Research Journal of Applied Sciences, Engineering and Technology 5(1): 23-29, 2013
DOI:10.19026/rjaset.5.5079
ISSN: 2040-7459; e-ISSN: 2040-7467
© 2013 Maxwell Scientific Publication Corp.
Submitted: July 26, 2012 Accepted: May 18, 2013 Published: January 01, 2013

Research Article

RFID Anti-Collision Algorithm Based on Occurrence Position

[1]Mingsheng Hu, [1, 2]Zhijuan Jia, [1]Xiaoyu Ji and [3]Liu Hong
[1]Zhengzhou Normal University, Network Center, Zhengzhou 450000, China
[2]Wuhan University of Technology, College of Computer, Wuhan 430000, China
[3]Huazhong University of Science and Technology, Institute of Systems Engineering,
Wuhan 430000, China

Abstract: Aiming at the data redundancy problem in process of anti-collision in RFID, an OP(occurrence position) algorithm is proposed in this study, OP algorithm is based on the theory of direct georeference and BS (binary search) algorithm, improve traditional binary search mechanism by changing data transformed by anti-collision commands, data transformed by anti-collision commands in OP algorithm contains the same information as those data in BS algorithm, but contains less data bits than that in BS algorithm. Analysis is made for binary search algorithm used in process of anti-collision and shortcomings are given, analysis for ALOHA algorithm is also given, then principles of OP algorithm which is proposed to improve those shortcomings existing in BS algorithm and ALOHA algorithm are elaborated, analysis is also made for OP algorithm. Finally a module for OP algorithm is established on transmission delay and data redundancy, results of simulations show that OP algorithm has even higher capability than other Binary Search algorithm and has better suitability on occasion of much more tags.

Keywords: Anti-collision, binary search, occurrence position, RFID

INTRODUCTION

Radio Frequency Identification (RFID), commonly known as electronic tag, is a non-contact automatic identification technology, which taking the RF signal as the transfer medium of information and energy, so as to complete the information exchange with the measured objects (Hu *et al*., 2011). Because RFID technology has the advantages of non-contact, can identify fast moving objects, high identify efficiency, can work in harsh environments, fast and convenient operation, etc. so it is widely used in access control, logistics, tracking, monitoring, security, identity, security, military, medical and other fields and with broad prospects for development (Chen and Liao, 2010). A major advantage of RFID technology is the multi-target recognition. When the system works, around the RFID reader there may be multiple tags exist and the conflict problem occurs when more than one tag transfer data to the reader. Existing RFID anti-collision algorithms are mainly two kinds: one is non-deterministic algorithm based on ALOHA and the other is a deterministic algorithm based on Binary Search (BS). ALOHA-based non-deterministic algorithm has a fatal flaw that tag is prone to "starve" (i.e., tag with the possibility of cannot be identified), deterministic algorithm based on binary search has solved this "starve" situation, but it also with the problems such as long identification cycle, label energy consumption, etc.

Usually RFID system is composed of reader, electronic tag and computer network. Electronic tag contains information about target objects, which can be got by reader through non-contact reading and writing. Reader transmits signals it has read by controlling RF module and receives responses of tag. Then identification information of electronic tag’s object is decoded. Finally this information and other relative information on the tag are transmitted into host through computer network for processing. Collision, in the context of RFID tag reads, occurs when either several tags or several readers are simultaneously present in the same field. Tag collision occurs when a reader attempts to simultaneously read several tags. Similarly, reader collision occurs when two or more readers are simultaneously present in the same field (Wang *et al*., 2010).

FINKENZELLER K proposed the binary search tree algorithm in RFID Handbook3 (Finkenzeller, 2003), the tag sends paging according to the decoding results of collision signals, when identify a tag then return to the starting point, which is the traditional binary search tree algorithm. YUSS etc. based on binary search algorithm proposed the Dynamic Binary Search (DBS) algorithm (Yuss *et al*., 2005), greatly reducing the data redundancy of traditional BS algorithm. Number of data which is sent during process of anti-collision is the same 8 bit all the time and there is large scale of data redundancy in the process, less than 8 bit

Corresponding Author: Mingsheng Hu, Zhengzhou Normal University, Network Center, Zhengzhou 450000, China

data should be used to transform information and the data bits should be dynamic of course (Wang *et al*., 2009)

Aiming at the data redundancy problem of binary search algorithm, this study proposed a mechanism which replaces the ID with the position, for the data amount of position is far lower than ID, the reader every time REQUEST an position of the highest collision bit and when the collision of current highest collision bit is removed, the position value of the highest collision bit in next REQUEST is the position value of the second highest collision bit in last REQUEST, until the reader finished every tag identification.

