

基于连续碰撞位探测的防碰撞算法研究

刘子龙, 纪金水, 刘彩虹, 齐爱琴, 张 炜, 陈秀萍

(西北民族大学数学与计算机科学学院, 甘肃兰州 730100)

摘 要: 文章通过对 IAMS 算法的分析, 提出了一种基于连续碰撞位探测 (CCBD) 的防碰撞算法. 在 CCBD 算法中引入连续碰撞位探测机制, 并把标签碰撞分为单独位碰撞和连续位碰撞两种情况进行处理. 连续碰撞位探测机制能够准确获得连续碰撞位的实际存在情况从而避免了空闲时隙以及不必要的碰撞时隙的产生. 理论分析和实验仿真表明, CCBD 算法克服了 IAMS 算法的不足, 在标签识别过程中表现出良好的性能.

关键词: 射频识别; 防碰撞算法; 连续碰撞位探测机制

中图分类号: TP399 **文献标识码:** A **文章编号:** 0372-2112 (2013) 11-2156-05

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2013.11.007

An Anti-Collision Algorithm Based on Continuous Collision Bit Detection

LIU Zi-long, JI Jin-shui, LIU Cai-hong, QI Ai-qin, ZHANG Wei, CHEN Xiu-ping

(Institute of Mathematics and Computer Science, Northwest University for Nationalities, Lanzhou, Gansu 730100, China)

Abstract: Through the analysis of IAMS algorithm, this paper proposed a zero idle timeslot anti-collision algorithm which is based on continuous collision bit detection (CCBD). CCBD algorithm introduced continuous collision bit detection mechanism, and divided tag collision into individual bit collision and continuous bits collision to process. CCBD algorithm can obtain the actual existence of the continuous collision bits by using continuous collision bit detection mechanism, thereby avoiding the generation of idle timeslots and unnecessary collision timeslots. The theoretical analysis and experimental simulation show that the CCBD algorithm has overcome the shortage of IAMS algorithm and has a good performance during RFID process.

Key words: radio frequency identification (RFID); anti-collision algorithms; continuous collision bit detection mechanism

1 引言

射频识别通常也称为 RFID (Radio Frequency Identification)^[1], 它借助基于电磁波的非接触式双向通信实现自动识别目标物体. RFID 技术广泛的被应用于畜牧业管理、门禁管理、图书管理、供应链管理和物流管理中. 射频识别系统的物理组成包括射频标签和阅读器两部分^[2]. 在 RFID 技术中存在着标签碰撞的问题, 即存在于阅读器作用范围内的多个标签同时发送各自的识别码给阅读器, 使阅读器因为接收到数据的相互重叠而无法正确识别所有标签的问题. 常见的标签防碰撞算法可分为基于 ALOHA 的防碰撞算法和基于树的防碰撞算法^[3], 后者又可分为 QT protocol 和 Tree algorithm^[3,4].

1.1 基本查询树算法 (QT protocol)

Dong-Her Shih 等阐述了 QT protocol 算法的原理, 其实现步骤如下^[4]:

- ①按照曼彻斯特码的编码规则形成最初的两个询问前缀 1 和 0 并依次压入栈, 形成询问栈 $S = (1, 0)$;
- ②将栈顶元素 $x \cdots x$ 出栈;
- ③阅读器发送②中出栈的询问前缀 $x \cdots x$, 其作用范围内的标签接收到后对比询问前缀与自身识别码, 两者相匹配的标签会做出响应. 若只有 1 个标签响应, 则识别该标签; 若有多于 1 个标签响应, 在当前询问前缀后面加 1 和 0, 形成新的询问前缀 $x \cdots x1$ 和 $x \cdots x0$ 并依次压入栈 S ; 若无有标签响应, 则什么都不做;
- ④若栈 S 为空, 算法结束; 否则跳至步骤②.

