Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6883591, 10 pages
https://doi.org/10.1155/2021/6883591

Research Article

Research on RFID Anticollision Algorithms in Industrial Internet of Things

Haizhong Qian

Information Engineering College, Jiangsu Maritime Institute, Nanjing 211199, China

Correspondence should be addressed to Haizhong Qian; hzhqian@jmi.edu.cn

Received 1 October 2021; Accepted 30 November 2021; Published 14 December 2021

Academic Editor: Yinghui Ye

Copyright © 2021 Haizhong Qian. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a perception enabling technology of the Internet of Things, RFID can quickly identify target objects. The tag-to-tag collision problem seriously affects the identification performance of the RFID system, which causes the reader to be unable to accurately identify any tag within the specific time. The mainstream anticollision algorithms are limited by the performance bottleneck under the standard framework. In this paper, we analyze the features and merits of three kinds of algorithms in detail and propose a new algorithm architecture for RFID anticollision. Through the extensive experimental results comparison, we prove that the new architecture is effective to improve the performance of DFSA algorithms. Finally, we summarize the future research trends in the RFID anticollision algorithms.

1. Introduction

Radio Frequency Identification (RFID) [1–3] technology is an emerging backscatter communication technology [4], which uses radio frequency signals to carry out information exchange in the wireless channel and realizes the noncontact identification between objects. A typical RFID system consists of readers, tags, and backend components. In many applications based on RFID technology, a mass of passive RFID tags is deployed in the RFID system. Due to the nature of passive tags, tags cannot communicate with each other, so all tags can only receive signals from readers. Since multiple tags share the same wireless channel to communicate with readers, when multiple tags send data to readers at the same time, the phenomenon of collision between more tags will occur, resulting in the failure of readers to identify tags. Obviously, the collision brings great challenges to information collection. Therefore, the research on RFID electronic tag information collection is very suitable for the background of the development of the times [5, 6] and has important reference value and practical significance. Therefore, when there are multiple tags at the same time in the covering area of the reader, the reader must use anticollision algorithm to improve the efficiency of tag recognition.

The current RFID multitag anticollision algorithms can be mainly divided into three categories, namely, probabilistic algorithms [7–10], deterministic algorithms [11–15], and hybrid algorithms [16–18]. Probabilistic algorithms are mainly derived from Aloha-based ideas. An Aloha-based algorithm takes the way of tag answering first and lets the tag randomly select a time period to respond to the query request of reader. If only one tag responded to the query in the current time period, the tag was successfully identified. In the Aloha-based algorithm, the most commonly used is the dynamic framed slotted Aloha (DFSA) algorithm. The DFSA algorithm formulates the concept of frame length (the frame length is equal to the number of time slots allowed by the tag), and then, the tag that receives the reader command will randomly select a time slot in a frame to respond and reply with its own ID. When a frame is over, the reader uses mathematical methods to estimate the number of remaining tags based on the slot status counted in the previous frame and starts the next frame identification until all tags are successfully recognized. The performance bound of the DFSA algorithm is 0.368 [19]. All DFSA algorithms are easy to implement and have low equipment cost but may have tag starvation. Deterministic algorithms are also called tree-walking algorithms. The core idea of this type of algorithm is to use bit tracking technology [20, 21] to lock the specific location of ID information collisions, so that the reader can adjust the query prefix in time to successfully identify subsequent tags. As the query progresses, the tree splits and all the child nodes of the query tree are retrieved. In other words, as long as there are enough queries, readers can accurately query the ID of each tag. Thus, tree-walk algorithm has a 100% recognition rate and can avoid the problem of tag starvation. However, once the tag IDs are not evenly distributed, the tree-based algorithm will generate many collision slots and lead to a gradual deterioration in performance. In order to maximize the multitag recognition performance of the RFID system, many researchers have merged the advantages of different types of algorithms and proposed a series of hybrid anticollision algorithms. The representative algorithms are BSTSA [16], BTSA [17], and GBSA [18]. Compared with traditional anticollision algorithm, the above algorithms have been significantly improved in tag recognition performance, but at the same time, they also caused some problems, such as high hardware cost, compatibility with existing RFID standards, and other issues. It is noted that there are also two kinds of collisions in the RFID systems, namely, reader-to-reader collision. The reader can use some scheduling methods to avoid such collisions. And this kind of collision is not the focus of this paper.

This paper mainly analyzes the performance of the typical representatives of the existing several types of anticollision algorithms and summarizes and analyzes the performance bottlenecks and other shortcomings of the existing methods. Based on the analysis, we propose a new algorithm architecture in which the system throughput can break through the bottleneck of existing DFSA algorithms. Different from the traditional DFSA algorithms, the proposed new method can separately cope with each collision slot with the independent small-size frame. Through theoretical derivation and analysis, we proved that the proposed new method can make the anticollision algorithm break the performance limit under the constraints of the existing framework. Finally, the future research trends of anticollision algorithms are summarized.

