Fast Wireless Anti-collision Algorithm in Ubiquitous ID System

Ho-Seung Choi, Jae-Ryong Cha, and Jae-Hyun Kim

School of Electrical and Computer Engineering
AJOU University
Suwon, Korea
Email: lastjoin@ajou.ac.kr, builder@ajou.ac.kr, and jkim@ajou.ac.kr

***Abstract*— This paper proposes and analyzes the fast wireless anti-collision algorithms(Modified Bit-by-bit Binary-Tree algorithm(MBBT) and Enhanced Bit-by-bit Binary-Tree algorithm(EBBT)) for Ubiquitous ID system. This paper mathematically compares the performance of the proposed algorithms with that of Binary Search algorithm(BS), Slotted Binary-Tree algorithm(SBT) using time slot, and Bit-by-bit Binary-Tree algorithm(BBT). We also validated analytical results using OPNET simulation. According to the analysis, comparing MBBT with BBT which is the best among existing algorithms, the performance of MBBT is about 5% higher when the number of the tags is 20, and 100% higher for 200 tags. Also, comparing EBBT with MBBT, the performance of EBBT is about 355% higher when the number of tags is 20, and 145% higher for 200 tags.**

I. Introduction

In a world of ubiquitous computing, the object identification is most useful for applications such as asset tracking(e.g. libraries, animals), automated inventory, stock-keeping, toll collecting, and similar tasks where physical objects are involved. The RFID(Radio Frequency Identification) systems are a simple form of ubiquitous sensor networks that are used to identify physical objects. In this paper, we call RFID system as Ubiquitous ID(*u*-ID) system. Instead of sensing environmental conditions, the *u*-ID system identifies the unique tags' ID or detailed information saved in them attached to objects. Passive *u*-ID systems generally consist of three components - a reader, passive tags, and a controller. A reader interrogates tags for their ID or detailed information through an RF communication link, and contains internal storage, processing power, and so on. Tags get processing power through RF communication link from the reader and use this energy to power up any on-tag computations and to communicate to the reader. A reader in *u*-ID system broadcasts the request message to the tags. Upon receiving the message, all tags send their response back to the reader. If only one tag responds, the reader receives just one response. But if there are more than one tag response, their responses will collide on the RF communication channel, and their response cannot be received by the reader[1]. The problem of resolving this collision is referred to as the Anti-Collision Problem, and the ability to resolve this problem is crucial in *u*-ID system performance. In *u*-ID system, important measures of performance include the time required to identify the tags, and the power consumed by the tags. Therefore, if the data from the tags are small, we can reduce the time to identify the tags and the power consumed by the tags. This paper mathematically compares the performance of the proposed algorithm with that of Binary Search algorithm(BS), Slotted Binary-Tree algorithm(SBT), and Bit-by-bit Binary Tree algorithm(BBT). We also validate analytical results using OPNET simulation.

II. Existing binary anti-collision algorithms

A. Binary Search algorithm(BS) [2]

BS resolves the collision by gradually reducing collided bits in tag ID. When a collision is occurred, the reader knows the position of collided bit. If there are tags whose first collided bit is 1, they do not respond to the reader's next request. But tags whose first collided bit is 0 transfer their ID to the reader's next request. By repeating this procedure, the reader can identify all the tags. More details are in [2]. Assuming that there are n tags, the number of iterations of BS(I_{BS}) is

$$I_{BS} = \frac{\log(n)}{\log(2)} + 1. \tag{1}$$

B. Slotted Binary Tree algorithm(SBT) [3]

According to this algorithm when a collision occurs, in slot i, all tags that are not involved in the collision wait until the collision is resolved. The tags involved in the collision split randomly into two groups, by(for instance) each selecting 0 or 1. The tags in the first group, those that selected 0, retransmit in slot i+1 while those that selected 1 wait until all tags that selected 0 successfully transmit their ID. If slot i+1 is either idle or contains a successful transmission, the tags of the second group, those that selected 1, retransmit in slot i+2. If slot i+1 contains another collision, the procedure is repeated[3]. Assuming that there are n tags, the number of iterations of SBT(I_{SBT}) is

$$I_{SBT} = 1 + \sum_{k=2}^{n} \binom{n}{k} \frac{2(k-1)(-1)^k}{\left[1 - p^k - (1-p)^k\right]}. \tag{2}$$

