IEICE TRANSACTIONS
on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024EAP1052

Publicized:2024/07/12

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

A Method to Enhance Tag Identification Efficiency Based on Tail Code Optimization Feature Sets

Xiaowu LI†, Wei CUI†, Runxin LI†, Lianyin JIA†, *and* Jinguo YOU†, *Nonmembers*

SUMMARY

Radio Frequency Identification (RFID) is crucial for the Internet of Things, with a key challenge being the efficient prevention of tag collisions for quick identification. This paper presents a novel approach for rapid tag recognition in small to medium-sized warehouses, combining a tag optimization feature set with a tail code recognition mechanism. To minimize the frequency of scanning for duplicate tags and reduce the occurrence of collisions, we construct an optimization feature set based on the reader's position. This set helps in assessing the likelihood of tag repetition through its linear variation. It also incorporates a tail code mechanism that recognizes only the last 22 digits of the tag's EPC code, significantly speeding up identification. The tail code length is dynamically adjusted based on the number of tags to maintain uniqueness. Simulation results indicate that our approach significantly reduces the identification of duplicate tags and minimizes the instances of collisions.

key words: *Tag optimization feature set, Tail code recognition mechanism, Anti-collision, Quick identification, RFID.*

1. Introduction

Radio frequency identification (RFID) has cemented its place as a transformational technology within the Internet of Things (IoT) landscape. RFID has risen in popularity as wireless communication technology has evolved, providing remarkable value across a wide range of industries. RFID enables remote and automatic object identification and tracking. When compared to traditional barcode systems, it has better readability, more data capacity, and more resilience to interference. As the Internet of Things evolves, RFID is projected to stay prominent, fostering innovation across a variety of industries. However, the widespread use of RFID presents technical and managerial obstacles [1]. Handheld readers have grown in popularity in a variety of RFID applications due to their ease of use and low cost of data collection. Passive RFID systems need efficient tag recognition technologies to enhance system efficiency and prolong reader battery life [2].

The grouping based bit-slot ALOHA protocol [3] effectively deals with collisions and idle slots but has an efficiency problem, leading us to introduce the optimization feature set concept. This set is intended to assess the tag environment before implementing the grouping-based protocol in order to minimize collisions and enhance system efficiency [4], [5], [6]. To determine this optimization feature set, we analyze the environment's characteristics, including tag density, movement patterns, and previous collision rates. Extreme tag quantities, whether minimal or excessive, lead to a notable reduction in the average number of recognitions, greatly affecting system efficiency [7], [8], [9]. In instances where tags are in a continuous response state and collisions occur, if the reader does not process these tags quickly, they may be missed [10], [11]. An increased number of simultaneous responses raises the chances of collisions, reducing recognition efficiency even more [12]. This approach allows us to tailor the protocol's operation to the specific conditions of the tag environment, ensuring more efficient and accurate tag recognition.

To address the challenge of tag collisions and inefficiency in tag recognition, particularly in environments with a dense tag population. The optimization feature set enables us to compare the similarity between the tags at the reader's current location and the previously recognized feature sets. The novelty of this idea is reflected in the fact that we can evaluate the duplication rate of the tags in the set in advance, so we further reduce the occurrence of tag collisions in advance before the grouping based bit-slot ALOHA protocol proceeds [13], which is consistent with the idea of the grouping based bit-slot ALOHA protocol. Therefore, the introduction of this novel approach allows us to accelerate the efficiency of tag recognition while further reducing collisions and energy consumption.

This paper aims to refine existing tag anti-collision algorithms [3], [8], [14]. We have introduced a tail code optimization feature set. The function of the optimization feature set is to further prevent collision occurrences and reduce duplicate responses and identification of the same tag. To measure the redundancy between tail code feature sets, we employ the Jaccard similarity metric. In addition, the tail code recognition mechanism is used to improve the efficiency of tag recognition. To avoid the problem caused by the short number of tail code bits, we use the birthday probability model to evaluate the possibility of tag tail code duplication. The primary objective of this paper is to reduce the response and recognition of duplicate tags based on the predefined tail code feature set, thereby further preventing conflicts. Experimental results confirm the effectiveness of our method in tag identification within medium and small-sized warehouse environments.

2. Bit-Slot ALOHA algorithm based on dynamic grouping

In the bit-slot ALOHA algorithm, let L denote the number of bit-slots. Considering a bit sequence length of 128 bits, $L = 128$. Let the probability of a tag selecting any given slot be $P = 1/L$ [3], and let the total number of tags be represented by n [6].

