

改进的树型结构 RFID 防碰撞算法

王雪, 钱志鸿, 刘晓慧, 程超

(吉林大学 通信工程学院, 吉林 长春 130012)

摘 要: 将整棵大查询树分为若干分支, 在每一个分支内进行标签识别, 减少碰撞发生的概率。算法分为前缀确定和分支查询 2 个阶段, 在前缀确定阶段读取所有标签前缀, 每个前缀代表一个分支, 依次遍历每一分支; 查询过程采用后退式搜索策略和动态搜索方式。理论分析和仿真实验表明, 算法能够从查询次数、时延、吞吐率多方面改善 RFID 系统的整体性能。

关键词: RFID; 防碰撞; 树型结构; 前缀查询; 后退式搜索

中图分类号: TN92

文献标识码: A

Improved tree structure anti-collision algorithm of RFID

WANG Xue, QIAN Zhi-hong, LIU Xiao-hui, CHENG Chao

(School of Communication Engineering, University of Jilin, Changchun 130012, China)

Abstract: The whole big query tree is divided into several branches, and the work of tag recognition is executed in each branch, thus reducing the probability of tag collision. The proposed algorithm can be divided into prefix determination and branch query two phases. In the prefix determination phase, the prefix of every tag is identified, and each prefix denotes a branch. Traverse each branch by turns to recognize tags; in the branch query phase, backward search strategy and dynamic search method are used. Theoretical analysis and simulation experiment show that when the improved algorithm can improved the overall performance of RFID system from the aspects of the number of queries, time delay, and throughput.

Key words: RFID; anti-collision; tree structure; prefix query; backward search

1 引言

射频识别技术 (RFID, radio frequency identification) 是新时代信息科技的产物。它以无线电波技术为基础, 利用空间电磁耦合实现无线双向数据通信和对物品的自动识别^[1]。RFID 技术的迅猛发展得益于计算机、网络、无线数字通信、电磁学、密码学等多项科学技术的成熟, 其本身具有穿透性强、无需直接接触、读取数据方便快捷、自动化程度高、易于操作等突出特点。

在物联网 (IOT, internet of things)^[2,3] 整个体系架构中, 利用 RFID 技术采集和传输底层信息, 其

优势之一就是多目标识别, 即快速准确地识别可读范围内的大量标签。但由于物联网特有的海量数据环境, 多个标签在争用同一信道传输数据时不可避免地导致了数据碰撞问题的出现, 影响系统数据传输的完整性和效率。因而合理地解决 RFID 系统中的数据碰撞问题成为物联网研究过程中的重点和热点。

标签数据碰撞实质是无线通信技术中的多路存取问题。在 RFID 系统中一般采用时分多址 (TDMA) 技术解决, 以基于 ALOHA 的概率型算法和基于树型结构的确定性算法为两大主要研究方向。

基于 ALOHA 的概率型算法执行过程相对简

收稿日期: 2014-08-08; 修回日期: 2014-11-27

基金项目: 国家自然科学基金资助项目 (61371092, 61401175); 教育部高等学校博士学科专项科研基金资助项目 (20130061120062); 中国博士后科学基金资助项目 (2014M551184)

Foundation Items: The National Natural Science Foundation of China (61371092, 61401175); The Research Fund of the Doctoral Program of Higher Education of China (20130061120062); China Postdoctoral Science Foundation (2014M551184)

单, 易于实现, 适用于实时性要求不高的场合, 一般用于高频段^[4]。但随机性比较大, 当标签数量急剧增多时, 吞吐率将会大幅下降。同时存在着标签长时间不能被识别的可能, 即标签“饥饿”。典型的 ALOHA 算法以纯 ALOHA 算法、时隙 ALOHA(SA, slotted ALOHA) 算法^[5]、帧时隙 ALOHA(FSA, framed-slotted ALOHA) 算法^[6]和动态帧时隙 ALOHA(DFSA, dynamic framed-slotted ALOHA) 算法^[7]为代表。此类算法的核心思想是让标签尽量选择不同的时隙响应阅读器, 从而降低碰撞的发生。

基于树型结构的防碰撞算法是一种确定性算法, 不存在标签“饥饿”的情况, 能够达到 100% 的识别效率^[8]。此类算法由多周期的阅读器查询命令和标签回复组成, 整个算法流程可以形象地借用数据结构中的树型结构来表示。典型算法有二进制搜索树(BS, binary search) 算法^[9]、动态二进制搜索树(DBS, dynamic binary search) 算法、查询树(QT, query tree) 算法^[10,11]等。离散处理防碰撞混合法^[12]采用二叉树查询与离散处理相结合的方法, 降低查询次数。自调整混合树多标签防碰撞算法^[13]根据最高 2 个碰撞位的特征, 在不增加额外查询的条件下, 自调整搜索树的叉数, 从而避免了一些碰撞时隙和空闲时隙。自适应碰撞树协议(ACT, adaptive collision tree)^[14]定义寻呼串根据标签数量动态自适应分为 2 或 4 个分支, 以此减少寻呼次数。

基于 ALOHA 的概率型算法时延较少, 但存在标签长时间不能被识别的可能; 基于树型结构的确定性算法能够准确识别所有标签, 但花费的时延较多。因此, 一些学者开始考虑把 2 种算法结合在一起, 提出了一些新型的混合防碰撞算法。双时隙查询树算法(BSQTA, bi-slotted query tree algorithm)^[15]是在查询树算法的基础上引入了时隙的概念, 采用时间分离思想, 根据碰撞位的不同将标签分配在不同时隙, 减少传输数据和发送查询命令的个数。智能分组双时隙二进制树型算法^[16](IS-ESBT, enhanced slotted binary tree algorithm with intelligent separation)采取标签分组的方法, 标签查询前缀后的两位 ID 序列中, 如果“1”的个数为偶数, 则被分配到 0 时隙响应; 如果“1”的个数为奇数, 则被分配到 1 时隙响应。多时隙(MS, multi-slotted)二进制树型算法^[17]以查询树为基础, 将查询树中的每一个节点分为若干时隙, 符合阅读器查询命令的标签

