This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Enhanced Binary Search with Time-Divided Responses for Efficient RFID Tag Anti-Collision

Ji Hwan Choi, Dongwook Lee, Hyoungsuk Jeon, Jongsub Cha, and Hyuckjae Lee
School of Engineering
Information and Communications University (ICU)
119 Munjiro, Yuseong-gu, Daejeon, 305-732, Korea
E-mail: {jihwanchoi, dwook, hschun, s1009, and hjlee}@icu.ac.kr

***Abstract*— RFID is a generic term for technologies which use RF waves to identify, track, or categorize any object. One of the research areas in RFID systems is a tag anti-collision protocol; how to reduce identification time with a given number of tags in the field of an RFID reader. There are two types of tag anti-collision protocols for RFID systems: tree based algorithms and slotted aloha based algorithms. Since the tree based tag anti-collision protocols achieve 100% read rate, we consider how to improve the performances of the tree based RFID tag anti-collision protocols. This paper proposes bi-slotted tree based tag anti-collision protocols, bi-slotted query tree algorithm (BSQTA) and bi-slotted collision tracking tree algorithm (BSCTTA), which reduce both prefix overhead and iteration overhead by time divided responses depending on whether the collided bit is '0' or '1'. According to the simulation results, the bi-slotted tree based RFID tag anti-collision protocols require less time consumption for tag identification than the present tag anti-collision protocols.**

***Index Terms*–Anti-collision protocols, tree based algorithms, tag collisions, RFID**

I. INTRODUCTION

Radio Frequency IDentification (RFID) is an emerging technology that guarantees to advance modern industrial practices in object identification and tracking, asset management, and inventory control [11]. Recently, several identification systems such as barcodes and smart cards are incorporated for automatic identification and data collection. However, these systems have several limits in read rate, visibility, and contact. RFID systems are a matter of grave concern because they provide fast and reliable communication without requiring physical sight or touching between readers and tags.

One of the areas of research is the speed with which a given number of tags in the field of RFID readers can be identified. For fast tag identification, anti-collision protocols, which reduce collisions and identify tags irrespective of occurring collisions, are required [1], [4]. There are two types of collisions: reader collisions and tag collisions. Reader collisions indicate that when neighboring readers inquire a tag concurrently, so the tag cannot respond its ID to the inquiries of the readers. These collision problem can be easily solved by detecting collisions and communicating with other readers. Tag collisions occur when multi tags try to respond to a reader simultaneously and cause the reader to identify no tag. For low-cost passive RFID tags, there is nothing to do except response to the inquiry of the reader. Thus, tag anti-collision protocols are necessary for improving the cognitive faculty of RFID systems.

Tag anti-collision protocols are classified into two categories: deterministic methods and probabilistic methods [4], [6]. The first one is tree based protocols such as bit-arbitration algorithm [9], splitting tree algorithm [1], [4] (memory based protocols), tree working algorithm [5], query tree algorithm [7]-[9], collision tracking tree algorithm [9], and scanning-based sequential searching algorithm [10] (memoryless based protocols). These algorithms split colliding tags into two subgroups until all tags are identified. The second one is based on ALOHA like slotted ALOHA, and frame slotted ALOHA. The frame slotted ALOHA is the basis of extended protocols such as FS-ALOHA [11], STAC [12], and bit-slot algorithm [13]. The ALOHA based protocols are designed to reduce the probability of occurring tag collisions how tags respond at the different time.

In this paper, we discuss how to improve the readability of low-cost passive RFID systems and its identification speed using the present memoryless tag anti-collision protocols. Bi-slotted tree based RFID tag anti-collision protocols, bi-slotted query tree algorithm (BSQTA) and bi-slotted collision tracking tree algorithm (BSCTTA), are presented as substitutes of the present RFID tag anti-collision protocols. Generally, the tree based algorithms send a prefix twice except the last bit in the same tree depth. Focusing on this characteristic, both prefix overhead and iteration overhead are reduced by the time-divided responses depending on whether the collided bit is '0' or '1'. We demonstrate via simulation results that the proposed bi-slotted tree based RFID tag anti-collision protocols achieve considerably better performance than the tree based RFID tag anti-collision protocols. Besides, the proposed algorithms require less time consumption for tag identification than the conventional schemes.

