捕获效应下 RFID 标签的 ABTS 防冲突协议

吴海锋 [1] 曾 玉 [1] 凌永发 [1]

摘 要 在被动式射频识别 (Radio frequency identification, RFID) 系统中, 即使有多个标签同时向阅读器发送信息, 捕获效应也能使其中一个标签被阅读器成功识别. 而且, 捕获效应还会导致标签漏读. 为防止标签漏读和进一步提高捕获效应下的识别效率, 我们提出一种新的防冲突协议, 该协议采用自适应和分配技术来减少标签间冲突. 其优点在于, 可合理地分配冲突标签和隐藏标签, 从而提高识别效率. 计算机仿真结果显示, 在捕获效应发生的环境下, 本文协议的识别效率优于现存协议.

关键词 射频识别, 标签防冲突, 捕获效应, 自适应, 二进制树

引用格式 吴海锋, 曾玉, 凌永发. 捕获效应下 RFID 标签的 ABTS 防冲突协议. 自动化学报, 2014, **40**(3): 506−515

DOI 10.3724/SP.J.1004.2014.00506

ABTS Protocol for RFID Tags Anti-collision Under Capture Effect

WU Hai-Feng[1] ZENG Yu[1] LING Yong-Fa[1]

Abstract In passive radio frequency identification (RFID) systems, capture effect will make a tag′s signal decoded even when multiple tags simultaneously transmit their signals to a reader. Besides, capture effect will make the reader miss some tags. In this paper, we propose a novel tag anti-collision protocol for capture effect to avoid missing tags and further enhance the efficiency of identification. The proposed protocol adopts an adaptive and allocation technique. The advantage of the adopted techniques is to reasonably allocate slots to hidden tags by capture effect and collided tags, reduce collision between the tags and thus enhance the identification efficiency. Computer simulation results show that the proposed protocol′s efficiency outperforms other existing protocols under a capture effect environment.

Key words Radio frequency identification (RFID), tag anti-collision, capture effect, adaptive, binary tree

Citation Wu Hai-Feng, Zeng Yu, Ling Yong-Fa. ABTS protocol for RFID tags anti-collision under capture effect. *Acta Automatica Sinica*, 2014, **40**(3): 506−515

物联网被视为继计算机、互联网之后信息技术发展的第三次浪潮, 它通过互联网对物品的信息进行共享和交换, 以此实现对物体识别、定位、跟踪、监控和管理[1]. 物联网的建立以传感器的广泛应用为基础, 因为成千上万的物品信息需通过传感器进行采集, 而射频识别 (Radio frequency identification, RFID) 技术正是这样一种传感技术, 因此, 也成为了物联网的关键技术之一. 在物联网中, 大量 RFID 标签被阅读器快速识别, 为提高识别效率, 阅读器需一次识别多个标签. 由于阅读器识别标签采用共享的无线信道, 因此标签冲突不可避免[2], 通常阅读器采用防冲突协议来解决标签间冲突[3], 所以标签的防冲突协议在物联网中起到了重要作用.

在被动式 RFID 系统中, 两个以上的标签同时发射信号并不必然导致冲突. 在阅读器的磁场范围内, 标签距离阅读器的位置有远有近, 因此, 发射的信号也有强弱差别, 而捕获效应可使信号较强的标签被阅读器识别, 而信号较弱的标签将被捕获效应所隐藏[4]. 由于许多传统的防冲突协议[3, 5−18] 均假设捕获效应不存在, 而捕获效应恰恰在 RFID 系统中又是一个普遍现象, 因此近几年, 如何在捕获效应下解决标签冲突问题得到越来越多的关注.

如何在捕获效应下识别标签, 现存多种解决方法[4, 19−21], 然而由于以下原因, 这些方法的识别效率并不高. 首先, 被捕获效应所隐藏的标签需要到下一个识别循环中继续选择时隙被识别, 而未识别的标签也要到下一个循环中选择时隙, 因此, 捕获效应下的识别效率不仅与未识别标签, 也与隐藏标签如何选择时隙相关. 遗憾的是, 现存方法并没有合理地将时隙分配给这两类标签, 导致它们之间的冲突增多. 其次是帧长设置的问题, 即设置一个识别循环的时隙数. 在被动式 RFID 标签防冲突协议中, 动

收稿日期 2012-11-06 录用日期 2013-05-02
Manuscript received November 6, 2012; accepted May 2, 2013
国家自然科学基金 (61262091), 云南省应用基础研究计划 (2011FB083), 云南省教育厅科学研究基金重大项目 (ZD2011009), 云南民族大学青年基金 (2012QN09), 云南民族大学创新团队项目资助
Supported by National Natural Science Foundation of China (61262091), Applied and Basic Research Foundation of Yunnan Province (2011FB083), the Major Special Project of Scientific Research Foundation of Yunnan Provincial Department of Education (ZD2011009), the Youth Foundation of Yunnan University of Nationalities (2012QN09), and Innovative Research Team in Yunnan University of Nationalities
本文责任编委 关新平
Recommended by Associate Editor GUAN Xin-Ping
1. 云南民族大学电信学院 昆明 650500
1. School of Electrical and Information Technology, Yunnan Nationalities University, Kunming 650500

态帧时隙 ALOHA 类协议应用非常广泛[5−11, 20−23]. 该协议的基本思想是每帧的长度动态变化, 其值等于每帧中待识别的标签数[6−9], 以使得识别效率能达到最优值. 然而, 当捕获效应发生时, 这样的条件并不能使识别效率达到最优, 因为此时的识别效率不仅与标签数相关, 还与捕获效应的概率相关[24], 而捕获效应的概率却难以获得, 它与识别环境有关, 例如标签的远近以及发射信号的强弱[25].