随机选择一个时隙响应, 在不同时隙响应的标签彼此之间不会发生碰撞, 因此减少了标签回复阅读器命令时碰撞的发生。

本文针对大规模标签情况下, 整棵树查询将会产生较多的标签碰撞这一缺点, 提出了一种改进的二进制搜索树算法。该算法增加前缀确定阶段, 具有相同前缀的标签属于同一分支, 从而将一整棵大树分解为若干小的分支, 在每一个小分支内查询标签, 以减少阅读器的查询次数和标签碰撞发生的概率。同时在对每一个分支查询时, 采用后退式搜索策略和动态搜索方式, 以进一步提高整体系统性能。仿真结果表明, 改进算法在标签数量众多的环境下, 能够有效减少查询次数, 降低传输时延, 提高系统吞吐率, 性能良好, 具有一定应用价值。

2 改进的二进制搜索树算法

改进的二进制搜索树算法旨在将一棵庞大复杂的二叉树分成若干小分支, 对标签的查询在每一个分支中进行, 从而缩小产生碰撞的范围, 提高系统效率。算法的整个流程分为 2 个阶段。第一阶段是标签前缀的确定阶段, 为达到把整棵二叉树分为若干小分支的目的, 在对标签进行查询前, 增加前缀确定阶段。在此阶段, 阅读器根据标签回复的命令确定阅读范围内所有标签的前缀, 拥有相同前缀的标签同属一个分支, 此后的查询命令是以这些标签前缀为基础进行。第二阶段是依次对各个分支内标签的识别, 对每一个分支内标签的查询采用带有后退策略的动态二进制树型搜索算法。

2.1 前缀确定规则

阅读器在确定标签前缀时需遵循一定的方法, 具体过程为: 若欲查询的标签前缀长度为 l bit, 所有标签接收到查询命令后, 把自身标签 ID 的前 l 位二进制序列换算成相应的十进制数 t 。标签回复给阅读器的序列长度为 2^l , 根据之前换算的十进制数, 将该序列中的第 t 位置“1”(从 0 开始计数), 其他 $t-1$ 位置“0”。阅读器接收到标签回复命令后, 判断碰撞发生的位置, 把十进制数表示的碰撞位置的位数换算成相应的二进制数, 即可得到所有标签的 1 位前缀, 完成标签前缀的确定。例如阅读范围内 6 个标签的 ID 分别为: 00001011(T1)、00100101(T2)、00100011(T3)、00110111(T4)、11010001(T5)、11010100(T6), 要查询的前缀长度

为 3，在接收到阅读器发送的前缀查询命令后，所有标签将自身 ID 的前三位二进制序列转换成十进制数，分别得到 0、1、6。标签回复 8 bit 的二进制序列，分别将该序列的第 0、1、6 位置“1”，其余置“0”。阅读器接收到标签回复的二进制序列后，在第 0、1、6 位发生碰撞，对应的二进制序列为 000、001、110，由此得到标签前缀，如图 1 所示。

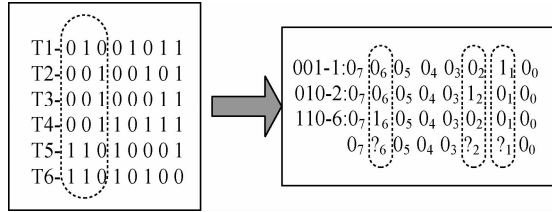


图 1 阅读器确定标签前缀示例

2.2 改进算法执行步骤

1) 阅读器广播查询 l bit 前缀的查询命令 Query(11...100...0)，其中连续“1”的个数为 l ，整个命令的长度与标签 ID 号的长度相等。

2) 若阅读器工作范围内有标签，则所有标签根据各自 ID 号的前 l 位回复长为 2^l bit 的二进制序列，该序列各比特位的具体内容按照之前所述规则进行设置；若阅读器工作范围内没有标签，转到步骤 1)。

3) 阅读器接收到标签回复序列后，根据碰撞位位置判断标签前缀，即确定了所有标签的前缀，每一个不同的前缀代表一个分支，并将这些前缀依次压入堆栈 S 中进行保存。

4) 弹出栈顶元素，阅读器发送包含该前缀序列的查询命令，即开始在该前缀代表的分支内进行标签查询，与此前缀相同的标签回复其剩余的 $k-l$ 位 ID 序列。如果有唯一标签响应，则正确识别该标签；如果发生碰撞，阅读器将最高碰撞位置 0，碰撞位之前不变，碰撞位之后舍去，形成新的查询命令，并发送。标签接收到该查询命令后，将其与自身 ID 比较，符合该查询命令的标签回复剩余的 ID 号。若还有碰撞发生，则按照如上所述规则生成新的查询命令并发送，直至只有唯一标签响应，正确读取该标签信息。每次正确读取标签信息后，返回到上一碰撞位，将碰撞位置 1，碰撞位之前不变，碰撞位之后舍去，形成查询命令并发送，即搜索碰撞节点的另一分支。如此循环，直到该分支的所有标签都被正确识别，转步骤 5)。

