An adaptive frame slotted ALOHA anti-collision algorithm based on tag grouping

Junsuo Qu\textsuperscript{1} | Ting Wang\textsuperscript{2}

\textsuperscript{1}School of Automation, Xi’an Key Laboratory of Advanced Control and Intelligent Process, Xi’an University of Posts & Telecommunications, Xi’an, China
\textsuperscript{2}School of Communication and Information Engineering, Xi’an University of Posts & Telecommunications, Xi’an, China

Abstract
Multi-tag anti-collision is an important problem in radio frequency identification (RFID) application. Solving the problem is of great significance to the RFID technology application and the future internet of things; therefore, an adaptive frame slotted ALOHA anti-collision algorithm based on tag grouping (IGA) is proposed. First, a novel method for estimating the number of tags accurately is proposed. Through theoretical research and the experimental verification, a relationship is obtained between the ratio of the collision time slot in the frame and the average number of tags in each collision slot, which helps us to calculate the number of tags. Second, the method of estimating the number of tags is applied to the IGA algorithm. The reader randomly groups the tags after the number of tags are estimated, and recognises the tags by grouping. In the identification process, the idle time slot is skipped automatically, and the collided tags can be identified with an additional frame until all tags are identified. The simulation results show that the total time slot of the IGA algorithm is relatively small, and the identification efficiency is about 71\%, which is 30\% better than the the improved RFID anti-collision algorithm and 90\% higher than the traditional ALOHA algorithm.

1 | INTRODUCTION

The internet of things is a hot technology in recent years, and it is a new technology revolution after the Internet and mobile communication technologies. In the realisation of the internet of things, automatic identification and information exchange using radio frequency identification (RFID) technology play an important role.

RFID is a new type of contactless communication technology, and information are transmitted through radio frequency spatial coupling, electromagnetic field or alternating magnetic field to achieve the purpose of identification [1]. RFID technology is widely used in highway tolls and intelligent transportation, electronic ticket, cargo tracking management and warehousing, anti-counterfeiting, and sharing umbrellas because of the characteristics of contactless recognition, fast recognition of multiple targets, strong adaptability, and low energy consumption and so on [2].

The RFID system is mainly composed of readers, electronic tags and application software, as shown in Figure 1 [3]. When the reader recognises the tags, the phenomenon of tag collision will occur because of the overlap of the multi-tag information, and it has greatly hindered the development of RFID technology. Therefore, solving the existing RFID multi-tag collision problem is of great significance to the RFID technology market and the future internet of things. There are also several ways to solve this problem. At present, the foremost algorithms to solve this problem are deterministic algorithm based on binary-based deterministic algorithms and ALOHA-based statistical algorithms [4]. The ALOHA-based statistical algorithms mainly include the slotted ALOHA algorithm, the frame slotted ALOHA algorithm (FSA), and the dynamic frame slotted ALOHA algorithm [5]. By studying the problem of multi-tag anti-collision, RFID technology will play a greater role in logistics management, transportation, automatic production, public information services and other domains.
 industries, which can greatly reduce costs and improve the efficiency of management and operation.

In the problem of tag anti-collision, scholars have conducted related research on binary-based deterministic algorithms and ALOHA-based statistical algorithms. In the research on binary-based deterministic algorithms, Wijayasekara et al. [6] proposed a new tree-based anti-collision protocol for RFID systems to achieve a very high tag identification efficiency. The proposed algorithm works in two phases. In the first phase, the number of competing tags is estimated through the proposed Bayesian estimation technique, while in the second phase, tags are identified using our modified dynamic tree algorithm. Su et al. [7] proposed a new query tree algorithm of bit query-based M-ray tree (BQBMT), which can quickly identify tags through the separation and collision of binary tree iteration, and the best switch between bit query mode and ID query mode. Yang et al. [8] proposed a new Q value algorithm based on EPC-C1G2. In this algorithm, the method of Q is improved effectively. Otherwise, combined with the label grouping algorithm, the proposed method achieves higher system efficiency in a large number of tag data.

