文章编号:1001 - 9081(2008)08 - 2141 - 03

RFID 系统中一种改进的防冲撞算法

张 颇,崔 喆
(中国科学院 成都计算机应用研究所,成都 610041)
(pozhang822@163.com)

摘 要:提出了一个在 RFID 系统中改进的防冲突算法,该算法基于 Aloha 算法并结合了二叉树算法。当有大量标签同时需要识别时,首先通过对标签上一轮的碰撞情况来估计待识别的标签数,然后对标签进行分类或改变帧的大小来降低标签发生碰撞的概率,从而提高识别的效率。

关键词:射频识别;防冲突算法;Aloha 算法
中图分类号: TP311.11 **文献标志码**:A

Improved anti-collision algorithm in RFID system

ZHANG Po, CUI Zhe
(*Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu Sichuan* 610041, *China*)

Abstract: Based on Aloha and binary tree algorithm, this paper proposed an improved anti-collision algorithm in RFID system . When lots of tags need to be identified at the same time, it first estimated the number of unread tags according to the last collision, then divided the tags into different subsets or adjusted the frame size to reduce the collision probability. As a result, the identification efficiency is improved with the proposed method.

Key words: Radio Frequency Identification (RFID); anti-collision algorithm; Aloha algorithm

0 引言

射频识别(RFID)是一种非接触式的自动识别技术,它通过射频信号自动识别目标对象并获取相关数据,识别工作无须人工干预,可工作于各种恶劣环境。RFID 技术可识别高速运动物体并可同时识别多个标签,操作快捷方便。在实际应用中,通常将电子标签附着在待识别物体的表面,通过阅读器读取并识别电子标签中所保存的电子数据,从而达到自动识别个体的目的。单独的 RFID 系统没有什么价值,RFID 真正的用处在于它与数据库,网络等的联合。通过 RFID 系统人们可以把现实生活的物体转变到网络中,以便对它们进行监视、跟踪和管理。目前 RFID 技术已应用于物流、身份识别、科学研究和畜牧业等方面。

RFID 系统主要由电子标签和读写器两个部分组成。电子标签中一般保存有约定格式的电子数据,通过这些数据能唯一区别出该电子标签。标签内部一般都含有芯片和天线,芯片一般用来处理简单的逻辑运算和数据存储,而天线用于同外界的交流。通过标签是否带有能量,可以把标签分成主动标签和被动标签两大类。主动标签自带能量,能够自由地发射信号和接受信号,其反应速度和可靠性都较高,但是其成本较高且体积较大。与之相反,被动标签没有内置电源,它通过读写器的电磁场来获取自身运行所需的能量。由于被动标签结构简单,成本便宜,适合大量使用,因而目前使用 RFID 识别的系统大部分都采用被动标签。阅读器的主要任务是控制射频模块向标签发射读取信号,并接收电子标签的应答,对标签的对象标识信息进行解码,将对象标识信息连带标签上其他相关信息传输到主机以供处理。

射频识别的工作原理为读写器周期的广播自己的信号,进入读写器感应范围的标签将获得运行的能量而被激活,在接受到信号后,标签将按照一定的协议规定发出自己的回执信号,由于所有标签的回执信号使用的是同一个信道,如果在同一时刻有多个标签发出信号,将在该信道上产生信号冲突从而造成读写器无法正确地识别标签返回的信号。为了解决上述问题而产生了许多防冲突算法,由于被动标签的结构简单,标签之间不能实现通信而只能进行简单的信号反射,所以避免信号冲突的任务主要落在了读写器上。目前主要有不确定性算法和确定性算法两种类型的防冲突算法。不确定性算法是基于 Aloha 机制的算法,确定性算法是基于二叉树机制的算法。基于 Aloha 的防冲突算法的主要原理是通过标签随机地选择发送回执信号的时间来避免冲突,而基于二叉树算法则是将电子标签进行分类来避免冲突。本文介绍的算法是结合了 Aloha 和二叉树这两种思想的一个改进算法,通过将标签进行分类且采用了随机挑选回执时间来避免标签之间的信号冲突,从而提高识别效率。