2. Related Works

Taking into account factors such as RFID equipment cost and implementation complexity, currently, in RFID readers, multitag conflict avoidance technologies based on time division multiplexing are mainly used, which mainly include three types of methods based on Aloha, binary splitting, and query tree. Among the methods based on Aloha, currently, the most widely used is the dynamic framed slotted Aloha algorithm, which has been adopted by the UHF international standard EPC C1 Gen2. At present, most research focuses on two aspects. One is to maximize the utilization of time slots by dynamically adjusting the frame length of each round, and the other is to avoid waste of resources by terminating low-utilization frames early. The EACAEA [22] algorithm proposed by Chen considers the difference in time duration between different types of time slots to optimize the frame length to maximize system throughput. Chen's research shows that when the length of the collision slot is 5 times that of the idle slot, the optimal frame length is 1.89 times the number of tags to be identified. However, this method only adjusts the frame length once in one round of recognition, which causes its performance to show large fluctuations when the number of tags changes, and its stability is poor. Su et al. [4] designed a frame length adjustment strategy based on time and energy saving and, based on this strategy, proposed a multitag identification algorithm named TES-FAS suitable for the EPC C1 Gen2 standard. The TES-FAS algorithm combines subframe observation and adaptive frame adjustment mechanisms and can achieve better recognition performance under different reader parameter configurations. However, the frame length optimization of the reader in TES-FAS algorithm is based on a static RFID environment, which makes it not suitable for dynamic RFID environments. Taking into account the mobility of tags, Zhu et al. [23] designed a scheduling-based RFID anticollision algorithm that satisfies a high recognition rate under a certain tag moving speed. Although the method takes into account the movement characteristics of the tag, its design is based on an ideal communication channel. In actual RFID application scenarios, there are path loss, signal attenuation, shadowing effects, etc., resulting in unsatisfactory communication channels between the reader and the tag. For this reason, many researchers [24–26] have studied multitag identification methods suitable for nonideal channels. The binary splitting method is intuitively a random access algorithm. Different from the Aloha-based algorithm, the collided tag set will be separated by a fixed probability of 0.5. In binary splitting methods, the tag starvation will be significantly weakened.

Law first used the query tree [21] algorithm to solve the signal conflict problem in the RFID tag identification process. The tag ID is essentially a sequence of 0/1 binary numbers. The recognition process of the query tree algorithm is similar to a virtual binary tree, the height of the tree is the ID length of the tag, and each branch is marked with the "left 0 right 1" method. The reader first sends a query command with a prefix of 0, and those tags whose ID prefix is 0 will respond to the reader and return their own ID. When a collision occurs, the reader divides the collided tag set into two subsets based on tag ID collision bit. These subsets get smaller and smaller until each subset contains only one tag. This type of algorithm requires a stack in the reader to store the query prefix information. The reader will continuously update the query prefix according to the collision bit and push the query prefix onto the stack. The entire recognition process will not end until the stack is null. The disadvantage of the query tree algorithm is that when the tags are dense, the signal conflicts are obviously intensified, which affects the system's recognition performance. Pan and Wu [27] proposed the STT algorithm, which learns the distribution of tags online based on the previous query results, thereby dynamically selecting the query depth of the query tree. The shortcoming of the STT algorithm is that it assumes that the tag distribution of the subsequent query is consistent with the previously learned tag distribution.

Table 1: Summary of various types of anticollision algorithms.

Types	Representatives	Features	Development trends
Aloha-based	MAP [19]	High estimation accuracy and high complexity	Reduce the complexity
	ECD [10]	Pulse detection	Improve the feasibility
	FEIA [22]	Slot-by-slot adjustment	Enhance initial performance
Tree-based	CT [29]	Remove the empty queries	Further improve the efficiency
	DPPS [30]	Identify multiple tags in the same time slot	Optimize query mechanism
	CwT [12]	Window mechanism to reduce the transmitted bits	Reduce the slot number
Hybrid-based	BSTSA [17]	First proposed hybrid architecture	Performance improvement
	GBSA [18]	Performance limit of UHF RFID	Optimize the architecture

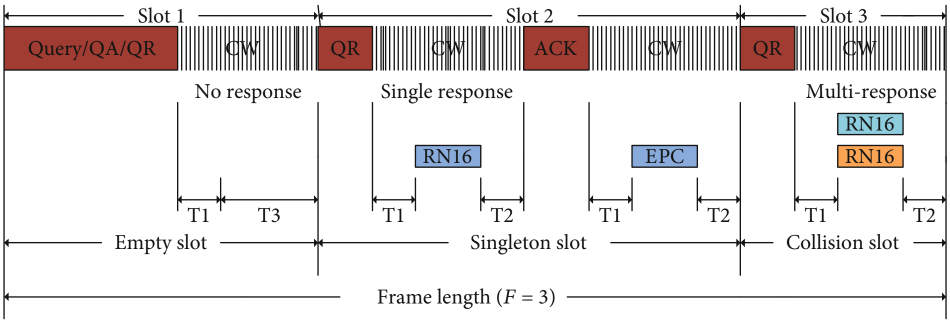

Figure 1: The timing link of EPC C1 Gen2 UHF RFID standard.

Shahzad and Liu [28] questioned this hypothesis and proposed the TH algorithm, which first uses the tag number estimation method to estimate the number of remaining tags and then estimates the ID distribution of unidentified tags and then jumps directly to the optimal layer for subsequent queries. In order to optimize the query prefix and improve the recognition efficiency of the query tree algorithm, subsequent researchers have presented a series of work on this basis, including the CT [29], CCMA [20], CwT [12], DPPS [30], and BQMT [13] algorithms. In Table 1, we summarize the features and development trends for existing representative anticollision algorithms.

3. Performance Analysis of Anticollision Algorithms

3.1. Analysis of DFSA Algorithms. The DFSA algorithm is a classical representative of Aloha-based algorithm, which strictly follows the timing link of EPC C1 Gen2 standard as illustrated in Figure 1. Based on the frame time slot algorithm, the DFSA algorithm estimates the number of tags in the current recognition stage through the tag estimation algorithm and adjusts the frame length at the beginning of each frame recognition stage according to the estimated number of tags to be recognized. Therefore, the key to the improvement of DFSA algorithm is (1) improve the accuracy of the estimation of the number of tags and adjust the frame length reasonably and (2) reduce the number of idle time slots and collision time slots. The workflow of the DFSA algorithm is illustrated in Figure 2.

