A bit-locking binary tree algorithm based on odd-even zone

Hongbin Tang[1, a *], Shangbo Zhou[2,b] and Karim Awudu[3,c]

[1]Chongqing University of Education, Chongqing China.

[2,3] Chongqing University, Chongqing China.

[a]bin_tang78@163.com, [b] shbzhou@cqu.edu.cn, [c] awudubody@yahoo.com

Keywords: radio frequency identification; odd-even zone; bit-locking; anti-collision; binary tree.

Abstract. Along with the constant development and maturity of internet of things (IoT), radio frequency identification (RFID) technology has been widely used, anti-collision technology is the core of the RFID technology.This paper presents a new tree based anti-collision algorithm for RFID. The proposed algorithm combines odd-even zone with bit-locking back off anti-collision algorithm. The dividing odd-even zone improves the efficiency of searching, because the number of rounds between reader and tags is reduced, while the bit-locking back off anti-collision algorithm reduces transmission delay, beacause the number of transmitted bits decreases. Both theoretical and simulation results show the effectiveness and superiority of the algorithm.

INTRODUCTION

The Internet of things (IoT) is considered to be the third IT revolution after the computer and the Internet. IoT needs to be based on widely used radio frequency identification (RFID) technology [1]. The key of multiple tags identification in RFID technology is anti-collision. At present, the commonly used anti-collision algorithm is based on time division multiplexing address (TDMA) algorithm, and collision algorithms can be divided into two main approaches: ALOHA and binary tree algorithms. ALOHA is based on probabilistic anti-collision algorithm, it decreases the probability of collision by scheduling the responses of tags at random time [2-5]. Binary tree algorithm on the other hand, performs the tag identification using a unique ID for each tag, which leads to lengthy queries issued by the reader which cause long identification delay [6].

The basic binary search algorithm (BS) solves collision by gradually decreasing collided bits in a tag's serial number (tag's identification, ID). The reader can detect the position of collided bit by using the Manchester code, thus it is possible to trace a collision to an individual bit. Transition can determine the value of the bit, the positive transition means the logic value is 0 while the negative transition means the logic value is 1. The "no transition" means an exception occurs. If two or more tags simultaneously transmit, because of variation of two arbitrary tags in the interrogation zone, the positive transitions cancel the negative transitions in some bits. No transition detected leads to an exception. Therefore, the collision in individual bits can be discerned [7].

First off, the reader sends a REUEST(UID) command and every bit of UID is 1, then the tags that have serial numbers less or equal to the UID respond. And UID is the highest serial number all tags respond to. If only one tag transmits at a time, the reader will identify it. Otherwise, if more than one tags transmit at the same time, it is certain that a collision will occur and the reader will detect all collided bits.

The highest collision bit is set to 0, the value of every bit higher than this bit remains unchanged, and the lowest bit is set to 1 in the new UID to be sent in the next request, the tags whose IDs are less or equal to the UID will respond, in other words, the tags whose ID bits are higher than the highest collision bit are the same as corresponding bits of UID. In this way the scope of the responses to the tags is narrowed and iteration continues until all tags are identified [8].

notes: UID in this paper is different from other papers, the UID refers to the sequence of instructions transmitted by the reader.

There are a variety of improved binary tree algorithms, typically，dynamic binary search algorithm (DBS) which overcomes the weakness of BS algorithm with a shorter ID and smaller delay, lower channel utilization and so forth. When the reader detects a conflict, the next request command only sends higher bits where no collision occurs, thus the amount of data transmitted is reduced to improve the transmission efficiency [9]. In regressive-style binary search algorithm (RBS), when a tag is identified, the search returns to the parent node based on memory but not to the root node. Compared to the BS algorithm, RBS reduces the number of redundant searches [10]. In addition, there are other improved algorithms such as pruning branches binary search algorithm (PBS), where the reader sends a request command initially and all tags in the interrogation zone respond. The reader divides the signal returned into idle slots, collision slots or valid slots. Upon detecting an idle slot, the reader cuts the corresponding subtree. Collision slot is extended to the next level subtree as valid slots read the corresponding tag. The algorithm is suitable for searches with longer tag ID and a large number of tags, the disadvantage is that it is more challenging [11]. For bit-locking backoff algorithm (BLBO), by locking collision bit, tags do not need to transmit the full bits of ID during every response. It is a disadvantage that the algorithm does not reduce the response times [12]. Adaptive binary search algorithm (ABS) requires that each tag has a counter, which is used to record the current comparing ID bit. The reader sends a comparative bit, the tags whose corresponding bits are same will respond, next bit is sent to the reader and the counter is updated. According to the signal received, the reader determines the status of the slot as idle, collision, or valid. The search algorithm's efficiency is low, moreover, because of the increased counter, the cost of the tag is raised[13].