1 防冲撞算法介绍

1.1 Aloha 算法

在 Aloha 算法中当标签进入读写器范围时,电子标签自动地向读写器广播自己的 ID(即唯一标识自身的数据,一般情况下为定长),在发送数据时如果有其他的标签也在发送数据,那么将会发生信号冲突,读写器将不能正确地识别标签的 ID 号。读写器在检查到信号冲突时,将发送一个停止发送信号的命令让所有标签停止当前发送并随机等待一个时间后再发送自己的信息。纯 Aloha 算法较简单、易实现,但标签之间发生信号冲突的概率很大,系统的识别率较低[1]。

1.2 Frame-Sloted Aloha

Frame-Sloted Aloha 算法是从 Aloha 算法发展而来的,在

收稿日期:2008 - 02 - 21;**修回日期**:2008 - 03 - 26。

作者简介:张颇(1982 -),男,四川广元人,硕士研究生,主要研究方向:RFID 系统; 崔喆(1970 -),男,四川巴中人,研究员,主要研究方向:基于局域网、Internet 的现代会议系统。

该算法中,时间被分成多个离散的时隙,标签只能在每个时隙的开始处发送信号,N 个时隙组成一个帧,读写器以一个帧为周期发送询问信号,当标签接收到来自读写器的询问请求时,每个标签有一个随机数发生器,随机地挑选一个时隙。并在该时隙发送自己的信号。如果一个时隙只被一个标签选中,则在这个时隙传输的信号将被读写器成功接收,标签被识别。如果有两个以上的标签选择了同一个时隙发送信号则碰撞产生,读写器不能识别发送信号的标签。如此循环,至到所有的标签都被识别。帧的大小是固定的,所以如果在某一时刻标签的个数远大于帧中时隙的个数,则在一个帧中发生碰撞的几率将被提高,被浪费的时隙也将增多,从而延长了识别所有标签的时间[2-3]。

1.3 Dynamic Framed Slot Allocation (DFSA)算法

该算法是对 Frame-Slot Aloha 算法的进一步拓展,在算法中通过对上一轮的碰撞情况进行统计,来估算周围有多少个标签,从而改变本次帧的大小,有两种估算方法。

估算方法1:

$$C_{\text{rate}} = 1 - P_{\text{succ}} - P_{\text{idle}} = 1 - (1 - \frac{1}{L})^n(1 + \frac{n}{L-1}) \tag{1}$$

当一个帧完成后,C_{rate} 表示发生碰撞的时隙和帧的大小的比率,可以根据统计出来,L 是帧的大小,从而可以估算出标签 n 的值。

估算方法2:

$$N_{\text{tags}} = 2.392\,2N_{\text{collslot}} \tag{2}$$

其中,N_{tags} 代表估算标签的个数,N_{collslot} 代表在一个帧中发生碰撞的时隙个数。该算法可以很好地提高识别标签的效率,但是该方法在有大量标签时,帧的长度也变得很大,这样对于系统将产生较大的负担[4]。

1.4 基本二叉树算法

在该算法中要求标签中有一个指针,最初该指针指向标签 ID 比特位的最高位,随着查询的进行该指针将逐步向最低位移动,每次读写器发送一个查询比特0或1,如果标签中指针所指的该位和查询比特相同,则标签发送自己的下一位给读写器,否则标签进入休眠状态不参与下一次查询。当读写器没有检查到冲突时,将把接受到的比特位作为下一次的查询条件,否则将0作为下次的查询值。直到标签的指针指向最低位,这样每轮循环能识别出一个标签[1]。