1.2 基于查询树算法的最近研究成果

在射频标签的识别码长度较长的时候, QT protocol 会产生大量碰撞周期; 无论树的节点是否有碰撞发生, QT protocol 都会遍历整个二进制树; 这些预示着 QT protocol 的效率不高. 为此, 人们对 QT protocol 不断改进, 以提高其性能. 丁治国先生在论文《自适应多叉树防碰撞

算法研究》中引入了碰撞因子 μ , 定义 μ 为^[5]:

$$\mu = \frac{n_c}{n} \quad (1)$$

其中, n_c 代表在阅读器的一次询问中收到的标签叠加反馈信息中发生碰撞的比特位数; n 代表标签的识别码长度; 张学军先生在《改进型自适应二叉树防碰撞算法研究》中提出了 IAMS 算法, IAMS 在处理碰撞之前要先计算 μ 的值, 根据 μ 的值, IAMS 处理碰撞有两种方法^[1]: ① $\mu < 0.75$, 采用二叉树分裂, 阅读器将处于高位的碰撞位分别赋值为 0 和 1, 生成两个新的询问前缀; ② $\mu \geq 0.75$, 采用四叉树分裂, 阅读器会探测发生碰撞的高两位的具体数值, 根据探测结果生成新的询问前缀. 下面说明一下 IAMS 的识别过程, 假设阅读器当前作用范围内有 6 个标签, 标签的识别码分别为: a : 11011001, b : 11011101, c : 11011010, d : 10111100, e : 10111110, f : 10111101; 存放询问前缀的栈 S 初始状态为空; 识别过程如表 1 所示.

表 1 IAMS 算法的识别过程

时隙	询问前缀	标签反馈	时隙状态	μ	2/4 叉树
1	ϵ	1xx11xxx	碰撞	0.625	2
2	11	11011xxx	碰撞	0.375	2
3	110111	11011101	成功识别 b		
4	110110	110110xx	碰撞	0.25	2
5	1101101	11011010	成功识别 c		
6	1101100	11011001	成功识别 a		
7	10	101111xx	碰撞	0.25	2
8	1011111	10111110	成功识别 e		
9	1011110	1011110x	碰撞	0.125	2
10	10111101	10111101	成功识别 f		
11	10111100	10111100	成功识别 d		
12	栈 S 为空, IAMS 算法结束, 识别过程结束				

从表中可以看到, IAMS 算法完成 6 个标签的识别一共用了 11 个时隙. 值得注意的是, 在整个识别过程中遇到有碰撞发生的时候, IAMS 算法都选择了二叉树分裂, 因此这个过程其实就是跳过没有发生碰撞的节点并且只在存在碰撞的节点遍历的 QT protocol; 整个识别过程中出现了约占总时隙数 45.5% 的碰撞时隙, 这使得在此类情况下 IAMS 算法的效率并不高, 它并没有按照最初预想的那样通过采用四叉树分裂减少碰撞时隙来提高 RFID 系统的识别效率. 分析其原因如下: 张学军先生把碰撞因子 $\mu = 0.75$ 做为 IAMS 算法处理碰撞时选择二叉树分裂还是四叉树分裂的判断标准; 假设式(1)中 n 取 8, 则当 $n_c \geq 6$ 的时候才会使用四叉树分裂, 在 $n_c < 6$ 的时候都使用二叉树分裂, 这大大减少了

IAMS 算法利用四叉树减少碰撞时隙的机会. 实际上, 在能够证明连续碰撞位的实际存在值的情况下, 可以尽量多的采用多叉树分裂来提高防碰撞算法的性能^[6].

鉴于 IAMS 算法存在的不足, 本文提出了一种基于连续碰撞位探测的防碰撞算法 (Continuous Collision Bit Detection Anti-Collision Algorithm, CCBDA), 该算法通过准确探测连续碰撞位的存在情况, 在把空闲时隙数减少到 0 的同时大幅减少了碰撞时隙数, 有效的提高了 RFID 系统的识别效率.

2 基于连续碰撞位探测的防碰撞算法

在介绍 CCBDA 算法前, 先定义以下概念.

探测命令: 用于探测连续碰撞位实际存在情况的二进制串. 探测命令由阅读器生成并发送, 它有 3 部分组成: 启动探测模式的信号 + 长度为标签 ID 的最高位序号减去连续碰撞位的最高位序号的全 0 序列 + 长度等于连续碰撞位位数的全 1 序列; 探测命令的第 1 部分是长度等于标签 ID 长度的全 1 序列 (在标签编码时候将全 1 的开头定义为启动探测模式的信号, 用于通知标签进入探测模式工作); 如果碰撞从标签 ID 的最高位开始则探测命令的第 2 部分轮空; 探测命令从第 2 部分开始 (第 2 部分轮空时从第 3 部分开始) 的高低位设置与标签 ID 的一致. 考虑到标签的计算能力, 把探测命令的第 3 部分的最大长度限制为 6, 若标签叠加反馈中的连续位碰撞位数大于 6, 则这一处连续位碰撞分作多次处理.