BIT-LOCKING ODD-EVEN ZONE ALGORITHM

Based on the existing binary tree algorithm, bit-locking odd-even zone algorithm (BLOE) is proposed. The zone-division method reduces the number of interrogations, besides, the bit-locking method makes responses without having to transmit the complete UID, thereby reducing the transmission delay. Theoretical analysis and simulation results show that the algorithm can effectively solve the tag collision problem.

Algorithm Conventions and Command

The reader can detect the position of a collided bit by Manchester code, the Manchester code makes it possible to discern a collision to an individual bit [7]. If the transmitted signal is positive, it means the logic is 0. However, a negative transition denotes the logic is 1. The "no transition" denotes an error. If more than one tags transmit bits of different values at the same time, the positive signals cancel the negative signals of the received bits. No transition is detected and that leads to an error.Thus the collision of an individual bit can be traced.

Here,we redesign the algorithm command as follows:

(1) REUEST(UID)

(2) SELECT(UID)

(3) READ_DATA

These three commands are introduced almost in every binary tree algorithm.

(4) REQUEST _ E(UID 0)

The bit-locking request command tags, the sum of whose bits is even and belong to the even-zone. If conflicts are detected, the reader sends a REQUEST _ E(UID 0). All tags within the even-zone respond. The reader receives responses and determines all conflicting bits. All conflicting bits are then converted to 1 and all non-conflicting bits are converted to 0, to generate a new sequence as the next bit-blocking command, the tags respond to the bit-blocking command as follows: Each tag in the even-zone will compare the bit-blocking command and its own UID. The tag just needs to focus on the '1' bits of the bit-blocking command, tags whose highest bit corresponding to 1 bit in the bit-blocking command will respond with 0. These tags return the rest of the bits corresponding to 1 in

the bit-blocking command to the reader. For example, the number of '1' bits in the bit-blocking command is *p*, then eligible tags only return *p*-1 bits to the reader.

(5) REQUEST_O(UID 0)

It is the bit-locking command that acts on odd-zone, the rules are similar to even-zone and need not be repeated here.

