

## Designing Fast Identification Algorithm in the RFID System for Known Tag Sets<sup>1</sup>

Chia-Sheng Tsai\* Yu-Cheng Wang

Department of Computer Science and Engineering, Tatung University, Taiwan

### Abstract

RFID is an automatic identification technology which exercises the radio frequency to retrieve the information saved in electronic tags. In a multi-tag environment, the collision problem caused by a number of tags simultaneously responding to a reading will jeopardize the speed of identification. Some scheme is necessary to reconcile the conflict and improve the reading time. A novel scheme is proposed in this paper to address this collision problem. In addition, our scheme can be trained to accelerate the identification of a subset of the known tags. Although there are similar schemes capable of achieving this function, our proposal accommodates rare presence of those tags outside the known set. In the simulation, we evaluate the performance of several tag anti-collision protocols. Specifically, we focus on the total reading time and the amount of messages required accomplishing the reading process. We observe that our scheme can effectively reduce both the reading time and the number of reading messages. In the best case, the proposed scheme saves 50% and 70% time when compared with the query tree algorithm (QTA) and binary tree algorithm (BTA) methods respectively. These improvements not only accelerate the reading speed but also save the on-board power in the reader.

*Key Words: Radio Frequency Identification, Anti-Collision Scheme, Electronic Tag Identification, Electronic Tag Collision*

---

<sup>1</sup> This work was supported in part by the National Science Council, Taiwan, under the Grants NSC 98-2221-E-036-033, NSC 99-2221-E-036-017, and by the Tatung University under the Grant B100-I09-038. Also, thanks are given to the anonymous reviewers for their helpful comments.

\* Corresponding author

E-mail: cstsai@ttu.edu.tw



# 快速辨識無線射頻電子標籤演算法之設計

蔡佳勝 王昱晟

大同大學資訊工程所

## 摘要

無線射頻辨識是一種自動辨識技術，這種技術利用無線電頻率去獲得電子標籤內的資訊。在多個電子標籤的環境中，數電子標籤同時回應讀取器將會成碰撞的問題而危及識別的速度。故此，某些為調解衝突與改善讀取時間的機制是必要的。本文提出一個新的機制以改善上述問題。另外，此機制可以被訓練來加快辨識已知的電子標籤。儘管有一些相似的機制能夠達到同樣的效果，但此機制可以識別未知的電子標籤等罕見狀況。在模擬情況中，本文針對數個電子標籤防碰撞機制進行績效評估。更精確而言，本文著重在完成讀取所須的時間和訊息數。我們觀察得知本機制能有效減少讀取時間和讀取訊息。在最佳情況下，相較於詢問樹演算法本機制能節省 50% 的時間；相較於二元樹演算法本機制能節省 70% 的時間。這些改善不只加快了讀取的速度，也節省了讀取器中之電力。

關鍵詞：無線射頻辨識、防碰撞機制、電子標籤識別、電子標籤碰撞

## 1. Introduction

Fast tag identification is an important performance objective in the RFID system since the information stored in the RFID tag must be read quickly and accurately in many applications. However, it is not an easy task to achieve fast tag identification among various configurations of the RFID system. In recent years, many RFID tag anti-collision schemes have been proposed. They can be divided into two classes, probabilistic algorithm (EPCglobal, 2005) and deterministic algorithm approaches (ISO/IEC, 2003). The probabilistic algorithms include the dynamic reading frame algorithm, the frame slotted aloha (FSA) algorithm, the dynamic framed slotted aloha (DFSA) algorithm, the estimation and binary selection frame slotted aloha (EB-FSA) algorithm, and so on (Vogt, 2002; Cha and Kim, 2005; Park et al., 2007). In probabilistic algorithm, the tag uses random access to respond to the reader. In these algorithms, the size of the frame can determine their reading performance. A frame can be divided into many slots. The tags can select whichever slot to send their IDs to the reader.



Generally, when the frame size increases, the probability of collision would decrease. But, the response time gets longer. In addition, certain mechanism may be needed to guarantee that each tag is successfully recognized by the reader. Otherwise, it is possible that some tags always collide with others and are never found by the reader. The deterministic algorithms contain the binary tree algorithm, the query tree algorithm, the cut-through algorithm, the bi-slotted query tree algorithm, along with other schemes (Finkenzeller, 2003; Law et al., 2000; Wang, 2006; Choi et al., 2006). In these deterministic algorithms, the reader sends out reading messages to identify the tags. The reading message includes the prefix of information. Those tags of the same prefix specified in the reading message will respond by sending the next bit after the prefix to the reader. The reader would continue using the prefix and following a tree structure to search for tags until all tags are identified. When the number of tags is small, the probability of collision with a specific prefix gets lower. However, this method is inefficient if the number of the tag is large due to the inflexibility of the tree structure. In addition, many existing schemes cannot remember the existing tags now. That is, in some applications, the possible tags in the field are known already. This information should be taken advantage of. Yet, most anti-collision algorithms do not put it in good use. Hence, they will waste many time to repeat the reading process even though the tags rarely change. Few existing algorithms can remember the existing tag such as the cut-through algorithm. But the algorithm cannot tolerate any tag outside the known tag set. In order to improve these schemes, the novel scheme proposed in this paper.

Our main idea is researching the RFID tag anti-collision schemes in the category of the deterministic algorithms. The proposed algorithm can train the reader to remember a set of existing IDs. Most importantly, dynamically adding IDs not in the existing set is considered. Hence, the scheme can achieve fast tag identification. In Section 3, we will give an in-depth introduction to our RFID tag anti-collision method. In our proposal, the RFID reader will make use of the tag identities discovered previously. This information will enable our algorithm to construct a search tree reducing the look up time for future identification.

The remainder of this paper is structured as follows. In Section 2, we will describe a variety of RFID anti-collision protocols and discuss their strengths and weaknesses. In Section 3, we will introduce our proposed RFID anti-collision scheme. In Section 4, the proposed algorithm compared with some of the existing ones. We will explain our simulation model and analyze the simulation results. Finally, we conclude the paper in Section 5, which highlights the main contributions of this research. It also points out future research directions.



## 2. Related Works

The RFID tag anti-collision schemes can be categorized into two classes, the reader anti-collision (Waldrop et al., 2003a; Waldrop et al., 2003b; Kim et al., 2006; Ho et al., 2006; Kim et al., 2007a; Birari and Iyer, 2005) and the tag anti-collision (Finkenzeller, 2003; Law et al., 2000; Wang, 2006; Choi et al., 2006; Myung et al., 2006; Choi et al., 2007a; Choi et al., 2007b; Kim et al., 2007b) techniques. Here, we focus on the latter issue in this paper.

