Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 609035, 20 pages
doi:10.1155/2012/609035

Research Article

Merged Search Algorithms for Radio Frequency Identification Anticollision

Bih-Yaw Shih,[1] Cheng-Wu Chen,[2,3] Chen-Yuan Chen,[1] and Ta-Wei Lo[1]

[1] *Department and Graduate School of Computer Science, National Pingtung University of Education, No. 4-18, Ming Shen Road, Pingtung 90003, Taiwan*
[2] *Institute of Maritime Information and Technology, National Kaohsiung Marine University, Kaohsiung City 80543, Taiwan*
[3] *Global Earth Observation and Data Analysis Center (GEODAC), National Cheng Kung University, No 1, Ta-Hsueh Road, Tainan 701, Taiwan*

Correspondence should be addressed to Chen-Yuan Chen, cyc@mail.npue.edu.tw

Received 16 July 2011; Accepted 11 December 2011

Academic Editor: Andrzej Swierniak

Nowadays, the Radio Frequency Identification (RFID) system enables the control of many devices over an open communication infrastructure ranging from a small home area network to the global Internet. Moreover, a variety of consumer products are tagged with remotely low-cost readable identification electromagnetic tags to replace Bar Codes. Applications such as automatic object tracking, inventory and supply chain management, and Web appliances were adopted for years in many companies. The arbitration algorithm for RFID system is used to arbitrate all the tags to avoid the collision problem with the existence of multiple tags in the interrogation field of a transponder. A splitting algorithm which is called Binary Search Tree (BST) is well known for multitags arbitration. In the current study, a splitting-based schema called Merged Search Tree is proposed to capture identification codes correctly for anticollision. Performance of the proposed algorithm is compared with the original BST according to time and power consumed during the arbitration process. The results show that the proposed model can reduce searching time and power consumed to achieve a better performance arbitration.

1. Introduction

Nowadays, Radio Frequency Identification (RFID) systems are widely used in different applications allowing users to identify an item so that it can quickly and accurately be reidentified, electronically when it moves through the utilization process. A RFID system basically is made of a reader and tags where each tag has a unique ID where the reader can identify and communicate with the tag via radio frequency.

There are two major types of tags: passive and active. A passive tag must listen to the signal emited from the reader then response to it, but it has lower manufacturing cost. An active tag has a power supply; it can emit RF signal to communicate with the reader actively, usually with longer access distance and better access time. Both of them are in widespread use in commercial applications.

When there has more than one tag in the interrogation field of a reader, the application of RFID identification device requires communicating with multiple tags simultaneously. The reader would send signals to each tag like calling “who is there?” and all the tags may answer simultaneously with their identification codes. Eventually, collision will happen and cause data loss, because the reader could not receive all the reflex signals simultaneously. The process to distinguish between all these answers and to capture their identification codes correctly is called anticollision. The environment with multitags is very common in commercial applications. There exist different anticollision methods to identify a device within a group including TDMA, SDMA, FDMA, CDMA, and Binary Search Tree algorithm.

The purpose of this study is the development of a splitting-based schema, called Merged Search Tree to improve the schema of Binary Search Tree defined in the anticollision protocol of Electronic Product Code (EPC) global Ultrahigh Frequency (UHF) RFID system. The proposed Merged Search Tree is expected to reduce the searching time and power consumed to accomplish the arbitration process.

2. Literature Review

An overview of RFID technique is introduced first. Then, the collision problem in RFID system and current solutions for anticollision are discussed.

2.1. Radio Frequency Identification (RFID)

Radio Frequency Identification (RFID) systems utilizing the radio frequency technology for identifying objects are widely used in commercial applications such as warehouse management, automatic object tracking, or medical management. RFID systems have higher capacity of variable data and can transmit data via RF signals without requiring a line of sight for longer communication distance. RFID systems are more reliable with higher data security and multi-tag readable; these characteristics promote the large-scale applications and immeasurable development.

There are five basic components to make up a RFID system: one or more RF tags, two or more antennas, one or more interrogators (readers), host computers, and appropriate software. Tags can be read remotely when they detect a radio frequency signal from a reader over a range of distances. A RFID tag contains its unique ID to communicate with the reader [1]. According to power supply, there are primarily two categories of RFID tags: passive and active Tags. The passive tags obtain operating power from the reader and have shorter read ranges. The active tags powered by an internal battery with longer read range. In brief, passive tags have lower cost and active tags are more powerful on function. A RFID system adopts the modulated backscatter to make reader communicate one or more tags as shown in Figure 1.

RFID systems were classified according to operating frequency. There are dissimilar characteristics and capability of each RF bandwidth. In substance, four classifications were indicated: Low Frequency (LF), High Frequency (HF), Ultrahigh Frequency (UHF) and Microwave. The Low Frequency system always works by passive tags with short range

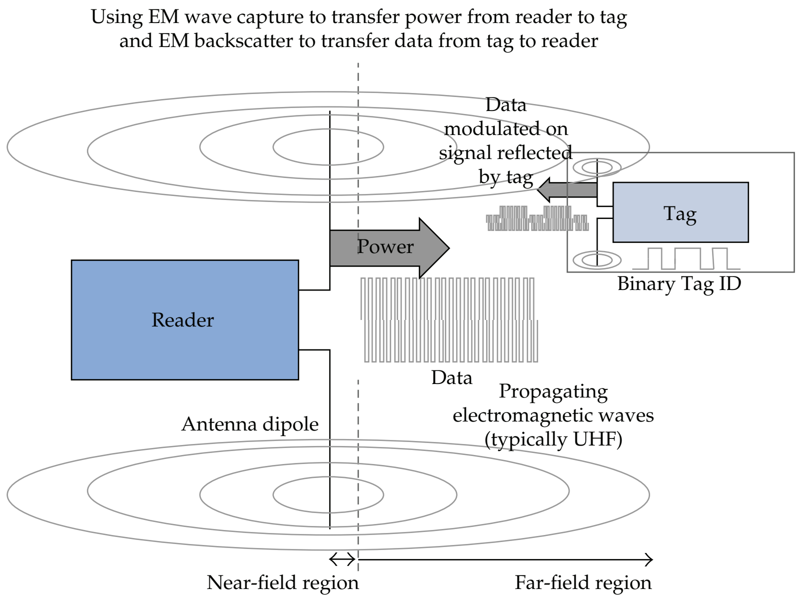

Figure 1: Power/communication mechanism for RFID tags [1].

and low data transmission rate. This system is widely used today such as animal ID or autokey. The High Frequency system is accepted worldwide and works in most environments but does not pierce through metal materials. Its application includes item level track and management of entrance guard. The Ultrahigh Frequency system has longer read range and better transmission rate; this makes it expand in commercial use. The most excellence is that UHF tag can be labeled on metal without interference. Even so, the interference between tags in close physical proximity is an issue. The Microwave system has long communicating distance and more functional but complex systems development [2, 3].