0-7803-8521-7/04/$20.00 (C) 2004 IEEE

TABLE I
USED TAGS' ID

Tag1	0001
Tag2	0010
Tag3	1010
Tag4	1011

C. Bit-by-bit Binary Tree algorithm(BBT) [4],[5]

In BBT, if the reader requests for ID bit to the tags, all the tags transfer 0 or 1 out of their ID bit. If not colliding, the reader saves the received bit in the memory before it requests next bit. But if colliding, the reader selects one group out of two groups according to the algorithm, then requests next bit. The reader repeats this procedure until all the bits of an ID are received. Assuming that there are *n* tags and the length of tag ID is *j* bits, then the number of iterations of BBT(I_{BBT}) is

$$I_{BBT} = n \times j. \tag{3}$$

Fig.1 shows the process to identify four tags in Table I. In Fig.1, the number of iterations to identify four tags is 16(4×4 bits).

III. THE PROPOSED ALGORITHMS

A. Modified Bit-by-bit Binary-Tree algorithm(MBBT)

In BBT, because the reader always requests all the bits of tags' ID, it takes much time to identify all the tags. To resolve this problem, we proposed MBBT. MBBT procedure is as follows. If the reader requests k^{th} bit to the tags, the tags transfer k^{th} bit.(In this algorithm, the initial value of k is 1). If the reader receives k^{th} bit without collision, it saves k^{th} bit in the memory. But if colliding, the reader saves k^{th} bit as 0 in the memory. The reader then makes all the tags whose k^{th} bit is 1 inactive. The tags in inactive state do not temporarily transfer their ID bits until one tag is identified. If there are two tags whose ID except for the last bit is identical, the last bit transferred from tags collides definitely. In this case, we do not need to request next bit in MBBT. Therefore, we can identify two tags simultaneously.

Fig.2 shows the process to identify four tags in Table I using MBBT. In this example two tags, whose IDs are respectively '1010' and '1011', have identical ID except the last bit. When the reader requests these tags' last bit, the collision occurs. However the reader can identify two tags simultaneously without further request. Therefore, in Fig.2, the number of iterations to identify four tags is 12 (3×4 bits).

B. Enhanced Bit-by-bit Binary-Tree algorithm(EBBT)

In MBBT, if IDs of used tags are sequential, the time to identify tags is decreased. Otherwise, MBBT shows same performance with BBT. Therefore, we propose EBBT to overcome weakness of MBBT and to enhance the performance of MBBT. EBBT procedure is as follows. First of all, the reader requests all the bits of the ID. When the reader receives k^{th} bit of tags' ID and if all the bits are 0(1), the reader saves