In this Section 2, the existing tag anti-collision schemes are surveyed. These schemes can be used to address the tag collision problem. We will introduce two tag anti-collision schemes. These schemes can be used to effectively address the reader collision problem. They are binary tree algorithm in Section 2.1 and query tree algorithm in Section 2.2.

### 2.1 Binary tree algorithm

In Finkenzeller (2003), a binary tree algorithm (BTA) was proposed. In this algorithm, the reader follows the binary tree and polls tags of the same prefix. Those tags of the same prefix specified in the reading message will respond by sending the next bit after the prefix to the reader. The reader will observe one of the three cases, i.e., a collision, a single response, and no response. If there is a collision, the reader will append a 0 or 1 to the previous prefix and form an extended prefix in the next reading message. Generally speaking, the reader goes down the binary tree to resolve the collision. If there is only one response, then the reader will attach the received bit to the previous prefix and send out the new prefix in a reading message. Since there is only one tag responding to the message, the ID of the tag will be fully discovered by the reader as it repeats this procedure. Finally, if there is no response, then the reader knows there is no tag in the branch and switch to neighbor branch.

### 2.2 Query tree algorithm

Law et al. (2000) proposed a protocol called query tree algorithm (QTA). In the QTA algorithm, a query-tree based algorithm proposed to govern the order of transmissions among the tags. In this algorithm, the reader follows a binary query tree to inquire the tags. If the prefix of a tag ID matches what is sent by the reader, then the tag responds by sending out all remaining bits in its ID after the prefix to the reader. If there is only one such tag, then the reader recognizes the tag immediately and no collision needs to be resolved. However, when there are two or more tags with the same prefix, all respond and the reader sees a collision from the garbled message. The reader can resolve the collision by being more and more pre-



cise on its query, i.e., specifying more bits in the prefix. In the algorithm, the tag is stateless. In other words, it does not have to remember if it already responds to the reader in a reading process. The only operation required of the tag is the comparison of its ID against the prefix in the reading message. Eventually, all tags will be successfully read exactly once since their IDs are unique.

So far, two existing schemes reviewed that can be used to find out the tags in the RFID system. These schemes choose to forget the location information of the tags. If the reader has a good idea on the locations of the tags, it can be much efficient in the RFID system. In next section, we will describe how this efficiency can be achieved.

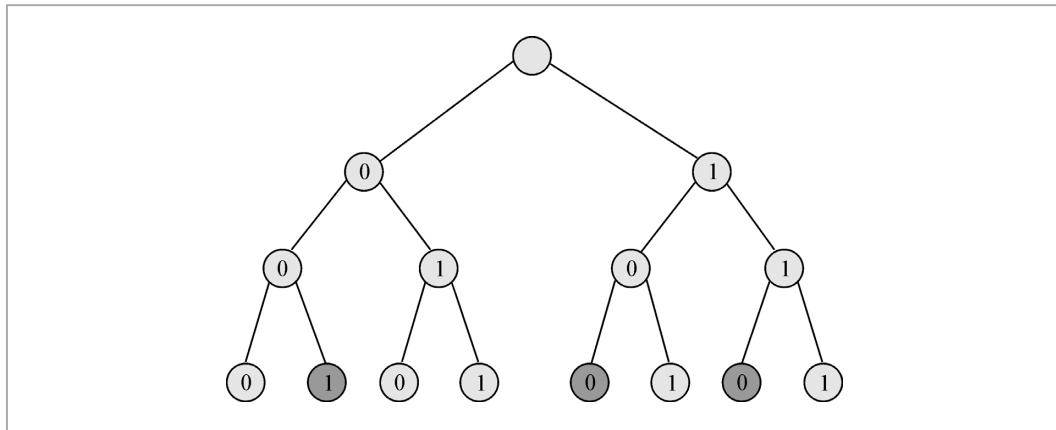
### 3. Fast Electronic Tags Identification Technology for RFID System

In this section, we explain the architecture of the proposed algorithm using the similar cut-through scheme (Wang, 2006). In the algorithm, a binary tree built whose height is the same as the length of the tag ID. A node in the binary tree is classified as one of the three types depending on how many tags are there in the sub-tree root at the node.

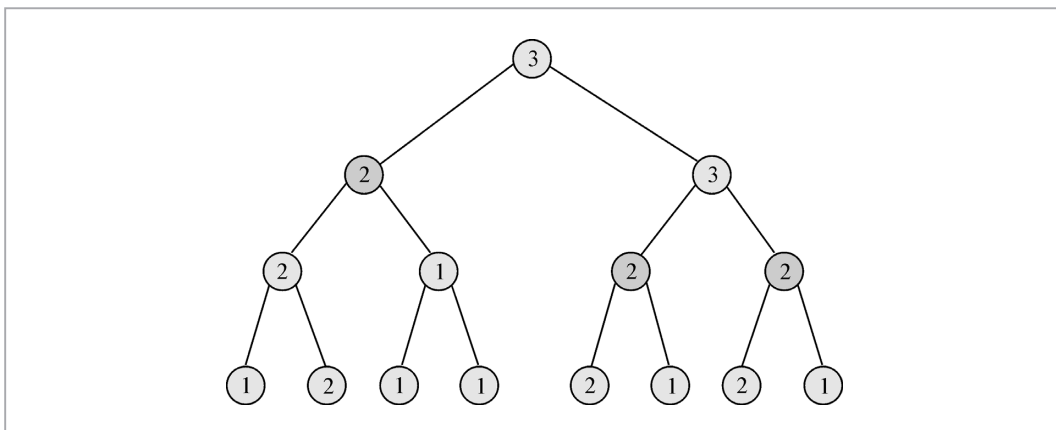
For every type 1 node, there is no tag in the sub-tree root at the node. Similarly, every type 2 node contains exact one tag while every type 3 node includes more than one tag in the sub-tree root at the node. The following is an example to further explain the proposed algorithm.

In Figure 1, we show a full binary tree of 8 leaves. Three tags whose IDs are 001, 100, and 110 are present and shaded in this example. First, the proposed algorithm identifies all tags with QTA anti-collision algorithm and remembers these tags. Then, it builds a new tree to enable expediting identification of these existing tags for next step. Figure 2 shows the node classification based on criteria just explained. The values in the circles indicate its types. It is evident that if the tags in the interrogation range of the reader must belong to the given set, then the optimal reading sequence contains the type 3 parent nodes. For instance, type 3 would always contain more than 1 tag; therefore, we would skip the type 3 reading point because collisions would always happen during type 3 processes.

