

一种基于冲突跟踪的 RFID 防冲突算法

熊 伟¹, 滕培俊¹, 梁 青²

(1. 空军工程大学 电讯工程学院, 陕西 西安 710077; 2. 西安邮电学院 电子与信息工程系, 陕西 西安 710061)

摘 要:在射频识别系统中,防冲突技术是 RFID 技术必须解决的关键问题之一。针对查询树算法因逐位增加查询前缀而增加了阅读器时间复杂度和通信复杂度这一问题,给出了一种利用冲突位来更新查询前缀的改进算法——冲突跟踪树算法。通过构建冲突跟踪树,从理论上分析了改进算法的通信复杂度。在冲突位是连续的假设前提下,给出了计算阅读器通信复杂度的 3 个引理和一个定理,并对其做了详细证明。最后利用 Matlab 对相关算法性能进行了对比仿真。结果表明,在冲突位是连续的前提下,冲突跟踪树算法阅读器的通信复杂度优于查询树算法阅读器的通信复杂度,从而得出在一般情况下,冲突跟踪树算法更能有效改善阅读器的通信复杂度,节省系统的开销。

关键词:无线射频识别;防冲突;冲突跟踪树;通信复杂度

DOI:10.3969/j.issn.1009-3516.2009.03.015

中图分类号: TN92 **文献标识码:** A **文章编号:** 1009-3516(2009)03-0068-05

无线射频识别(Radio Frequency Identification, RFID)技术是一种非接触的自动识别技术,其基本原理是利用射频信号的空间耦合(电感或电磁耦合)传输特性,实现对被识别物体的自动识别^[1]。RFID 技术不仅在民用上应用十分广泛有着良好的应用前景。在 RFID 系统中,如何快速有效地识别多个标签,即防冲突问题,是 RFID 技术必须解决的关键问题之一。当前超高频段的防冲突算法^[2]主要采用二进制树算法,可分为二进制搜索算法、动态二进制搜索算法^[1,3-5]和查询树算法(Query Tree, QT)^[6-8]等。

本文在研究查询树算法的基础上,提出了一种根据冲突位来增加查询前缀的算法,即冲突跟踪树型算法(Collision Tracking Tree, CTT),该算法有效避免了查询树算法逐位增加查询前缀的弊端。与查询树算法相比,冲突跟踪树型算法能明显减少阅读器查询次数和通信复杂度,降低阅读器的数据传输量。

1 查询树算法

查询树算法是一种无记忆的算法,标签不必存储以前的查询信息,对标签的功能要求较低,因此标签成本可以降低。该算法包括一系列的查询和应答,在每一轮中,阅读器向标签发出查询命令(一个 k 位前缀),询问标签是否含有该特定的前缀,如果有多于一个标签应答,则阅读器能检测出冲突,则在已有的前缀后增加一个 0 或 1,形成一个新的前缀继续询问。当只有一个标签响应查询时,该标签就被识别。

查询树算法主要从时间复杂度和通信复杂度这两个性能指标来衡量^[9-10]。所谓时间复杂度就是阅读器完全识别 n 个标签所需的查询次数。用 T_s 表示查询树算法中的识别次数,当标签个数 $n \geq 4$ 时,文献[9]提出统计关系式:

$$2.881n - 1 \leq E[T_s] \leq 2.887n - 1 \quad (1)$$

式中 $E[T_s]$ 表示识别次数的数学期望。在最坏的情况下,识别 n 个标签所需的查询次数为:

* 收稿日期:2008-10-27

基金项目:陕西省自然科学基金资助项目(2006F13)

作者简介:熊 伟(1965-),男,安徽凤阳人,副教授,主要从事 RFID 和 EDA 的研究. E-mail:peijun0501@126.com

$$N \leq n(L + 2 - \log_2 n) \tag{2}$$

式中 L 为标签 ID 长度。

阅读器的通信复杂度是指识别 n 个标签时阅读器所发送的总比特数。由于标签 ID 长度为 L , 故阅读器每次发送的比特数不超过 L bit, 而 $E[T_s] \leq 2.887n - 1$ 。因此, 阅读器的通信复杂度有近似关系式^[9]:

$$L_r \leq 2.89kn \tag{3}$$

式中 L_r 为阅读器发送的比特数。

2 冲突跟踪树算法

2.1 算法的核心思想

通过对查询树算法的研究可知, 阅读器向标签广播一个查询前缀, 当有多个标签响应时, 响应的标签将各自剩余的 ID 传送给阅读器, 假设各个标签发送的 ID 中前 $n-1$ 位都相同, 第 n 位冲突, 对于 QT 算法, 阅读器至少要再经过 $n-1$ 位重复查询, 才能检测到第 n 位冲突。针对该算法查询前缀每次只增加 1 位 (0 或 1) 的缺陷, 本文采用冲突跟踪的方法, 仅当阅读器检测到冲突位时才更新查询前缀, 更新的查询前缀为当前的查询前缀加上新接收到的冲突位以前的比特。用公式表示就是: “新前缀” = “原前缀” + “接收到的比特 ($k+1$ 至 $n-1$)” + “0/1”, 其中, k 表示上一次冲突位置。冲突跟踪树型算法的流程见图 1。