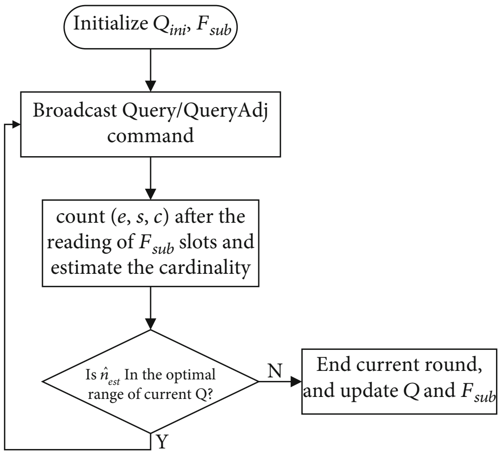
Initialize Q_{ini}, F_{sub}
Broadcast Query/QueryAdj command
count (e, s, c) after the reading of F_{sub} slots and estimate the cardinality
Is $\hat{n}_{est}$ In the optimal range of current Q?
N
End current round, and update Q and F_{sub}
Y

Figure 2: The flowchart of the DFSA algorithm.

We suppose the number of effective tags is$N$, and the length of frame size is$F$, since the probability of choosing the same time slot for any$r$tags can be expressed as

$$P_r = C_N^r \left(\frac{1}{F}\right)^r \left(1 - \frac{1}{F}\right)^{N-r}. \quad (1)$$

For a successful slot, there is only one tag that sends its IDs to the reader. So, the probability of successful slot is

$$P_s = C_N^1 \left(\frac{1}{F}\right)^1 \left(1 - \frac{1}{F}\right)^{N-1}. \quad (2)$$

When the current time slot has no tag send ID, it is defined as an idle time slot, and its probability can be expressed as

$$P_e = \left(1 - \frac{1}{F}\right)^N. \tag{3}$$

There are only three slot statuses during the recognition process: successful slot, idle slot, and collision slot. Thus, the probability of collision slot is

$$P_c = 1 - P_s - P_e = 1 - \frac{N + F - 1}{F}\left(1 - \frac{1}{F}\right)^{N-1}. \tag{4}$$

According to Equations (1), (2), and (3) above, we can get the expected values of successful slot, idle slot, and collision slot as follows:

$$\begin{cases} E_s = N(1 - 1/F)^{N-1}, \\ E_e = F(1 - 1/F)^N, \\ E_c = F - (N + F - 1)(1 - 1/F)^{N-1}. \end{cases} \tag{5}$$

We denote the system throughput as the ratio of successful slots to total slots F, denoted as U. Thus, U is expressed as

$$U = \frac{N}{F}\left(1 - \frac{1}{F}\right)^{N-1}. \tag{6}$$

In order to satisfy U maximum, we take the derivative of F on both sides of Equation (6) and make it equal to 0; we then have

$$\frac{dU}{dF} = \frac{N(N - F)(F - 1)^{N-2}}{F^{N+1}} = 0. \tag{7}$$

From Equation (7), we can know that in order to maximize U, there are two possibilities for the value of F, either $F = 1$ or $F = N$. Considering the actual situation, the number of tags in the coverage of the reader is usually much greater than 1. Therefore, we find that the F value that satisfies maximal U is N. That is, when the given frame length is equal to the number of tags to be identified, the anticollision algorithm can achieve the maximum system throughput.

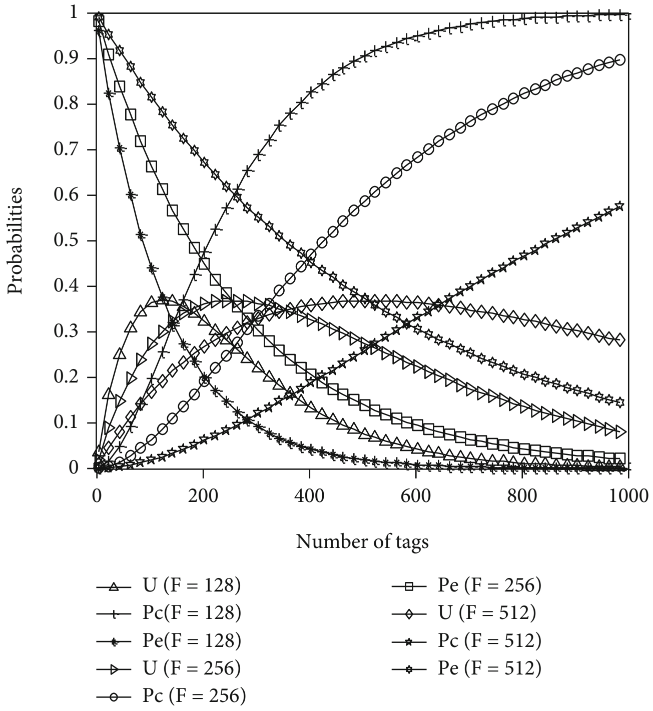

Figure 3: U, P_e, and P_c for different N and F values.

Figure 3 reveals the relationship between U, P_e, and P_c under different tag numbers and different frame lengths. We can see from Figure 2 that each curve about U corresponds to a peak point, which is approximately equal to 0.368, which is obtained when $F = N$. Since the reader does not know the specific number of tags before identification, the purpose of the DFSA algorithm is to estimate the number of remaining tags so as to approach the performance to 0.368. Therefore, 0.368 has also become the performance bottleneck of the traditional DFSA algorithm. My research found that the reason for this bottleneck is that the current DFSA algorithm framework is aimed at all unread tags, and the reader sets a relatively large frame length to identify them. From Equation (6), we know that when F approaches infinity, the value of U approaches 0.368. When F is small, the value of U can be greater than 0.368. For example, when $F = N = 2$, the value of U is 0.5. This phenomenon stimulates the design of an independent frame recognition scheme, so that system throughput of the anticollision algorithm exceeds 0.368. Let us do a simple theoretical derivation to verify this point.