The rest of this paper is organized as follows. In Section II, the previous tree based anti-collision protocols are reviewed. Section III describes the proposed protocol, bi-slotted tree based RFID tag anti-collision protocols. The simulation environment and the performance analysis are explicated in Section IV. The conclusion is discussed in Section V.

1-4244-0353-7/07/$25.00 ©2007 IEEE

II. THE PRESENT TREE BASED RFID TAG ANTI-COLLISION PROTOCOLS

For reliable RFID systems, read rate is a very important factor. The read rate indicates the percentage of the succeeding frequency over cognitive trials. While 100% read rate cannot be achieved by the slotted ALOHA based anti-collision protocols with small amount of tags, it can be fulfilled by the tree based anti-collision protocols. Thus, we first overview several tree based RFID tag anti-collision protocols in the following subsections.

A. Binary Tree Working Algorithm

A reader chooses '0' or '1' for the initiative. If the reader makes a choice, the identification process should keep the way of choice order when the tree splits at a node. Then the binary tree working algorithm (BTWA) is operated as follows:

Step 1.The reader transmits k-length prefix.

Step 2.Tags send $(k+1)_{th}$ bit if the first k bits of tag IDs are the same as the prefix.

Step 3.If the received bits collide, the extended prefix attached '0' or '1' to the prefix is retransmitted by the reader. If they do not collide, the received bit is attached to the prefix for the next prefix. If there is no response, the branch is ignored. Also, if the last bits collide, the reader assumes there are two tags because of the uniqueness of the tag IDs.

Step 4.The reader repeats the procedure until all the tags are identified [5].

B. Query Tree Algorithm

The query tree algorithm (QTA) is based on BTWA. The difference between QTA and BTWA is as follows. A reader transmits the k-length prefix. Then tags send from $(k+1)_{th}$ bit to the end bit of tag IDs if the first k bits of tag IDs are the same as the prefix. Also, if the received tag IDs collide, the extended prefix attached '0' or '1' to the prefix is retransmitted. Furthermore, if there is no collision, the reader identifies one of the tags [7]-[9].

C. Collision Tracking Tree Algorithm

The collision tracking tree algorithm (CTTA) is based on QTA except that this scheme uses collision tracking. In CTTA, the tags send their IDs from $(k+1)_{th}$ bit to the end bit if the prefix is the same as tags' first k bits. Then the reader transmits a signal to stop sending IDs from tags if there is a collision. It looks similar to QTA, but the difference is that the collision tracking tree algorithm constructs the next prefix with the bits received before collision and reduces the waste of time caused by collisions occurred at the received bits. In other words, the next prefix is '$\textit{the former prefix } k \textit{ bits } + \textit{ the received } n-1 \textit{ bits } + \textit{ 0 or 1}$' when the collision occurs at n_{th} bit in the received sequences of the reader [9].

D. Scanning based Sequential Searching Algorithm

This algorithm finds the positions of collided bits first with scanning all tag IDs. A reader sends inquiring bits to receive tag IDs simultaneously. After tags send their whole length of IDs, the reader can learn the location of the collided bits with received sequences. Then the reader makes 2^{n-1} prefixes from LSB ('0') to MSB ('1') if n-bit collisions occur. With this prefix set, the reader identifies all the tags sequentially. The reason is that if there is only one collision at any position, the reader assumes there are two tags because of the uniqueness of the tag IDs [10].

III. BI-SLOTTED TREE BASED RFID TAG ANTI-COLLISION PROTOCOLS

The present schemes are not efficient enough because of its processing time. Thus, we propose new efficient tag anti-collision protocols for passive RFID systems.

A. The crux of the proposed algorithm

The aspect of bi-slotted tree based RFID tag anti-collision protocols, bi-slotted query tree algorithm (BSQTA) as shown in Fig.1 and bi-slotted collision tracking tree algorithm (BSCTTA) as shown in Fig.2, is that the present tree based algorithms generally send two 'n length inquiring bits', which have the same first $n-1$ bits and the different last bit. Focusing on this characteristic, the RFID systems can reduce the identification time with the procedure as follows:

1) REQUEST : A reader sends $n-1$ length inquiring bits (prefix) to tags.

