

# 分裂二进制追踪树标签防碰撞协议

李占青 李光顺 吴俊华 孔令增

(曲阜师范大学信息科学与工程学院 日照 276826)

**摘要** 针对大规模RFID系统中的标签碰撞问题,提出了一种位追踪技术与最优分割理论相结合的标签防碰撞协议。该协议由二进制分裂过程与二进制追踪树识别过程这两个阶段组成。在分裂过程中,随机选择0或1可将当前响应标签集合进行二分,在得到可读时隙或空闲时隙时停止。在识别过程中,由于二分得到的左右子集合标签数近似相等,因此运用最优分割理论对左子集合标签数进行处理以得到右子集合的时隙数,自底向上在各右子集合上采用二进制追踪树时隙算法来完成对标签的识别。分裂过程简单、易实施,识别过程无需预先估计标签数量,对设备计算能力的要求低,设置最优分隔可明显减少空闲时隙。理论分析及仿真结果表明,该协议能够提高RFID系统的识别效率,在大规模RFID系统中性能更优。

**关键词** 标签碰撞,位追踪,最优分割,二进制分裂,二进制追踪树

**中图分类号** TP391.44 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.12.016

## Novel Tag Anticollision Protocol with Splitting Binary Tracking Tree

LI Zhan-qing LI Guang-shun WU Jun-hua KONG Ling-zeng

(School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China)

**Abstract** To solve the problem of tag collision in large scale RFID (Radio Frequency Identify) systems, a new tag anticollision protocol was proposed which combines the bit tracking technology and the optimal partition theory. The protocol consists of two phases, namely, a binary splitting phase and a binary tracking tree identifying phase. The first phase repeatedly divides the set of the current response tags into two subsets by choosing "0" or "1" randomly until a readable slot or an idle slot is obtained. The second phase first handles the number of tags in the left subset by using the optimal partition theory to obtain the size of slots in the right subset. And then the second phase finishes the identification of tags by utilizing the binary tracking tree slot algorithm on all right subsets in a bottom-up manner. The splitting process is simple and easy to implement, and the recognition process is not required to estimate the number of tags in advance, so the computing power of the device is low, moreover, the optimal partition can obviously reduce the idle slots. Theoretical analysis and simulation results demonstrate that the protocol can improve identification efficiency and do better in large scale RFID system.

**Keywords** Tag collision, Bit tracking, Optimal partition, Binary splitting, Binary tracking tree

## 1 引言

射频识别(RFID)技术是一种方便快捷的非接触式自动识别技术,其通过无线电信号识别特定目标并读写相关数据<sup>[1]</sup>,可以替代传统的条形码识别方式。一般情况下,RFID系统由一个阅读器和多个标签组成,其中标签可分为有源标签、半有源标签和无源标签。无源标签不需要自身携带电源,而是从阅读器的传输信号中获得工作中所需要的能量,具有体积小、成本低的优点,在实际中被广泛应用。因此,本文主

要研究无源标签组成的RFID系统。

当多个标签同时响应阅读器时,它们由于共享相同的无线信道而发生碰撞,从而使阅读器不能正确识别标签,该问题即为常见的标签碰撞问题。标签碰撞将增加系统延迟,降低系统识别效率。在大规模RFID系统中存在成千上万的标签,碰撞会严重增加系统负担,因此设计高效的标签防碰撞算法来提高大规模RFID系统的识别效率尤为重要。

当前越来越多的研究者关注位追踪技术在标签防碰撞算法中的应用。文献[2]提出的基于查询树(QT)的碰撞追踪树

到稿日期:2016-09-05 返修日期:2017-01-03 本文受国家自然科学基金项目(61373027),山东省自然科学基金项目(ZR2012FM023),山东省优秀中青年科学家科研奖励基金(BS2014DX005),山东省高校科技计划项目(J14LN38),山东省研究生教育创新计划项目(SDY14052),曲阜师范大学科技计划项目(xkj201525)资助。

李占青(1991—),男,硕士生,主要研究方向为RFID技术,E-mail:914115569@qq.com;李光顺(1977—),男,博士,副教授,硕士生导师,CCF高级会员,主要研究方向为RFID技术、计算机网络与通信,E-mail:30752585@qq.com(通信作者);吴俊华(1979—),女,博士,副教授,硕士生导师,主要研究方向为RFID技术、计算机网络与通信。

