

## Research Article

# Flood Diversion Algorithm for Anticollision in RFID System

**Hua Huo, Jun Qiang Liu, and Yong Jie Wang**

*School of Software, Key Laboratory of Intelligent Information Processing, Henan University of Science and Technology, Luoyang 471003, China*

Correspondence should be addressed to Hua Huo; [huohua\\_haust@163.com](mailto:huohua_haust@163.com)

Received 25 October 2014; Accepted 27 February 2015

Academic Editor: Luca Catarinucci

Copyright © 2015 Hua Huo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Radio frequency identification (RFID) provides a contactless approach for object identification. If there are multiple tags in the interrogation zone of a reader, tag collision occurs due to radio signal interference. To solve tags identification collision and improve identification efficiency in RFID system, a flood division anticollision (FDAC) algorithm has been presented. Firstly, the algorithm launches an estimation of the number of tags and according to the estimation result decides whether a flood diversion processing needs to be started or not. Secondly, when the flood diversion processing needs to be done, all tags are grouped and assigned to different models in which the tags are to be processed and identified in parallel. Thirdly, in the identification processing, for reducing data transmission, the reader needs only to send a three-dimensional-vector command to tags, tags respond to the command with part of collision-bit parameters, and stack and queue are adopted to store precious request command and tags' ID to avoid repeatedly transmitting them between the reader and the tags. Simulation experiment results show that FDAC is superior to the dynamic frame slotted (DFS) Aloha algorithm, the binary-retreat tree algorithm (BRT) and the dynamic binary-search tree (DBST) algorithm, in the performances of data bits transmission, identification time delay, and energy consumption by the reader.

## 1. Introduction

Radio frequency identification (RFID) is an automatic identification scheme which is a contactless, low power, and low cost wireless communication technology. In recent years, a still growing number of articles have developed the RFID in technology and engineering, such as the cooking and dietary management system, the meat freshness monitoring system, the smooth medical service, the equipment management system via the wireless sensor network, the wireless monitoring of household electrical power meter, the system for tracking and behavior analysis of small animals [1–3], the intelligent traffic flow control system, and the building access. In addition, more and more novel RFID sensing systems have been designed to satisfy some special requirements in engineering, and many effective measures have been proposed to improve the technology on RFID [4–7].

An RFID system consists of a reader and one or more tags. Tag is attached to object; a reader recognizes an object by issuing RF signals to the attached tag [8]. RFID tag contains two major parts. One is an integrated circuit for storing and processing information, modulating, and demodulating

an RF signal and perhaps other specialized functions. The second is an antenna to receive and transmit the signal. In an RFID system, every tag has a unique identification code (ID) [9]. The length of ID may be different in different RFID standards.

While a tag and a reader are close enough, they can communicate with each other. For such a situation, we say that the tag is in the interrogation zone of the reader. To figure out which tags are within the interrogation zone, a reader initiates an interrogation procedure to request tags to send back their IDs. When multiple tags respond to the reader simultaneously, tag collisions occur and no tag can be identified by the reader successfully [10]. The RFID system will become very low efficient and high power consumption while tag collision occurs frequently [11, 12]. Reducing the tag collisions and signal denoise [13] are often employed to improve efficiency of the RFID system.

How to reduce tag collisions to speed up the identification procedure and lower energy consumption is thus important. There are several tag anticollision protocols proposed for reducing tag collision. To achieve this goal of efficient tag reading, various tag reading protocols have been proposed in

the literature. They are classified into two categories: probabilistic protocols and deterministic protocols. Probabilistic protocols are based on the Aloha framework. In Aloha-based algorithms, tags respond to the reader by transmitting IDs in a probabilistic manner. For example, in slotted Aloha algorithms, the whole interrogation procedure period is divided into several time slots, and each tag randomly chooses a time slot for transmitting its ID to the reader. Aloha-based algorithms are simple; however, they have the tag starvation problem that a tag may never be successfully identified because its responses always collide with others [14–17].

Deterministic protocols are based on binary tree framework where each root-to-leaf path represents a unique tag ID. The basic idea of the binary tree-based algorithms is to repeatedly split the tags encountering collisions into subgroups until there is only one tag in a subgroup to be identified. The binary tree-based algorithms do not have the problem of tag starvation [18].

Both of the above category algorithms have their advantages and shortcomings [14]. This paper presents a flood division anticollision (FDAC) algorithm, which uses the flood division idea and integrates the advantages of both Aloha-based algorithm and classical binary tree-based algorithm. FDAC firstly estimates the number of tags and then deals with tags by parallel processing with flood diversion idea according to the estimated result. FDAC makes the reader only send one three-dimensional-vector command, and then tags respond to the command with part of collision-bit parameters, which makes both read-write operations and collision processing work simultaneously and the read-write time can be ignored. We simulate and analyze FDAC and compare it with other algorithms in terms of the performances of data transmitting, time delay, and energy consumption.

The rest of this paper is organized as follows. Some related work is introduced in Section 2. In Section 3, we describe FDAC by elaborating the concepts and mechanism of anticollision in the RFID system. In Section 4, we simulate and analyze FDAC and compare it with related algorithms. And finally, conclusion is drawn in Section 5.

## 2. Related Work

Almost all of the tag reading protocols are classified into two types: probabilistic protocols and deterministic protocols; the former use probabilistic algorithm and the latter use deterministic algorithm sometimes which may be a hybrid of the two type algorithms [16].

*2.1. Probabilistic Algorithm.* Aloha-based protocols are classical probabilistic algorithms, which try to stagger tag response times in a probabilistic manner to reduce collisions. There are several variants of Aloha-based protocols, such as classical Aloha (CA) protocol, slotted Aloha (SA) protocol, frame slotted Aloha (FSA) protocol, and dynamic frame slotted (DFS) Aloha protocol [19]. In these protocols, each tag reading round comprises three phases. The first phase is the

broadcast phase, where the reader broadcasts the frame size to all tags. Frame size refers to the number of time slots available in a frame [20]. The second phase is the transmission phase, during which each tag randomly chooses a time slot within the frame and transmits its ID number. If more than one tag chooses the same time slot, their transmissions collide and the slot will ultimately be wasted. A slot is also wasted when none of the tags choose it for transmitting their IDs. Only those slots that are chosen by exactly one tag each end up actually being used and such slots are also referred to as successful slots [21]. The third phase is the acknowledgment phase, wherein the reader informs a tag if its transmission was successful or not. While the aforesaid working is common for all the probabilistic protocols, they differ in terms of the way in which they choose the frame size and the way in which they choose the tags that respond in a given round [22].

In classical Aloha protocol, while receiving the reader's interrogation request, each tag in the interrogation zone independently chooses a random back-off time and responds with its tag ID to the reader at that time. If an ID is received by the reader without collision, it can be identified properly and acknowledged by the reader. A tag with acknowledged ID will stop responding to the reader. On the other hand, an unacknowledged tag will repeatedly select a random back-off time to send its ID until it is identified and acknowledged by the reader [23].

