

# 基于分组处理的 RFID 标签防碰撞算法<sup>①</sup>

谈诗文<sup>1</sup>, 付桂英<sup>1</sup>, 彭 勇<sup>1</sup>, 张建国<sup>2</sup>, 王晓涧<sup>2</sup>

<sup>1</sup>(江南大学 物联网工程学院, 无锡 214122)

<sup>2</sup>(无锡市交通局航道管理处, 无锡 214122)

**摘 要:** 由于现有的确定性 RFID 标签防碰撞算法存在识别效率不高, 数据交换量大等问题, 提出了一种引入部分响应机制的分组式 RFID 标签防碰撞算法. 算法采用分组策略与部分响应相结合的方式, 读写器按一定次序依次识别每个分组内的标签, 减少了标签碰撞概率和标签识别数量; 对每个分组内的标签采取部分响应机制进行识别, 能有效减少数据通信量. 仿真结果表明, 该算法相比其他几种算法, 具有识别效率高、数据交换量小等优势.

**关键词:** RFID; 防碰撞; 标签识别

## Group-Based Anti-Collision Algorithm for RFID Tag

TAN Shi-Wen<sup>1</sup>, FU Gui-Ying<sup>1</sup>, PENG Yong<sup>1</sup>, ZHANG Jian-Guo<sup>2</sup>, WANG Xiao-Jian<sup>2</sup>

<sup>1</sup>(School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China)

<sup>2</sup>(Water Route Management Office, Wuxi Transportation Bureau, Wuxi 214122, China)

**Abstract:** To solve the problem of the inefficient identification and the large amount of data transmission, an enhanced group-based anti-collision algorithm based on partial responses for RFID tag was proposed. The algorithm used the grouping strategy and partial response mode combines. The tags in each group were identified by reader in turn, it could reduce the probability of collision and the number of the identified tags. The partial response mechanism was used to identify all tags in each group, it reduce the data traffic effectively. The simulation results show that, compared with several other algorithms, the proposed algorithm has the advantage of efficient identification and a small amount of data exchange.

**Key words:** radio frequency identification (RFID); anti-collision; tag identification

射频识别(Radio Frequency Identification, RFID)技术<sup>[1]</sup>是一种利用无线通信实现信息识别的技术, 在物联网中扮演着极其重要的角色. 一套完整的 RFID 系统主要是由读写器和标签两部分组成的, 其工作原理是读写器向标签发射特定频率的射频信号, 以驱动标签内部的电路将内部数据送出, 此时读写器便依序接收并解读数据, 实现对标签的识别. 由于多个标签共用同一信道, 因此在读写器依序接收标签返回的数据时, 可能产生信道争用, 发生数据冲突, 导致读写器不能准确识别标签<sup>[2]</sup>, 这就是标签碰撞问题. 标签碰撞严重制约着标签的识别率, 如何减少标签碰撞以提

高 RFID 系统的效率变得尤为重要. 因此, 防碰撞算法实现是 RFID 最为关键的技术之一.

当前对防碰撞算法的研究主要分为两大类: 随机性算法和确定性算法. 随机性算法又被称为概率性算法, 这类算法基本上都基于 ALOHA 机制<sup>[3]</sup>, 即标签随机选择一个时间向读写器传送信息, 如果发生标签碰撞, 则延迟一段随机的时间后再次尝试发送. 主要包括纯 ALOHA(P-ALOHA)算法<sup>[4]</sup>、时隙 ALOHA (S-ALOHA)算法<sup>[5]</sup>、动态时隙 ALOHA (DFSA)算法<sup>[6]</sup>等. 而确定性算法则是基于树的深度优先思想, 根据标签编码的唯一性选择合适的标签进行数据交换.

① 基金项目:江苏省交通运输厅资助项目(2012X08-2)

收稿时间:2015-01-05;收到修改稿时间:2015-02-16

具体讲就是当碰撞发生时, 将碰撞的标签集分解成两个小的子标签集, 然后依次识别这两个子标签集, 如果在识别子标签集的过程中仍然有碰撞发生, 则继续分解, 直到最后每个子集中只有一个标签. 基于树的深度优先思想的算法包括二进制搜索(Binary Search)算法<sup>[7]</sup>、自适应二进制分割(ABS)算法<sup>[8]</sup>、查询树(Query Tree)<sup>[9]</sup>算法等.