The probability that m tags choose the same slot simultaneously is denoted by P_m, and this probability follows a binomial distribution:

$$P_{\mathrm{m}} = C_{\mathrm{n}}^{\mathrm{m}} P^{\mathrm{m}} (1 - P)^{\mathrm{n-m}} \tag{1}$$

T refers to a certain time slot, which is a randomly specified one. No tags select time slot T, and the probability of free time slot is:

$$P_{\mathrm{free}} = C_{\mathrm{n}}^{0} P^{0} (1 - P)^{\mathrm{n}} \tag{2}$$

The probability that only one tag successfully selects time slot T is:

$$P_{\mathrm{succeed}} = C_{\mathrm{n}}^{1} P^{1} (1 - P)^{\mathrm{n-1}} \tag{3}$$

The probability of multiple tags selecting time slot T is:

$$P_{\mathrm{collided}} = 1 - (P_{\mathrm{free}} + P_{\mathrm{succeed}}) \tag{4}$$

If a certain time slot is selected by multiple tags, it means that the time slot has collided, and the mathematical expectation of the number of bit-slots that have collided is:

$$\begin{aligned} E_{\mathrm{collided}} &= L \times (P_{\mathrm{succeed}} + P_{\mathrm{collided}}) \\ &= L \times (1 - P_{\mathrm{free}}) \end{aligned} \tag{5}$$

The number of tags in the entire reader recognition area is:

$$\mathrm{n} = \ln\left(1 - \frac{E_{\mathrm{collided}}}{L}\right) / \ln\left(1 - \frac{1}{L}\right) \tag{6}$$

The mathematical expectation for a bit slot without collision is:

$$E_{\mathrm{succeed}} = L \times P_{\mathrm{succeed}} \tag{7}$$

The recognition efficiency is:

$$\begin{aligned} \eta\,(\mathrm{n}) &= E_{succeed} / 1 + E_{collided} \\ &= n \times \left(1 - \tfrac{1}{L}\right)^{n-1} \Big/ 1 + L \times \left[1 - \left(1 - \tfrac{1}{L}\right)^{n}\right] \end{aligned} \tag{8}$$

As depicted in Fig. 1, the system's identification efficiency decreases when the number of tags is either very low or exceedingly high, resulting in numerous slots being underutilized or prone to collisions. When there are 15 tags, the system identification efficiency is at its highest, at 0.8838. Utilizing dynamic grouping to categorize tags into designated groups can enhance performance. We provide a bit-slot ALOHA protocol that utilizes dynamic grouping to control the number of responsive tags [3] and introduces a grouping parameter represented by G.

Upon the receipt of a query command, tags generate random values ranging from 0 to $2^G - 1$, effectively categorizing them into distinct groups [13]. Only tags generating a random value of 0 will respond immediately. If there are fewer collisions in bit slots, the protocol reduces the number of groups by reducing G. In contrast, if collisions occur frequently, G will increase.

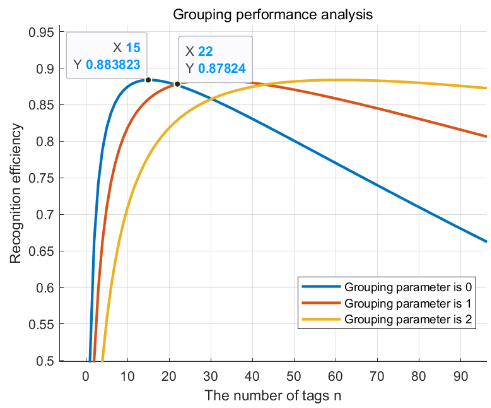

Fig. 1 Grouping performance analysis.

Table 1 Adjusting G Value.