5) 若栈 S 非空，转步骤 4)；若栈 S 空，则所

有标签都已经被识别，算法结束。

整个算法的流程如图 2 所示。

3 算法性能分析

3.1 阅读器查询次数分析

阅读器所要读取的标签前缀长度为 l bit，根据如前所述在标签前缀确定阶段阅读器发送查询命令标签回复的规则，可能存在 2^l 个前缀。为减少传输比特数，前缀查询阶段标签回复的比特数应小于标签 ID 的长度，即 $1 < 2^l \leq k$ 。

设 $p_c(t, n, l)$ 、 $p_s(t, n, l)$ 、 $p_l(t, n, l)$ 分别代表 n 个标签中，长为 l 的前缀发生碰撞的概率、成功识别的概率和空闲的概率，则

$$p_l(t, n, l) = (1 - 2^{-l})^n \quad (1)$$

$$p_s(t, n, l) = \frac{n}{2^l} (1 - 2^{-l})^{n-1} \quad (2)$$

$$p_c(t, n, l) = 1 - p_l(t, n, l) - p_s(t, n, l), 1 < 2^l \leq k \quad (3)$$

由此可以得出空闲前缀的个数 N_l 、成功识别的前缀个数 N_s 和发生碰撞的前缀个数 N_c 分别是

$$N_l = 2^l p_l = 2^l (1 - 2^{-l})^n \quad (4)$$

$$N_s = 2^l p_s = 2^l \frac{n}{2^l} (1 - 2^{-l})^{n-1} = n(1 - 2^{-l})^{n-1} \quad (5)$$

$$N_c = 2^l p_c = 2^l \left[1 - (1 - 2^{-l})^n - \frac{n}{2^l} (1 - 2^{-l})^{n-1} \right], 1 < 2^l \leq k \quad (6)$$

在未来物联网的应用中，标签数量是海量的，即 n 的取值非常大，当 $n \gg 2^l$ 时， $(1 - 2^{-l})^n \rightarrow 0$ ， $(1 - 2^{-l})^{n-1} \rightarrow 0$ ，代入 N_c 的表达式，可以得到 $N_l \rightarrow 0, N_s \rightarrow 0, N_c \rightarrow 2^l$ 。说明当标签数量非常大时，即标签数量远大于可能存在的前缀时，几乎不存在空闲前缀和成功识别前缀，而几乎所有前缀都发生碰撞，则每一个前缀所代表的分支都有 $\frac{n}{2^l}$ 个标签。

改进的二进制搜索树算法分为前缀确定和分支查询 2 个阶段，2 个阶段阅读器的查询次数不同，因此对阅读器查询次数的分析也相应地分为 2 个阶段。在标签前缀的确定阶段，阅读器需要广播一个查询命令以获取标签信息；第二个阶段是对每一个分支内标签的查询。根据前面分析得出，每个分支都有 $\frac{n}{2^l}$ 个标签，因此可以以其中一个分支为例进行分析。

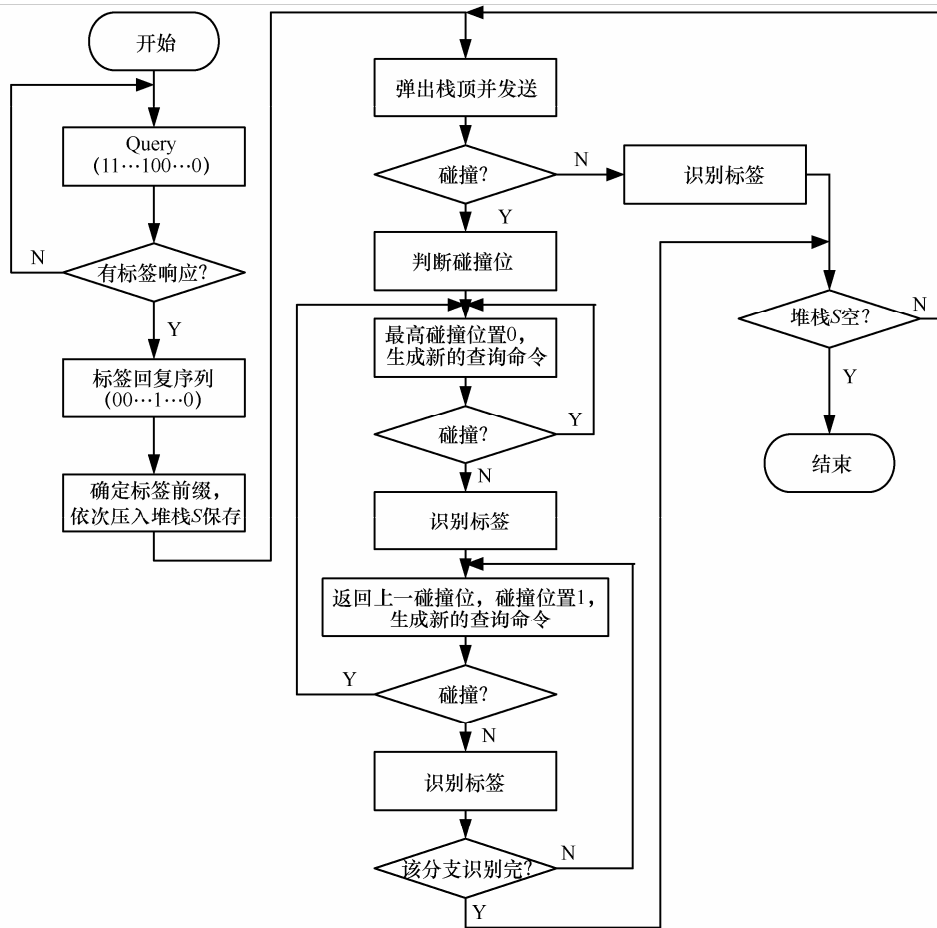


图 2 改进的二进制搜索树算法流程

对每一个分支的查询过程都可以用一棵二叉树表示，每个标签对应二叉树中的叶子节点，发生碰撞的位置对应二叉树中的分叉节点，对标签查询相当于对二叉树执行遍历操作，并且该二叉树中只有度为 2 和度为 0 的节点。对此，可以简单证明。