PROTOCOLS FORREDUCING COLLISIONS

Binary search algorithm: The ability to identify many tags simultaneously is crucial for more advanced applications such as to check out lots of items at the supermarket. It might take a lot of time to identify them one by one, which requires an efficient identification method checking large number of goods at one time without any delay. However, if multiple RFID tags are being read synchronously, the radio signals will interfere with each other and may cancel each other's communication data. Widely used are two kinds of methods to solve this type of collision problem. One is based on a probabilistic method and the other is based on a deterministic method.

The tree-based algorithms operate by sequentially traversing the tree from top to bottom, where each node of the tree has an additional component of a tag's information. The root of the tree is common to all the tags and each leaf node of the tree represents a specific tag. These algorithms narrow down the tag of interest by successively muting tags that are involved in collisions. Drawbacks of these algorithms include the number of iterations involved before the reader can identify any given tag, the complexities of circuits required in the tag itself to be able to respond to such readers, among others. The ALOHA-type of protocols, on the other hand, let the tags choose when (i.e., time slot) they respond. When collision occurs, the tags involved in the collision participate in further iterations until resolution occurs. The ALOHA-type of protocols clearly take less time to complete, although there is a positive probability for a tag to not be read within a given number of iterations.

The main principle of the binary tree algorithm is to require electronic tag has a pointer which points to the highest bit of tag ID first and gradually moves to the lowest bit with the process of query. Reader sends a query bit 0 or 1 each time, if the bit the pointer points is the same with query bit, then tag transmit its next bit to reader, or tag stop transmitting and doesn't take part into the next query any longer. When reader does not find collision, it will take the bits it received as the next query condition, or 0 will be the next query value. Reader will operate in accordance with this process until the pointer points to the lowest bit, so each cycle reader can identify one tag. Binary tree anti-collision algorithm requires memory function of tag and has high requirement on power. Therefore, people suggested a non-memory binary search algorithm.

Step 1: The reader send order REQUEST (SNR), all tags who receive this order from reader will send their ID back to reader and reader will check out that whether there are collisions among these tags, if no collision appears the process turns to Step 6, if there are collisions emerging the process turns to Step 2

Step 2: Reader will check out the collision by xor all the received IDs number, from the result of the xor calculating reader will compute out another SNR number and this new SNR number is set at the current highest collision bit and the position changes from 0 to n-1, then reader send order REQUEST (SNR) and the process will turn to Step 3

Step 3: Reader send order REQUEST (SNR) by broadcasting, so all tags will receive the new SNR. Tags will compare the received SNR to their own IDs and if their IDs are less than received SNR they will send their IDs back to readers, else they will not do anything. And tags who send their IDs back to reader are called 0-branch tags and the others are called 1-branch tags. Then the process will turn to Step 4

Step 4: Reader will deal with tags separately after tags are divided into 0-branch and 1-branch, so reader will get two results of xor calculating, then the reader will send SNR of 0-branch and SNR of 1-branch separately to tags. If there are still collisions existing in 0-branch tags or 1-branch tags, 0-branch tags and 1-branch tags will be divided into another 0-branch and 1-branch tags separately, this principle will not finish until the entire anti-collision process is finished. Then the process come into Step 5

Step 5: If reader have recognized all the tags which means that there is no collision appearing, the process will turn to Step 6, if not, the process will turn to Step 4

Step 6: The process finish. Look at Fig. 1

Command REQUEST (SNR) represents to request a serial number; Command SELECT (SNR) represents to select a serial number; Command READ- DATA represents to read data.

