

# Energy-Efficient Active Tag Searching in Large Scale RFID Systems

Shigeng Zhang<sup>a</sup>, Xuan Liu<sup>b</sup>, Jianxin Wang<sup>a</sup>, Jiannong Cao<sup>b,a</sup>, Geyong Min<sup>c</sup>

<sup>a</sup>*School of Information Science and Engineering, Central South University, China*

<sup>b</sup>*Department of Computing, Hong Kong Polytechnic University, Hong Kong*

<sup>c</sup>*College of Engineering, University of Exeter, United Kingdom*

---

## Abstract

Radio Frequency Identification (RFID) is a key enabling technology of Internet of Things (IoTs) and has attracted much research attention in recent years. RFID can support automatic information tracing and management during the management process in many fields. A typical application of RFID is modern warehouse management, where products are attached with RFID tags and the inventory is managed by retrieving tag IDs. Many practical applications require searching a group of tags to determine whether they are in the system or not. The existing studies on tag searching have mainly focused on improving time efficiency and paid little attention to energy efficiency that is extremely important for active tags powered by built-in batteries. To fill in this gap, this paper investigates the tag searching problem from the angle of energy efficiency. We first propose an Energy-efficient tag Searching protocol in Multiple reader RFID systems, namely ESiM, which pushes per tag energy consumption to the limit as each tag needs to exchange only one bit

---

*Email addresses:* [sgzhang@csu.edu.cn](mailto:sgzhang@csu.edu.cn) (Shigeng Zhang),  
[csxuanliu@comp.polyu.edu.hk](mailto:csxuanliu@comp.polyu.edu.hk) (Xuan Liu), [jxwang@csu.edu.cn](mailto:jxwang@csu.edu.cn) (Jianxin Wang),  
[csjcao@comp.polyu.edu.hk](mailto:csjcao@comp.polyu.edu.hk) (Jiannong Cao), [g.min@exeter.ac.uk](mailto:g.min@exeter.ac.uk) (Geyong Min)

data with the reader. We then develop a time efficiency enhanced version of ESiM, namely TESiM, which can dramatically reduce the execution time while increasing transmission overhead only slightly. Extensive simulation experiments reveal that, compared to state-of-the-art solution in the current literature, TESiM reduces per tag energy consumption by more than one order of magnitude subject to comparable execution time. Furthermore, in most scenarios, TESiM even reduces the execution time by more than 50 percent.

*Keywords:* Internet of Things, RFID tag searching, Energy efficiency, Active tags, IoT in industry

---

## 1. Introduction

Internet of Things (IoT) has been considered as a novel paradigm that has the potential to bring revolutionary changes to our lifetime [1]. IoT integrates Radio Frequency Identification (RFID) technology, sensor technology, actuators, and novel wireless technologies like near field communications (NFC) to build Internet-like infrastructure for objects that are identifiable. All the things in IoT could be automatically managed and inventoried by computers, and thus the management efficiency could be improved greatly. Currently, IoT is exploiting its application in a wide scope of industry fields [1–4].

As a key enabling technology of IoT, RFID can be used in many industrial fields to support intelligent process management, e.g., retailing industry, transportation and logistics industry, healthcare industry, and construction industry [5–9]. For example, a typical application of RFID technology in logistics industry is RFID-enabled warehouse management. The RFID sys-

tem deployed in a modern warehouse usually consists of a large number of RFID tags that are attached to products and multiple RFID readers. RFID tags can be accessed by readers wirelessly from a distance without line-of-sight interaction, and thus are more flexible than the traditional barcode tags that are severely limited in operational range. Multiple RFID readers are deployed in different places of the warehouse in order to cover the whole area. By collecting all the tag IDs, the warehouse can manage and update the inventory of products in an automatic manner, and thus can improve management efficiency significantly.

There are mainly two types of RFID tags [10, 11]: passive tags and active tags. Passive tags harvest energy from the radio signal broadcasted by the reader to backscatter their data, and thus have very limited operational range. Passive tags are suitable for small range applications like fast checkout. Active tags are powered by built-in batteries, and thus have much longer operational distance. In large scale RFID systems that cover a very wide area, e.g., a big warehouse, active tags are more preferable. Furthermore, active tags are necessary in many application scenarios that require sensing environment data because they have rich on-chip sensors. Although currently passive tags are more sold and used than active tags [12, 13], it is forecasted that the market of active tags will raise to 25% of the total RFID market in 2020. Thus it is necessary and meaningful to investigate active RFID tags.

Rather than collecting all the tag IDs, many applications in warehouse require determining whether a certain group of tags are in the system or not. Consider a big warehouse that stores products for many different manufac-

turers. Given a list of tag IDs that represent flawed products, a manufacture wants to search which of them are in the warehouse in order to recall and fix them. Such a task is referred to as *tag searching* where the tags to be searched are called *wanted tags*. Tag searching is of great importance to many practical applications in industry. For example, a manufacture may store its products in different warehouses due to the constraint in logistic budget. It can learn the distribution of its products by searching which products are in which warehouse. Tag searching can also help update the inventory of a (or several) specified type(s) of products, or provide input to RFID polling protocols that aim to collect information from some specified tags [14]. In this paper, rather than consideration of the special scenarios for searching a single tag, we consider the generalized scenarios in which a group of tags are searched simultaneously.

Although the tag searching problem can be solved by collecting the IDs of all the tags in the system, it is far from efficiency in terms of both time and energy, especially in large scale RFID systems that may contain tens of thousands of active tags. The identification throughput is only several hundreds tags per second [15–17]. It might be imagined that we can solve the tag searching problem by using a dedicated database to trace which tags enter or leave the system. This approach, however, faces several problems as follows. First, the system might have no infrastructure to record which tags enter or leave the system. For example, if the RFID system is temporarily built with mobile readers, it may have no specially designed database to record which tags enter or leave the system. Second, in the current standard [18][19][17], because the reader has no simple way to distinguish the tags that

are not enrolled yet from those that have already been enrolled, it has to collect all the tags in its interrogation region. In fact, how to efficiently read only the tags that have not been enrolled into the system is an interesting problem called *unknown tag identification* [20–22] that has not been thoroughly solved. Third, tags in the system might be stolen or missing, making it difficult to exactly trace which tags have left the system. If such stolen or missing tags are not detected, the searching accuracy might be affected. In fact, missing tag detection is also an interesting problem that have attracted much research attention in recent years [23–25].

In [15] the authors proposed the Compact Approximator based Tag Searching (CATS) protocol to search tags in a large scale RFID system. CATS aims to reduce the searching time by avoiding tag ID collection. It employs Bloom filter to compact the information exchanged between the tags and readers, and finds the searching result by estimating the intersection of the two bloom filters respectively representing the set of wanted tags and the set of all the tags in the system. However, CATS paid little attention to energy consumption. In CATS, each tag needs to receive a large volume of data from the reader, causing very high per tag energy consumption. In [26] the authors proposed a time-efficient tag searching protocol in RFID systems containing multiple readers. However, both of them are not suitable to RFID systems that are built with active tags powered by built-in batteries.

Energy efficiency is an important objective in designing RFID tag searching protocols for the systems built upon active tags, and is still an open research issue. With the advantages in longer operational distance and rich on-chip sensors, active tags are more likely to be used in large scale RFID

systems. However, to the best of our knowledge, energy efficient tag searching in large scale RFID systems has not been thoroughly investigated and remains a challenging problem. To fill in this gap, we study the tag searching problem from the angle of energy efficiency. The major contributions of this paper include:

- We propose an Energy-efficient tag Searching protocol in Multiple reader RFID systems, namely ESiM, which pushes per tag energy consumption to a limit. Each tag in ESiM needs to exchange only one bit data with the reader, which is two orders of magnitude less than the best of the existing solutions.
- In order to further improve the time efficiency of ESiM, we develop the TESiM (i.e., Time efficiency enhanced ESiM) protocol that adopts a multiple round method to shorten the frame size and hence dramatically reduce the execution time. Meanwhile, TESiM increases energy consumption on each tag only slightly.
- Extensive simulation experiments are conducted to evaluate the performance of the two proposed protocols. The results demonstrate that, compared to state-of-the-art solution in the current literature, TESiM reduces per tag energy consumption by more than one order of magnitude subject to comparable execution time.

