

一种改进的 RFID 自适应防碰撞算法

徐海峰, 姜 晖, 刘 振

(合肥电子工程学院信息系, 合肥 230037)

摘 要: 当读写器响应范围内的标签数目较多时, 自适应搜索矩阵算法的性能会迅速下降。为此, 提出一种改进的无线射频识别自适应防碰撞算法。采用查询机制, 利用碰撞信息, 获得查询前缀, 通过构造动态搜索矩阵进行逐段查询, 并根据时隙状态自适应调整搜索路径, 对算法的碰撞时隙数、空间时隙数、吞吐量和传输比特数进行性能分析。实验结果表明, 该算法能提高系统吞吐量和识别效率。

关键词: 无线射频识别; 防碰撞; 动态搜索矩阵; 自适应搜索矩阵算法; 吞吐量; 识别效率

An Improved Adaptive Anti-collision Algorithm in RFID

XU Hai-feng, JIANG Hui, LIU Zhen

(Department of Information, Electronic Engineering Institute of Hefei, Hefei 230037, China)

【Abstract】 In order to solve the problem that tags can't be identified quickly and reliably in the Adaptive Search Matrix(ASM) algorithm, an improved adaptive anti-collision algorithm in Radio Frequency Identification(RFID) is presented. The novel algorithm adopting a new search mechanism uses the information of collision bits to gain the proper prefixes from tags, and then constructs a dynamic search matrix to identify tags. It adjusts the search paths adaptively based on the states of timeslots to decrease the collision timeslots and avoid the idle timeslots. Theory and simulations show that the new algorithm can improve the throughput of system, and enhance the identification efficiency effectively.

【Key words】 Radio Frequency Identification(RFID); anti-collision; dynamic search matrix; Adaptive Search Matrix(ASM) algorithm; throughput; identification efficiency

DOI: 10.3969/j.issn.1000-3428.2012.17.078

1 概述

无线射频识别(Radio Frequency Identification, RFID)技术是一种利用无线通信实现的非接触式自动识别技术。其因具有许多优点已广泛应用于工业自动化、智能交通运输控制管理、物流等众多领域。在读写器有效通信范围内, 多个标签同时与读写器进行通信, 可能会使得读写器接收到的标签信息发生重叠, 导致读写器无法准确识别标签。因此, 标签防碰撞算法已成为 RFID 领域研究的热点之一。

通常可分为随机性算法和确定性算法^[1]2 种, 具体如下:

(1)随机性算法大都采用 ALOHA 机制, 包括动态帧时隙 ALOHA(Dynamic Frame Slotted ALOHA, DFSA)算法^[2], 分群时隙 ALOHA 算法(Grouping Slotted ALOHA, GSA)^[3]和精确标签估计(RFID Explicit Tag Estimation Scheme, RETES)算法^[4]等。该类算法是标签随机进行分配响应读写器的查询, 实现简单、成本较低, 但存在某一标签在相当长一段时间内无法识别的情况, 即“Tag Starvation”现象。

(2)基于二进制树搜索的算法, 主要包括自适应二叉树搜索(Adaptive Binary Splitting, ABS)算法^[5]、后退式索引算法^[6]、自适应搜索矩阵(Adaptive Search Matrix, ASM)算法^[7]和改进的基于二叉树搜索混合防碰撞算法等。该类算法比较复杂、延时较长, 但不存在“Tag Starvation”现象, 能够识别读写器有效通信范围内所有标签。

ASM 算法通过利用读写器首次查询获得的碰撞位信息, 构造一个固定的搜索矩阵进行标签查询, 在一定程度上提高了标签识别速度, 但当读写器响应范围内标签数目较多时, 该算法性能会迅速下降。为此, 本文提出一种改进的自适应搜索矩阵算法。利用标签自身序列号之间的碰撞信息进行特定位查询, 从而获得准确的查询前缀, 以实现逐段查询。

