

Received April 8, 2020, accepted April 21, 2020, date of publication April 28, 2020, date of current version May 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991027

TMIA: A Tree-Based Multi-Reader Interactive Anti-Collision Algorithm for RFID Tag Identification

JIANHUA SONG^{1,2,3,4}, SHUAI HE¹, AND HUI YAO¹

¹School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China

²Engineering and Technical Research Center of Hubei Province in Educational Informatization, Wuhan 430062, China

³Engineering and Technical Research Center of Hubei Province in Software Engineering, Wuhan 430062, China

⁴Engineering Research Center of Hubei Province in Intelligent Government Affairs and Application of Artificial Intelligence, Wuhan 430062, China

Corresponding author: Shuai He (optimus_he@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61803149 and Grant 61901163, in part by the Technology Innovation Special Program of Hubei Province under Grant 2018ACA13, and in part by the Hubei Province Project of Key Research Institute of Humanities and Social Sciences at Universities (Research Center of Information Management for Performance Evaluation).

ABSTRACT In the RFID system, the tag collision problem seriously affects the efficiency of tag identification. Among the anti-collision algorithms, tree-based anti-collision algorithms are popular for the reason that they can ensure that the tags can theoretically 100% be identified by readers. As multi-reader scenarios are more and more widely used in complex internet of things environments, there are more collision slots and idle slots in traditional anti-collision algorithms, which affects the efficiency of the algorithm. We propose a tree-based multi-reader interactive anti-collision algorithm (TMIA) for multi-reader tag identification scenarios to solve the problems above. Readers optimize the broadcast prefix sequence by sharing the broadcast results. The proposed algorithm works in two phases: In the first phase, the reader selects a suitable broadcast prefix by the multi-reader inverse probability function (MIPF) to reduce the initial redundant collision in tag identification. In the second phase, the priority of the broadcast prefix is adjusted through the information exchange between readers, the readers thus avoid broadcasting a large number of invalid prefixes. Theoretical analysis and simulation results show that TMIA has a lower total number of slots and better system efficiency than existing tree-based algorithms. TMIA also greatly reduce the number of collision slots.

INDEX TERMS RFID, tag recognition, multi-reader, tree-based algorithm, tag anti-collision.

I. INTRODUCTION

Radio frequency identification (RFID) is a non-contact automatic identification technology that uses radio waves as a transmission medium to accomplish bidirectional data communication [1], [2]. A typical RFID system consists of readers, tags and the server. Readers can read the contents of tags by wireless transmission. In the process of tags identification, the tag collision occurs when multiple tags send signals to the reader at the same time. Tag anti-collision algorithms can be separated into Aloha-based [3]–[5] and tree-based anti-collision algorithms [6]–[11]. Aloha algorithm is based on probability. The basic principle is that tags send data to the

reader, when a collision occurs, the reader sends a terminate transmission command. Each tag randomly delays for some time and then sent data to the reader. this mechanism is called a random concession mechanism [12], [13]. This method works effectively and performs well when the number of tags is small. However, since the slots are randomly generated, there may be some tags that have not been identified for a long time, which is the tag starvation problem [14]. To solve the problem of tag starvation, tree-based anti-collision algorithms were proposed. The tree-based algorithms have gradually become the mainstream of anti-collision algorithms due to the reason that they can theoretically guarantee that tags can be 100% identified. Tree-based tag anti-collision algorithms can be classified as query tree based algorithms [15]–[22], and binary search tree-based (BST)

The associate editor coordinating the review of this manuscript and approving it for publication was Vyasa Sai.

algorithms [23]–[25]. Tree-based algorithms do not suffer from tag starvation, but usually require a large number of slots during the identification process, and there are also a large number of collision slots.

With more and more multi-reader RFID systems applying in the complex industrial IoT environment, such as inventory check in a large warehouse, transportation, production assembly line and so on, the efficiency of traditional tree-based tag anti-collision algorithms in multi-reader scenarios is not as good as in single-reader scenarios. There are two main challenges:

(1) A longer length of tag ID is needed in multi-reader scenarios. Under the same quantity of tags being searched by each reader, a multi-reader scenario will force each reader to use a deeper tree structure to store every bit of tag ID, which will lead to more query commands as well as more collision bits during identifications. (2) There is no cooperation between readers in traditional tree-based algorithms. Readers can only acquire next broadcast prefixes through their own acknowledges, which decreases the efficiency of the algorithm. Therefore, an efficient anti-collision algorithm in the multi-reader scenario is essential for the development of RFID technology.

To solve the problems above, we propose a tree-based multi-reader interactive anti-collision algorithm(TMIA). TMIA focuses on the benefits of different broadcast results to tag anti-collision, especially the idle slots, which were useless for their own reader but can increase the tag identification efficiency for other readers. By sharing the broadcast results between readers, TMIA greatly reduces collision slots and invalid nodes, thereby improved the efficiency of the protocol.

There are three main contributions of this paper: (1)In multi-reader scenario, all broadcast prefixes can be presented by their IDs into a full binary tree. We proposed a multi-reader inverse probability function (MIPF) to make all the possibility that a node contains tags at each layer measurable. (2)In the tag anti-collision research area, the perspective of former algorithms to improve system efficiency is usually to improve the communication between one reader and a lot of tags. This research method has a limited effect in reducing the number of collision slots. We found that the broadcast result of a single reader has benefit to other readers, we use this benefit to improve the efficiency of the tree-based algorithm, which avoids a large number of potential collision slots. (3)We improved the initial layer selection strategy of broadcasting. In PQT [23], a mapping table method is used to derive the optimal initial layer, but it did not consider the effect of tags distribution in the actual broadcasting to the initial layer. we used the effect above and bit tracking technology to improve the selection strategy of the initial layer.

The other parts of this paper are as follows: Section II describes tree-based tag anti-collision algorithms research background and related works; Section III describes TMIA core idea of the algorithm, ideal environment and algorithm

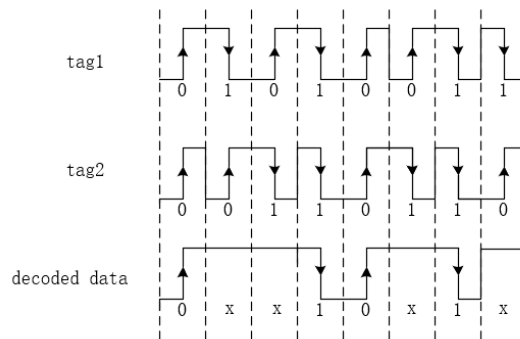


FIGURE 1. Tag ID identify using bit-tracking technology in a RFID system.

pseudo-code; Section IV analyzes the performance of TMIA algorithm; In Section V, the simulation experimental results for the proposed algorithm are shown and compared with other related algorithms; The sixth section summarizes the work of this paper.

II. RELATED WORKS

A. BIT TRACKING TECHNIQUE

Bit-tracking technology is usually based on Manchester coding and is used to track the collision bit in broadcasting. Manchester code is a synchronous clock coding technology, which is introduced in detail in [26], [27]. Its core idea is to express the jump from low to a high level by logic code ‘0’ and to use logic code ‘1’ to express the jump from high to a low level. In the RFID system, all tags signal transmission is based on the Manchester method. Many scholars improved the tree anti-collision algorithm from the perspective of Manchester coding [28], [29], this improved method is called “bit tracking” technology. The proposed algorithm will also adopt this strategy. In the process of tag response, the reader can receive the same electric signal of multiple tags, the same signal of tags and corresponding logic code “1” or “0” at the same time. However, if a bit code superimposes “0” and “1” signals at the same time, it will interfere with each other to produce a collision bit, which leads to an invalid symbol as shown in Figure 1. It can be seen that the collision bits are the second, third, sixth and eighth bit. This information helps the reader further split into subsets so tags identification will be faster when collided.