1.5 二进制搜索算法

该算法是一种无记忆的算法,标签不必记录查询状态,读写器每次发送的不是一位的查询比特而是一个查询比特前缀a1a2a3a4a5,读写器逐渐地增加查询前缀直到获得唯一的一个标签标识。当标签收到一个由读写器发出的查询前缀且自己的标识符合查询条件时,标签就将自己的标识发送给读写器。如果没有冲突则识别到一个标签,当有两个或者更多的标签回应读写器则将发生冲突,并在下次查询中在原来的查询前缀后面增加0或1,如此循环直到识别完所有的标签[5]。

2 改进的算法

动态帧时隙分配算法(Dynamic Framed Slot Allocation, DFSA)算法能够根据标签的多少来动态地改变帧的大小,这提高了时隙的利用率,但是当周围有大量的标签时,帧的大小也变得很大。这对于标签识别将会造成局部的延迟,并消耗更多的存储空间。因此当有大量的标签同时出现时,需要将标签进行分类,缩小识别的范围。

本文提出的算法的主要思想是在有大量标签出现的时候把标签分成多个数量较少的子集以缩小识别范围,然后再用上文提到的 Frame-Sloted Aloha 方法对这些子集进行识别。首先通过上一帧中碰撞情况来估算当前范围内有多少标签有待识别,然后决定是否进行分类,估算方法利用了前文提到的计算方法2。用一个字符串 Qstr 表示当前标签集合其 ID 的前 n 位值,只有标签 ID 的前 n 是 Qstr 才属于该集合,Qstr 被初始化为空以表示所有的标签。通过改变 Qstr 的值来实现对标签的分类和遍历,每当需要对该集合进行分类的时候就在 Qstr 的尾部增加一个0,从而将当前集合的标签分成两个子集,一个是标签 ID 前 n 位是 Qstr 且第 $n+1$ 位是0的集合,一个是标签 ID 前 n 位是 Qstr 且第 $n+1$ 位是1的集合,通过不断地执行这个分类过程直到标签集合的大小达到一个满意的值。如果当前集合不需要分类,则根据估算标签的多少来改变查询帧的大小,然后让标签 ID 的前缀是 Qstr 的标签在帧中随机挑选一个时隙发送自己的信号,这样如果在该时隙中没有发生信号冲突,标签将被正确识别。当识别完一个集合后,表示该集合的 Qstr 的最后一位将被变成1,这样当所有的标签都被识别后 Qstr 应该是一个全1的字符串或者是空串。具体方法如下:

根据帧中发生碰撞的时隙数 N_{collslot},通过式(2)估算出标签数 N_{tags},如果 $N_{\text{tags}} < FrameSize$(帧的大小)则说明当前帧的时隙数大于目前发生碰撞的标签数,不需要分支;如果 $N_{\text{tags}} \neq 0$,则 $FrameSize = N_{\text{tags}}$,否则 $FrameSize$ = 初始化的值,并且 $N_{\text{tags}} = 0$ 说明标签 ID 前缀为 Qstr 的标签已全部识别了,下一步将查询另一个子集。通过修改查询字符串 Qstr 来实现对另一个子集的查询,如果 Qstr 全为字符1或者为空则说明识别了所有的标签,程序将退出,否则从 Qstr 的最后一位开始查找,直到找到一个为字符0的位置,抛弃该位以后的字符串,并将该位置成字符1,然后开始识别标签 ID 的前 n 位是 Qstr 的标签集合。例如:如果当前 Qstr 为001110111,我们找到了倒数第四位为0,将该位变成1,并抛弃最后三位,则 Qstr 将变成001111。