2.2. RFID Anticollision

When more than one tag appears in the interrogation field of a reader, the reader would send signals to each tag and all the tags would reflex signals with their Tag IDs back to the reader at the same time. This is a problem called collision that tags collide with each other to cause data loss, because the reader could not receive all the reflex signals simultaneously. Some tags would not be identified and retransmissions are required which results in the increase of total delay. However, the environment of multitags is very general in commercial applications and many researchers are dedicating in resolving this problem. Space Division Multiple Access (SDMA) [4], Frequency Division Multiple Access (FDMA) [5], Code Division Multiple Access (CDMA) [6], and Time Division Multiple Access (TDMA) [7] are the well known arbitration algorithms which are developed to overcome the collision between tags.

Recently, soft computing and artificial intelligence have been successfully applied to the different fields including value calculation [8, 9], and dynamic system [10], and channel selection [11]. TDMA is the most popular algorithm for RFID anticollision where it allows tags to use the same frequency by allocating a random time slot to each tag. Tags will response to the reader according to the unique time slots. The protocol, Interrogator Talks First (ITF) is the standard solution declared for anticollision in Electronic Product Code

(EPC) standard where a two-way communication link will be established between reader and tags [7]. The tag ID of each tag will be identified, and reader will send out commands to each tag according to the sequence defined by the algorithm. The role of which device takes initiative in the communication link are categorized into two types of ITF protocol: tag talks first (TTF) and reader talks first (RTF). TTF is relatively slow and inflexible to RTF, the synchronous procedure which is also known as time duplex procedure that reader initiates the communication actively. RTF is subdivided into Polling [4], Splitting Method [12], I-code Protocol [13], and Contact-less Protocol [14]. Meanwhile, several algorithms were proposed to improve the original algorithms [15]. A quadtree search algorithm is developed to avoid collision and it outperforms the traditional DFSA algorithm with respect to reducing the cost and tag reading time [16].

In addition to the EPC Global Inc., the International Standards Organization (ISO) also formulated RFID standards as part of the ISO 18000 family. There are different classifications and attributes between these two standards. For the purpose of anticollision, EPC Global Inc. proposed bit-based Binary Search Tree algorithm (deterministic) for EPC global CLASS 0 (UHF) and Bin slot Binary tree (Probabilistic) for EPC global CLASS 1 (UHF). ISO proposed Dynamic Framed ALOHA (Probabilistic) algorithm for ISO 18000-6 TYPE-A (UHF) and Binary Search Tree algorithm for ISO 18000-6 TYPE-B (UHF) [17].

In both of the specifications, Binary search tree algorithm is defined as the standard ITF anticollision protocol for Radio Frequency Identification Tags. There are many researches devoted to improve this algorithm, including improving the procedure for query conflicting tags or modifying the frame to place tags. Wang [18] proposed an enhanced binary search tree with cut-through operation [18]. The cut-through operation will replace the intermediate node with one leave node, which is the node with no collision occurred, by the leave node itself to shorten the stages of binary search tree algorithm. The analytic results show that this simplifying of tree structure outperforms when the number of existing tags is small relative to all possible nodes.

Nanjundaiah and Chaudhary [19] proposed a modification of Binary Search Tree algorithm by resetting to the appropriate node for read cycle [19]. Instead of going to the root node in every read cycle in the original protocol, the modified protocol begins traversal cycle with the bit position form where the last difference happened in the bit stream of tag IDs. Although the complexity of tag is increasing, the overall read time is reducing.

The principal excellence of Binary Search Tree algorithm is simple to be implemented in the digital integrated circuits within the tag and low power consumption during the arbitration process [18, 20]. These features make this algorithm applicable to industrial manufacture of tags with a longer working distance. Because of that, Binary Search Tree algorithm is the official anticollision protocol; presently, the objective of this study is to modify Binary Search Tree to reduce arbitration time and power consumed.

2.3. General Binary Search Tree Algorithm

According to EPC global specification, Ultrahigh frequency (UHF) adopt binary tree scheme as anticollision algorithm. The advantages of binary tree schema are easy tag implementation and efficiency [2, 21]. Binary Search Tree for RFID system is developed from the tree algorithm for packet broadcast channels by Capetanakis [12].

In the arbitration process of this algorithm, reader will send out requests to ask "Which tags existing?" according to the bits sequence of IDs from MSB. If more than one tag response, collisions occur. Then the collided tag split into two subsets, and reader keep on requesting

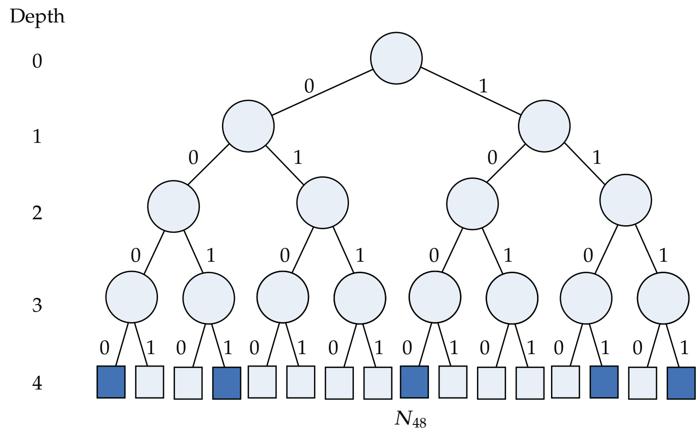

Figure 2: The structure of Binary Search Tree ($m = 5,\ k = 4$).

the next bit of the first subset. If there is still a collision, the further splitting proceeds recursively; otherwise tags in this subset are identified and no subsets will be queried until no collision in the former subset.

Figure 2 shows an example of binary tree whose depth is 4. Each tag corresponds to a leaf node in the binary search tree. There are 5 tags with Tag ID of 4-bit length in this example. For example, the ID of tag N_{48} is 1000.

Some parameters of the searching procedure of Binary Search Tree are defined as follows:

n: the number of all the possible tags in this RFID system

k: the length of bit stream to present a tag, it is equal to the depth of the binary tree. $k = \lceil \log_2 n \rceil$

m: the number of tags exist in the interrogation field of reader

N_{xy}: the yth node at depth x of the tree

T_{xy}: the subtree whose root node is N_{xy}.

The searching procedure of Binary Search Tree is stated as follows.

(1) Begin with the root node N_{00}, and ask if there is zero, one or more than one tag in the subtree T_{00}.

(2) If there are more than one tag in T_{00}, it means there is a collision that happened at N_{00}. Then ask the 2 succeeding nodes, N_{10} and N_{11}, to check whether there is zero, one, or more than one tag in their subtrees. If the first node N_{10} has collisions, let the second node N_{11} wait and ask the same question to succeeding nodes of N_{11}. The process follows the principle of "first in, first out."

(3) Recursively, if there is any collisions at nay node, repeat the same question to its succeeding nodes until there is no collision at the succeeding node.

