

基于分组动态帧时隙的 RFID 防碰撞算法

尹君, 何怡刚, 李兵, 邓晓, 谭阳红, 肖迎群

(湖南大学电气与信息工程学院, 长沙 410082)

摘要: 为了解决射频识别(RFID)系统中的多标签防碰撞问题, 在分析帧时隙 ALOHA 算法的基础上, 提出一种基于分组动态帧时隙的 RFID 防碰撞算法。当标签数量庞大时, 该算法可以通过分组限制响应标签数量达到较高的识别效率。仿真结果表明, 当标签数为 1 000 时, 与传统算法相比, 该算法能使时隙利用率提高 80% 以上。

关键词: 帧时隙; 防碰撞; 分组

RFID Anti-collision Algorithm Based on Grouping Dynamic Frame Slotted

YIN Jun, HE Yi-gang, LI Bing, DENG Xiao, TAN Yang-hong, XIAO Ying-qun

(College of Electrical and Information Engineering, Hunan University, Changsha 410082)

【Abstract】 In order to solve the problem of collision between multi-tag in radio frequency identification(RFID) system. This paper proposes RFID anti-collision algorithm based on grouping dynamic frame slotted by analyzing frame slotted ALOHA algorithm. When there are a large number of tags in the field, it can achieve high discernment efficiency by grouping to restrict the number of response tags. Simulation results show that the algorithm improves the slot utilization rate above 80% comparing with the conventional algorithms, when the number of tag is 1 000.

【Key words】 frame slotted; anti-collision; grouping

1 概述

射频识别(Radio Frequency Identification, RFID)是于 20 世纪 90 年代提出的一种利用无线信道实现双向通信的识别技术。由于其显著的特点, 因此被认为可以替代传统条形码、磁卡和 IC 卡等技术, 目前已被广泛应用于生产过程、物流仓储、商业零售、商品防伪、保安和交通管理等领域。RFID 系统最大的难题在于多标签碰撞时较低的标签数据识读率, 多标签碰撞是指当多个标签同时存在于同一个射频信道内时, 阅读器无法读取标签数据的现象。

目前, 解决 RFID 标签阅读冲突问题最广泛的是帧时隙 ALOHA 算法和二进制搜索算法。由于简单实用, 因此帧时隙 ALOHA 算法应用更为频繁^[1], 例如 ISO/IEC18000-6 Type A 协议和 EPC Class1 协议都是使用帧时隙 ALOHA 算法。

2 帧时隙 ALOHA 算法^[2]

帧时隙 ALOHA(Framed Slotted ALOHA, FSA)算法是一种随机时分多址方式的非同步通信收发算法。FSA 算法的信息帧时分多址如图 1 所示。

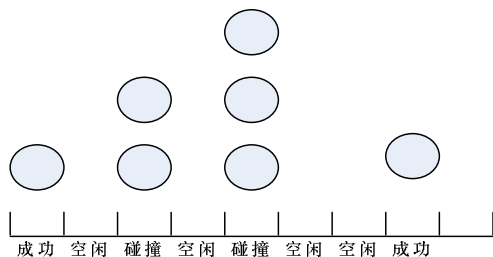


图 1 FSA 算法的信息帧时分多址

该算法将信道用信息帧表示, 其中, 帧是指由阅读器要

求的包含若干时隙的时间间隔。信息帧可以分成多个时隙, 其中, 时隙是指标签发送自身标识的时间长度。当一个时隙只被一个标签占有时, 阅读器才会正确识别该标签, 而当一个时隙内有 2 个或 2 个以上标签时, 会发生碰撞, 读写器无法正确识别, 若时隙为空则跳过。根据帧内时隙数是否变化可以分为普通帧时隙 ALOHA 算法和动态帧时隙 ALOHA 算法。

2.1 普通帧时隙 ALOHA 算法

普通帧时隙 ALOHA(Basic Framed Slotted ALOHA, BFSa)算法在识别过程中, 每帧时隙数相同。在开始识别时, 读写器向场内所有标签发送一个包含时隙数 L 的命令, 标签从 $1 \sim L$ 中随机选择 1 个数作为其时隙数值, 当该数值与时隙计数器的值相同时, 标签向读写器发送应答信息。标签被成功识别后, 读写器发送去活命令, 该标签退出识别系统。在一帧完成后, 读写器开始时隙数仍为 L 的新帧。

BFSa 算法简单, 但当标签数量巨大时, 会产生过多碰撞; 反之, 会产生很多空闲。上述 2 种情况都会降低识别率, 只有当标签数和时隙数相当时, 系统吞吐率最大。

