

# 改进型自适应多叉树防碰撞算法研究

张学军,蔡文琦,王锁萍

(南京邮电大学电子科学与工程学院,江苏南京 210003)

**摘要:** 文章在自适应多叉树防碰撞算法的基础上,通过优化阅读器所发送前缀的方法,提出了一种改进型自适应多叉树(IAMS)防碰撞算法.该算法通过计算碰撞因子来动态地选择搜索树的叉数,并对四叉树的查询前缀进行优化,从而避免了大量的空闲时隙.文章通过数学分析,准确地描述了 IAMS 算法识别标签所需的时隙数.仿真结果表明,改进型自适应多叉树防碰撞算法具有更快的识别速度和更高的系统吞吐率.

**关键词:** 射频识别; 防碰撞算法; 多叉树; 自适应

**中图分类号:** TN92      **文献标识码:** A      **文章编号:** 0372-2112 (2012) 01-0193-06

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2012.01.032

## One Anti-Collision Algorithm Based on Improved Adaptive Multi-Tree Search

ZHANG Xue-jun, CAI Wen-qi, WANG Suo-ping

(College of Electronic Science and Engineering, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China)

**Abstract:** Based on the adaptive multi-tree search anti-collision algorithm, we proposed an improved adaptive multi-tree search anti-collision algorithm (IAMS) by optimizing the prefix sent by readers. Our algorithm reduces idle time slots by choosing the number of search tree branches adaptively according to the collision factor and optimizing the query prefix of quadtree. By mathematical analysis, we accurately predict the total number of time slots required for the tag identification in IAMS algorithm. Simulation results show that the IAMS algorithm has faster identification speed and higher system throughput.

**Key words:** RFID(Radio Frequency Identification); anti-collision algorithms; multi-tree; adaptive

### 1 引言

射频识别(Radio Frequency Identification, RFID)是利用射频信号通过空间电磁耦合实现无接触式自动识别技术的一种. RFID 技术无需直接接触、无需光学可视、无需人工干预即可完成信息输入和处理,具有操作方便快捷、存储数据量大、保密性好、反应时间短、对环境适应性强、识别距离远、穿透能力强、多物体识别、抗污染等优点,现在已广泛应用于工业自动化、商业自动化、交通运输管理、产品证件防伪、防盗等众多领域. RFID 技术、传感器技术、纳米技术、智能嵌入技术是实现物联网的四大核心技术<sup>[1]</sup>. 随着物联网技术和应用的不断深入, RFID 技术已成为当前研究的热点.

典型的 RFID 系统主要包括电子标签(Tag)、阅读器(Reader)和数据处理子系统三个部分,如图 1 所示.

在很多 RFID 的应用场合,阅读器要在很短时间内尽快识别多个标签. 由于阅读器和标签通信共享无线信

道,阅读器或标签的信号可能发生冲突,使阅读器不能正确识别标签,我们称这种现象为碰撞,是 RFID 技术必须解决的关键问题.

RFID 系统有两类标签的防碰撞算法,基于树的确定性算法和基于 ALOHA 的不确定算法<sup>[2]</sup>. 基于树的算法将标签分成多个子群直到所有的标签都被识别出来,例如二进制搜索树(Binary Search Tree, BST)<sup>[3]</sup>算法、查询树(Query Tree, QT)<sup>[4]</sup>算法、双时隙碰撞跟踪树(Bi-Slotted Collision Tracking Tree Algorithm, BSCCTA)<sup>[5]</sup>算法和改进型四叉查询树算法(Improved 4-ary Query Tree Algorithm,

#### 3. 阅读器采集信号并解码

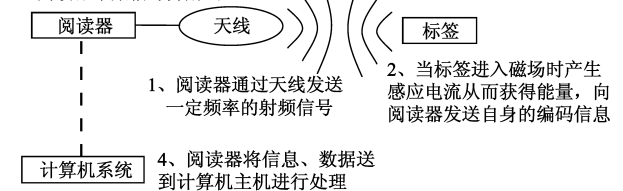


图1 RFID系统结构图

14QTA)<sup>[6]</sup>等. 基于 ALOHA 协议的算法主要有: 时隙 ALOHA (Slotted ALOHA, SA) 算法<sup>[7]</sup>、帧时隙 ALOHA (Frame Slotted ALOHA, FSA) 算法<sup>[8]</sup>、动态帧时隙 ALOHA (Dynamic Frame Slotted ALOHA, DFSA) 算法<sup>[9,10]</sup>等. 由于 ALOHA 算法具有随机性, 可能存在某一标签在相当长的一段时间内都无法被识别, 即出现标签“饥饿”问题, 而基于树的标签防碰撞算法可以达到 100% 的读取率, 但其算法设计复杂, 鉴别时间较长.

