

智能包装技术

RFID 包装系统中防冲突算法研究

金迪

(哈尔滨商业大学, 哈尔滨 150028)

摘要: **目的** 解决现阶段包装箱管理过程中 RFID 标签冲突的问题。**方法** 在分析已有的解决标签冲突算法的基础上, 采用多线程技术、后退式二进制防冲突算法和优化的数据结构等方法, 设计并实现一种新的防冲突算法。RFID 读写器发送一个三元组命令, RFID 标签应答冲突位的信息, 减少数据传输量; 采用堆栈存储 RFID 读写器发送的命令, 减少识别冲突的次数; 利用多线程处理思想, 对标签进行分类处理, 缩短冲突处理的时间。**结果** 经过仿真分析, 这种并发执行的后退式二进制 RFID 防冲突算法效率可提高约 51%。**结论** 该算法解决了 RFID 标签冲突, 提高了多标签情况下的读写效率, 很好地解决了包装箱管理系统中 RFID 标签的功能和性能的问题。

关键词: RFID; 标签冲突; 防冲突算法; 多线程; 后退技术

中图分类号: TB487; TP301.6 **文献标识码:** A **文章编号:** 1001-3563(2018)01-0001-05

Anti-collision Algorithm in RFID Packaging System

JIN Di

(Harbin University of Commerce, Harbin 150028, China)

ABSTRACT: The work aims to solve the RFID tag conflict in the process of managing packaging box at current stage. Based on the analysis of the existing conflict resolution algorithm, a new anti-collision algorithm was designed and implemented with multi-thread technology, backward binary anti-collision algorithm and optimized data structure. The RFID reader sent a command of triples and responded the information at the conflict position. This process reduced the amount of transmission data; the command that RFID sent was stored by the stack to reduce the times of identifying conflicts; with the thought of multithreading, the labels were classified to shorten the processing time of conflicts. The simulation analysis showed that, the efficiency of the multi-threading backward binary RFID anti-collision algorithm could be increased by about 51%. The algorithm solves the problem of RFID tag conflict, improves the efficiency of reading and writing in the case of multiple labels, and well solves the problems regarding the function and performance of RFID tag in the packaging box management system.

KEY WORDS: RFID; label conflict; anti-collision algorithm; multithreading; backward technology

在现代物流运输中, 射频识别技术 (RFID) 被广泛应用于包装箱管理^[1]。RFID 技术的使用可以对包装箱的装箱、运输、入库和储存等各个物流环节进行跟踪, 实现包装箱的实时监控和管理^[2]。RFID 可以无接触采集数据, 是一种自动识别的通信技术^[3]。这种技术可以通过电磁波双向传输数据实现数据的通信。RFID 标签、RFID 读写器和中央信息处理器组成了 RFID 系统, 其中 RFID 标签可以粘贴在包装箱

上, 可存储包装箱的具体信息, RFID 标签具有唯一的编码, 该编码可以作为包装箱的唯一标识; RFID 读写器可以对包装箱上的 RFID 标签进行更改、写入和读取, 将读取的 RFID 标签存储信息传输到中央处理器; 中央处理器可以对读取的 RFID 标签信息进行分析处理, 实现对包装箱的管理^[4]。

RFID 包装箱技术存在多个 RFID 标签信息冲突的问题^[5]。RFID 读写器向有效范围的 RFID 标签发出

收稿日期: 2017-05-19

基金项目: 2017 年度黑龙江省哲学社会科学规划项目 (17JYD238); 黑龙江省哲学社科“十九大专题”项目 (17YYH54)

作者简介: 金迪 (1980—), 女, 哈尔滨商业大学讲师, 主要研究方向为信息管理与信息系统。

通信请求信号,所有包装箱上的 RFID 标签会同时返回标签信息至 RFID 读写器^[2]。RFID 标签的通信使用共享的通信信道,因此多个 RFID 标签信息在同一信道中产生冲突。如何在避免冲突的条件下快速、正确地读出 RFID 标签信息,并高效地利用通信信道,是 RFID 应用在包装系统中主要需要研究和解决的问题之一^[6]。

