RFID 二进制树型折半搜索防碰撞算法①

廖剑锋, 刘宇航

(华中科技大学 文华学院, 武汉 430074)

摘 要: 为提高 RFID 系统中多标签读取的工作效率, 分析了二进制树型搜索防碰撞算法, 提出一种改进型的二进制树型折半搜索算法. 通过构建哈夫曼树, 使用自定义编码, 采用堆栈操作确定碰撞位, 用折半查找方式减少重复路径访问量, 并最终确定最短路径长度. 数据分析及实验结果表明, 二进制树型折半搜索防碰撞算法可以明显降低搜索深度, 显著提高 RFID 多标签读取的工作效率.

关键词: 射频识别; 防碰撞; 二进制树; 路径长度; 堆栈; 折半搜索

RFID Anti-Collision Algorithm Based on Binary Tree Half Search

LIAO Jian-Feng, LIU Yu-Hang

(School of Wenhua, Huazhong University of Science & Technology, Wuhan 430074, China)

Abstract: Aiming at improving multiple tags reading efficiency in RFID, it analyses the binary search anti-collision algorithm, and puts forward an improved binary tree half search algorithm. By constructing a Huffman tree, using a custom code, using the stack operation to determine the collision bit, with a half search method to reduce duplication of path, ultimately determines the shortest path length. Data analysis and experimental results show that, the binary tree half search anti-collision algorithm can significantly reduce the search depth, and improve the work efficiency of RFID tag reading.

Key words: RFID; anti-collision; binary tree; path length; stack; half search

1 引言

在射频识别(RFID)系统中, 数据传输的完整性存在两个方面问题: 一是各种外界干扰可能使数据传输产生错误, 二是多个标签 Tag(也称为应答器)同时争用信道使数据产生碰撞. 运用差错检测技术可以检测和发现错误数据, 运用防碰撞算法可以解决数据碰撞问题. RFID 系统在工作时, 可能会有一个以上的标签同时处于阅读器(Reader)覆盖范围内. 这样, 如果有两个或两个以上的标签同时给阅读器发送数据, 因数据争抢信道而相互干扰, 产生冲突, 也就是碰撞. 当有多个阅读器覆盖范围之内的多个标签同时发送数据时, 数据碰撞问题就更为突出.

RFID 系统通常采用时分多址(Time Division Multiple Access, TDMA)技术解决信道争用问题, 其算法主要包括帧时隙 ALOHA 算法和二进制树型搜索算法[1]. 传统的 ALOHA 算法简单, 但识别效率较低, 不适合大规模标签同时读取. 一些学者对其进行改进, 提出了一些新算法. 比如, 针对 ALOHA 算法中时隙空白现象, 周信、刘晔设计出一种基于码距反演和时隙预定的防碰撞算法[2]. 针对传统 ALOHA 协议算法弊端, 吴海峰提出动态二进制树时隙协议、自适应二进制树时隙协议和分裂二进制树时隙协议三个新的防碰撞协议[3]. 针对主动标签环境, 王荃等人提出一种基于仲裁集的使用多频道资源的防碰撞 MCMA 协议[4]. 二进制树型搜索算法比较复杂, 耗时较长, 比 ALOHA 算法识别度和吞吐率高, 所以适合于大规模标签应用场景. 为提高二进制树型搜索算法效率, 很多学者对之进行改进. 比如, 针对使用 RFID 技术的隧道人员定位系统, 米根锁等学者提出一种基于后退策略去除搜索信息冗余位的二进制树型搜索改进算法[5], 该算法使搜索次数和碰撞