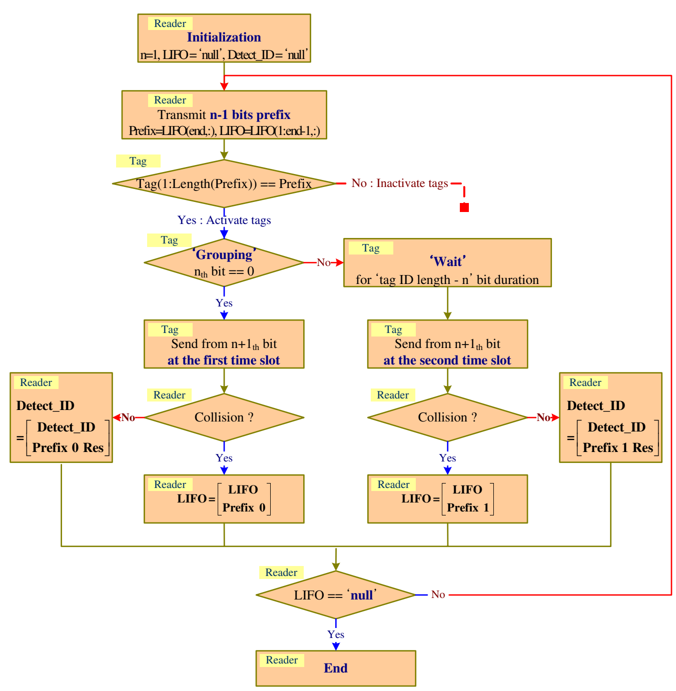


Fig. 1. The Flow Chart of Bi-Slotted Query Tree Algorithm

2) GROUPING : Tags in the field of the reader respond their tag IDs to the reader if the inquiring bits are the same as the first $n - 1$ bits of tag IDs.
   - When the tags respond their IDs to the reader, they choose one of two time slots depending on whether n_{th} bit is '0' (first slot) or '1' (second slot). Thus, the time slot indicates the value of n_{th} bit.
   - BSQTA: tags send their IDs from $(n + 1)_{th}$ bit to the end bit.
   - BSCTTA: tags send their IDs from $(n + 1)_{th}$ bit to the time that ACK signal, which is sent from the reader when a collision occur, is received.
3) DECISION : Depending on whether collision have occurred or not, the reader decides on proceeding procedure with the following conditions.
   - If there is a collision, the reader saves a new prefix at the last input first output (LIFO).
     – BSQTA: the connection of $n - 1$ length inquiring bits and the indication of the chosen slot
     – BSCTTA: the connection of $n - 1$ length inquiring bits, the indication of the chosen slot, and the bits received before collisions occur
   - If a collision occurs at the last bit in tag IDs, the reader assumes there are two tags because of the uniqueness of the tag IDs.
   - If there is no collision, the reader identifies a tag in the multi tags.
4) Do these steps until the LIFO is '*null*'.