证明 由于在二叉树中每一个中间节点都代表一个碰撞位。发生碰撞后，都是将碰撞位置 0 和置 1，分为 2 个树枝，不存在只有一个树枝的情形，即中间节点的度数都是 2，没有度为 1 的中间节点。由此得到该二叉树中只有度为 0 和度为 2 的节点，不存在度为 1 的节点。

设对每个分支进行查询，阅读器所需发送的查询命令个数为 Q_d ，阅读器发送的每一个命令对应二叉树中的每一个节点，因此阅读器发送查询命令的个数等于二叉树中所有节点的个数，问题转化为求二叉树中所有节点的个数，其中该二叉树中叶子节点的个数为 $\frac{n}{2^l}$ ，树中只有度为 2 和度为 0 的节点。下面证明

搜索一个分支所需的查询次数 $Q_d = \frac{n}{2^{l-1}} - 1$ ，即二叉树中所有节点的个数为 $\frac{n}{2^{l-1}} - 1$ 。

证明 设在该二叉树中节点总数为 n ，根据前提条件，该二叉树中叶子节点个数等于标签个数 $\frac{n}{2^l}$ ，即度为 0 的节点 n_0 个数为 $\frac{n}{2^l}$ 。根据二叉树基本性质，对任意一棵二叉树，度为 2 的节点（叶子节点）总是比度为 0 的节点少一个，可以得到该二叉树中度为 2 的节点个数为 $n_2 = \frac{n}{2^l} - 1$ ，即发生碰撞的节点个数为 $\frac{n}{2^l} - 1$ 。由于该二叉树中只存在度数为 2 或 0 的节点，则

$$n = n_0 + n_2 = \frac{n}{2^l} + \frac{n}{2^l} - 1 = \frac{n}{2^{l-1}} - 1 \quad (7)$$

即 $Q_d = \frac{n}{2^{l-1}} - 1$ ，问题得证。

改进算法对应的搜索树共有 2^l 个分支，每一个

分支的搜索所需的查询次数 $Q_d = \frac{n}{2^{l-1}} - 1$; 前缀确定阶段需要一个查询命令, 可以得出

$$Q = 2^l Q_d + 1 = 2^l \left(\frac{n}{2^{l-1}} - 1 \right) + 1 = 2n - 2^l + 1 \quad (8)$$

其中, $1 < 2^l \leq k, n \gg 2^l$ 。

同时可知, 对每一个标签的查询平均需要的查询命令个数 Q_e 为

$$Q_e = \frac{Q}{n} = \frac{2n - 2^l + 1}{n}, 1 < 2^l \leq k, n \gg 2^l \quad (9)$$

3.2 传输时延分析

改进的二进制搜索树算法传输时延的长短取决于算法执行过程中阅读器和标签之间通信时传送比特位的多少, 传送的比特位越多, 则时延越大, 传送的比特位越少则时延越小。

在前缀确定阶段, 为确定标签的长为 l bit 的前缀, 阅读器首先广播长为 k bit 的前缀查询命令, 标签接收到命令后, 回复长为 2^l bit 的二进制序列。因此可以得出在前缀确定阶段, 信道上传送的比特长度 L_1 为

$$L_1 = k + 2^l \quad (10)$$

在分支查询阶段, 每次阅读器发送查询命令的长度与标签回复长度之和等于标签 ID 序列长度, 即为 k 。因此, 每一个分支中传输的数据比特长度等于阅读器发送查询命令的个数与标签 ID 号长度之积, 即

$$L_d = k Q_d = k \left(\frac{n}{2^{l-1}} - 1 \right) \quad (11)$$

可得搜索 2^l 个分支, 信道中一共需要传送的比特数 L_2 为

$$L_2 = 2^l L_d = 2^l k \left(\frac{n}{2^{l-1}} - 1 \right) \quad (12)$$

因此可得, 改进的二进制搜索树算法信道中所需传送的比特数总数 L 为

$$\begin{aligned} L &= L_1 + L_2 = k + 2^l + 2^l k \left(\frac{n}{2^{l-1}} - 1 \right) \\ &= k(2n - 2^l + 1) + 2^l \end{aligned} \quad (13)$$

设信道中数据传输速率为 v bit/s, 则可得改进的二进制搜索树算法传输时延 T 为

$$T = \frac{L}{v} = \frac{k(2n - 2^l + 1) + 2^l}{v} \quad (14)$$

平均每一个标签所需传输时延 L_e 为

$$L_e = \frac{T}{n} = \frac{k(2n - 2^l + 1) + 2^l}{vn} \quad (15)$$

3.3 吞吐量分析

RFID 系统采用改进的二进制搜索树算法后, 吞吐量 S 的表达式为

$$S = \frac{n}{Q} \times 100\% = \frac{n}{2n - 2^l + 1} \times 100\% \quad (16)$$

4 实验仿真与结果分析

本文采用 Matlab 仿真软件对该算法进行仿真。仿真参数设置为: 阅读器可读范围内标签个数 100~500, 且标签 ID 均匀分布, 每个标签都有长度为 64 bit 的唯一 ID 号, 信道中数据传输速率为 50 kbit/s, 选择标签前缀长度为 3 bit。分别在标签数量众多的情况下, 对阅读器查询次数、传输时延、系统吞吐量 3 个性能指标验证分析, 同时与 BS、DBS、ABD 算法^[15]、BLBO 算法^[8]、QT 算法^[10]进行比较。