The rest of this paper is organized as follows. Section 2 overviews the related work. Section 3 presents the system model and problem statement. In Section 4, the detailed descriptions of ESiM and TESiM are given, along with theoretical analysis on their performance in terms of energy, time, and

sensitivity to different system parameters. In Section 5, the simulation results are reported and analyzed. Finally, Section 6 concludes this paper.

## 2. Related Work

Tag identification protocols can be naturally used to solve the tag searching problem, but they are neither energy efficient nor time efficient in performing the task of tag searching. Generally, the existing RFID tag identification protocols can be classified into two categories [10, 11]: ALOHA-based protocols [27–30] and tree-based protocols [19, 31]. In ALOHA-based protocols, on average a tag needs to transmit its ID  $e$  times to the reader before being successfully identified [10, 27], where  $e$  is the base of natural logarithm whose value is approximately equal to 2.72. The typical length of a tag ID is 96 bits [17, 32]. Thus, it incurs  $2.72 \times 96 \approx 261$  bits data transmission from every tag to the reader. Meanwhile, the time efficiency of tag identification is also low because the throughput of tag identification is only 100 to 200 tags per second [15, 16, 33]. As a comparison, in the searching protocol proposed in this paper, every tag needs to transmit only ten to twenty bits data to the reader, more than one order of magnitude less than that in tag identification protocols.

The RFID tag searching protocol, CATS, proposed in [15] aims to improve high time-efficiency. CATS is a two-phase protocol that uses Bloom filters to quickly find which of the wanted tags are in the system. In the first phase, a Bloom filter representing all the wanted tags is constructed and broadcasted to all the tags in the system. After receiving the filter, tags in the system check whether they are in the filter and determine whether they

should participate in the second phase or not accordingly. The goal of the first phase is to reduce the number of tags participating in the second phase. In the second phase, the reader constructs a virtual Bloom filter representing all the remaining tags in the system by scanning replies from tags, and filters out those wanted tags that are not in the virtual filter. CATS achieved much better time efficiency than tag identification protocols [15], but it paid little attention to energy consumption of tags. In the first phase of CATS, every tag needs to receive a very long filter, and thus consumes a lot of energy (note that for active tags receiving one bit consumes the same energy as transmitting one bit[14, 24]). Compared to CATS, our protocols proposed in this paper reduce per tag energy consumption by more than two orders of magnitude.

Multiple reader scheduling has attracted much research attention in recent years [34–37]. Most of the reader scheduling algorithms target to improve the tag identification throughput by allowing as many readers as possible to work simultaneously. Waldrop, Engels, and Sarma [34] proposed a distributed reader scheduling algorithm based on graph coloring called Colorwave. Zhou et al. [35] proposed a centralized reader scheduling algorithm based on STDMA (Spatial Time Division Multiple Access). Tang et al. [36] proposed the RASpberry protocol that tries to make the system work in a stable way in a long term when the arrival rate of tags is within the capacity region of the readers. They further proposed a scheduling algorithm to maximize the number of served tags per time slot while avoiding interference among readers in [37]. Our reader scheduling algorithm is also based on graph coloring [34] but makes important enhancement in order to guarantee the high energy



and time efficiency of the tag searching protocols proposed in this paper.

### 3. System Model and Problem Statement

#### 3.1. System Model

We consider a large scale RFID system consisting of multiple readers and a large number of active tags. A back end server communicates with all the readers and coordinates them to avoid collisions between nearby readers. The communication between the back end server and readers can be either wired or wireless. Because the interrogation range of a single reader is very limited, large RFID systems that cover a very wide area usually need to deploy multiple readers to cover the whole area. In this paper, we consider the generalized scenarios where multiple readers are needed. However, our solutions can also be applied to the special cases where only a single reader is contained.

We mainly focus on RFID systems built with active tags that are powered by built-in batteries, e.g., Philips I-code tags [17]. Tags adopt the *frame slotted ALOHA* [18] protocol as the basic communication protocol. In the ALOHA protocol, the reader issues queries to and receives replies from tags in consecutive *frames*. Every frame is further divided into many slots. At the beginning of each frame, the reader broadcasts a query that contains the frame size  $f$  and a random seed  $s$ . After receiving the query, a tag calculates a hash value  $S = H(ID_t || s) \bmod f$ , where  $ID_t$  denotes the tag's ID. It then replies to the reader in the  $S$ -th slot. Because tags may collide with each other, the reader may need to issue multiple frames to collect all the replies from tags. It has been proven that, on average, a tag needs to transmit its

ID  $e$  times before it could be successfully identified [18]. In our tag searching protocol, when the searching task completes, tags enter sleeping mode to save energy and prolong lifetime.

According to the number of tags that transmit in each slot, there are three different types of slots. A slot is called an *empty* slot if no tags transmit in it, or is called a *non-empty* slot otherwise. A non-empty slot can be either a *singleton* slot, in which only one tag transmits to the reader, or a *collision* slot, in which more than one tags transmit to the reader simultaneously. In our protocols, a reader only needs to distinguish between empty and non-empty slots. To achieve this goal, a tag can transmit a short response containing only one bit to the reader [14, 24]. In contrast, in the traditional tag identification protocols, a tag transmits its ID to the reader. Let  $t_e$ ,  $t_b$ , and  $t_{id}$  denote the time duration of an empty slot, a slot in which a one-bit short response is transmitted, and a slot in which a tag ID is transmitted, respectively.

Nearby readers cannot work simultaneously due to potential collisions. In this paper we consider two types of collisions between readers [35, 36]: *Reader-Tag collisions* and *Reader-Reader collisions*. If a tag  $t$  is in reader  $A$ 's interrogation range and reader  $B$ 's interference range simultaneously, its reply to reader  $A$  may be ruined by the signal from reader  $B$ , which causes a Reader-Tag (R-T) collision. If  $t$  is in the interrogation range of both  $A$  and  $B$ , it then cannot correctly receive the commands sent by either reader, which causes a Reader-Reader (R-R) collision. The readers should be scheduled to avoid R-R and R-T collisions.

### 3.2. Problem Statement

This section defines the tag searching problem. Let  $\mathcal{S}$  and  $\mathcal{T}$  denote the set of all the wanted tags and the set of all the tags in the system, respectively. Given  $\mathcal{S}$ , we want to find which tags in  $\mathcal{S}$  are present in  $\mathcal{T}$ , i.e., the intersection  $\mathcal{T} \cap \mathcal{S}$ . The goal is to reduce the energy consumption of tags during the searching procedure. Meanwhile, we also want to minimize the time spent in performing the searching task.

In some cases it is acceptable to include some false positive results, i.e., the searching result could contain some tags in  $\mathcal{T} - \mathcal{T} \cap \mathcal{S}$ . For example, when the warehouse manager wants to search a particular set of products with manufacturing flaws, it is acceptable to include a few extra normal products provided that all the flawed products are found [15]. We use the parameter  $\alpha$  to denote the tolerable false positive rate threshold. Denote by  $\mathcal{R}$  the searching result. We aim to guarantee that  $\frac{|\mathcal{R} - (\mathcal{T} \cap \mathcal{S})|}{|\mathcal{T} - \mathcal{S}|} \leq \alpha$ . More specifically, for any tag in  $\mathcal{S}$  but not in  $\mathcal{T}$  (and thus should not be included in the searching result), the probability that it is included in the searching result should not exceed  $\alpha$ .

## 4. The Proposed Protocols

In this section, we first develop the ESiM protocol that pushes per tag energy efficiency to the limit as each tag needs to transmit only one bit to the reader. Then in Section 4.2 we develop the TESiM protocol that uses a novel approach to reduce the execution time but increases energy consumption slightly, and analyze the sensitivity of its execution time to several key system parameters. Finally, we consider the effects of reader

collisions and propose the new reader scheduling algorithm in Section 4.3.

