Exploring Tag Distribution in Multi-reader RFID Systems

Feng Zhu, Bin Xiao, *Senior Member, IEEE,* Jia Liu, *Member, IEEE,* Bin Wang, Qingfeng Pan, Li-jun Chen

Abstract—Radio Frequency Identification (RFID) brings a revolutionary change in a range of applications by automatically monitoring and tracking products. With the proliferation of RFID-enabled applications, multiple readers are needed for ensuring the full coverage of numerous RFID tags. In this paper, we focus on the tag distribution problem in multi-reader RFID systems. The problem is to fast identify the tag set beneath each reader, which is a fundamental premise of efficient product inventory and management. Only with such tag set information can we localize specific tags in a reader and expedite the tag query information collection. As an RFID system usually contains a large number of tags and multiple readers, the traditional solution to identify tags by individual readers is highly time inefficient. We propose an Inference-Based protocol (IB) that identifies the tag distribution based on information inference rules and the aggregated physical signals to improve operational efficiency. In our protocol, three kinds of inference rules based on internal information reported by a single reader, external information shared by multiple readers, and history information retained by the system are fully exploited to infer tag distribution. With these rules all readers can cooperatively work together and quickly obtain the tag distribution in the system. We also build a prototype RFID system using the USRP-based reader and WISP programmable tags, and then implement the IB protocol. The experimental results and extended simulations show that IB outperforms the state-of-the-art protocols.

Index Terms—RFID system; tag distribution identification; time efficiency; multiple readers; topological relations

1 Introduction

Radio Frequency Identification (RFID) brings a revolutionary change in a range of applications, such as inventory control [1], [2], [3], supply chain logistics [4], [5], [6] and object tracking [7], [8]. With the proliferation of these RFID-enabled applications, deploying multiple readers is essential to ensure the full coverage of numerous RFID tags. The problem presented in this paper is how to quickly explore which tags are beneath which readers, with the knowledge of all tag IDs a priori. We refer to such a problem simply as the *tag distribution problem*, which plays a crucial role in facilitating warehouse management and inventory control in multi-reader RFID-enabled applications.

To understand its practical importance, let us consider a common scenario. In an RFID-assisted warehouse, when tagged products move through the gate equipped with RFID readers (as shown in Fig.1), tag IDs can be automatically scanned by the readers and stored on the server side. However, even with this global information of tag IDs, we are unware of the tag distributions after these tagged objects are stocked under different storage zones. This makes each reader have to take the global tag population as the input to execute various inventory operations, e.g., missing/unknown tag identification [9], [10], [11], information collection [12], [13], tag searching [2], [14], [15], greatly lowering the inventory efficiency. Especially when the number of tags in the region of interest is much smaller than that of all tags in the warehouse, the performance gap will be further widened as much more extra overhead is required for separating tags of interest from others. In contrast, knowing the tag distribution can help quickly zoom in a local surveillance region and complete the inventory operations based on a much smaller tag subset, greatly improving the inventory performance. For example, consider a warehouse deployed 50 readers, each of which covers 2,000 tags on average. The execution time of missing tag identification with Protocol-3 [9] decreases sharply from 17.8s to only 1.8s, producing a 89.8% performance gain. Therefore, exploring the tag distribution is a fundamental premise of efficient inventory and management under multi-reader RFID systems.

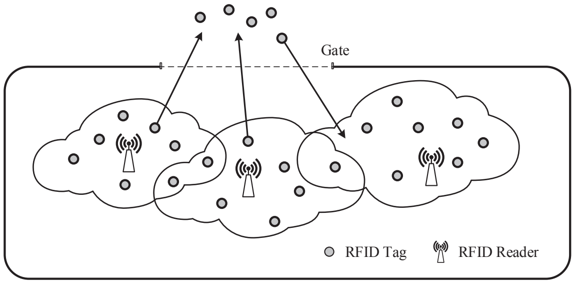


Fig. 1. Problem statement of exploring tag distribution.

The tag distribution problem has not been thoroughly investigated yet. An intuitive solution to tackle this problem is tag identification [16], [17], which sequentially collects IDs from tags by each reader. This approach is simple but collecting tedious tag IDs is time-consuming. Another solution is to transplant the missing tag identification technique to the tag distribution problem. Specially, each reader individually execute the missing tag identification

- *Feng Zhu, Jia Liu, Bin Wang, Qingfeng Pan and Li-jun Chen are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China. E-mail:{zhufeng, liujia, wangbin, qfpan}@smail.nju.edu.cn, chenlj@nju.edu.cn.*
- *Bin Xiao is with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China. E-mail: csbxiao@comp.polyu.edu.hk.*
- *Li-jun Chen is the corresponding author.*

protocols by taking overall tags as the input and treating tags outside its coverage as missing tags. The frame sizes in these protocols, however, are proportional to the number of overall tags, hard to meet real-time application requirements. The performance is even worse than that of tag identification when each reader only covers a small portion of tags. Besides above approaches, the Bloom filter based information collection protocol (BIC) [12] can also be tailored to the tag distribution problem. In BIC, the reader identifies tags in its interrogation region by constructing and transmitting the Bloom filters. However, the use of Bloom filter suffers from false positives. To remove the false positives, BIC needs to execute tag identification, resulting in extra communication overhead. In conclusion, the low efficiency of existing approaches is essentially due to the individual work of each reader instead of the cooperative work of multiple readers.

In this paper, we propose an efficient protocol called Inference-Based protocol (IB) to identify the tag distribution. Unlike previous work, the performance gain of IB benefits from two new design principles: 1) Instead of using only the application-level information, the inter-round physical signals are exploited in IB to extract more tag distribution information. 2) Instead of working individually, all readers in IB cooperate with each other to accelerate the identification process. The above mechanisms greatly reduce transmission overhead during the identification process and overcome the drawbacks of existing solutions.

IB infers the tag distribution based on three kinds of information, which are *internal information*, *external information* and *history information*. Internal information is the aggregated physical signals identified by a single reader. Upon obtaining such information, a reader can check whether a tag replies or not and thus infer its presence or absence. External information is the shared local tag distribution among readers, such as the topology of the system. As a result, the reader can cooperate with each other to accelerate the identification of its local tag distribution. History information is the aggregated physical signals gathered in previous rounds for each reader. On updating the local tag distribution in a round, more distribution information can be inferred by re-utilizing the aggregated signals retained in the current round and previous rounds. Based on these three kinds of information, we then formulate six inference rules to identify the tag distribution. We comprehensively analyze the protocol, including the investigation of protocol efficiency, the impact on execution time with varied parameters and the setting of optimal frame size.

We implement a prototype system and validate IB design based on the Universal Software Radio Peripheral (USRP) platform and the Intel Wireless Identification and Sensing Platform (WISP). Furthermore, we conduct extensive simulations to evaluate our protocol performance. The simulation results show that IB outperforms the state-of-the-art protocols [9], [12], [18], [19], [20].

The paper makes the major contributions as follows:

- We are the first to formulate the tag distribution problem. With the tag distribution, each reader in the system can take the local tag subset rather than the global tag population as the input to execute various inventory protocols, greatly improving time and energy efficiency.
- We propose a new idea of protocol design that identifies the tag distribution based on information inference rules and the physical signals of RFID tags to improve efficiency. With these rules all readers can cooperatively work together and quickly obtain the tag distribution.
- We build a real prototype system using USRP and WISP platforms to validate our protocol. Besides, we conduct extensive simulations to evaluate the protocol performance. Our protocol produces great performance improvement compared with the existing advanced work.

The rest of this paper is organized as follows. In Section 2, we overview related work. Section 3 introduces the system model and gives the problem description. In Section 4, we describe the Inference-Based protocol (IB) that identifies the tag distribution based on six inference rules. In Section 5, we comprehensively analyze the protocol efficiency. In Section 6, we propose a cardinality estimation scheme, which can have no extra overhead to estimate tag set cardinality. We build a prototype system with USRP-based reader and WISP tags, and then implement IB protocol in Section 7. Section 8 evaluates the performance with the state-of-the-art protocols and investigates the efficiency of IB. Finally, Section 9 concludes this paper.

2 Related Work

The tag distribution problem can be naturally solved with tag identification approaches. Existing tag identification protocols fall into two categories: Tree-based protocols [21], [22] and Aloha-based protocols [16], [17]. In Tree-based protocols, the reader applies a dynamic ID prefix of tag IDs to progressively split a tag set into smaller subsets until only one tag is left. This process is iteratively executed until all tags are successfully identified. In Aloha-based identification protocols, the reader carries out a slotted frame and each tag individually picks a slot to reply. A tag can successfully transmit its ID only when its slot has no other tags. The left tags which fail to report their IDs will move to the next slotted frame. Although the identification protocols can be used to address the tag distribution problem, it needs to transmit tag IDs and does not make use of the prior knowledge of all tag IDs, which is time-consuming.

Techniques for identifying missing tags can be transplanted to solve the tag distribution problem. The target of the missing tag identification is to monitor a set of RFID tags and identify the complete missing ones. By treating the tags out of the reader as missing tags, the missing tag identification schemes can be adopted to identify the tag distribution. Protocol-3 proposed in [9] is a deterministic missing tag identification scheme which significantly reduces the missing tag identification time by more efficiently scheduling multiple readers. In [19], a Slot Filter-based Missing Tag Identification (SFMTI) protocol is proposed to reconcile some expected collision slots into singleton slots and filter out the expected empty slots as well as the unreconcilable collision slots, thereby improving time-efficiency in missing tag identification. However, the performance is even worse than that of tag identification approaches when each reader only covers a small portion of all tags.

Besides above approaches, the Bloom filter based information collection protocol (BIC) [12] can also tackle the tag distribution problem, which was initially designed for collecting sensor-augmented tag information. In BIC, the reader identifies the tags in its interrogation region by using the Bloom filters. A Bloom filter representing the interrogated tag set is distributively constructed and transmitted to the reader. However, the use of Bloom filter suffers from false positives. To remove the false positives, BIC needs to collect tedious tag IDs, resulting in extra communication overhead. Due to the individual work of each reader instead of the cooperative work of multiple readers, the existing approaches are essentially of low efficiency.

3 Preliminary

3.1 System Model

We consider a large-scale RFID system with multiple readers, a massive number of tags and a backend server. The tags attached to items are under the different surveillance regions covered by different readers. After communicating with the tags, the reader transmits tag information to a backend server, which provides powerful computation ability to process such data. Similar to previous work [9], [12], [20], [23], [24], we assume that the server has the knowledge of all tag IDs in the system a priori.

Considering the tagged objects move around the system, the tag distribution is likely to dynamically change. Since our protocol can identify the tag distribution in real time, we can simply treat the dynamic environment as a static one within a short period of time. Even if few tags are incorrectly classified due to frequent tag motion, their distribution will be updated in the next execution of our protocol.

In the RFID system, the communication between the reader and tags initiates with a high power continuous wave (CW) which energizes RFID tags. The tags transmit their signals by reflecting back the CW using on-off keying modulation: a binary '1' or '0' represents the presence or absence of CW [25]. Due to the low data rate, the bit-level clock synchronization of tags can be achieved in practice [20], [26].