算法(CTTA)与文献[3]提出的基于二进制树(BT)的增强防碰撞算法(EAA)是两种使用位追踪技术的典型算法,它们的思想类似,都根据标签 ID 中第一个碰撞位将碰撞的标签分成两个子集合。文献[4]提出一种新的增强型防碰撞算法(NEAA),该算法根据标签 ID 中“1”的位数将标签分成一些集合,减少识别开始时发生碰撞的次数,在识别过程中再使用位追踪技术以在一个时隙内识别多个标签。文献[5]提出一种分配树时隙协议(ATSA),然后加入位追踪技术得到改进的分配树时隙协议(ImATSA)。文献[6]和文献[7]分别提出最优查询追踪树协议(OQTT)与最优二进制追踪树协议(OBTT),这些协议将位追踪技术与树时隙算法相结合来对标签进行识别。文献[8]提出一种高效无线射频识别自适应型跟踪树防碰撞算法(HACT),该算法借鉴 CT 算法的碰撞位定位思想,设计二叉树碰撞位跟踪算法来对新到标签进行识别。文献[9]提出基于碰撞位跟踪的分组 N 叉跟踪树型 RFID 防碰撞算法(CBGN),该算法通过检测 ID 的碰撞来实现对分组内标签的 N 叉分隔。文献[10]提出一种基于信息位编码的自适应搜索防碰撞算法(SA),阅读器充分利用碰撞位信息自适应地生成有效查询前缀,从而避免空闲时隙,减少查询次数。

位追踪技术采用曼彻斯特编码方式,它允许阅读器检测碰撞位发生的位置,以此进行准确定位,可以更好地解决碰撞,进而显著提高系统的识别效率。二进制分裂凭借其简单、易实施的分裂方式,也在标签防碰撞算法的设计中得到了广泛应用。文献[11]将二进制分裂用于对标签数的估计,提出了二进制分裂树时隙 Aloha 协议(STSA)。文献[12]和文献[13]将分裂过程用于对帧长度的调整,提出了分裂二进制树时隙协议(SBTS)。文献[14]则利用二进制分裂来消除空闲时隙,提出了基于二元分裂的空闲时隙消除防碰撞算法(ISEBS)。本文提出分裂二进制追踪树标签防碰撞协议(Splitting Binary Tracking Tree, SBTT),该协议充分运用位追踪技术与二进制分裂的优势,同时在设置帧长度时应用最优分割理论对标签进行分组以减少碰撞,从而提高大规模 RFID 系统的识别效率。

## 2 位追踪技术与最优分割理论

### 2.1 位追踪技术

位追踪技术采用曼彻斯特编码方案,在一位时间内,当电平由低到高转变时(上升沿)编码值为“0”;当电平由高到低转变时(下降沿)编码值为“1”。在 RFID 系统中,若阅读器接收到的曼彻斯特编码信号在一位时间内未发生转变,则可知该位发生了碰撞。

图 1 给出了一个曼彻斯特编码的实例。

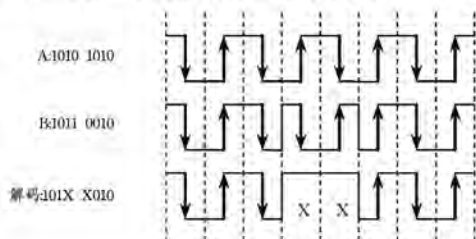


图 1 曼彻斯特编码实例

图 1 中,A 标签的 ID 编码为“1010 1010”,B 标签的 ID 编码为“1011 0010”,当标签 A 和标签 B 使用曼彻斯特编码方法同时发送 ID 给阅读器时,阅读器解码得到“101X X010”,其中“X”表示一个碰撞位。在该实例中,碰撞位的位置分别是第四位和第五位。

### 2.2 最优分割理论

为了减少识别开始时的碰撞次数,文献[6]和文献[7]运用最优分割理论对标签集合进行分组,对识别开始时的初始分组数进行了严密的分析计算。已知二进制追踪树中所消耗的平均时隙数为:

$$H_{BTT}(n, m) = 2n - m + 2m(1 - \frac{1}{m})^n$$

其中,  $n$  是标签数,  $m$  是分配的初始时隙数(初始分组数)。为了找到理想的初始时隙数,使协议所消耗的总时隙数最少,对上式求导,令:

$$\begin{aligned} H'_{BTT}(n, m) &= 0 \\ \Rightarrow -1 + 2(1 - \frac{1}{m})^{n-1}(1 - \frac{1}{m} + \frac{n}{m}) &= 0 \\ \Rightarrow (1 - \frac{1}{m})^{n-1}(\frac{m+n-1}{m}) &= \frac{1}{2} \end{aligned}$$

由该方程可知  $m \approx 0.595824n$ ,即待识别的标签数为  $n$ ,则该组标签在识别开始时的理想时隙数(帧长度)应为  $0.595824n$ 。在分裂过程中,由于左右子集合的标签数近似相等,得到左子集合标签数  $\frac{n}{2}$  之后,可得出对右子集合进行识别的初始时隙数为  $0.595824 \times \frac{n}{2}$ 。

## 3 分裂二进制树时隙协议(SBTS)

分裂二进制树时隙协议由分裂算法和二进制树时隙算法两部分构成<sup>[12-13]</sup>。首先对当前响应的碰撞标签进行分裂,直到产生可读时隙或空闲时隙为止。分裂结束后,以左子集合标签数作为右子集合时隙数,在得到的右集合上执行二进制树时隙算法。整个协议的执行过程如图 2 所示。