Number of collision slots	Adjusting G value
$> E_{collided\_point}$ (20)	$G = G + 1$
$E_{collided\_\mathrm{reduce}}$ (11) ~ $E_{collided\_point}$ (20)	$G = G$
$< E_{collided\_\mathrm{reduce}}$ (11)	$if\ G = 0, G = 0$ $if\ G > 0, G = G - 1$

Upon juxtaposing the curve for a grouping of 0 with that of 1, an intersection point emerges, symbolized as N_{point}. At N_{point}, recognition efficiency is consistent for grouping parameters of 0 and 1. We find that N_{point} is approximately 21.51, which rounds up to 22. At $N_{\mathrm{point}} = 22$, the efficiency $\eta(22)$ is about 0.87824. Using N_{point}, we calculate that $E_{\mathrm{collided\_point}}$ is approximately 20.29. Consequently, the expected number of collision bit slots is 20. When the reader detects more than 20 collision bit slots, it stops the current query and increases the value of G by 1 to enhance the system's efficiency. By utilizing Eq.(9), we can calculate the collision bit slots as grouping decreases.

$$\eta\,(N_{\mathrm{point}}) = \eta\,(N_{\mathrm{reduce}}) \tag{9}$$

We suppose that N_{reduce} should be inferior to 22, prompting us to evaluate potential values between 1 and 21. Utilizing Eq.(7), our objective is to ascertain a value satisfying Eq.(9) or one that closely aligns with it. Post-calculation, we find that N_{reduce} is approximately 12.21, which implies that when the number of tags is 12, the system efficiencies are similar. Therefore, when the grouping is reduced, the number of collision bit slots $E_{\mathrm{collided\_reduce}}$ is approximately 11. From this, we can derive Adjustment Grouping Table 1.

3. Tail code recognition mechanism

In the grouping based bit-slot ALOHA protocol, a reserved sequence spans 128 bits. Given that the tag's EPC code is also 128 bits, we allocate the last 22 bits as the tail code and the initial 106 bits as the residual code. This paper presents a strategy in which a tag transmits only its tail code to the reader rather than the entire EPC code. If there is a repetition in the tail codes, the reader will identify the residual code. In systems without this tail code recognition mechanism, identifying the entire 128-bit sequence occupies one time slot. However, our proposed method only identifies the 22-bit tail code, which takes approximately 0.1719 of a time slot. In other words, we identified almost six tags in one time slot [2], [15]. The tail-code-centric technique significantly reduces the time required [16]. The time cost comparison is shown in Fig. 2.

The parameter G in Fig. 2 represents the grouping parameter. The upper graph illustrates the efficiency of tag recognition at grouping levels of 0, 1, or 2. As the number of tags grows, the efficiency of recognition reduces due to collisions, so we need to change the number of groupings. The time cost in the following figure is the number of time slots required to recognize a tag with and without a tail code when the grouping is 0, 1, or 2. The following figure mainly expresses the comparison of the time cost with and without the tail code recognition mechanism when the time slot required to recognize the tag is fixed. All time units in this paper have been converted to the number of time slots. In addition, the focus at $X = 160$ is the number of time slots required to recognize 160 tags with the grouping parameter of 0, with and without tail code. This is an illustrative point for comparing the time cost visually.

However, the brief nature of tail codes significantly raises the likelihood of repetition. Addressing this issue is crucial. The repetitive tail codes can significantly hinder the authentication of tag information, leading to verification conflicts and undermining the system's identification efficiency. Consequently, it is vital to strike a balance between optimizing identification efficiency and reducing conflict risks. A key challenge lies in developing robust strategies to prevent the recurrence of identical tail codes.

Assume there are N tags in the area to be recognized, with each tag's tail code length being l. If each tag's tail codes are dispersed randomly, the birthday paradox can be used to calculate the chance of tail code duplication.

Specifically, the problem is this: Suppose we have N tags, and each tag's tail code is randomly distributed in the range of 1 to 2^l with equal probability. The purpose is to determine the number of these tags with the same tail codes.

A.Set simulation parameters:

- Tail code range:1-2^l
- The number of tags:1000
- Simulation iterations: 10,000 (This volume ensures a reliable estimate.)

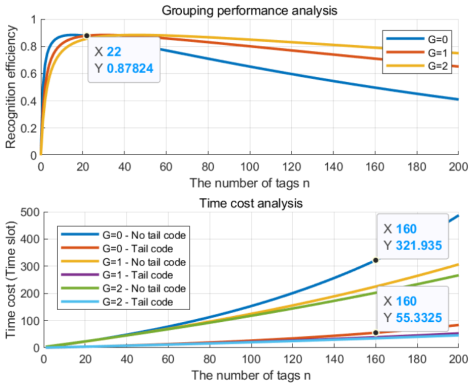

Fig. 2 Time cost analysis.

Table 2 The Average Number of Repeated Tail Codes.

