r^o}{n_r} \ln 2. \quad (10)$$

The minimum execution time T_r^o can be expressed as follows:

$$T_r^o = w_r^o \times \tau_{\text{bit}} + \left(\frac{e \times t_{\text{id}}}{96} + \tau_{\text{inf}}\right) \times \left[n_r + p_r^o \times (n - n_r)\right]. \quad (11)$$

C. Cardinality Estimation

In order to determine the parameters of the Bloom filter, such as p_r^o, w_r^o, and k_r^o, the reader r must know the number of the target tags n_r, which might not be available in some application scenarios. In that case, we have to estimate the cardinality of the target tag set before running our information collection protocol BIC. In the literature, many estimation algorithms [11]–[15] have been designed to quickly and accurately estimate the size of tag population for RFID systems, which can be integrated into our protocol for deriving the estimated value of n_r.

1) Estimation Error: When the number of target tags n_r is unknown, the reader r will estimate it and then determine the values of p_r, w_r, and k_r based on the estimated value $\hat{n_r}$ rather than n_r. Since $\hat{n_r}$ returned from the estimation algorithms is usually not exactly equal to n_r, the parameters of the Bloom filter calculated based on $\hat{n_r}$ might not be optimal with respect to n_r, which will lead to the performance degradation of our protocol. Next, we investigate the sensitivity of the execution time of BIC to the estimation error.

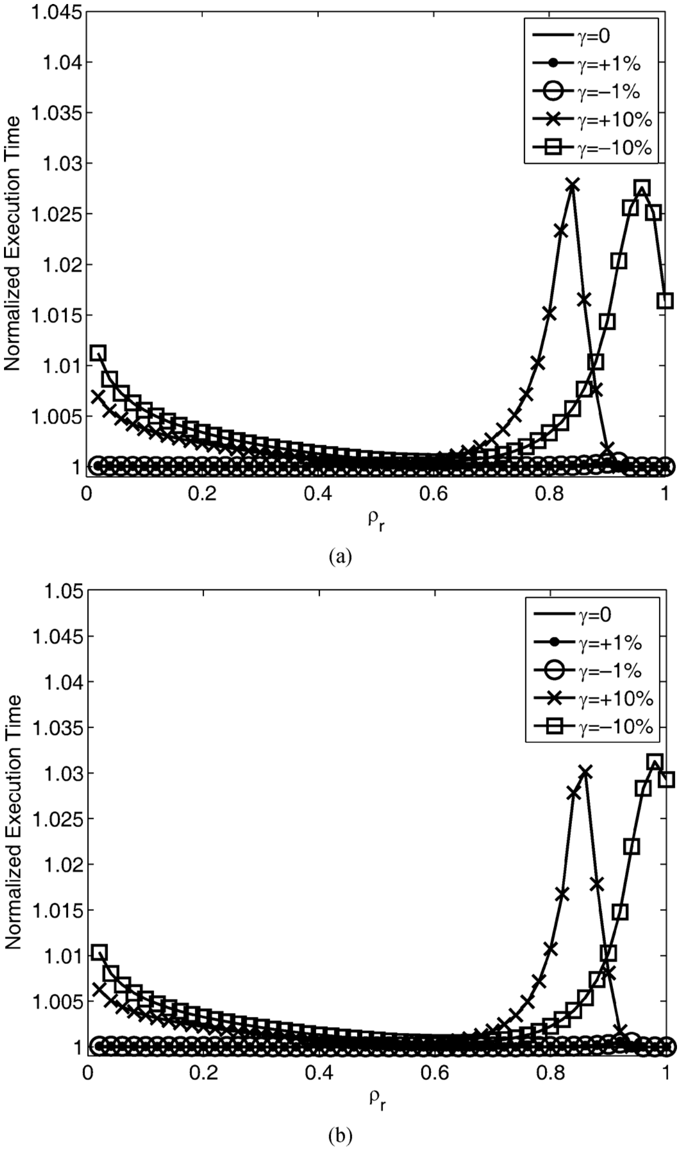
We define $\hat{\rho_r}$ as the ratio of $\hat{n_r}$ to n, i.e., $\hat{\rho_r} = \hat{n_r}/n$. Then, the optimal false positive probability calculated based on $\hat{n_r}$ is

$$\hat{p_r^o} = \min\left(\frac{\tau_{\text{bit}} \times \hat{\rho_r}}{(\ln 2)^2 \left(\frac{e \times t_{\text{id}}}{96} + \tau_{\text{inf}}\right)(1 - \hat{\rho_r})}, 1\right). \quad (12)$$

Given the false positive probability $\hat{p_r^o}$, the length of the Bloom filter and the number of hash functions are determined as follows:

$$\hat{w_r^o} = -\frac{\hat{n_r} \times \ln \hat{p_r^o}}{(\ln 2)^2} \quad (13)$$

$$\hat{k_r^o} = \frac{\hat{w_r^o}}{\hat{n_r}} \ln 2. \quad (14)$$



(a)

(b)

Fig. 3. Impact of estimation error on the execution time of BIC. (a) $l = 1$ bit. (b) $l = 8$ bits.