B. QUERY TREE ALGORITHM

The query tree (QT) algorithm [30] is a very classic tree-based anti-collision algorithm. It uses the bit tracking technology to identify the collision position of tags ID number. The reader queries by sending the tag ID prefix, and tags matching the ID prefix respond. The reader contains a stack for storing the query prefix. Based on the previous query and different query results, ‘0’ or ‘1’ is added to the next query prefix. QT lays the basic structure of the tree-based tag anti-collision algorithm,

but its performance is not outstanding, and there are still many places that can be improved.

C. ENHANCED ANTI-COLLISION ALGORITHMS BASED ON QT ALGORITHM

Since the QT algorithm relies on the information of the collision bit, once the initial collision potential cannot provide effective information, the QT algorithm will fall into the initial redundancy problem. Thus, algorithms such as OQTT [31], I4QT [32] and PQT [23] are proposed to improve QT. The Optimal Query Tracking Tree (OQTT) [31] improves QT from the perspective of separating all tags into smaller sets. OQTT estimates the number of tags based on the position of the tags collision bits in the broadcast process, splits the optimal initial subset according to this number, then divides the optimal initial subset according to the number of tags, and then generates appropriate queries according to the optimal initial subset number by tracking the query tree, so that the number of tags scattered on the initial query approximately obeys uniform distribution. Finally, the query tracking tree further divides the tag into two subsets according to the collision bits. OQTT has lower hardware complexity because the reader does not need to memorize data during the whole recognition process.

Besides the perspective of optimizing the initial subset, the anti-collision algorithms based on the multi-ary query tree have also achieved good results in tags identification. It can be expressed as a B-ary tree ($B \geq 2$), [33] proved that the best performance when the value of B is 3, but for all tags to be recognized in the tree structure, 'B' should be in the form of 2^n ($n \geq 1$), an increase in the value of B results in a decrease in the number of collision slots, while the number of idle slots gradually increases. On this basis, [32] proposed an improved 4-ary query tree algorithm (I4QT). I4QT makes better use of the bit change model (BCM). by using the BCM that only detects the existing tag ID, when a collision slot occurs in 4-ary QT. Therefore, I4QT improves efficiency by eliminating a part of the idle slots as well as a lower communication complexity.

Recently, [23] proposed a probability-based query tree algorithm (PQT). PQT works in two phases: In the first phase, the initial broadcast layer is calculated by the number of tags and the ID number length of the tags, by calculating the inverse probability (the probability of nodes that do not contain any tag) of each layer, PQT proposed an inverse probability function and proved that in the case of uniform distribution, the layer which probability is bigger than 0.2 is the PQT initial layer, In the second phase, it uses the QT as the subsequent broadcast strategy.

III. THE PROPOSED ALGORITHM

A. ASSUMPTIONS AND DEFINITIONS

1) ASSUMPTIONS

This paper follows the same working conditions in [34]: (1)the reader contains a control channel and a data channel.

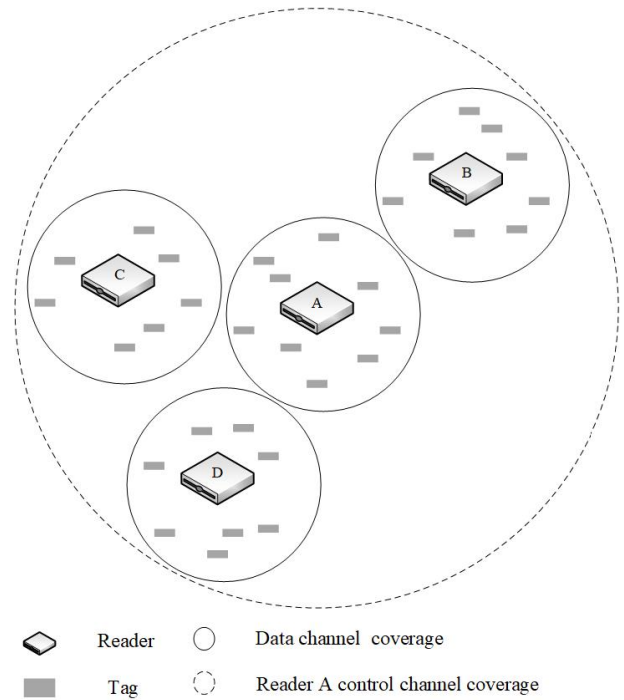


FIGURE 2. Slot definition in multi-reader scenario.

The control channel is used for communication between readers, and the data channel is used for communication between readers and tags. (2) The coverage of the control channel must ensure that all readers can communicate with each other using the control channel through the signals that may interfere with each other in the data channel. By increasing the transmission power of the control channel, the coverage of the control channel can be guaranteed to be larger than that of the data channel, and there is no interference between the data channel and the control channel. (3) Readers are deployed outside the coverage of other readers data channel, since the practical reader anti-collision protocols have almost the same impact on the communications,so the communications between the readers in this paper is considered ideal under the above conditions. Figure 2 is an example of the working environment.

2) DEFINITIONS

The slot is the period during which the reader sends a triggering (or feedback) signal to all tags and the tags respond the signals to the reader. Broadcast slots are usually classified as readable slots, collision slots, and idle slots [35], [36]. Collision slots indicate that two (or more) tags respond during one broadcast. Readable slots indicate that only one tag responds, while an idle slot indicates that no tag response. For the convenience of the information interaction in the multi-reader scenario, based on three classifications above, in this paper, the collision slot is further classified as the readable collision slot and the collision slot. Figure 3 shows the difference between those two slots.

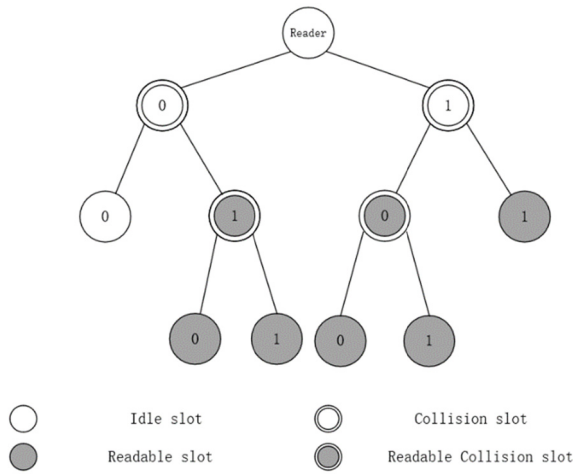


FIGURE 3. Slot definition in multi-reader scenario.

We classified the slot where the collision position at the last bit as a readable collision slot. In a multi-reader scenario, when the collision bit occurs in the last bit, the current prefix contains only two tags and does not contain any tags in other readers. When readable collision slots occur, other readers should avoid broadcasting the prefix. While the collision slot does not contain such characteristics when it appears in other positions, so we further classify the collision slot.

B. BASIC IDEA OF TMIA

The proposed algorithm in this paper can be divided into two phases: In the first phase, the optimal layer of each reader is calculated through MIPF (will be explained in Section III-B-2), and the weights of all nodes are calculated so that each reader can choose a prefix from the optimal layer. In the second phase, the readers exchange broadcast prefixes and broadcast results in each round broadcast and generate the next broadcast prefix by calculating the weight of the prefix and using the bit tracking technology.

