

## 一种基于碰撞位指示的射频识别标签防碰撞算法

李志坚<sup>\*①②</sup> 赖顺桥<sup>②</sup>

<sup>①</sup>(华南理工大学电子与信息学院 广州 510640)

<sup>②</sup>(广州市光机电技术研究院 广州 510663)

**摘要:** 多标签碰撞是射频识别(RFID)技术在推广应用时必须克服的一个问题。针对目前 RFID 标签防碰撞算法存在识别效率低的不足, 该文提出一种基于碰撞位指示的 RFID 标签防碰撞的碰撞位指示算法(CBIA)。通过跟踪待识别标签的碰撞位, 采用碰撞位编解码技术, 对待识别标签进行重复分组, 直到所有标签都被正确识别。算法通过确定性分组, 避免了空闲时隙的产生。仿真结果表明, 采用 CBIA 算法的多标签识别系统, 吞吐率可以达到每时隙 0.7 个标签, CBIA 算法识别效率优于优化查询跟踪树算法(OQTT)和碰撞跟踪树算法(CTTA)算法。

**关键词:** 射频识别; 防碰撞算法; 位跟踪技术; 吞吐率

中图分类号: TN92

文献标识码: A

文章编号: 1009-5896(2014)12-2842-06

DOI: 10.3724/SP.J.1146.2013.01759

## An Anti-collision Algorithm Based on Collided Bits Indicator in Radio Frequency Identification Systems

Li Zhi-jian<sup>①②</sup> Lai Shun-qiao<sup>②</sup>

<sup>①</sup>(School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China)

<sup>②</sup>(Guangzhou Institute of Optics Mechanics and Electrics, Guangzhou 510663, China)

**Abstract:** Multiple tags collision becomes an important factor blocking the popularization of Radio Frequency Identification (RFID). To improve the identification efficiency and reduce the communication overhead, an novel algorithm, anti-Collided Bits Indicator Algorithm (CBIA) is proposed base don Bits Indicator Algorith (BIA). Using the collision bit tracking technology and collided bits coding technology, the reader splits the tags into smaller subsets according to the identified collided bits. This process is repeated until all collided bits are solved. CBIA groups the tags into determinate subsets to avoid generating idle slots. The analysis and simulation results show that the average throughput of CBIA is 0.7 tags per slot, which is better than that of other algorithms, such as Optimal Query Tracking Tree protocol (OQTT) and Collision Tracking Tree Algorithm (CTTA).

**Key words:** Radio Frequency IDentification (RFID); Anti-collision algorithm; Bit-tracking technology; Throughput

### 1 引言

射频识别(Radio Frequency IDentification, RFID)是上世纪 90 年代兴起的一种非接触式自动识别技术。近年来, 随着物联网技术的发展, RFID 被广泛应用于军事、供应链、卫生保健和公共管理等领域<sup>[1,2]</sup>。由于 RFID 系统标签和读写器共用一个无线频道, 当多读写器和多标签同时占用频道通信时, 信号相互干扰, 发生了碰撞。

RFID 多标签防碰撞算法主要使读写器正确识别有效通信范围内的多个标签。根据所采用的无线

通信多址接入方式, 这些算法可以分为时分多址(TDMA)类, 码分多址(CDMA)类, 空分多址(SDMA)类和频分多址(FDMA)<sup>[3]</sup>。然而, 受 RFID 系统成本和资源约束, 时分多址技术在 RFID 多标签防碰撞算法中应用最为广泛。基于 TDMA 技术的防碰撞算法又可分为 ALOHA 算法<sup>[4,5]</sup>和基于树的分组算法<sup>[6-10]</sup>两类。

ALOHA 算法易于实现, 成本低廉, 但是由于算法的时隙是随机分配, 因此可能造成某些标签长时间无法识别, 亦即存在标签饥饿(tag starvation)问题。

基于树的分组算法是确定型算法, 算法依据确定的分组规则, 对活动标签反复分组, 直到每个组内只含一个标签或者不含标签, 从而识别所有标签。根据分组规则的不同, 基于树的分组算法又进一步分为二进制树搜索算法(Binary Tree protocol, BT)

