

改进型动态时隙冲突跟踪树标签防碰撞算法

黄以华, 陈小若

(中山大学电子与通信工程系, 广东广州 510006)

摘 要: 射频识别 RFID(Radio Frequency Identification)系统内的读卡器在识别大数量标签时常因信号的碰撞而导致系统识别效率降低. 动态时隙冲突跟踪树算法 DSCTTA(Dynamic Slots Collision Tracking Tree Algorithm)采用动态时隙应答机制可减少前缀开销和迭代开销, 但会产生大量空闲时隙. 本文将 DSCTTA 和比特转换方式(BCM)相结合, 得到改进型动态时隙冲突跟踪树标签防碰撞算法(IDSCTTA)以防止标签冲突, 加快标签识别速度. 理论分析及仿真结果表明, IDSCTTA 不仅具有 DSCTTA 的全部优点, 而且能够有效地减小识别时延和提高时隙效率, 并且标签数目越大, 算法性能越优越.

关键词: 射频识别; 防碰撞; 冲突跟踪树; 空闲时隙; 比特转换方法

中图分类号: TN918.7 **文献标识码:** A **文章编号:** 0372-2112 (2015)08-1465-06

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2015.08.001

Improved Dynamic Slots Collision Tracking Tree for RFID Anti-Collision Algorithm

HUANG Yi-hua, CHEN Xiao-ruo

(Department of Electronic and Communication Engineering, Sun Yat-Sen University, Guangzhou, Guangdong 510006, China)

Abstract: The identification efficiency of a RFID system decreases when a reader attempts to identify a large number of tags, because a collision occurs. Dynamic slots collision tracking tree algorithm (DSCTTA) reduces the prefix and iteration overhead by using time-divided responding scheme, meanwhile a lot of idle time slots are generated. In this paper, an improved dynamic slots collision tracking tree anti-collision algorithm (IDSCTTA) is proposed to prevent the tag collision by combining DSCTTA with bits change method (BCM). IDSCTTA not only has all the advantages of DSCTTA, but also can suppress the generation of idle slots therefore improve the system throughput and reduce the identification delay. Simulation results show that, IDSCTTA can effectively reduce the identification delay and improve the slot efficiency. And the greater the number of tags is, the better the algorithm performance is.

Key words: RFID(radio frequency identification); anti-collision; collision tracking tree; idle slot; bits change method

1 引言

射频识别(RFID)系统中由于读卡器和标签之间的通信需共享无线信道, 因此存在碰撞. 碰撞类型可分为读卡器碰撞和标签碰撞. 这些碰撞不仅导致标签无法识别, 还会增加读卡器识别标签的开销, 降低系统识别效率. 因此, 研究高效的防碰撞算法以提高系统识别效率是必不可少的.

目前标签防碰撞算法主要分为基于 Aloha^[1~6]的随机性算法, 如帧时隙 Aloha^[1,2] 以及动态帧时隙 Aloha^[3~6]等, 以及基于树^[7~18]的确定性算法, 比如查询树

算法(Query Tree Algorithm, QTA)^[11,12], 冲突跟踪树算法(Collision Tracking Tree Algorithm, CTTA)^[13,14]以及双时隙冲突跟踪树算法(Bi-slotted Collision Tracking Tree Algorithm, BSCTTA)^[14]等. 与基于 Aloha 的算法相比, 基于树的算法不致出现标签饥饿问题, 可保证 100% 的识别率, 但识别延迟相对较长, 因此学者们提出了一系列改进算法, 主要有: 4-ary Query Tree 算法^[15], 可减少碰撞发生的几率, 剔除大量不必要的查询, 然而该算法也引入了大量空闲时隙; Improved 4-ary QTA(I4QTA)^[16]算法, 在 4QTA 的基础上采用比特转换方法(Bits Change Method)来检测那些导致碰撞的标签 ID(存在的), 进而剔除空

闲时隙,减少不必要的查询. Liang 等人基于 BSCITTA 提出了动态时隙冲突跟踪树算法 (Dynamic Slots Collision Tracking Tree Algorithm, DSCITTA)^[17], 可减少大量不必要的查询,但同时引入了大量空闲时隙.