对于 RFID 系统中的信息冲突问题,一般采用时分多址 (TDMA) 的技术来解决。基于树的算法和基于 ALOHA 的算法是时分多址的两类主要算法^[7-10]。基于 ALOHA 算法的主要思想是发现冲突后,重新产生识别标签。当标签数量增加,冲突则呈线性增长,系统性能急剧下降。基于树的算法是一类确定性算法,例如查询树算法 (QTA),二进制搜索算法 (BAS) 等^[11-14]。QTA 算法需传输并检验标签前缀,信息处理速度较慢。BAS 算法的主要思想是将读入的信息生成 2 个不相交的子集,循环处理,直到子集中只有唯一可识别的标签。当标签数量增大,也会产生大量标签的冲突,使系统效率降低。后退式二进制算法以 BAS 算法为基础,采用栈和后退原则减少冗余,提高系统的工作效率。以上 2 种算法虽然可以解决多个包装箱 RFID 标签的传输冲突问题,但随着同时读取包装箱数量的增多,传输和处理的数据量也随之大量增加。为了解决冲突和效率的问题,这里将提出并发执行的后退式二进制 RFID 防冲突算法。

1 算法基础

1.1 曼彻斯特编码

多个同时读取的 RFID 标签信息在共享信道中产生冲突,首先应找到冲突产生的位置,曼彻斯特编码则是确定冲突位置的基础编码方式。曼彻斯特编码采用在一位传输周期中的电平跳变规则,正跳变为 0,负跳变为 1。例如有 2 个 RFID 标签 (8 bit),信息分别为 10010111 (标签 TagB) 和 10110011 (标签 TagA)。这 2 个标签在共享信道中同时被 RFID 读写器读出,则读出的信息为 10X10X11 (其中 X 为无电平)。由于 TagA 和 TagB 的第 2 位和第 5 位相反,上升和下降电平相互抵消,可以说明第 2 位和第 5 位产生冲突,见图 1。

1.2 后退式二进制 RFID 防冲突算法

后退式二进制 RFID 防冲突算法识别 RFID 标签的步骤如下所述。

1) RFID 读写器广播发送同步信号及请求命令,将此命令入栈,便于下次命令的确定。各标签响应请求,并判断自己的 ID 是否小于等于 S, S 为 RFID 读写器的参数,表示该读写器能够读取的 RFID 标签的最大长度。如果小于等于 S,则返回自身 ID。

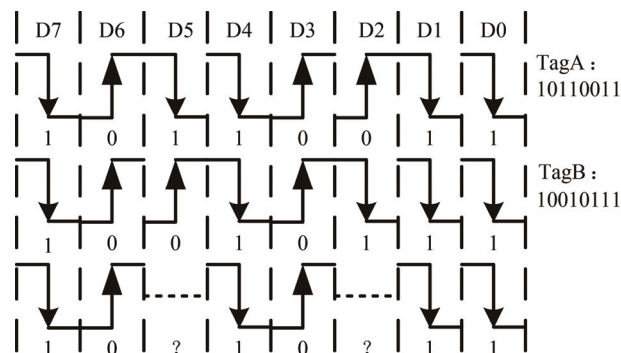


图 1 曼彻斯特编码确定的冲突位
Fig.1 The conflict bits identified in Manchester

2) 读写器根据读回的信息,判断是否有标签产生冲突,如果没有冲突,转到步骤 3); 如果有冲突,将读回的序列的最高冲突位置“0”,其余冲突位置“1”,不冲突位保持读回序列的原状态。将修改后的序列赋值给 S,返回步骤 1)。

3) 如没有冲突,证明已识别出一个 RFID 标签的 ID,可对此标签执行操作。从栈中取出请求命令,确定下次命令的参数,返回步骤 1)。

1.3 并发执行

并发执行主要采用多线程技术实现,多线程技术可以同步运行多个独立的程序片段,这种并行可以提高系统的效率。RFID 中央处理器可以采用多核处理器,采用多线程的编程方法,对 BAS 算法中生成的 2 个不相交的自己进行同步处理,既解决了冲突问题,又解决了效率问题。