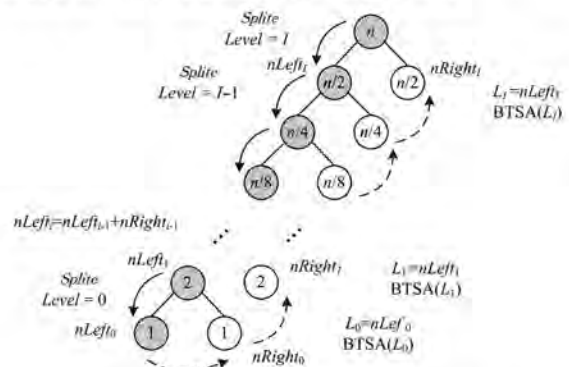


图 2 分裂二进制树时隙协议的执行过程

一方面,由于该协议中的标签随机选择二进制数“0”和“1”,因此碰撞的处理时间较长;另一方面,当标签数为  $n$  时,在 Aloha 算法中最优的初始时隙数为  $n$ ,而在二进制树算法中最优的初始时隙数小于  $n$ 。因此,本文将位追踪技术与最优分割理论相结合,提出分裂二进制追踪树标签防碰撞协议,解决了 RFID 系统中的标签碰撞问题。

### 4 分裂二进制追踪树标签防碰撞协议(SBTT)

本文提出分裂二进制追踪树标签防碰撞协议(SBTT)。首先,采用最优分割得到的时隙数初始化帧长度;然后,采用位追踪技术对标签识别过程中出现的碰撞进行准确定位;最后,根据标签自身 ID 碰撞位的编码值,置计数器值为“0”或“1”。在识别之前,通过最优分割将标签分成一系列子集能够有效减少识别开始时的碰撞次数;在识别过程中,利用位追踪技术能够有效地解决标签碰撞问题。

本文提出的 SBTT 协议分为两部分:分裂过程和二进制追踪树识别过程。在分裂过程中对当前响应的标签持续二分,直至出现空闲时隙或可读时隙为止。分裂结束后得到一棵二叉树,由于树中根节点左右子集合的标签数近似相等,因此运用最优分割理论对左子集合标签数进行处理,并将其结果作为右子集合响应时隙数(即帧长  $L$ ),然后在右子集合上采用二进制追踪树时隙算法识别标签,依次自底向上完成对所有标签的识别。

本文算法在标签内部设置计数器  $Counter$ ,当  $Counter=0$  时标签将自身 ID 传输给阅读器;在阅读器内部设置计数器  $SC$ ,当  $SC < L$  (其中  $L$  为帧长) 时进行循环,识别当前右集合中的标签。 $SplitQuery$  表示分裂命令,  $Query$  表示识别命令,  $f$  表示时隙状态,其中  $f \in \{collision, idle, successful\}$ 。

本文提出的分裂二进制追踪树标签防碰撞协议分为阅读器算法和标签算法两部分。阅读器算法的流程图如图 3 所示;标签算法的流程图如图 4 所示。其中  $BTTSA()$  函数的流程图单独给出,如图 5 所示。

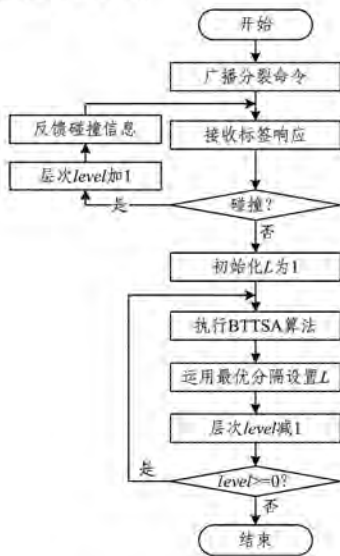


图3 阅读器算法的流程图

阅读器算法和标签算法的伪代码描述如算法 1 所示。

#### 算法 1 阅读器算法的伪代码描述

```

1. [level, f]=Splitting()
2. SBTTSA(level, f)
3. function Splitting()
4. broadcast SplitQuery, and level=0
5. do{ Receive tag response and detect collision
6.   if tag collision

```

```

7.     f=collision, and level=level+1
8.   elseif no tag response
9.     f=idle
10.  elseif only a tag response
11.    Receive ID from tag and store it
12.    f=successful
13.  end
14.  Transmit f } while f=collision
15. return level and f
16. function SBTTSA(level, f)
17. if f=idle
18.   nLeft=0
19. elseif f=successful
20.   nLeft=1
21. end
22. L=1
23. do{ BTTSA(L)
24.   nRight=CountNumSuccess()
25.   // Count the number of successful slots
26.   nLeft=nLeft+nRight
27.   L=0.595824 * nLeft
28.   Level=level-1 } while level >= 0
29. function BTTSA(L)
30. Broadcast Query with L, and SC=0
31. do{ Receive tag response and detect collision
32.   if tag collision
33.     Find the location i of first collided bit
34.     Transmit f=collision with i
35.     BinTrackTree()
36.   elseif no tag response
37.     Transmit f=idle
38.   else only a tag response
39.     Receive ID from tag
40.     Transmit f=successful
41.   end
42.   SC=SC+1 } while SC < L
43. function BinTrackTree()
44. B=2
45. do{ Receive tag response and detect collision
46.   if tag collision
47.     Find the location i of first collided bit
48.     Transmit f=collision with i, and B=B+1
49.   elseif no tag response
50.     Transmit f=idle, and B=B-1
51.   else only a tag response
52.     Receive ID from tag
53.     Transmit f=successful
54.     B=B-1
55.   end } while B > 0

