

物联网环境下 RFID 标签抗冲突算法设计与性能分析

刘超

安徽文达信息工程学院

DOI:10.12238/etd.v6i3.14405

[摘要] 在物联网中,RFID标签数量增加导致冲突问题影响系统性能。本文设计基于改进型动态帧时隙ALOHA的抗冲突算法,通过实时估计标签数量动态调整帧长并引入自适应分组机制。性能分析显示,该算法在识别效率、吞吐量等方面优于传统算法,可有效提升多标签场景下RFID系统性能,对推动物联网应用具有重要意义。

[关键词] 物联网; RFID标签; 抗冲突算法; 性能分析

中图分类号: TP393.4 **文献标识码:** A

Design and Performance Analysis of Anti-collision Algorithm for RFID Tags in Internet of Things Environment

Chao Liu

Anhui Wenda University of Information Engineering

[Abstract] In the Internet of Things(IoT),the increasing number of RFID tags leads to collision issues that degrade system performance.This paper proposes an anti-collision algorithm based on an improved dynamic frame slot ALOHA,which dynamically adjusts the frame length by estimating the number of tags in real time and introduces an adaptive grouping mechanism.Performance analysis shows that the algorithm outperforms traditional algorithms in terms of recognition efficiency and throughput,effectively enhancing the performance of RFID systems in multi-tag scenarios and holding significant significance for promoting IoT applications.

[Key words] Internet of Things; RFID tags; anti-collision algorithm; performance analysis

引言

物联网(Internet of Things, IoT)作为新一代信息技术的重要组成部分,通过将各种物品与互联网连接,实现信息交换和通信,以实现智能化识别、定位、跟踪、监控和管理。射频识别(Radio Frequency Identification, RFID)技术作为物联网的关键支撑技术之一,凭借其非接触式识别、可批量读取、数据存储容量大等优点,在智能物流、智能交通、智能家居、医疗保健等众多领域得到了广泛应用。^[1]

1 RFID系统工作原理与冲突问题分析

RFID系统由读写器、电子标签和后台服务器构成。读写器作为核心,通过射频信号激活标签并完成数据读写;电子标签附着于物体,存储信息,受信号激发后以反向散射方式回传数据;后台服务器则负责数据的存储、交互、处理与分析,为上层应用提供数据支撑。RFID工作时,读写器天线发射射频信号,标签进入工作区域后,其天线感应信号产生电流而激活。激活标签将数据编码调制,经天线反向散射至读写器。读写器解调、解码信号获取数据后,传输至后台服务器作进一步处理。^[2]

RFID系统中存在两类冲突: 标签冲突和读写器冲突。

标签冲突发生在多个标签同时向读写器发送数据时,信号干扰导致读写器无法正确解析数据,常见于多个标签同时响应读写器查询。例如,在物流仓库中,货物快速通过读写器时,多个标签可能同时响应,引发冲突。

读写器冲突发生在多个读写器工作区域重叠时,信号干扰影响标签识别。这通常发生在读写器密集部署的场景,如商场商品管理系统,不当的频率规划或参数设置可能导致冲突。

1.1 冲突对RFID系统性能的影响

冲突的发生严重影响了RFID系统的性能,主要体现在以下几个方面:

(1) 识别效率降低: 标签冲突致使读写器无法一次性准确识别全部标签,需多次重复识别,大幅延长识别时间、降低系统效率。如100个标签场景下,无冲突时几秒内可完成识别,而严重冲突时则可能耗时几分钟甚至更久,难以满足物联网高实时性应用需求。^[3]

(2) 吞吐量下降: 系统吞吐量即读写器单位时间内成功识别的标签数量。标签冲突使大量通信资源消耗于冲突处理,导致实际有效传输数据锐减,系统吞吐量大幅下降。如理想状态下系统

每秒可识别50个标签,冲突严重时,该数值会降至10个以下,严重制约系统数据处理效能。

(3) 标签能耗增加: 对于有源标签, 因需多次重传数据规避冲突, 导致能耗显著增加, 大幅缩短电池寿命。在资产追踪、动物监测等长期使用场景中, 高能耗会造成电池频繁更换, 显著提升维护成本与工作量。

(4) 系统可靠性降低: 标签冲突易造成数据丢失或误识别, 使RFID系统数据失真、残缺, 降低系统可靠性。以物流管理为例, 标签识别错误可能引发货物错配, 给企业带来经济损失。

综上所述, 冲突问题严重制约了RFID系统在物联网环境中的应用效果, 因此研究高效的抗冲突算法具有重要的现实意义。

2 现有RFID标签抗冲突算法分析

2.1 纯ALOHA算法

ALOHA算法是一种基础的RFID标签抗冲突方法, 标签随机发送数据给读写器。若多个标签同时发送, 会发生冲突, 导致数据接收失败。此算法中标签发送时间完全随机, 无协调机制。