本文将我们先前的工作 — 自适应 BTSA (Binary tree slotted aloha, BTSA) 协议[17] 扩展到捕获效应的环境下, 提出一种新的自适应 BTSA 协议 (Adaptive binary tree slotted aloha, ABTSA). 该协议首先采用 EPC C1 Gen2 标准中的 RN16 序列[21] 来检测隐藏标签, 然后, 采用一种分配策略来减少隐藏标签和其他未识别标签间的冲突, 最后, 再利用一种自适应技术调整帧长, 以获得捕获效应下较高的识别效率. 本文的贡献在于: 1) 将原先的工作扩展到捕获效应环境下, 提出了一种新的防冲突协议, 该协议优于现存的其他协议. 2) 提出了一种在捕获效应下的分配策略, 它可减少隐藏标签和其他未识别标签的冲突, 从而提高识别效率.

1　捕获效应问题

对于捕获效应, 至少有两个问题值得考虑. 第一个是如何检测被捕获效应隐藏的标签. 在一个发生捕获效应的时隙中只有一个标签能够被识别, 而传统的防冲突协议认为它是可读时隙, 因此, 会漏读其他[4]. 为检测隐藏标签, 阅读器将解码的标签 ID 返回给该时隙的所有标签, 然后, 标签把返回的 ID 和自己的 ID 进行比较, 相匹配的标签为被识别标签, 而不匹配的则是隐藏标签. 由于文献 [4, 23] 用该方法有效地解决了隐藏标签的检测, 因此, 本文不再进行过多讨论.

另外一个问题是捕获效应下的识别效率. 识别效率定义为成功识别的时隙数与总的时隙数比值, 越少的冲突时隙数将得到越高的效率. 由于以下原因, 有无捕获效应下的识别效率会有所不同. 1) 捕获效应使原先假定为冲突的时隙变为成功时隙, 因此识别效率得到了提高; 2) 被捕获效应隐藏的标签会与其他未识别标签冲突. 由于隐藏标签还要继续到下一个识别循环中选择时隙, 当其他未识别的标签也选择了同一个时隙, 两者之间的冲突不可避免, 因此, 合理分配时隙给这两类标签将有助于减少冲突时隙、提高识别效率.

2　相关工作

GBT (General binary tree) 协议[4] 是针对捕获效应而提出的一种二进制树 (Binary tree, BT) 类协议[11], 该协议解决了传统 BT 类协议无法检测隐藏标签的问题, 从而防止了隐藏标签被漏读. 然而在 GBT 协议中, 所有隐藏标签在下一循环中都是选择同一个时隙, 如图 1, 当隐藏标签数量比较大时, 冲突时隙数也会增加.

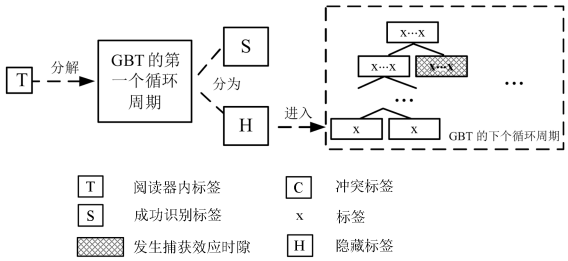

图 1　GBT 的标签识别示意图

Fig. 1　Tag identification in GBT with capture effect

Q 算法[21] 是 EPC C1 Gen2 标准中的防冲突算法, 在该算法中帧长会自动调整以保证一个合理值, 若当前帧出现了过多的冲突时隙, 阅读器会提前结束该帧, 而重新开启一个更长的帧; 相反, 若出现过多的空时隙, 会开启一个更短的帧. 但是, 当捕获效应发生时, Q 算法把当前帧中的隐藏标签和冲突标签全部放到下一个帧中继续识别, 如图 2 所示, 这可能造成这两者在下一个帧中产生冲突, 因此, Q 算法的识别效率也不能达到较优值.

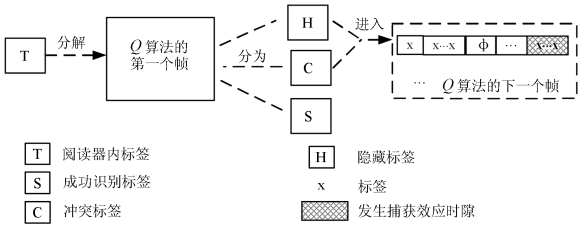

图 2　Q 算法的标签识别示意图

Fig. 2　Tag identification in Q algorithm with capture effect

最优 Q 算法[19] 假定标签数是已知的, 然后, 让帧长等于标签数, 这能保证无捕获效应下的识别效率能到最优. 然而, 当捕获效应发生时, 最优效率却无法得到保证, 因为此时的帧长不仅与标签数还与捕获效应发生的概率相关[24].

我们曾在文献 [17] 中提出一种标签防冲突协议 — 自适应 BTSA 协议, 该协议的优点在于, 无需标签估计, 而且其识别效率不会受标签数变化的影响. 然而, 同大多数已存在的协议一样, 自适应 BTSA 协议没有考虑捕获效应, 它会产生标签漏读. 表 1 给出了识别率, 定义为

$$R = \frac{n_{\text{read}}}{n} \times 100\,\% \tag{1}$$

其中, n 为待识别的标签数, n_{read} 是被识别的标签数. 从表 1 可以看到, 随着捕获效应的概率增大, 漏读的标签数也在增大. 由于这样的原因, 自适应 BTSA 协议难以用到实际工程中, 例如在商场的结算处, 如果有商品的标签被漏读, 那么顾客就不需要付钱购买这些商品, 那么商家将蒙受损失.

