An Investigation on Tree-Based Tags Anti-Collision Algorithms in RFID

Su Xiaohao [1] , Liu Baolong [2]

[1] School of Computer Science and Engineering, Xi'an Technological University,

Xi'an 710021, China, Email:529489890@qq.com

[2] School of Computer Science and Engineering, Xi'an Technological University,

Xi'an 710021, China, Email:liu.bao.long@hotmail.com

Abstract. The tree-based tags anti-collision algorithm is an important method in the anti-collision algorithms. In this paper, several typical tree algorithms are evaluated. The comparison of algorithms is summarized including time complexity, communication complexity and recognition, and the characteristics and disadvantages of each algorithm are pointed out. Finally, the improvement strategies of tree anti-collision algorithm are proposed, and the future research directions are also prospected.

Keywords: Radio Frequency Identification, Anti-collision algorithm, Electronic tags, Deterministic algorithm

1. Introduction

With the rapid development of the Internet of things technology, Radio frequency identification (RFID) technology has been used more and more in life and production. RFID technology is an important part of the Internet of Things system. It is a non-contact intelligent identification technology. Compared with the traditional automatic identification technology, it has many advantages such as small size, low cost, large amount of data storage, high security and reusable. It has great application value in industrial production, logistics and transport and cargo management, commodity trade, health and safety inspection and intelligent transportation and other fields. RFID system mainly consists of reader (including antenna), the application system and a large number of electronic tags [1-3] as shown in Fig.1. Label is mainly used to store the information encoded by the marked object and security encryption, the reader used to read, change and verify the label information. However, there are several problems in the application of RFID system.

1.1 Collision problem

When multiple tags and readers in the same channel and signal transmission, collision problem is generated because of mutual interferences between tags and readers.

1.2 Security authentication and privacy protection

The security problem is how to ensure the authenticity and validity of the label information and the reader when there are eavesdropping attacks and replay attacks.

1.3 Efficient storage of large amounts of data

RFID devices in real-world applications continue to generate large amounts of data. How to build an efficient data storage model to reduce system overhead and improve query efficiency is an urgent problem to be solved.

1.4 Precise target location and tracking

How to obtain the accurate position information of the location object in outdoor and indoor environment will be a hot research field in the future.

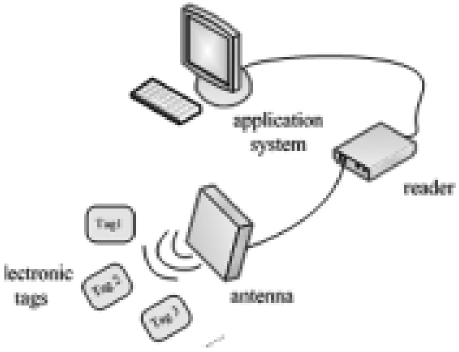

Figure.1 Diagram of RFID system composition

The collision problem is a key problem affecting the performance of RFID identification, which restricts the recognition efficiency in large-scale label recognition environment. When multiple readers or multiple tags at the same time, the same frequency to send data, the data signal will interfere with each other in the wireless channel, resulting in data loss, missing, misreading and other phenomena [4]. According to the different reasons of the collision, it can be divided into three categories: tag-tag collision, tag-reader collision and reader-reader collision [5]. As for tag-tag collision problem, the anti-collision algorithm has been divided into three categories [6-10]: (1) Non deterministic anti-collision algorithms based on ALOHA; (2) Deterministic anti-collision algorithms based on tree structure; (3) Hybrid algorithms.

This paper summarizes the tag-tag anti-collision algorithms based on tree structure. Some of the typical algorithms at present are analyzed and compared for algorithm on time complexity, communication complexity and recognition efficiency. The way to improve the existing algorithm is discussed, and the future research directions are points out. This provides reference for the research on anti-collision algorithms the structure of the tree.

2. Reviews of Several Typical Tree-Based Anti-Collision Algorithms

The tree-based anti-collision algorithms transform the label number to a binary string consisting of ‘0’ and ‘1’. When a collision occurs, it is divided into “0” and “1” branches according to the label number. In each branch, the algorithm repeats the query process until it can correctly identify a label. In the end, a tree structure is built. Compared to ALOHA algorithm for random reading of labels, tree-based algorithm can avoid the label hunger phenomenon. In this section, we have selected several tree-based algorithms with the most research at present, and analyzed the implementation process of the algorithm and the latest research progress。

2.1 BT (Binary Tree) Algorithm

The binary tree algorithm (BT) is a basic binary anti-collision algorithm, in which each tag contains a counter and a random number generator [11].When the counter value is 0, the label sends the tag information. When the counter value is not 0, the label is in a wait state and does not respond, and the value of each counter is 0 in initial state. The algorithm steps are as follows.

