Novel Bit-by-bit Binary Tree Algorithm in Ubiquitous ID System1)

Ho-Seung Choi, Jae-Ryong Cha, and Jae-Hyun Kim
School of Electrical and computer Engineering
AJOU University
Sowon, Korea
{lastjoin, builder, and jkim}@ajou.ac.kr

Abstract—We propose and analyze the fast wireless anti-collision algorithm(Enhanced Bit-by-bit Binary-Tree algorithm(EBBT)) for Ubiquitous ID system. We mathematically compares the performance of the proposed algorithm 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 EBBT with BBT which is the best of existing algorithms, the performance of EBBT is about 288% higher when the number among the tags is 20, and 380% higher for 200 tags.

Keywords : Anti-collision algorithm, Ubiquitous ID system, and Binary Tree algorithm

1. 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 on which 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 it 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 the 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.

2. Existing binary anti-collision algorithms

2.1. Binary Search algorithm(BS) [2]

BS resolves the collision by gradually reducing collided bits in a 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

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

2.2. 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

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

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

In BBT, if the reader requests for the 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 procedure is as follows. If the reader requests k^{th} bit from 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. 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

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

3. Proposed EBBT algorithms

In BBT, because the reader always requests all the bits of tags' ID, it takes long time to identify all the tags. To resolve this problem, we proposed EBBT. 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 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-request only 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 BBT. And if the tags' ID is sequential, the number of collision bits reduces. Therefore, the performance of EBBT is much better than that of BBT.

4. Performance analysis

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

4.1. The number of iterations

4.1.1. When the used tags' ID is not sequential.

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 shown in (4).

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

where [*] is a maximum integer less than or equal to *.

4.1.2. When the used tags' ID is sequential.

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 given by (5).

$$\begin{aligned} I_{EBBT} &= \left(\frac{2^r}{n}\times m+1\right)(r+1)+\frac{m\left(1-\frac{2^r}{n}\right)}{2}r+\frac{m\left(1-\frac{2^r}{n}\right)}{4}(r-1)+\cdots \\ &\quad+\frac{m\left(1-\frac{2^r}{n}\right)}{2^{k_{\max}}}(r+1-k_{\max})+\left(\frac{m\left(1-\frac{2^r}{n}\right)}{2^{k_{\max}}}-1\right)(r-k_{\max}) \\ &= \sum_{k=1}^{\log_2 m\left(1-\frac{2^r}{n}\right)}\frac{m\left(1-\frac{2^r}{n}\right)}{2^k}(r+1-k)+\left(\frac{2^r}{n}\times m+1\right)(r+1)+\left(\frac{m\left(1-\frac{2^r}{n}\right)}{2^{k_{\max}}}-1\right)(r-k_{\max}) \end{aligned} \tag{5}$$