2^l \ A, N	200	500	1000	1500	2000
l = 16	0.6280	3.8079	15.1495	33.9720	60.1657
l = 18	0.1534	0.9865	3.7561	8.6261	15.3122
l = 20	0.0354	0.2338	0.9225	2.1800	3.7861
l = 22	0.0084	0.0480	0.2252	0.5920	0.9564

Table 3 The Probability of Tags Having a Repeating Tail Code.

2^l \ p, N	200	500	1000	1500	2000
l = 16	0.0031	0.0076	0.0151	0.0226	0.0301
l = 18	0.0008	0.0020	0.0038	0.0058	0.0077
l = 20	0.0002	0.0005	0.0009	0.0015	0.0019
l = 22	0.0000	0.0001	0.0002	0.0004	0.0005

B.Monte Carlo simulation:

- For each iteration:

a. Assign tail codes to 1000 tags randomly according to the range of tail codes.

b. By counting the frequency of each tail code, it is possible to determine which tail codes are shared and how many tags share the same tail code.

c. For each simulation, calculate the total number of tags with duplicate tail codes.

C.Analysis and results:

- From the accumulated simulation results, determine the average number of tags with duplicate tail codes as well as the probability of tail code duplication.

If we determine the number of tags (N) and the tail code length (l), we can theoretically calculate the probability of the same tail code. However, this actually involves the problem of the birthday paradox. Therefore, we utilize the birthday probability to determine the likelihood of a tag having a repeating tail code of (p). We use Monte Carlo simulation to deal with some birthday problems, such as determining how many tags in a large group have a repeated tail code with other tags. This is because a straight mathematical study of certain situations might be difficult or confusing. The Monte Carlo method provides a practical solution, especially in cases where direct analysis is not feasible.

Tables 2 and 3 show the number of tags (N), the length of the tail code (l), the probability of a tag having a duplicate tail code (p), and the average number of repetitions of tail code (A). After conducting 10,000 Monte Carlo simulations, it was noted that the frequency of tail code duplicates rises as the number of tags grows. Increasing the number of tail code bits can effectively reduce these repetitions. In small to medium-sized warehouses, the length of the tail code should be adjusted based on the number of tags to minimize or eliminate the occurrence of duplicate tail codes. Even if a tag has a duplicate tail code, the time required to recognize the remaining code is negligible [17].

In designing the scheme for implementation in small and medium-sized warehousing environments, we adjusted the tag parameters in the simulation to 200–2000. This quantity of tags already meets the requirements of small and medium-sized warehouses. In practice, dynamically altering the tail code's bit length based on tag count ensures a negligible chance of tail code duplication.

For this study, the tail code's length was set at 22 bits, with a total of 1,000 tags. Referring to Table 3, following 10,000 Monte Carlo simulation iterations, the average number of repeated tag tail codes stands at a mere 0.2252, with a probability of 0.0002. This indicates that the likelihood of tail code repetitions among tags is extremely low. If a duplicate tail code occurs in this paper, the time needed to recognize its residual code has a negligible effect on the system's efficiency [12], [17].

4. Tag optimization feature set

The bit-slot ALOHA algorithm, enhanced by dynamic grouping, is effective in reducing collision issues in tag-based systems. However, this approach is not without its drawbacks, particularly regarding time efficiency. Changes in the number of tags or environmental conditions need adjustments in grouping and slot allocation, resulting in longer response times. A significant issue occurs when the reader's detection zone has a significant overlap. When multiple active tags respond simultaneously, the likelihood of a collision occurring rises. If tags that consistently respond are not processed quickly, their data transmission protocols must be reorganized, negatively impacting system performance [3]. As a result, it is critical to avoid repeated tag recognition and optimize the movement strategy, allowing the reader to transition more quickly to new areas.

In response, we introduce a tag tail code optimization feature set. Assuming the tag's EPC code is 128 bits, we designate the last 22 bits as the tail code and the remaining 106 bits as the residual code. The collective tail codes within the reader's detection range are combined into a feature set. Every time the reader moves to a new position, a new tail code feature set is generated. A series of such tail code sets form the comprehensive optimization set. For every new location, the reader evaluates the similarity between the current feature set and previously identified ones. If this similarity exceeds a predetermined threshold, it suggests a high level of duplication in tag identification, leading the reader to skip recognizing that particular set.