1) After the reader sends the query command for the first time, all the tags responses within the recognition range and sent the respective tag numbers.

2) If a collision occurs, each tag with a counter value of 0 randomly generates 0 or 1 and assigns it to the counter. If the counter value is greater than 0, the counter value is incremented by 1.

3) If no collision occurs, it indicates that a label has been identified or a free node is generated, and the counter value for all tags is decremented by one.

4) Cycle the step 2 until all the tags are identified.

Fig. 2 shows an example of using the BT algorithm to identify tags. With the increasing of the number of tags, the number of queries increases dramatically. This leads to a large number of idle time slots and reduces the efficiency of the system recognition. Otherwise, the BT algorithm needs to update the value of the tag counter for each identification cycle with requiring the tag to have storage and counting function [12]. As the result, it increases the overhead of the system. In [13], the ABS algorithm was proposed based on the BT algorithm. Each tag in the algorithm has two counters: the reader’s allocates slot counter Rc and the label’s progress slot counter Pc [14]. These two counters and random numbers make the tags identically identifiable, but the algorithm is designed to be more complex. There are still idle slots in the recognition process. The value of the counter is also stored every time, which increases the time complexity of the algorithm. Jiang et al [15] introduced a fallback strategy and search tree algorithm to improve the ABS algorithm. The algorithm eliminates idle time, and reduces the identification delay, but it does not reduce the system overhead because each query still needs to change the value of the counters.

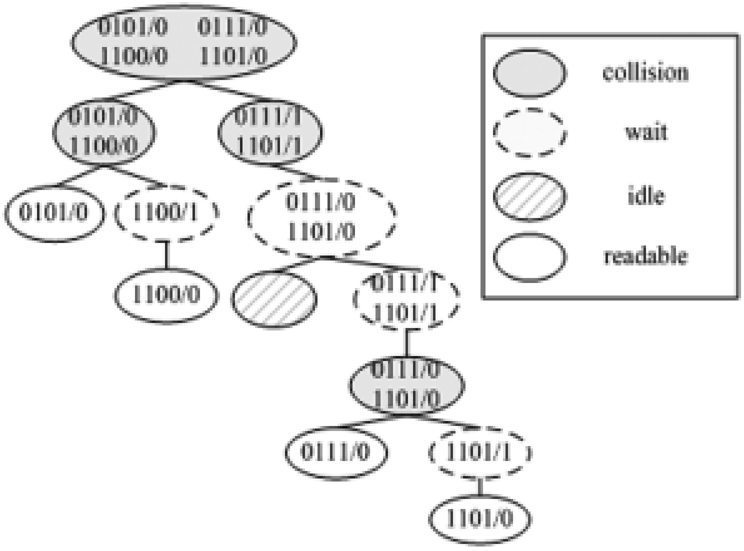

Figure.2 An example diagram for BT algorithm

2.2 BS (Binary Search) Algorithm

Binary search tree anti-collision algorithm (BS) is a memory less classic tree-based anti-collision algorithm [16]. In the algorithm, the label does not need to store any information, when the collision occurs, the reader can change the query string to identify labels. The algorithm steps are as follows.

1) The reader sends the query command to tags in binary string. The query string length is equal to the label number length and the initial value is composed of all ’ 1’ . All tags to be identified in the recognition range response and send the numbers to the reader.

2) If no collision occurs, there is only one tag identified. If the collision occurs, the reader will set query string’ s the highest collision position ‘0’ and the encoding higher than the highest collision bit unchanged, the highest collision to the lowest are set to “1” to form a new query string.

3) The reader broadcasts a new query string, looping the step 2. When a tag is recognized, the reader reads its number and sends the command to the “dormant” state.

4) Restart from step 1 until all tags are identified.

The binary search algorithm (BS) eliminates the idle time slot and does not need to store additional data. However, each recognition after the query string will start from the initial state, and it increases the number of queries. At the same time, when the reader sends a binary string, the tag needs to send a complete binary string. In fact, the same part of the binary string sent by the reader does not need to be sent to the reader. Given the three labels A, B, C {110100101011011010101101}, using the BS algorithm to identify the process of these three labels are shown in table 1.