Complying with the EPCglobal Gen-2 standard, the communication between the readers and tags follows the Reader-Talk-First model. Namely, the tag talks only if receiving the reader's commands. A reader initializes each round of our protocol by sending a request. On receiving the order, tags then backscatter signals. Besides, the reader collision will occur when two or more readers attempt to communicate with a tag concurrently, since the mixed signals cannot be correctly decoded at the tag side. Many existing reader anti-collision schemes [27], [28] have been proposed to achieve dynamical reader schedule. We can resort to these works to avoid the communication collision among multiple readers.

3.2 Problem Formulation

Consider a set of readers $\mathcal{M} = \{r_1, ..., r_m\}$ and a set of tags $\mathcal{N} = \{t_1, ..., t_n\}$ that are deployed in the system, where m is the number of readers and n is the number of tags. Symbol $\mathcal{N}_i$ represents the set of tags under the surveillance region covered by the reader $r_i (i \in [1, m])$. Due to the overlapping area, a tag may be covered by more than one reader. Then we have $\mathcal{N} = \bigcup_{i\in[1,m]} \mathcal{N}_i$ and $|\mathcal{N}_i \cap \mathcal{N}_j| \geq 0$.

As Fig. 1 depicts, the target of the tag distribution problem is to identify which tags are beneath which readers, i.e., identifying the set $\mathcal{N}_i$ for the reader r_i $(i \in [1, m])$. Since all the tag IDs in $\mathcal{N}$ are known in advance, how to take full use of this information is the key to improve protocol performance.

Besides, the topology of readers is another information that can be used to assist the identification of tag distribution. It is formally a graph $\mathcal{G} = (\mathcal{M}, \mathcal{E})$, where $\mathcal{M}$ is the set of readers and $\mathcal{E}$ is the set of edges each connecting two readers which have overlapping monitor area. We refer to these two overlapped readers as adjacent (neighboring) readers. Clearly, if the readers r_i and r_j are adjacent (neighboring), $(r_i, r_j) \in \mathcal{E}$ holds. Otherwise, if two readers r_i and r_j are not adjacent (non-neighboring), $(r_i, r_j) \notin \mathcal{E}$ holds. We denote the set of r_i's neighboring readers as $\Gamma(r_i)$ and the set of r_i's non-neighboring readers as $\Upsilon(r_i)$. The number of r_i's adjacent readers is referred to as $d(r_i)$, which is actually the degree of the node r_i in $\mathcal{G}$. Clearly, the reader topology relies on only the practical deployment, regardless of the tag distribution. It can be easily observed when an RFID system is set up. In our protocol design, both the tag IDs and the reader topology will be fully taken into account to explore the tag distribution.

4 IB: Inference-Based protocol

To explore the tag distribution in an efficient way, we propose an Inference-Based protocol (IB), which mainly identifies the tag distribution by inference rules and the physical signals. After presenting the protocol design, we then analyze the parameter setting to achieve the optimal performance.

4.1 Overview

Inference-Based protocol (IB) follows two guidelines to achieve high time efficiency. First, IB can exploit the inter-round physical signals to extract more tag distribution information. Second, all readers in IB cooperate with each other to accelerate the identification process. The above mechanisms greatly reduce transmission overhead during the identification process and overcome the drawbacks of existing solutions. Consider an arbitrary tag t_i. If a reader is sure about whether t_i is under its coverage or not, we say that t_i is identified by the reader, or unidentified otherwise. The number of unidentified tags for each reader decreases after each round. The server traces the number of identified tags. When all the tags of each reader have been identified, then the protocol ends.

4.2 Inference Rules

IB identifies the tag distribution based on three kinds of information, which are internal information, external information and history information. Internal information is the aggregated physical signals identified by a single reader. When

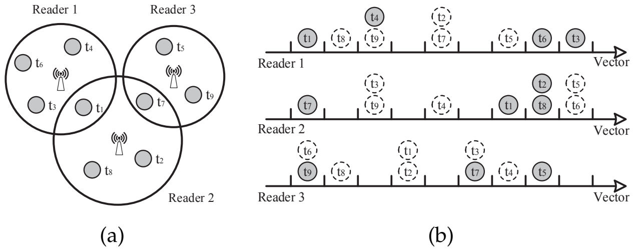


Fig. 2. An example to illustrate the tag distribution under a multi-reader scenario. (a) A tag distribution instance with 3 readers and 9 tags. (b) The mapping relationship between the tag ID and the index of a 9-bit vector.

obtaining such information, a reader can check whether a tag replies or not and thus infer its presence or absence. External information is the shared local tag distribution among readers, such as the topology of the system. As a result, the reader can cooperate with each other to accelerate the identification of its local tag distribution. History information is the aggregated physical signals gathered in previous rounds for each reader. On updating the local tag distribution in a round, more distribution information can be inferred by re-utilizing the aggregated signals retained in the current round and previous rounds. Based on the three kinds of information, we then formulate six inference rules to identify the tag distribution.

Consider an arbitrary reader r_i, $\mathcal{N}_i$ and $\mathcal{L}_i$ represent the set of tags inside and outside r_i's coverage. Thus the following equations hold:

$$\mathcal{N}_i = \mathcal{N}_{i,0} \cup \mathcal{N}_{i,1},$$
$$\mathcal{L}_i = \mathcal{L}_{i,0} \cup \mathcal{L}_{i,1},$$

where the second binary subscript indicates whether tag is identified: '1' is identified and '0' is not identified. Note the set $\mathcal{N}_{i,0}$ and $\mathcal{L}_{i,0}$ cannot be surely identified in the obtaining process, thus the server cannot exactly know what tags in the set $\mathcal{N}_i$ at the beginning.

Initially, because no tag distribution is identified, both $\mathcal{N}_{i,1}$ and $\mathcal{L}_{i,1}$ are the empty set, that is:

$$\mathcal{N}_{i,0} \cup \mathcal{L}_{i,0} = \mathcal{N},$$
$$\mathcal{N}_{i,1} \cup \mathcal{L}_{i,1} = \varnothing.$$

After the execution of IB, the server successfully identifies all the tags in unidentified set, and put them into identified set. Then the equations hold:

$$\mathcal{N}_{i,0} \cup \mathcal{L}_{i,0} = \varnothing,$$
$$\mathcal{N}_{i,1} \cup \mathcal{L}_{i,1} = \mathcal{N}.$$

Thus the set of tags beneath appointed readers can be surely identified, i.e., $\mathcal{N}_i = \mathcal{N}_{i,0} \cup \mathcal{N}_{i,1} = \mathcal{N}_{i,1}$.

4.2.1 Example Case

For ease of presentation, we detail the six inference rules based on a toy example. Consider the system provided in Fig. 2(a), where $m = 3$ and $n = 9$. Both the reader r_1 and r_3 are adjacent to the reader r_2. The set of total tags in the system is $\mathcal{N} = \{t_1, t_2, t_3, ..., t_8, t_9\}$.

Initially, the reader broadcasts a request with a random number. Each tag chooses an index by hashing its ID with the received random number. Only in the slot that the tag picks, it replies. Otherwise, it keeps silent. Based on the tag ID database, the server knows exactly which bit a tag chooses under each random number. However, from each reader's view, only a subset of tags reply to the reader. As shown in Fig. 2(b), only the tags with solid line reply in the actual system. Each tag's response forms a logical bit vector, where response represents bit '1' and silence represents bit '0', as shown in Table 1. On receiving vectors, each reader uses a logical 'OR' operation and constructs a reader vector.

TABLE 1
Procedure of Vector Construction

	Reader 1	Reader 2	Reader 3
Tag Vector	t_1: 100000000 t_3: 000000001 t_4: 001000000 t_6: 000000010	t_1: 000000100 t_2: 000000010 t_7: 100000000 t_8: 000000010	t_5: 000000010 t_7: 000001000 t_9: 100000000
Reader Vector	r_1: 101000011	r_2: 100000110	r_3: 100001010

TABLE 2
Identification Results of IB in the First Round

Reader	Tag								
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
r_1	$\circ_2$	$\times_1$	$\circ_2$	$\circ_3$	$\times_1$	$\circ_2$	$\times_1$	$\times_1$	-
r_2	$\circ_2$	$\circ_3$	$\times_1$	$\times_1$	$\times_1$	$\times_1$	$\circ_2$	$\circ_3$	$\times_1$
r_3	$\times_1$	$\times_1$	$\times_4$	$\times_1$	$\circ_3$	$\times_4$	$\circ_5$	$\times_1$	$\circ_5$

Note the target of IB is to gain the tag distribution in the system. We can easily map the relationship of the tag distribution to a two-dimensional matrix with 3×9 as illustrated in the Table 2. The symbol '$\circ$' represents the relation that a tag is covered by a reader. On the contrary, '$\times$' represents the relation that a tag is not covered by a reader and '-' represents the relation that needs to be inferred. The subscript represents the rule number used to infer the tag distribution. The tag distribution is identified until matrix is crammed.

4.2.2 Rules Based on Internal Information

We first introduce the inference rules based on internal information. To infer the tag distribution under reader r_i's coverage, the server generates a virtual reader vector of r_i by assuming all tags in $\mathcal{N}$ are under r_i's coverage. After collecting the actual reader vector, the server can infer the tag distribution by observing the the differences between the virtual vector and the actual vector.

Rule 1: A tag t_k can be classified into $\mathcal{L}_{i,1}$, when $t_k \in \mathcal{L}_{i,0}$ holds and no tag is from both $\mathcal{N}_{i,0}$ and $\mathcal{N}_{i,1}$.

If no tag is from the set $\mathcal{N}_{i,0}$ and $\mathcal{N}_{i,1}$, r_i will receive the bit '0' from tags. In the meantime, if there is also at least one tag from the set $\mathcal{L}_{i,0}$, the server assures the tags supposed to reply are out of the r_i region. Then the server could successfully add these tags into $\mathcal{L}_{i,1}$. We take the reader r_1 for an instance in the example case. In the second bit, only t_8 is supposed to reply '1'. But the second bit of the reader vector '100000011' equals '0'. Then the server could identify $t_8 \in \mathcal{L}_{1,1}$. The same case exists for the fifth bit. Because there are two tags t_2 and t_7 supposed to reply in the fifth bit, with that the fifth bit of the reader vector equals '0', the server can also infer $t_2, t_7 \in \mathcal{L}_{1,1}$.

Rule 2: A tag t_k can be classified into $\mathcal{N}_{i,1}$, when $t_k \in \mathcal{N}_{i,0}$ holds and no tag is from the set $\mathcal{L}_{i,0}$ and $\mathcal{N}_{i,1}$.

