

The Research of RFID Anti-Collision Algorithm Based on the Frame Slots of Multi-branched Tree

Huai WEN^{1, a}, Pengcheng WEI^{2, b}, Li LI^{3, c}, Lie LEI^{4, d} and Weihong WU^{5, e}

¹College of Automation, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

²Department of mathematics and Information Engineering, Chongqing university of education, Chongqing, 400065, China

^aemail: 991371841@qq.com, ^bemail:199771778@qq.com, *Huai WEN

Keywords: RFID, Frame Slots, Multi-branched Tree, The Prefix of Packet.

Abstract. This paper proposed a new RFID anti-collision algorithm on the basis of the frame slots of Multi-branched Tree. This new algorithm used for multi-grouped lies in the label prefix discrimination, which greatly prevents the situation that the frame length is not enough and reduces the transmission of bits effectively. Simultaneously, this algorithm reduces the Reader's search range and the probability of collisions. In the performance of decreasing the collisions slots, reducing the search depth and system delay and improving the system throughput, especially in the case of the presence of large number of tags, the new algorithm is superior to the traditional method. Numerical simulations demonstrate the efficiency of theoretical results.

Introduction

In the radio frequency identification system, the data collision exists when multiple tags simultaneously occupy channel within the range of a reader [1]. At present, the anti-collision algorithm is divided into two categories, i.e. anti-collision algorithm based on ALOHA [2] and binary tree [3] respectively. However, both the two kinds of algorithms have a number of weak points. To take the dynamic frame slotted algorithm as an example, when the number of tags is greater than the length of the longest frame length, this situation will increase the probability of collisions and it will make the system throughput decreased.

The recognition rate of the algorithm based on Binary Tree can reach 100%. However, when the number of labels in the range of reader is huge, the depth of the binary tree will greatly increase and the complexity of the reading will be improved, so that the delay of system will be increased. Obviously the efficiency of the system is unsatisfactory.

At present a lot of researchers has proposed an improved algorithm based on these two types of algorithms [4][5], but in most of the algorithms, since the branch of the tree in these algorithms is relatively simple in the case of the huge number of tags, the proportion of the collision slots is great as well. Then the system throughput and efficiency is still unable to reach the optimal state.

If there is a better anti-collision algorithm to reduce the proportion of the collision slot and the depth of the tree, then the efficiency of the system will be greatly enhanced.

The algorithm based on the ALOHA and the Binary Tree anti-collision algorithm

The label is divided into two sub-groups in the algorithm based on the tree, such as Binary tree, Decomposition Tree and Detecting conflicts tree. The randomness is large on the ALOHA algorithm when the number of tags is huge, the performance of system to be worsen. It is not suitable to large-scale to read. The rate of read can be reach to 100% as the tree of anti-collision algorithm. Then consider how to improve the performance of tree-based anti-collision protocol RFID tags [6][7][8].

Therefore, to improve the recognition efficiency, the key is to find ways to reduce the number of collision nodes and spare nodes.

The principle of improved algorithm

In this paper, the collision avoidance algorithm, discriminating the bit collision by Manchester encoding[9], it is based on the fusion of dynamic frame time slot and the backward search tree algorithm[10]. When the collision has happened, the jumping forward search is based on the discriminant of the collision and the backward strategy is adopted to realize the ordered reading. Compared with the deterministic algorithm, it reduces the probability of the tag collision and has no idle nodes. It reduces the depth and complexity of the algorithm. With the addition of the prefix to determine the stage of the premise, the determine of the prefix length to reduce the amount of information transmission and queries play an important role. Relative to the non deterministic algorithm, in the case of a large number of tags, we can group the tags according to the label ID prefix, and at the same time the same prefix label is divided into a branch, so it can appropriately prevent the use of the frame length being not enough, the using of the backward strategy can effectively reduce the search times and greatly save system time consumption.

