Anti-collision Algorithm based on Multi-threaded RFID Lock Position

YanMin Ma and Di Jin

School of computer and information engineering
Harbin University of Commerce, Harbin, China
mym654321@163.com

Abstract

Combined with lock algorithm, retreat algorithm and multithreaded processing thoughts; it puts forward an anti-collision algorithm based on multi-threaded RFID lock position. RFID reader sends a quad command, RFID tag response the information of collision position, reduce the amount of data transmission; Fused with lock and multi-status thoughts, the algorithm checks the lock position of card numbers, and continuously separates the status of card numbers in the searching process. Using multithreaded processing idea, tags are classified by the algorithm, and collision handling time is shortened. The algorithm solves the problem of idle thread by tag group. From the test results, compared with the original algorithm, this algorithm reduces the times of collision identification and digit number of data transmission, and improves the system performance.

Keywords: *RFID tags conflict; lock position; Retreat strategy; Multithreaded*

1. Introduction

RFID (radio frequency identification RFID), electronic tag technique, this kind of technique uses the magnetic wave as the transmission media, and communicates parallelly by radio frequency. RFID system consists of RFID tag, RFID reader, and central information processing system. The relevant information of a product is preserved in a RFID tag, and the unique electronic code could use as the product ID; RFID reader can read and write the information of a tag, and transmit the information to central information processing system; Central information processing system could manage, analyze and transmit the information [1-5].

In a system, Information interference and collision happens when RFID reader reads or writes more than one tag at the same time, then all of the RFID tags share the same communication tunnel and transmit information to the RDIF reader. There are two main anti-collision algorithms based on RDIF: algorithm based on ALOHA and algorithm based on tree. The algorithm based ALOHA implements retransmission after collision to solve the problem, but, with the increasing number of tag, tag collision becomes serious, and entire performance declines continuously.

Anti-collision algorithm based on tree(certain algorithm), references [2-3] binary search algorithm (BSA)will come back to the start point every time when it identifies a tag, but, as the increasing number of tag, the system efficiency will decline; References [4-5] back binary

Project Supported: social science research project of Heilongjiang province (12D056); natural science foundation of Heilongjiang province (F201243); 2012 higher education teaching reform project of Harbin University of Commerce (201242).

search algorithm (BBSA)reduces identification times, but do not the digit number of transmission; References [6] dynamic binary algorithm based on retreat reduce the identific ingation times and digit number of transmission; References [7-8] the method of lock position turns the transmission position lock to collision position; References [9-12] divides the tag into 3 categories of status; References [1] lets tags distribute into multithreaded processing, and improves the processing efficiency, but do not solve the problem of idle thread.

To solve the efficiency problem caused by transmission digit and identification times, this essay, coalesced with the thoughts in algorithms above, implements the improved multithreaded method to solve the problem of idle thread and anti-collision algorithm based on lock position in every thread, and puts forward anti-collision algorithm based on multithreaded lock position.

2. The Basis of Algorithm

2.1. Manchester code

The premise to make the anti-collision algorithm reality is confirming the collision position. Manchester code can be used in the RFDI code, the code represents the bit by level switch in a half of bit cycle (positive switch means a binary 0 / negative switch means a binary 1) , if level were not switched, coding would be error or misrepresentation. If you have two RFID tags (12 bits): the tag A: 1001 0011 0110, tag B: 1001 1011 0010. RFID reader reads the information: 1001 ?011 0?10, the clock level offsets with each other on the D2, D7 position of tag A and B, collision on D2 and D7 shown in Figure 1.