为了抑制空闲时隙的产生,本文提出了改进型动态时隙冲突跟踪树算法 (Improved-DSCITTA). 该算法在 DSCITTA 的基础上,为导致碰撞的标签提供一次额外的 BCM 序列应答机会. IDSCTTA 通过对跟踪到的连续碰撞比特进行 BCM 处理后再次应答,检测出实际存在的 ID,进而剔除空闲时隙. 因此,IDSCTTA 具有更快的识别速率和时隙效率.

2 改进型动态时隙冲突跟踪树技术

本文在 DSCITTA 的基础上引入了比特转换方法^[16]. 在每一轮查询过程中,IDSCTTA 对标签的应答比特进行跟踪. 若冲突比特数大于 2,IDSCTTA 则为这些标签额外提供一次应答 BCM 序列的机会. 而发生碰撞的标签则对读卡器跟踪到的连续比特进行 BCM^[16] 转换,然后在其额外应答时隙中将 BCM 序列应答给读卡器. 通过检测 BCM 序列,IDSCTTA 即可剔除空闲时隙.

2.1 改进型动态时隙冲突跟踪树算法

类似 DSCITTA 算法,IDSCTTA 算法也需要保存两个数据:查询序列和碰撞状态序列(后面用 Cs 表示),分别存入堆栈 PS 和 CS,区别在于 Cs 序列末位为标志位(1 表示 Cs 序列为 BCM 序列;0 则表示为 C 的二进制码,C < N, N 为算法最大的冲突跟踪比特数),同时增加一个替补队列 CQ,以保存需再次响应的新查询序列. M 存储已识别标签 ID. D_C 为各应答时隙对应的 C 比特 ID.

(1) 读卡器

查询:读卡器弹出 PS 和 CS 堆栈顶部的数据,向标签发送查询命令(含 $q_1q_2 \cdots q_k$, 碰撞状态序列),初始化自身时隙计数器,等待标签应答.

决策:当前时隙计数器是否在 T(Cs 序列的标志位为 0 时为 2^C , 否则为 Cs 中 1 的个数,标志位除外)内:

(I) 若计数器在 T 内,则读卡器将计数器加 1,等待标签应答序列 $r_{k+C+1}r_{k+C+2} \cdots r_{ACK}$,并检测连续碰撞的位数 C',记录首个碰撞位 r_F .

(a) 无碰撞:读卡器在接收到最后一个比特后向标签发送 ACK 信号. 此时只有一个标签在该时隙内应答,读卡器将结合查询序列、D_C 以及应答序列就可知道该标签的 ID,并将 $q_1q_2 \cdots q_k | D_C | r_{k+C+1}r_{k+C+2} \cdots r_{ACK}$ 存入 M 中并识别该标签. 然后等待下一时隙到来.

(b) $C' < 2$: 读卡器在接收到连续碰撞位后的首个非碰撞位时立刻向标签发送 ACK 信号,将 $q_1q_2 \cdots q_k | D_C | r_{k+C+1}r_{k+C+2} \cdots r_{F-1}$ 存入堆栈 PS, C 转换成二进制码后加

上标志位存入堆栈 CS. 注意,其中若首个碰撞位 r_F 为标签 ID 最末位时,读卡器收到末位后向标签发送 ACK 信号, r_F 设置为 0 和 1,将 $q_1q_2 \cdots q_k | D_C | r_{k+C+1}r_{k+C+2} \cdots r_F$ 存入 M 中,识别这两个标签,然后等待下一时隙到来.

(c) $C' \geq 2$: 读卡器在收到连续碰撞位后的首个非碰撞位或第 N 位碰撞时立刻向标签发送含 r_F 、C' 以及碰撞时隙数(与存入 CQ 的序列个数相同)的 ACK 信号. 同时将 $q_1q_2 \cdots q_k | D_C | r_{k+C+1}r_{k+C+2} \cdots r_{F-1}$ 存入替补队列 CQ. 然后等待下一时隙标签到来.

(II) 若时隙计数器超出了 T,读卡器则判别替补队列是否为空.

(a) 若 CQ 不为空,读卡器则弹出 CQ 第一个序列(设为 STR),并等待标签 BCM^[16] 序列的再次应答. 收到 BCM 后,读卡器将碰撞比特用 1 替换并在末尾加上比特 1 后存入 CS,STR 存入 PS. 注意,若碰撞达到最末位,读卡器可根据收到的 BCM 并结合 STR 识别出相应的标签 ID,存入 M,就不需要将 STR, BCM 序列分别存入 PS 和 CS. 然后等待下一时隙标签到来.