The steps of this algorithm

The step of the improved dynamic frame slotted multi-tree algorithm as follows (assuming the tag ID of length N bits, $X = 0$, the stack $S = \text{null}$):

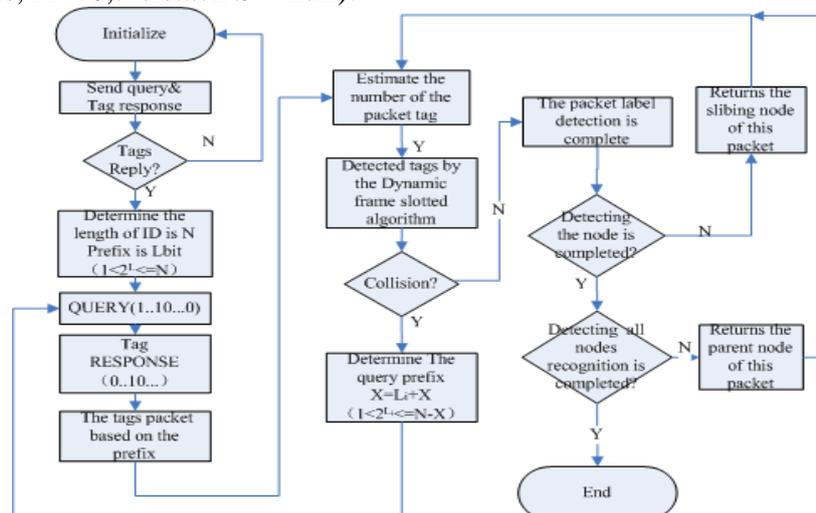


Figure 1. The step of this algorithm on the tag identification

(1) The reader sends a query command, all tags respond to it, which do the next step, if the label is reply, otherwise it will repeated the inquiries if the label is not reply

(2) Determining the length of query prefix is L bits, the reader broadcasts a query command, which length is L Query (11 ... 100 ...) the successive of "1", it is equal to the length of L . The entire length of the tag query command is equal to the length of ID. According to the ID number of L bit, all tags reply 2^L bit binary sequence, which is set by the previous steps.

(3) The tag prefix determine by the collision position when the reader receiving the labels response, each of prefix represent of a branch and these prefixes save to stack S successively.

(4) Pop the top of element, the reader send a query command including the prefix, the label reply the surplus of $N-L$ bit ID.

(5) Detect the packet labels by the Dynamic Frame Slot Aloha (DFSA), if tangs collision has been happened, redefining the sub-branch prefix node, make $X = L + X$, $1 < 2^L <= N - X$, return to step 2, if the collision has not occurs, the packet tag identification successfully, send a sleep command, so the identification tags into hibernation, the packet label query is completed.

(6) Detect the node of the packet identifying whether completed, if it is not, then query the sibling group node of this packet, return to Step 4. If it is completed, query the parent node of this packet and return to the step 4.

(7) All the tag distinguished and the query is finished when the stack S is empty.

The number of queries analysis of the this algorithm

Suppose in the range of the reader, the number of tag is N_0 and the bits of tag $N = 96$, The query prefix of the reader is $L_i (i=0...23)$. The parameters of the tag prefix X is initialized to zero, make $X = L_i + X$, according to the rules of the prefix tag reader queries tags reply stage, there may be 2^{L_i} prefix has be exist, to reduce the number of bits transmitted, the number of bits prefix tag query stage reply should be less than the length of the label, set $1 < 2^{L_i} \leq N - X$.

It is supposed that there may be a number of k tags with the same prefix in N_i tags. The probability is expressed as $P(k, N_i, L_i)$. Then

$$P(k, N_i, L_i) = \binom{N_i}{k} p^k (1-p)^{(N_i-k)}, p = 2^{-L_i}, 1 < 2^{L_i} \leq N - X \quad (1)$$

Set $p_i(k, N_i, L_i)$ is the probability of idle prefix, set $p_s(k, N_i, L_i)$ is the success probability of identifying prefix and set $p_c(k, N_i, L_i)$ is the success probability of collision prefix, Then

$$p_i(k, N_i, L_i) = P(0, N_i, L_i) = (1 - 2^{-L_i})^{N_i} \quad (2)$$

$$p_s(k, N_i, L_i) = P(1, N_i, L_i) = N_i 2^{-L_i} (1 - 2^{-L_i})^{N_i-1} \quad (3)$$

$$p_c(k, N_i, L_i) = 1 - P_i(k, N_i, L_i) - p_s(k, N_i, L_i), 1 < 2^{L_i} \leq N - X \quad (4)$$

It can be obtain the number of idle slots label is NI , the number of successful slot tags is NS and the number of collision slots label is NC :

$$NI_i = 2^{L_i} p_i = 2^{L_i} (1 - 2^{-L_i})^{N_i} \quad (5)$$

$$NS_i = 2^{L_i} p_s = N_i (1 - 2^{-L_i})^{N_i-1} \quad (6)$$

$$NC_i = 2^{L_i} p_c = 2^{L_i} (1 - (1 - 2^{-L_i})^{N_i} - N_i 2^{-L_i} (1 - 2^{-L_i})^{N_i-1}), 1 < 2^{L_i} \leq N - X \quad (7)$$