In the slotted Aloha protocol, the random back-off time must be a multiple of a prespecified slot time. If collisions occur in a slot, the reader will notify the colliding tags to reselect a response time randomly. As it is shown, the performance of SA is twice that of CA since there is no partial collision of tag ID responses in the former protocol [24].

The frame slotted Aloha protocol is similar to SA. However, to limit the response time, FSA divides the whole interrogation procedure into a set of frames. Each frame has a fixed number of time slots, and a tag sends its ID to the reader in only one randomly chosen slot during a frame period. In FSA, when collisions occur, FSA adopts the avoiding strategy which is waiting for the next round to send response again. The efficiency of FSA very closely related to the number of tags. According to one statistic, lots of time slots are not fully utilized when the number of tags is less than the number of time slots which results in a very low efficiency of FSA. While the two numbers are approximately equal, the maximum efficiency is only approximate to 36.8%. Once the number of tags is greater than the number of frame time slots, collision increases significantly and the efficiency fell sharply [25].

The dynamic frame slotted Aloha protocol tries to eliminate the drawback by dynamically adjusting the frame size according to the estimated number of tags. It has better performance than SA and FSA. But it needs many communication rounds to optimize the frame size before the identification process. Under the assumption that the tag IDs are with the same series in production (i.e., tags have the continuous tag ID numbers), [22] proposed a lottery frame protocol to reduce the number of communication rounds with the help of the geometric distribution hash function. Although DFS improves the tags identification efficiency by

dynamically changing the frame time slot, its efficiency has not been satisfactory.

In general, probabilistic algorithms are simple and have fair performance. However, some probabilistic algorithms have the tag starvation problem that a tag may never be identified when its responses always collide with others' or need many data bits transmitted in the communication rounds [26].

**2.2. Deterministic Algorithm.** Tree-based (TB) protocols are classical deterministic algorithms, which rely on tag IDs to repeatedly split colliding tags into subgroups until there is only one tag in a subgroup to be identified successfully. The query tree (QT) protocol and the binary tree (BT) protocol are two representatives of tree-based protocols [27].

In the query tree (QT) protocol, a reader first broadcasts a bit string  $S$  which has a specified length. The tag with an ID whose prefix matches with  $S$  will respond with its whole ID to the reader. If only one tag responds at a time, the tag is identified successfully. But if multiple tags respond simultaneously, the responses collide. In such case, the reader appends string  $S$  with bit 0 or 1 and broadcasts again the longer bit string. In this manner, the colliding tags are divided into two subgroups. If there is only one tag in a subgroup, it can be identified successfully. The reader keeps track of the request strings needed to broadcast with the help of a stack and perform tag identification procedure until all tags are identified [28]. QT does not have the tag starvation problem and its identification time delay is affected by the distribution and the length of tag IDs. Specifically, if the tags have long and continuous IDs, the request bit string will grow very quickly for identifying all tags. The delay time of the identification procedure will then increase significantly [29].

In the binary tree (BT) protocol, while receiving a reader's interrogation request, each tag responds with the first bit of its tag ID. The reader then records and broadcasts 1 (resp., 0) if the received bit is 1 (resp., 0 or a colliding signal). Only the tags with the first bit being 1 (resp., 0) will respond with its next ID bit; other tags will go into a sleep mode [25]. The above procedure will be repeated bit-by-bit until the last ID bit is reached. The reader can then identify and mute one tag and reset tags in the sleep mode to go through the interrogation procedure from some ID bit position. The bit-by-bit procedure is performed recursively and all tags can be identified. BT requires tags to be equipped with writable on-tag memory so that tags can keep track of the inquiring bit position [30, 31].

In recent reference, there are three improved variants of the binary tree protocol, that is, binary-search tree (BST) protocol, binary-retreat tree (BRT) protocol, and dynamic binary-search tree (DBST) protocol. BST adopts a recursive method to perform tags identification, in which all ID bits of a tag are used as transmitting parameters in the recursive process. And after finishing each tag's identification, the system will repeat sending the command used in the last time identification, where sometimes the command may not be useful. So, the disadvantage of BST is too much information needed for transmission which leads to low identification

efficiency [32]. In BRT, the data structure technologies such as stack and queue are used to store and remember the previously used commands which will only be imported from the stack and queue while they need to be used again. So, the use of the data structure technologies, in part, makes BRT reduce the number of times of tags identification and the amount of information that needed transmitting [26]. In DBST, the tag ID is divided into two parts: known part and unknown part. For example, the tag ID has  $n$  bits; bit 1 to bit  $x - 1$  is the known part and bit  $x$  to bit  $n$  is the unknown part. In the search process, only the known part needs to be transmitted and then if the tags are matched, by the known part, the unknown part of these tags will be transmitted and used as response, or else the unknown part will not be transmitted [33]. This technology also partly helps to reduce the amount of information to be transmitted and the time of transmitting in DBST.

**2.3. Energy Consumption of RFID System.** RFID systems consist of a reading device called an RFID reader and a finite number of tags. RFID systems can be divided into two classes: active RFID systems and passive RFID systems. The readers in both active and passive RFID systems have their own batteries. In an active RFID system, all tags are active tags which are provided with energy by their own batteries. In a passive RFID system, all tags are passive tags which have no battery and rely on RF energy transferred from the reader [34]. The energy consumed by the reader and a tag is an important issue that affects battery life time of mobile reader and active tags [35]. This issue has a twofold cause: on the one hand, the radio transceiver implies a higher power consumption compared to the other components of the embedded device; on the other hand, the communication phase is associated with phenomena such as collisions, overhearing (i.e., listening of messages addressed to another node), overmitting (i.e., transmission of data to a node that cannot receive them), and idle listening (i.e., listening to the channel in absence of communications), which substantially reduce the nodes battery [11]. To evaluate and reduce energy consumption, many works have been presented in the literatures [34, 36, 37]. Most of these works are mainly focused on energy evaluating or saving solutions on the level of anticollision protocols. In recent researches [11, 38, 39], Catarinucci et al. had proposed the most effective method by combining new MAC protocols with hardware solutions to further reduce the node's power consumption in wireless sensor networks.

### 3. The Proposed Algorithm

**3.1. Idea of the Proposed Algorithm.** Both DFS and BRT have their respective advantages. For example, the DFS algorithm is simple and rapid when tags are less and deterministic, and the BRT algorithm also has good identification efficiency because it utilizes data structure to reduce the number of times of tags identification and the amount of information that needs transmitting. In this paper, we try to utilize the advantages of both DFS and BRT by performing a novel

method which is a hybrid algorithm of DFS and BRT. The flood division method is suitable for being used to parallel processing the unidentified tags, especially while the number of tags is large and nondeterministic. We call this method a flood division anticollision (FDAC) algorithm.