The Flow Chart of the Algorithm

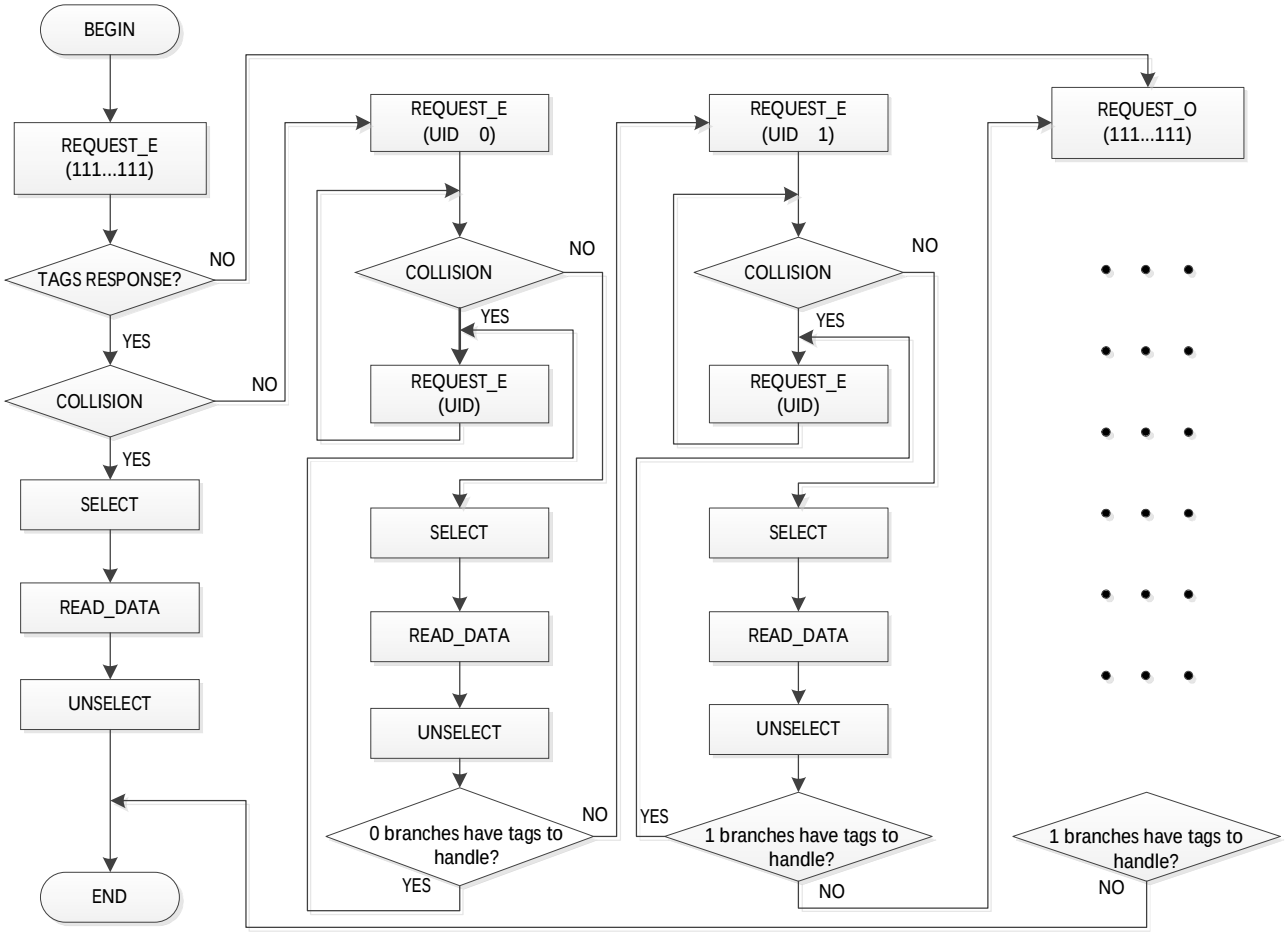

Fig. 1 The flow chart of BLOE

The flow chart of the algorithm as shown in Fig. 1 above, consists of 2 parts, even-zone and odd-zone, meanwhile, odd-zone is similar to even-zone, so it is omitted.

The algorithm can be described as follows:

1) The reader sends REQUEST_E(111...111) command, all tags in the even-zone respond and send UID to the reader.
2) If the reader does not detect any signal, then jump to stage 6, otherwise, jump to stage 3.
3) The reader decodes all the signals sent by the responding tags and determines whether there is a collision. If there is no collision, the reader sends "SELECT" and "READ_DATA" command consecutively, so as to select the tag and write the tag's ID to cache, and then the reader sends "UNSELECT" command, which contains ID of the selected tag to set it to sleep status. If there has been a collision, jump to stage 4.
4) The collision bit position will be set to 1, no collision bit position will be set to 0. The description above is the main composition of the UID. The reader sends REQUEST_E(UID,0) command, the tags in the even-zone receive the command, compare the UID with their respective IDs and lock collision bits. The tags whose highest locking bit is 0 will respond to this command, so they send the remaining locking bit to the reader. If there is no collision, the reader sends SELECT and READ_DATA commands to select tag and write the tag's ID to memory, then sends

UNSELECT command to let it sleep. If there is a collision, the reader decodes it again, repeats the previous process until there is a tag to be identified, then uses the rollback strategy to return to the last parent collision node. The same method can identify another branch of this node. This recursion goes on until all tags whose highest locking bit is 0 are identified and then jump to stage 5.

5) The reader sends REQUEST(UID, 1) command, the tags whose highest locking bit is 1 in the even-zone will respond and send the remaining locking bit to the reader. If there is no collision, the reader sends SELECT and READ_DATA commands to select tag and write the tag's ID to memory, then sends UNSELECT command to let it sleep. Recursion is similar to the 0 branch until all tags whose highest locking bit is 1 are identified.
6) Similarly to the tags having been identified in the even-region, the same method will be used to identify tags in the odd-area, until all the tags are identified.