#### 4.1. ESiM: *Energy-Efficient Tag Searching in Multiple Reader RFID Systems*

##### 4.1.1. *Protocol Design*

Our tag searching protocol is motivated by the following observation: In a multiple reader RFID system, if a wanted tag is in the system, it must reside in at least one reader’s interrogation range. In contrast, if a wanted tag is absent from all the readers’ interrogation range, it must be absent from the system. With this observation, for any wanted tag, we test its existence in all the readers’ interrogation range and determine whether it is in the system or not according to the testing results. If a wanted tag is absent from all the readers’ range, it is excluded from the wanted tag set  $\mathcal{S}$ . After all such tags are excluded, the remaining tags constitute the searching result.

ESiM utilizes empty slots in a frame to test whether a wanted tag is in a reader’s interrogation range or not. Let  $\mathcal{T}(R_i)$  denote the *local system tags of reader  $R_i$* , i.e., those tags residing in  $R_i$ ’s interrogation range.  $R_i$  starts a frame with broadcasting two parameters  $f_i$  and  $s_i$ . All the local system tags of  $R_i$  reply with one-bit short responses. The reader then scans the frame and constructs a reply pattern  $RPN = \{b_0, \dots, b_i, \dots, b_{f_i}\}$ , where  $b_i$  indicates the status of the  $i$ -th slot in the frame. If the  $i$ -th slot is empty, then  $b_i = 0$ ; otherwise,  $b_i = 1$ . For every wanted tag  $t \in \mathcal{S}$ , ESiM calculates the *expected slot index  $j$*  for  $t$  assuming that  $t$  is in  $\mathcal{T}(R_i)$ . It then checks  $b_j$  in  $RPN$ . If  $b_j$  is equal to zero, then it can be judged that  $t$  must not be in  $\mathcal{T}(R_i)$  and can be safely excluded from  $\mathcal{S}$ . After all the wanted tags are tested, the remaining tags in  $\mathcal{S}$  represent those wanted tags residing in  $R_i$ ’s range. We call these tags the *local searching result of reader  $R_i$* , and denote

them by  $\mathcal{S}(R_i)$ . After all the readers are tested, we combine all the local searching results to obtain the final result  $\mathcal{R} = \cup_{i=1}^M \mathcal{S}(R_i)$ , where  $M$  is the number of readers.

There may be some false positive results in  $\mathcal{R}$ . For example, for a wanted tag  $t$  that is not in the system, there may happen to be a local tag  $t'$  that selects the same slot as  $t$  does. The reader cannot distinguish  $t'$  from  $t$  with only a one-bit short response in the slot. In this case,  $t$  cannot be excluded from  $\mathcal{S}$  by ESiM, and thus it will be incorrectly included in the local searching result and consequently be included in the final result.

We now present how to set the frame length  $f_i$  to guarantee the false positive rate, i.e., to guarantee that

$$P_{FP} \leq \alpha, \tag{1}$$

where  $P_{FP}$  is the false positive rate (i.e., the probability a wanted tag that does not exist in the system is incorrectly included in the final result), and  $\alpha$  is a system parameter. In order to calculate  $P_{FP}$ , we need to know the probability  $P_w$  that a tag is incorrectly included in the local searching result of a reader  $R_i$  when it is not in  $R_i$ 's range. Recall that a tag not in the system is incorrectly included in the final result only when it is incorrectly included in the local searching results of at least one of the  $M$  readers. Thus,  $P_{FP}$  is given by

$$P_{FP} = 1 - (1 - P_w)^M. \tag{2}$$

Without loss of generality, we consider how to calculate  $P_w$  for the reader  $R_i$ . For a random slot in the frame, the probability that none of the local

system tags of  $R_i$  (i.e., tags in  $\mathcal{T}(R_i)$ ) selects this slot is

$$P_0 = \left(1 - \frac{1}{f_i}\right)^{|\mathcal{T}(R_i)|} \approx e^{-|\mathcal{T}(R_i)|/f_i}. \quad (3)$$

The expected number of empty slots in the frame can thus be calculated as  $N_0 \approx f_i \times P_0$ . If a wanted tag  $t$  is not in  $R_i$ 's range (i.e.,  $t \notin \mathcal{T}(R_i)$ ), only when it selects a non-empty slot in the frame it will be incorrectly included in the local searching result. Thus,  $P_w$  is equal to the probability that  $t$  selects a non-empty slot in the frame, and can be written as

$$P_w = \frac{f_i - N_0}{f_i} \approx 1 - e^{-|\mathcal{T}(R_i)|/f_i}. \quad (4)$$

Substituting Eqs. (2), (3), and (4) into Eq. (1), we get

$$f_i \geq \frac{-|\mathcal{T}(R_i)| \times M}{\ln(1 - \alpha)}. \quad (5)$$

Define the *local load factor of reader  $R_i$*  as  $\rho_i = f_i/|\mathcal{T}(R_i)|$ . From Eq. (5) reveals that in order to guarantee a false positive rate lower than  $\alpha$ , the minimum local load factor for reader  $R_i$  should be  $\frac{-M}{\ln(1-\alpha)}$ .

Figure 1 plots  $\rho_i$  in a system containing 16 readers when  $\alpha$  changes from 0.01 to 0.1. This figure shows that when  $\alpha$  is small,  $\rho_i$  could be vary large, which means that we need to set a very large frame size ( $f_i$ ) to guarantee a low false positive rate. Furthermore, from Eq. (5) we know that the minimum frame length increases linearly along with the number of readers  $M$ . This limits the application of ESiM in very large RFID systems that may contain a huge number of readers. In order to overcome this limitation, we develop

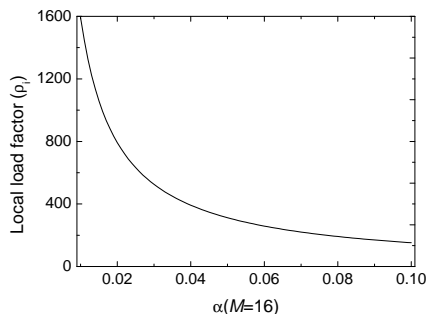


Figure 1: Local load factor ( $\rho_i$ ) when the false positive rate threshold ( $\alpha$ ) varies. The number of readers in the system ( $M$ ) is 16.

the TESiM protocol that reserves the high energy efficiency property of ESiM but dramatically improves the time efficiency and scales well for large scale RFID systems.

#### 4.1.2. Energy Efficiency of ESiM

We now analyze the energy efficiency of ESiM. For active tags, most energy is consumed in transmitting data between the reader and the tag. Thus, the energy efficiency could be measured by the total number of data exchanged between the reader and the tag. For every tag that should be included in the final result (i.e., tags in  $\mathcal{T} \cap \mathcal{S}$ ), it needs to transmit at least one bit to the reader. Otherwise, it is impossible for the reader to judge whether the tag is in the system or not. Thus, the number of bits to be exchanged between tags and readers in *any possible solutions to the tag searching problem* is strictly no less than  $|\mathcal{T} \cap \mathcal{S}|$ . This represents an upper bound on the energy efficiency of any tag searching protocols. ESiM achieves this upper bound when  $\mathcal{T} \subseteq \mathcal{S}$ , i.e., all the tags in the system are wanted tags.

When some tags in the system are not wanted tags, ESiM's energy efficiency is lower than the upper bound. However, in practice, every tag in the system should transmit at least one bit to the reader; otherwise, it is difficult for the reader to judge whether the tag is a wanted tag or not. Thus the total number of bits transmitted from the tags to the readers should be no less than  $|\mathcal{T}|$ . In ESiM, every tag in  $\mathcal{T}$  needs to transmit only one bit data to the reader, so the total number of bits transmitted to the readers in ESiM is exactly  $|\mathcal{T}|$ . Therefore, ESiM actually achieves the limit in energy efficiency in practice, as it requires every tag to transmit only one bit data to the reader.

#### *4.2. TESiM: Time-Efficiency Enhanced ESiM*

##### *4.2.1. Protocol Design*

In order to achieve low false positive rate, ESiM needs to use an extremely long frame to test the existence of wanted tags. This leads to the low time efficiency of the ESiM protocol. Intuitively, we can use a short frame to perform the existence test and reduce the execution time. However, this will lead to higher false positive rate. In order to guarantee that the false positive rate is below the desired threshold, we can perform the existence test for several rounds with several short frames. By carefully selecting the number of frames and the length of each frame, we show that the execution time of ESiM can be dramatically reduced while the transmission overhead of tags remains low. We call this time efficiency enhanced ESiM protocol as TESiM.