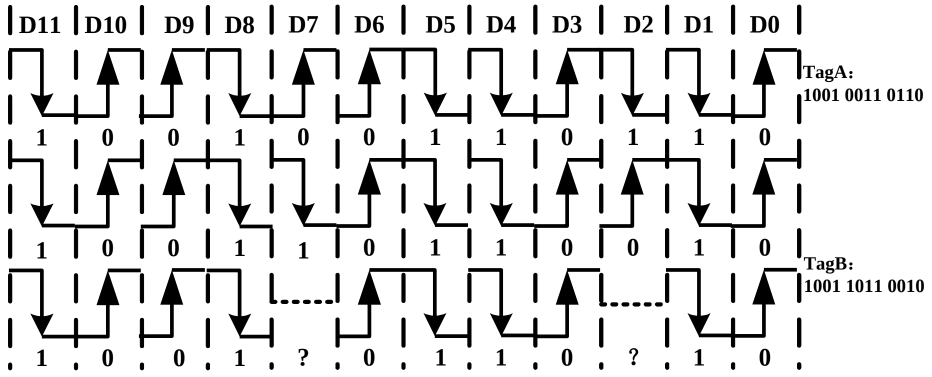


Figure 1. Manchester code to determine a collision Database contexts

2.2.State arbitration

To solve the searching time problem of dynamic back binary tree algorithm, the improved algorithm introduced multi status thought, the tag is divided into living state, resting state and preparing state. Mark each elected tag’s status to living state, turn the tag’s status which confirm to the request to preparing state, or resting state. In each tag search, the next search scope can be limited in preparing state tags. All the tags will participate in the arbitration process, when there are only two tags in preparing state, reader can read two tags directly, *i.e.*, one of the tags selected, another tag in the following arbitration process will no longer sends a signal to the reader, card reader can process corresponding data directly. Only when card number in preparing state have been searched, tag in dormancy state could turn to preparing state, then the search could continue.

2.3. Lock position processing

Due to the application of tag card number's length is 8 Byte ~ 10 Byte commonly, and the collision position is limited, so this paper through the lock position paging instruction locked collision's bit position, and makes anti-collision operation in the collision position, and this improved algorithm based on dynamic back algorithm reduces data redundancy bit once again.

2.4. Multithreaded

In a program, the program fragment operating independently called "thread" , programming concept used it is known as "multithreaded processing". Multithreading completes a number of tasks synchronously, and improves the system efficiency by making the resource usage efficiently. If the processor is multiple processors, each thread is equivalent to assign a processor, "parallel arithmetic" is reality. In programming, the most valuable characteristic of "multithreaded processing" is that programmers don't need to care about the processor's model, only need to programme multithreaded. This paper using 4 threads completes its algorithm, groups all tags (8 group), and solves the idle process problems effectively through scheduling grouped tags orderly.

2.5. Command Definition

For the convenience of the algorithm description, definiting request command Req (Di, Mi, Tj, Gk) and response command Res (S) under the premise that the original RFID system command is preserved. Parameter Di in the request command is the highest collision position (a RFID is 12 bits ID, and the Di is a 4 binary number 0000 ~ 1100, which indicates D0 ~ D11 position), Mi is the parameters on collision position (0/1), Tj is thread number (4 threads, two binary number indicates four threads), Gk is tag's group number (8 group, three binary number indicates eight groups).

3. The Process of Algorithm

Under the assumption that RFID tag code is 16 bit, there 20 tags need to be identified, as it shown in Table 1.

The tree form of improved algorithm is shown in Figure 2, 4 threads and 8 tag groups deal with this algorithm, the tree node indicates different request commands according to the highest collision position.

Algorithm process is as following:

3.1. Multithreaded Grouped Scheduling

RFID reader sends sync signal to readable tags, tags response with their own ID. RFID reads back the binary sequence:1??? ?0?? ?0??, and judges D10, D9, D8, D7, D5, D4, D3, D1, D0. RFID reader will sent the collision position to the RFID tag.

According to the highest three collision position to group tag, RFID tags judge the group which they should belong to, according to their own ID sequence and the collision position. For example, according to the highest collision position:D10, D9, D8, tags are divided into eight groups, the first 4 groups(0 to 3 group)add into four threads(0-3 thread) for processing, detailed grouping situation is shown in Table 2. "0" group only one tag, so it's identified quickly, thread "0" state is empty. The fourth set of tags could be scheduled into the thread "0" for identification, until all the tag groups have been recognized. This method is effective to solve the problem of the idle thread, and improves the efficiency of recognition. The third group of tag identification procedure is used as example to explain the process of lock position anti-collision algorithm, the collision tree to deal with collision is shown in Figure 2, flow diagram is shown in Figure 3.