(b) 若 CQ 为空,则读卡器开启新的查询.

读卡器反复执行以上过程,直到 PS 和 CS 都为空.

(2) 标签

等待:标签处于等待状态,直到收到读卡器的查询比特.

决策:若标签 ID 的前 k 位与 $q_1q_2 \cdots q_k$ 相同,根据 Cs 设定自身应答时隙,若 Cs 标志位为 0,则按照第 k 位后的 C 比特 ID 在 $0 \sim 2^C$ 中的位置(从小到大)设定,否则,根据 C 比特 ID 对应 Cs 中比特 1 的位置(从高位到低位,跳过比特 0)设定,然后等待应答.

(I) 时隙计数器在 T 内. 则在对应时隙开始时向读卡器发送其第 k + C 位后的剩余 ID,同时等待读卡器发来的 ACK 信号(收到立刻停发). 此时,若 ACK 内信息不含碰撞位等信息,则标签只需停发即可,否则依照 ACK 信号中包含的信息设定好自身的应答时隙(再次应答),将连续碰撞 ID 进行 BCM 处理(如表 1)并在既定

表 1 BCM 模型

连续比特数 C = 2	BCM 序列	连续比特数 C = 2	BCM 序列
00	1	000	1
01	01	001	01
10	001	010	001
11	0001	011	0001
		100	00001
		101	000001
		110	0000001
		111	00000001

的时隙应答给读卡器. 然后转入等待状态.

(II) 时隙计数器不在 T 内. 需要再次应答的标签按上一次应答过程中读卡器分配的时隙(碰撞时隙数加 1 得到)向读卡器应答 BCM^[16] 序列. 然后转入等待状态.

3 算法性能分析

查询树类算法以及其变种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. 由于 BSCTTA^[14] 和 DSCTTA^[17] 等在一次查询中有多个标签应答, 这无疑会使每一次迭代查询的响应时间变长. 故此将时隙细化, 为读卡器一次查询和标签的一次应答各分配一个时隙. 时间复杂度定义为完全识别 n 个标签所需的读卡器查询次数与标签应答次数的总和. 而算法通信复杂度则定义为识别 n 个标签时读卡器发送和接收的总比特数. 下面对 CTTA^[13,14]、DSCTTA 以及 IDSCTTA 等算法的时间复杂度和通信复杂度进行分析.

3.1 算法时间复杂度

为了简化分析, 假定 n 个标签构成的 CTT, DSCTT 等都为完全树, 即 n 个标签具有连续碰撞 ID. 现实中, 读卡器不能发送空查询序列 ξ , 但此处假设读卡器能够发送空查询序列. 树的有关性质在文献[18]中有详细证明. 关于算法的时间复杂度, 鉴于篇幅所限, 此处直接给出以下定理.

定理 1 若采用 CTTA 进行识别具有连续碰撞 ID 的 n 个标签, 首位碰撞发生在第 k 位(最高位为第 1 位), 则算法时间复杂度为 $D_{CTTA} = 4n - 2$.

定理 2 若采用 BSCTTA 进行识别具有连续碰撞 ID 的 n 个标签, 首位碰撞发生在第 k 位, 则算法时间复杂度为 $D_{BSCTTA} = 3n - 2$.

定理 3 若采用 DSCTTA(最大冲突跟踪能力 $N = 2$ 时)进行识别具有连续碰撞 ID 的 n 个标签, 首位碰撞发生在第 k 位, 则算法时间复杂度为 $D_{DSCTTA} = (5n - 2)/3$.

定理 4 若采用 IDSCTTA(最大冲突跟踪能力 $N = 2$ 时)进行识别具有连续碰撞 ID 的 n 个标签, 首位碰撞发生在第 k 位, 则算法时间复杂度为 $D_{IDSCTTA} = 2n - 1$.

3.2 算法通信复杂度

令标签最高位为第 1 位, n 个标签具有连续碰撞 ID, 即 CTT, DSCTT 等都为完全树. 设首个碰撞 ID 为第 k 位. 关于算法的通信复杂度, 鉴于本文的篇幅所限, 此处我们给出以下定理.