2013-11-08 收到, 2014-06-30 改回

广东省科技计划 (2010A011300016), 广州市科技计划 (2011J4100034)和中央高校基本科研业务费自主项目(2009ZM0097)资助课题

\*通信作者: 李志坚 leemu1001@126.com

和查询树算法(Query Tree protocol, QT)。基本的 BT 算法和 QT 算法都存在空闲时隙,降低了系统效率。

采用碰撞位跟踪技术可以提高基于树的分组算法的系统效率。增强防碰撞算法(Enhanced Anti-collision Algorithm, EAA)和碰撞跟踪树算法(Collision Tracking Tree Algorithm, CTTA)<sup>[11,12]</sup>分别是对基本的 BT 算法和 QT 算法,都采用了碰撞位跟踪技术。这两种算法都检测最高碰撞位,之后按照最高碰撞位分别为‘0’和‘1’把碰撞标签分成两组,从而避免了空闲时隙的产生,提高了系统效率。

但是,当活动标签数量很大时,EAA 算法和 CTTA 算法都存在大量的碰撞时隙。减少碰撞时隙可以进一步提高系统效率。文献[8]提出的优化查询跟踪树算法(Optimal Query Tracking Tree, OQTT)算法。该算法在识别开始时,先通过碰撞位置估计标签数量,之后依据最优分组规则把待识别标签分成若干组,每一组都采用 CTTA 算法进行识别。OQTT 减少了识别最初的碰撞时隙数量,提高了识别效率。

OQTT 算法只在查询树首层增加了分支,而在每个子查询树里面仍采用 CTTA 算法。因此,在标签数量很大的时候,识别初期的碰撞时隙还是比较多。进一步减少碰撞时隙,可以进一步提高系统效率。

本文正是基于进一步减少碰撞时隙的思想,提出一种基于碰撞位指示的 RFID 多标签防碰撞算法(Collided Bits Indicator Algorithm, CBIA)。CBIA 通过跟踪待识别标签的碰撞位,采用碰撞位编解码技术,对待识别标签进行重复分组,直到所有标签都被正确识别。算法的创新点在于增加了查询树每一层的分枝数量,同时避免空闲时隙的产生,进一步压缩了查询树的层数,减少了碰撞时隙的数量,提高了系统效率。本文通过数学分析,给出了 CBIA 算法的占用时隙的估计模型,并在此技术上得出了系统吞吐率估计。仿真结果表明,相对于现有查询树算法 CTTA 和 OQTT 算法,CBIA 具有更快的识别速率和更高的吞吐率。

## 2 基于碰撞位指示(CBI)的 RFID 防碰撞算法

### 2.1 算法的预设条件

(1)标签 ID 采用曼彻斯特编码<sup>[13]</sup>,长度为  $L$ ,读写器能够正确识别非碰撞位和碰撞位。读写器中设置一个查询前缀堆栈  $S$  和两个寄存器碰撞位模板(Collided Bits Mask, CBM)和识别位模板(Recognized Bits Mask, RBM),分别用于记录碰撞位和已识别位数据。

(2)标签中设置一个  $2^m$  位的碰撞位指示(CBI)寄存器。当接收读写器广播的 CBM 时,标签顺序取出 ID 中的高  $m$  位碰撞位,形成一个二进制数,这个二进制数成为碰撞位编码 CBC(Collided Bits Code),然后根据 CBC 把 CBI 的第 CBC 位置‘1’,其他位置‘0’。

(3)标签中设置一个状态寄存器 SF,标签根据 SF 的值在不同的时候响应读写器命令。

### 2.2 算法的命令格式

(1)Act (0)命令:读写器广播这个命令,有效通信范围内所有标签回送各自 ID 给读写器同时把状态标志位 SF 置‘1’。

(2)Req ( $x$ , opt): 查询命令。其中参数‘ $x$ ’是一个长度为  $l(x)$  的二进制数,可以是查询前缀序列或者 CBM,由参数 opt 区分。参数 opt 为‘0’时,‘ $x$ ’是一个查询前缀序列。若标签 ID 的高  $l(x)$  位与‘ $x$ ’一致,则标签回送剩下的  $L-l(x)$  位给读写器并把状态标志位 SF 置‘1’。参数 opt 为‘1’时,‘ $x$ ’是一个 CBM。状态标志位 SF 为‘1’的标签回送 CBI 给读写器并把 SF 清零。