① 基金项目:华中科技大学文华学院培育基金(2013py04)
收稿时间:2013-04-22;收到修改稿时间:2013-05-27

概率显著降低, 缩短了传输时间. 针对查询树搜索算法在标签数量较多识别效率较低的问题, 陆冰清等人提出一种新的动态多叉树查询算法[6], 该算法充分利用曼彻斯特编码可以识别碰撞位的特性, 动态调整搜索叉树, 可以有效提高搜索效率和时隙吞吐率. 熊昌慧等人提出一种基于粒子群优化 PSO 的改进自适应分组和预发信号 ASPS 的算法[7], 通过将接收信号二维化, 建立粒子群优化算法模型, 从而改进 ASPS 算法, 该算法运算时间短, 处理碰撞次数少. 张文欣提出一种基于后退式二进制搜索算法[8], 通过动态调整发送指令长度, 有效减少搜索次数和传输工作量, 在无碰撞时采用后退策略快速识别标签. 李飞等人提出一种基于平衡不完全区组设计 BIBD 编码的二进制搜索改进算法[9], 通过引入标签多状态机制, 对标签进行逐节识读, 在标签较多或标签 ID 较长方面识别优势明显.

本文研究了传统二进制树型搜索算法和各种改进算法, 针对被动式标签环境, 提出一种新的改进的二进制树型折半搜索算法.

2　二进制树型搜索算法

文献[10]描述了传统二进制树型搜索算法, 国际标准化组织也制定了 ISO/IEC 14443 防碰撞协议. RFID 系统二进制搜索算法采用曼彻斯特编码方法, 理想情况下要求阅读器覆盖范围内所有标签均可同时对阅读器做出应答. 为方便后文实例描述, 这里引入 Request(SN)、Pause(SN)、Stack-Push(Site)和 Stack-Pop()命令作为前提条件:

① Request(SN)-询问(序号): 阅读器发送一个序列给作用范围内的所有标签, 范围内的电子标签接到询问后将其与自身序列比较. 若自身序列小于或等于这个询问序列, 标签则以自身序列做应答处理.

② Pause(SN) – 暂停(序号): 使包含该序列的标签直到下次激活后才“苏醒”. 在本次进入阅读器有效的范围内, 不再理会阅读器的任何请求.

③ Stack-Push(Site, Bit) – 压入碰撞栈(碰撞比特位, 比特值): 将对应的碰撞比特位最高位压到栈中. 栈的最大深度为对应标签的最大序列长度.

④ Stack-Pop() – 弹出碰撞栈: 将碰撞栈的栈顶碰撞位弹出栈.

2.1 二进制树型搜索算法工作流程

整体流程图如图 1 所示. RFID 开始工作时, 阅读器发射广播询问 Request, 阅读器分析范围内所有标签的应答信号. 当检测到碰撞时, 找到最高位碰撞位置 1, 并将位置信息压栈, 再置其位为 0, 其余碰撞位置 1 组成新的序列 SN 后执行 Request(SN)命令. 当处理完一个标签后, 弹出栈内信息后, 修改序列重新询问即可. 当阅读器询问新序列, 如接收到的标签应答序列仍出现碰撞, 则反复压栈直到确定唯一标签. 如仅有一个标签应答, 则接收这个标签的数据. 然后系统设定标签在该阅读器作用范围内, 并且不再对阅读器的询问做出响应. 同时将碰撞栈栈首元素出栈, 寻找同根节点的其它标签. 重复上述所有操作, 直到遍历全树. 读取完该范围内所有的标签后(碰撞栈为空). 阅读器恢复发送广播询问.

