

密集标签环境下 RFID 标签防冲突算法的分析研究*

杨 健¹, 王永华¹, 詹宜巨², 余松森³

(1. 广东工业大学自动化学院, 广东 广州 510006;

2. 中山大学工学院, 广东 广州 510006

3. 华南师范大学南海校区计算机工程系, 广东 佛山 528225)

摘 要: 分析了密集标签环境下几种传统的 RFID 标签防冲突算法的识别总耗时、系统吞吐率、读写器发送数据量和单标签发送数据量等主要性能指标, 并依此提出了一种对传统 ALOHA 算法的改进算法, 其系统吞吐率达到 0.41 左右, 读写器发送数据量和单标签发送数据量与树形算法相比均减少了一半以上, 最后指出了下一步的研究方向。

关键词: RFID; 标签防冲突; 动态帧 ALOHA; 二叉树; 查询树

中图分类号: TN91 **文献标识码:** A **文章编号:** 0529-6579(2009)06-0147-04

Research on Tag Anti-collision Algorithms of RFID System under Dense Tag Environment

YANG Jian¹, WANG Yonghua¹, ZHAN Yiju², YU Songsen³

(1. Faculty of Automation, Guangdong University of Technology, Guangzhou 510006, China);

2. School of Engineering, Sun Yat-sen University, Guangzhou 510006, China)

3. Department of Computer Engineering, Nanhai Campus, South China Normal University, Foshan 528225, China ;

Abstract: This paper firstly analyses the performances of several RFID anti-collision algorithms under the environment of dense tags. The performances discussed include total time consumed, system efficiency, total bits transmitted by the reader and average bits sent by each tag and so on. Then a novel improved algorithm based on ALOHA algorithms is present. The new algorithm has high system efficiency around 0.41 and less amount of data transmitted which is decreased by 50 percent at least compared with tree-based algorithms. Lastly, conclusions are made and directions of further work are given.

Key words: RFID; tag anti-collision; framed slotted ALOHA; binary tree; query tree

射频识别(Radio Frequency Identification, RFID)技术是从上世纪90年代兴起的一项自动识别技术,具有快速、准确、可靠性高等特点。RFID系统利用无线射频方式进行非接触双向通信,以实现和标签之间的数据交换。

RFID被动标签因性价比高而广泛应用,但其结构简单,在密集读写器/标签环境下,当多个读

写器同时读取某标签或多个标签同时与某读写器通信时,会导致所有标签数据无法读出,发生RFID冲突。RFID冲突可分为读写器冲突和标签冲突,前者是指多个读写器信号同时作用于某标签,干扰其正常工作时序,导致识别失败;而后者是指多个标签处在某读写器信号覆盖范围内,会同时尝试与该读写器通信,数据间相互干扰导致所有标签无法

* 收稿日期: 2008-12-10

基金项目: 国家自然科学基金资助项目(60673132); 广东省科技计划资助项目(2008B010200037); 广州市科技计划资助项目(2008Z1-D141)

作者简介: 杨健(1982年生),男,博士; E-mail: yangjn3@mail2.sysu.edu.cn

识别。本文主要对后者进行研究。

1 传统 RFID 标签防冲突算法

传统 RFID 标签防冲突算法主要分为 ALOHA 算法和树形算法两大类。前者代表有 DFSA (动态帧 ALOHA) 算法^[1-3]、EDFSA (增强型动态帧 ALOHA) 算法等^[4]; 后者代表有 BS (二叉树) 算法^[5]、QT (查询树) 算法和 QTI (改进的查询树) 算法等^[5-6]。衡量的主要指标有: 识别某一数量标签的总耗时、单标签功耗以及读写器功耗。前者反映 RFID 系统的系统吞吐率和反应速度, 后者则体现了标签最大作用距离及读写器功耗情况。因此本文主要考察以上算法识别某一数量标签的总耗时 T 、单标签发送的平均数据量 TB 、以及读写器发送的总数据量 RB 。