本文在自适应多叉树防碰撞算法 (Adaptive Multi-Tree Search Anti-Collision Algorithm, AMS)<sup>[11]</sup>的基础上, 通过优化阅读器所发送前缀的方法, 提出了一种改进型自适应多叉树防碰撞算法 (Improved Adaptive Multi-Tree Search Anti-Collision Algorithm, IAMS). 该算法通过计算碰撞因子来动态地选择搜索树的叉数, 并对四叉树的查询前缀进行优化, 从而避免了大量的空闲时隙. 文章通过数学分析, 准确地描述了 IAMS 算法识别标签所需的时隙数. 仿真结果表明, 改进型自适应多叉树防碰撞算法具有更快的识别速度和更高的系统吞吐率.

## 2 自适应多叉树防碰撞算法

现有防碰撞算法大多通过查询每个时隙里标签的响应状态来鉴别标签, 这些时隙有三种可能情况: 成功时隙, 在该时隙只有一个标签响应阅读器发出的查询命令, 此时可成功鉴别该标签; 碰撞时隙, 在这个时隙有两个或两个以上的标签同时响应阅读器发出的查询命令; 空闲时隙, 在该时隙没有标签响应阅读器发出的查询命令. 为了提高算法的效率, 应该尽可能减少碰撞时隙和空闲时隙的产生.

在基于树的算法中, 四叉树、八叉树等树形算法经常被用来减少标签鉴别过程中的碰撞时隙. 当多叉树的叉数增多时, 碰撞时隙不断减少, 但空闲时隙也在不断增加, 从而导致最后使用的总时隙也有了一定的增加. 所以, 单一的使用某种多叉树必然导致较多的碰撞时隙或大量的空闲时隙. 如果在鉴别过程中能估算出分支内标签的数量, 自适应地选择搜索叉树, 那么就能有效地提高算法的搜索效率, 减少识别时间.

AMS 就是采用了这一思想, 但该算法的前提是要辨识出阅读器中数据碰撞的准确位置, 因此, 采用了 Manchester 编码. 现阶段大多数树形算法仅利用了碰撞位首位或前几位的信息, 剩余的碰撞信息并没有充分利用. 一般来说, 当分支内标签的数量越多时, 出现碰撞的位数就越多, 碰撞位占总比特位的比例也就越大. 因此, 可以根据这个比例来动态的选择使用二叉树还是四叉树. AMS 算法定义了碰撞因子  $\mu$  来表示这个比例.

假设每个标签的 ID 码的长度为  $n$  比特, 其中产生

碰撞的比特位为  $n_c$ , 定义碰撞因子  $\mu = \frac{n_c}{n}$ , 如果系统中存在  $m$  个待识别标签, 则其中任意 1 位不发生碰撞的概率为  $C_2^1 \cdot \left(\frac{1}{2}\right)^m = \left(\frac{1}{2}\right)^{m-1}$ , 由此可以推出

$$\mu = \frac{n \left[1 - \left(\frac{1}{2}\right)^{m-1}\right]}{n} = 1 - \left(\frac{1}{2}\right)^{m-1} \quad (1)$$

式(1)表明, 当系统中待识别标签的数量越大时, 碰撞因子越高, 反之, 碰撞因子越低, 这说明, 碰撞因子的大小与分支内所含标签数量的多少有着密切的关系.

在文献[11]中分析了 AMS 算法中采用二叉树和四叉树时  $\mu$  的取值范围, 即: 假设系统内有  $m$  个待识别标签, 系统分配的叉树为  $L$ , 当搜索深度为 1 时, 标签的识别概率为:  $p(1) = \left(1 - \frac{1}{L}\right)^{m-1}$ , 在搜索深度为  $k$  时, 识别概率为:  $p(k) = p(1)[1 - p(1)]^{k-1}$ , 则所需搜索深度的均值为

$$E(k) = \sum_{k=0}^{\infty} [1 - p(1)]^k = \frac{1}{1 - [1 - p(1)]} = \frac{1}{p(1)} = \frac{1}{\left(1 - \frac{1}{L}\right)^{m-1}} \quad (2)$$