The system of binary tree search algorithm used the above commands, now taking four RFID tags which in the scope of a reader as an example. Each tag has a unique serial number within 00~~FFH (equal to the decimal range of 0 to 255, or binary range of 00000000 to 11111111):

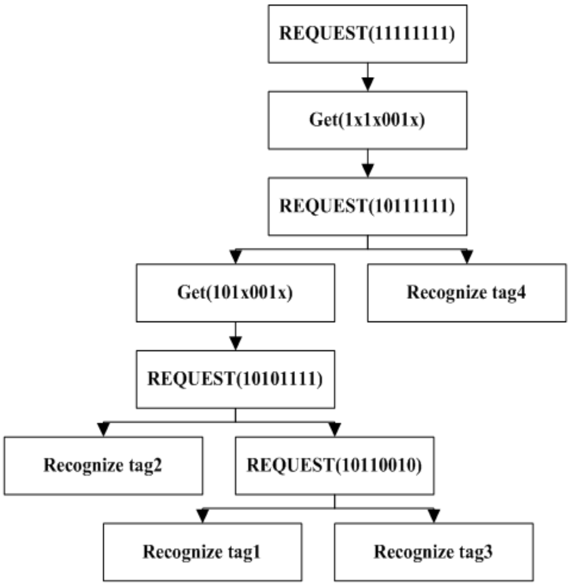

Fig. 1: BS algorithm

Electronic tag 1: 10110010; electronic tag 2: 10100011; electronic tag 3: 10110011; electronic tag 4: 11100011.

Reader conducts the REQUEST (11111111) operation, the tag whose ID is less than or equal to 11111111 need to make a response (that is, of all calls). Then the reader decodes the received ID responded by tags and finds 1x1x001x had three bits collision; reader conducts the REQUEST (10111111) operation, the tag whose ID is less than or equal to 10111111 need to make a response, then tag 1, 2 and 3 make responses and tag 4 is identified and finds 101x001x still had two bits collision; reader conducts the REQUEST (10101111) operation, tag 2 makes a response and is identified by the result of 1011001x; reader conducts the REQUEST (10110010) operation, tag1 makes a response and tag 3 does not answer and because there are only two tags so tag 1 and 3 can be identified, then every identification is completed. The identification process is shown in Fig. 1, the results of each REQUEST are divided into branches of 0 and 1 where the branch 0 is on the left and branch 1 is on the right. Look at Fig. 1.

Analysis of binary search algorithm: BS algorithm's basic idea is divided the in-collision tags into two subsets as 0 and 1, which first queries the subset 0, if there is no conflict, then identifies the tags correctly, if there is still conflict then subset 0 is divided into 00 and 01 two new subsets and so on, until the all tags in subset 0 has been identified and then follows these steps to query the subset 1. In ISO 18000-6B protocol, the process of using BS algorithm to achieve the anti-collision is: reader continue to get tags into binary fission until the tag on the left-most branch is 1 or 0 and after the reader finished the tag inventories in left branch, it keeps inventorying the tags in follow-up branches in accordance with the previous procedure and so on until every tag complete the inventory 5.

The biggest flaw is that each REQUEST data is a fixed 8-bit data, if the number of collision bits is large, the reader will issue massive REQUEST data, while a larger amount of data in wireless communications will bring great communication delay. DBS algorithm which on the basis of the BS algorithm can eliminate the front bits which have been identified and then in REQUEST operation where the sending data is the remaining unrecognized bit, so that reducing the amount of redundant data in the identification process, but there is no improvement in the nature. So there are mainly two aspects of problems existing in Binary Search algorithm:

- Readers send a number-fixed ID sequence to all the collision tags, which is a waste of data, because with the progress of anti-collision algorithm, more and more collision tags are figured out, so we can only send less information to left tags to get the same results to method of number-fixed sending. All the anti-collision process is changing dynamically all the time, so using fixed data to represent collision information is irrational obviously
- Flexible data numbers should be adopted to send information so that there are less data bits transformed from reader to tags, which reduces the burden of RFID communication
- Less data bits consume less time to process the data bits, in another side, if we reduce the data bits in anti-collision process, it will cost less time to finish the anti-collision process, as long as the RFID communication

Aloha algorithm and analysis: ALOHA was developed in the 1970s for a packet radio network at the University of Hawaii. Whenever a station has data, it transmits the data. The sender station finds out whether the transmission was successful or experienced a collision by listening to the broadcast. If there is a collision, the sender retransmits after a random wait period. This protocol involves the simplest of all reader design where the reader just listens. The tags periodically send data packets with random quiet periods. This protocol quickly adapts to varying number of tags. The theoretically proven maximum utilization with this protocol (assuming Poisson arrival) is 18.4%. Figure 2 provides an illustration of ALOHA, with four tags and a reader. Collision occurs, as illustrated by the dark rectangles at the reader side, when two or more signals are received by the reader during the same (overlapping) time period.