这两类算法各有其优劣势, 基于 ALOHA 机制的算法由于其随机性, 在标签的识别过程中可能会出现一个或几个标签一直无法被识别的情况, 即所谓的标签“饿死”问题; 确定性算法虽不存在标签的“饿死”问题, 但识别过程较长, 数据交换量较大.

针对确定性算法存在的上述问题, 本文提出了一种基于部分响应的分组式 RFID 标签防碰撞算法(Group-based anti-collision Algorithm based on Partial Response, GAPR). GAPR 首先对读写器范围内的所有标签进行随机分组<sup>[10]</sup>, 接着对每一组标签依次使用基于部分响应机制的防碰撞方法<sup>[11]</sup>进行识别. 使用 GAPR 对标签进行的防碰撞处理能减少标签碰撞概率以及数据通信量, 加快标签的识别率和识别速度.

## 1 系统概述

分组开始时, 每个标签均会产生一个随机数  $n$ ,  $n$  将保存在与标签相对应的存储器中, 所产生随机数相等的标签将被分在同一组, 组号以随机数命名, 用  $Group(n)$  表示. 标签分组将大量碰撞标签分裂成多个子部分, 使得每一组内的标签数量显著减少, 这样能降低标签间的碰撞率.

完成分组后, 读写器按照组号从小到大的顺序, 依次对分组内的标签进行部分响应式标签识别<sup>[12]</sup>, 将每组内的标签逐个识别出来. 至此完成整个 GAPR 算法过程. 部分响应式标签识别的主要优势在于响应阅读器的标签仅回复部分数据, 而非传统算法中回复全部标签数据, 这样的处理方式则有效降低了识别过程中的数据通信量.

GAPR 算法由以上所述的标签分组和部分响应式标签识别两个步骤构成. 经过两个步骤依次处理后, 显然该算法对标签识别率和识别速度的提升有显著效果.

### 1.1 标签分组数 $t$ 的最优解推导

假设共有  $n$  个标签, 将之分为  $t$  组, 即标签产生的随机数范围为  $[1, t]$ . 在标签识别阶段, 当识别其

中任意一组标签时, 由于不同时刻每个标签所产生的位于  $[1, t]$  之间的随机数的概率是相同的, 因此识别过程可以等效于在同一时刻, 该分组中的每个标签都随机产生一个  $[1, t]$  之间的随机数.

令  $S_n$  为识别出  $n$  个标签所需的平均搜索次数,  $S_i$  为识别  $i$  个标签时阅读器的搜索次数. 由于选择  $[1, t]$  之间任何一个随机数的标签数目都服从二项分布, 则有:

$$\begin{aligned} S_n &= \sum_{i=1}^n \left( \binom{n}{i} (1/t)^i (1-1/t)^{n-i} t S(i) \right) \\ &= \sum_{i=1}^n \left( \binom{n}{i} (1/t)^i (1-1/t)^{n-i} t(2i-1) \right) \\ &= \sum_{i=0}^n \left( \binom{n}{i} (1/t)^i (1-1/t)^{n-i} t(2i-1) \right) + \\ &\quad t(1-1/t)^n \\ &= 2 \sum_{i=0}^n \left( \binom{n}{i} (1/t)^{i-1} (1-1/t)^{n-i} i \right) - \\ &\quad t \sum_{i=0}^n \left( \binom{n}{i} (1/t)^i (1-1/t)^{n-i} \right) + t(1-1/t)^n \\ &= 2 \sum_{i=1}^n \left( \binom{n-1}{i-1} (1/t)^{i-1} (1-1/t)^{n-i} n \right) - \\ &\quad t \sum_{i=0}^n \left( \binom{n}{i} (1/t)^i (1-1/t)^{n-i} \right) + t(1-1/t)^n \\ &= 2n(1/t + 1-1/t)^{n-1} - t(1/t + 1-1/t)^n + \\ &\quad t(1-1/t)^n \\ &= 2n - t + t(1-1/t)^n \end{aligned}$$

