

CHI 标签估计下自适应帧长调整 DFSA 算法

卢迪, 李绅龙, 许成舜

(哈尔滨理工大学 电气与工程学院, 黑龙江 哈尔滨 150080)

摘要: 标签碰撞问题在 RFID 技术中很常见, 对整个系统效率影响很大, 如何有效解决这个问题对于整个系统十分重要. 针对目前常用的 DFSA 算法, 将准确性高、复杂性低的 CHI 标签数量估计法与自适应的帧长及分组数调整方法结合起来, 构造出一个解决标签碰撞问题的方法, 并给出理论推导过程. 仿真结果显示 CHI 估计算法错误率低, 帧长和分组数调整方法有效地提高了系统吞吐率和稳定性, 构造的方法提升了解决标签碰撞问题的效率.

关键词: RFID; 防碰撞; 自适应; 吞吐率

DOI: 10.15938/j.jhust.2015.01.011

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1007-2683(2015)01-0056-05

Adaptive Frame Size and Group Division in DFSA Algorithms on the Basis of CHI Tags' Estimation

LU Di, LI Shen-long, XU Cheng-shun

(School of Electrical and Electronic Engineering, Harbin University of Science and Technology, Harbin 150080)

Abstract: Anti-collision is a very common problem in RFID systems, it has a great influence on the whole system efficiency. How to solve this problem effectively is very important for the entire system. For the most commonly used algorithms based on DFSA, Combines CHI which are both high accuracy and low complexity in estimating the tags to be identified with adaptive frame size and group division, Constructing a method to solve the problem of tag collision and giving the theoretical derivation. Simulation results show that the error rate of CHI is low, and adaptive frame size and group division can effectively enhance the system's throughput and its stability, the method under construction improves the efficiency to solve the problem of tag collision.

Key words: RFID; Anti-collision; adaptive; throughput

0 引言

射频技术(radio frequency identification, RFID)是自动识别技术在无线通信方面的应用. RFID 技术具有识别精度高、操作速度快、读写范围广、数据存储量大、可加密数据、环境适应能力强、使用寿命长等许多优点, 它显著提高了管理维护、跟踪、定位等

用途的处理速度^[1-3]. 然而, 对多个目标进行识别时数据传输量很大, 不可避免的会产生冲突即碰撞, 这种碰撞使得系统的工作效率受到严重影响. 因此, RFID 系统中提高吞吐率的关键是如何有效解决碰撞问题^[4-5].

基于时隙的 ALOHA 算法是电子标签防碰撞算法中常用的一种方法, 其中动态帧时隙 ALOHA(dynamic framed slotted aloha, DFSA) 算法实现简单、性

收稿日期: 2014-10-08

作者简介: 卢迪(1971—), 女, 博士, 教授, 硕士研究生导师;

李绅龙(1988—), 男, 硕士研究生, E-mail: lishenlong000@126.com;

许成舜(1987—), 男, 硕士研究生.

能较好,能够适应电子标签数量不同的场合,具有较高的电子标签吞吐率,被广泛应用于 RFID 系统中^[6-8]. 在 DFSA 算法中,帧长能够随阅读区域中电子标签的数量变化而动态改变,当标签数量和时隙数量相等时系统效率最佳,可以根据每次识别周期开始前估算出的标签数量确定帧的长度. 如果待识别标签量比帧长大就增加帧长度^[9-10];反之,则减小帧长度. 而由于成本要求标签寄存器最高为 8 位,导致标签数量过多时难以解决碰撞问题,所以标签数大于 256 时需要将标签进行分组识别^[11]. 因此在 DFSA 算法中,确定帧长及分组数是提高电子标签吞吐率的关键技术. 为解决这个问题,本文将 CHI 估计法与自适应的帧长及分组数调整方法结合起来,从而提高系统效率中^[12].