When such a Bloom filter is used to determine the target tag set $\mathcal{N}_r$, the resulting false positive probability $\hat{p_r}$ is

$$\hat{p_r} = \left[1 - \left(1 - \frac{1}{\hat{w_r^o}}\right)^{\hat{k_r^o} n_r}\right]^{\hat{k_r^o}} \approx \left(1 - e^{-\frac{\hat{k_r^o} n_r}{\hat{w_r^o}}}\right)^{\hat{k_r^o}}. \quad (15)$$

Therefore, the expected cardinality of the set $\tilde{\mathcal{N}_r}$ is $n_r + \hat{p_r} \times (n - n_r)$, and the execution time $\hat{T_r}$ can be calculated as

$$\hat{T_r} = \hat{w_r^o} \times \tau_{\text{bit}} + \left(\frac{e \times t_{\text{id}}}{96} + \tau_{\text{inf}}\right) \times \left[n_r + \hat{p_r} \times (n - n_r)\right]. \quad (16)$$

Fig. 3 shows the expected execution time of BIC under different levels of estimation error when the information is 1 bit and 8 bits long. Here, all results are normalized with respect to the expected execution time of BIC with no estimation error. We use γ to represent the level of estimation error, where $\gamma = \frac{\hat{n_r} - n_r}{n_r}$. When ρ_r is small, much time is taken for the transmissions of the Bloom filter bases. As ρ_r increases, the effect of estimation error on the determination of the length of the Bloom filter bases decreases based on (9), and so does the overall execution time. When ρ_r is large, the time for information transmissions dominates the performance of our protocol. As ρ_r increases, the performance degradation due to estimation error increases because the false positive probability $\hat{p_r^o}$ calculated based on $\hat{n_r}$ becomes more inaccurate compared to the optimal value p_r^o. When ρ_r is close to 1, we have $p_r^o = \hat{p_r^o} = 1$ according to (8) and (12). In this case, target tag identification with Bloom filter is skipped, and the performance of BIC under estimation error is the same as that without estimation error. Therefore, the normalized execution time reduces to 1. This also explains why we have a peak on each curve with estimation error as shown in Fig. 3. It can be observed that BIC is very robust to the estimation error: When $\gamma = \pm 1\%$, the execution time of BIC is almost the same as that without estimation error. Even when γ reaches $\pm 10\%$, the expected execution time of BIC increases only by up to 3%.

TABLE I
ESTIMATION OVERHEAD

n_r	2500	5000	10000	15000	20000
$l = 1$bit	36.5%	19.1%	10.0%	7.0%	5.3%
$l = 8$bits	30.2%	15.6%	8.2%	5.6%	4.3%
$l = 16$bits	25.3%	13.0%	6.7%	4.6%	3.5%
$l = 32$bits	19.1%	9.8%	5.0%	3.4%	2.6%

2) Estimation Overhead: We also measure the additional time overhead for estimating the cardinality of the target tag set. Here, we use the Enhanced Zero-Based (EZB) estimator designed in [12] to estimate n_r. We set the confidence interval $\beta = 10\%$ and the reliability $\alpha = 99\%$. Table I illustrates the ratio of the running time of the estimation algorithm to the execution time of BIC for information collection under different information length l and the number of target tags n_r when $n = 50\,000$. It shows that the estimation overhead is moderate, and it will decrease as the information length or the number of target tags increases. In Section VI, we demonstrate that BIC outperforms other information collection protocols with consideration of the estimation overhead via extensive simulations.

D. Channel Error

Until now, the wireless communication channels are considered to be error-free. If this is not true, the normal execution of BIC might be disturbed. 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 case of false negatives, some target tags in $\mathcal{N}_r$ will not be identified into $\tilde{\mathcal{N}_r}$ and hence will not be intentionally allocated a time-slot by the reader r. Therefore, these tags will either have no chance to report their information to the reader during the execution of our protocol 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, such as AIC, to read sensor information from the tags that have not successfully submitted their information to the reader.