Table 1. Tag's ID

Tag	Tag ID											
Name	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Tag1	1	0	0	0	0	0	0	1	0	0	0	1
Tag2	1	0	0	1	0	0	1	0	0	0	1	0
Tag3	1	0	1	0	0	0	1	1	0	0	1	1
Tag4	1	0	1	0	0	0	0	1	1	0	0	1
Tag5	1	0	1	0	0	0	0	0	0	0	0	0
Tag6	1	0	1	0	1	0	0	0	1	0	0	0
Tag7	1	0	1	0	1	0	1	0	0	0	1	0
Tag8	1	0	1	1	0	0	1	0	1	0	1	0
Tag9	1	0	1	1	0	0	0	0	1	0	0	0
Tag10	1	0	1	1	0	0	1	1	0	0	1	1
Tag11	1	0	1	1	0	0	1	0	0	0	1	0
Tag12	1	0	1	1	1	0	0	1	1	0	0	1
Tag13	1	0	1	1	1	0	0	1	0	0	0	1
Tag14	1	0	1	1	1	0	1	0	0	0	1	0
Tag15	1	0	1	1	1	0	0	1	1	0	0	0
Tag16	1	1	0	0	0	0	0	1	0	0	0	1
Tag17	1	1	0	0	0	0	1	1	0	0	1	1
Tag18	1	1	0	0	1	0	1	1	1	0	1	1
Tag19	1	1	0	1	1	0	1	1	1	0	1	1
Tag20	1	1	1	1	1	0	0	0	1	0	0	0

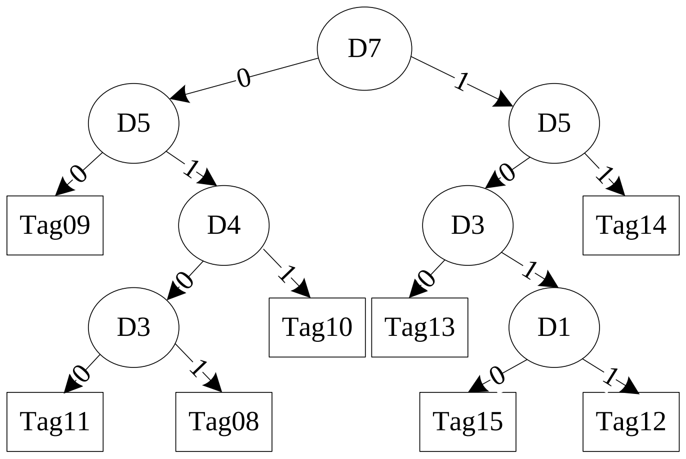


Figure 2. Thread 3 Dealing with Collision in the Form of Tree

Table 2. Tags Classification

D10D9D8	Group	Tags
000	0	Tag1
001	1	Tag2
010	2	Tag3、Tag4、Tag5、Tag6、Tag7
011	3	Tag8、Tag9、Tag10、Tag11、Tag12、Tag13、Tag 14、Tag15
100	4	Tag16、Tag17、Tag18
101	5	Tag19
110	6	
111	7	Tag20

3.2. Lock position anti-collision algorithm procedure is as following

① Reader sends Req: 1011 1111 1111, gets eight tags, and detects conflict positions, then processes lock positions, finally, presses stack in no collision position, and gets new collision sequence.

② Reader sends request number for new conflict sequence, according to whether meet the requirement responses and marks state, counter counts the number of preparing state.

ance of the state arbitration is superior.

③ Sending request number, if there is any collision in card number returned, process comes back to step 2; If there is no collision, tag can be read and marked at successful state, counter minus 1.