There are three such type 2 nodes shaded in Figure 2. All of these nodes are shaded in the picture. Both the prefixes sent in the reading messages and the number of reading messages are optimized in this reading sequence. However, it is possible that a few tags not in the given set, then this method may miss these unexpected tags. In the following, we describe



▲ Figure 1 A binary tree example



▲ Figure 2 A tree constructed using the proposed algorithm

how to compensate for prediction errors.

First, the binary tree construction and node classification is carried out as stated previously. The reader will follow the tree to send out the reading message. Since not all tags are from the given set, the algorithm needs to send a reading message when it hits a type 1 node. If a collision is observed, then the query tree algorithm is activated to resolve the collision. If a single or no response is received, then the tag is identified and the branch will be abandoned. When a type 2 node is reached, the reading message of the adequate prefix will be sent out. If there is a single response as expected, then the algorithm identifies the tag and discards the branch. Otherwise, the algorithm invokes the query tree method to resolve the collision.



The algorithm will simply move down the branch when it sees a type 3 node because it is no need to waste a reading cycle on an expected collision. These three different cases are outlined in Figure 3. Figure 4 illustrates the flowchart of the proposed algorithm. In Figure 3, we further explains the particular step of “Performing the proposed algorithm” in details.

#### Subroutine 1-Initialization

- Given a set of tag IDs:
  - Build a binary tree and mark the tags.
  - Classify all nodes in the tree based on the number of tags in the subtree rooted in the node.

#### Subroutine 2-Type 1 node handling

- Send out a reading message with the corresponding prefix
  - collision: go down the branch in the tree
  - no collision: either identify a tag or ignore the branch

#### Subroutine 3-Type 2 node handling

- Send out a reading message with the corresponding prefix
  - collision: go down the branch in the tree
  - no collision: either identify a tag or ignore the branch

#### Subroutine 4-Type 3 node handling

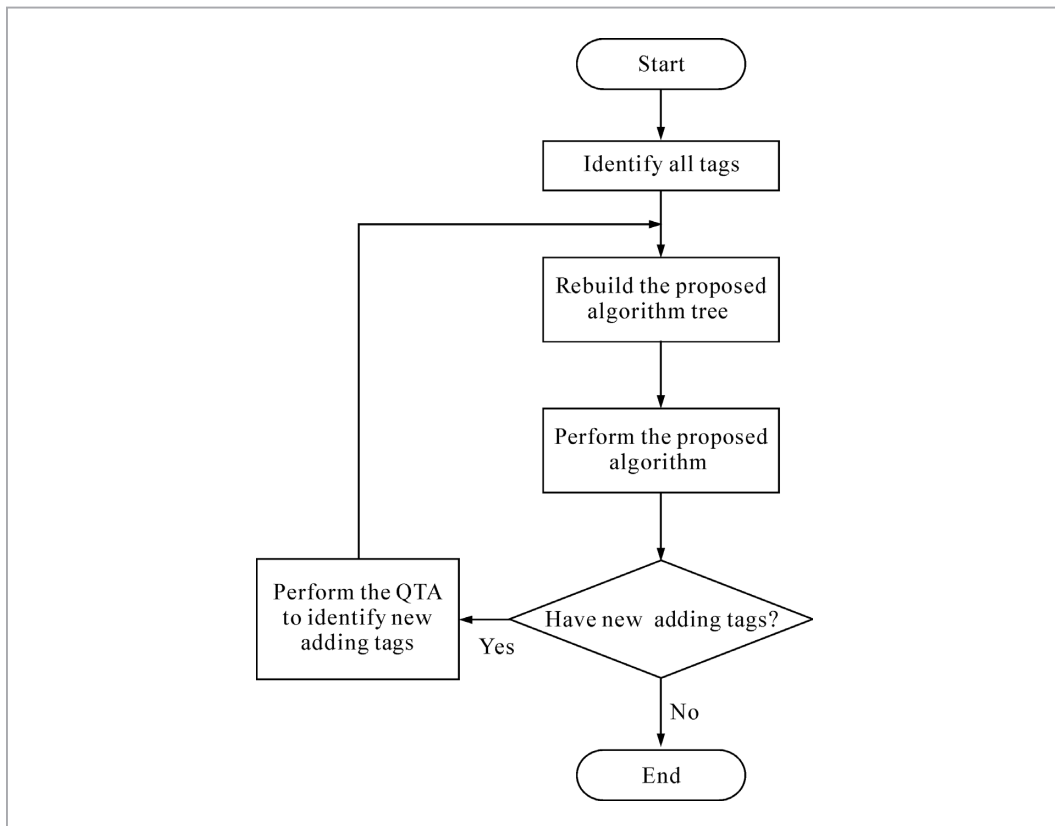
- Go down the branch without sending any reading message for the node

▲ Figure 3 The associated subroutines of the proposed algorithm

## 4. Performance Evaluations

### 4.1 The proposed algorithm description

In this section, the performance of the proposed scheme would be presented by simulations. The Table 1 shows the essential parameters used in our simulations. These parameters would be including L represents the length of the tag IDs, S represents the number of tags, P is the S to the number of leaves, and T represents the simulation times. Our scheme implemented in C language. We explored the impacts on performances during various scenarios with parameters. Specially, we consider the number of reading message required while the number of tags and various length of the tag ID change to different values. The proposed algorithm compared with the BTA and QTA in the static and the dynamic scenario. We took the average performance of 1000 times of simulations for each set of parameters and scheme combinations. We will discuss the advantages and drawbacks of these deterministic algorithms by examining the performance statistics.