1 算法介绍

1.1 标签数量估计

常见的标签估计方法有下限值法(low bound, LB)、Schoute 估计法和 CHI 估计法(chebyshev's inequality, CHI) 等. LB 法中假设前一识别周期有 C 个碰撞时隙, LB 以每个碰撞时隙最少有两个碰撞标签来估计,即用 $N = 2C$ 来估计阅读范围内未识别的标签数量; Schoute 法在 LB 基础上做了改进,考虑到每个时隙标签碰撞数可能大于 3 个,用 $N = 2.39C$ 来估计待识别标签数量^[13-14]. CHI 方法将期望值与实际执行得到的结果相结合得出 n 值,使实际时隙值 C_e, C_s, C_c 和理论期望值 N_e, N_s, N_c 间的差距最小,即式(1)在标签数 n 取值范围 $[C_s + 2C_c, \dots, 2 \times (C_s + 2 \times C_c)]$ 内找到最小的 ε 值,所对应 n 值即为预测的标签数^[15-17]. 其中: N 为帧长, n 为阅读器有效读取范围内的标签数量总和, n 个标签在帧长大小为 N 时的系统吞吐率为 P_N , r 是一帧内成功识别的时隙数, r 的期望值为 N_s 、空时隙数的期望值为 N_e 、碰撞时隙数的期望值为 N_c .

$$\varepsilon = \min_n \left| \begin{pmatrix} N_e \\ N_s \\ N_c \end{pmatrix} - \begin{pmatrix} C_e \\ C_s \\ C_c \end{pmatrix} \right| \quad (1)$$

图1中对 LB 法、Schoute 法、CHI 法 3 种估计方法的错误率进行仿真. 设 n_0 为标签数目的估计值, 错误率 $\eta = |(n - n_0)/n|$, 标签数范围为 1 ~ 500. 当标签数量较小时, 3 种方法的错误率都较小; 在标签数大于 150 时, LB 法和 Schoute 法错误率线性增

加; CHI 法估计的错误率一直稳定在 5% 以下. 图2是 DFSA 算法在 3 种估计方法下吞吐率情况. 使用 CHI 法估计的吞吐率在标签数约为 120 处达最大值 35%; 使用 LB 和 Schoute 估计的吞吐率在标签数约为 80 处达到最大值 27%. 当标签数量增多时, CHI 法估计的吞吐率要比另外两种方法变化缓慢, 在标签数量约为 500 时, 约达到 28%, 而 LB 和 Schoute 估计的吞吐率变化的相对较快, 在标签为 500 时仅为 15%.

考虑到 LB 法以及 Schoute 法在多标签情况下效率低且误差较大, 本文采用吞吐率和复杂程度都比较合适的 CHI 估计方法.

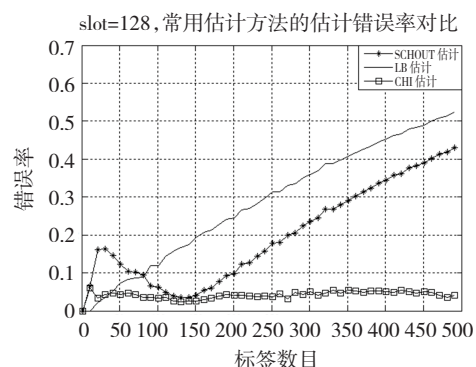


图1 3种估计方法错误率

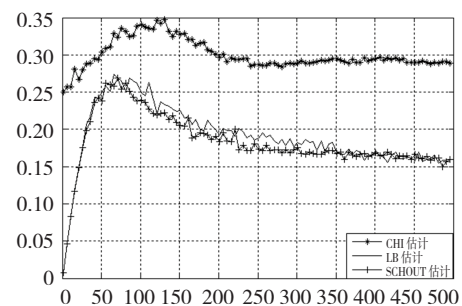


图2 3种估计方法下 DFSA 吞吐率

1.2 标签帧长及分组调整

目前动态调整帧长的方法(如 DFSA 算法)为在已知待识别标的标签数时,按照式(1)结果偏差最小的方式调整帧长,如果标签数大于 2^k 且小于 2^{k+1} (k 取正整数),则取帧长为 2^k ,即采用向下取整法. 它的缺点在于划分不清楚、计算量较大,如当标签数为 17 时设定帧长为 32,而理论上为当帧长取 16 时效果最佳. 由于碰撞具有随机性,导致该方法的系统稳定性很差,而且帧长初始值同样对系统性能有很大影响^[18].