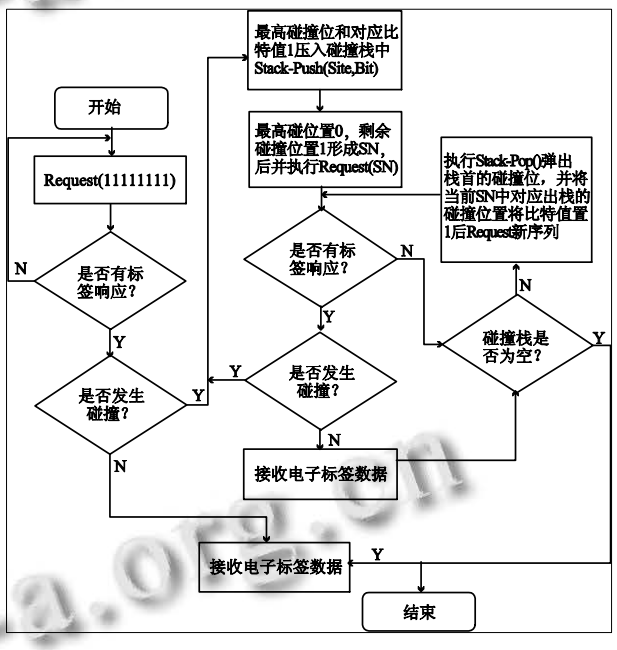

图 1　二进制树型搜索防碰撞图

2.2 二进制树型搜索防碰撞算法实例

有 4 个均为 8 位长度的电子标签, 采用曼彻斯特编码, 分别为 10100101、10101101、11010101 和 11101101, TagX 中 X 表示一个碰撞比特位. 与此对应, 规定用于存放碰撞位的栈为 8 个单位空间.

如图 2 所示, 为阅读器 Request 树型示意图. 流程概述如下:

① 阅读器发出广播询问 Request(11111111), 范围内标签与自身做出应答. 如果小于等于请求序列, 则将自身的序列进行应答. 此处标签均做出应答, 阅读器获取应答后判别发生碰撞.

② 通过Manchester编码，得到对应的碰撞比特位为 $D_3D_4D_5D_6$. 先执行 Stack-Push(3,1)操作，将对应的碰撞位最高位 D_3 置 1 后压入栈中. 再将 D_3 置 0，其余碰撞位置 1，产生新的序列. 执行 Request(11110101).

③ 因Tag1和Tag3标签小于询问序列，以自身SN应答. 阅读器检验出碰撞，最高碰撞位为 D4. 执行 Stack-Push(4,1)命令后，将 D4 位置 0，其余碰撞位置 1，再次形成新的序列 11100101. 执行 Request 命令.

④ Tag1 小于询问序列做出应答，没有发生碰撞. 阅读器获取到 Tag1 序列，同时执行 Pause(11100101)和 Stack-Pop()命令. Stack-Pop()命令将栈顶元素弹出，因而产生新询问序列 11110101，执行 Request 命令.

⑤ Tag3 应答，未出现碰撞. 接着执行 Pause 和 Stack-Pop().

⑥ 以此类推直到碰撞栈为空栈，阅读器重新发送广播询问.