Table 1 Binary search algorithm identification table

	First search	Second search	Third search
Reqest	11111111	10111111	10101111
Label A	11010010	——	——
Label B	10110110	10110110	——
Label C	10101101	10101101	10101101
Collision bit decoding results	1×××××××	101××1××	
Identified label	null	null	C

The improved algorithm based on BS algorithm includes dynamic binary search algorithm (DBS), dynamic back off binary search algorithm [20] and so on. Compared to the BS algorithm, the DBS algorithm has the same number of queries, but the data between the label and the reader is reduced by half and the query efficiency is improved. Dynamic binary back off algorithm is improved on the basis of the DBS algorithm, which reduces the number of queries while ensuring that DBS transmits less data. On the basis of BS algorithm, the BS - BLG algorithm is proposed by Di Chunyu from Jilin University [21], whose main improvement is to increase the lock bit group and group paging instructions. The role of the lock bit group is to extract the bits that produce the collision and to prioritize groups according to the number of consecutive ‘1’ . It reduce the number of queries by grouping paging and use backward strategy to reduce the number of queries. BLBO algorithm is proposed on the basis of BS and DBS algorithm [22]. In this algorithm the collision bit is locked by the lock strategy, and then the reader only sends the collision bit information, which reduces the communication complexity. By using the backward strategy, after each tag is identified, it is returned to the node where the collision occurred, and the recognition efficiency of the BLBO algorithm is close to 50%. In [23], the NBLBO algorithm is proposed on the basis of BS algorithm and BLBO algorithm. The algorithm introduces the idea of adaptive bifurcation, which adjusts the next query prefix by calculating the collision factor after each collision. With the same number of tags, the recognition efficiency of the algorithm is about 3 times of that the BLBO algorithm and it also has a significant reduction in the amount of data transmission and the number of queries.

2.3 QT (Query Tree) Algorithm

The QT Algorithm [2, 24-26] is also a memory less algorithm， the tag only needs to be compared to the query prefix sent by the reader each time, and the tag is sent back to the reader with the prefix. The algorithm is first introduced into the stack to save the query prefix to improve the query efficiency. QT algorithm steps are as follows.

1) In the initial stage of the query, an empty string is pushed onto the stack, and the reader sends an empty string. At this time, all tag responses within the scope of the identification.

2) If there is a collision, the ‘0’ and ‘1’ followed by the last prefix are added to form a new query prefix, and the two new prefixes are pressed into the stack. If there is no collision, a unique tag is identified.

3) If the stack is not empty, popping a query prefix of the stack. After receiving the query command, the tag matching the query prefix returns the code of the remainder. If there is no tag response, an empty cycle will occur, and a query prefix will be popped.

4) Repeat the above 2, 3 steps until the stack is empty and all tags are identified.

QT algorithm is simple and easy. It has no additional storage requirements for the tag, and the hardware cost is low. However, the QT algorithm is greatly affected by the label distribution, it does not prejudge the location of the collision of the label, and the update of the query prefix is more mechanical. It will produce a large number of idle time slots in the process of identification. The following table lists the number of idle time slots in the case where the label length is 12 bits and the number distribution is a random distribution. The number of tags increases from 100 to 600.

Table 2 Slot table for QT algorithm

Number of tags	Query time slot	Idle time slot
100	289	45
200	543	71
300	785	93
400	1031	116
500	1283	142
600	1512	157

It can be seen that as the number of tags increases, the free time slot increases. The query efficiency is reduced, increasing the reader’s energy consumption. But the QT algorithm still inevitably produces a large number of free time slots and collision time slots. In the literature [27], the MBQT algorithm is proposed based on the QT algorithm. The label is changed from binary code to multiple coding, which reduces the amount of transmission data in the communication process and improves the recognition efficiency. But the algorithm needs to use the reader more storage space, and the performance of the algorithm is also affected by the length of the tag. In [28], a hybrid query tree algorithm is proposed. The algorithm changes the query prefix according to the collision information in each recognition cycle. Compared with the QT algorithm, the algorithm reduces the number of queries and increases the recognition efficiency to about 59%. But the algorithm still does not solve the problem of idle time slot. Zhou Qing proposed the IHQT algorithm based on the QT algorithm [29] which sets a full adder in each tag. This algorithm uses the number of three consecutive “1” after the query prefix to determine the order of the response to the reader. This method avoids the generation of idle time slots and reduces communication complexity.

2.4 CT(Collision Tree) Algorithm

CT algorithm is an improved algorithm in QT algorithm, which was proposed by Dr. Jia Xiaolin in 2012 [12]. Unlike the QT algorithm, the CT algorithm only updates the query prefix for the collision position, reduces the unnecessary query, and eliminates the idle time slot. In the CT algorithm, first, the initial reader sends a query instruction to labels. if there is no label collision, then directly identify, or the reader update the query prefix according to the first collision bit, that is, all the bits before the first collision bit unchanged, Followed by the addition of “0” and “1” .Thus, two new query prefix are pushed onto the stack. The reader sends the query prefix at the top of the query stack, and labels compare their own number with the received query prefix. If the same, labels will remove the same part of the prefix sent to the reader, the reader again to judge until the stack is empty so far.

