doi:10.3772/j.issn.1002-0470.2017.06.004

判断二进制树中标签识别完毕的方法①

崔英花②

(北京信息科技大学信息与通信工程学院 北京 100101)

摘 要 分析了射频识别(RFID)系统阅读器与标签通信的二进制树算法,指出在标签识别过程中,阅读器并不知道是否识别完标签,阅读器会以连续多次没有接收标签响应为依据结束对标签的查询,这样往往会造成标签漏读或浪费时间在已识别完的标签上。基于此分析,提出了判断二进制树中标签识别完毕的方法。该方法通过在阅读器中设置计数器,就可以很好地跟踪标签的识别情况,准确地判断出标签是否识别完毕。分析结果表明,该方法可以准确地判断标签是否识别完毕,增加系统识别效率和可靠性。

关键词 射频识别(RFID),标签识别完毕,二进制树,树的深度

0 引 言

射频识别(radio frequency identification, RFID)技术是一种自动识别技术。RFID 系统中的阅读器可以在短时间内识别大量的标签。由于 RFID 可以快速识别物体,并且成本低廉、管理智能方便,因而被广泛应用于仓储、货运、物流等行业。

当前的 RFID 标准中阅读器与标签的通信仲裁过程可分为基于 Aloha 的算法[1-5]和基于二进制树的算法[6-9]。无论何种方法,协议都没有对何时结束对标签的查询做出明确的解释。在实际应用中,一般做法是阅读器发出查询命令,若标签在约定次数下没有应答,便视为标签已全部识别完毕。这种情况下很容易出现漏读标签,或在标签已识别完毕的情况下阅读器还在查询,从而增加系统功耗,降低系统识别效率。本文针对二进制树算法,在阅读器中设置计数器,通过计数器的变化来跟踪识别范围内标签的动态情况,判断标签是否识别完毕。通过该方法,阅读器可以明确知道是否识别完标签,避免不必要的查询、时间浪费和功耗,而且也可以防止漏读标签情况发生,增强系统识别的可靠性。

1 二进制树算法分析

二进制树算法是概率型算法,标签通过产生随机数 0、1 来决定自己是否与阅读器进行通信,其树形结构是由于碰撞标签不断分裂为左右两支而形成的。具体做法如下:

阅读器发出选择命令,所有标签的随机数产生器产生随机数 0 或 1,产生 0 的标签发送数据给阅读器(以后是计数器为 0 的标签向阅读器回复数据)。产生 1 的标签计数器置 1 进入休眠状态。

阅读器接收数据,会有以下三种情况发生:(a)只有一个标签回复,该标签被成功识别,所有标签计数器的计数减 1;(b)没有标签回复,所有标签计数器计数减 1;(c)有多个标签回复,产生碰撞。这些碰撞标签的随机数产生器产生随机数 0 或 1,产生 0 的标签继续发送数据给阅读器。产生 1 的标签的计数器置 1,并进入休眠状态等待下一轮查询,其他计数器计数加 1。如此反复,直至所有标签被识别。ISO18000-6B[6]标准采用的就是二进制树算法,其算

① 国家自然科学基金(61340005),北京市自然科学基金(4132012),北京市教委科技发展计划(KM201411232011)和北京市优秀人才培养(2013D5007000006)资助项目。

② 女,1973 年生,博士,副教授;研究方向:无线通信,射频识别,移动通信;联系人,E-mail: cui_ying_hua@sina.com

(收稿日期:2017-03-01)