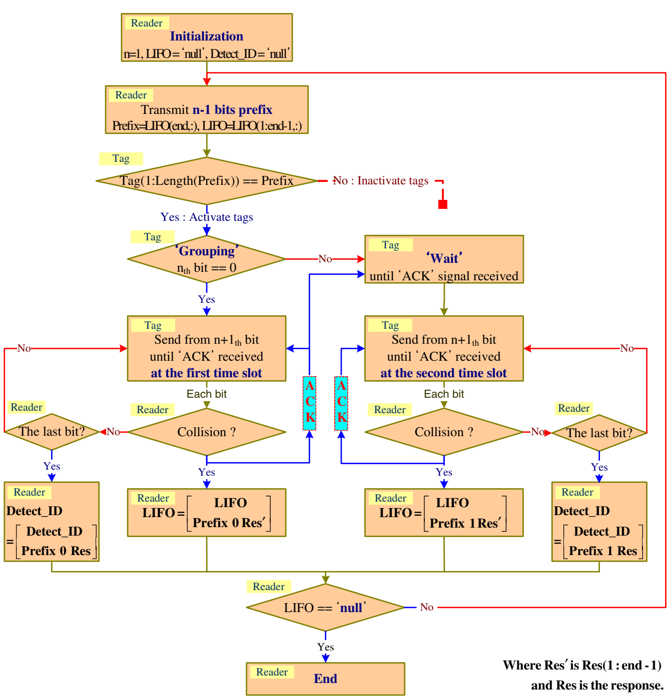


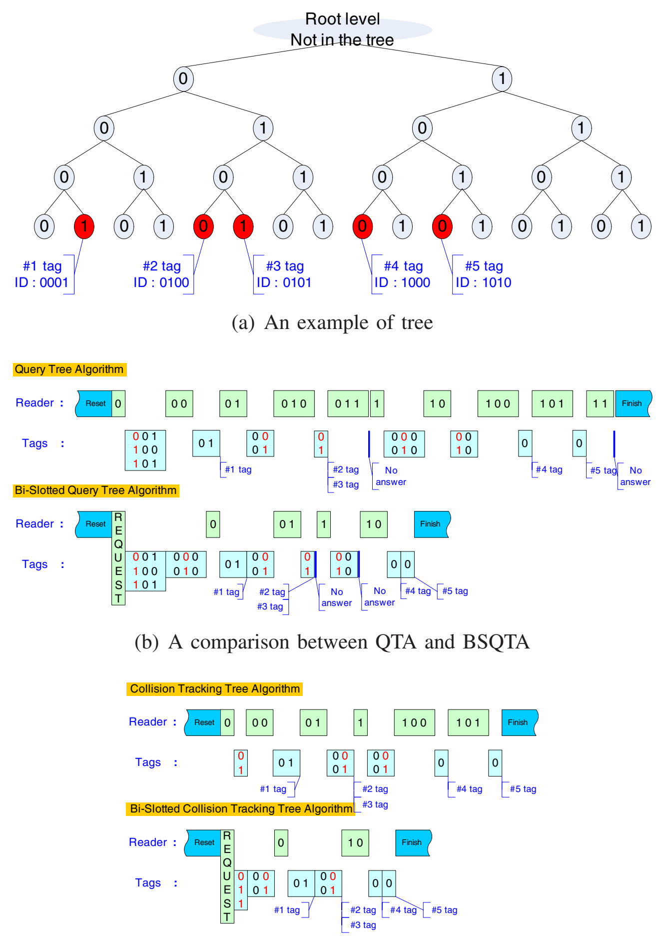
Fig. 2. The Flow Chart of Bi-Slotted Collision Tracking Tree Algorithm

With BSQTA and BSCTTA, we can reduce the average required prefix overhead to '*half* − *1bit*' of the prefix overhead in the present tree based RFID tag anti-collision protocols, QTA and CTTA, without any increase in the tag response overhead.

B. An example

To facilitate the understanding of the proposed algorithms, an example is given as follows.

Let's assume that there are five tags, which the tag IDs are '0001', '0100', '0101', '1000', and '1010', in the field of an RFID reader as shown in Fig.3 (a). Then we observe the total required bits for one-tag identification to compare QTA with BSQTA and to compare CTTA with BSCTTA. Here, the tree searching order is from LSB ('0') to MSB ('1'). Since we use LIFO, we don't search from the root level every time, but search from the closest node at the tree. In Fig.3 (b) and (c), the stream of time is from left to right. Also, the slot sizes depend on the contained number of bits. The bits on the reader side are the prefixes, and the bits on the tag side are the corresponding responses. Finally, the length from left to right shows the time delay for tag identification.



(a) An example of tree

(b) A comparison between QTA and BSQTA

(c) A comparison between CTTA and BSCTTA

Fig. 3. An example : comparison of tag identification

According to Fig.3 (b), for identifying all five tags, QTA requires 22 bits for the prefixes and 15 bits for the responses, so 37 bits in total are needed for tag identification. On the other hand, BSQTA requires 7 bits for the prefixes and 15 bits for the responses, so 22 bits in total are needed for tag identification. Thus, BSQTA reduces the prefix overhead more than QTA.