定理 5 对 n 个标签, 假设第一次碰撞发生在第 k 位, 且碰撞位是连续的, CTTA 的通信复杂度为

$$T_{CTTA} = \sum_{j=k}^{k+\lfloor \log_2^n \rfloor - 1} j \cdot 2^{j-k+1} + \lfloor k + \log_2^n \rfloor \cdot \left[2n - 1 - \sum_{j=0}^{\lfloor \log_2^n \rfloor} 2^j \right]$$

$$\begin{aligned} & + (n + k - 2) + \left[1 - n + \sum_{j=0}^{\lfloor \log_2^n \rfloor} 2^j \right] \\ & \cdot \left[L - k - 1 - \lfloor \log_2^n \rfloor \right] \\ & + \left[2n - 1 - \sum_{j=0}^{\lfloor \log_2^n \rfloor} 2^j \right] \cdot \left[L - \lfloor k + \log_2^n \rfloor \right] \end{aligned} \quad (1)$$

而同样条件下 QTA 的通信复杂度为:

$$T_{QTA} = (k - 1)kL + (2n - 1)L \quad (2)$$

其中, 上式右边第一部分对应标签前 $k-1$ 位的通信复杂度, 第二部分则对应后续 ID 的通信复杂度.

定理 6 对 n 个标签, 假设第一次碰撞发生在第 k 位, 且碰撞位是连续的, DSCTTA($N = \log_2 m$) 的通信复杂度为:

$$\begin{aligned} T_{DSCTTA} & = \frac{n-1}{m-1}N + k - 1 + \sum_{j=0}^{\lfloor \log_2 n \rfloor - 1} (k + j \cdot N - 1)m^j \\ & + \left[k + N \lfloor \log_m n \rfloor - 1 \right] \cdot \left[\frac{n-1}{m-1} - \sum_{j=0}^{\lfloor \log_2 n \rfloor - 1} m^j \right] \\ & + \left[\frac{m-n}{m-1} + \sum_{j=0}^{\lfloor \log_2 n \rfloor} m^j \right] \\ & \cdot \left[L - k - N \lfloor \log_m n \rfloor + 1 \right] \\ & + \left[\frac{m(n-1)}{m-1} - \sum_{j=1}^{\lfloor \log_2 n \rfloor} m^j \right] \\ & \cdot \left[L - k - N \lfloor 1 + \log_m n \rfloor + 1 \right] \end{aligned} \quad (3)$$

特别地, 当 $N = 1$ 时, 定理 6 中 DSCTTA 的通信复杂度即为 BSCTTA 的通信复杂度, 有

$$\begin{aligned} T_{BSCTTA} & = \sum_{j=0}^{\lfloor \log_2 n \rfloor - 1} (k + j - 1)2^j + \left[k + \lfloor \log_2 n \rfloor - 1 \right] \\ & \cdot \left[n - 1 - \sum_{j=0}^{\lfloor \log_2 n \rfloor - 1} 2^j \right] \\ & + \left[n + k - 2 \right] + \left[2 - n + \sum_{j=0}^{\lfloor \log_2 n \rfloor} 2^j \right] \\ & \cdot \left[L - k - \lfloor \log_2 n \rfloor + 1 \right] \\ & + \left[2n - 1 - \sum_{j=0}^{\lfloor \log_2 n \rfloor} 2^j \right] \cdot \left[L - k - \lfloor \log_2 n \rfloor \right] \end{aligned} \quad (4)$$

定理 7 对 n 个标签, 假设第一次碰撞发生在第 k 位, 且碰撞位是连续的, IDSCTTA($N = \log_2 m$) 的通信复杂度为:

$$\begin{aligned} T_{IDSCTTA} & = \frac{(n-1)}{m-1}(2m + N) + k - 1 + \sum_{j=0}^{\lfloor \log_2 n \rfloor - 1} (k + j \cdot N - 1)m^j \\ & + \left[k + N \lfloor \log_m n \rfloor - 1 \right] \cdot \left[\frac{n-1}{m-1} - \sum_{j=0}^{\lfloor \log_2 n \rfloor - 1} m^j \right] \\ & + \left[\frac{m-n}{m-1} + \sum_{j=0}^{\lfloor \log_2 n \rfloor} m^j \right] \cdot \left[L - k - N \lfloor \log_m n \rfloor + 1 \right] \\ & + \left[\frac{m(n-1)}{m-1} - \sum_{j=1}^{\lfloor \log_2 n \rfloor} m^j \right] \cdot \left[L - k - N \lfloor 1 + \log_m n \rfloor + 1 \right] \end{aligned} \quad (5)$$