If any node is waiting, resolve it by repeating step (3). If there is more than one waiting node, resolve them following the priority of "first in, first out." The pseudo codes of this algorithm are shown in Algorithm 1.

At the end of the process, each leaf will contain at most one tag. In the example of Figure 2, the order in which we find the tags is 1111, 1101, 1000, 0011, 0000. This algorithm needs to search 4 stages for finding a tag successfully.

```
/*binary search tree initialize and arbitrate*/
main()
{
    Initialize: BST = null;

    for each tag existing
        {
         BST_insert (BST,D1);
        }
    BST_arbitration (BST,D1);
}

/*binary search tree initialize*/
    BST insert (BST:a binary search tree,Da:the ath bit of tagID)
{
if (a = k) then return tag;

if (Da = 0) then {
                        a = a + 1;
                        BST_insert (left_subtree of BST,Da);
                        }
if (Da = 0) then {
                        a = a + 1;
                        BST_insert (right_subtree of BST,Da);
                        }
}

/*binary search tree arbitrate*/
   BST_arbitration (BST:a binary search tree,Da:the ath bit of tagID)
{
    if (collision happened at Da)
           then {
                    a = a + 1;
                    BST_arbitration (left_subtree of BST,Da);
                    BST_arbitration (right_subtree of BST,Da);
                    }
else ifentify the tagID;

}
```

Algorithm 1: Pseudo codes of the Binary Search Tree.

Several researches were conducted to improve tree algorithm and showed that some operation can improve the efficiency [18, 19].

3. System Model

3.1. Merged Search Tree Algorithm

In the original binary search tree, the researchers assume there are m out of n possible tags in the RFID system that exist in the field of reader with a bit stream of length $k = \lceil \log_2 n \rceil$

to present a tag. Meanwhile, a full binary tree with depth k is constructed to present all the tags. Since the drawback of Binary Search Tree is serious collisions when the majority of possible tags exist at the same time. Collisions always happen at each query and a skewed tree may form in certain situations to cause data clustering which may result in an increase in searching time. The purpose of this study is modifying the schema of Binary Search Tree to adapt to better performance.

The analysis of RFID anticollision performance that brings up the number of stages for searching a tag successfully is interrelated to searching cost [18, 20]. In order to reduce the number of stages for searching a tag successfully, the length of the bit stream is shortened to represent a tag. Therefore, Stages of Binary Search Tree are merged into one or more stage and representation of Tag ID is transformed from binary number to a new form combined with decimal and binary numbers. The Merge Operations is defined as follows.

The process of Merge Operation

(1) Merging Stages Determining.

  (i) Computing the Synthesis Evaluating Function, $\mathrm{SEF}(s, m, n)$, for $s = 0$ to $\lceil \log_2 n \rceil - 1$, s stands for the number of stages which is intended to merge. The s with minimum $\mathrm{SEF}(s, m, n)$ will be chosen.

(2) Rebuilding

  (i) Increase the branches of root node and number of nodes in depth 1 form 2 to 2^s. Take the 2^s nodes at depth 1 as root nodes to build binary subtrees, each has a depth of $\lceil \log_2 n \rceil - \mathrm{s} + 1$.

  (ii) Update the label of each branch of the root node to decimal number 0 to $2^s - 1$.

After rebuilding the tree, nodes are searched according to the ordinary search steps of Binary Search Algorithm. Figure 3 shows the result of merging the tree with $s = 2$ and it successfully reduces the depth of the tree from 4 to 3. Eventually, nodes have new representation and new Tag IDs. For example, the leaf node N_{48} becomes N_{38} and its new ID is 200.

Especially, the stages of tree are reduced from k to $k - s + 1$, and the number of nodes in the Merge Search Tree is $2^{k+1} - 2^s + 1$, which is less than $2^{k+1} - 1$, the number of nodes in Binary Search Tree. At the same time, the distribution of tags is relatively dispersed due to the increasing of branches in level 1 and this would help decrease the collisions between tags. Therefore, the structure of the tree has been successfully simplified to adapt more efficient search process.

3.2. The Synthesis Evaluating Function

The Synthesis Evaluating Function, SEF, is a function of s, m, n. According to m and n, SEF (s, m, n) can be used to determine the most appropriate s for merging operation to optimize the performance. Since the time and power consumed during the arbitration process are the most important issues to determine the performance of an RFID anticollision algorithm [20, 22], both time and power consumed are taken into account when there are different amounts of existing tags.

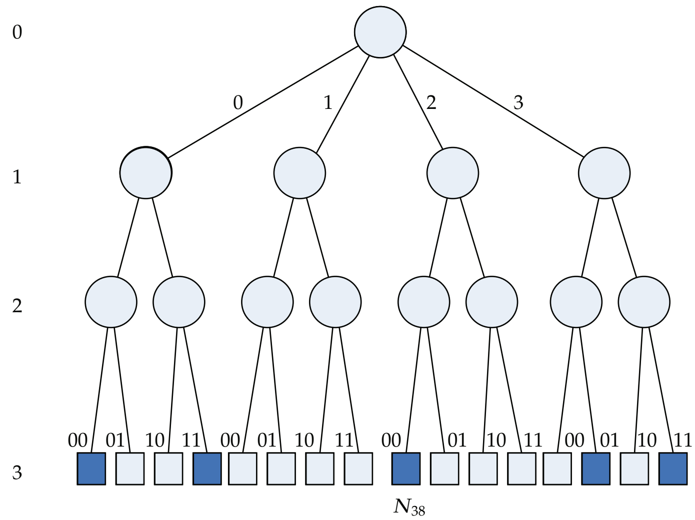

Figure 3: The structure of Merged Search Tree ($s = 2,\ m = 5$).

3.2.1. *Time Evaluating Function*

Hush and Wood [22] proposed the model to evaluate the time consuming of binary tree [22]. The total time required to recognize each tag is proportional to the total time slots to visit all the nodes. $q_{Li,m}$ is the probability that the algorithm will visit the ith node in depth L. The algorithm begins at root node, so $q_{00,m}$ is equal to 1. In the depth $L > 0$, a node will be visited if and only if there is a collision that happened in their parent nodes. In the research of tree algorithm, the uniform distribution of j out of the m tags which fall in a node of depth L is assumed and given by the binomial distribution:

$$p\left(\frac{j}{m}\right) = \binom{m}{j} B^{-L}\left(1 - B^{-L}\right)^{m-j}. \tag{3.1}$$

Assume $\beta_{Li,m}$ is the probability of a collision that happened at the ith node in depth L. It is equal to subtract the probability of zero reply and only one tag reply form 1, as shown in

$$\beta_{Li,m} = \beta_{L,m} = 1 - \left(1 - 2^{-L}\right)^{m} - m2^{-L}\left(1 - 2^{-L}\right)^{m-1}. \tag{3.2}$$

Therefore the probability $q_{Li,m}$ is

$$\begin{aligned} q_{Li,m} &= q_{00,m} = 1, \quad L = 0, \\ q_{Li,m} &= q_{L,m} = \beta_{L-1,m}, \quad L > 0. \end{aligned} \tag{3.3}$$