The slotted ALOHA protocol was an improvement where time is divided into discrete time intervals (slots)

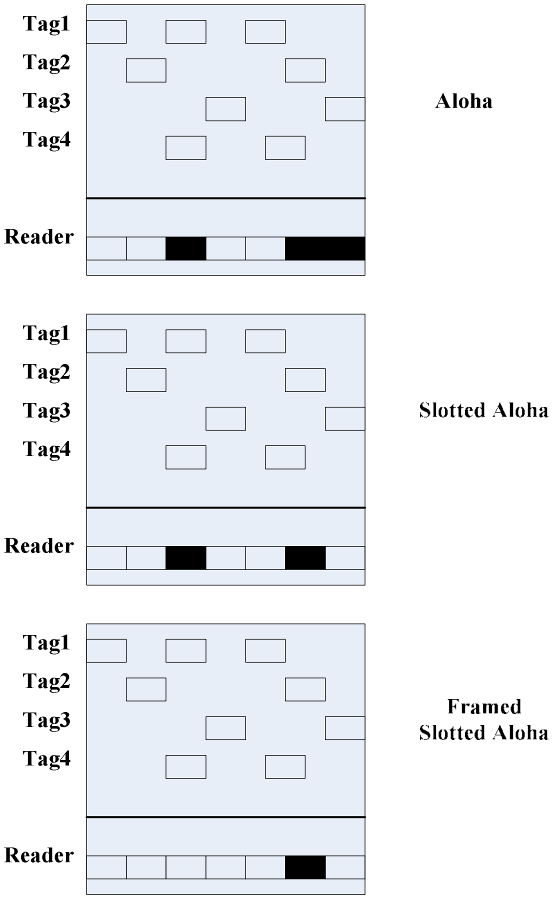

Fig. 2: Aloha algorithm

and a packet can only be transmitted at the beginning of a slot. It reduced the collision duration. Here, packets either collide completely or do not collide at all, i.e., no partial collisions like in the pure ALOHA case. This protocol doubles the channel utilization of pure ALOHA to 36.8%. Fig. 2 provides an illustration of the Slotted ALOHA protocol.

The framed slotted ALOHA protocol incorporated further discretization of time by grouping medium accesses into frames, with N slots per frame. Within a frame, tags transmit at most once in a randomly selected slot. Thus signal transmitted per tag can have at most one collision within a given frame. Initially, frames were of uniform size with the same number of slots. This was later extended to incorporate flexibility by letting the reader adaptively expand and contract the number of slots per frame in the next upcoming round. The number of slots per frame is selected as per the number of tags in the field. There is no deterministic way to handle this problem of adaptively varying the number of slots per frame at any given point in time. We address this issue in the next section.

The framed slotted ALOHA has been proposed as a means to alleviate problems associated with collisions when multiple RFID tags are simultaneously present in a reader's field. Almost all of these approaches assume that the reader first transmits its frame size to the tags. The tags then randomly pick a slot and reply in that slot. From the reader's side, in any given slot, it receives (a) no signal, (b) signal from one tag, or (c) signal from more than one tag. The slot under case (a) is considered a loss since no useful information was transmitted during that slot. The slot under case (b) is a successful case since the signal from this tag is received by the reader. Collision occurs in case (c) and the number of tags that replied in this slot is generally unknown.

It is hard to estimate the initial number of tags present in the field of the reader. The frame length depends on the number of tags. Existing papers on RFID collision (as well as ALOHA, in general) assume that the tags choose a uniform distribution to determine the slots in which to reply.

OCCURRENCE POSITION ALGORITHM

Basic principles: The previous studies could be classified into two different approaches: one is based on deterministic model and the other is probability model. The deterministic method has been discussed in some papers, like Marcel Jacomet's work (Marcel *et al.*, 1999). His algorithm used a bit-wise arbitration model during the identification phase of the target device. In Weis's paper, he showed one deterministic algorithm based on the binary-tree walking scheme. Christian Floerkemeier's research (Christian and Matthias, 2006) gave some illustrations of the sources of error by some already existing methods which deal with the collision problem. He also pointed out how to make a system more appropriate to handle this multiple identifying situation.