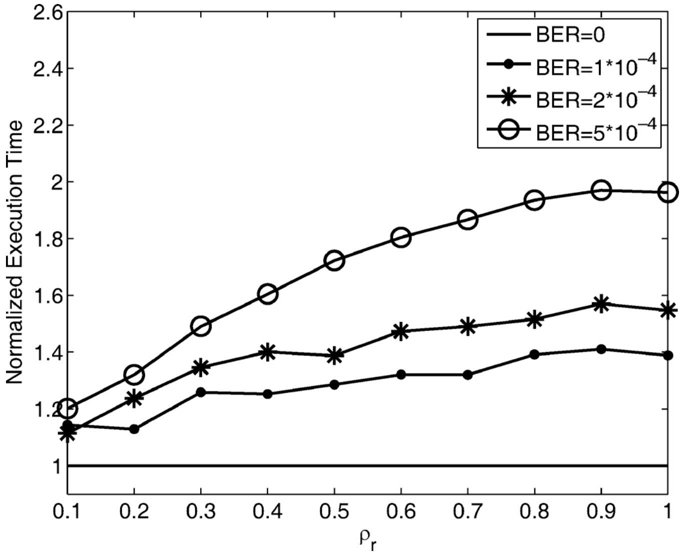


Fig. 4. Impact of channel error on the execution time of BIC.

During information collection, to ensure the correctness of the information received at the reader via an unreliable communication channel, we include 16-bit checksum to the information for error detection. Each segment of 96 bits in the allocation vectors also carries 16-bit checksum. In BIC, a target 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 1's in the previous segments. When a target 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 1's appearing before its indicator bit in the current segment, no matter whether 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-based protocol as we mentioned above.

Next, we evaluate the performance of BIC under different bit error rates (BERs), and the results are shown in Fig. 4. Here, all results are normalized with respect to the execution time of BIC when the channel is error-free. We observe that as the BER increases, the execution time of BIC increases. When the BER is small (e.g., 1×10^{-4}), the execution time of BIC only increases by 40%. Even when the BER reaches 5×10^{-4}, the execution time of BIC is still within 2 times of that under error-free channel.

E. Synchronization

Similar to [18] and [20], in our protocol, we assume that each bit in the Bloom filter transmitted from different target tags to the reader is synchronized. RFID systems work in low-rate channels, and hence the bit time is long when compared to other wireless technologies such as WiFi. Based on the Philips I-Code specification [36], the time interval for the transmission of one bit from a tag to a reader is 18.88 μs. According to the data in [1], we assume that the communication range of the tags is 15 m. Then, the propagation delay is $\frac{15}{3\times10^8} = 0.05$ μs, which is negligible compared to the bit time 18.88 μs. Therefore, after the reader's signal synchronizes the tags' clocks, the tags are able to transmit during specific bit times even though their distances to the reader are different. Click drift might constrain the maximum number of consecutive bits transmitted. If the size of the Bloom filer is large, we can divide the Bloom filter basis into smaller segments, and the reader will transmit signal to resynchronize the tags at the end of each segment.

VI. Evaluation

In this section, we evaluate the performance of our protocol BIC. We compare the execution time of BIC with AIC, TIC, PIC, MIC, as well as the universal lower bound and demonstrate the efficiency of BIC for unknown-target information collection in large-scale RFID systems.