The CT algorithm completely eliminates the idle cycle in the query process, and does not need the information in the process of memory tag recognition. The recognition efficiency is more than 50%. On the basis of CT algorithm, a multi-period identification algorithm (MCT) [30] is proposed, which divides the identification cycle of communication between reader and tag in RFID multi-tag identification into three sub-periods (Q, R0, R1).In the algorithm, labels determines the flag bit according to the query command, and select the response cycle (R0 or R1) according to the value of the flag bit to respond to the reader’ s query request. But in this way, each time the reader sends the request, the label must judge the flag bit, according to the flag bit to divide the response cycle, so that the communication efficiency of the tag and the reader is greatly reduced. In paper [31], the CT algorithm is improved by using the double prefix and collision bit continuity, and the DPPS algorithm is proposed. Compared with CT algorithm, the average recognition efficiency is improved by 36%, and the average communication complexity is reduced by 35.6%.

In summary, although the research on the anti-collision algorithm has made some achievements in recent years, there are still some problems such as the complexity of the algorithm design and the inability to adapt to the practical application. The algorithm identification efficiency is affected by the large distribution of labels and so on. An efficient and stable anti-collision algorithm is the focus of future research.

3. Comparison of Several Algorithms

At present, most of the references for the improvement of tree based anti-collision algorithm focus on the BS algorithm, BLBO algorithm, QT algorithm and CT algorithm. In this section, we compare the performance of these algorithms with experimental analysis

3.1 Performance Analysis

Table 3 shows the performance parameters of the four algorithms. Where, n is the number of tags to be identified, k is the number length of the tag, x is the number of bits in which the tags collide, l_{pre} is the length of the query prefix sent by the reader, n_{res} is the tag that responds to the reader in each query cycle number.

Table 3 Algorithm performance comparison

Anti-collision algorithm	Time complexity	Communication complexity	Recognition efficiency
BS algorithm [32,33]	$n+\log_2(n!)$	$2k*(n+\log_2(n!)$	$1(1+\log_2(n!)/n)$
QT algorithm [33,34]	$n*(k+2-\lg n)$ (The worst situation)	$(n+1)(2.21k*\log_2 n+4.19k)$	$1(k+2-\lg n)$ (The worst situation)
BLBO algorithm [22]	$2n-1$	$3k-2x-21+(2x+44)*n$	$1(2-1/n)$
CT algorithm[35,36]	$2n-1$	$(2n-1)*[l_{pre}+(k-l_{pre})*n_{res}]$	$1(2-1/n)$

As can be seen from the above table, the time complexity of the BS algorithm is $n+\log_2(n!)$. Due to the existence of $n!$, the time complexity and communication complexity increase rapidly with the increase of the number of labels and the length of the tag number, and the recogniti only the most complicated case, but it can be seen that the QT algorithm is greatly influenced on efficiency will be lower and lower. Obviously, there is still much room for improvement in the BS algorithm. The time complexity of the QT algorithm listed in the above table isby the tag number length. When the tag number is long, the query range of the reader is greatly increased, which increase the number of free time slots. Normally, QT algorithm recognition efficiency is about 34% [12]. The BLBO algorithm and the CT algorithm have the same time complexity, which determines the recognition efficiency of the two algorithms is the same. Because the value of n is greater than or equal to 1, the recognition efficiency is above 50% and the time complexity is only affected by the number of tags, regardless of the length of the label number, so these two algorithms are more stable algorithm for large-scale label recognition occasions.

3.2 Simulation Experiment

The number of queries and the number of bits transmitted are important indicators that reflect the complexity of time and communication complexity. In order to compare the performance of these algorithms more intuitively, the MATLAB simulation tool is used to simulate the query times, communication transmission bits and recognition efficiency of several algorithms. Among them, the number of labels increased from 100 to 600 and the increment is 100.The label number is randomly distributed, and the number is 12 bits. The simulation results are shown in Fig.3 to Fig.5.