According to the definition of system throughput, we know that it can be redefined as N divided by the total number of time slots to identify N tags. Accordingly, Equation (6) can be rewritten as the following formula:

$$U = \frac{N}{F + A}, \tag{8}$$

where A represents the expected number of remaining time slots required for the reader to recognize N tags except for the initial frame F. Similarly, according to the calculation formula of system throughput, we can get

$$A = \frac{N_{\text{rest}}}{(1 - (1/N_{\text{rest}}))^{N_{\text{rest}}-1}}, \tag{9}$$

in which N_{rest} means the number of remaining tags. We know that before the end of the entire tag identification process, there are always collision slots, and the sum of the tags involved in all collision slots is the remaining tags. Thus, we have

$$N_{\text{rest}} = \sum_{i=1}^{m} r_i. \tag{10}$$

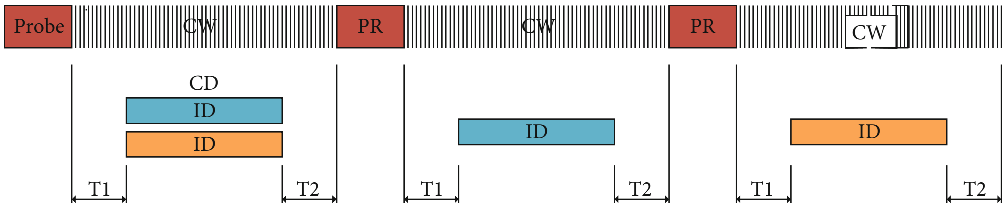

Figure 4: The timing link of ISO 18000-6B RFID standard.

Herein, m is the number of collision slots counted in the initial frame, and r_i represents the number of tags involved in i-th collision slot. We assume that the idea of the new anticollision algorithm (designed in this paper) is to independently allocate a small frame for each collision slot to identify it; then, the system throughput of the algorithm can be derived as follows:

$$U_{\text{new}} = \frac{N}{F+B}, \quad B = \sum_{i=1}^{m} \frac{k_i}{\left(1-(1/k_i)\right)^{k_i-1}}. \tag{11}$$

Assume that among all collision slots, the lth time slot involves the largest number of tags, and the number of tags included is k_l. So, we have

$$B < B^* = \frac{\sum_{i=1}^{m} k_i}{\left(1-(1/k_l)\right)^{k_l-1}}. \tag{12}$$

Obviously, the function of $(1-(1/x))^{x-1}$ is a monotonically decreasing function. From this, we can deduce

$$U_{\text{new}}^* = \frac{N}{F+B^*} > U_{\text{DFSA}} = \frac{N}{F+A}. \tag{13}$$

The reason why the above formula holds is that $N_{\text{rest}} > k_l$. Therefore, we have

$$U_{\text{new}} > U_{\text{new}}^* > U_{\text{DFSA}}. \tag{14}$$

We prove through the above derivation that in the same multitag recognition scenario, the system throughput of the new algorithm can break through the performance bottleneck of the traditional DFSA algorithm. It is worth noting that the new algorithm here only refers to the concept and idea we put forward, and the detailed algorithm design is not the focus of this paper.

3.2. Analysis of Query Tree Algorithms. Unlike the DFSA algorithms, the query tree- (QT-) based algorithms do not have a common performance upper bound. Therefore, different optimization methods for query prefixes may result in large differences in final performance. The timing of QT-based algorithms usually follows the ISO-18000-6B standard, as shown in Figure 4. The workflow of QT-based algorithm is illustrated in Figure 5. Since there will be differences in the performance of each QT algorithm, we choose several representative QT algorithms for performance analysis here.

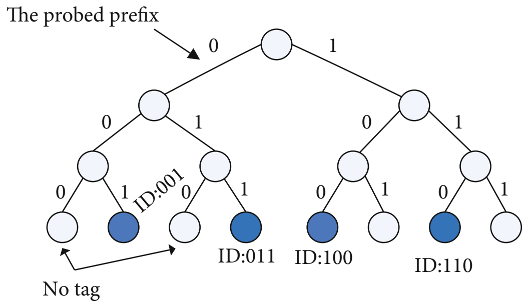

Figure 5: The workflow of the QT-based algorithm.

(1) Collision tree (CT) algorithm

The basic principle of the CT algorithm is that the reader sets the highest collision bit of the received string to 0 and 1, respectively; generates a new query prefix; and sends a new query command in the next time slot. The tag that receives the query command only needs to return the remaining ID information after matching the prefix. Compared with the traditional QT algorithm, the advantage of the CT algorithm is that it eliminates idle time slots in the query process and reduces the amount of data transmission in the time slot.

Assuming that there are 5 tags (A, B, C, D, and E) in the current reader's working domain, their IDs are "0010," "0101," "1101," and "1110." The reader uses the QT algorithm to recognize them, and the detailed recognition process is shown in Table 2.

We can observe in Table 2 that it takes 9 time slots for the reader to recognize these 5 tags using the QT algorithm. The recognition process of the QT anticollision algorithm is similar to traversing a binary tree, and each node on the tree will be detected. We can see that in the QT algorithm, each tag replies with a complete ID, which undoubtedly increases the communication complexity in the recognition process. In addition, when the number of tags to be identified is large, the QT algorithm will generate a large number of idle time slots, thereby further reducing performance. Since the CT algorithm eliminates idle nodes on the basis of the QT algorithm, the binary traversal tree corresponding to the CT algorithm only contains collision nodes (intermediate nodes) and leaf nodes, thereby improving the recognition efficiency.

Table 2: An recognition example by using QT algorithm.