Fig.3 (c) shows a comparison between CTTA and BSCTTA. For all five-tag identification, CTTA requires 12 bits for the prefixes and 9 bits for the responses, so 21 bits in total are needed for tag identification. On the contrary, BSCTTA requires 4 bits for the prefixes and 9 bits for the responses, so 13 bits in total are needed for tag identification. Hence, BSCTTA reduces the prefix overhead more than CTTA.

Consequently, RFID systems when either BSQTA or BSCTTA is applied achieve faster tag identification than RFID systems when either QTA or CTTA is applied.

IV. SIMULATIONS AND RESULTS

This simulation is focused on the performance comparison between the present tree based RFID tag anti-collision protocols and the proposed protocols.

A. Simulation environment

The simulation condition is as follows. There is only one reader. In the field of the reader, the number of tags increases from 2 to 65536. Both tag-to-reader data rate and reader-to-tag data rate are chosen as 80k bps. The reason is that the middle speed in EPC Class 1 Gen. 2 proposed by EPCglobal to ISO/ICE 18000-6 C is equal to the chosen data rate [2], [3]. 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 tag IDs are randomly generated. Moreover, query tree algorithm (QTA) and collision tracking tree algorithm (CTTA) are taken for comparing with the proposed algorithms, bi-slotted query tree algorithm (BSQTA) and bi-slotted collision tracking tree algorithm (BSCTTA).

The observation of the simulations has three important points. The first point of observation is the average required bits for one-tag identification with 96-bit tag ID length determined in EPC Class 1 Gen. 2. The second point is the average required iterations for one-tag identification with 96-bit tag ID length. One iteration is determined by one request and one of the possible responding states: collision, no collision, and no response. The third point is the tendency of the prefix overhead and the response overhead to compare the performance between the present protocols and the proposed protocols in detail. The prefix overhead and the response overhead are related to the amount of power consumption at the reader and the tags.

B. Results

We start with the comparison between BSQTA and QTA. Fig.4 shows the average required prefix and response for one-tag identification. According to the figure, BSQTA reduces the average required prefix overhead to '$half - 1bit$' of the prefix overhead in QTA without any increase in the average required response bits for one-tag identification. Thus, BSQTA requires less average required bits for one-tag identification than QTA as shown in Fig.5. The performance gap between BSQTA and QTA in the average required bits for one-tag identification increases with the number of tags because the average required prefix overhead linearly increases with the number of tags as shown in Fig.4 and Fig.5. In other words, the average required prefix overhead for one-tag identification depends on the number of collisions, which are similar to the number of tags. The average required iterations for one-tag identification are another important factor in evaluating the RFID system performance because iteration overhead affected by the processor in both the reader and the tags gives time delay to the tag identification. Fig.7 indicates that BSQTA needs half of the required iterations in QTA. Accordingly, BSQTA achieves somewhat better performance than QTA in both the average required bits and the average required iteration for one-tag identification.