### 2.3 算法流程图

CBIA 算法的流程图如图 1 所示,其主要步骤如下:

步骤 1 读写器初始化,清空堆栈  $S$ ,清零 CBM 和 RBM。广播 Act (0)命令。识别范围内所有标签回送 ID 并置位 SF;

步骤 2 读写器检测碰撞位,设置 CBM 和 RBM。如果没有检测到碰撞,读写器识别一个标签;若只存在一个碰撞位,则读写器识别两个标签。如果存在多个碰撞位,读写器跟踪所有碰撞发生的位置,并把 CBM 对应位置‘1’。读写器识别所有未碰撞位,把 RBM 对应位置上置位已识别数码;

步骤 3 读写器发送命令 Req(CBM,1)。有效识别范围内, SF 为‘1’的标签回送 CBI 并把 SF 置‘0’;

步骤 4 读写器接收 CBI,并跟踪所有碰撞位位置,然后根据碰撞位位置译码得到 CBC。例如,若读写器接收到 8 位 CBI 为“000x0xx0”, $x$  表示该位发生碰撞。则表示碰撞发生在第 1,第 2 和第 4 位,那么读写器可以译码出 3 个 CBC,分别是“001”,“010”,“100”;

步骤 5 读写器把 CBM 中高  $m$  个碰撞标志位置‘0’。若 CBM 全部数位为零,则说明碰撞位数不高于  $m$ ,说明本次响应标签数目不大于  $2^m$  个,可根据 CBC 全部识别。若碰撞位数高于  $m$ ,读写器把 CBC 按顺序嵌入到 RBM 中,形成新的查询前缀序列。举个例子,若读写器译码得到的 CBC 为

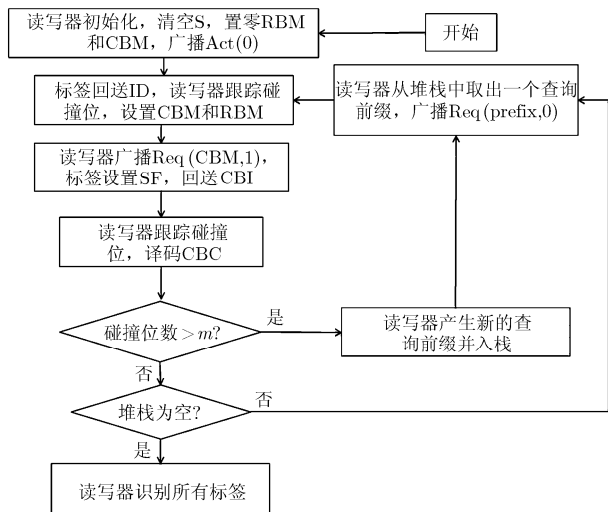


图 1 CBIA 流程图

“ $c_2c_1c_0$ ”, RBM 为 “ $r_{L-1}r_{L-2} \dots 0r_jr_{j-1} \dots 0r_kr_{k-1} \dots 0r_p r_{p-1} \dots 0r_q r_{q-1} \dots r_0$ ”, 读写器用 CBC 更新 RBM 得到新的 RBM 为 “ $r_{L-1}r_{L-2} \dots c_2r_jr_{j-1} \dots c_1r_kr_{k-1} \dots c_0r_p r_{p-1} \dots 0r_q r_{q-1} \dots r_0$ ”, 并生成新的查询前缀 “ $r_{L-1}r_{L-2} \dots c_2r_jr_{j-1} \dots c_1r_kr_{k-1} \dots c_0$ ”。最后读写器把新产生的查询前缀序列压入堆栈 S 中;

步骤 6 若 S 不空, 则读写器从 S 中取出一个查询前缀序列, 并广播 Req(prefix,0)命令。有效识别范围内, ID 高位与前缀一致的标签回送剩余 ID, 并把 SF 置 ‘1’。算法跳转到步骤 2;

步骤 7 若 S 读空, 则所有标签都被识别, 识别过程结束。

### 2.4 算法应用举例

图 2 给出了一个利用 CBIA 算法识别 7 个标签 TagA~TagG 的示例。图中 x 表示该位发生碰撞。CBC 长度  $m$  为 4。在第 1 个时隙, 读写器广播启动命令, 所有标签都响应读写器命令。在第 2 个时隙, 读写器广播 CBM “11111111111011”。SF 为 ‘1’ 的标签回送 CBI, 并置 SF 为 ‘0’。在本示例中, 标签 A, B, F 和 G 的 CBI 相同, 均为 “00000100000000”, 对应 CBC 为 “1010”; 标