In traditional anti-collision algorithms of RFID, the command of REQUEST sends binary bits of tags' ID throughout, 8 stationary binary bits are sent by REQUEST in BS algorithm and less than 8 dynamic binary bits are sent by REQUEST in DBS algorithm. Binary bits of tags' ID are replaced with Binary bits of position in OP algorithm, only binary bits of position which is the position of the current highest collision bit in the whole ID sequence, command of REQUEST (ID) is replaced with REQUEST (POSITION) and the corresponding decoding mechanism should be adopted in tags, so that correct communication is established between reader and tags based on REQUEST (POSITION), which decides whether the tags should send their own ID back to reader.

Assuming there are n binary bits of tags' ID, so the position of collision bits are 0~n-1, the processes of OP algorithm are as follows:

Step 1: The reader send order REQUEST (n), all tags who receive this order from reader will send their ID back to reader and reader will check out that whether there are collisions among these tags, if no collision appears the process

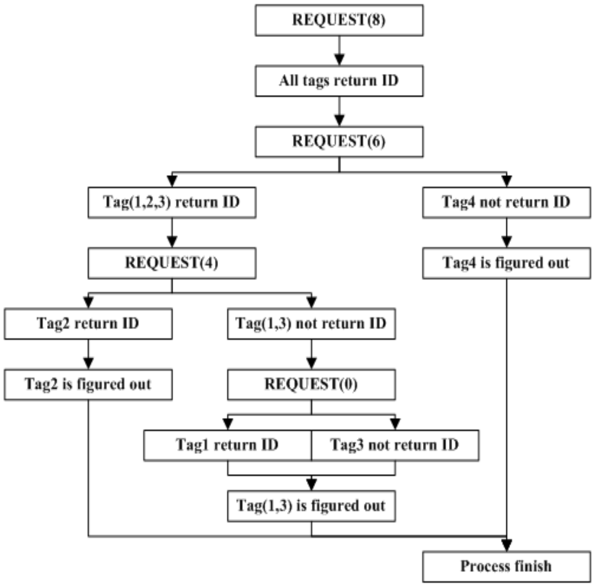

Fig. 3: Reader of OP algorithm

turns to Step 6, if there are collisions emerging the process turns to Step 2

Step 2: Reader will check out the position of the current highest collision bit and the position changes from 0 to n-1, then reader send order REQUEST (position) and the process will turn to Step 3

Step 3: Reader send order REQUEST (position) by broadcasting, so all tags will receive the order. Tags will find the corresponding position of their own ID sequences after decoding the received position, then Tags whose bit on this position is 0 will send their ID back to reader and Tags whose bit on this position is 1 will not send their ID back to reader. In this way, Tags are divided into two branches including 0-branch and 1-branch and the process will turn to Step 4; Look at Fig. 3

Step 4: Reader will deal with tags separately after tags are divided into 0-branch and 1-branch, so reader will get two positions from two branches separately, then reader will send order REQUEST (position of 0-branch) in 0-branch and send order REQUEST (position of 1-branch) in 1-branch, the process will turn to Step 5

Step 5: If reader have recognized all the tags which means that there is no collision appearing, the process will turn to Step 6, if not, the process will turn to Step 4

Step 6: The process finish. Look at Fig. 4

Analysis for occurrence position algorithm: Assuming that tags have 1 bit ID sequence, then the position of current highest collision bit will not be larger than 0th, so binary bits of position will not be larger than 0, also 1 bit data, at this time SI, DBS, BS algorithms have the same count of bits; assuming that tags have 2 bit ID sequence, then the position of current highest collision bit will not be larger than 1^{st}, so binary bits of position will not be larger than 1, also 1 bit data, at this time OP algorithm has 1 bit data, DBS and BS all have 2 bit data, so OP algorithm has 1 bit less than DBS and BS in count of data; so when tags have n bit ID sequence, then the position of current highest collision bit will not be larger than (n-1)th, so binary bits of position will not be larger than logarithmic of position based on 2, because the position is decimal number and reader will send logarithmic of position based on 2 bits of binary numbers when position is converted from decimal to binary:

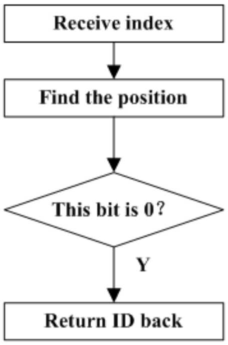

Fig. 4: Tags of OP algorithm

$$bit\_count = \begin{cases} 1, n = 1 \\ floor(\log_2(n-1)) + 1, n \neq 1 \end{cases} \tag{1}$$