▲ Figure 4 Flowchart of the proposed algorithm

▼ Table 1 The parameters in simulations

Parameters	Descriptions	Values
L	The length of the tag IDs	5、10、15
S	The number of tags	$2^1 - 2^{10}$
P	The S to the number of leaves	25%、50%、75%、100%
T	The simulation times	1000

## 4.2 Performance evaluation

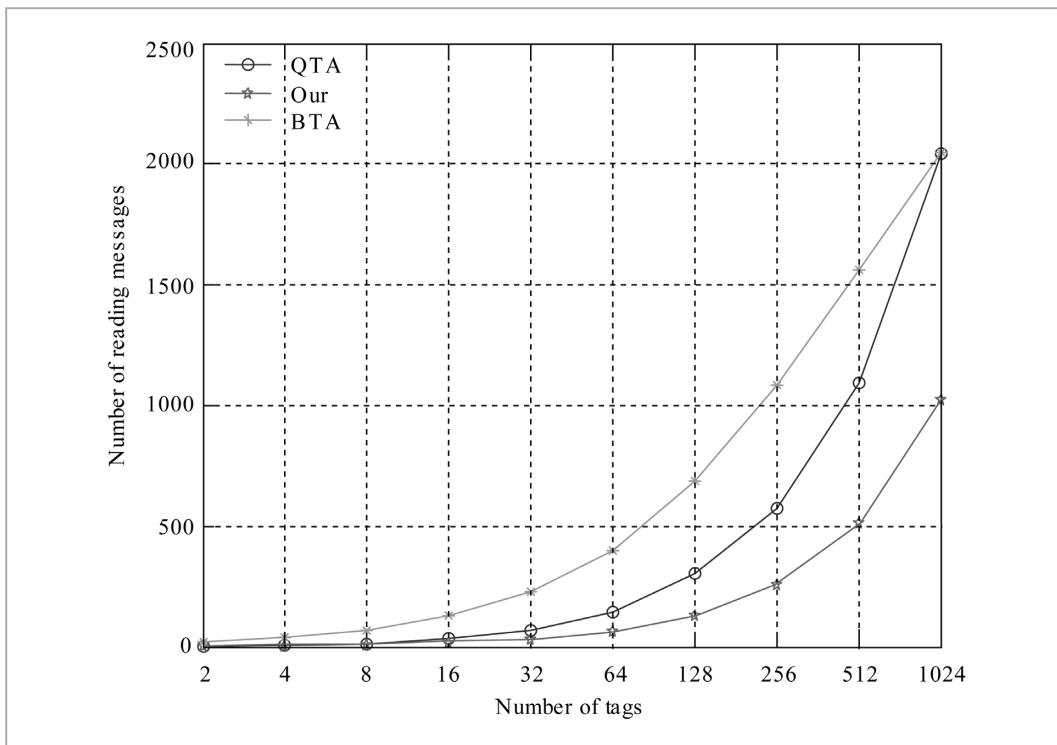
### 4.2.1 The static scenario

In this scenario, there are no adding and subtracting tags within this reading process. In Figure 5, each tag has a 10-bit ID. And, the IDs are distributed evenly over the leaves of the full binary tree. Additionally, the number of tags changed from  $2^1$  to  $2^{10}$  and observed the re-





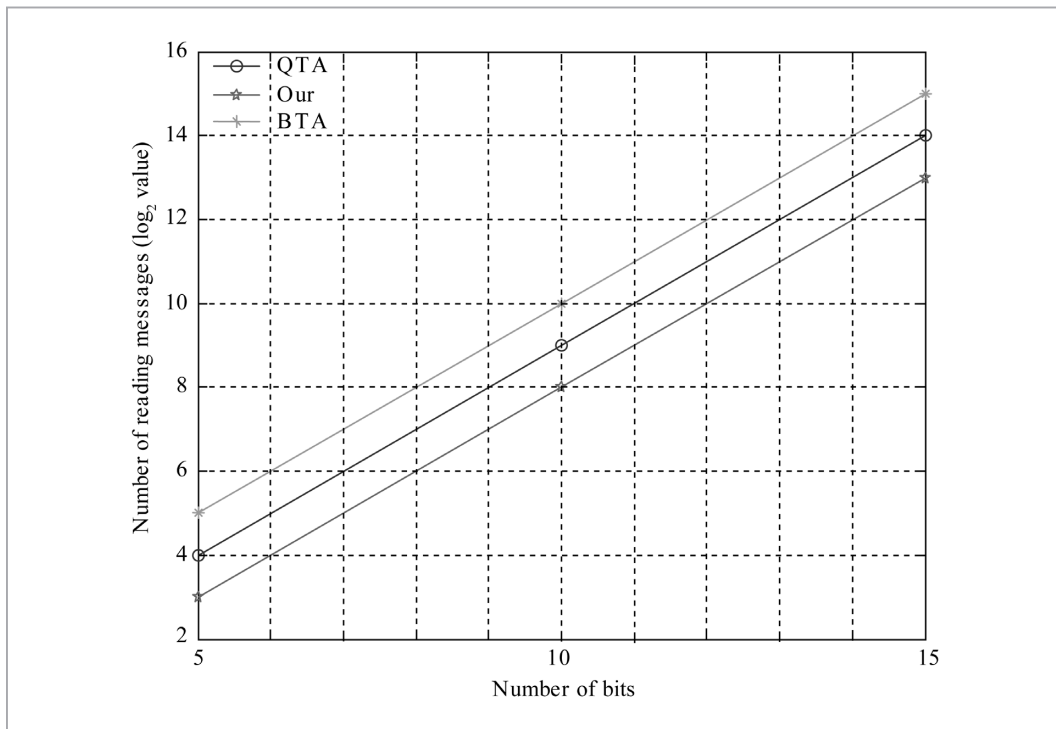
sults. The number of reading messages required in the BTA, the QTA, and the proposed algorithm would increase linearly as the number of tags increase. It should be noted that the x-axis in Figure 5 is exponential because it's representing log scale. Nevertheless, in the worst case scenario, the BTA and the QTA would generate twice as many reading messages as the proposed algorithm. When the number of tags remains small, the BTA, QTA and the proposed algorithm performances are quite similar.