为达到系统最佳性能,需要每个识别周期取和

标签数最近的帧长大小. 不同帧长系统吞吐率变化曲线如图3所示, 求出相邻帧长曲线的交点值, 该值当作调整帧长大小或分组数的临界值, 当估计的待识别标签数量到达该临界点就自适应的调整相应帧长大小或分组数^[19-20].

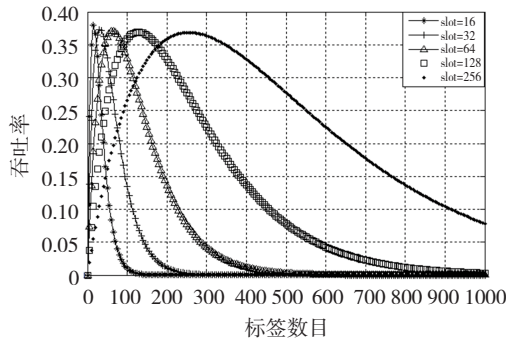


图3 不同帧长系统吞吐率

1 个标签在帧长为 N , 未读标签数为 n 时占用的时隙数 N_s 为:

$$N_s = Nn \left(\frac{1}{N} \right) \left(1 - \frac{1}{N} \right)^{n-1} \quad (2)$$

则 n 个标签在帧长为 N 时吞吐率为:

$$P_N = \frac{N_s}{N} = n \left(\frac{1}{N} \right) \left(1 - \frac{1}{N} \right)^{n-1} \quad (3)$$

可得到两相邻帧性能曲线交点 $P_N = P_{2N}$ 处的标签数为:

$$n_{N,2N} = \frac{\ln(2)}{\ln\left(\frac{2N-1}{2N-2}\right)} \quad (4)$$

因此, 若估计出的标签数超过 $n_{N,2N}$ 就将帧长增大一倍, 若标签数低于 $n_{0.5N,N}$ 则帧长减小一半. 同理在计算分组数时, 将标签划分为 m 组与 $m+1$ 组时相邻组性能曲线的交点值为:

$$p = \left(\frac{n}{m} \right) \left(\frac{1}{256} \right) \left(1 - \frac{1}{256} \right)^{\frac{n}{m}-1} = \left(\frac{n}{m+1} \right) \left(\frac{1}{256} \right) \left(1 - \frac{1}{256} \right)^{\frac{n}{m+1}-1} \quad (5)$$

通过求解式(5), 可以得到式(6)所示的 n 与 m 之间的关系.

$$n \sim m(m+1) \frac{\ln(m+1) - \ln(m)}{\ln(256) - \ln(255)} \quad (6)$$

估计出标签个数 n 后, 将 $m=1, 2, 3, 4, 5 \dots$ 分别代入式(6)中, 令式(6)结果与 n 值之差最小, 此时式(6)结果对应的 m 值作为此次识别周期的分组个数. 本文方法与常用的 DFSA 算法帧长和分组数调整方案如表1所示.

表1 本文方法与 DFSA 算法帧长和分组调整

待识别 标签数	本文方法 最佳帧长	本文方法 分组数	DFSA 算法 最佳帧长	DFSA 算法 分组数
1	2	1	1	1
2 ~ 3	4	1	2	1
4 ~ 7	8	1	4	1
8 ~ 11	8	1	8	1
12 ~ 15	16	1	8	1
16 ~ 22	16	1	16	1
23 ~ 31	32	1	16	1
32 ~ 44	32	1	32	1
45 ~ 63	64	1	32	1
64 ~ 88	64	1	64	1
89 ~ 127	128	1	64	1
128 ~ 176	128	1	128	1
177 ~ 255	256	1	128	1
256 ~ 354	256	1	256	2
355 ~ 511	256	2	256	2
512 ~ 622	256	2	256	3
623 ~ 883	256	3	256	3
884 ~ 1023	256	4	256	3
1024 ~ 1141	256	4	256	4
⋮	⋮	⋮	⋮	⋮

1.3 算法实现过程

1) 无源标签进入阅读器有效读取范围后, 限定系统最大帧长为 256. 阅读器设置初始帧长并向其有效识别范围内的标签发出请求阅读指令, 开始第一个识别周期, 标签随机选定一个时隙响应.

2) 阅读器判断标签的响应情况, 对被识别标签开始读取信息, 成功读取后向其发送一个停止响应指令, 则该标签进入到沉默状态且本次阅读周期不再被识别.