签 C 的 CBC 是 “0101” 回送 CBI “0000000000100000”; 标签 D 的 CBC 是 “0001” 回送 CBI “0000000000000010”; 标签 E 的 CBC 是 “0111” 回送 CBI “0000000010000000”。读写器接收端接收到的 CBI 为 “00000x00x0x000x0”。这样读写器可以根据碰撞位发生位置计算所有的 CBC, 分别是 “1010”, “0111”, “0101”, “0001”。然后读写器就可以生产 4 个查询前缀序列 “1010”, “0111”, “0101”, “0001”, 并压栈。在第 3 个时隙, 读写器取出一个查询前缀序列 “0001”, 并广播查询命令。标签 D 高 4 位与这个前缀一致, 所以它回送剩余 ID 位, 这样标签 D 成功识别。同样标签 C 和 E 分别在第 4 个和第 5 个时隙被识别。第 6 个时隙, 读写器广播查询前缀 “1010”, 标签 A, B, F 和 G 同时回送剩余 ID 位并置各自 SF 为 ‘1’。读写器接收 CBI “0x0xx0x0”。此时碰撞位数为 4, 那么这 4 个标签可以在下个时隙一次识别。在第 7 时隙, 读写器广播 CBM “000001011010”, 标签 A, B, F 和 G 回送 CBI。读写器接收 CBI “00x00x0000x 0x000” 经译码得 CBCs: “1101”, “0101”, “1010”, “0011”, 然后把 CBC 嵌入到 RBM 中可识别这 4 个标签。此时, S 为空, 识别过程结束, 所有标签都被识别。

### 3 算法性能分析

算法性能可以通过估计识别  $N$  个标签所需要的时隙数和估算系统吞吐量进行衡量。

在数学分析之前, 有必要区分 CBIA 算法的两种碰撞:

可识别碰撞: 读写器在接收标签回送 ID 数据时, 如果检测出碰撞位数不大于 CBC 位数  $m$ , 则碰撞的多个标签可通过译码 CBC 全部识别。

不可识别碰撞: 读写器在接收标签回送 ID 数据时, 如果检测出碰撞位数大于 CBC 位数  $m$ , 则碰撞的多个标签不可在一次译码中识别, 这种碰撞为不可识别碰撞。

时隙序号	读写器命令	标签响应	读写器接收数据	时隙状态	识别标签	标签 A	标签 B	标签 C	标签 D	标签 E	标签 F	标签 G
1	Act(0)	多个 IDs	xxxx xxxx x0xx	碰撞	-	1010	1010	0101	0110	-	-	-
2	Req(11111111111011,1)	多个 CBI	00000x00x0x000x0	碰撞	-	1010	0001	0110	1111	-	-	-
3	Req(0001,0)	10101111	10101111	单个识别	标签 D	0101	1101	1101	-	-	-	-
4	Req(0101,0)	11011101	11011101	单个识别	标签 C	0111	0101	1101	-	-	-	-
5	Req(0111,0)	01011101	01011101	单个识别	标签 E	1010	0100	1100	-	-	-	-
6	Req(1010,0)	多个 IDs	00000x0xx0x0	碰撞	-	1010	0000	1110	-	-	-	-
7	Req(000001011010,1)	多个 CBI	00x00x0000x0x000	多个识别	标签 A, B, F 和 G	$m=4$						

