Multiple Resolution Bit Tracking Protocol for Continuous RFID Tag Identification

Weiping Zhu*, Mingzhe Li*, Jiannong Cao†, Zongjian He‡ and Rong Xie*§

*School of Computer Science, Wuhan University, P. R. China

†Department of Computing, The Hong Kong Polytechnic University, Hong Kong

‡Centre for eResearch, Faculty of Science, University of Auckland, New Zealand

Email: {wpzhu,mzli,xierong}@whu.edu.cn, csjcao@comp.polyu.edu.hk, jason.he@auckland.ac.nz

***Abstract*—In recent years, radio frequency identification (RFID) technology has been applied in various fields to identify objects efficiently. Anti-collision protocols are important for RFID tag identifications because it can overcome the problem of unsuccessful identification caused by simultaneous transmission of IDs from multiple tags. Considering that identification is usually performed multiple times, the latest anti-collision approach uses the previous identification results for later identifications, and can identify two tags per unit time. Existing anti-collision protocols ignore bit information, causing low performance problems. In this study, we propose a new approach called multiple resolution bit tracking protocol (MRB) to improve this performance further. This approach dynamically computes a tag set that can be unambiguously identified irrespective of any missing sub-set. The tags in the tag set are requested to transmit their IDs in the same time slot and terminate their identifications after proper processing. We perform extensive simulations to validate the performance of our proposed approach. The results show that MRB can achieve 3.7 tags per unit time, which is 1.85 times the number achieved using the existing approaches.**

***Index Terms*—RFID, multiple resolution, collision tracking tree, MRB.**

I. Introduction

Radio frequency identification (RFID) is an automatic identification technology that uses radio communication to obtain the IDs stored in tags. RFID is widely used in many applications such as supply chain management, human surveillance, and animal management. Compared to the barcode technology, RFID has various advantages, such as non-line-of-sight capability, long distance identification, and high reliability.

A typical RFID system consists of a reader and several tags. During the identification process, the reader sends out a request to the tags, and the tags reply with their pre-stored IDs over a shared wireless medium. When more than one tag transmit their IDs simultaneously, their signals collide, and the reader cannot identify any tags. Therefore, an efficient approach is required to reduce such collisions, which is called a tag anti-collision approach in the RFID field.

The tag identification process is usually performed continuously; thus, the result of the first identification can facilitate and speed up latter identifications. The tree-based anti-collision approach is an important type of RFID tag anti-collision approach for such kind of identification. A tree-based approach [1], [2] follows a tree traversal to continuously split a set of tags into two subsets until each set has only one tag and can be identified without collision. The tree-based approaches are developed mainly based on two algorithms: binary tree (BT) [3], [4] and query tree (QT) [5], [6] algorithms. According to a previous study [7], BT usually outperforms QT in terms of identification time; thus, we will focus on the BT-based approaches.

BT consists of several rounds of identification, and in each round the reader sends a request and requires the tags to respond. Every time after the tags reply with their IDs, the reader notifies the tags about the result of identification, which can be idle, singleton, or collision, denoting no tag, one tag, or more tags responded, respectively. Subsequently, the collided tags are separated into two groups randomly. Such a process is repeated; eventually, there is only one tag in a group, which is identified. Based on BT, the adaptive binary splitting protocol (ABS) [8] records the identification sequence number (called slot number) in the first identification for each identified tag, and later identification can be performed based on such slot numbers rather than starting from scratch. The pair resolution blocking ABS algorithm (PRB) [9] identify two tags together each time unit, thereby achieving performance better than that of ABS. Recently, the bit collision tracking technique [10], [11], [12], [13], [14] is used to speed up the identification process. This technique can determine the location of collisions at the bit level when multiple tags transfer their IDs. This provides more information than previous tag anti-collision approaches.

However, this technique has not been used for multiple times' identifications. We believe that when using the bit collision tracking technique, a performance of two tags per unit time is not optimal because bits of tags can provide more information. It can be improved further. More tags can be identified per unit time by taking advantage of the tags' bit information.

In this study, we proposed a new tree-based RFID anti-collision protocol called multiple resolution bit tracking protocol (MRB). When identifying the tags, we introduce the concept of unique collided set. Each unique collided set includes tags with successive IDs and any sub-set tags of it can be determined using the bit tracking technique. We develop an approach to dynamically compute a unique collided set each

§Corresponding Author: Rong Xie, Email: xierong@whu.edu.cn time and request the tags available in the unique collided set to reply and get identified. The total identification time is proportional to the number of the unique collided set. We perform extensive simulations to validate our proposed approach. In summary, this study makes the following contributions:

- We proposed the concept of unique collided set. This concept contributes to the theory of identifying multiple tags simultaneously in the tree-based anti-collision approaches. This concept also extends the bit tracking technique to provide more useful information based on the identification result of tags in the previous rounds.
- We developed the MRB protocol for RFID tag anti-collisions. MRB can achieve 3.7 tags per unit time when identifying tags, which is 1.85 times the number achieved using the existing approaches.
- We performed extensive simulations to validate the performance of our proposed approach. The results show that our approach achieves good performance in various scenarios.

The remainder of this study is organized as follows. Section II briefly surveys the related works. Section III describes the system model and problem formulation. MRB is presented in Section IV. Section V shows the mathematical analyses for MRB. Section VI reports the results of the simulation experiments. Finally, Section VII concludes this study.

II. Related Work

We briefly review the related works to our problem in this section before describing our proposed algorithm.

A. Anti-collision Approaches

RFID tag anti-collision approaches can be classified into two categories: tree-based and Aloha-based approaches. The Aloha-based approaches [15], [16], [17], [18] divide the identification process into several frames, and in each frame there are several time slots. In a frame, each tag randomly selects a time slot to respond, and the response is received successfully only if one tag selects that time slot. The tree-based approaches [1], [2], [19], [20] follow a tree traversal to continuously split a set of tags into two subsets until each set has only one tag and can be identified without collision. Generally, the Aloha-based approaches are faster in identifying new tags, while the tree-based approaches can more easily utilize the previous identification results for later identifications. Considering that the identification is usually performed continuously, we focus on tree-based approaches in this study.