为了推导出  $t$  的最优解, 令  $t = fn$ ,  $f$  为分组因子. 代入式中, 有:

$$\begin{aligned} S_n &= 2n - fn + fn(1-1/fn)^n \\ \text{两边同时除以 } n, \text{ 取 } n \rightarrow \infty, \text{ 有} \\ \lim_{n \rightarrow \infty} (S_n/n) &= \lim_{n \rightarrow \infty} (2 - f + f(1-1/fn)^n) \\ &= 2 - f + fe^{-1/f} \end{aligned}$$

在 MATLAB 中对分组因子  $f$  进行仿真, 结果如下图所示:

由图所示结果可知, 当分组因子  $f = 0.6$ , 即分组数  $t = 0.6n$  时, 标签的识别效率最高. 即  $t = 0.6n$  为分组数最优解.

### 1.2 部分响应式标签识别过程

在以 QT 算法<sup>[13]</sup>为代表的传统二叉树识别算法中, 标签每响应一次阅读器的请求都必须回复标签本身的

全部数据作为应答, 这样会造成大量不必要的数据在信道间传输, 增加整个系统的数据传输量. 如果能够减少标签响应过程中携带的数据量, 则可以有效减轻系统的通信负担.

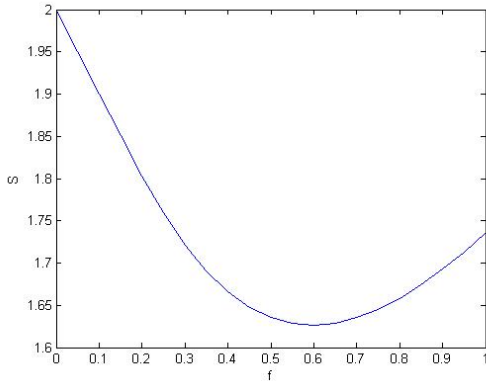


图 1 分组因子 f 仿真示意图

对每一组内的标签, 采用基于部分响应的防碰撞方法对其进行识别, 标签采用曼彻斯特编码<sup>[14]</sup>方式. 读写器在发出请求命令后, 通过标签回复的曼彻斯特编码, 能获取到具体的碰撞位, 此时只有发生碰撞的位置的信息<sup>[15]</sup>是未知的, 其他位置的信息可以准确获得.

引入部分响应机制的标签识别算法具体步骤如下:

设标签的编码为  $D_n D_{n-1} \dots D_k D_{k-1} \dots D_1$ , 假设长度为  $n - k + 1$  的标签前缀  $D_n D_{n-1} \dots D_k$  未发生碰撞, 用  $s$  表示. 具体识别方法描述如下: 读写器首先将请求前缀  $s$  送入识别区域中, 位于区域中的前缀同样为  $s$  的标签发生响应, 此时读写器判断  $D_{k-1}$  的值, 若该标签返回的  $D_{k-1}$  为 0, 则回复该标签的  $D_{k-1} \dots D_2 D_1$  给读写器; 若该标签返回的  $D_{k-1}$  为 1, 则回复  $D_{k-1}$  给读写器. 读写器在收到回复后, 便能根据  $D_{k-1}$  的值进行如下判断:

如果  $D_{k-1}$  为 0, 那么意味着识别区域中没有前缀为  $s1$  的标签. 此时如果有标签回复的数  $D_{k-1} \dots D_2 D_1$  能被读写器准确识别, 即未发生碰撞, 则这个标签能被识别出来; 否则读写器将  $s0$  作为请求前缀送入识别区域中, 并重复识别过程.

如果  $D_{k-1}$  为 1, 那么意味着识别区域中没有前缀为  $s0$  的标签. 此时读写器将  $s1$  作为请求前缀送入识别区域中, 并重复识别过程.

如果  $D_{k-1}$  的值无法准确判断出来, 那么意味着前缀为  $s0$  和  $s1$  的标签均存在. 此时如果有标签回复的数据  $D_{k-1} \dots D_2 D_1$  能被读写器准确识别, 即未发生碰撞, 则这个标签能被识别出来; 否则读写器依次将  $s0$  和  $s1$

作为请求前缀送入识别区域中, 并重复识别过程.

