

基于改进后退策略的按位二进制防碰撞算法

李 全

(湖北师范学院教育信息与技术学院, 湖北 黄石 435002)

摘 要: 为解决无线射频识别系统中的标签碰撞问题, 提出一种基于改进后退策略的按位二进制防碰撞算法。该算法利用标签 ID 的唯一性, 使阅读器只需通过标签 ID 的部分比特位就能准确地识别某个标签。仿真结果表明, 改进的搜索算法能减少阅读器与标签之间的识别通信量及阅读器的搜索次数, 提高标签的识别速度。

关键词: 无线射频识别; 防碰撞算法; 标签识别; 二进制搜索

Bit-by-bit Binary Anti-collision Algorithm Based on Improved Backtracking

LI Quan

(College of Educational Information & Technology, Hubei Normal University, Huangshi 435002, China)

【Abstract】 In order to effectively identify an electronic tag in Radio Frequency Identification(RFID) system, a bit-by-bit binary anti-collision algorithm based on improved backtracking is proposed. This algorithm takes advantage of unique feature of tag identification, and makes the reader identify a tag accurately by only parts of bits of tag identification. The experiment fully considers two important performance index of bits transferred between the reader and the tags, and the search number of reader. The analysis on simulation result indicates that the improved algorithm performs significantly better than the other binary tree algorithms. It can not only greatly reduce the bits transferred between the reader and the tags, but also reduce the search number of reader. The speed of identifying tags can be enhanced greatly. Therefore the application has good prospect.

【Key words】 Radio Frequency Identification(RFID); anti-collision algorithm; tag identification; binary search

DOI: 10.3969/j.issn.1000-3428.2012.03.092

1 概述

无线射频识别(Radio Frequency Identification, RFID)技术是一种非接触式的自动识别技术。它通过无线射频信号自动识别目标对象并获取相关数据的信息。RFID 系统一般由阅读器、标签和天线组成。该技术具有识别距离远、穿透能力强、多目标识别和抗污染等优点。现已被广泛应用于诸多领域, 如零售业、物流业^[1]。

在同一个 RFID 系统中, 所有的电子标签都工作在相同的频率段。因此, 如果阅读器的作用范围内有多个标签存在, 同一时刻可能有多个标签获得能量并通过共享无线信道向阅读器发送信息。这样就会出现相互干扰, 使阅读器不能正常识别标签。该现象称为标签冲突或碰撞。因此, 需要一种防碰撞技术来识别多个标签, 如防碰撞算法。现有的 RFID 防碰撞算法主要 2 种: 基于 ALOHA 的不确定性算法^[2]和基于二进制的确定性算法^[3]。前者一般包括时隙 ALOHA 算法、帧时隙 ALOHA 算法、动态帧时隙 ALOHA 算法等; 后者一般包括二进制搜索(Binary Search, BS)算法、按位二进制树(Bit-by-Bit Binary Tree, BBT)算法和后退策略的按位二进制搜索(Bit-by-Bit Binary Search of Backtracking, BBSB)算法等。ALOHA 算法由于随机性较大, 可能导致“标签饥饿问题”, 即不能保证所有的标签都被识别。二进制搜索算法虽然可以解决“标签饥饿问题”, 但也存在着识别周期长、标签能耗大等问题^[4]。

基于以上分析, 在 RFID 系统中, 一个好的防碰撞算法应该具有以下特点:

(1) 在阅读器的识别范围内, 阅读器成功搜索标签的次数较少。

(2) 在整个识别过程中, 阅读器与标签之间需要发送的总比特数较小。

(3) 保证阅读器作用范围内的标签都能被正确、完整地识别。

针对目前二进制防碰撞算法存在的识别周期长、标签功耗大等问题, 本文提出一种基于改进后退策略的按位二进制防碰撞算法, 以减少阅读器的搜索次数和阅读器与标签之间的识别通信量。

2 现有的二进制搜索算法

以下分析假设现有 N 个标签待识别, 且每个标签 ID 号的长度为 L 比特。

2.1 BS 算法

BS 算法^[1]的主要思想是阅读器发送一个长度为 L 的序列号, 各个标签将自身的 ID 号与接收到的序列号进行比较。其二进制数值小于或者等于该序列号的标签将回送自身的 ID 号给阅读器, 否则不作响应。这样可以缩小预选标签的范围。