The tree-based approaches are developed mainely based on binary tree (BT) [3], [4], [21] and query tree (QT) [5], [6] algorithms. The former uses random binary numbers to split tags while the latter uses tag IDs. According to the previous study [7], BT outperforms QT in terms of identification time. Based on BT, adaptive binary splitting protocol (ABS) [8] takes advantage of the identification result in the last round to speed up the identification process. Based on ABS, PRB [9] blocks the identification of new arriving tags from the previous identified tags (called staying tags) by memorizing its interrogating reader's ID in each tag. Moreover, two tags are allowed to be transmitted their IDs simultaneously. If both tags do not leave, the slot will be collided. If only one tag responds, the reader stores its ID and infers that the other tag has been left. If no tag responds, the two tags are confirmed to be left. BCTT [22] is proposed to further improve the execution time by allocating a unique short ID to each identified tag and using such short ID in later data transmission to reduces the data amount to be transmitted. In this study, we will further reduce the time slots required for identification.

B. Bit Tracking based Approaches

Recently, the bit tracking technique [10], [11], [12], [13], [14] is used to speed up the identification process. This technique can determine the collisions in the bit level when multiple tags transfer their IDs. In [10], collision tree protocol (CT) utilizes the collided bits to generate proper prefixes to be sent and split tag groups. In [11], Chen *et al.* presents an algorithm named new enhanced anti-collision algorithm (NEAA) based on BT. All tags are initially partitioned into multiple sets according to number of bit 1 in their IDs; each set of tags will reply the reader at the same time. The ID transmission can be split into several parts according to the collision results. In a special case, several tags can be identified in one time slot if all their IDs have only one bit 1 or 0. In [13], Lai *et al.* proposed an optimal binary tracking tree protocol (OBTT) to separate the tags into proper sets to reduce collisions in the identification. It first estimates the number of tags based on the locations of collided bits, splits the tags into an optimal number of sets based on the estimation and the first collided bit, and and then follows BT to identify tags. Multi-reader RFID tag identification using bit tracking algorithm (MRTI-BT) [14] improves OBTT by transferring only a part of tag ID for collision resolution. Multibit identification protocol (MBI) [12] identifies the tag IDs multibit by multibit. The collided ID strings of the previous time slot are grouped according to a special scheme and then deduced simultaneously. In this study, we further use bit tracking technique to allow as many as possible tags to transfer their information, based on previous identification results.

III. System Model and Problem Description

We assume that in a warehouse, library, or shopping mall, there are N objects required to be monitored. Each object is attached with an RFID tag. We use an RFID reader to obtain the IDs of the tags from time to time. Each time identification is called a round of identification. The tags may move in and out the interrogation region of an RFID reader. We call the tags arriving tags if they are not present in the previous round and arrive in current round, and call the tags staying tags if they keep present in the previous and current round.

Tree-based anti-collision protocol is assumed used for the ID collection. Each round of identification is divided into several frames and each frame includes several slots. A slot is called *idle*, *signlton*, or *collision* when no tag responds, only one tag responds or multiple tags respond, respectively.

In the physical layer, Manchester coding are used in the communications, where a 0 bit is coded by a positive transition, while a 1 bit is coded by a negative transition. Consequently, if two tags simultaneously transmit bits of different values, then the positive and negative transitions of the received bits cancel each other out; thus, a subcarrier signal is received for the duration of an entire bit. The collisions of multiple tags can be determined in the bit level. Supposing that the ID of tag A is "10101" and the ID of tag B is "10110", when tags A and B send their IDs simultaneously using the Manchester coding method, the signal received by the reader is "101xx," where "x" represents a collided bit.

The identification is performed for multiple times. Each identified tag is assumed assigned a unique short ID, and the association between a short ID and a full tag ID is recorded in the reader. In the later identification process, a tag transits its short ID, rather than its tag ID, to reduce the cost of data transmission.

Given the system models described above, we need to design a protocol to identify the tags in the interrogation region of an RFID reader as quickly as possible.

IV. The Procedure of MRB

In this section, we propose our solution, MRB. MRB allows several tags to reply simultaneously to reduce the number of time slots. In the following subsections, we illustrate the key procedures used in MRB.

A. First Round Identification

At first, following ABS [8], our approach adopts a tree-based approach to eliminate collisions. Each tag has two counters, *allocated slot counter (ASC)* and *progressed slot counter (PSC)*. The reader also has two counters, *terminated slot counter (TSC)* and *progressed slot counter (PSC)*. The *PSC*s of the reader and all the tags are the same, recording the number of tags that have been identified. *ASC* is used to allocate tags into different groups to reduce collisions. Initially, both *PSC* and *ASC* are set to 0.

In each slot, the tags, whose *ASCs* are equal to *PSC*, transmit their IDs. If a tag's *ASC* is less than its *PSC*, it is recognized and keeps silent in the later process. For the tags that have transmitted the data, the reader notifies them about the identification result. The result can be idle, singleton, or collision, denoting no tag, one tag, or more than one tag replied, respectively. The tags perform different operations according to the result, which can be described as follows:

- Idle: Each tag decreases its *ASC* by 1.
- Collision: The tags that transmitted IDs in the slot randomly add a binary number to their *ASC*s. The other tags add 1 to their *ASC*s.
- Singleton: Each tag adds 1 to its *PSC*.

On the reader side, *PSC* is increased by 1 when the slot is a singleton slot. *TSC* keeps a track of the largest *ASC* in all tags. The reader adds 1 to its *TSC* in the collision slot and decreases 1 from its *TSC* in the idle slot. When the reader's $TSC < PSC$, it means that all the tags have been identified and the entire process is completed.

Suppose that there are N tags in the reader interrogation region, after the identification for the first round, the *ASCs* of the tags are unique and continuous and have the values 0, 1, 2 ,..., N-1.

B. Later Rounds Identification

After the first round identification, MRB computes the set of tags that can transmit their IDs simultaneously. The tags can be uniquely identified based on the collided bits with each other. We call the tags in such a set *unique collided set*. This approach can identify multiple tags in a slot, therefore we call such processing *multiple resolution technique*.