图 2 CBIA 示例

假设在读写器可识别范围内存在  $N$  个待识别标签，每个标签具有唯一的  $L$  位 ID 和一个  $2^m$  位的碰撞位指示器 CBI。用  $N_C$  表示最初碰撞时隙中碰撞位数，显然有  $N_C < L$ 。同时标签数目  $N < 2^{N_C}$ 。标签 ID 服从均匀分布，则  $N$  个标签 ID 中，同一数位上‘0’的个数  $x$  服从二项分布

$$P(x, N) = \binom{N}{x} p^x (1-p)^{N-x} \quad (1)$$

由于每个数位等概率取得‘0’和‘1’。所以  $p=1/2$ 。当  $N$  个标签 ID 的同一位上均为‘0’或者‘1’时，该位不会发生碰撞。这样  $N$  个标签某位不发生碰撞的概率为

$$P(x=0|x=N, L, N) = P(0, N) + P(N, N) = 2^{-(N-1)} \quad (2)$$

相应地，该位发生碰撞的概率为

$$P(x \neq 0 \& x = N, L, N) = 1 - P(x=0|x=N, L, N) = 1 - 2^{-(N-1)} \quad (3)$$

进而有  $N$  个  $L$  位标签 ID 同时回送读写器时，有  $N_C$  个碰撞位的概率是

$$P_C(N_C, L, N) = \binom{L}{N_C} [P(x \neq 0 \& x = N, L, N)]^{N_C} \cdot [P(x=0|x=N, L, N)]^{L-N_C} \quad (4)$$

在一棵完整的 CBIA 查询树中，每个父节点可以产生  $2^m$  个子节点，第  $K$  层有  $2^{mK}$  子节点。每个子节点对应一个查询前缀序列。 $N$  个标签 ID 中具有相同前缀的标签 ID 数量  $j$  服从二项分布

$$P_K(j, N) = \binom{N}{j} q^j (1-q)^{N-j} \quad (5)$$

同样，由于标签 ID 是均匀分布，因此标签前缀等概率出现，从而有

$$q = 2^{-mK} \quad (6)$$

当只有一个标签具有某一前缀时，该标签可以直接识别。由式(4)，单标签被识别的概率为

$$P_{K\_SI}(1, N) = Nq(1-q)^{N-1} \quad (7)$$

当多个标签具有相同前缀，但回送数据仅有一个碰撞位时，说明当前只有两个标签响应读写器查询命令，这样两个标签可以一次识别。识别概率为

$$P_{K\_BI} = P_K(2, N)P_C(1, N_C - Km, 2) \quad (8)$$

当多个标签具有相同前缀，但回送数据碰撞位数  $n_C$  不大于  $m$  时，通过下一个时隙 CBC 译码，这些标签都可以识别出来，这种情况发生的概率为

$$P_{K\_MI} = \sum_{n_C=2}^m \sum_{j=3}^{2^{n_C}} P_K(j, N)P_C(n_C, N_C - Km, j) \quad (9)$$

当多个标签具有相同前缀，且回送数据碰撞位数  $n_C$  大于  $m$  时，这些标签不可能在写一个时隙或者下一次 CBC 译码识别。在 CBIA 树的第  $K$  层，回送数据碰撞位的最大值为  $N_C - Km$ 。因此这种情况发生的概率为

$$P_{K\_C} = \sum_{n_C=m+1}^{N_C-Km} \sum_{j=3}^{2^{n_C}} P_K(j, N)P_C(n_C, N_C - Km, j) \quad (10)$$

如果用  $P_{Ki/N}$  表示 CBIA 查询树上第  $K$  层的第  $i$  个节点被访问的概率。显然当  $K=0$ ,  $P_{00/N}=1$ ，也就是 CBIA 的根节点总是被访问到。

每个子节点被访问到的前提是其父节点发生了不可识别碰撞。假设第  $K$  层的第  $i$  个节点发生不可识别碰撞的概率为  $\beta_{Ki/N}$ 。由于同一层上，各节点发生不可识别碰撞的概率相同，因此

$$\beta_{Ki/N} = \beta_{K/N} = P_{K\_C} \quad (11)$$

进而有

$$P_{Ki/N} = P_{K/N} = \begin{cases} 1, & K=0 \\ \beta_{K-1/N}, & K>0 \end{cases} \quad (12)$$

这样，CBIA 算法访问节点的数学期望为

$$\hat{t}_{\text{Node}} = \sum_{K=0}^{\infty} \sum_{i=0}^{2^{mK}-1} P_{Ki/N} = 1 + \sum_{K=1}^{\infty} 2^{mK} P_{K/N} = 1 + 2^m \sum_{K=1}^{\infty} 2^{mK} \beta_{K/N} \quad (13)$$