In TESiM, every reader  $R_i$  uses  $k$  frames to perform the existence test for wanted tags. In each of the  $k$  frames,  $R_i$  issues a frame with length  $f_k$  and



excludes tags from  $\mathcal{S}$  in the same way as in the ESiM protocol. After the  $k$  frames, the reader takes the remaining tags as the local searching result  $\mathcal{S}(R_i)$ . We can obtain the final result by combining all the local searching results of all the  $M$  readers.

We now analyze how to set  $k$  and  $f_k$  to minimize the execution time. For a wanted tag  $t$  that does not reside in the interrogation range of reader  $R_i$ , the only chance that it remains in the local searching result of  $R_i$  is that TESiM fails to exclude it from  $S$  in all the  $k$  frames. In any frame, the probability that TESiM fails to exclude  $t$  is given by

$$P_{W,1} = 1 - e^{-|\mathcal{T}(R_i)|/f_k}. \quad (6)$$

The probability that TESiM fails to exclude  $t$  after all the  $k$  frames can be written as

$$P_{W,k} = P_{W,1}^k = (1 - e^{-|\mathcal{T}(R_i)|/f_k})^k. \quad (7)$$

If tag  $t$  does not exist in the system (i.e.,  $t \notin \mathcal{S} \cap \mathcal{T}$ ), the probability that it is correctly excluded from the final result equals the probability that it is correctly excluded by all the  $M$  readers, which is

$$P_c = (1 - P_{W,k})^M. \quad (8)$$

So the probability that tag  $t$  is incorrectly included in the final result (thus it is a false positive result) is given by

$$P_{FP,k} = 1 - P_c = 1 - [1 - (1 - e^{-|\mathcal{T}(R_i)|/f_k})^k]^M. \quad (9)$$

In order to guarantee that the false positive rate is below the threshold  $\alpha$ , we should make

$$P_{FP,k} \leq \alpha. \quad (10)$$

Substituting Eqs. (6), (7), (8), and (9) into Eq. (10), we find that in order to guarantee the false positive rate, the length of each frame should satisfy

$$f_k \geq \frac{-|\mathcal{T}(R_i)|}{\ln(1 - (1 - (1 - \alpha)^{1/M})^{1/k})}. \quad (11)$$

It can be seen that when  $\alpha$  is fixed,  $f_k$  is a function of both  $M$  and  $k$ . As the value of  $M$  is often fixed in a warehouse, we only need to consider how to set  $k$  to minimize the searching time, which is given by

$$T_k = k * f_k * t_b. \quad (12)$$

It is obvious that when  $k$  is fixed,  $T_k$  takes its minimal value when  $f_k$  takes the minimal feasible value

$$f_k = \frac{-|\mathcal{T}(R_i)|}{\ln[1 - (1 - (1 - \alpha)^{1/M})^{1/k}]}. \quad (13)$$

In this case, we have

$$T_k = k * t_b * \frac{-|\mathcal{T}(R_i)|}{\ln[1 - (1 - (1 - \alpha)^{1/M})^{1/k}]}. \quad (14)$$

To find the value of  $k$  that minimizes  $T_k$ , we let

$$\frac{\partial T_k}{\partial k} = 0. \quad (15)$$

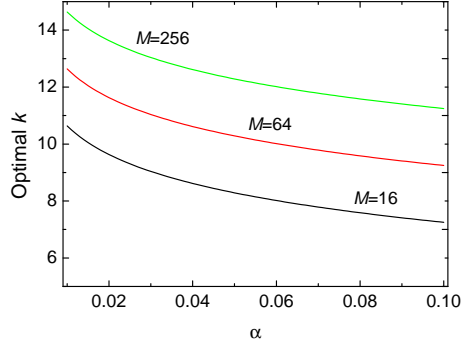


Figure 2: Optimal number of frames ( $k$ ) in TESiM when the false positive rate threshold ( $\alpha$ ) varies.  $M$  indicates the number of readers in the system.

The value of  $T_k$  is minimized when  $k$  satisfies the following equation

$$-|\mathcal{T}(R_i)| \frac{\ln[1 - (1 - (1 - \alpha)^{1/M})^{1/k}] - k \ln k}{\ln^2[1 - (1 - (1 - \alpha)^{1/M})^{1/k}]} = 0. \quad (16)$$

Eq. (16) shows that the optimal  $k$  is determined by both  $\alpha$  and  $M$ . In Figure 2 we plot the optimal  $k$  for different  $\alpha$  when  $M = 16, 64,$  and  $256$ , respectively. It can be observed that  $k$  gradually increases when  $\alpha$  decreases, and it also increases slightly when  $M$  increases. More analysis on the impact of  $M$  and  $\alpha$  on  $T_k$  will be given in the next section.

Similar to that in the ESiM protocol, we define the local load factor for reader  $R_i$  as  $\rho_i = T_k/|\mathcal{T}(R_i)|$  and plot its value for different combinations of  $\alpha$  and  $M$  in Figure 3. Compared with Figure 1, we observe dramatic decrease of  $\rho_i$ . For example, when  $M = 16$  and  $\alpha = 0.01$ ,  $\rho_i = 1592$  in the ESiM protocol and drops to 15.34 in TESiM. The improvement is more than two orders of magnitude. Even when there are as many as 256 readers ( $M = 256$ ),  $\rho_i$  remains small in TESiM: Its value is only 16.23 when  $\alpha = 0.1$

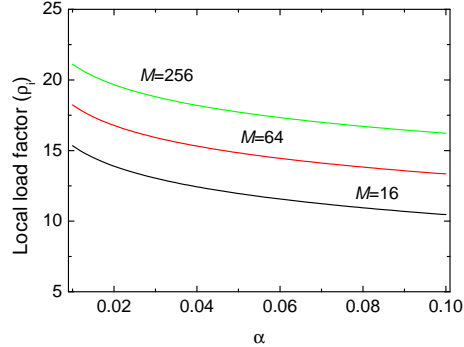


Figure 3: Local load factor ( $\rho_i$ ) when the false positive rate threshold ( $\alpha$ ) varies.  $M$  indicates the number of readers in the system.

and is 21.12 when  $\alpha = 0.01$ .

#### 4.2.2. Sensitivity Analysis on $T_k$

$T_k$  is affected by both the number of readers ( $M$ ) and the required false positive rate ( $\alpha$ ). In this section, we first give a simplified expression of  $T_k$ , then analyze its sensitivity to  $M$  and  $\alpha$ .

We observe that when  $k$  takes optimal value, the ratio of  $f_k$  to  $|\mathcal{T}(R_i)|$  remains at a constant  $\eta = 1.4427 \approx 1/\ln 2$ . With this observation, we can get a simplified expression of  $T_k$  and obtain a closed form solution of optimal  $k$  based on this simplified expression.

According to Eq. (13), when  $k$  takes optimal value, we have

$$\frac{f_k}{|\mathcal{T}(R_i)|} = \frac{-1}{\ln[1 - (1 - (1 - \alpha)^{1/M})^{1/k}]} \approx \frac{1}{\ln 2} = \frac{-1}{\ln \frac{1}{2}}. \quad (17)$$

Note here we use observation that  $f_k/|\mathcal{T}(R_i)| \approx 1/\ln 2$ . According to Eq.