The unique collided set is formally defined as follows. When the tags in a set (denoted by A) transmit their IDs simultaneously, the resulting collision bits are denoted by *collisionBits*(A). Two collision bits are defined equal if each bit in them is the same, otherwise, are defined unequal. For example, for tag set A = {"001","100"} and set B = {"001","010"}, *collisionBits*(A) = "X0X" and *collisionBits*(B) = "0XX", which are not equal. The tags in a unique collided set A should meet the following two conditions:

a) The ASCs of the tags should be continuous

b) $\forall A_1, A_2, A_1 \subseteq A, A_2 \subseteq A,$ and *collisionBits*$(A_1) \neq$ *collisionBits*(A_2)

The tags in a unique collided set are requested to transmit their IDs in the same time slot. The reader notify the tags the collided bits. Considering that some tags may leave the interrogation region of the reader, the tags can compute such leaving tags and fill in their left ASCs. After the identification, the staying tags' ASCs keeps continuous.

C. The Details of MRB

We illustrate the details computing of unique collided set dynamically and the processing of ASCs as follows:

The approach includes the operations for the reader and tags. The reader operations can be seen in Algorithm 1 and the tag operations can be seen in Algorithm 2. The reader begins the operation using the function *sendIdenRequest*, followed by the replies of the tags using the function *receiveReaderRequest*, then the reader notifies the tags the identification result using the function *receiveTagReplies*, and finally the tags change their statuses using the function *changeASC*.

There are several variables used in the reader. *uniqueCollidedSet* is used to store the *ASC*s of tags in a unique collided set up to now. *collisionBitMapping* is a data structure of key-value, where the keys are all possible collision bits in a tag set, considering that some of the tags in it may be absent, and the values are IDs of tags lead to the collision bits. *curCollisionBitMapping* is a similar data structure; where the keys are the collision bits between the current tag under consideration and the previous considered tags. Initially, *collisionBitMapping* is added an element (*noSignal*, $\emptyset$) denoting no signal received in the channel and the inferred absences of all the tags in a unique collided set, *curCollisionBitMapping* is set to (ID_0, ID_0) denoting only the first tag is present, and *uniqueCollidedSet* is set to empty.

As shown in Algorithm 1, MRB traverses all *ASC*s of tags and forms unique collided sets. From lines 5-9, *curCollisionBitMapping* is computed considering that the current tag and any subset of previously considered tags co-exist. Initially, *curCollisionBitMapping* includes only the current tag (line 4). If the union of *curCollisionBitMapping* and *collisionBitMapping* have no repeated elements, the reader is able to differentiate all situations when the current tag and all the tags in *uniqueCollidedSet* transmit their IDs in one slot. Therefore the *uniqueCollidedSet* and corresponding *collisionBitMapping* are extended, and the next tag is considered following the same process (lines 11-14). The process of extending the unique collided set is stopped in either of two conditions, one is the union of *curCollisionBitMapping* and *collisionBitMapping* has repeated elements (line 15), the other is all staying tags have been identified (line 23). Consequently, the starting *ASC* (i.e. *i* in the protocol, denoted by *PSC*) and the size of *uniqueCollidedSet* (denoted by *groupSize*) are broadcast to all the tags (lines 16-18 and 24-26). When all the tags are handled, *isFinished* is set to *true*.

On receiving the request from the reader, a tag handles it using the function *receiveReaderRequest* of Algorithm 2. The tags whose *ASC* values range from *PSC* and *PSC*+*groupSize*-1 reply their IDs when receiving the request from the reader.

Further, the reader listens to the channel and handles the collision bit received in the function *receiveTagReplies* of Algorithm 1. We stored all the collision bits and corresponding tag IDs in *collisionBitMapping*; thus, it is easy to obtain the replied tags according to the *collisionBits* (line 1). The tags that did not reply are inferred absent and hence removed from current tag list, i.e., *collisionBitMapping*. Correspondingly, the tags' ASCs are changed to fill in the removed tags' (lines 8-12). Finally, a string denoting the collision result is generated and sent to the tags (lines 5, 7, and 15).

Finally, on receiving the notification from the reader, the tags handle it using the function *changeASC* of Algorithm 2. Each tag participated in the replies calculates the number of tags whose *ASC*s are less then it and do not reply, and then its *ASC* is decreased by this value. After this function, all the staying tags keep continuous *ASC*s, which is compatible with the ABS. It is noted that for the operations of tags, the tags available only in a unique collided set reply and all of them are confirmed present or missing in a time slot.

D. Example of Identification

We illustrate the computing of unique collided set using an example shown in Table II. Suppose that *PSC* is 5, and there are four staying tags whose *ASC*s are more than 4. The tags have the IDs of "001," "010," "100," and "111", respectively. MRB tries to add the tags one by one into *uniqueCollidedSet* if the new added tag make the *uniqueCollidedSet* satisfying the definition of the unique collided set. Initially, *uniqueCollidedSet* and *curCollisionBitMapping* is set empty, and *collisionBitMapping* is set to $\{noSignal\}$ denoting no tag transmission. At the step 1, tag "001" is added into *uniqueCollidedSet*. Since this is the first tag considered, *curCollisionBitMapping* is the result when sending tag "001" and *collisionBitMapping* is the empty set together with *curCollisionBitMapping*. At the step 2, tag "010" is added into *uniqueCollidedSet*. The possible collision bits between it and the tag "001" can be "010" if tag "001" is absent or "0XX" if tag "001" is present, which constitute *curCollisionBitMapping*. *curCollisionBitMapping* is further added to *collisionBitMapping*. At the step 3, tag "100" is added into *uniqueCollidedSet*. *curCollisionBitMapping* is "100" if all previous tags are absent, and it is "XX0," "XXX," or"X0X" in other cases. Similarly, at the step 4, tag "111" is added into *uniqueCollidedSet*. Further, in the computed *curCollisionBitMapping*, *XXX* present 4 times (denoted by *XXX*(4)), which denotes some transmissions cannot be distinguished. For example, if tag "001," "010," and "100" transmit signals to readers in a time slot simultaneously, the reader will receive signal "XXX," which is the same with the transmission of tag "100" and "111". Tag "111" is removed from the *uniqueCollidedSet* and generates a new unique collided set.

