

射频识别系统冲突防范算法^①

周朝阳

(湖北大学 图书馆, 武汉 430062)

摘要: 射频识别是物联网技术的核心, 标签冲突问题是射频识别必须要解决的一个关键问题. 针对射频识别系统中的标签冲突问题, 分析了 Aloha 算法及其改进算法. 提出了一个基于 Fibonacci Number 的动态帧时隙 Aloha 改进算法, 并建立了动态调整帧长的策略. 仿真结果表明该算法能提高系统的吞吐率和系统负载.

关键词: RFID; Fibonacci Number; Aloha 算法; 防冲突; 算法设计

Algorithms for Anti-collision in RFID Systems

ZHOU Zhao-Yang

(Library of Hubei University, Wuhan 430062, China)

Abstract: RFID plays the core role in the internet of thing, tag-collision is a key issue and must be addressed for radio frequency identification. Aim at the problem of tag-collision in RFID system, this paper analyzed ALOHA algorithm and its enhanced algorithm in detail. An new ALOHA algorithm based on Fibonacci Number was proposed, and the strategy of changing the frame length was established. The simulation indicated that the proposed algorithm can improve the throughput and load of RFID system.

Key words: RFID; Fibonacci Number; Aloha algorithm; anti-collision; algorithm design

射频识别(Radio Frequency Identification, RFID)技术是逐渐兴起的物联网^[1]的核心技术, RFID 通过无线射频方式进行非接触双向通信, 以达到识别目的并交换数据. RFID 在资产管理、物流、生产控制以及图书管理等领域产生了重要影响^[2], 将成为未来自动识别技术的主流.

标签冲突问题是 RFID 技术必须要解决的一个关键问题. 标签冲突是指在阅读器的射频作用域内存在多个标签, 并有多个(大于等于两个)标签同时响应阅读器, 这时阅读器将无法接收到正确的标签 ID 码, 这种现象称之为标签冲突. 解决冲突的算法称为防冲突算法. 良好的防冲突算法可以有效解决标签的冲突问题, 实现标签的快速、高效的识别.

常见的 RFID 防冲突算法主要包括基于二进制树的防冲突算法和基于 Aloha 的防冲突算法两类. 前者使用二叉搜索树将冲突的标签集合划分为两个标签子集, 对于可能冲突的标签集合, 采用默认的方式来解

决冲突^[7,8]. Aloha 算法最初被用来解决网络阻塞问题, 文献[6][9][10]提出使用 Aloha 方法来解决 RFID 标签冲突问题, 实现标签的高效识别.

本文在详细分析 Aloha 防冲突算法^[3]及其改进算法的基础上, 提出一种基于 Fibonacci Number 的动态帧时隙 Aloha 算法, 通过数学模型分析及其仿真模拟, 证明该算法吞吐量及系统负载两方面的特性较优.

1 Aloha算法分析研究

Aloha 算法是一种 TDMA 算法, 最初用来解决网络通信中数据包拥塞问题. 该算法被广泛应用在 RFID 系统中以解决标签冲突问题.

1.1 纯 Aloha 算法

在纯 Aloha 算法中, 当标签到达读写器区域时, 标签会自动向读写器发送自己的 ID 码. 如果在某个标签发送自己 ID 码的时段内没有其他的标签发送其 ID 码, 那么读写器将能正确的识别到此标签; 如果在某

^① 收稿时间:2013-03-31;收到修改稿时间:2013-05-16

个标签发送自己 ID 码的时段内还有其他的标签也在发送自己的 ID 码, 那么读写器将不能正确识别该标签, 此时即产生冲突. 当冲突发生时, 读写器将发送一个停止发送的命令让所有标签停止当前发送并等待一个随机时间后再重发自己的 ID 码. 如此循环, 直到所有标签都被正确识别. 纯 Aloha 算法思想比较简单, 模拟运行如图 1 所示. 当两标签开始发送自身 ID 码的时间间隔小于 T_0 时, 就会发生标签冲突. 纯 Aloha 算法中发生冲突率的概率很大.

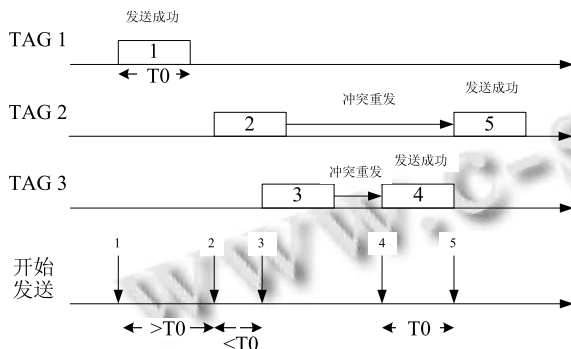


图 1 纯 Aloha 算法模拟图

1.2 静态帧时隙 Aloha 算法

静态帧时隙 Aloha 算法是对 Aloha 算法的改进. 在该算法中, 时间被分成多个离散的时隙, N 个时隙组成一个帧, 标签只能在每个时隙的开始处发送信号. 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, 每个标签随机地选择一个时隙, 并在该时隙发送自己的 ID 码. 如果一个时隙只被一个标签选中, 则在这个时隙传输的 ID 码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 ID 码则冲突产生, 读写器不能识别发送 ID 码的标签. 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.