(17), we can get the relationship between  $k$ ,  $\alpha$ , and  $M$  as

$$1 - (1 - (1 - \alpha)^{1/M})^{1/k} = 1/2, \quad (18)$$

which implies that

$$k = \log_{0.5}(1 - (1 - \alpha)^{1/M}). \quad (19)$$

We rewrite the expression of  $T_k$  as

$$\begin{aligned} T_k = f_k \times k &= |\mathcal{T}(R_i)| * \frac{1}{\ln 2} * \log_{0.5}(1 - (1 - \alpha)^{1/M}) \\ &= \frac{-|\mathcal{T}(R_i)|}{\ln^2(2)} * \ln(1 - (1 - \alpha)^{1/M}). \end{aligned} \quad (20)$$

*Sensitivity to  $M$ .* We first analyze the sensitivity of  $T_k$  to  $M$ . Taking the first order derivative of  $T_k$  on  $M$ , we get

$$\frac{\partial T_k}{\partial M} = \frac{-|\mathcal{T}(R_i)|}{\ln^2(2)} * \frac{\ln(1 - \alpha)}{M^2[(1 - \alpha)^{-1/M} - 1]}. \quad (21)$$

Expanding  $(1 - \alpha)^{-1/M}$  with the Taylor series, we get

$$(1 - \alpha)^{-1/M} \approx 1 + (\ln(1 - \alpha)) * \frac{-1}{M} + \frac{(\ln(1 - \alpha))^2}{2!} \frac{1}{M^2} + o(\frac{1}{M^3}). \quad (22)$$

Combining Eq. (21) and Eq. (22), we have

$$\frac{\partial T_k}{\partial M} \approx \frac{-|\mathcal{T}(R_i)|}{\ln^2(2)} * \frac{\ln(1 - \alpha)}{-M \ln(1 - \alpha) + \frac{(\ln(1 - \alpha))^2}{2!} + o(\frac{1}{M})} \propto \frac{1}{M}. \quad (23)$$

It can be seen that the first order derivative is approximately inversely proportional to  $M$ . This means that when  $M$  changes, the change of  $T_k$  will not be large. Thus  $T_k$  is not sensitive to the change of  $M$ .

*Sensitivity to  $\alpha$ .* We now analyze the sensitivity of  $T_k$  to  $\alpha$ . Taking the first order derivative of  $T_k$  on  $\alpha$ , we have

$$\begin{aligned} \frac{\partial T_k}{\partial \alpha} &= \frac{-|\mathcal{T}(R_i)|}{\ln^2(2)} * \frac{1}{M} \frac{1}{(1-\alpha)^{1-1/M} - (1-\alpha)} \\ &\approx \frac{-|\mathcal{T}(R_i)|}{\ln^2(2)} * \frac{1}{M} \frac{1}{\frac{\alpha}{M} + \frac{1-M}{2M^2}\alpha^2 + o(\alpha^2)} \propto \frac{1}{\alpha}. \end{aligned} \quad (24)$$

Similarly, we can observe that the value of the first order derivative is inversely proportional to  $\alpha$ . Thus  $T_k$  is also not sensitive to the changes of  $\alpha$ .

### 4.3. Multiple Reader Scheduling

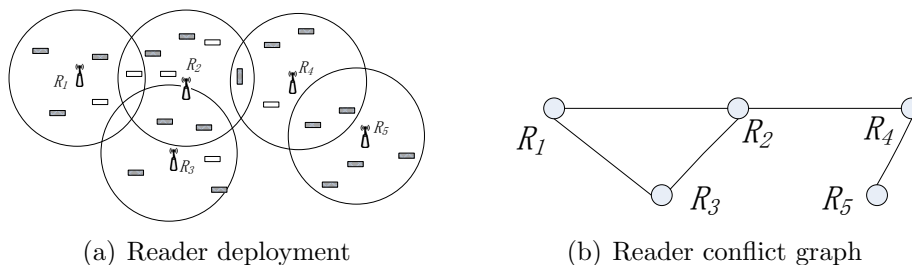


Figure 4: An example reader conflict graph: (a) a system containing five readers; (b) the corresponding conflict graph.

We did not consider the time delay caused by reader scheduling when designing the tag searching protocols in the previous sections. As mentioned in Section 3.1, adjacent readers cannot work together because there may be collisions between them, e.g., R-R collisions or R-T collisions. In this section, we discuss how to schedule readers to avoid such collisions.

Consider that all the  $M$  readers are scheduled to work in  $L$  different

rounds. The total execution time is

$$T_{total} = T_k \times L. \quad (25)$$

In order to minimize  $T_{total}$ , we should minimize  $L$ . In other words, we should use as few rounds as possible to schedule all the readers to work. This naturally maps to the minimum coloring problem on the *conflict graph* of the readers (we will explain how to construct the conflict graph soon). The minimum coloring problem has been proven to be a NP-hard problem [38]. Considering that the conflict graph of RFID readers is usually sparse, we adopt a sequential coloring algorithm called DSATUR [39] that can generate near optimal solution when the graph is sparse. The complexity of the DSATUR algorithm is  $O(n^3)$ , where  $n$  is the number of vertices in the graph.

For the TESiM protocol, there is a little more attention to be paid. Recall that in TESiM per tag energy consumption ( $k$ ) and the execution time in every round ( $T_k$ ) both increase along with  $M$ . Thus, in order to keep  $k$  and  $T_k$  as small as possible, we should use as few readers as possible to cover the whole system. This is different from existing RFID reader scheduling algorithms that aim to maximize identification throughput by scheduling as many readers as possible to work simultaneously. In order to achieve our goal, we add a *reader pruning* phase before finding a feasible scheduling of readers. For each reader, we check whether it is redundant, i.e., its interrogation region can be covered by its nearby readers. We then remove all the redundant readers from the reader set and schedule only the left readers to work.

Algorithm 1 shows our reader scheduling algorithm. First, we prune the redundant readers. For every reader  $R_i$ , we check whether its interrogation

---

**Algorithm 1** Reader Scheduling for ESiM/TESiM

---

- 1: Prune redundant readers, and get the remaining reader set  $\{R_1, \dots, R_{M'}\}$ .
  - 2: Construct the conflict graph  $G = \langle V, E \rangle$  for the remaining reader set.
  - 3: Find a minimum coloring on  $G$  with the DSATUR algorithm.
  - 4: Construct a scheduling of readers based on the coloring result, and activate readers to run ESiM/TESiM in the order determined by the coloring result.
- 

range can be covered by some nearby readers. If  $R_i$  is redundant, it is removed from the reader set. The pruning process is repeated until there are no redundant readers. Second, we construct the conflict graph  $G = \langle V, E \rangle$  according to the remaining readers. In the conflict graph, vertex  $v_i \in V$  corresponds to reader  $R_i$  in the remaining reader set. There is an edge between two vertices  $v_i$  and  $v_j$  if and only if there is an R-R collision or an  $R-T$  collision between  $R_i$  and  $R_j$ . Figure 4 shows an example of the conflict graph constructed for a five reader RFID system. After the conflict graph is constructed, we run the DSATUR algorithm to find a minimum coloring on the conflict graph  $G$ . Finally, we construct a scheduling according to the coloring result: If  $v_i$  is colored with integer  $l$ , we let  $R_i$  work in the  $l$ -th round. Assume that there are totally  $c$  colors in the result, i.e.,  $\{1, \dots, c\}$ . We first activate all the readers with color 1 and let them execute ESiM/TESiM, and then activate all the readers with color 2 to run our protocols, and so on. Note that the parameters in ESiM/TESiM are computed with the number of readers after the pruning phase.



## 5. Performance Evaluation and Comparison

### 5.1. Performance Metrics and Simulation Settings

Three key metrics are used to evaluate the performance of the proposed searching protocol:

- The first metric is the *precision of the searching result*, i.e., the ratio of wanted tags that are actually in the system to the total number of tags in the searching result. This metric is defined as

$$precision = \frac{|\mathcal{S} \cap \mathcal{T}|}{|\mathcal{R}|}, \quad (26)$$

where  $\mathcal{R}$  is the searching result.

- The second metric is *per tag energy consumption*. As the majority of energy is consumed in transmitting data between tags and readers, we use the total number of bits exchanged between a tag and the reader covering it to measure energy efficiency. Note that this metric considers both the data sent to and received from the reader, because for active tags it takes nearly the same energy to send or receive a bit.
- The third metric is the *execution time*, which refers to the time spent in performing the searching task. We use the timing scheme of the Philips I-Code active tags [17] to calculate the execution time.

In the evaluation, we compare ESiM and TESiM with the state-of-the-art CATS protocol proposed in [15] as well as two baseline approaches, namely *Collection* and *Broadcast*. In the *Collection* approach, the readers simply collect IDs of all the tags in the system (i.e.,  $\mathcal{T}$ ) and find the searching result

by comparing  $\mathcal{S}$  and  $\mathcal{T}$ . In the Broadcast approach, the readers broadcast tag IDs in  $\mathcal{S}$  one by one. If a tag finds that the received ID matches its own ID, it transmits a one-bit short response to the reader to notify the matching. The reader then adds the tag into the searching result. After all the tag IDs in  $\mathcal{S}$  have been broadcasted, the readers can find the result.