TABLE I
NOTATIONS IN MRB

uniqueCollidedSet	The ASC of candidate tags belonging to a unique collided set
collisionBitMapping (bitStr, IDList)	All possible collision bits of *uniqueCollidedSet* and corresponding IDs that lead to the collision bits
curCollisionBitMapping (bitStr, IDList)	All possible collision bits between current considering tag with previous considered tags, and corresponding IDs that lead to the collision bits
keys(*collisionBitMapping*) keys(*curCollisionBitMapping*)	The collision bits stored in *collisionBitMapping* or *curCollisionBitMapping*
noSignal	It denotes no signal received in the channel
ID_i	short IDs of the tag whose *ASC* is *i*
size(A)	The number of elements in A
collisionBits	Collision bits received in the channel
collisionBits(*a*,*b*)	Collision bits when transmitting *a* and *b* together
groupSize	The number of tags in the current unique collided set
isFinished	It is used to denote whether the identification of staying tags is finished
collisionResult	It is used to record which bits are collided, 0 for no-collision and 1 for collision
collisionResult[*a*,*b*]	It is the sub string of cr from the left a*th* bit to the b*th* bit, both included

We further illustrate the change of ASCs of tags using an example shown in Figure 1. In the first frame, there are three

Algorithm 1: MRB (Reader Operations)

```
Variable : collisionBitMapping=$\emptyset$, isFinished=false, i=0
Function: sendIdenRequest( )
uniqueCollidedSet = $\emptyset$
collisionBitMapping = $\{(noSignal,\emptyset)\}$
while $i < N$ do
    curCollisionBitMapping = $\{(ID_i, ID_i)\}$
    foreach $e \in$ keys(collisionBitMapping) do
        if $e \neq noSignal$ then
            tags = collisionBitMapping(e) $\cup$ $\{ID_i\}$
            curCollisionBitMapping = curCollisionBitMapping
            $\cup$ (collisionBits($ID_i$,e), tags)
        end
    end
    if there is no repeated elements in
    keys(collisionBitMapping $\cup$ curCollisionBitMapping) then
        collisionBitMapping = collisionBitMapping $\cup$
        curCollisionBitMapping
        uniqueCollidedSet = uniqueCollidedSet $\cup$ $\{i\}$
        $i = i + 1$
    else
        groupSize = size(uniqueCollidedSet)
        PSC = i-groupSize
        broadcast PSC and groupSize
        receiveTagReplies()
        uniqueCollidedSet = $\emptyset$
        collisionBitMapping = $\{(noSignal, \emptyset)\}$
    end
end
groupSize = size(uniqueCollidedSet)
PSC = i-groupSize
broadcast PSC and groupSize
isFinished = true
receiveTagReplies()

Variable : j=0
Function: receiveTagReplies(collisionBits)
identifiedTags = collisionBitMapping(collisionBits)
while $j <$ groupSize do
    if $ID_{i\text{-}groupSize+j} \in$ identifiedTags
    then
        collisionResult = collisionResult + "1"
    else
        collisionResult = collisionResult + "0"
        remove $ID_{i\text{-}groupSize+j}$ from collisionBitMapping
        foreach tag whose $ASC >$ i-groupSize+j do
            decrease the ASC by 1
        end
        i = i - 1
    end
end
broadcast collisionResult and PSC
if isFinished=true then
    return keys(collisionBitMapping)
end
```

tags that have been identified, whose IDs are "001", "010" and "100", and ASCs are 0, 1 and 2, respectively. In the second frame, tag "001" (whose ASC is 0) leaves reader's recognition region. By using our identification approach, the reader firstly generates a unique collided set and put the three tags into the same set. Then, the reader broadcast the start *ASC* as 0 and the group size as 3. Tag "010" and

Algorithm 2: MRB (Tag Operations)

```
Variable : ASC
Function: receiveReaderRequest(int PSC, int groupSize)
if $ASC \geq$ PSC and $ASC <$ PSC +groupSize then
    transmit the ID
end

Function: changeASC(collisionResult, int PSC)
if $ASC \geq$ PSC then
    s = number of "0"s in collisionResult[0, ASC-PSC]
    ASC = ASC − s
end
```

"100" reply while tag "001" does not. So the reader receives *collisionBits* "XX0". The reader identifies the tags "010" and "100", generates *collisionResult* "110", and broadcast it to the tags. On receiving it, tags "010" and "100" decrease their ASC by 1 and keep the ASCs continuous and starting from 0.

E. Compliance with Current Standard

We further discuss the compliance of MRB to ISO/IEC 18000-6B standard (i.e., BT) [3]. MRB is based on ABS; therefore, a few modifications are required to the ISO/IEC 18000-6B standard to implement ABS. Besides this, in MRB, a short ID is required in the tag. The reader should include *PSC* and *groupSize* in the GROUP_SELECT, SUCCESS, FAIL, and DATA_READ commands. The collision result should be further included in the FAIL command. When receiving these commands, the tags should reply their short IDs if their *ASC*s are between *PSC* and *PSC+groupSize*. When receiving the FAIL command, a tag should further extracts the collision result and change its *ASC* based on it. The other process in MRB are built above the physical layer, including calculating *uniqueCollidedSet*, *curCollisionBits*, *collisionBits* and analyzing tags' response of the collision bits, which is compliance with the ISO/IEC 18000-6B standard.

V. Performance Analysis

In this section, we analyze the performance of MRB. Suppose there are *n* tags, and each tag has a short ID of length *m*. First, we analyze the probability of identifying these *n* tags in a slot , then we calculate the mathematical expectations of the number of tags identified in a slot.

A. Identification Probability

In our problem, a tag can be distinguished from other tags if it owns at least one unique bit in its ID. A unique bit is a bit for which all the other tags in the unique collided set have a different value. For example, if tag A's ID has the value of 1 at the second bit, this bit is a unique bit only if the second bit of all the other tags in the unique collided set is 0.