阅读器通过二进制搜索算法在搜索范围内搜索 N 个标签的总搜索次数为: $T = \text{Int}(\lg N! + N)$, $\text{Int}()$ 表示取整数。在整

基金项目: 湖北师范学院青年基金资助项目(2010C33)

作者简介: 李 全(1982—), 男, 硕士, 主研方向: 无线射频识别, 网络通信

收稿日期: 2011-07-06 **E-mail:** cleverlq@hotmail.com

个识别过程中, 读写器与 N 个标签之间需要发送的总比特数为: $B = 2 \times \text{Int}(\lg N) + N \times L$ 。对于该算法, 阅读器的搜索次数比较多, 且每次标签应答都需要发送完整的序列号, 因此, 识别通信量也比较大。

2.2 BBT 算法

BBT 算法^[5]的基本思路是: 阅读器发送请求命令, 请求标签回送序列号。响应的标签每次只发送 1 位序列号。如果阅读器端没有发生冲突, 则在内存中保存该接收位, 然后请求下一位; 如果阅读器处发生冲突, 则将该冲突位分为 2 支, 即分支 0 和分支 1, 从中选择一个分支, 然后请求下一位。阅读器重复上述过程直至序列号的每一位都被识别。该算法搜索到 N 个标签的总搜索次数为: $I = N \times L$, 且发送的总比特数为: $B = 2 \times N \times L$ 。

2.3 BBSB 算法

BBSB 算法^[6]在 BBT 算法的基础上进行了改进。在 BBSB 算法中, 首先阅读器发送请求命令, 如果发生碰撞, 则记录每一个碰撞位, 然后阅读器根据所选取的状态(0 或 1)重复发送请求命令。当阅读器成功选取标签并读写完毕后, 返回上一个记录的碰撞点, 最后修改阅读器原本选取的状态, 再次进行标签的选取工作。当完成一个标签的读写再重新选取一个标签时, 不需要全部重新筛选。该算法搜索到 N 个标签的总搜索次数为:

$$I = \sum_{k=1}^{k_{\max}} \frac{N(L+1-k)}{2^k} + L + \left(\frac{N}{2^{k_{\max}}} - 1\right)(L - k_{\max})$$

其中, $k_{\max} = \lceil \lg N \rceil$; $[A]$ 表示一个小于等于 A 的最大整数。

BBBS 算法发送的总比特数为:

$$B = 2 \times \sum_{k=1}^{k_{\max}} \frac{N(L+1-k)}{2^k} + L + \left(\frac{N}{2^{k_{\max}}} - 1\right)(L - k_{\max})$$

3 基于改进后退策略的按位二进制搜索算法

虽然后退策略按位二进制防碰撞算法在一定程度上减少了搜索次数和识别通信量, 但是当标签的数量不断增加时, 搜索次数和识别通信量仍然很大。因此, 本文在对后退策略按位二进制防碰撞算法进行分析研究的基础上提出了一种改进的防碰撞算法。

3.1 算法的相关指令

为了实现改进的算法, 需要一组指令, 这组指令由电子标签处理。此外, 每个电子标签拥有一个唯一的序列号^[1]。具体指令如下:

(1)Request($B[i]$, $Num[i]$): 表示请求标签 ID 号某一位的下一位。当 $i \geq 1$ 且 $i \leq L$ 时, $B[i]$ 表示标签 ID 号第 i 位的值为 0 或 1, $Num[i]$ 表示标签 ID 号的第 $Num[i]$ 位。 i 的初始值为 0。

(2)Sleep($Slep$): 表示标签发生碰撞时, 当碰撞位为 1 时, 休眠值 $Slep$ 加 1; 当碰撞位为 0 时, 休眠值 $Slep$ 不变。当 $Slep > 0$ 时, 该标签处于休眠状态; 当 $Slep = 0$ 时, 该标签处于待识别状态。 $Slep$ 的初始值为 0。

3.2 算法的工作流程

假设现有 5 个长度为 8 比特的标签需要识别, 它们的 ID 号分别为: 标签 1: 0101 0110; 标签 2: 0100 1110; 标签 3: 0111 0010; 标签 4: 0100 1010; 标签 5: 0110 1100。