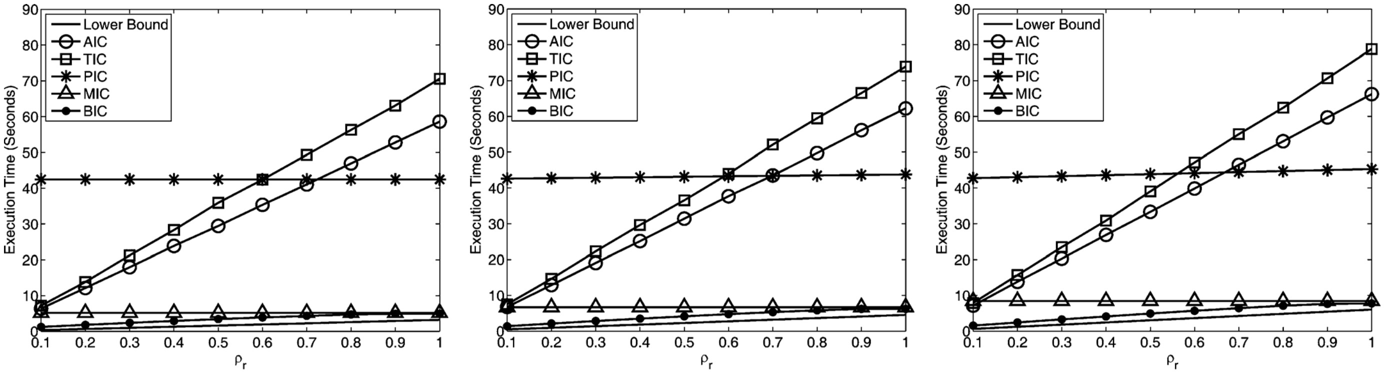
A. Simulation Setting

The simulation setting is based on the Philips I-Code specification [36] and the EPCGlobal Gen2 standard [39]. 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 μs. The transmission rate of the reader is 26.5 kb/s. Thus, the time for the reader to transmit an ID or a segment of allocation vectors is 3927 μs with a time interval included, i.e., $t_{\rm id} = 3927$ μs. The transmission rate of a tag is 53 kb/s, which is different from that of the reader. It takes 18.88 μs for a tag to transmit 1 bit, i.e., $\tau_{\rm bit} = 18.88$ μs. The value of $\tau_{\rm inf}$ is calculated as the sum of a time interval and the information transmission time that equals to 18.88 μs multiplied by the length of the information l. For example, if the sensor information is 8 bits long, $\tau_{\rm inf}$ is equal to 452 μs. Recall that τ represents the length of a time-slot for a tag to transmit both the ID and information. Similarly, we have $\tau = (18.88 \times l + 2114)$ μs.

We use EZB to estimate the cardinality of each target tag set, where the confidence interval $\beta = 10\%$ and the reliability $\alpha = 99\%$. Under the same setting, we take the average values of 100 simulation runs as results.

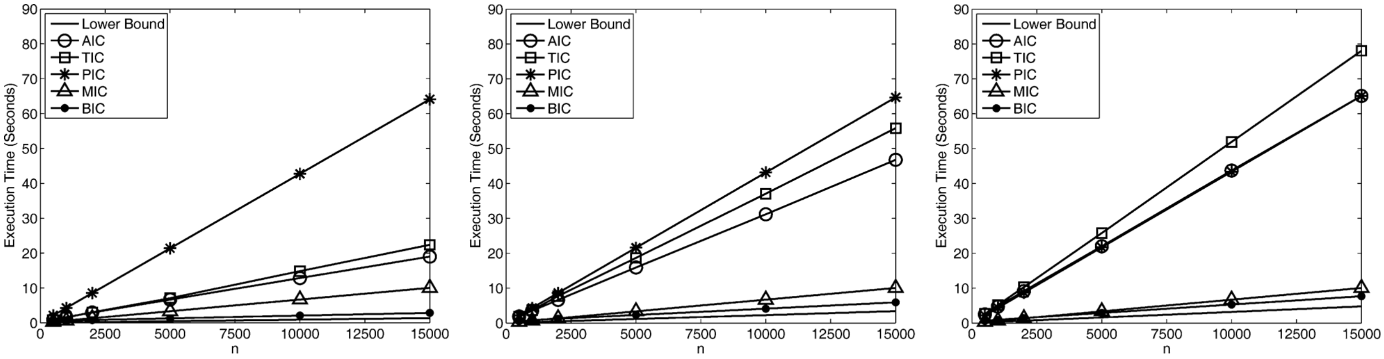
B. Performance Comparison Under Different Ratio ρ_r

We first evaluate the performance of the protocols for unknown-target information collection under different values of ρ_r. Three sets of simulations are conducted, where $n = 10\,000$ and the length of sensor information is 1, 8, and 16 bits, respectively. The simulation results are shown in Fig. 5. The execution time of AIC and TIC increases linearly as ρ_r increases. Previous work [40]–[42] has theoretically demonstrated that the expected number of time-slots needed for a reader to collect IDs from m tags with ALOHA-based ID-collection protocols and tree-based ID-collection protocols is approximately $2.72 \times m$ and $2.88 \times m$, respectively. Therefore, the execution times of AIC and TIC should be linear functions of ρ_r when n is a constant, which is confirmed by our simulation results. In addition,



(a) (b) (c)

Fig. 5. Execution time comparison of information collection protocols under different ratio ρ_r. (a) $l = 1$ bit. (b) $l = 8$ bits. (c) $l = 16$ bits.