```

该阅读器算法由分裂函数 Splitting() 和识别函数 SBTTSA() 组成。

算法 1 的第 3—15 行是分裂函数 Splitting() 的伪代码描述。它根据响应标签是否碰撞来判断时隙状态,若时隙状态

为“collision”,则继续对当前响应标签进行分裂。分裂完成后,返回整个二叉树的层数以及最后一个时隙的状态。

算法 1 的第 16—28 行是识别函数 SBT TSA() 的伪代码描述。它首先根据分裂过程得到的结果来判断已识别的标签数为 0 还是为 1,然后循环对每一层右集合标签的识别,在循环过程中需要计算已经识别的标签数,并将其累加得到上一层左集合的标签数以便于进行最优分割。

算法 1 的第 29—42 行是右集合标签的识别函数 BIT TSA() 的伪代码描述。它根据帧长度  $L$  来判断何时完成对该右集合的识别,若识别过程中出现碰撞,则先找到发生碰撞的位置,然后将带有碰撞位置信息的时隙状态发送给标签,最后执行碰撞处理函数 BinTrackTree()。

算法 1 的第 43—55 行是碰撞处理函数 BinTrackTree() 的伪代码描述。它使用二进制追踪树算法来解决碰撞,本质上属于二进制树。由于二进制树中父节点有两个孩子节点,因此初始  $B=2$ ,然后根据标签响应判断是否碰撞来改变  $B$  的值,直到  $B=0$  时碰撞处理完成。

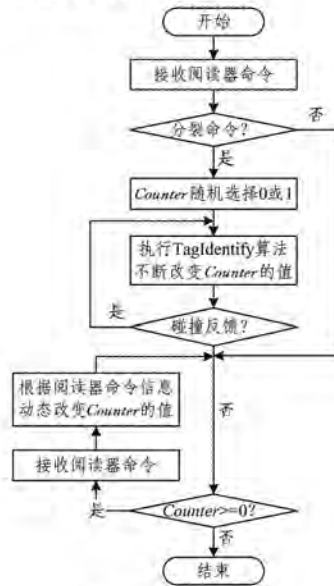


图 4 标签算法的流程图

#### 算法 2 标签算法的伪代码描述

```

1. Receive Reader's Command
2. if receive Reader's SplitQuery
3.   Counter=a random binary number 0 or 1
4.   do{ [f,Counter]=TagIdentify(Counter) } while f=collision
5. end
6. while Counter >=0
7.   do{ Receive Reader's Command
8.     if receive Reader's Query with L
9.       if Counter=0
10.        Counter=a random number from 0 to L-1
11.        SendID()
12.       else
13.        Counter=Counter+L-1
14.       end
15.     elseif receive Reader's f with i
16.       if Counter=0
17.         if f=successful
18.           Identified, and Counter=Counter-1

```

```

19.         // Keep silent
20.       else
21.         Counter=Counter+ID(i)
22.         SendID()
23.       end
24.     else
25.       if f=collision
26.         Counter=Counter+1
27.       else
28.         Counter=Counter-1
29.         SendID()
30.       end
31.     end
32.   end)
33. function TagIdentify(Counter)
34.   if Counter=0
35.     Send ID and receive f from reader
36.     if f=successful
37.       Identified, and Counter=Counter-1
38.     else
39.       Counter=Counter+a random binary number 0 or 1
40.     end
41.   else
42.     Receive f from reader
43.     if f=collision
44.       Counter=Counter+1
45.     else
46.       Counter=Counter-1
47.     end
48.   end
49.   return Counter and f
50. function SendID()
51.   if Counter=0
52.     Send ID
53.   end

```

与阅读器操作相对应,标签操作也主要由分裂步骤(算法 2 的第 1—5 行)和识别步骤(算法 2 的第 6—32 行)组成。

标签在接收到分裂命令后,随机选择“0”或“1”设置计数器的值,具体的分裂操作由函数 TagIdentify() 给出,该函数的参数是标签自身计数器的值。算法 2 的第 33—49 行是函数 TagIdentify() 的伪代码描述。若计数器的值为 0,则作出响应,并继续接收阅读器的反馈信息,根据阅读器反馈的时隙状态信息做出下一步操作;若计数器的值不为 0,则根据阅读器反馈的时隙状态进行“+1”或“-1”操作。