综上，有 CBIA 识别  $N$  个标签的时隙树为

$$\hat{t}_{\text{Node}} = 1 + 2^m \sum_{K=1}^{\infty} \sum_{n_C=m+1}^{N_C-Km} \sum_{j=3}^{2^{n_C}} 2^{Km} P_K(j, N) \cdot P_C(n_C, N_C - Km, j) \quad (14)$$

进而可得系统的吞吐率为

$$s = \frac{N}{\hat{t}_{\text{Node}}} = \frac{N}{1 + 2^m \sum_{K=1}^{\infty} \sum_{n_C=m+1}^{N_C-Km} \sum_{j=3}^{2^{n_C}} 2^{Km} P_K(j, N) P_C(n_C, N_C - Km, j)} \quad (15)$$

需要说明的是，在实际识别过程中，CBIA 查询树的层数不可能无穷大，它与系统设置的标签 CBC 位数  $m$  有关。

#### 4 实验仿真分析

设标签的长度是 64 bit，标签 ID 服从均匀分布，标签数目分别为 10, 100, 200, ..., 1000。每设置 1 次标签数目，实验重复 100 次，平均时隙为 100 次取平均。以下是对 CBIA, OQTT 和 CTTS 的平均识别时隙、吞吐率两个指标进行比较仿真，仿真中，标签碰撞位编码 CBC 的位数  $m=6$ 。仿真结果如图 3 所示。

仿真结果表明：相比于CTTA, OQTT算法, CBIA算法平均识别时隙数优势明显,且随着待识别标签数据的增大,算法性能优势越明显。在系统吞吐率方面, CBIA算法性能优于其他两种算法,平均每个时隙可识别0.7个标签,优于OQTT算法的吞吐率仅为0.6和CTTA的0.5。

图4反应了CBC位数和标签ID位数对CBIA算法性能的影响。图4(a)中,标签ID位数固定为64位,标签CBC位数分别为2,4,6,8,10位。图4(b)中,标签数量固定为1000个,标签ID位数分别为32,48,64,96和128位。仿真结果表明,当CBC位数为6时,识别相同数量的标签所用时隙最少。

## 5 算法的性能对比优势和标签硬件开销分析

本节主要分析CBIA算法与CTTA, OQTT的对比优势,并分析CBIA获取这些优势所付出的硬件成本。

### 5.1 算法的性能对比优势

CTTA, OQTT和CBIA都是查询树算法QT的改进,其核心思想是不断地对待识别标签进行分组,直到组内标签可以一次识别。这几种算法性能差异,是由各自采取的分组协议引起的。

CTTA每次只处理一个碰撞位。每次检测到碰撞位,CTTA自动把待识别标签分成碰撞位为‘0’和‘1’的两组,直到组内仅含一个标签或者仅含有

两个只有一个冲突位的标签,识别这些标签。CTTA采用确定性分组,避免了空闲时隙的产生。然而,CTTA每次只把组内待识别标签划分成两个小组。这样,在标签数量较大时,CTTA搜索深度会很深。而且,在识别初期,每一组内待识别标签数量很大,碰撞概率很高,在标签数量较大时,碰撞时隙数量非常可观,识别效率不高。

OQTT对QT算法的改进仅在于识别最初的最优化分组,而在之后的各组内识别过程仍旧采用CTTA算法。尽管采用了碰撞位跟踪技术,但每次也仅处理一个碰撞位。这一方面,CTTA算法的不足,在OQTT算法中仍然存在。另一方面,OQTT的最优化分组是随机分组。当标签ID不服从均匀分布时,OQTT的最初分组会不平衡<sup>[9]</sup>,从而使得某些组内待识别标签非常多,而某些组内待识别标签很少,甚至没有。其结果是OQTT算法性能下降。

CBIA算法与CTTA算法比较,其最大特点在于一次处理 $m$ 位碰撞位。CBIA根据CBI译码结果,把待识别标签分成 $K(K \geq 2)$ 组。当组内所含标签ID碰撞位不大于 $m$ ,CBIA可以识别组内所有标签。CBIA每次分组数量不小于2,因此,识别相同数量的标签,CBIA的搜索深度是CTTA的 $1/m$ 。同时,CBIA每次分组数量大于CTTA,因此,在同一搜索深度,CBIA组内待识别标签数量少于CTTA。进而减小了标签ID碰撞概率。这些,使得CBIA算法性能优于CTTA。