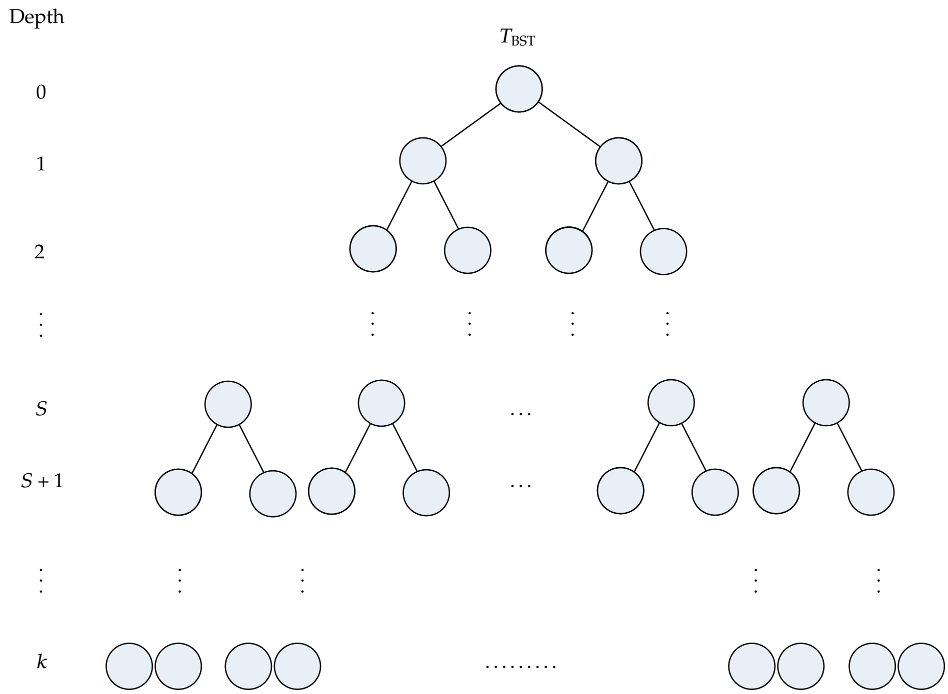

Figure 4: T_{BST}, the original Binary Search Tree.

The total time slots needed to arbitrate the m tags during the process of original Binary Search Tree is $\overline{T}_{\mathrm{BST}}(m)$. Numerical calculating of $\overline{T}_{\mathrm{BST}}(m)$ is shown in

$$\begin{aligned}
\overline{T}_{\mathrm{BST}}(m) &= \sum_{L=0}^{\infty}\sum_{i=0}^{2^L-1} q_{Li,m} \\
&= 1 + \sum_{L=1}^{\infty} 2^L \beta_{L-1,m} \\
&= 1 + \sum_{L=1}^{s} 2^L \beta_{L-1,m} + \sum_{L=s+1}^{\infty} 2^L \beta_{L-1,m} \\
&= 1 + \sum_{L=1}^{\infty}\left[2^L\left(1 - \left(1 - 2^{-(L-1)}\right)^m - m2^{-(L-1)}\left(1 - 2^{-(L-1)}\right)^{m-1}\right)\right].
\end{aligned} \tag{3.4}$$

$\overline{T}_{\mathrm{BST}}(m)$ is the total time slots needed during the arbitration process of Merged Binary Tree. A Binary Search Tree with depth k, T_{BST}, is shown in Figure 4. The result after merging with s, T_{MST}, is shown in Figure 5. Compare T_{MST} with T_{BST}, and the result shows the nodes at depth $1'$ in T_{MST} corresponding to the nodes at depth s in T_{BST}, as well as the nodes at depth $2'$ in T_{MST} corresponding to the nodes at depth $s+1$ in T_{BST}. To reason by analogy, the total time slots of probing nodes at depth L in T_{MST} are equal to time slots probing nodes at depth $L+s-1$ in T_{BST}.

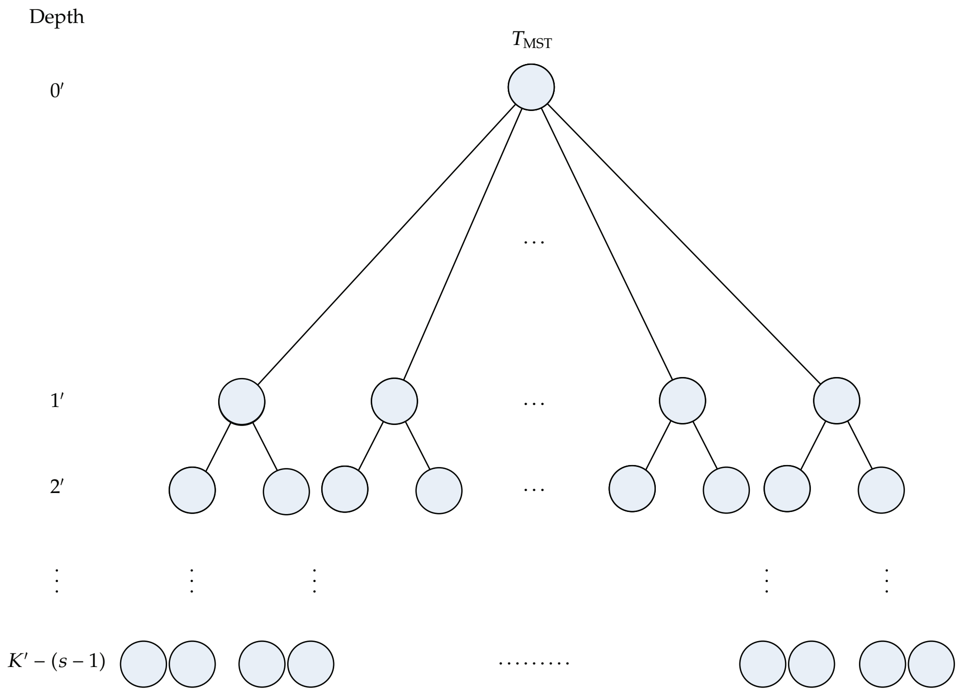

Figure 5: T_{MST}, the tree merged from T_{BST}.