1) INITIAL OPTIMAL LAYER SELECTION STRATEGY

The reader broadcasts a INI_Q which is a prefix from the full “1” of length N before identifying starts, to avoid broadcasting the non-collision bits which contain potential invalid nodes, and check whether the initial layer needs to be optimized. When the initial layer returns a collision bit, all layers’ inverse probabilities are calculated through PQT initial layer selection strategy, the reader selects the first layer in which probability is bigger than 0.2 as the initial layer. When the layer is a non-collision bit, we set the initial layer to one layer deeper to avoid repeated broadcasts. Since all nodes in this layer have the same weight at the beginning of broadcasting, the reader randomly selects a node from the layer as the initial broadcast prefix of the reader, and after confirming the initial layer, delete invalid nodes through bit tracking technology, and the priority of the prefix to be broadcast

is continuously adjusted through the subsequent broadcast calculation strategy.

2) THE WEIGHT OF PREFIX CALCULATION STRATEGY

In the process of identification, readers transmit the broadcast result and prefix of each round to all other readers in its coverage through the control channel. Since each ID of the tag is mapped to a node in a binary tree in readers, the reader receives the prefix from other readers. As a result, the weight of the prefix in the queue to be broadcasted is calculated. The weight of the prefix indicates the number of tags it may contain. The larger the weight, the more likely the prefix is to be a valid node. According to the principle that nodes should broadcast if they have tags, otherwise, they should avoid broadcast, this paper analyses the possibility of nodes not containing tags and their changing rules, and then comes to the optimal adjustment strategy. Since the distribution of tags in real situations is random, it is nearly impossible to know the exact distribution of tags, so we use probability to describe it.

If N denotes the length of the tag, R denotes the number of readers, and the number of all tags contained in readers is T . To get the initial weights of all the nodes to be broadcast, the probability of the nodes in the N layer should be calculated first. Assuming that the probability of tag inclusion in node A is P_A and the tag is a uniform distribution since each node contains only two child nodes, the probability of tag inclusion in its child nodes is expressed by P_B and P_C . Obviously, the probability of node A depends on the probability of node B and C , in the case of unknown tag distribution, P_B and P_C are completely equivalent, so $P_B = P_C = \frac{T}{R \cdot 2^N}$. When A node does not contain tags, there must be node B and C do not contain any tag at the same time. Let $P_B = P_C = P$, the expression can be obtained as:

$$P_A = (1 - P)^2 \tag{1}$$

Take $P = \frac{T}{R \cdot 2^N}$ into formula (1), and get:

$$P_A = 1 - (1 - \frac{T}{R \cdot 2^N})^2 \tag{2}$$

Strictly speaking, the existence of ID in B node and ID in C node are not independent of each other, so the probability multiplication rule of independent events is not satisfied. However, in the case of a larger cardinal number, the correlation between the two events is very small, so it can be considered as two events that are independent of each other and satisfy the rule of probability multiplication. The inverse probability function obtained by the probability multiplication rule contains errors, but all errors are within the controllable range. For example, in the first layer, the inverse probability of Layer 0 should be 0 strictly, which means the node must contain ID, and the actual inverse probability is not equal to 0. However, the value of the actual inverse probability is very close to zero, so we can still calculate the probability of each node in this case. The error analysis will be explained in detail in the performance analysis section.

Extend formula (2) to all layers, and get:

$$P_n = 1 - (1 - P)^{2^{N-n}} \quad (3)$$

Bring $P = \frac{T}{R \cdot 2^N}$ into formula (3) and get:

$$P_A = 1 - \left(1 - \frac{T}{R \cdot 2^N}\right)^{2^{N-n}} \quad (4)$$

The function (4) is to calculate the inverse probability of any layer in multi-reader scenario, which is called the multi-reader inverse probability function (MIPF). When a collision slot occurs, other readers reduce the weights of the current node and all its parent nodes to $1 - (1 - \frac{T-2}{R \cdot 2^N})^{2^{N-n}}$ by MIPF, so that the priority of those nodes are lower than other nodes in the layer. When a readable collision slot occurs, other readers remove the node from its binary tree, and recursively check whether its sibling node is also removed. Since all tags containing this node are within the coverage of the current reader, this node is an invalid node for other readers, and should avoid broadcast. When an idle slot occurs, other readers increase the weights of the current node and all parent nodes to $1 - (1 - \frac{1}{(R-1) \cdot 2^N})^{2^{N-n}}$. When a readable slot occurs, for the current reader there is $P_B = 0$, and $P_A = P_B = \frac{T}{R \cdot 2^N}$, the current reader removes the node and quiesces the current tag, other readers reduce the weights of the current node and all parent nodes to $1 - (1 - \frac{1}{R \cdot 2^N})^{2^{N-n}}$ to make its priority higher than the collision node in the same layer and lower than the idle slot node.

Since the length of the prefix is fixed, each prefix will be broadcasted only once, regardless of any broadcast result, this prefix will not be broadcast in the current reader. Therefore, after a single broadcast, and before the start of the next broadcasting, readers remove its current prefix from broadcast queue. The next prefix is the one with the highest weight among the nodes to be selected, if multiple nodes simultaneously at the highest priority, readers will randomly select a prefix as the next broadcast prefix. When a prefix in the initial layer returns an idle slot, the prefix with the highest weight in the initial layer is selected as the next prefix. When there is no prefix to be broadcast in the initial layer, which indicates that the reader has finished searching for the tags, and the reader ends the broadcast.

C. THE PSEUDO-CODE AND FLOWCHART

Figure 4 shows the flowchart of TMIA, in which MIPF is used as a initial layer and next broadcast prefix calculation strategy. 'INI_Q' is a query to allow all tags respond to the reader with their full ID. It is introduced at the part one of Section B.

Figure 5 shows the TMIA pseudo-code, the pseudo-code contains five inputs, where N denotes the tag ID number length, T denotes the number of tags, R denotes the number of readers, 'initial layer' denotes the layer to broadcast, 'prefix queue[]' denotes a multidimensional array which corresponds to the 'prefix_probability' where the prefix weights are stored in. Bit-tracking technology identifies collision bits,

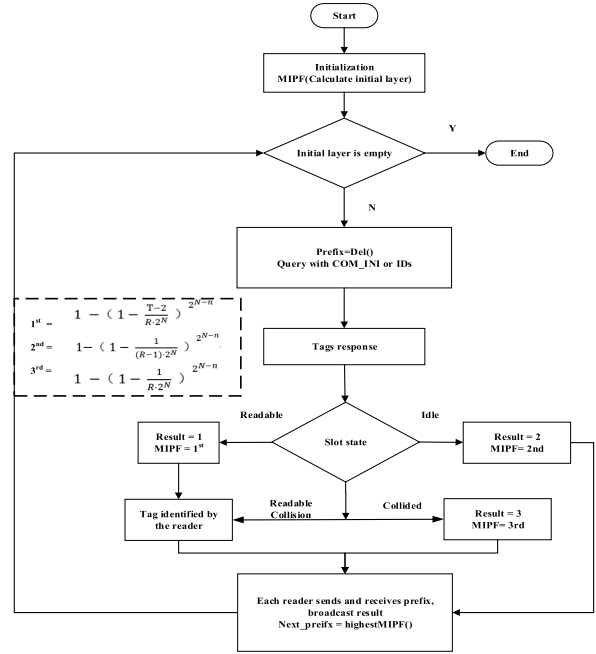


FIGURE 4. The flowchart of TMIA.

build a binary tree with its depth, and the PQT initial layer selection strategy, these three existing methods are omitted in this figure.

TMIA firstly calculates the initial layer through PQT and determines whether there is a collision in this layer through the first round of broadcast and Manchester encoding. If there is no collision, it expands the initial layer downward, and then randomly selects prefix from the layer as the first broadcast prefix, and in subsequent broadcasts, adjust the weight of the prefix to select the next broadcast prefix.

