

基于维码数的 RFID 混合防碰撞算法

黄庆欢 郑嘉利 韦冬雪 邓林

(广西大学计算机与电子信息学院 南宁 530004)

摘要 根据树搜索算法和时隙 Aloha 算法思想,提出了一种基于维码数的自适应混合防碰撞算法。针对 RFID 系统中同个阅读器可读范围内多个标签的碰撞问题,新算法先根据标签位数进行自适应分维,通过检测碰撞位,对不同维码数的标签组采用不同的策略推算标签维 ID 序列。算法还引入了维码数堆栈,减少了不必要的空闲时隙。仿真实验结果表明,新算法搜索次数较传统算法减少了 75%,搜索深度减少了 50%,提高了系统性能。

关键词 RFID,维码数,自适应,防碰撞

中图分类号 TP301.6,TP391 **文献标识码** A

Mixed RFID Anti-collision Algorithm Based on Dimensional Code Number

HUANG Qing-huan ZHENG Jia-li WEI Dong-xue DENG Lin

(School of Computer and Electronic Information, Guangxi University, Nanning 530004, China)

Abstract Combining ideas of Binary Search algorithm and Slotted Aloha algorithm, an adaptive hybrid anti-collision algorithm based on the code number of every dimension was proposed. To solve the collision problem about numbers of tags in the range of a same reader in RFID(Radio Frequency Identify), the new algorithm firstly divides the tags into groups based on the bits of tags adaptively. Through detecting the collision bits, different tags adapt different tactics to calculate the ID numbers. The new algorithm also introduces a stack to save the dimensional code numbers, reduces unnecessary idle time slot. Simulation results show that, comparing with the traditional algorithm, the new algorithm improves system performance, reduces search times by 75% and search depth by 50%.

Keywords RFID, Dimensional code number, Adaptive, Anti-collision

物联网是在计算机互联网的基础上,利用数据通信、射频识别等技术,通过计算机互联网实现物品的自动识别和信息的互联和共享^[1]。RFID 技术是物联网技术的核心,基本原理是射频信号的在空间中传输具有反射特性,标签附着在所要跟踪的物品上。通过目标物品上的标签响应,对物体进行跟踪识别和信息采集。

在日常的应用系统中,通常会存在多个阅读器和多个标签,多个标签接收到阅读器指令后选择同一个时隙在同一个频段回复阅读器,造成信息之间的相互干扰,碰撞问题由此产生。为了解决该类碰撞问题,通过设置相关命令操作,使得同一个阅读器范围内的标签能有序接入,该过程就是防碰撞算法^[2]。

1 防碰撞算法研究现状

RFID 防碰撞算法主要分为:阅读器防碰撞算法和标签防碰撞算法。解决阅读器碰撞问题的算法主要分为协调计划算法和功率控制算法^[3]。本文研究的是标签防碰撞算法。文献^[4]提出了一种基于码分多址的防碰撞算法。现在最常用的标签防碰撞算法是采用时分多址(TDMA)的方法,大致可分为基于 Aloha 不确定型防碰撞算法、基于树搜索的确定型

防碰撞算法和混合改进型防碰撞算法。

1.1 基于 Aloha 不确定型防碰撞算法

N. Abramson 首次提出了 Aloha 算法,用于解决数据包通信网络中的传输问题^[5]。Aloha 算法的优点是防碰撞流程简单,其中 DFSA 算法能根据每次查询中碰撞和空闲时隙的情况动态改变下一个阅读循环中每帧的时隙个数,以符合实际系统中标签数量不确定的需求。但基于 Aloha 的防碰撞算法属于不确定性算法,接入标签的总数决定了系统的识别效率,当标签饱和度较高时,识别效率明显降低。

1.2 基于树搜索确定型防碰撞算法

该类算法主要基于二叉树思想,将发生碰撞的标签分为两个子集^[6]。树搜索防碰撞算法是基于轮询的方法,标签和阅读器之间的通信通过握手原则进行通信^[7],主要分为二进制搜索算法(BS)、动态二进制搜索算法(DBS)、跳跃式动态树形反碰撞算法(JDBS)等^[8]。JDBS 是在 DBS 基础上进行改进的,当无碰撞时该方法后退到父节点而不是根节点,这减少了搜索次数。文献^[9]提出了一种自适应的多叉树算法,当碰撞位为 2 时,采用二叉树搜索算法,当碰撞位大于 2 时,采用四叉树算法。基于树的搜索算法属于确定性的算法,但该算法