We consider three parameters that may affect the performance of different protocols. The first parameter is the false positive rate threshold  $\alpha$ , which affects the performance of probabilistic approaches including ESiM, TESiM and CATS. The second parameter is the ratio of wanted tags to the tags in the system, which is defined as  $\gamma = |\mathcal{S}|/|\mathcal{T}|$ . This parameter affects the performance of approaches in which tags need to receive large volume of data from the reader, e.g., Broadcast and CATS. The third parameter is the scale of the system, which is represented by the number of readers deployed in the system. When  $\gamma$  is fixed, this parameter affects the number of tags in  $\mathcal{S}$  and consequently affects the performance of protocols like Broadcast and CATS whose performance heavily depends on the size of  $\mathcal{S}$ . This parameter also affects the performance of TESiM and ESiM because their execution time is affected by the number of readers.

Following previous studies on multiple reader protocols [15, 40], in the default settings we deploy 64 readers in a grid topology to cover a  $10r \times 10r$  area, where  $r$  is the interrogation radius of readers. We randomly distribute 1,000,000 tags in the system, resulting that every reader covers approximately 31400 tags. We take the same default settings as in the CATS protocol, i.e.,  $\alpha = 0.05$  and  $\gamma = 0.1$ .

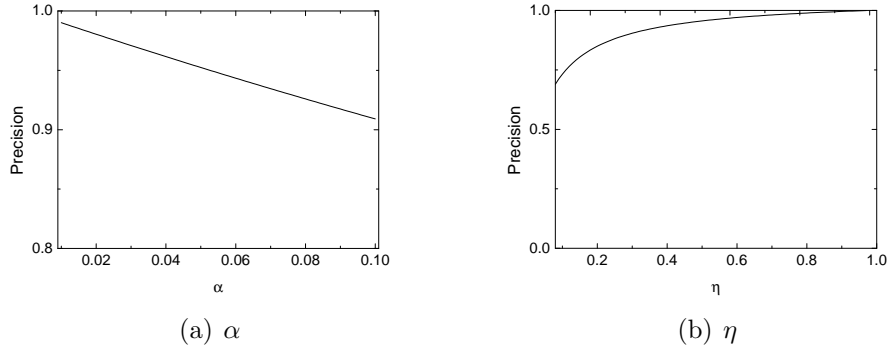


Figure 5: Precision of ESiM and TESiM: (a) when the false positive rate threshold ( $\alpha$ ) changes ( $\eta = 0.5$ ); (b) when the ratio of found wanted tags ( $\eta$ ) changes ( $\alpha = 0.05$ ).

## 5.2. Precision

When  $\alpha$  is fixed, the precision of ESiM and TESiM is determined by the ratio of the number of tags in  $|\mathcal{S} \cap \mathcal{T}|$  to the total number of wanted tags, i.e.,

$$\eta = \frac{|\mathcal{S} \cap \mathcal{T}|}{|\mathcal{S}|}. \quad (27)$$

According to Eq. (26), we have

$$\begin{aligned} precision &= \frac{|\mathcal{S} \cap \mathcal{T}|}{|\mathcal{R}|} \\ &\approx \frac{|\mathcal{S}| * \eta}{|\mathcal{S}| * \eta + |\mathcal{S}| * (1 - \eta) * \alpha} = \frac{\eta}{\eta + (1 - \eta) * \alpha}. \end{aligned} \quad (28)$$

It is obvious that in Collection and Broadcast the precision is always 1. Figure 5(a) plots how the precision of ESiM and TESiM changes when  $\alpha$  changes ( $\eta = 0.5$ ). The precision drops when  $\alpha$  increases but remains high ( $\geq 0.9$ ) even when  $\alpha$  is as large as 0.1. Figure 5(b) shows how  $\eta$  affects the precision of ESiM and TESiM when  $\alpha = 0.05$ . When  $\eta$  is small, the precision

is low. For example, when  $\eta = 0.1$ , the precision is only 0.69. However, when  $\eta$  increases, the precision improves quickly. when  $\eta \geq 0.3$ , the precision is higher than 0.9. In practice, the system operator should have some priori knowledge about the wanted tag set  $\mathcal{S}$ , and thus  $\eta$  should be relatively large, in which case our searching protocols will achieve very high precision.

### 5.3. Energy Consumption

In Figure 6 depicts per tag energy consumption in different protocols. The energy consumption in Collection and ESiM are both constant (1 and 261, respectively). We observe that the energy consumption in Broadcast and CATS are much higher than that in other protocols. Compared with Broadcast and CATS, the per tag energy consumption in TESiM is more than four orders of magnitude lower. Compared with Collection, the energy consumption in TESiM is more than one order of magnitude lower. We further observe that in TESiM and CATS per tag energy consumption increases along with the decrease of  $\alpha$ . However, as we have analyzed in Section 4.2.2, per tag energy consumption in TESiM is not sensitive to the change of  $\alpha$ . In Figure 6(a), when  $\alpha$  drops from 0.1 to 0.01, the total number of bits every tag needs to exchange with readers in TESiM increases only from 9.25 to 12.6. In contrast, in CATS per tag communication overhead increases with a much faster speed, from  $3.6 * 10^5$  bits when  $\alpha = 0.1$  to  $5.1 * 10^5$  bits when  $\alpha = 0.01$ .

The energy consumption in Broadcast and CATS greatly depends on the size of the wanted tag set  $\mathcal{S}$ . In Figure 6(b) we plot the energy consumption of Broadcast, CATS and TESiM when the system scale (measured in the reader number  $M$ ) increases. When the system scales up, per tag energy con-

sumption in Broadcast and CATS increases dramatically. When the reader number increases from 16 to 128, per tag energy consumption in Broadcast and CATS increases 8 times and 3.09 times, respectively. In Broadcast and CATS, every tag needs to receive a large volume of data from the reader that greatly depends on the number of wanted tags ( $|\mathcal{S}|$ ), which increases when the system scales up. In contrast, in TESiM per tag energy consumption increases only 1.36 times in the same scenario, from 8.29 to 11.3. Compared to Broadcast and CATS, the energy consumption in TESiM is far less sensitive to changes of the system scale.

The per tag energy consumption in Broadcast and CATS also depends on  $\gamma$  when the system scale is fixed. Figure 6(c) plots the per tag energy consumption in Broadcast and CATS when  $\gamma$  increases from 0.01 to 0.2. We also plot the data of TESiM for comparison, although in TESiM per tag energy consumption is independent to  $\gamma$  when  $M$  and  $\alpha$  is fixed. We observe 20 times and 6.1 times increase in energy consumption in Broadcast and CATS, respectively, when  $\gamma$  increases from 0.01 to 0.2. Thus the energy consumption in Broadcast and CATS is sensitive to both  $M$  and  $\gamma$ .

#### 5.4. Execution Time

When calculating the execution time of different protocols, we use the timing scheme specified in the Philips I-Code specification [17]. In this specification, two consecutive transmissions are separated by a waiting time of  $302 \mu s$ . Thus  $t_e = 0.302 ms$ . We set the transmission rate at 26.5 Kb/sec, with which it takes  $37.76 \mu s$  to transmit one bit from the tag to reader or vice versa. According to this specification,  $t_{id} = 3.927 ms$  and  $t_b = 0.34 ms$ . For better readability, we do not plot the execution time of ESiM because it

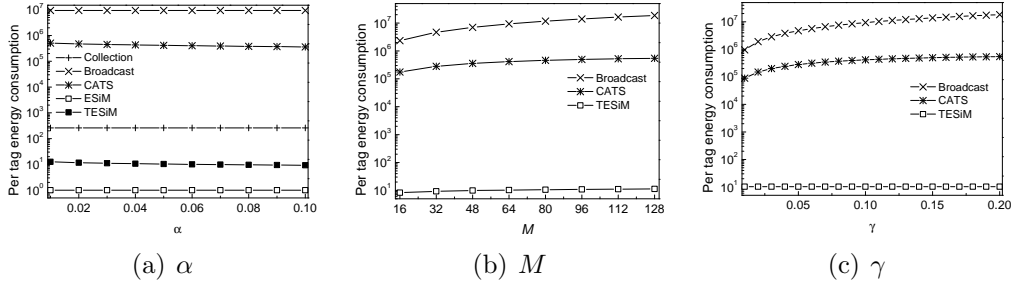


Figure 6: Energy consumption comparison: (a) when the false positive rate threshold ( $\alpha$ ) changes; (b) when the system scale ( $M$ ) changes; (c) when the ratio of the number of wanted tags to the total number of tags in the system ( $\gamma$ ) changes.

is much higher than other protocols. Instead, we list the execution time of ESiM in different simulation settings in Table 1 and discuss the performance of ESiM in a dedicated paragraph.