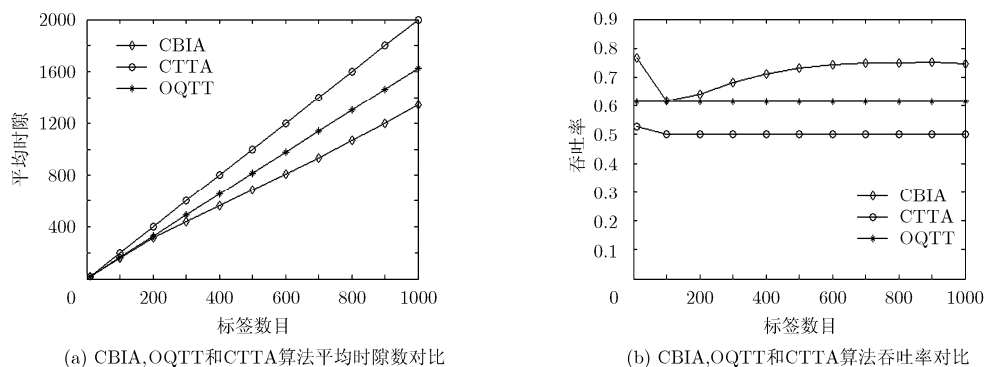


图3 算法性能比较

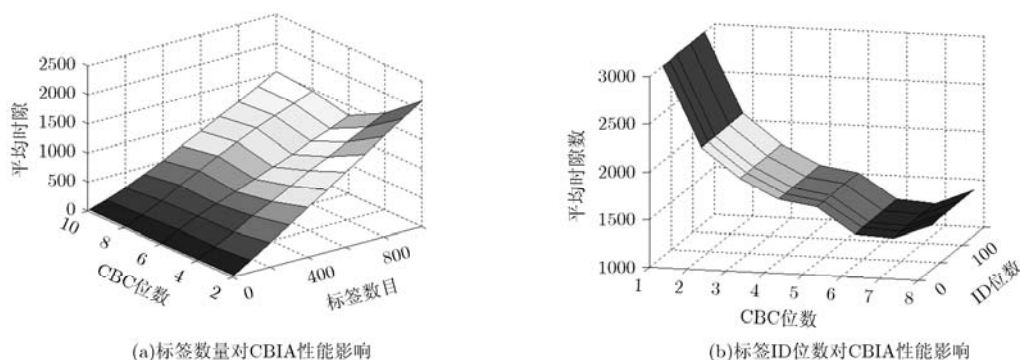


图4 CBC位数和ID位数对CBIA性能影响仿真

CBIA 算法与 OQTT 算法比较, 其最大特点是确定性分组。与 OQTT 的随机分组不同, CBIA 算法在最初分组时采用确定性分组, 每个小组内都有待识别标签, 不会出现无标签的小组。因此, CBIA 可以避免 OQTT 算法在最初分组时的空闲时隙的产生。而之后的识别过程, OQTT 组内采用 CTTA 算法, 这样, CBIA 算法对 CTTA 算法的优势也同样体现在对 OQTT 算法的比较中。

## 5.2 标签硬件开销分析

CTTA 是采用了位跟踪技术的 QT 算法。QT 算法是一种无记忆防碰撞协议, 标签在识别过程中不需要存储任何数据, 每次回送剩余标签 ID。因此, CTTA 算法标签中无存储资源, 硬件成本低廉。

OQTT 是对 CTTA 算法的改进, 继承了 CTTA 算法的无记忆的特点。但是, OQTT 算法在最初的进行标签数量估计时, 需要标签根据协议产生一个  $\log_2 L$  位的随机数, 并根据这个随机数对一个  $L$  位的二进制数(其中  $L$  是标签 ID 的位数)进行位调制<sup>[8]</sup>。因此, OQTT 算法在硬件开销方面, 将增加随机数生成模块和位调制模块。