The feature set for each reader position is saved in a tail code table while tag scanning is being done simultaneously. RFID systems inherently lack memory. Thus, if the reader powers off or completes a loop along its movement trajectory and unidentified tags remain, we cannot tell whether the tags that re-enter the recognition range have previously been identified. To make this determination, we can go to the tail code table. If a tag's tail code is already present in the tail code table, it will be handled silently [14].

We use the Jaccard similarity coefficient to assess feature set similarities. This metric, defined as the ratio of the intersection to the union of two sets, helps determine thresholds to reduce duplicate tag identification [18].

Assuming the reader's recognition range is a circle with a radius of r, there could be recognition blind spots. So we must compute on the assumption that the reader can cover the entire recognition range and determine the position of the next reader based on the different overlap rates. We employed binary search to optimize overlap rates, ensuring maximal recognition and minimal redundancy. Initial tests at 50% overlap revealed reader blind spots. Subsequent testing showed overlap rates ranging from 60% to 90%. S_{area} is the area of overlap between two sets. D represents the distance traveled by the reader along the recognition path. S_{new} represents the additional area (un-overlapped section) following each reader movement. We determined the optimal overlap rate using a defined similarity computing methodology.

Step 1. Determine the reader's movement distance for varying overlap rates. The calculation of the overlap rate is as follows:

$$R_{\text{overlapping}} = \frac{S_{\text{area}}}{\pi \times \mathrm{r}^2} \tag{10}$$

Depending on the overlap rate, the distance moved by the reader is calculated as follows:

$$D_{\text{reader}} = \mathrm{r} \times \left(1 - R_{\text{overlapping}}\right) \tag{11}$$

Step 2. Calculate the additional area (un-overlapped section) following each reader movement, ensuring complete coverage of the recognition area.

$$S_{\text{new}} = \left(\pi \times \mathrm{r}^2 - S_{\text{area}}\right) \times \frac{L}{\mathrm{r} \times \left(1 - R_{overlapping}\right)} \tag{12}$$

Step 3. Identify the overlap rate that minimizes the total newly added area, and derive the Jaccard similarity based on this rate.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{S_{\text{area}}}{S_{new} + \pi \times \mathrm{r}^2} \tag{13}$$

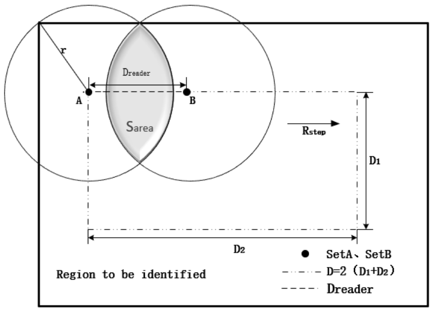

Fig. 3 Reader moving model diagram.

Figure 3 depicts the reader mobility model in an RFID system. Assume that the tags are randomly scattered around the region to be recognized. In Fig. 3, D indicates the distance traveled by the reader to complete one revolution along the recognition route, while R_{step} represents the reader's speed. A and B are the sets where the reader is positioned, and D_{reader} represents the distance from the reader traveling from point A to point B. S_{area} refers to the area of the shaded part. $R_{overlapping}$ denotes the overlapping rate of sets A and B, which is the ratio of shaded to total area. S_{new} represents the additional area (un-overlapped section) following each reader movement.

Iterative computations identify an optimal overlap rate of 81% for minimizing tag duplicate identification. The simulation results, shown in Fig.4, affirm the overlap degree and Jaccard similarity exhibit a clear linear relationship and positive correlation. Utilizing an overlap rate of 81%, the corresponding Jaccard similarity is approximately 0.6807, with a threshold of around 0.681. When the duplication rate of tags between two sets reaches 0.681, it indicates a high degree of environmental similarity. Consequently, we will abandon the recognition of the new set, and the reader will continue to advance. This optimization effectively shortens the total time spent on dispatching queries, awaiting responses, and managing collisions by avoiding responses to duplicate tags and reducing the number of queries.

A Jaccard similarity of 0.681 (equivalent to an 81% overlap rate) represents a balanced choice derived from the movement and coverage properties of circles, aimed at ensuring complete coverage of the area to be recognized while minimizing the recognition of duplicate tags. However, this conclusion assumes specific conditions, such as the radius of the circle r and the area of coverage S_{area}. In fact, for different r, it is necessary to make adjustments to find the optimal overlap rate. By optimizing the relationship between the reader's travel distance D_{reader} and the overlap rate, the best strategy can be found for different system configurations under different conditions.