In the research on ALOHA-based statistical algorithms, Solić et al. [9] proposed an early frame interruption strategy for ALOHA-based RFID systems, showing how to apply the slot-by-slot (SbS) estimation method and the early frame interruption strategy. He et al. [10] proposed an improved ALOHA anti-collision algorithm, which determines the frame size of handling collision tags by the ratio of the selected time slot to the total time slot. Xu et al. [11] proposed the improved RFID anti-collision algorithm (IAA) algorithm to determine the frame size of the collision tag by estimating the number of tags in the collision time slot from the sequence returned by the tag.

Because the binary-based deterministic algorithm is complicated to achieve high cost, and ALOHA-based statistical algorithms has the advantages of fast recognition efficiency, so this paper studies on the basis of ALOHA-based statistical algorithms. Compared with other algorithms that have been proposed, Yejun He algorithm and IAA algorithm have higher recognition efficiency and lower complexity. However, their algorithm is not suitable for a large number of tags, and the selection of frame length cannot vary with the number of tags in IAA algorithm. Based on this, we propose an adaptive frame time slotted Aloha anti-collision algorithm based on tag grouping (IGA). The IGA algorithm improves the efficiency and stability of the system by estimating the number of tags and identifying grouping tags. After the number of tags is estimated, the reader randomly groups the tags according to the number of tags. Then, the reader recognizes the tag by grouping. In the identification process of tags, the idle time slot is determined according to the slot sequence returned by the tag and is skipped automatically. The tag in the collision time slot can be identified with an additional frame until all tags are identified. The identification efficiency of the algorithm is improved, and the complexity is low, which is suitable for a large number of tags.

2 | TRADITIONAL ALOHA ALGORITHM

There are three kinds of the traditional ALOHA algorithm, pure ALOHA algorithm (P-ALOHA), slotted ALOHA algorithm (S-ALOHA) and FSA. The P-ALOHA algorithm is basic, and it is recognized immediately as long as there is a tag in recognition area. Once collision, the tag will wait for a random amount of time and retransmits the data till it transmits successfully. No synchronisation is required, and the tag collision is a complete collision or a partial collision. The schematic diagram is shown in Figure 2, and the maximum identification efficiency is 18.4% [12].

The S-ALOHA algorithm divides the time into discrete intervals called time slots, and all tags can transmit data only at the beginning of the slot. If there is only one tag in a time slot, the tag will be recognized successfully. If there are two or more tags in a time slot, collision will occur. The schematic diagram is shown in Figure 3 and the maximum identification efficiency is 36.8% [12].

The FSA algorithm is improved on the basis of the S-ALOHA algorithm. When the number of tags is far more than the number of slots, the efficiency of the S-ALOHA algorithm will be reduced, and the FSA algorithm will solve the problem of a large number of tags. In the FSA algorithm, every tag will be identified in the time slot of this frame after the frame is divided into several slots with the same length [13,14]. If a collision occurs, the collision tags will be identified in the next frame. The schematic diagram of the FSA algorithm is shown in Figure 4. The maximum identification efficiency is 36.8%.

3 | IGA ALGORITHM DESCRIPTIONS

The IGA algorithm selects the same type of tag to identify. Assuming that the IGA algorithm selects passive electronic tags, the quality and size of tags are exactly the same. All tags are identified under a RFID reader, and the distribution of tags is even.
3.1 Analysis of time slot in IGA algorithm

Assuming that the number of tags is \( N \) and the frame size is \( L \), \( N \) tags will be identified in \( L \) slots. Each tag randomly selects a time slot to send tag information. According to the binomial distribution theorem, the probability of a tag occupying a slot in a frame is \( P = \frac{1}{L} \). And the probability of having tags in the same slot \([15]\) is

\[
P_r = C_N^r \left( \frac{1}{L} \right)^r \left( 1 - \frac{1}{L} \right)^{N-r}
\]  

(1)