Assuming the tags' IDs follow a uniform distribution, a bit of an ID has a probability of $\left(\frac{1}{2}\right)^{n-1}$ to be a unique bit. There are total m bits that can be allocated to the tags as unique bits. Considering the tags sequentially, we assume that the *i*th tag has k_i unique bits in its ID. Furthermore, after the *i-1* tags

TABLE II
AN EXAMPLE OF UNIQUE COLLIDED SET COMPUTATION

step number	current ASC	current ID	uniqueCollidedSet	keys(curCollisionBitMapping)	keys(collisionBitMapping)	continue?
1	5	001	{5}	{001}	{001,*noSignal*}	yes
2	6	010	{5,6}	{0XX,010}	{010,0XX,001,*noSignal*}	yes
3	7	100	{5,6,7}	{XX0,XXX,X0X,100}	{100,X0X,XX0,XXX,010,0XX,001,*noSignal*}	yes
4	8	111	{5,6,7,8}	{1XX,XXX(4),X1X,XX1,111}	{1XX,XXX(5),X1X,XX1,111, 100,X0X,XX0,010,0XX,001,*noSignal*}	no

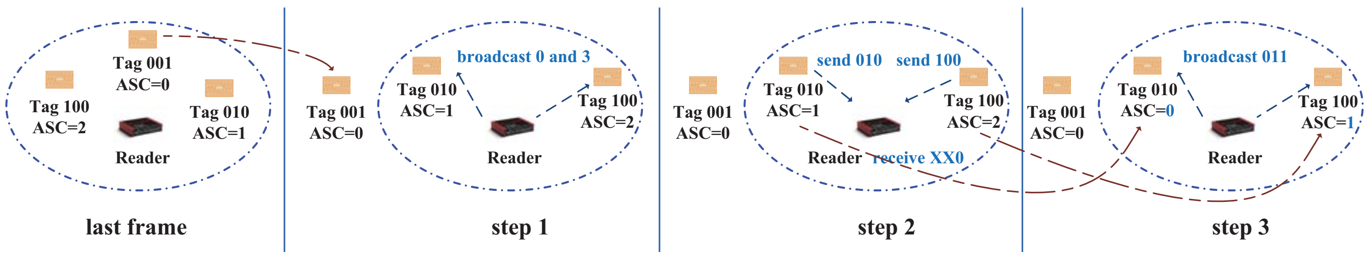

Fig. 1. An example of the ASC Change

determined their unique bits, the probability that the *i*th tag has k_i unique bits is

$$C^{k_i}_{m-\sum_{t=1}^{i-1} k_t}\left(\frac{1}{2}\right)^{k_i(n-1)} \tag{1}$$

Further, we calculate the probability of a tag set, in which their unique bits are $k_1, k_2 ...., k_n$, as shown below:

$$\prod_{i=1}^{n} C^{k_i}_{m-\sum_{t=1}^{i-1} k_t}\left(\frac{1}{2}\right)^{k_i(n-1)} \tag{2}$$

Let P(*n*, *m*) denote the probability that all *n* tags in a set can be identified, where the length of the tags' ID is *m*. We enumerate all the possibilities of unique slots allocated to the tags, and obtain P(*n*, *m*) as follows:

$$P(n,m) = \sum_{\sum_{i=1}^{n} k_i \le m}\left(\prod_{i=1}^{n} C^{k_i}_{m-\sum_{t=1}^{i-1} k_t}\left(\frac{1}{2}\right)^{k_i(n-1)}\right) \tag{3}$$

It is noted that the sum of k_i cannot exceed *m*.

B. Mathematical Expectation

Let Q(*n+1*, m) denote the probability that n tags can be identified, while the (*n+1*)th tag cannot be identified together in a slot. This means that the (*n+1*)th cannot have a unique bit with the previous considered tags. First, we calculate the probability that there are no tags that have unique bit as

$$1 - C^2_{n+1} \times \left(\frac{1}{2}\right)^{n+1} \times 2 \tag{4}$$

We compute Q(*n+1*, m) as follows:

$$Q(n+1,m) = P(n,m) \times \left(1 - C^1_{n+1}\left(\frac{1}{2}\right)^{n-1}\right)^{m-\sum_{i=1}^{n} k_i} \tag{5}$$

Further, we compute the mathematical expectations of the number of tags identified in a time slot. At least two tags are identified within one time slot; thus, *n* is not less than 2. In addition, each tag must own at least one unique bit; thus, *n* must not exceed *m*. We calculate the mathematical expectation as follows:

$$E = \sum_{n=2}^{m} n \times Q(n+1,m) \tag{6}$$

When *n* is 50 and m is 8, this value is 3.43 tags per time slot; when *n* is 50 and m is 16, this value is 3.61 tags per time slot. This result is much better than 2 tags per time unit achieved by existing approaches.

C. Discussions

In this paper, we assumed that the communication is reliable, i.e., the requests from the readers and responses from the tags are intact. Although this assumption is commonly used in the existing tag identification algorithms [23], [24], [25], it may not be true for real applications. To solve this problem, we can build a probability model for the communication like in [1], [26], [27], and require the readers and tags to communicate for sufficient times according to a required readability. The detailed design is out of the scope of this study and is left as future work.

VI. PERFORMANCE EVALUATION

We evaluate the performance of MRB by comparing it with existing tree-based algorithms and bit tracking based algorithms including ABS, PRB, BCTT, OBTT, MRTI-BT, and MBI. A hundred simulations are repeated to obtain each data point of the figures.

