

智能包装中的 RFID 标签防碰撞算法研究

周伟辉¹, 蒋年德²

(1. 东华理工大学长江学院, 抚州 344000; 2. 东华理工大学 信息工程学院, 南昌 330013)

摘要: **目的** 解决目前物流系统中智能包装 RFID 标签冲突的问题。**方法** 在研究已有防碰撞算法的性能和缺点基础上, 针对大量标签场景设计一种分组动态帧时隙的混合查询树 (GDFSA-HQT) 算法。该算法在每一轮识别之后估计还没有识别的标签数量, 如果还没有识别的标签数量小于或等于 354 个, 则先采用动态帧时隙 ALOHA (DFSA) 算法对标签进行识别, 再采用混合树查询算法 (HQT) 进行标签识别; 若未被识别标签数大于 354 个, 则先对标签进行分组处理, 再分别采用 DFSA 和 HQT 进行标签识别。**结果** 仿真实验表明, GDFSA-HQT 算法的吞吐率能够保持在 0.82 左右。**结论** GDFSA-HQT 算法解决了标签碰撞问题, 在大量智能包装的物流系统中具有良好的应用前景。

关键词: 智能包装; RFID; 混合树查询算法; 动态帧时隙 ALOHA 算法; 物流系统

中图分类号: TB487 **文献标识码:** A **文章编号:** 1001-3563(2018)21-0011-06

DOI: 10.19554/j.cnki.1001-3563.2018.21.003

Anti-collision Algorithm of RFID Tag in Intelligent Packaging

ZHOU Wei-hui¹, JIANG Nian-de²

(1. East China Institute of Technology, Yangtze River College, Fuzhou 344000, China;

2. Information Engineering College, East China Institute of Technology, Nanchang 330013, China)

ABSTRACT: The work aims to solve the intelligent packaging RFID tags conflict in the logistics system at current stage. On the basis of studying the performance and shortcomings of existing anti-collision algorithms, a grouped dynamic frame slotted ALOHA-Hybrid Query Tree (GDFSA-HQT) algorithm was designed for a large number of tags scenarios. The GDFSA-HQT algorithm estimated the number of unrecognized tags after each round of tags identification. If the number of unrecognized tags was less than or equal to 354, the dynamic frame slotted ALOHA (DFSA) algorithm was used to identify the tags firstly, and then the hybrid tree query algorithm (HQT) was used for tags identification. If the number of unrecognized tags was greater than 354, the tags should be first grouped, and then DFSA and HQT would be used for tags identification. The simulation experiment showed that GDFSA-HQT algorithm throughput rate could be maintained at around 0.82. The GDFSA-HQT algorithm solves the tag collision problem and has a good application prospect in a large number of intelligent packaging logistics systems.

KEY WORDS: intelligent packaging; RFID; hybrid query tree algorithm; dynamic frame slotted ALOHA algorithm; logistics system

无线射频识别(Radio Frequency Identification, RFID)是一种非接触式实现标签和阅读器之间通信的自动识别技术^[1], 广泛应用于物流、图书管理、大型购物超市、智能交通、智能包装管理等领域^[2]。文中主要研究 RFID 在智能包装管理中的应用, 其原理是在智能包装上植入 RFID 标签, 通过阅读器就能实现

数据的识别、统计和判断, 不再需要人工逐个扫描, 大大节省了人力和物力^[3]。由于在物流系统的智能包装管理中会产生 RFID 多标签碰撞现象, 因此如何在避免碰撞的情况下准确、快速地读取 RFID 标签数据, 是 RFID 应用在智能包装系统中需要研究和解决的问题之一^[4]。目前已经提出了 2 种类型的防碰撞