FDAC comprises three phases. In the first phase, FDAC first starts a simple estimation with the collision-bits detecting method to estimate the number of collision tags. Then, FDAC judges whether an accurate estimation needs to be done or not according to the simple estimation result. If the number of collision tags is greater than a predefined threshold, an accurate estimation process will be started. In this case, all collision tags are flood-divided and distributed into multiple modules automatically, in which the collision tags are processed in parallel. In the third phase, every module takes process and identification for all assigned tags with stack storage, queue storage, back strategy, collision-bit tracking technologies, and so forth. In the process, the reader only needs to send a three-dimensional-vector command, and then the tags respond to the command with part of the collision-bit parameters. Finally, the IDs of all successfully identified tags will be saved into a data structure-queue, and the operation of read-write for the tags is performed under the control of the processing center. The process of FDAC is shown in Figure 1.

### 3.2. Key Terms and Commands Explanation

(1) *Division Modules.* The modules are divided into several parts: module\_1, module\_2, module\_3, module\_4, and so forth, which are labeled, respectively, with module IDs. The thresholds (which can be changed) are set and used to start modules working according to the number of collision tags. For example, if the number of collision tags is in the range of 1–16, only module\_1 starts working, when the number is in 17–64, two modules (module\_1, module\_2) start working, when the number is in 65–521, three modules (module\_1, module\_2, and module\_3) start working, when the number is in 513–2048, four modules (module\_1, module\_2, module\_3, and module\_4) start working, and so forth.

(2) *Module Labeling.* Module labeling is an operation to assign IDs to the started working modules, and the IDs are sequences of 0 or 1. The module labeling operation does not need to be done if the RFID tags are less and need only one module to work. If two modules (i.e., module\_1 and module\_2) work, the two modules are labeled with 0 and 1, respectively. If three or four modules (i.e., module\_1, module\_2, module\_3, or module\_4) work, they are labeled with 00, 01, 10, or 11, and so on when more modules need to work.

(3) *Tag Collision Criterion.* Tag collision criterion is whether there are collisions between the parameters sent by tags. If only one collision bit is detected, two collision tags can be directly identified without the need for a command from reader. If collision does not happen, we can conclude that only one tag exists in the system which can be identified directly.

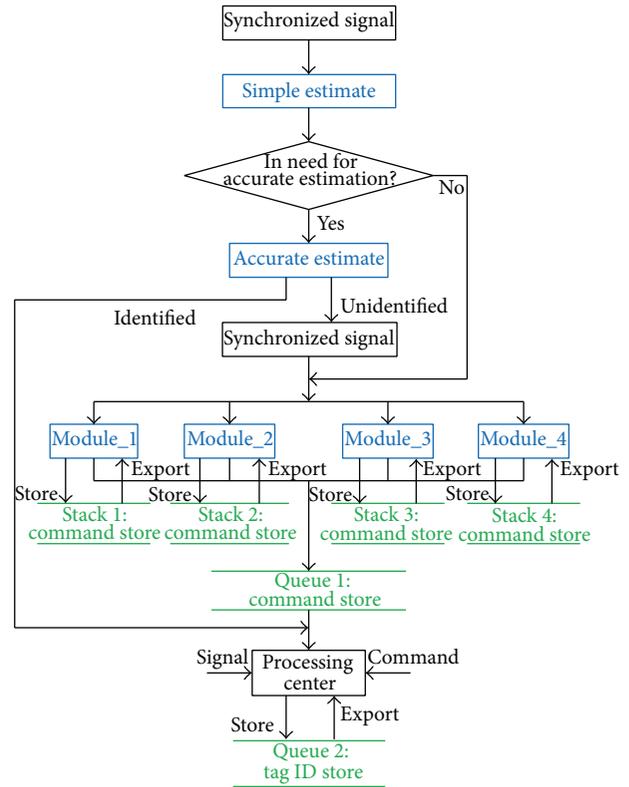


FIGURE 1: Flow chart of flood division anticollision (FDAC) algorithm.

When a collision tag needs to respond to the reader, only the parameters (0 or 1) of the collision bits (not all bits of the tag ID) are sent to the reader, which can decrease the data transmitted between the collision tag and the reader.

(4) *Back Tracking Strategy.* After a request command from reader is stored into a stack, it can be called out from the stack while it is in need again. This strategy is also called retrospective strategy.

(5) *Parallel Processing.* Parallel processing exists in two levels. Firstly, among the modules, all started modules work and process simultaneously. Secondly, in the module, the read-write operation and the collision processing are done simultaneously.

(6) *Collision-Bit Tracking.* When a reader sends a three-dimensional-vector command to the tags, each tag responds to the command with the collision-bit parameters of the tag ID. The collision bits are basic clue and are used in the process whether the reader sends a command or the tags respond to the command. So, tag collision identifying can be simply named collision-bit tracking.

(7) *Request Command.* When collision occurs, a request command is to be sent to the collision tags from the reader. For example,  $REQUEST(B, P, M)$  is a request command, “ $B$ ” represents a collision bit, “ $P$ ” is the collision parameter (0/1)

of the collision bit, “ $M$ ” is a mark of a module in which the REQUEST command exits, and “ $M$ ” may be NULL when only module\_1 works. When collision is detected, the reader will send a REQUEST command again to the collision tag in which the parameter “ $B$ ” is assigned a value with the highest bit of the collision bits that the collision tag responded to the last REQUEST of the reader with. If no collision occurs, the back tracking strategy will be started by the reader. In the process of the request command, the collision tag does not every time need to send the completed ID to the reader (only in the first time of the REQUEST), and meanwhile invalid commands are avoided as far as possible. So, the data bits that need to be transmitted can be reduced to a great extent.

(8) *Answer Command.* The answer command is used to answer to the request command of the reader. For example, ANSWER( $f$ ) is an answer command in this algorithm and the parameter “ $f$ ” is a sequence comprised of the other collision bits which is lower than “ $B$ ”.

3.3. *Setting of Tag State.* One tag, while it is within the interrogation zone of the reader, always possesses one of the four states: *Standby*, *Silence*, *Killed*, and *Ready*.

*Standby.* After having done a response with ANSWER( $f$ ) to a REQUEST command, the tag turns into the *Standby* state, and the REQUEST is stored.

*Silence.* After the tag ID is identified, the tag turns into the *Silence* state and does not participate in the competition with other tags and waits for writing the operation.

*Killed.* After completing the writing operation, the tag turns into *Killed* state and exits from the set of tags.

*Ready.* After REQUEST( $B, P, M$ ) has been sent out by the reader, the *Ready* tags just respond to the request commands in which “ $B$ ” is the currently second high collision bit (next to the highest collision bit currently), and the tags decide whether to send a response command or not according to the parameter “ $P$ .” The *Ready* tag refers to the tag whose state is not *Standby*, *Silence*, and *Killed*.