Let *l* denote the length of tag ID, *s* denote the length of a short ID, and *n* denote the number of tags. The execution time is computed as a function of the bits transmitted. The requests sent from the reader are the same in these approaches; thus,

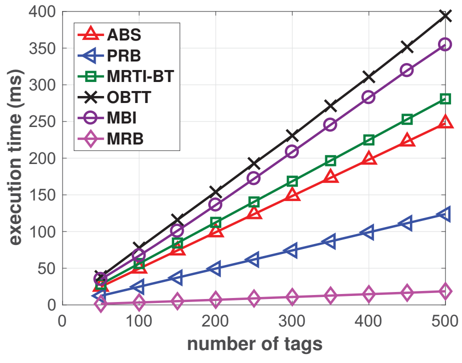

Fig. 2. Execution times of different approaches with varying number of tags

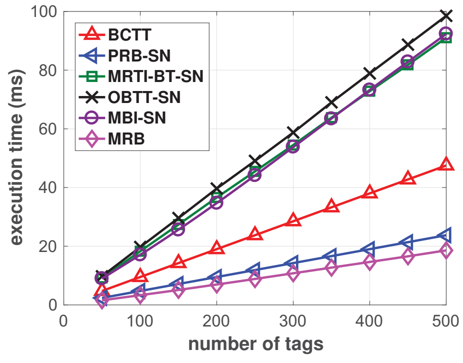

Fig. 3. Execution times of different approaches (replying short ID) with varying number of tags

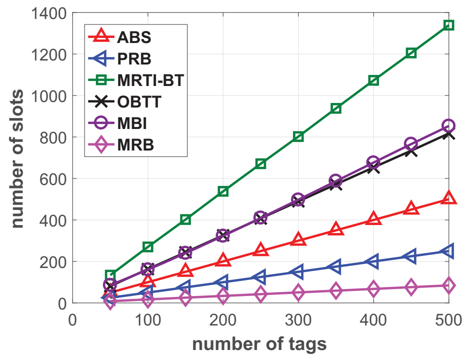

Fig. 4. Number of slots of different approaches with varying number of tags

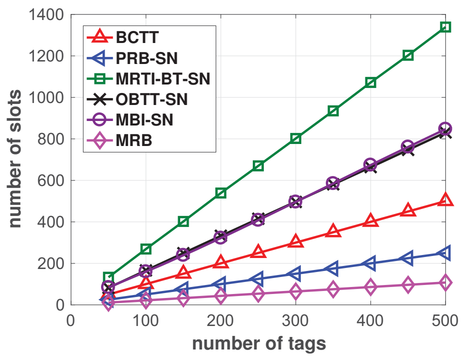

Fig. 5. Number of slots of different approaches (replying short ID) with varying number of tags

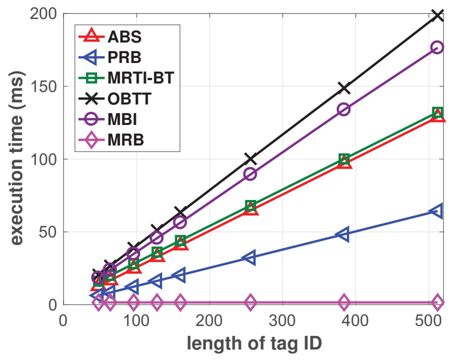

Fig. 6. Execution times of different approaches with varying length of tag ID

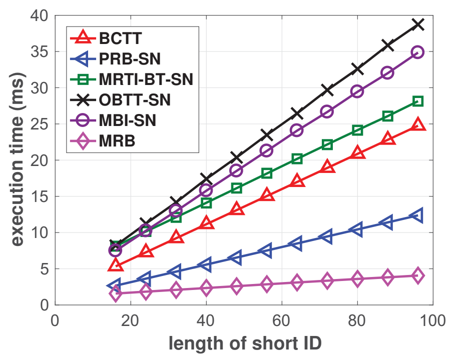

Fig. 7. Execution time of different approaches with varying length of short ID

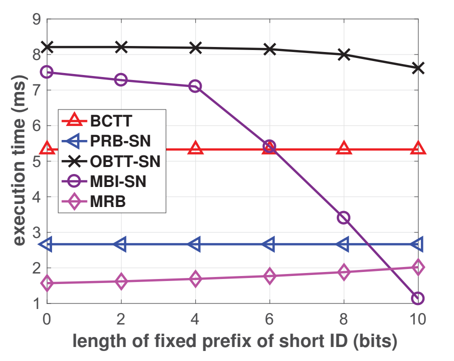

Fig. 8. Execution time of different approaches with varying length of a fixed prefix of short ID

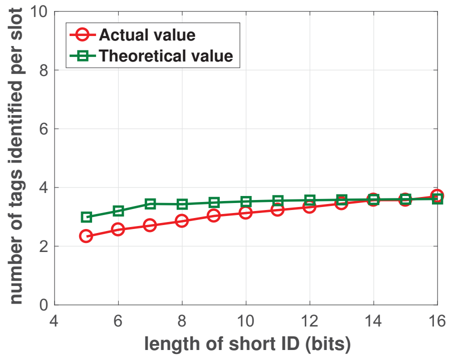

Fig. 9. Theoretical and actual values of the identified tags in a time slot

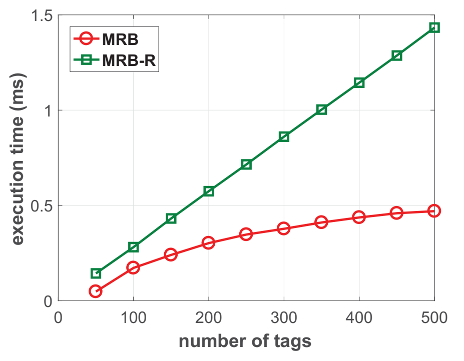

Fig. 10. Execution time of MRB and MRB-R with varying number of tags

they are not considered. Each bit is transmitted for $b\mu$s. By default, we set l=96 bits, s=16 bits, n=50, and b=5.

A. Impact of Number of Tags

First, we vary the number of tags to compare the execution time for the different approaches; the result is shown in Fig. 2. It can be seen that the execution times of all the approaches increase with the increase of the number of tags. The execution times of OBTT, MBI, and MRTI-BT are greater than those of ABS, PRB, and MRB, because the information in previous identification is not utilized. The execution time of PRB is half of that of ABS, because of PRB's pair resolution strategy. MRB further reduces the execution time by allowing multiple tags to be identified in a time slot. When the number of tags is 500, the execution time of MRB is 3.26% of ABS, 6.52% of PRB, 2.05% of BCTT, 2.28% of MBI, and 2.87% of MRTI-BT, respectively.