Figure 5 shows that at low tag counts, the optimization feature set technique greatly reduces time expenditure. As the number of tags increases, so does the overall system time required to process conflicts and identify tags, rendering the time savings from employing the optimization feature set approach less obvious but still considerable. Specifically, the 'Time Cost' on the vertical axis is defined as the time taken by the reader to identify all the tags in the entire region to be identified. Our calculations convert the units to slots. As shown in the figure, when the tag count reaches 300, the time cost of the two strategies is 5387 and 4602, respectively. This means that the strategy for the optimization feature set delivers a 24.6% time reduction, and for 700 tags, the drop is around 8.7%.

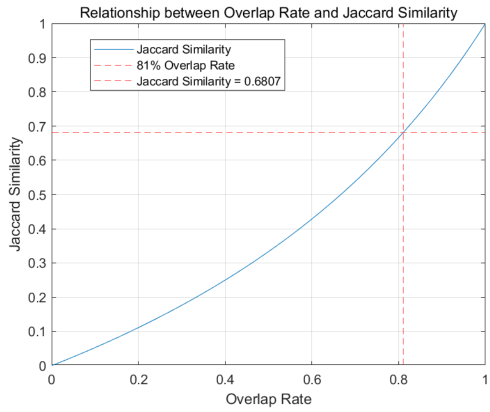

Fig. 4 Relationship between Overlap Rate and Jaccard Similarity.

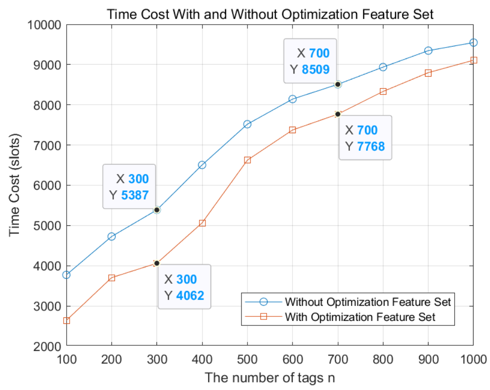

Fig. 5 Time cost with and without Optimization Feature Set.

5. Improved algorithm based on the tail code optimization feature set

Due to the presence of optimization feature sets, the reader may only need to send fewer queries to cover the entire area, avoiding repetitive responses and recognition. This reduces the total time needed for sending queries, waiting for responses, and completing recognition, while also further preventing the occurrence of collisions [12]. At the same time, the approach using the tail code recognition mechanism only takes 17.2% of the time compared to the approach that does not use it. As a result, we offer an improved approach that focuses on the tail code optimization feature set.

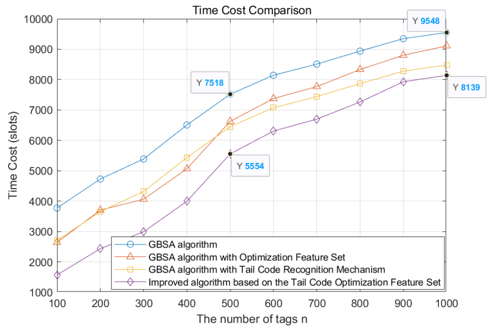

Fig. 6 Time cost comparison.

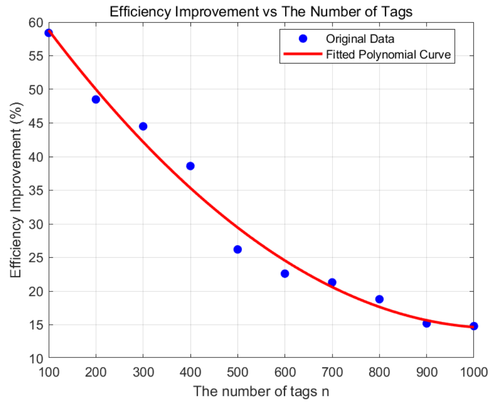

Fig. 7 Efficiency improvement.

This revamped Bit-Slot algorithm is a fusion of the bit-slot ALOHA algorithm based on dynamic grouping, the tag optimization feature set, and the tail code identification mechanism. The implementation steps of the algorithm are as follows:

Step 1. The reader sends a query command to start a new identification cycle. This command will contain a parameter G, based on which the tags will be grouped.

Step 2. Tag sends its tail code.

Step 3. The reader computes the Jaccard similarity between the present and preceding tail code feature sets. The initial position of the Jaccard similarity is set to 0.