### 3.4. Estimation of Tag Number

(1) *Simple Estimation.* The reader broadcasts synchronization signal to all tags, and after receiving the signal each tag sends its ID to the reader. If collision occurs, the reader detects the collision bits. Let the number of the collision bits be  $X$ . The number of tags is  $2^X$  at most, and if  $2^X$  is greater than a threshold  $\alpha$  preset for module\_1, the accurate estimation will be started by the reader [31]. That is to say, it is decided by  $X$  and  $\alpha$  that the accurate estimation needs to be started.

(2) *Accurate Estimation.* In RFID system,  $L$  and  $n$  are assumed to be, respectively, the frame size and the number of tags. In practice, it is reasonable to assume that  $n$  is not known and has to be estimated based on the observed read results.

For an observed read result  $C = \langle c_0, c_1, c_k \rangle$  (where  $c_0$ ,  $c_1$ , and  $c_k$  are the number of the empty time slot, successful time slot, and collision time slot), there have been many methods to estimate tag number  $n$  in RFID system. The most well-known methods are lower bound algorithm, Schoute algorithm, collision ratio algorithm, and Vogt algorithm [40].

Lower bound algorithm was proposed by Harald Vogt, which is obtained through the observation that a collision involves at least two different tags. Therefore, a lower bound on the value of  $n$  can be obtained by the simple estimation function  $nLB(L, c_0, c_1, c_k) = c_1 + 2 * c_k$ . This method had validated the larger estimation error especially in much more tags systems [32].

Schoute tag estimation mechanism is based on a prerequisite hypothesis that the system has the maximum throughput rate of tags in a frame. The number of tags is estimated by multiplying the number of collision slots ( $c_k$ ) in a frame by the a posteriori expectation on the number of tags that choose the same time slot simultaneously ( $nSchoute = 2.39 * c_k$ ), which is the same for all frames regardless of the various number of tags and frame size. In a practical RFID system, the maximum throughput rate of tags is not always got in a frame, so the Schoute mechanism also has larger error of tag estimation [41].

Collision ratio mechanism assumes that the slot collision ratio is equal to the expectation slot collision probability

$$C_{ratio} = \frac{c_k}{c_k + c_0 + c_1}, \quad (1)$$

$$P_c = 1 - \left(1 - \frac{1}{L}\right)^n - \left(\frac{n}{L}\right) \left(1 - \frac{1}{L}\right)^{n-1},$$

where  $P_c$  is the expectation of the slot collision probability. By letting  $C_{ratio}$  be equal to  $P_c$ , the number of tags  $n$  can be obtained. This mechanism ignores the difference between the real collision ratio and the expectation of the slot collision probability and also has larger error of tags estimation [42].

Vogt method is based on the fact that the outcome of a random experiment is most likely somewhere near the expected value. Thus, an estimation function uses the distance between the read result and the expected value vector to determine the value of  $n$  for which the distance becomes minimal [43]. In this paper, an improved Vogt method with *Chebyshev* inequality is used. We suppose the matter in which one tag appears in a time slot subject to a binomial distribution. The probability that  $r$  tags appear in a time slot can be computed as in the following equation:

$$p(r) = \binom{n}{r} \left(\frac{1}{L}\right)^r \left(1 - \frac{1}{L}\right)^{n-r}. \quad (2)$$

Let  $p(0)$  be the probability that the time slot is an empty time slot, that is, no tag to choose this time slot, let  $p(1)$  be the probability that the time slot is a successful time slot, that is, one of  $n$  tags to choose this time slot, and let  $p(k)$  be the

probability that the time slot is a collision time slot, that is,  $k$  tags in  $n$  tags to choose this time slot. So,

$$\begin{aligned} p(0) &= \left(1 - \frac{1}{L}\right)^n, \\ p(1) &= \left(\frac{n}{L}\right) \left(1 - \frac{1}{L}\right)^{n-1}, \\ p(k) &= 1 - p(0) - p(1) \\ &= 1 - \left(1 - \frac{1}{L}\right)^n - \left(\frac{n}{L}\right) \left(1 - \frac{1}{L}\right)^{n-1}. \end{aligned} \quad (3)$$

In a frame, the average values of empty time slots, successful time slots, and collision time slots can be represented as

$$\begin{aligned} s_0(L, n) &= L \left(1 - \frac{1}{L}\right)^n, \\ s_1(L, n) &= \left(\frac{n}{L}\right) \left(1 - \frac{1}{L}\right)^{n-1}, \\ s_k(L, n) &= L \left(1 - \left(1 - \frac{1}{L}\right)^n - \left(\frac{n}{L}\right) \left(1 - \frac{1}{L}\right)^{n-1}\right). \end{aligned} \quad (4)$$

Actually,  $s_0(L, n)$ ,  $s_1(L, n)$ , and  $s_k(L, n)$  are, respectively, the expectation values of empty time slots, successful time slots, and collision time slots, and the expectations are relevant to the length of frame and the number of tags. We denote the estimation function by

$$\xi(L, C) = \min_{\hat{n} \in \Phi} |S(n) - C|, \quad (5)$$

where  $S(n) = [s_0(L, n), s_1(L, n), s_k(L, n)]^T$ ,  $C = [c_0, c_1, c_k]^T$ ,  $\Phi = \{n \mid c_1 + 2c_k \leq n \leq N\}$ , and  $N$  is the maximum number of tags in the system. We can get the estimation number of tags, that is,  $\hat{n}$ , by using the EM (*expectation maximization*) algorithm [44] as follows:

$$\hat{n} = \arg \xi(L, C) = \arg \min_{n \in \Phi} |S(n) - C|. \quad (6)$$

Figure 2 shows the mean estimation error of the four methods, which is defined as the mean difference between the real number and the estimated number of tags. The Vogt with *Chebyshev* method presents the most efficient estimation performance among the methods. So, in this paper, we use this method to estimate the tags in the accurate estimation.

**3.5. Identification Algorithm.** The process of the flood division anticollision (FDAC) algorithm has been presented in Figure 2. The detailed algorithm is described as shown in Algorithm 1.

After being divided into groups, the RFID tags are assigned to the models. The operation that tags are identified in model<sub>i</sub> is described as shown in Algorithm 2.

An example that tags are processed with Operation run( $i$ ) is as follows. We assume that 8 activated RFID tags are grouped and assigned to model<sub>i</sub>, and the IDs of the tags are tag<sub>i.1</sub>: 10110011, tag<sub>i.2</sub>: 11011011, tag<sub>i.3</sub>: 10010001, tag<sub>i.4</sub>:

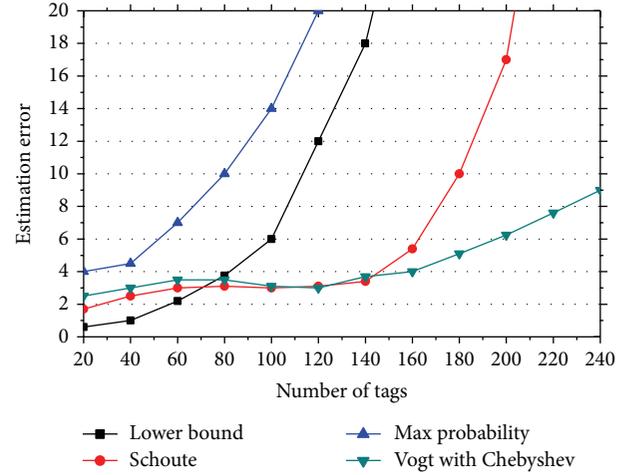


FIGURE 2: Difference between the real number and the estimated number of tags.