连续碰撞位探测机制: 是在连续位碰撞发生时用于探明连续碰撞位实际存在情况的策略. 标签在收到阅读器发出的命令并对其进行判断, 确定是探测命令后要作 3 项工作 (假设要探测的连续碰撞位位数是 N , $N \leq 6$): ① 去掉探测命令的第 1 部分, 只保留其后的内容, 得到新探测命令; ② 从自身 ID 最高位开始截取与新探测命令等长的二进制串并将其与新探测命令按照从高位到低位的顺序依次进行按位“与”; 再将按位“与”得到的二进制串转化为十进制数 M , 并按照高位在前低位在后的顺序生成长度为 2^N (N 等于探测命令中第 3 部分的长度) 的二进制串, 将该串中位序等于 M 的那一位位置 1, 其他位置 0; ③ 将第 ② 步中生成的二进制串反馈给阅读器. 在连续碰撞位探测机制中, 阅读器要完成 2 项工作: ① 生成并发送探测命令; ② 对各标签在其第 3 项工作中反馈的叠加信息进行解析, 即将其中各碰撞位的位序值分别转换为长度为 N 的二进制串; 生成新的询问前缀 (将解析结果拼接在探测命令中第 2 部分所对应标签 ID 的后面).

探测模式: 在探测模式下, 阅读器和标签分别执行连续碰撞位探测机制中的操作. 与探测模式相对应的

是询问模式,阅读器利用询问模式来正常询问标签.

探测时隙:在探测模式下工作的时隙.与探测时隙相对应的是询问时隙.

下面对连续碰撞位探测机制的使用进行举例.假设有两个识别码长度为 4 的标签 1001 和 0111,询问模式下阅读器收到的反馈为 xxx1,则探测过程如下:

①阅读器对 xxx1 进行检测,发现存在 1 处长度为 3 的连续位碰撞,于是阅读器生成探测命令 1111111 并向其作用范围内的标签广播;

②两标签收到命令并判断它是以 1 开头且有 4 个连续的 1,确认阅读器要求进入探测模式工作,于是两标签分别对 1111111 进行处理后得到新探测命令 111;

③两标签从自身识别码截取 100 和 011;分别与 111 进行按位与得到 100 和 011;将按位与的结果转换为十进制数 4 和 3;根据 4 和 3 生成长度为 2^3 的二进制串 00010000 和 00001000 并反馈给阅读器;

④阅读器收到的叠加反馈是 000xx000;对此信息进行解析后,阅读器就可以知道连续碰撞位的实际值是长度为 3 且对应的十进制数是 4 和 3 的两个二进制串,据此生成新的询问前缀 100 和 011;探测模式结束.

CCBD 算法的流程如图 1 所示.

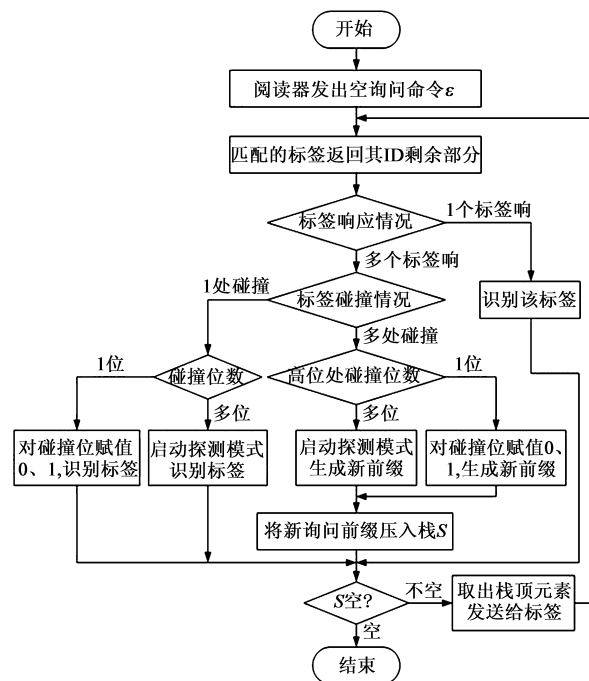


图1 CCBD算法的流程图

CCBD 算法的基本思想:用询问前缀栈来存储询问前缀,若栈为空且没有已经出栈但还未完成询问的前缀,识别过程结束;系统有询问模式和探测模式两种工作模式;询问模式下阅读器进行正常的标签询问,与询问前缀相匹配的标签返回自身识别码除询问前缀后的

剩余部分;探测模式用于处理标签反馈中的连续位碰撞,生成新的询问前缀并压入询问前缀栈;若碰撞位只有 1 位则阅读器将该位分别置为 0 和 1,生成两个新的询问前缀并压入询问前缀栈;若标签的叠加反馈中碰撞只有 1 处(单独位或连续位),则在将该单独位分别置为 0、1 或连续位用探测得到的实际值替代后直接识别标签.