Hush and Wood [22] proposed that if the tags are uniformly distributed, visiting a node can be viewed as probing an interval of size 2^{-L}, hence that the probability of collisions happened in a node at depth L' in T_{MST} is equal to the probability at depth $L + s - 1$ in T_{BST} [22]. Assume the probability that the Merged Tree algorithm will visit the ith node at depth L' in T_{MST} is $q_{L'i,m}$; therefore the probability $q_{L'i,m}$ is

$$\begin{aligned} q_{L'i,m} &= q_{0',m} = 1, \quad L' = 0 \\ q_{L'i,m} &= q_{1',m} = \beta_{0,m}, \quad L' = 1 \\ q_{L'i,m} &= \beta_{L+S-2,m}, \quad L' > 1. \end{aligned} \tag{3.5}$$

The number of total time slots is computed as shown in

$$\begin{aligned} \overline{T}_{\text{MST}}(m) &= \sum_{L'=0}^{\infty} \sum_{i=0}^{2^{L'}-1} q_{L'i,m} \\ &= 1 + 2^s q_{1',m} + \sum_{L=2}^{\infty} 2^{L'} q_{L',m} \\ &= 1 + 2^s \beta_{0,m} + 2^{s+1}\beta_{s,m} + 2^{s+2}\beta_{s+1,m} + \cdots + 2^{\infty}\beta_{\infty,m} \\ &= 1 + 2^s \beta_{0,m} + \sum_{L=s+1}^{\infty} 2^L \beta_{L-1,m}. \end{aligned} \tag{3.6}$$

Compare (3.4) with (3.6) and obtain

$$\overline{T}_{\text{MST}}(m) = \overline{T}_{\text{BST}}(m) + 2^s\beta_{0,m} - \sum_{L=1}^{s} 2^L \beta_{L-1,m} \tag{3.7}$$

$$= \overline{T}_{\text{BST}}(m) + 2^s - \sum_{L=1}^{s}\left[2^L\left(1-\left(1-2^{-(L-1)}\right)^m - m2^{-(L-1)}\left(1-2^{-(L-1)}\right)^{m-1}\right)\right] \tag{3.8}$$

$$= 1 + 2^s + \sum_{L=s+1}^{\infty}\left[2^L\left(1-\left(1-2^{-(L-1)}\right)^m - m2^{-(L-1)}\left(1-2^{-(L-1)}\right)^{m-1}\right)\right]. \tag{3.9}$$

Here the number of total time slots of Merged Search Tree arbitration is obtained. Equations (3.7) and (3.8) stand for the comparison between $\overline{T}_{\text{MST}}(m)$ and $\overline{T}_{\text{BST}}(m)$ and (3.9) shows the numerical calculation of $\overline{T}_{\text{MST}}(m)$.

3.2.2. Power Evaluating Function

The power consumed during arbitration process is another important issue of evaluating a RFID anticollision algorithm. The tags' power consumption will determine the working distance between the tags and reader. The lower power consumption will allow longer working distance [20]. The power consumption evaluating model of Hush and Wood [22] is applied and modified for Merged Search Tree in this research. The total number of tag replies is computed, which is proportion to the total power consumed during RFID arbitration process [22]

$$u_{Li,m} = P\left(\frac{1}{m}, L\right) = \binom{m}{k} p^1 (1-p)^{m-1}. \tag{3.10}$$

The probability of single reply happened at node N_{Li} is represented as $u_{Li,m}$ as shown in (3.10) and the average number of tags identified by algorithm, up to and including level L, $\overline{S}_{\text{BST}}(m, L)$, is equal to the sum of $u_{Li,m}$ overall the nodes at level L, as shown in

$$\overline{S}_{\text{BST}}(m, L) = \sum_{i=0}^{2^L-1} U_{Li,m} = m\left(1-2^{-L}\right)^{m-1}. \tag{3.11}$$

Let $\overline{R_{\text{BST}}}(m)$ be the total sum of tag replies of each node at each levels. There are m replies at level 0 and the number of replies is equal to $m - \overline{S}_{\text{BST}}(m, L-1)$ at the other level. Therefore, $\overline{R_{\text{BST}}}(m)$ is shown in

$$\overline{R_{\text{BST}}}(m) = m + \sum_{L=1}^{k}\left(m - \overline{S}_{\text{BST}}(m, L-1)\right). \tag{3.12}$$

According to (3.12),

$$\begin{aligned}\overline{R_{\mathrm{BST}}}(m) &= m + \sum_{L=1}^{k}\left(m - m\left(1 - 2^{-(L-1)}\right)^{m-1}\right)\\ &= 2m + \sum_{L=2}^{k}\left(m - m\left(1 - 2^{-(L-1)}\right)^{m-1}\right).\end{aligned} \tag{3.13}$$

Here the number of total tag replies for Binary Search Tree algorithm arbitrating m tags is obtained.

$\overline{R_{\mathrm{MST}}}(m)$, the total number of tag replies in Merged Search Tree, is accounted in similar way to Binary Search Tree. Comparing the results in Figures 11 and 12, there are m tag replies at the level $0'$ in T_{MST}, the same as level 0 of Binary Search Tree. At level $1'$ of T_{MST}, $m - m(1-(2^s)^0)^{m-1} = m$ tag replies are needed. For levels $L = 2'$ to $k' - (s-1)$ of T_{MST}, the number of tag replies at level L is equal to the number of tag replies at level $L+s-1$ of T_{BST}. $\overline{R_{\mathrm{MST}}}(m)$ is counted as the sum of tag replies at each level as shown in

$$\begin{aligned}\overline{R_{\mathrm{MST}}}(m) &= m + m + \sum_{L=s+1}^{k}\left(m - \overline{S}_{\mathrm{MST}}(m, L-1)\right)\\ &= 2\,m + \sum_{L=s+1}^{k}\left(m - m\left(1 - 2^{-(L-1)}\right)^{m-1}\right).\end{aligned} \tag{3.14}$$

3.2.3. The SEF Function

Since both of the time and power performances are computing the summation of expected value which the parameters are considering each tags as a unit, the function $\mathrm{SEF}(s, m, n)$ is defined as

$$\mathrm{SEF}(s,m,n) = \begin{cases} \left[1 + 2^s + \displaystyle\sum_{L=s+1}^{\infty}\left(2^L\left(1 - \left(1 - 2^{-(L-1)}\right)^m - m2^{-(L-1)}\left(1 - 2^{-(L-1)}\right)^{m-1}\right)\right)\right] \\ \quad + \left[2\,m + \displaystyle\sum_{L=s+1}^{\lceil \log_2 n\rceil}\left(m - m\left(1 - 2^{-(L-1)}\right)^{m-1}\right)\right], \quad \text{if } 2 \le s \le \lceil \log_2 n\rceil \\ \left[1 + \displaystyle\sum_{L=1}^{\infty}2^L\left(\left(1 - \left(1 - 2^{-(L-1)}\right)^m - m2^{-(L-1)}\left(1 - 2^{-(L-1)}\right)^{m-1}\right)\right)\right] \\ \quad + \left[2\,m + \displaystyle\sum_{L=2}^{\lceil \log_2 n\rceil}\left(m - m\left(1 - 2^{-(L-1)}\right)^{m-1}\right)\right], \quad \text{if } s = 0 \text{ or } 1. \end{cases} \tag{3.15}$$

Equation (3.15) is used to choose s. The lower value of $\mathrm{SEF}(s, m, n)$ is, the better s is chosen to achieve best performance. The pseudo codes of this algorithm is shown in Algorithm 1.