只在标签计数器的值大于或等于 0 时执行识别步骤(计数器的值小于 0 说明该标签已被识别)。标签先等待接收阅读器命令,若接收到带有帧长度  $L$  的识别命令 Query,则判断自身计数器的值是否为 0。若计数器的值为 0,则在  $0 \sim (L-1)$  之间随机选择一个数重置计数器的值,并执行 SendID() 函数;若计数器的值不为 0,则将计数器的值加上  $(L-1)$ 。若计数器为 0 的标签接收到的时隙状态为“successful”,则进行“-1”操作;否则,根据自身 ID 碰撞位的编码值重置计数器的值,并执行 SendID() 函数。对于计数器不为 0 的标签,只需要根据具体的时隙状态对其进行“+1”或“-1”操作。SendID()

函数只对计数器值为0的标签发送ID,算法2中的第50-53行给出了其伪代码描述。

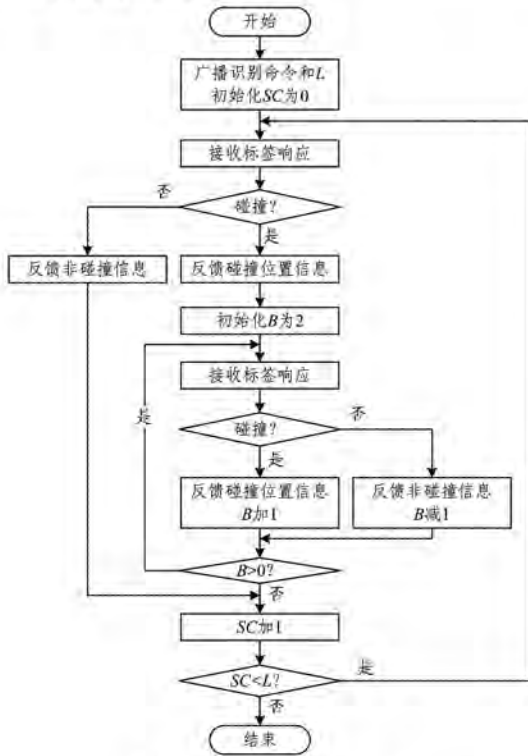


图5 BTSSA()函数的流程图

### 5 算法性能分析

在标签识别过程中包含3种时隙:成功时隙、空闲时隙和碰撞时隙。一般情况下,假设3种时隙的持续时间相同。成功时隙数与3种时隙总数的比值表示识别效率,它可以间接反映算法的时间复杂度,是评价标签防碰撞算法性能的重要指标。

令  $R_{SBTT}$  为本文分裂二进制追踪树协议识别  $n$  个标签所需总时隙数的期望值,根据文献[11]可得:

$$R_{SBTT} = 1 + \sum_{k=0}^n P\{B_{n,1/2}(k)\} \times (R_{SBTT}[k] + R_{BTT}[k_1]) \quad (1)$$

其中,1表示1次初始分裂消耗的1个时隙; $P\{B_{n,1/2}(k)\} = C_n^k 2^{-n}$ 表示  $n$  个标签分裂到两个集合的概率,其中  $k+k_1=n$ ;  $R_{BTT}[i]$ 表示使用二进制追踪树算法识别  $i$  个标签集合所需时隙数的期望值。

由于左集合递归地进行二进制分裂,假设分裂次数为  $\lambda$ ,那么理想情况下对  $n$  个标签进行分裂直到出现单标签或空标签集合需要的分裂次数  $\lambda = \log_2 n$  (由  $2^\lambda = n$  可以得出),因此:

$$R_{SBTT} = \lambda + \sum_{k=0}^n P\{B_{n,1/2}(k)\} \times R_{SBTT}[k] + \sum_{k=0}^n P\{B_{n,1/2}(k)\} \times R_{BTT}[k_1] + \dots + \sum_{k=0}^n P\{B_{n,1/2}(k)\} \times R_{BTT}[k_\lambda] \quad (2)$$

其中  $k \in \{0, 1\}$  (分裂结束后得到空闲时隙或成功时隙,其对应的标签数分别为0或1)。由于二进制分裂得到的左右集合标签数近似相等,因此右式中:

$$\sum_{k=0}^n P\{B_{n,1/2}(k)\} \times R_{BTT}[k_1] + \dots + \sum_{k=0}^n P\{B_{n,1/2}(k)\} \times R_{BTT}[k_\lambda] \approx R_{BTT}[k] \quad (3)$$

由此可得:  $R_{SBTT} = \log_2 n + R_{BTT}[n]$ 。由文献[7]得  $R_{BTT}[n] \approx 1.6266354n$ ,则  $R_{SBTT} = \log_2 n + 1.6266354n$ 。

仿真实验中以标签数目为变量,依次增加标签数目,首先执行分裂算法,得到一系列标签集合;接着在分裂得到的标签集合上执行识别算法,计算每个集合中所需要的时隙数;最后得到总的时隙数,并对其进行分析。

### 6 仿真结果

采用 MATLAB 对本文 SBTT 协议进行仿真实验,并将仿真结果与理论分析结果、SBTS 协议结果、OBTT 协议结果进行比较。

图6为仿真结果与理论分析结果的对比图。可以看出,仿真结果与理论分析结果基本一致,但也存在一定误差。当标签数目相对较少时,理论分析结果优于仿真结果;随着标签数目的不断增加,两条曲线不断接近,最终基本重合。出现误差的原因在于:当标签数目较少时,二进制分裂得到的左右两个集合中标签数目的差值相对较大。