1111011, tag<sub>i.5</sub>: 10011011, tag<sub>i.6</sub>: 11110001, tag<sub>i.7</sub>: 11010011, and tag<sub>i.8</sub>: 10011001. The reader can detect a signal “1??1?0?1” in model<sub>i</sub>, which expresses that collisions occur in four bits: Di1, Di3, Di5, and Di6. The process that the tags are identified with Module<sub>i</sub> is as in Figure 3. Finally, all eight tags have been identified. Writing operation, if tags need, can be started after the tags turn into Silence state.

## 4. Simulation and Analysis

The simulation condition is as follows. There is only one reader, and in the field of the reader, the number of tags increases from 2 to 1000. Both tag-to-reader data rate and reader-to-tag data rate are chosen as 40 kbps. The reason is that the middle speed in EPC Class 1 Gen.2 proposed by EPC global to ISO/IEC 18000-6 B is equal to the chosen data rate [43]. There is some iteration overhead because of propagation delay from the channel and latency from the signal processing. Here, the iteration overhead is not considered for the simulation. The IDs are randomly generated.

We simulate FDAC and compare it with the dynamic frame slotted (DFS) Aloha algorithm, the binary-retreat tree (BRT) algorithm, and the dynamic binary-search tree (DBST) algorithm in terms of the number of data bits transmitted by reader, the number of data bits transmitted by tags, and the identification time delay [44, 45]. In the simple estimation of tag numbers, we set the threshold  $\alpha = 17$ , which is an empirical value that lets FDAC get better comprehensive performance, under the simulation conditions in which FDAC has 4 models and the maximum number of tags is 1000. The results are the average values of 50 times experiments.

**4.1. Number of Data Bits Transmitted by Reader.** DFS needs many communication rounds to optimize the frame size in the identification process and needs the reader to send much more data bits to tags. In DBST, the tag ID is divided into known part and unknown part, and in the search process only the known part needs to be transmitted between tags

```

(1) Input: Set of Tags,  $\alpha$ ;
    //  $\alpha$  is the parameter to start accurate estimation;
(2) Output: List of identified tag Ids;
(3) BEGIN
(4)   Sort Tag Ids; // In ascending order;
(5)   Byte_Collision_bits := Detect(Collision bits);
    // Detect get collision bits from collision signal;
(6)   Bit_Num := Count(Byte_Collision_bits);
(7)   If  $2^{\text{Bit\_Num}} \leq \alpha$  Then // Simple estimation
(8)   { Assign all tags to Model 1;
(9)     Start (Model_1 (run()));
    // Model_1 operates the run() to identified tags;
(10)  Output (List of identified tag Ids); }
(11) Else
(12) { Tag_Num := Accurate_Estimation(Tags);
    // Start accurate estimation;
(13)  Do Case
(14)    Case Tag_Num in [1–16]
(15)    { Assign all tags to Model_1;
(16)      Start (Model_1(run(NULL))); };
(17)    Case Tag_Num in [17–64]
(18)    { Make tags into 2 groups;
    // Set groups in IDs order;
(19)      Assign Group 1 to Model_1;
(20)      Assign Group 2 to Model_2;
(21)      Do in parallel { Start (Model_1(run(1));
(22)                        Start (Model_2(run(2))); }; }
(23)    Case Tag_Num in [65–521]
(24)    { Make tags into 3 groups;
    // Set groups in IDs order;
(25)      Assign Group 1 to Model_1;
(26)      Assign Group 2 to Model_2;
(27)      Assign Group 3 to Model_3;
(28)      Do in parallel { Start (Model_1(run(1));
(29)                        Start (Model_2(run(2)));
(30)                        Start (Model_3(run(3))); }; }
(31)    Case Tag_Num in [523–2048]
(32)    { Make tags into 4 groups;
    // Set groups in IDs order;
(33)      Assign Group 1 to Model_1;
(34)      Assign Group 2 to Model_2;
(35)      Assign Group 3 to Model_3;
(36)      Assign Group 4 to Model_4;
(37)      Do in parallel { Start (Model_1(run(1));
(38)                        Start (Model_2(run(2)));
(39)                        Start (Model_3(run(3)));
(40)                        Start (Model_4(run(4))); }; }
(41)    ...
(42)  } End Case
(43) End If
(44) Output (List of identified tag Ids);
(45) END FDAC

```

ALGORITHM 1: FDAC.

and reader, which reduce data transmission between the tags and the reader. In BRT, some data structure technologies are used to store and remember the previously used commands which will only be imported while it would be in need again. In FDAC, the back tracking strategy is used to process

the REQUEST command which reduces the times of the REQUEST command sent by the reader. And while needing to send a REQUEST command, the reader only needs to send a three-dimensional-vector command, and the tags respond to the command with part of the collision-bit parameters.

```

(1) Begin
(2)   Byte_Collision_bits := Get_Byte_from_Collision_Tag_IDs;
(3)   High_bit.Index := Get_Highest_bit_Index(Byte_Collision_bits);
      //Get the highest collision bit Index;
(4)   High_bit.value := Get_Highest_bit_Value(Byte_Collision_bits);
      //Get the value of the highest collision bit;
(5)   While High_bit <> NULL
(6)   {
(7)     REQUEST(High_bit.Index, High_bit.value, i);
      //Reader send REQUEST to the tags;
(8)     ANSWER (f); //Tags respond with f;
(9)     If f <> NULL
(10)    { Store REQUEST(High_bit.Index, High_bit.value, i) into the stack;
(11)      Collision Tags turn into Standby;
(12)      Store collision information;
(13)      Assign next high collision bit to the High_bit; }
(14)    Else
(15)      { Tags be identified and turn into the Silence;
(16)        If need write, wait for write operation; }
(17)    End if
(18)    Call REQUEST(High_bit.Index, High_bit.value, i);
      //Back tracking strategy;
(19)    High_bit.value := ~(High_bit.value)
(20)  } End While
(21) } End run

```

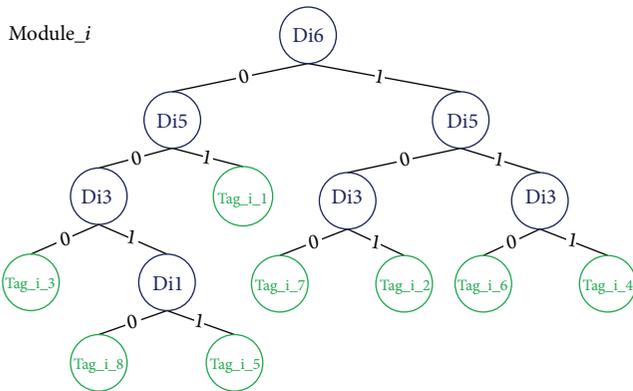
ALGORITHM 2: Operation run( $i$ ).FIGURE 3: Process of 8 RFID tags identified in Module. $i$ .