When \( r = 0 \), it indicates that there is no tag in the time slot, which is called an idle time slot, and its occurrence probability is recorded as \( P_0 \). When \( r = 1 \), it means that there is only one tag in the time slot, called a successful time slot, and its probability is expressed as \( P_1 \). When \( r \geq 2 \), it means that there are multiple tags in the time slot, called a collision time slot, and the probability is represented as \( P_{\geq 2} \). Probability can be expressed as

\[
\begin{align*}
P_0 &= \left( 1 - \frac{1}{L} \right)^N \\
P_1 &= N \times \frac{1}{L} \times \left( 1 - \frac{1}{L} \right)^{N-1} \\
P_{\geq 2} &= 1 - P_0 - P_1
\end{align*}
\]  

(2)

The theoretical expected number of idle slots \( (Q_0) \), the number of successful slots \( (Q_1) \) and the number of collision slots \( (Q_{\geq 2}) \) can be expressed as

\[
\begin{align*}
Q_0 &= L \times P_0 = L \times \left( 1 - \frac{1}{L} \right)^N \\
Q_1 &= L \times P_1 = N \times \left( 1 - \frac{1}{L} \right)^{N-1} \\
Q_{\geq 2} &= L \times P_{\geq 2} = L - L \times \left( 1 - \frac{1}{L} \right)^N - N \times \left( 1 - \frac{1}{L} \right)^{N-1}
\end{align*}
\]  

(3)

The identification efficiency of the RFID system \([16]\) can be expressed as

\[
S = \frac{\text{The number of successful slots}}{\text{Frame length}} = \frac{Q_1}{L} = P_1
\]  

(4)
3.2 The estimation of the number of tags

The IGA algorithm needs to estimate the number of tags before the reader starts to identify the tags. The estimation of the number of tags is critical to the performance of the collision algorithm. The closer to the actual number of tags, the better performance.

The accuracy and simplicity of the tag number estimation method are two directions to improve the method. The accuracy refers to the error between the estimated number of tags and the actual number of tags. The simplicity refers to whether the implementation method of tag number estimation is complex, whether there are certain requirements for hardware, and whether there is impact on the performance of the system.

A variety of methods to estimate the number of tags has been proposed, such as Khandelwal algorithm [17], Vogt algorithm, Low Bound algorithm, Schoute algorithm and Craio algorithm and so on [18], which use the idle slot or a probabilistic way. These methods either require a large estimation error or a very high complexity, which is very disadvantageous for the identification of the radio frequency tag. In this paper, we use a new improved method to estimate the number of tags by the ratio of collision time slots in the frame size more simply and accurately.

If \( P_c \) is the ratio of collision time slots in the frame size, that is \( P_c = \frac{Q_{\text{coll}}}{L} \), and \( e \) is the average number of tags in each collision slot, that is \( e = \frac{N - Q_{\text{coll}}}{Q_{\text{coll}}} \), it is expressed as follows through Equation 3.

\[
\begin{align*}
P_c &= 1 - \left(1 + \frac{N}{L - 1}\right) \left(1 - \frac{1}{L}\right)^N \\
e &= \frac{N - N \left(1 - \frac{1}{L}\right)^{N-1}}{L \cdot P_c}
\end{align*}
\]  

In Equation (5), the value of \( P_c \) cannot be 0. But in practice, when the number of tags is 0 and the frame length is not 0, there will be no tag collision phenomenon, that is, the ratio of the collision time slot in the frame is 0. Therefore, it is considered when \( P_c = 0, e = 0 \) in the actual situation. At the same time, if all the time slots are collision time slots, it is impossible to estimate the number of tags. Even if \( P_c = 1 \) and the value of \( e \) is variable, it is considered that \( P_c \neq 1 \) is used in this estimation method.