表 1 适应的 BTSA 协议标签识别率

Table 1 Identification rate of adaptive BTSA

发生概率	0	0.2	0.4	0.6	0.8	1.0
识别率 (%)	100	75	53	38	17	3

在本文中, 我们把自适应 BTSA 扩展到捕获效应环境下, 提出了 ABTS 协议. ABTS 协议可以有效检测隐藏标签, 防止标签漏读, 而且 ABTS 协议为进一步提高效率, 还解决了如何合理地把时隙分配给隐藏标签和其他未识别标签的问题.

3 协议设计

3.1 捕获效应下的 RN16 序列

本协议首先利用 RN16 序列去解决捕获效应下的标签冲突问题, RN16 序列[21] 源于 EPC C1 Gen2, 该标准对 RN16 做了详细规定, 它由标签随机产生, 其位长应短于 ID 序列, 例如 ID 序列位长为 96 bit, 随机产生的 RN16 可选 16 bit. 图 3 给出了采用 RN16 和未采用 RN16 的设计对比图, 识别开始时, 标签先发送较短的 RN16: 1) 若发生冲突, 则不用较长的 ID 序列以避免较长的冲突时间; 2) 若信道空闲, 阅读器只需等待较短的 RN16 的发送时间就可避免较长的信道空闲时间; 3) 若成功, 阅读器发回成功接收到的 RN16 和确认 (Acknowledgement, ACK) 帧, 然后, 标签再发送其长 ID 序列以供识别, 此时仍然少于标签直接向阅读器发送长 ID 序列后, 阅读器再返回确认帧的通信时间.

利用 RN16, ABTS 可以成功地检测隐藏标签. 当多个标签同时向阅读器发送 RN16 时, 捕获效应可以让阅读器成功地解码出其中一个标签的 RN16. 根据上文的情况 3), 阅读器把解码的 RN16 返回给标签, 那样隐藏的标签将会发现其自身发送的 RN16 与返回的 RN16 不匹配, 从而被检测出. 在 GBT 协议中, 阅读器用返回的 ID 序列, 而不是 RN16 来检测隐藏标签, 如图 3 (a). 由于 RN16 的位长短于 ID, 因此, 采用 RN16 序列的方法将会花费较短的通信时间.

3.2 捕获效应下的分配策略

ABTS 的分配策略包括两部分, 分别是隐藏标签的分配和冲突标签的分配. 同传统的帧时隙 ALOHA 协议一样, ABTS 把标签的识别过程分为若干个帧, 标签随机选择一个帧中的某个时隙向阅读器发送 RN16 序列. 在隐藏标签的分配中, 若一个标签被捕获效应所隐藏, 那么阅读器将让这个标签进入到下一个帧中再被识别, 而不是当前帧中剩下的时隙, 这就避免了隐藏标签不会在当前帧中与其他未识别标签再次发生冲突. 在冲突标签的分配中, 只要冲突产生, 冲突标签将立即进行 BT 分解, 因此被分解的标签将不会进入到下一个帧中与隐藏标签发生冲突.

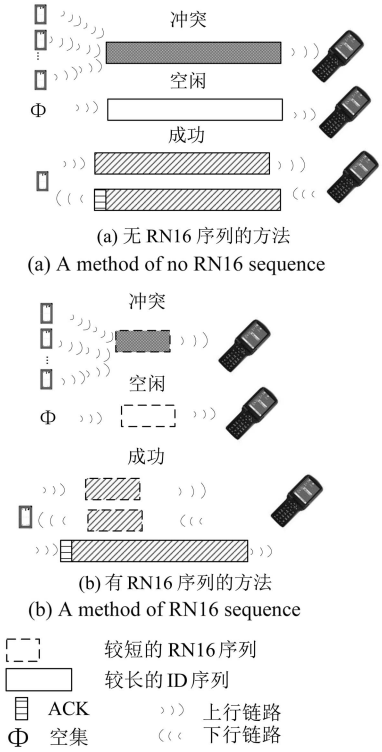

图 3 采用 RN16 和未采用 RN16 的设计对比图

Fig. 3 Comparison between a method of RN16 and no RN16

图 4 给出了一个分配策略的例子, 在第 i 帧中, A、B、C 和 D 分别选择时隙发送其 RN16 序列, 此时 A 和 B 的 RN16 发生冲突, 则立即进行二进制树分解, 而 D 为隐藏标签, 将进入到第 i+1 个帧中. 由于 A 和 B 不会进入第 i+1 个帧, 因此, D 也不可能与 A、B 发生冲突.