收稿日期: 2018-09-12

作者简介: 周伟辉 (1990—), 男, 硕士, 东华理工大学长江学院助教, 主要研究方向为 RFID 技术、物联网。

通信作者: 蒋年德 (1971—), 男, 博士, 东华理工大学副教授, 主要研究方向为 RFID 技术、数字图像处理与智能信息融合。

算法,分别是基于树的防碰撞算法和基于 ALOHA 的防碰撞算法^[5]。基于树的防碰撞算法是将标签分裂在 N 个子树上,直到全部标签被识别,主要有:碰撞树(Collision Tree, CT) 算法^[6]、自适应多叉树(Adaptive Multi-tree Search, AMS)算法^[7]、改进型自适应多叉树(Improved Adaptive Multi-tree Search, IAMS)算法^[8]、自调整混合树(Adjustive Hybrid Tree, AHT)算法^[9]、改进的混合查询树(Improved hybrid query tree, IHQT)算法^[10]等。基于 ALOHA 算法的工作原理是利用时隙随机分配,标签随机选择时隙响应来达到识别标签的目的。经典的 ALOHA 算法有:动态帧时隙 ALOHA(Dynamic Frame Slotted ALOHA, DFSA)算法^[11]、分组自适应分配时隙 ALOHA (Grouped Adaptive Allocating Slot, GAAS) 算法^[12]、标签估计(Tag Estimation Method, TEM)算法^[13]、分组动态帧时隙 ALOHA (Grouped Dynamic Frame-slotted ALOHA, GDFSA)算法^[14]等。

当标签数量较大时,DFSA 算法存在吞吐量降低等缺点,HQT 算法存在碰撞时隙、空闲时隙增多等缺点。文中提出一种分组动态帧时隙的混合查询树(Grouped Dynamic Frame Slotted ALOHA - Hybrid Query Tree, GDFSA-HQT) 算法,其思想是在每轮识别以后估计还没有识别的标签数目,若还没有识别标签数小于等于 354 个,则先采用 DFSA 算法(DFSA)对标签进行识别,再采用混合树查询算法(HQT)进行标签识别;若未被识别标签数大于 354 个,则先对标签进行分组处理,再分别采用 DFSA 和 HQT 进行标签识别。

1 GDFSA-HQT 算法描述

1.1 标签数量估计算法

文中算法在下一轮识别之前都需要预估上一轮所剩标签的数量,根据标签数量来选择帧长和分组数,因此标签数量估计算法在文中是必不可少的。典型的标签数量估计算法包括 LB 估计算法、Schoute 估计算法和 CHI 估计算法等^[15]。若标签数量较大,相较于 LB 估计法和 Schoute 估计法,CHI 估计法能使误差达到最小,且吞吐率可达到最大。

CHI 估计算法将期望值与实际值相结合,以获得标签数 n ,也就是使实际时隙值 C_e , C_s , C_c 和理论时隙期望值 N_e , N_s , N_c 之间的差最小。在标签数量为 n 时的取值范围 $[C_s + 2C_c, \dots, 2 \times (C_s + 2C_c)]$ 中找到 ε 值,所对应 n 值即为估计的标签数,见式(1)^[16-18]。

$$\varepsilon(N, C_e, C_s, C_c) = \min_n \left| \begin{pmatrix} N_e \\ N_s \\ N_c \end{pmatrix} - \begin{pmatrix} C_e \\ C_s \\ C_c \end{pmatrix} \right| \quad (1)$$

式中: n 为每轮要识别标签的数量; N 为帧长度; N_e , N_s , N_c 为每轮读取之后空闲时隙、成功时隙和碰撞时隙的期望值; C_e , C_s , C_c 为每轮读取之后空闲时隙、成功时隙和碰撞时隙的实际值。

1.2 帧长和分组数的调整

通过分析 DFSA 可知,为了达到系统的最大吞吐率,需要使帧长无限接近标签数量。在实际应用中,帧长不能取任意值,必须是 2 的整数幂,并且由于硬件的限制,帧长 N 不能大于 256,即 $N_{\max} = 256$,一般取值为 2, 4, 8, 16, 32, 64, 128, 256。