1.1 DFSA 算法 DFSA 是一种可根据标签数量动态调整帧长的 ALOHA 算法。识别开始前, 读写器估计未读标签数量 n , 并据此设置帧长 L 。标签在 L 个时隙中随机选择一个作为自身时隙存储, 并在轮询到该时隙时响应读写器, 该策略在很大程度上降低了潜在的冲突可能性。估计未读标签数量采用以下方法: 定义 N_c 为冲突时隙数, N_s 为单标签时隙数, C_r 为 N_c 与 n 的比值并可由式 (1) 计算, $N_c/C_r - N_s$ 即是当前帧结束后剩余的标签数量, 以此值设置下帧帧长可使系统吞吐率达到最大。

$$C_r = 1 - \left(1 - \frac{1}{L}\right)^n \cdot \left(1 + \frac{n}{L-1}\right) \quad (1)$$

若 T_s 为单个时隙时间长度, L_i 为初始帧长, L 为实际帧长, L_b 为帧长长度, $P_s(k)$ 为标签经 k 次被识别的概率, QR_b 为时隙间隔标志长度, ACK_b 为读写器收到标签 ID 后确认信号长度, ID_b 为标签 ID 长度, 则 DFSA 算法的主要指标如下:

$$T = T_s \cdot (L_i + L \cdot \sum_{k=1}^{\infty} kP_s(k)) \quad (2)$$

$$RB = L_b \cdot \left(\sum_{k=1}^{\infty} kP_s(k) + 1\right) + QR_b \cdot L \cdot \sum_{k=1}^{\infty} kP_s(k) + n \cdot ACK_b \quad (3)$$

$$TB = ID_b \cdot \sum_{k=1}^{\infty} kP_s(k) \quad (4)$$

1.2 EDFSA 算法 EDFSA 在 DFSA 基础上增加了对标签的预分组, 改善标签数量庞大时的系统性能。在完成对未读标签数量的估计后, 根据吞吐率最大条件将标签分为若干组, 逐组进行识别, 每组都相当于一个已定帧长的 DFSA。EDFSA 采用文献 [3-4] 中的标签数量估计方法, 定义向量 $c = \{c_0, c_1, c_k\}$ 为上轮识别过程中的空时隙数、单个

标签的时隙数和冲突时隙数; 定义 $i = \{i_0, i_1, i_k\}$ 为理论的空时隙数、单个标签时隙数和冲突时隙数。由切比雪夫不等式, 当向量 i 和 c 的距离 ε 达到最小时, n 的取值即被看作标签数量的最优估计值, 即:

$$\varepsilon(N, c_0, c_1, c_k) = \min_n \left\| \begin{pmatrix} i_0 \\ i_1 \\ i_k \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right\| \quad (5)$$

若 T_s 为单个时隙时间长度, L_i 为初始帧长, L 为实际帧长, G_i 为初始分组数, G 为实际分组数, G_b 为分组长度, $P_s(k)$ 为标签经 k 次被识别的概率, QR_b 为时隙间隔标志长度, ACK_b 为读写器收到标签 ID 后确认信号长度, ID_b 为标签 ID 长度, 则 EDFSA 算法的主要指标为:

$$T = T_s \cdot (G_i \cdot L_i + G \cdot L \cdot \sum_{k=1}^{\infty} kP_s(k)) \quad (6)$$

$$RB = G_b \cdot L_b \cdot \left(\sum_{k=1}^{\infty} kP_s(k) + 1\right) + QR_b \cdot G \cdot L \cdot \sum_{k=1}^{\infty} kP_s(k) + n \cdot ACK_b \quad (7)$$

$$TB = ID_b \cdot \sum_{k=1}^{\infty} kP_s(k) \quad (8)$$

1.3 BS 算法 在 BS 中标签当其内部计数器的值为 0 时响应读写器。读写器根据标签响应情况有三种不同的操作: 1. 发生标签冲突, 将冲突标签内部计数器随机加 '0' 或 '1', 未冲突的未读标签内部计数器加 1; 2. 有唯一标签响应, 识别该标签, 并将未读标签内部计数器减 1; 3. 无标签响应, 将未读标签的内部计数器减 1。直到识别所有标签。