1.3 动态帧时隙 Aloha 算法

动态帧时隙算法是对静态帧时隙算法的进一步改进. 在静态帧时隙算法中, 帧的长度是固定的, 这成为影响该算法性能的一个主要瓶颈, 一方面, 当标签数远大于帧的长度时, 在帧周期内将会有频繁的冲突发生, 影响到标签的正确识别; 另一方面, 当标签数

远小于帧的长度时, 将会有大量时隙处于空闲状态, 从而导致时间上的浪费, 进而影响到整体的吞吐量. 动态帧时隙 Aloha 算法的思想是在算法的运行过程中动态的调整帧的长度, 尽量减少在一个帧周期内发生标签冲突的概率, 并提高帧内时隙的利用率.

调整帧长的方法大致可以分为两类:

第一类: 通过实时估计读写器射频作用域内的标签数量 N , 设定帧内的时隙数(帧的长度) $F=N$. 具体的估计方法有多种, 文献[4]中有详细的介绍, 此处不再赘述.

第二类: 按照一定的变化规律来改变帧的长度. 常见的方法是使用公比 $q=2$ 的等比数列 $\{2, 4, 8, 16, 32, 64, 128, 256\}$ 来调整帧的长度^[5], 当冲突率较高时, 帧的长度增倍, 当冲突率较低时, 帧的长度减半.

2 基于 Fibonacci Number 的改进的算法

本算法是基于动态帧时隙 Aloha 算法的第二类帧长度调整方法设计的, 在算法中引入改进的 Fibonacci Number, 使帧长随算法的运行动态的按照 Fibonacci Number 的规律调整. 改进后的 Fibonacci Number 为: $f(0)=1, f(1)=2, f(2)=3$, 以此类推, 当 $n \geq 2$ 时, $f(n)=f(n-1)+f(n-2)$. RFID 读写器内保存 $f(n-2), f(n-1), f(n)$ 三个变量的值.

算法的基本思想是: RFID 读写器以初始帧长 $F=f(2)=3$ 发送询问指令, 根据当前帧周期内接收到标签 ID 信号的情况, 计算出当前帧周期的冲突率 p_c 和 p_e 空闲率, 然后 RFID 读写器根据 p_c 和 p_e 的大小来决定是否对下一个帧周期的帧长做出调整. 具体的调整策略为: 当冲突率 p_c 大于某个值时, 按照 Fibonacci Number 增加帧长; 当空闲率 p_e 大于某个值时, 按照 Fibonacci Number 减小帧长; 其他情况下则保持帧长不变.

下面来推导 p_c 和 p_e 的阈值. 设当前帧周期长度 $F=f(n)$, 当前标签总数为 r , 冲突率为 p_c , 空闲率为 p_e . 由于每个标签在帧内选择某一个时隙发送 ID 码是随机事件, 其概率为 $p = \frac{1}{f(n)}$.

(1) 需要增加帧长的情况.

在需要调整增加帧长的状态下, 当前标签总数为 r 满足以下关系①.

$$r > f(n) \quad (1)$$

由于在某一时刻产生标签冲突的标签数量均值为 $r \times p = r/f(n)$, 因此, 在当前帧周期内冲突标签总数为 $r \times p \times f(n) \times p_c = r \times p_c$. 根据 Fibonacci Number 的递增规律, 当 $r \times p_c < f(n-2)$ 时, 如果做增加帧长的调整会导致下一个帧周期内的空闲率 p_e 急剧增加, 从而导致帧长调整震荡(马上又做减小帧长调整), 因此此时应保持帧长不变; 当 $r \times p_c \geq f(n-2)$ 时, 做增加帧长的调整会使效果最佳. 因此, 应该做增加帧长调整时冲突标签总数应满足以下关系②.

$$r \times p_c \geq f(n-2) \tag{2}$$

由①、②两式得冲突率 p_c 的最大值 $p_{cmax} = f(n-2)/f(n)$. 即当冲突率 $p_c \geq p_{cmax} = f(n-2)/f(n)$ 时, 需要对下一个帧周期的长度作增加调整, 下一个帧周期长度 $F = f(n+1) = f(n) + f(n-1)$.

(2) 需要减小帧长的情况.