如果标签数量很少时,DFSA 算法的吞吐率可以维持得较高;如果标签数较大时,DFSA 算法的吞吐率降低,识别时间也会相应地增长。为了保证帧长和识别标签数相近,可以通过标签分组来控制响应的标签数量,使算法吞吐率维持较高水平。

在文中算法中,阅读器根据 CHI 估计算法预估标签数量 n 值,再和 $N_{\max}$ 比较。如果小于 $N_{\max}$,则标签不分组;如果大于 $N_{\max}$,则对标签分组,并且在每轮识别中,只能有 1 组标签参与识别。

成功时隙数 N_s 为:

$$N_s = Nn \left(\frac{1}{N} \right) \left(1 - \frac{1}{N} \right)^{n-1} \quad (2)$$

则 n 个标签响应在帧长为 N 的吞吐率为:

$$P_N = \frac{N_s}{N} = n \left(\frac{1}{N} \right) \left(1 - \frac{1}{N} \right)^{n-1} \quad (3)$$

当 $P_N = P_{2N}$, 可以获得 2 个相邻帧性能曲线交叉处的标签数量:

$$n_{N,2N} = \frac{\ln(2)}{\ln\left(\frac{2N-1}{2N-2}\right)} \quad (4)$$

当 $N_{\max} = 256$, 将响应标签分成 m 组和 $m+1$ 组时,系统性能曲线交点处的值为:

$$P = \frac{n}{m} \left(\frac{1}{256} \right) \left(1 - \frac{1}{256} \right)^{\frac{n}{m}-1} = \frac{n}{m+1} \left(\frac{1}{256} \right) \left(1 - \frac{1}{256} \right)^{\frac{n}{m+1}-1} \quad (5)$$

则可以得出 n 与 m 的关系为:

$$n = m(m+1) \frac{\ln(m+1) - \ln(m)}{\ln(256) - \ln(255)} \quad (6)$$

将 $m=1, 2, 3, 4, \dots$ 分别带入式(6)中可以得到分组标签临界值,如 $m=1$ 时, n 为:

$$n = 2 \frac{\ln(2) - \ln(1)}{\ln(256) - \ln(255)} \approx 354 \quad (7)$$

也就是说标签分成 2 组的临界值为 354 个,依此类推,那么标签数量与分组数 m 和帧长度 N 的关系见表 1。

表 1 标签数量与帧长 N 和分组数 m 的关系
Tab.1 Relationship between the number of tags and the frame length N and the number of groups m

标签数量 n	帧长 N	分组数 m
1	2	1
2~5	4	1
6~11	8	1
12~22	16	1
23~44	32	1
45~88	64	1
89~176	128	1
177~354	256	1
355~622	256	2
623~883	256	3
884~1141	256	4
$\vdots$	$\vdots$	$\vdots$

1.3 GDFSА-HQT 算法实现步骤

GDFSА-HQT 算法的核心是在进行每轮识别之前，都要估计标签数目，以确定分组数和帧长，再采用 GDFSА 算法对标签进行分组识别，然后剩下的碰撞标签采用混合树查询算法进行识别。算法实现步骤可分为 3 个阶段：标签数量估计及分组阶段、帧时隙处理阶段和混合树碰撞标签处理阶段。算法的流程见图 1。

- 1) Query(n,N): 查询命令，在帧时隙处理阶段，阅读器会对标签发送查询命令。
- 2) request($q_1q_2\cdots q_k$): 混合树碰撞标签处理阶段查询命令，其中 $q_1q_2\cdots q_k$ 是查询前缀，碰撞标签收到该命令后把 ID 前缀与查询前缀对比，若相同，则标