3.3 自适应方案设计

ABTS 的自适应方案设计包括调整帧长的自适应算法和标签定位的自适应算法. 调整帧长的自适应算法采用逐个时隙的方式调整帧长, 使得帧长与未识别的标签数始终保持一定的比例关系, 以此保证较高的系统效率, 而标签定位的自适应算法让标签自行选择时隙, 以避免信道的冲突与空闲. 现对这两个算法分述如下:

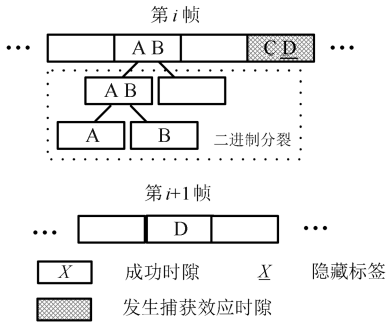

图 4　捕获效应下的分配策略示例

Fig. 4　An example of allocation strategy for capture effect

1) 帧长调整的自适应算法. 调整帧长的自适应算法基于 Q 算法, 当一个帧中出现过多的冲突时隙时, 阅读器提前结束该帧, 发送一个新的较长帧; 当一个帧中出现过多的空闲时隙时, 阅读器也提前结束该帧, 发送一个新的较短帧. 若一个帧长为 $L = 2^Q$, 在第 k 个时隙的 $Q = \mathrm{Round}(Q_{fp}(k))$ (Q_{fp} 是 Q 的小数值, 即 Q 值经 Q_{fp} 四舍五入得到), 那该自适应算法可表示为

$$Q_{fp}(k+1) = Q_{fp}(k) + cg(k) \tag{2}$$

其中, 步长 $0.1 \le c \le 0.5$, 且 Q 值大时, c 值取小, Q 值小时, c 值取大,

$$g(k) = \begin{cases} 1, & 若\ slot_k = C \\ -1, & 若\ slot_k = I \\ 0, & 若\ slot_k = S \end{cases} \tag{3}$$

其中, C, I 和 S 分别代表第 k 个时隙 $slot_k$ 的状态为冲突、空闲和成功. 阅读器根据 $\mathrm{Round}(Q_{fp}(k))$ 是否等于 Q 来决定是否重新开始一个帧, 只要 $\mathrm{Round}(Q_{fp}(k))$ 不等于 Q, 那么阅读器就重新开始一个帧, 而该帧长由新的 Q 值所确定.

ABTS 的自适应算法与 Q 算法最大不同之处在于, 嵌入了第 3.2 节中的二进制树分配策略, 当 $\mathrm{Round}(Q_{fp}(k)) = Q$ 时, 冲突的标签采用二进制树进行冲突分解, 直到冲突的标签分解完成后再进行下一个时隙的标签识别.

2) 标签定位的自适应算法. ABTS 协议的标签定位信息通过自适应算法获取, 该定位信息其实就是标签中选择何时隙向阅读器发送 ID. 标签定位的自适应算法利用 RN16 实现, 标签先向阅读器发送 RN16, 在阅读器端 RN16 将发生冲突、成功和空这三种情况, 然后, 阅读器再把这一情况反馈给标签, 标签根据反馈信息由自适应算法获取定位信息. 标签的定位信息可放在标签的计数器中, 标签的计数器初始值为 0 到帧长 $L-1$ 之间的一个数, 当标签的计数器值变化为 0 时, 标签就发送其 RN16, 当计数器值小于 0 时, 计数器将不再工作. 假定某个标签在第 k 个时隙的计数器值为 $Counter(k)$, 那该自适应算法可表示为

$$Counter(k+1) = Counter(k) + c'g'(k) \tag{4}$$

其中, 步长 $c' = 1$, $g'(k)$ 的取值规定为, 当 $Counter(k) > 0$ 时,

$$g'(k) = \begin{cases} 1, & 若\ slot_k = C \\ -1, & 若\ slot_k = I\ 或\ S \end{cases} \tag{5}$$

当 $Count(k) = 0$ 时,

$$g'(k) = \begin{cases} 0\ 或\ 1 & 若\ slot_k = C \\ -1, & 若\ slot_k = S \end{cases} \tag{6}$$

无论标签初始状态的计数器值为何值, 经式 (4) ~ (6) 的计算, 每个标签都会最终定位到一个时隙, 在这个时隙中将不会发生冲突. 表 2 给出了标签定位的自适应算法的一个例子, 假定有三个标签 A、B 和 C, 其初始计数器值为 1, 1 和 3, 经式 (4) ~ (6) 的自适应算法后, C、B 和 A 将分别定位到时隙 2、3 和 5 后被识别.

表 2　适应的 BTSA 协议标签识别率

Table 2　Identification rate of adaptive BTSA

时隙	标签: *counter* 值			时隙类型
$k=0$	A: 1	B: 1	C: 3	空
$k=1$	A: 0	B: 0	C: 2	冲突
$k=2$	A: 1	B: 0	C: 3	成功
$k=3$	A: 0	B: /	C: 2	成功
$k=4$	A: /	B: /	C: 1	空
$k=5$	A: /	B: /	C: 0	成功

"/" 代表标签的计数器不再工作

4　协议描述

4.1　协议概述

ABTS 协议的设计依据 RN16、捕获效应的分配策略和自适应方案进行设计. 在 ABTS 协议中, 标签先选择时隙发送 RN16, 然后, 阅读器通过接收到的 RN16 自适应地调整帧长, 而标签通过自适应算法对其计数器值进行运算以完成标签的时隙定位, 定位完成后, 标签在定位的时隙中发送其 ID 序列. 同时, 整个协议还把捕获效应下的分配策略嵌入其中.

4.2 阅读器和标签操作

整个协议由阅读器操作和标签操作两部分构成, 下面分别给出了阅读器和标签的操作伪代码. 阅读器的操作主要由函数 ABTS() 所完成. 一个帧的帧长设为 2^Q, 初始时 $Q = Q_0$. 在 ABTS() 函数中, 有一个时隙计数器 SC, SC 在帧开始时被置为 0, 以后每经过一个时隙就自加 1, 当 SC 超过一个帧的帧长时, 阅读器将结束该帧. 在每一个帧开始时, 阅读器会发送一个带有 2^Q 的 Query 命令, 假如阅读器要提前结束一个当前帧而开始一个新帧, 阅读器再发送一个带有新的 2^Q 的 Query 命令.