所需的平均时隙数为

$$T_{L-ary} = E(k)L = \frac{L}{\left(1 - \frac{1}{L}\right)^{m-1}} \quad (3)$$

对于二叉树, 所需的平均时隙数为

$$T_{2-ary} = \frac{2}{\left(1 - \frac{1}{2}\right)^{m-1}} \quad (4)$$

对于四叉树, 所需的平均时隙数为

$$T_{4-ary} = \frac{4}{\left(1 - \frac{1}{4}\right)^{m-1}} \quad (5)$$

由式(4)和式(5)可以算出, 当  $m < 3$  时,  $T_{2-ary} < T_{4-ary}$ , 而当  $m \geq 3$  时,  $T_{2-ary} > T_{4-ary}$ , 该算法中将  $m = 3$  作为二叉树和四叉树的临界值, 算出此时的碰撞因子  $\mu = 1 - \left(\frac{1}{2}\right)^{3-1} = 0.75$ , 即当检测出的  $\mu < 0.75$  时, 采用二叉树进行搜索, 当  $\mu \geq 0.75$  时, 改用四叉树搜索.

## 3 改进型自适应多叉树防碰撞算法

AMS 算法通过检测 ID 码的碰撞位信息动态地调整当前碰撞节点应该采用何种多叉树, 在一定程度上减少了碰撞时隙和空闲时隙, 但通过碰撞因子  $\mu$  并不能准确地判断当前碰撞节点下存在标签的多少, 甚至会产生误判. 例如, 如果产生碰撞的标签为 0000 和 1111,

阅读器通过碰撞位检测得到的数据为  $\times\times\times$ , 其碰撞因子  $\mu = 1$ , 应该选用二叉树, 而实际上系统中只存在两个未读标签, 发出的二叉树中产生了两个空闲时隙, 大大增加了鉴别时间。

为了减少 AMS 算法中所出现的空闲时隙, 本文通过优化阅读器发送的前缀, 即只发送系统中存在的前缀, 提出了一种改进型自适应多叉树防碰撞算法 (I-AMS), 具体思想如下: 若碰撞因子  $\mu < 0.75$ , 则采用二叉树, 阅读器根据第一个碰撞位发出两个查询前缀, 此时不会产生空闲时隙; 若碰撞因子  $\mu \geq 0.75$ , 则采用二叉树, 但阅读器并不直接发送碰撞前两位的四个查询前缀, 而是在发送查询前缀前先发送一个命令给当前节点的碰撞标签, 要求其返回前两位碰撞位的具体数值。

例如, 产生碰撞的标签为 1010 和 0100, 阅读器接收到的数据为  $\times\times\times 0$ , 现将这个四位 ID 码从左到右依次定义为 D3、D2、D1、D0, 可得知 D3、D2、D1 位均产生碰撞, 碰撞因子  $\mu = 0.75$ , 应采用二叉树。现要读取的是产生碰撞的前两位, 因此, 阅读器发送一个最高两位碰撞位 D3、D2 为“1”, 其它位为“0”的命令, 即 1100 给标签。标签将接受到的命令 1100 与自身 ID 码相“与”后分别得到两个四位码 1000 和 0100, 其 D3、D2 位分别是 10 和 01, 转换成十进制数为 2 和 1, 接着它们分别返回一个 D2 位和 D1 位为“1”, 其它位为 0 的四位码 0100 和 0010 给阅读器。此时, 阅读器第二次收到的数据就为  $0\times\times 0$ , 根据 D2、D1 位有碰撞, 检测出系统中存在 10 和 01 两个前缀, 阅读器在下一个时隙只发送这两个查询前缀, 从而避免了空闲时隙。

IAMS 算法的鉴别步骤为:

**步骤 1** 阅读器初始化前缀栈, 使之空, 并发出  $\epsilon$  命令;

**步骤 2** 与当前阅读器发送前缀相符的标签响应;

**步骤 3** 判断时隙状态, 如果只有一个标签响应, 则成功鉴别, 跳转到步骤 6; 如果没有标签响应, 则该分叉内无需再继续搜索, 跳转到步骤 6; 如果有多个标签响应, 则为碰撞时隙;

**步骤 4** 计算碰撞因子  $\mu$ 。若  $\mu < 0.75$ , 选择二叉树, 并根据碰撞比特的首位信息, 确定两个新的查询前缀; 若  $\mu \geq 0.75$ , 选择二叉树, 阅读器向标签发送查询高两位碰撞前缀命令, 标签返回一个包含碰撞位信息的 4 位码, 阅读器根据接收到的信息确定系统中存在的前缀;