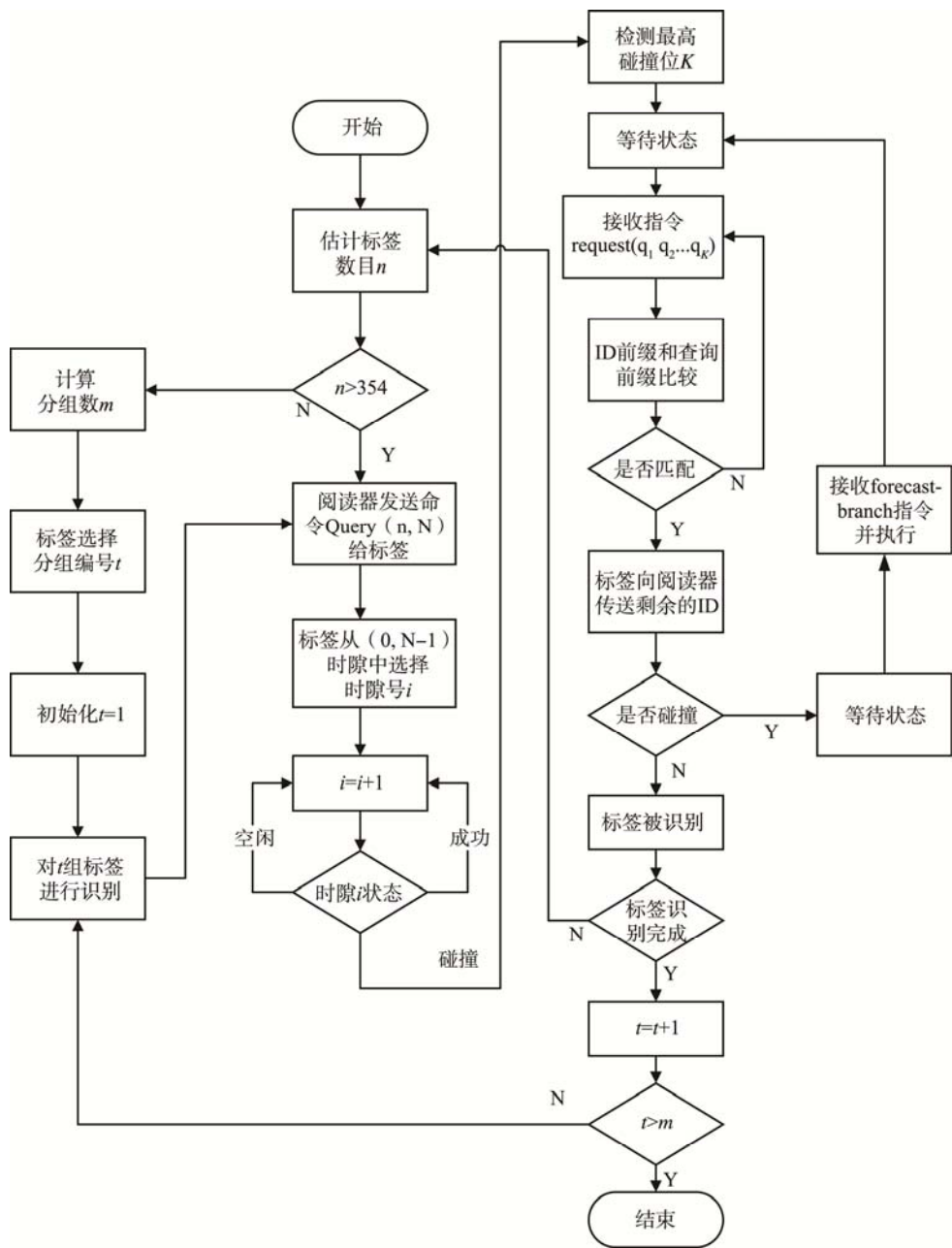


图 1 GDFSА-HQT 算法流程
Fig.1 Flow chart of GDFSА-HQT algorithm

签响应。

3) forecast-branch($q_1q_2 \cdots q_k 111 \cdots, L$ 个1): 叉树选择命令, L 的取值由碰撞标签数目决定。碰撞标签产生后, 阅读器会对碰撞标签发送混合树叉数选择命令, 以便选择最佳叉树。