ABTS 协议操作伪代码

阅读器操作

```
Q = Q_0
ABTS(Q)
function ABTS(Q)
  SC = 0, Q_fp = Q and broadcast Query with 2^Q
  //Query command originating from EPC initiates a new frame
  do{Receive tag response
      if RN16 collision, Q_fp = min(Q_fp+C, l5)
          if Round(Q_fp)! = Q, break; else transmit QueryBT and BT()
//QeuryBT command instructs tags to be re solved by BT
        elseif no tag response, Q_fp = max(Q_fp − C, 0)
            if Round(Q_fp)! = Q, break; else transmit QueryRep
//QueryRep command originating from EPC instructs tags to
//decrement their counters
        elseif RN16 decoded, transmit ACK with RN16
            if received ID, store ID and trans mit QuervRep
            SC = SC + 1} while SC < 2^Q
  if number of idle slots = 2^Q, return; else ABTS (Round(Q_fp))
function BT()
  B = 2
  do {Receive tag response and detect a collision
    if RN16 collision, transmit QuervBT and B = B + 1
    elseif no tag response, transmit QueryRep and B = B − 1
    elseif RN16 decoded, transmit ACK with RNl6, B = B − 1
      if received ID, store ID and transmit QuervRep
  } while B > 0
```

标签操作

```
Receive Reader's Query with 2^Q and TagQIdentify(2^Q)
function TagQIdentify (2^Q)
  Counter = a random number from 0 to 2^Q − 1
  while Counter >= 0
  do {if Counter=0, transmit RN16
          if receive Query with 2^Q, break
          elseif receive ACK with RN16 from reader
              if received RN16=tag's own RN16,
                  Counter = Counter− 1 and transmit ID
              else wait until reader's Query with 2^Q, break
          elseif receive QueryBT from reader
              Counter = Counter+ a random number 0 or 1
      elseif Counter! = 0
          if receive Query with 2^Q, break
          elseif receive QueryBT from reader
              Counter = Counter + l
          elseif receive QueryRep form reader
              Counter = Counter − 1}
  if Counter < 0 return; else TagQIdentify(2^Q)
```

阅读器根据接收到的 RN16 的情况可知当前时隙是否为空、可读和冲突, 然后, 据此调整帧长. 帧长的调整通过调整 Q_{fp} 的值, 具体计算由式 (1) 和 (2) 确定. 无论何时只要 Round(Q_{fp}) 不等于 Q, 阅读器将提前终止该帧. 如果 Round(Q_{fp}) = Q, 阅读器将发送不同的命令反馈给标签当前时隙为何时隙: 1) 假如是空时隙, 发送 QueryRep, 表示开始下一个时隙; 2) 假如是成功时隙, 发送带有 RN16 的 ACK, 在成功接收到一个标签的 ID 后, 再发送一个 QueryRep 来表示下一个时隙开始; 3) 假如是冲突时隙, 发送 QueryBT, 表示冲突标签立即进入二进制树分解. 阅读器采用 BT() 函数来执行 BT 分解. 首先, 让一变量 B 的初始值为 2, 若当前时隙为冲突时隙, $B = B + 1$, 若为非冲突时隙, $B = B - 1$. 当 $B = 0$ 时, 阅读器将结束当前的二进制树分解.

另外, 当一个帧结束时, 并且空时隙数等于这个帧的帧长数, 那么整个识别结束; 否则, 将进入下一个 ABTS 的循环.

标签操作由 TagQIdentify() 函数来执行. 在 TagQIdentify() 函数中, 标签有一计数器 $Counter$, 其可存放标签的定位信息, 标签的初始 $Counter$ 将

从 0 到帧长 2^Q 之间随机选择一个值.

假如标签的 *Counter* 选择 0, 根据阅读器发送的命令可分为三种操作: 1) 若接收到的是带有 2^Q 的 Query 命令, 标签将进入下一个帧中, 当前帧将提前结束; 2) 若收到的是带有 RN16 的 ACK 命令, 标签将判断该 RN16 是否与自身的匹配, 如果匹配, 该标签将发送 ID, 然后, 其计数器自减 1, 并且在后面的帧中均不再工作; 否则, 该标签为被捕获效应隐藏的标签, 将进入下个帧中识别; 3) 若收到的是 QueryBT 命令, 标签将进行二进制分解, 此时的 *Counter* 将加上一个随机二进制数 0 或 1.

假如标签的初始 *Counter* 不选择 0, 标签的操作也分为二种: 1) 若接收到的是带有 2^Q 的 Query 命令, 标签将进入下一个帧中, 当前帧将提前结束; 2) 若收到的是 QueryBT 命令, 标签的 *Counter* 将自加 1; 3) 若收到的是 QueryRep 命令, 标签的 *Counter* 将自减 1.