2.2 动态帧时隙 ALOHA 算法

动态帧时隙 ALOHA(Dynamic Frame Slotted ALOHA, DFSa)算法根据识别标签的时隙数目和产生碰撞的时隙数目

基金项目: 国家“863”计划基金资助项目(2006AA04A104); 国家自然科学基金资助项目(50677014); 高校博士点基金资助项目(20060532002); 湖南省科技计划基金资助项目(06JJ2024)

作者简介: 尹君(1983-), 男, 硕士研究生, 主研方向: RFID 测试技术, 电路与系统; 何怡刚, 教授、博士生导师; 李兵、邓晓, 博士研究生; 谭阳红, 副教授; 肖迎群, 博士研究生

收稿日期: 2009-04-07 **E-mail:** yinjun6451664@yahoo.cn

等信息来决定帧长度，当待识别标签数大于时隙数而造成过多碰撞时会增加帧长度；反之，当空闲时隙过多时会减小帧长度，只有使时隙数与标签数量相当才能达到最佳吞吐量。因此，DFSA 算法基本上可以解决 BFSa 算法标签识别效率低的问题。

另一方面，如果现场有大量标签且远大于帧时隙数时，由于硬件关系，帧长度增大有限 ($L_{\max}=256$)，那么此时系统的识别效率会急剧降低，标签碰撞概率会增大，识别标签的耗时也会呈指数增加。为了解决该问题，本文提出一种将响应的标签数目限制在识读效率最高的标签数目之内的基于分组动态帧时隙的 RFID 防碰撞算法。

3 基于分组动态帧时隙的 RFID 防碰撞算法

基于分组动态帧时隙的 RFID 防碰撞算法是在 DFSA 算法的基础上提出的，针对大规模标签进行快速识别的一种改进型算法。此算法很好地改善了标签识读效率问题，即使有大量标签同时存在，该算法也能线性地增加请求时间来识读标签。

3.1 算法描述

由于各标签距阅读器距离不同，近距离标签发送的信息可能覆盖远距离标签发出的信息，因此即使发生碰撞，阅读器也可能只识别近距离标签的信息，忽略远距离的信息。同样，由于其他环境噪声的影响，即使在 1 个时隙内只有 1 个 RFID 标签应答，阅读器也可能无法识别，因此本文不考虑上述 2 种不理想条件即捕获效应和环境噪声的影响^[2]。假定识别区域内标签数量固定且不变化。

将阅读器工作范围内的标签设定为 3 种状态：

- (1)待命态，等待参与读写器识别；
- (2)休眠态，暂时不参与识别，处于未激活状态；
- (3)去活态，识别完毕的进入去活状态，对读写器的询问命令不做应答。

本文算法首先要估算未识别的标签个数，与给定的最大帧长度 $L_{\max}=256$ 进行对比，以此作为是否分组的依据。当未识别的标签个数远大于给定的最大帧长度时，将场内标签分组：待命组和休眠组，并且只允许待命组标签响应，其中，待命组标签个数定为 256 个，以便与最大帧长度相适应，休眠组的标签暂不响应。每次识别周期结束后，阅读器要统计上一轮识别周期中成功识别的时隙数、碰撞的时隙数、空时隙数等信息来进行标签数目估算，根据估算结果来确定帧长度以及决定下一帧中是否要进行分组操作。因此，只要有标签处于休眠状态时，场内待命状态标签数大致相当于 256 个，而帧长度 $L=256$ 为定值，因此，在每一帧的识别周期中，读写器都能最大限度地识别标签，直到标签个数低于某一阈值时，读写器不再对标签进行分组，只是动态地调整帧长度，此时 L 就不再为定值，此后标签全部处于待命状态。另外，成功识别的标签处于去活状态，不响应读写器的任何命令。

当阅读器限制了响应标签的数量时，在每一个帧识别周期的间隔中间，阅读器在广播请求信息中发送分组信息，接收到信号 0 的标签处于待命状态，可以对阅读的请求响应，若接收到 1 的静默，则暂不响应。而当估计标签的数目低于一个极限的时候，阅读器便不再对标签进行分组操作。此时，阅读器以一个帧长度和标签分组命令来广播一个阅读请求，在每一个阅读周期结束后，阅读器估算未读标签的数目并调整自己的帧长度，该过程一直持续到所有标签都被成功识别。

3.2 算法分析及数学模型

在帧时隙 ALOHA 算法中，随着标签个数的增加，系统的吞吐量呈下降趋势。假设时隙数 L ，标签总数为 n ，根据统计学的原理，有 r 个标签选择 1 个时隙的概率为

$$B_{n,1/L}(r) = \binom{n}{r} \left(\frac{1}{L}\right)^r \left(1 - \frac{1}{L}\right)^{n-r} \quad (1)$$