若 T_c 为读周期长度, C_b 为读指令长度, ID_b 为标签 ID 长度。则 BS 算法的主要指标为:

$$T = T_c \cdot$$

$$\left(1 + 2 \cdot \sum_{k=0}^{\infty} \left\{2^k - (2^k + n - 1) \cdot (1 - 2^{-k})^{n-1}\right\}\right) \quad (9)$$

$$RB = C_b \cdot$$

$$\left(1 + 2 \cdot \sum_{k=0}^{\infty} \left\{2^k - (2^k + n - 1) \cdot (1 - 2^{-k})^{n-1}\right\}\right) \quad (10)$$

$$TB = ID_b \cdot \log_2 n \quad (11)$$

式 (9)、式 (10) 为树的深度 k 趋向无穷大时的一般表达式, 为简化计算还可以进一步推导出 T 和 RB 的下限。BS 识别过程为严格二叉树, 当其为完全二叉树时的中间结点数 $C_F(n)$ (总冲突次

数) 最小, 由于严格二叉树的总结点数 $N_n(n)$ (读写器发出的读周期总数) 为 $2C_F(n) + 1$, 即 $N_n(n) \geq 1 + 2C_F(n) = 1 + 2 \sum_{k=0}^{\log_2 n - 1} 2^k = 2n - 3$, 可得:

$$T_B \geq T_c \cdot (2n - 3) \quad (12)$$

$$RB \geq C_b \cdot (2n - 3) \quad (13)$$

1.4 QT 算法 在 QT 中标签何时响应读写器是由查询前缀和标签 ID 共同决定的。读写器广播给标签的读指令中包含了一个查询前缀, 标签接收到后将其与自身 ID 比较, 若包含该前缀则响应读写器。若有多个标签同时响应, 则将查询前缀长度增加 1 位, 并按先左孩子结点后右孩子结点的顺序将此两个新查询前缀依次向标签广播, 若新的查询前缀仍导致冲突, 则重复以上过程, 直到有唯一标签响应。读写器完成对标签的识别后更改查询前缀为兄弟结点, 若无或已为右孩子结点则返回父结点, 重复以上过程直至识别出所有标签。需要注意的是, 与 BS 相比 QT 中所有标签冲突只发生在中间结点处, 而不包括根结点。

若 T_c 为读周期长度, ID_b 为标签 ID 长度。则 QT 算法的主要指标为:

$$T \leq T_c \cdot (2.887n - 1) \quad (14)$$

$$RB \leq ID_b \cdot (2.887n - 1) \quad (15)$$

$$TB = ID_b \cdot (2.21 \log_2 n + 4.19) \quad (16)$$

1.5 QTI 算法 由于 QT 中存在着大量的无效查询前缀, 既增加了读写器的无效读周期数, 又降低了系统吞吐率。QTI 对 QT 会引起冲突的查询前缀进行了预过滤。如: 查询前缀 “q” 已引起标签冲突, 而 “q0” 又已无标签响应, 则判定 “q1” 为必然导致标签冲突的无效查询前缀, 滤除 “q1” 并直接以 “q10” 作为下一读周期的查询前缀。

QTI 与 QT 相比仅降低了读写器发送数据量, 标签发送数据量并没有改变。若 T_c 为读周期长度, ID_b 为标签 ID 长度。则 QTI 主要指标为:

$$T \leq T_c \cdot (2.665n - 1) \quad (17)$$

$$RB \leq ID_b \cdot (2.665n - 1) \quad (18)$$

$$TB = ID_b \cdot (2.21 \log_2 n + 4.19) \quad (19)$$

2 对现有算法的改进

DFSA、EDFSA 等动态帧 ALOHA 算法本质上都是基于竞争时隙, 因此理论识别率小于 100%; BS、QT 和 QTI 等树形算法本质上基于二叉树, 在一个标签成功识别前不会转而识别其他标签, 理论上可达到无遗漏识别, 但增加了读写器与标签之间