Step 4. Should similarity surpass the 0.681 threshold, the current feature set's identification is sidelined, prompting the reader's continued movement.

Step 5. If the similarity is below the threshold, tags randomly select a value within the range of $0 \sim 2^G - 1$. Only those landing on 0 instantly respond, transmitting a reservation sequence.

Step 6. The reader detects the reservation sequences. If collision bit-slot counts oscillate between the predicted count and the lower boundary:

6.1 If no tag's tail code is duplicated, the reader logs the colliding slots. If only one tag responds, the tail code of that tag is captured and subsequently deactivated. If multiple tags respond, these tags will have to wait for the next query cycle.

6.2 When the tail code of a tag is detected to be duplicated, it is necessary to further identify the residual code. If only one tag responds based on the recorded collision slots, all EPC code is captured and deactivated. Otherwise, wait for the next query cycle.

Step 7. When the number of collision bit slots stray from expected counts and the lower boundary, G is recalibrated via Table 1, altering the group count.

Step 8. The above steps cyclically persist until all tags are discerned.

The time cost comparison is shown in Fig. 6. The GBSA algorithm represents the bit-slot ALOHA algorithm based on dynamic grouping. Predefined tag optimization feature sets are instrumental in tailoring the reader protocol, reducing both collisions and the frequency of scanning duplicate tags. When compared to conventional algorithms, the approach that integrates optimization feature sets, as well as the tail code identification mechanism, takes significantly less time. With a batch of 1000 tags, the time cost of the GBSA algorithm is 9458 time slots, while the improved technique is 8139 time slots. The new method decreases the time cost by 1409 slots, resulting in an efficiency improvement of around 14.8%. For 500 tags, the efficiency improvement is around 26.2%. As a result, it is obvious that the fewer the tags, the more time our algorithm saves and the higher its efficiency.

Figure 7 illustrates the trajectory of the algorithm's efficiency improvement as the number of tags rises. Further analysis reveals that, as the number of tags increases, the extent of efficiency benefits decreases but eventually stabilizes. This discovery is especially relevant to medium and small warehouses, where the number of tags is quite modest. This enhanced technique not only improves the system's overall performance, but also prevents responding to duplicate tags while being compatible with previous collision resolution methods. In summary, our approach effectively prevents the occurrence of collisions by performing predefined operations before reading the tags, proving its effectiveness in enhancing recognition efficiency.

6. Conclusion

This study presents an algorithm enhanced with the tail code optimized feature set, skillfully designed to reduce idle and collision slots through reservation sequences and grouping tactics. By refining the reading strategy with optimization feature sets, we can reduce collisions further. This optimization effectively shortens the total time spent on dispatching queries, awaiting responses, and managing collisions by avoiding responses to duplicate tags and reducing the number of queries. Additionally, the algorithm improves the reader's navigational strategy, resulting in less time spent on unnecessary inquiries and movements. We give actual experimental data to illustrate the efficacy of our solution, which shows a considerable reduction in overall system load when compared to previous methods. Specifically, by combining tail code optimization feature sets with grouping techniques, our program reduces the likelihood of collisions, increasing system throughput. Empirical results from comparative tests demonstrate the huge efficiency gains realized with our strategy.

7. Acknowledgments

The authors would like to thank the National Natural Science Foundation of China for their support under Grant Nos. 62262035 and 62062046.