To keep the comparison fair, we revise the approaches by allowing them to use a short ID rather than a tag ID to reply to the reader. The result is shown in Fig. 3. We add the suffix "-SN" to their names to distinguish the modified approaches. BCTT in fact equals to ABS-SN in this environment. It can be seen that the execution times are greatly reduced compared with those shown in Fig. 2, while their trends are similar. When the number of tags is 500, the execution time of MRB is 16.99% of BCTT, 33.99% of PRB-SN, 8.19% of OBTT-SN, 8.74% of MBI-SN, and 8.86% of MRTI-BT-SN, respectively.

The execution time computed above depends only on the data amount to be transmitted; thus, we further compare the number of slots in different approaches while varying the number of tags. The results are shown in Fig. 4 and 5. Similarly, the slots of OBTT, MBI, MRTI-BT (and also OBTT-SN, MBI-SN, and MRTI-BT-SN) are more than those of ABS, PRB, and MRB (BCTT, PRB-SN, and MRB). The number of slots of ABS is the same with BCTT, and so with PRB and PRB-SN. This is because the identification processes of them are the same, but using short ID instead of tag ID in the transmission. According to the figure, the trends of these approaches are similar to that shown in Fig. 2 and 3. Therefore, if the execution time also considers the transition of time slots, the result is similar to the one achieved. When the number of tags is 500, the number of time slots of MRB is 16.89% of ABS and BCTT, 33.78% of PRB and PRB-SN, 10.34% of OBTT, 10.16% of OBTT-SN, 9.90% of MBI, 9.97% of MBI-SN, 8.56 % of MRTI-BT-SN, and 6.31% of MRTI-BT, respectively.

B. Impact of Length of Tag ID

Further, we vary the length of tag ID to evaluate the execution time of different approaches; the result is shown in Fig. 6. The approaches that use a short ID rather than a tag ID in the data transmission are not affected by this factor, therefore they are not included in this figure. It can be seen that the execution times of OBTT, MBI, and MRTI-BT are still greater than those of ABS, PRB, and MRB. The execution times of ABS and PRB increase with the increased length of ID, because they directly transmit their IDs in each reply to the reader. When the length of tag ID is 96 bits, the execution time of MRB is 6.67% of ABS, and 13.34% of PRB, respectively. It is reasonable that each staying tag is allocated a short ID because they will be identified multiple times later.

C. Impact of Length of Short ID

To explore the effect of the length of short ID on the execution time, we varied the length of short ID from 16 to 96 bits in our evaluation; the result is shown in Fig. 7. BCTT, PRB-SN, MRTI-BT-SN, OBTT-SN, MBI-SN, and MRB are included in the figure. It is noted that the length of short ID depends on the number of tags identified, because a short ID represents a unique ID of a tag.

As shown in the figure, the approaches have a higher execution time with the increase of the length of short ID. MRB has a slower increasing speed compared to those of other approaches. This is because the increased length of short ID identifies more tags together in a same slot. Among all the approaches, MRB achieves the least execution time.

D. Impact of Distribution of Short ID

In the above simulations, we assume that tag IDs and short IDs follow a uniform distribution. However, in the practice, there is a high probability that most tags' IDs or short IDs share the same prefix. To simulate the occurrence of this situation, we fix a prefix in a short ID and make the other parts of short ID uniformly distributed. The performance of ABS and PRB are not affected by the distribution of short ID or tag ID because they do not use this information. We vary the length of the fixed prefix from 0 to 10 bits to check the performance. The result is shown in Fig. 8.

It can be seen that with less fixed prefix of short ID, MRB has better performance because each tag has more freedom to choose unique bits and it is more likely to identify more tags in a slot. MBR's performance deteriorates when the length of fixed prefix increases. The execution time of MBR when the length of fixed prefix is 0 is 70.63% of that when the length of fixed prefix is 10. The execution times of BCTT and PRB-SN are stable because the distribution of short ID has not affect their communications. The execution time of OBTT-SN is slightly decreased with the increase of the length of fixed prefix of short ID, while MBI has a significant reduction. When the length of fixed prefix of short ID is more than 10, the execution time of MBI is less than that of MRB. However, in this case, the short ID has less than 6 bits valid ID, and hence can support no more than 64 tags. In a common scenario, MRB has the least execution time.

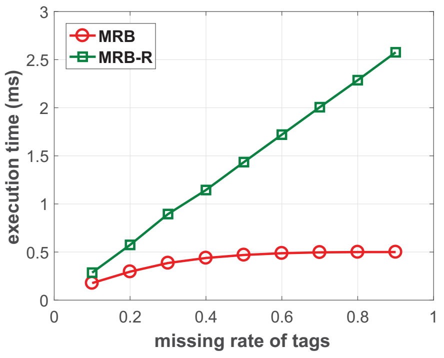

Fig. 11. Execution time of MRB and MRB-R with varying missing rate of tags

E. Comparison of Theoretical Value and Actual Value

In section V, we compute the average number of tags that can be identified in a slot. Further, we compare the computed values with those obtained in the simulations.

We change the length of short ID and compare their values. The result is shown in Fig. 9. As shown in the figure, when the length of short ID increases from 5 to 16 bits, the theoretical and actual values are quite close. This confirms the correctness of the computation in the performance analysis. According to the figure, in our experiments, MRB can achieve 3.7 tags per unit time when using a 16 bits short ID, which is 1.85 times the number achieved using PRB.

F. MBR Performance in a Dynamic Environment

ABS has a desirable performance in a dynamic environment, in which the tags may move out of the interrogation region of the reader and the new tags may move into the region.

The data structure in ABS, including ASC and TSC, changes accordingly. To make MBR compatible with ABS, the tags should be allocated ASCs continuously. There are several approaches to implement this. One is the MRB approach proposed in this study. An alternative approach is to record the ASCs of all missing tags and allocating them to the arriving tags. We call this approach as MRB-R.

We compare the execution time of these two approaches by varying the number and missing rate of the tags. The results are shown in Fig. 10 and Fig. 11. The result shown in Fig. 10 is performed in a scenario with both 10% missing tags and 10% arriving tags. It can be seen that the execution time of MRB-R increases linearly with the number of tags, whereas the execution time of MRB increases quite slowly as compared to that with MRB-R. Further, we vary the missing rate from 0.1 to 0.9 to check the performance of MRB and MRB-R. The result is shown in Fig. 11 and the trend is similar to that observed in Fig. 10. This is because the number of missing tags depends on both the missing rate and the total number of tags. These results clearly show that MRB outperforms MRB-R.