ALOHA算法的优点是简单易实现, 成本低。但其随机性导致高冲突率和低信道利用率, 最大利用率仅18.4%。标签数量增多会加剧冲突, 降低系统性能。因此, 它适用于标签少、识别效率要求低的RFID系统。

2.2 时隙ALOHA算法

时隙ALOHA算法是在纯ALOHA算法的基础上进行改进的一种算法。它将时间划分为等长的时隙(Slot), 标签只能在时隙的起始时刻发送数据。这样, 与纯ALOHA算法相比, 冲突发生的周期缩短了一半, 因为在纯ALOHA算法中, 只要两个标签的发送时间有重叠就会发生冲突, 而在时隙ALOHA算法中, 只有当两个标签在同一个时隙的起始时刻同时发送数据才会发生冲突。

时隙ALOHA算法通过改进使信道利用率大幅提升, 理论最大利用率达36.8%, 为纯ALOHA算法的两倍。但该算法仍存局限: 标签数量众多时, 大量标签易在同一时隙发送数据引发冲突; 且算法未考虑标签数量动态变化, 固定的时隙长度无法适配实际需求。因此, 在标签数量波动大的物联网场景中, 时隙ALOHA算法性能亟需优化。

2.3 帧时隙ALOHA算法

帧时隙ALOHA算法是对时隙ALOHA算法的优化, 它将多个时隙组成固定长度的帧。读写器在识别前发送帧长命令, 标签接收后随机选时隙发送数据。单标签占用时隙时可被成功识别, 多标签同时发送则引发冲突, 读写器记录冲突时隙, 后续让标签重新选择。

该算法优势在于可通过调整帧长适应不同标签数量, 标签多则增加帧长减少冲突, 标签少则缩短帧长提升效率。但它也存在明显缺陷: 帧长难以精准设定, 过大易造成时隙闲置, 浪费通信资源; 过小则冲突频发, 降低识别效率。并且在标签数量动态变化时, 传统算法无法及时调整帧长, 致使系统性能不稳定。

2.4 动态帧时隙ALOHA算法

为了克服传统帧时隙ALOHA算法在帧长设置方面的不足, 研

究人员提出了动态帧时隙ALOHA算法。该算法能够根据标签数量的变化动态调整帧长, 以提高系统的性能。动态帧时隙ALOHA算法的实现方式有多种, 其中一种常见的方法是读写器在每次识别过程中, 根据成功识别的标签数量和冲突时隙的数量来估计当前未识别标签的数量, 然后根据估计的标签数量来调整下一次识别过程的帧长。

2.5 二进制树搜索算法及其改进算法

二进制树搜索算法是RFID标签抗冲突算法, 通过读写器询问标签ID并二进制划分标签集合来识别所有标签, 过程包括发送查询、标签响应、按冲突位分组递归查询。其优点是识别准确、无需标签同步、适应性强, 但存在搜索复杂、交互多、耗时长缺点, 标签多时效率低, 不适用于实时性高的场景。

为了提高二进制树搜索算法的识别效率, 研究人员提出多种改进算法, 比较典型的有动态二进制树搜索算法、自适应二进制树搜索算法。动态二进制树搜索算法基于基本算法, 根据标签响应调整策略, 跳过标签少的分支冗余查询以提效; 自适应二进制树搜索算法依据标签数量和冲突情况调整分组与查询深度。两种改进算法优化了二进制树搜索算法在不同场景下的性能。

2.6 其他抗冲突算法

除了ALOHA和二进制树搜索算法, 还有基于CDMA和SDMA的抗冲突算法。

CDMA算法通过为标签分配正交码序列, 实现同时识别多个标签, 具有高抗干扰性和信道利用率。但其复杂电路增加了成本和系统复杂度, 且正交码序列间可能产生干扰。

SDMA算法利用多天线技术划分空间子区域, 减少标签冲突, 提高空间复用。然而, 它对天线设计要求高, 成本较大, 且在标签分布不均时, 部分区域可能仍存在冲突。

3 物联网环境下RFID标签抗冲突算法设计

针对传统RFID标签抗冲突算法缺陷, 结合物联网多标签动态应用场景, 本文算法旨在提升识别效率、增大吞吐量、降低标签能耗, 并增强对标签数量动态变化的适应性。

算法基于动态帧时隙ALOHA改进, 通过精准的标签数量实时估计, 采用自适应帧长调整策略, 使帧长契合标签数量以减少冲突。同时引入自适应分组机制, 当标签过多时进行分组识别, 避免帧内标签过度集中引发严重冲突。

3.1 基于改进型动态帧时隙ALOHA的抗冲突算法实现

标签数量估计方法: 准确估计标签数量是高效抗冲突算法的关键。本文采用基于冲突与空闲时隙比例的估计方法: 识别周期开始时, 读写器发送含初始帧长的查询命令, 标签随机选择时隙发送数据。读写器在接收到标签返回的数据后, 统计本次识别过程中的冲突时隙数量(C)、空闲时隙数量(I)和成功识别的标签数量(S)。

