Adaptive M-ary Query Tree Search Algorithm for RFID Tag Identifications

Dongmin Yang', Jongmin Shine, Taekeun Parka

1 '
Daej eon University, Department of Information and Communications Engineering, Daej eon, Korea
dmyang@dju.kr

2 Samsung Electronics, Visual Display Division, Suwon, Korea
jm0621.shin@samsung.com

3 Dankook University, Department of Multimedia Engineering, Cheonan, Korea
tkpark@dankook.ac.kr, Corresponding author

Abstract. The query tree based RFID anti-collision scheme guarantees to identify all the tags, where all tag IDs are uniformly distributed. However, in real implements, the beginning part of tag ID is assigned according to a RFID code policy and the remaining part is sequentially given. In this paper, we propose an adaptive M-ary tree search protocol (AMTS), which effectively reduces unnecessary query-response cycles between the tags with the similar tag IDs. The results of simulations show that the AMTS significantly outperforms other schemes.

Keywords: RFID, anti-collision, query tree, tag identification.

1 Introduction

Radio frequency identification (RFID) technologies are powerful tools for object identification in various applications. A RFID system consists of a reader that identifies all objects in its field and a number of small and low-cost devices, called tags, each of which is attached to an object [1].

There are two types of anti-collision approaches: deterministic methods and probabilistic methods. The deterministic methods, which are on the basis of tree based protocols utilize the binary or n-ary tree structure to split tags into smaller subsets until the collision is resolved within a bounded delay [2][3][4][5][6][7]. In contrast, the probabilistic methods, which have a basis of the Aloha algorithm, where the tags respond to the reader's query at random time [1][8][9][10]. Since probabilistic protocols experience significant performance degradation in a dense tag environment and does not guarantee 100% tag identification rate, we focus on the deterministic methods, tree based protocols.

In this paper, we propose an adaptive M-ary tree search scheme (AMTS) which dramatically reduces the identification delay by coupling of M-ary tree search (MTS) [7], and Manchester code [1]. According to EPC Class 1 Gen. 2, a tag ID (96 bit) is made up of 4 data fields: a header (8 bits), a company prefix (28 bits), an object class (24 bits), and a serial number (36 bits). If items are manufactured by the same company or belong in the same product category, they may have very similar tag IDs. In a warehouse or a shopping center, where the similar products are stacked together, the tree-based protocols with bit-by-bit identification result in too many unnecessary query-response cycles. We further decrease the identification delay by Manchester code which can trace an individual bit about a collision or not.

The remainder of this paper is organized as follows. We discuss briefly related works in section 2. The proposed AMTS is shown in Section 3. Section 4 evaluates performance through simulations. Finally, we conclude the paper in section 5.

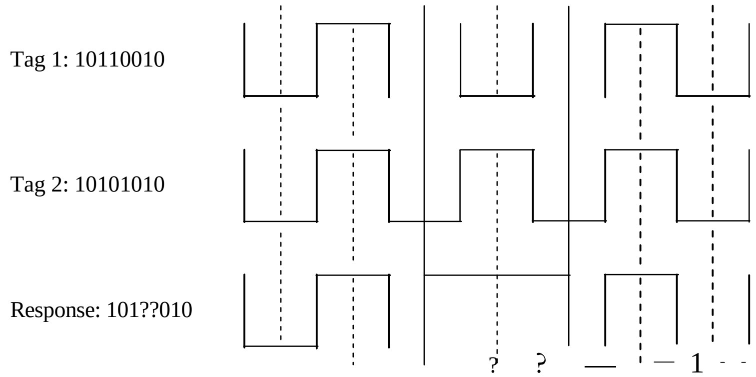

Fig. 1. The Manchester code can trace a collision to an individual bit.

2 Related works

In Manchester coding, the value of a bit is defined by the change in level (positive or negative transition). A logic 0 is coded by a positive transition; a logic 1 is coded by a negative transition. If two (or more) tags simultaneously transmit bits of different values, then the positive and negative transitions of the received bits cancel each other out, i.e., there is "no transition". This state is not permissible during data transmission and is recognized as an error. Therefore, Manchester code makes it possible to "trace a collision to an individual bit" as shown in Fig. 1 [4].

The query tree protocol (QT) [2][3][4][5] consists of query-response cycles. In each cycle, the reader deques a query *Q* from a queue initially containing an empty string A and broadcasts it. When the prefix of a tag ID matches *Q*, that tag transmits a response with its ID. If there is no collision in the responses of tags, the reader successfully recognizes one unique tag. Otherwise, the reader enques two 1-bit longer queries QO and Qlinto the queue. These cycles are repeated until the queue is empty. The tag identification process in QT is achieved by the single bit arbitration with 1-bit longer query. In the collision tree protocol (CT), a collision tree structure is used to capture the complete communications between the reader and the tags [6]. The prefixes generation and tag group splitting are based on the collided bit information from Manchester code. MTS [7], instead of the single bit arbitration, uses M-ary tree search scheme with a mapping function for *m* bits in Table 1, and *m* bits arbitration is made at a time, where *m* is $\log_2 M$. It reduces the query-response cycles for identification considerably.

Table 1. Mapping function for *m* bits

m bits	M(=2") bits
00...00	**0000...0001**
00...01	**0000...0010**
11...01	**0010...0000**
11...10	**0100...0000**
11...11	**1000...0000**

3 Adaptive M-ary tree search (AMTS)

While QT follows the single bit arbitration step as stated in Section 2, AMTS is based on MTS, where *m* bits arbitration can be achieved in every query-response cycle. To reduce unnecessary iterations, we use the information about the first collision bit in the response of tags. Table 1 shows the mapping function for *m* bits (m>1), which makes the arbitration feasible for multiple bits. The algorithm can be described as follows.