2.2 冲突跟踪树的构建

构建冲突跟踪树以便于对冲突跟踪树算法过程的理解。除根节点外, 冲突跟踪树的每一个节点表示阅读器要发送的查询前缀, 树的高度 k 表示标签发生冲突的位置, 例如 $k=2$, 表明标签第 2 位产生冲突, 高度为 k 的节点的每一条边表示阅读器发送的比特数, 权值为 k bit。树的一个叶节点表示一个标签, 或者表示一个无响应节点 (只在第 1 次查询时出现)。以 ID 长度为 8 的标签集 {10110010, 10111010, 10010010, 00110010, 00010010} 为例, 所构建的冲突跟踪树见图 2。第 1 次查询, 所有标签的第 1 位发生冲突, 于是在高度为 1 处分成“0”枝和“1”枝; 对于“0”枝, 标签在第 3 位发生冲突, 于是在高度为 3 处分成“000”和“001”两枝, 结果是“00010010”和“00110010”标签被识别。同样, 对于“1”枝, 在高度为 3 处分成“100”和“101”两枝, 此时“10010010”标签被识别。对于“101”枝, 标签在第 5 位发生冲突, 于是在高度为 5 处分成“10110”和“10111”两枝, 剩余的 2 个标签被识别。

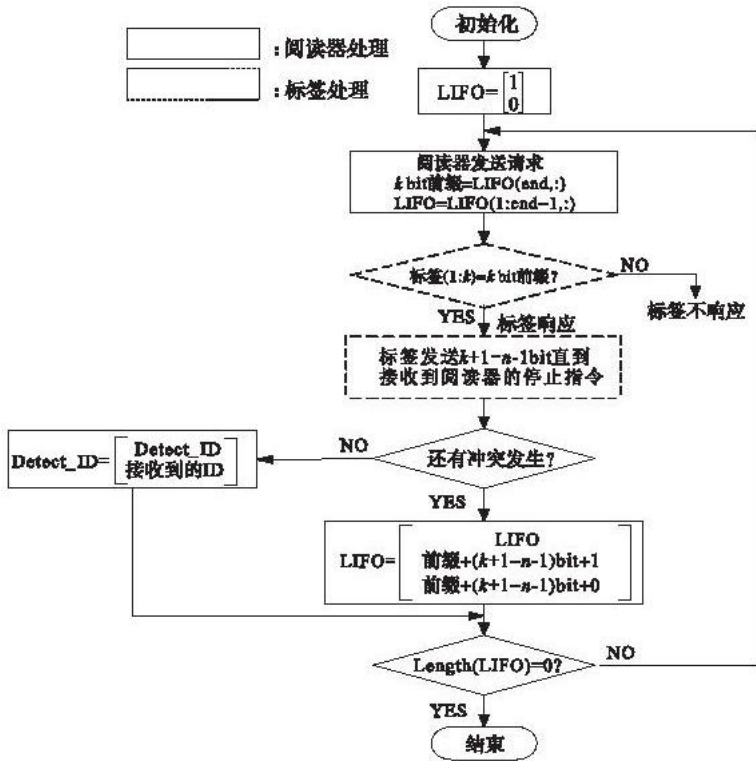


图 1 CTT 算法流程图

Fig. 1 The flow chart of collision tracking tree algorithm

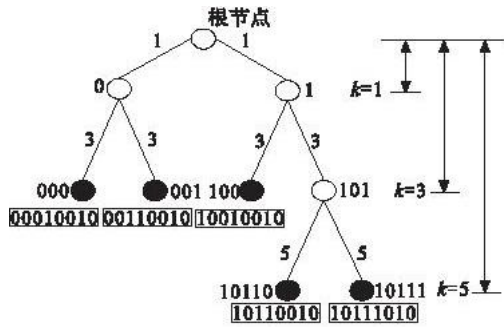


图 2 冲突跟踪树

Fig. 2 Collision tracking tree

3 算法通信复杂度分析

令标签的最高位为第 1 位, 从图 2 中的冲突跟踪树可以看出, 高度为 k 的节点的每一条边表示的比特数为 k bit。假设冲突只发生在第 i 位, 阅读器需要发送的比特数为 $2i$ bit。因此, 标签产生冲突的位越高, 阅读器需要发送的比特数越少。