4. Performance Evaluations

The performance of Binary Search Tree, and Merged Search Tree are evaluated.

4.1. Time Consumption

Equation (3.8) shows the difference between tag replies of Binary Search Tree and Merged Search Tree. In the environment of RFID systems for commercial application, there are usually a lot of tags existing in the field of reader, such as the warehouse or automation control. In these RFID systems, the number of m is large. Since $1 - 2^{-(L-1)}$ is a fractional number, the terms $(1 - 2^{-(L-1)})^m$ and $(1 - 2^{-(L-1)})^{m-1}$ will approach 0 when m is greater than a threshold which results in the situation as shown in (4.1). The equation is established for most of the generality of situations

$$2^s - \sum_{L=1}^{s}\left[2^L\left(1 - \left(1 - 2^{-(L-1)}\right)^m\right) - m2^{-(L-1)}\left(1 - 2^{-(L-1)}\right)^{m-1}\right] \cong 2^s - \sum_{L=1}^{s}2^L \leq 0. \tag{4.1}$$

Different number of s is applied to observe the threshold of m to achieve a situation that is adapted to the merging stages number in the RFID system with $n = 256$. The result is shown in Figure 6. For example, if $s = 2$, while there are more than 3 tags existing in the operation range in a system with $n = 512$, (4.1) is established. Roughly speaking, when there are more than 14 tags, the better performance is achieved no matter which s is, Figure 7 shows the threshold of m with different n, when m is greater than the threshold, (4.2) is established as follows:

$$\begin{aligned}\overline{T}_{\mathrm{MST}}(m) &= \left[1 + 2^s + \sum_{L=s+1}^{\infty}\left(2^L\left(1 - \left(1 - 2^{-(L-1)}\right)^m\right)\right) - m2^{-(L-1)}\left(1 - 2^{-(L-1)}\right)^{m-1}\right] \\ &= \overline{T}_{\mathrm{BST}}(m) + 2^s - \sum_{L=1}^{s}\left[2^L\left(1 - \left(1 - 2^{-(L-1)}\right)^m\right) - m2^{-(L-1)}\left(1 - 2^{-(L-1)}\right)^{m-1}\right] \\ &\leq \overline{T}_{\mathrm{BST}}(m).\end{aligned} \tag{4.2}$$

Equation (4.2) shows that a better time performance is expected in the Merged Search Tree.

4.2. Power Consumption

The power consumption of each model is evaluated by total number of tag replies. Subtract (3.14) from (3.13) and obtain

$$\overline{R}_{\mathrm{BST}}(m) - \overline{R}_{\mathrm{MST}}(m) = \sum_{L=1}^{s}\left(m - m\left(1 - 2^{-(L-1)}\right)^{m-1}\right). \tag{4.3}$$

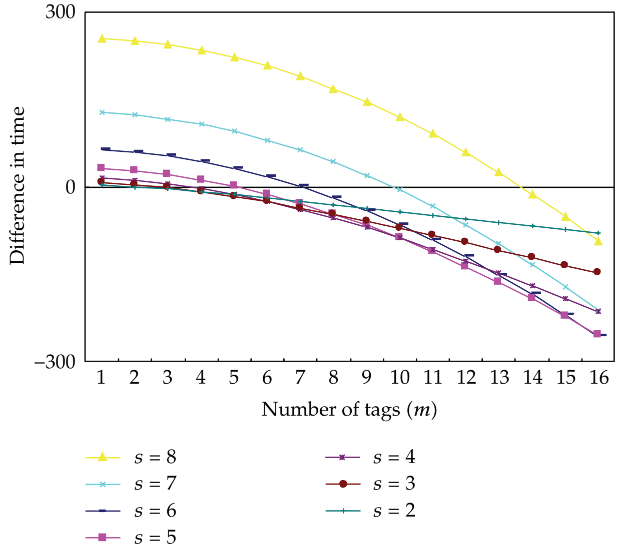

Figure 6: Different number of s with the threshold of m to achieve a situation.

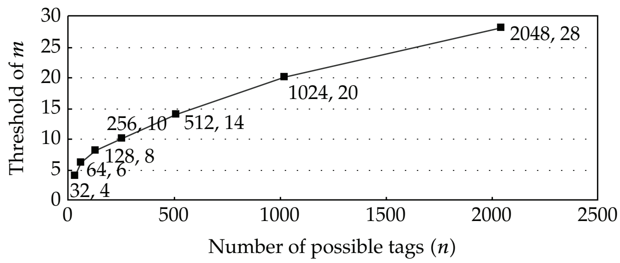

Figure 7: The threshold of m with different n.

Since the item $(1-2^{-(L-1)})^{m-1}$ is a fractional number, the number m would be greater than $m(1-2^{-(L-1)})^{m-1}$, such that

$$\overline{R}_{\mathrm{BST}}(m) - \overline{R}_{\mathrm{MST}}(m) = \sum_{L=1}^{s}\left(m - m\left(1-2^{-(L-1)}\right)^{m-1}\right) \geq 0. \tag{4.4}$$

It reveals that Merged Search Tree consumed lower power to arbitrate all the tags than Binary Search Tree algorithm.

$\overline{R}_{\mathrm{QT}}(m)$ is the total tag replies of Quadtree algorithm and substitute the item $2^{-(L-1)}$ in (3.13) by $4^{-(L-1)}$. However, the depth of T_{QT} is decreased to $[k/2]$, so $\overline{R}_{\mathrm{QT}}(m)$ is computed as

$$\overline{R}_{\mathrm{QT}}(m) = 2m + \sum_{L=2}^{[k/2]}\left(m - m\left(1-4^{-(L-1)}\right)^{m-1}\right). \tag{4.5}$$

Equations (3.13), (3.14), (3.15), and (4.5) are used to evaluate the performances of power consumed in different algorithms.

4.3. Simulation Results and Discussion

4.3.1. Simulation Results

In the improved schemas, as the number of branches in level 1 increased, the more dispersion of existing tags is expected to reduce the number of tags at which collisions happened. In the view of power, fewer collisions cause lower number of tag replies and tag communication overhead [23]. Regarding the time consumption, the total number of time slots is equal to the sum of the time slots of collisions, single reply, and zero reply. Reducing of collisions will increase zero replies, which may increase the total number of time slots. The Synthesis Evaluating Function is applied to consider time and power consumption simultaneously by determining b and s. The simulation results of different number of m are discussed in the following.

(A) Total Number of Possible Tags = 128

Figures 8 and 9 show the simulation of performance on the original Binary Search Tree and Merged Search Tree in the environment that at most 128 tags ($n = 128$) could exist simultaneously. Consider the situations of 1 to 128 ($m = 1$ to 128) tags existing at the same time.In Figure 8, all of the algorithms need almost the same time slots when less than 30 tags exist. Accurately, the time slots of Merged Search Tree are less than Binary Search Tree but not obvious in the figure due to the small volume. However, when more than 30 tags exist, it is apparent that Merged Search Tree is similar to each other on time performance and both are better than the other. Especially, Binary Search Tree almost has the same performance on time. In Figure 9, the difference among them is increasing. The power consumption of Merged Search Tree is reduced to almost 35%. Both algorithms have the trend that when more tags exist, the more time slots and tag replies increase linearly.