Slot	Query prefix	Responding data from tags	Slot status
1	Empty string ε	xxxx	Collided
2	0	0xxx	Collided
3	00	0010	Successfully identify A
4	01	0101	Successfully identify B
5	1	11xx	Collided
6	10		Idle
7	11	11xx	Collided
8	110	1101	Successfully identify C
9	111	1110	Successfully identify D

Assuming that there are n tags to be identified in the RFID system, the reader uses CT algorithm to identify them; the total number of time slots required can be expressed as

$$N_{\mathrm{t}} = N_{\mathrm{c}} + n, \tag{15}$$

where N_{c} represents the number of collision nodes in the traversal tree. We know that a collision node will produce 2 child nodes, so N_t can be further rewritten as

$$N_{\mathrm{t}} = 2 \times N_{\mathrm{c}} + 1. \tag{16}$$

Comparing Equations (15) and (16), we can have

$$N_{\mathrm{c}} = n - 1. \tag{17}$$

Therefore, the total number of slots required by the CT algorithm to identify n tags can be expressed as

$$N_{\mathrm{t}} = 2 \times n - 1. \tag{18}$$

According to the definition of system throughput, we can know that the maximum system throughput of CT algorithm is

$$U_{\mathrm{CT}} = \lim_{n \longrightarrow \infty} \frac{n}{2n-1} = \lim_{n \longrightarrow \infty} \frac{1}{2-(1/n)} = \frac{1}{2}. \tag{19}$$

Through formula (19), we know that the performance of the CT algorithm is relatively stable, and its system throughput is maintained at 50%. However, it is essentially a binary tree search. When the collision is obvious, the search process cannot be accelerated.

(2) CCMA algorithm

The CCMA algorithm is a multiary search algorithm, which introduces a custom query command and a collision string mapping mechanism. The mapping relationship of collision data is shown in Table 3. The main idea of the CCMA algorithm is that if the reader detects that the first and second collision bits are continuous, the reader will send a custom query command, namely, QueryP in the next time slot to make the involved tag return a 4-bit mapping string to replace the original ID prefix information. This mapping string can accurately reflect the collision information of the tag. The reader can accurately identify the first 2 bits of collision information of the tag through the received mapping string to determine the next query command. If the first and second collision bits are not consecutive, the reader will use the CT algorithm to identify the tag.

Table 3: The mapping table used in CCMA algorithm.

Collision information	Mapped string
00	0001
01	0010
10	0100
11	1000

The authors in [20] only give the simulation results of the CCMA algorithm. In order to better evaluate the performance of the QT algorithm, we do the following analysis of the CCMA algorithm. Similarly, we assume that there are n tags in the RFID system, and the reader uses the CCMA algorithm to recognize them. The total number of time slots required is

$$N_{\mathrm{t}} = P_{\mathrm{cc}} \times N_{\mathrm{cc}} + P_{\mathrm{sc}} \times N_{\mathrm{sc}}, \tag{20}$$

where N_{cc} is the number of time slots required to identify n tags when all collisions are continuous collisions and N_{sc} is the number of time slots required to identify n tags when all collisions are noncontinuous. P_{cc} and P_{sc} are the probability of continuous collision and discontinuous collision, respectively.

Lemma 1. *Assuming that all collisions in the multitag recognition process are noncontinuous collisions, the number of time slots required by the CCMA algorithm to identify n tags is* $N_{ccma} = 2 \times n - 1$.

Proof. In the entire tag recognition process, when all the detected collisions are noncontinuous, the recognition process of the CCMA algorithm is similar to the CT algorithm, at this time, $N_{\mathrm{sc}} = 2 \times N_{\mathrm{c}} + 1 = 2 \times \mathrm{n} - 1$. Thus, Lemma 1 is proved. □

Lemma 2. *In the entire tag recognition process, when all the detected collisions are continuous, the number of time slots used by the CCMA algorithm to recognize n tags is* $N_{cc} = (20 \times n - 11)/9$.

Proof. When all detected collisions are continuous, the recognition process of the CCMA algorithm is similar to a complete quaternary traversal tree. The collision node in the tree will produce 4 child nodes, so $N_4 = N_l + N_c = 4 \times N_c + 1$, where N_l and N_{c} denote the number of leaf nodes and collision nodes in the tree; we have

$$N_l = 3 \times N_{\mathrm{c}} + 1. \tag{21}$$

In the quaternary traversal tree, the leaf nodes may be successful nodes or idle nodes, so the above formula can be rewritten as

$$n + N_{\mathrm{i}} = 3 \times N_{\mathrm{c}} + 1. \tag{22}$$

In the CCMA algorithm, the number of idle nodes that a continuous collision node may produce is between 0 and 2. The CCMA algorithm can eliminate idle time slots through the QueryP command. Next, we consider the following three scenarios, respectively.

Scenario A: when the number of idle nodes generated by a continuous collision node is 0, $N_i = 0$, and $N_{\mathrm{c}} = (n-1)/3$ can be obtained from Equation (22). We substitute it into $N_4 = 4 \times N_{\mathrm{c}} + 1$ to get $N_4 = (4 \times n - 1)/3$. According to the principle of the CCMA algorithm, the number of sending the custom command QueryP in the CCMA algorithm is the number of consecutive collisions (i.e., N_{c}), so $N_{\mathrm{cc}} = N_4 - N_{\mathrm{i}} + N_{\mathrm{c}} = (5 \times n - 2)/3$.

Scenario B: when the number of idle nodes generated by a continuous collision node is 1, $N_{\mathrm{i}} = N_{\mathrm{c}}$, $N_{\mathrm{c}} = (n-1)/2$ can be obtained from Equation (22), and substituting it into $N_4 = 4 \times N_{\mathrm{c}} + 1$, we can find $N_4 = 2 \times n - 1$. Further, we can find $N_{\mathrm{cc}} = 2 \times n - 1$.

Scenario C: when the number of idle nodes generated by a continuous collision node is 2, similarly, $N_{\mathrm{cc}} = 3 \times n - 2$ can be obtained.