(a) (b) (c)

Fig. 6. Execution time comparison of information collection protocols under different number of tags. (a) $\rho_r = 0.2$. (b) $\rho_r = 0.5$. (c) $\rho_r = 0.7$.

we observe that when n is fixed, the performance of PIC is insensitive to change in ρ_r because almost all the execution time is devoted to the broadcast transmissions of tag IDs in $\mathcal{N}$ from the reader, which only depends on n rather than n_r. Moreover, when ρ_r is small (e.g., less than 0.5), the execution time of PIC is much larger than that of AIC and TIC, due to its significant communication overhead for the reader to broadcast the IDs of the tags in $\mathcal{N}\backslash\mathcal{N}_r$. As ρ_r increases, PIC works better than AIC and TIC since it avoids the severe collision among tag transmissions occurred to AIC and TIC. MIC completely removes the time-consuming transmissions of tag IDs during information collection, and thus further reduces the protocol execution time as compared to AIC, TIC, and PIC. BIC constructs a Bloom filter for the reader to efficiently determine the target tags, and it uses the allocation vectors to coordinate the transmissions from different tags and thus reduce collision. Therefore, it achieves the best performance among all the protocols, and its execution time is within 2 times of the universal lower bound $n_r \times \tau_{\inf}$ in all simulations as shown in Fig. 5. For example, in the case that $l = 8$ bits and $\rho_r = 0.4$, the execution time of BIC is 3.5 s, which is 10% of the time required by AIC, 12% of the time required by TIC, 8% of the time required by PIC, 52% of the time required by MIC, and 1.4 times of the universal lower bound, respectively.

C. Performance Comparison Under Different Number of Tags

Next, we study the execution time of different information collection protocols with respect to the number of tags in the system. In our simulations, the sensor information is 8 bits long, and the number n of tags varies from 500 to 15 000. Fig. 6 presents the execution time comparison among AIC, TIC, PIC, MIC, and BIC when the ratio ρ_r is 0.2, 0.5, and 0.7, respectively.

From the results in Fig. 6, we observe that when ρ_r is fixed, the execution times of all the information collection protocols increase approximately linearly as the number of tags in the system increases. When n is small (e.g., $n = 500$), the execution time of these information collection protocols is almost the same because the time overhead due to transmission collision and ID broadcast is also low for the warm-up solutions. Such overhead increases as n increases, and BIC significantly outperforms the other protocols when n is large. We also observe that the rate of increase in the execution time varies for different protocols. The execution times of AIC and TIC increase faster under a larger value of ρ_r because the transmission collision among the target tags becomes more severe as ρ_r increases. The rate of increase in the execution time for PIC and MIC remains constant under different values of ρ_r for the same reason as we discussed in Section VI-B. BIC has the minimum rate of increase in the execution time among all the information collection protocols, which indicates that it is efficient and scalable for large-scale RFID systems.

D. Performance Comparison Under Different Information Length

We compare the performance of different information collection protocols with various information lengths. In the

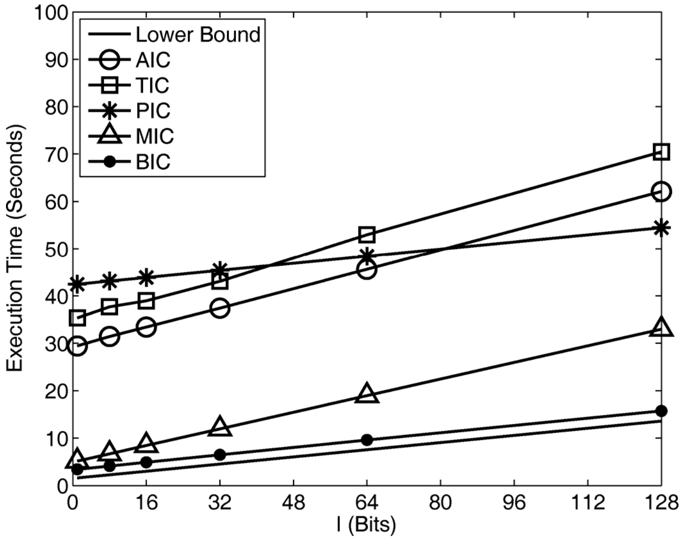


Fig. 7. Execution time comparison of information collection protocols under different information length.

simulations, we set $n = 10\,000$ and $\rho = 0.5$. The information length changes from 1 bit to 128 bits. The simulation results are presented in Fig. 7, where the execution times of all the information collection protocols increase as the length of the sensor information increases. The execution time of BIC is much smaller than that of the warm-up solutions under different lengths of information. Moreover, the rate of increase in the execution time under AIC, TIC, or MIC is larger than that of PIC or BIC. In AIC and TIC, due to transmission collision, each tag has to transmit its information multiple times on average in order for the reader to successfully receive the information, which results in fast increase of execution time when the information length increases. For MIC, its high rate of increase in execution time is caused by the wasted time-slots allocated to the large number of tags in $\mathcal{N} \backslash \mathcal{N}_r$.