When the number of tags \( N \) is the same and the frame size \( L \) is different, it was found that the ratio of collision time slots in the frame size \( P_c \) and the average number of tags in each collision slot show a certain rule. If \( P_c \) is used as the abscissa and \( e \) is used as the ordinate, the relationship between \( e \) and \( P_c \) is shown in Figure 5. It is found that when the frame size is different, the relationship between \( P_c \) and \( e \) is basically the same, so it can be said that the relationship between \( P_c \) and \( e \) is independent of the frame length \( L \). Through curve fitting, the relationship between \( P_c \) and \( e \) is expressed as

\[
e = \begin{cases} 
8.826P_c^3 - 11.512P_c^3 + 5.481P_c^2 + 0.286P_c + 2.101, & 0 < P_c < 1 \\
0, & P_c = 0
\end{cases}
\]  

If the number of tags is \( N \), the number of unrecognized tags is \( N_u = N - Q_{\text{cos}} \). The number of unrecognized tags can be calculated as follows: First, counting the number of collision time slots \( Q_{\text{coll}} \) and \( P_c \) by identifying the tag in frame size \( L \). Then, the average number of tags in the collision time slot can be calculated according to the Equation (6). Finally, the number of unrecognized tags can be calculated by \( N_u = e \times Q_{\text{coll}} \). The total number of tags \( N \) is \( N = N_1 + N_u \).

Through experimental research, it is found that when different numbers of tags are identified at different frame lengths, the relationship between the ratio of the collision time slot in the frame and the average number of tags in the collision time slots is close to the curve obtained by Equation (6), and its floating maintain basic stability, which conforms to the theoretical calculation. Especially when the value of \( P_c \) is 0.2 to 0.6, the error between the experimental value and the theoretical value is the smallest, the performance is the best, and the estimated number of tags is closest to the true value.

3.3 Optimal frame size and the method of tags grouping

The selection of frame size is very important in the IGA algorithm. By Equation (4), when the frame size \( L \) is 16, 32, 64, 128, and 256 respectively, the identification efficiency of the system is shown in Figure 6.

To achieve high recognition efficiency, the number of tags corresponding to the intersection of curves is the critical value of adjusting the frame length. The relationship between the number of tags and the frame size as shown in Table 1 can be obtained [19]. Because of the hardware of the tag, the frame size needs to be taken as a power of two.
The frame size cannot increase indefinitely as the number of tags increases because of cost and system efficiency. When the number of tags exceeds 354 in the identification area, the tags will be grouped [20]. If the tags are divided into k groups, the identification efficiency of the system after grouping can be expressed as

\[ S = \frac{N}{kL} \times \left( 1 - \frac{1}{L} \right)^{\frac{N}{L} - 1} \]  

(7)

Considering the imbalance problem encountered when the tags are actually grouped, the tags are now divided into even arrays such as 2, 4, 6, and 8 for identification. When \( k \) is 2, 4, 6, 8, 10, and \( L = 256 \), the system identification efficiency after grouping is shown in Figure 7.

When we set \( k \) different values, the intersection of adjacent curves is the critical value of the group in Figure 7, and the relationship between the number of tags and the number of groups is shown in Table 2 [19].

### 3.4 Estimation of frame size for collision tags

In the IGA algorithm, when the tag collision problem occurs in a time slot, an additional frame is selected to identify the collision tags, thereby solving the collision problem.

### 3.5 The slot sequence of IGA algorithm

The idea of sequence of time slots is used in IGA algorithm. The tag contains a counter whose length is \( k \) binary bits. The tag generates the number of time slots through the counter. The maximum number of time slots is \( 2^{k-1} \).
3.6 | IGA algorithm

3.6.1 | Symbols and commands used by the IGA algorithm

(i) N: Number of tags.
(ii) L: Frame length.
(iii) group: Number of tags grouped.
(iv) s: Slot number.
(v) G: A register for storing the group number selected by the tags.
(vi) S: A register for storing the slot number selected by the tags.
(vii) Query (L): The reader sends a Query (L) command to the tags, and all tags in the reader identification area randomly select a time slot from 0 to L − 1. Meanwhile, the selected number of time slots will be saved in the register S, and the time slot sequence will be sent to the reader.
(viii) Query group (group): The reader sends a Query group (group) command to the tags, and all tags will randomly select a number as group ID from 0 to group − 1. Meanwhile, the selected group ID will be saved in the register G.
(ix) Query slot(s): The reader sends a Query slot(s) command to the tags to identify the tag selected slots.
(x) Query Add (s, Add F): If the tags collide in a time slot, the reader will send a Query Add (s, Add F) command to the collision tags to add additional frame with Add F frame size. Then, the collision tag will randomly select a time slot from 0 to Add_F − 1.