在需要调整减小帧长的状态下当前标签总数为 r 满足 $r < f(n)$. 但是, 当 $r \geq f(n-1)$ 时, 如果减小帧长则会使下一个帧周期内的冲突率 p_c 显著增大, 从而会导致帧长调整震荡, 因此此时应保持帧长不变; 当 $r < f(n-1)$ 时做减小帧长调整效果最佳. 因此当前标签总数为 r 应满足以下关系③

$$r < f(n-1) \tag{3}$$

此时, 处于空闲状态的时隙数为 $f(n) \times p_e$, 由于在同一个时隙中发送 ID 码的标签均值为 $r \times p = r/f(n)$,

所以平均的空闲标签数为 $f(n) \times p_e \times r \times p = r \times p_e$, 根据 Fibonacci Number 的变化规律, 当 $r \times p_e < f(n-2)$ 时, 如果做减小帧长的调整则会增加下一个帧周期内的冲突率, 同样导致帧长调整的震荡; 当 $r \times p_e \geq f(n-2)$ 时减小帧长效果性能最佳. 因此, 应该减小帧长时平均的空闲标签数应满足以下关系④.

$$r \times p_e \geq f(n-2) \tag{4}$$

为了使帧周期内的空闲时隙尽量减少, 提高时隙的利用率, 应取 p_e 的最小值. 由③、④两式得到 $p_e \geq p_{emin} = f(n-2)/f(n-1)$. 即当空闲率 $p_e \geq f(n-2)/f(n-1)$ 时, 应该对下一帧周期的长度作减小调整, 下一个帧周期长度 $F = f(n-1) = f(n) - f(n-2)$.

综上所述, 帧长的调整策略为: 当冲突率

$p_c \geq f(n-2)/f(n)$ 时, 按照 Fibonacci Number 增加帧长; 当空闲率 $p_e \geq f(n-2)/f(n-1)$ 时, 按照 Fibonacci Number 减小帧长; 其他情况下则保持帧长不变. 此外, 冲突率 p_c 对空闲率 p_e 具有优先判决权, 即当冲突率 p_c 和空闲率 p_e 同时满足调整条件时, 优先按增加帧长调整. 算法流程如图 2 所示

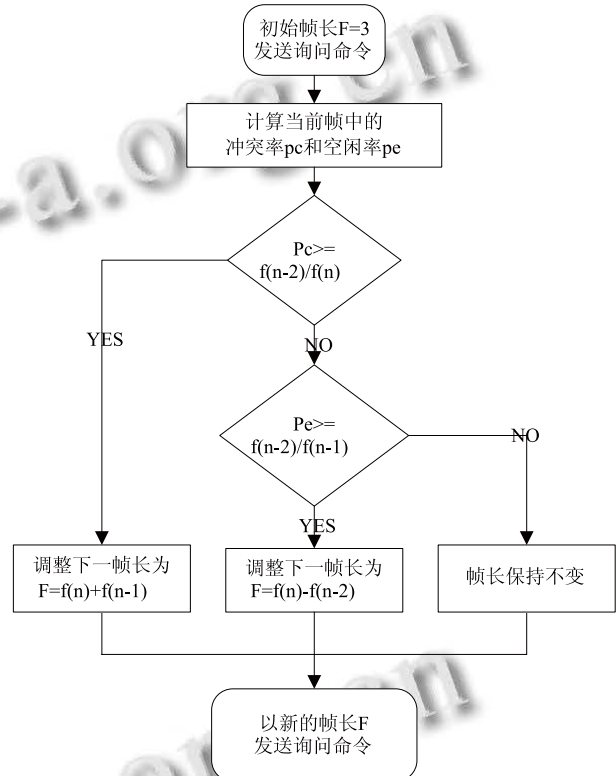


图 2 算法流程

3 仿真分析

用于比较算法性能的两个参数是吞吐量 S 和系统负载 G. 吞吐量 S 是指在帧的一个时隙 T0 内成功发送的平均标签数. 系统负载 G 是指在一个时隙 T0 内总共发送的平均标签数, 包括发送成功的标签数和因冲突未发送成功的标签数. 在标签到达服从泊松分布的条件下^[6], 吞吐量 $S = (r/f(n)) \times e^{-r/f(n)}$, 系统负载 $G = r/f(n)$.

用于参考对比的算法分别是固定帧长 $F = 32$ 的静态帧时隙 Aloha 算法、初始帧长 $F = 2$ 的按 2 的幂次递增的动态帧时隙 Aloha 算法以及本文提出的改进算法.