法流程如图 1 所示。

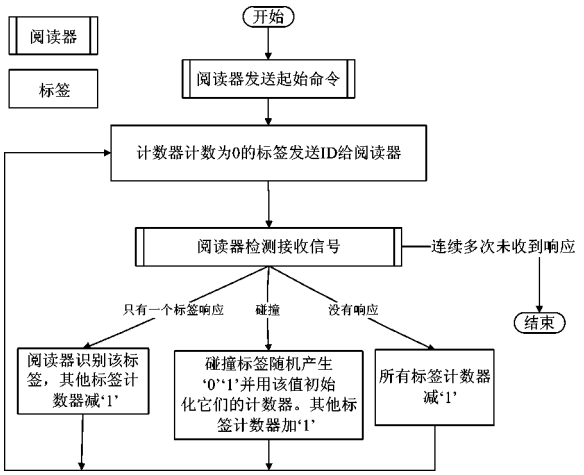

图 1　二进制树算法流程图

采用二进制树算法,经过初始几次连续碰撞后,形成如图 2 所示的树结构。

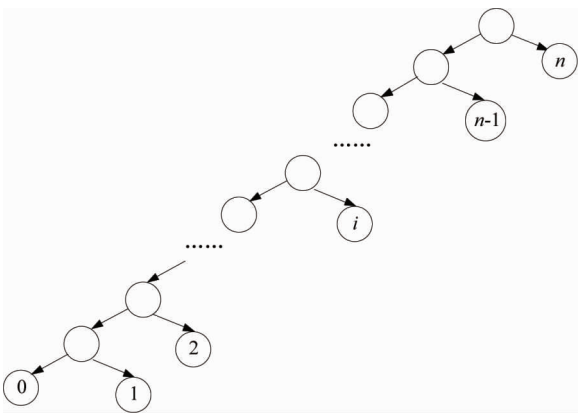

图 2　二进制树算法几次连续碰撞所形成的树结构示意图

图 2 中空白圆圈表示碰撞节点,而标有数字的圆圈表示计数器值为该数值的标签的集合,所有具有相同计数的标签处于同一个节点。假设图 2 中,计数器值为 0 的集合为空或者只有一个标签,即该集合对应于第一个没有标签回复或只有一个标签回复的节点。在识别完该节点后,此时记二进制树的深度为 n,如图 3 所示。计数器值最大的标签计数器取值为 $n-1$。

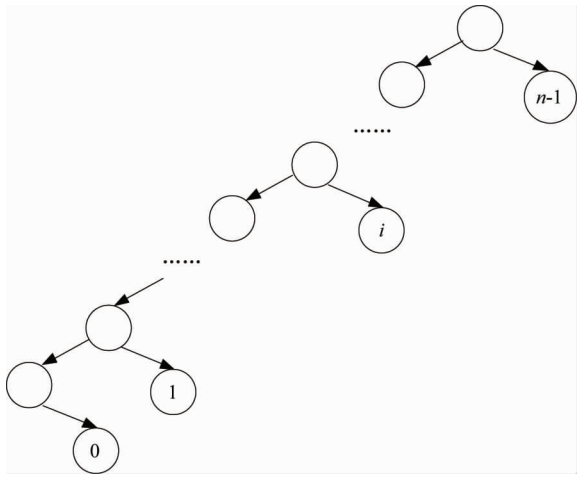

图 3　二进制树算法识别完第一个节点后所形成的树结构

在随后的标签识别中,当把图 3 中最底层的节点识别完后,则二进制树如图 4 所示。此时二进制树的深度为 $n-1$,计数值最大的标签计数器取值为 $n-2$。依此类推,假设当前树的深度为 m,则当标签的最大计数值变为 $m-2$ 时,表示已将树中最底层的节点识别完毕,树的深度也将相应减 1。此时树的深度变为 $m-1$,标签的最大计数值为 $m-2$。这个过程反复进行,直到所有标签被识别完毕。

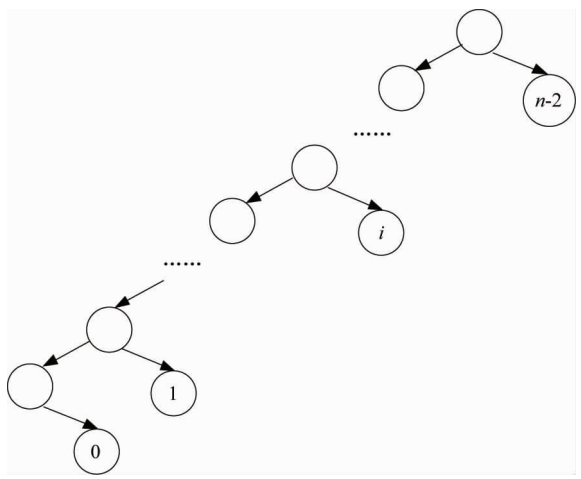

图 4　二进制树算法识别完最底层节点后所形成的树结构

2　判断标签识别完毕的方法

由图 3 可知,当识别完最底层的节点后,树的深度减少,当树的深度减少为 0 时,则表示所有标签都已识别完毕。根据该特点在阅读器中设置一个计数器 S,其计数记为 S,设置其初始值为 1。标签的变