本文受广西自然科学基金面上项目(2011GXNSFA018169),广西自然科学基金青年项目(2012GXNSFBA053160),广西科学研究与技术开发计划课题(12118017-9A)资助。

黄庆欢(1988-),女,硕士生,主要研究方向为物联网 RFID 技术,E-mail:lemontree312@live.cn;郑嘉利(1979-),男,博士,副教授,主要研究方向为多媒体通信、视频编码、物联网技术、RFID;韦冬雪(1988-),女,硕士生,主要研究方向为 RFID 标签防碰撞技术;邓林(1990-),男,硕士生,主要研究方向为 RFID 室内定位。

要遍历所有可能的组合,当存在很多标签时,会导致系统识别时间过长。

1.3 混合改进型标签防碰撞算法

混合改进型标签防碰撞算法结合了基于时隙和基于树搜索思想。在基于码距反演的 RFID 防碰撞算法^[10]中,所有标签根据码重即标签 ID 中“1”的个数分成不同的码组,但该算法仅仅根据码重并不能完全区分码组,并且所有位数参与码重计算,效率降低,当标签数量较少时会产生很多空白时隙。文献^[11]首次提出了一种维分编码 RFID 防碰撞算法,该算法将二进制代码按照空间维数进行编码,每维以 4bit 二进制数据编码,但其需要对维进行再次编码,过程复杂且数据传输量大。本文结合以上算法的优点提出了改进算法。首先,针对文献^[10]中对相同码重的码组根据其分组原则进行必要的二次分组的问题,新算法提出了新的原则即根据标签的位数关系进行自适应分维,选择每 3 或 4bit 为一维,有效地对标签进行了分组,无需二次分组;然后新算法根据维顺序分维识别标签,引入堆栈存放标签组的维码数,减少了标签与阅读器之间的通信量,针对文献^[10]空白时隙较多的问题,维码数依次出栈,可以有效地控制空白时隙的产生;最后,阅读器根据不同的维码数分组采用不同策略进行维 ID 的推算,大大减少了搜索次数。若不能分成整除维,按照分维策略,剩余位只可能是一位或两位,当剩余位为 1 位时,可直接发送前缀,通过检测碰撞位推测该位信息;当剩余位为 2 位时,异或结果为 0 的标签先回复指令,根据碰撞位推算出序列,用相同的方法识别异或结果为 1 的标签组。

2 基于维码数的自适应混合防碰撞算法

2.1 关于算法的约定

1) 码数指的是标签 ID 中为“1”的位数的个数。维码数指的是根据标签 ID 位数自适应分维,每 3 或 4bit 为一维,计算某维中比特位为 1 的位数,即为该维的维码数。例如,某标签 ID 号为 001100000111,该 ID 号总共 12 位,因此分为三维,分别是 0011、0000、0111,则对应的每维维码数为 2,0,3。

2) 常用指令集:

a) 查询指令 SearID(null, 1), 标签返回其第 1 维维码数,并将维码数压栈。

b) 查询指令 SearID(null, 1, m), 第一维维码数为 m 的标签返回其第 1 维 ID。

c) 查询指令 SearID(prefix, n, m, [z]), m 指标签第 n 维的维码数,当 m=null 时,标签回复其第 n 维维码数; m 为数字时,标签回复其第 n 维 ID。z 是可选参数,可能的取值为“0”和“1”,该参数只在维码数为 2 且碰撞位为 4 位时使用。标签自身设有维数计数器,初始值为 1。

2.2 算法步骤

(1) 逻辑分维阶段,首先根据标签位数的倍数关系进行自适应分维。标签位数能被 3 或 4 整除的比特数为一维;若同时能被 3 和 4 整除,则优先选择 4bit 为一维;若不能被整除,则选择余数较小的进行分维;若余数相同,则选择分维数较小的位数。

(2) 阅读器发送 SearID(null, 1) 指令,阅读器范围内所有标签返回其第 1 维维码数,阅读器将各标签返回的第 1 维维码数压栈,转到步骤(3)。

(3) 阅读器发送 SearID(prefix, n, null) 指令,阅读器范围

内所有标签返回其第 n 维维码数,若只有一个标签回复,则转到步骤(7);若有多个标签回复,阅读器将各标签返回的第 n 维维码数压栈。

(4) 维码数出栈,根据每维维码数选择步骤:

i. 若按照每 4bit 为一维分维:

1) 针对维码数为 0 或 4 的标签组,即可推算出第 n 维 ID 号为 0000 或 1111,标签维计数器加 1,转到步骤(5);

2) 针对维码数为 1 或 3 的标签组,阅读器发送 SearID(prefix, n, 1 或 3) 命令,维码数为 1 或 3 的标签发送其第 n 维 ID 号,若无碰撞直接识别,则转到步骤(7);若碰撞,则检测碰撞位即可推算出标签第 n 维 ID 号,标签维计数器加 1,转到步骤(5);

3) 针对维码数为 2 的标签组,阅读器发送 SearID(prefix, n, 2) 命令,第 n 维维码数为 2 的标签发送其第 n 维 ID 号,检测碰撞位:

a) 若碰撞位小于 4 位,阅读器可直接推算出标签第 n 维 ID 号,标签维计数器加 1,转到步骤(5);

b) 若碰撞位等于 4 位时,采用二叉树的思想,阅读器先发送 SearID(prefix, n, 2, “0”),第 n 维最高位为 0 的标签发送该维剩余 3 位 ID 号,通过检测碰撞位可以推算出该维完整的 ID 号;阅读器再发送 SearID(prefix, n, 2, “0”),第 n 维最高位 0 的标签发送该维剩余 ID 号,推算出该维完整的 ID 号,标签维计数器加 1,转到步骤(5)。算法流程如图 1 所示。

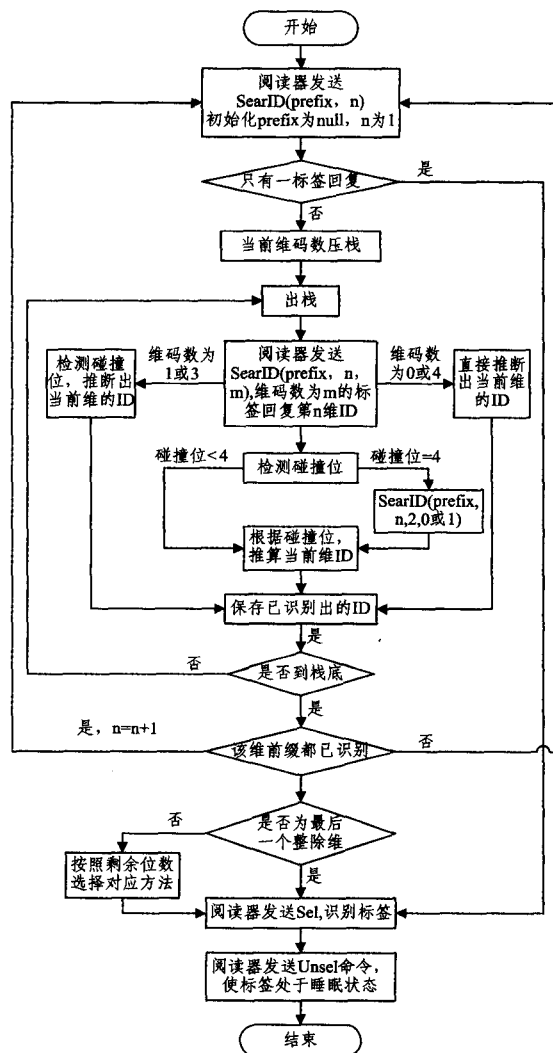


图 1 以 4bit 分维的算法流程图

ii. 若按照 3 比特为一维分维:

1) 针对维码数为 0 或 3 的标签组, 说明该维 ID 序列为 000 或 111, 阅读器可以直接将该维 ID 序列保存, 转到步骤(5)。

2) 针对维码数为 1 或 2 的标签组, 标签返回其该维 ID 序列, 检测碰撞位, 即可按照维码数推算出该维 ID。算法流程如图 2 所示。

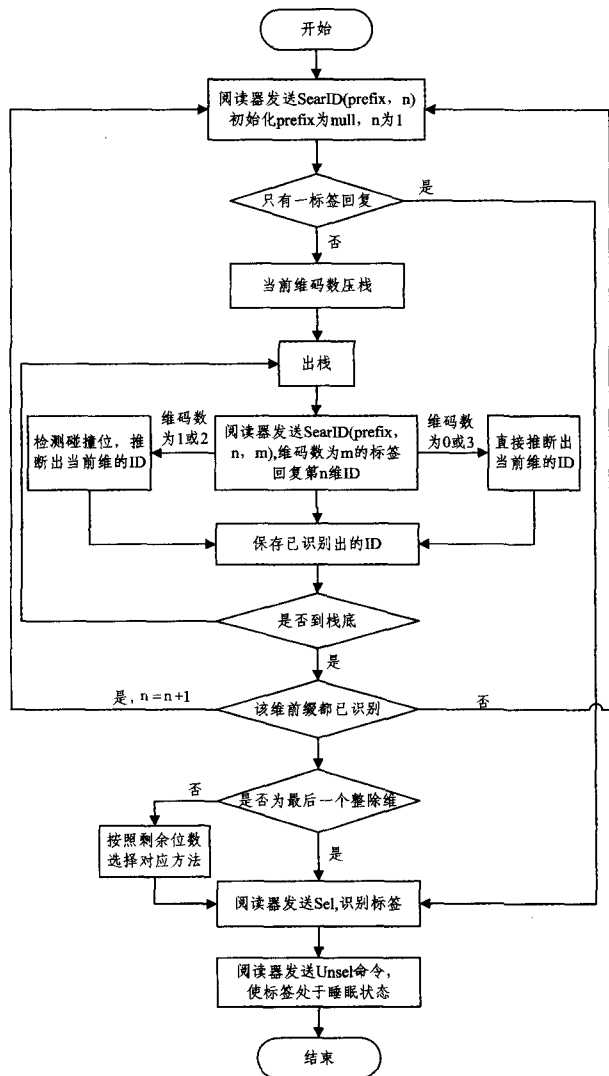


图 2 以 3bit 分维的算法流程图

(5) 阅读器将识别出的第 n 维 ID 作为前缀保存, 判断是否到达栈底, 若不是栈底, 则转到步骤(4); 若为栈底, 即已识别完第 n 维的 ID 号, 阅读器维计数器加 1 (初始值为 1), 当前 $n = n + 1$, 转到步骤(3)。

(6) 判断是否到最后一维, 若是最后一维且能分成整维的标签组, 算法结束; 若是最后一维但不能分成整维的标签组, 只剩 1 位时, 阅读器发送 SearID(prefix), 通过检测碰撞位来推算剩余 ID 序列; 剩余位为 2 位时, 异或结果为 0 的标签先返回其剩余位 ID 信息 (ID 位数 - 阅读器维计数器 $\times 4$ 或 3), 通过检测碰撞位来推算剩余 ID 序列, 再用相同的方法来识别异或结果为 1 的标签。

(7) 阅读器对对应的标签依次发送 Sel、Read、Unsel 指令, 重复以上步骤, 依次将已存储的前缀对应的标签识别完。

2.3 算法举例

假如阅读器范围内现有 6 个标签, 标签的 ID 分别为 A: 0000 0100, B: 1011 0001, C: 0101 1010, D: 0111 1100, E: 0101

1100, F: 0101 0101; 标签位数为 8bit, 能被 4 整除, 则按照每 4bit 为一维分维。

1) 阅读器发送 SearID(null) 指令, 其范围内所有标签回复其第一维维码数, 将维码数堆栈, 依次为 0, 3, 2。

2) 栈顶元素 0 先出栈, 维码数为 0 可直接推算其第一维 ID 为 0000。

3) 维码数 3 出栈, 阅读器发送 SearID(1, 3), 标签 B 和 D 回复, 解码为 ?? 11, 因为维码数为 3, 可直接推算出 ID 为 0111 和 1011;

4) 维码数 2 出栈, 阅读器发送 SearID(1, 2), 标签 C、E 和 F 返回其第一维 ID, 无碰撞, 译码得出 0101。

5) 判断已到栈底, 阅读器和标签维计数器加 1。阅读器发送 SearID(0000, 2, null), 只有标签 A 回复其第二维维码数, 直接识别, 并使标签 A 进入休眠状态。阅读器发送 SearID(1011, 2, null), 标签 B 回复其第二维 ID 号, 无碰撞, 直接识别, 并使标签 B 进入休眠状态。阅读器发送 SearID(0111, 2), 标签 D 回复其第二维 ID 号, 无碰撞, 直接识别, 并使标签 D 进入休眠状态。

6) 阅读器发送 SearID(0101, 2, null), 标签 C、E、F 返回其第二维维码数, 不止一个标签回复, 维码数 2 压栈。阅读器再发送 SearID(0101, 2, 2), 前缀为 0101 的第二维维码数为 2 的标签 C、E、F 回复其第二维 ID, 译码得 ????