The number of tags is massive in the things applications in the future when $N_i \gg 2^{L_i}$, $(1 - 2^{-L_i})^{N_i-1} \rightarrow 0$, $(1 - 2^{-L_i})^{N_i} \rightarrow 0$, Get into the expression, it can be obtained $NC \rightarrow 2^{L_i}$. That is, when the number of tags is much larger than the number of prefixes that can be represented, almost no rough idle prefix and prefix successfully identified, each bit of prefix are collision, so there are $N_i / 2^{L_i}$ tags in the prefix represent of each branches.

In the frame slot algorithm, in order to be able to get a reasonable identification tag distribution and to reduce collision position as well as idle position, we should make the allocation of time slots equal to the number of tags as much as possible. After research and analysis to obtain the dispensing of the number is close to the number of labels that need to meet the proportion of relationship, such as formula (8).

$$N_i : N_s : N_c = 2L : 2N_i L : N_i^2 (1 + \frac{N_i}{3L}) \quad (8)$$

Research shows that dynamic frame slotted algorithm to maximize system throughput was 36.8% when $L = Ni$, that is when $N_i : N_s : N_c = 1 : 1 : \frac{2}{3}$ can be obtain optimal system efficiency. In this case, $L = Ni$, $ns = \frac{3}{8} N_i$ tags has been successfully identified in the first phase, There be $nc = \frac{5}{8} N_i$ tags collide in the number of $\frac{1}{4} L = \frac{5}{8} N_i$ collision slots, the collision of the label needs to reacquire the length of the prefix and branch pressed again by the prefix, then complete the identification. If it is in the j -th collision slots, set the number of collision labels is k_j and then expressed the sum of collision

tag as: $nc = \sum_{j=1}^{N_i/4} k_j$. The number of branches after each collision is N_{c_i} , the value of i is from 23 to 1, when the tag bits is $N = 96$. Consider the situation we put the tag identification with a worst-case scenario to calculate the number of searches, when L_i 's i value is one, the last two remaining branches of the tree identification tag was completed that is the prefix L_i position lies in the label reply number changes up to $N = 23$ times. According to the above analysis. the algorithm of the total number of

searches as follows:

This is the bits of the prefix Li calculate method,,which is rounded down :

$$X = L_{i-1} + X, 1 < 2^L_i \leq N - X \quad (9)$$

This is the branches of Nc_i calculate method,which is Rounded on:

$$Nc_i = 2^{L_i} p_c = 2^{L_i} (1 - (1 - 2^{-L_i})^{N_i} - N_i 2^{-L_i} (1 - 2^{-L_i})^{N_i-1}), 1 < 2^{L_i} \leq N - X \quad (10)$$

This is the number of branches N_i calculate method:

$$N_i = \sum_{j=1}^{N_{i-1}} k_j = \frac{5N_{i-1}}{8} \frac{1}{2^{L_{i-1}}} = \frac{5N_{i-1}}{2^{L_{i-1}+3}} \quad (11)$$

The total number of queries of this algorithm as follows:

$$F(i) = Nc_0 \left(\frac{N_0}{2^{L_0}} + f(i+1) \right), \text{Where } f(i) = Nc_i (N_{i+1} + f(i+1)) \text{ and } 1 \leq i \leq 21 \quad (12)$$

Consequently the throughput of this algorithm is:

$$S = \frac{N_0}{F(i)} \quad (13)$$

The change of the number of searches analysis based on the Binary tree and Frame slots of Multi-branched Tree algorithm.

Given the number of tags within the reader range from 0 to 10000. We can compare the number of searches between the binary algorithm and this algorithm within this range changes can be seen from figure 2:

The number of Tags	Algorithm	
	Binary Tree	Frame slots of Multi-branched Tree
256	1940	295
512	4387	576
1000	1525	1064
2000	21053	2192
3000	33331	3384
4000	46100	4576
5000	59233	5768
6000	72656	7024
7000	86321	8216
8000	100193	9472
9000	114245	10664
10000	128458	11920

Figure 2: The number of searches on the binary tree and Frame slots of Multi-branched Tree

The figure 2 shows that the number of tags in a case from less to more. The number of searches on this algorithm is less than the amount of the binary tree.