当每一个分组内的标签均被依次识别出来后, 查询过程完成.

### 1.3 算法相关命令

#### ① Request(s)

查询命令, 将  $s$  送入识别区域中.  $s$  为查询前缀.

#### ② Rand\_devide(t)

分组命令, 产生区间  $[1, t]$  内的随机数<sup>[16]</sup>作为组号分配给标签, 并保存在标签中. 拥有相同组号的标签被分入同一组.

#### ③ Group\_select(g)

选中分组命令, 对选中的组进行标签识别.  $g$  为组号.

#### ④ Select(p)

选中标签命令, 表明标签被成功识别.  $p$  为被识别的标签.

### 1.4 算法流程与实例

GAPR 算法的流程主要包括标签分组与标签识别两个部分, 如图 2 所示.

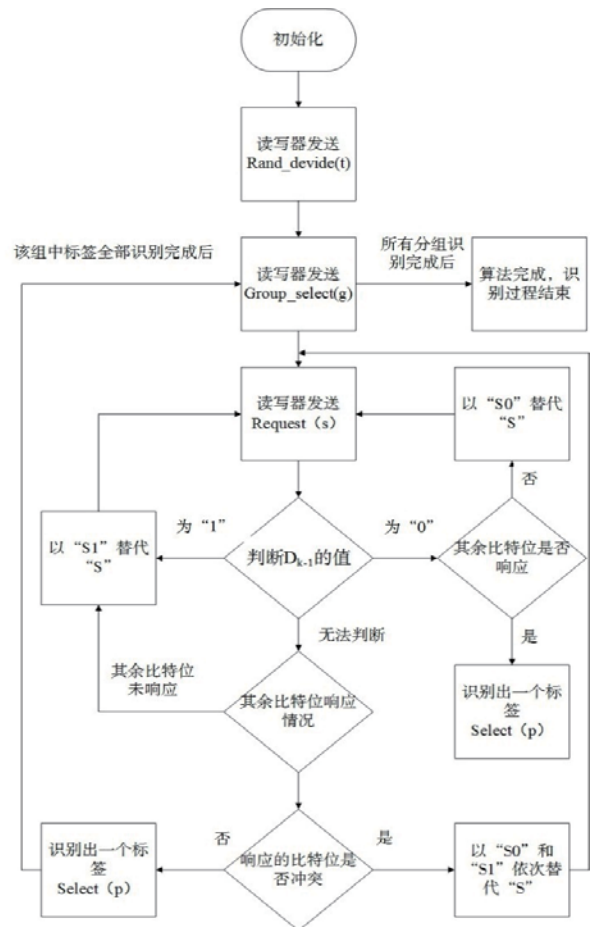


图 2 GPRA 算法的防碰撞处理流程图

为简单起见, 假设对标签的分组已经完成, 取其中一组, 组内标签个数为 6 个, 标签长度为 8 位. 依次表示为 A(01001011), B(01000010), C(01010001), D(01000001), E(01000100), F(01001010).

利用 GAPR 算法对这 6 个标签进行识别的具体过程见表 1.

表 1 GAPR 算法实例

请求前缀	栈中数据	响应标签	响应数据	识别情况
-	0100	-	-	-
	0101			
		B-01000010	0010	
	01000	D-01000001	0001	
0100	01001	E-01000100	0100	发生碰撞
	0101	A-01001011	1	
		F-01001010	1	
	010000			
	010001	B-01000010	010	
01000	01001	D-01000001	001	发生碰撞
	0101	E-01000100	1	
	0100001			
010000	010001	D-01000001	10	识别出 D
	01001	B-01000010	1	
	0101			
	010001			
0100001	01001	B-01000010	0	识别出 B
	0101			
	010001			
010001	01001	E-01000100	00	识别出 E
	0101			
	010010			
01001	010011	A-01001011	11	发生碰撞
	0101	F-01001010	10	
	0100101			
010010	010011	A-01001011	1	发生碰撞
	0101	F-01001010	1	
	01001011			
0100101	010011	F-01001010	0	识别出 F
	0101	A-01001011	1	
	010011			
01001011	0101	A-01001011	-	识别出 A
	0101			
010011	0101	-	-	-
0101	-	C-01010001	0001	识别出 C