(B) Total Number of Possible Tags = 512

Figures 10 and 11 show the simulation results of performance of each algorithm when the total number of possible tags is 512. In Figure 10, Merged Search Tree 1 outperforms on time performance than Binary Search Tree. However, Binary Search Tree and Merged Search Tree need similar time slots to arbitrate. In Figure 11, as the number of tags increase, the power consumption improved. For example, in the case $m = 400$, average tag replies involved by Binary Search Tree and Merged Search Tree are 3897.839 and 1116.0838. A great improvement on power consumption is achieved.

(C) Total Number of Possible Tags = 1024

Figure 12 shows that the simulation result of the merged algorithm still outperform the unmerged. However, the performance of Merged Search Tree is not entirely linear as Binary Search Tree. This is the effect of considering time and power consumption simultaneously and therefore there is no obvious improvement in few cases such as $m = 122$. However, two

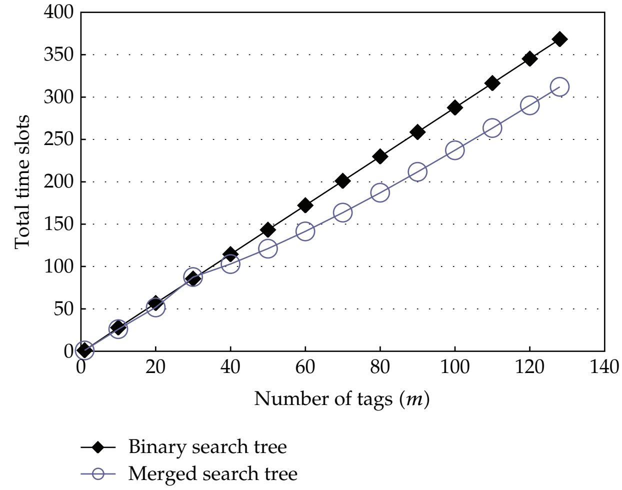

Figure 8: Time consumed with $m = 1$ to 128.

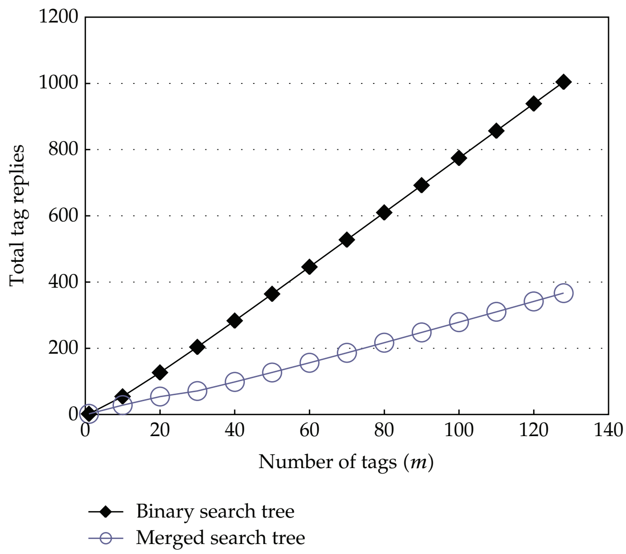

Figure 9: Power consumed with $m = 1$ to 128.

merged methods perform better than the two unmerged. In Figure 13, tag replies Merged Search Tree is reduced almost 75% of Binary Search Tree.

4.3.2. Discussion

The results of simulation on proposed methods reveal the following.

(1) When m is relatively close to n, both the two algorithms will determine the same combination of b and s to rebuild the tree. In other words, considering the branch number on each node in the merged tree is improved limitedly that it cannot

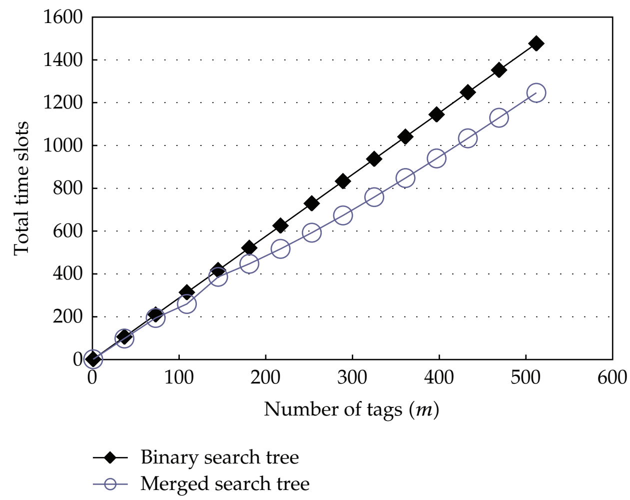

Figure 10: Time consumed with $m = 1$ to 512.

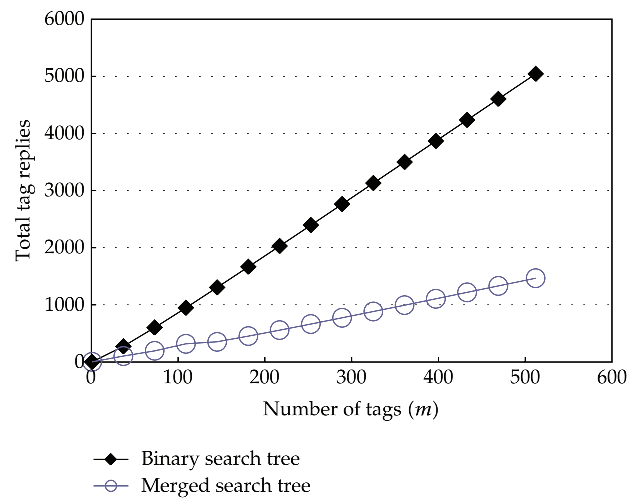

Figure 11: Power consumed with $m = 1$ to 512.

improve a Merged Search Tree as the degree that quadtree improved Binary Search Tree. Nevertheless, the proposed merged tree is better than binary tree.

(2) The function SEF considered time and power consumption at the same time. It is possible that this mechanism determined a structure not the fastest or cost lowest, but both of the two performances can be optimized to achieve a balance between speed and cost.