1.4 定义命令

为方便对算法进行描述,在不改变原 RFID 系统命令的前提下,定义请求命令 Request (D, m, T) 和应答命令 Response (s)。其中 Request 命令中的 D 参数为冲突最高位的编号 (设 RFID 为 8 位 ID, D 为 3 位二进制数,如 110 表示冲突的最高位为 D6 位), m 为冲突位的参数 (0/1), T 为线程编号 (4 线程, 2 位二进制表示 00, 01, 10, 11 这 4 个线程的编号)。

2 算法流程

假设 RFID 标签编码为 16 位二进制数,现有 20 个待识别的标签,见表 1。改进算法采用 4 个线程处理该算法,树中结点表示根据最高冲突位发出的不同请求命令。

算法的流程如下所述。

1) RFID 读写器向可读标签发送同步信号,标签返回自身 ID。RFID 读回 1??1?0??的二进制序列,判断 D6, D5, D3, D1, D0 发生冲突。RFID 读写器将产生冲突的位置发送给 RFID 标签。

2) 根据最高冲突位和次高冲突位进行分类, RFID 标签根据自身 ID 的序列和产生冲突的位置, 判断应进入哪个分类, 由哪个线程进行处理。例如, 可用最高冲突位 D6 与次高冲突位 D5 对标签进行分类, 共分为 4 类, 每类对应进入相应的线程中进行下一步处理, 详细分类情况见表 2。下面以线程 1 的流程说明算法的整个流程。

表 1 标签的 ID
Tab.1 The label ID

标签名	标签 ID							
	D7	D6	D5	D4	D3	D2	D1	D0
Tag1	1	0	0	1	0	0	0	1
Tag2	1	0	1	1	0	0	1	0
Tag3	1	1	0	1	0	0	1	1
Tag4	1	0	0	1	1	0	0	1
Tag5	1	0	1	1	0	0	0	0
Tag6	1	0	0	1	1	0	0	0
Tag7	1	1	0	1	0	0	1	0
Tag8	1	0	0	1	1	0	1	0
Tag9	1	0	1	1	1	0	0	0
Tag10	1	1	1	1	0	0	1	1
Tag11	1	0	0	1	0	0	1	0
Tag12	1	1	0	1	1	0	0	1
Tag13	1	0	1	1	0	0	0	1
Tag14	1	1	1	1	0	0	1	0
Tag15	1	1	1	1	1	0	0	1
Tag16	1	1	1	1	0	0	0	1
Tag17	1	0	1	1	0	0	1	1
Tag18	1	1	1	1	1	0	1	1
Tag19	1	1	0	1	1	0	1	1
Tag20	1	1	1	1	1	0	0	0

表 2 标签分类
Tab.2 The classification of the label

D6D5 位的值	线程号	标签名
00	1	Tag1, Tag4, Tag6, Tag8, Tag11
01	2	Tag2, Tag5, Tag9, Tag13, Tag17
10	3	Tag3, Tag7, Tag12, Tag19
11	4	Tag10, Tag14, Tag15, Tag16, Tag18, Tag20

3) RFID 读写器发送 Request (011, 0, 00)。其中, 参数“011”表示标签的最高冲突位为 D3 位; 参数“0”表示将最高冲突位置“0”; 参数“00”表示上述标签所属线程的标号为“00”, 即线程 1。满足上述条件的 RFID 标签 (Tag1, Tag11) 返回信息, 发送命令 Response (D1D0), D1D0 为其余冲突位的标签信息,

即 Tag1 发送 Response (01), Tag11 发送 Response (10)。RFID 读写器就收到 Tag1 与 Tag11 返回的叠加后的信息“??”, 说明仍在 D1 与 D0 位存在冲突。将 Request (011, 0, 00) 命令入栈。

4) RFID 读写器发送 Request (001, 0, 00), 此时满足其条件的标签只有 Tag1, 该标签发送命令 Response (1)。RFID 读写器收到“1”的信息, 表示此时无冲突, 表示识别了一个标签 (Tag1)。Request (001, 0, 00) 命令入栈。

5) 执行出栈, 出栈的命令为 Request (001, 0, 00), 将第 2 个参数“0”改为“1”。RFID 读写器发送 Request (001, 1, 00), 此时满足其条件的标签只有 Tag11, 该标签发送命令 Response (0)。RFID 读写器收到“0”的信息, 表示此时无冲突, 并识别了一个标签 (Tag11)。