改进算法的主要步骤如下:

(1)阅读器对磁场区域内的标签发送 Request(0, NULL)命令, 要求区域内所有的标签返回自己的首位, 此时阅读器收到 5 位数据, 分别为 0、0、0、0、0, 没有发生碰撞, $i = i + 1$, 所以, $B[1] = 0$, $Num[1] = 1$ 。

(2)阅读器发送 Request(0, 1)命令, 要求所有第 1 位为 0 的标签返回自己的第 2 位, 此时阅读器收到 5 位数据, 分别为 1、1、1、1、1, 没有发生碰撞, $i = i + 1$, 所以, $B[2] = 1$, $Num[2] = 2$ 。

(3)阅读器发送 Request(1, 2)命令, 要求所有第 2 位为 1 的标签返回自己的第 3 位, 此时阅读器收到 5 位数据, 分别为 0、0、0、1、1, 发生碰撞, $Slep_{\text{标签}3} = 1$, $Slep_{\text{标签}5} = 1$ 。第 3 位为 1 的标签 3 和标签 5 进入休眠状态, 暂时不响应阅读器的命令, 第 3 位为 0 的标签 1、标签 2 和标签 4 返回自己的第 3 位数据, $i = i + 1$, 所以, $B[3] = 0$, $Num[3] = 3$, 并将 $B[2] = 1$ 和 $Num[2] = 2$ 压入堆栈 S 。

(4)阅读器发送 Request(0, 3)命令, 要求所有第 3 位为 0 的标签返回自己的第 4 位, 此时阅读器收到 3 位数据: 0, 0, 1, 发生碰撞, $Slep_{\text{标签}3} = 2$, $Slep_{\text{标签}5} = 2$, $Slep_{\text{标签}1} = 1$ 。第 4 位为 1 的标签 1 进入休眠状态, 暂时不响应阅读器的命令, 第 4 位为 0 的标签 2 和标签 4 返回自己的第 4 位数据, $i = i + 1$, 所以, $B[4] = 0$, $Num[4] = 4$, 并将 $B[3] = 0$ 和 $Num[3] = 3$ 压入堆栈 S 。

(5)阅读器发送 Request(0, 4)命令, 要求所有第 4 位为 0 的标签返回自己的第 5 位, 此时阅读器收到 2 位数据, 分别为 1、1, 没有发生碰撞, $i = i + 1$, 所以, $B[5] = 1$, $Num[5] = 5$ 。

(6)阅读器发送 Request(1, 5)命令, 要求所有第 5 位为 1 的标签返回自己的第 6 位, 此时阅读器收到 2 位数据, 分别为 0、1, 发生碰撞, $Slep_{\text{标签}3} = 3$, $Slep_{\text{标签}5} = 3$, $Slep_{\text{标签}1} = 2$, $Slep_{\text{标签}2} = 1$ 。第 6 位为 1 的标签 2 进入休眠状态, 暂时不响应阅读器命令, 第 6 位为 0 的标签 4 返回自己的第 6 位数据, $i = i + 1$, 所以, $B[6] = 0$, $Num[6] = 6$, 并将 $B[5] = 1$ 和 $Num[5] = 5$ 压入堆栈 S 。

(7)阅读器发送 Request(0, 6)命令, 要求所有第 6 位为 0 的标签返回自己的第 7 位, 此时阅读器只收到 1 位数据, 且值为 1, 没有发生碰撞, $i = i + 1$, 所以, $B[7] = 1$, $Num[7] = 7$ 。阅读器可以根据标签 ID 号的前 7 位数据唯一地确定标签 4, 且 $Slep_{\text{标签}3} = 2$, $Slep_{\text{标签}5} = 2$, $Slep_{\text{标签}1} = 1$, $Slep_{\text{标签}2} = 0$ 。