E. Performance Comparison in Multi-Reader Scenarios

Recall that one notable application of our protocol is the information collection in the multi-reader RFID systems, as we described in Section I. We have compared the execution time of different protocols for unknown-target information collection at an individual reader in Sections VI-B–VI-D. In this section, we will evaluate the performance of our protocol in the multi-reader scenarios for completeness, i.e., the overall execution time for a number of readers to collect sensor information from all the tags in an RFID system.

In the simulations, we distribute $4 \times 4 = 16$ RFID readers in a square area of size 50×50 m^2, where the readers are arranged in a grid formation. The interrogation region of the readers is assumed to be a disk of radius 15 m. The tags are randomly distributed within the area. In a large-scale RFID system where multiple readers are installed, adjacent readers will interfere with each other and cause reader–tag collision and reader–reader collision [28], [29]. The reader–tag collision occurs when one reader, say r_1, is in the interrogation region of another reader r_2. In that case, the commands broadcast from reader r_1 might drown the response transmitted to reader r_2 from its interrogated tags. For two readers r_1 and r_2 that share an overlapped interrogation region, if both of them issue a command at the same time, the tags in the overlapped region cannot resolve the commands from the readers, which is called reader–reader collision. To avoid such collisions and achieve time-efficient information collection in the multi-reader RFID systems, a coordination mechanism must be introduced to determine the schedule for activation of different readers. To address this issue, in our simulations, we divide the set of readers into the minimum number of subsets, where the readers in each subset can be activated at the same time without transmission collision. Each time, one subset of readers is scheduled to execute the unknown-target information collection protocol, until the sensor information of all the tags in the system are collected. This coordination mechanism could effectively reduce the interference among the readers, and it can work with all the information collection protocols we consider in the simulations.

TABLE II
EXECUTION TIME COMPARISON (IN SECONDS) WHEN $n = 20\,000$

	l (Bits)				
	1	8	16	32	64
AIC	31.7	33.3	35.2	39.7	47.6
TIC	36.2	37.7	40.1	43.8	53.1
PIC	103.0	103.2	103.4	103.7	104.5
MIC	41.2	53.5	67.5	95.6	151.8
BIC	5.9	6.6	7.4	9.0	12.1

TABLE III
EXECUTION TIME COMPARISON (IN SECONDS) WHEN $n = 50\,000$

	l (Bits)				
	1	8	16	32	64
AIC	75.1	79.0	85.0	94.2	115.7
TIC	89.1	93.3	97.1	107.3	129.2
PIC	257.5	257.9	258.4	259.3	261.2
MIC	103.1	133.8	169.0	238.9	379.5
BIC	10.9	12.7	14.7	18.6	26.4

TABLE IV
EXECUTION TIME COMPARISON (IN SECONDS) WHEN $n = 100\,000$

	l (Bits)				
	1	8	16	32	64
AIC	147.0	157.2	166.2	187.2	228.3
TIC	176.6	184.6	198.5	212.7	257.6
PIC	515.0	515.9	516.8	518.7	522.5
MIC	206.2	267.5	337.8	477.9	758.9
BIC	19.3	22.9	26.9	36.4	50.5

Tables II–IV show the execution time of each information collection protocol under different information length when n is 20 000, 50 000, and 100 000, respectively. We continue to observe that BIC has the minimum execution time among all the protocols. For example, in the case that $l = 8$ bits and $n = 20\,000$, the execution time of BIC is only 6.6 s, which is 20% of the time required by AIC, 17.5% of the time required by TIC, 6.4% of the time required by PIC, and 12.3% of the time required by MIC. Even when the total number of tags in the system reaches 100 000 and the information is 64 bits long, BIC could still achieve the information collection within 1 min as shown in Table IV, which indicates that BIC is also very efficient for multi-reader RFID systems.

VII. Conclusion

This paper studies the problem of collecting information from an unknown subset of tags in a sensor-enabled RFID system. Different from information collection where the reader has prior knowledge about its set of target tags, for unknown-target information collection, the reader must first figure out all target tags before reading information from them. We begin with two categories of solutions that are derived from the existing ID-collection protocols and information collection protocols and demonstrate that they cannot efficiently solve the unknown-target information collection problem due to significant time overhead for identifying the target tags. We then propose a novel solution called the Bloom-filter-based Information Collection protocol (BIC). A Bloom filter representing the target tag set is distributively constructed and transmitted to the reader, which drastically reduces the time overhead and improves the performance of the information collection protocol. Extensive simulations show that our protocol significantly outperforms other solutions.