6) 执行出栈, 出栈的命令为 Request (011, 0, 00), 将第 2 个参数“0”改为“1”。RFID 读写器发送 Request (011, 1, 00), 此时满足其条件的标签有 Tag4, Tag6 和 Tag8, 标签分别发送命令 Response (01), Response (00) 和 Response (10)。RFID 读写器收到“??”的信息, 表示 D1 和 D0 存在冲突。

7) RFID 读写器发送 Request (010, 0, 00), 此时满足其条件的标签有 Tag4, Tag6, 标签分别发送命令 Response (1) 和 Response (0)。RFID 读写器收到“?”, 表示 D0 位存在冲突。此时只有一位冲突, 表示 D0 位可有 2 种取值, 识别了 2 个标签 (Tag4, Tag6)。Request (010, 0, 00) 命令入栈。

8) 执行出栈, 出栈的命令为 Request (010, 0, 00), 将第 2 个参数“0”改为“1”。RFID 读写器发送 Request (010, 1, 00), 此时满足其条件的标签有 Tag8, 标签发送命令 Response (0)。RFID 读写器收到“0”的信息, 表示此时无冲突, 识别了一个标签 (Tag8)。

9) 栈为空, 该线程结束。

线程 2、线程 3、线程 4 也采取同样的处理方式。

3 算法相关问题说明

1) Request 命令入命令栈。由于算法中对树的操作是从左至右, 所以按照命令中的 m 参数值确定是否命令入栈, m 为“0”入栈, 为“1”不入栈。

2) 出命令栈。由算法中的后退原则, 识别出标签后, 应后退查找树的结点。此时可执行出栈操作, 并将出栈命令中的 m 由“0”置“1”, 再执行下次搜索。

3) 多线程处理。如果 RFID 读写器是多频的, 并且 CPU 是多核处理器, 则该多线程为“并行处理”; 如 RFID 是单频的, 则需增加一个命令队列, 暂时存储同时发出的 Request 命令, 此时为“多线程处理”, 可利用 RFID 读写器在处理信息的同时, 读写 RFID

标签内容。这2种方式都可提高系统的效率。

4 性能分析

4.1 BAS、后退式BAS和并发式BAS算法比较

评估RFID系统防冲突算法的好坏,要看算法搜索完所有标签需要的查询次数和识别的时间。设RFID标签数量为 N ,RFID标签的编码长度为 L ,则BAS算法中RFID标签的编码长度为 L ;后退式BAS算法中,RFID标签的编码长度为 $\log_2 L+1$;在该算法Request命令中, D 的长度与编码长度有关,为 $\log_2 L$, m 为1位, T 与线程的数目(N_T)有关,为 $\log_2 N_T$ 。在该算法中发送的二进制编码长度为 $\log_2 L+1+\log_2 N_T$ 。

BAS算法中,需循环搜索RFID标签,因此识别数量为 N 的标签查询次数为 $N(N+1)/2$;后退式BAS算法识别数量为 N 的标签需要的查询次数为 $2N-1$;文中算法采用并发执行后退式BAS,每个线程平均查询次数为 $(2N-1)/N_T$ 。

BAS、后退式BAS、并发处理BAS算法的传输二进制数据长度分别为 $N(N+1)/2L$, $(2N-1) \times (\log_2 L+1)$, $[(2N-1)/N_T] \times (\log_2 L+1+N_T)$ 。

以表1的20个标签为样本,比较BAS、后退式BAS和并发执行BAS算法的性能,比较指标见表3。随着标签数量、编码位数以及并发数的增加,文中算法的优势将越来越明显。对后退式BAS和并发执行BAS算法进行Matlab仿真,对RFID标签的编码为8位、64位,线程数为4和8的2种算法的吞吐量进行了比较,见图2。

表3 算法比较指标
Tab.3 Algorithm comparison index

算法	发送命令次数	Request命令传输数据总量/bit
BAS	210	1680
后退式BAS	39	156
并发执行BAS	10	60

可以看出,新算法发送命令次数和数据量都大幅度减少,分析其原因应有以下两点:后退方式减少了命令的发送;命令的参数不是RFID标签的全部ID,而是冲突的位置及数值。当标签的ID位数越多,会越明显。