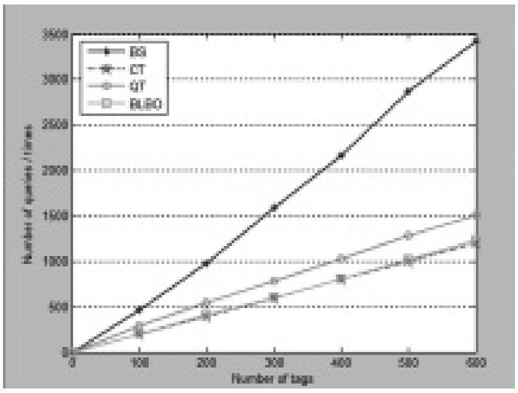

Figure.3 Comparison of query times

It can be seen from Fig.3, BS algorithm in the case of increasing the number of tags, the reader needs to send the most queries, we can see that the BS algorithm is most suitable for use in large-scale label environment, and the number of queries of QT algorithm is about half that of BS algorithm, and the performance of BLBO algorithm is similar to that of CT algorithm, which is better than QT algorithm.

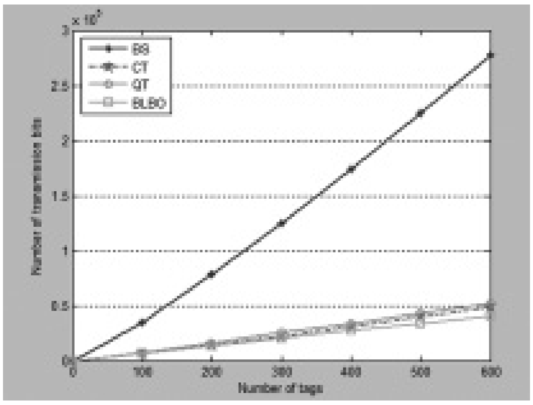

Figure.4 Comparison of transmission bit number

In Fig.4, we can see BLBO algorithm and CT algorithm’ s recognition efficiency is about 50%, and with the increase of the number of tags, the performance of the algorithm tends to be stable. QT algorithm’ s recognition efficiency is between 35% and 40%. BS algorithm’ s recognition efficiency is the lowest and with the increase of the number of tags, the recognition efficiency is gradually reduced.

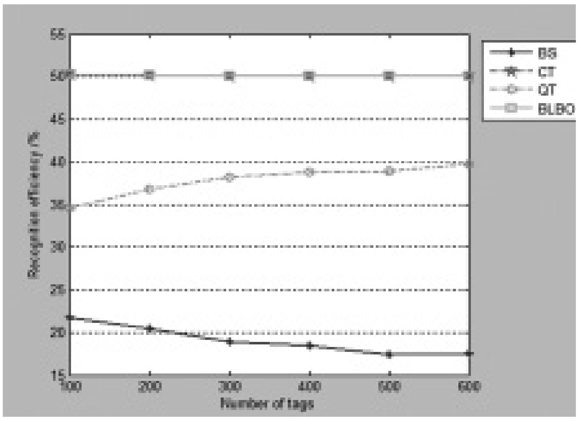

Figure.5 Comparison of recognition efficiency

The number of bits in the transmission between the reader and tags are represented in Fig.5 when use the four algorithms to identify tags. It can be seen from the figure that the BS algorithm the BS algorithm performs the worst, and with the increase in the number of tags, the amount of data needed to grow faster and faster, increasing the cost of the system. QT, CT and BLBO algorithm transmission bits are significantly lower than the BS algorithm. Among them, due to the introduction of the lock back strategy in the BLBO algorithm, it performs the best, followed by CT and QT algorithm.

4. Improvement Strategy of Tree-based Anti-collision Algorithm

In recent years, the research on RFID multi-label anti-collision algorithm has been paid more and more attention by many scholars, and many improvement strategies have been put forward for the anti-collision algorithm of tree, and some research results have been obtained. On the basis of reviewing the relevant literature, this paper summarizes several relatively typical improvement strategies.

4.1 Bit locked

In the tree-based anti-collision algorithm, if the label collides, then only the collision code is not known. In this way, locking the collision bits when collision occurs, and then the reader only sends collision bit information, the tags will only return the collision bit information, which eliminate redundant data. As shown in Figure 6, after the collision, the reader’ s decoding information is 1×0××000.Then the next time, the reader only request the 2, 4, 5 bit to send coded information, so whether the label side or the reader will reduce the information Storage to improve communication efficiency.

In [37], the DBS algorithm is improved by using the method of bit locking, which makes the new algorithm have great improvement in transmission delay and tag energy consumption. In the literature [38], a multi-collision joint lock-bit dynamic adjustment algorithm is proposed. The algorithm uses the highest collision bit and the lowest collision bit to form the lock bit paging instruction. It is preferable to estimate the number of labels in the collision slot, and then to detect the number of collision bits, and dynamically select the algorithm to deal with the collision. The instruction cost of the new algorithm is 12% and 39% lower than that of DBS algorithm and QT algorithm.