ALGORITHM PERFORMANCE ANALYSIS

In the BIBO algorithm, there are n tags in the query zone, $Q(n)$ denotes the number of interrogation [12], then

$$Q(n) = 2n - 1$$

The number of tags in the assumed odd region and even region is p and q respectively. Then

$$Q(n) = Q(p) + Q(q) = (2p - 1) + (2q - 1)$$

$$= 2(p+q) - 2 = 2n - 2$$

To avoid omissions, whether odd zone or even zone has 0 tags, the reader repeats the query once again in each zone. In this case,

$$Q(n) = Q(p) + Q(q) = 1 + (2q - 1) = 2q = 2n$$

The probability of such a situation is $1/2^{n-1}$, so the total number of interrogations is as follows:

$$Q(n) = 2n * 1/2^{n-1} + (2n-2) * (1 - 1/2^{n-1}) \quad (1)$$

If there ever was n tags within reader's scope, then two bits collision might have C_n^2. The probability of x bits collision is

$$p = 1 - (1 - \frac{C_k^x}{2^k})^{C_n^x} \quad (2)$$

Obviously, when tag length and tag number are invariant, the probability of 2 bits collison is more than that of 1 bit collison. It means that BLOE's number of interogations is less than that of BLBO's.

Transmission Delay

Because the reader's command processing time is much less than the transmission time of the command, the main consideration of general analysis of RFID is transmission delay [17]. In some systems, the transmission rate between the reader and the tag is constant, the transmission delay on the other hand is determined by the number of interogations and the length of each transmission (bit). The reader's request command and the tag's response command are used as a data frame, its head and tail both have 5 bits free time. The reader also needs to spend 1 bit time more on receiving the tag response data. Having accepted the tag response data, the reader also needs to spend 1 bit time to check. The reader takes 2 bits time to record the ID that the tag returns. After recognizing a tag, the reader is required to send an UNSELECT command (it includes tag id) to put it into a sleep state.

There are n tags within a reader's scope, assuming the length of UID is k bit, collision has y bit. The communication between reader and tags needs time L_0: $L_0 = 2k + 21$; bit-locking command and response need k+y+21 bit time. If there is no collision, another 2 bits time will be used to record tag ID. In this case, $L_0 = y + 23$. Therefore the bit-locking backoff alogorithm requires the overall length L:

$$L = (2k + 21) + (k + y + 21)(n - 1) + (k + y + 23)n \quad (3)$$

For bit-locking based on odd-even zone algorithm, inquiry command needs to add one bit to identify the odd-even zone, so BLOE algorithm's average overall length L' is

$$L' = 2*(2k+22)+(k+y+22)(n-2)+(k+y+24)n \quad (4)$$

$$L'-L = 2n+k-y$$

As can be seen from the difference between equation(3) and equation(4), if $2n+k>y$, the BLBO algorithm has advantages over BLOE algorithm, otherwise, the BLOE algorithm has higher optimizing efficiency. In fact, the above equation does not consider the following situation: the BLBO algorithm can discern two tags when only 1 bit collision occurs, however, unlike BLBO, the BLOE algorithm can discern two tags and 2 bits collision besides 1 bit collision. According to equation(2), when k=8 and n=5, the probability of two tags collision bit x=1 is 14.6%, but the probability of two tags collision bit x=2 is 44%. Let's look at the effect by simulation.

SIMULATIONS AND ANALYSIS

This article uses the c++ programming to verify the performance of the algorithm. Conflicting bits are randomly generated, we vary the number of tags every time and compare the improved algorithm （BLOE） with the BS and BLBO alogorithms in the ideal channel.