4.3　协议举例

图 5 给出了 ABTS 协议识别标签 A、B、C 和 D 的例子, 其中步长 $c = 0.5$, $Q_0 = 2.0$. 我们假定标签 A、B、C 和 D 的 *Counter* 分别为 1、1、3 和 3, 那么在第一个帧的第二个时隙, 标签 A 和 B 冲突, 由于 $\mathrm{Round}(Q_{fp}) = Q$, 因此, 冲突的标签立即进行二进制树分解, 而在分解中, A 和 B 又同时选择一个时隙, 但是发生捕获效应, 因此, A 的 RN16 被成功接收, 而标签 B 为被隐藏标签, 其计数器处于等待状态直到开启一个新帧. 在第一帧中的最后个时隙也发生捕获效应, C 的 RN16 被成功接收. 在第二帧的第一个时隙为空, 此时 $\mathrm{Round}(Q_{fp}) \neq Q$, 因此, 该帧的剩余时隙被终止. 在第三帧中, Q 由 2.0 变为 1.0, B 和 D 的 RN16 被成功接收. 在第 4 帧中, 所有时隙为空, 表明标签识别结束. 表 3 给出了该例子中标签 *Counter* 变化的过程.

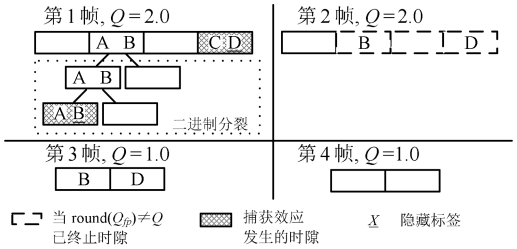

图 5　一个周期内 ABTS 协议识别标签示例

Fig. 5　An example of ABTS protocol identifying tags in a cycle

5　性能分析

在本节中, 我们分析 ABTS 协议中识别标签所花费的时隙数. 令 T 为阅读器在一个识别周期内识别所有标签所花费的时隙数和, 那么该时隙数应为发生冲突、空和成功的时隙数、和之和, 即

$$T = T_C + T_I + T_S \tag{7}$$

由于当阅读器识别标签时, 在未发生和发生捕获效应下所花费的时隙数不相同, 因此, 我们将对这两种情况下所花费的时隙数分别进行分析.

表 3　适应的 BTSA 协议标签识别率

Table 3　Identification rate of adaptive BTSA

帧	Q	时隙	Q_{fp}	*Counter* A	*Counter* B	*Counter* C	*Counter* D	反馈
1	2.0	1	1.5	1	1	3	3	空
		2	2.0	0	0	2	2	冲突
		3	2.0	0	0	3	3	冲突
		4	2.0	0	*	4	4	成功
		5	2.0	/	*	3	3	空
		6	2.0	/	*	2	2	空
		7	1.5	/	*	1	1	空
		8	1.5	/	*	0	*	成功
2	2.0	1	1.0	/	1	/	3	空
3	1.0	1	1.0	/	0	/	1	成功
		2	1.0	/	/	/	0	成功
4	1.0	1	0.5	/	/	/	/	空
		2	0	/	/	/	/	空
		3	0	/	/	/	/	空

"*" 代表该标签为隐藏标签, 其计数器不工作直到当前帧结束, "/" 代表标签已被识别, 其计数器不工作

5.1　无捕获效应的识别时隙数

若在帧时隙 ALOHA 中有 n 个标签在一个帧长为 L 的帧中随机选择时隙, 那么有 r 个标签选择同一时隙的个数的数学期望可表示为

$$\mathrm{E}(L,n,r) = L\begin{pmatrix} n \\ r \end{pmatrix}\left(\frac{1}{L}\right)^r\left(1-\frac{1}{L}\right)^{n-r} \tag{8}$$

此时, 在帧时隙 Aloha 中嵌入二进制树分解策略所花费的总时隙数可表示为

$$T_{BTA}(L,n) = L + \sum_{r=2}^{n}\mathrm{E}(L,n,r)T_{BT}(r) \tag{9}$$

其中, $T_{BT}(r)$ 表示用二进制树算法识别 r 个标签所花费的总时隙数, 表示为

$$T_{BT}(n)=\begin{cases}4, & n=2\\ \dfrac{2+\sum\limits_{r=2}^{n-1}\begin{pmatrix}n\\r\end{pmatrix}0.5^{n-1}T_{BT}(r)}{1-0.5^{n-1}}, & n>2\end{cases}\tag{10}$$

在 ABTS 协议中, 由于帧长逐个时隙动态调整, 当冲突时隙或空时隙过多时会提前结束一个帧, 因此帧长并不是固定值, 假定用平均帧长 L_{Av} 代替, 那么 ABTS 协议所花费的总时隙数可近似表示为

$$T_{BTA}(L_{Av},n)\approx L_{Av}+\sum_{r=2}^{n}\mathrm{E}(L_{Av},n,r)T_{BT}(r)\tag{11}$$

由文献 [12] 可知, 当帧长 L_{Av} 与识别标签数 n 相等时, 时隙数最 $T_{BTA}(L,n)$ 小, 此时, 所花费的时隙数为最优. 由于

$$\mathrm{E}(L_{Av},n,r)|_{L_{Av}=n,L_{Av}\to+\infty}=\frac{L_{Av}}{r!e}\tag{12}$$

因此, ABTS 协议识别 n 个标签所花费的最优时隙数为

$$T_{ABTA}(L_{Av},n)|_{L_{Av}=n,L_{Av}\to+\infty}\approx L_{Av}+L_{Av}\times\sum_{r=2}^{n}\frac{T_{BT}(r)}{r!e}\approx 2\tag{13}$$