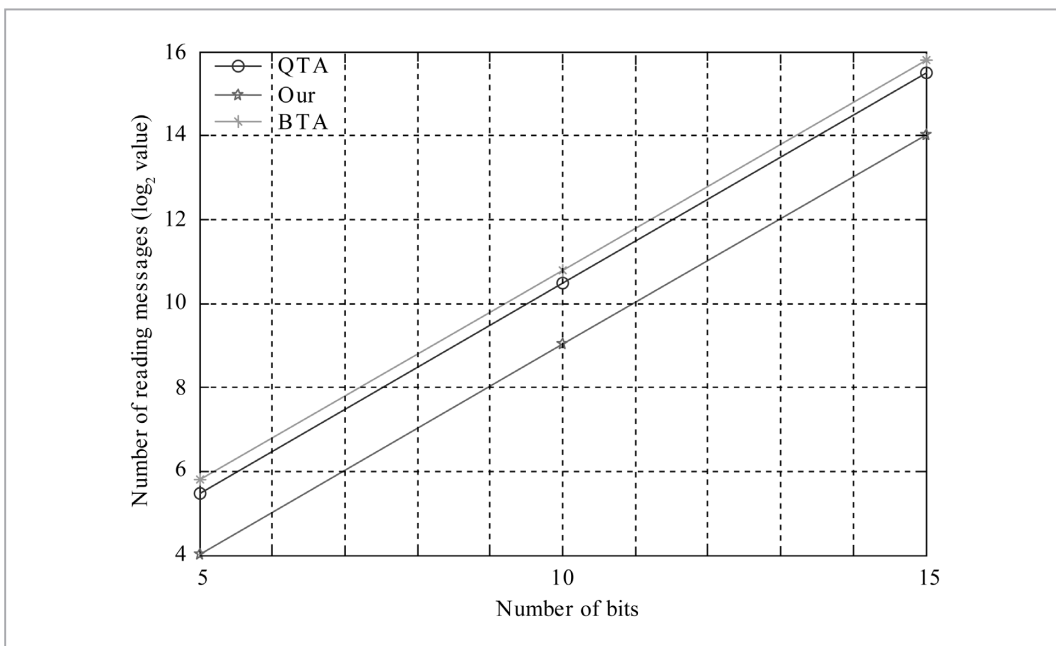
▲ Figure 5 Number of reading messages ( $L = 10$ )

When the number of tags remains small, collision impact is very minimal. As the number of tags increases, these three algorithms would be starting to react differently due to the fact that these three algorithms have different methods to handle collisions. Since all the tags are predefined, the optimal reading sequence produced by the proposed algorithm can avoid most of the collisions and would greatly outperform the other two schemes.

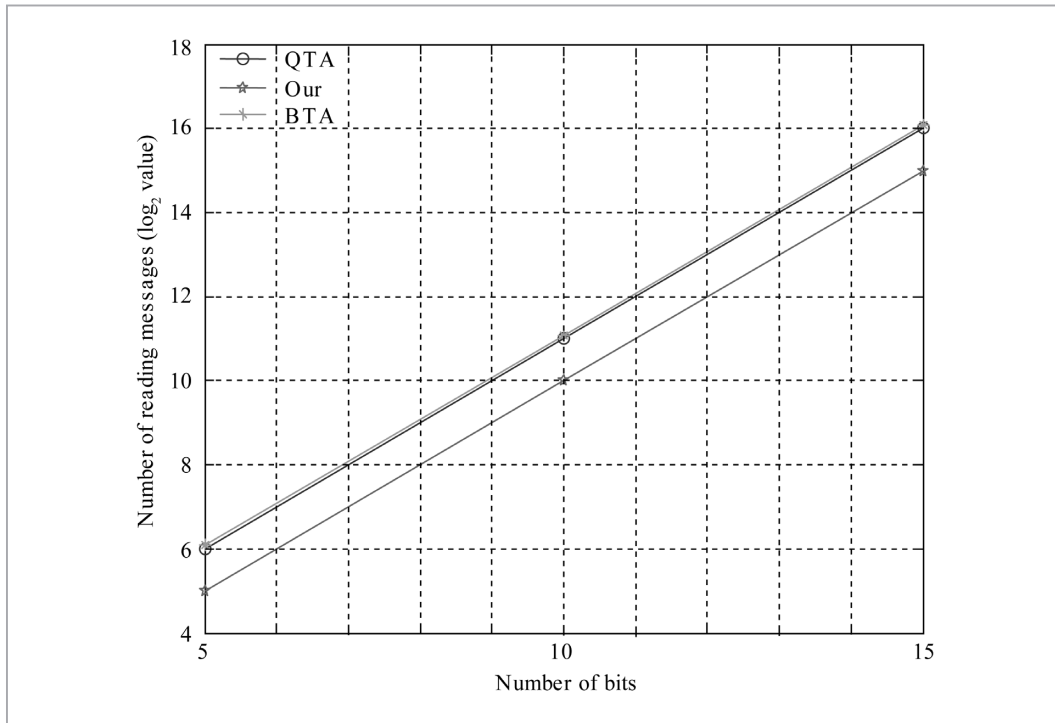
The x-axis in Figures 6, 7, 8, and 9 each represents 5, 10, and 15 bits of ID lengths. Additionally, these 4 Figures are displaying 25%, 50%, 75%, and 100% in relation with the x-axis respectively. For example, in Figure 6, when the x-axis is at 5 bits, the maximum num-



▲ Figure 6 Number of reading messages vs.  $L$  ( $P = 25\%$ )

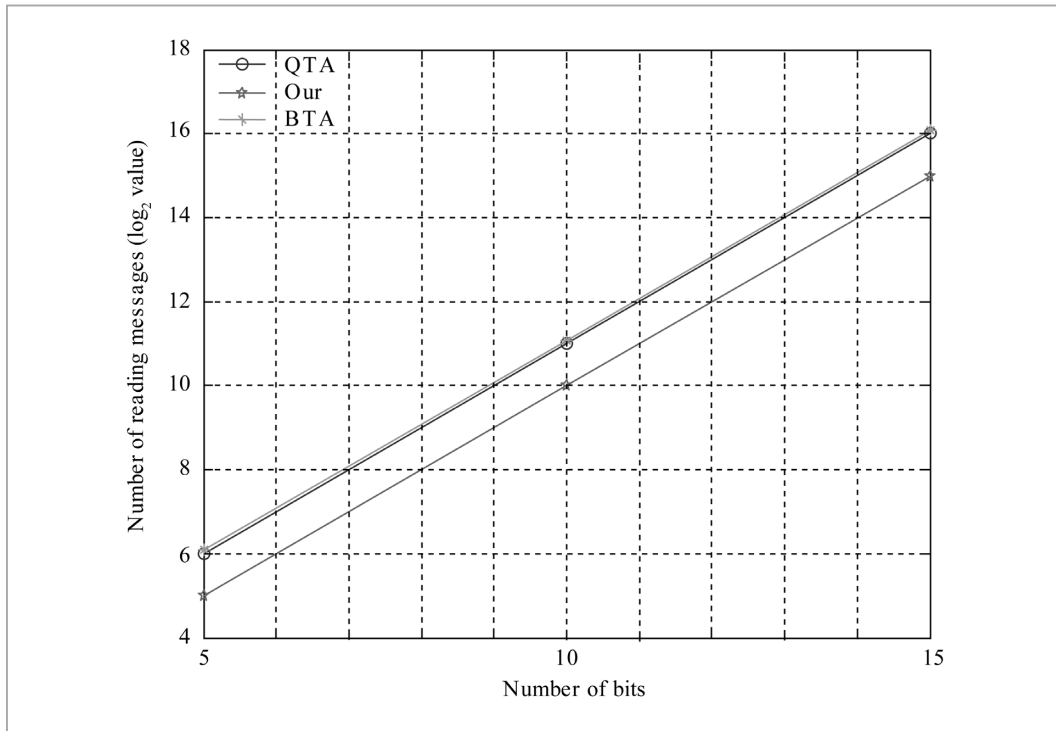


▲ Figure 7 Number of reading messages vs.  $L$  ( $P = 50\%$ )