7) 阅读器发送 SearID(0101, 2, 2, 0), 符合前缀 0101 第二维维码数为 2 且最高位为 1 的标签 F 回复第二维剩余 ID, 无碰撞, 直接识别, 阅读器发送 Unsel 使标签 F 休眠。

8) 阅读器发送 SearID(0101, 2, 2, 1), 符合前缀 0101 第二维维码数为 2 且最高位为 1 的标签 C、E 回复其第二维剩余 ID 号, 检测碰撞位, 译码可得 ?? 0, 因为维码数为 2, 最高位已为 1, 可推算出标签 C、F 该维 ID 为 010 和 100, 已是最后一个整除维, 标签全部识别。

3 算法性能分析与分析

3.1 性能分析

如图 3 所示, 新算法结合了二进制搜索和时隙 Aloha 算法的思想, 图 3 右边所示, 时隙 2 和时隙 8 虽然是碰撞时隙, 但可根据碰撞位推断出第一维 ID。由图可以看出, 针对算法举例中的 6 个标签, 动态二进制搜索算法搜索深度为 6, 而新算法为 3, 搜索深度减少了一半, 同时还避免了空白搜索时隙的产生。

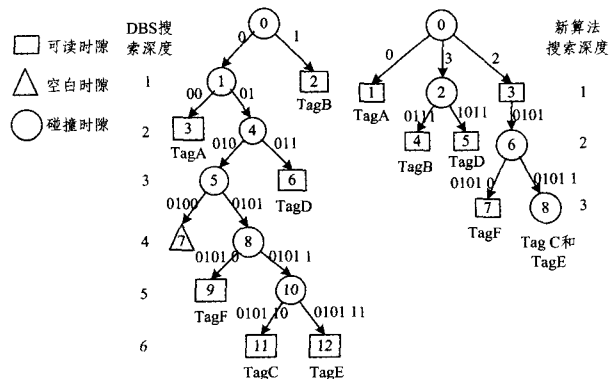


图 3 DBS 和新算法搜索深度比较

当标签位数为 m 时, 标签数目最大为 2^m 。DBS 算法需要的总搜索次数^[6]为:

$$N_{DBS} = 2 * 2^m - 1$$

新算法需要的总搜索次数为:

$$N_{new} = (2 * 2^m - 1) / 8$$

新算法的搜索次数是 DBS 算法的 1/8。

分析该算法的搜索次数, 设标签编码位数为 n , 能分成 m 个整维和 k 位剩余, 假设标签个数为 2^n , 那么

(1) 当标签按照每 3bit 分维时, 搜索次数为:

$$N = 3 \times \frac{2^{3m} - 1}{2^3 - 1} + x, \begin{cases} x=0 (k=0) \\ x=2^{n-1} (k=1) \\ x=3 \times 2^{n-1} (k=2) \end{cases}$$

(2) 当标签按照每 4bit 分维时, 搜索次数为:

$$N = 6 \times \frac{2^{4m} - 1}{2^4 - 1} + y, \begin{cases} y=0 (k=0) \\ y=2^{n-1} (k=1) \\ y=3 \times 2^{n-1} (k=2) \end{cases}$$

证明:

i. 当编码位数为 1 时, 只需发送一次搜索指令即可得到标签 ID;

ii. 当编码位数为 2 时, 发送第一次 SearID, 检测到碰撞位, 阅读器发送第二次 SearID(0), 异或结果为 0 的标签回复 ID, 产生碰撞, 由于异或结果为 0 且会产生碰撞的情况只有在 00 和 11 两种标签同时存在时才会出现, 因此可推算出标签 ID。用相同的方法得到异或结果为 1 的标签;