Figure 4 shows the amount of data bits transmitted by the reader in varied number of the tags. Although the numbers of data bits transmitted by the reader of the four algorithms nearly increase linearly with varied number of tags, the FDAC has the best data transmitting performance among the four algorithms. The data transmitted by the reader in FDAC is only 26%, 42%, and 63% of that in DFS, DBST, and BRT, respectively, for 1000 tags.

**4.2. Number of Data Bits Transmitted by Tag.** As described previously, in DBST, the tag ID is divided into known part and unknown part, and in the search process only the known part needs to be transmitted between tags and reader, which

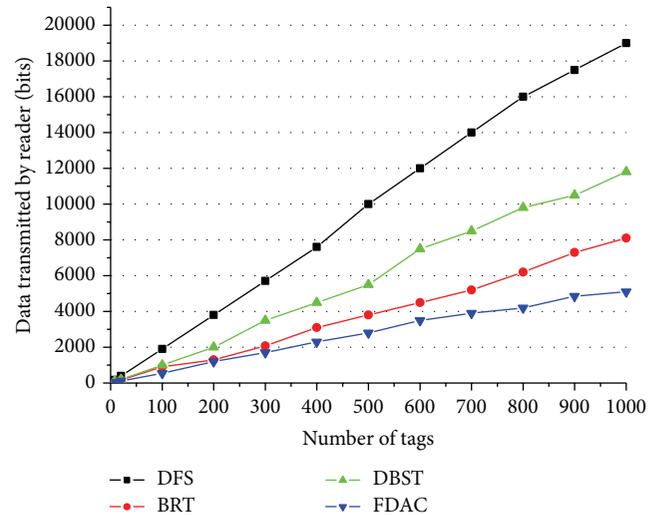


FIGURE 4: Plot of the data transmitted by reader of the four algorithms.

also reduce data transmission from the tags to the reader. In BRT, the data structure technologies are used to store and remember the previously used commands, which reduce not only the data transmission from the reader to the tags but also the data transmission from the tags to the reader. In FDAC, while responding to the request command, the collision tag does not every time need to send the completed ID to the reader (only in the first time of the REQUEST).

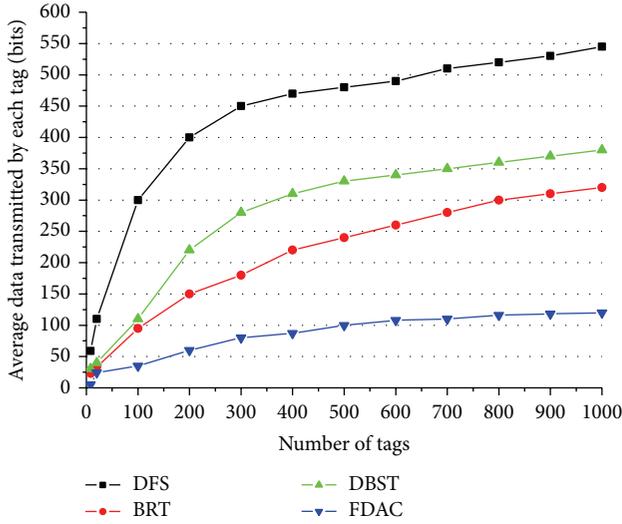


FIGURE 5: Plot of the data transmitted by tags of the four algorithms.

Figure 5 shows the average data bits transmitted by each tag. FDAC has also the best data transmitting performance by tags among the four algorithms. The average data transmitted by each tag in FDAC is only 22%, 32%, and 38% of that in DFS, DBST, and BRT, respectively, for 1000 tags.

**4.3. Tag Identification Time Delay.** We compare FDAC with DFS, DBST, and BRT in terms of the tag identification time delay, which is the elapsed time for a reader to identify all tags in the interrogation zone. The tag identification time delay  $T_d$  is defined to be [37, 38]

$$T_d = (n_c + n_{nc}) \times t_{id} + n_{cmd} \times t_m + n_{np} \times t_a. \quad (7)$$

The notations used in (7) are shown as follows:

- $n_c$ : the number of cases with tag collisions;
- $n_{nc}$ : the number of cases with no tag collisions;
- $n_{cmd}$ : the number of commands sent by the reader;
- $n_{np}$ : the number of cases of no tag response;
- $t_{id}$ : the elapsed time for a tag to transmit its ID;
- $t_m$ : the elapsed time for the reader to transmit a command;
- $t_a$ : the elapsed time for a reader to be aware of null response.

In FDAC, the mechanisms that reduce data transmission between the tags and the reader help to save the identification time. More importantly, the parallel processing technology among the modules with the flood division anticollision algorithm mainly contributes to better identification time performance, especially in the case of much more tags.

Figure 6 presents the comparison of the identification time delay between the four algorithms, and FDAC has the best performance of identification time. The identification time delay in FDAC is only 17%, 23%, and 29% of that in DFS, DBST, and BRT, respectively, for 1000 tags.

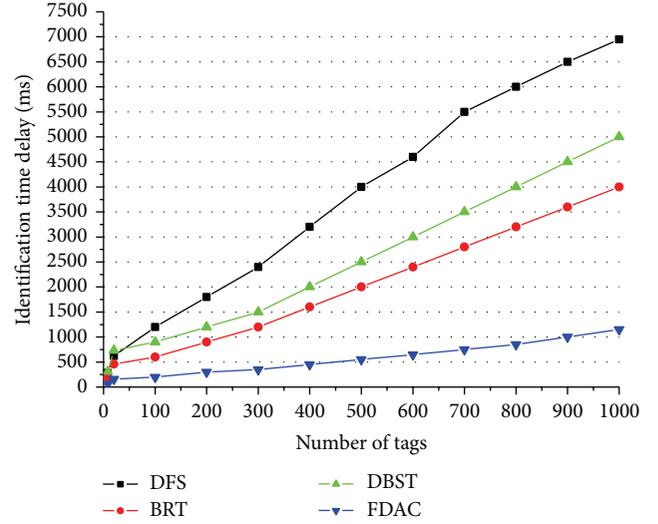


FIGURE 6: Plot of identification time delay of the four algorithms.