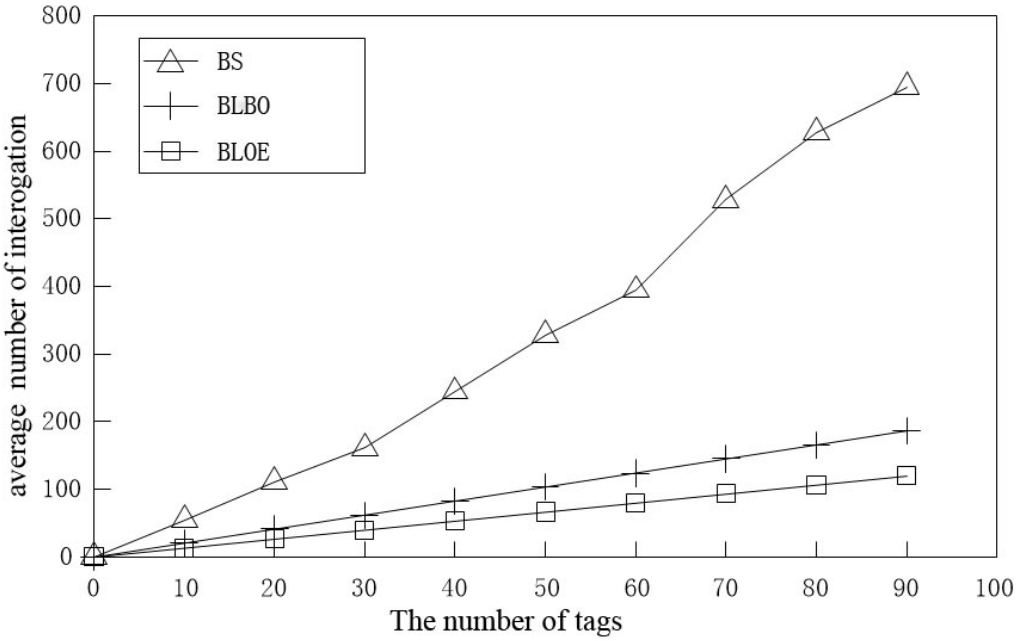

Fig.2 The average number of interogations used to identify tags with the varying number of tags

Tag ID is randomly generated, the amount varied within 90, constant length of 64 bits, test run for 50 times and the average taken and then compared as in figure 2. It is obvious that the improved algorithm is superior. This is because the BLOE algorithm can identify one or two conflict bits at a time. But other algorithms can only identify one collided bit at a time.

To minimize the problem of transmission delay being larger than reader processing delay, in the following simulation result, transmission delay is set as a parameter to compare the performance of the three algorithms.

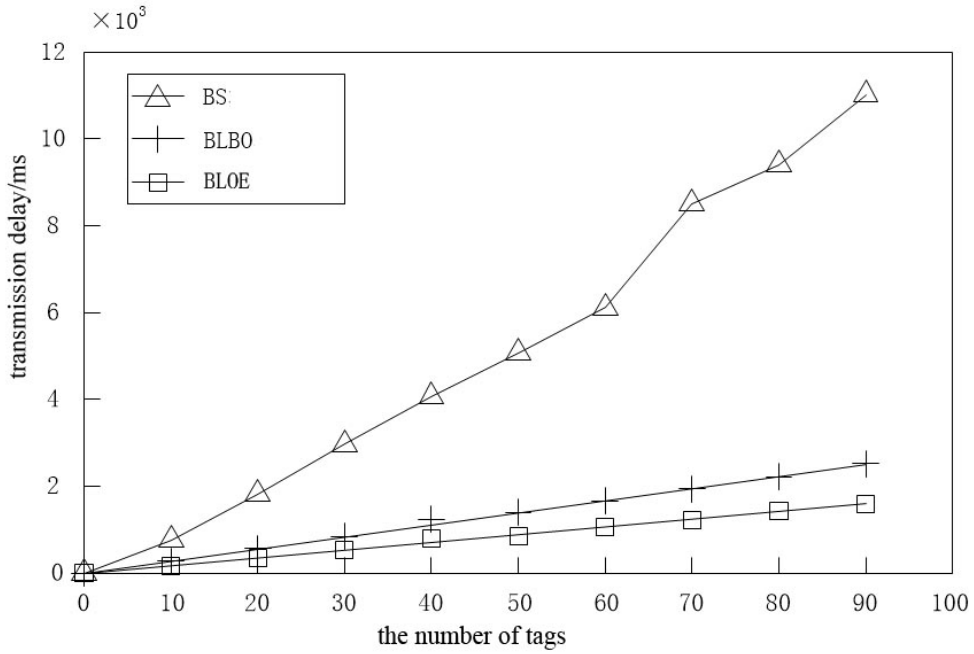

Fig.3 Change of transmission delay with the number of tags

The DB, BLBO and the improved algorithm of transmission delay trend with the number of tags as shown in Fig.3, where transmission rate unit is set to 100kbit/s. As can be seen from the chart, the transmission delay of the improved algorithm is less than that of the DB and BLBO algorithms. It can be seen from the above illustration that BLBO algorithm and the improved algorithm only transfer conflicting bit when tag responds, so their transmission delay are smaller than that of DB algorithm. For each query, the reader must send an instruction one bit more than BLBO algorithm to identify odd-even zone. Without considering the case of two tags and two bits collision at this point, we can state BLOE's performance is close to BLBO's. However, due to the fact that the fronter can identify one and two bits conficts in one query, its total number of interrogations is less than that of BLBO, therefore, the transmission delay is less than the latter. It is interesting that the corresponding parameters have the same trend with the increase of the tag number as in figure 2 and figure 3, This is determined by the characteristics of each algorithm.