iii. 当按照 3bit 分维时, 首先识别第一维, 发送第一次 SearID, 得到第一维维码数, 维码数 0 或 3 可以直接推断其维 ID; 发送 SearID(1, 1), 根据碰撞位推测出第一维维码数为 1 的标签维 ID; 发送 SearID(1, 2), 根据碰撞位可推算出第一维维码数为 2 的标签维 ID, 假设标签有 2^n 个, 此时碰撞位为???, 只有一种可能, 因此可推测出 011、101 和 110 3 种 ID 都存在; 因此, 识别第一维总共的搜索次数为 3 次。识别第二维时, 在第一维保存了 2^3 个前缀, 因此要执行 3×2^3 次... 识别第 m 维时, 需要的搜索次数为 3×2^{3m} 。因此, 总的搜索次数为:

$$\begin{aligned} N &= 3 + 3 \times 2^3 + 3 \times 2^6 + \dots + 3 \times 2^{3m} \\ &= 3 + 3 \times 2^3 + 3 \times (2^3)^2 + \dots + 3 \times (2^3)^m \\ &= 3 \times \left[\frac{1 \times [1 - (2^3)^{m+1}]}{1 - 2^3} \right] \end{aligned}$$

若有一位剩余, 则只要再发送 2^{n-1} 次搜索即可得到全部的标签 ID。若有两位剩余, 则每保存的一个前缀都需加上 3 次搜索指令才可识别出全部标签的 ID(运用 ii 的方法), 因此需要加上 $3 \times 2^{n-1}$ 次。

iv. 当按照 4bit 分维时, 首先识别第一维, 发送第一次 SearID, 得到第一维维码数, 维码数 0 或 4 可以直接推断其维 ID; 发送 SearID(1, 1), 根据碰撞位推测出第一维维码数为 1 的标签维 ID; 发送 SearID(1, 3), 根据碰撞位可推算出第一维维码数为 2 的标签维 ID, 假设标签有 2^n 个, 此时碰撞位为????, 只有一种可能, 因此可推测出 0111、1011、1101 和 1110 4 种 ID 都存在; 发送 SearID(1, 2), 产生碰撞, 需要再次发送 SearID(1, 2, "0"), 识别出最高位为 0 的标签, 用相同的方法识别出最高位为 1 的标签。因此, 识别第一维总共的搜索次数为 6 次。识别第二维时, 在第一维保存了 2^4 个前缀, 因此要执行 6×2^4 次... 识别第 m 维时, 需要的搜索次数为 6×2^{4m} 。因此, 总的搜索次数为:

$$\begin{aligned} N &= 6 + 6 \times 2^4 + 6 \times (2^4)^2 + \dots + 6 \times 2^{4m} \\ &= 6 \times \left[\frac{1 \times [1 - (2^4)^{m+1}]}{1 - 2^4} \right] \\ &= 6 \times \left(\frac{2^{4m} - 1}{2^4 - 1} \right) \end{aligned}$$

若有一位剩余, 则只要再发送 2^{n-1} 次搜索即可得到全部的标签 ID。若有两位剩余, 则每保存的一个前缀都需加上 3 次搜索指令才可识别出全部标签的 ID(运用 ii 的方法), 因此需要加上 $3 \times 2^{n-1}$ 次, 证毕。

3.2 仿真结果

本文采用 JAVA 语言编程。对查询树算法(QTA)、碰撞位检测的查询树算法(QT-CBP)^[12]、基于二进制树分解的动态防碰撞算法(BTD)^[6]进行比较。实验中取标签 ID 编码位数为 8bit 和 16bit, 标签 ID 序列随机产生, 仿真实验循环 100 次取平均值, 对 4 种算法的搜索次数进行了比较。

图 4 和图 5 分别为编码位数为 8 位和 16 位时搜索次数的比较, 三角标记线为新算法的搜索次数, 标签个数随机产生, 当编码位数为 8 位, 标签个数少于 50 时, 即标签饱和度比较低时, 4 种算法所用的搜索次数无很大区别, 当标签个数等于 100 时, 新算法所用的搜索次数为 67 次, 而其他 3 种算法的搜索次数依次为 225、184、146 次。当编码位数为 16 位时, 可以看到 BTD 和 QT-CBP 算法的搜索次数相近, QT-CBP 略优于 BTD 算法, 因为 BTD 算法检测碰撞位后按照二叉树的算法对标签进行识别, 每次增加两位前缀, 但并不是增加的前缀的 4 种排列都存在, 因此增加了不必要的搜索次数。随着标签数量的增加, 新算法的搜索次数较其他 3 种算法明显减少。通过仿真实验结果计算可得, 新算法的搜索次数分别比 QTA、BTD 和 QT-CBP 算法平均减少了 54.2%、36.9% 和 42.3%。