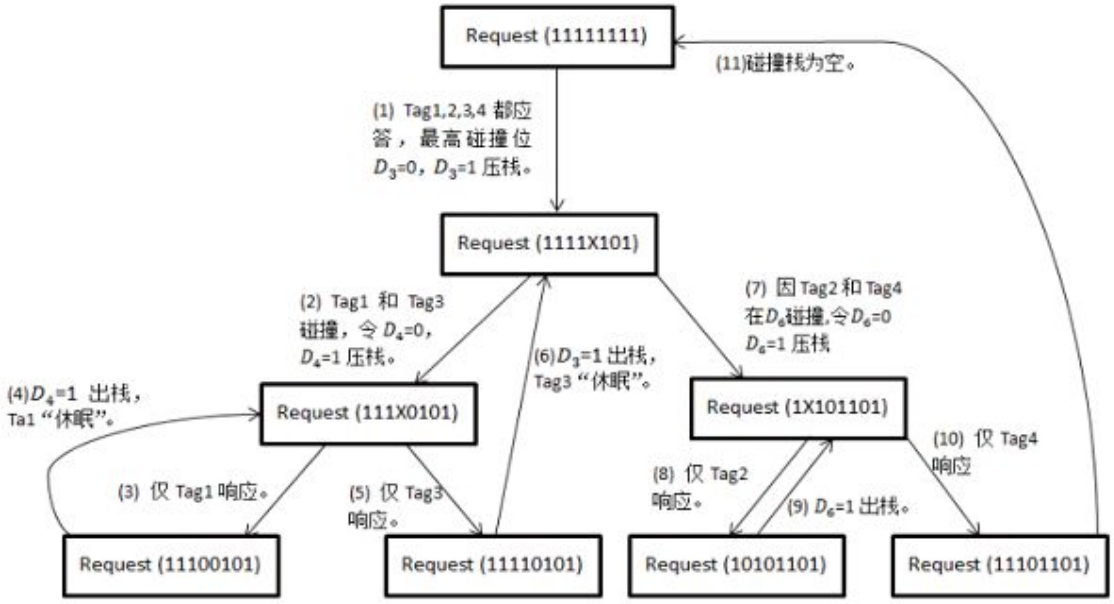

图 2 阅读器 Request 树型图

3 二进制树型搜索防碰撞算法改进

3.1 算法改进思路

可以将二进制树型搜索防碰撞算法完成路径看成二叉树结构，并以此作为算法改进思路. 哈夫曼树又称为最优二叉树，是一种加权路径长度最短的树. 可以通过模仿构建哈夫曼树来提高算法的工作效率. 但构建哈夫曼树的前提是必须已知所有项出现的频数，即碰撞位碰撞的次数，这在实际应用中是不可能的，因为知道了碰撞位碰撞的次数，也就知道了每个电子标签所发出的序列. 经过仔细分析二叉树结构，可以将折半查找的方式融入算法改进中. 二进制树型搜索算法可以采用自行定义编码来尽可能寻找出路径最短的树，这里将这种改进算法称为二进制树型折半搜索算法.

3.2 二进制树型折半搜索算法

根据折半性质定义编码 X_n，以 X 表示发生了碰撞的位，n 定义为碰撞位的个数. 当 n=1 即为 X1 时，即碰撞位只有一位，0 或 1，故定义为 1. 当 n=2 即为 X_2 时，因碰撞位变为两位，则有 00、01、10、11 这四种可能，取半定义为 10. 当 n=3 即为 X_3 时，因 X_3 可以折半分解成为 X_1+X_2，则在前取高位定义编码为 101. 同理得到 X_4 为 X_2+X_2=0101. X_5=01101、X_6=101101、X_7=1010101、X_8=01010101. 如图 3 所示.

低位 高位

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
X_1								1
X_2							0	1
X_3						1	0	1
X_4					0	1	0	1
X_5				0	1	1	0	1
X_6			1	0	1	1	0	1
X_7		1	0	1	0	1	0	1
X_8	0	1	0	1	0	1	0	1

图 3 二进制树型折半搜索定义图

当检测到碰撞，就不再发送整个序列，而是仅发送碰撞位及对应的十六进制碰撞位位数. 如检测到发生 1 次碰撞，碰撞位为 1 则发送 Request(X_1,1)，如果第 3 次碰撞，碰撞位 0、1 和 7 发送 Request(X_3,710)，以此类推每次发送的信息. 当 Request 前后信息相等且不为广播序列时(即 Request(X_3,710)为前次发送询问，发送后得到反馈信息且等于 Request(X_3,710)说明碰撞位的折半信息不存在对半，仅存在树的一边，而非划分为两边)，此时将采用原二进制树型搜索防碰撞算法来处理本次操作. 将最高碰撞位置 0，其余碰撞位置 1，小于此询问序列的标签以自身序列对阅读器进行应答. 重新检查碰撞位再次利用二进制树型折半搜索算法，继续沿用原二进制树型搜索算法压栈方式，减少重复路径的访问量. 当找到子节点为空或是未发生碰撞时，如果栈内不为空，则将栈顶元素出栈. 将 X_n 所对应的所有位均置 1，则可以访问同根节点的右边子树或叶子节点. 流程图如图 4 所示.