1.3.1 标签数量估计及分组阶段

1) 在每一轮识别之前, 通过 CHI 估计法预估出标签数目 n 的值。

2) 如果标签数量 n 小于或等于 354, 通过 DFSA 算法进行第 1 阶段的识别, 根据表 1 选取帧长 N , 直接进入帧时隙处理阶段; 若标签数目大于 354, 先对标签进行分组操作, 由表 1 计算出分组数 m 。

3) 标签随机选择 1 和 m 之间数字作为自己的分组号 t , 并记录每组中标签的数量。

4) 初始化 $t=1$, 即从分组号为 1 的标签开始识别, 然后识别第 t ($t>1$) 组中的标签, 跳到帧时隙处理阶段步骤 1。

1.3.2 帧时隙处理阶段

1) 阅读器向第 t 组中的标签发送 Query(n, N) 命令, 此时全部标签都会应答。

2) 响应标签从 $(0, N-1)$ 时隙中按照规则选择时隙号 i 。

3) 判断时隙 i 的情况, 若时隙 i 上面有且仅有 1 个标签响应, 那么该时隙为成功时隙, 进入时隙 $i+1$ 继续识别; 若时隙 i 上没有标签响应, 那么该时隙为空时隙, 进入到时隙 $i+1$ 继续识别; 若时隙 i 上面至少有 2 个标签响应, 那么这些标签进入混合树碰撞标签处理阶段继续识别。

1.3.3 混合树碰撞标签处理阶段

1) 阅读器检测最高冲突位 K , 发送命令 request($q_1q_2 \cdots q_K$), 标签接收请求, 将 ID 前缀与查询前缀比较。

2) 判断是否相同。如果相同则响应, 响应标签向阅读器发送剩余的 ID 码; 如果不相同, 则跳到步骤 1。

3) 检测响应标签有无碰撞, 如果无碰撞, 则被成功识别; 如果存在碰撞, 则阅读器发送 forecast-branch($q_1q_2 \cdots q_K 111 \cdots, L$ 个1)命令, 并且碰撞标签接收该命令跳转到步骤 1)。

4) 判断一组中的标签是否全部识别完成, 若未全部识别完成, 则跳转到标签数目估计及分组阶段中的步骤 1 继续执行; 若全部识别完成, 则组编号加 1, 即执行 $t=t+1$ 。

5) 判断分组号 t 的大小, 此时会执行下面 2 种情况之一: $t \leq m$, 即还存在没有被识别的组标签, 跳转到标签数目估计及分组阶段中的步骤 4 继续下

一组标签的识别; $t > m$, 则表示所有的组标签全部被识别完成, 算法结束。

2 算法性能及实验仿真分析

在 Windows10 操作系统和 4G 内存环境中, 利用 MATLAB 软件分别对 GDFSA 算法、DFSA 算法、AHT 算法及文中 GDFSA-HQT 算法进行实验。实验标签数目范围是 $[0, 1000]$ 。为了确保实验的准确性, 每次仿真结果都取 100 次实验的平均值。

2.1 总时隙数分析及对比

GDFSA-HQT 算法中的帧时隙处理阶段所需时隙数通过 GDFSA 算法进行标签识别, 所耗时隙数是每一轮选取帧长之和; 在 GDFSA-HQT 算法中的混合树碰撞标签处理阶段, 根据碰撞标签树来选择树的叉数, 阅读器会自动避开空闲节点, 所以在混合树碰撞标签处理阶段可以避免空闲时隙产生。当有 2 个碰撞标签, 采用二进制树算法, 所需要的时隙数为 3 个; 当有 3 个碰撞标签, 采用三叉树算法, 所需要的时隙数为 4 个, 依此类推, 所以假设碰撞标签数为 k 个, 则可以得到该阶段所需的时隙数 M : $M=k+1$ 。