5.2 捕获效应下的识别时隙数

我们先来推导在捕获效应发生的概率为 p 下, 采用二进制树识别 n 个标签所花费的时隙数. 当 $n=2$, 有:

$$T'_{BT}(2)=2+(1-p)\mathrm{E}(2,2,2)T'_{BT}(2)\tag{14}$$

把式 (8) 代入式 (14), 可得:

$$T'_{BT}(n)=\frac{4}{1+p},\qquad n=2\tag{15}$$

当 $n>2$ 时, 有:

$$T'_{BT}(n)=2+(1-p)\mathrm{E}(2,n,n)T'_{BT}(n)+(1-p)\times\sum_{r=2}^{n-1}\mathrm{E}(2,n,r)T'_{BT}(n)\tag{16}$$

把式 (8) 代入式 (16), 可得:

$$T'_{BT}(n)=\frac{2+(1-p)\sum\limits_{r=2}^{n-1}\begin{pmatrix}n\\r\end{pmatrix}0.5^{n-1}T'_{BT}(r)}{1-(1-p)0.5^{n-1}},\qquad n>2\tag{17}$$

由式 (15) 和 (17) 可知, 采用二进制树花费的时隙数可用迭代方法求的. 同样, 若把平均帧长 L_{Av} 代替 L, 在捕获效应发生的概率为 P 下, ABTS 协议识别 n 个标签所花费的时隙数为

$$T'_{ABTA}(L_{Av},n)\approx L_{Av}+(1-p)\sum_{r=2}^{n}E(L_{Av},n,r)T'_{BT}(r)\tag{18}$$

式 (18) 表示的仅仅是, ABTS 协议识别未被捕获效应隐藏的标签所花费的时隙数, 而要将所有标签识别完, 还需要对隐藏的标签再次识别, 此时所花费的时隙数可同样用式 (18) 进行计算.

6 计算机仿真结果

6.1 仿真系统设置

本部分用计算机仿真的实验结果验证本文提出的 ABTS 协议, 在仿真实验中, 我们考虑一个阅读器识别多个标签的情况, 并且我们假定在识别过程中标签不会离开阅读器的磁场范围, 也不会有新的标签进入阅读器磁场. 为了评价标签识别的性能, 我们考虑以下指标:

1) 总时隙数: 观测在一个识别周期内, 阅读器成功识别完所有标签所花费的时隙数, 较少的时隙数将得到较短的识别时间.

2) 识别效率: 观测在一个识别周期内的识别效率, 它定义为一个周期内成功时隙数与总的时隙数的比值, 越少的空时隙数和冲突时隙数将得到越高的识别效率.

在仿真中, 我们对每个实验重复执行 500 次, 最后的实验结果为 500 次结果的平均值. 我们把所有仿真实验分成两部分, 第 6.1 节我们把 ABTS 协议与 Q 算法、GBT 协议和 Optimal Q 算法进行对比, 这些协议的参数如下:

1) Q 算法: 根据 EPC C1 Gen2[21] 的标准, Q 的初始值 Q_0 取 4.0, 步长 $0.1\le c\le 0.5$. 因此, 步长 c 采用以下方法选取: 若 $0\le Q\le 2$, $c=0.5$; 若 $Q\ge 10$, $c=0.1$; 否则, $c=1/Q$.

2) GBT 协议: 初始时, 所有标签的计数器均为 0. 被捕获效应隐藏的标签将进入下一个二进制树循

环[4], 其中, 一个二进制树循环定义为在这个循环中所有冲突时隙均被分解为可读时隙或空时隙.

3) Optimal Q 算法: Q_0 取 4.0, 这与上面的 Q 算法中参数的选择一致. 假定第 n 个帧长的标签数 T_n 已知, 那么由文献 [19] 可知, 第 n 个帧长的 Q_n 为 $\lfloor \log_2 T_n \rfloor$.

4) ABTS 协议: Q_0 同样取 4.0, 这与上面的 Q 算法和 Optimal Q 算法中参数的选择一致. 并且步长 c 选取的方法也与上面的 Q 算法一致, 即, 若 $0 \le Q \le 2$, $c = 0.5$; 若 $Q \ge 10$, $c = 0.1$; 否则, $c = 1/Q$.

在第 6.2 节中, 我们将给出不同 p 和 Q_0 下 ABTS 协议的识别性能, 该仿真的目的是展示 p 和 Q_0 如何影响 ABTS 协议的性能.

6.2　同现存协议的比较

在本节中, 我们把 ABTS 协议与现存协议进行比较. 图 6 给出了当捕获效应发生的概率从 0 变化到 0.6 之间、标签数 $N = 500$ 时, ABTS 协议、Q 算法、GBT 以及 Optimal Q 算法的识别效率. 从图 6 中可以看到, 本文的 ABTS 协议的效率曲线在其余几种协议曲线的上方. 这一结果表明, 当捕获效应发生概率在 0～0.6 之间时, ABTS 协议的识别效率要高于现存的这几种协议, 其原因在于: 1) ABTS 协议允许让隐藏的标签在下一个帧中随机选择时隙, 相反, GBT 协议却只能让所有隐藏标签选择同一个时隙, 这导致了过多的冲突; 2) ABTS 协议中的标签一旦发生冲突将立即进行二进制树分解, 而 Q 算法和 Optimal Q 算法却让隐藏标签和冲突的标签均进入下一个时隙, 这也会产生过多的标签冲突; 3) ABTS 协议采用自适应技术来获得一个合理的帧长, 而 Optimal Q 算法让帧长等于标签数, 这在捕获效应条件下并不是最优帧长.