5. Conclusion

In the applications of RFID, when there exist multitags in the reading field of reader, collisions happen due to the failure of tags synchronous transmission. This limits the development

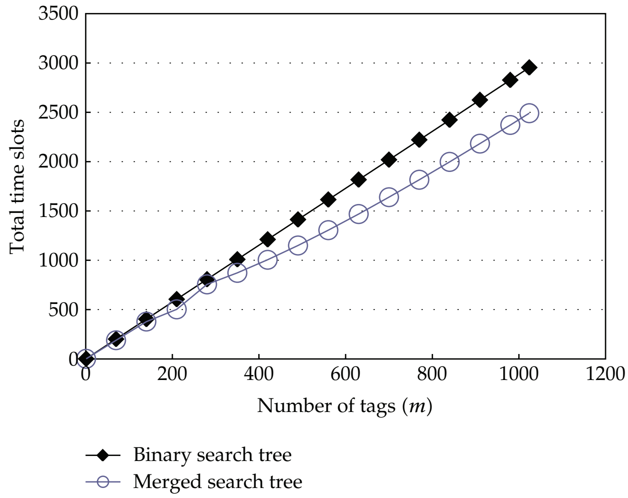

Figure 12: Time consumed with m = 1 to 1024.

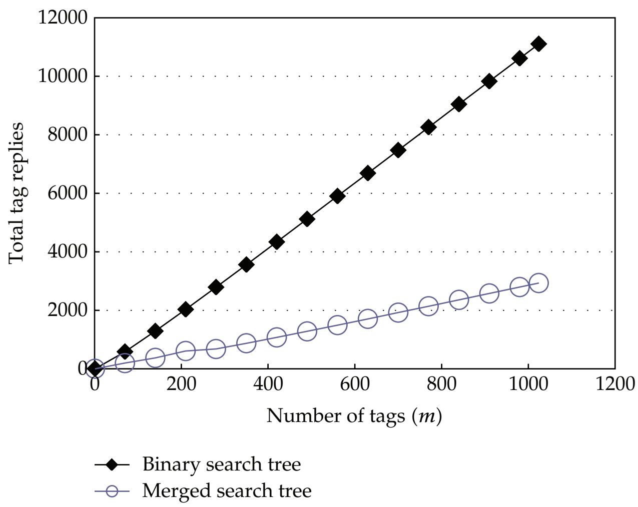

Figure 13: Power consumed with m = 1 to 1024.

of RFID applications. Although many mechanisms were introduced to avoid this problem, they still have some limits and are out of flexibility. The current study has successfully developed a splitting-based schema to improve the schema of Binary Search Tree defined in the anticollision protocol of EPC global UHF RFID system. In this study, Merged Search Tree is successfully developed. According to the relation between number of existing tags and possible tags, the function SEF will determine the appropriate tree structure. The performance of each algorithm is evaluated based on time and power consumption during arbitration process and compared with each other. Merged Search Tree can improve greatly both on time and power performance simultaneously.

Acknowledgments

The authors are appreciative of the financial support in the form of research grants from the National Science Council, Republic of China under Grant nos. NSC 99–2628-E-153-001, NSC 100-2221-E-022 -013 -MY2, NSC 100-2628-E-022 -002 -MY2 and NSC 100–2628-E-153-001. The authors are also most grateful for the kind assistance of Professor Andrzej Swierniak, Editor of Mathematical Problems In Engineering, and for the constructive suggestions from the anonymous reviewers, all of which has led to the making of several corrections and have greatly aided us to improve the presentation of this paper.

References

[1] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.
[2] EPC global, "900 MHz Class 0 Radio Frequency (RF) Identification Tag Specification," http://www.epcglobalinc.org/standards_technology/specifications.html.
[3] EPC global, "13.56 MHz ISM Band Class 1 Radio Frequency (RF) Identification Tag Interface Specification," http://www.epcglobalinc.org/standards_technology/specifications.html.
[4] D. H. Shih, P. L. Sun, D. C. Yen, and S. M. Huang, "Taxonomy and survey of RFID anti-collision protocols," *Computer Communications*, vol. 29, no. 11, pp. 2150–2166, 2006.
[5] S. Sarma, D. Brock, and D. Engels, "Radio frequency identification and the electronic product code," *IEEE Micro*, vol. 21, no. 6, pp. 50–54, 2001.
[6] A. Rohatgi and G. D. Durgin, "Implementation of an anti-collision differential-offset spread spectrum RFID system," in *Proceedings of the IEEE Antennas and Propagation Society International Symposium*, pp. 3501–3504, 2006.
[7] EPCglobal Inc., "Specification for RFID Air Interface," Version1.0.9, 2005.
[8] W. I. Lee, "The development of a qualitative dynamic attribute value model for healthcare institutes," *Iranian Journal of Public Health*, vol. 39, no. 4, pp. 15–25, 2010.
[9] W. Lee and C. H. Lin, "Consumer hierarchical value map modeling in the healthcare service industry," *African Journal of Business Management*, vol. 5, no. 3, pp. 722–736, 2011.
[10] W. I. Lee, C. W. Chen, and C. H. Wu, "Relationship between quality of medical treatment and customer satisfaction—a case study in dental," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 4, pp. 1805–1822, 2010.
[11] B. Y. Shih, C. J. Chang, A. W. Chen, and C. Y. Chen, "Enhanced mac channel selection to improve performance of ieee 802.15.4," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 12, pp. 5511–5526, 2010.
[12] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 505–515, 1979.
[13] H. Vogt, "Multiple object identification with passive RFID tags," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 6–9, October 2002.
[14] M. Jacomet, A. Ehrsam, and U. Gehrig, "Contactless identification device with anticollision algorithm," in *Proceedings of the IEEE (CSCC '99)*, Athen Greece, 1999.
[15] F. Zhou, D. Jin, C. Huang, and M. Hao, "Optimize the power consumption of passive electronic tags for anti-collision schemes," in *Proceedings of 5th International Conference on ASIC*, vol. 2, pp. 1213–1217, 2003.
[16] B. Y. Shih, T. W. Lo, and C. Y. Chen, "The research of quadtree search algorithms for anti-collision in radio frequency identification systems," *Scientific Research and Essays*, vol. 6, no. 25, pp. 5342–5350, 2011.
[17] ISO/IEC 18000-6:2003(E), Part 6, "Parameters for air interface communications at 860–960 MHz," November 2003.
[18] T. P. Wang, "Enhanced binary search with cut-through operation for anti-collision in RFID systems," *IEEE Communications Letters*, vol. 10, no. 4, pp. 236–238, 2006.
[19] M. Nanjundaiah and V. Chaudhary, "Improvement to the anticollision protocol specification for 900 MHz class 0 radio frequency identification tag," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA '05)*, vol. 2, pp. 616–620, March 2005.

[20] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, "Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems," in *Proceedings of the International Symposium on Lower Power Electronics and Design (ISLPED '04)*, pp. 357–362, August 2004.

[21] MIT Auto-ID Center, "Draft protocol specification for a 900 MHz Class 0 Radio Frequency Identification Tag," February 2003, http://auto-id.mit.edu.

[22] D. R. Hush and C. Wood, "Analysis of Tree algorithms for RFID arbitration," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '98)*, p. 107, 1998.

[23] J. Myung, W. Lee, and J. Srivastava, "Adaptive binary splitting for efficient RFID tag anti-collision," *IEEE Communications Letters*, vol. 10, no. 3, pp. 144–146, 2006.