的数据量。这两种算法都存在着时隙浪费严重的现象: 前者在识别后期存在大量空闲时隙浪费, 这是由于帧长和分组数不能随标签逐渐识别而实时调整造成; 后者在识别前期响应的标签数量远大于时隙数量而导致大量的冲突时隙, 而 QT、QTI 还受标签 ID 分布影响较大。

为此, 我们提出了一种同时减小初期冲突时隙和后期空闲时隙数量的 ALOHA 改进算法 (Improved ALOHA, I-ALOHA), 首先对未识别标签的数量进行估计, 由此选取最佳的分组数和帧长, 确保最大的系统吞吐率。对于冲突时隙中的标签, 则采用类似 BS 中的二叉树加以识别。为了深入研究 I-ALOHA 的总耗时和通信复杂度, 我们先进行以下设置: 设初始分组数 G_i 、实际分组数 $G(n)$, 位数为 G_b ; 初始帧长 L_i 、实际帧长 $L(n)$, 位数为 L_b ; 单标签时隙的概率 P_s , 冲突时隙的概率 P_c , 空闲时隙的概率 P_i , 单个标签发送概率 p ; 时隙间隔标志位数为 QR_b ; 读写器向可识别标签发送 ID 请求数据位数为 IDR_b ; 二叉树递归识别过程读写器单个读周期指令位数为 R_b ; 标签发送的测试数据位数为 IDT_b , 标签全部落在二叉树相同结点中的概率不大于 ε 。限于篇幅, 这里仅给出了各指标的最终表达式。

$$T \approx \{G_i \cdot L_i + G(n) \cdot L(n) + 2.7844 \cdot G(n) \cdot L(n) \cdot P_c\} \cdot T_c \quad (20)$$

其中

$$P_c = \left[1 - \frac{n}{G(n)} \cdot p(1-p)^{\frac{n}{G(n)}-1} - (1-p)^{\frac{n}{G(n)}}\right]$$

$$RB \approx 2(G_b + L_b) + [G_i L_i + G(n) L(n)] \cdot QR_b + n \cdot IDR_b + 0.3922 G(n) L(n) P_c R_b \quad (21)$$

TB 满足在一范围内, 上限:

$$\frac{G(n) \cdot L(n)}{n} \cdot [(2.3922 \cdot P_c \cdot \log_2 2.3922 + P_s) \cdot IDT_b] + ID_b \quad (22)$$

下限:

$$\frac{G(n) \cdot L(n)}{n} [P_c \cdot (\log_2 2.3922 \cdot \log_2 \varepsilon^{-1} + 0.5820) + P_s] \cdot IDT_b + ID_b \quad (23)$$