Next, we compare BSCTTA with CTTA. Similarly,

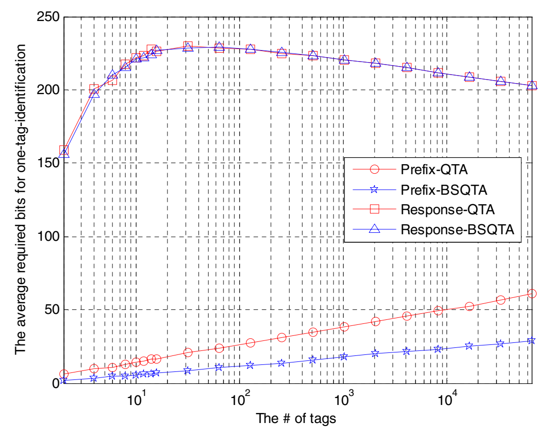


Fig. 4. The average required prefix and response bits for one-tag identification

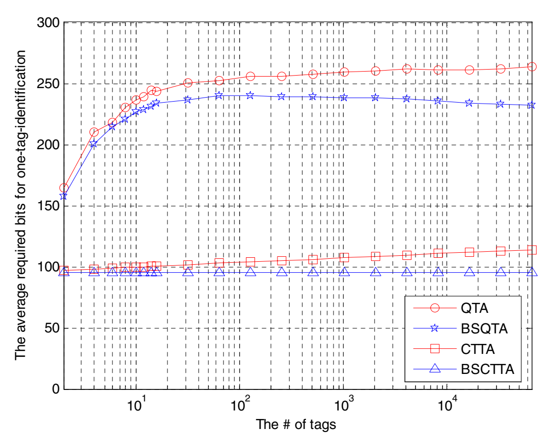


Fig. 5. The average required bits for one-tag identification

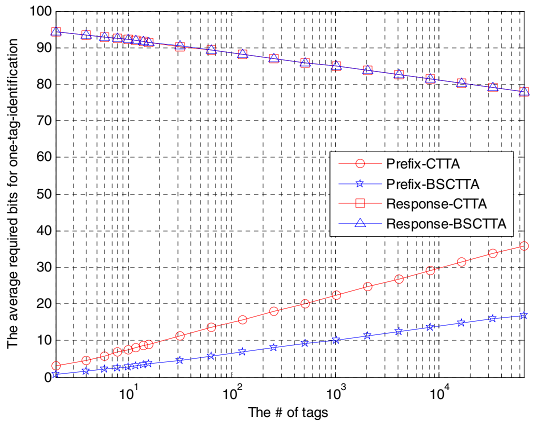


Fig. 6. The average required prefix and response bits for one-tag identification

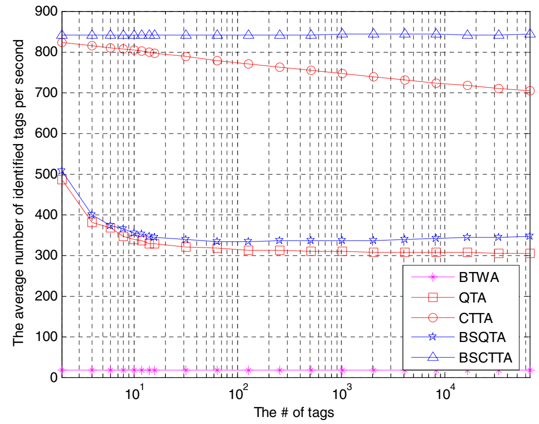


Fig. 8. The average number of identified tags per second

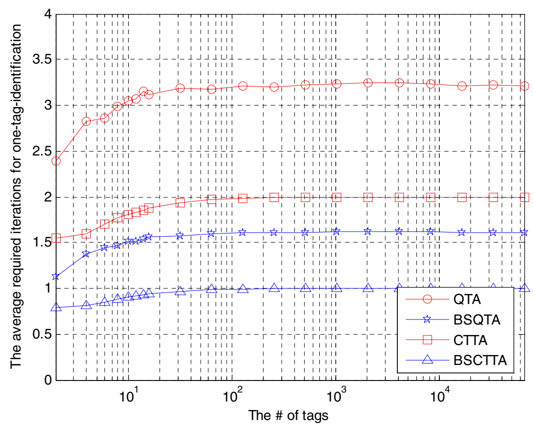


Fig. 7. The average required iterations for one-tag identification

BSCTTA gets better performance than CTTA in the average required bits for one-tag identification. Also, the average required prefix overhead for one-tag identification in BSCTTA is '$half - 1bit$' of that in CTTA without any performance degradation on the average required response bits for one-tag identification as shown in Fig.6. Thus, BSCTTA requires less average required bits for one-tag identification because of the reduced prefix overhead as shown in Fig.5. Furthermore, the average required iterations for one-tag identification in BSCTTA are half of those in CTTA as shown in Fig.7.

Finally, Fig.8 shows the average number of identified tags per second in each algorithm. In conformity with the simulations, BSQTA achieves faster identification than QTA, and BSCTTA also achieves faster identification than CTTA. Consequently, the proposed protocols, BSQTA and BSCTTA, accomplish faster tag identification than the present tree based tag anti-collision protocols, QTA and CTTA.

V. CONCLUSION

RFID systems are coming the most promising technologies used for contactless object identification. For fast tag identification, anti-collision protocols, which reduce the frequency of collisions occurred and identify tags independent of occurring collisions, are required. Moreover, tag anti-collision protocols are more important than reader anti-collision protocols because of the incompetence of tags in low-cost passive RFID systems.