在一帧的识别周期后，只有一个标签的时隙数(成功识别)、没有标签的时隙数、产生碰撞的时隙数，它们的期望值分别为

$$a_1^{L,n} = L \times B_{n,1/L}(1) = n \left(1 - \frac{1}{L}\right)^{n-1} \quad (2)$$

$$a_0^{L,n} = L \times B_{n,1/L}(0) = L \left(1 - \frac{1}{L}\right)^n \quad (3)$$

$$a_k^{L,n} = L - a_0^{L,n} - a_1^{L,n} \quad (4)$$

定义 系统的吞吐量

$$S = \frac{\text{成功标识的标签数}}{\text{帧长度}} = \frac{a_1^{L,n}}{L} = \frac{n}{L} \left(1 - \frac{1}{L}\right)^{n-1} \quad (5)$$

对式(5)求导：

$$\frac{dS}{dn} = \frac{1}{L} \left(1 - \frac{1}{L}\right)^{n-1} + \frac{n}{L} \left(1 - \frac{1}{L}\right)^{n-1} \ln \left(1 - \frac{1}{L}\right) = \frac{1}{L} \left(1 - \frac{1}{L}\right)^{n-1} \left[1 + n \ln \left(1 - \frac{1}{L}\right)\right] = 0 \quad (6)$$

解方程得：

$$n = \left[\frac{1}{-\ln \left(1 - \frac{1}{L}\right)} \right] \quad (7)$$

最佳帧长度：

$$L = \frac{1}{1 - e^{-\frac{1}{n}}} \quad (8)$$

当 n 的值很大时，使用泰勒级数简化式(8)得：

$$L \approx \frac{1 + 1/n}{1 + 1/n - 1} = n + 1, \quad n \gg 1 \quad (9)$$

以上推导证明：当待识别标签数与帧长度基本相当时，系统吞吐量最大，即一个帧长度识别周期中能够成功识别的标签数最多。图 2 给出了 L 取不同值时系统吞吐率的仿真结果。

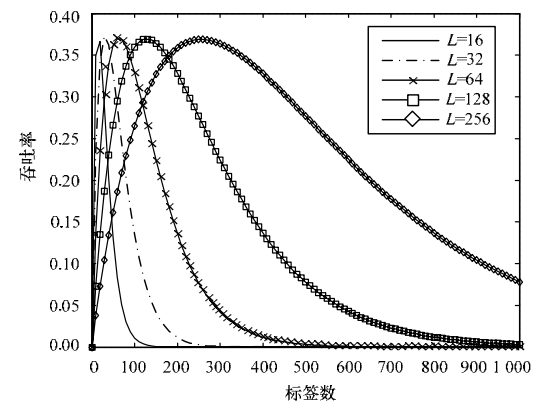


图 2 帧长度取不同值时的系统吞吐量

另一方面，读写器能设定的时隙数通常是定值，如 1, 8, 16, 32, 64, 128, 256。因此，读写器根据上一轮识别过程结束后，剩余未识别标签个数中选择 1 个数作为下一帧的长度，具体选择标准：当碰撞的时隙数高于 70% 的总时隙数时，下一帧长度加倍；当空时隙数高于 30% 的总时隙数时，下一帧

长度减半；当标签数远大于 L_{\max} 时，对标签进行分组处理。不同标签个数对应的帧长度和分组情况如表 1 所示。这样就可以在估算出未识别标签数后，在下一帧中选择匹配的时隙数，从而获得最佳系统吞吐率。

表 1 不同标签个数对应的帧长度和分组情况

帧长度(L)	未识别标签个数(n)	是否分组
256	> 256	是
256	177~256	否
128	89~176	否
64	46~88	否
32	23~45	否
16	12~22	否
8	6~11	否

标签的估算方法有很多，可以估算参与识别的标签总数，也可以直接估算出未识别的标签数，本文采用后一种方法。通过比较实际的成功、空闲、冲突时隙数与理论的成功、空闲、冲突时隙数得出误差最小的结果来预测未知标签数，即

$$\varepsilon(L, c0, c1, ck) = \min_N \left(\begin{matrix} c0 \\ c1 \\ ck \end{matrix} - \begin{matrix} a0 \\ a1 \\ ak \end{matrix} \right) \quad (10)$$

其中， $c1, c0, ck$ 为实际测得的成功、空闲、冲突时隙数值； $a1, a0, ak$ 为其对应的期望值。在标签数 n 取值范围 $[c1+2 \times ck, \dots, 2 \times (c1+2 \times ck)]$ 内找到最小的 ε 值^[3]，所对应的 n 值就是预测的标签数。