(xii) QueryAddslot(s): When the tag collides, the reader sends the Query Add (s, Add F) command to add additional frame. After the tag selects the slot in 0 ∼ Add_F − 1, the reader sends the QueryAddslot(s) command to identify the tag in the additional frame. The tag in slot s will send its own information to the reader.

(xiii) Sleep (s): When a tag is successfully identified, the reader will send a Sleep (s) command to the time slot. The tag will enter the sleep state and no longer participate in the tag identification of this round.

3.6.2 | IGA algorithm flow

The flow chart of IGA algorithm is shown in Figure 10.

Step1. The IGA algorithm needs to estimate the number of tags before it begins to identify tags. This paper uses the tag estimation method in Section 3.2 to estimate the number of tags.

Step2. If the estimated number of tags is greater than 354, tags need to be grouped. The number of groups can be determined from Table 2, recorded as group. After the reader sends a Query group (group) command to the tags, tags will randomly select a number as the group number from 0 to group − 1 and save it to the register G at the same time. After each group tag is recognised with the frame size L = 256, adds 1, and reader will continue to recognise the next group until t ∼ group − 1 ends.

Step3. If the number of tags is less than 354, the system uses dynamic frame time slot strategy to dynamically adjust the length of recognition frame through Table 1. When the reader sends a Query (L) command, tags will randomly select a time slot from 0 to L − 1. At the same time, the reader calculates the x value by the strategy of determination collision tag frame size and the additional frame size. The reader sends the command Query slot (s) to scan tags slot by slot. Three statuses will occur during the scan, which are successful identification, idle time slots, and collision slots.

(i) Successful identification: If there is only one tag in the time slot, the reader will successfully recognise it. Each time, it is successfully identified, and the reader will send a Sleep (s) command to make the tags in the sleep state and slot adds 1. If the current time slot is more than L − 1, this round of recognition ends.

(ii) Idle time slots: When the reader sends a Query (L) command, the reader can judge which time slots are idle.
time slots by the sequence of time slots returned by each tag. During the reader scanning time slot, the reader will skip the idle time slot directly.

(iii) Collision slots: If there are multiple tags in a time slot, a collision will occur [21]. The reader gets \( Add_F \) by Equation (9) and sends Query Add \((s, Add_F)\) command to identify the collision tags with frame size \( Add_F \). After the tag selects the slot in \( 0 \sim Add_F - 1 \), the reader sends the Queryaddslot \((s')\) command to identify the tag in the additional frame. Each time it is successfully identified, \( slot' \) adds 1. If the current time slot is more than \( Add_F - 1 \), the tags in the additional slot has been identified, go to the original time slot.

The pseudo code of IGA algorithm is described as follow.

1. **Input:** number of tags
2. Get the number of groups
3. **Output:** total number of slots consumed and the recognition efficiency of tag
4. **Reader Side**
5. **Initialization:** \( L, t, s, group, Add_F, used Slot, Slot, and collision Deal(s,Add_F) \)
6. **Broadcast:** Query group \((group)\)
7. **Broadcast:** Query \((L)\)
8. Get the responses of time slot from all tags, and \( Add_F \)
9. Analyze the response of time slot information