化同二进制树算法。标签计数器为 0 的标签均回复阅读器,阅读器计数器数值随标签回复的情况作变化,模拟标签计数器的变化情况。当 S 变为 0 时,表示当前所有标签识别完毕,结束查询。

阅读器根据检测到的回复标签,在阅读器端有如下三种规则:

(a)没有标签回复,S 值减 1,且所有的标签计数器数值都减 1。

(b)只有一个标签回复,S 值减 1,且所有的标签计数器数值都减 1。

(c)有两个以上标签回复,S 值加 1,且标签计数器为 0 的标签随机产生 0 或 1,并用该值初始化它的标签,计数器不为 0 的标签将自己的计数器数值加 1。

直到阅读器的计数器 S 值为 0 时,完成标签识别。

3　判断标签识别完毕的方法实例分析

假设有 5 个标签,标签仲裁协议为二进制树算法。阅读器的计数器 S 值的变化按照上述定义的规则。即当阅读器检测到标签碰撞时,S 值加 1;当识别一个标签时,S 值减 1;当没有标签回应时,S 值减 1。图 5 说明阅读器的计数器 S 值在整个识别过程中的变化过程。其中:方框内数字依次表示查询次数和计数器 S 的数值,例如 0:1 是第 0 次查询(即初始状态),阅读器计数器 S 的初始值为 1。圆圈内数字表示要识别标签数。从图中可以看出,通过在阅读器中设置计数器的方法可以判断标签是否识别完毕。

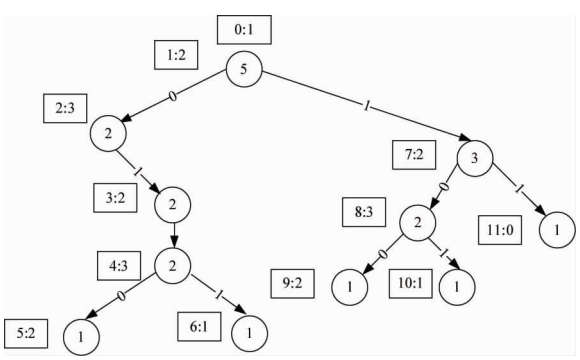

图 5　阅读器计数器 S 的变化实例图

图 5 中共有 11 个步骤。二进制树的左支标签识别为步骤 1 到步骤 6,右支标签识别为步骤 7 到步骤 11,其判断标签识别完毕的具体步骤如下:

步骤 0:设置 S 初始值为 1,开始查询标签时,阅读器向标签发出查询命令。

步骤 1:设定 5 个标签计数器的初始值都为 0,向阅读器发送数据,阅读器检测到碰撞,发送识别失败 FAIL 命令。碰撞标签随机产生 0 或 1,并用该值初始化标签。实施例中碰撞后产生两个随机数为‘0’的标签,3 个随机数为‘1’的标签。阅读器计数器 S 适用上述规则(c),S 加 1 变成 2。

步骤 2:由于有两个计数器为 0 的标签向阅读器发送数据,阅读器检测到碰撞,发送 FAIL 命令。碰撞标签随机产生 0 或 1,并用该值初始化标签,实施例中,碰撞后产生两个随机数都为‘1’的标签,即标签计数器的数值为 1。阅读器计数器 S 适用上述规则(c),S 加 1 变成 3。

步骤 3:由于碰撞的两个标签产生的随机数为‘1’,不能向阅读器发送数据,所以阅读器没有检测到标签响应。阅读器计数器 S 适用上述规则(a),S 减 1 变成 2。同时阅读器发送识别成功命令 SUCCESS 命令,标签计数器计数减 1,即这两个碰撞标签计数器计数值为 0。