对表 1 中的 6 个标签采用 CCBD 算法的识别过程如表 2 所示.

表 2 CCBD 算法的识别过程

时隙	询问命令	标签反馈	时隙模式	备注
1	ϵ	1xx11xxx	询问	多处连续位碰撞
2	11111111011	0xx0	探测	探测模式工作
3	110	11xxx	询问	1 处连续位碰撞
4	1111111100000111	00x00xx0	探测	探测模式识别 a、b、c
5	101	111xx	询问	1 处连续位碰撞
6	111111110000011	0xxx	探测	探测模式识别 d、e、f
7	栈 S 为空,CCBD 算法结束,识别过程结束			

3 算法对比以及性能分析

表 1 和表 2 所对应的询问树分别如图 2 和图 3 所示,图中的矩形代表碰撞时隙,圆角矩形代表成功时隙;平行四边形代表探测时隙;四边形中的内容代表询问命令.

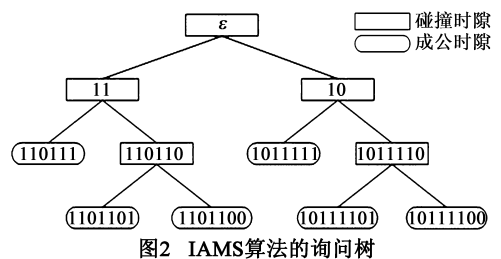


图2 IAMS算法的询问树

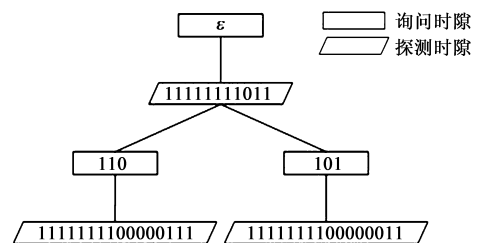


图3 CCBD算法的询问树

对比图 2 和图 3 可以看出,CCBD 算法大大的减少了阅读器的询问次数.下面通过计算总时隙数和吞吐率来分析 CCBD 算法的性能.假设系统中存在 n 个长度为 m 的待识别标签,因为在 CCBD 算法中避免了空闲

时隙出现,所以该算法的询问树一定满足:第 0 层只有 1 个结点,其余各层的节点数最多不会超过 n . 据此,算法的满询问树是这样的:根节点只有 1 个,其余 i 层的节点均为 n 个. 对于满询问树的任意非根节点层, n 个未读标签中有 k 个选择同一节点响应的概率服从二项分布^[7]:

$$P(k/n) = C_n^k p^k (1-p)^{n-k} \quad (2)$$

因为除根节点外的任意层都有 n 个节点,所以上式中 $p = n^{-1}$. 设算法识别 n 个标签所需的总时隙为 $T(n)$,分析算法的原理和图 3 可知:

$$T(n) = \text{根节点}(1) + \text{碰撞节点}(t_1) + \text{探测节点}(t_2) + \text{中间的成功节点}(t_3) \quad (3)$$

式(3)中的各部分计算如下:

碰撞节点个数: 等于所有发生碰撞的节点的个数,也即有一个以上的标签响应的节点的个数.

空闲节点的概率:

$$P(0/n) = C_n^0 p^0 (1-p)^n = (1-n^{-1})^n \quad (4)$$

成功节点的概率:

$$P(1/n) = C_n^1 p^1 (1-p)^{n-1} = (1-n^{-1})^{n-1} \quad (5)$$

碰撞节点的概率:

$$P(k > 1/n) = 1 - P(0/n) - P(1/n) = 1 - (1-n^{-1})^n - (1-n^{-1})^{n-1} \quad (6)$$

$$P(2/n) = C_n^2 p^2 (1-p)^{n-2} = 2^{-1} (1-n^{-1})^{n-1} \quad (7)$$

$$P(k > 2/n) = 1 - P(0/n) - P(1/n) - P(2/n) = 1 - (1-n^{-1})^n - 1.5(1-n^{-1})^{n-1} \quad (8)$$