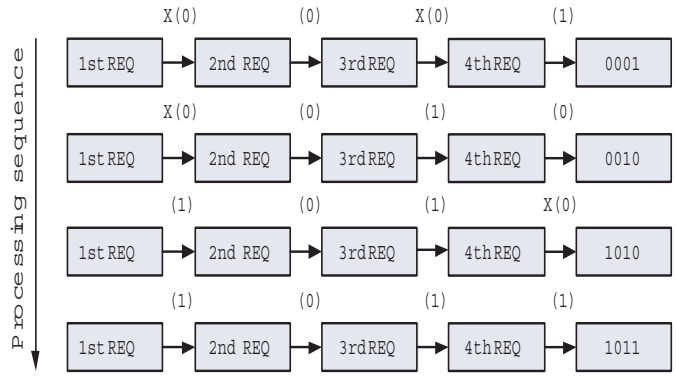

Fig. 1. Bit-by-bit binary-tree algorithm

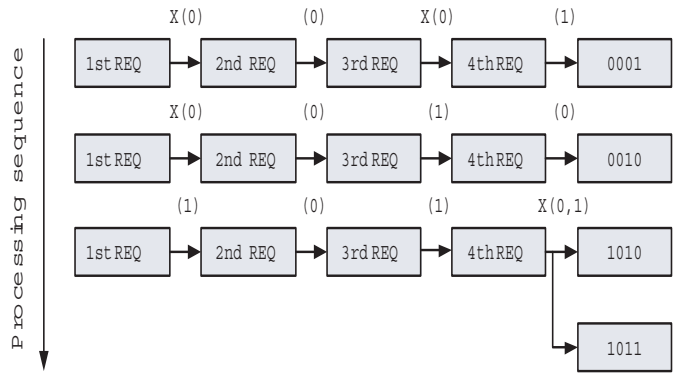

Fig. 2. Proposed MBBT algorithm

k^{th} bit as 0(1). But if the collision occurs, the reader saves k^{th} bit as X in the memory. After the reader receives all the bits of the tags, the reader knows which bit is collided. The reader sequentially re-requests only the collided bit to the tags by the method of bit-by-bit. As a result, the number of iterations of EBBT is less than that of MBBT. And if the tags' ID is sequential, the number of collision bits reduces. Therefore, the performance of EBBT is better than that of MBBT.

IV. PERFORMANCE ANALYSIS

In this section, we analyze the performance of proposed algorithms from two points of view. One is the number of iterations and the other is total transferred bits from the tags.

A. The number of iterations

1) Modified Bit-by-bit Binary-Tree algorithm(MBBT): We assume that the total number of tags is 2*n*, which is sequential, and the length of ID is 36bits[5]. The number of iterations of MBBT(I_{MBBT}) is as follows.

a) When the number of used tags is even(2m): When the number of used tags(2*m*) is less than or equal to 50%($0 < 2m \leq n$) out of total tags(2*n*), the number of iterations of

MBBT(I_{MBBT}) is

$$I_{MBBT} = \sum_{k=0}^{m} \frac{\binom{n}{k}\binom{n-k}{2(m-k)} 2^{2(m-k)}}{\binom{2n}{2m}} \times (2m-k). \tag{4}$$

When the number of used tags(2*m*) is more than 50%($n < 2m \leq 2n$) out of total tags(2*n*), the number of iterations of MBBT(I_{MBBT}) is

$$I_{MBBT} = \sum_{k=2m-n}^{m} \frac{\binom{n}{k}\binom{n-k}{k-2m+n}}{\binom{2n}{2m}} \times 2^{2(m-k)} \times (2m-k). \tag{5}$$

b) When the number of used tags is odd(2m-1): When the number of used tags(2*m*-1) is less than 50%($0 < 2m - 1 < n$) out of total tags(2*n*), the number of iterations of MBBT(I_{MBBT}) is

$$I_{MBBT} = \sum_{k=0}^{m-1} \frac{\binom{n}{k}\binom{n-k}{2m-2k-1}}{\binom{2n}{2m-1}} \times 2^{(2m-2k-1)} \times (2m-1-k). \tag{6}$$

When the number of used tags(2*m*-1) is more than 50%($n < 2m - 1 < 2n$) out of total tags(2*n*), the number of iterations of MBBT(I_{MBBT}) is

$$I_{MBBT} = \sum_{k=2m-n-1}^{m-1} \frac{\binom{n}{k}\binom{n-k}{k-2m+n+1}}{\binom{2n}{2m-1}} \times 2^{(2m-2k-1)} \times (2m-1-k). \tag{7}$$

2) Enhanced Bit-by-bit Binary-Tree algorithm(EBBT): We assume that the number of the used tags which do not have sequential ID is *m*, and the length of ID is 36 bits. The number of iterations of EBBT(I_{EBBT}) is

$$\begin{aligned} I_{EBBT} &= \left(\frac{m}{2}+1\right) \times 36 + \frac{m}{4} \times 35 + \frac{m}{8} \times 34 + \cdots \\ &\quad + \frac{m}{2^{k_{\max}}}(37-k_{\max}) \\ &\quad + \left(\frac{m}{2^{k_{\max}}} - 1\right)(36-k_{\max}) \\ &= \sum_{k=1}^{\log_2 m} \frac{m(37-k)}{2^k} + 36 \\ &\quad + \left(\frac{m}{2^{k_{\max}}} - 1\right)(36-k_{\max}) \end{aligned} \tag{8}$$

where [*] is a maximum integer less than or equal to *, and $k_{\max} = [\log_2 m]$.