If one tag is from the set $\mathcal{N}_{i,0}$, r_i will receive the bit '1' from tags. If the server knows exactly only one tag replies '1' at the bit, it will easily know this tag is covered by the reader r_i. Obviously, no matter how many tags from the set $\mathcal{L}_{i,1}$ select the bit, the reader r_i receives no reply from them. Therefore, if there is one tag from the set $\mathcal{N}_{i,0}$ and no tag from the set $\mathcal{L}_{i,0}$ and $\mathcal{N}_{i,1}$, the server could add the tag into $\mathcal{N}_{i,1}$. Here, we also take the reader r_1 for an instance in the example case. Based on the tag ID database, only the tag t_1 is assumed to reply '1' at the first bit. Upon checking the reader vector, the first bit is also '1'. The server knows tag t_1 replies, i.e. $t_1 \in \mathcal{N}_{1,1}$. In the same way, $t_3, t_6 \in \mathcal{N}_{1,1}$ can be known. But for the third bit, both tag t_4 and t_9 are supposed to transmit '1'. With receiving '1' in the corresponding bit, the server could not tell which tag(s) actually reply.

4.2.3 Rules Based on External Information

To further accelerate the identification process, IB makes the readers cooperate with each other by sharing their vectors to narrow down the each searching set as the following rule.

Rule 3: A tag t_k can be added to $\mathcal{N}_{i,1}$, when the tag has been identified absent from other $m-1$ readers.

Consider an arbitrary tag t_j, the relation $t_j \in \mathcal{L}_{k,1}$ exists. The server easily infers $t_j \in \mathcal{N}_{i,1}$. For example, the eighth bit of the r_2 aggregated vector is '1', possible replying from both t_2 and t_8 hinders the distribution obtaining. The server is unable to add t_2 or t_8 to $\mathcal{N}_{2,1}$. But consider both the reader r_1 and r_3 receive no reply from t_2. Thus the server could assure t_2 in the region of r_2.

Rule 4: A tag t_k can be classified into $\mathcal{L}_{j,1}$, when $t_k \in \mathcal{N}_{i,1}$ and $r_j \in \Upsilon(r_i)$ both hold.

The membership $t_k \in \mathcal{N}_{i,1}$ holds means that the tag t_k is under the reader r_i's coverage, which includes two cases. In the first case, the tag t_k is in the overlapping area between the reader r_i and any other neighboring readers of r_i, i.e., $t_k \in \bigcup_{r_j \in \Gamma(r_i)} (\mathcal{N}_i \cap \mathcal{N}_j)$ holds. In the second case, the tag t_k is reachable to only the reader r_i rather than others, i.e., $t_k \in \bigcap_{r_j \in \Gamma(r_i)} (\mathcal{N}_i - \mathcal{N}_j)$ holds.

The server cannot figure out which case of these two is correct, but it can make sure t_k is under r_i's coverage zone with the membership $t_k \in \mathcal{N}_{i,1}$. In other words, if a reader r_j is far away from r_i, i.e., r_j is not the neighbor of r_i ($r_j \in \Upsilon(r_i)$), we can infer that t_k must not be covered by r_j with certainty. Namely, t_k belongs to $\mathcal{L}_{j,1}$. As t_k has been identified, the server can update $|\Upsilon(r_i)|$ relations for it. In the example case, because both t_3 and t_6 are under r_1's coverage zone, the server can infer $t_3 \in \mathcal{L}_{3,1}$ and $t_6 \in \mathcal{L}_{3,1}$.

4.2.4 Rules Based on History Information

As previously described, the server can update the tag distribution with Rule 1-4. But the reader vectors gathered in current round and previous rounds can also be utilized to infer more relations, which highly reduces the total execution rounds. Thus, two rules can be used to mine more information.

Rule 5: After updating the new tag distribution, the server can infer more relations with Rule 1-4 by checking current vectors again.

By executing Rule 1-4 once, some bits in the vector can be used to infer the tag distribution, while others cannot, such as the first bit and sixth bit of the reader r_3. To avoid this waste, we take a closer look at the tag distribution and find that a new tag distribution generates after the one-round execution of Rule 1-4. Hence, if the server takes the new distribution as the input and repeats the Rule 1-4 again, the useless bits in the previous execution are likely to be useful under the newly updated distribution.

We take the vector of r_3 for example. During the first execution of Rule 1-4, both tag t_3 and t_7 are supposed to transmit '1' in the sixth bit. Therefore, there are two possible results. One is $t_3 \in \mathcal{L}_{3,1}, t_7 \in \mathcal{N}_{3,1}$ and the other is $t_7 \in \mathcal{L}_{3,1}, t_3 \in \mathcal{N}_{3,1}$. According to the sixth bit, the server cannot determine whether t_3 and t_7 are under the reader r_3's coverage. However, after the first execution of Rule 1-4, the relation $t_3 \in \mathcal{L}_{3,1}$ is available for the server. Then the reader can take these newly updated tag distribution as the input and run Rule 1-4 again, which will generate the relation $t_7 \in \mathcal{N}_{3,1}$.

Rule 6: After updating the new tag distribution, the server can infer more relations with Rule 1-4 by checking the vectors in previous rounds again.

Because the example case only executes IB in one round, the server has no previous vectors. But after several rounds, the server could also infer new relations to update the tag distribution as Rule 5.

We denote the total overhead of r_i in the current round as F_i, which is equal to $F_{i,[1-4]} + F_{i,[5]} + F_{i,[6]}$, where $F_{i,[1-4]}$, $F_{i,[5]}$ and $F_{i,[6]}$ represent the overhead of Rule 1-4, Rule 5 and Rule 6, respectively. Assume the optimal vector size for the reader r_i in the k-th round is $f_{i,k}$. We have $F_{i,[1-4]} = f_{i,k}$, $F_{i,[5]} = f_{i,k}$ and $f_{i,[6]} = \sum_{j=1}^{k} f_{i,j}$. Therefore, the overhead of distribution update in the current round is $F_i = F_{i,[1-4]} + F_{i,[5]} + F_{i,[6]} = 2f_{i,k} + \sum_{j=1}^{k} f_{i,j}$. Considering m readers, the total computation overhead is $F = \sum_{i=1}^{m} F_i = 2\sum_{i=1}^{m} f_{i,k} + \sum_{i=1}^{m} \sum_{j=1}^{k} f_{i,j}$.

4.3 Protocol Description

The Inference-Based protocol (IB) contains multiple rounds. Each round contains two communication steps: the reader-to-tag broadcasting step and the tag-to-reader replying step. To initialize the reader-to-tag broadcasting step, the server generates optimal parameters to achieve high identification efficiency. The parameters consist of the random number and the frame size. Here, for an arbitrary reader i, we denote the random number and the frame size as d_i and f_i which are different from other readers in each round. We will detail how to set the optimal frame size f_i in the following section. The server generates the server array $\mathbf{A}_i$ based on the parameter and the database. The symbol $\mathbf{A}_i[k]$ means the set of tags supposed to reply '1' at the k-th bit. Then the readers in the system broadcast each pre-calculated parameter pair $\langle d_i, f_i \rangle$ to tags.

In the tag-to-reader replying step, benefiting the efficiency of identification process, each tag sends a vector simultaneously. As previously mentioned, IB also utilizes the synchronized physical layer transmissions to distributively construct each tag replying vector and efficiently gathers the distribution with information aggregated from tags. Consider the tag under r_i's coverage zone, after receiving the parameter pair $\langle d_i, f_i \rangle$, it calculates the f_i-bit tag vector. The tag sets all the bits to '0' except that the bit of index $\mathcal{H}(ID, d_i)\ mod\ f_i$ is '1'. All the tags then send the vectors concurrently to the reader.

At the reader, the gathering of physical signals is equivalent to binary 'OR' operation. The reader r_i then generates the reader vector $\mathbf{V}_i$. The symbol $\mathbf{V}_i[k]$ ranges between binary bit value '0' and '1'. Formally, for an arbitrary reader i, we denote the j^{th} replying bit as $b_{i,j}$. When the channel is an idle carrier, the reader sets $b_{i,j} = 0$. When the channel is a busy carrier, the reader sets $b_{i,j} = 1$. After the transmissions of all the vector bits, reader i combines calculated bits to construct new vector $\mathbf{V}_i = b_{i,1}b_{i,2}...b_{i,f}$. For example, if three tags transmit vectors '000010000', '001000000' and '010000000' to a reader respectively. Then the reader constructs the vector $\mathbf{V}_i =' 011010000'$.

Therefore, if the system has m readers, the server can collect m vectors $\mathbf{V}_i$. Obviously, the server exactly knows which bit the tag in the system is supposed to select. Upon gathering the array $\mathbf{A}_i$ with the vector $\mathbf{V}_i$, the server is able to infer the tag distribution with the Algorithm 1.

```
Algorithm 1: Tag Distribution Inference
  Input: $\mathbf{V}_i$, $\mathbf{A}_i$, $\mathcal{N}_{i,1}$, $\mathcal{L}_{i,1}$ for $r_i$ ($i \in [1, m]$).
  Output: Updated $\mathcal{N}_{i,1}$, $\mathcal{L}_{i,1}$ for $r_i$ ($i \in [1, m]$).
  for $i = 1 \to m$ do
      for $j = 1 \to$ LENGTH($\mathbf{V}_i$) do
          if $\mathbf{V}_i[j] =' 0'$ then
              $\mathcal{L}_{i,1} = \mathbf{A}_i[j] \cup \mathcal{L}_{i,1}$
          else if $\mathbf{V}_i[j] =' 1'$ and $|\mathbf{A}_i[j] \setminus \mathcal{L}_{i,1}| = 1$ then
              $\mathcal{N}_{i,1} = \mathbf{A}_i[j] \setminus \mathcal{L}_{i,1} \cup \mathcal{N}_{i,1}$
              for $r_k \in \Upsilon(r_i)$ do
                  $\mathcal{L}_{k,1} = \mathbf{A}_i[j] \setminus \mathcal{L}_{i,1} \cup \mathcal{L}_{k,1}$
  for $j = 1 \to n$ do
      if $r_i, t_j \in \mathcal{L}_{k,1}$ ($k \in [1, m], k \neq i$) then
          $\mathcal{N}_{i,1} = \mathcal{N}_{i,1} \cap \{t_j\}$
```

After the inference of the tag distribution with current vector, the server can update more tag distribution information based on the Algorithm 2.

```
Algorithm 2: Accelerating Tag Distribution Inference
  Input: Previous and current vectors for $r_i$ ($i \in [1, m]$).
  Output: Updated tag distribution.
  for $i = 1 \to m$ do
      Check the current $\mathbf{V}_i^{current}$ with Alg 1.
      for $j = 1 \to CurrentRoundIndex$ do
          Check the $j^{th}$ previous $\mathbf{V}_i^j$ with Alg 1.
```

If the set $\mathcal{L}_{i,1} \cup \mathcal{N}_{i,1} = \mathcal{N}$, the server stops identification of the reader r_i. Until all the readers stop, the server can achieve obtaining the entire tag distribution.

5 Protocol Analysis

We first analyze the protocol efficiency, and then give the number tendency of identified relations as the frame size changes. Finally, we represent the setting of frame size that optimizes the efficiency of IB.