**步骤 5** 将新生成的前缀写入堆栈, 读取栈首前缀, 并发送给标签, 返回步骤 2;

**步骤 6** 判断前缀堆栈是否为空, 如果不是, 读取栈首前缀, 并发送给标签, 返回步骤 2; 否则, 识别过程结束。

IAMS 防碰撞算法流程如图 2 所示。

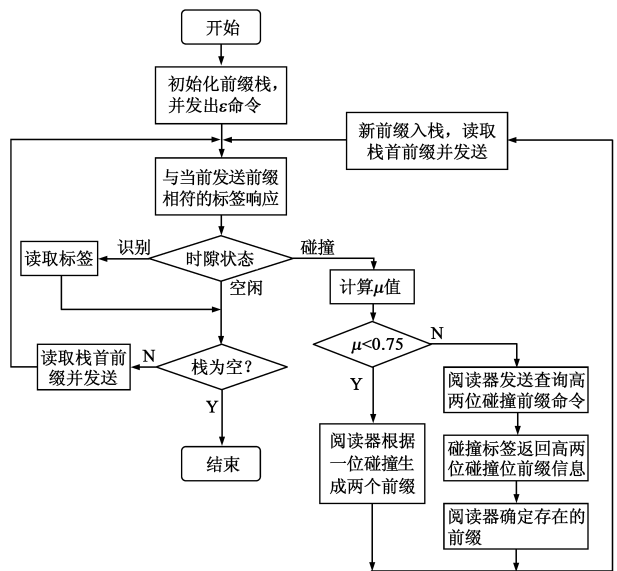


图 2 IAMS 防碰撞算法流程图

## 4 算法性能分析

通过计算总时隙数和吞吐率来分析 IAMS 算法的性能。从上面的算法流程可知, 当  $\mu < 0.75$  时, 采用二叉树, 当  $\mu \geq 0.75$  时, 采用无空闲时隙的二叉树, 所以总时隙数应该为两部分之和。

假设系统中存在  $m$  个待识别标签, 当搜索深度为  $k$  时, 每个子节点上的平均标签数为 3, 那么, 当搜索深度小于  $k$  时, 将采用二叉树, 反之则用二叉搜索。搜索深度  $k = \lfloor \log_4(m/3) \rfloor$ ,  $\lfloor \cdot \rfloor$  表示截尾取整运算。

如果从 0 到  $k$  层的二叉树都没有去空闲时隙, 那么可以得到

$$T_{4-ary} = \sum_{i=0}^k 4^i = \sum_{i=0}^{\lfloor \log_4(m/3) \rfloor} 4^i \quad (6)$$

当搜索深度大于等于  $k$  时, 算法采用二叉搜索, 根据二叉搜索的公式可以得到

$$T_{2-ary} = \frac{m}{3}(2 \times 3 - 1) = \frac{5m}{3} \quad (7)$$

由算法流程图可知, 虽然新算法中采用了无空闲时隙的二叉树, 但阅读器在检测到碰撞信号之后, 第二次发送的查询命令仍需占用一个时隙, 而这个命令所占用的时隙数  $T_{comm}$  正好等于碰撞时隙数, 即  $T_{comm} = T_{4-coll}$ 。

下面, 计算二叉树中的空闲时隙  $T_{4-idle}$  和碰撞时隙  $T_{4-coll}$  [12]。

对于一棵完全二叉树, 在第  $L$  层的时候, 有  $4^L$  个节点, 假设现在有  $m$  个未读标签, 其中有  $k$  个标签在第  $L$  层选择同一个节点响应的概率服从二项分布

$$P(k/m, L) = C_m^k p^k (1-p)^{m-k} \quad (8)$$

这里,  $p = 4^{-L}$  表示在  $4^L$  个节点中选择一个节点的

概率.