可以得到:

$$t_1 = i * n * P(k > 1/n) = in[1 - (1-n^{-1})^n - (1-n^{-1})^{n-1}] \quad (9)$$

探测节点个数: 等于发生连续位碰撞的时隙的个数. 因为 $P(k > 1/n) = P(2/n) + P(k > 2/n)$, 而 $P(k > 2/n)$ 中 $k > 2$ 意味着发生的碰撞是碰撞位数大于或等于 2 的连续位碰撞(若只有 1 位发生碰撞,则响应的标签有且只有两个); $k = 2$ 时发生的碰撞可能是单独位碰撞,也可能是碰撞位数大于或等于 2 的连续位碰撞,因此只需从 $P(k > 1/n)$ 中减去单独位碰撞的发生概率,即可得到探测节点发生的概率. 两个标签对同一个节点响应的时候,标签 ID 只有一位发生碰撞的概率计算如下:

标签各位的取值相互独立并且符合二项分布,所以两个标签中任意一位不发生碰撞的概率:

$$P_1 = 2C_2^0 (2^{-1})^0 (2^{-1})^2 = (2^{-1})^{2-1} = 2^{-1} \quad (10)$$

两个标签中任意一位发生碰撞的概率:

$$P_2 = 1 - P_1 = 1 - 2^{-1} = 2^{-1} \quad (11)$$

两个标签中只有一位发生碰撞的概率:

$$P_3 = C_m^1 (P_2)^1 (P_1)^{m-1} = C_m^1 (2^{-1})^1 (2^{-1})^{m-1} = m2^{-m} \quad (12)$$

可以得到:

$$t_2 = i * n * [P(k > 1/n) - P_3] = in[1 - (1-n^{-1})^n - (1-n^{-1})^{n-1} - m2^{-m}] \quad (13)$$

中间的成功节点个数: 这是指算法的满询问树中除最后一层外的其余各层中的成功时隙数,因为在碰撞只有一处的情况下算法通过探测或对单独碰撞位的赋值直接识别标签而不再询问,所以最后一层的成功时隙不应计算在总时隙内.

可以得到:

$$t_3 = (i-1)nP(1/n) = n(i-1)(1-n^{-1})^{n-1} \quad (14)$$

所以由上面计算得到:

$$\begin{aligned} T(n) &= 1 + t_1 + t_2 + t_3 \\ &= 1 + in[1 - (1-n^{-1})^n - (1-n^{-1})^{n-1}] \\ &\quad + in[1 - (1-n^{-1})^n - (1-n^{-1})^{n-1} - m2^{-m}] \\ &\quad + n(i-1)(1-n^{-1})^{n-1} \\ &= 1 + 2in - 2in(1-n^{-1})^n \\ &\quad - (in+n)(1-n^{-1})^{n-1} - inm2^{-m} \end{aligned} \quad (15)$$

在满询问树中共有 n 个成功时隙,所以有:

$$i * n * P(1/n) = n \quad (16)$$

由式(16)可以得到询问树的总层数:

$$i = [P(1/n)]^{-1} = [(1-n^{-1})^{n-1}]^{-1} = (1-n^{-1})^{1-n} \quad (17)$$

将式(17)代入式(15)并化简得到:

$$\begin{aligned} T(n) &= 3 - 3n - n(1-n^{-1})^{n-1} \\ &\quad + n(2 - m2^{-m})(1-n^{-1})^{1-n} \end{aligned} \quad (18)$$

所以吞吐率计算为:

$$S = \frac{n}{T(n)} \quad (19)$$

4 算法的实验仿真和分析

从实验仿真的角度对 CCBD 算法进行验证,基于数量从 0 到 1000 的 20 组标签分别对 CCBD 算法和 IASM 算法的识别过程进行了模拟;每组标签的识别总时隙数取 50 次仿真的平均值. 仿真结果如下页图 4 和图 5 所示.

图 4 中,当标签数量取(0, 100)的时候,CCBD 算法所耗费的总时隙数略少于 IASM 算法;随着标签数量的增加,CCBD 算法连续碰撞位探测机制的优势也愈加明显,其所耗费的总时隙越来越明显的少于 IASM 算法;在标签数量取到 1000 的时候 CCBD 算法仅耗费约 1150 个时隙,而 IASM 算法却耗费接近 2000 个时隙.