④ If the counter is 1, tags can be directly processed data according to the situation arbitration, after that, the tags can be marked at successful state, counter minus 1; If the counter is 0, the dormant state tags can be changed to preparing state, counter is reset, and then turn to step 5; If the counter is 0 and dormant state has not been tagged, then the searching finish.

⑤ According to the thought of back algorithm, process backs to the last collision position, and gets the new request command, then, turns to step 3. The example above shows the recognition process of this algorithm under the situation that 5 tags conflict, when any parameters of these: the tag's length of card number, the number of tags, the number of collision position, appear to change, this algorithm can identify each tag quickly, and suitable for all kinds of tag situation. When the tag's card number is continuous, the chance of appearing 2 preparing states increases greatly, so the perform

4. The Description of Problems related to Algorithm

Request command enters into the command stack. Due to the operation of the algorithm to the tree is from left to right, so the algorithm can determine whether the command can enter into the stack according to the mi parameter value of command, mi is "0" means entering into the stack, "1" means not.

Command of output stack. According to the back principle of algorithm, algorithm should be back and searched tree node after tags have been identified. The output stack operation is executable at that time, and mi of output stack command turns from "0" to "1", and the next search can be implemented.

Multithreaded processing. If RFID reader is multifrequency , and the CPU is multiple processors, this multithreaded is "parallel processing"; IF RFID is single frequency, a command queue storing the Req commands which sent out at same time need to be added, this is "multithreaded processing", which can use RFID reader to process information and read/write RFID tag content at the same time. The two methods can improve the efficiency of system.