▲ Figure 8 Number of reading messages vs.  $L$  ( $P = 75\%$ )

ber of tags is at  $2^5 = 32$ . For this particular figure, we are testing these algorithms at 25% of tags. Therefore, the maximum number of tags in Figure 6 would be  $32/4 = 8$ . In order to assure quality illustration of the simulation results; in these Figures, the y-axis is representing  $\log_2$  value. Additionally, the length of the IDs changed from 5 to 15 and range of the  $P$  ratios range from 25% to 100%. In Figure 6, the performance of the BTA, the QTA, and the proposed algorithm differences when the number of tags is at 25%. As the result show, the performance of three tree-based schemes closes when the number of tags is at 50% in Figure 7. In Figure 8 and Figure 9, the number of reading messages required can observe in the BTA and the QTA. They were performing almost equally. However, the proposed algorithm would still be more efficient when compared with the BTA and the QTA. Hence, the proposed algorithm can efficiently reduce the number of reading messages required for tag identification.

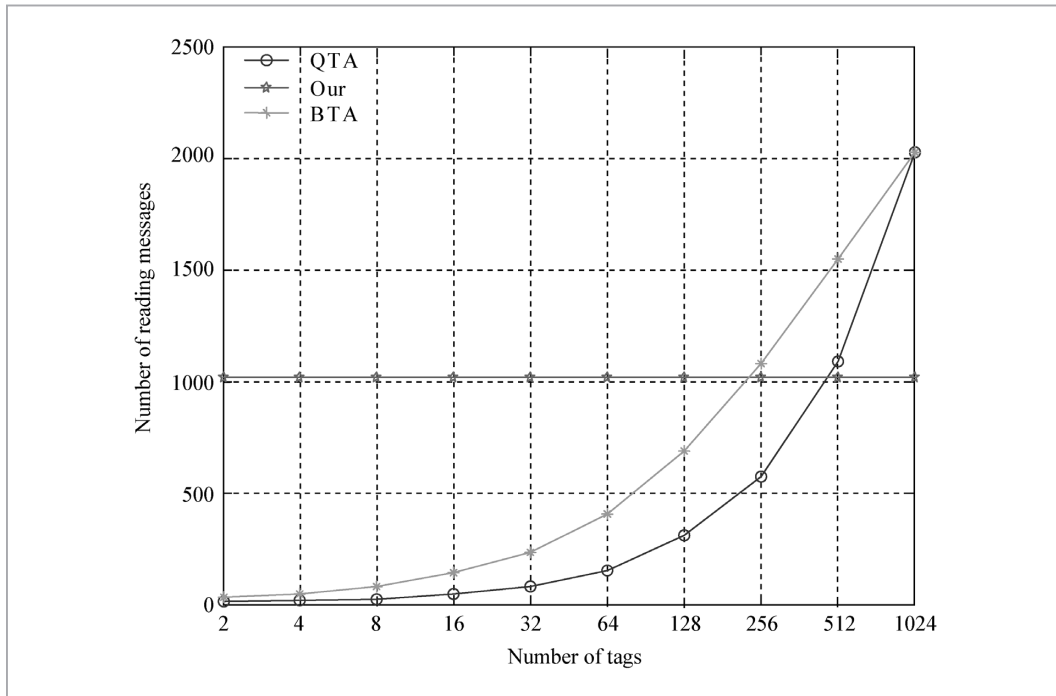


▲ Figure 9 Number of reading messages vs.  $L$  ( $P = 100\%$ )