空闲概率

$$P(0/m, L) = (1-p)^m = (1-4^{-L})^m \quad (9)$$

成功识别概率

$$P(1/m, L) = mp(1-p)^{m-1} = \frac{m}{4^L}(1-4^{-L})^{m-1} \quad (10)$$

碰撞概率

$$\begin{aligned} P(k > 1/m, L) &= 1 - P(0/m, L) - P(1/m, L) \\ &= 1 - (1-p)^m - mp(1-p)^{m-1} \\ &= 1 - (1-4^{-L})^m - \frac{m}{4^L}(1-4^{-L})^{m-1} \end{aligned} \quad (11)$$

令  $q_{Li/m}$  表示在  $L$  层的第  $i$  个节点被搜索到的概率, 显然, 当  $L=0$  的时候, 根节点总是可以被访问到的, 即  $q_{0i/m} = q_{00/m} = 1$ . 对于除了根节点以外的其他节点, 能被访问到的前提条件是它的父节点产生碰撞, 所以

$$q_{Li/m} = q_{L/m} = \begin{cases} 1, & L=0 \\ \beta_{L-1/m}, & L>0 \end{cases} \quad (12)$$

令  $\beta_{Li/m}$  表示在  $L$  层的第  $i$  个节点产生碰撞的概率, 当同一层的每个节点产生碰撞的概率都相同时, 可以得到

$$\begin{aligned} \beta_{Li/m} &= \beta_{L/m} \\ &= P(k > 1/m, L) \\ &= 1 - (1-4^{-L})^m - \frac{m}{4^L}(1-4^{-L})^{m-1} \end{aligned} \quad (13)$$

平均总时隙为对所有的  $q_{Li/m}$  求和, 即

$$\begin{aligned} \bar{t}(m) &= \sum_{L=0}^{\infty} \sum_{i=0}^{4^L-1} q_{Li/m} = 1 + \sum_{L=1}^{\infty} 4^L q_{L/m} \\ &= 1 + \sum_{L=1}^{\infty} 4^L \beta_{L-1/m} = 1 + 4 \times \sum_{L=0}^{\infty} 4^L \beta_{L/m} \end{aligned} \quad (14)$$

将  $\beta_{L/m}$  代入式(14)可得

$$\bar{t}(m) = 1 + 4 \times \sum_{L=0}^{\infty} 4^L \left[ 1 - (1-4^{-L})^m - \frac{m}{4^L}(1-4^{-L})^{m-1} \right] \quad (15)$$

碰撞总时隙为对所有的  $\beta_{Li/m}$  求和, 即

$$\bar{c}(m) = \sum_{L=0}^{\infty} \sum_{i=0}^{4^L-1} \beta_{Li/m} = \sum_{L=0}^{\infty} 4^L \beta_{L/m} \quad (16)$$

用  $\bar{t}(m)$  可以表示为

$$\bar{c}(m) = \left( \frac{1}{4} \right) (\bar{t}(m) - 1) \quad (17)$$

空闲时隙

$$\bar{z}(m) = \bar{t}(m) - \bar{c}(m) - m = \left( \frac{3}{4} \right) \bar{t}(m) - m + \frac{1}{4} \quad (18)$$

由上述分析可知, 总时隙  $\bar{t}(m)$ 、碰撞时隙  $\bar{c}(m)$ 、空闲时隙  $\bar{z}(m)$  都是关于  $m$  的函数, 通过仿真可得到图 3 所示的结果.

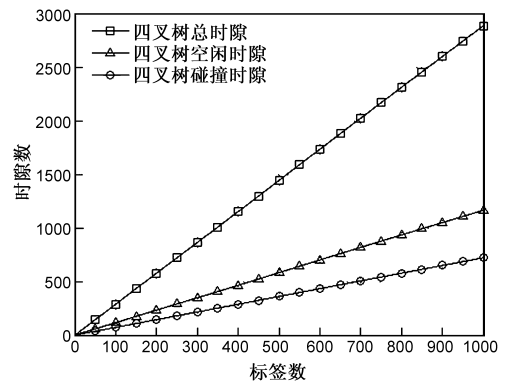


图3 四叉树的各种状态时隙数

由图 3 可见,  $\bar{t}(m)$ 、 $\bar{c}(m)$ 、 $\bar{z}(m)$  与  $m$  均呈线性关系, 求得其近似线性关系为

$$\bar{t}(m) \approx 2.88m, \bar{c}(m) \approx 0.72m, \bar{z}(m) \approx 1.16m$$

根据上述关系可知: 如果完全采用四叉树, 那么空闲时隙和碰撞时隙的个数分别为  $1.16m$  和  $0.72m$ , 但是在 IAMS 算法中, 当子节点上的平均标签数为 3 时采用二叉树搜索, 也就是说, 将原来四叉树中的最后一层改为二叉树搜索, 那么, 在计算四叉树的空闲时隙和碰撞时隙时也需要减去最后一层里有可能产生的空闲和碰撞, 即