4 算法仿真与分析

我们采用文献[16]和文献[17]所规定的环境对 IDSCCTA 的性能进行评估,并将其与 QTA, BSCCTA, DSCCTA, I4QTA 的性能进行对比. N 值设定为 2, 3 和 4. DSCCTA3 和 IDSCCTA3 分别表示 $N = 3$ 时 DSCCTA 和 IDSCCTA 的仿真结果,其它的类似. 仿真过程中,标签 ID 长度为 96bits.

首先,假设所有标签的 ID 碰撞位连续,各算法的通信复杂度的仿真结果见图 1. 当标签个数为 $n = 43$ ($n-1$ 必须是同时是 3 和 7 倍数才能构成完全 4 叉树和完全 8 叉树)时,且碰撞位连续时,随着首次碰撞位 k 的增大,除 CTTA 外的其它算法通信复杂度呈线性下降趋势,而 CTTA 的通信复杂度则是不断增加. 当 $k = 10$ 和 $k = 30$ 时,CTTA, BSCCTA, DSCCTA ($N = 2, 3$)以及 IDSCCTA ($N = 2, 3$)的通信复杂度分别为 4659bit, 4148bit, 3787bit, 3672bit, 3843bit, 3720bit 和 5499bit, 4088bit, 3207bit, 2932bit, 3263bit, 2980bit. 从这两组数据可以看出,CTTA 算法通信复杂度高于其它算法, BSCCTA 次之, DSCCTA 通信复杂度最小 ($N = 3$). IDSCCTA 通信复杂度始终低于 BSCCTA,但却高于 DSCCTA,而且同条件下与 DSCCTA 的差距保持恒定. 图 1 显示上述算法两两之间的差距也会随着首个碰撞位置的增加而增大(同条件下 IDSCCTA 与 DSCCTA 除外). 这是因为 BSCCTA, DSCCTA 和 IDSCCTA 都采用分时隙应答策略,使得查询次数(或比特)减少. 随着 k 的增大,每次应答的比特数会减小,而且一次查询所增加的比特数要小于多次应答所减少得比特数. QTA 的通信复杂度超过 8000bit,而且随着 k 增加而增大,因此没有在图中画出.

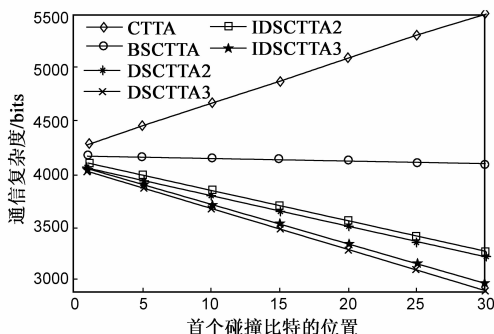


图1 算法的通信复杂度对比

显然, IDSCCTA 的通信复杂度较 DSCCTA 的要大. 然而在现实应用中,碰撞位连续这种情况非常罕见. 为不失一般性,所有标签都按一致分布随机产生 96bits ID. 然后,针对各算法的通信复杂度和时间复杂度进行仿真.

图 2 给出了识别一个 96 比特的 RFID 标签的通信

复杂度. 不难看出 IDSCCTA 与 BSCCTA 之间有着明显的差距, IDSCCTA 比 BSCCTA 要少传送大约 2 到 6 个比特, 原因是 IDSCCTA 采用多比特仲裁机制及动态时隙响应策略, 在同样采用冲突跟踪方法的情况下, IDSCCTA 省去 BSCCTA 大量的不必要查询. 而 IDSCCTA 相对于 DSCCTA 算法来说, 只增加了约 3 到 5 个比特, 这是因为 IDSCCTA 中增加了标签碰撞部分的 BCM 序列再次应答. IDSCCTA 中, $N = 3$ 时的传送开销又比 $N = 2$ 和 4 时的开销平均少约 0 到 3 比特. 同时 $N = 4$ 时算法传送开销又有增大的趋势. 然而, 由于 QTA 和 I4QTA 在每一轮查询中标签要将剩余的 ID 全部应答, 故识别一个标签平均所需要传送的比特数分别大于 240 和 160 比特, 因此这两种方法的结果没展示出来. 类似于 DSCCTA, 随着标签数目的增加, IDSCCTA 识别一标签所需传送的平均比特数呈缓慢下降趋势.