5.1 Performance Efficiency

As the previous discussion, the set $\mathcal{N}_{i,0}$ and $\mathcal{L}_{i,0}$ cannot be surely identified. But on optimizing the frame size f_i, the server will estimate $n_{i,0}$ and $l_{i,0}$ in each round (that will be discussed in Section 6). Here we denote $n_{i,0}$, $n_{i,1}$, $l_{i,0}$ and $l_{i,1}$ as the cardinality of $\mathcal{N}_{i,0}$, $\mathcal{N}_{i,1}$, $\mathcal{L}_{i,0}$ and $\mathcal{L}_{i,1}$ respectively.

The IB protocol is designed for accomplishing the collection with the shortest time. Here, we consider the reader r_i in an arbitrary round without loss of generality. Let λ_i be the expected number of newly identified relations and τ_i be the execution time. Thus the number of newly identified relations per unit time (denoted as Θ_i) can be represented as:

$$\Theta_i = \frac{\lambda_i}{\tau_i}. \tag{1}$$

When the server heads to a new round, without considering the rules based on history information, two kinds of information can be adopted to identify the distribution. Here we refer to the relation revealed from the internal information as *internal relation* (the expected number is denoted as φ_i). On the contrary, the relation revealed from the external information is called *external relation* (the expected number is denoted as ρ_i). The total number of newly revealed relations can be represented as:

$$\lambda_i = \varphi_i + \rho_i. \tag{2}$$

We then first analyze the expected number φ_i of internal relations revealed in a round. The internal relations identified in a round generally compromises two portions. One portion is the relation that a tag t_k is not in r_i's scope, i.e., $t_k \in \mathcal{L}_{i,1}$ (the expected number is denoted as α_i). On the contrary, the other portion is the relation that $t_k \in \mathcal{N}_{i,1}$ (the expected number is denoted as β_i). We then take the following lemmas to analyze φ_i.

Lemma 1. *For a reader r_i in an arbitrary round, the expected number α_i of revealed relations that indicates a tag is out of r_i's coverage is*

$$\alpha_i = l_{i,0}\left(1 - \frac{1}{f_i}\right)^{n_{i,0}+n_{i,1}}. \tag{3}$$

Proof. As aforementioned, by checking each bit in the vector, we are able to reveal some useful relations. Now, we first derive the expected number of useful relations revealed by analyzing one bit, which is denoted as $E[A]$. The server can infer the relation only if there is no tag from the set $\mathcal{N}_i$ and at least one tag from the set $\mathcal{L}_{i,0}$. The probability that a tag selects current bit is $\frac{1}{f_i}$, where f_i is vector size. We have the probability P_a that no tag is from the set $\mathcal{N}_i$, which is $P_a = (1 - \frac{1}{f_i})^{n_{i,0}+n_{i,1}}$. Since there are $l_{i,0}$ tags in the set $\mathcal{L}_{i,0}$ is, the probability $P_{b,j}$ that j tags are from the set $\mathcal{L}_{i,0}$ is $P_{b,j} = \binom{l_{i,0}}{j}(\frac{1}{f_i})^j(1-\frac{1}{f_i})^{l_{i,0}-j}$. Thus, by taking the two probabilities into account, we have the expected number E[A] of useful relations revealed by analyzing one bit as follows:

$$\begin{aligned} E[A] &= \sum_{j=0}^{l_{i,0}} jP_aP_{b,j} = P_a \sum_{j=0}^{l_{i,0}} jP_{b,j} \\ &= P_a \sum_{j=0}^{l_{i,0}} j \binom{l_{i,0}}{j} \left(\frac{1}{f_i}\right)^j \left(1-\frac{1}{f_i}\right)^{l_{i,0}-j} \\ &= P_a \frac{l_{i,0}}{f_i} = \frac{l_{i,0}}{f_i}\left(1-\frac{1}{f_i}\right)^{n_{i,0}+n_{i,1}} . \end{aligned} \tag{4}$$

Since the vector consists of f_i bits, the expected number of revealed relations that indicates a tag is out of r_i's coverage is $\alpha_i = f_iE[A] = l_{i,0}(1-\frac{1}{f_i})^{n_{i,0}+n_{i,1}}$. □

Lemma 2. *For a reader r_i in an arbitrary round, the expected number β_i of revealed relations that indicates a tag is under r_i's coverage is*

$$\beta_i = n_{i,0}\left(1-\frac{1}{f_i}\right)^{n_{i,0}+n_{i,1}+l_{i,0}-1} . \tag{5}$$

Proof. We denote the probability that a bit can be used to identify the relation that a tag is covered by the reader r_i as P_B. The server can reveal the relation only if one tag is from the set $\mathcal{N}_{i,0}$ and no tag is from the set $\mathcal{L}_{i,0}$ and $\mathcal{N}_{i,1}$. We denote the probability that one tag is from the set $\mathcal{N}_{i,0}$ as P_c and the probability that no tag is from the set $\mathcal{L}_{i,0}$ and $\mathcal{N}_{i,1}$ as P_d. Thus we can have the following equation:

$$P_c = \binom{n_{i,0}}{1}\frac{1}{f_i}\left(1-\frac{1}{f_i}\right)^{n_{i,0}-1} = \frac{n_{i,0}}{f_i}\left(1-\frac{1}{f_i}\right)^{n_{i,0}-1} . \tag{6}$$

Then the probability that no tag is from the set $\mathcal{L}_{i,0}$ and $\mathcal{N}_{i,1}$ can be represented as:

$$P_d = \left(1-\frac{1}{f_i}\right)^{n_{i,1}+l_{i,0}} . \tag{7}$$

Thus, multiplying Eqn (6) and Eqn (7) yields the probability P_B:

$$P_B = P_cP_d = \frac{n_{i,0}}{f_i}\left(1-\frac{1}{f_i}\right)^{n_{i,0}+n_{i,1}+l_{i,0}-1} . \tag{8}$$

The expected number β_i of the relations that indicates a tag is under r_i's coverage is $\beta_i = f_iP_B = n_{i,0}(1-\frac{1}{f_i})^{n_{i,0}+n_{i,1}+l_{i,0}-1}$. □

Corollary 1. *For a reader r_i in an arbitrary round, the expected number φ_i of the newly identified internal relations is*

$$\varphi_i = \left(1-\frac{1}{f_i}\right)^{n_{i,0}+n_{i,1}}\left[l_{i,0}+n_{i,0}\left(1-\frac{1}{f_i}\right)^{l_{i,0}-1}\right] . \tag{9}$$

Proof. Obviously, combining Eqn (3) and Eqn (5) yields the result. □

Then we analyze the expected number ρ_i of the external relations. From above discussion, we know the revealed relations of Rule 3 depend on the results of the other rules (Rule 1, Rule 2 and Rule 4), thus maximizing the efficiency of the other rules also facilitates the efficiency of Rule 3. Here we only consider the revealed external relations from Rule 4 to make a approximation.

Lemma 3. *For the reader r_i in an arbitrary round, the expected number ρ_i of newly revealed external relations is:*

$$\rho_i = n_{i,0}|\Upsilon(r_i)|\left(1-\frac{1}{f_i}\right)^{n_{i,0}+n_{i,1}+l_{i,0}-1} - \varepsilon_i, \tag{10}$$

where ε_i is the number of relations which have been revealed in previous round and it is in the interval of $[0, n_{i,0}|\Upsilon(r_i)|(1-\frac{1}{f_i})^{n_{i,0}+n_{i,1}+l_{i,0}-1}]$.

Proof. If the reader r_i knows $t_k \in \mathcal{N}_i$, each r_i's non-neighboring reader r_j ($r_j \in \Upsilon(r_i)$) can easily infer $t_k \notin \mathcal{N}_j$. Therefore, once one relation about r_i is revealed, the server can infer up to $|\Upsilon(r_i)|$ relations. From Lemma 2, we know that the expected number β_i of revealed relations that indicate a tag is under r_i's coverage is $n_{i,0}(1-\frac{1}{f_i})^{n_{i,0}+n_{i,1}+l_{i,0}-1}$. Therefore, the expected number of relations that could be inferred is $n_{i,0}|\Upsilon(r_i)|(1-\frac{1}{f_i})^{n_{i,0}+n_{i,1}+l_{i,0}-1}$. These relations fall into two types: the newly revealed and the previous revealed. Let the former be ε_i, which ranges from 0 to $n_{i,0}|\Upsilon(r_i)|(1-\frac{1}{f_i})^{n_{i,0}+n_{i,1}+l_{i,0}-1}$. Finally, subtracting the expected number of previous revealed relations yields the value of ρ_i. □

Corollary 2. *For the reader r_i in an arbitrary round, the expected total number λ_i of newly revealed relations for the reader r_i is approximately*

$$\lambda_i \approx \left(1-\frac{1}{f_i}\right)^{n_{i,0}+n_{i,1}}\left[l_{i,0}+n_{i,0}(1+|\Upsilon(r_i)|)\left(1-\frac{1}{f_i}\right)^{l_{i,0}-1}\right] . \tag{11}$$

Proof. From Eqn 2, the expected number of newly revealed relations can be represented as $\lambda_i = \varphi_i + \rho_i$. From Corollary 1 and Lemma 3, we can have $\varphi_i = (1-\frac{1}{f_i})^{n_{i,0}+n_{i,1}}[l_{i,0}+n_{i,0}(1-\frac{1}{f_i})^{l_{i,0}-1}]$ and $\rho_i = n_{i,0}|\Upsilon(r_i)|(1-\frac{1}{f_i})^{n_{i,0}+n_{i,1}+l_{i,0}-1} - \varepsilon_i$. Thus, the expected total number λ_i of newly revealed relations can be deduced as:

$$\begin{aligned} \lambda_i &= \varphi_i + \rho_i \\ &= \Psi^{n_i}(l_{i,0}+n_{i,0}\Psi^{l_{i,0}-1})+n_{i,0}|\Upsilon(r_i)|\Psi^{n_{i,0}+n_{i,1}+l_{i,0}-1}-\varepsilon_i \\ &\approx \Psi^{n_i}\left[l_{i,0}+n_{i,0}\left(1+|\Upsilon(r_i)|\right)\Psi^{l_{i,0}-1}\right], \end{aligned}$$

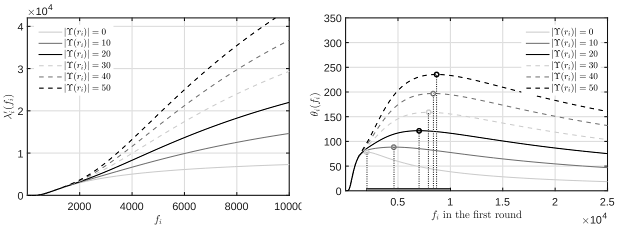
where $\Psi = 1-1/f_i$ and $n_i = n_{i,0}+n_{i,1}$. Because ε_i is related to not only the set $\mathcal{N}_{i,1}$ but also the set $\mathcal{L}_{j,0}$ ($r_j \in \Upsilon(r_i)$), and each reader runs separately, we then omit the residual ε_i to make an approximate for λ_i. □

Lemma 4. *When the parameter of frame size is set to f_i for the reader r_i, the expected execution time in a round is*

$$\tau_i = f_it_b. \tag{12}$$

Proof. The frame size of a vector is set to f_i. All the tags in the region of a reader transmit their vectors concurrently. We denote the transmission time of a bit as t_b. The total time of sending the vector equals ft_b. Thus, the expected execution time is $\tau_i = f_it_b$. □

From Corollary 2 above, we get the equation of λ_i. When the frame size is set as f_i, the total execution time is represented in Eqn (12). Thus we can utilize the equations



(a)

(b)

Fig. 3. Theoretical curve of λ_i and the optimal value for f_i. (a) Relationship between λ_i and f_i. (b) Effectiveness of optimal frame size f_i. The unit time is microsecond.

to yield the number of newly identified relations per unit time Θ_i:

$$\Theta_i = \frac{\lambda_i}{\tau_i} = \Psi^{n_i}\left[\frac{l_{i,0}}{f_i t_b} + \frac{n_{i,0}\left(1+|\Upsilon(r_i)|\right)}{f_i t_b}\Psi^{l_{i,0}-1}\right], \quad (13)$$

where $n_i = n_{i,0} + n_{i,1}$.