$$T_{4-idle} = \bar{z}(m) - \overline{z_{k+1}}(m) \quad (19)$$

$$T_{4-coll} = \bar{c}(m) - \overline{c_{k+1}}(m) \quad (20)$$

$$\begin{aligned} \text{其中, } \overline{z_{k+1}}(m) &= 4^{k+1}(1-4^{-(k+1)})^m \\ \overline{c_{k+1}}(m) &= 4^{k+1} \left[ 1 - (1-4^{-(k+1)})^m - \frac{m}{4^{k+1}}(1-4^{-(k+1)})^{m-1} \right] \end{aligned} \quad (21)$$

所以, IAMS 算法成功鉴别  $m$  个标签的总时隙数为

$$\begin{aligned} T(m) &= T_{4-ary} + T_{2-ary} - T_{4-idle} + T_{comm} \\ &= T_{4-ary} + T_{2-ary} - T_{4-idle} + T_{4-coll} \\ &= \sum_{i=0}^{\lfloor \log_2(m/3) \rfloor} 4^i + \frac{5m}{3} - (1.16m - 4^{k+1}(1-4^{-(k+1)})^m) \\ &\quad + \{0.72m - 4^{k+1}[1 - (1-4^{-(k+1)})^m - \frac{m}{4^{k+1}}(1-4^{-(k+1)})^{m-1}]\} \end{aligned} \quad (22)$$

吞吐率为

$$S_{IAMS} = \frac{m}{T(m)} \quad (23)$$

## 5 算法仿真分析

经过上面的算法分析, 可以得到鉴别标签所用的总时隙, 而衡量防碰撞算法的性能主要是从鉴别延时  $D$  和吞吐率  $S$  两个方面进行分析.

$$D = (D_{reader} + D_{tag}) \times T \quad (24)$$

$D_{reader}$  是阅读器发送一个命令所需的时间,  $D_{tag}$  是一个标签发送其自身信息的时间,  $T$  是鉴别所有标签所需的

时隙数. 由于  $D_{reader}$  和  $D_{tag}$  在系统中都被设置为常量, 所以鉴别延时  $D$  主要取决于总时隙数  $T$ .  $T$  越大, 鉴别延时  $D$  就越大, 效率也就越低.

$$S = m/T(m) \quad (25)$$

同样, 在式(25)中, 由于每次仿真的标签数量  $m$  都是人为设定的, 所以  $S$  仍然由  $T$  决定.  $T$  越小, 吞吐率就越高, 性能也就越好.

下面, 将 IAMS 算法与几种常见的树形算法进行仿真, 从总时隙数和吞吐率两个方面进行性能上的比较.

IAMS 算法解决了 AMS 算法中存在空闲时隙的问题, 也解决了无空闲时隙四叉树算法采用单一的四叉树存在较多碰撞时隙的问题. 由图 4 可见, 在标签总数从 0 变化到 1000 的过程中, IAMS 算法的性能始终优于 AMS 算法和无空闲时隙四叉树算法, 当标签数量达到 1000 时, 无空闲时隙四叉树算法需要将近 2500 个时隙, AMS 算法需要约 2000 个时隙, 而 IAMS 算法只需要约 1700 个时隙, IAMS 算法所需的时隙数最少, 成功识别标签的速度最快.

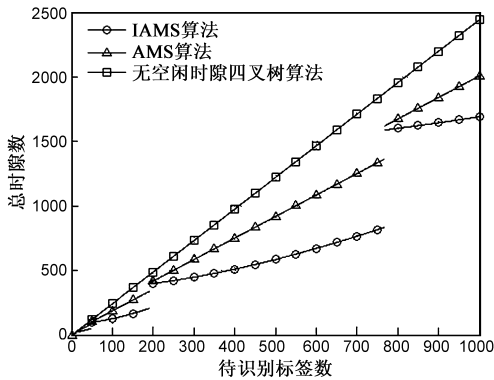


图4 IAMS算法与AMS算法、无空闲时隙四叉树算法总时隙数比较

由图 4 也可以发现, AMS 算法和 IAMS 算法的仿真结果并不像无空闲时隙四叉树算法是一条连续平滑的曲线, 而是间断式的, 随着待识别标签数的不同在几个点上产生了分叉. 这是因为 AMS 和 IAMS 算法均能自适应的调整搜索叉数, 由文献[11]及式(22)可知 AMS 算法所消耗的时隙数都与搜索深度  $k = \lfloor \log_4(m/3) \rfloor$  有关,

$k$  是非负整数,  $\sum_{i=0}^{\lfloor \log_4(m/3) \rfloor} 4^i$  也必然是一些非连续的整数, 因此 AMS 和 IAMS 算法的总时隙数在某些点是跳变的.