如果 $N_{\text{tags}} > FrameSize$,则说明需要分支,通过在查询字符串 Qstr 的末尾来增加一位来实现分类,即在查询字符串 Qstr 的后面添加一个0字符,这样就将当前集合的标签集分成了两部份。这时需要增长 $FrameSize$ 的值,如果 $N_{\text{tags}} > MaxFrameSize$($MaxFrameSize$ 表示系统能允许的最大帧长),则 $FrameSize = MaxSize$,否则 $FrameSize = N_{\text{tags}}$,然后执行下一次查询。举例来说,如果当前的查询字符串 Qstr 为'010101',如果需要分支,则我们修改查询字符串为'0101010',那么下次查询的条件将变成标识号前7位是'0101010'的标签,当所有符合'0101010'查询字符串的标签都被识别后,下次查询的字符串将变成'0101011',如此循环直到查询字符串中所有的字符全都变成1为止。算法流程图如图1所示。

3 模拟分析

在模拟分析中将标签数以100为步长从0逐步增加到1 000,用 Matlab 进行了算法的模拟,计算出每次所消耗掉的时隙数,其初始化的帧的大小是256,图2分别显示了 Frame Slot Aloha 算法、DFSA 算法、一种分组帧时隙 Aloha 算法(GFSA)和改进算法识别完所有的标签所消耗的时隙数。在

这四种算法中,FSA 算法不能动态地改变帧的大小,帧初始值对系统性能影响较大,当标签数目增大时,系统识别标签所需的时隙数迅速增加。DFSA 算法能够估算出外围待识别的标签数从而改变自身帧的大小,在标签数小于 600 时,系统识别效率很好,但是随着标签数目进一步的增大,系统识别标签所消耗的时隙数近似地成指数增长。一种分组帧时隙算法(GFSA)不能动态地改变帧的大小,但当标签冲突达到一定比例时,算法可以将标签按照一定的方式分组从而降低冲突发生的概率[7],在标签数较少时其性能和 FSA 算法相同,在标签数目较大时由于其分组特性使系统的识别效率有所提高。从图 2 中可以看出,改进的算法所消耗的时隙数基本上与标签数目成线性增长,帧的初始值对于识别没有太大影响,在标签数目较大时,对于系统性能的提升较为明显。图 3 为 DSFA 算法和改进后的算法的系统效率比较,从图中可以看出改进后的算法的在标签数量超过 600 时,系统的利用率大大超过了 DSFA 算法,且系统效率一直保持在一个比较平稳和高效的状态。改进的算法在基于 DSFA 算法的基础上采用类二叉树算法的分组方法来提升系统的效率,当需要分组时,每次把当前标签集合分成两组,然后不断地迭代直至达到一个合理的大小,当子集识别完成后回溯到上一层以进行对其他集合标签进行识别。这种分组方法能应用到各种不同标签 ID 分布中。在算法中实现这种方法只需要一个能容纳标签 ID 号的数组即可,相对于其他的分组策略实现起来更为简单容易。