## 2 分析与仿真

假设读写器识别范围内的标签总数量为  $N$ , 每个标签所产生随机数的范围为  $1, t$ , 即共生成  $t$  个分组. 设  $S_i$  和  $C_i$  分别为识别第  $i$  个分组中所有标签所需要的查询次数和数据通信量, 那么总的查询次数为:

$$S(N) = S(0) + S(1) + \dots + S(t-2) + S(t-1) = \sum_0^{t-1} S(i)$$

总的通信量为:

$$C(N) = C(0) + C(1) + \dots + C(t-2) + C(t-1) = \sum_0^{t-1} C(i)$$

由于标签产生随机数, 在标签数量足够大的情况下, 可认为标签产生的随机数服从均匀分布, 此时每个分组中的标签数量的期望值应为  $N/t$ . 对将要识别的标签优先进行分组处理后, 每个分组内的标签数量显著减少, 显然这一处理能大大减小标签的碰撞概率, 即  $S(N)_{GAPR} \ll S(N)_{ELSE}$ .

我们知道诸如查询树算法等普通的二叉树查询算法, 标签每次响应过程均回复该标签的全部数据给读写器, 而 GAPR 算法对于返回的  $D_{k-1}$  为 1 的标签, 仅回复 1bit 的数据给读写器, 这样能大大减少标签识别过程中数据的通信量, 即  $C(N)_{GAPR} \ll C(N)_{ELSE}$ .

### 2.2 仿真结果

针对本文提出的 RFID 标签防碰撞算法, 在 MATLAB8.0 环境下, 不计前后缀、校验冗余等开销的理想信道下进行了仿真实验. 仿真标签数量从 25 逐渐增加到 1000, 标签的长度为 96bit, 仿真程序连续运行 100 次取平均值. 把本文提出的 GAPR 与查询树算法、仅分组算法和仅采取部分响应算法相比较, 得到如下结果.

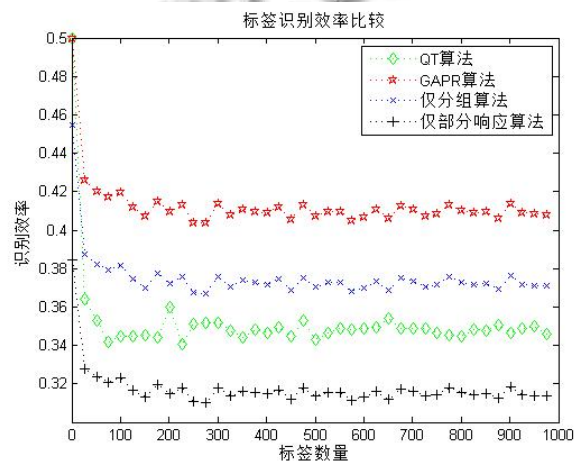


图 3 标签识别率比较

图 3 所示为四种不同算法下标签的识别效率比较, 实验结果表明相对于其他三种算法, GAPR 算法的标签识别效率最高, 四种算法的标签识别效率基本保持

稳定,不随标签数量的增加而变动。

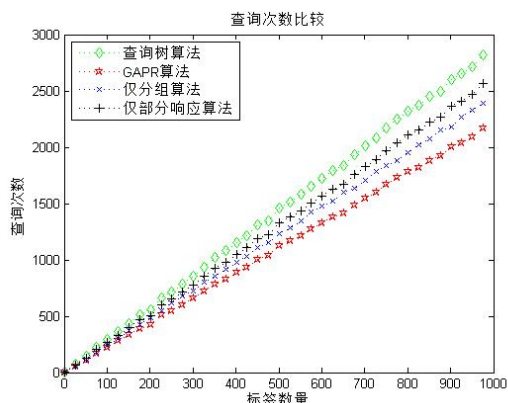


图4 查询次数比较

图4显示了四种不同算法在标签数量逐渐增加时查询次数的变化情况,可以看到四种不同算法在标签数量逐步增加时,查询次数几乎与标签数量成正比,在查询次数方面,各算法的差别不显著,但仍可观察到GPR算法的查询次数最少.减少查询次数可以作为下一步算法优化的关键点。