对 DFSA 算法、GDFSA 算法、AHT 算法和文中 GDFSA-HQT 算法的总时隙数做了对比实验, 见图 2。标签数量从 0~1000 变化, DFSA 算法在识别过程中所耗总时隙数呈指数增长, GDFSA 算法、AHT 算法和文中 GDFSA-HQT 算法呈线性增长, 但是相比较其他 2 种算法来说, GDFSA-HQT 算法增长得最慢。特别是标签数大的情况下, GDFSA-HQT 算法的优势最大; 当标签数量达到 1000, GDFSA-HQT 算法只需要 1051 个时隙左右, 而 DFSA 算法、GDFSA 算法、AHT 算法分别需要 5127 个、2777 个、1833 个时隙左右。分别比它们减少了约 4076 个、1726 个、782 个。

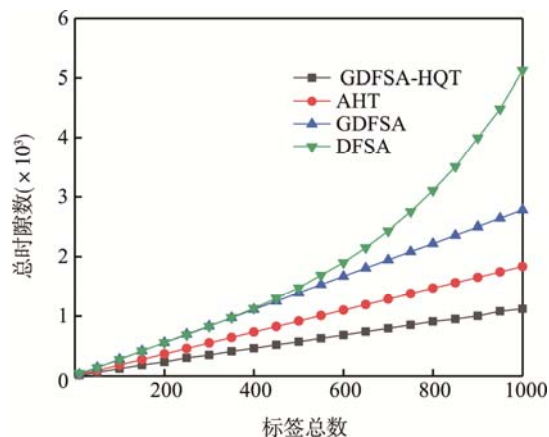


图 2 不同算法总时隙数对比
Fig.2 Comparison of total number of slots in different algorithms

2.2 碰撞时隙数分析及对比

GDFS-HQT 算法在识别过程中所需的碰撞时隙数是帧时隙处理和混合树碰撞标签处理两阶段所需碰撞时隙之和。值得注意的是，在混合树碰撞标签处理阶段，由于所选树的叉数等于碰撞标签数，即有 k 个碰撞标签就选择 k 叉树进行识别，这时子节点全部是成功时隙，则整个过程中只需根节点是碰撞时隙，所以在混合树碰撞标签处理阶段所需碰撞时隙数为 1 个。

对 DFSA 算法、GDFS-HQT 算法、AHT 算法和 GDFS-HQT 算法的碰撞时隙数做了对比实验，见图 3。标签数量从 0~1000 变化，DFSA 算法所耗总时隙数呈指数增长，而 GDFS-HQT、AHT 和文中 GDFS-HQT 等算法呈线性增长，GDFS-HQT 算法增长得最慢。当标签数目取 1000，GDFS-HQT 算法所耗碰撞时隙数为 542 个左右，比 AHT 算法减少约 396 个，比 DFSA 算法减少约 3370 个，比 GDFS-HQT 算法减少约 418 个。

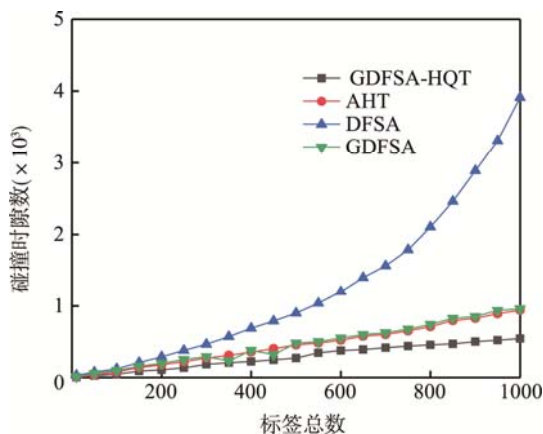


图 3 不同算法碰撞时隙数对比

Fig.3 Comparison of collision slots in different algorithms

2.3 算法吞吐率分析及对比

RFID 系统的吞吐率是衡量系统性能好坏的一个重要参数指标。若整个识别过程中响应标签数目为 n ，所耗总时隙数为 M ，则 GDFS-HQT 算法的吞吐率 E 为：

$$E = \frac{n}{M} \quad (8)$$

DFSA 算法、AHT 算法、GDFS-HQT 算法及文中 GDFS-HQT 算法的吞吐率仿真实验对比见图 4。当标签数目在 $[0, 1000]$ 变化时，AHT 的吞吐率保持在 0.55 左右，GDFS-HQT 的吞吐率始终维持在 0.36 左右；当标签数目大于 354 时，DFSA 算法的吞吐率随着标签数目的增大而急剧减小，文中 GDFS-HQT 算法的吞吐率一直维持在 0.82 左右，尤其是标签数目增加到 1000 个时，吞吐率可以达到 0.89 左右。与 AHT