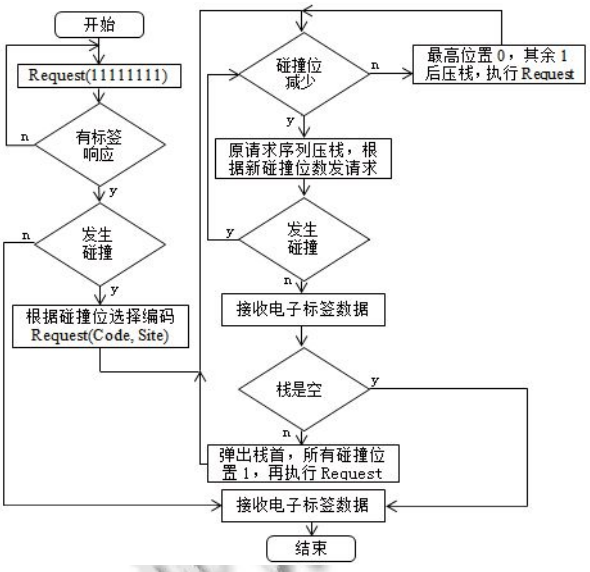

图 4　二进制树型折半搜索算法流程图

此改进算法的最优情况是：询问路径最终路径树与哈夫曼树一样，此时阅读器检测碰撞效率最高. 该改进算法的最差情况是，碰撞位多且碰撞的标签多的情况下，所有产生的碰撞位为两两相互仅有一位碰撞，折半搜索法与原算法效率相同，即树的深度相同且均在一侧. 如果碰撞的标签数量较少和标签的曼彻斯特的编码差值小且整体偏低或偏高的情况下，改进算法的效率基本等于原算法的效率.

3.3 二进制树型折半搜索算法实例

在二进制树型折半搜索算法基础上，现通过实例分析算法可行性. 如图 5 所示，首先定义电子标签的曼彻斯特编码. 为模拟实际情况，将电子标签设定为 8 位并且全部发生碰撞，设定电子标签数为 7 个. 分别定义 Tag1: 11010110, Tag2: 11001101, Tag3: 10000100, Tag4: 10000010, Tag5: 00000110, Tag6: 11110110 和 Tag7: 00100111.

位数	7 6 5 4 3 2 1 0
Tag1	11010110
Tag2	11001101
Tag3	10000100
Tag4	10000010
Tag5	00000110
Tag6	11110110
Tag7	00100111

图 5　8 位 7 个电子标签曼彻斯特编码定义图

如图 6 左所示，按照原算法获取到解决碰撞后的路径图. 在原命令约定下添加一条新命名 Request (Code, Site)，将 Stack-Push(Site, Bit)修改为 Stack-Push (Command). 再利用二进制树型折半搜索算法获取路径. 流程概述如下:

① 阅读器发送广播询问，在该范围内的七张电子标签接收到广播询问，以自身序列做出应答响应.

② 阅读器接收到应答数据，检测到发生碰撞，且碰撞位的个数为 8. 选用自定义编码 X_8(01010101)执行 Request(X_8,76543210).

③ 标签检测到询问命令不为广播命令，则获取十六进制比较位，将自身序列对应位与 X_7 进行比较，如包含，则以自身序列做出应答响应. 根据响应得到新的碰撞位个数.

④ 将新碰撞位个数与发送前相比较. 如果碰撞位个数减少则将发送前的命令进行压栈处理，然后根据新碰撞比特位个数选用定义编码. 此处将 Request(X_8,76543210)命令压入栈中. 根据新碰撞位个数为 6，发送命令 Request(X_6,765421).

⑤ 阅读器读取标签应答信息，检测到 2 个碰撞位. 将 Request(X_6,765421)压入栈后执行 Request(X_2,21)命令.

⑥ 得到标签应答，未发生碰撞. 执行 Pause()命令，Tag3 进入“休眠”状态. 因栈不为空，则 Stack-Pop()弹出栈顶元素，将 Xn 对应的全部位均置 1，就可以得到

同根节点的右侧子树. 再以相同方式识别全部标签.