引理1 n 个标签,阅读器若要全部识别,标签至少需要有 $\lceil \log_2 n \rceil$ 位冲突。

证明 假设一个十进制数 n ,若用二进制表示需要 m 位,则有关系式: $2^m \geq n$,即 $m \geq \log_2 n$,说明一个十进制数 n ,至少需要 $\lceil \log_2 n \rceil$ 位二进制表示。可见, n 个标签,其ID至少有 $\lceil \log_2 n \rceil$ 位不同,即至少有 $\lceil \log_2 n \rceil$ 位冲突。

引理2 一棵高度为 k 的完全二进制树,阅读器的通信复杂度为:

$$L = \sum_{i=1}^k i2^i \quad (4)$$

证明 假设完全二进制树的高度为 k ,由完全二进制树的查询过程可知,第 i 层($i \leq k$)的节点个数为 2^i ,第 i 层的边数也为 2^i 。因此,在第 i 层阅读器发送的比特数为 $i2^i$,由于阅读器的通信复杂度为每一层发送的比特数之和。因此一棵高度为 k 的完全二进制树的通信复杂度为:

$$L = 1 \times 2 + 2 \times 2^2 + \dots + k \times 2^k = \sum_{i=1}^k i2^i \quad (5)$$

引理3 n 个标签,阅读器若要全部识别,需要的查询次数为: $N = 2n$ 。

证明 从图2所描述的冲突跟踪树可知,除根节点外,冲突跟踪树的节点数表示阅读器的查询次数。由于系统的初始查询都是从“0”和“1”开始,因此任意冲突跟踪树都有“0”和“1”2个节点,且冲突跟踪树只在最高冲突位分成左右2个节点。因此,当标签数 $n = 2$ 时,冲突跟踪树的节点个数为: $2 + 2 = 2(2 - 1) + 2$;若增加1个标签,意味着系统增加了1次冲突,冲突跟踪树相应增加2个节点,节点个数变为: $2 \times 2 + 2 = 2(3 - 1) + 2$ 。依此类推, n 个标签,冲突跟踪树的节点数为: $N = 2(n - 1) + 2 = 2n$,即查询次数为 $N = 2n$ 。若标签第1位就发生冲突,则查询次数修正为: $N = 2(n - 1)$ 。

定理 对 n 个标签,假设第1次冲突发生在第 k 位($k \geq 2$),且冲突位是连续的,则阅读器的通信复杂度为:

$$L_r = \sum_{i=k}^{k+\lceil \log_2 n \rceil - 1} i2^{i-k+1} + \lfloor k + \log_2 n \rfloor \left(2n - 2 - \sum_{i=1}^{\lfloor \log_2 n \rfloor - 1} 2^i \right) + 2 \quad (6)$$

证明 由引理1可知, n 个标签至少有 $\lceil \log_2 n \rceil$ 位冲突位。由于冲突位是连续的,因此标签从第 k 位到第 $k + \log_2 n$ 位为冲突位。根据假设条件,可以将阅读器的查询过程分成3部分:首先,阅读器的查询都是从“0”、“1”开始,即初始查询阅读器需要发送2 bit;其次,由于冲突位是连续的,因此第 k 位到第 $k + \lceil \log_2 n \rceil - 1$ 位构成一棵完全冲突跟踪树 T_1 。由于第1次冲突发生在第 k 位,则该节点处的每条边的权值为 k bit。根据引理2, T_1 发送的比特数为:

$$L'_r = 2^1 k + 2^2 (k + 1) + \dots + 2^{\lfloor \log_2 n \rfloor} (k - 1 + \lceil \log_2 n \rceil) = \sum_{i=k}^{k+\lceil \log_2 n \rceil - 1} i2^{i-k+1} \quad (7)$$

由引理3可知,系统总的查询次数为 $N = 2n$,初始查询次数为2,并且高度为 $\lceil \log_2 n \rceil$ 的完全冲突跟踪树的查询次数为 $\sum_{i=1}^{\lfloor \log_2 n \rfloor} 2^i$ 。因此,对第 $k + \log_2 n$ 位冲突,需要的查询次数为: $2n - 2 - \sum_{i=1}^{\lfloor \log_2 n \rfloor} 2^i$,针对该位阅读器发送的比特数为:

$$L''_r = \lfloor k + \log_2 n \rfloor \left(2n - 2 - \sum_{i=1}^{\lfloor \log_2 n \rfloor} 2^i \right) \quad (8)$$