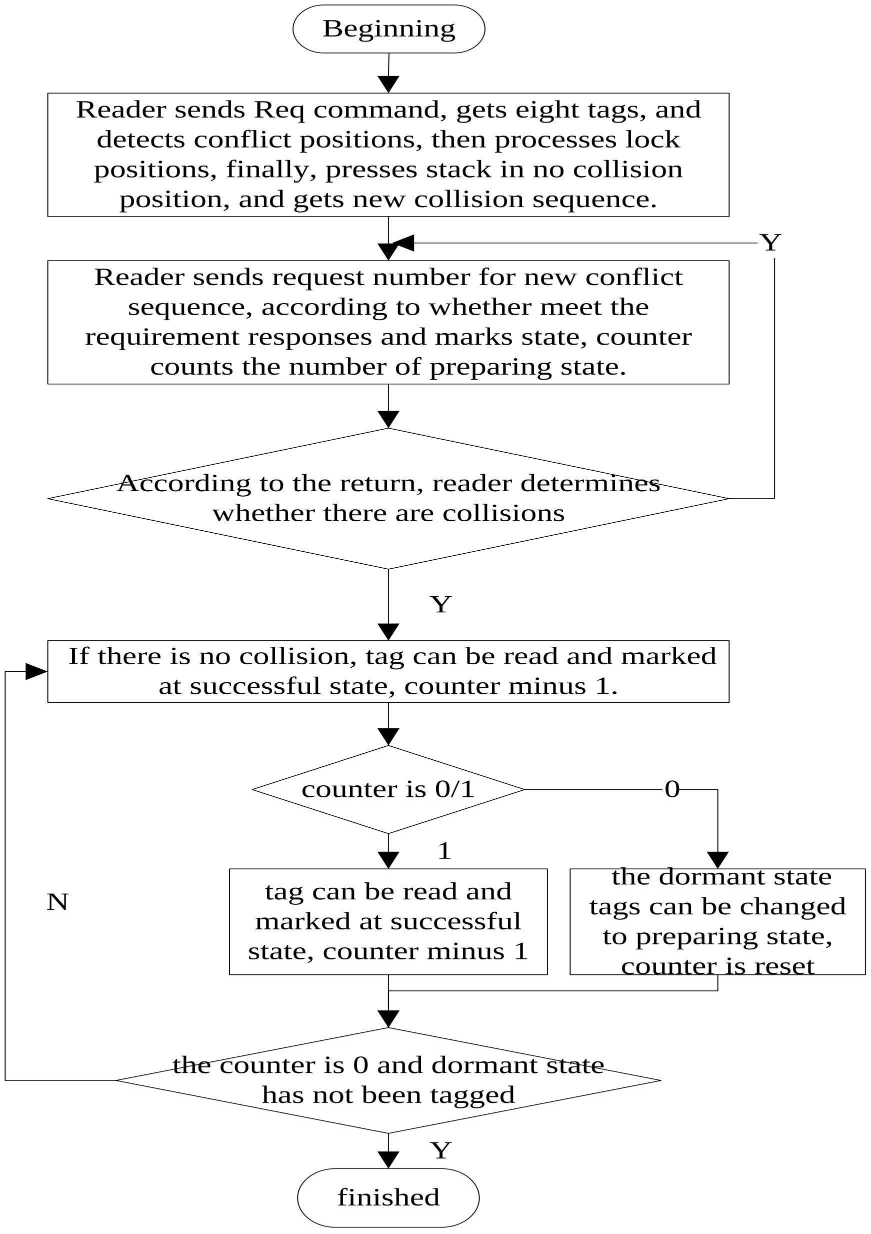

Beginning
Reader sends Req command, gets eight tags, and detects conflict positions, then processes lock positions, finally, presses stack in no collision position, and gets new collision sequence.
Y
Reader sends request number for new conflict sequence, according to whether meet the requirement responses and marks state, counter counts the number of preparing state.
According to the return, reader determines whether there are collisions
Y
If there is no collision, tag can be read and marked at successful state, counter minus 1.
counter is 0/1
0
1
N
tag can be read and marked at successful state, counter minus 1
the dormant state tags can be changed to preparing state, counter is reset
the counter is 0 and dormant state has not been tagged
Y
finished

Figure 3. Algorithm Flow Chart

5. The Performance Analysis

Evaluation of an anti-collision algorithm based on RFID system depends on the query times and time consuming of identification which the algorithm needs to search all tags. If the number of RFID tags is Num, the code's length of RFID tag is Len.

In the BAS algorithm, the coding length of RFID tag is Len; In BBAS algorithms, RFID tag coding length is log2Len+1; In the Req command of this algorithm, the length of Di is relevant to the length of code, mi is in "1" position, Tj, related to the number of threads (Tnumber), is log2Tnumber, Gk, related to Group (Group), is log2Group. So the length of binary code sent in this algorithm is log2Len+1+log2Tnumber+log2Group.

In BAS algorithm, algorithm open a loop to search RFID tags , so the inquiring times to identify a tag with "Num" is Num(Num+1)/2; In BBAS algorithm, the inquiring times to identify a tag with "Num" is 2*Num-1; The average inquiring times for each thread is (2*Num-1)/Tnumber in this algorithm.

In algorithms above, the binary data length transmitted is:

BAS algorithm:

$$Num(Num+1)/2*Len \ . \qquad (1)$$

BBAS algorithm:

$$(2*Num-1)*(\log_2 Len+1) \ . \qquad (2)$$

Multithreaded processing BAS algorithm:

$$((2*Num-1)/Tnumber)*(log2Len+1+log2Tnumber+log2Group). \qquad (3)$$

Using 20 tags in Table 1 as samples and comparing performance of the BAS, BBAS and multithreaded processing BAS algorithm, and indexes of the comparison are shown in Table 3.

Table 3. The Index of Algorithm Comparison

Algorithm	The times of sending command	The amount of bits transmitted by Req command （bit）
BAS	210	2520
BBAS	39	195
MBAS	10	90

Along with the increasing number of tags and code bits, and threads processed, the advantage of this algorithm will become more and more obvious. Figure 4, Figure 5 is Matalab simulation of retreating BAS and MBAS algorithm, x axis is the number of tags, y axis is the system's throughput. In Figure 2, RFID tag code is 12 bits, the number of thread is 4, the number of group is 8; In Figure 4, RFID tag code is 64 bits, the number of thread is 8, the number of group is 16.