4.1 阅读器查询次数仿真

由文献[18]可知, ABD 算法识别 n 个标签, 阅读器所需发送的查询命令个数 Q_{ABD} 为

$$Q_{ABD} = n(lbn - l + 1) + 2^l \quad (17)$$

识别每个标签阅读器平均所需发送的查询命令个数为

$$Q_{ABDe} = \frac{n(lbn - l + 1) + 2^l}{n} \quad (18)$$

在 Matlab 仿真平台下对各个算法进行仿真实验, 改进算法与 BS 算法、DBS 算法、ABD 算法、BLBO 算法、QT 算法所需的阅读器查询次数比较如图 3 和图 4 所示。

图 3 是识别阅读器可读范围内所有标签, 改进算法与 BS 算法、DBS 算法、ABD 算法、BLBO 算法和 QT 算法所需查询次数的比较, 其中图 3(a)是改进算法与以上 5 种算法的比较, 考虑到 QT 算法与其他算法相比性能差异较大, 从而导致其他算法之间的差异未能明了直观地显现, 因此图 3(b)为除 QT 算法以外的其他算法的详细比较。可以看出, 在标签数量众多的条件下, 随着标签数量的不断增加, 以上 5 种算法所需的阅读器查询次数都会越来越多, 但是改进的二进制搜索树算法所需的阅读器查询次数远远小于 QT 算法, 明显少于 BS 算法、DBS 算法和 ABD 算法, 并且标签数量越多, 差距

将会越明显。BLBO 算法是性能较为优异的 RFID 防碰撞解决方案，在查询次数上与本文所提算法相近似，但理论分析上其查询次数仍略多于本文所提的改进算法。

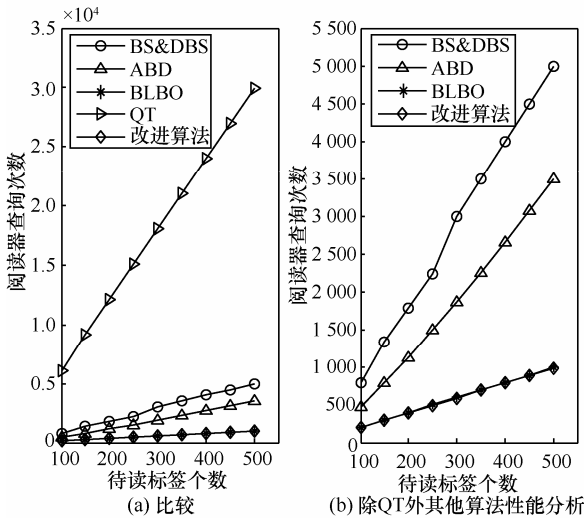


图 3 识别所有标签所需查询次数比较

图 4 是对平均识别每一个标签所需查询次数的仿真验证，从图中可以看出，改进算法所需的查询次数少于 BS 算法、DBS 算法、ABD 算法、BLBO 算法和 QT 算法。改进算法所需的识别一个标签所需的平均查询次数大致趋于平稳，均值在 2 左右，这是由于在标签数量众多的情况下，式(10)约等于 2，即识别一个标签，阅读器平均需要发送 2 个查询命令。由此可以得出，改进的二进制搜索树算法在阅读器查询次数方面的性能优于 BS 算法、DBS 算法、ABD 算法和 QT 算法，在标签数量较少时，本文所提算法略优于 BLBO，随着标签数量的增大，二者性能近似。