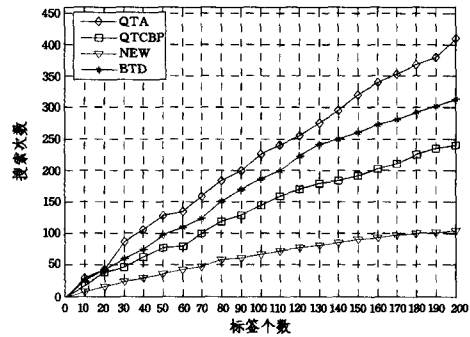


图 4 8bit 数据对比图

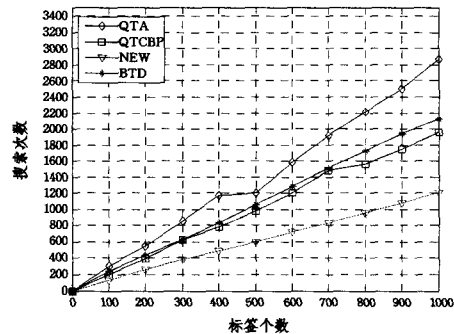


图 5 16bit 数据对比图

图 6 和图 7 分别是标签 ID 位为 8bit 和 16bit 4 种算法吞吐率的对比图, 吞吐率是衡量系统性能指标之一, 定义为单

位时间内成功识别的标签个数。当标签编码位数为 8bit 时, QTA、QT-CBP 和 BT-D 算法在标签个数较少时吞吐率相近, 新算法在标签饱和度较低时, 能保持较高的吞吐率。随着标签饱和度升高, 新算法较其他 3 种算法的优势明显, 因为新算法根据堆栈顺序, 维码数依次出栈, 减少了不必要的搜索时隙。当编码位数相同时, 新算法较 QT-CBP 和 BT-D 算法, 吞吐率平均增加了 59.3%, 78.8%。

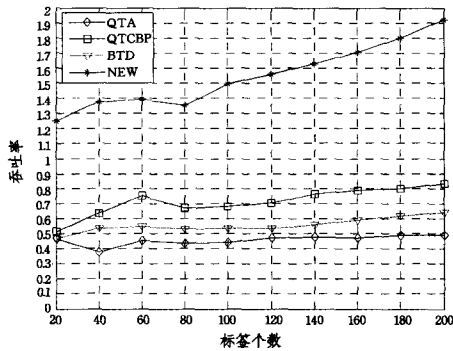


图 6 8bit 吞吐率对比图

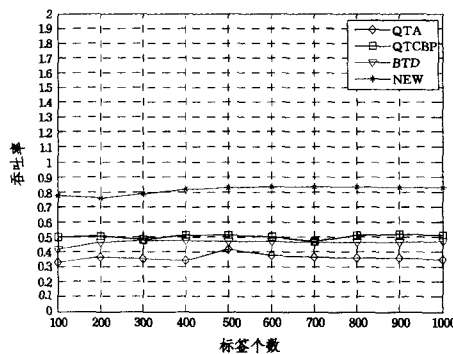


图 7 16bit 吞吐率对比图

结束语 本文结合智能分离的增强型时隙二进制算法 (IS-ESBT) 和自适应维分编码的思想, 提出了一种基于维码数的自适应混合防碰撞算法。新算法较传统算法有以下几点优势: 1) 对可读范围内的标签进行逻辑分组, 自适应选择分维

位数, 实现标签的有序识别; 2) 维码数堆栈阶段, 只回复其维码数, 减少了阅读器和标签之间的通信量; 3) 根据堆栈的维码数, 采用不同的策略推算其维 ID, 减少了空白时隙的产生。仿真实验证明, 改进算法具有较大的优越性。