(8)阅读器从堆栈 S 中取出 $B[5] = 1$ 和 $Num[5] = 5$, 并发送 Request(1, 5)命令, 要求所有第 5 位为 1 的待识别标签返回自己的第 6 位, 此时阅读器只收到 1 位数据, 且值为 1, 没有发生碰撞, $i = i + 1$, 所以 $B[6] = 1$, $Num[6] = 6$ 。阅读器可根据标签 ID 号的前 6 位数据唯一地确定标签 2, 且 $Slep_{\text{标签}3} = 1$, $Slep_{\text{标签}5} = 1$, $Slep_{\text{标签}1} = 0$ 。

(9)阅读器从堆栈 S 中取出 $B[3] = 0$ 和 $Num[3] = 3$, 并发送 Request(0, 3)命令, 要求所有第 3 位为 0 的待识别标签返回自己的第 4 位, 此时阅读器只收到 1 位数据, 值为 1, 没有发生碰撞, $i = i + 1$, 所以, $B[4] = 1$, $Num[4] = 4$ 。阅读器可根据标签 ID 号的前 4 位数据唯一地确定标签 1, 且 $Slep_{\text{标签}3} = 0$, $Slep_{\text{标签}5} = 0$ 。

(10)阅读器从堆栈 S 中取出 $B[2] = 1$ 和 $Num[2] = 2$, 并发送 Request(1, 2)命令, 要求所有第 2 位为 1 的标签返回自己的第 3 位, 此时阅读器收到 2 位数据, 分别为 1、1, 没有发生碰撞, $i = i + 1$, 所以, $B[3] = 1$, $Num[3] = 3$ 。

(11)阅读器发送 Request(1, 3)命令, 要求所有第 3 位为 1 的标签返回自己的第 4 位, 此时阅读器收到 2 位数据, 分别为 0、1, 发生碰撞, $Slep_{\text{标签}3} = 1$ 。第 4 位为 1 的标签 3 进入休眠状态, 暂时不响应阅读器的命令, 第 4 位为 0 的标签返回自己的第 4 位数据, $i = i + 1$, 所以, $B[4] = 0$, $Num[4] = 4$,

并将第3位 $B[3]=1$ 和 $Num[3]=3$ 压入堆栈 S 。

(12)阅读器发送 $Request(0, 4)$ 命令, 要求所有第4位为0的待识别标签返回自己的第5位, 此时阅读器只收到1位数据, 且值为1, 没有发生碰撞。第4位为0的标签返回自己的第5位数据, $i=i+1$, 所以, $B[5]=1, Num[5]=5$, 阅读器可根据标签ID号的前5位数据确定标签5, 且 $Slep_{\text{标签}5}=0$ 。

(13)阅读器从堆栈 S 中取出 $B[3]=1$ 和 $Num[3]=3$, 并发送 $Request(1, 3)$ 命令, 要求所有第3位为1的标签返回自己的第4位, 此时阅读器只收到1位数据, 且值为1, 没有发生碰撞。 $i=i+1$, 所以, $B[4]=1, Num[4]=4$ 。阅读器可以根据标签ID号的前4位数据唯一地确定标签3。