5.2 Relationship between λ_i and f_i

We now investigate the relationship between the expected number λ_i of identified relations and the frame size f_i.

Corollary 3. *Consider an arbitrary round of the reader r_i, with the growth of the frame size f_i, the expected number λ_i of newly identified relations increases.*

Proof. From Corollary 2, we have the representation of λ_i with the expression of f_i. To this end, solving the derivatives f_i generates as follows:

$$\frac{\partial \lambda_i}{\partial f_i} = \frac{n_{i,0}(1+|\Upsilon(r_i)|)(l_{i,0}+n_i-1)\Psi^{n_i+l_{i,0}-2} + l_{i,0}n_i\Psi^{n_i-1}}{f_i^2},$$

where $\Psi = 1 - 1/f$ and $n_i = n_{i,0} + n_{i,1}$.

The frame size f_i is greater than 0, so $\Psi \in [0, 1)$. We then have both $\Psi^{n_i+l_{i,0}-2} \geq 0$ and $\Psi^{n_i-1} > 0$. Because the number of tags under r_i's coverage zone is at least equal to 1, i.e. $n_i = n_{i,0} + n_{i,1} \geq 1$. The factor $l_{i,0} + n_i - 1$ is also greater than 0. Consider the number of non-neighbor readers $|\Upsilon(r_i)|$, it can range from $[0, n-1]$. Then we have $1 + |\Upsilon(r_i)| > 0$. The value of both $n_{i,0}$ and $l_{i,0}$ are more than 0 except for the finishing of IB in which case the reader r_i identified all the tags (i.e. $n_{i,0} + l_{i,0} = 0$). Therefore, the derivatives $\frac{\partial \lambda_i}{\partial f_i} \geq 0$ exists. The expected total number λ_i of newly revealed relations is the increasing function of the frame size f_i. □

From above corollary, we know as the increase of the frame size f_i, λ_i always increases. Fig. 3(a) shows the tendency of λ_i with growing frame size f_i, where we fix $n_{i,0} = 2,000$, $n_{i,1} = 2,000$, $l_{i,0} = 4,000$ and $l_{i,1} = 2,000$ and vary $|\Upsilon(r_i)|$ from 0 to 50. It can be clearly seen that all the curves raises as the increasing of f_i. For example, when $f_i = 3,000$ and $|\Upsilon(r_i)| = 10$, we have $\lambda_i = 4,892$. If f_i raises to $6,000$, λ_i also increases to $9,887$. As can be seen from the figure, the number of identified relations λ_i also shares the trend of steady rise, when varying $|\Upsilon(r_i)|$ and fixing f_i. When $f_i = 6,000$, λ_i is $13,665$, $17,443$ and $21,220$ with respect to $|\Upsilon(r_i)| = 20$, $|\Upsilon(r_i)| = 30$ and $|\Upsilon(r_i)| = 40$.

To improve the efficiency running in parallel mode (the readers perform the execution simultaneously), the server could set the maximum among pre-calculated frame sizes for each reader in the system. Thus the server can increase the expected total number of newly revealed relations in each round without extra time.

5.3 Setting Optimal f_i

Now we derive how to set the frame size f_i to maximize the efficiency of IB. To achieve the maximal efficiency, the number of newly identified relations per unit time Θ_i should be maximized. The Eqn (13) is hard to achieve the general algebraic solution to set the optimal frame size. However, it is a function with respect to f_i when the other parameters are foreknown. We can get the numerical solution to optimize frame size easily. The following lemma shows the interval of f_i to peruse the maximal efficiency Θ_i.

Lemma 5. *The function Θ_i has a unique maximum with respect to the frame size f_i ranges from $[n_{i,0}+n_{i,1}+1, n_{i,0}+n_{i,1}+l_{i,0}]$.*

Proof. We first compute the derivatives of Θ_i with respect to the frame size f_i as following:

$$\frac{\partial \Theta_i}{\partial f_i} = \mathfrak{U} + \mathfrak{V},$$

$$\begin{cases} \mathfrak{U} = \dfrac{l_{i,0}\Psi^{n_i-1}(n_i+1-f_i)}{{f_i}^3 t_b} \\ \mathfrak{V} = \dfrac{n_{i,0}(1+|\Upsilon(r_i)|)\Psi^{l_{i,0}+n_i-2}(l_{i,0}+n_i-f_i)}{{f_i}^3 t_b}. \end{cases}$$

We then consider the case with the interval of f_i in which Θ_i increases. Only when both $\mathfrak{U}$ and $\mathfrak{V}$ are greater than 0, the function Θ_i increases.

Due to the frame size is at least equal to 1 (i.e. $f_i > 0$), so ${f_i}^3 t_b > 0$. Here we suppose $l_{i,0} > 0$ which will be discussed later. Then considering $\Psi > 0$ and $n_i + 1 - f_i > 0$ yields $1 < f < n_i + 1$. Only if $f_i \in (1, n_i+1)$, the part $\mathfrak{U} > 0$ exists.

Due to the frame size f_i at least equals 1 (i.e. $f_i > 0$), so ${f_i}^3 t > 0$. With $|\Upsilon(r_i)| \geq 0$, we also have $1 + |\Upsilon(r_i)| > 0$. It is no meaningful to calculate the frame size f_i for both $n_{i,0} = 0$ and $l_{i,0} = 0$. We only consider the case with $n_{i,0} > 0$. Thus when $\mathfrak{V} > 0$, it is equivalent to $\Psi^{l_{i,0}+n_i-2}(l_{i,0}+n_i-f_i) > 0$, that is

$$\begin{cases} \Psi = 1 - \dfrac{1}{f_i} > 0 \\ l_{i,0} + n_i - f_i > 0 \end{cases} \Rightarrow f_i \in (1, l_{i,0} + n_i).$$

At this end, combining the interval results of $\mathfrak{U}$ and $\mathfrak{V}$ yields the increasing interval of f_i is $(1, l_{i,0}+1)$. In the same way of calculation with the increasing interval of f_i, the decreasing interval of f_i is $f_i \in (l_{i,0} + n_i, \infty)$.

Thus, it is easily to know Θ_i' has a unique maximum with respect to the frame size f_i ranges from $[n_{i,0} + n_{i,1} + 1, n_{i,0} + n_{i,1} + l_{i,0}]$. □

We then evaluate the effectiveness of optimal frame size f_i for the reader r_i in a round. Fig. 3(b) shows the identification efficiency of λ_i with growing frame size f_i, where we fix $n_{i,0} = 2,000$, $n_{i,1} = 0$, $l_{i,0} = 8,000$ and $l_{i,1} = 0$ and vary $|\Upsilon(r_i)|$ from 0 to 50. The gray bar attached to the x-axis shows the interval of f_i in which Θ_i has

TABLE 3
Optimal Frame Size in the First Round

$n_{i,0}$	$l_{i,0}$						
	1,000	2,000	3,000	4,000	5,000	6,000	7,000
1,000	1,320	1,166	1,058	1,020	1,007	1,003	1,001
3,000	3,696	3,950	3,958	3,837	3,665	3,495	3,354
5,000	5,808	6,291	6,539	6,624	6,595	6,491	6,341
7,000	7,860	8,468	8,874	9,121	9,246	9,276	9,234

maximum. In the example, the interval ranges from $2,001$ to $10,000$. For the case with $|\Upsilon(r_i)| = 0$, when $f_i = 2,040$, Θ_i attains the maximum 78. Also with revealing extra external relation, the greater $|\Upsilon(r_i)|$, the higher efficiency of Θ_i. Compared with $|\Upsilon(r_i)| = 0$, if $|\Upsilon(r_i)| = 50$, Θ_i achieves 235 revealed relations per microsecond. As illustrated in Table 3, when varying both $n_{i,0}$ and $l_{i,0}$ values and fixing $|\Upsilon(r_i)| = 0$, the optimal frame size is also bounded with the interval $[n_{i,0} + n_{i,1} + 1, n_{i,0} + n_{i,1} + l_{i,0}]$. For example, when $n_{i,0} = 3,000$ and $l_{i,0} = 4,000$, the optimal frame size is 3,837 which is at the interval [3,001, 6,000].

The optimal vector size that minimizes the global execution time is determined by two factors: the number of revealed relations inferred by Rule 1-6 and the reader scheduling mode. In IB, we consider the former factor rather than the later. That is because to get the optimal reader scheduling is equivalent to finding a minimum coloring for the conflict graph, which has been proven to be an NP-Complete problem [29], [30]. Hence, by considering only the number of revealed relations, the optimal frame size and the minimize execution time for each reader are the local optimum instead of the global optimum. Even so, IB is still able to achieve the distribution inference in an efficient way. The detailed performance analysis can be seen in Section 8.

6 Cardinality Estimation

As mentioned in previous section, to set the optimal frame size, the server needs to estimate the cardinality for the set of tags (i.e., $\mathcal{N}_{i,0}$ and $\mathcal{L}_{i,0}$). Many estimation schemes have been proposed [31], [32] to achieve fast and reliable estimation. But utilizing those separate estimation protocols will increase the execution time. We propose an estimation scheme without extra time consuming by using the information identified in each round.

Consider the previous round of the reader r_i, there are $n_{i,0}$ tags actually in the set $\mathcal{N}_{i,0}$. Also $n_{i,1}$, $l_{i,0}$ and $l_{i,1}$ are the actual cardinality of the set $\mathcal{N}_{i,1}$, $\mathcal{L}_{i,0}$ and $\mathcal{L}_{i,1}$ respectively. To set the optimal frame size in each round, the server should know the number of them. Although the server can precisely count the number of $n_{i,1}$ and $l_{i,1}$, the number of $n_{i,0}$ and $l_{i,0}$ can hardly be known. So it must take the scheme as following to estimate the numbers which are denoted as $\hat{n_{i,0}}$ and $\hat{l_{i,0}}$.

In IB, the server will generate the reader vector $\mathbf{V}_i$ and the server array $\mathbf{A}_i$ for the reader r_i. Thus, utilizing the difference between $\mathbf{V}_i$ and $\mathbf{A}_i$ reveals tag relations. For example, in Fig. 2(b), the reader r_2 attains $\mathbf{V}_2$='100000110'. The server can ascertain two tags are in the scope of the reader r_2 with Rule 2. In this way, by counting how many tags are in the region of r_2 with Rule 2, we can estimate the cardinality.