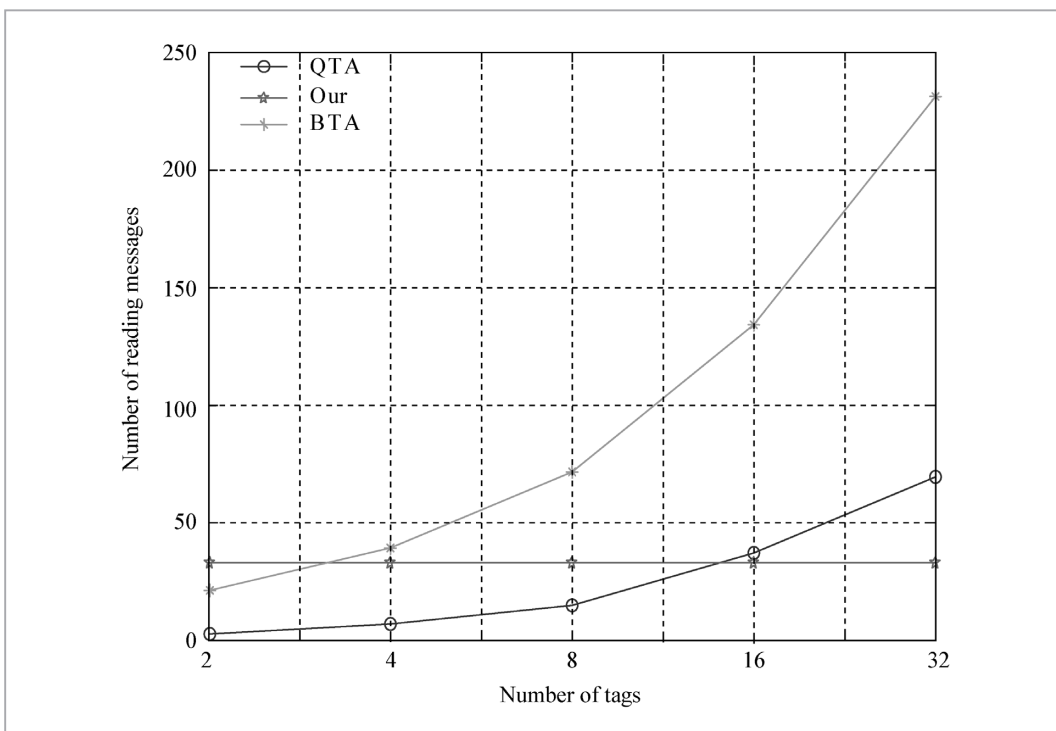
#### 4.2.2 The dynamic scenario

In this scenario, there are adding and subtracting tags within this reading process. In Figure 10 and Figure 11, the tags have 10 bit-ID and 5-bit ID respectively. All known tags are distributed over the leaves of the full binary tree. In Figure 10, the number of tags changed from  $2^1$  to  $2^{10}$  and observed results. The number of tags changed from  $2^1$  to  $2^5$  in Figure 11 as well. It can be observed that the number of reading messages required for BTA and QTA increase linearly as the number of tags increase. The proposed algorithm would react differently compared to BTA and QTA. The proposed algorithm has the ability to memorize what has been found before. The reader can utilize this memory of existing tags to save valuable time during the identifying process. Hence, the number of reading messages required does not increase linearly as the number of tags increase due to the fact that the proposed algorithm identifies all tags from a known tag set. Consequently, the number of reading messages required does not change as the number of tags increase in Figure 10 and Figure 11.

On the other hand, the proposed algorithm would spend a lot of time reading messages in Figure 10 and Figure 11. The efficiency problem would reduce as the number of tags in-



▲ Figure 10 Number of reading messages ( $L = 10$ ,  $P = 100\%$ )



▲ Figure 11 Number of reading messages ( $L = 5$ ,  $P = 100\%$ )

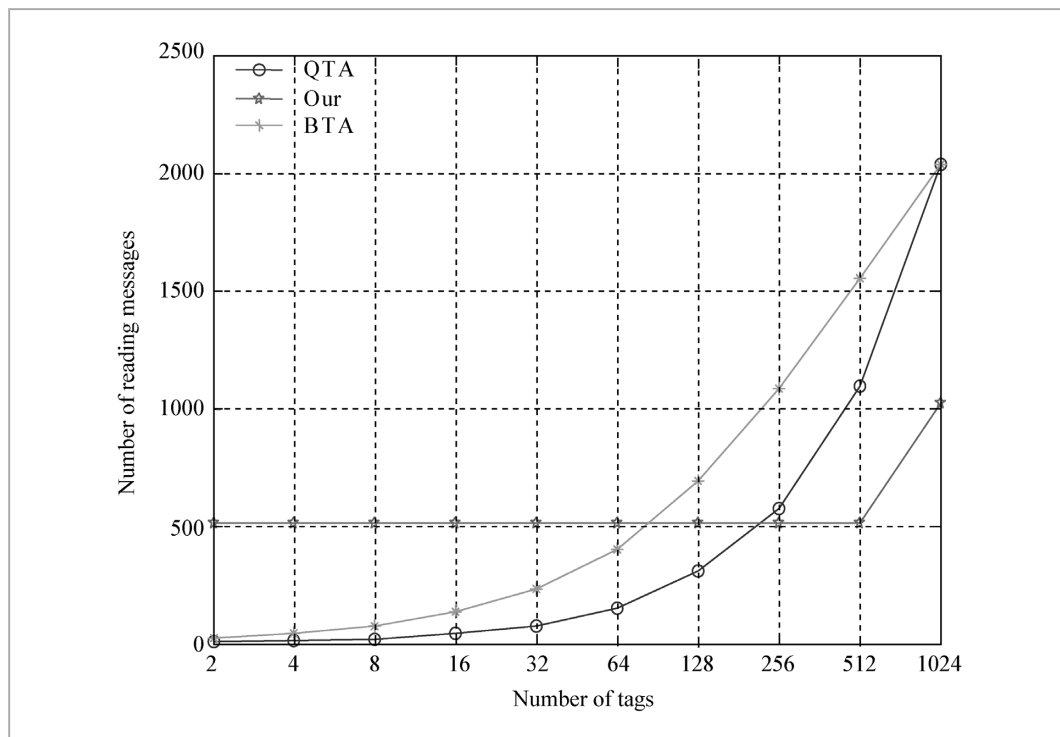


crease. It is because that whether the mobile number of tags is small or large, the number of reading messages stays the same.

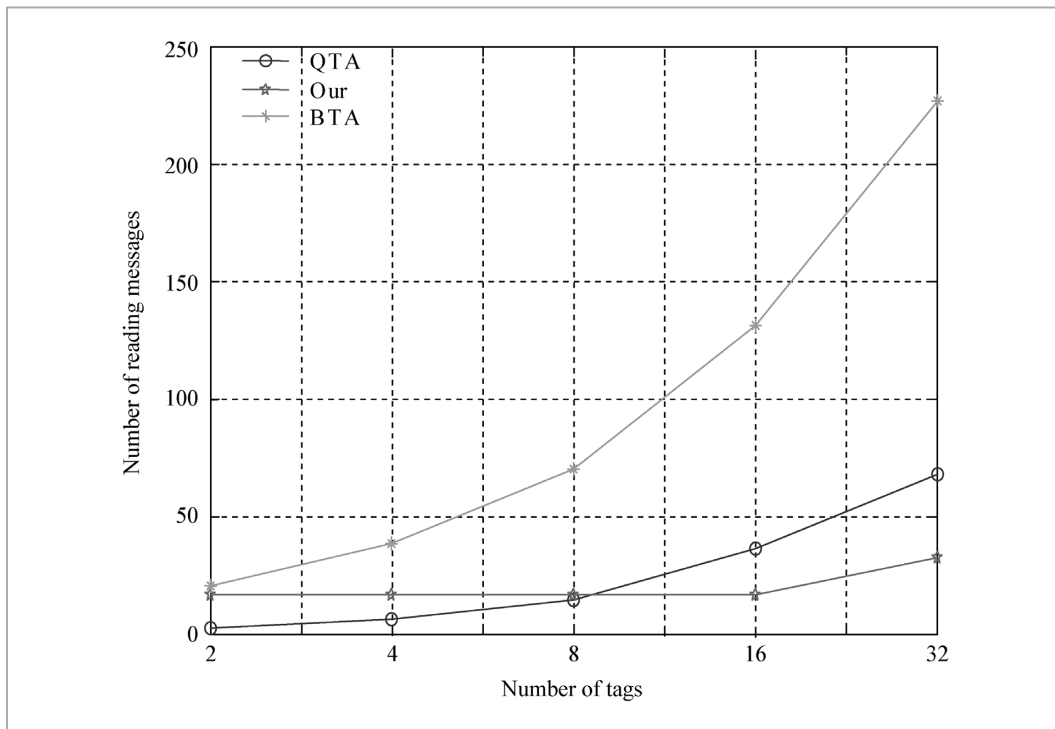
Additionally, the condition discussed when half of the known tags are distributed over the leaves of the full binary tree. The efficiency problem is minimal in Figure 12 and Figure 13. This is because the known number of tags is small. Therefore, the proposed algorithm was not required to send out too many reading messages to identify existing tags.

