Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 154 (2019) 460–467

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Congress of Information and Communication Technology, ICICT 2019

Research on Anti-Collision Algorithm of RFID Tags in Logistics System

Wei Hui Zhou[a,], Nian De Jiang[b], Cheng Cheng Yan[a]

[a]East China Institute of *Technology* Yangtze River College, *Fuzhou* 344000, China
[b] East China *Institute of Technology Nangchang 330013, China*

Abstract

In order to solve the problem of RFID tags collision in the current logistics system. Based on the study of existing anti-collision algorithms, a grouped dynamic frame slot with query tree (GDQT) algorithm was designed for a large number of labeling scenarios in logistics management. The GDQT algorithm first estimates the number of tags that have not been identified after each round of identification by the tag estimation algorithm, if less than or equal to 354, the tags were first identified using the dynamic frame slot ALOHA (DFSA) algorithm., and then using the query tree algorithm (QT) for the remaining tags identification. If the number of unrecognized tags was greater than 354, the tags should be first grouped, and then DFSA and QT would be used for tags identification. The simulation results showed that GDQT algorithm can improve system throughput rate and keep throughput rate at around 0.82, which is 300%, 120%, 50% higher than the DFSA algorithm, GDFSA algorithm and AHT algorithm respectively. Therefore, The GDQT algorithm has a good application prospect in the logistics system.

Keywords: logistics system; GDQT; throughput rate; anti-collision algorithms; RFID

1. Introduction

Radio frequency identification (RFID) is a non-contact automatic identification technology for communication*

* Corresponding author: Wei-hui Zhou, Tel: +1-510-794-1562.
E-mail address: canlly12@ecit.cn

10.1016/j.procs.2019.06.065

between tags and readers [1]. And it is widely used in logistics system, book management, large shopping supermarkets, intelligent transportation, intelligent packaging management and other fields [2]. This paper mainly studies the application of RFID in logistics system management. The principle is to implant RFID tags in the logistics system, and the data can be recognized, counted and judged by the reader, no need to manually scan one by one, which greatly saves manpower and material resources [3]. Because of the multi-tag collision of RFID in the logistics system, how to read the tag data accurately and quickly without collision is one of the problems that need to be studied and solved in the application of RFID in the logistics system management [4]. At present, two kinds of anti-collision algorithms have been proposed, which are tree-based anti-collision algorithm and ALOHA-based anti-collision algorithm [5]. Tree-based anti-collision algorithm divides tags into N sub-trees until all tags are recognized. The main algorithms are Query Tree (QT) [6], Adaptive Multi-tree Search (AMS) [7], Improved Adaptive Multi-tree Search (IAMS) [8], Adjustive Hybrid Tree (AHT) [9] and Improved hybrid query tree algorithm (IHQT) [10]. The principle of ALOHA algorithm is to identify tags by random slot allocation and random slot response of tags. Classical ALOHA algorithms include dynamic frame slotted ALOHA(DFSA) algorithm [11], grouped adaptive allocating slot ALOHA (GAAS) algorithm [12], tag estimation method (TEM) algorithm [13], grouped dynamic Frame- slotted ALOHA(GDFSA) algorithm [14], etc.

When the number of tags is large, DFSA algorithm has some shortcomings such as throughput reduction, HQT algorithm has some shortcomings such as collision slots, increase of idle slots and so on. In this paper, a grouped dynamic frame slot with query tree (GDQT) algorithm is proposed, the idea is to estimate the number of tags that have not been identified after each round of recognition. If the number of tags that have not been identified is less than 354, the DFSA algorithm (DFSA) is used to identify the tags first, and then the tree query algorithm (QT) is used to identify the tags; If the number of unrecognized tags is more than 354, the tags are grouped first, then the tags are identified by DFSA and QT respectively.

2. GDQT Algorithm Description

2.1 Tags number estimation algorithm

Before the next round of recognition, the algorithm in this paper needs to estimate the number of tags left in the previous round, and choose the frame length and the number of groups according to the number of tags, so the number of tags estimation algorithm is essential in this paper. Typical tag number estimation algorithms include LB estimation algorithm, Schoute estimation algorithm and CHI estimation algorithm [15]. If the number of tags is large, compared with LB estimation and Schoute estimation, CHI estimation can minimize the error and maximize the throughput rate.