---

**FIGURE 10** Flow chart of IGA algorithm
12. **Broadcast**: Query slot \( s \)
13. Query time slots one by one
14. \( t = 1, s = 0 \)
15. while \( t \leq \) group
16. while \( s < L \)
17. if idle time slots
18. continue;
19. else if successful time slots
20. Slot ++;
21. **Broadcast**: Sleep \( s \)
22. else Collision slots
23. **Broadcast**: Query Add \( (s, \text{Add}_F) \)
24. go to used Slot = collision Deal \( (s, \text{Add}_F) \);
25. Slot = Slot + used Slot;
26. ends
27. ends
28. Broadcast end round messages
29. 
30. **Respond** Query group (group): all tags randomly select
31. group ID
32. **Respond** Query \( (L) \): all tags randomly select group ID
33. **Respond** Query slot \( (s) \): tags return information
34. **Respond** Sleep \( (s) \): the tag in slot \( s \) will enter the sleep
35. state
36. **Respond** Query Add \( (s, \text{Add}_F) \): the tag in slot \( s \) will
37. randomly select a time slot from 0 to \( \text{Add}_F-1 \), and
38. return the slot sequence

### 3.6.3 IGA algorithm example

Figure 11 shows the recognition process of eight tags and omits the process of putting the tag to sleep state. The relationship between the number of tags and the frame size is shown in Table 1, and the frame size should be 8. After the reader sends the command Query \( (8) \), tags will return the sequences of time slots 01,000,000, 00,010,000, 00,000,100, 10,000,000, 00,001,000, 00,000,100, 01,000,000 and 01,000,000. The reader will calculate that the number of idle time slots is three which are time slots 0, 1, and 5 by receiving the sequence of slot \( **0 **0 **0 **0 **0 **0 **0 **0 \).

In the process of recognition, the reader will automatically skip time slots 0, 1, and 5. When trader sends command Query slot \( (2) \) to scan the time slot 2, there was a collision between tag 3 and tag 6. Through the collision tag processing method in the IGA algorithm, we use an additional frame size \( \text{Add}_F = 4 \) to identify collision tags by calculating \( \bar{x} = \lfloor \log_2 3.301 \rfloor = 2 \) in Equations (8) and (9). Then readers send Query slot \( (3) \) and Query slot \( (4) \), tag 5 and tag 2 were identified, respectively. When readers send Query slot \( (6) \) to tags, tag 1, tag 7 and tag 8 collided. Through the processing method of collision tags, these tags have been identified. Lastly, readers send Query slot \( (7) \) to identify the tag 4 successfully. Finally, all tags are correctly identified through TGA algorithm.

### 4 ANALYSIS OF IGA ALGORITHM PERFORMANCE

The IGA algorithm proposed in this study has obvious advantages, compared with the IAA algorithm and the Yejun He algorithm. In order to verify the validity of the IGA algorithm from the experimental point of view, the IGA algorithm, IAA algorithm, and Yejun He algorithm were simulated by the simulation software MATLAB in the environment of Windows 7 operating system and 6G RAM.

The experiment has selected 5 to 1000 tags for 100 times to get the average value. When the number of tags is less than or equal to 100, the frame size is 64, and when the number of tags is greater than or equal to 100, the frame size is 256 in the IAA algorithm and the Yejun He algorithm. But the IGA algorithm adaptively adjusts the frame size by the number of tags, which is more suitable for practical applications.

#### 4.1 Analysis of total time slot consumption

The consumption of time slots is a decisive factor in the performance of the system. The smaller number of time slots, the better performance [22]. In the IGA algorithm, the total number of slots consumed is composed of two parts, which are the number of slots of the tag identified successfully and the number of slots consumed to process collision tags. The total number of time slots can be expressed as

\[
Q = \sum_i Q_{1,i} + \sum_j Q_{2,2,j} \quad (10)
\]

In the above equation, \( \sum_i Q_{1,i} \) represents the total number of time slots successfully identified, and \( Q_{1,i} \) represents the number of time slots successfully identified in the \( i \)-th frame. \( \sum_j Q_{2,2,j} \) is the sum of the number of collision slots, and \( Q_{2,2,j} \) is the number of collision slots in the \( j \)-th frame.