按位二进制防碰撞算法工作流程如图1所示。

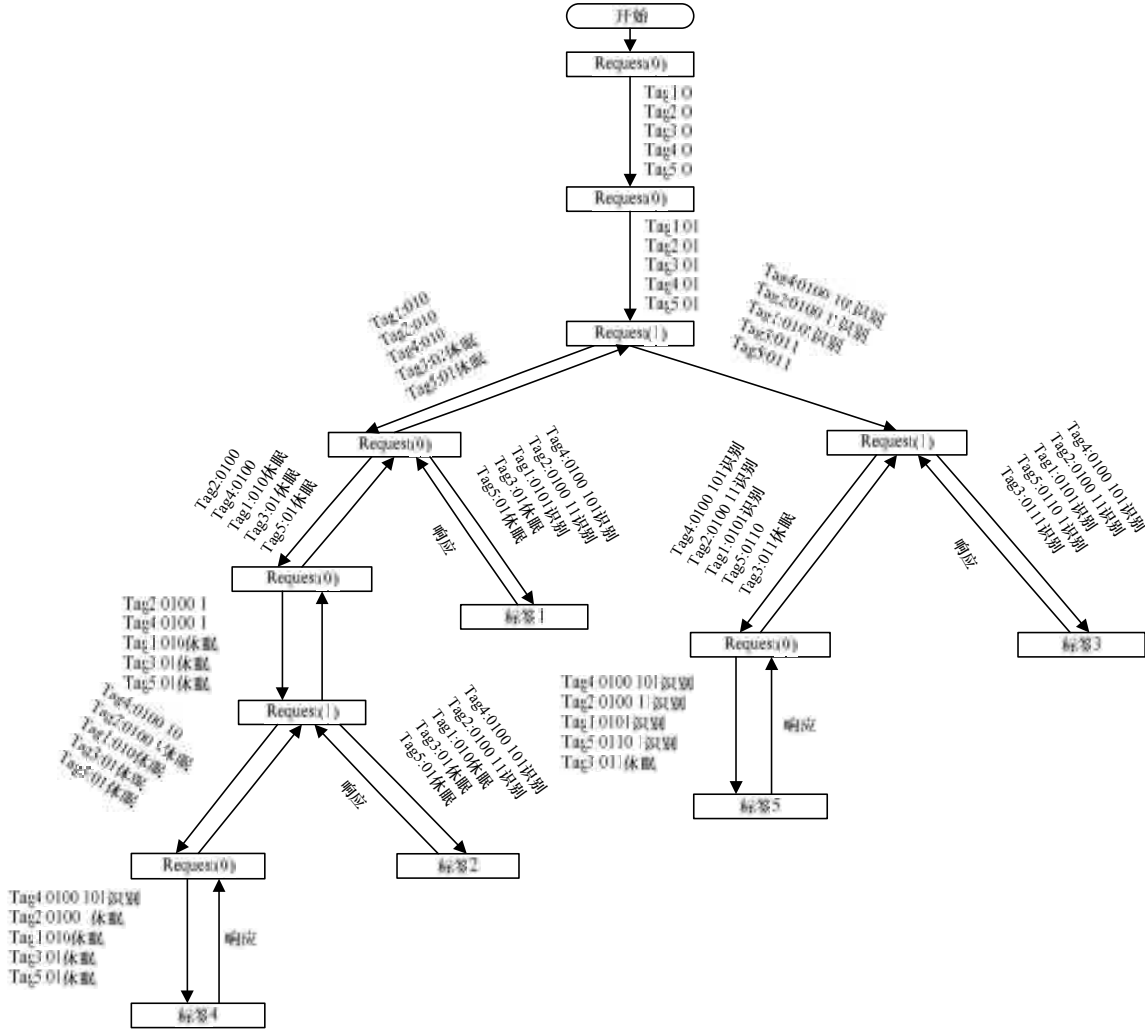


图1 基于改进后退策略的按位二进制防碰撞算法工作流程

4 改进算法的性能分析

若有 N 个电子标签在阅读器的工作区域内, 且其比特位数相同, 为识别 N 个电子标签, 设 D_i 为第 i 个电子标签的十进制数, 则 N 个电子标签的十进制数之和 $D_{\text{sum}} = \sum_{i=1}^N D_i$; D_{max} 为 N 个标签十进制数之和的最大值。设 $Q(N)$ 为阅读器搜索到所有标签的搜索次数; 当 $D_{\text{sum}} = D_{\text{max}}$ 时, 阅读器搜索到所有标签的搜索次数最大, 且最大值为:

$$Q(N)_{\text{max}} = \begin{cases} 1 & N = 1 \\ 2 \times m + 3 & N \geq 1 \end{cases} \quad (1)$$

例如有 3 个比特数为 5 的标签, 分别为 Tag1、Tag2 和 Tag3。当 Tag1: 11111、Tag2: 11110、Tag3: 11101 时, 3 个标签十进制数之和为最大值, 且 $D_{\text{max}} = 31 + 30 + 29$ 。当 Tag1、Tag2、Tag3 为任意值时, $0 < D_{\text{sum}} \leq D_{\text{max}}$, 且 $Q(N) \leq Q(N)_{\text{max}}$ 。

下面采用数学归纳法来证明式(1)。

(1)当 $N=1$ 时, 表示阅读器工作区域内只有 1 个电子标

签, 所以, 不会产生碰撞, 显然 $Q(1)_{\text{max}} = 1$ 成立。

(2)当 $N=2$ 时, 表示阅读器工作区域内只有 2 个电子标签, 当 Tag1: 11111、Tag2: 11110 时, 2 个标签的十进制数之和为最大值。此时, 读写器和阅读器执行以下任务:

1)读写器发送 $Request(0, NULL)$ 命令, 要求区域内所有的标签返回自己的首位, 此时阅读器收到 2 位数据, 分别为 1、1, 没有发生碰撞, $i=i+1$, 所以, $B[1]=1, Num[1]=1$ 。

2)读写器发送 $Request(1, 1)$, 要求所有第 1 位为 1 的标签返回自己的第 2 位, 此时阅读器收到 2 位数据, 分别为 1、1, 没有发生碰撞, $i=i+1$, 所以, $B[2]=1, Num[2]=2$ 。

3)读写器发送 $Request(1, 2)$, 要求所有第 2 位为 1 的标签返回自己的第 3 位, 此时阅读器收到 2 位数据, 分别为 1、1, 没有发生碰撞, $i=i+1$, 所以, $B[3]=1, Num[3]=3$ 。

4)读写器发送 $Request(1, 3)$, 要求所有第 3 位为 1 的标签返回自己的第 4 位, 此时阅读器收到 2 位数据, 分别为 1、1, 没有发生碰撞, $i=i+1$, 所以, $B[4]=1, Num[4]=4$ 。

5)读写器发送 Request(1, 4),要求所有第4位为1的标签返回自己的第5位,此时阅读器收到2位数据,分别为0、1,发生碰撞, $Slep_{\text{标签}1}=1$ 。第5位为1的标签1进入休眠状态,暂时不响应阅读器的命令,第5位为0的标签返回自己的第5位数据, $i=i+1$,所以, $B[5]=0$, $Num[5]=5$,并将第4位 $B[4]=1$ 和 $Num[4]=4$ 压入堆栈 S 。

6)阅读器发送 Request(0, 5)命令,要求所有第5位为0的标签返回自己的第6位,此时阅读器只收到1位数据,且值为 NULL,没有发生碰撞, $i=i+1$,所以, $B[6]=NULL$, $Num[6]=6$ 。阅读器可以根据标签 ID 号的前5位数据唯一地确定标签2,且 $Slep_{\text{标签}1}=0$ 。

7)阅读器从堆栈 S 中取出 $B[4]=1$ 和 $Num[4]=4$,并发送 Request(1, 4)命令,要求所有第4位为1的待识别标签返回自己的第5位,此时阅读器只收到1位数据,且值为1,没有发生碰撞, $i=i+1$,所以, $B[5]=1$, $Num[5]=5$ 。阅读器可以根据标签 ID 号的前5位数据唯一地确定标签1。所以, $Q(2)_{\text{max}}=7$ 成立。

(3)假设有 N 个标签,且 D_{sum} 为最大值时, $Q(N)_{\text{max}}=2 \times N + 3$ 成立。当阅读器工作范围内第 $N+1$ 个标签进入阅读器的工作范围时,只需在原有二进制搜索树上增加2次搜索,所以, $Q(N+1)_{\text{max}}=Q(N)_{\text{max}}+2=2 \times N + 3 + 2=2(N+1)+3$ 。由此得到有 $N+1$ 个标签时结论也成立,即式(1)成立。

5 仿真及结果分析

本文不计控制、前后缀、校验冗余等开销,在理想信道条件下进行仿真。标签 ID 均匀分布,长度固定为 16 bit,标签的数量在 0~60 之间动态变化,对阅读器的搜索次数、阅读器和标签之间的识别通信量这2个性能指标进行分析,与 BS 算法、BBT 算法和 BBSB 算法进行比较。图2为改进算法与 BS 算法、BBT 算法和 BBSB 算法的阅读器搜索次数的比较。