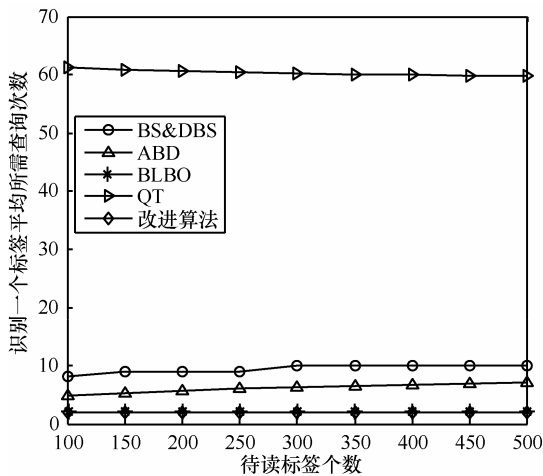


图 4 识别一个标签平均所需查询次数比较

4.2 传输时延仿真

对改进的二进制搜索树算法、BS 算法、DBS 算法、ABD 算法、BLBO 算法和 QT 算法执行过程中的时延进行仿真验证比较，结果如图 5 和图 6 所示。

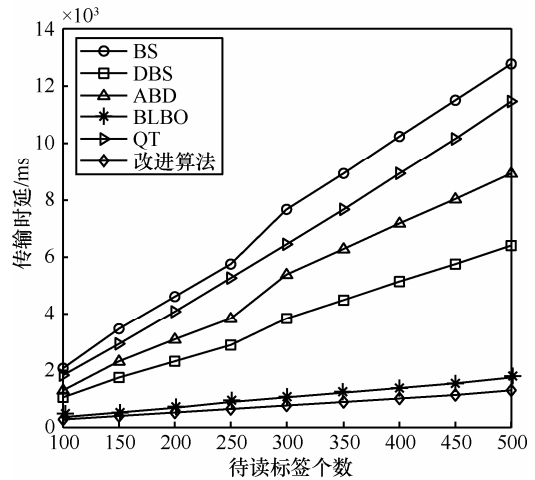


图 5 识别所有标签系统所需时延比较

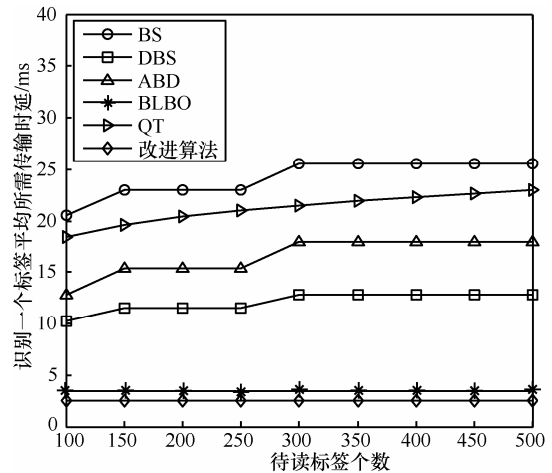


图 6 识别一个标签系统平均所需时延比较

本文所提算法无论在总传输时延，还是单个标签传输时延方面，都优于其他 5 种防碰撞算法，随着标签数量的增加，性能改善越明显。因此，结合仿真可以证明本文所提算法具有更少的查询次数和更低的传输时延。由此可见，对 RFID 防碰撞系统而言，本文所提算法减少了寻呼的标签数量，每次寻呼所需时间也大幅度降低，因此算法运行所需总时间将明显降低，算法的时间复杂度更低。

4.3 系统吞吐量仿真

对 BS 算法、DBS 算法、ABD 算法、BLBO 算法、QT 算法以及改进的二进制搜索树算法的吞吐量进行仿真验证，结果如图 7 所示。

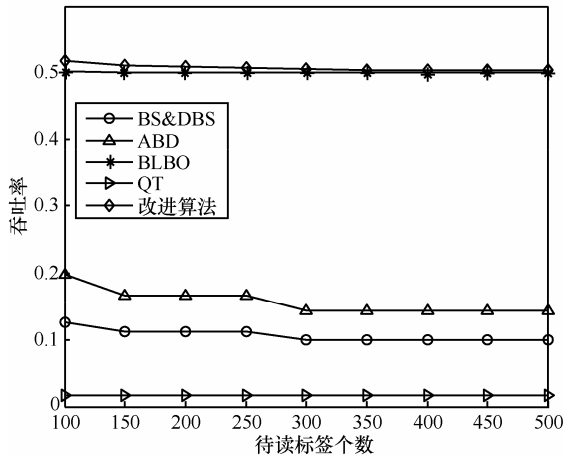


图 7 吞吐率比较

由图 7 可知，在标签数量众多的情况下，改进的二进制搜索树算法的吞吐率高于 50%，并且趋势平稳；QT 算法的吞吐量最低，性能较差，BS 算法和 DBS 算法的吞吐率接近于 10%，ABD 算法吞吐率高于 BS 算法、DBS 算法，约为 15%，BLBO 算法吞吐量远高于 BS 算法、DBS 算法和 ABD 算法，约为 35%，改进的二进制搜索树算法吞吐率高于 BS 算法、DBS 算法 40% 左右，高于 ABD 算法 35%，优于 BLBO 算法约 4%，也就是说，采用改进算法的 RFID 系统具有更强大的数据传输和处理能力，相同条件下，能够正确识别更多的可读范围内的标签信息，因此系统的整体性能更好。

为了进一步验证和比较改进算法吞吐率方面的性能，本文同时对概率性算法中的纯 ALOHA 算法、时隙 ALOHA 算法、帧时隙 ALOHA 算法的吞吐率进行仿真验证，结果如图 8 和图 9 所示。

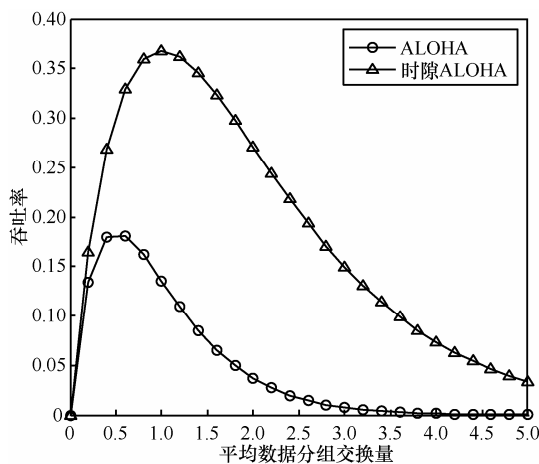


图 8 纯 ALOHA 算法和时隙 ALOHA 算法吞吐率

图 8 和图 9 清晰地展示出了纯 ALOHA 算法、

时隙 ALOHA 算法、帧时隙 ALOHA 算法吞吐率的变化趋势。可以看出，随着平均数据分组交换量和待读标签个数的变化，纯 ALOHA 算法在 $G=0.5$ 时达到最大吞吐率，为 18.4%，时隙 ALOHA 算法在 $G=1$ 时达到最大吞吐率，为 36.8%，帧时隙 ALOHA 算法在帧长与待读标签个数相等时达到最大吞吐率，为 36.8%。在非极值点处，吞吐率会出现较大程度的下滑。改进的二进制搜索树算法的吞吐率变化比较平稳，且高于 ALOHA 算法、时隙 ALOHA 算法、帧时隙 ALOHA 算法的最大值。