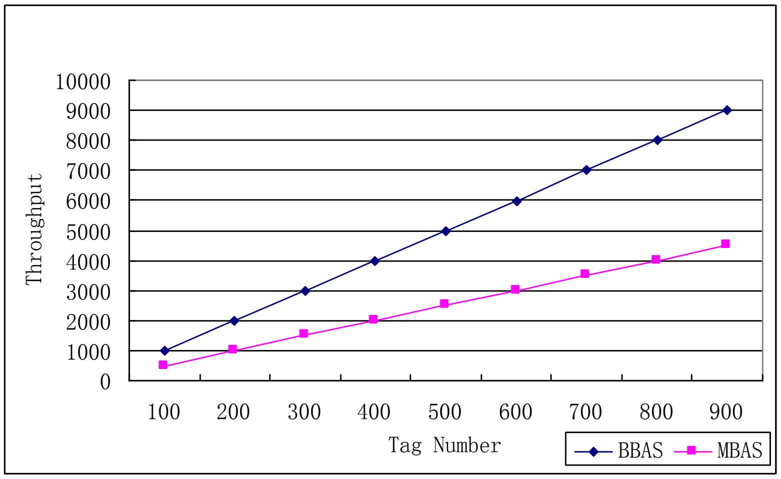


Figure 4. Algorithm Throughput

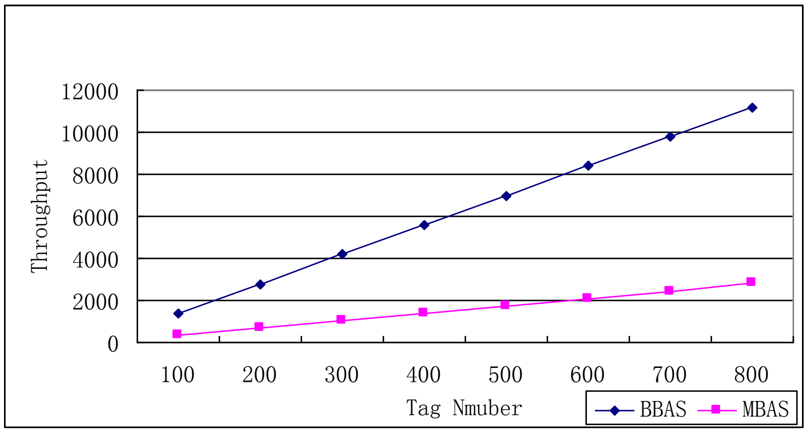


Figure 5. Algorithm Throughput

From the table, the order sending number and the data quantity are greatly reduced in the new algorithm; the causes should be the following.

① Lock position and back method reduce the number of orders to be sent.

②The parameters of the command is not all RFID tag's ID, but the collision position and value are represented by position. When the tags ID's bits increase, the characteristic will be more obvious.

③With the increasing number of thread, the number of groups also increases, although RFID tag's coding length increases, the total number of bits transmitted reduces.

The comparison of execution time of algorithms, as shown in Table 4. If the time of RFID reader sending 1 bit data is T1; the time of RFID tag responsed 1 bit data time is T2; the time of RFID processing collision of 1 bit data time is T3.

Table 4. The Time Comparison of Algorithms

Algorithm	Time of Request Command	Time of Response Command	Time of processing collision
BAS	200T1	132T2	T3
BBAS	304T1	80T2	T3
MBAS	64T1	10.7T2	T3/4

As Table 4 presented, the improved algorithm is improved in speed, the causes should be the following.

①Multithreaded processing, the collision handling time is reduced.

②Solving the problem of the idle thread effectively.

③The data of command transmission is reduced, and data identification time is reduced.

④Collision handling and command transmission can be multithreaded processing, then the operation speed improves.