The CHI estimation algorithm combines the expected value with the actual value to obtain the number of tags n, that is, to minimize the difference between the actual time slot values C_e , C_s, C_c and the theoretical time slot expectations N_e , N_s , N_c . Equation (1) finds the value of ε from $C_s + 2C_c$ to $2 \times (C_s + 2C_c)$, and the corresponding n value is the estimated number of tags [16-18].

$$\varepsilon\left(N, C_e, C_s, C_c\right) = \min_n \left\| \begin{pmatrix} N_e \\ N_s \\ N_C \end{pmatrix} - \begin{pmatrix} C_e \\ C_s \\ C_c \end{pmatrix} \right\| \tag{1}$$

In Equation (1): n is the number of tags to be identified per round; N is the frame length; N_e , N_s , N_c are the expected values of idle time slots, success time slots and collision time slots after each round of identification; C_e , C_s, C_c are The actual value of the idle time slot, the success time slot, and the collision time slot after each round of identification.

2.2 Adjustment of frame length and group numbers

The analysis of DFSA shows that in order to achieve the maximum throughput of the system, it is necessary to make the frame length infinitely close to the number of tags. In practical applications, frame length cannot be arbitrary, it must be an integer power of 2. Because of hardware limitations, frame length N cannot be greater than 256, that is, $N_{max} = 256$. Generally, frame length N is 2, 4, 8, 16, 32, 64, 128, 256.

If the number of tags is small, the throughput rate of DFSA algorithm can be maintained high; If the number of tags is large, the throughput rate of DFSA algorithm will be reduced, and the recognition time will increase accordingly. In order to ensure that the frame length and the number of identification tags are similar, the number of tags in response can be controlled by tag grouping, so that the throughput of the algorithm can be maintained at a higher level.

The algorithm of this paper, the reader estimates the n value of the tag according to the CHI estimation algorithm, and then compares it with N_{max}. If it is less than N_{max}, tags are not grouped; if it is larger than N_{max}, tags are grouped, and only one group of tags can participate in each round of recognition.

The number of successful slots N_s is:

$$N_s = Nn\left(\frac{1}{N}\right)\left(1 - \frac{1}{N}\right)^{n-1} \tag{2}$$

The throughput rate of n tags response in a system with a frame length of N is:

$$P_N = \frac{N_s}{N} = n\left(\frac{1}{N}\right)\left(1 - \frac{1}{N}\right)^{n-1} \tag{3}$$

When $P_N = P_{2N}$, the number of tags at the intersection of 2 adjacent frame performance curves is:

$$n_{N,2N} = \frac{\ln(2)}{\ln\left(\frac{2N-1}{2N-2}\right)} \tag{4}$$

When $N_{max} = 256$, the response tag is divided into m groups and m+1 groups, the value at the intersection of system performance curves is:

$$P = \frac{n}{m}\left(\frac{1}{256}\right)\left(1 - \frac{1}{256}\right)^{\frac{n}{m}-1} = \frac{n}{m+1}\left(\frac{1}{256}\right)\left(1 - \frac{1}{256}\right)^{\frac{n}{m+1}-1} \tag{5}$$

It can be concluded that the relationship between N and M is:

$$n = m\left(m+1\right)\frac{\ln\left(m+1\right)-\ln\left(m\right)}{\ln\left(256\right)-\ln\left(255\right)} \tag{6}$$

The group threshold can be obtained by bringing m =1, 2, 3, 4 into equation (6). For example, when m = 1, n is:

$$n = 2\frac{\ln\left(2\right)-\ln\left(1\right)}{\ln\left(256\right)-\ln\left(255\right)} \approx 354 \tag{7}$$

That is to say, the threshold value of the tags divided into two groups is 354, and so on, then the relationship between the number of tags and the number of groups m and the frame length N is shown in Table 1.

2.3 Steps of GDQT algorithm

The core of GDQT algorithm is to estimate the number of tags before each round of recognition to determine the number of groups and frame length. Then GDFSA algorithm is used to group tags, and the remaining collision tags are identified by hybrid tree query algorithm. The steps of the algorithm can be divided into 3 stages: the number of tags estimation and grouping stage, frame time slot processing stage and collision tags processing stage. The algorithm flow is shown in Figure 1.

Query (n, N): Query command in frame time slot processing stage, reader will send query commands to tags.

request (q q1 2 qk): Query command in the collision tags processing stage, and $q_1 q_2 \cdots q_k$ is the query prefix, and the collision tags compared the ID prefix with the query prefix after receiving the command. If the same, the tag will be responded

forecast-branch ($q_1 q_2 \cdots q_k$ $\cdots$): Branch prediction command. The number of 1 is L of $q_1 q_2 \cdots q_k$ $\cdots$,and the value of L is determined by the number of collision tags. If there are collision tags, the reader sends a branch prediction command to the collision tags to select the optimal tree.