The IGA algorithm, IAA algorithm, and Yejun He algorithm are simulated in the process of increasing the number of tags from 5 to 1000. In IGA algorithm, when the number of tags is less than 354, tags need not be grouped. According to the estimated number of tags, dynamic frame time slot strategy is used to determine the frame length for recognition. When the number of tags is greater than 354, tags are grouped according to the grouping theory. In each group, the appropriate frame length is selected by the number of tags after grouping. The results are shown in Figures 12 and 13. It can be found that when the number of tags is more than 40, the total number of slots consumed by IGA algorithm is smaller than that of the other two algorithms. With the increase of the number of tags, the increase trend of the total number of slots consumed by IGA algorithm is obviously less than that of the other two algorithms. When
the number of tags is 1000, the number of slots consumed by the IGA algorithm is 86.9% of the IAA algorithm, which is 65.3% of the Yejun He algorithm.

### 4.2 Analysis of IGA algorithm identification efficiency

The efficiency of tag identification is another important factor to measure the performance of the system. The higher the identification efficiency, the better the system performance [23]. If defined, the ratio of the number of successfully identified slots in the i-th frame to the frame length \( L_i \) is the frame identification efficiency, that is:

\[
\eta_i = \frac{Q_{i,i}}{L_i}
\]  

(11)

Then, the recognition efficiency of the system is the ratio of the total number of time slots successfully recognised and the total number of frame lengths in the recognition process, which is expressed as:

\[
\eta = \frac{\sum_i Q_{i,i}}{\sum_i L_i}
\]

(12)

where \( \sum_i L_i \) represents the sum of all frame sizes.
The identification efficiency of identifying 100 to 1000 tags are shown in Figure 14 and Figure 15 in the IGA algorithm, IAA algorithm, and Yejun He algorithm. It can be found that when the number of tags is more than 40, the recognition efficiency of IGA algorithm is higher than that of the other two algorithms. With the increase of the number of tags, the recognition efficiency of IGA algorithm tends to be stable and remains at about 71%, while the recognition efficiency of the other two algorithms has a downward trend. Among them, when the identification efficiency of the IGA algorithm is about 71%, the IAA algorithm is 61% and the Yejun He algorithm is 48% at the best time. Obviously, the system identification efficiency of IGA algorithm is higher than the other two algorithms.

5 | CONCLUSION

An adaptive frame slotted ALOHA anti-collision algorithm based on tag grouping is proposed. After the algorithm estimates the number of tags, tags need to be identified by grouping when the number of tags is greater than 354. In the process of recognition, the reader automatically skips the idle time slot. When encountering collision slots, the system allocates additional slots to identify collision tags. The experimental results show that when the number of tags reaches more than 50, the identification efficiency of the system is stable at about 71%, which is 18.3% higher than the IAA algorithm, 47.9% higher than the Yejun He algorithm, and 90% higher than the traditional ALOHA algorithm.

ACKNOWLEDGEMENTS

This research was supported in part by grants from the National Natural Science Foundation of China under Grant 51875457, the Program of China Scholarships Council under Grant 201908615015, the International Cooperation and Exchange
REFERENCES