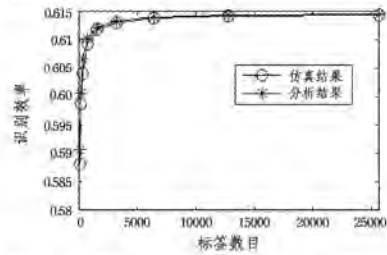


图6 仿真结果与理论分析结果的对比图

图7给出了二进制分裂得到的左右两个集合中标签数目的差值随标签数目变化的关系。可以看出,随着标签数量的不断增大,曲线斜率逐渐变小,这说明标签数目越多时分裂出现的相对误差越小。同理,标签数目越少时分裂出现的相对误差越大。因此,识别效率的仿真结果与理论分析结果在标签数目较少时存在一定误差。

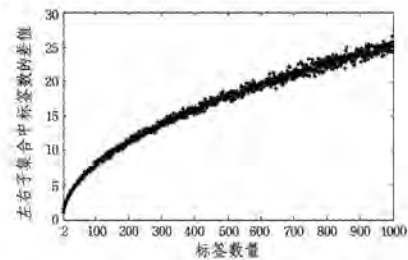


图7 二进制分裂左右集合中标签数的差值与标签总数的关系图

图8为分裂过程中消耗的平均时隙数随标签数量变化的示意图。图9给出了分裂过程中消耗时隙数占标签总数的百分比。当标签数量较大时,分裂次数占标签总数的百分比相对较小;而当标签数量较少时,分裂次数占标签总数的百分比相对较大。显然,当标签数量较少时,分裂过程所花费的时间占总识别时间的百分比比较大;而标签数量较大时,分裂过程花费的时间占总识别时间的百分比则比较小。由此看出,本文提出的 SBTT 协议在标签数量较大时具有更好的性能。

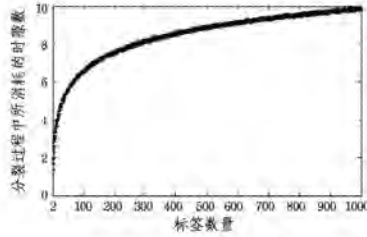


图 8 分裂过程中所消耗的时隙数随标签数目变化的示意图

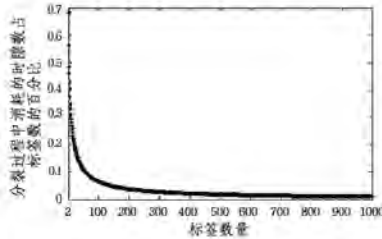


图 9 分裂过程中消耗的时隙数与标签总数的比值示意图

文献[12]提出 SBTS 协议,并分析得出 BTSA 算法与 SBTS 协议的最优识别效率都约为 0.429。本文 SBTT 协议的识别效率与 SBTS 协议的识别效率(最优值为 0.429)的对比结果如图 10 所示(选取的标签数量变化范围为 10~1000)。可以看出,标签数量在 10~1000 的变化范围内时,本文 SBTT 协议的识别效率较 SBTS 协议的识别效率平均提高了 17.28%,并且随着标签数量的增加,SBTT 协议的识别效率还会缓慢提高。

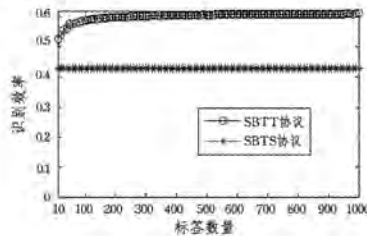


图 10 本文 SBTT 协议与 SBTS 协议识别效率的对比图

文献[7]提出 OBTT 协议,其识别效率在现有标签防碰撞算法中是最高的,约为 61.4%。将本文提出的 SBTT 协议与 OBTT 协议进行比较,两者识别效率的对比结果如图 11 所示。

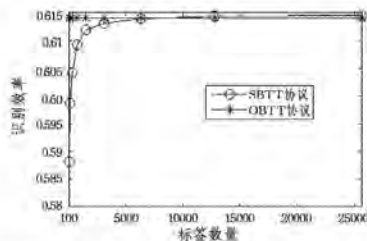


图 11 本文 SBTT 协议与 OBTT 协议的识别效率对比图

可以看出,当标签数为 100 时,本文提出的 SBTT 协议的识别效率达到 58.82%;当标签数为 200 时,其识别效率达到 60%;当标签数为 6400 左右时,其识别效率达到 61.4%;当标签数大于 23100 时,其识别效率稳定在 61.45%左右。由图 11 可知,在标签数相对较少时,SBTT 协议的识别效率低于 OBTT 协议的识别效率,原因在于标签数目较少时分裂过程

消耗的时隙数占总时隙数的比例较高,识别效率较低;但随着标签数目的增加,识别效率也不断增加。在标签数目较大时,分裂过程所消耗的时隙数占总时隙数的比例较低,证明本文提出的 SBTT 协议的识别效率略高于 OBTT 协议的识别效率。