各算法的执行时间比较见表4。其中RFID读写器发送1bit数据时间为 t_1 ,RFID标签应答1bit数据时间为 t_2 ,RFID进行冲突处理1bit数据时间为 t_3 。由表4可以看出,改进算法在速度上也得到了提高,分析其原因应有以下3点:并发执行处理,使冲突处

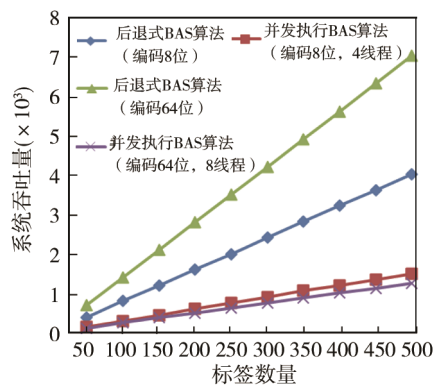


图2 后退式BAS算法和并发执行BAS算法
Fig.2 Backward BAS algorithm and multi-threading BAS algorithm

表4 算法时间比较
Tab.4 Algorithm time comparison

算法	Request命令时间	Response命令时间	冲突处理时间
BAS	$200t_1$	$132t_2$	t_3
后退式BAS	$304t_1$	$80t_2$	t_3
并发执行BAS	$64t_1$	$10.7t_2$	$t_3/4$

理时间减少;命令传输的数据减少,识别数据的时间减少;处理冲突可与命令传输多线程处理,提高了运行的速度。

4.2 并发执行BAS算法与其他算法的效率比较

算法的系统效率是比较算法好坏的标准之一,因此对ALOHA算法、QTA算法和BAS算法进行系统效率的比较。ALOHA算法的思想是发现冲突后,重新产生识别标签,当标签数量增加,冲突则呈线性增长,系统性能急剧下降。QTA算法需传输并检验标签前缀,信息处理速度较慢。BAS算法的思想是将读入的信息生成2个不相交的子集,循环处理,直到子集中只有唯一可识别的标签。

算法的系统效率定义如下:系统效率=(发送成功的标签信号/发送总标签信号) $\times 100\%$ 。

这几种算法的效率比较曲线见图3。可以看出,ALOHA算法的效率最低,随着阅读器数目的增多,冲突也随之增加,当阅读器数量达到64时,ALOHA算法的系统效率几乎为0;QTA算法虽然很好地避免了节点的冲突,但由于需要检查标签的前缀,当阅读器数目达到64时,系统效率约为40%;BAS算法将读入的信息分成2个不相交的子集,但单线程会造成系统空闲,当阅读器数目达到64时,系统效率约为67%;文中算法是在BAS算法的基础上,引入多线程后退式技术,使系统效率一直保持在90%左右。

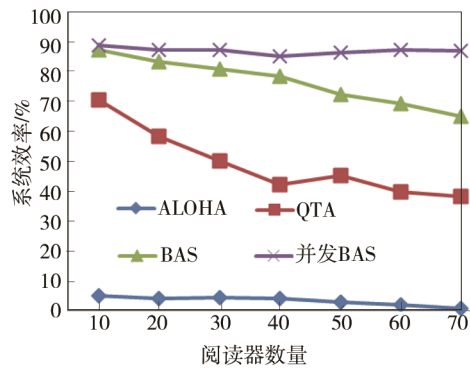


图 3 算法效率比较

Fig.3 Comparison of algorithm efficiency

5 结语

提出了并发执行的后退式防冲突算法,减少了命令传输数据的数量,缩短了读写器的识别时间,提高了包装系统的处理速度。并发执行的后退式防冲突算法融合了 BAS 算法思想,将读入的信息生成 2 个不相交的子集,循环处理,直到子集中只有唯一可识别的标签;进一步采用了多线程并发技术,对 BAS 算法中系统空闲时间进行有效的利用,提高了系统的工作效率。性能分析表明,该算法优于 BAS 算法和后退式 BAS 算法,更适用于包装标签 ID 长、标签数量大的情况。

参考文献:

- [1] 苏琦, 郭彦峰, 王思宇. 基于 RFID 技术的仓储管理系统的优化与设计[J]. 包装工程, 2009, 30(6): 82—83.
SU Qi, GUO Yan-feng, WANG Si-yu. Optimization and Design of Warehouse Management System Based on RFID Technology[J]. Packaging Engineering, 2009, 30(6): 82—83.
- [2] 谢关友, 李良春. 基于 RFID 技术的包装物流配送中心业务流程分析[J]. 物流技术, 2008(2): 122—124.
XIE Guan-you, LI Liang-chun. Analysis of Business Process of Packaging Logistics Distribution Center Based on RFID Technology[J]. Logistics Technology, 2008(2): 122—124.
- [3] 王志伟, 闫秀霞, 孙宝连. RFID 技术应用研究综述及研究趋势展望[J]. 物流技术, 2014(9): 1—5.
WANG Zhi-wei, YAN Xiu-xia, SUN Bao-lian. RFID Technology Application Research and Research Trends[J]. Logistics Technology, 2014(9): 1—5.
- [4] 马衍民. 基于多线程的后退式二进制 RFID 防冲突算法研究[J]. 中国电子商务, 2014(1): 60—61.
MA Yan-min. Based on Multithreaded Regressive Binary RFID Anti-conflict Algorithm Research[J]. Chinese E-commerce, 2014(1): 60—61.
- [5] 张婕好, 刘嵘, 汝文文. 基于 RFID 的药品生产管理及包装信息化[J]. 包装学报, 2015(4): 25—28.
ZHANG Jie-yu, LIU Rong, RU Wen-wen. Electronic Drug Production Management and Packaging Informationization Based on RFID[J]. Chinese Journal of Packaging, 2015(4): 25—28.
- [6] 韩彪, 袁海, 徐昌彪. 基于 RFID 技术的包装防伪应用系统[J]. 重庆工学院学报(自然科学版), 2008(2): 104—107.
HAN Biao, YUAN Hai, XU Chang-biao. Based on RFID Technology Packaging Anti-counterfeiting Application System[J]. Journal of Chongqing Institute of Technology (Natural Science Edition), 2008(2): 104—107.
- [7] 王雪, 钱志鸿, 胡正超. 基于二叉树的 RFID 防碰撞算法的研究[J]. 通信学报, 2010(6): 49—57.
WANG Xue, QIAN Zhi-hong, HU Zheng-chao. Study on RFID Anti-collision Algorithm Based on Binary Tree[J]. Journal of Communications, 2010(6): 49—57.
- [8] MARCZEWSKI T, MA Y P, SUN W B. Evaluation of RFID Tags to Permanently Mark Trees in Natural Populations[J]. Frontiers in Plant Science, 2016, 7: 24—26.
- [9] 王勇, 李婷. 改进的基于 ALOHA 的 RFID 防碰撞算法[J]. 电信科学, 2016(8): 77—81.
WANG Yong, LI Ting. Improved RFID Anti-collision Algorithm Based on ALOHA[J]. Telecommunications Science, 2016(8): 77—81.
- [10] SRIWAN A, SITTICHIVAPAK S. Performance Analysis of ALOHA Framework under Limited Access of Data Transmission for Active RFID System[J]. Procedia Computer Science, 2016, 86: 19—22.
- [11] HU Ming-sheng, GUO Ning. Anti-collision Algorithm for RFID Based on Dynamic Packet Query Tree[J]. Advanced Materials Research, 2011, 159: 550—555.
- [12] LI Cheng-yong, HUANG Yan, ZHAO You-gui. Based on RFID Technology of High-speed Mobile Multi-objective Binary Tree Anti-collision Algorithms[J]. Applied Mechanics and Materials, 2013, 427: 793—796.
- [13] LI Gang-sheng, PANG Yue-xin, LUO Wei. Design of RFID Based Robot System for Automatic Search of Books in Libraries[J]. Advanced Materials Research, 2012, 433: 6856—6860.
- [14] ZHENG Yuan-qing, LI Mo. Fast Tag Searching Protocol for Large-scale RFID Systems[J]. IEEE/ACM Transactions on Networking, 2013, 21(3): 924—934.