2 自适应二叉树搜索算法和自适应搜索矩阵算法

2.1 自适应二叉树搜索算法

文献[5]提出自适应二叉树搜索算法。该算法根据标签的响应时将时隙分为碰撞、空闲和可读 3 种状态。其要求系统内各标签具有一个计数器, 记录当前正在比较的 EPC 比特位 $i(1 \leq i \leq m)$, 其中, m 为标签电子产品码(Electronic

作者简介: 徐海峰(1986—), 男, 硕士研究生, 主研方向: 无线射频识别技术, 嵌入式系统; 姜 晖, 教授; 刘 振, 硕士研究生

收稿日期: 2011-10-24 **修回日期:** 2011-12-26 **E-mail:** xuhai Feng8349@163.com

Product Code, EPC)的长度。读写器对标签 EPC 进行逐位搜索, 其发送正在比较的 EPC 第 i 位, 而只有相应位与读写器查询相一致的标签进行响应, 将其 EPC 第 $i+1$ 位发送给读写器, 同时标签的计数器自动更新, $i=i+1$ 。读写器根据收到的 EPC, 判断时隙是否为碰撞、空闲还是可读, 从而自适应对标签进行搜索识别。

ABS 算法在每一个时隙内只有 2 bit 的数据通信量, 进一步提高了标签的响应速度, 但当 2 个标签的 EPC 前 $j-1$ 位相同而第 j 位不同时, 算法需要进行深度为 j 的搜索, 才能将 2 个标签识别, 此时算法的搜索效率不高。

2.2 自适应搜索矩阵算法

为提高 RFID 系统的搜索效率, 在 ASM 算法中读写器利用曼彻斯特编码可有效识别多标签 EPC 之间碰撞比特出现的位置, 获得一个固定的 $2 \times k (1 \leq k \leq m)$ 阶前缀搜索矩阵 R , 其中, k 为读写器通信范围内标签 EPC 之间出现碰撞的次数, 亦是读写器进行标签识别时的搜索深度; 2 是每一深度都需要查询的分支数。该算法引入了碰撞堆栈的概念, 用来保存读写器进行查询发生碰撞时的搜索深度, 并为空闲时隙和可读时隙中读写器返回上层未完成全分支搜索的碰撞结点处, 进行查询提供所需的搜索深度, 从而做到根据不同时隙状态自适应调整搜索路径。

当标签 EPC 之间碰撞位置不连续时, 读写器每次是逐段进行前缀查询的, 克服了 ABS 算法中逐位进行查询效率低的缺点, 能大幅度提高标签的识别效率, 但这种情况一般只出现在标签数目较少的场合。当标签数目较多时, 标签 EPC 之间碰撞位置将连续, 搜索深度增加, 算法的性能也会降低。

3 改进的自适应搜索矩阵算法

为克服 ASM 算法在标签数目较多的情况下, 因为标签 EPC 之间出现连续碰撞位置而导致搜索效率较低的缺点, 本文对其前缀搜索矩阵 R 进行改进, 使其做到动态调整, 进一步提高标签的识别效率。

为了便于分析, 假设本文算法与 ASM 算法具有共同的前提条件, 即在整个标签识别过程中标签不会发生变

化。本文算法的前缀搜索矩阵是 $8 \times k (1 \leq k \leq m)$ 阶, 这里 k 仍为读写器进行标签识别时的搜索深度, 亦是读写器对利用曼彻斯特编码方法进行首次查询获得的标签 EPC 碰撞信息分段的数目。其分段规则如下:

(1) 当标签 EPC 之间的某一碰撞位与其他碰撞位不连续时, 该段为其前一碰撞位之后直到当前碰撞位这一部分。

(2) 当标签 EPC 之间的某 2 位或 3 位连续碰撞位与其他碰撞位不连续时, 该段为其前一碰撞位之后直到当前连续碰撞位这一部分。

(3) 当标签 EPC 之间的某连续碰撞位超过 3 位时, 该段为其前一碰撞位之后直到当前连续碰撞位前 3 位这一部分, 多余的碰撞位则按照此规则继续分段。

本文算法首先利用曼彻斯特编码方法进行碰撞位识别, 然后按照分段规则分段后构造初始前缀搜索矩阵。在标签搜索过程中, 如果有多个标签进行响应, 读写器则采用一种新的查询机制获得准确的查询前缀, 以动态调整搜索矩阵, 其调整方式如下:

(1) 如果查询的分段内只有一位碰撞位, 则在 R 相应位置直接写入 0 和 1, 形成 2 个分支。

(2) 如果查询的分段内有 2 位或 3 位碰撞位, 则读写器对当前响应的标签相应位置进行特定位置查询, 以获得准确的查询前缀, 并对 R 进行调整。

特定位置查询即为新的查询机制。设标签 EPC 为 $q_{m-1}, q_{m-2}, \dots, q_0$, 其中, $q_i (0 \leq i \leq m-1)$ 可以为 0 或 1, $Req(q_{m-1}, q_{m-2}, \dots, q_0)$ 为读写器的查询命令, 如果 $q_i (0 \leq i \leq m-1)$ 为 1, 则标签的相应位被查询。标签响应时则把当前查询分段内的碰撞位转化为十进制数 $d (0 \leq d \leq 7)$, 并返回 8 位比特信息 $q_7, q_6, \dots, q_0, q_d$ 被置 1, 其余为 0。读写器根据标签返回的信息可以获得准确的查询前缀, 进而调整搜索矩阵。由于本文算法通过特定位置查询可获得准确的查询前缀, 其避免了空闲时隙的产生。该算法同样采用碰撞堆栈, 保存当前查询碰撞发生时的深度。

本文算法流程如图 1 所示。

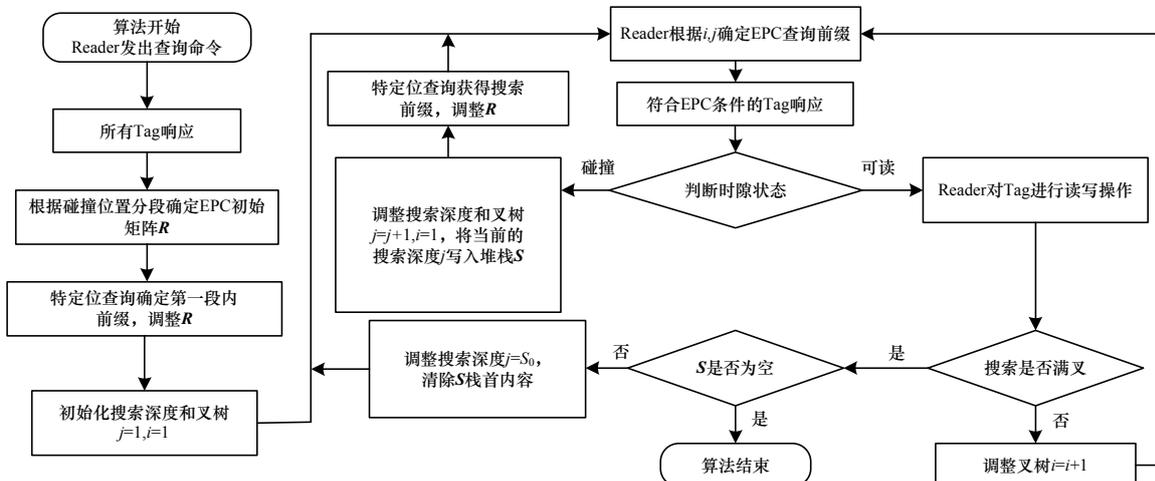


图 1 本文算法流程

对本文算法进行详细描述:

(1)初始化:首先读写器发出查询命令,所有标签应答。读写器根据碰撞信息和分段规则对 EPC 序列进行分段,并构造一个 $8 \times k(1 \leq k \leq m)$ 阶初始前缀搜索矩阵 R , 此时各段均为满叉, 未被写入的位置用相应段碰撞位长个 0 填充。令 $i=1, j=1$, i 对应矩阵的行表示搜索的叉树, j 对应矩阵的列表示搜索的深度。读写器利用新的查询机制, 获得第一分段准确的查询前缀, 并写入 R 中。