From the Eqn (5), the expected number β_i of the relations a tag is revealed by Rule 2 within the scope of the reader r_i is $\beta_i = n_{i,0}(1 - \frac{1}{f_i})^{n_{i,0}+n_{i,1}+l_{i,0}-1}$. After the execution of previous round, the server easily ascertains the number Λ_i of relations which are revealed by Rule 2 as follows:

$$\Lambda_i = \hat{n_{i,0}} \left(1 - \frac{1}{f_i}\right)^{\hat{n_{i,0}}+n_{i,1}+\hat{l_{i,0}}-1} \tag{14}$$

In the system, the total number of tags is n. So the relationship $n = n_{i,0} + n_{i,1} + l_{i,0} + l_{i,1}$ exists. The server utilizes the following estimator:

$$\hat{l_{i,0}} = n - \hat{n_{i,0}} - n_{i,1} - l_{i,1} \tag{15}$$

Therefore, combining the two equations above yields $\Lambda_i = \hat{n_{i,0}}(1 - 1/f_i)^{n-l_{i,1}-1}$. From the equation, we can derive:

$$\hat{n_{i,0}} = \Lambda_i \left(1 - \frac{1}{f_i}\right)^{l_{i,1}+1-n} \tag{16}$$

Using the Eqn (16), the server estimates the number of tags $\hat{n_{i,0}}$ which are determined without the scope of the reader i in the previous round. We then denote $n'_{i,0}$, $n'_{i,1}$, $l'_{i,0}$ and $l'_{i,1}$ as the current round number of each set. The increasing number of identified tags in set $\mathcal{N}_{i,1}$ is $\Delta n_{i,1} = n'_{i,1} - n_{i,1}$. Then the server infers the estimate number of the set $\mathcal{N}_{i,0}$ as follows:

$$\begin{aligned} \hat{n'_{i,0}} &= \hat{n_{i,0}} - \Delta n_{i,1} \\ \hat{l'_{i,0}} &= n - \hat{n'_{i,0}} - n'_{i,1} - l'_{i,1} \end{aligned} \tag{17}$$

The estimation scheme needs the reader vector of the previous round. For the first round of our protocol, we average the number of tags in each coverage of a reader. Therefore the server estimates the number as $\hat{n_{i,0}} = \frac{n}{m}$ and $\hat{l_{i,0}} = n - \frac{n}{m}$.

7 Implementation

In this section, we build a prototyping system by USRP-based readers and programmable WISP tags, and then implement IB protocol.

7.1 Prototype Setup

Due to the limitation of commercial RFID readers that provides little physical layer information interfaces, we build a tag distribution identification prototyping system based on the USRP software-defined platform and programmable WISP tags. Our test environment is shown in Fig.4(a). For more specifically, we implement the system with USRP1 based on Gen2 RFID project and WISP firmware. USRP1 in the prototype has two complete RFX900 daughterboards which are designed for operation in the 900 MHz band. The RFID tag is implemented with the WISP programmable device based on the DL-WISP4.1 firmware. The WISP tag generally comprises three parts: the first part is the MSP430F2132 microcontroller which can work in ultra-low power, the second part is an antenna circuitry which can gather and backscatter signals, and the third part is sensor which is not used in our system.

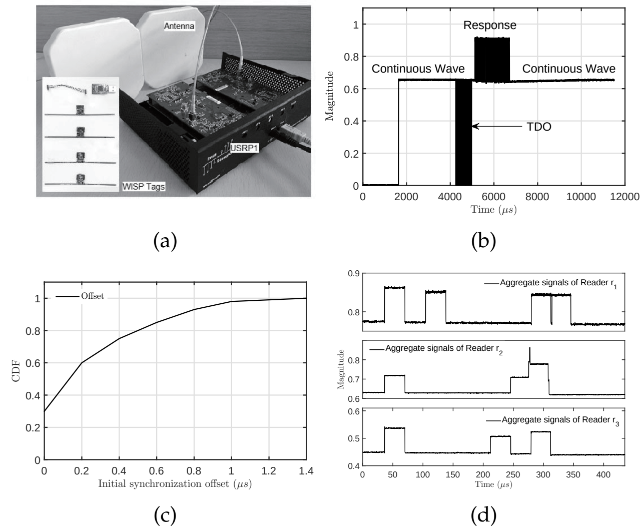


Fig. 4. Implementation. (a) Prototype Setup: The USRP1 with two complete RFX900 daughterboards and the WISP tags. (b) Communication process of TDO. (c) CDF of initial synchronization offset. (d) Received aggregated signals from tags under the regions of three readers.

In the firmware of DL-WISP4.1, it has already complemented the most commonly used features of the EPCglobal Gen-2 standard (e.g., QUERY, ACK, SELECT). To transplant our protocol on the WISP tag, we extend the original firmware by adding tag distribution obtaining command (TDO) which signals the beginning of the obtaining round. Upon communicating with a tag, a reader must transmit a continuous RF wave (CW). Fig.4(b) shows communication between the USRP-based reader and a WISP tag as probed by command TDO. The tag does not transmit any energy, but it powers itself by technically harvesting the signal energy from CW. The backscattered tag response can be apparently seen as a blend of the reflected CW and the incident CW. After receiving TDO, the aggregated responses from tags can be decoded at the reader.

7.2 Synchronization

The RFID tag captures the energy in the reader's RF signal to power itself up. On probing by the reader order, the tag transmits reply on the continuous wave. Theoretically, our protocol assumes that the transmissions of tags are synchronized. In practice, on concurrently replying to a reader's query, each tag has different delay from the start of transmission. But the response offset in synchronization has little impact on the performance. Here, we use 10 WISP tags to measure the offsets.

Fig.4(c) shows the CDF of the initial offsets. The offset's 50^{th} percentile is $0.35\mu s$. The 90^{th} percentile of the offset is $0.72\mu s$ and the maximum is less than $1.4\mu s$. When the default bit rate if $26.5Kb/s$, a $1.4\mu s$ offset is about 3.71%. The synchronization accuracy is sufficient for the performance. Thus our protocol is tolerable to the offsets in the system.

7.3 Tag Distribution Identification

To test the practicability of our protocol with the USRP-based reader and WISP tags, we deploy a prototyping system with the same tag distribution as in the case example (Section 4.2.1). As defined, the system has 3 readers and 9 tags. The reader r_1 and r_3 are both adjacent to the reader r_2. Due to the constrain of the number of the available USRP-based reader, we run the protocol in sequential mode. Thus we can perform the process of tag distribution obtaining for each reader one by one.

On being probed, the tags under the coverage of each reader reply signals concurrently. Fig. 4(d) plots the received signals at the readers. In the top of Fig. 4(d), due to the varied signal strength of different tags, the reader r_1 receives the aggregated signals from tags with different magnitude. But in our protocol, the reader only needs to distinguish empty bit from busy bit. Thus the reader r_1 easily interprets the signals to the reader vector $\mathbf{V}_1$='101000011' which is the same result as the analysis in Section 4.2.1. In the middle and the bottom of Fig. 4(d), the reader r_2 and r_3 obtain $\mathbf{V}_2$='100000110' and $\mathbf{V}_3$='100001010' respectively. Note in the seventh bit of $\mathbf{V}_2$, both t_2 and t_8 reply '1', the aggregated signal also can be identified apparently. The reader r_2 interprets that the seventh bit of $\mathbf{V}_2$ is '1'. The readers can examine the state of each bit to gain the set of tags. The server then can adopt IB to infer the distribution of tags in the system.

8 Performance Evaluation

8.1 Simulation Settings

We implement a simulator with Python to evaluate the performance of our protocol in large-scale systems. Our protocol is compared with the state-of-the-art protocols, including EDFSA [18], Protocol-3 [9], SFMTI [19], P-MTI [20] and BIC [12]. For fair comparison, we adopt the timing scheme defined in the EPCglobal C1G2 UHF tags [33] as the unit of the execution time for all protocols. The transmission rate of the tag is $53Kb/s$. It takes $18.88\mu s$ to transmit a bit from a tag to a reader, i.e. $t_b = 18.88\mu s$. The transmission rate of the reader is $26.5Kb/s$. Any two consecutive transmissions are separated by a time interval of $302\mu s$. Based on this we can calculate that the time for the reader to transmit an ID or a segment of allocation vectors is $3,927\mu s$. In both Protocol-3 and SFMTI, the tags need to send 10-bit long responses for helping the reader to distinguish empty, singleton, and collision slots. Thus the length of a slot for this purpose is $18.88 \times 10 + 302 = 490\mu s$. We fix the parameter $\rho = 1.68$ (the optimal value) in SFMTI. Since BIC is designed to collect tag information and adopts Bloom filter, we set the length of sensor information 1 bit and the false positive probability of the Bloom filter $p = 1 \times 10^{-3}$ and $p = 1 \times 10^{-4}$ as the same setting in [12].

In our prototype system, the server can control readers to run protocols in either sequential mode or parallel mode. In the sequential mode, the reader starts the execution after the previous one finished. On the contrary, in the parallel mode, the readers perform the execution simultaneously. In the multi-reader cases, the communication conflict would exist. There are many existing reader scheduling algorithms [27], [28], which propose the conflict-free schedules for multi-reader cases. We can adopt those algorithms to arrange the identification sequence for the readers. Thus, we focus on two extreme cases that are sequential mode and parallel

TABLE 4
Execution Time of Different Protocols in Parallel Mode (s)

Scenario	Parameter					Scheme						
	m	μ	σ	d	n	EDFSA	Protocol-3	SFMTI	P-MTI	BIC($p=10^{-4}$)	BIC($p=10^{-3}$)	IB
1	20	2,000	500	20	40,000	17.1	21.3	11.8	9.8	2.3	2.0	0.8
2	30	2,000	500	30	60,000	17.7	32.1	17.8	15.6	2.4	2.1	1.1
3	50	2,000	500	50	100,000	18.3	53.8	29.7	26.3	2.4	2.2	1.3
4	50	3,000	750	50	150,000	27.5	80.7	44.6	40.8	3.7	3.2	1.8
5	50	4,000	1,000	50	200,000	36.5	107.3	59.3	53.7	4.8	4.3	2.5
6	50	4,000	1,000	0	200,000	36.7	107.7	59.5	53.4	4.9	4.3	2.3

TABLE 5
Execution Time of Different Protocols in Sequential Mode (s)

Scenario	Parameter					Scheme						
	m	μ	σ	d	n	EDFSA	Protocol-3	SFMTI	P-MTI	BIC($p=10^{-4}$)	BIC($p=10^{-3}$)	IB
1	20	2,000	500	20	40,000	229.8	391.7	216.7	188.4	31.1	27.5	10.6
2	30	2,000	500	30	60,000	344.7	585.2	327.0	284.7	46.5	42.1	17.4
3	50	2,000	500	50	100,000	574.5	983.5	546.9	473.1	77.2	70.6	31.6
4	50	3,000	750	50	150,000	862.0	1,462.4	814.5	708.3	117.4	105.9	49.9
5	50	4,000	1,000	50	200,000	1,145.8	1,934.4	1,083.3	937.1	156.7	140.9	68.8
6	50	4,000	1,000	0	200,000	1,147.6	1,937.3	1,084.8	935.9	156.4	140.3	59.5

mode. On adopting the scheduling algorithms, the execution time is bounded at the range from the time of sequential mode to the time of parallel mode. Each simulation experiment was conducted for 100 times and then we report the averaged results of the independent trials.