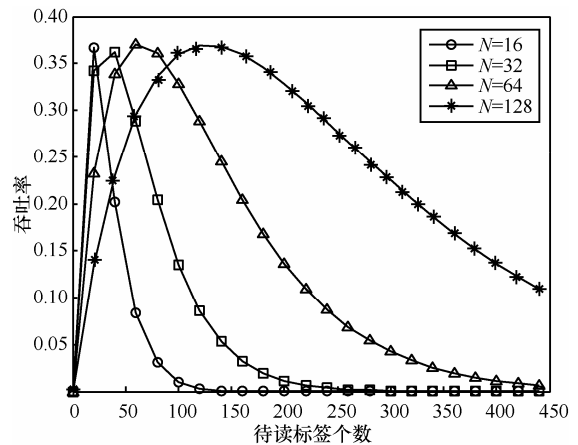


图 9 帧时隙 ALOHA 算法吞吐率

5 标签前缀长度的选择与分析

由于改进算法需要首先确定标签前缀，同时根据第 4 节中对改进算法性能的分析可以得出，阅读器读取的标签前缀长度 l 与查询次数、传送的比特数以及系统吞吐率三方面性能指标密切相关，因而将对整个系统的综合性能产生较大影响。

为进一步观察标签前缀长度对算法性能的影响，本文在标签数量相同的情况下选取不同的 l 值进行仿真实验，以观察不同 l 值下阅读器查询次数、传输时延以及吞吐率的变化。如前所述，查询前缀长度的取值应该满足 $1 < 2^l \leq k$ 。仿真时分别取标签数量为 100、200、300，结果如图 10~图 12 所示。

根据图 10~图 12 所示仿真结果，可以看出，在标签数量众多的环境下，同时满足条件 $1 < 2^l \leq k$ 和 $n \gg 2^l$ ，随着查询前缀长度 l 的增大，阅读器的查询次数和传输时延逐渐变小，吞吐率变大，说明查询前缀长度 l 越大，算法性能越好，系统整体性能越能够得到提高。然而过大的前缀长度，也会在能耗上对系统产生影响与负担，因此可以在满足前提条件

$1 < 2^l \leq k$ 和 $n \gg 2^l$ 的情况下, 取查询前缀长度 l , 在不明显增加能耗的同时, 使算法的防碰撞性能与效率达到最优。

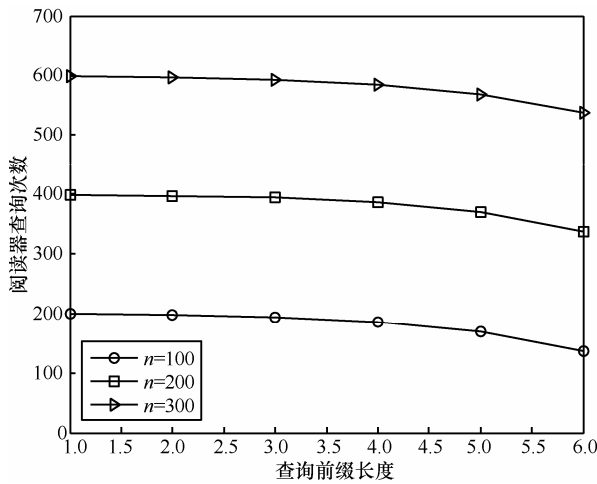


图 10 查询次数随查询前缀长度 l 的变化趋势

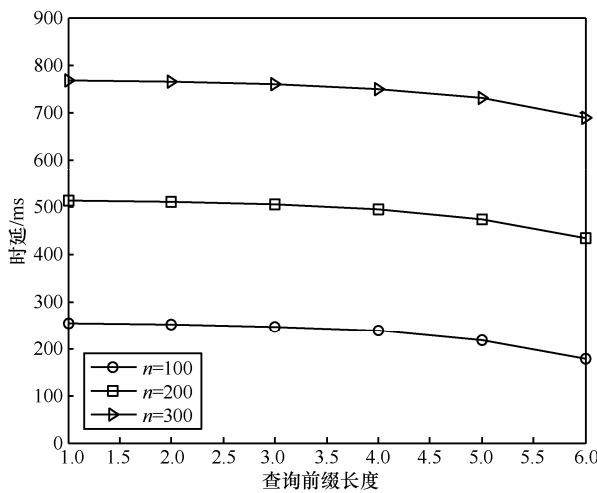


图 11 传输时延随查询前缀长度 l 的变化趋势

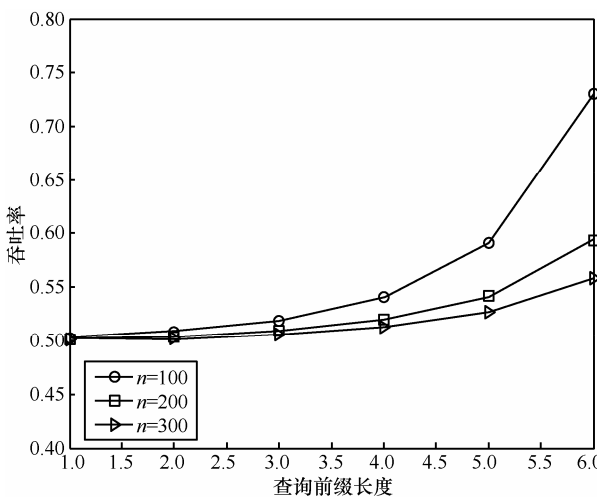


图 12 吞吐量随查询前缀长度 l 的变化趋势

6 结束语

本文以减少二进制搜索树算法中的碰撞次数为立足点, 同时考虑减少传输时延, 提出了一种改进的二进制搜索树算法, 该算法将所有标签的识别过程分为前缀确定和分支查询两个阶段。在前缀确定阶段中, 阅读器通过发送前缀查询命令, 根据标签回复确定可读范围内所有标签的前缀, 并把前缀压入堆栈保存, 具有相同前缀的标签在同一分支上, 旨在将一整棵庞大的搜索树分成若干分支, 减少碰撞的发生; 第二阶段依次弹出栈顶元素, 以此前缀为起点对每一个分支的标签进行查询, 查询过程中采用后退策略, 阅读器和标签动态发送比特位, 达到减少查询次数和降低时延的目的, 从而使整个系统性能有所提高。同时, 对改进算法的查询次数、传输时延、吞吐率三方面性能进行了理论分析, 并在 Matlab 仿真平台上对改进的算法进行仿真验证, 进一步证实了大规模标签情况下改进算法的优越性能。