步骤 4:上述两个计数器为 0 的标签向阅读器发送数据,阅读器检测到碰撞,发送 FAIL 命令。碰撞标签随机产生 0 或 1,并用该值初始化标签,实施例中,碰撞后产生一个随机数为‘0’的标签,一个随机数为‘1’的标签,即一个标签的计数器值为 0,一个标签的计数器值为 1。阅读器计数器 S 适用上述规则(c),S 加 1 变成 3。

步骤 5:此时只有一个计数器为 0 的标签向阅读器发送数据,阅读器识别到该标签,发送 SUCCESS 命令,其他标签计数器计数减 1,产生计数器数值为 0 的标签。阅读器计数器 S 适用上述规则(b),S 减 1 变成 2。

步骤 6:此时只有一个计数器为 0 的标签向阅读器发送数据,阅读器识别到该标签,发送 SUCCESS 命令,所有标签计数器计数减 1,产生计数器数值为 0 的标签。阅读器计数器 S 适用上述规则

(b),S 减 1 变成 1。

步骤 7:由于有三个计数器为 0 的标签向阅读器发送数据,阅读器检测到碰撞,发送 FAIL 命令。碰撞标签随机产生 0 或 1,并用该值初始化标签,实施例中,碰撞后有两个标签产生随机数‘0’,一个标签产生随机数‘1’,即两个标签计数器为 0,一个标签计数器为 1。阅读器计数器 S 适用上述规则(c),S 加 1 变成 2。

步骤 8:由于有两个计数器为 0 的标签向阅读器发送数据,阅读器检测到碰撞,发送 FAIL 命令。碰撞标签随机产生 0 或 1,并用该值初始化标签。实施例中,碰撞后,有一个标签产生随机数‘0’,另一个标签产生随机数‘1’,即一个标签计数器为 0,一个标签计数器为 1。阅读器计数器 S 适用上述规则(c),S 加 1 变成 3。

步骤 9:此时只有一个计数器为 0 的标签向阅读器发送数据,阅读器识别到该标签。阅读器发送 SUCCESS 命令,其他标签计数器计数减 1,产生计数器为 0 的标签。阅读器计数器 S 适用上述规则(b),S 减 1 变成 2。

步骤 10:此时只有一个计数器为 0 的标签向阅读器发送数据,阅读器识别到该标签。阅读器发送 SUCCESS 命令,所有标签计数器计数减 1,产生计数器为 0 的标签。阅读器计数器 S 适用上述规则(b),S 减 1 变成 1。

步骤 11:此时只有一个计数器为 0 的标签向阅读器发送数据,阅读器识别到该标签。阅读器发送 SUCCESS 命令,所有标签计数器计数减 1,产生计数器为 0 的标签。阅读器计数器 S 适用上述规则(b),S 减 1 变成 0。

此时,S 值为 0,完成标签识别,所有标签被识别完毕。

从识别过程来看,阅读器中设置的计数器在步骤 0 ~ 步骤 11 中,会随着标签回复的情况做变动。当 $S=0$ 时,阅读器识别完全部标签。因此根据阅读器中计数器的状态,就可以判断标签是否识别完毕。同时,该方法不需要改变阅读器和标签任何的空中接口协议,可以与现存的标准很好地兼容,是一个很好的判读是否识别完标签的方案。

4 多分支结构中判断标签识别完毕的方法

为了提高系统识别效率,二进制树算法经常会被改进为多分支树算法。上节讨论的判断标签是否识别完毕的方法还可以很容易地推广到多进制树方法中,阅读器的计数器 S 值的变化按照上述定义的规则。

阅读器根据检测到的回复标签,在阅读器端有如下三种规则:

(a)没有标签回复,S 值减 1,且所有的标签计数器数值都减 1。

(b)只有一个标签回复,S 值减 1,且所有的标签计数器数值都减 1。

(c)有两个以上标签回复,S 值加 $N-1$,N 为分支数。且标签计数器为 0 的标签随机产生 0 或 1,并用该值初始化它的标签,计数器不为 0 的标签将自己的计数器数值加 1。