8.2 Simulation Performance

In this subsection, we evaluate the performance of our protocol with the five protocols, i.e., EDFSA, Protocol-3, SFMTI, P-MTI, BIC ($p = 10^{-3}$), BIC ($p = 10^{-4}$) with various parameter settings. We first evaluate the execution efficiency under six fixed scenarios. Then we study the impacts of different parameters, i.e, the total number m of readers, the average number μ of tags in each coverage, the standard deviation σ and the total degree d of the topology.

8.2.1 General Performance

In Table 4 and 5, we compare the execution time among EDFSA, Protocol-3, SFMTI, P-MTI, BIC ($p = 10^{-3}$), BIC ($p = 10^{-4}$) and IB under six fixed scenarios in both the parallel and sequential mode. In each scenario, we vary one factor and fix others to study an overview of the influence of each factor. We first evaluate the impact of m, which fixes $\mu = 2,000$ and $\sigma = 500$, and varies m from 20 to 50. The total number of tags n is from $40,000$ to $100,000$. To keep the same of average degree, we also vary d from 20 to 50. Then we fix $m = 50$ and $d = 50$, and vary μ from 2,000 to 4,000, σ from 500 to 1,000. The total number of tags n is from $100,000$ to $200,000$. At last, we fix $m = 50$, $\mu = 4,000$ and $\sigma = 1,000$, and vary d from 50 to 0.

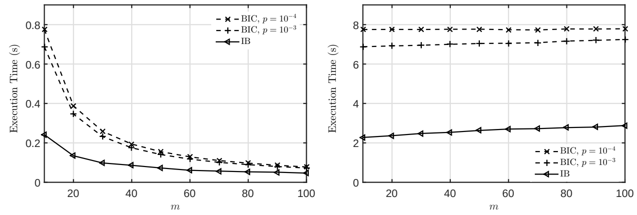
From the tables, we observe that IB performs better than the other six protocols. With different scheduling algorithms, the execution time will be bounded at the interval of the execution time of sequential mode (upper bound) and the execution time of parallel mode (lower bound). The performance gain of IB under different scenarios benefits from following design principles: (1) Instead of using only the application-level information, the inter-round physical signals are exploited in IB to extract more tag distribution information. (2) Instead of working individually, all readers in IB cooperate with each other to accelerate the identification process. (3) The more relations IB reveals in tag distribution through topology, the less time IB consumes.

But for EDFSA, Protocol-3, SFMTI, P-MTI and BIC protocols, the optimal frame size is determined by the number of tags in the reader's coverage region or the number of tags in the system. Therefore, the different degrees of each reader will also have no impact on the total execution time. And the readers cannot work together and share their tag distribution information with each other.

To enable Protocol-3, SFMTI and P-MTI to tackle our problem, each reader must individually execute the missing tag identification protocols by taking the overall tags as input and treating tags outside its coverage as missing tags. This works but is time-consuming, since the length of the frames used in these protocols is proportional to the number of overall tags instead of local tags. Due to the huge performance gap and for the ease of presentation, we remove the comparison of EDFSA, Protocol-3, SFMTI and P-MTI in the following subsections.

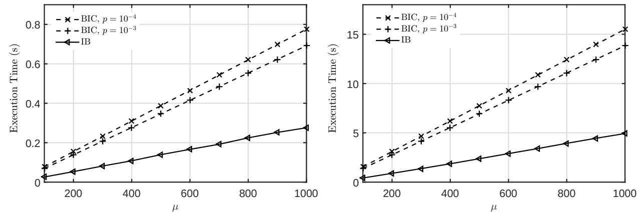
8.2.2 Impact of m

Fig. 5 plots the execution time of different protocols as the reader number increases. Here we set the total number of tags in the system $n = 10,000$. In the meantime, we fix $\sigma = 0$, $\mu = \frac{n}{m}$, $d = 0$ and vary the number of readers m from 10 to 100. Generally, IB outperforms over BIC. In the parallel mode, as the increase of the reader number m, the execution time of the three protocols have a sharp decline. But both BIC ($p = 10^{-3}$) and BIC ($p = 10^{-4}$) take more time on execution than IB. When $m = 20$, the execution time of BIC ($p = 10^{-3}$) and BIC ($p = 10^{-4}$) are



(a) (b)

Fig. 5. Execution time of BIC and IB with respect to m. (a) Parallel reader scheduling, (b) Sequential reader scheduling.



(a) (b)

Fig. 6. Execution time of BIC and IB with respect to μ. (a) Parallel reader scheduling, (b) Sequential reader scheduling.

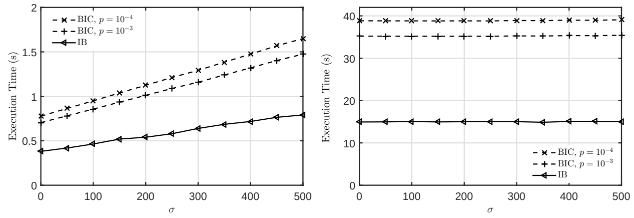
$0.34s$ and $0.38s$ respectively. In the meantime, IB finishes obtaining in $0.15s$ respectively. In the sequential mode, the total execution time of both IB and BIC have a slight increase as the rise of the reader number. Compared with BIC, IB can reduce execution time by up to 56.2% in the parallel mode and 62.1% in the sequential mode, respectively.

8.2.3 *Impact of* μ

In Fig. 6, we evaluate the impact of the average tag number under each reader's coverage μ among BIC ($p = 10^{-3}$), BIC ($p = 10^{-4}$) and IB. We fix $\sigma = 0$, $m = 20$, $d = 0$ and vary μ from 100 to $1,000$. As shown in the figures, the execution time of all protocols shares a steady rise tendency as the average tag number increases in both parallel and sequential mode. However, IB uses less time than BIC. In more detail, when $\mu = 800$, IB consumes approximately $3.96s$ to complete the identification in sequential mode. While BIC ($p = 10^{-3}$) and BIC ($p = 10^{-4}$) consume about $11.06s$ and $12.40s$. Compared with BIC, IB can reduce execution time by up to 64.9% in the parallel mode and 68.1% in the sequential mode, respectively.

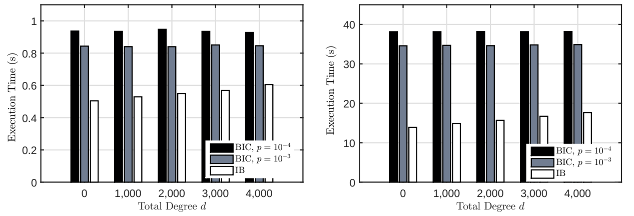
8.2.4 *Impact of* σ

In Fig. 7, we evaluate the standard derivation σ with respect to the execution time of BIC ($p = 10^{-3}$), BIC ($p = 10^{-4}$), and IB. In the simulation, we set the parameters $m = 50$, $\mu = 1,000$, $d = 50$, and vary σ from 0 to 500. In the parallel mode (Fig. 7(a)), since the optimal frame size is constrained to the region of maximum tag number, all protocols share a steady rise as σ increases. But in the sequential mode (Fig. 7(b)), all the protocols remain stable because the total number of the groups remains the same regardless of the variation σ. In either parallel mode or sequential mode, IB outperforms BIC. Compared with BIC, IB can reduce execution time by up to 48.1% in the parallel mode and 60.9% in the sequential mode, respectively.



(a) (b)

Fig. 7. Execution time of BIC and IB with respect to σ. (a) Parallel reader scheduling, (b) Sequential reader scheduling.



(a) (b)

Fig. 8. Execution time of BIC and IB with respect to d. (a) Parallel reader scheduling, (b) Sequential reader scheduling.

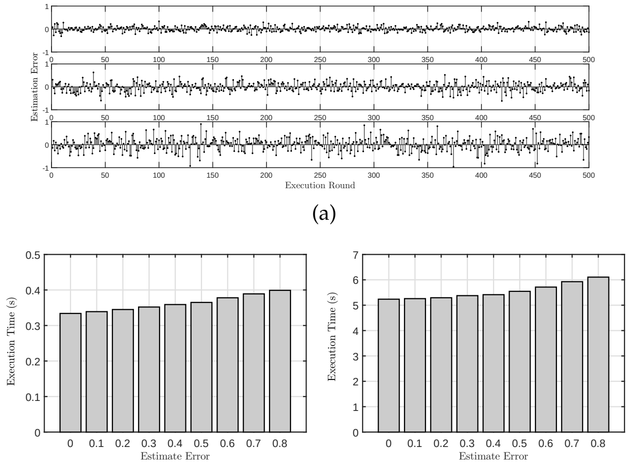
8.2.5 *Impact of* d

We also evaluate the impact of the different degrees among BIC ($p = 10^{-3}$), BIC ($p = 10^{-4}$) and IB. Here we fix $\mu = 1,000$, $\sigma = 100$, $m = 50$ and vary d from 0 to $4,000$. Without losing generality, we generate the random topologies in the simulation. As shown in the Fig. 8, the more relations IB reveals in tag distribution through topology, the less time IB consumes. For example, when the degree d decreases from $3,000$ to $1,000$ in sequential mode, both BIC ($p = 10^{-3}$) and BIC ($p = 10^{-4}$) share a stable execution time, but IB has a decline with $1.7s$ in the execution time respectively.

8.3 Performance Evaluation

8.3.1 *Cardinality Estimation Scheme*

To achieve the high efficiency in obtaining tag distribution, the server must set the optimal frame size by utilizing the cardinality of the sets $\mathcal{N}_{i,0}$, $\mathcal{N}_{i,1}$, $\mathcal{L}_{i,0}$ and $\mathcal{L}_{i,1}$. In our method, we propose the estimation scheme to attain the cardinality of the set $\mathcal{N}_{i,0}$ and the set $\mathcal{L}_{i,0}$. However, the estimated $\hat{n_{i,0}}$ and $\hat{l_{i,0}}$ may deviate from the actual value. In Fig. 9(a), we plot three scenarios to investigate the estimation error by running 500 times, varying $l_{i,0}$ from 1,000 to 20,000 and fixing $n_{i,0} = 1,000$. The estimation error has slightly increase with the increase of $l_{i,0}$, and is smaller than 0.8 in most cases. Then we plot Fig. 9(b) and 9(c) to investigate the impact of estimation error on the total execution time. The figure illustrates the results with $\mu = 1,000$, $\sigma = 0$, $m = 20$ and $d = 0$, and the estimation error varies from 0 to 0.8. Obviously, when no estimation error exists, the execution time is the shortest. But when the estimation error grows larger, the execution time only has a small increase. Thus, the estimation error is tolerable for our protocol IB.