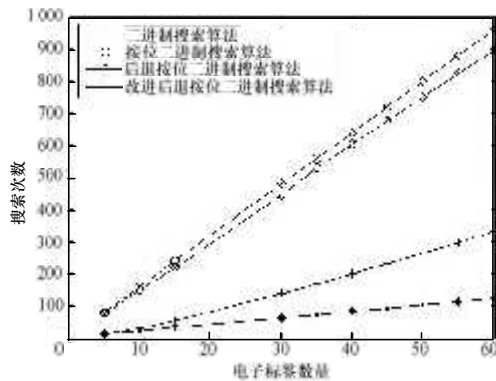


图2 4种算法搜索次数比较

从图2可以看出, BBSB 算法的搜索次数少于 BBT 算

法, BS 算法少于 BBT 算法和 BBSB 算法。改进算法的最大搜索次数明显少于其他3种算法。随着 N 的增大,改进算法的优势越明显。

图3为4种算法阅读器与标签之间识别通信量的比较。由图可以看出 BS 算法的识别通信量最大。BBSB 算法小于 BBT 算法。改进算法识别通信量的最大值明显小于其他3种算法。

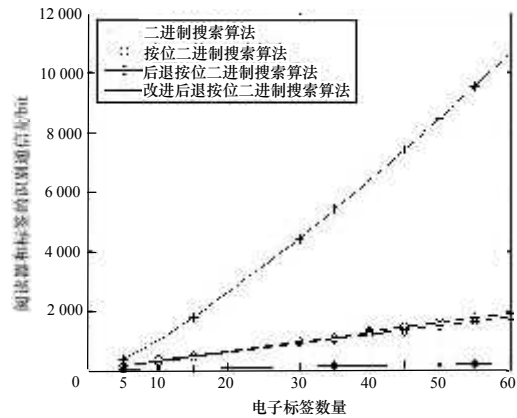


图3 4种算法识别通信量比较

6 结束语

理论分析和实验结果表明,本文算法在阅读器与标签之间的识别通信量及阅读器的搜索次数等方面都优于 BS 算法、BBT 算法和 BBSB 算法,可以获得更好的性能,具有良好的应用前景。

参考文献

- [1] Finkenzeller K. RFID Handbook Fundamentals and Applications in Contactless Smart Cards and Radio Frequency Identification and Near-field Communication[M]. 2nd ed. New York, USA: Wiley, 2003.
- [2] 尹君,何怡刚. 基于分组动态帧时隙的 RFID 防碰撞算法[J]. 计算机工程, 2009, 35(20): 267-269.
- [3] Cho H, Lee W, Baek Y. LDFSA: A Learning-based Dynamic Framed Slotted ALOHA for Collision Arbitration in Active RFID Systems[C]//Proc. of Conf. on Advances in Grid and Pervasive Computing. [S. l.]: IEEE Press, 2007: 655-665.
- [4] 王雪,钱志鸿. 基于二叉树的 RFID 防碰撞算法的研究[J]. 通信学报, 2010, 31(6): 50-57.
- [5] 康国斌. 无线射频识别系统中的防碰撞算法研究[D]. 上海: 上海交通大学, 2009.
- [6] 孙明. RFID 中防碰撞算法研究[D]. 合肥: 合肥工业大学, 2009.

编辑 张正兴

(上接第 279 页)

- [5] 王晟中,陈伟男,彭澄廉. 可重构计算硬件平台的改进设计[J]. 计算机工程, 2010, 36(5): 250-252.
- [6] Li Zhiyuan, Hauck S. Configuration Compression for Virtex FPGAs[C]//Proc. of IEEE Symposium on FPGAs for Custom Computing Machines. [S. l.]: IEEE Press, 2001.
- [7] Li Zhiyuan, Hauck S. Configuration Prefetching Techniques for Partial Reconfigurable Coprocessor with Relocation and Defragmentation[C]//Proc. of ACM/SIGDA Symposium on Field-programmable Gate Arrays. New York, USA: ACM Press, 2002.
- [8] 邹祎,吴强,赵远宁. 支持动态可重构硬件透明编程的预

- 配置调度[J]. 计算机工程与应用, 2008, 44(27): 52-55.
- [9] 梁樑,周学功,王颖,等. 采用预配置策略的可重构混合任务调度算法[J]. 计算机辅助设计与图形学报, 2007, 19(5): 635-641.
- [10] Chen Yu, Li Renfa, Wu Qiang. Automatic Reconfigurable System-on-Chip Design with Run-time Hardware/Software Partitioning[C]//Proc. of Conference on Computer-Aided Design and Computer Graphics. Huangshan, China: [s. n.], 2009.

编辑 张帆