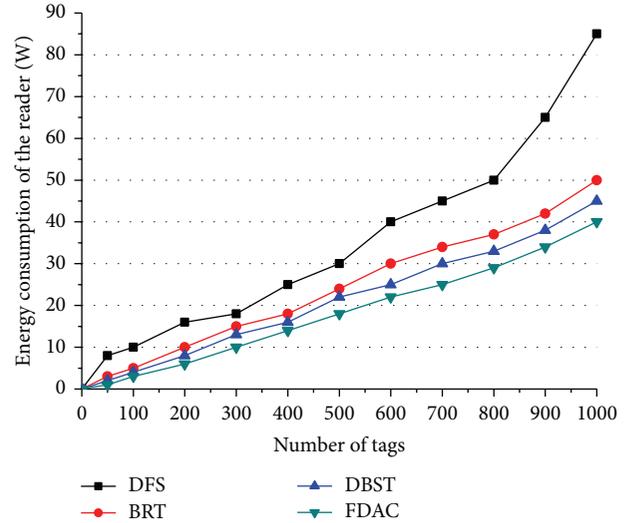


FIGURE 7: Plot of energy consumption of the reader with the four algorithms.

**4.4. Energy Consumption of Reader and Tag.** We compare FDAC with DFS, DBST, and BRT in terms of the energy consumption of the reader and tags, according to the energy consumption evaluation framework presented by Yan and Liu in [35]. In the evaluation framework, the energy consumed by the reader in a tag collision resolution cycle had been classified as the energy consumed in broadcasting command messages and that in listening to the tag responses, and the energy consumed by the tag had been classified as that consumed in transmitting its modulated signals and in listening to the command messages broadcasted by the reader.

Figure 7 presents the energy consumed by the reader with the four algorithms. The FDAC performs the best, and the DFS consumes much more energy than the other three algorithms. We think it is due to the fact that DFS needs

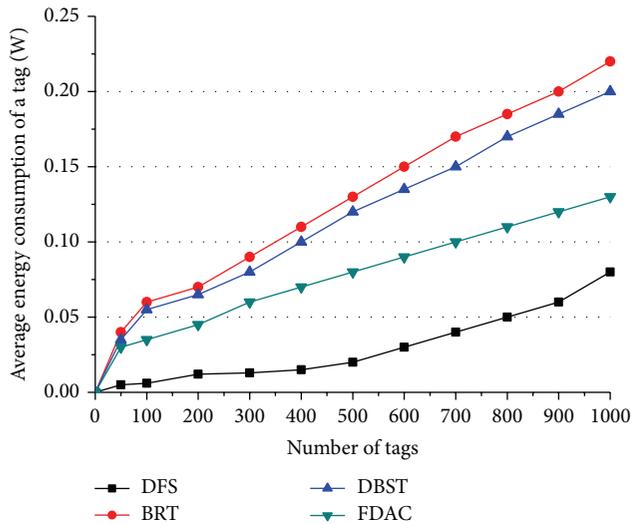


FIGURE 8: Plot of average energy consumption of a tag with the four algorithms.

much more communication rounds to optimize the frame size in the identification process and needs the reader to send much more data and that the FDAC reduces the REQUEST command sent by the reader.

Figure 8 presents the average energy consumed by a tag with the four algorithms. The FDAC needs less energy than BRT and DBST but needs more energy than DFS. The reason why the FDAC needs more energy than the DFS is that the FDAC needs more listening to the command messages broadcasted by the reader than the DFS.

## 5. Conclusions and Future Research

In this paper, we investigated the tag identification problem in RFID system. Firstly, we surveyed the existing algorithms for anticollision of tags. Then, we proposed a new anticollision algorithm based on flood division idea to reduce data bits transmitted and identification time delay. And, we simulate the FDAC and compare it with the DFS, the BRT, and the DBST in terms of the number of data bits transmitted by reader, the number of data bits transmitted by tags, the identification time delay, and the energy consumption by the reader and tag.

The results of the simulation experiment show that FDAC has better performance than DFS, BRT, and DBST, whatever in data transmitted and identification time delay or in energy consumption by the reader. About the energy consumption by a tag, the FDAC shows better performance than DFS but less than the BRT and the DBST.

In the future works, we will improve the FDAC especially in the energy consumption.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant 6074308, the Key Scientific and Technological Projects of Henan Province in China under Grant 142102210045, and the project sponsored by SRF for ROCS, SEM. The authors would like to thank the reviewers for their valuable comments.

## References

- [1] P. H. Chen, Y. H. Liang, and T. C. Lin, "Implementing a cooking and dietary management system using RFID technology," *Mathematical Problems in Engineering*, vol. 2014, Article ID 234614, 10 pages, 2014.
- [2] L.-P. Hung, H.-K. Lu, C.-S. Wang, and D.-J. Chiang, "Building a smooth medical service for operating room using RFID technologies," *Journal of Applied Mathematics*, vol. 2014, 11 pages, 2014.
- [3] L. Catarinucci, R. Colella, L. Mainetti et al., "Near field UHF RFID antenna system enabling the tracking of small laboratory animals," *International Journal of Antennas and Propagation*, vol. 2013, Article ID 713943, 10 pages, 2013.
- [4] L. Catarinucci, R. Colella, L. Mainetti et al., "Smart RFID antenna system for indoor tracking and behavior analysis of small animals in colony cages," *IEEE Sensors Journal*, vol. 14, no. 4, pp. 1198–1206, 2014.
- [5] H. Zeng, J. Zhang, G. Dai, Z. Gao, and H. Hu, "Security visiting: RFID-based smartphone indoor guiding system," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 212741, 13 pages, 2014.
- [6] D. Guo, Y. Zhang, Q. Xiang, and Z. Li, "Improved radio frequency identification indoor localization method via radial basis function neural network," *Mathematical Problems in Engineering*, vol. 2014, Article ID 420482, 9 pages, 2014.
- [7] D. Xie, J. Xiao, G. Guo, and T. Jiang, "Processing uncertain RFID data in traceability supply chains," *The Scientific World Journal*, vol. 2014, Article ID 535690, 22 pages, 2014.
- [8] E. Ilie-Zudor, Z. Kemény, F. van Blommestein, L. Monostori, and A. van der Meulen, "A survey of applications and requirements of unique identification systems and RFID techniques," *Computers in Industry*, vol. 62, no. 3, pp. 227–252, 2011.
- [9] X.-Q. Yan, Y. Liu, B. Li, and X.-M. Liu, "Numeric evaluation on the system efficiency of the EPC Gen-2 UHF RFID tag collision resolution protocol in error prone air interface," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 716232, 9 pages, 2014.
- [10] X. Zhu, S. K. Mukhopadhyay, and H. Kurata, "A review of RFID technology and its managerial applications in different industries," *Journal of Engineering and Technology Management*, vol. 29, no. 1, pp. 152–167, 2012.
- [11] L. Catarinucci, R. Colella, G. D. Fiore et al., "A cross-layer approach to minimize the energy consumption in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 268284, 11 pages, 2014.
- [12] D. Klair, K.-W. Chin, and R. Raad, "On the energy consumption of Pure and Slotted Aloha based RFID anti-collision protocols," *Computer Communications*, vol. 32, no. 5, pp. 961–973, 2009.
- [13] S. L. Mei and D. H. Zhu, "HPM-based dynamic sparse grid approach for Perona-Malik equation," *The Scientific World Journal*, vol. 2014, Article ID 417486, 15 pages, 2014.