Figure 7(a) plots the execution time of Collection, Broadcast, CATS and TESiM when  $\alpha$  changes. The execution time of Collection and Broadcast is not affected by  $\alpha$ . In contrast, the execution time of CATS and TESiM increases when  $\alpha$  decreases. The reason is that when  $\alpha$  is large, more false positive results can be tolerated and TESiM can use less rounds to complete the searching task. Compared with Collection, the execution time in CATS and TESiM is reduced by 33% and 56% on average, respectively. We can also observe that the execution time of CATS is more sensitive to  $\alpha$  than TESiM is. When  $\alpha$  decreases from 0.1 to 0.01, the improvement of CATS over Collection drops from 49% to 2%. This indicates that when we require very precise searching result ( $\alpha \leq 0.01$ ), CATS performs nearly the same as or might be worse than Collection. In contrast, in the same scenario TESiM can always effectively improve time efficiency over Collection. The

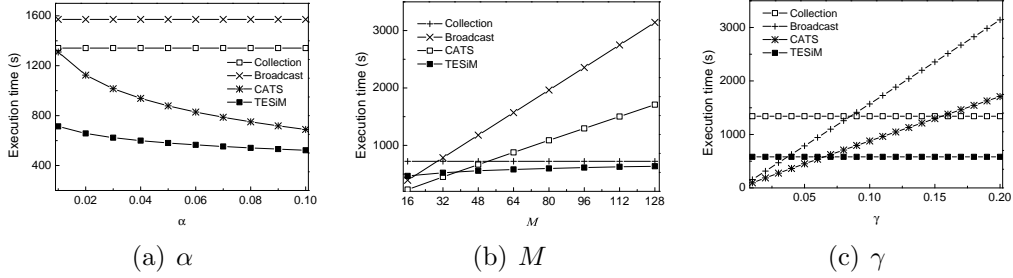


Figure 7: Execution time comparison: (a) when the false positive rate threshold ( $\alpha$ ) changes; (b) when the system scale ( $M$ ) changes; (c) when the ratio of the number of wanted tags to the total number of tags in the system ( $\gamma$ ) changes.

improvement of TESiM over Collection drops only slightly, from 61% to 48%, when  $\alpha$  decreases from 0.1 to 0.01. Thus TESiM can effectively reduce searching time even when the precision requirement is very high.

Figure 7(b) shows how the system scale affects the execution time of different protocols. The searching time of Collection is almost not affected by the system scale. This is reasonable because the execution time of Collection depends on only the number of local tags in one reader's interrogation range and the number of rounds used to schedule all the readers. The execution time in other three protocols increases when the system scales up. For Broadcast and CATS, the increase in execution time is mainly due to the increase in the number of wanted tags ( $|\mathcal{S}|$ ). We can observe that the execution time increases more significantly in Broadcast than in CATS. However, the increase in the execution time of TESiM is not due to the increase of  $|\mathcal{S}|$ . As we have pointed out in Section 4.2, when  $M$  increases, TESiM needs more rounds to guarantee that the false positive rate does not exceeds  $\alpha$ , and thus the frame size in each round is larger. Furthermore, the execution time

increase in TESiM is much slower than that in Broadcast and CATS.

We also observe crossovers between the execution time of different protocols in Figure 7(b). When the system scale is small (e.g.,  $M < 48$ ), the execution time of CATS is shortest, even shorter than TESiM. The execution time of Broadcast is also shorter than that of Collection when  $M \leq 32$ . However, the execution time of Broadcast and CATS increases rapidly when  $M$  becomes large. They even use longer time than Collection when  $M > 32$  (Broadcast) and  $M > 48$  (CATS), respectively. In contrast, the execution time of TESiM is still 12% less than that in Collection even when there are as many as 128 readers.

We see similar change trend in the execution time of the four protocols in Figure 7(c) and also observe crossovers. The execution time of Broadcast and CATS increases when  $\gamma$  increases, because the number of wanted tags  $|\mathcal{S}|$  increases when  $\gamma$  becomes large. However, the execution time of TESiM remains unchanged when  $\gamma$  increases. Compared with Collection, the execution time in TESiM is 55% less in average. In most cases, the execution time of TESiM is less than that of Broadcast and CATS. Compared with Broadcast and CATS, the execution time of TESiM is 62% and 34% less respectively when  $\gamma = 0.1$ , and is 81% and 66% less respectively when  $\gamma = 0.2$ .

Broadcast and CATS perform well when the number of wanted tags is extremely small. For example, when  $M = 16$  and  $\gamma = 0.1$ , the execution time of CATS is only about 1/3 of the execution time of Collection, or about one half of the execution time of TESiM. In the default setting where  $M = 64$  and  $\gamma = 0.01$ , the execution time of CATS is only about 1/7 of the execution time of Collection, and only one third of the execution time of TESiM. However,



Table 1: Execution time of ESiM

$\alpha(M = 64)$	0.02	0.04	0.05	0.06	0.07	0.08	0.09	0.19
Time( $\times 10^4$ s)	12.0	5.95	4.74	3.93	3.35	2.91	2.58	2.31
$M(\alpha = 0.05)$	16	32	48	64	80	96	112	127
Time( $\times 10^4$ s)	1.18	2.37	3.55	4.74	5.92	7.10	8.29	9.47

the execution time of CATS increases quickly and is longer than that of TESiM when  $M > 32$  and  $\gamma > 0.6$ , respectively. To summarize, when  $|\mathcal{S}|$  is extremely small, Broadcast and CATS can be applied if we only consider time efficiency. In other cases, TESiM is more suitable.

In Table 1 we list the execution time of ESiM in different settings. The execution time of ESiM is much longer than other protocols, about one order of magnitude longer. We can also observe that the execution time of ESiM increases when  $\alpha$  decreases and when the system scale ( $M$ ) increases. Thus, ESiM is suitable in cases where energy consumption is the top consideration and a long execution time can be tolerated.

## 6. Conclusion

As a key enabling technology of IoT, RFID have attracted a lot research attention in recent years. RFID tag searching is very important to many industrial applications, e.g., warehouse management in logistics industry, and inventory control in retailing industry. Although there are some prior studies on improving time efficiency of tag searching, energy efficiency in tag searching has not been investigated thoroughly. In this paper, we study the tag searching problem from an energy efficient angle. Two energy efficient tag searching protocols are proposed for large scale RFID systems built with active tags: ESiM and TESiM. ESiM is extremely energy efficient as it re-

quires each tag to exchange only one bit data with readers, but its execution time may become long in large-scale RFID systems. TESiM greatly reduces the execution time while increasing per tag energy consumption only slightly, achieving a better balance between energy consumption and execution time. The per tag energy consumption in TESiM is more than one order of magnitude less than the best of existing solutions. Moreover, compared with state-of-the-art solutions to tag searching, in most cases TESiM even reduces execution time by more than 50 percent.

### **Acknowledgement**

This work is partially supported by the National Science Foundation of China (Grant No.61073036, 61103203, and 61232001/F02). We appreciate the reviewers' comments on improving the quality of this paper.