1. Huang, Z., et al.: A novel cross layer anti-collision algorithm for slotted ALOHA-based UHF RFID systems. IEEE Access. 7(1), 3620–36217 (2019)
2. Wang, H.G., et al.: ALOHA based anti-collision algorithm for RFID tag identification under capture environment. 2018 IEEE International Conference on Smart Internet of Things (SmartIoT), 117–121. Xi'an, China (August 2018)
3. Tian, K.J.: Application and analysis of radio frequency identification technology in identification and location for underground pipelines. Colliery Mech. Elect. Technol. 21(5), 71–73 (2018)
4. Zhang, J.R., Jiang, C., Liu, Y.L.: Multi-tags anti-collision algorithms in RFID system. J. Xi'an Univ. Post. Telecommun. 20(6), 28–32 (2015)
5. Ferreira, H.P.A., Assis, F.M., Serres, A.R.: Novel RFID method for faster convergence of tag estimation on dynamic frame size ALOHA algorithms. IET Commun. 13(9), 1218–1224 (2019)
6. Wijayasekara, S.K., et al.: A collision resolution algorithm for RFID using modified dynamic tree with Bayesian tag estimation. IEEE Commun. Lett. 22(11), 2238–2241 (2018)
7. SU, J., et al: Bit query based M-ray tree anti-collision identification protocol for RFID tags. Acta Electron. Soc. 47(2), 422–427 (2019)
8. Yang, Z.W., et al.: New Q value anti-collision algorithm based on label grouping. Comput. Sci. 45(9), 152–155 (2018)
9. Šolić, P., Radić, J., Rožić, N.: Early frame break policy for ALOHA-based RFID systems. IEEE Trans. Autom. Sci. Eng. 13(1), 876–881 (2016)
10. He, Y.J., Wang, X.Y.: An ALOHA-based improved anti-collision algorithm for RFID systems. IEEE Wireless Commun. Lett. 20(5), 152–158 (2013)
11. Xu, Y., Chen, Y.F: An improved dynamic framed slotted ALOHA Anti-collision algorithm based on estimation method for RFID systems. 2015 IEEE International Conference on RFID, 15–17. San Diego (2015)
12. Zhou, S.K., Deng, M.L.: Survey on ALOHA tag anti-collision algorithm. Comput. Eng. Appl. 53(14), 9–17 (2017)
13. Lee, S.R., Joo, S.D., Lee, C.W.: An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification. In: The Second Annual International Conference on Mobile and Ubiquitous Systems, pp. 1–7. CA, USA (Aug 2005)
14. Zhang, X.H., Hu, Y.M.: Research on a grouped adaptive allocating slot anti-collision algorithm in RFID system. Acta Electron. Soc. 44(6), 1328–1355 (2016)
15. Keat, C.S., Shyan, L.N.: Dynamic framed slotted ALOHA algorithm for RFID systems with enhanced tag estimation technique. 2013 IEEE International Conference on RFID-Technologies and Applications (RFID-TA), pp. 1–4. Johor Bahru, Malaysia (September 2013)
16. Wang, C.Y., Lee, C.C.: A grouping-based dynamic framed slotted ALOHA anti-collision method with fine groups in RFID systems. 2010 5th International Conference on Future Information Technology, 1–5. Busan, South Korea (May 2010)
17. Eom, J.R., Lee, T.J: Accurate tag estimation for dynamic framed-slotted ALOHA in RFID systems. IEEE Commun. Lett. 14(1), 60–62 (2010)
18. Liu, D., et al: Method for detecting the collision of multiple RFID tags. J. Chin. Comput. Syst. 30(9), 1890–1894 (2009)
19. Wang, H., et al: Group improved enhanced dynamic frame slotted ALOHA anti-collision algorithm. J. Supercomput. 69(3), 1235–1253 (2014)
20. Jiang, A., Li X.Y.: Tag anti-collision algorithm based on information bit grouping. Comput. Eng. 44(3), 294–300 (2018)
21. Cui, Y.H.: An adaptive multiple branches (AMB) anti-collision algorithm based on binary tree in RFID. Chin. High Technol. Lett. 27(5), 398–403 (2017)
22. Li, C., et al: Performance analysis and research of anti-collision algorithms based on query tree. Acta Electron. Soc. 27(11), 2671–2678 (2018)
23. Wei, J., Feng, X.F.: Research on a frame-slotted ALOHA based on adaptive splitting method for RFID. Comput. Technol. Develop. 22(11), 57–60 (2012)

How to cite this article: Qu, J., Wang, T. An adaptive frame slotted ALOHA anti-collision algorithm based on tag grouping. Cogn. Comput. Syst. 2021;3:17–27. https://doi.org/10.1049/ccs2.12001