Now it is necessary to prove that bit_count is smaller than n, then it is obvious that OP algorithm has less data consume than DBS and BS and when n = 1

$$bit\_count = n = 1$$

So it is necessary to prove that the Eq. (3) below is always correct when n is larger than 1. And because the condition (2) below is always correct:

$$floor(\log_2(n-1)) + 1 < \log_2(n-1) + 1 \tag{2}$$

$$\log_2(n-1) + 1 \leq n \tag{3}$$

So it is necessary to prove that Eq. (4) is permanent correct:

$$n - 1 \leq 2^{n-1} \tag{4}$$

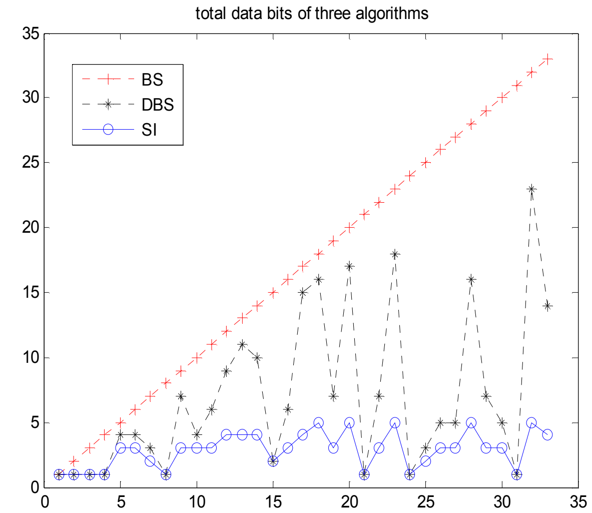

Fig. 5: Total data bits of algorithms

Because n is larger than 1, Eq. (4) is permanent correct, so bit_count is always smaller than n, we can say OP algorithm has less data consume than BS and DBS. So save of OP algorithm in data consume compared to BS and DBS.

For example, there are 8 bits binary data in ID sequences, so position in decimal is from 0 to 7, so the current highest collision bit is less than 7th bit, so the largest position is 7 and 7 is 111 when converted to binary and count of binary bits is 3, but the binary data that reader should send to tags is always 01111111 and its count of binary bits is always 8, at this time OP algorithm save 5 bits data compared to BS algorithm. If the current highest collision bit is 6th bit, so the largest position is 6 and 6 is 110 when converted to binary and count of binary bits is 3, but the binary data that reader should send to tags is always 00111111 and its count of binary bits is always 8, at this time OP algorithm save 5 bits data compared to BS algorithm. Similarly, if the current highest collision bit is 0th bit, so the largest position is 0 and 0 is 0 when converted to binary and count of binary bits is 1, but the binary data that reader should send to tags is always 00000000 and its count of binary bits is always 8, at this time OP algorithm save 7 bits data compared to BS algorithm, the data save is considerable.

The improved algorithm OP is always sending less data bits than BS algorithm, but it still adopt the method of binary search, so sending times in OP algorithm is the same as BS and DBS algorithm, then all the data bits sent in entire anti-collision process of OP algorithm is less than that of BS and DBS algorithm. Additionally, mechanism of occurrence position is adopted in OP algorithm, an position of current highest bit is sent by reader every time, if all the positions are sent by reader at one time, tags who receive these positions can only find all the positions that collision bits exist in ID sequence, in fact we only need the position of current highest collision bit, other positions of collision bit which is not the highest is redundant, so it is fair to send only the position of current highest collision bit at one time.

SIMULATIONS

Simulation is built based on Matlab6.5 environment and analysis and compare are made in aspects of data redundancy and transmission delay among SI, DBS and BS algorithm, process of mathematical modeling is as follow:

Variable 1: Count of bit in tags' ID sequence. SI, DBS and BS algorithms, in these three algorithms, sending times of readers are all the same and the total data bits are all deal with sending times and data bits sent by reader every time, so factors which really affect the total data bits are count of bit in tags' ID. Assuming that count of bit in tags' ID is n, n changes from 1 to n, it is counted as an x-label variable;

According to the features of ISMR, inspired by the metasynthesis, the ways and means of achieving comprehensive and integrated are concluded as follow.