参考文献:

- [1] 单承颖, 单玉峰, 姚磊等. 射频识别 (RFID) 原理与应用[M]. 北京: 电子工业出版社, 2010.
SHAN C G, SHAN Y F, YAO L, *et al.* Radio Frequency Identification (RFID) Theory and Application [M]. Beijing: Electronic Industry Press, 2010.
- [2] 钱志鸿, 王义君. 面向物联网的无线传感器网络综述[J]. 电子与信息学报, 2013, 35(1): 215-227.
QIAN Z H, WANG Y J. Internet of things-oriented wireless sensor networks review [J]. Journal of Electronics and Information Technology, 2013, 35(1): 215-227.
- [3] 钱志鸿, 王义君. 物联网技术与应用研究[J]. 电子学报, 2012, 40(5): 1023-1029.
QIAN Z H, WANG Y J. IoT technology and application [J]. Chinese Journal of Electronics, 2012, 40(5): 1023-1029.
- [4] 李萌, 钱志鸿, 张旭等. 基于时隙预测的 RFID 防碰撞 ALOHA 算法[J]. 通信学报, 2011, 32(12): 43-50.
LI M, QIAN Z H, ZHANG X, *et al.* Slot-predicting based ALOHA algorithm for RFID anti-collision[J]. Journal on Communications, 2011, 32(12): 43-50.
- [5] ISO/IEC FDIS 18000-6. Information Technology-Radio Frequency Identification for Item Management-Part 6: Para-Meters for air interface Communications at 860 MHz to 960 MHz[S].
- [6] VOGT H. Multiple object identification with passive RFID tags[A]. 2002 IEEE International Conference on Systems, Man and Cybernetics[C]. 2002, 3: 651-656.
- [7] LEE D, CHOI J, LEE W, *et al.* A time-optimal anti-collision algorithm for FSA-based RFID systems[J]. ETRI Journal, 2011, 33(3): 458-461.
- [8] 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 X, QIAN Z H, HU Z C. Research on RFID anti-collision algorithms based on binary tree [J]. Journal on Communications, 2010, 31(6): 49-57.

- [9] FINKENZELLER K. RFID Handbook: Radio-Frequency identification Fundamentals and Applications(2nd ed)[M]. John Wiley and Sons Ltd, 2003.
- [10] LAW C, LEE K, SIU K Y. Efficient memoryless protocol for tag identification[A]. Proc 4th International Workshop on DIALM[C]. Boston, 2000.75-84.
- [11] LANDALUCE H, PERALLOS A. A fast RFID identification protocol with low tag complexity[J]. IEEE Communications Letters, 2013, 17(9): 1704-1706.
- [12] 李平, 孙利民, 吴佳英等. 基于可离散处理的 RFID 防碰撞混杂方法研究[J]. 通信学报, 2013, 34(8): 10-17.
LI P, SUN L M, WU J Y, *et al.* Research on RFID anti-collision hybrid algorithm based on discrete processing [J]. Journal on Communications, 2013, 34(8): 10-17.
- [13] 宋建华, 郭亚军, 韩兰胜等. 自调整混合树 RFID 多标签防碰撞算法[J]. 电子学报, 2014, 42(4):685-689.
SONG J H, GUO Y J, HAN L S, *et al.* Anadjustive hybrid tree anti-collision algorithm for RFID multi-tag identification [J]. Chinese Journal of Electronics, 2014, 42(4):685-689.
- [14] LIU X H, QIAN Z H, ZHAO Y H, *et al.* An adaptive tag anti-collision protocol in RFID wireless systems[J]. Communications China, 2014, 11(7):117-127.
- [15] CHOI J H, LEE D W, LEE H J. Bi-slotted tree based anti-collision protocols for fast tag identification in RFID systems[J]. IEEE Communications Letter, 2006, 10(12): 861-863.
- [16] KIM S, KIM Y, AHN K. An enhanced slotted binary tree algorithm with intelligent separation in RFID systems[A]. IEEE Symposium on Computers and Communications(ISCC 2009)[C]. 2009.237-242.
- [17] NAMBOODIRI V, GAO L X. Energy-aware tag anti-collision protocols for RFID systems[J]. IEEE Transactions on Mobile Computing, 2007, 9(1): 44-59.
- [18] 丁治国. RFID 关键技术研究与实现[D]. 合肥: 中国科学技术大学, 2009.
DING Z G. Research and Realization on Key Technologies of RFID

[D]. Hefei: University of Science and Technology of China, 2009.

作者简介:



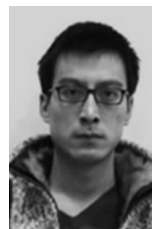
王雪(1984-), 女, 吉林白山人, 博士, 吉林大学讲师, 主要研究方向为无线传感器网络技术、RFID(射频识别)技术、超宽带通信技术和物联网等。



钱志鸿(1957-), 男, 吉林长春人, 吉林大学教授, 博士生导师, 主要研究方向为近程无线网络通信技术、无线传感器网络技术、RFID(射频识别)技术、UWB(超宽带)通信技术和物联网等。



刘晓慧(1989-), 女, 吉林长春人, 吉林大学硕士生, 主要研究方向为射频识别技术。



程超(1984-), 男, 吉林长春人, 吉林大学博士生, 主要研究方向为无线网络技术。