6. Conclusion

This paper puts forward a lock position anti-collision algorithm based on multi-threading to reduce the quantity of command transmission data and the reader's recognition time, and to improve the system's processing speed. Performance analysis shows that this algorithm is better than the BAS algorithm, back BAS algorithm, also it processes the idle process, and is more suitable for the long tag ID and big quantity tags. The next work step is to adjust the number of threads and groups by adaptive algorithm.

References

[1] D. Jin, Y. Ma, Z. P. Fan and X. Fu, "A RFID anti-collision algorithm based on multithread regressive-style binary system", 2012 International Conference on Measurement, Information and Control, **(2012)**, pp. 365-369.

[2] Q. Y. Dai, R. Y. Zhong, M. L. Wang, X. D. Liu and Q. Liu, "RFID-enable Real-time Multi-experiment Training Center Management System, IJAST, vol. 7, **(2009)** June, pp. 27-48.

[3] M. -Y. Chen, C. -N. Yang and C. -S. Laih, "Authorized Tracking and Tracing for RFID1-14.IJSIA, vol. 1, no. 1, **(2007)** July.

[4] B. King and X. Zhang, "Applying RFID to Secure the Pharmaceutical Supply Chain", IJSIA, vol. 1, no. 2, **(2007)** October, pp. 71-84.

[5] S. I. Ahamed, F. Rahman, E. Hoque, F. Kawsar and T. Nakajima, "Secure and Efficient Tag Searching in RFID Systems using Serverless Search Protocol", IJSIA, vol. 2, no. 4, **(2008)** October, pp. 57-66.

[6] J. Myung, W. Lee and J. Srivastava, "Adaptive binary splitting for efficient RFID tag anti-collision", IEEE Communications Letters, vol. 10, no. 3, **(2006)**, pp. 144-146.

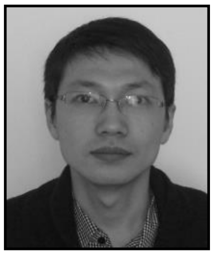
[7] K. Finkenzeller, "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification", 2nd ed., John Wiley & Sons Ltd., **(2003)**.

[8] K. Finkenzeller, "RFID Handbook:radio-frequency identificationfundamentals and applications", 2nd ed.New York, NY, USA: Wiley and Sons, **(2003)**.

[9] K. Finkenzeller, "Fundamentals and applications in contactless smart cards and identification", England: RFID Handbook, **(2003)**, pp. 1-38.

[10] S. R. Lee, S. D. Joo and C. W. Lee, “An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification”, Suwon, South Korea: Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Service, Piscataway, NJ, USA: IEEE, **(2005)**, pp. 166-172.
[11] J. Ryu, H. Lee and Y. Seok, “A hybrid query Tree protocol for tag collision arbitration in RFID system”, IEEE Inter-national Conference on Communications, **(2007)**, pp. 5981-5986.
[12] H. S. Choi, J. R. Cha and J. H. Kim, “Fast wireless anti-collision algorithm in ubiquitous ID system”, Los Angeles, CA, USA: Proceedings of the 60th Vehicular Technology Conference, Piscataway, NJ, USA: IEEE, **(2004)**, pp. 4589-4592.
[13] Z. Xie, S. Lai and P. Chen, “RFID technology and anti-collision algorithm”, Computer Engineering and Applications, vol. 43, no. 6, **(2007)**, pp. 223-225 (in Chinese).
[14] H. Meng and D. Chen, “Research and realization of RFID campus telephone system”, Application Research of Computers, vol. 26, no. 8, **(2009)**, pp. 2985-2988 (in Chinese).
[15] J. H. Choi, D. Lee and Y. Youn, “Scanning-based pre-processing forenhanced tag anti-collision protocols”, International Sym-posium on Communications and Information Technologies, **(2006)**, pp. 1207-1211.
[16] J. Guo, H. Yang and F. Deng, “Intrusion de-tection model for RFID system based on immune network”, The Journal of Computer Application, vol. 28, no. 10, **(2008)**, pp. 2481-2484 (in Chinese).

Author



YanMin Ma, master, lecturer, School of computer and information engineering, Harbin University of Commerce.