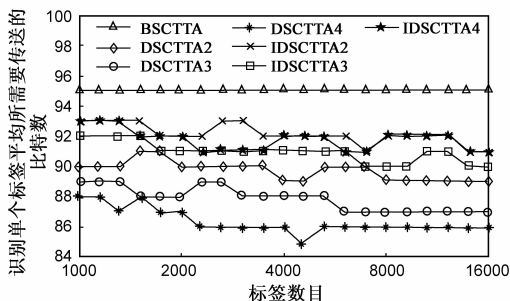


图2 识别一个96比特的RFID标签的通信复杂度

图 3 显示了 QTA, I4QTA, BSCCTA, DSCCTA 以及 IDSCCTA 识别不同数量标签所需要的时间. 总体而言, 随着标签数量的增加, 上述算法所花费的时间呈线性递增的趋势. 识别同样数量的标签, IDSCCTA 所花时间最少, 识别速率最快——IDSCCTA 识别速率至少比 QTA 快 54%, 比 I4QTA 快 35%, 而比 BSCCTA 快 12%. 然而, DSCCTA 虽然相对于 QTA 和 I4QTA 识别时间要快得多, 但是在 $N > 3$ 时识别速率却不如 BSCCTA. 相比之下, IDSCCTA 却始终快于 DSCCTA. 这是因为随着 N 的增加, 虽然 DSCCTA (IDSCCTA 也如此) 的查询迭代次数减小, 但引入的空闲应答时隙相应的增多, 所以 DSCCTA 在

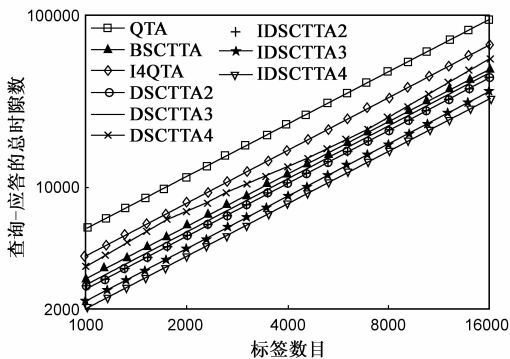


图3 识别一定数量标签(96bits)的总时隙

$N > 3$ 时性能不如 BSCITTA, 而 IDSCTTA 采用 BCM 剔除了空闲时隙, 所以比 BSCITTA 和 DSCITTA 更优.

图 4 显示了 IDSCTTA 与 DSCITTA 算法识别相同数量的标签所需识别时间的差异. 显然, 随着 N 的增大, DSCITTA 算法识别时间逐渐增大. 相反, IDSCTTA 算法识别时间却逐渐减少, 当然传送开销会有相应地增加. IDSCTTA 虽然在 $N = 2$ 时相对于 DSCITTA 性能改善并不大, 但是在 $N > 2$ 时性能却有显著的提升. IDSCTTA 在 $N = 3$ 时的时隙数比 DSCITTA 在 $N = 2$ 时 (此时 DSCITTA 算法的系统吞吐率最高) 减少了约 17%, 而在 $N = 4$ 时, 比 DSCITTA 在 $N = 2$ 时减少了约 22%. 这是因为随着 N 的增加, DSCITTA 引入的空闲时隙就越多, 这样白白浪费了大量系统的资源.