**结束语** 本文将二进制分裂算法与二进制追踪树时隙算法相结合,提出分裂的二进制追踪树标签防碰撞协议。该协议不单独对标签数量进行估计,对阅读器计算能力的要求低,分裂算法简单、易实施。算法运用最优分割理论对标签进行分组,减少了识别开始时碰撞的发生次数,同时大量减少了识别过程中的空闲时隙。分裂算法与追踪树算法的结合为标签防碰撞提供了新的解决思路。由仿真结果可知,该协议能够实现目前协议所实现的最高识别效率,在标签数量很大时,性能表现优越;但该协议仍有不足之处,比如在标签数量很少时识别效率相对较低。在今后的标签防碰撞算法研究中,应重点考虑追踪树算法的应用,同时关注更加快速的标签分裂方法,并设计智能型算法。无论在标签数量很大还是标签数量很小时,依靠算法在识别过程中进行最优的自适应调整都能高效地解决碰撞从而实现识别效率的最大化,这是今后研究标签防碰撞算法的更高目标。

### 参考文献

- [1] XIA J M, XIAO G Q, CHEN K, et al. Framed slotted ALOHA anti-collision algorithm using hybrid spill-tree [J]. Computer Science, 2014, 41(12): 67-69. (in Chinese)  
夏静满, 肖国强, 陈凯, 等. 基于混合溢出树搜索的帧时隙 ALOHA 防碰撞算法[J]. 计算机科学, 2014, 41(12): 67-69.
- [2] ZHOU F, JIN D, HUANG C, et al. Optimize the power consumption of passive electronic tags for anti-collision schemes[C]// Proceedings of the 5th International Conference on Association for Science and Information on Coffee. New York: IEEE Press, 2003: 1213-1217.
- [3] CHEN W C, HORNG S J, FAN P. An enhanced anti-collision algorithm in RFID based on counter and stack[C]// Proceedings of the Second International Conference on Systems and Networks Communications. New York: IEEE Press, 2007: 1-4.
- [4] CHEN Y H, HORNG S J, RUN R S, et al. A novel anti-collision algorithm in RFID systems for identifying passive tags[J]. IEEE Transactions on Industrial Informatics, 2010, 6(1): 105-121.
- [5] ZHANG L, ZHANG J, TANG X. Assigned tree slotted Aloha RFID tag anti-collision protocols [J]. IEEE Transactions on Wireless Communications, 2013, 12(11): 5493-5505.
- [6] LAI Y C, HSIAO L Y, CHEN H J, et al. A Novel Query Tree Protocol with Bit Tracking in RFID Tag Identification[J]. IEEE Transactions on Mobile Computing, 2012, 12(10): 2063-2075.
- [7] LAI Y C, HSIAO L Y, LIN B S. Optimal Slot Assignment for Binary Tracking Tree Protocol in RFID Tag Identification[J]. IEEE/ACM Transactions on Networking, 2015, 23(1): 255-268.
- [8] WANG X, JIA Q X, GAO X, et al. Highly efficient RFID adaptive tracking tree anti-collision algorithm[J]. Journal of Jilin University (Engineering and Technology Edition), 2015, 45(4): 1225-1233. (in Chinese)

本文采用多尺度分解来对原始图像进行分块,考虑了图像的纹理复杂性,提高了视觉质量;在同质块中通过预测误差扩展算法来嵌入水印,并在非同质块中在 IWT 的中高频系数中嵌入水印,有效地提高了嵌入的视觉质量;在同质块中通过预测误差扩展来嵌入水印,在理想状态下嵌入率可达 1;在非同质块中在 IWT 的中高频系数中嵌入水印,其嵌入率可达 0.75。该方法有效地提高了整个算法的嵌入率并增加了嵌入容量。这里给出的结果表明,本文提出的基于多尺度分解与预测误差扩展的可逆图像水印算法大大增强了有效载荷的能力,同时仍保持了较高的视觉质量。

**结束语** 最大嵌入量和视觉质量是可逆图像水印算法中两个重要的性能评价指标,两者之间相互矛盾,在保证较好视觉质量的前提下能够有效地提高水印嵌入率是目前亟需解决的重要问题。本文基于此提出了一种基于多尺度分解和预测误差扩展的可逆图像水印算法,在提取嵌入的水印后,该算法能够无损恢复出原始图像。该算法能够有效增加水印嵌入量并且具有较高的视觉质量,其主要贡献在于和其他一些算法相比在维持较高视觉质量的前提下提高了水印嵌入率。