<Reader>

1) The reader sends out a query *Q*. Initially, it starts with an empty string *2.*
2) Two possible cases can arise based on the mapping part from a response *R.* The mapping part is a string of *r* bits from MSB of the entire response of tags, which is denoted by R_{mp}. Note that '*', '0' and`1' represent collision, logic 0 and logic 1, respectively.
   - A single '1' is in the mapping part and no '*' is in the response: One unique tag is identified.
   - A single `l'is in the mapping part and one more is in the response: If the response is $R_{mp}bib2...b_c\_ib,...$, where $b_iE10,11$ and b_e is the first collided bit, two queries $Qb_lb_2...b,\_0$ and $Qb_1b_2...b_{c\_1}0$ are prepared in the next step.
   - Otherwise: Let the number of 'k's *t* in the mapping part, then , r_t are the strings that can be inversely mapped from the mapping function. Queries *Qr* . . , *Qr,* are prepared in the next step.
3) Repeat steps 1)-2) until all tags are identified.

<Tag>

When the tag receives a query from the reader and the query is 2 or matches the prefix of the tag ID, then it responses the rest part of the tag ID except the prefix part. At this time, *m* bits from the MSB of the rest part are mapped into *M (=2m)* bits by the mapping function for *m* bits. If the size of the rest part *m*' is less than *m,* the mapping function for *m*' is used.

The communication procedure for an example with four tag IDs of 00011001, 00011010, 11100100 and 11100001 is illustrated in Table 2.

Table 2. Communication procedure for example

Cycle	query *(Q)*	response *(R)*		identified tag ID
		mapping part *(R,,,,)*	rest part	
1	*2*	**00**	******	
2	00	0010	10**	
3	0001100	10		00011001
4	0001101	01		00011010
5	11	0100	0*0*	
6	111000	0010		11100001
7	111001	0001		11100100

4 Performance Evaluation

To validate MTS, we have developed our own event-driven simulator. CT, MTS and QT are implemented and compared with AMTS in terms of required cycles for one tag identification. The simulation parameters are according to the IS0-14233 [1]. Tag IDs are grouped into two categories, each of which includes the same number of tags. Tag IDs in one group have an identical prefix of 60 bits and a uniquely assigned serial number of 36 bits. Due to significant differences in performance, we do not include QT in Fig. 2. AMTS always outperforms CT and MTS as shown in Fig. 2. We can know that, on the average, AMTS, CT, MTS and QT take 1.46, 1.95, 2.74 and 14.00 cycles to identify one tag, separately.

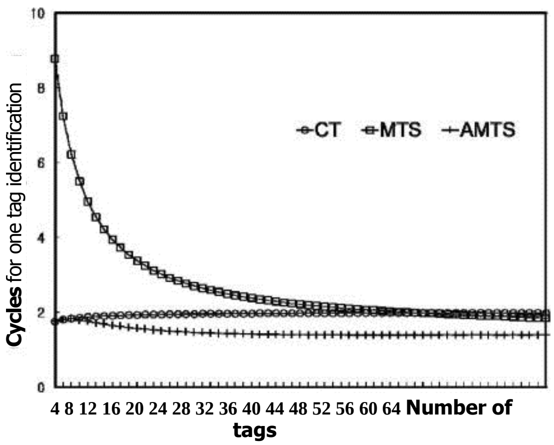

Fig. 2 Identification cycles to recognize all tags

5 Conclusion

We proposed the AMTS which effectively reduces the identification delay in real environment. Through simulation results, we show that AMTS significantly outperforms previously proposed schemes.

References

[1] K. Finkenzeller: RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification. pp. 206--219. John Wiley &Sons, (2003)

[2] C. Law, K. Lee, and K. Y. Siu: Efficient Memoryless Protocol for Tag identification. In: ACM DIALM '00, Aug. (2000)

[3] J. Myung, W. Lee, J. Srivastava, and T. K. Shih: Tag-splitting: adaptive collision arbitration protocols for RFID tag identification. In: IEEE Transactions on Parallel Distributed Systems, vol. 18, no. 6, pp. 763--775, June (2007)

[4] J. H. Choi, D. Lee, and H. Lee: Query tree-based reservation for efficient RFID tag anti-collision. In: IEEE Communications Letters, vol. 11, no. 1, pp. 85--87, Jan. (2007)

[5] P. Hawkes: Anti-collision and Transponder Selection Methods for grouped vicinity cards and RFID tags, In: BTG International Limited for IEEE, 25 Oct. (1999)

[6] Xiaolin Jia, Quanyuan Feng, and Chengzhen Ma: An Efficient Anti-Collision Protocol for RFID Tag Identification. In: IEEE Communication Letters, vol. 14, no. 11, pp. 1014--1016, Nov. (2010)

[7] Jongmin Shin and Dongmin Yang: M-ary tree search scheme for multiple tag arbitration. Technical Report DU-ICE-WINET-OOFF00, Daejeon University, Department of Information and Communications Engineering (2012)

[8] H. Vogt: Efficient Object Identification with Passive RFID tags. In: Proc. International Conference on Pervasive Computing, LNCS Springer-Verlag, Zurich, (2002)

[9] M. A. Bonuccelli, F. Lonetti, and F. Martelli: Instant collision resolution for tag identification in RFID networks. In: Ad Hoc Networks, vol. 5, no. 8, pp. 1220--1232, 2007

[10] G. Bagnato, G. Maselli, C. Etrioli, and C. Vicari: Performance analysis of anti-collision protocols for RFID systems. In: VTC Spring 2009 - IEEE 69th Vehicular Technology Conference, Barcelona, Spain, pp. 1--5, Apr. (2009)