References

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, Second ed. New York, NY, USA: Wiley, 2003.

[2] D. K. Klair, K. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *IEEE Commun. Surveys Tutorials*, vol. 12, no. 3, pp. 400–421, Apr. 2010.

[3] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. Pervasive*, Zurich, Switzerland, Aug. 2002.

[4] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed ALOHA for multiple RFID objects identification," *IEICE Trans. Commun.*, vol. E88-B, no. 3, pp. 991–999, Mar. 2005.

[5] S. R. Lee, S. D. Joo, and C. W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proc. MOBIQUITOUS*, San Diego, CA, USA, Jul. 2005, pp. 166–172.

[6] J. R. Cha and J. H. Kim, "Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID system," in *Proc. IEEE CCNC*, Las Vegas, NV, USA, Jan. 2006, pp. 768–772.

[7] V. Sarangan, M. R. Devarapalli, and S. Radhakrishnan, "A framework for fast RFID tag reading in static and mobile environments," *Comput. Netw.*, vol. 52, no. 5, pp. 1058–1073, Apr. 2008.

[8] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, "Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems," in *Proc. ISLPED*, Newport, CA, USA, Aug. 2004, pp. 357–362.

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

[10] N. Bhandari, A. Sahoo, and S. Lyer, "Intelligent query tree (IQT) protocol to improve RFID tag read efficiency," in *Proc. ICIT*, Orissa, India, Dec. 2006, pp. 46–51.

[11] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. ACM MobiCom*, Los Angeles, CA, USA, Sep. 2006, pp. 322–333.

[12] M. Kodialam, T. Nandagopal, and W. Lau, "Anonymous tracking using RFID tags," in *Proc. IEEE INFOCOM*, Anchorage, AK, USA, May 2007, pp. 1217–1225.

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

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

[15] M. Shahzad and A. X. Liu, "Every bit counts: Fast and scalable RFID estimation," in *Proc. ACM MobiCom*, Istanbul, Turkey, Aug. 2012, pp. 365–376.

[16] C. Qian, H. Ngan, and L. Hu, "Cardinality estimation for large-scale RFID systems," in *Proc. PerCom*, Hong Kong, Mar. 2008, pp. 30–39.

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

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

[19] R. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast identification of the missing tags in a large-scale RFID system," in *Proc. IEEE SECON*, Salt Lake City, UT, USA, Jun. 2011, pp. 278–286.

[20] Y. Zheng and M. Li, "Fast tag searching protocol for large-scale RFID systems," in *Proc. IEEE ICNP*, Vancouver, BC, Canada, Oct. 2011, pp. 363–372.

[21] 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.

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

[23] R. Want, "Enabling ubiquitous sensing with RFID," *Computer*, vol. 37, no. 4, pp. 84–86, Apr. 2004.

[24] M. Miura, S. Ito, R. Takatsuka, T. Sugihara, and S. Kunifuji, "An empirical study of an RFID mat sensor system in a group home," *J. Netw.*, vol. 4, no. 2, pp. 133–139, Apr. 2009.

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

[26] Y. Qiao, S. Chen, T. Li, and S. Chen, "Energy-efficient polling protocols in RFID systems," in *Proc. ACM MobiHoc*, Las Vegas, NV, USA, Sep. 2011, Art. no. 25.

[27] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: An anticollision algorithm for the reader collision problem," in *Proc. IEEE ICC*, Anchorage, AK, USA, May 2003, pp. 1206–1210.

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

[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*, Shanghai, China, Apr. 2011, pp. 3092–3100.

[30] L. Xie, B. Sheng, C. Tan, H. Han, Q. Li, and D. Chen, "Efficient tag identification in mobile RFID systems," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[31] A. Juels, "RFID security and privacy: A research survey," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 381–394, Feb. 2006.

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

[33] L. Lu, J. Han, L. Hu, Y. Liu, and L. M. Ni, "Dynamic key-updating: Privacy-preserving authentication for RFID systems," in *Proc. IEEE PerCom*, White Plains, NY, USA, Mar. 2007, pp. 13–22.

[34] L. Lu, J. Han, R. Xiao, and Y. Liu, "ACTION: Breaking the privacy barrier for RFID systems," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1953–1961.

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

[36] Philips Semiconductors, "I-CODE UID smart label IC functional specification," Jan. 2004 [Online]. Available: http://www.nxp.com/acrobat_download/other/identification/SL092030.pdf

[37] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet Math.*, vol. 1, no. 4, pp. 485–509, Mar. 2004.