In Figure 12, the number of reading messages required in the proposed algorithm increases abruptly as the number of tags increase from  $2^9$  to  $2^{10}$ . It is because that only half of known tags existed. At the same time, the proposed algorithm identifies new tags with QTA anti-collision algorithm and memorizes these tags as the number of tags increase from  $2^9$  to  $2^{10}$ . Hence, the number of reading messages required in the proposed algorithm increases abruptly. Similarly, the condition occurs in Figure 13. As a result, the proposed algorithm has the ability to dynamically add new tags for tag identification. The proposed algorithm is a good method for RFID system, but there is an efficiency problem for messages reading.

Nevertheless, the proposed algorithm can apply in the static scenario such as storehouses. It is a good environment to apply this method because the merchandises of storehouses



▲ Figure 12 Number of reading messages ( $L = 10$ ,  $P = 50\%$ )



▲ Figure 13 Number of reading messages ( $L = 5$ ,  $P = 50\%$ )

usually remain at the same quantity. Additionally, we did not compare the proposed algorithm with the cut-through algorithm due to the fact that they share common advantages.

## 5. Conclusions

In this paper, we address two tag anti-collision schemes and propose a novel scheme for the reader to quickly identify tags in the RFID system. It has been proven in the past that the way to optimize message reading efficiency is to recognize all possible tag IDs in advance. However, to the best of our knowledge, the proposed scheme is the first attempt in considering dynamic participation of tags which are not in the predefined set. It reflects the real situation in the RFID application.

According to the simulation results, the proposed algorithm exhibits short tag identification time when compared with the prevailing deterministic algorithms such as BTA and QTA. Best case scenario, we observed that the proposed algorithm would spend 50% of the time required in the QTA method and 30% of the time required in the BTA method. This im-



provement not only accelerates the reading speed but also protects the on-board power in the reader.

## References

- Birari, S.M. and Iyer, S. (2005), "Mitigating the reader collision problem in RFID networks with mobile readers," in *Proceedings of the 2005 13th IEEE International Conference on Networks*, Kuala Lumpur, Malaysia, 16-18.
- Cha, J. and Kim, J. (2005), "Novel anti-collision algorithms for fast object identification in RFID system," in *Proceedings of the 2005 11th International Conference on Parallel and Distributed System*, Fukuoka, Japan, 63-67.
- Choi, J.H., Lee, D., and Lee, H. (2006), "Bi-slotted tree based anti-collision protocols for fast tag identification in RFID systems," *IEEE Communications Letters*, 10(12), 861-863.
- Choi, J.H., Lee, D., and Lee, H. (2007a), "Query tree-based reservation for efficient RFID tag anti-collision," *IEEE Communications Letters*, 11(1), 85-87.
- Choi, J.H., Lee, D., Jeon, H., Cha, J., and Lee, H. (2007b), "Enhanced binary search with time-divided responses for efficient RFID tag anti-collision," in *Proceedings of the 2007 IEEE International Conference on Communications, Glasgow, Scotland*, 3853-3858.
- EPCglobal (2005), "EPCTM radio-frequency identification protocols class-1 generation-2 UHF RFID protocol for communications at 860MHz-960MHz version 1.0.9," EPCglobal.
- Finkenzeller, K. (2003), *RFID handbook: Radio-frequency identification fundamentals and applications* (2nd ed.), John Wiley and Sons Ltd.
- Ho, J., Engels, D.W., and Sarma, S.E. (2006), "HiQ: A hierarchical Q-learning algorithm to solve the reader collision problem," in *Proceedings of the International Symposium on Applications and the Internet Workshops (SAINTW'06)*, Arizona, USA, 23-27.
- ISO/IEC (2003), "Identification technology automatic identification and data capture techniques—radio frequency identification for item management air interface—part 6: parameters for air interface communications at 860–960MHz," (ISO/IEC FDIS 18000-6), ISO/IEC.
- Kim, J., Lee, W., Kim, E., Kim, D., and Suh, K. (2007a), "Optimized transmission power





- control of interrogators for collision arbitration in UHF RFID systems," *IEEE Communications Letters*, 11(1), 22-24.
- Kim, S.H., Park, P., Kim, E., Kim, D., and Suh, K. (2007b), "An efficient tree-based tag anti-collision protocol for RFID systems," *IEEE Communications Letters*, 11(5), 449-451.
- Kim, S., Lee, S., and An, S. (2006), "Reader collision avoidance mechanism in ubiquitous sensor and RFID networks," in *Proceedings of the 1st International workshop on Wireless network testbeds, experimental evaluation & characterization (WiNTECH'06)*, NY, USA, 101-102.
- Law, C., Lee, K., and Siu, K.Y. (2000), "Efficient memoryless protocol for tag identification," in *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Boston, MA, USA, 75-84.
- Myung, J., Lee, W., and Srivastava, J. (2006), "Adaptive binary splitting for efficient RFID tag anti-collision," *IEEE Communications Letters*, 10(3), 144-146.
- Park, J., Chung, M.Y., and Lee, T.J. (2007), "Identification of RFID tags in framed-slotted ALOHA with robust estimation and binary selection," *IEEE Communications Letters*, 11(5), 452-454.
- Vogt, H. (2002), "Multiple object identification with passive RFID tags," in *Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics*, Yasmine Hammamet, Tunisia, 651-656.
- Waldrop, J., Engels, D.W., and Sarma, S.E. (2003a), "Colorwave: A Mac for RFID reader networks," in *Proceedings of the Wireless Communication and Networking Conference (WCNC'2003)*, New Orleans, LA, USA, 1701-1704.
- Waldrop, J., Engels, D.W., and Sarma, S.E. (2003b), "Colorwave: An anticollision algorithm for the reader collision problem," in *Proceedings of the 2003 IEEE International Conference on Communications, Anchorage Alaska, USA*, 1206-1210.
- Wang, T.P. (2006), "Enhanced binary search with cut-through operation for anti-collision in RFID systems," *IEEE Communications Letters*, 10(4), 236-238.