CBIA 算法也是 CTTA 算法的改进。根据协议需要, 标签接收读写器广播的 CBM, 并根据 CBM 的非零位位置和标签 ID 的对应位产生一个  $m$  二进制数, 并用这个二进制数对  $2^m$  位 CBI 进行位调制。因此, CBIA 算法与 OQTT 类似, 都增加了二进制数生成和位调制的硬件开销。但是, 由于 CBC 位数小于  $\log_2 L$ , CBI 位数小于  $L$ , 因此, 与 OQTT 比较, CBIA 算法所消耗的硬件资源更少。考虑到标签需把调制后的二进制数回送读卡器, OQTT 算法回送一个  $L$  位的二进制数, 而 CBIA 算法仅回送一个  $2^m$  位的 CBI, 因此, CBIA 比 OQTT 的通信数据量更少。

## 6 结束语

本文提出了一种新颖的 RFID 多标签防碰撞算法, CBIA。该算法利用碰撞位跟踪技术, 对待识别标签进行重复分组, 直到所有标签都被正确识别。算法增加了查询树的每一层分叉数目, 同时避免了空闲时隙的产生, 压缩了查询树的层数, 减少了碰撞时隙的数量。仿真结果表明, 采用 CBIA 算法的多标签识别系统, 吞吐率可以达到每时隙 0.7 个标签, CBIA 算法识别效率优于 OQTT 和 CTTA 算法。最后, CBIA 和 OQTT 都增加了一些硬件开销, 但是 CBIA 硬件开销低于 OQTT。

## 参考文献

[1] Zuo Y J. Survivable RFID systems: issues, challenges and techniques[J]. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 2010, 40(4): 406-418.

- [2] Mohamed B, Adel M, and Belkacem F. Dual antenna for physical layer UHF RFID collision cancelling[C]. 2012 International Conference on Multimedia Computing and Systems, Melbourne, Australia, 2012: 623-628.
- [3] Lee C C and Lin S Y. A double blocking dynamic framed slotted ALOHA anti-collision method for mobile RFID systems[C]. 2012 Sixth International Conference on Genetic and Evolutionary Computing, Kyushu, Japan, 2012: 581-584.
- [4] Jiang Y J, Xu Y F, and Wang Q. Cancellation strategy in dynamic framed slotted ALOHA for RFID system[C]. 2013 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, 2013: 854-859.
- [5] Wang S, Hong W J, and Li S F. A slot-wise LMMSE estimate algorithm for frame slotted aloha protocol of RFID system[C]. 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai, China, 2012: 1-5.
- [6] Xue J B, Wang W H, Li S B, et al. Anti-collision algorithm based on counting mechanism and multi-state binary[C]. 2013 Fifth Conference on Measuring Technology and Mechatronics Automation, Hong Kong, China, 2013: 276-282.
- [7] Landaluce H, Perallos A, and Zuazola I J G. A fast RFID identification protocol with low tag complexity[J]. *IEEE Communications Letters*, 2013, 17(9): 1704-1706.
- [8] Lai Y C, Hsiao Y L, Chen H J, et al. A novel query tree protocol with bit tracking in RFID tag identification[J]. *IEEE Transactions on Mobile Computing*, 2012, 12(10): 2063-2075.
- [9] Yang Y K, Cui C S, Zhou T F, et al. Improvement on RFID-based binary anti-collision algorithm[C]. 2012 International Conference on Computer Science and Service System, Nanjing, China, 2012: 515-518.
- [10] Jin D, Ma Y M, Fan Z P, et al. A RFID anti-collision algorithm based on multithread regressive-style binary system[C]. 2012 International Conference on Measurement, Information and Control, Harbin, China, 2012: 365-369.
- [11] Liang C K and Lin H M. Using dynamic slots collision tracking tree technique towards an efficient tag anti-collision algorithm in RFID systems[C]. 2012 9th International Conference on Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing, Fukuoka, Japan, 2012: 272-277.
- [12] Bai Y, Xuan X W, Teng J F, et al. An anti-collision algorithm based on collision bit position and splitting[C]. 2010 6th International Conference on Wireless Communications Networking and Mobile Computing, Chengdu, China, 2010: 1-4.
- [13] Piao C H, Fan Z J, Yang C Y, et al. Research on group-based polling anti-collision algorithm for RFID tag identification[C]. 2010 International Forum on Information Technology and Applications. Kunming, China, 2010: 185-188.

李志坚: 男, 1981年生, 讲师, 研究方向为物联网 RFID 关键技术与应用、雷达信号处理。

赖顺桥: 男, 1981年生, 工程师, 研究方向为物联网 RFID 关键技术与应用。