Since the above three scenarios appear with equal probability, we can get $N_{\mathrm{cc}} = (20 \times n - 11)/9$; therefore, Lemma 2 is proved. □

Considering that in the process of tag recognition, the probability of continuous collision and discontinuous collision is 0.5, combining Lemmas 1 and 2, formula (20) can be rewritten as

$$N_{\mathrm{t}} = \frac{1}{2} N_{\mathrm{sc}} + \frac{1}{2} N_{\mathrm{cc}} = \frac{19 \times n - 10}{9}. \tag{23}$$

Therefore, the maximum system throughput that the CCMA algorithm can achieve can be expressed as

$$U_{\mathrm{CCMA}} = \lim_{n \longrightarrow \infty} \frac{9n}{19n - 10} = \frac{9}{19 - (10/n)} \approx 0.4736. \tag{24}$$

From Equation (24), we can know that the system throughput that the CCMA algorithm can maintain is slightly lower than that of the CT algorithm, but the amount of data transmission required is lower than that of the CT algorithm.

(3) DPPS algorithm

The performance analysis of the current mainstream QT anticollision algorithms shows that the recognition performance of the QT algorithm has a higher room for improvement. A typical idea is to optimize the response mechanism of the tag, and its representative is the DPPS algorithm. The basic principle of the DPPS algorithm is that the reader sends a query command to all tags within its coverage area, and when the tag receives this command, it will return its own complete ID. Once a collision is detected, the reader will update the query prefix based on the parsed data and send the PROBE_EQ command to probe the tag in the next time slot. There are 3 key parameters in the PROBE_EQ command. First is COM_Str, which represents the data part of the tag ID before the collision bit; second is Pre1, which represents prefix 1; and third is Pre2, which represents prefix 2. The value of Pre2 is equal to the value of Pre1 minus one. We can also make the following derivation to analyze the performance of the DPPS algorithm. In the DPPS algorithm, we use N_{slots} to represent the sum of the number of successful slots, complete collision slots, and identifiable collision slots. Let $E(N_{\mathrm{slots}})$ be defined as the expected value of N_{slots}. According to the principle of the DPPS algorithm, $E(N_{\mathrm{slots}})$ can be expressed as

$$E(N_{\mathrm{slots}}) = 1 + N_{\mathrm{c}}, \tag{25}$$

where N_{c} is the number of ID collisions during the recognition process.

Table 4: The experimental parameters used in this paper.

Parameter	Value
The number of tags	(100, 1000)
The length of IDs	96
Communication rate	40 kbps
T1	25 μs
T2	25 μs
T3	12.5 μs

Lemma 3. *Similarly, we assume that there are n tags to be identified in the system, and the reader uses the DPPS algorithm to identify them; then, the expected total number of time slots $E(N_{slots})$ is equal to n.*

Proof. The recognition process of the DPPS algorithm is similar to a variant of a binary traversal tree, and each node on the tree corresponds to a time slot. For the convenience of analysis, we treat both identifiable collisions and complete collisions as collision nodes. Obviously, a collision node will produce 2 child nodes, so the total number of nodes on the traversal tree is $2N_{\mathrm{c}} + 1 = N_{\mathrm{c}} + n$. It can be seen that $N_{\mathrm{c}} = n - 1$. According to formula (25), we know that $E(N_{\mathrm{slots}}) = 1 + n - 1 = n$. Therefore, Lemma 3 is proved. □

Above, we analyzed the performance of three typical QT algorithms. Through analysis, we know that the performance improvement of QT algorithms mainly depends on the query mechanism of the reader and the response mechanism of the tag. By optimizing these two mechanisms, the performance of tag recognition can be continuously improved. Through Lemma 3, we know that the DPPS algorithm only needs n time slots to identify n tags, and its system throughput reaches 100%. However, this does not mean that the performance of QT anticollision algorithms has reached the limit.

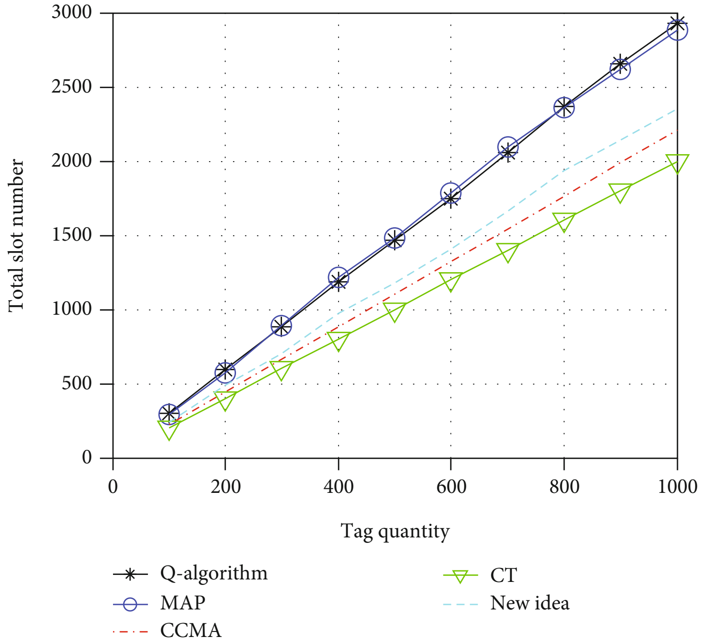

Figure 6: Experimental results: the total slot numbers by various algorithms.

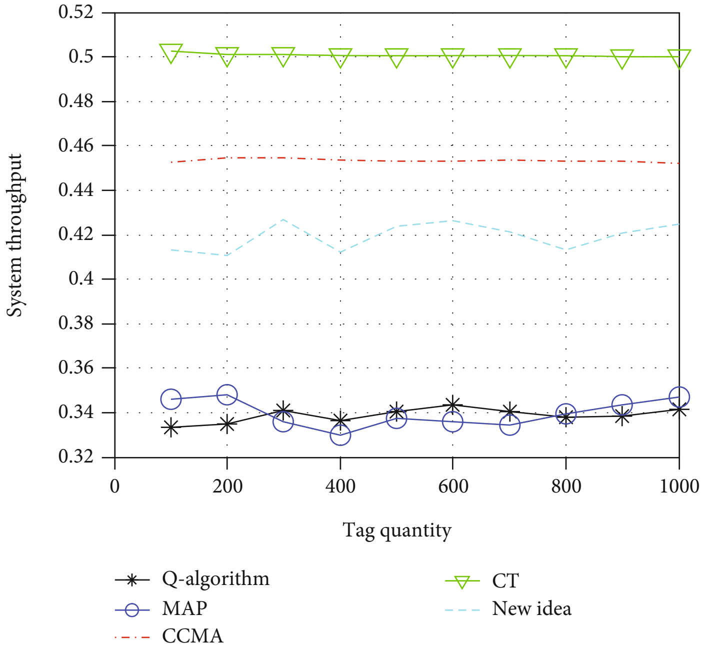

Figure 7: Experimental results: system throughput of various algorithms.