### 参考文献

- [1] LEI L, GUO S X, WANG L. SVD Digital Image Watermarking Algorithm Based on Wavelet Transform [J]. Computer Simulation, 2013, 30(9): 169-172, 260. (in Chinese)  
雷蕾, 郭树旭, 王雷. 基于小波变换的 SVD 数字图像水印算法研究[J]. 计算机仿真, 2013, 30(9): 169-172, 260.
- [2] TIAN J. Reversible data embedding using a difference expansion [J]. IEEE Transactions Circuits Syst. Video Technol., 2003, 13(8): 890-896.
- [3] ZHENG S L, WANG M L, XING H F, et al. A difference expansion reversible watermarking algorithm based on bit plane [J]. Journal of Hefei University of Technology (Natural Science), 2012, 35(10): 1345-1348. (in Chinese)  
郑淑丽, 王美玲, 邢慧芬, 等. 一种基于位平面的差值扩展可逆水印算法[J]. 合肥工业大学学报(自然科学版), 2012, 35(10): 1345-1348.
- [4] MAITY H K, MAITY S P. Reversible image watermarking using modified difference expansion [C] // 2012 Third International Conference on Emerging Applications of Information Technology (EAIT). IEEE, 2012: 320-323.
- [5] LI X L, LI J, LI B, et al. High-fidelity reversible data hiding scheme based on pixel-value ordering and prediction-error expansion [J]. Signal Processing, 2012, 93(1): 198-205.
- [6] LIN S L, HUANG C F, LIU M H, et al. Improving histogram based reversible information hiding by an optimal weight-based prediction scheme [J]. Journal of Information hiding and Multimedia Signal Processing, 2013, 1(1): 19-33.
- [7] ZHANG Z W, WU L F, LAI H G, et al. Reversible watermarking algorithm with IWT and general difference expansion [J]. Computer Engineering and Applications, 2016, 52(8): 84-89. (in Chinese)  
张正伟, 吴礼发, 赖海光, 等. 基于 IWT 和广义差值扩展的可逆水印算法[J]. 计算机工程与应用, 2016, 52(8): 84-89.
- [8] YAN Y Y, GUO Z B, YANG J Y. Face recognition based on multi-scale singular value features [J]. Journal of Tsinghua University (Science and Technology), 2008, 48(10): 1688-1692. (in Chinese)  
严云祥, 郭志波, 杨静宇. 基于多尺度奇异值特征的人脸识别[J]. 清华大学学报(自然科学版), 2008, 48(10): 1688-1692.
- [9] LI Y Q, LI J X, LIANG Y H. Image Watermarking Algorithm based on Image Characteristics and Huffman Coding [J]. Computer Applications and Software, 2013, 30(9): 128-130, 140. (in Chinese)  
李亚琴, 李金祥, 梁颖红. 基于图像特征和霍夫曼编码的图像水印算法[J]. 计算机应用与软件, 2013, 30(9): 128-130, 140.
- [10] GAO Z Y, YANG X M, GONG J M, et al. Research on image complexity description methods [J]. Image and Journal of Graphics, 2010, 15(1): 129-135.
- [11] EL-SAYED H S, EL-ZOGHDY S F, FARAGALLAH O S. Adaptive Difference Expansion-Based Reversible Data Hiding Scheme for Digital Images [J]. Arabian Journal for Science and Engineering, 2016, 41(3): 1091-1107.
- [12] ABDULLAH S M, MANAF A A. Multiple Layer Reversible Images Watermarking Using Enhancement of Difference Expansion Techniques [M] // Networked Digital Technologies. Springer Berlin Heidelberg, 2010: 333-342.
- [13] WU H F, ZENG Y, FENG J H. Passive RFID tag anti-collision binary tree slotted protocol without tags quantity estimation [J]. Journal of Computer Research and Development, 2012, 49(9): 1959-1971. (in Chinese)  
吴海峰, 曾玉, 丰继华. 无标签数估计的被动 RFID 标签防冲突二进制树时隙协议[J]. 计算机研究与发展, 2012, 49(9): 1959-1971.
- [14] WU H, ZENG Y, FENG J, et al. Binary tree slotted ALOHA for passive RFID tag anticollision [J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(1): 19-31.
- [15] SU J, SHENG Z, XIE L, et al. Idle-Slots elimination based binary splitting (ISE-BS) anti-collision algorithm for RFID [J]. IEEE Communications Letters, 2016, 20(12): 2394-2397.

(上接第 85 页)

王鑫, 贾庆轩, 高欣, 等. 高效无线射频识别自适应跟踪树防碰撞算法[J]. 吉林大学学报(工学版), 2015, 45(4): 1225-1233.

- [9] WANG X, JIA Q X, GAO X, et al. Research on grouping n-ray tracking tree RFID anti-collision algorithm [J]. Acta Electronica Sinica, 2016, 44(2): 437-444. (in Chinese)  
王鑫, 贾庆轩, 高欣, 等. 分组 N 叉跟踪树型 RFID 防碰撞算法研究[J]. 电子学报, 2016, 44(2): 437-444.
- [10] HU Y M, ZHANG X H. Research of an adaptive searching anti-collision algorithm for RFID based on information-bit encoding [J]. Acta Electronica Sinica, 2016, 44(8): 1791-1798. (in Chinese)  
胡应梦, 张小红. 基于信息位编码的自适应搜索 RFID 防碰撞算法研究[J]. 电子学报, 2016, 44(8): 1791-1798.
- [11] LA P T, MASELLI G, PETRIOLI C. Anticollision protocols for