得到 n 值后，由式(11)计算出 c 值，若 $c=0$ ，则对标签进行分组；若 $c \neq 0$ ，则不分组。

$$c = \left\lfloor \frac{n}{L_{\max}} \right\rfloor, \quad L_{\max} = 256 \quad (11)$$

4 仿真分析及结论

将本文提出算法与 BFSA 算法、DFSA 算法进行比较，标签个数取 0~1 000，通过对比不同标签个数下读写器成功识别所有标签所需的总时隙数，来比较算法效率。假设一帧最大时隙数取 256，BFSA 算法的固定帧长度为 256，DFSA 的帧长度为 16~256，选择标准如表 1 所示。本文算法初始值 $L_{\min}=16$ ^[4]。当在一帧的识别周期中，如果没有标签被读写，那么可以认为场内所有标签都被成功识别并结束仿真。为了提高仿真精度，本文进行了 500 次仿真取其平均值。

可见，BFSA 算法和 DFSA 算法随着标签个数的增加，所需的时隙数呈指数增加，而本文算法呈线性增长。主要原因是当标签数量巨大时，它能动态地调整参与识别的标签数量，使之与帧长度相匹配，这样便能保持系统的平均吞吐率

在 36.8% 左右。当标签数量小于 256 后，本文算法的规则等同于 DFSA 算法。仿真结果比较如图 3 所示。

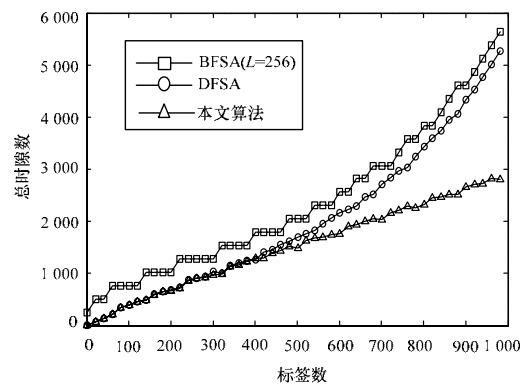


图 3 仿真结果比较

在 0~256 的范围内，2 种算法数据是重合的。本文提出的算法在识别大量标签的情况下，相比前 2 种算法效果较好，特别当标签个数达到 1 000 时，本文算法较前 2 种算法，性能可以至少提高 80%。

本文通过统计总时隙数来评估算法，其最终结果很接近实际情况下整个识别过程花费的时间。其原因是一个时隙对应的时间是固定的，读写器只在一帧读写周期开始前才发送命令，未引入其他的时间消耗。

5 结束语

传统 RFID 防碰撞算法识读标签时需要的时隙数随标签数目的增加呈指数增加，本文提出的算法通过限制响应标签数量解决此问题，且其执行过程简单，可以使系统效率达到最大理论值。

参考文献

- [1] Finkenzeller K. RFID Handbook, Fundamentals and Applications in Contactless Smart Cards and Identification[M]. 2nd ed. Chichester, UK: John Wiley and Sons Ltd., 2003.
- [2] Finkenzeller K. 射频识别技术[M]. 3 版. 吴晓峰, 陈大才, 译. 北京: 电子工业出版社, 2006.
- [3] Vogt H. Multiple Object Identification with Passive RFID Tags[C]// Proc. of IEEE International Conference on Systems, Man and Cybernetics. Piscataway, USA: [s. n.], 2002: 6-9.
- [4] 吴晶, 熊璋, 王晔. 利用动态时间槽分配的多目标防冲突射频识别[J]. 北京航空航天大学学报, 2005, 31(6): 618-622.

编辑 陆燕菲

(上接第 266 页)

参考文献

- [1] Blunden B. Memory Management: Algorithms and Implementation in C/C++[M]. [S. l.]: Wordware Publishing Inc., 2002.
- [2] Hu Zhe, Zhang Jun, Luo Xiling. Design and Realization of Efficient Memory Management for Embedded Real-time Application[C]// Proc. of the 6th International Conference on ITS Telecommunications. Chengdu, China: [s. n.], 2006: 174-177.

- [3] Yi Sangho, Lee Seungwoo, Cho Yookun. SESAME-P: Memory Pool-based Dynamic Stack Management for Sensor Operating Systems[C]//Proc. of the 4th IEEE International Conference on Distributed Computing in Sensor Systems. Santorini Island, Greece: [s. n.], 2008: 544-549.
- [4] 王泽民, 芦东昕, 徐立峰, 等. 嵌入式系统软件内存泄漏监测的算法和实现[J]. 计算机工程, 2005, 31(13): 84-86.

编辑 陆燕菲