3) 一个阅读周期结束后, 阅读器计算系统效率并采用 CHI 法估计剩余标签数量, 如果为 0 则转到 5); 如果超过 354 个则转到 4); 若少于 354 个则根据表1 选择合适帧长转到 1), 开始新一轮阅读周期.

4) 根据表1, 帧长定为 256 并对标签分组, 然后每组分别完成识别过程, 转到 1) 开始识别.

5) 若估计出剩余标签数量为零, 则再进行一个阅读周期, 保证所有标签均被成功识别后, 结束本次识别.

算法实现流程图如图4所示, 标签分组及帧长

调整如表 1 所示.

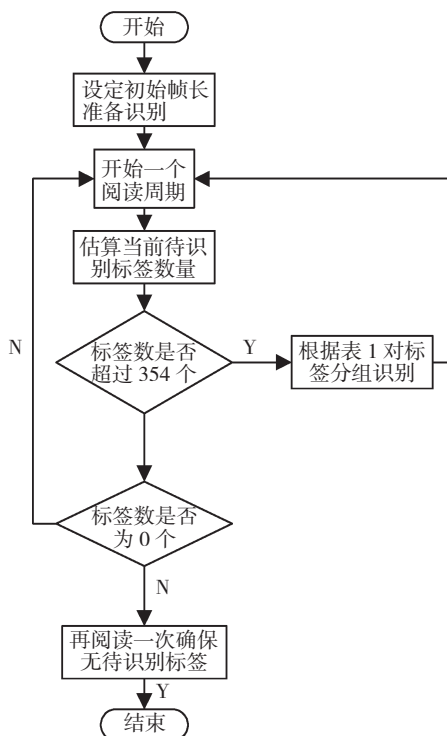


图 4 算法实现流程图

2 仿真结果与分析

用 Matlab 对算法进行性能仿真,为使得仿真结果有比较性,每次仿真设定初始帧长为 128.

图 5 是全分组算法、EDFSA、GDFSA 以及本文方法吞吐率情况,标签数目在 3 500 张以内时,本文方法吞吐率稳定在 38% 左右,当标签数量增多吞吐率也不会下降.在标签数大于 1 000 时,EDFSA 的吞吐率有所降低,而在标签数达 3 500 时,吞吐率则维持在 30% 以上. GDFSA 算法吞吐率保持稳定且较高,但基本不超过 35%. 图 6 是全分组算法、EDFSA、GDFSA 以及本文方法消耗时隙数情况,4 种算法时隙增加都比较缓慢,在标签数目小于 1 000 时,这些算法消耗的时隙基本相同;而当标签数大于 1 000 以后,本文方法消耗时隙数明显低于其它算法.

当标签数量较少时,常用调整方法和自适应的帧长调整方法的系统性能相差很小,由于后者在每一识别周期都精确采用理论上的最佳帧长使系统稳定性得到提高,避免了因采用 DFSA 算法等方案时帧长划分方法不当导致的问题.采用本文方法标签吞吐率和系统稳定性效果都有所提高.