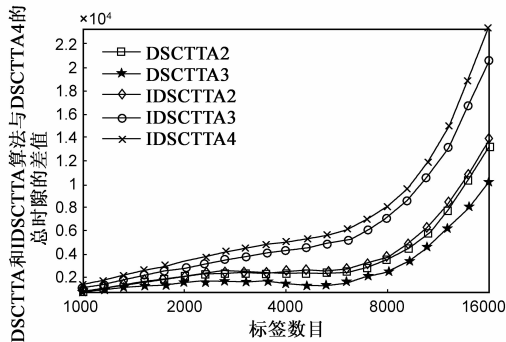


图4 IDSCTTA与DSCITTA的性能对比

图 5 给出了 QTA, DSCITTA 以及 IDSCTTA 在识别标签过程中产生的空闲时隙个数. 其中, QTA 在识别过程中产生的空闲时隙数比较大, 因为它采用单比特仲裁机制, 在识别过程中进行了大量不必要查询. 虽然 DSCITTA 采用冲突跟踪机制和动态时隙应答策略, 减少了大量不必要查询, 但同时也引入了大量空闲时隙, 并且仲裁比特数越多, 空闲时隙就越多, 甚至比 QTA 还多. 而 IDSCTTA 因为采用 BCM 策略, 剔除了 DSCITTA 中全部的空闲时隙, 因而节约了系统资源. I4QTA、BSCITTA 同 IDSCTTA 一样空闲时隙数为零 (未列出), 前者是因为采用了 BCM 方案剔除了空闲时隙, 后者是因为采用

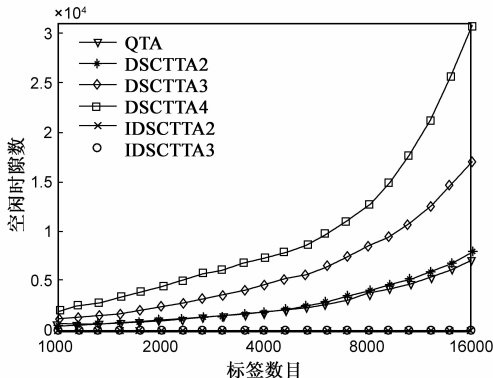


图5 IDSCTTA与DSCITTA空闲时隙个数对比

冲突跟踪和双时隙应答方法不存在引入空闲时隙的问题.

算法的时隙效率定义为可读时隙与总时隙之比. 由表 2 可以得出, 冲突跟踪树算法的时隙效率明显优于 QTA 和 I4QTA. 采用冲突跟踪技术不仅减少了通信开销, 又减少了大量的查询次数, 缩短了识别时延. 另一方面, DSCITTA ($N < 4$) 和 IDSCTTA 的时隙效率高于 BSCITTA. 因为随着 N 的增大, DSCITTA (和 IDSCTTA) 的查询迭代次数越少, 即查询次数更少, 进而算法的时间复杂度越小, 在相同条件下, 时隙效率就越高. 而当 $N \geq 4$ 时, DSCITTA 引入大量的空闲时隙, 故此时隙效率越加不如 BSCITTA. IDSCTTA 既减少了查询迭代次数, 又通过以较少的 BCM 应答时隙剔除了全部的空闲应答时隙, 故此时隙效率最高, 而且, 随着 N 的增大, 其时隙效率越高.

表 2 算法时隙效率

算法	时隙效率	算法	时隙效率	算法	时隙效率
QTA	17.3%	I4QTA	24.5%	BSCITTA	33.3%
DSCITTA2	37.2%	DSCITTA3	35.1%	DSCITTA4	29.6%
IDSCTTA2	37.8%	IDSCTTA3	44.9%	IDSCTTA4	49.7%

5 结论

本文提出了 IDSCTTA 算法, 以便加快读卡器对其通信域内标签的识别速度. 理论与仿真结果表明, IDSCTTA 的时间复杂度要小于其它几种算法, 其通信复杂度也小于除 DSCITTA 外的其它算法. 相对于 DSCITTA 而言, IDSCTTA 以增加极少的传送开销为代价, 极大地减小了识别时延, 加快了系统的识别速率, 节省了大量系统资源.