图 5 中,当标签数量取 100 到 150 时,CCBD 算法的吞吐率约为 0.63;随着标签数量增加,CCBD 算法的高吞吐率得到体现,当测试标签数量增加到 1000 的时候,

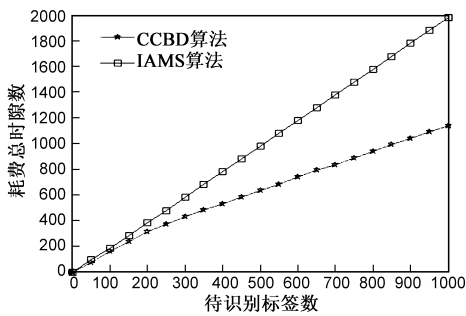


图4 CCBD算法和IAMS算法耗费总时隙数的比较

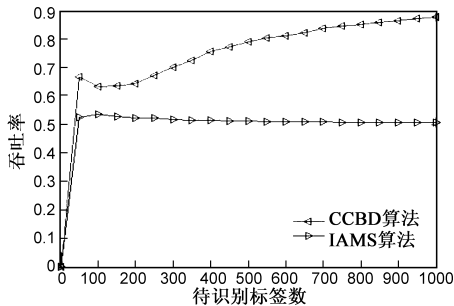


图5 CCBD算法和IAMS算法吞吐率的比较

CCBD算法的吞吐量接近0.9;而IASM算法的吞吐量整体较为稳定的保持在(0.5,0.6)之间.总的看来,在同等条件下,CCBD算法较IASM算法所耗费的总时隙数较少,吞吐量较高.

5 结语

本文提出了一种基于连续碰撞位探测的RFID防碰撞算法—CCBD算法.在CCBD算法中引入了连续碰撞位探测机制,阅读器通过在探测模式下发送探测命令来获取连续碰撞位的实际存在情况,据此生成新的询问前缀,这就避免了空闲时隙以及不必要碰撞时隙的产生.CCBD算法采用不同的方法处理单独位碰撞和连续位碰撞;对于只有一处碰撞(单独或连续)的情况,CCBD算法则采用更为节省时隙的处理办法.理论分析和仿真表明,CCBD算法克服了IASM算法因为采用碰撞因子而减少了二叉树使用频率的问题,在连续碰撞位探测机制的作用下较IASM算法有较好的性能表现.

参考文献

[1] 张学军,蔡文琦,王锁萍.改进型自适应二叉树防碰撞算法研究[J].电子学报,2012,40(1):193-198.

Zhang Xue-Jun, Cai Wen-qi, Wang Suo-ping. One anti-collision

algorithm based on improved adaptive multi-tree search[J]. Acta Electronica Sinica, 2012, 40(1): 193-198. (in Chinese)

[2] Pupunwiwat P, Stantic B. Unified Q-ary tree for RFID tag anti-collision resolution[A]. Proceedings of the 20th Australasian Database Conference (ADC) [C]. Australia: Australian Computer Society Inc, 2009. 49-58.

[3] Ryu J, Lee H, Seok Y, et al. A hybrid query tree protocol for tag collision arbitration in RFID systems[A]. Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing [C]. Glasgow: IEEE, 2007. 5981-5986.

[4] Shih D H, Sun P L, Yen D C, et al. Taxonomy and survey of RFID anti-collision protocols[J]. Computer Communications, 2006, 29(11): 2153-2156.

[5] 丁治国,朱学永,郭立,古今.自适应二叉树防碰撞算法研究[J].自动化学报,2010,36(2):237-241.

Ding Zhi-Guo, Zhu Xue-Yong, Guo Li, Gu Jin. An adaptive anti-collision algorithm based on multi-tree search[J]. Acta Automatica Sinica, 2010, 36(2): 237-241. (in Chinese)

[6] Kim Y, Kim S, Lee S, et al. Improved 4-ary query tree algorithm for anti-collision in RFID system[A]. International Conference on Advanced Information Networking and Applications [C]. Bradford: IEEE Computer Society, 2009. 699-704.

[7] D R Hush, C Wood. Analysis of tree algorithms for RFID arbitration[A]. Proceedings of IEEE Symposium on Information Theory [C]. Cambridge, MA, USA, 1998. 107-116.

作者简介



刘子龙 男,1984年2月出生于河南原阳,2008年、2011年分别在河南工业大学、西南交通大学获得工学学士学位、工学硕士学位.现为西北民族大学教师,助教.主要研究方向为嵌入式系统、物联网技术、无线射频识别技术.

E-mail: 163lzl163@163.com



纪金水 男,1965年出生于河北衡水.1985年在西北工业大学获得工学学士学位,现为西北民族大学教授、硕士生导师.主要研究方向为嵌入式系统设计与应用、无线传感器网络及物联网技术应用.