[38] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[39] EPCglobal, "EPC radio-frequency identity protocols Class-1 Generation-2 UHF RFID protocol for communications at 860 MHz–960 MHz," Oct. 2008 [Online]. Available: http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2/uhfc1g2_1_2_0-standard-20080511.pdf

[40] L. G. Roberts, “ALOHA packet system with and without slots and capture,” *Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28–42, Apr. 1975.

[41] J. Massey, “Collision resolution algorithms and random-access communication,” University of California, Los Angeles, CA, USA, Rep. UCLA-ENG-8016, 1981.

[42] T. F. L. 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.

Hao Yue (S’11) received the B.Sc. degree in telecommunication engineering from Xidian University, Xi’an, China, in 2005, and is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of Florida, Gainesville, FL, USA.

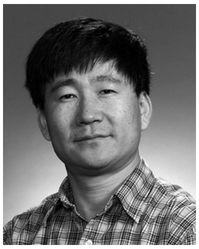
His research interests include wireless networks and mobile computing, cyber-physical systems, and security and privacy in distributed systems.

Chi Zhang (S’06–M’11) received the B.E. and M.E. degrees in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2011.

He is now an Associate Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His current research interests include network and distributed system security, wireless networking, and mobile computing.

Miao Pan (S’07–M’12) received the B.Sc. degree in electrical engineering from Dalian University of Technology, Dalian, China, in 2004, the M.A.Sc. degree in electrical and computer engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2007, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2012.

He is now an Assistant Professor with the Department of Computer Science, Texas Southern University, Houston, TX, USA. His research interests include cognitive radio communications and networking, wireless security and network economics, renewable integration in smart grids, and resource management in cloud computing.



Yuguang “Michael” Fang (S’92–M’97–SM’99–F’08) received the Ph.D. degree in systems engineering from Case Western Reserve University, Cleveland, OH, USA, in 1994, and the Ph.D. degree in electrical engineering from Boston University, Boston, MA, USA, in 1 997.

He was an Assistant Professor with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA, from 1998 to 2000. He then joined the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA, in 2000 as an Assistant Professor, got an early promotion to an Associate Professor with tenure in 2003, and to a Full Professor in 2005. He holds a University of Florida Research Foundation (UFRF) Professorship from 2006 to 2009, a Changjiang Scholar Chair Professorship with Xidian University, Xi’an, China, from 2008 to 2011, and a Guest Chair Professorship with Tsinghua University, Beijing, China, from 2009 to 2012. He has published over 250 papers in refereed professional journals and conferences.

Dr. Fang is a member of the Association for Computing Machinery (ACM). He is currently serving as the Editor-in-Chief for *IEEE Wireless Communications* and serves/served on several editorial boards of technical journals including the IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, *IEEE Wireless Communications*, and *Wireless Networks*. He was an Editor for the IEEE Transactions on Mobile Computing and currently serves on its Steering Committee. He has been actively participating in professional conference organizations such as serving as the Steering Committee Co-Chair for QShine from 2004 to 2008, the Technical Program Vice-Chair for IEEE INFOCOM 2005, Technical Program Symposium Co-Chair for IEEE GLOBECOM 2004, and a member of the Technical Program Committee for IEEE INFOCOM in 1998, 2000, and 2003 to 2010. He received the National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He is the recipient of the Best Paper Award in the IEEE International Conference on Network Protocols (ICNP) in 2006 and the IEEE TCGN Best Paper Award in the IEEE High-Speed Networks Symposium, IEEE GLOBECOM, in 2002.

Shigang Chen (M’04–SM’12) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 1993, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana–Champaign, Urbana, IL, USA, in 1996 and 1999, respectively, all in computer science.

He is an Associate Professor with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA. After graduation, he worked with Cisco Systems, San Jose, CA, USA, for three years before joining the University of Florida in 2002. He served on the technical advisory board for Protego Networks from 2002 to 2003. He published more than 100 peer-reviewed journal/conference papers. He holds 11 US patents. His research interests include computer networks, Internet security, wireless communications, and distributed computing.

Dr. Chen is a member of the Association for Computing Machinery (ACM) and an Associate Editor for the IEEE/ACM Transactions on Networking, *Computer Networks*, and the IEEE Transactions on Vehicular Technology. He served in the Steering Committee of IEEE IWQoS from 2010 to 2013. He received the IEEE Communications Society Best Tutorial Paper Award in 1999 and the NSF CAREER Award in 2007.