This paper proposes bi-slotted query tree algorithm (BSQTA) and bi-slotted collision tracking tree algorithm (BSCTTA), which reduce the prefix overhead for fast tag identification. The crux of these protocols is that an RFID reader transmits one '$n - 1$ length inquiring bits' at a node which is related to a collided n_{th} bit in the received tag IDs instead of two 'n length inquiring bits', which have the same first $n - 1$ bits and the different last bit. Then the corresponding responses consist of two time slots depending on whether n_{th} bit is '0' (first slot) or '1' (second slot). This technique reduces the average required prefix overhead to '$half - 1bit$' of the prefix overhead in the present tree based RFID tag anti-collision protocols, QTA and CTTA, without any increase in the tag response overhead. According to the simulation results, RFID systems when either BSQTA or BSCTTA is applied achieve faster tag identification than RFID systems when either QTA or CTTA is applied.

In conclusion, the proposed tree based RFID tag anti-collision protocols, BSQTA and BSCTTA, are more efficient than the present tree based RFID tag anti-collision protocols, QTA and CTTA. Thus, the proposed algorithms allow the RFID systems to operate faster than the present RFID systems.

VI. ACKNOWLEDGMENT

This work is supported by the "Development of Sensor tag and Sensor node technology for RFID/USN" project of ETRI through IT Leading R&D Support Programs of MIC (Ministry of Information and Communication), Korea.

REFERENCES

[1] J. Myung, W. Lee, and J. Srivastava, "Adaptive binary splitting for efficient RFID tag anti-collision," *IEEE Communications Letters*, Vol. 10, Iss. 3, pp 144-146, Mar. 2006.

[2] "Information technology automatic identification and data capture techniques - radio frequency identification for item management air interface - part 6: parameters for air interface communications at 860-960 MHz," *ISO/IEC FDIS 18000-6*, Nov. 2003.
[3] "EPC^{TM} radio-frequency identification protocols class-1 generation-2 UHF RFID protocol for communications at 860MHz-960MHz Ver. 1.0.9," *EPCglobal*, Jan. 2005.
[4] J. Myung and W. Lee, "An adaptive memoryless tag anti-collision protocls for RFID networks," *IEEE INFORCOM'05*, Poster Session, Mar. 2005.
[5] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Information Theory*, vol. 25, pp. 505-515, Sept. 1979.
[6] J.Zhai and G. Wang, "An anti-collision algorithm using two-functioned estimation for RFID tags," *Proc. ICCSA'05, LNCS.3483*, pp. 702-711, May 2005.
[7] D. R. Hush and C. Wood, "Analysis of Tree Algorithms for RFID Arbitration," *Proc. IEEE Inter. Symp. on Information Theory*, pp 107-127, 1998.
[8] "860MHz-930MHz Class I Radio Frequency Identification Tag Radio Frequency and Logical Communication Interface Specification Candidate Recommendation, Version 1.0.1," *Auto-ID Center*, Nov. 14, 2002.
[9] F. Zhou, D. Jin, C. Huang, and M. Hao, "Optimize the Power Consumption of Passive Electronic Tags for Anti-collision Schemes," *Proc. The 5th Inter. conf. on ASIC*, Vol. 2, pp.1213-1217, Oct. 21-24, 2003.
[10] L. Liu, Z. Xie, J. Xi, and S. Lai, "An Improved Anti-collision Algorithm in RFID System," *2005 2nd Inter. Conf. on Mobile Technology, Applications and Systems*, pp 1-5, Nov. 15-17, 2005.
[11] H. Vogt, "Efficient Object Identification with Passive RFID Tags," *Proc. Inter. Conf. on Pervasive Computing, LNCS.2414*, Springer-Verlag. pp 98-113, Aug. 2002.
[12] "13.56MHz ISM Band Class 1 Radio Frequency Identification Tag Interface Specification: Candidate Recommendation, Ver. 1.0.0," *Auto-ID Center* May 2003.
[13] C. Kim, K. Park, H. Kim, and S. Kim, "An Efficient Stochastic Anti-collision Algorithm using Bit-Slot Mechanism," *Proc. Inter. Conf. on Parallel and Distributed Processing Techniques and Applications(PDPTA)*, pp 652-656, Jun. 2004.