### **References**

- [1] L. Atzori, A. Iera, G. Morabito, The Internet of things: A survey, *Computer Networks* 54 (15) (2010) 2787–2805.
- [2] M. A. Feki, F. Kawsar, M. Boussard, L. Trappeniers, The internet of things: The next technological revolution, *Computer* 46 (2) (2013) 24–25.
- [3] Y.-J. Huang, W.-C. Lin, H.-L. Li, Efficient implementation of RFID mutual authentication protocol, *IEEE Transactions on Industrial Electronics* 59 (12) (2012) 4784–4791.

- [4] E. Masciari, Smart: Stream monitoring enterprise activities by rfid tags, *Information Sciences* 195 (2012) 25–44.
- [5] X. Zhu, S. K. Mukhopadhyay, H. Kurata, A review of RFID technology and its managerial applications in different industries, *Journal of Engineering and Technology Management* 29 (1) (2012) 152–167.
- [6] P.-S. Fan, C.-C. Tsao, Y.-C. Liaw, Exploring the innovative value of the RFID industry, *International Journal of Automation and Smart Technology* 3 (3) (2013) 193–201.
- [7] C. Sun, F. Jiang, S. Jiang, Research on RFID applications in construction industry, *Journal of Networks* 8 (5) (2013) 1221–1228.
- [8] R. Kuo, S. Hung, W. Cheng, Application of an optimization artificial immune network and particle swarm optimization-based fuzzy neural network to an rfid-based positioning system, *Information Sciences* 262 (2014) 78–98.
- [9] X. Liu, B. Xiao, S. Zhang, K. Bu, Unknown Tag Identification in Large RFID Systems: An Efficient and Complete Solution, *IEEE Transactions on Parallel and Distributed Systems* pp (99) (2014) 1–14.
- [10] D. K. Klair, K.-W. Chian, R. Raad, A survey and tutorial of RFID anti-collision protocols, *IEEE Communications Survery and Tutorials* 2 (3) (2010) 400–421.
- [11] D. Shih, P. Sun, D. Yen, S. Huang, Taxonomy and survey of RFID anti-collision protocols, *Computer communications* 29 (11) (2006) 2150–2166.

- [12] R. Das, P. Harrop, Rfid forecasts, players and opportunities 2014-2024, [http://www.centrenational-rfid.com/docs/users/file/RFID\\_Forecasts\\_2014\\_2024.pdf](http://www.centrenational-rfid.com/docs/users/file/RFID_Forecasts_2014_2024.pdf) (February 14 2014 [Online]).
- [13] P. Harrop, R. Das, Active RFID and sensor networks 2011-2021, RFID Report, IDTechEx, Cambridge, UK.
- [14] Y. Qiao, S. Chen, T. Li, S. Chen, Energy-efficient polling protocols in RFID systems, in: Proc. of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc), 2011, Article no. 25.
- [15] Y. Zheng, M. Li, Fast tag searching protocol for large-scale RFID systems, IEEE/ACM Transactions on Networking 21 (3) (2013) 924–934.
- [16] Q. Xiao, K. Bu, B. Xiao, Efficient monitoring of dynamic tag populations in RFID systems, in: Proc. of the IFIP 9th International Conference on Embedded and Ubiquitous Computing (EUC), 2011, pp. 106–113.
- [17] P. Semiconductors, I-code smart label rfid tags, available: [http://www.nxp.com/documents/data\\_sheet/SL092030.pdf](http://www.nxp.com/documents/data_sheet/SL092030.pdf) (April 2013 [Online]).
- [18] Z. Bin, M. Kobayashi, M. Shimizu, Framed aloha for multiple RFID objects identification, IEICE Transactions on Communications 88 (3) (2005) 991–999.
- [19] C. Law, K. Lee, K.-Y. Siu, Efficient memoryless protocol for tag identification, in: Proc. of the 4th International Workshop on Discrete

Algorithms and Methods for Mobile Computing and Communications, 2000, pp. 75–84.

- [20] B. Sheng, Q. Li, W. Mao, Efficient continuous scanning in RFID systems, in: Proc. of the 29th IEEE International Conference on Computer Communications (Infocom), IEEE, 2010, pp. 1–9.
- [21] X. Liu, S. Zhang, K. Bu, B. Xiao, Complete and fast unknown tag identification in large RFID systems, in: Proc. of 9th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), IEEE, 2012, pp. 47–55.
- [22] X. Liu, K. Li, Y. Shen, G. Min, B. Xiao, W. Qu, H. Li, A fast approach to unknown tag identification in large scale RFID systems, in: Proc. of 22nd IEEE International Conference on Computer Communications and Networks (ICCCN), IEEE, 2013, pp. 1–7.
- [23] T. Li, S. Chen, Y. Ling, Identifying the missing tags in a large RFID system, in: Proc. of the 11th ACM international symposium on Mobile ad hoc networking and computing (Mobihoc), ACM, 2010, pp. 1–10.
- [24] W. Luo, S. Chen, T. Li, S. Chen, Efficient missing tag detection in RFID systems, in: Proc. of the 30th IEEE International Conference on Computer Communications (Infocom), 2011, pp. 356–360.
- [25] W. Luo, S. Chen, T. Li, Y. Qiao, Probabilistic missing-tag detection and energy-time tradeoff in large-scale RFID systems, in: Proceedings of the 13th ACM international symposium on Mobile Ad Hoc Networking and Computing (Mobihoc), ACM, 2012, pp. 95–104.

- [26] S. Zhang, X. He, H. Song, D. Zhang, Time efficient tag searching in multiple reader rfid systems, in: Proc. of IEEE International Conference on Internet of Things, 2013, pp. 481–488.
- [27] B. Zhen, M. Kobayashi, M. Shimizu, Framed ALOHA for multiple RFID objects identification, IEICE Transactions on Communications 88 (3) (2005) 991–999.
- [28] V. Namboodiri, L. Gao, Energy-aware tag anticollision protocols for RFID systems, IEEE Transactions on Mobile Computing (2009) 44–59.
- [29] D. Klair, K. Chin, R. Raad, An investigation into thie energy efficiency of pure and slotted aloha based RFID anti-collision protocols, in: Proc. of IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2007, pp. 1–4.
- [30] J. Wang, H. Hassanieh, D. Katabi, P. Indyk, Efficient and reliable low-power backscatter networks, ACM SIGCOMM Computer Communication Review 42 (4) (2012) 61–72.
- [31] D. Husn, C. Wood, Analysis of tree algorithm for RFID arbitrtion, in: Proc. of the IEEE International Symposium on Information Theory, 1998, pp. 107–107.
- [32] EPCglobal, Brussels, Belgium, EPC radio-frequency identity protocols class-1 gen-2 uhf RFID protocol for communications at 860 mhzc960 mhz, available: <http://www.epcglobalinc.org/standards/uhfc1g2> (April 2013 [Online]).

- [33] X. Liu, S. Zhang, K. Bu, B. Xiao, Complete and fast unknown tag identification in large RFID systems, in: Proc. of the 9th IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), 2012, pp. 47–55.
- [34] J. Waldrop, D. Engels, S. Sarma, Colorwave: An anticollision algorithm for the reader collision problem, in: Proc. of IEEE International Conference on Communications (ICC), Vol. 2, 2003, pp. 1206–1210.
- [35] Z. Zhou, H. Gupta, S. Das, X. Zhu, Slotted scheduled tag access in multi-reader RFID systems, in: Proc. of the 15th IEEE International Conference on Network Protocols (ICNP), 2007, pp. 61–70.
- [36] S. Tang, J. Yuan, X. Li, G. Chen, Y. Liu, J. Zhao, Raspberry: A stable reader activation scheduling protocol in multi-reader RFID systems, in: Proc. of the 17th IEEE International Conference on Network Protocols (ICNP), 2009, pp. 304–313.
- [37] S. Tang, C. Wang, X. Li, C. Jiang, Reader activation scheduling in multi-reader RFID systems: A study of general case, in: Proc. of IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2011, pp. 1147–1155.
- [38] V. V. Vazirani, Approximation Algorithms, Springer, 2001, Ch. 2, pp. 15–17.
- [39] D. Brélaz, New methods to color the vertices of a graph, Communications of the ACM 22 (4) (1979) 251–256.

- [40] R. Zhang, Y. Liu, Y. Zhang, J. Sun, Fast identification of the missing tags in a large RFID system, in: Proc. of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), IEEE, 2011, pp. 278–286.