We assume that arbitrary *m* tags are used out of *n* tags whose ID is sequential, and the length of ID is 36 bits. The number of iterations of EBBT(I_{EBBT}) is driven by

$$\begin{aligned} I_{EBBT} &= \left(\frac{2^r}{n}+m+1\right)(r+1) + \frac{m\left(1-\frac{2^r}{n}\right)}{2} r \\ &\quad + \frac{m\left(1-\frac{2^r}{n}\right)}{4}(r-1) + \cdots \\ &\quad + \frac{m\left(1-\frac{2^r}{n}\right)}{2^{k_{\max}}}(r+1-k_{\max}) \\ &\quad + \left(\frac{m\left(1-\frac{2^r}{n}\right)}{2^{k_{\max}}} - 1\right)(r-k_{\max}) \\ &= \sum_{k=1}^{\log_2 m\left(1-\frac{2^r}{n}\right)} \frac{m\left(1-\frac{2^r}{n}\right)}{2^k}(r+1-k) \\ &\quad + \left(\frac{2^r}{n} \times m + 1\right)(r+1) \\ &\quad + \left(\frac{m\left(1-\frac{2^r}{n}\right)}{2^{k_{\max}}} - 1\right)(r-k_{\max}) \end{aligned} \tag{9}$$

where $k_{\max} = [\log_2 m\left(1-\frac{2^r}{n}\right)]$, $2^r < n \leq 2^{(r+1)}$, and $0 \leq r \leq 35$(*r* is an integer).

B. Total transferred bits from the tags

Let I be the number of iterations for each algorithm and B_I be the transferred bits from the tags in each iteration, then the number of total transferred bits(B_{total}) from the tags is

$$B_{total} = I \cdot B_I. \tag{10}$$

In EBBT, when the reader initially requests all ID bits from the tags, the tags send all ID bits to the reader. So, the number of total transferred bits from the tags should be added by 35(the length of ID-1).

V. Numerical and Simulation Results

Fig.3 shows the total transferred bits from the tags for the number of tags in each algorithm by mathematical analysis. In proposed algorithms, all tags transfer one bit whenever the reader requests ID to the tags. But all other algorithms except for BBT transfer all the bits of tags' ID. Therefore, the proposed algorithms reduce the time to identify tag ID and the energy consumed by the tag because the number of transferred bits from the tags is smaller compared to the existing algorithms. In Fig.3, to identify 100 tags, BS, SBT, and BBT transfer 22491, 9482, and 3600 bits respectively. For the same scenario, MBBT and EBBT only transfer 2704 and 770 bits respectively. Therefore, we found that proposed algorithms have higher performance compared to the existing algorithms.

Fig.4 shows the number of iterations for the number of tags in BBT, MBBT and EBBT. In Fig.4, lines represent analytic results and symbols mean simulation results using OPNET. Analytic results are very closed to the simulation results. Comparing MBBT with BBT, we found that the