VII. Conclusion

In the study, we propose a new RFID anti-collision protocol called MRB by using the multiple resolution technology. We proposed the concept of unique collided set in which the tags can transmitted simultaneously. The tags in it can be identified without collisions based on Manchester coding system and our designed protocol. According to our evaluation, MRB can achieve 3.7 tags per unit time when identifying tags, which is 1.85 times the number achieved using the existing approaches. MRB is compatible with ABS and hence can handle various dynamic environment.

Acknowledgment

We thank Yifei Ma for analyzing the performance of MRB. This research is supported in part by National Natural Science Foundation of China No. 61502351, National Key R&D Program of China No. 2018YFC1604000, Chutian Scholars Program of Hubei, and 2018 Science and Technology Transformation Project of Grain Administration of Hubei Province "Grain and Oil Quality & Safety Assurance System Research".

References

[1] 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.

[2] 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 Transactions on Mobile Computing*, vol. 12, no. 10, pp. 2063–2075, 2013.

[3] ISO/IEC 18000-6, "Information technology automatic identification and data capture techniques–radio frequency identification for item management air interface–part 6: Parameters for air interface communications at 860–960 mhz," 2004.

[4] W.-C. Chen, S.-J. Horng, and P. Fan, "An enhanced anti-collision algorithm in RFID based on counter and stack," in *Proc. of the 2nd International Conference on Systems and Networks Communications (ICSNC)*, 2007, pp. 21–21.

[5] T.-P. Wang, "Enhanced binary search with cut-through operation for anti-collision in RFID systems," *IEEE communications letters*, vol. 10, no. 4, pp. 236–238, 2006.

[6] J. Ryu, H. Lee, Y. Seok, T. Kwon, and Y. Choi, "A hybrid query tree protocol for tag collision arbitration in RFID systems," in *Proc. of IEEE International Conference on Communications*, 2007, pp. 5981–5986.

[7] G. Bagnato, G. Maselli, C. Petrioli, and C. Vicari, "Performance analysis of anti-collision protocols for RFID systems," in *Proc. of IEEE the 69th Vehicular Technology Conference (VTC Spring)*, 2009, pp. 1–5.

[8] J. Myung, W. Lee, and J. Srivastava, "Adaptive binary splitting for efficient RFID tag anti-collision," *IEEE communications letters*, vol. 10, no. 3, pp. 144–146, 2006.

[9] Y.-C. Lai and C.-C. Lin, "Two blocking algorithms on adaptive binary splitting: single and pair resolutions for RFID tag identification," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 3, pp. 962–975, 2009.

[10] X. Jia, Q. Feng, and C. Ma, "An efficient anti-collision protocol for RFID tag identification," *IEEE Communications Letters*, vol. 14, no. 11, pp. 1014–1016, 2010.

[11] 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 Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 105–121, 2010.

[12] Y. Wang, L. Yi, H. Leung, R. Chen, and L. An, "A multi-bit identification protocol for RFID tag reading," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3527–3536, 2013.

[13] Y.-C. Lai, L.-Y. Hsiao, and B.-S. Lin, "Optimal slot assignment for binary tracking tree protocol in RFID tag identification," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 255–268, 2015.

[14] A. Fahim, T. Elbatt, A. Mohamed, and A. Al-Ali, "Towards extended bit tracking for scalable and robust RFID tag identification systems," *IEEE Access*, vol. 6, no. 99, pp. 27 190–27 204, 2018.

[15] 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.

[16] X. Liu, B. Xiao, S. Zhang, and K. Bu, "Unknown tag identification in large RFID systems: An efficient and complete solution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1775–1788, 2015.

[17] Y. Wang, J. Liu, X. Wang, F. Zhu, and L. Chen, "Missing tag identification in open RFID systems," in *Proc. of IEEE International Conference on Communications*, 2017, pp. 1–6.

[18] Y. Yin, L. Xie, J. Wu, and S. Lu, "Focus and shoot: Exploring auto-focus in RFID tag identification towards a specified area," *IEEE Transactions on Computers*, vol. 65, no. 3, pp. 888–901, 2016.

[19] H. Gou, H.-c. Jeong, and Y. Yoo, "A bit collision detection based query tree protocol for anti-collision in RFID system," in *Proc. of IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2010, pp. 421–428.

[20] L. Zhang, W. Xiang, X. Tang, Q. Li, and Q. Yan, "A time-and energy-aware collision tree protocol for efficient large-scale RFID tag identification," *IEEE Transactions on Industrial Informatics*, 2017.

[21] M. Jacomet, A. Ehrsam, and U. Gehrig, "Contactless identification device with anticollision algorithm," in *Proc. of IEEE Conference on Circuits, System, Computers and Communications*, 1999, pp. 269–273.

[22] J. Liu and Q. Feng, "A blocking collision tracking tree algorithm in mobile RFID systems," in *Progress In Electromagnetics Research Symposium-Spring (PIERS)*, 2017, pp. 2520–2525.

[23] X. Liu, B. Xiao, S. Zhang, and K. Bu, "One more hash is enough: Efficient tag stocktaking in highly dynamic RFID systems," in *Proc. of IEEE International Conference on Communications*, May 2016, pp. 1–6.

[24] T. Li, S. Chen, and Y. Ling, "Identifying the missing tags in a large RFID system," in *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2010, pp. 1–10.

[25] X. Liu, K. Li, G. Min, Y. Shen, A. X. Liu, and W. Qu, "Completely pinpointing the missing RFID tags in a time-efficient way," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 87–96, Jan 2015.

[26] L. Xie, B. Sheng, C. C. Tan, H. Han, Q. Li, and D. Chen, "Efficient tag identification in mobile RFID systems," in *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*, 2010, pp. 1–9.

[27] K. Fyhn, R. M. Jacobsen, P. Popovski, and T. Larsen, "Fast capturełrecapture approach for mitigating the problem of missing RFID tags," *IEEE Transactions on Mobile Computing*, vol. 11, no. 3, pp. 518–528, 2012.