(2)根据不同时隙状态, 自适应得调整搜索路径。

碰撞时隙: 判断将当前搜索是否为满叉, 如果不是, 将当前搜索深度值将当前搜索深度值 j 保存到碰撞堆栈矩阵 S 中。同时, 读写器利用新查询机制, 获得查询前缀, 记录当前的分叉树, 并对 R 进行动态调整。特定位查询时发生碰撞不改写碰撞堆栈, 令 $i=1, j=j+1$, 读写器根据当前的 i, j , 确定 EPC 继续查询。

可读时隙: 读写器对标签进行识别和相应读写操作。判断当前搜索是否满叉, 如果不是, 则 $j=j+1$, 继续下一叉树的搜索。如果是, 则返回到碰撞堆栈栈首所表示的那一深度。同时, 清除堆栈的栈首内容, 读写器根据当前的 i, j , 确定 EPC 继续搜索。

算法结束: 如果堆栈的内容为空, 当前的深度的搜索是满叉, 时隙状态为非碰撞时, 算法结束。

4 算法性能分析

对本文算法碰撞时隙数、空闲时隙数、吞吐量和传输比特数进行性能分析。为便于分析, 假设读写器通信范围内标签数目较多, 容易使得标签 EPC 之间每一比特位发生碰撞。

标签 EPC 序列长为 m , 标签个数为 n , 则前缀搜索矩阵 R 的搜索深度为:

$$k = \lceil m/3 \rceil$$

其中, $\lceil \cdot \rceil$ 为向上就近取整。

由于本文算法避免了空闲时隙数的产生, 且特定位查询所需时隙数和碰撞时隙数相同, 则算法识别标签所需要的总时隙数为:

$$T_{IASM} = 2T_c + T_r \tag{1}$$

其中, T_c 和 T_r 分别代表碰撞时隙数和可读时隙数, 总时隙数与算法的吞吐量密切相关, 其定义为 $S = (n/T_n) \times 100\%$ 。

对于深度为 j 的搜索, 如果最多可确定 8^j 个时隙, 那么在该深度下, 估计会出现的可读时隙数和碰撞时隙数分别为:

$$T_r(n, j) = 8^j \frac{n}{8^j} (1 - \frac{1}{8^j})^{n-1} = n(1 - \frac{1}{8^j})^{n-1} \tag{2}$$

$$T_c(n, j) = 8^j [1 - T_r(n, j)] = 8^j [1 - \frac{n}{8^j} (1 - \frac{1}{8^j})^{n-1}] \tag{3}$$

而 T_c 和 T_r 为不同深度 j 对应的时隙数的总和。则它们可以分别表示为:

$$T_c = \sum_{j=0}^{\lceil m/3 \rceil} T_c(n, j) \tag{4}$$

$$T_r = \sum_{j=0}^{\lceil m/3 \rceil} T_r(n, j) \tag{5}$$

将式(2)~式(4)和式(5)代入式(1)即可得算法的总时隙数, 当标签 EPC 序列不是每一位都发生碰撞时, 其性能要好于上述分析结果。

对于 ASM 算法, k 为标签 EPC 中碰撞的次数, 则其碰撞时隙数和空闲时隙数分别为:

$$T_c = \sum_{k=1}^m \sum_{i=0}^k \left\{ 2^i [1 - (1 - \frac{1}{2^i})^n - \frac{n}{2^i} (1 - \frac{1}{2^i})^{n-1}] \cdot C_m^{m-k} (1 - \frac{1}{2^{n-1}})^k (\frac{1}{2^{n-1}})^{m-k} \right\} \tag{6}$$

$$T_e = \sum_{k=1}^m \sum_{i=0}^k 2^i (1 - \frac{1}{2^i})^n C_m^{m-k} (1 - \frac{1}{2^{n-1}})^k (\frac{1}{2^{n-1}})^{m-k} \tag{7}$$