算法、DFSA 算法和 GDFS-HQT 算法相比较，GDFS-HQT 算法的吞吐率提高了 104.7%，358.7% 和 145.2%。

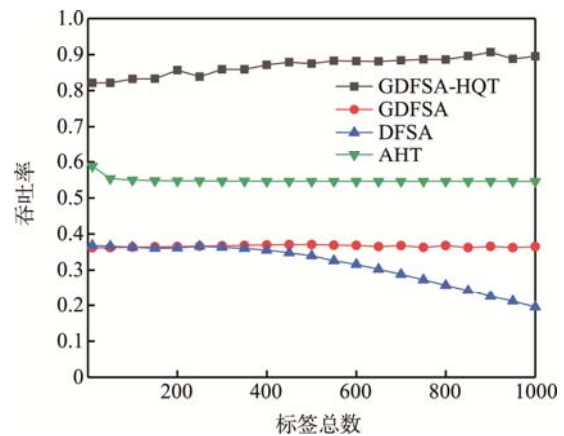


图 4 不同算法吞吐率的对比

Fig.4 Comparison of throughput rates in different algorithms

3 结语

针对物流系统智能包装中 RFID 多标签碰撞的问题，文中设计了一种分组动态帧时隙的混合查询树（GDFS-HQT）算法。通过每轮识别之前估算标签数目，再依据标签数目对标签进行分组，并结合 GDFS-HQT 算法和 HQT 算法进行识别。仿真实验表明，GDFS-HQT 算法的吞吐率能够保持在 0.82 左右，明显高于 DFSA 算法、AHT 算法和 GDFS-HQT 算法，因此 GDFS-HQT 算法可以更快、更高效地解决标签碰撞问题，它在大量智能包装的物流系统场合具有广阔的应用前景。

参考文献：

- [1] 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 Zhi-hong, WANG Yi-jun. IoT Technology and Application[J]. Acta Electronic Sinica, 2012, 40(5): 1023—1029.
- [2] 金迪. RFID 包装系统中防冲突算法研究[J]. 包装工程, 2018, 39(1): 1—5.
JIN Di. Anti-collision Algorithm in RFID Packaging System[J]. Packaging Engineering, 2018, 39(1): 1—5.
- [3] 贺彩玲, 殷峰社. RFID 技术在仓储物流行业中的应用研究[J]. 电子设计工程, 2013, 21(14): 12—14.
HE Cai-ling, YIN Feng-she. Research of Application of RFID Technology in Logistics Industry[J]. Electronic Design Engineering, 2013, 21(14): 12—14.
- [4] 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. RFID Tech-