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

4.2. Total transferred bit from the tags

Let I be the iterations of 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

$$B_{total} = I \cdot B_I. \tag{7}$$

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

5. Numerical and simulation result

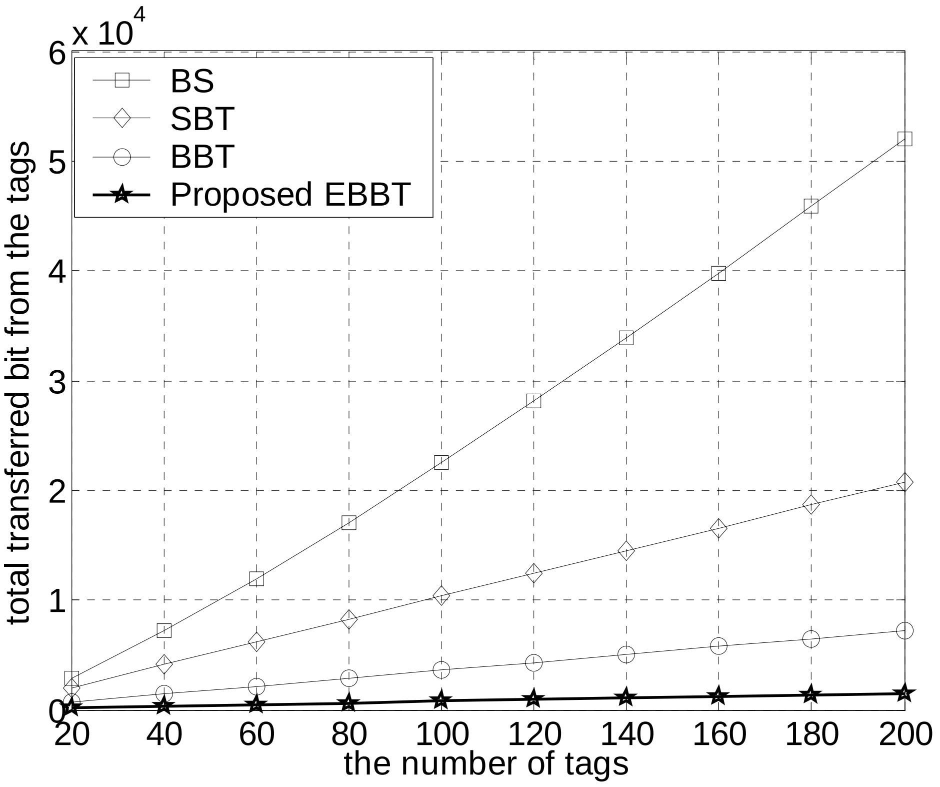

Figure 1. total transferred bits from the tags vs. the number of the tags

Fig.1 shows the total transferred bits from the tags for the number of tags in each algorithm and the analytic results. In proposed algorithm, all tags transfer one bit whenever the reader requests ID from the tags except for the first transmission. But all other algorithms except for BBT transfer all the bits of tags' ID. So, the proposed algorithm reduces 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.1, to identify 100 tags, BS, SBT, and BBT transfer 22491, 9482, and 3600 bits respectively. For the same scenario, EBBT only transfers 770 bits. Therefore, we found that proposed algorithm has very high performance compared to the existing algorithms.

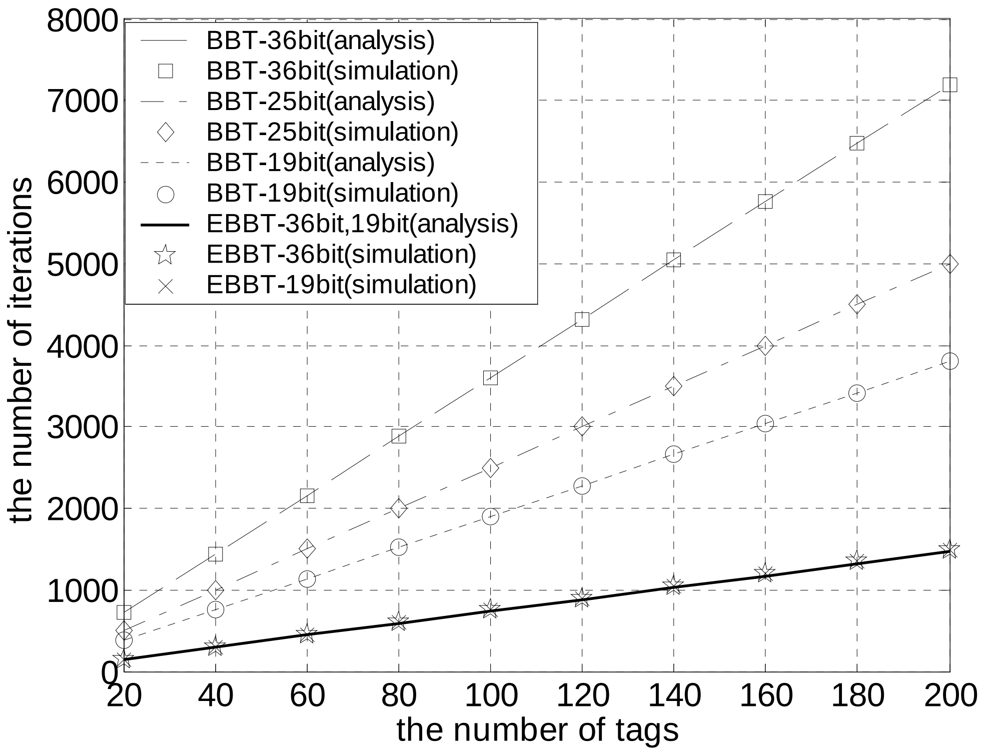

Figure 2. the number of iterations vs. the number of tags

Fig.2 shows the number of iterations for the number of tags in BBT and EBBT according to the number of tag's ID bit[6]. In Fig.2, lines represent analytic results and symbols represent simulation results using OPNET. Analytic results are very closed to the simulation results. Comparing EBBT with BBT, we can conclude that the number of iterations in BBT increases as the number of tag's ID bit increases, while in EBBT does not increase according to the number of tag's ID bit. That is why the EBBT shows better performance for the larger tag's ID bit. We found that the performance of EBBT is 288% higher when the number of tags is 20 for the 36 bit tag's ID. Moreover, we found that the performance of EBBT is about 380% higher when the number of tags is 200. As the number of tags increases the performance of EBBT is better than that of exiting algorithms.

6. Conclusion

We proposed and analyzed the fast wireless anti-collision algorithm(EBBT) for *u*-ID system. We mathematically compared the performance of the proposed algorithm with BS, SBT, and BBT. We also validated analytical results using simulation. EBBT shows better performance than existing algorithms as the number of tag's ID bit increases. The energy consumption is also lower than existing algorithms since the number of bits transferred from the tags is smaller. Furthermore, the performance of EBBT is better than exiting algorithms as the number of tags increases. On conclusion, if we apply the proposed algorithm 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.

Reference

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

[2] K. Finkenzeller, RFID Handbook: *Radio-Frequency Identification Fundamentals and applications.* John Wiley & 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 CSCC'99, 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.

[6] EPC Global, *EPC^{TM} Tag Data Standards Version 1.1 Rev.1.24*, Apr., 2004.

---

1) 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.