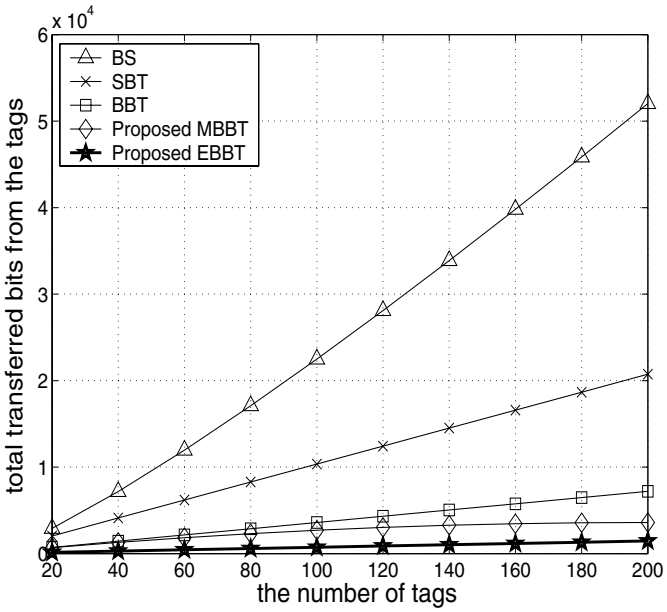

Fig. 3. Total transferred bits from the tags vs. the number of the tags

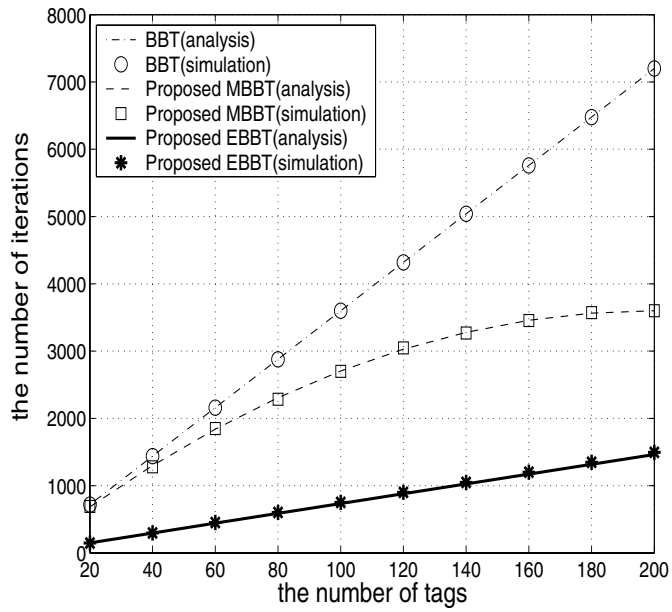

Fig. 4. The number of iterations vs. the number of tags

performance of MBBT is 100% higher when the number of tags is 200. Also, comparing EBBT with MBBT, we found that the performance of EBBT is about 145% higher when the number of tags is 200.

VI. Conclusion

We proposed and analyzed the fast wireless anti-collision algorithms(MBBT and EBBT) for *u*-ID system. We mathematically compared the performance of the proposed algorithms with BS, SBT, and BBT. We also validated analytical results using OPNET simulation. MBBT shows better performance than existing algorithms as the number of used tags increases. The energy consumption is also smaller than existing algorithms since the number of bits transferred from the tags is smaller. Furthermore, the performance of EBBT is better than that of MBBT. On conclusion, if we apply the proposed algorithms to the *u*-ID system, it will contribute to improve the performance of the *u*-ID system because the reader can identify more tags within short time and with low energy consumption.

Acknowledgment

This research is partially supported by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Science and Technology(MOST) 21st Century Frontier R&D Program in Korea.

References

[1] H. Vogt, Efficient Object Identification with Passive RFID tags, In International Conference on Pervasive Computing, Zurich, 2002.

[2] K. Finkenzeller, RFID Handbook: *Radio-Frequency Identification Fundamentals and applications*. John Wiley and Sons Ltd, 1999.

[3] J. L. Massey, Collision resolution algorithms and random-access communications, Univ. California, Los Angeles, Tech. Rep. UCLAENG -8016, Apr., 1980.

[4] M. Jaoccurt, A. Ehrsam, U. Gehrig, Contactless Identification Device With Anticollision Algorithm, IEEE Computer Society CSCC99, Conference on Circuits, Systems, Computers and Communications, Jul. 4-8 Athens, 1999.

[5] Auto-ID Center, *Draft Protocol Specification for a Class 0 Radio Frequency Identification tag*., 2003.