Variable 2: Total data bits. When count of bit in tags' ID sequence changes, there are big differences in the total data bits of these three algorithms, count of bit in tags' ID is defined as n and the current highest collision bit is defined as k, so k changes from 0 to n-1, then there are always confirm n bits data being sent by reader every time in BS algorithm from the analysis before; there are confirm k bits data being sent by reader every time in DBS algorithm and k randomizes from 0 to n-1; there are floor(log2(k))+1 bits data being sent by reader every time in OP algorithm, so count of total data bits changes as a logarithmic curve. Look at Fig. 5.

Variable 3: Transmission delays as shown in Fig. 6. Assuming that there are m tags in the region of reader and there are x collision bits existing in recognition, so the count of collision bits which is decoded by reader from tags at the first time, its range is [integer(lbn),k]6. System clock cycle is defined as T and current highest collision bit is defined as m which changes from 0 to k, only one time of paging need to be compared because these three algorithms have the same times of paging, transmission delays of BS, DBS and OP algorithms are as follows:

BS algorithm:

$$\tau = k*T + k*T = 2kT$$

DBS algorithm:

$$\tau = m*T + k*T = (m+k)T, m \le k$$

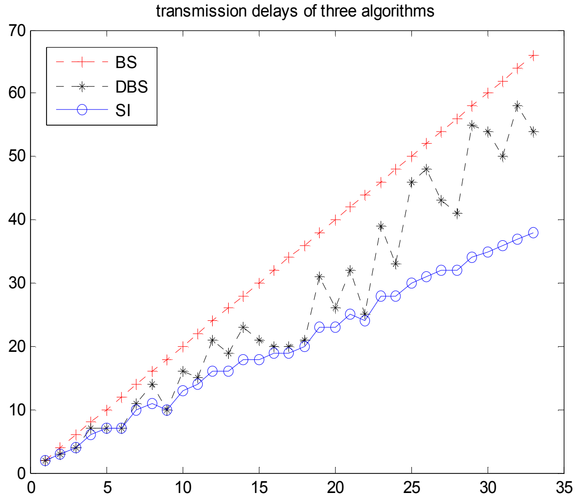

Fig. 6: Transmission delays of algorithms

OP algorithm:

$$\tau = floor(\log_2(m)+1)*T + k*T$$

$$= floor(\log_2(m)+k+1)T$$

CONCLUSION

The traditional BS algorithm is with large volume of data and long transmission delay, which seriously affected the real-time and reliability of RFID anti-collision process. In this study, the OP algorithm changed the communication mechanism between the reader and tag by using the position of the highest collision bit as the only data sent in anti-collision process, in order to minimize the data generated by the anti-collision process. Finally, the OP algorithm, the BS algorithm and the DBS algorithm were analyzed and compared by the performance parameter of data redundancy and transmission delay, simulation results show that the transmission delay and data redundancy of the OP algorithm are the smallest.

ACKNOWLEDGMENT

This study is sponsored by the National Natural Science Foundation of China (NSFC, Grant 60903174), the Key Scientific and Technological Project of Henan Province (122102310004), the Fundamental Research Funds for the Central Universities (HUST: 2012QN087, 2012QN088) and the Innovation Scientists and Technicians Troop Construction Projects of Zhengzhou City (10LJRC190).

REFERENCES

Christian, F. and W. Matthias, 2006. Comparison of transmission schemes for framed ALOHA based RFID. Proceedings of the International Symposium on Applications on Internet Workshops SAINT-W '06, pp: 92-97.

Chen, Z. and M. Liao, 2010. Rapid anti-collision algorithm for RFID. J. Comput. Appl., 30: 18-20.

Finkenzeller, K., 2003. RFID Handbook. 2nd Edn., John Wiley and Sons, England.

Hu, M., L. Guo Ning, 2011. Anti-Collision algorithm for RFID based on dynamic packet query tree. Trans Tech Publicat., 159(2011): 550-555.

Marcel, J., E. Adrian and G. Urs, 1999. Contactless Identification Device With Anticollision. IEEE Computer Society CSCC99, Conference on Circuits, System, Computers and Communications, Athens.

Wang, C., J. Sun and Y. Zhang, 2009. Research on anti-collision algorithm for RFID. IEEE T. Ind. Electron., 6: 49-54.

Wang, X., Z. Qian and Z. Hu, 2010. Research on anti-collision algorithm based on binary tree for RFID. J. Communicat., 31: 49-56.

Yuss, Z., Z. Wang and Z. Tang, 2005.Anti-collision algorithm based on jumping and dynamic searching and its analysis. Comput. Eng., 31: 19-20.