参考文献

- [1] Vogt H. Efficient object identification with passive RFID tags [A]. Proc Inter Conf on Pervasive Computing, LNCS. 2414 [C]. Zurich, Switzerland: Springer-Verlag, 2002. 98 - 113.
- [2] Park J H, Chung M Y, Lee T J. Identification of RFID tags in framed-slotted ALOHA with robust estimation and binary selection [J]. IEEE Communications Letters, 2007, 11 (5): 452 - 454.
- [3] Lee S R, Joo S D, Lee C W. An enhanced dynamic framed slotted aloha algorithm for FRFID tag identification [A]. Proc Second Annual International Conference Mobile and Ubiquitous Systems: Networking and Services [C]. San Diego, CA, USA: IEEE, 2005. 166 - 172.
- [4] EPC global Inc. EPCTM. Radio-frequency identity protocols class 1 generation-2 UHF rFID protocol for communications at

- 860-960MHz version 1.2.0[S]. EPC global Inc. EPC™. 2008.
- [5] Cha J R, Kim J H. Dynamic framed slotted aloha algorithms using fast tag estimation method for RFID system[A]. Proc Consumer Communications and Networking Conf. [C]. Las Vegas, Nevada, USA: IEEE, 2006. 2: 768 – 772.
- [6] Zhu L, Yum T P. Optimal framed aloha based anti-collision algorithms for RFID systems[J]. IEEE Transactions on communications, 2010, 58(12): 3583 – 3592.
- [7] Myung J H, Lee W J, Srivastava J. Adaptive binary splitting for efficient RFID tag anti-collision [J]. IEEE Communications Letters, 2006, 10(3): 144 – 146.
- [8] Finkenzell K. RFID Handbook: Radio-Frequency Identification Fundamentals and Applications[M]. John Wiley & Sons Ltd, 1999.
- [9] Myung J H, Lee W J, SHIH T K. An adaptive memoryless protocol for RFID tag collision arbitration[J]. IEEE Transactions on Multimedia, 2006, 8(5): 1096 – 1101.
- [10] Myung J H, Lee W J, Srivastava J, Shih T K. Tag-splitting: adaptive collision arbitration protocols for RFID tag identification[J]. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(6): 763 – 775.
- [11] Auto-ID Center. 860MHz-930MHz Class I Radio Frequency Identification Tag Radio Frequency and Logical Communication Interface Specification Candidate Recommendation Version 1.0.1[R]. 2002.
- [12] Cho J S, Shin J D, Kim S K. RFID tag anti-collision protocol: query tree with reversed ID[A]. Proc 10th Inter Conf Advanced Comm Technology [C]. Gangwon-Do, South Korea: IEEE, 2008. 1: 225 – 230.
- [13] Zhou F, Jin D, Huang C, Hao M. Optimize the power consumption of passive electronic tags for anti-collision schemes [A]. Proc of the 5th Inter conf on ASIC[C]. Piscataway, NJ, USA: IEEE, 2003. 2: 1213 – 1217.
- [14] Choi J H, Lee D W, Jeon H, Cha J, Lee H. Enhanced binary search with time-divided responses for efficient RFID tag anti-collision[A]. Proceedings of the IEEE International Conference on Comm [C]. Glasgow, Scotland, UK, 2007. 3853 – 3858.
- [15] Ryu J, Lee H, Seok Y, Kwon T. A hybrid query tree protocol for tag collision arbitration in RFID systems[A]. IEEE International Conference on ICC 07[C]. Glasgow, Scotland, UK, 2007. 5981 – 5986.
- [16] Kim Y, Kim S, Lee S, Ahn K. Improved 4-ary tree algorithm for anti-collision in RFID system[A]. Proc of International Conference on AINA[C]. Bradford, UK: IEEE, 2009. 699 – 704.
- [17] Liang C, Lin H. Using dynamic slots collision tracking tree technique towards an efficient tag anti-collision algorithm in RFID systems[A]. Proc of the 9th International Conference on UIC-ATC[C]. Fukuoka, Japan: IEEE, 2009. 272 – 277.
- [18] Kenneth H Rosen. Discrete Mathematics and Its Applications-7th Edition[M]. New York: McGraw – Hill, 2012. 752 – 755.

作者简介



黄以华 男, 2002年毕业于中国科学技术大学, 获理学博士学位. 现为中山大学电子与通信工程系教授, 博士生导师. 主要从事近场通信、芯片与系统设计及固态电子器件等方面的研究工作.

E-mail: huangyh@mail.sysu.edu.cn



陈小若 男, 1989年生, 现为中山大学信息科学技术学院电路与系统专业硕士研究生, 主要研究领域为 RFID 标签防冲突, 无线定位等.

E-mail: chenxr6@163.com