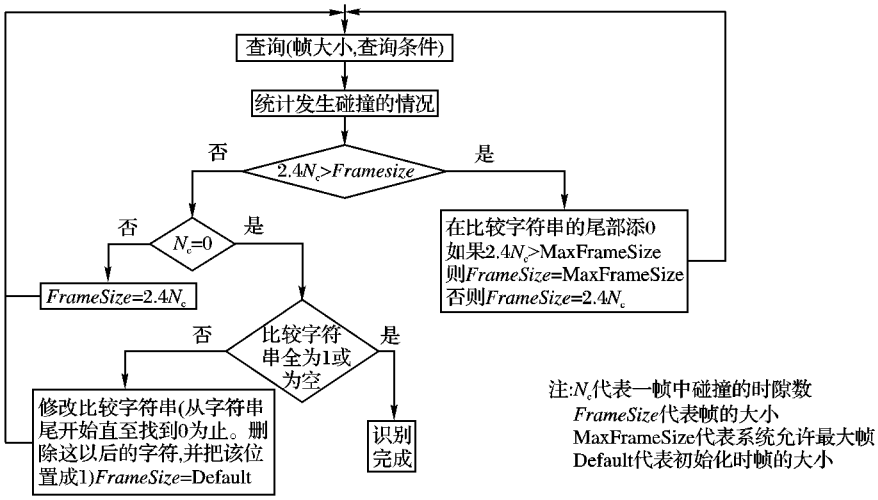

图 1　算法流程

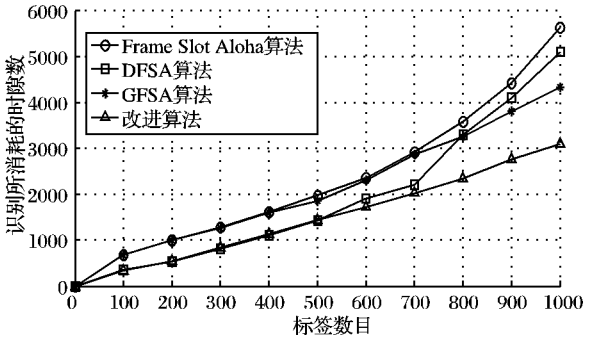

图 2　四种算法的消耗时隙比较

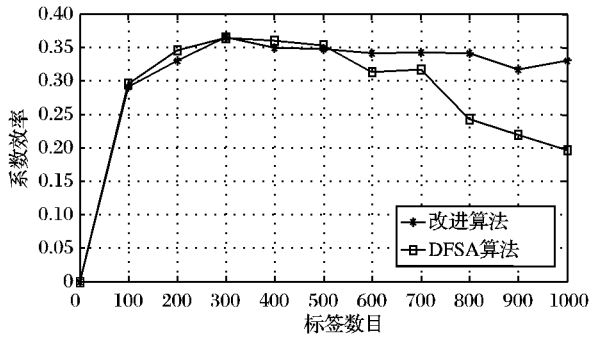

图 3　DFSA 算法和改进后算法系统效率比较

4　结语

RFID 系统需要在短时间内识别大量的标签,基于 Aloha 的算法由于其协议简单,易于实现而得到了广泛的应用,但是标签冲突会降低程序的执行效率,所以为了缩短识别时间,提高 RFID 系统的效率,主要采用改变帧的大小和对标签进行分类这两种方法。结合这两种方式本文提出了一种简单易行的方法。从前面的性能分析中可以看出,改进算法在不使帧的大小超过系统允许的最大值的同时,通过动态地调整帧的值和对待识标签分组来降低发生碰撞的概率,提高了系统的性能并减少了操作的复杂度和系统的开销。

参考文献:

[1] 陈香,张思东,薛小平. RFID 防碰撞技术的研究[J]. 电脑与电信,2006(4):63 - 66.

[2] ZHEN BIN, KOBAYASHI M, SHIMIZU M. Framed Aloha for multiple RFID objects identification [J]. IEICE Transactions on Communications, 2005, E88(1):991 - 999.

[3] VOGT H. Efficient object identification with passive RFID tags [C]// International Conference on Pervasive Computing, LNCS 2414. Berlin: Springer-Verlag, 2002:98 - 113.

[4] CHA J R, KIM J H. Novel anti-collision algorithms for fast object identification in RFID system [C]// Proceedings of the 11th International Conference on Parallel and Distributed Systems Workshops. Washington: IEEE Computer Society, 2005:63 - 67.

[5] LAW C, LEE K, SIU K Y. Efficient memoryless protocol for tag identification [C]// Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. New York: ACM, 2000:75 - 84.

[6] TAO CHENG, LI JIN. Analysis and simulation of RFID anti-collision algorithms[C]// The 9th International Conference on Advanced Communication Technology. New York: IEEE, 2007, 1:697 - 701.

[7] HWANG T W, LEE B G, KIM Y S, *et al*. Improved anti-collision scheme for high speed identification in RFID system[C]// Proceedings of the First International Conference on Innovative Computing, Information and Control. Washington: IEEE Computer Society, 2006, 135 - 139.