⑦ 最终得到的路径图如图 6 右所示. 分析图 6 可以明显看出, 8 位 7 个标签改进前后二叉树在深度上明显不同. 根据公式 $PL=\sum_{k=1}^{n} lk$ (n 为节点数, l 为对应节点路径长度), 可以计算和比较改进前后路径长度 PL, 传统二进制树型搜索算法的 PL=34, 改进后的二进制树型折半搜索算法 PL=30, 改进后的 PL 值比改进前减少 4.

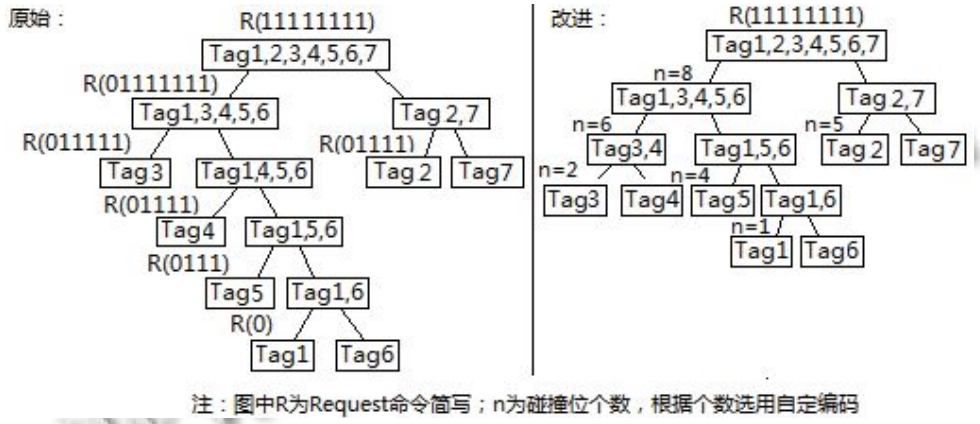

图 6 改进前后路径图

4 实验

4.1 实验环境

本实验主要用于测试二进制树型折半搜索防碰撞算法的有效性, 对传统算法和改进后算法进行比较. 实验设备采用北京烽火联拓公司生产的 RFID 阅读器和有源电子标签, 如图 7(a)和图 7(b)所示. 标签工作模式使用被动式, 要求上位机主动从读写器获取 RFID 数据. 阅读器使用 2.5GHz 工作频率, 天线增益范围 1-31dB, 传输波特率为 9600. 上位机软件系统类型为 X86-based PC; 处理器为 AMD Athlon(tm) 64 Processor 3200+; 操作系统为 Windows2003 Server SP2; 物理内存为 512 MB. 需要说明的是, 本实验目的是测试单阅读器条件下的多标签防碰撞改进算法. 对于多阅读器条件下的多标签防碰撞算法, 因其涉及到密集部署和交叉读冗余数据清洗和标签归属仲裁问题, 本实验不做讨论.

(a) RFID 阅读器

(b) 有源电子标签

图 7 本实验所用 RFID 阅读器和有源电子标签

4.2 实验评估

根据测试要求, 阅读器内置天线覆盖区域为扇形, 抽象出典型的阅读器与标签工作场景. 按照场景要求, 标签均匀放置在阅读器探测区域范围内, 如图 8 所示. 设定测试时间为 10 分钟, 统计出标签数据搜索深度 PL 值, 结果如表 1 所示. 在真实环境中, 使用 20、40 和 60 三段数值进行测试, 得到三个不同的搜索深度 PL 值. 通过计算机模拟仿真, 得到仿真环境下搜索深度 PL 值.