D. A CASE STUDY OF TMIA

For the convenience of understanding, three readers and twelve tags are used. Tags with ID numbers '0110', '0100', '1011', and '1001' are distributed within the coverage of reader A. Tags with ID numbers '1010', '1000', '0111', and '1111' are distributed within the coverage of reader B. Tags with ID numbers '0010', '0000', '0011', '0001' are distributed within the coverage of reader C. According to Manchester encoding, readers send '1111', reader A and B receive 'xxxx' and reader C receives '00xx'. The initial layer can be calculated from the ID number length and the number of tags which is the second layer. Therefore, the first broadcast prefix of reader C is '00'. Since the first two bits of the prefixes of the reader A and B are both collision bit, they randomly select nodes from the second layer as the initial broadcast prefix. Assume Reader A and B initial prefixes are '01', '10'. We select the search process of reader A for a detailed explanation. After completing the search of two tags of '0100' and '0110', in the scenario without other readers' assistance, according to the search strategy of PQT, reader A All '1' prefixes should be broadcast and the next round of

```

MITA $\leftarrow$ 
Input: prefix_queue[], initial_layer, N, T, R $\leftarrow$ 
1. if Manchester(N, T, R) != 'x' $\leftarrow$ 
2.   initial_layer = PQT() $\leftarrow$ 
3. else $\leftarrow$ 
4.   initial_layer = PQT() + 1 $\leftarrow$ 
5. End if $\leftarrow$ 
6. n = N - initial_layer $\leftarrow$ 
7. while n <= N $\leftarrow$ 
8.   prefix_queue[] = bulidTree(n) $\leftarrow$ 
9.   prefix_probability = 1 - [(1 - T / (R * 2^N))]^(2^(N-n))
10.  n += 1 $\leftarrow$ 
11. End while $\leftarrow$ 
12. First_prefix = random select prefix from prefix_queue[]
13. while prefix_queue [initial_layer] $\leftarrow$ 
14.   readers broadcast prefix with data channel $\leftarrow$ 
15.   readers broadcast with control channel $\leftarrow$ 
16.   r = 0 $\leftarrow$ 
17.   while r <= R $\leftarrow$ 
18.     receive prefix, result from other readers $\leftarrow$ 
19.     r += 1  $\leftarrow$ 
20.     if result == 0 // Collision slot $\leftarrow$ 
21.       prefix_probability -= [(1 - T - 2 / (R * 2^N))]^(2^(N-n))
22.     else if result == 1 // Readable or readable collision slot $\leftarrow$ 
23.       prefix_probability = 0 $\leftarrow$ 
24.       remove prefix from childprefix[] $\leftarrow$ 
25.       If all childprefix has been removed $\leftarrow$ 
26.         remove prefix from prefix_queue[] $\leftarrow$ 
27.       else if result == 3 //Idle slot $\leftarrow$ 
28.         prefix_probability -= (1 - T / ((R - 1) * 2^N))^2 $\leftarrow$ 
29.     End if $\leftarrow$ 
30.     next_prefix = the highest probability prefix $\leftarrow$ 
31.   End while $\leftarrow$ 

```

FIGURE 5. The pseudo-code of TMIA.

prefixes should be obtained based on the information of the collision information. In TMIA, reader B marks '10' as a readable collision slot, and reader C marks '00' as collision slots, in reader A, the '00' weight becomes 0.11, reader A delete '11' nodes then '10' is selected as the next prefix since it is the highest weight node, thus TMIA eliminating the round of '1111' broadcasts and avoiding the broadcasts of the two invalid nodes, which are '00' and '11'. The detailed broadcast steps are as follow.

IV. PERFORMANCE ANALYSIS

A. STABILITY ANALYSIS OF TMIA

In formula (2), N is the length of tag ID, n is the length of the ID of any node, P_A is the probability of tag in layer N , P is the probability of tag in layer n .

Theorem 1: TMIA performance is independent of tag ID length N .

Take any two points on the formula (2) function ($n\alpha, \alpha$), ($n\beta, \beta$). If the interval length $n\alpha - n\beta$ is a value independent of N , it can be proved that the original function is independent of N .

Put $n\alpha, n\beta$ into formula (2), the following results are obtained:

$$\alpha = (1 - P_A)^{2^{N-n\alpha}} \tag{5}$$

$$\beta = (1 - P_A)^{2^{N-n\beta}} \tag{6}$$

$n\alpha$ is extracted from formula 5 to obtain:

$$n\alpha = \log_2\left(\frac{2^N \cdot \ln(1 - P_A)}{\ln \alpha}\right) \tag{7}$$

$n\beta$ is obtained by the same method:

$$n\beta = \log_2\left(\frac{2^N \cdot \ln(1 - P_A)}{\ln \beta}\right) \tag{8}$$

By subtracting formula 7 and 8, we get

$$m\beta - n\alpha = \log_2\left(\frac{\ln \alpha}{\ln \beta}\right) \tag{9}$$

Since both α and β are fixed values, formula (9) is a fixed value independent of N .

B. TOLERANCE ANALYSIS OF TMIA

In this paper, we use multiplication to calculate the probability of two non-independent events in 3.2.2, which will bring errors. This section will prove that the errors are within the controllable range and will not affect the efficiency of the algorithm.

The inverse probability function is:

$$1 - P = (1 - P_A)^{2^{N-n}} \tag{10}$$

The first derivative and the second derivative are obtained:

$$(1 - P)' = (1 - P_A)^{2^{N-n}} \cdot 2^{N-n} \cdot \ln(1 - P_A) \cdot \ln \frac{1}{2}$$

$$(1 - P)'' = (1 - P_A)^{2^{N-n}} \cdot 2^{N-n} (2^{N-n} \cdot \ln(1 - P_A) + 1) \cdot \ln(1 - P_A) \cdot \ln\left(\frac{1}{2}\right)^2 \tag{11}$$

Let the second derivative be equal to 0, and we can get:

$$2^n = 2^N \cdot \ln\left(\frac{1}{1 - P_A}\right) \tag{12}$$

Let N approach to infinity:

$$\lim_{N \rightarrow \infty} 2^n = \lim_{N \rightarrow \infty} \frac{\ln(2^N) - \ln(2^N - 1)}{\left(\frac{1}{2}\right)^N} = 1$$

$$\Rightarrow \lim_{N \rightarrow \infty} n = 0 \tag{13}$$

By substituting formula (12) in (10), the maximum error is constant to $\frac{1}{e}$, which is about 0.3678.

We find that this error will decrease rapidly with the increase of the number of tags. When the number of tags in a single reader is more than 5, the error will fall below 0.02. When the number of tags is 1 to 5, the probability error is large, because TMIA has rejected invalid nodes according to bit-tracking, the probability error will not affect the algorithm. Only a few cases will mislead the reader to choose wrong nodes to broadcast, so it can be concluded that the error of probability is always within the controllable range.

TABLE 1. TMIA tag identification process (Reader A).

READER A	ROUND 1	ROUND 2	ROUND 3	ROUND 4
DATA CHANNEL BROADCAST	1111	01	10	11
TAGS RESPOND	XXXX	01X0	10X1	-
TAG IDENTIFIED	-	0100,0110	1001 , 1011	-
CONTROL CHANNEL BROADCAST	-	01X0 , COLLISION SLOT	10X1 , READABLE COLLISION SLOT	11XX, IDLE SLOT
CONTROL CHANNEL RECEIVES	-	10XX, COLLISION SLOT 000X , READABLE COLLISION SLOT	1111 , READABLE SLOT 001X, READABLE COLLISION SLOT	0111 , READABLE SLOT
INITIAL LAYER PREFIXES[WEIGHT]	00[0.16],01[0.16], 10[0.16],11[0.16]	00[0.11] , 10[0.16],11[0.16]	11[0.12]	Null
NEXT PREFIX	01	10	11	-
IS IDENTIFICATION COMPLETED	X	X	X	√

TABLE 2. TMIA tag identification process (Reader B).