综合上述3部分内容,阅读器的通信复杂度为:

$$L_r = L'_r + L''_r + 2 = \sum_{i=k}^{k+\lceil \log_2 n \rceil - 1} i2^{i-k+1} + \lfloor k + \log_2 n \rfloor \left(2n - 2 - \sum_{i=1}^{\lfloor \log_2 n \rfloor} 2^i \right) + 2 \quad (9)$$

当 $k = 1$ 时,阅读器的通信复杂度为:

$$L_r = \sum_{i=1}^{\lfloor \log_2 n \rfloor} i2^i + \lfloor 1 + \log_2 n \rfloor \left(2n - 2 - \sum_{i=1}^{\lfloor \log_2 n \rfloor} 2^i \right) \quad (10)$$

在相同条件下,同理可推导出查询树算法的阅读器通信复杂度为:

$$L_q = k(k - 1) + \sum_{i=k}^{k+\lceil \log_2 n \rceil - 1} i2^{i-k+1} + \lfloor k + \log_2 n \rfloor \left(2n - 2 - \sum_{i=1}^{\lfloor \log_2 n \rfloor} 2^i \right) \quad (11)$$

4 仿真结果

由于 EPC 标准第 2 代第 1 类中规定 RFID 标签的 ID 长度为 96 bit,因此仿真过程中假定标签的 ID 长度 $L=96$ bit;此外,假设所有标签的冲突位是连续的。阅读器通信复杂度的仿真结果见图 3。图 3(a)是第 1 次冲突发生的位置与阅读器检测到第 1 次冲突时发送的比特数的关系曲线。可见,当第 1 次冲突发生在第 1 位时,两种算法阅读器发送的比特数相同,但当第 1 次冲突发生位置为 ($k \geq 2$) 时,CTT 算法阅读发送的比特数少于 QT 算法阅读器发送的比特数。如当 $k=20$ 时,QT 算法阅读器发送的比特数为 380 bit,CTT 算法阅读器发送的比特数为 40 bit,明显少于 QT 算法。图 3(b)为当标签个数 $n=50$,且冲突位置是连续的前提下,第 1 次冲突位置与阅读器的通信复杂度之间的关系曲线。当 $k=15$ 时,QT 算法的阅读器通信复杂度为 1 982 bit,CTT 算法的阅读器通信复杂度为 1 872 bit,相差 210 bit;当 $k=25$ 时,QT 算法和 CTT 算法的阅读器通信复杂度分别为 3 426 bit 和 2 826 bit,二者差 600 bit。

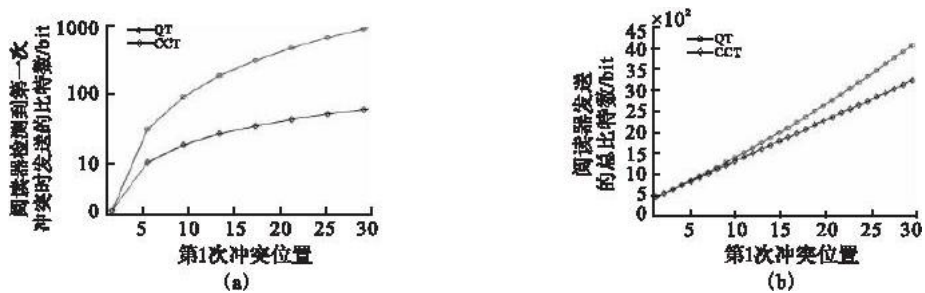


图3 阅读器通信复杂度分析

Fig. 3 Analyze of reader communication complexity

由此可见,当发生冲突的位置 k 越大,优越性体现得越明显。所以,CTT 算法的阅读器通信复杂度比 QT 算法的阅读器通信复杂度优越。在更一般的情况下,当冲突位置不连续时,CTT 算法的优越性将体现得更为明显。

5 结束语

基于二进制树的防冲突算法已经应用于 EPC 标准中,各种改进的二进制冲突算法也有广泛的应用。本文提出一种基于冲突跟踪的 RFID 防冲突算法,其核心在于阅读器根据标签的冲突位来构建冲突跟踪树,阅读器接收标签发送的数据,直至检测到冲突,根据冲突位调整查询前缀,重复以上查询过程,直至完全识别工作范围内的所有标签。文章通过对冲突跟踪树算法性能的理论分析及 Matlab 仿真,结果表明,冲突跟踪树算法在阅读器通信复杂度上优于查询树算法。