对于 ABS 算法^[5], 其碰撞时隙数为:

$$T_c = \sum_{i=0}^m 2^i [1 - (1 - \frac{1}{2^i})^n - \frac{n}{2^i} (1 - \frac{1}{2^i})^{n-1}] \tag{8}$$

当标签数目较多时, 标签 EPC 之间每一位都发生碰撞, ASM 算法也只能逐位进行搜索, 其性能和 ABS 算法相同, 这 2 种算法所需总时隙数都满足: $T_n = 2T_c + 1$ 。对于算法识别标签所需传输的数据通信量, 本文算法中每个碰撞时隙和可读时隙传送的数据比特数平均为 6 位, 而在特定位查询时隙中需要传送 16 位数据比特, 则该算法识别标签所需传输的数据通信量为:

$$D_{IASM} = 6(T_c + T_r) + 16T_c = 10T_c + 6T_r \tag{9}$$

而对于 ABS 算法和 ASM 算法, 其识别标签所需传输的数据通信量分别为:

$$D_{ABS} = 2T_{ABS} \tag{10}$$

$$D_{ASM} = 2 \sum_{k=1}^m \frac{m}{k} T_{ASM}(k) C_m^{m-k} (1 - \frac{1}{2^{n-1}})^k (\frac{1}{2^{n-1}})^{m-k} \tag{11}$$

5 仿真实验

为了验证本文算法的性能, 这里用 Matlab 对本文算法、ASM 算法、ABS 算法和 BT 算法进行仿真。由于当标签数目较多时, ASM 算法和 ABS 算法性能相同, 因此, 本文对 BT 算法也进行仿真用作对照。设标签的 EPC 为随机产生的 32 位序列, 并对多次相同条件下的仿真结果取 50 次平均。图 2、图 3 给出了 4 种算法在碰撞时隙数、空闲时隙数和吞吐量以及传输比特数的比较。

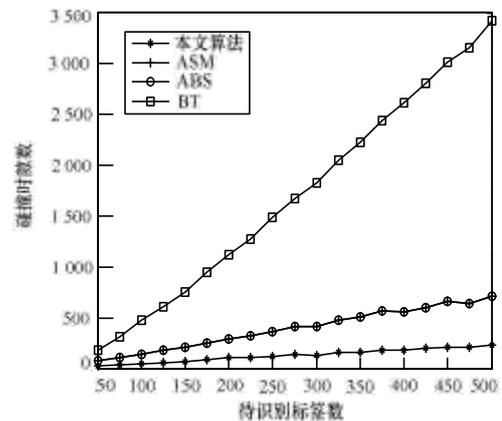
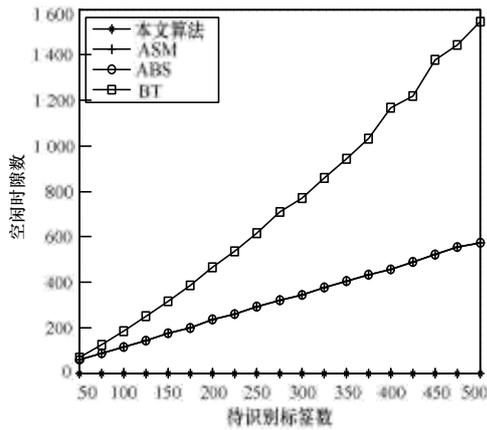
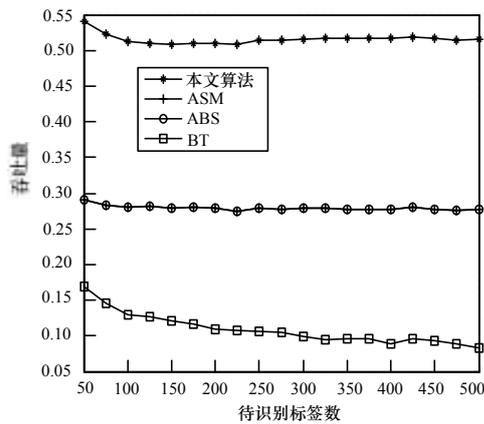