表 1 RFID 标签测试评估表

参数	真实环境			仿真环境	
标签数	20	40	60	80	100
深度 PL	196	411	618	831	1221

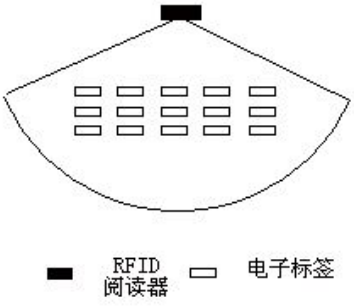

图 8　阅读器与标签工作场景抽象示意图

4.3 验证结果

通过分析表 1 中真实环境和计算机仿真环境得到的 RFID 验证数据，构建如图 9 所示验证结果图. 横轴为 Tags 标签数，纵轴为 PL 路径长度. 可以看到，改进后的二叉树型折半搜索算法路径长度要比原算法小. 搜索路径长度变小会直接提高搜索的效率. 结果验证表明，二进制树型折半搜索算法比原算法效率高.

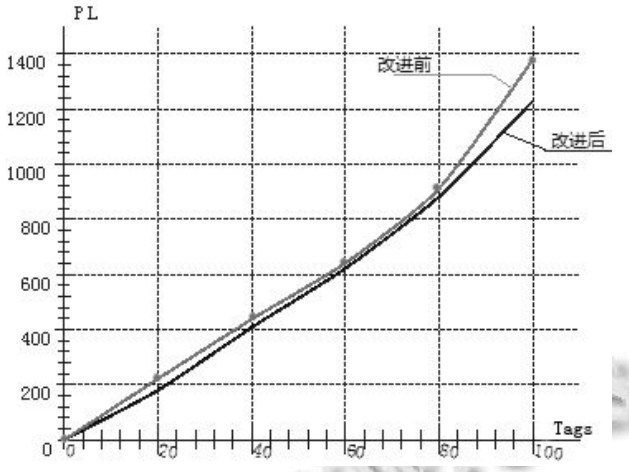

图 9　二叉树型折半搜索算法和原算法标签碰撞路径长度比较

5　结论

在 RFID 系统中，提高阅读器对标签的识别效率、降低识别时间和尽量避免识别错误是提高系统可靠性的基本要求. 本文分析了被动式标签模式下二叉树型搜索防碰撞算法和各种改进的防碰撞算法，提出一种二进制树型折半搜索防碰撞算法. 数据分析和验证结果表明，二进制树型折半搜索算法比原二进制树型搜索算法路径长度小，可以显著提高识别效率.

参考文献

1 Yang CN, He JY.An effective 16-bit random number aided query tree algorithm for RFID tag anti-collision. Communications Letters, IEEE, 2011, 5(15): 539–541.

2 周信,刘晔.一种基于码距反演的 RFID 防碰撞算法.计算机工程与应用,2012,48(8):214–217.

3 吴海峰,曾玉,丰继华.标签数估计的被动 RFID 标签防冲突二进制树时隙协议.计算机研究与发展,2012,49(9):1959–1971.

4 王荃,滑楠,张璐.基于仲裁集的 RFID 主动式标签防碰撞 MCMA 协议.科学技术与工程,2013,13(4):1037–1044.

5 米根锁,王彦快,马学霞.隧道人员定位系统中 RFID 防碰撞算法的研究.计算机工程与应用,2012,48(24):72–76.

6 陆冰清,牛国柱,赵英臣.一种新型 RFID 动态多叉树查询防碰撞算法.制造业自动化,2012,34(8):12–15.

7 熊昌慧,丁永生,郝矿荣.一种基于 PSO 优化的 RFID 防碰撞算法.计算机应用与软件,2012,29(6):8–10.

8 张文欣,昂志敏,尹夕振.一种改进的后退式二进制搜索 RFID 多标签防碰撞算法.合肥工业大学学报,2012,35(7):919–921.

9 李飞,曹敦,傅明.一种 BIBD 编码的 RFID 防碰撞算法的改进.计算机应用与软件,2012,29(6):151–154.

10 单承认赣,单玉峰,姚磊.射频识别(RFID)原理与应用.北京:电子工业出版社.2008:109–111.