图 5 是把 IAMS 算法和二进制查询树以及碰撞跟踪树作了比较. 参照前文中对四叉树总时隙数的推导方法, 可得二进制查询树算法的总时隙数为  $T_{2-ary} = 1 + 2 \times \sum_{L=0}^{\infty} 2^L [1 - (1 - 2^{-L})^m - \frac{m}{2^L} (1 - 2^{-L})^{m-1}]$ , 而碰撞跟踪树算法由于采用曼彻斯特编码检测碰撞位, 避免

了很多不必要的查询命令, 其所需时隙数为  $T_{track} = 2m - 1$ . 从仿真结果可以发现, 同样在鉴别过程中采用曼彻斯特编码的碰撞跟踪树算法, 其性能只在少数点上与 IAMS 算法相当, 大多数情况下其性能都不及 IAMS 算法好.

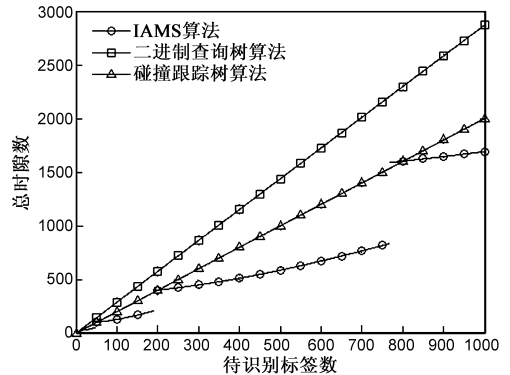


图5 IAMS算法和二进制查询树算法、碰撞跟踪树算法总时隙数比较

图 6 中对 IAMS 算法、AMS 算法及无空闲时隙四叉树算法的吞吐率进行了比较, 可以看出, IAMS 算法的吞吐率明显要比 AMS 算法和无空闲时隙四叉树算法要高. AMS 算法的吞吐率基本在 0.5 到 0.6 之间, 无空闲时隙四叉树算法则维持在 0.41 附近, 而 IAMS 算法的吞吐率最高能达到 0.9 左右, 但其变化的幅度相对于其它两种算法都要大一些.

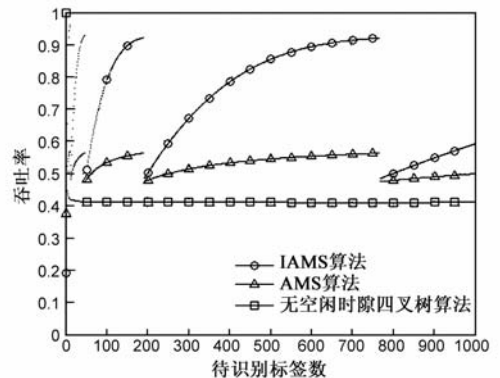


图6 IAMS算法与AMS算法、无空闲时隙四叉树算法吞吐率比较

## 6 结论

本文在 AMS 算法的基础上提出了一种 IAMS 防碰撞算法, 新算法利用曼彻斯特编码可以准确识别碰撞位的特性, 通过计算碰撞因子来估计标签数量, 从而自适应的调整搜索叉树, 即在标签数较多的分支内选择四叉树, 反之则选用二叉树. 在四叉树中利用无空闲时隙四叉树优化前缀的方法来跳过空闲时隙, 即通过阅读器单独发送的一条查询命令来获知系统中存在的前缀, 从而避免了空闲时隙. 经过理论和仿真分析, 新算法解决了 AMS 算法和无空闲时隙四叉树中的问题, 在

减少碰撞时隙的基础上,又减少了空闲时隙,大幅度地提高了阅读器识别标签的速度和系统的吞吐率。

IAMS 算法是在标签数量一定条件下,如何用最少的时间,最高的效率来完成识别,并没有对标签的运动情况、信道环境和标签识别的优先级等因素进行充分考虑,而这些因素在实际应用中势必将影响到算法的性能。标签端算法设计越复杂,算法的效率就越高,随之带来的标签成本也就越高,带着这些问题,IAMS 算法还有很多需要改进的地方。