READER B	ROUND 1	ROUND 2	ROUND 3	ROUND 4
DATA CHANNEL BROADCAST	1111	10	11	01
TAGS RESPOND	XXXX	10X1	1111	0111
TAG IDENTIFIED	-	1000,1010	1111	0111
CONTROL CHANNEL BROADCAST	-	10X1 , COLLISION SLOT	1111, READABLE SLOT	0111 , READABLE SLOT
CONTROL CHANNEL RECEIVES	-	01X0 , COLLISION SLOT 000X, READABLE COLLISION SLOT	10x1, READABLE COLLISION SLOT 001X, READABLE COLLISION SLOT	11XX, IDLE SLOT 0111, READABLE SLOT
INITIAL LAYER PREFIXES[WEIGHT]	00[0.16],01[0.16], 10[0.16],11[0.16]	00[0.11] , 01[0.11] 11[0.16]	01[0.11]	Null
NEXT PREFIX	10	11	01	Null
IS IDENTIFICATION COMPLETED	X	X	X	√

C. TOTAL NUMBER OF SLOTS

The total number of slots refers to the total number of broadcasts made by a single reader during the process of identification. The total number of slots of TMIA is the initial value of the binary tree which is $2(2^N - 1)$. Known tag ID length is N , the number of readers is R . The number of tags that each reader needs to identify is T . We use S_T to represent the total number of slots in a single reader of TMIA, use M_0 to represent the number of nodes deleted by the dynamic layer, M_1 to represent the number of invalid nodes rejected by the broadcast of the test round, and M_2 to represent the number of valid nodes from other readers' collision slots. M_3

indicates the number of invalid nodes that can be eliminated by the receiving reader when other readers' broadcast results are readable slots, then the total number of slots can be expressed as:

$$S_T = 2(2^N - 1) - M_0 - M_1 - (R - 1)M_2 - (R - 1)M_3 \quad (14)$$

Since the selection strategy of the dynamic layer is inherited from PQT, we get:

$$M_0 = \frac{2(2^N - 1) \cdot \ln(1 - \frac{T}{2^N})}{\ln 0.2} \quad (15)$$

TABLE 3. TMIA tag identification process (Reader C).

READER C	ROUND 1	ROUND 2	ROUND 3
DATA CHANNEL BROADCAST	1111	000	011
TAGS RESPOND	00XX	000X	001X-
TAG IDENTIFIED	-	0000,0001	0010,0011
CONTROL CHANNEL BROADCAST	-	000x , READABLE COLLISION SLOT	001x , READABLE COLLISION SLOT
CONTROL CHANNEL RECEIVES	-	01X0 , COLLISION SLOT 10X1 , COLLISION SLOT	1100 READABLE SLOT 001, IDLE SLOT
INITIAL LAYER PREFIXES[WEIGHT]	000[0.16],001[0.16],	001[0.08]	Null
NEXT PREFIX	00	110	-
IS IDENTIFICATION COMPLETED	X	X	√

Next, M_1 is calculated as follows: According to the Manchester code principle, for each layer of the binary tree, only when the ID of the current layer are all “0” or “1”, the reader receive the unique coding, and eliminate $\frac{1}{2}$ invalid nodes. When the second unique coding occurs, it can continue to eliminate $\frac{1}{2^2}$ invalid nodes. Use k to denote the number of a unique ID that can be confirmed by Manchester code, and then the number of invalid nodes eliminated can be denoted as:

$$M_1 = 2(2^N - 1)(\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} \cdots + \frac{1}{2^k}) \quad (16)$$

and:

$$k \in [0, N - 1]$$

When $k = 0$, the reader receives N collision bits. At this time, the reader can not eliminate any invalid nodes through this round broadcasting. When $k = N - 1$, each response from the first bit to $N - 1$ bit is unique and readable. Meanwhile, all nodes except the current node can be excluded. When $k = N$, there is only one tag in the coverage of the reader, which is same as $k = N - 1$. Since this situation is not practical, it is not included in the discussion. It can be found that the number of nodes that can be removed by test-round broadcasting is only related to the number of collision bits, but not to the order in which the collision bits occur. Under this assumption, in each layer of the binary tree, the probability that the tag is a whole “0” code or whole “1” code equals the T times of N -weight experiment. Since each experimental result is a discrete random event with only two mutually exclusive results, the problem obeys a binomial distribution with $X \sim B(N, P)$:

$$P(X = k) = C_N^k \cdot p^k \cdot (1 - p)^{N-k} \quad (17)$$

Among them, p is the probability that a certain bit in T tags is coded as all “0” code or all “1” code, which can be obtained:

$$p = (\frac{1}{2})^T + (\frac{1}{2})^T \Rightarrow p = \frac{1}{2^T - 1} \quad (18)$$

Take formula (18) substituting formula (17):

$$P(X = k) = C_N^k \cdot (\frac{1}{2^T - 1})^k \cdot (1 - \frac{1}{2^T - 1})^{N-k} \quad (19)$$

The distribution is then used to find its expectations:

$$E(k) = \sum_{k=0}^{N-1} P_k \cdot k \quad (20)$$

Take formula (19) substituting formula (20), and get :

$$E(k) = \sum_{k=0}^{N-1} [C_N^k \cdot (\frac{1}{2^T - 1})^k \cdot (1 - \frac{1}{2^T - 1})^{N-k}] \cdot k \quad (21)$$

At this point, M_1 is obtained:

$$M_1 = 2(2^N - 1) - 2 \sum_{k=0}^{N-1} [C_N^k \cdot (\frac{1}{2^T - 1})^k \cdot (1 - \frac{1}{2^T - 1})^{N-k}] \cdot k - 1 \quad (22)$$

Next, the M_2 and M_3 are calculated as follows: When M_2 is a reader in the collision slot, the number of nodes can be eliminated by other readers. M_2 can be calculated through the probability of a node in the $N-1$ layer which its child-nodes collide multiple tags in one reader:

$$M_2 = 2(2^N - 1) \cdot \frac{2}{2^N - (R - 1) \cdot T} \cdot \frac{1}{2^N - (R - 1) \cdot T - 1} \quad (23)$$

Similarly, we get M_3 :

$$M_3 = C_{R-1}^2 \cdot 2(2^N - 1) \cdot \left(\frac{2}{2^N - (R - 1) \cdot T}\right)^2 \quad (24)$$

The total number of slots is obtained:

$$S_T = 2(2^N - 1) \cdot \left\{ 1 - \frac{\ln(1 - \frac{T}{2^N})}{\ln 0.2} - 2^{-2^{-N(T-1)} \cdot \sum_{k=0}^{N-1} [k(2^{T-1} - 1)^{N-k}]} - \frac{2(R-1)}{2^N - (R-1) \cdot T} \cdot \left[\frac{2}{2^N - (R-1) \cdot T - 1} + \frac{2C_{R-1}^2}{2^N - (R-1)T} \right] \right\} \quad (25)$$

D. SYSTEM EFFICIENCY AND COMMUNICATION COMPLEXITY

System efficiency (η) is the ratio of tags to query times for a single reader. It is one of the key indicators to measure tree-based anti-collision algorithms. The system efficiency of TMIA can be expressed in (26), as shown at the bottom of this page.