Table 1. Relationship between the number of tags and the frame length N and the number of groups m

total number of tags *n*	frame length *N*	number of groups *m*
1	2	1
2～ 5	4	1
6～11	8	1
12～22	16	1
23～44	32	1
45～88	64	1
89～176	128	1
177～354	256	1
355～622	256	2
623～883	256	3
884～1141	256	4
⋮	⋮	⋮

2.3.1 The number of tags estimation and grouping stage

Step 1: Before each round of identification, the value of the number of tags n is estimated by the CHI estimation method.

Step 2: If the number of labels n is less than or equal to 354, the first stage is identified by the DFSA algorithm, and the frame length N is selected according to Table 1, then entering the frame time slot processing stage; If the number of tags is greater than 354, the tags are grouped first. And the number of groups m is calculated from Table 1.

Step 3: The tags randomly select the number between 1 and m as its own group number t, and records the number of tags in each group.

Step 4: Initialization t=1, that is, starting from the tags with the group number 1 and then identifying the label in the t(t>1) group, jumping to step 1 in the frame slot processing stage.

2.3.2 The frame time slot processing stage

Step 1: The reader sends a Query (n, N) command to the tags in group t, at which point all tags respond.

Step 2: The response tags select the slot number i from the (0, N-1) time slot according to the rules.

Step 3: Judging the case of slot i, if there is only one tag response on the slot i, then the slot is a successful slot, and continues to be identified by entering slot i+1; if there is no tag response on the slot i, then the time slot is an empty slot, and continues to be identified by entering slot i+1; if there are at least 2 tags response on slot i, then these tags enter the collision tags processing stage to continue to be identified.

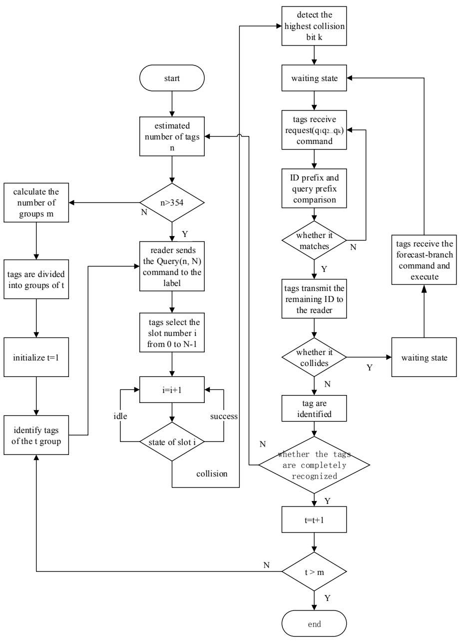

Fig. 1. Flow chart of GDQT algorithm

2.3.3 The collision tags processing stage

Step 1: The reader detects the highest conflict bit K, sends a request ($q_1 q_2 \cdots q_K$), the tag receives the request, and compares the ID prefix with the query prefix.

Step 2: Judge whether they are the same, if they are the same, the response tags sends the remaining ID to the reader. If not, jump to step 1.

Step 3: Detect whether the response tags has a collision, if there is no collision, they are successfully identified. If there is a collision, the reader sends a forecast-branch ($q_1q_2 \cdots q_k \quad \cdots$) command, and the collision tags receives the command and jumps to step 1.

Step 4: Judge whether all the tags in the group are completely identified. If not, then jump to step 1 in the number of tags estimation and grouping stage are continued. If all the tags are identified, the group number is incremented by 1, that is, t=t+1.

Step 5: To judge the value of the group number t, one of the following two situations is performed: $t \leq m$, that is, there is still a group of tags are not identified, and it will jump to the step 4 in the number of tags estimation and grouping stage to continue the identification of the next group of tags; $t > m$, indicating that all group of tags are all identified and the algorithm is finished.

3 Algorithm Performance and Simulation Result Analysis

In the Windows 10 operating system and 4G memory environment, the GDFSA algorithm, DFSA algorithm, AHT algorithm and the GDQT algorithm in the paper were experimented with MATLAB software. The number of tags is from 0 to 1000. In order to ensure the accuracy of the experiment, the average of 100 experiments was taken for each simulation result.