直到阅读器的计数器 S 值为 0 时,完成标签识别。图 6 为判断 10 个标签识别完毕过程示例。

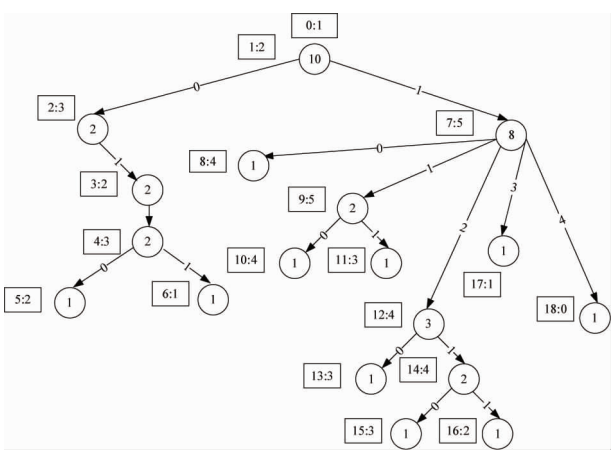

图 6　判断 10 个标签识别完毕过程

图 6 方框内数字依次为查询次数:计数器值,例如 0:1 是第 0 次查询(即初始状态),阅读器计数器 S 的初始值为 1;圆圈内数字为要识别的标签数。从图中可以看出,该方法同样适用于多分支树算法。

5 结 论

通过上述分析,可以看出,在阅读器中设置计数器可以很好地跟踪标签的识别情况,并对是否识别完标签做及时判断。避免了为标签是否识别完毕而做的查询过程,节省时间,减少了功耗,又很好地避免出现标签漏读的现象,增加了系统可靠性。同时该方法还适用于多分支的多叉树上。该方法只需在阅读器中设置计数器,并不更改任何空中接口协议,标签端不需要做任何变动,因此可以与现有协议很好兼容。

参考文献

[1] Finkenzeller K. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, 5th ed, New York :Wiley, 2010

[2] Arshad S, Nazir M H, Anwar S M, et al. Optimal enhanced dynamic framed slotted Aloha OEDFSA. In: Proceedings of the 6th International Conference on Innovative Computing Technology, Dublin, Ireland, 2016. 567-57

[3] Cui Y H, Zhao Y P. Performance evaluation of a multi-branch tree algorithm in RFID. *IEEE Transactions on Communications*, 2010, 58(5): 1356-1364

[4] Liu F, Wang L, Cai Z, et al. Research on RFID tag anti-collision algorithm based on aloha. *C e Ca*, 2017, 42(3): 1069-1072

[5] Elshabrawy T, Shereen E, Ashour M. Throughput evaluation of dynamic frame slotted ALOHA for spatially distributed RFID tags. In: Proceedings of the IEEE 84th Vehicular Technology Conference, Montreal, Canada, 2017. 166-172

[6] International Organization for Standardization. Information Technology Radio Frequency Identification for Item Management Part 6: Parameters for Air Interface Communications at 860 MHz to 960 MHz. ISO/IEC 18000-6, 2013

[7] International Organization for Standardization. Information Technology——Radio Frequency Identification for Item Management——Part 4: Parameters for Air Interface Communications at 2.45GHz. ISO/IEC FDIS 18000-4, 2015

[8] Cui Y H. A collision recovery algorithm using optimal Q parameter based on BIBD in RFID, *High Technology Letters*, 2016, 22(4): 350-354

[9] Zheng X Y, Cho C, X Y. Algorithms and stability analysis for content distribution over multiple multicast trees. *IEEE Transactions on Parallel and Distributed Systems*, 2015, 26(5): 1217-1227

A scheme of judging the end of tag identification in binary tree

Cui Yinghua

(School of Information and Communication Engineering, Beijing Information Science & Technology University, Beijing 100101)

Abstract

The binary tree algorithm for the communication between tags and a radio frequency identification (RFID) system's reader was analyzed, and it was pointed that in the RFID tag recognition process, the reader does not know whether all the tags are identified, and the reader will end up querying if it does not receive the tag response many times, thus often resulting in tag read off or wasted time on the tags identified. Based on the analysis, a scheme for judging the end of tag identification in the binary tree was proposed. The scheme can well track the identification of the tags by setting a counter in the reader, and accurately judge if all the tags are identified. The analysis results show that this method can accurately judge whether all the tags are identified or not, with the higher identification efficiency and reliability.

Key words: radio frequency identification (RFID), end of tag identification, binary tree, depth of the tree