CONCLUSION

Based on the advantages of the bit-locking backoff algorithm (BLBO) and the parity of binary tree odd-even zone search algorithm, we propose an improved algorithm (BLOE). A reader in the odd zone and even zone implements the BLBO algorithm. The bit-locking backoff algorithm has a lower time delay, because only conflicting bits are transfered. The improved algorithm introduced into the odd-even zone makes the reader further reduces the number of queries. Theoretical analysis and simulation results show that the performance of the improved algorithm further improves the efficiency of the system identification compared with the common anti-collision binary tree search algorithms.

References

[1] Haifeng Wu and Yu Zeng,Passive RFID Tag Anticollision Algorithm for Capture Effect, IEEE SENSORS JOURNAL, VOL. 15, NO. 1, JANUARY 2015.

[2] J. van der Geer, J.A.J. Hanraads, R.A. Lupton, The art of writing a scientific article, J. Sci. Commun. 163 (2000) 51-59.B. Glover and H. Bhatt, RFID essentials. Farnham: O'Reilly, 2006.

[3] L. Liu and S. Lai, "ALOHA-Based Anti-Collision Algorithms Used in RFID System," in Wireless Communications, Networking and Mobile Computing. China 2006.

[4] H. Cho, W. Lee, and Y. Baek, "LDFSA: A Learning-Based Dynamic Framed Slotted ALOHA for Collision Arbitration in Active RFID Systems," Advances in Grid and Pervasive Computing, pp. 655-665, 2007.

[5] C. Lee, H. Cho, and S. W. Kim, "An Adaptive RFID Anti-Collision Algorithm Based on Dynamic Framed ALOHA," IEICE TRANS.COMMUN, vol. VOL.E91–B, 2008

[6] J. Myung, W. Lee, J. Srivastava, and T.K. Shih,"Tag-Splitting: adaptive collision arbitration protocols for RFID tag identification," In IEEE Trans. on Parallel and Distributed Systems, vol. 18 (6), pp. 763-775, Jun. 2007.

[7] MAJID ALOTAIBI, ADAM POSTULA, and MARIUS PORTMANN, Tag Anti-collision Algorithms in RFID Systems - a New Trend, WSEAS TRANSACTIONS on COMMUNICATIONS, vol. 8 (12), pp. 1216-1232, December. 2009.

[8] Ziqiang YU,Xianqiang Liu, Improvement of Dynamic Binary Search Algorithm Used in RFID System, 2011 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference, 1046-1049, July. 2011.

[9] Fabao YAN1, Xianhe SHAO2,Qian SUN, The Dynamic Anti-collision Algorithm Based on the Similar Binary in RFID System. Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on. 2008: 444 - 447.

[10] Xiaoyun Chen, Guohua Liu, Yukai Yao, Yi Chen, Shengfa Miao, Youli Su, An Improved RFID Anti-Collision Algorithm Based on Regressive-style Binary Search Tree ,/ 2010 International Forum on Information Technology and Applications, 2010:403-406.

[11] Songsen Yu，Yiju Zhan，A Binary-tree Searching Anti-collision Algorithm Based on Pruning Away Branches and Its Practice [J]，Computer Engineering，2005，Vol.31 No16:217-230.

[12] Xue Wang，Zhihong Qian, Zhengchao Hu,Yinan Li，Research on RFID anti-collision algorithms based on binary tree [J]. Journal on Communications，2010，31 (6) : 49-57.

[13]Dongxue Wei,Jiali Zhen,Liangliang Li,Fushi Yao. Study of Novel Adaptive Multi-tree Anti-collision Search Algorithm. Computer Science. Vol.40 No.10 Oct 2013.

[14] SU-RYUN LEE, SUNG-DON JOO, CHAE-WOO LEE.An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification [C]. Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems,2005:166-172.

[15] WANG Fei,ZHANG Wu. Based on the grouping of dynamic frame-slotted ALOHA algorithm[J]. Computer Systems & Applications,2013,22(7):77-80.

[16] Zhiguo Xia, Yigang He, Zhouguo Hou. Binary-tree bit-detecting RFID tag anti-collision algorithm.Computer Engineering and Applications，2010，46（20）：245-248.

[17] LAI Y C,HSIAO L Y.General Binary Tree Protocol for Coping with the Capture Effect in RFID Tag Identification[J]. IEEE COMMUNICATIONS LETTERS,2010,14(3):208-210.

[18] Liang Liu, Huan-ge Xing,Jin-wei Guo, Anti-collision algorithm based on odd-even zone search and its simulation analysis.Computer Engineering and Design,2010,31(12):2740-2743.