Communication complexity is the total number of bits to be transmitted to identify all tags, expressed in C_T . We use the same principle in bit tracking technology: in the single search, the reader broadcasts the prefix ID of the tag through the data channel, the tag returns the remaining ID if the ID of the tag matches the broadcast prefix. The sum of the two IDs is the full length of the tag. After a single search, the reader broadcasts the prefix through the control channel. If it is sent to other readers, the length of the broadcast prefix $N - 1$, we use “0”, “1”, “2” to represent different broadcast results, so the communication complexity of the two channels is the length of the ID of the tag multiplied by the total number of slots. The total communication complexity (C_T) of TMIA is the sum of data channel communication complexity and control channel communication complexity.

$$C_T = 2N \cdot 2(2^N - 1) \cdot \left\{ 1 - \frac{\ln(1 - \frac{T}{2^N})}{\ln 0.2} - 2^{-2^{-N(T-1)} \cdot \sum_{k=0}^{N-1} [k(2^{T-1} - 1)^{N-k}]} - \frac{2(R-1)}{2^N - (R-1) \cdot T} \cdot \left[\frac{2}{2^N - (R-1) \cdot T - 1} + \frac{2C_{R-1}^2}{2^N - (R-1)T} \right] \right\} \quad (27)$$

V. SIMULATION RESULTS

In this section, we use a passive RFID system, which consists of multi-reader and a large number of passive tags. The ID of each tag is unique. The tag IDs are uniformly distributed.

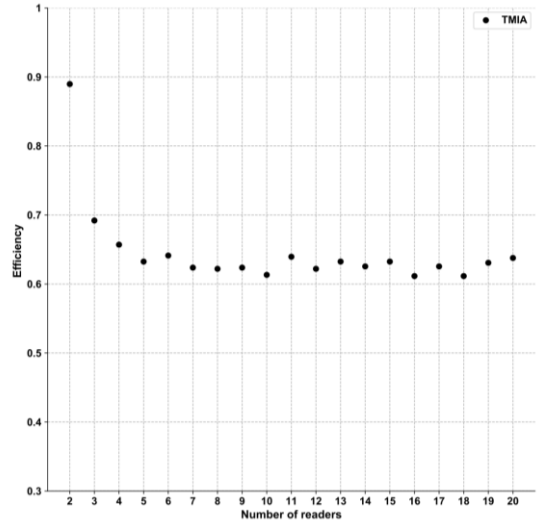


FIGURE 6. Total slots of TMIA in different number of readers.

The tag set is unchanged during the identification process, and the reader has no knowledge of their IDs before the identification process. Since the practical environments have almost the same impact on the comparative protocols, the communication channels between the reader and tags are assumed to be ideal. All the tags’ response signals are assumed to be correctly detected. TMIA is compared with three relative algorithms (PQT [23], OQTT [31], I4QT [32]) which are also enhanced from QT. The tags are evenly distributed near the reader to directly receive the reader’s commands. A widely used benchmark [10], [23], [31], [32], [37] containing 96 bits IDs, total time slots, throughput rates, and communication complexity is used to evaluate the performance under different numbers of tags. Tags are randomly generated, the number of tags increases from 2000 to 20000, and all results are the average of 100 experiments.

A. EFFECT OF READER NUMBER ON ALGORITHM PERFORMANCE

In research of performance under a different number of readers, 2 to 20 readers scenarios have been used in [38], [39]. Since the proposed algorithm is for multi-reader scenarios, the single-reader scenario is not suitable for us. 2-20 readers scenarios were simulated in the experiment, Under the condition of the 96 bits tags ID and each reader contains 10000 tags, the total number of slots is obtained by simulation, the results are shown in figure 6.

It can be found in figure 6 that in two readers scenario, the TMIA average system efficiency of the algorithm can

$$\eta = \frac{T}{2(2^N - 1) \cdot \left\{ 1 - \frac{\ln(1 - \frac{T}{2^N})}{\ln 0.2} - 2^{-2^{-N(T-1)} \cdot \sum_{k=0}^{N-1} [k(2^{T-1} - 1)^{N-k}]} - \frac{2(R-1)}{2^N - (R-1) \cdot T} \cdot \left[\frac{2}{2^N - (R-1) \cdot T - 1} + \frac{2C_{R-1}^2}{2^N - (R-1)T} \right] \right\}} \quad (26)$$

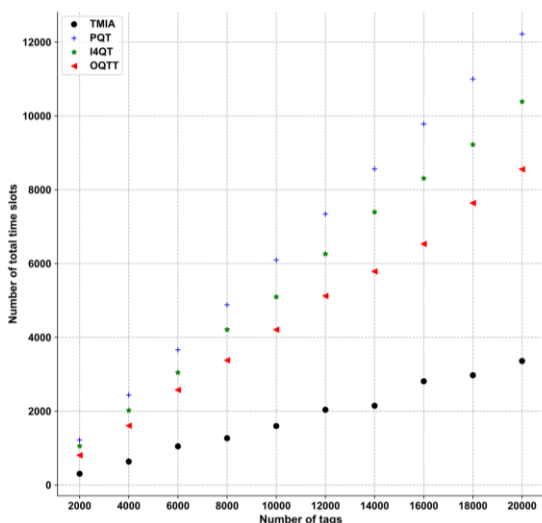


FIGURE 7. Comparison of total slots of tree-based algorithms.

reach 89.79%. The reason is under the case of dual readers, the broadcast result of each reader is 100% valuable for the other reader. When the broadcast slot of one of a reader is an idle time slot, there must be a tag response for the corresponding prefix on the other reader, so a large number of invalid prefixes are avoided broadcast. As the number of readers is bigger than 2, the system efficiency decreases rapidly and stabilizes at about 62.7%. The decrease in system efficiency is caused by the readers' broadcast results no longer 100% valuable to other readers, so there will be readers with a small probability that the result leads to an idle slot. In other readers, the broadcast result is still an idle slot. However, the benefits slots outweigh the misleading slots in the process of identification, which has been proved in section 4.2. Simulation results also confirmed this conclusion. Besides, although the system shows excellent efficiency in the dual readers' scenario since the dual readers' scenario is too demanding for the reader in RFID application scenarios, we use more universal ones in subsequent simulation experiments, which is 3-20 readers scenarios and calculate the average for comparison with other algorithms.

B. THE NUMBER OF TOTAL SLOTS

We analyze the total number of slots of the mentioned algorithms. Figure 7 and Figure 8 shows the average of the number of collision slots and the total number of time slots of the four algorithms under the same conditions. The four algorithms have the same number of readable slots, so they are omitted. It can be observed from the figure that the total number of slots in TMIA, especially the number of collision slots, is less than the other three algorithms. The main reasons are twofold. One is that in the interactive identification mode, the Information shared avoids a large number of repeated broadcasts in existing algorithms. Secondly, other three algorithms only use the bit-tracking in the initial broadcast,

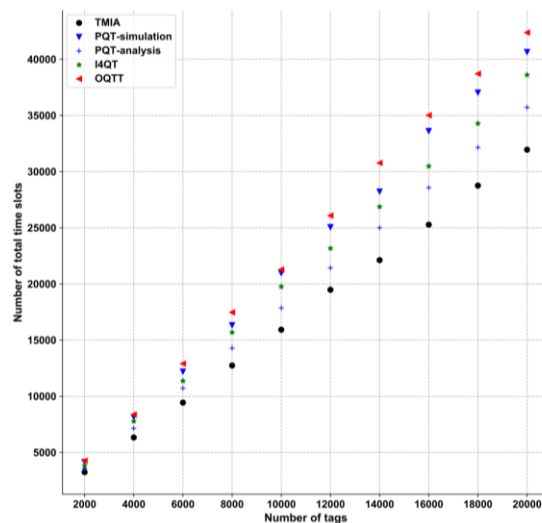


FIGURE 8. Comparison of collision slots of tree-based algorithms.