The reason is that in the DPPS algorithm, the length of a single time slot is longer than the length of the time slot in the traditional algorithm, so the actual improvement effect is not so high. Therefore, there is still much room for improvement in the performance of QT algorithms.

4. Experimental Results and Analysis

4.1. Experiment Setup. In the experiment section, we measure the performance of the RFID anticollision through some common metrics such as system throughput, total number of time slots, and recognition efficiency and compared with our analysis results to verify the effectiveness of our theoretical analysis. The simulations are performed on the desktop computer with Intel Core i5-4590 CPU and 8 GB RAM. All experiments are realized through MATLAB program. Since the performance analysis results of most DFSA algorithms are similar and close (below 0.368), we choose the more classic Q-algorithm and MAP algorithm as reference. In the QT-based algorithms, we choose the

CCMA and CT algorithm as the reference algorithms. The experimental parameters are listed in Table 4 [31–33]. In order to ensure the convergence of the results, we independently repeat the experiments 1000 times and then take the average value as the final result [34].

4.2. Result Analysis. Figure 6 compares the total number of time slots required by different algorithms to recognize the same quantity of tags. The comparative algorithms include Q-algorithm [9], MAP [19], CCMA [20], CT [29], and the new design concept proposed in this paper (named new idea in the simulations). It can be observed from the figure that when the QT algorithm recognizes the same number of tags, the number of time slots spent is significantly lower than that of the DFSA algorithm. Among the two QT algorithms, the CT algorithm spends less time slots than the CCMA algorithm, which is also consistent with the theoretical analysis results. Among three DFSA algorithms, the number of time slots consumed by the Q-algorithm and the MAP algorithm is very close, while the number of time slots consumed by the new idea is significantly lower than that by the other two DFSA algorithms.

Figure 7 depicts the system throughput that can be achieved under different algorithms. From the simulation results, we can further observe that the system throughput of QT algorithm is generally higher than that of DFSA algorithms. Through our previous theoretical analysis, we can see that the system throughput of most DFSA algorithms is lower than 0.368. This experimental result also further verifies our theoretical analysis. The throughput of the Q-algorithm and MAP algorithm is very close. Their average throughput is 0.3388 and 0.3398, respectively, which is 8.6% and 8.3% away from the theoretical maximum value. We can further observe that the average throughput that new idea algorithm can achieve is 0.4192 and 13.9% higher than the maximum throughput of the existing DFSA algorithms. The average throughput of the CT and CCMA algorithm is 0.5 and 0.453, respectively. The gap between the average throughput of the CCMA algorithm and the theoretical analysis is 4%. The reason is that the number of simulations is not set large enough.

4.3. Challenges and Future Trends. We have verified through theoretical analysis and experiments that many of the current anticollision algorithms can alleviate the problem of multitag recognition to a certain extent, but there are still some problems in the recognition efficiency, complexity, stability, etc., which cannot meet the large-scale RFID systems. In summary, there are following problems:

(1) The existing tag quantity estimation methods cannot be applied to low-cost RFID systems. Existing research on anticollision algorithms is excessively pursuing accuracy in the estimation of the tag quantity, which makes the complexity of the algorithm increase sharply. Taking into account the limitation of the computing power of the RFID platform, the high-complexity algorithm cannot be applied to the low-cost RFID systems

(2) Existing anticollision technologies are facing performance bottlenecks. Most anticollision algorithms are based on a channel with only one tag reply, so that the reader can decode it correctly. The collision information is discarded, and the tag needs to be retransmitted. This undoubtedly wastes a lot of useful information and restricts the further improvement of the recognition efficiency of the anticollision algorithm

5. Conclusions

This paper analyzes the current mainstream anticollision algorithms and conducts theoretical analysis and simulation on their performance. Based on the analysis of the reasons for the limited performance of the existing DFSA algorithm, a new DFSA algorithm architecture is proposed, and theoretical analysis and simulation results prove that it can break through the performance bottleneck of the current DFSA algorithm. Finally, we summarize the existing research on anticollision algorithms and present the current challenges and future research trends.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest to report regarding the present study.

References

[1] R. Jayadi, Y. C. Lai, and C. C. Lin, “Efficient time-oriented anti-collision protocol for RFID tag identification,” *Computer Communications*, vol. 112, pp. 141–153, 2017.

[2] J. Su, Z. Sheng, A. Liu, Z. Fu, and Y. Chen, “A time and energy saving-based frame adjustment strategy (TES-FAS) tag identification algorithm for UHF RFID systems,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 2974–2986, 2020.

[3] X. Liu, J. Zhang, S. Jiang et al., “Accurate localization of tagged objects using mobile RFID-augmented robots,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1273–1284, 2021.

[4] Y. Ye, L. Shi, R. Qingyang Hu, and G. Lu, “Energy-efficient resource allocation for wirelessly powered backscatter communications,” *IEEE Communications Letters*, vol. 23, no. 8, pp. 1418–1422, 2019.

[5] L. Zhu and T. S. P. Yum, “A critical survey and analysis of RFID anti-collision mechanisms,” *IEEE Communications Magazine*, vol. 49, no. 5, pp. 214–221, 2011.