(a)

(b)

(c)

Fig. 9. Evaluation of the cardinality estimation scheme. (a) Three scenarios to investigate the estimation error (From the top to the bottom: $l_{i,0} = 1,000, 10,000$ and $20,000$). (b) Impact of estimation error on the total execution time in parallel mode. (c) Impact of estimation error on the total execution time in serial mode.

TABLE 6
Total Rounds under Five Scenarios

Scenario	Parameter			Total Rounds		
	n_i	m	n	Avg	Min	Max
1	2,000	20	40,000	9	6	15
2	2,000	30	60,000	11	8	16
3	2,000	50	100,000	10	7	14
4	3,000	50	150,000	11	8	15
5	4,000	50	200,000	11	6	16

8.3.2 Efficiency Evaluation

We then investigate the efficiency of IB in five fixed scenarios. Since the standard deviation σ varies among the readers in the system, we set $\sigma = 0$ in the five scenarios to have a clear investigation of the efficiency. For the simplicity of description, we only focus on the i-the reader. The settings of the five scenarios are shown in Table 6 and 7. We first fix $n_i = 2,000$, $d = 0$ and vary m from 20 to 50. And then we fix $m = 50$ and vary $n_i = 4,000$. Table 6 plots the total rounds of the reader r_i uses under the five scenarios. We can observe in all cases, IB can execute in no more than 16 rounds. On average, the reader r_i finishes identification in about 10 rounds, which helps the server achieve high efficiency of the tag distribution identification. Table 7 plots the total vector size of the reader r_i uses under five scenarios. We can observe that the total vector size increases as the increase of the total number of tags in the system.

9 Conclusion

This paper studies a new problem of how to efficient explore tag distribution in multi-reader RFID systems. The problem is to fast identify the tag set beneath each reader, which is a fundamental premise of efficient product inventory and management. Only with such tag set information can we localize specific tags in a reader and expedite the tag query information collection. We present a sophisticated protocol IB that identifies the tag distribution based on information inference rules and the aggregated physical signals to improve operational efficiency. To validate the protocol efficiency, we build a prototype system, which is based on the Universal Software Radio Peripheral (USRP) platform and the Intel Wireless Identification and Sensing Platform (WISP), and then implement IB. From conducting extensive simulations, the evaluation results show that IB outperforms the state-of-the-art protocols.

TABLE 7
Total Vector Size under Five Scenarios

Scenario	Parameter			Total Vector Size ($\times 10^3$)		
	n_i	m	n	Avg	Min	Max
1	2,000	20	40,000	31.2	27.4	37.1
2	2,000	30	60,000	33.7	29.0	39.8
3	2,000	50	100,000	35.3	28.4	42.9
4	3,000	50	150,000	47.5	42.7	51.7
5	4,000	50	200,000	63.1	56.6	70.8

Acknowledgments

This research is financially supported by the National Natural Science Foundation of China (No.61272418, 61373181), the National Science and Technology Support Program of China (No.2012BAK26B02), the Future Network Prospective Research Program of Jiangsu Province (No.BY2013095-5-02), the Lianyungang City Science and Technology Project (No.CG1420, JC1508), the Fundamental Research Funds for the Central Universities, HK PolyU G-SB40 and the program B for Outstanding PhD candidate of Nanjing University.

References

[1] L. Xie, H. Han, Q. Li, J. Wu, and S. Lu, "Efficient protocols for collecting histograms in large-scale RFID systems," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2015.

[2] X. Liu, B. Xiao, S. Zhang, K. Bu, and A. Chan, "Step: A time-efficient tag searching protocol in large RFID systems," *IEEE Transaction on Computers (TOC)*, 2015.

[3] J. Liu and L. Chen, "Placement of multiple RFID reader antennas to alleviate the negative effect of tag orientation," in *Proc. of IEEE ICPADS*, 2012.

[4] X. Liu, S. Zhang, B. Xiao, and K. Bu, "Flexible and time-efficient tag scanning with handheld readers," *IEEE Transactions on Mobile Computing (TMC)*, 2014.

[5] J. Liu, B. Xiao, K. Bu, and L. Chen, "Efficient distributed query processing in large RFID-enabled supply chains," in *Proc. of IEEE INFOCOM*, 2014.

[6] F. Zhu, B. Xiao, J. Liu, X. Liu, and L. jun Chen, "Plat: A physical-layer tag searching protocol in large RFID systems," in *Proc. of IEEE SECON*, 2016.

[7] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile RFID tags to high precision using cots devices," in *Proc. of ACM MOBICOM*, 2014.

[8] J. Han, C. Qian, X. Wang, D. Ma, J. Zhao, P. Zhang, W. Xi, and Z. Jiang, "Twins: Device-free object tracking using passive tags," in *Proc. of IEEE INFOCOM*, 2014.

[9] Y. Zhang, Y. Liu, Y. Zhang, and J. Sun, "Fast identification of the missing tags in a large RFID system," in *Proc. of IEEE SECON*, 2011.

[10] F. Zhu, J. Liu, and L. Chen, "Fast physical-layer unknown tag identification in large-scale RFIDsystems," in *Proc. of IEEE GLOBECOM*, 2014.

[11] X. Liu, B. Xiao, S. Zhang, and K. Bu, "Unknown tag identification in large RFID systems: An efficient and complete solution," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2014.

[12] H. Yue, C. Zhang, M. Pan, Y. Fang, and S. Chen, “A time-efficient information collection protocol for large-scale RFID systems,” in *Proc. of IEEE INFOCOM*, 2012.

[13] J. Liu, B. Xiao, X. Liu, and L. Chen, “Fast rfid polling protocols,” in *Proc. of ICPP*, 2016.

[14] M. Chen, W. Luo, Z. Mo, S. Chen, and Y. Fang, “An efficient tag search protocol in large-scale RFID systems,” in *Proc. of IEEE INFOCOM*, 2013.

[15] S. Zhang, X. Liu, J. Wang, J. Cao, and G. Min, “Energy-efficient active tag searching in large scale RFID systems,” *Information Sciences*, 2015.

[16] V. Namboodiri and L. Gao, “Energy-aware tag anticollision protocols for RFID systems,” *IEEE Transactions on Mobile Computing (TMC)*, 2010.

[17] B. Zhen, M. Kobayashi, and M. Shimizu, “Framed aloha for multiple RFID objects identification,” *IEICE Transactions on Communications*, 2005.

[18] S.-R. Lee, S.-D. Joo, and C.-W. Lee, “An enhanced dynamic framed slotted aloha algorithm for RFID tag identification,” in *Proc. IEEE MobiQuitous*, 2005.

[19] X. Liu, K. Li, G. Min, Y. Shen, A. Liu, and W. Qu, “Completely pinpointing the missing RFID tags in a time-efficient way,” *IEEE Transactions on Computers (TOC)*, 2015.

[20] Y. Zheng and M. Li, “P-mti: Physical-layer missing tag identification via compressive sensing,” in *Proc. of IEEE INFOCOM*, 2013.

[21] J. Myung, W. Lee, J. Srivastava, and T. Shih, “Tag-splitting: Adaptive collision arbitration protocols for RFID tag identification,” *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2007.

[22] M. Shahzad and A. X. Liu, “Probabilistic optimal tree hopping for RFID identification,” in *ACM SIGMETRICS Performance Evaluation Review*, 2013.

[23] J. Liu, M. Chen, B. Xiao, F. Zhu, S. Chen, and L. Chen, “Efficient RFID grouping protocols,” *IEEE/ACM Transactions on Networking*, 2016.

[24] S. Chen, M. Zhang, and B. Xiao, “Efficient information collection protocols for sensor-augmented RFID networks,” in *Proc. of IEEE INFOCOM*, 2011.

[25] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2003.

[26] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, “Efficient and reliable low-power backscatter networks,” in *Proc. of ACM SIGCOMM*, 2012.

[27] J. Waldrop, D. W. Engels, and S. E. Sarma, “Colorwave: an anticollision algorithm for the reader collision problem,” in *Proc. of IEEE ICC*, 2003.

[28] 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 IEEE INFOCOM*, 2011.

[29] V. V. Vazirani, *Approximation algorithms*. Springer, 2001.

[30] D. P. Williamson and D. B. Shmoys, *The design of approximation algorithms*. Cambridge university press, 2011.

[31] W. Gong, K. Liu, X. Miao, and H. Liu, “Arbitrarily accurate approximation scheme for large-scale RFID cardinality estimation,” in *Proc. of IEEE INFOCOM*, 2014.

[32] B. Chen, Z. Zhou, and H. Yu, “Understanding RFID counting protocols,” in *Proc. of ACM MOBICOM*, 2013.

[33] *Epcglobal. epc radio-frequency identity protocols class-1 generation-2 uhf RFID protocol for communications at 860 mhz-960mhz version 1.2.0*, Tech. Rep., 2008.

Feng Zhu received his B.E. degree in Information Security from Harbin Institute of Technology, China, in 2011. He is currently a Ph.D. student with the Department of Computer Science and Technology at Nanjing University of China. His research interests include RFID technologies and wireless sensor networks.

Dr. Bin Xiao is currently an Associate Professor in the Department of Computing, The Hong Kong Polytechnic University. Dr. Xiao received the B.Sc and M.Sc degrees in Electronics Engineering from Fudan University, China, and Ph.D. degree in computer science from University of Texas at Dallas, USA. His research interests include distributed wireless systems, network security, and software-defined networks (SDN). Dr. Xiao has published more than 100 technical papers in international top journals and conferences. Currently, he is the associate editor of the Journal of Parallel and Distributed Computing (JPDC) and Security and Communication Networks (SCN). He is the IEEE Senior member, ACM member and the recipient of the best paper award of the international conference IEEE/IFIP EUC-2011.

Jia Liu received his B.E. degree in Software Engineering from Xidian University, Xi’an, China, in 2010. He is currently a Ph.D. student with the Department of Computer Science and Technology at Nanjing University of China. His research interests include RFID technologies and wireless sensor networks. He is a member of IEEE.

Bin Wang received his B.E. degree in Information and Computing Science from Jiangnan University, China, in 2012. He recieved a master degree in the Department of Computer Science and Technology at Nanjing University of China, in 2015. His research interests include RFID technologies and wireless sensor networks.

Qingfeng Pan received his B.E. degree in Computer and Application from Qingdao University of Science and Technology, Qingdao, China, in 2002. He is currently a Ph.D. student with the Department of Computer Science and Technology at Nanjing University of China. His research interests include RFID technologies and wireless sensor networks.

Dr. Li-jun Chen received his B.S. degree in Electrical Engineering from the Xian University of Science and Technology, P. R. China, in 1982 and his M.S. degree and Ph.D. degree, both in Automatic Control from China University of Mining and Technology, P. R. China, in 1993 and 1998 respectively. He was a Post Doctoral Follow at the Nanjing University (1998 -2000), P. R. China and the Michigan State University (2001-2002), U.S.A, and a Visiting Scholar at The Hong Kong Polytechnic University in 2007. His current research interests include distributed computing and ubiquitous network.