3.1 Total number of time slots analysis and comparison

The number of slots required in the frame time slot processing stage in the GDFSA-HQT algorithm is identified by the GDFSA algorithm. The required number of slots is the sum of the selected frame lengths in each round. With the collision tags processing stage in the GDQT algorithm, according to the collision tags, the number of branches of the tree is selected, and the reader automatically avoids the idle node, so the idle time slot can be avoided in the frame time slot processing stage. When there are two collision tags, using the binary tree algorithm, the number of slots required is three; When there are three collision tags, using the trinomial tree algorithm, the number of slots required is four. And so on, so if the number of collision tags is k, the number of slots M required for this stage can be obtained: M=k+1.

The comparison of the total number of time slots of DFSA algorithm, GDFSA algorithm, AHT algorithm and GDQT algorithm is shown in Figure 2. The number of tags varies from 0 to 1000, the total number of time slots consumed by the DFSA algorithm increases exponentially. And the GDFSA algorithm, the AHT algorithm and the GDQT algorithm increase linearly, but compared to the other two algorithms, GDQT algorithm increases the slowest. Especially when the number of tags is large, the GDQT algorithm has the greatest advantage. When the number of tags reaches 1000, the GDQT algorithm only needs the number of time slots is about 1051, while the DFSA algorithm, GDFSA algorithm and AHT algorithm need 5127, 2777 and 1,833 time slots. They were reduced by about 4,076, 1,726, and 782 respectively.

3.2 Collision time slots analysis and comparison

The number of collision slots required by the GDQT algorithm in the identification process is the sum of the collision time slots required for the two stages of the frame time slot processing stage and the collision tags processing stage. It is worth noting that in the collision tags processing stage, since the number of branches of tree is equal to the number of collision tags, that is, there are k collision tags to select the k-tree to identify. And then the child nodes are all successful time slots, then in the whole process, only the root node is a collision time slot, so the number of collision time slots required in the collision tags processing stage is one.

The comparison of the number of collision slots of DFSA algorithm, GDFSA algorithm, AHT algorithm and GDQT algorithm is shown in Figure 3. The number of tags varies from 0 to 1000, the number of collision slots consumed by the DFSA algorithm increases exponentially. And the GDFSA algorithm, the AHT algorithm and the GDQT algorithm increase linearly, but compared to the other two algorithms, GDQT algorithm increases the slowest. When the number of tags reaches 1000, the GDQT algorithm only needs the number of collision slots is about 542, which is about 396 less than the AHT algorithm, about 3370 less than the DFSA algorithm, and about 418 less than the GDFSA algorithm.

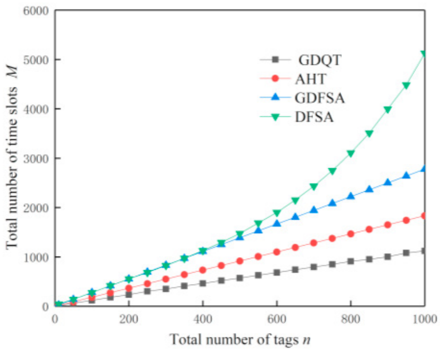

Fig. 2. Comparison of total number of time slots

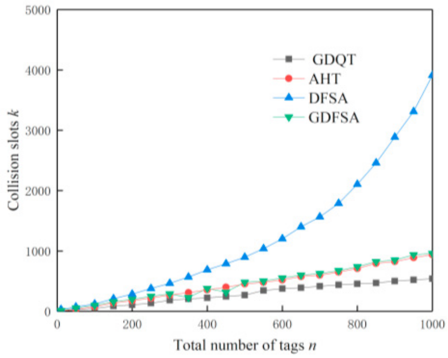

Fig. 3. Comparison of number of collision slots

3.3 Collision time slots analysis and comparison

The throughput rate of RFID system is an important parameter to measure the performance of the system. If the number of response tags in the whole identification process is n and the total number of time slots consumed is M, the throughput rate of the GDQT algorithm is:

$$E = \frac{n}{M} \tag{8}$$

The comparison of the throughput rate of DFSA algorithm, AHT algorithm, GDFSA algorithm and GDQT algorithm is shown in Figure 4. When the number of tags varies from 0 to 1000, the throughput rate of AHT is kept at about 0.55, and the throughput rate of GDFSA is always maintained at about 0.36; When the number of tags is greater than 354, the throughput of the DFSA algorithm decreases sharply with the increase of the number of tags. The throughput of the GDQT algorithm has been maintained at around 0.82, especially when the number of tags increases to 1000, it can reach about 0.89. Compared with AHT algorithm, DFSA algorithm and GDFSA algorithm, the throughput of GDQT algorithm is increased by 104.7%, 358.7% and 145.2%.