仿真实验采用 Matlab 7 平台, 记录标签数从 1 到 50 均匀变化时的吞吐量和系统负载值做对比. 从图 3 可见, 在系统吞吐率性能上, 本文提出的基于

Fibonacci Number 的改进算法对静态帧时隙算法有显著改进, 对按 2 的幂次递增的动态帧时隙 Aloha 算法也有明显的改进, 该改进算法能在平均吞吐率大于 0.35 的稳定状态下工作. 图 4 是系统负载图, 从图中可知, 基于 Fibonacci Number 的改进算法的系统负载比其他两个算法更接近于 1, 这表明该改进算法的冲突率和空闲率总体均值更小、更均衡.

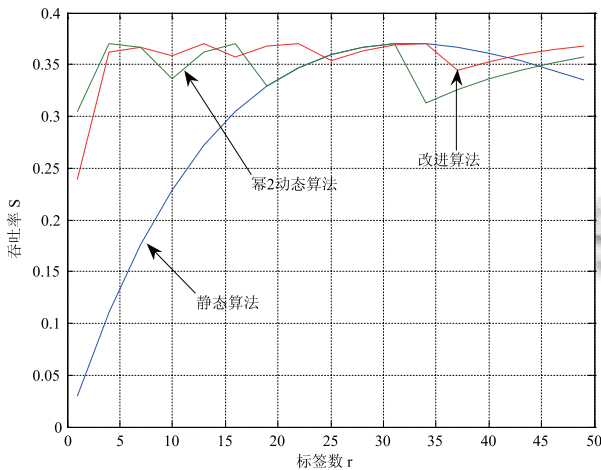


图 3 吞吐量

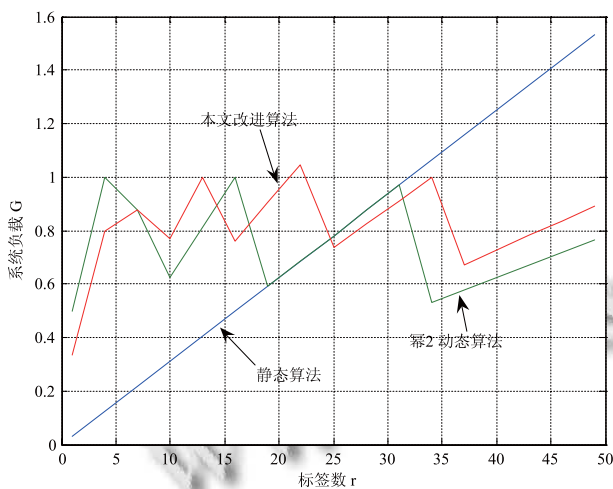


图 4 系统负载

仿真结果表明, 通过引入 Fibonacci Number, 并且综合考虑当前帧周期内的冲突率和空闲率来决定如何调整下一个帧周期的长度的方法, 能够提高动态帧时隙 Aloha 算法的性能.

4 结语

防冲突算法是提高 RFID 系统识别率的关键. 本文通过对纯 Aloha 算法及其改进算法进行详细的分析研究, 进而提出了一种基于 Fibonacci Number 的动态帧时隙 Aloha 算法, 并以严格的数学推导确定了调整帧长的条件, 建立了帧长调整策略. 在标签到达服从泊松分布的条件下, 通过 matlab 软件的仿真比较, 结果表明该改进算法吞吐率和系统负载两方面的性能都有明显的提高.

参考文献

- 冯波. 一种新的 RFID 标签识别防冲突算法. 自动化学报, 2008, 6: 632-639.
- 谢振华, 赖声礼, 陈鹏. RFID 技术和防冲撞算法. 计算机工程与应用, 2007, 6: 227-239.
- Tanenbaum A. Computer networks (4th Edition). New Jersey, Prentice Hall, 2003.
- 张颇, 崔结. RFID 系统中一种改进的防冲撞算法. 计算机应用, 2008(8): 2141-2143.
- 吴伟贞, 黄云鹰, 郭栋, 郭东辉. 基于时隙 ALOHA 的 RFID 防冲突算法及其系统实现方案的分析研究. 中国集成电路, 2008(4): 85-90.
- Schoute F. Dynamic frame length ALOHA. Communications, IEEE Transactions on, 1983, 31(4): 565-568.
- Myung J, Lee W, Srivastava J. Adaptive binary splitting for efficient RFID tag anti-collision. Communications Letters, IEEE, 2006, 10(3): 144-146.
- Pan L, Wu H. Smart trend-traversal: A low delay and energy tag arbitration protocol for large RFID systems. INFOCOM 2009, IEEE. IEEE, 2009: 2571-2575.
- Bin Z, Kobayashi M, Shimizu M. Framed ALOHA for multiple RFID objects identification. IEICE Transactions on Communications, 2005, 88(3): 991-999.
- Lee SR, Joo SD, Lee CW. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification. Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on. IEEE, 2005: 166-172.