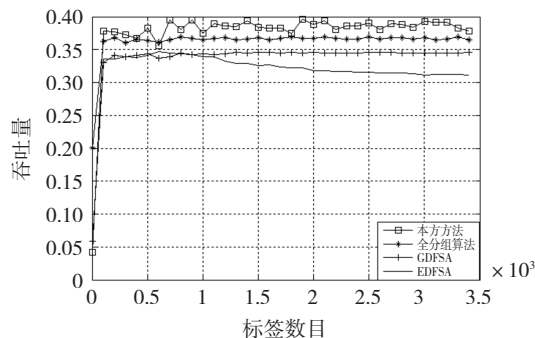


图 5 本文方法与其它算法吞吐率

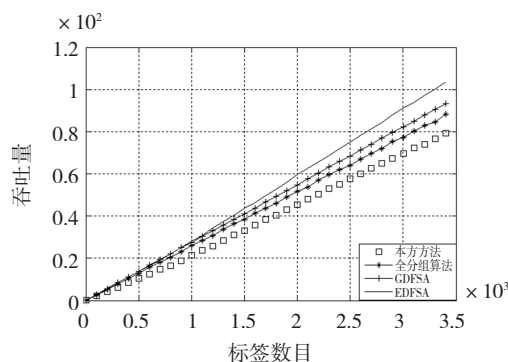


图 6 本文方法与其它算法使用时隙数

3 结 语

本文将 CHI 估计法与自适应的帧长分组调整方法结合起来,对无源超高频 RFID 技术中的标签防碰撞算法进行改进.改进后的系统提高了系统吞吐率而且不受标签数量的限制.与现有的 DFSA 算法、EDFSA 算法和 GDFSA 算法相比改善了系统识别效率,提高吞吐率到 38% 左右.仿真结果表明本文方法使得系统开销和算法性能达到了较好的平衡.

参 考 文 献:

- [1] 王永. 基于 EDFSA 算法的改进研究 [J]. 计算机与数学工程, 2011, 39(3): 17-18.
- [2] 陈燕, 李娜娜, 张娜. 一种改进的动态帧时隙 ALOHA 算法 [J]. 郑州轻工业学院学报: 自然科学版, 2012, 27(6): 41-42.
- [3] 陈荣伶, 王玉峰, 刘伟, 等. 基于标签分组的 RFID 防碰撞算法 [J]. 计算机应用, 2013, 33(8): 2132-2133.
- [4] 王飞, 张武. 基于分组的动态时隙 ALOHA 算法 [J]. 计算机系统应用, 2013, 22(7): 77-78.
- [5] 张学军, 田侃, 王锁萍. 增强型自适应时隙数防碰撞算法研究 [J]. 南京邮电大学学报: 自然科学版, 2012, 32(1): 59-60.
- [6] 王萌, 陈殿仁. RFID 系统多标签识别碰撞算法 [J]. 太赫兹科学与电子信息学报, 2013, 11(4): 601-602.

- [7] 舒远仲,田蕾,张丽,等. 基于排队理论的时隙 ALOHA 防碰撞算法[J]. 计算机工程与设计,2013,34(7):2632-2633.
- [8] 周朝阳. 射频识别系统冲突防范算法[J]. 计算机系统应用,2013,22(10):133-134.
- [9] 徐艳群,张斌,秦小铁. 基于动态时隙 ALOHA 的多标签识别防碰撞算法[J]. 西南师范大学学报,2013,38(5):23-24.
- [10] 谭海波,景凤宣. 基于 RFID 电子标签的 ALOHA 防冲突算法研究[J]. 福建电脑,2013,4:10-11.
- [11] 宋瑞玲,高仲合. 基于分组动态帧时隙 ALOHA 防碰撞算法研究[J]. 通信技术,2013,7(46):37-38.
- [12] 徐劼,林菁,李绍香. 一种高效低成本的 RFID 系统防碰撞改进算法[J]. 计算机应用与软件,2013,30(7):124-125.
- [13] 刘永,熊兴中,李晓花. RFID 防碰撞技术的研究[J]. 电信科学,2012,2:140-141.
- [14] 王萌,陈殿仁. RFID 系统多标签识别防碰撞算法[J]. 太赫兹科学与电子信息学报,2013,11(4):602-603.
- [15] 陈春明,冯玉田,付良成. RFID 动态帧时隙防碰撞改进算法研究[J]. 通信与网络,2013,39(1):86-87.
- [16] 卢迪,凌晓艳. 全分组不等长时隙的 DFSA 算法[J]. 测控技术,2013,32(9):56-57.
- [17] 潘峥嵘,尚凯. 一种改进的 RFID 动态帧时隙 ALOHA 算法[J]. 传感器与微系统,2012,31(7):141-142.
- [18] 李玮玮,郭志涛,张晶,等. 基于标签分组的 DFSA 抽样训练规则防冲突算法[J]. 计算机应用研究,2013,30(10):2969-2970.
- [19] 王玉青,李开宇,孙纯鹏. 改进动态帧时隙 ALOHA 算法[J]. 电子科技,2012,25(7):76-77.
- [20] 庞宇,彭琦,林金朝,等. 基于分组动态帧时隙的射频识别防碰撞算法[J]. 物理学报,2013,62(14):4-5.

(编辑: 关 毅)