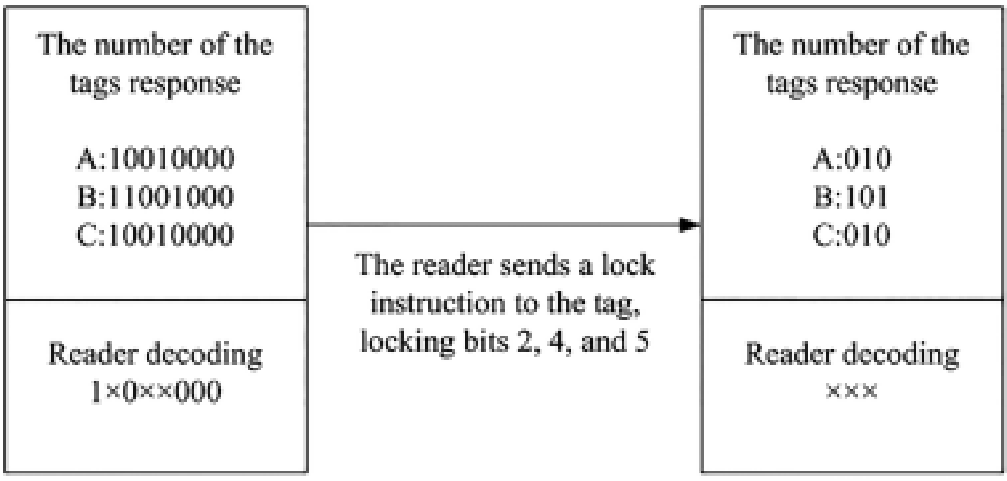
The number of the tags response
A:10010000
B:11001000
C:10010000
Reader decoding
1×0××000
The reader sends a lock instruction to the tag, locking bits 2, 4, and 5
The number of the tags response
A:010
B:101
C:010
Reader decoding
×××

Figure.6 Lock bits strategy diagram

4.2 Adaptive bifurcation

The existing bifurcation of several tree-based anti-collision algorithms is only binary, which leads to the depth of the tree is too deep and increases the number of queries. If the label set is allocated on a number of sub trees, it can effectively reduce the query tree depth and tag collision probability [22]. Quadtree, octree and other tree bifurcation algorithms are often used to reduce the collision time slot in the tag identification process, accordingly, the number of free time slots increase. Some researchers put forward adaptive bifurcation algorithms to reduce idle time slots. Mr Ding Zhiguo proposes an adaptive anti-collision algorithm. The collision factor is introduced into the algorithm, and the collision factor is the ratio of the collision bit to the label response bit in the collision slot [39]:

$$\mu = \frac{n_c}{n} \tag{1}$$

When the label collides, the collision factor is calculated. When $\mu < 0.3$, it select the binary tree to generate two query prefixes according to the first collision bit; when $\mu \geq 0.3$, it use quadtree to generate four query prefixes according to the first two collision bits. In order to avoid the use of quadtree to generate idle time slots, the literature [40] proposed a method that the reader send detection commands to determine the specific value to optimize the quadruple query prefixes. Instead, Ren [41] proposes a method that uses the XOR operation to eliminate the idle time slot before the determination of the quadruple query prefixes. Wang [42] first use the MLE algorithm to estimate the number of tags, and then use the proposed CBGN (Collision Bit Tracking Tree Algorthm Based on Grouping N-ray) algorithm to deduce the optimal grouping coefficients under different trees. The new algorithm eliminates the idle time slot and reduces the communication complexity significantly.

4.3 Backward

In the traditional BS algorithm, after successful identification of a tag, it is necessary to return to the root node to get the query command. After identifying a label in backward strategy, it will fall back to the label node’s parent node or other non-root node, which can greatly reduce the number of searches, but also can reduce the idle time slot and collision time slot Number of. the most typical algorithm is the intelligent traversal algorithm- STT（Smart Trend-Traversal） [43] proposed by Pan et al. The algorithm takes a preorder traversal node in the collision time slot, and the idle slot takes the mechanism of retreating to the upper layer node, which effectively reduces the number of collisions and the idle time slots. In [44], an enhanced STT algorithm is proposed, which can adaptively adjust the length of the query string according to the last query. Compared with the STT algorithm, the total number of queries is reduced by about 6%. In the paper [45], the backward paging is added to the query instruction of the reader, and the label information is divided into four parts, which reduces the unnecessary backward query path and reduces the amount of communication.