参考文献

- [1] 黄玉兰. 射频识别(RFID)核心技术详解[M]. 北京: 人民邮电出版社, 2010: 2-5
- [2] 康东, 石喜勤, 李勇鹏, 等. 射频识别(RFID)核心技术与典型应用开发实例[M]. 北京: 人民邮电出版社, 2008: 165-178
- [3] 陈颖. 一种新的多阅读器防碰撞算法的研究[J]. 杭州电子科技大学学报, 2013, 32(5): 112-115
- [4] 石封茶, 崔琛, 余剑. 基于标签运动的一种新型 RFID 防碰撞算法[J]. 计算机科学, 2013, 40(6): 76-79
- [5] Abramson N. The ALOHA System-Another Alternative for Computer Communications[J]. Fall Joint Computer Conference, AFIPS Conference Proceedings, 1970, 37: 281-285
- [6] 萧耀友, 胡钢, 魏钦伟, 等. 基于二进制树分解的动态防碰撞算法[J]. 通信技术, 2011, 44(1): 99-101
- [7] 郑嘉利, 覃团发, 倪光南. Tree-based backoff protocol for fast RFID tag identification[J]. 中国邮电高校学报 (EI 源期刊), 2013, 20(2): 37-41
- [8] 宋瑞玲, 高仲合. RFID 防碰撞算法研究[J/OL]. 计算机工程与应用, <http://www.cnki.net/kcms/doi/10.3778/j.issn.1002-8331.1309-0084.html>, 2014-02-13
- [9] 韦冬雪, 郑嘉利, 李亮亮, 等. 一种新颖的自适应多叉树防碰撞算法的研究[J]. 计算机科学, 2013, 40(10): 52-55
- [10] 周信, 刘晔. 一种基于码距反演的 RFID 防碰撞算法[J]. 计算机工程与应用, 2012, 48(8): 214-217
- [11] 李致金, 周杰, 乔杰, 等. 自适应维分编码 RFID 防碰撞算法研究及优化[J]. 通信学报, 2013, 34(9): 185-190
- [12] Lee H, Kim J. QT-CBP: A new RFID tag anti-collision algorithm using collision bit positioning[M]. Emerging Directions in Embedded and Ubiquitous Computing, Springer Berlin Heidelberg, 2006: 591-600

(上接第 6 页)

其提高的比例达到了 60%, 持平的比例仅为 40%, 没有 ΔF 降低的现象出现, 处理器的利用率在 0.2~0.3 时 ΔF 提高的比率最大。图 6 中得到提高的比例达到了 50%, 持平的比例为 50%, 最大提高幅度出现在图 5 中。相比而言, SCP 策略比 ED/FR/S 策略提高得更多一些。

结束语 本文的主要贡献是, 针对突发性故障容错策略所存在的缺陷, 提出可以改变容错任务优先级策略 SCP 和 MCP, 分析计算了采用新策略下实时任务的最坏情况响应时间, 经过研究和实验, 证明了这两种策略能够有效地提高实时系统在突发性故障模式下的容错能力。

参考文献

- [1] Burns A, Davis R. I, Punnekkat S. Feasibility analysis of fault-tolerant real-time task sets[C]//Proceedings of the Eighth Euro-micro Workshop Real-Time Systems, L'Aquila, 1996: 29-33
- [2] Many F, Doose D. Scheduling analysis under fault bursts[C]//2011 17th IEEE Real-time and Embedded Technology and Ap-

plications Symposium (RTAS), Chicago, IL, 2011: 113-122

- [3] Liu C L, Lavland J W. Scheduling algorithm for multiprogramming in a hard real-time environment[J]. Journal of ACM, 1973, 20(1): 40-61
- [4] Audsley N C, et al. Hard Real-Time Scheduling: The Deadline Monotonic Approach[C]//Proceedings of Eighth IEEE Workshop on Real-time Operating Systems and Software, Atlanta, CA, USA, 1991: 133-137
- [5] 邓建波, 张立臣, 邓惠敏. 异构分布式系统混合型实时容错调度算法[J]. 计算机科学, 2011, 38(3): 87-92
- [6] 郭锐锋, 刘娟, 丁万夫, 等. 回卷恢复模型下容错实时系统的可调度性分析[J]. 计算机科学, 2013, 34(6): 1334-1338
- [7] Joseph M, Pandya P. Finding response times in a real-time system[J]. The Computer Journal, 1986, 29(5): 390-395
- [8] 李俊, 阳富民, 卢炎生. 一种可行的容错实时系统可调度性分析[J]. 软件学报, 2005, 16(8): 1513-1522
- [9] Lima G M A, Burns A. An optimal fixed-priority assignment algorithm for supporting fault-tolerant hard real-time systems[J]. IEEE Transactions on Computers, 2003, 52(10): 1332-1346