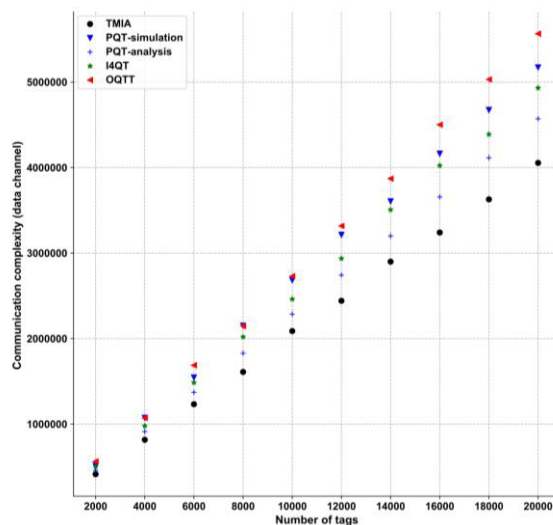


FIGURE 9. Comparison of communication complexity of tree-based algorithms (data channel).

while TMIA uses the bit-tracking and adjusts the weight of the nodes during the entire broadcast. In the process of simulation of the algorithms, we marked the collision slots and calculated its proportion in the total number of slots. The proportion of collision slots in TMIA total number of slots is between 12.44% and 17.92%, while OQTT, I4QT, and PQT are 40.71% to 44.13%, 50.15% to 55%, and 60.34% to 63.12%.

C. SYSTEM EFFICIENCY AND COMMUNICATION COMPLEXITY

Figure 9 shows the communication complexity of the data channel which is used to communicate with the tag by the reader. Figure 10 is a comparison of the total communication complexity. Since the communication complexity of TMIA is the sum of the control channel and the data channel. As can

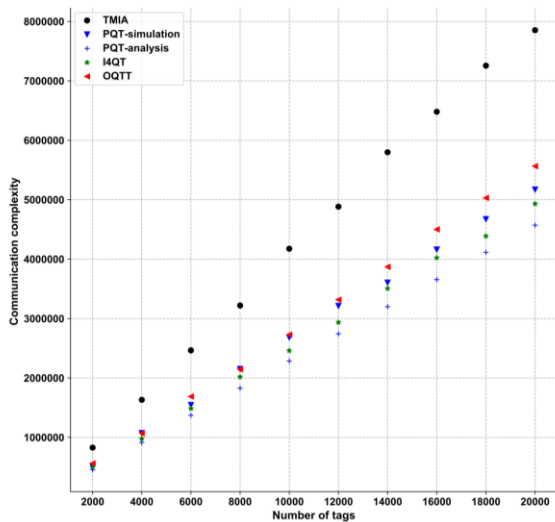


FIGURE 10. Comparison of communication complexity of tree-based algorithms.

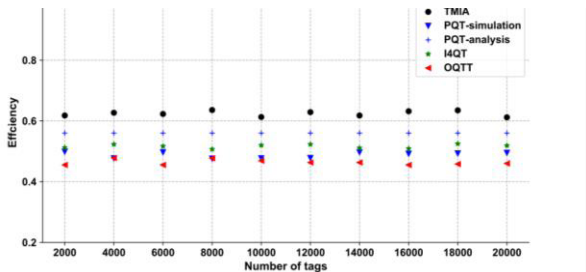


FIGURE 11. Comparison of system efficiency of tree-based algorithms.

be seen from the figure, TMIA communication complexity is lower than the other algorithms, but in comparison of the total communication complexity, the communication complexity of the TMIA is higher than other three algorithms. It is since the TMIA has two channels. Generally speaking, a higher communication complexity will shorten the available work time of the reader battery in the passive RFID system, but in the active RFID system, which is more common in multi-reader scenarios, a higher communication will cause no other negative impact except a higher power consumption, so TMIA's communication complexity is acceptable in this situation.

Figure 11 shows the system efficiency of the four algorithms. The system efficiency is the ratio of the number of tags to the total number of slots, which directly reflect the performance of the algorithm. The system efficiency can be calculated by taking the corresponding number of tags into the formula (25). the average efficiency is 62.31%, the result of the simulation system efficiency average is 62.7%, which is close to the theoretical value.

VI. CONCLUSION

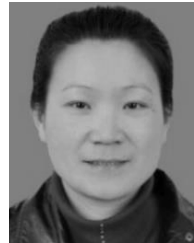
In this paper, the problem of anti-collision for tag identification in the RFID system is discussed. We proposed a tree-based multi-reader interactive anti-collision algorithm.

TMIA takes into consideration a multi-reader synergic method in the RFID tag anti-collision research area. Before the start of the broadcast, the reader calculated the dynamic layer through MIPF and selected an appropriate prefix. Through bit tracking technology, a large number of idle slots and invalid prefixes are eliminated. In further broadcasts, TMIA focuses on the benefits of different broadcast results, especially the idle slots which were useless for their own readers but can increase the tag identification efficiency for other readers. To avoid a large number of potential collision slots in further broadcast, the priority of the prefixes is calculated through information shared by other readers. We calculated expressions of the total number of slots, system efficiency, and communication complexity of the algorithm, making the algorithm's performance verifiable by calculation. In general, TMIA has a positive effect in reducing collision slots. Although it has a higher communication complexity than other algorithms, TMIA has a clear advantage in total slots and system efficiency.

REFERENCES

- [1] *Information Technology, Radio Frequency Identification for Item Management, Part 6: Parameters for Air Interface Communications at 860 MHz to 960 MHz, Amendment 1: Extension with Type C and Update of Types A and B, ISO/IEC, Standard ISO/IEC 18000-6:2004/Amd. 1:(E)*, Jun. 2006.
- [2] M. Azambuja, C. A. M. Marcon, and F. P. Hessel, "Survey of standardized ISO 18000-6 RFID anti-collision protocols," in *Proc. 2nd Int. Conf. Sensor Technol. Appl. (sensorcomm)*, Aug. 2008, pp. 468–473.
- [3] L. Zhu and T.-S.-P. Yum, "The optimal reading strategy for EPC Gen-2 RFID anti-collision systems," *IEEE Trans. Commun.*, vol. 58, no. 9, pp. 2725–2733, Sep. 2010.
- [4] J. Vales-Alonso, V. Bueno-Delgado, E. Egea-Lopez, F. J. Gonzalez-Castano, and J. Alcaraz, "Multiframe maximum-likelihood tag estimation for RFID anticollision protocols," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 487–496, Aug. 2011.
- [5] C.-F. Lin and F. Y.-S. Lin, "Efficient estimation and collision-group-based anticollision algorithms for dynamic frame-slotted ALOHA in RFID networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 4, pp. 840–848, Oct. 2010.
- [6] J. Su, Z. Sheng, L. Xie, and G. Wen, "Idle-slots elimination based binary splitting anti-collision algorithm for RFID," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2394–2397, Dec. 2016.
- [7] L. Zhang, W. Xiang, X. Tang, and Q. Yan, "A time- and energy-aware collision tree protocol for efficient large-scale RFID tag identification," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2406–2417, Nov. 2018.
- [8] 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 Trans. Commun.*, vol. 67, no. 3, pp. 2527–2538, Mar. 2019.
- [9] 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 Process. Lett.*, vol. 23, no. 3, pp. 386–389, Mar. 2016.
- [10] J. Shin, B. Jeon, and D. Yang, "Multiple RFID tags identification with M-ary query tree scheme," *IEEE Commun. Lett.*, vol. 17, no. 3, pp. 604–607, Jan. 2013.
- [11] J. Su, Z. Sheng, A. X. 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 Trans. Wireless Commun.*, early access, Feb. 3, 2020, doi: 10.1109/TWC.2020.2969634.
- [12] 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 Commun.*, vol. 26, no. 3, pp. 118–124, Jun. 2019.
- [13] J. Su, G. Wen, and D. Hong, "A new RFID anti-collision algorithm based on the Q-Ary search scheme," *Chin. J. Electron.*, vol. 24, no. 4, pp. 679–683, Oct. 2015.
- [14] S. Nithya, M. A. Jerlin, and S. Priya, "Tag starvation and tag collisions in RFID system—A solution," *Int. J. Eng. Technol.*, vol. 5, no. 3, pp. 2519–2522, Jun. 2013.