The change of throughput analysis based on the three algorithm .

Given the number of tags within the reader range from 0 to 10000. We can compare the throughput of among the binary algorithm, dynamic frame slotted anti-collision algorithm and this collision algorithm within this range changes can be seen from Figure

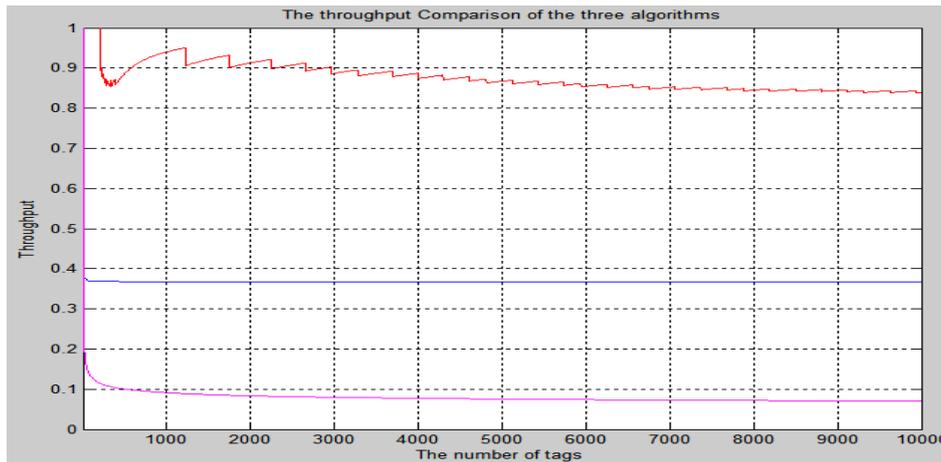


Figure 3 three kinds of anti-collision algorithm change throughput comparison chart

To sum up, the figure 3 shows that the number of tags in a case from less to more. The throughput of this algorithm is greater than the throughput of the binary and dynamic frame slotted algorithm, and the throughput is more than 2.3 times to the dynamic frame slotted Algorithm.

Conclusion

This paper is based on the improved the backward search tree and dynamic frame slotted algorithm. Solve the problem of the large number of tags RFID systems collide. The algorithm is different from the conventional anti-collision algorithm. is that According to the judgment prefix collision, make the length of prefix change to the Multi-packet, when a collision has happened and the bits of tag has changed as well as the depth of tree has changed. Each branch of the tree using dynamic frame slot algorithm distinguished the tag. Take back strategy when the collision has not happened, orderly realization tag read. The simulation results clearly shows that new algorithm on the number of searches and the throughput is superior to the traditional anti-collision algorithm.

References

- [1] Wenjing Fan, Shanshan Zhang. The Research of RFID Anti-Collision Algorithm based on the retreat binary tree[J]. Computer Software and Applications, 2012.
- [2] Hong Wan, Yanzhao Yang, The Research and Improved of RFID Anti-Collision Algorithm[J]. Microcomputer Information, 2009. 25(3-2). 230-232.
- [3] Tingzhao Lin, The Research and Improved of Binary Tree Anti-Collision Algorithm[J]. Journal of Fujian University of Technology, 2008. 25(3-2). 732-736.
- [4] ISO/IEC14443-3, identification cards contactless integrated circuit card-proximity cards part 3 initialization and anti-collisions.[J], 2003.
- [5] Songseng Yu. Anti-collision operator Regressive Index based on a binary tree search and its realization[J]. Computer Engineering and Applications, 2004. 40(16). 26-28.
- [6] Hwang T W, Lee B G, Kim Y S. Improved anti-collision scheme for high speed identification in RFID system,[J]. proceedings of the First International Conference on Innovative Computing, Information and Control.
- [7] Choi J H. Bi-slotted tree based anti-collision protocols for fast tag identification in RFID systems[J]. IEEE Communications Letters, 2006, 10(12): 861-863
- [8] Lifan Jiang, guizhang Lu. The Research of fast tag identification in RFID systems[J]. Computer Engineering and Applications, 2007, 43(15): 29-32.
- [9] Nian Liu. The Research of RFID Anti-Collision Algorithm Based on the Dynamic Frame

slotted ,[D].Shanghai Normal University,2014,9-10.

[10]ZhongyongWang, Xiangchuan Gao,The Research of RFID Anti-Collision Applications[J],2009.45(3).210-2