5. Conclusion

The tree-based anti-collision algorithm has the advantages of stability and high recognition efficiency, but there are some shortcomings such as large amount of communication data between reader and tag, high system delay and large influence on the label number distribution. However, the algorithm based on Aloha has strong adaptability and low system overhead, so the hybrid anti-collision algorithm based on tree algorithm and Aloha algorithm will be a hot research topic in the future. In addition, the label recognition for mobile state and capture effect is also the future research direction.

Sponsors or Supporters

This work is partially supported by Science & Technology Program of Weiyang District of Xi’an City with project “201609” .

References

[1]Yuan-Cheng Lai , Ling-Yen Hsiao, Bor-Shen Lin. An RFID anti-collision algorithm with dynamic condensation and ordering binary tree[J].Computer Communications,2013,36:1754-1767

[2] Xin-Qing Yan , Yang Liu , Bin Li, et.al. A memoryless binary query tree based Successive Scheme for passive RFID tag collision resolution[J].Information Fusion,2015,22:26-38

[3] Filippo Gandino,Renato Ferrero,Bartolomeo Montrucchio,et al.Probabilistic DCS:An RFID reader-to-reader anti-collision protocal[J].Journal of Network and Computer Applications,2011,34:821-832.

[4] Wu Nan. Research on RFID anti-collision algorithm based on a tree structure [D]. Changchun: Jilin University, 2014

[5] Yue Keqiang.RFID multi tag anti-collision algorithm research and application [D]. Hangzhou: Zhejiang University, 2014

[6] Overview of multi-tag anti-collision algorithms in RFID[J]. Computer Integrated Manufacturing Systems, 2014,20 (2): 440-451.

[7] Haidar Safa,Wassim El-Hajj,Christine Meguerditchian. A distributed multi-channel reader anti-collision algorithm for RFID environments[J].Computer Communications,2015,64:44-56.

[8] Yuan-Cheng Lai , Ling-Yen Hsiao, Bor-Shen Lin. An RFID anti-collision algorithm with dynamic condensation and ordering binary tree[J].Computer Communications,2013,36:1754-1767.

[9] Journal of Xidian University (Natural Science Edition), 2013,40 (6): 162-167 .Research on Adaptive Anti-collision Algorithm Based on Adaptive Multidimensional Dimensional Isolation [J]. Journal of Xidian University (Natural Science Edition), 2013,40 (6): 162-167.

[10] Klair K,Chin W,Raad R. A survey and tutorial of RFID anti-collision protocols[J].IEEE Communications Surveys and Tutorials,2010,12(3):400-421.

[11] Wang Yu .Research on RFID anti-collision algorithm [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2011.

[12] Jia Xiaolin. Research on Anti-collision Algorithm Based on collision tree for RFID multi label recognition [D]. Chengdu: Southwest Jiao Tong University, 2013.

[13] J.Myung,W.Lee,J,Srivastava,et al.Tag-splitting:adaptive collision arbitration protocols for RFID tag identification[J].IEEE Transactions on Parallel and Distributed System,2007,18(6):763-775.

[14] Mingxing He,Shi-Jinn Horng,Pingzhi Fan,et al.A fast RFID tag identification algorithm based on counter and stack[J].Expert System with Application,2011,38:6829-6838.

[15] Jiang An, WU Jixiong, Huang Shengye, et al. Improved RFID binary search anti-collision algorithm [J]. Computer Engineering and Applications, 2009,45 (5): 229-235.

[16] Chen Feiyu.Improvement of Anti-collision Algorithm Based on Dynamic Binary [D]. Shanghai: Shanghai Jiaotong University, 2013.

[17] Sarkar S, Kak AC, Rama GM. Metrics for measuring the quality of modularization of large-scale object-

oriented software. IEEE Trans. on Software Engineering, 2008,34(5):700-720.

[18] Mustapha Djeddou,Rafik Khelladi,Mustapha Benssalah.Improved RFID anti-collision algorithm[J]. International Journal of Electronics and Communications(AEÜ) ,2013,67:256-262.

[19] Chen Chong, XU Zhi, HE Minghua.Study on a New RFID Anti-collision Algorithm [J]. Journal of Fuzhou University, 2009,37 (3): 367-371.

[20] Li Qingqing.Research on RFID anti-collision algorithm [D]. Nanchang: Nanchang Aviation University, 2012.

[21] Di Chunyu.Research on RFID Anti-collision Algorithm Based on Search Tree [J]; Changchun: Jilin University; 2014.

[22] Wang Xue, Qian Zhi-hong, HU Zhengchao, et al.Study on RFID anti-collision algorithm based on binary tree [J] .Journal of Communications, 2010,31 (6): 49-56.