[6] D. Klair, Kwan-Wu Chin, and R. Raad, “A survey and tutorial of RFID anti-collision protocols,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, pp. 400–421, 2010.

[7] M. Benssalah, M. Djeddou, B. Dahou, K. Drouiche, and A. Maali, “A cooperative Bayesian and lower bound estimation in dynamic framed slotted ALOHA algorithm for RFID

systems," *International Journal of Communication Systems*, vol. 31, no. 13, pp. 1–13, 2018.

[8] A. Bekkali, S. Zou, A. Kadri, M. Crisp, and R. V. Penty, "Performance analysis of passive UHF RFID systems under cascaded fading channels and interference effects," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1421–1433, 2015.

[9] Z. He and H. Luo, "An efficient early frame breaking strategy for RFID tag identification in large-scale industrial Internet of Things," *Scientific Programming*, vol. 2021, 6 pages, 2021.

[10] J. Su, R. Xu, S. Yu, B. Wang, and J. Wang, "Idle slots skipped mechanism based tag identification algorithm with enhanced collision detection," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 5, pp. 2294–2309, 2020.

[11] C. N. Yang, L. J. Hu, and J. B. Lai, "Query tree algorithm for RFID tag with binary-coded decimal EPC," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1616–1619, 2012.

[12] H. Landaluce, A. Perallos, E. Onieva, L. Arjona, and L. Bengtsson, "An energy and identification time decreasing procedure for memoryless RFID tag anticollision protocols," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 4234–4247, 2016.

[13] J. Su, Y. Chen, Z. Sheng, Z. Huang, and A. X. Liu, "From M-ary query to bit query: a new strategy for efficient large-scale RFID identification," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2381–2393, 2020.

[14] Y. C. Lai, L. Y. Hsiao, H. J. Chen, C. N. Lai, and J. W. Lin, "A novel query tree algorithm with bit tracking in RFID tag identification," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 2063–2075, 2012.

[15] J. Shin, B. Jeon, and D. Yang, "Multiple RFID tags identification with M-ary query tree scheme," *IEEE Communications Letters*, vol. 17, no. 3, pp. 604–607, 2013.

[16] T. F. la Porta, G. Maselli, and C. Petrioli, "Anticollision protocols for single-reader RFID systems: temporal analysis and optimization," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 267–279, 2011.

[17] H. Wu, Y. Zeng, J. Feng, and Y. Gu, "Binary tree slotted ALOHA for passive RFID tag anticollision," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 19–31, 2013.

[18] J. Su, Z. Sheng, A. X. Liu, and Y. Chen, "A partitioning approach to RFID identification," *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2160–2173, 2020.

[19] 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.

[20] J. Su, G. Wen, and J. Han, "An efficient RFID anti-collision algorithm for ISO 18000-6B algorithm," *Acta Electronica Sinica*, vol. 42, no. 12, pp. 2515–2519, 2014.

[21] K. W. Chiang, C. Hua, and T. S. P. Yum, "Prefix-randomized query-tree algorithm for RFID systems," in *Proceedings of IEEE International Conference on Communications*, pp. 1653–1657, Istanbul, Turkey, 2006.

[22] W. T. Chen, "Optimal frame length analysis and an efficient anti-collision algorithm with early adjustment of frame length for RFID systems," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3342–3348, 2016.

[23] W. Zhu, J. Cao, H. C. B. Chen, X. Liu, and V. Raychoudhury, "Mobile RFID with a high identification rate," *IEEE Transactions on Computers*, vol. 63, no. 7, pp. 1778–1792, 2014.

[24] Y. Chen, Q. Feng, X. Jia, and H. Chen, "Modeling and analyzing RFID generation-2 under unreliable channels," *Journal of Network and Computer Applications*, vol. 178, article 102937, 2021.

[25] L. Xie, B. Sheng, C. C. Tan, H. Han, and Q. Li, "Efficient tag identification in mobile RFID systems," in *29th IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–9, San Diego, CA, USA, 2010.

[26] L. Yang, Y. Chen, X. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: real-time tracking of mobile RFID tags to high precision using COTS devices," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, pp. 237–248, Hawaii, US, 2014.

[27] L. Pan and H. Wu, "Smart trend-traversal: a low delay and energy tag arbitration algorithm for large RFID systems," in *28th IEEE Conference on Computer Communications (INFOCOM)*, pp. 2571–2575, Rio De Janeiro, Brazil, 2009.

[28] M. Shahzad and A. X. Liu, "Probabilistic optimal tree hopping for RFID identification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 796–809, 2015.

[29] X. Jia, Q. Feng, and L. Yu, "Stability analysis of an efficient anti-collision protocol for RFID tag identification," *IEEE Transactions on Communications*, vol. 60, no. 8, pp. 2285–2294, 2012.

[30] J. Su, Z. Sheng, G. Wen, and V. C. M. Leung, "A time efficient tag identification algorithm using dual prefix probe scheme (DPPS)," *IEEE Signal Processing Letters*, vol. 23, no. 3, pp. 386–389, 2016.

[31] J. Su, Z. Sheng, A. X. Liu, Y. Han, and Y. Chen, "Capture-aware identification of mobile RFID tags with unreliable channels," *IEEE Transactions on Mobile Computing*, pp. 1–14, 2020.

[32] 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*, pp. 1–12, 2021.

[33] J. Su, Z. Sheng, L. Xie, G. Li, and A. X. Liu, "Fast splitting-based tag identification algorithm for anti-collision in UHF RFID system," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2527–2538, 2019.

[34] J. SU, X. ZHAO, D. HONG, Z. LUO, and H. CHEN, "Q-value fine-grained adjustment based RFID anti-collision algorithm," *IEICE Transactions on Communications*, vol. E99.B, no. 7, pp. 1593–1598, 2016.