#### 参考文献

- [1] 宁焕生,徐群玉.全球物联网发展及中国物联网建设若干思考[J].电子学报,2010,38(11):2590-2599.  
Ning Huan-sheng, Xu Qun-yu. Research on global internet of things developments and its construction in China[J]. Acta Electronica Sinica, 2010, 38(11): 2590-2599. (in Chinese)
- [2] 王雪,钱志鸿,胡正超,等.基于二叉树的 RFID 防碰撞算法的研究[J].通信学报,2010,31(6):49-57.  
Wang Xue, Qian Zhi-hong, HU Zheng-chao, et al. Research on RFID anti-collision algorithms based on binary tree[J]. Journal on Communications, 2010, 31(6): 49-57. (in Chinese)
- [3] K Finkenzeller. RFID Handbook; Fundamentals and Applications in Contactless Smart Cards and Identification (2nd Edition)[M]. New York, USA: John Wiley & Sons Inc, 2003. 8-156.
- [4] C Law, K Lee, K Y Siu. Efficient memoryless protocol for tag identification[A]. Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications[C]. New York, USA: ACM Press, 2000. 75-84.
- [5] J H Choi, D W Lee, H J Lee. Bi-slotted tree based anti-collision protocols for fast tag identification in RFID systems[J]. IEEE Communication Letter, 2006, 10(12): 861-863.
- [6] Yonghwan Kim, Sungsoo Kim, et al. Improved 4-ary query tree algorithm for anti-collision in RFID system[A]. International Conference on Advanced Information Networking and Applications[C]. Bradford, United Kingdom: IEEE Computer Society, 2009. 699-704.
- [7] L Liu, S Lai. ALOHA-based anti-collision algorithms used in RFID system[A]. Wireless Communications, Networking and Mobile Computing (WiCOM 2006)[C]. Wuhan, China, 2006. 1-4.
- [8] H Vogt. Efficient object identification with passive RFID tags

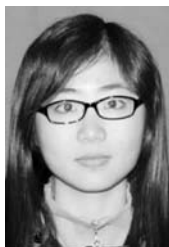
[A]. International Conference on Pervasive Computing[C]. Zurich, Switzerland, 2002. 98-113.

- [9] M Kodialam, Thyaga Nandagopal. Fast and reliable estimation schemes in RFID systems[A]. Proceedings of ACM Mobicom[C]. Los Angeles, California, USA, 2006. 322-333.
- [10] S Lee, S D Joo, C W Lee. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification[A]. Proceedings of Mobiquitous 2005[C]. San Diego, CA, USA: IEEE Computer Society, 2005. 166-172.
- [11] 丁治国,朱学永,等.自适应多叉树防碰撞算法研究[J].自动化学报,2010,36(2):237-241.  
Ding Zhi-guo, Zhu Xue-yong, et al. An adaptive anti-collision algorithm based on multi-tree search[J]. Acta Automatica Sinica, 2010, 36(2): 237-241. (in Chinese)
- [12] D R Hush, C Wood. Analysis of tree algorithms for RFID arbitration[A]. Proceedings of IEEE Symposium on Information Theory (ISIT)[C]. Cambridge, MA, USA, 1998. 107-116.

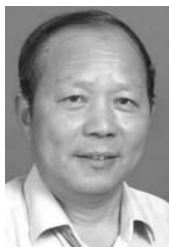
#### 作者简介



张学军 男,1969年8月出生于江苏南通。1993年、2001年分别在中国石油大学、东南大学获工学学士和工学硕士学位。现为南京邮电大学副教授、硕士生导师。主要研究方向为无线射频识别技术、通信网络的性能分析、流量控制、QoS理论与技术。  
E-mail: xjzhang@njupt.edu.cn



蔡文琦 女,1987年10月出生于江苏宜兴,2010年在南京邮电大学获理学学士学位,现为南京邮电大学电子与通信领域硕士研究生。主要研究方向为无线射频识别技术。



王锁萍 男,1946年9月出生于上海。1970年、1981年分别在清华大学、南京邮电大学获工学学士和工学硕士学位。现为南京邮电大学教授、博士生导师。主要研究方向为通信网络的性能分析、流量控制、QoS理论与技术、单播组播路由算法、通信及网络安全理论与技术,以及信源编码、信道编码的理论与技术。