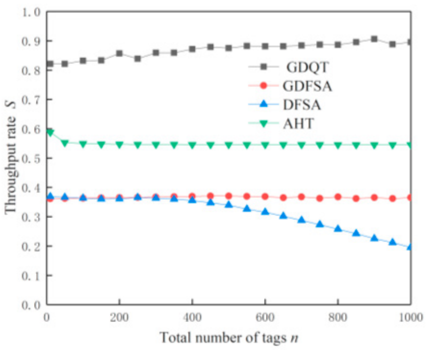

Fig. 4. Comparison of throughput rate

4. Conclusion

Aiming at the problem of RFID multi-tag collision in logistics system, a grouped dynamic frame slot with query tree (GDQT) algorithm was designed. The number of tags is estimated before each round of identification, and the tags are grouped according to the number of tags, and combined with the GDFSA algorithm and the HQT algorithm for identification. Simulation results show that the throughput rate of the GDQT algorithm can be maintained at around 0.82, which is significantly higher than the DFSA algorithm, AHT algorithm and GDFSA algorithm. Therefore, the GDQT algorithm can solve the tags collision problem faster and more efficiently. So, it has a good application prospect in the logistics system.

References

1. Deng C H. High-efficient and security RFID authentication protocol based on EPC-C1G2 standard. Control engineering of China, 2017, 24(1): 141-147.
2. Zhang X H, Zhou W H. Research on dynamic framed binary tree anti-collision algorithm for RFID system. Journal of system simulation, 2018, 30(3): 1063-1073.
3. He C L, Yin F S. Research of application of RFID technology in logistics industry. Electronic design engineering, 2013, 21(14): 12-14.
4. Fu Y, Qian Z H, Cheng C, et al. Bit arbitration query tree anti-collision algorithm based on grouping mechanism. Journal on communications, 2016, 36(1): 123-129.
5. Wang X, Hu Z C, Qian Z H. Research on RFID anti-collision algorithms based on binary tree. Journal on communications, 2010, 31(6): 49-57.
6. Yang F, Xu H L, Xie, et al. A RFID anti-collision algorithm based on double idle factors. Computer engineering & science, 2016, 38(7): 1140-1146.
7. Wang Y, Tang X H, Zhang L J. Group strategy for remaining tags in anti-collision algorithm for RFID system. Journal of electronics & information technology, 2016, 38(3): 594-599.
8. Zhang X J, Cai W Q, Wang S P. One anti-collision algorithm based on improved adaptive multi-tree search. Acta electronic sinica, 2012, 40(1): 193-198.
9. Song J H, Guo Y J, Han L S, et al. An adjustive hybrid tree anti-collision algorithm for RFID multi-tag identification. Acta electronic sinica, 2014, 42(4): 685-689.
10. Shi L Q, He Y G, Luo Q W, et al. Improved hybrid query tree anti-collision algorithm of RFID. Journal of electronic measurement and instrumentation, 2017, 31(8): 1281-1288.
11. Chen H L. Research on anti-collision algorithm of RFID system. Computer technology and development, 2016, 26 (10): 108-117.
12. Zhang X H, Hu Y M. Research on a grouped adaptive allocating slot anti-collision algorithm in RFID system. Acta electronic sinica, 2016, 6(6):1328-1334.
13. Cha J R, Kim J H. Novel anti-collision algorithms for fast object identification in RFID system. Proc of the 11th international conference on parallel and distributed systems, 2005: 63-67.
14. Lin C F, Lin F Y S. Efficient estimation and collision-group-based anti-collision algorithms for dynamic frame-slotted ALOHA in RFID networks. IEEE transactions on auto-mation science and engineering, 2010, 7(4) :840-848.
15. Lu D, Li S L, Xu C S. Adaptive frame size and group division in DFSA algorithms on the basis of CHI Tags' estimation. Journal of harbin university of science and technology, 2015, 20(1): 56-60.
16. Chen C M, Feng Y T, Fu L C. Improved anti-collision DFSA algorithm for RFID system. Application of electronic technique, 2013, 39(1): 86-89.
17. Lu D, Ling X Y. All grouping unequal slot DFSA algorithm. Measurement & Control technology, 2013, 32(9): 55-59.
18. Pan Z R, Shang K. A modified dynamic Frame slotted ALOHA algorithm for RFID. Transducer and microsystem technologies, 2013, 32(9): 55-59.