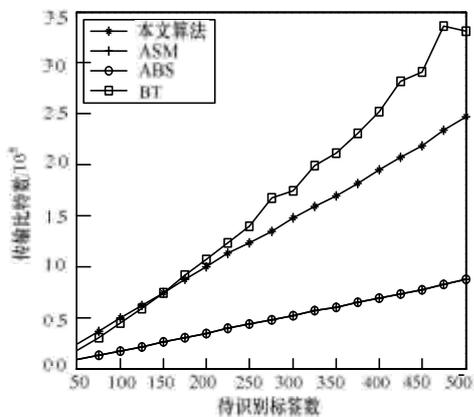
图 2 4 种算法碰撞时隙数比较



(a) 空闲时隙数比较



(b) 吞吐量比较



(c) 总传输比特数

图 3 4 种算法性能比较

由图 2、图 3 可知, 本文算法在碰撞时隙、空闲时隙和吞吐量方面明显优于另外 3 种算法, 其由于特定位置查询, 获得准确的查询前缀, 有效避免了空闲时隙的产生, 大幅度减少了碰撞时隙数, 提高了算法的吞吐量, 能使其保持在 50% 以上。当标签数目较多时, 读写器有效通信范围内标签 EPC 之间每一位都会发生碰撞, ASM 算法进行搜索时只能逐位进行搜索, 其性能将和 ABS 算法相同,

仿真结果也验证了这一点。

本文算法有效提高了标签的识别速度, 但是其由于利用特定位置查询, 增加了算法的复杂度, 也使得算法识别标签所需的传输比特数有所增加。在算法实现方面, 本文算法中读写器需要具有 2 个寄存器, 一个作为碰撞堆栈使用, 另一个则用来记录动态的搜索叉树。此外, 算法还要求标签能够实现特定位置响应功能。目前, 有的标签具有微控制器, 能够使标签具备上述能力, 虽其在一定程度上增加了标签成本, 但能应用于有源 RFID 系统中, 满足人们在社会信息化和科学技术迅速发展的条件下对高速移动标签的识别要求。

6 结束语

本文提出一种改进的自适应搜索矩阵算法。给出算法的具体步骤, 对算法性能进行了对比分析, 并基于 Matlab 进行了仿真验证。实验结果表明, 该算法能大幅度减少碰撞时隙数, 避免了空闲时隙的产生, 从而提高了标签识别效率, 使其时隙吞吐量能达到 50% 以上。今后将对动态环境中的算法性能进行研究。

参考文献

- [1] Vogt H. Multiple Object Identification with Passive RFID Tags Systems[C]//Proc. of IEEE International Conference on Man and Cybernetics. [S. l.]: IEEE Press, 2002.
- [2] Lee S R, Joo S D, Lee C W. An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification[C]//Proc. of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. [S. l.]: IEEE Press, 2005.
- [3] Tae W, Byong G. Improved Anti-collision Scheme for High Speed Identification in RFID System[C]//Proc. of the 1st International Conference on Innovative Computing, Information and Control. [S. l.]: IEEE Press, 2006.
- [4] Pupunwiwat P, Stantic B. A RFID Explicit Tag Estimation Scheme for Dynamic Framed-slot ALOHA Anti-collision[C]//Proc. of the 6th International Conference on Wireless Communications, Networking and Mobile Computing. [S. l.]: IEEE Press, 2010.
- [5] Jihoon M, Lee W. Adaptive Binary Splitting: A RFID Tag Collision Arbitration Protocol for Tag Identification[J]. Mobile Networks and Applications, 2005, 11(5): 711-722.
- [6] Yu Songsen, Zhang Yiju, Peng Weidong, et al. An Anti-collision Algorithm Based on Binary-tree Searching of Regressive Index and Its Practice[J]. Computer Engineering and Application, 2004, 26(16): 26-28.
- [7] 丁治国, 郭立, 刘琦. 一种基于搜索矩阵的自适应防碰撞算法[J]. 模式识别与人工智能, 2008, 21(4): 476-481.

编辑 刘冰