[23] Liu Xiao-hui.Study on RFID anti-collision algorithm based on binary number [D]. Changchun: Jilin University, 2015.

[24] Shuen -Chih Tsai,Yu-Min Hu,Chen-Hsun Chai et al.Efficient tag reading protocol for large-scale RFID systems with pre-reading[J]. Computer Communications,2016,88:73-83.

[25]Zhang Bo. Research on Anti-collision Technology of CFD [D]. Shanghai: Donghua University, 2010.

[26] Yang,C N,Lin C C.Two couple-resolution blocking protocols on adaptive query splitting for RFID tag identification[J].IEEE Transactions on Mobile Computing,2012,11(10) :1450 — 1463.

[27] Jiao Chuan-hai,Wang Ke-ren.Multi-branch query tree protocol for solving RFID tag collision problem[J].The Journal of China University of Posts and Telecommunications,2008,15(4):51-54.

[28] Jiang Wu, Yang Heng-xin, Zhang Yun.An improved Anti-collision Algorithm for Query Tag RFID Tag [J]. Computer Technology and Development, 2015,25 (2): 86-89.

[29] Zhou Qing, Cai Ming.Improved RFID Hybrid Query Tree Anti-collision Algorithm [J]. Computer Engineering and Design, 2012,33 (1): 209-213.

[30]Jia Xiaolin,Feng Quanyuan,Lei Quanshui.Multi-cycle Collision Tree Algorithm for RFID Tag Identification[J]. Journal of Southwest University of Science and Technology, 2014,29 (1), 39-44.

[31] Jian Su,Zhengguo Sheng,Guangjun Wen.A Time Ffficient Tag Identification Algorithm Using Dual Prefix Probe Scheme(DPPS)[J].IEEE Signal Processing Letters,2016,23(3):386-389.

[32] He Na.Study on anti-collision algorithm in radio frequency identification system and its hardware implementation [D]. Guangzhou: Jinan University, 2007.

[33] Klaus Finkenzeller,RFID Handbook[M].Hanser Verlag,Munich,March 2003.

[34] Shi Hao. A Hybrid RFID Anti-collision Algorithm [D]. Nanjing: Nanjing University of Posts and Telecommunications, 2012.

[35] Jia Xiao-Lin,Feng Quan-yuan,Ma Cheng-zhen.An efficient anti-collision protocal for RFID tag identification[J].IEEE Communication Letters,2010,14(11):1014-1016.

[36] Jia Xiao-Lin,Feng Quan-yuan, Yu Li-shan.Stability analysis of an efficient anti-collision protocol for RFID tag identification[J].IEEE Transactions on Communications.2012,60(8):2285-2294.

[37] Hu Zhenchao.Study on RFID anti-collision algorithm based on binary tree [D]. Changchun: Jilin University, 2009.

[38] Zhang Li, Yuan Meng-Meng, He Xing.Study on Dynamic Adjustable Algorithm for Multi-collision Bit Joint Lock Position [J] .Journal of Electronics Measurement and Instrumentation, 27 (8): 773-779.

[39] Ding Zhi-guo, Zhu Xue-yong, Guo Li, et al.Study on adaptive multi-tree anti-collision algorithm [J] .2010,36 (2): 237-241.

[40] Zhang Xue-jun, Cai Wen-qi, et al.Study on Improved Adaptive Multi-tree Anti-collision Algorithm [J]. Journal of Electronics, 2012,40 (1): 193-198.

[41] Ren Shao-jie, Hao Yong-sheng, et al.A New self-adjusting multi-tree anti-collision algorithm [J]. Computer Measurement and Control, 2015,23 (12): 4180-4183.

[42]Wang Xin,Jia Qing Xuan,Gao Xin, et al.Research on Grouping N-ray Tracking Tree RFID Anti-collision Algorithm[J]. Journal of Electronics,2016,44(2):437-444.

[43] Pan L,Wu H.Smart Trend-Traversal:A Low Delay and Energy Tag Arbitration Protocal for Large RFID System[C].In Proceedings of 28th Annual Joint Conference of the IEEE Computer and Communications. Rio de Janeiro,Brazil,2009:2571-2575.

[44] Shuen-Chih Tsai,Yu-Min Hu,Chen-Hsun Chai,et al.Efficient tag reading protocol for large-scale RFID systems with pre-reading[J].Computer Communications,2016,88:73-83

[45]Tang Zhijun,Yang Yi,Wu Xiaofeng, et al.Novel Back-off Binary Tree Anti-Collision Algorithm for Ultra High Frequency RFID[J]. Computer science and exploration, 2016,10 (5): 612-621.