$$\text{其中 } P_s = \frac{n}{G(n)} \cdot p(1-p)^{\frac{n}{G(n)}-1}。$$

3 性能分析及对比

图 5~7 分别为 I-ALOHA 和各传统算法的 S 、 RB 以及 TB 的比较。

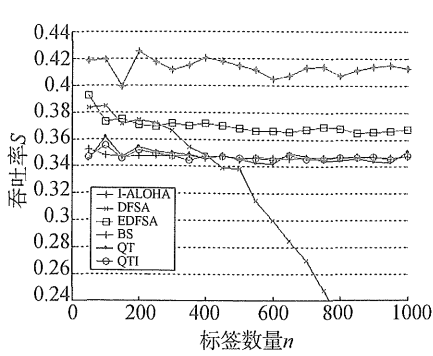


图 5 I-ALOHA 与各算法 S 比较

Fig. 5 The comparison of S

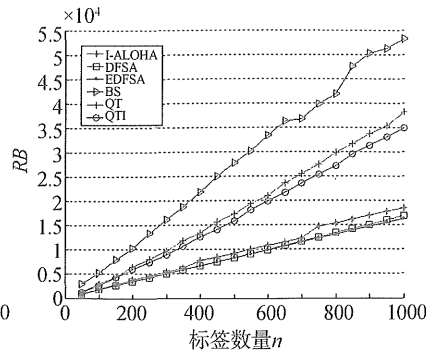


图 6 I-ALOHA 与各算法 RB 比较

Fig. 6 The comparison of RB

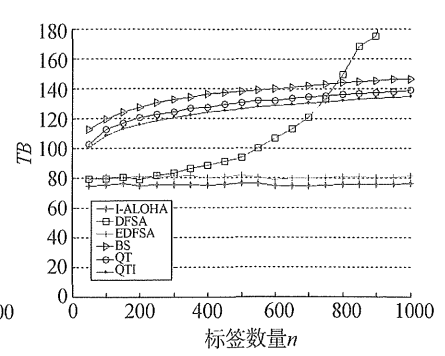


图 7 I-ALOHA 与各算法 TB 比较

Fig. 7 The comparison of TB

由图 5 可知, I-ALOHA 的吞吐量 S 远高于其他算法, 达到 0.41 左右。DFSFA 在标签数量较小时的 S 和 EDFSA 相似, 但当标签数量超过最大帧长时, 冲突时隙数量呈指数增长, S 急剧恶化; BS、QT 和 QTI 的 S 近似且随标签数量变化不大。

图 6 为 I-ALOHA 与各传统算法的 RB 比较。其中, 树形算法 RB 明显较 ALOHA 算法大, 因为树形算法识别初期发生大量的标签冲突; ALOHA 算法通过将每帧分为多个时隙, 降低了单个时隙内发生标签冲突的可能性, 因此 DFSFA 和 EDFSA 的 RB 达到最小。I-ALOHA 首轮识别采用了最优分组和帧长选取策略, 降低了初期读写器重发的概率, 因此其 RB 与树形算法相比减少了一半以上。

图 7 为 I-ALOHA 与各传统算法 TB 的比较。其中, 树形算法的 TB 最大, 其识别初期的大量冲突导致标签重发数据的次数较多, 因而单标签发送的数据量也较多; I-ALOHA 的 TB 最小, 仅为树形算法的 50% 左右, 且相对稳定; ALOHA 算法的 TB 也较理想, 但随着标签数量的增大, DFSFA 的 TB 由近似 EDFSA 到急剧增长, 这是由于标签数量超过其最大帧长所导致。

4 结论及展望

本文针对 ALOHA 算法和树形算法各自缺陷, 提出了一种 ALOHA 改进算法, 其吞吐率高达 0.41 左右, 通信复杂度也相对较低, 综合了两类算法的优点。但该算法还未考虑到标签识别过程中大量移

动标签快速进出读写区域对算法性能产生的影响, 而这可以作为下一步的研究方向。

参考文献:

- [1] CHA J R, KIM J H. Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID System [C] // 2006 3rd IEEE Consumer Communications and Networking Conference, 2006: 768 - 772.
- [2] CHA J R, KIM J H. Novel anti-collision algorithms for fast object identification in RFID system [C] // Proceedings of the 2005 11th International Conference on Parallel and Distributed System, 2005: 63 - 67.
- [3] VOGT H. Multiple object identification with passive RFID tags [C] // 2002 IEEE International Conference on System, Man and Cybernetics, 2002: 6 - 9.
- [4] LEE S R, JOO S D, LEE C W. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag [C] // Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005: 166 - 172.
- [5] MYUNG J, LEE W, SRIVASTAVA J. Tag-splitting: adaptive collision arbitration protocols for RFID tag identification [J]. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(6): 763 - 775.
- [6] LAW C, LEE K, SIU K. Efficient memoryless protocol for tag identification [C] // Proc. Fourth Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm., 2000: 75 - 84.