- [15] Y. Jiang and R. Zhang, "An adaptive combination query tree protocol for tag identification in RFID systems," *IEEE Commun. Lett.*, vol. 16, no. 8, pp. 1192–1195, Aug. 2012.
- [16] J. Sung, D. Kim, T. Kim, and J. Choi, "Heuristic query tree protocol: Use of known tags for RFID tag anti-collision," *IEICE Trans. Commun.*, vol. E95-B, no. 2, pp. 603–606, 2012.
- [17] H. Gou and Y. Yoo, "A bit collision detection based hybrid query tree protocol for anti-collision in RFID system," in *Proc. IEEE 11th Int. Conf. Comput. Inf. Technol.*, Aug. 2011, pp. 158–163.
- [18] X. Jia, Q. Feng, and L. Yu, "Stability analysis of an efficient anti-collision protocol for RFID tag identification," *IEEE Trans. Commun.*, vol. 60, no. 8, pp. 2285–2294, Aug. 2012.
- [19] M.-K. Yeh, J.-R. Jiang, and S.-T. Huang, "Parallel response query tree splitting for RFID tag anti-collision," in *Proc. 40th Int. Conf. Parallel Process. Workshops*, Sep. 2011, pp. 6–15.
- [20] J. Hwan Choi, D. Lee, and H. Lee, "Query tree-based reservation for efficient RFID tag anti-collision," *IEEE Commun. Lett.*, vol. 11, no. 1, pp. 85–87, Jan. 2007.
- [21] 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 Trans. Commun.*, vol. 68, no. 2, pp. 998–1012, Feb. 2020, doi: [10.1109/TCOMM.2019.2958426](https://doi.org/10.1109/TCOMM.2019.2958426).
- [22] 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 Trans. Commun.*, vol. 68, no. 4, pp. 2381–2393, Apr. 2020, doi: [10.1109/TCOMM.2020.2968438](https://doi.org/10.1109/TCOMM.2020.2968438).
- [23] X. Liang and Y. Guo, "A probability-based anti-collision protocol for RFID tag identification," *Wireless Pers. Commun.*, vol. 107, no. 1, pp. 57–79, Jul. 2019.
- [24] G. Bagnato, G. Maselli, C. Petrioli, and C. Vicari, "Performance analysis of anti-collision protocols for RFID systems," in *Proc. IEEE 69th Veh. Technol. Conf. (VTC Spring)*, Apr. 2009, pp. 1–5.
- [25] X. Jia, Q. Feng, and C. Ma, "An efficient anti-collision protocol for RFID tag identification," *IEEE Commun. Lett.*, vol. 14, no. 11, pp. 1014–1016, Nov. 2010.
- [26] Y.-H. Chen, S.-J. Horng, R.-S. Run, J.-L. Lai, R.-J. Chen, W.-C. Chen, Y. Pan, and T. Takao, "A novel anti-collision algorithm in RFID systems for identifying passive tags," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 105–121, Feb. 2010.
- [27] K. Finkenzeller, *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*, 2nd ed. Hoboken, NJ, USA: Wiley, 2003.
- [28] X. Liu, S. Zhang, B. Xiao, and K. Bu, "Flexible and time-efficient tag scanning with handheld readers," *IEEE Trans. Mobile Comput.*, vol. 15, no. 4, pp. 840–852, Apr. 2016.
- [29] Y.-C. Lai, L.-Y. Hsiao, and B.-S. Lin, "Optimal slot assignment for binary tracking tree protocol in RFID tag identification," *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 255–268, Feb. 2015.
- [30] C. Law, K. Lee, and K. Siu, "Efficient memoryless protocol for tag identification," in *Proc. 4th Int. Workshop Discrete Algorithms Methods Mobile Comput. Commun.*, Aug. 2000, pp. 75–84.
- [31] Y.-C. Lai, L.-Y. Hsiao, H.-J. Chen, C.-N. Lai, and J.-W. Lin, "A novel query tree protocol with bit tracking in RFID tag identification," *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 2063–2075, Oct. 2013.
- [32] Y. Kim, S. Kim, S. Lee, and K. Ahn, "Improved 4-ary query tree algorithm for anti-collision in RFID system," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.*, 2009, pp. 699–704.
- [33] D. R. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," in *Proc. IEEE Int. Symp. Inf. Theory*, Aug. 1998, p. 107.
- [34] W. Xin, *Research on Anti-Collision Algorithms for RFID Tags and Readers*. Chengdu, China: Univ. of Electronic Science and Technology of China, 2009.
- [35] F. Zhou, D. Jin, C. Huang, and M. Hao, "Optimize the power consumption of passive electronic tags for anti-collision schemes," in *Proc. 5th Int. Conf. (ASIC ICASIC)*, Oct. 2003, pp. 1213–1217.
- [36] W.-C. Chen, S.-J. Horng, and P. Fan, "An enhanced anti-collision algorithm in RFID based on counter and stack," in *Proc. 2nd Int. Conf. Syst. Netw. Commun. (ICSNC)*, Aug. 2007, pp. 1–4.
- [37] W. Zhang, Y. Guo, X. Tang, G. Cui, L. Wu, and Y. Mei, "An efficient adaptive anticollision algorithm based on 4-Ary pruning query tree," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 12, Dec. 2013, Art. no. 848746.
- [38] Y. Chen and F. Zhang, "Study on reader anti-collision algorithm in RFID sensor networks," *Chin. J. Sens. Actuators*, vol. 2, Feb. 2010.
- [39] K.-I. Hwang, K.-T. Kim, and D.-S. Eom, "DiCa: Distributed tag access with collision-avoidance among mobile RFID readers," in *Proc. Int. Conf. Embedded Ubiquitous Comput.* Berlin, Germany: Springer, 2006.



JIANHUA SONG was born in Xiangyang, Hubei, China, in 1973. She is currently an Associate Professor with Hubei University. She is also a member of the Engineering and Technical Research Center of Hubei Province in Educational Informatization, the Engineering and Technical Research Center of Hubei Province in Software Engineering, and the Engineering Research Center of Hubei Province in Intelligent Government Affairs and Application of Artificial Intelligence. Her research interests

include information security and network security.



SHUAI HE was born in Ürtümqi, Xinjiang, China, in 1994. He received the B.S. degree in software engineering from Hubei University, Wuhan, China. He is currently pursuing the master's degree with the School of Computer Science and Information Engineering, Hubei University, Wuhan, China. His research interests include the Industrial Internet of Things and network security.



HUI YAO was born in Jingzhou, Hubei, China, in 1995. She is currently pursuing the master's degree with the School of Computer and Information Engineering, Hubei University, Wuhan, China. Her research interests include information security and cybersecurity.

...