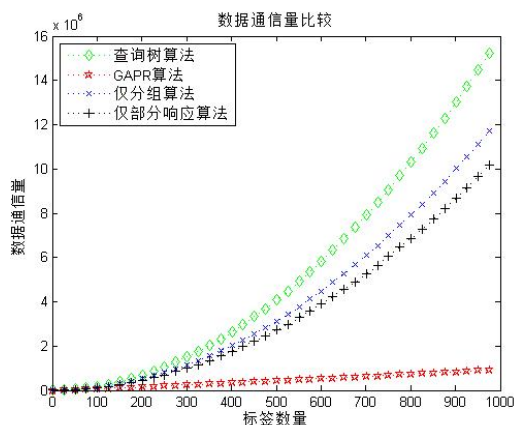


图5 数据通信量比较

数据通信量的仿真结果如图5所示,查询树算法和仅分组算法等由于标签响应时回复给阅读器全部数据,导致数据通信量很大,而引入部分响应机制后,标签在响应阅读器请求时,仅回复部分有效数据,由图可知,GPR算法的数据通信量远远小于其他三种算法,具有明显优势。

### 3 结语

本文提出了一种基于部分响应的分组式RFID标签防碰撞算法GPR.对算法进行分析与仿真后的结

果表明该算法能显著减少标签识别过程中的数据通信量,显著提高标签识别效率.在实验结果中可以看到,由于对查询次数的有效减少效果不够明显,在进一步的工作中将主要对该算法减少查询次数做出改进。

### 参考文献

- 1 陈剑,冀京秋,陈宝国.我国射频识别(RFID)技术发展战略研究.科学决策,2010,(1).
- 2 张棋飞,刘威,孙宝林,桂超,严冰.基于冲突分类模型的冲突解析算法.软件学报,2010,(3).
- 3 吴海锋,曾玉.自适应帧 Aloha 的 RFID 标签防冲突协议.计算机研究与发展,2011,(5).
- 4 Hu YB, Yang WW, Cai YM. Throughput analysis of slotted ALOHA with cooperative transmission using successive interference cancellation. Science in China, 2009(12).
- 5 李萌,钱志鸿,张旭,王义君.基于时隙预测的 RFID 防碰撞 ALOHA 算法.通信学报,2011,(12).
- 6 Wang YQ, Jiang GP. Dynamic framed slotted ALOHA algorithm used in RFID system. Journal of Nanjing University of Posts and Telecommunications (Natural Science), 2010, (1).
- 7 王海涛,朱洪.改进的二分法查找.计算机工程,2006,(10).
- 8 吕敬祥,曾小芸,杨建平.一种基于自适应前缀查询树的 RFID 防碰撞算法题名.制造业自动化,2013,(6).
- 9 南敬昌,单晓艳,高明.RFID 系统中改进的混合查询树防碰撞算法.计算机工程,2012,(23).
- 10 王亚奇,蒋国平.基于分组机制的跳跃式动态二进制防碰撞算法.自动化学报,2010,(10).
- 11 白宇,郭显娥.深度优先稳定原地归并排序的高效算法.计算机应用,2013,(4).
- 12 彭佳厚,唐明浩.一种分组式 RFID 防碰撞算法的研究.电子测量技术,2011,(11).
- 13 Pupunwiwat R, Darcy P, Stantic B. Conceptual selective RFID anti-Collision technique management. Procedia Computer Science, 2011, 5: 827-834.
- 14 孙枫叶,张科峰,刘冬生,李孝煌.高频RFID标签曼彻斯特编码电路的设计.电子技术应用,2006(10).
- 15 Parveen M, Khader SAP, Rabbani MAM. Analysis of bit grouping algorithm for collision resolution in passive RFID tags. International Journal of Engineering Science and Technology, 2010, 2(9), 4192.
- 16 李国鸿,梁红.一种 K 分布随机数产生方法.系统仿真学报,2007,(2).