References

[1] R. Want, "An introduction to rfid technology.," IEEE Pervasive Computing, vol.5, no.25-33, 2006.
[2] J. Su, Z. Sheng, V.C.M. Leung, and Y. Chen, "Energy efficient tag identification algorithms for rfid: Survey, motivation and new design," IEEE Wireless Communications, vol.26, no.3, pp.118–124, 2019.
[3] C.P. Wong and Q. Feng, "Grouping based bit-slot aloha protocol for tag anti-collision in rfid systems," IEEE Communications Letters, vol.11, no.12, pp.946–948, 2007.
[4] L. Zhang, W. Xiang, X. Tang, Q. Li, and Q. Yan, "A time- and energy-aware collision tree protocol for efficient large-scale rfid tag identification," IEEE Transactions on Industrial Informatics, vol.14, no.6, pp.2406–2417, 2018.
[5] J. Su, Z. Sheng, A.X. Liu, Z. Fu, and C. Huang, "An efficient missing tag identification approach in rfid collisions," IEEE Transactions on Mobile Computing, vol.22, no.2, pp.720–731, 2023.
[6] L. Zhang, W. Xiang, and X. Tang, "An adaptive anti-collision protocol for large-scale rfid tag identification," IEEE Wireless Communications Letters, vol.3, no.6, pp.601–604, 2014.
[7] H. Wang, S. Xiao, F. Lin, T. Yang, and L.T. Yang, "Group improved enhanced dynamic frame slotted aloha anti-collision algorithm," The Journal of Supercomputing, vol.69, pp.1235–1253, 2014.
[8] A. Abbasian and M. Safkhani, "Cncaa: A new anti-collision algorithm using both collided and non-collided parts of information," Computer Networks, vol.172, p.107159, 2020.
[9] L. Zhang, W. Xiang, and X. Tang, "An adaptive anti-collision protocol for large-scale rfid tag identification," IEEE Wireless Communications Letters, vol.3, no.6, pp.601–604, 2014.
[10] L. Zhang, W. Xiang, and X. Tang, "An efficient bit-detecting protocol for continuous tag recognition in mobile rfid systems," IEEE Transactions on Mobile Computing, vol.17, no.3, pp.503–516, 2018.
[11] L. Zhang, W. Xiang, I. Atkinson, and X. Tang, "A time-efficient pair-wise collision-resolving protocol for missing tag identification," IEEE Transactions on Communications, vol.65, no.12, pp.5348–5361, 2017.
[12] D. Liu, Z. Wang, J. Tan, H. Min, and J. Wang, "Aloha algorithm considering the slot duration difference in rfid system," 2009 IEEE International Conference on RFID, pp.56–63, 2009.
[13] J. Su, Z. Sheng, A.X. Liu, Y. Han, and Y. Chen, "A group-based binary splitting algorithm for uhf rfid anti-collision systems," IEEE Transactions on Communications, vol.68, no.2, pp.998–1012, 2020.
[14] C. Yihong, F. Quanyuan, Z. Ma, and T. Liu, "Multiple-bits-slot reservation aloha protocol for tag identification," IEEE Transactions on Consumer Electronics, vol.59, no.1, pp.93–100, 2013.
[15] C. Floerkemeier, "Bayesian transmission strategy for framed aloha based rfid protocols," 2007 IEEE International Conference on RFID, pp.228–235, 2007.
[16] W.T. Chen, "An accurate tag estimate method for improving the performance of an rfid anticollision algorithm based on dynamic frame length aloha," IEEE Transactions on Automation Science and Engineering, vol.6, no.1, pp.9–15, 2009.
[17] K. Pal, "A novel frame-slotted aloha algorithm for radio frequency identification system in supply chain management," Procedia Computer Science, vol.184, pp.871–876, 2021.
[18] H. Lv, X. Li, L. Jia, R. Li, J. Ding, and J. You, "Suffix-based double reader tag movement rfid anti-collision method," Journal of Cloud Computing, vol.10, no.1, p.12, Jan 2021.

Xiaowu Li is the first author of this paper. He graduated with a Ph.D. in Computer Applications from Southwest Jiaotong University in 2015. His main research interests include Radio Frequency Identification (RFID) technology, blockchain, and artificial intelligence. He is currently serving at Kunming University of Science and Technology in Yunnan, China. Contact him at lxwlxw66@kust.edu.cn.

Wei Cui is the second author of this paper and is currently pursuing a Master's degree in Computer Technology at Kunming University of Science and Technology, Yunnan, China. Contact him at 20222204122@stu.kust.edu.cn.

Runxin Li graduated in 2013 with a Ph.D. in Applied Mathematics from Yunnan University, focusing primarily on machine learning and optimization algorithms. Currently, the author holds a position at Kunming University of Science and Technology in Yunnan, China, and serves as the corresponding author of this paper. Contact him at rxli@kust.edu.cn.

Lianyin Jia obtained a doctoral degree from South China University of Technology in 2012, with a primary focus on Artificial Intelligence and Data Mining. Currently, Li serves as an Associate Professor at the Kunming University of Science and Technology in Yunnan. Contact him at lianyinjia@kust.edu.cn.

Jinguo You received his Ph.D. from South China University of Technology in December 2008, with a primary focus on big data analysis, data warehousing, and machine learning. He is currently a professor at Kunming University of Science and Technology in Yunnan, China. Contact him at jgyou@kust.edu.cn.