- [14] V. Sarangan, M. R. Devarapalli, and S. Radhakrishnan, "A framework for fast RFID tag reading in static and mobile environments," *Computer Networks*, vol. 52, no. 5, pp. 1058–1073, 2008.
- [15] M. He, S.-J. Horng, P. Fan et al., "A fast RFID tag identification algorithm based on counter and stack," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6829–6838, 2011.
- [16] Y. I. Joo, D. H. Seo, and J. W. Kim, "An efficient anti-collision protocol for fast identification of RFID tags," *Wireless Personal Communications*, vol. 77, pp. 767–775, 2013.
- [17] Y.-Q. Wang, G.-P. Jiang, and J. Wang, "Framed slotted ALOHA with grouping tactic and binary selection for anti-collision in RFID systems," *The Journal of China Universities of Posts and Telecommunications*, vol. 16, no. 4, pp. 47–52, 2009.
- [18] M.-K. Yeh and J.-R. Jiang, "A novel query tree protocol based on partial responses for RFID tag anti-collision," in *Proceedings of the 19th IEEE International Conference on Parallel and Distributed Systems (ICPADS '13)*, pp. 617–622, December 2013.
- [19] D. K. Klair, K.-W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 3, pp. 400–421, 2010.
- [20] V. Namboodiri, M. Desilva, K. Deegala, and S. Ramamoorthy, "An extensive study of slotted Aloha-based RFID anti-collision protocols," *Computer Communications*, vol. 35, no. 16, pp. 1955–1966, 2012.
- [21] P. Pupunwiwat, P. Darcy, and B. Stantic, "Conceptual selective RFID anti-collision technique management," *Procedia Computer Science*, vol. 5, pp. 827–834, 2011.
- [22] N. Kamineni and X. Li, "Analysis of anti-collision multi-tag identification algorithms in passive RFID systems," in *Proceedings of the International Conference on Computing Communication and Networking Technologies (ICCCNT '10)*, pp. 1–8, Karur, India, July 2010.
- [23] Z.-E. Wael and E. Shaaban, "An enhanced binary tree anti-collision technique for dynamically added tags in RFID systems," in *Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICCET '10)*, pp. V7349–V7353, April 2010.
- [24] C. Qian, H. Ngan, Y. Liu, and L. M. Ni, "Cardinality estimation for large-scale RFID systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1441–1454, 2011.
- [25] Z. Li, C. He, J. Li, and X. Huang, "RFID reader anti-collision algorithm using adaptive hierarchical artificial immune system," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2126–2133, 2014.
- [26] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.
- [27] H. Hao, S. Bo, C. C. Tan, L. Qun, M. Weizhen, and L. Sanglu, "Counting RFID tags efficiently and anonymously," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, San Diego, Calif, USA, March 2010.
- [28] Y.-C. Lai, L.-Y. Hsiao, and B.-S. Lin, "An RFID anti-collision algorithm with dynamic condensation and ordering binary tree," *Computer Communications*, vol. 36, no. 17–18, pp. 1754–1767, 2013.
- [29] L. Xie, Y. Yin, A. V. Vasilakos, and S. Lu, "Managing RFID data: challenges, opportunities and solutions," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1294–1311, 2014.
- [30] Y. C. Lai, L. Y. Hsiao, and B. S. Lin, "Optimal slot assignment for binary tracking tree protocol in RFID tag identification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 255–268, 2015.
- [31] H. Landaluce, A. Perallos, and I. J. G. Zuazola, "A fast RFID identification protocol with low tag complexity," *IEEE Communications Letters*, vol. 17, no. 9, pp. 1704–1706, 2013.
- [32] W. Y. Huo Hua, "RFID anti-collision algorithm based on parallel processing," *Computer Engineering*, vol. 37, pp. 263–265, 2013.
- [33] W.-T. Chen, "An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 9–15, 2009.
- [34] S. Xu and X. Yang, "Analysis of energy consumption for multiple object identification system with active RFID tags," *Wireless Communications and Mobile Computing*, vol. 8, no. 7, pp. 953–962, 2008.
- [35] X. Yan and X. Liu, "Evaluating the energy consumption of the RFID tag collision resolution protocols," *Telecommunication Systems*, vol. 52, no. 4, pp. 2561–2568, 2013.
- [36] T. Agrawal and P. K. Biswas, "Energy consumption evaluation framework for passive RFID tag anti-collision algorithms," *International Journal of Radio Frequency Identification Technology and Applications*, vol. 4, no. 3, pp. 229–243, 2014.
- [37] K.-W. Chin and D. Klair, "E<sup>2</sup>MAC: an energy efficient MAC for RFID-enhanced wireless sensor networks," *Pervasive and Mobile Computing*, vol. 7, no. 2, pp. 241–255, 2011.
- [38] L. Catarinucci, S. Guglielmi, L. Patrono, and L. Tarricone, "Switched-beam antenna for wireless sensor network nodes," *Progress In Electromagnetics Research C*, vol. 39, pp. 193–207, 2013.
- [39] L. Catarinucci, S. Guglielmi, L. Mainetti et al., "An energy-efficient MAC scheduler based on a switched-beam antenna for wireless sensor networks," *Journal of Communications Software and Systems*, vol. 9, no. 2, pp. 117–127, 2013.
- [40] T. Li, S. S. Wu, S. Chen, and M. C. K. Yang, "Generalized energy-efficient algorithms for the RFID estimation problem," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 1978–1990, 2012.
- [41] H. Vogt, "Efficient object identification with passive RFID tags," in *Pervasive Computing*, vol. 2414 of *Lecture Notes in Computer Science*, pp. 98–113, Springer, Berlin, Germany, 2002.
- [42] X. Fan, I. C. Song, and K. H. Chang, "Gen2-based hybrid tag anti-collision Q algorithm using Chebyshev's inequality for passive RFID systems," in *Proceedings of the IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '08)*, pp. 1–5, Cannes, France, September 2008.
- [43] E. Inc, "EPC radio-frequency identity protocols class 1 generation-2 UHF RFID protocol for communications at 860–960 MHz version 1.2.0," Specification for RFID Air Interface 1-108, 2008.
- [44] F. Marino, G. Massei, and L. Paura, "Modeling and performance simulation of EPC Gen2 RFID on OPNET," in *Proceedings of the 2nd IEEE International Workshop on Measurements and Networking (M & N '13)*, pp. 83–88, October 2013.
- [45] A. Zavvari, M. Shakiba, M. T. Islam, E. Sundararajan, and M. J. Singh, "Computational cost analysis on securing RFID protocols conforming to EPC class-1 generation-2 standard," *Procedia Technology*, vol. 11, pp. 778–784, 2013.