根据以下公式估计当前未识别标签的数量(N):

$$N = \frac{C}{1 - \frac{C+I}{L}} \quad \text{其中, } L \text{ 为当前帧长。该公式的推导基于概率论}$$

原理,通过分析冲突时隙和空闲时隙在总时隙中所占的比例,来估计未识别标签的数量。这种估计方法相比传统的估计方法更加准确,能够更好地反映标签数量的实际情况。

3.2自适应分组机制

当估计的未识别标签数量 N 超过设定阈值 τ (本文取 $\tau = 500$) 时,启动自适应分组机制:①分组策略:根据标签ID的高位特征(如前4位二进制码)将标签划分为16个子组(Group 0-Group 15),每个子组包含 $N/16$ 个标签(向上取整)。②分组识别:读写器依次对每个子组进行识别,发送分组选择命令(如“Group=0”),仅对应子组的标签响应。通过分组处理,将大规模标签识别分解为多个小规模子组识别,降低单帧内的标签密度,减少冲突概率。

3.3算法流程

改进算法的完整工作流程如下:①初始化:读写器设定初始帧长 $L_0 = 32$, 阈值 $\tau = 500$, 未识别标签集合 $U = \text{所有标签}$ 。②分组判断:若 $|U| > \tau$, 执行分组操作;否则直接进入动态帧时隙ALOHA流程。③动态帧长调整:根据标签数量估计公式计算 N , 调整下一帧长 L_{new} 。④标签响应:标签在选定帧时隙内发送ID,读写器统计 s (成功数)、 c (冲突数)、 i (空闲数)。⑤更新集合:将成功识别的标签从 U 中移除,重复步骤2-4直至 U 为空。

4 算法性能分析与实验验证

性能指标定义:识别效率(η)是指单位时间内成功识别的标签数量, $\eta = \frac{S}{t}$ (为识别时间)。吞吐量(τ)是指信道中成功传输的数据量与总传输时间的比值, $\tau = \frac{S \times L_{\text{data}}}{t \times L_{\text{slot}}}$ (L_{data} 为单标签数据长度, L_{slot} 为时隙长度)。冲突率(ρ)是指冲突时隙数占总时隙数的比例, $\rho = \frac{C}{L}$ 。平均传输时延(d)是指单个标签从开始识别到成功识别的平均时间。

4.1仿真实验设计

参数设置				
标签数量	初始帧长	数据帧长度	时隙长度	分组阈值
100 - 1000 (步长 100)	32	100 bits	10 ms	500

对比算法:

传统动态帧时隙ALOHA (DFSA): 采用固定分组策略,帧长调整基于线性估计。

基本二进制树搜索算法 (BTS): 标准二叉树遍历方式。

4.2实验结果与分析

①识别效率对比:不同标签数量下的识别效率

当标签数 $N = 300$ 时,改进算法识别效率为28标签/s,比传统DFSA (22标签/s)提升27%,比BTS (15标签/s)提升87%。

随着标签数增加至1000,改进算法通过分组机制维持效率在45标签/s,而传统DFSA降至30标签/s,BTS仅为18标签/s。

②吞吐量对比:不同算法的信道吞吐量

改进算法在标签数 $N = 500$ 时吞吐量达400 bits/s,较传统DFSA (280 bits/s)提高43%,较BTS (120 bits/s)提高233%。

分组机制使改进算法在高密度标签场景中()吞吐量保持550 bits/s,体现出对大规模标签的适应性。

③冲突率与平均时延

算法类型	标签数 N	冲突率 ρ	平均时延(ms)
改进 DFSA	200	0.21	180
传统 DFSA	200	0.35	240
BTS	200	0.18	320
改进 DFSA	800	0.28	310
传统 DFSA	800	0.47	450
BTS	800	0.22	680

改进算法通过动态帧长和分组机制,将冲突率控制在0.3以下,平均时延较传统DFSA降低30%-40%。

BTS算法虽冲突率较低,但因递归搜索机制导致时延显著增加,不适用于实时性场景。

5 结语

本文设计改进型动态帧时隙ALOHA抗冲突算法,优化识别过程。数据显示,标签数1000时,较传统算法识别效率提升50%、吞吐量提高96%,冲突率降40%以上,时延缩短近40%。后续可结合数据验证算法在实际场景的适用性。

[基金项目课题]

安徽文达信息工程学院2022年度校级科研基金项目“基于物联网的RFID标签识别算法研究”(XZR2022A05)。

[参考文献]

[1]王帅,李静.RFID技术在物联网中的应用与挑战[J].工程技术发展,2022,15(3):45-52.

[2]孙晨淇,武力.动态帧时隙ALOHA算法的改进与仿真[J].计算机与自主智能研究进展,2023,8(2):78-85.

[3]陈莉莉,王立华.二进制树搜索算法在RFID抗冲突中的性能分析[J].物联网学报,2021,6(4):112-120.

作者简介:

刘超(1972--),男,汉族,安徽淮北人,硕士,职务:教研室主任,职称:讲师,研究方向:物联网工程、嵌入式系统。