- nology-based Application System in Goods Packing Encryption System[J]. Journal of Chongqing Institute of Technology (Natural Science), 2008(2): 104—107.
- [5] 王雪, 胡正超, 钱志鸿, 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 Xue, HU Zheng-chao, QIAN Zhi-hong. Research on RFID Anti-collision Algorithms Based on Binary Tree[J]. Journal on Communications, 2010, 31(6): 49—57.
- [6] JIA X, FENG Q, YU L. Stability Analysis of an Efficient Anti-collision Protocol for RFID Tag Identification[J]. Communications, IEEE Transactions on, 2012, 60(8): 2285—2294.
- [7] 丁治国, 朱学永, 郭立, 等. 自适应多叉树防碰撞算法研究[J]. 自动化学报, 2010, 36(2): 237—241.
DING Zhi-guo, ZHU Xue-yong, GUO Li, et al. An Adaptive Anti-collision Algorithm Based on Multi-tree Search[J]. Acta Automatic Sinica, 2010, 36(2): 237—241.
- [8] 张学军, 蔡文琦, 王锁萍. 改进型自适应多叉树防碰撞算法研究[J]. 电子学报, 2012, 40(1): 193—198.
ZHANG Xue-jun, CAI Wen-qi, WANG Suo-ping. One Anti-collision Algorithm Based on Improved Adaptive Multi-tree Search[J]. Acta Electronic Sinica, 2012, 40(1): 193—198.
- [9] 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 Jian-hua, GUO Ya-jun, HAN Lan-sheng, et al. An Adjustive Hybrid Tree Anti-collision Algorithm for RFID Multi-tag Identification[J]. Acta Electronic Sinica, 2014, 42(4): 685—689.
- [10] 史露强, 何怡刚, 罗旗舞, 等. 改进的混合查寻树 RFID 防碰撞算法[J]. 电子测量与仪器学报, 2017, 31(8): 1281—1288.
SHI Lu-qiang, HE Yi-gang, LUO Qi-wu, et al. Improved Hybrid Query Tree Anti-collision Algorithm of RFID[J]. Journal of Electronic Measurement and Instrumentation, 2017, 31(8): 1281—1288.
- [11] DENG D J, TSAO H W. Optimal Dynamic Framed Slotted ALOHA Based Anti-collision Algorithm for RFID Systems[J]. Wireless Personal Communications, 2011, 59(1): 109—122.
- [12] 张小红, 胡应梦. 分组自适应分配时隙的 RFID 防碰撞算法研究[J]. 电子学报, 2016, 44(6): 1328—1334.
ZHANG Xiao-hong, HU Ying-meng. Research on a Grouped Adaptive Allocating Slot Anti-collision Algorithm in RFID System[J]. Acta Electronic Sinica, 2016, 44(6): 1328—1334.
- [13] CHA J R, KIM J H. Novel Anti-collision Algorithms for Fast Object Identification in RFID System[C]// Proc of the 11th International Conference on Parallel and Distributed Systems, 2005: 63—67.
- [14] LIN C F, LIN F Y S. Efficient Estimation and Collision-group-based Anti-collision Algorithms for Dynamic Frame-slotted ALOHA in RFID Networks[J]. IEEE Transactions on Auto-mation Science and Engineering, 2010, 7(4): 840—848.
- [15] 卢迪, 李绅龙, 许成舜. CHI 标签估计下自适应帧长调整 DFSA 算法[J]. 哈尔滨理工大学学报, 2015, 20(1): 56—60.
LU Di, LI Shen-long, XU Cheng-shun. Adaptive Frame Size and Group Division in DFSA Algorithms on the Basis of CHI Tags' Estimation[J]. Journal of Harbin University of Science and Technology, 2015, 20(1): 56—60.
- [16] 陈春明, 冯玉田, 付良成. RFID 动态帧时隙防冲撞改进算法研究[J]. 电子技术应用, 2013, 39(1): 86—89.
CHEN Chun-ming, FENG Yu-tian, FU Liang-cheng. Improved Anti-collision DFSA Algorithm for RFID System[J]. Application of Electronic Technique, 2013, 39(1): 86—89.
- [17] 卢迪, 凌晓艳. 全分组不等长时隙的 DFSA 算法[J]. 测控技术, 2013, 32(9): 55—59.
LU Di, LING Xiao-yan. All Grouping Unequal Slot DFSA Algorithm[J]. Measurement & Control Technology, 2013, 32(9): 55—59.
- [18] 潘峥嵘, 尚凯. 一种改进的 RFID 动态帧时隙 ALOHA 算法[J]. 传感器与微系统, 2012, 31(7): 140—142.
PAN Zheng-rong, SHANG Kai. A Modified Dynamic Frame Slotted ALOHA Algorithm for RFID[J]. Transducer and Microsystem Technologies, 2012, 31(7): 140—142.