参考文献:

- [1] Finkenzerler K. RFID Handbook: Fundamentals and Application in Contact-less Smart Card and Identification (Second Edition) [M]. England: John Wiley and Sons, 2003.
- [2] Auto-ID Center. 860 MHz-930 MHz Class I Radio Frequency Identification Tag Radio Frequency and Logical Communication Interface Specification Candidate Recommendation, Version 1.0. 1 [R]. Technical Report MIT-AUTOID-TR-007, 2002.
- [3] 杜海涛, 徐昆良, 王威廉. 基于返回式二进制树形搜索的反碰撞算法 [J]. 云南大学学报: 自然科学版, 2006, 28(SI): 133-136.
DU Haitao, XU Kunliang, WANG Weilian. An Anti-collision Algorithm Based on Binary Tree Searching of Backtracking [J]. Journal of Yunnan University, 2006, 28(SI): 133-136. (in Chinese)
- [4] 冯波, 李锦涛, 郑为民, 等. 一种新的 RFID 标签识别防冲突算法 [J]. 自动化学报, 2008, 34(6): 632-638.
FENG Bo, LI Jintao, ZHENG Weimin, et al. A Novel Anti-collision Algorithm for Tag Identification in RFID Systems [J]. Acta Automatic & Sinica, 2008, 34(6): 632-638. (in Chinese)
- [5] 王彤, 晁爱农, 张红梅. 一种动态 Huffman 优化算法的设计与应用 [J]. 空军工程大学学报: 自然科学版, 2005, 6(2):

76 – 78.

WANG Tong, CHAO Ainong, ZHANG Hongmei. A Design and Application on Dynamic Huffman Optimization Algorithm [J]. Journal of Air Force Engineering University: Natural Science Edition, 2005, 6(2): 76 – 78. (in Chinese)

- [6] Choi Ji Hwan, Lee Dongwook, Lee Hyuckjae. Query Tree – Based Reservation for Efficient RFID Tag Anti – Collision [J]. IEEE Commun Lett, 2007, 11(1): 85 – 87.
- [7] Zhou F, Jin D, Huang C, et al. Optimize the Power Consumption of Passive Electronic Tags for Anti – collision Schemes [C]// Proc the 5th Inter conf on ASIC. New Jersey: IEEE, 2003: 1213 – 1217.
- [8] Choi Ji Hwan, Lee Dongwook, Jeon Hyongsuk. Enhanced Binary Search with Time – Divided Responses for Efficient RFID Tag Anti – Collision [C]// IEEE Communications Society Subject Matter Experts for Publication in the ICC 2007 Proceedings. New Jersey: IEEE, 2007: 3853 – 3858.
- [9] Law C, Lee K, Siu K. Efficient Memory Less Protocol for Tag Identification [C]// Proceedings of the ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. Manchester: ACM Press, 2000: 75 – 84.
- [10] Myung Jihoon, Lee Wonjun, Srivastava Jaideep. Adaptive Binary Splitting for Efficient RFID Tag Anti – Collision [J]. IEEE Communications Letters, 2006, 10(3): 144 – 146.

(编辑: 徐楠楠)

Research of an Anti – Collision Algorithm Based on Collision Tracking Of RFID System

XIONG Wei¹, TENG Pei – jun¹, LIANG Qing²

(1. Telecommunication Engineering Institute, Air Force Engineering University, Xi'an 710077, China; 2. Department of Electronics and Information Engineering, Xi'an Institute of Posts and Telecommunications, Xi'an 710061, China)

Abstract: Anti – Collision is one of the key problems in RFID system. Aiming at the deficiency of Query Tree Algorithm that the complexities of query and communication are increased due to adding prefix seriatim, an improved algorithm called Collision Tracking Tree Algorithm is proposed, which updates query prefix using collision position. By designing the collision position tree, the communication complexity of this improved algorithm is analyzed theoretically. On the assumption that the collision bits are continuous, three lemmas and a theorem about the reader communication complexity are presented and proved in detail. Finally, the performance of the algorithm is simulated by using MATLAB. The simulation results show that the reader communication complexity of collision tracking tree algorithm is better than the query tree algorithm based on the assumption that the collision bits are continuous. So, in the normal instance using the proposed algorithm can reduce the complexity of communication and save costs of the system effectively.

Key words: RFID; anti – collision; collision tracking tree; communication complexity