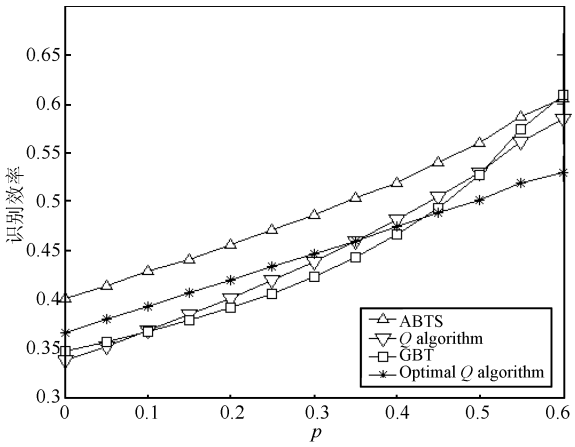

图 6　仿真结果: 识别效率, $0.1 \le p \le 0.6$, $N = 500$

Fig. 6　Simulation results: identification efficiency, $0 \le p \le 0.6$, $N = 500$

当然从图 6 中, 我们还可以看到, 随着 p 的增大, ABTS 和 GBT 的间距在逐渐减小, 当 $p = 0.6$, 这个间距将减小到 0. 可以预见, 随着 $p > 0.6$, GBT 的识别效率将超过 ABTS. 然而, $p > 0.6$ 意味着, 每两个假定冲突的时隙将产生一个捕获效应的时隙, 这种情况并不常见[19], 因此, 本文没有考虑 $p > 0.6$ 的情形. 图 7 还给出了以上协议在捕获效应发生的概率从 0.1 变化到 0.6 之间、标签数 $N = 500$ 时的识别总时隙数, 从中可以看到, ABTS 协议所花费的时隙数少于其余的协议, 这个结果与图 6 也一致, 即越高的识别效率将花费越少的时隙数, 而越少的时隙数将带来越少的识别时间.

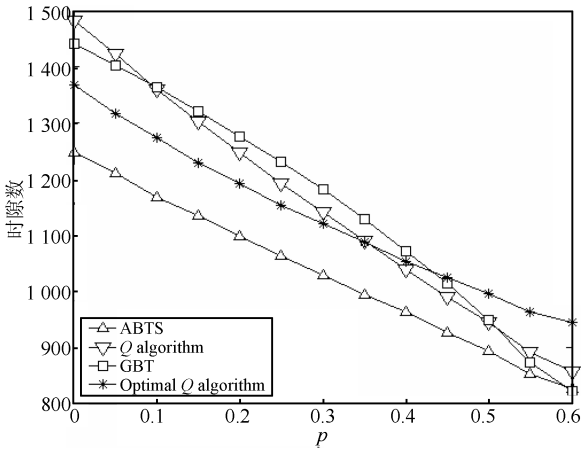

图 7　仿真结果: 识别时隙数, $0.1 \le p \le 0.6$, $N = 500$

Fig. 7　Simulation results: the number of slots, $0 \le p \le 0.6$, $N = 500$

图 8 给出了 $P = 0.4$ 和 $100 \le N \le 4\,000$ 时各协议的识别效率曲线, 这 4 条曲线由上到下分别为: ABTS 协议、Q 算法、Optimal Q 算法以及 GBT. 这一结果表明, 当捕获效应发生概率为 0.4, 且标签数从 100 变化到 4 000 时, ABTS 协议的识别效率均高于现存的这几种协议, 它并不依赖标签数的变化. 另外, 从图 8 中, 我们还可以看到, 当标签从 100 变化到 4 000 时, ABTS 协议的识别效率仅从约 0.51 变化到 0.53, 这也说明 ABTS 协议的识别效率的稳定性较好, 它并不会因为标签数的剧烈变化产生效率值的剧烈变化. 图 9 还给出了 $p = 0.4$ 和 $100 \le N \le 4\,000$ 时的识别总时隙数. 从图 9 中可以看到, ABTS 协议比其他协议花费了更少的时隙数, 这个结果也与图 8 一致.

6.3　p 和 Q_0 值的影响

图 10 给出了 ABTS 协议在 $p = 0$、0.2、0.4、0.6 和 0.8 时, 标签数从 0 变化到 1 000 时的识别效率变化曲线. 从图 10 中可以看出, 当标签数大于 100 时, 这 4 条曲线接近水平, 分别稳定在 0.4、0.45、0.5、0.6

和 0.7. 这个结果反映了, 无论标签数是大还是小, ABTS 的识别效率曲线均未发生明显抖动, 具有较好的稳定性.

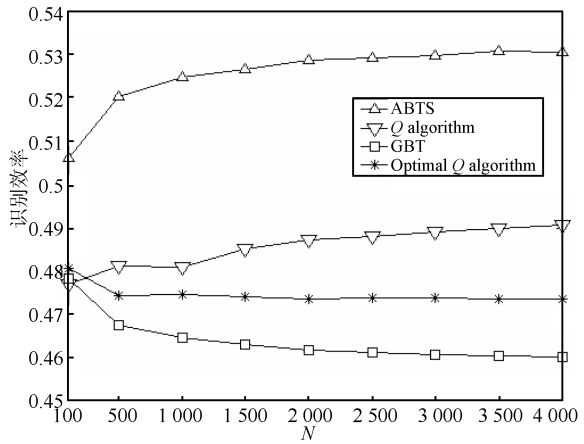

图 8 仿真结果: 识别效率, $100 \le N \le 4\,000$, $p = 0.4$

Fig. 8 Simulation results: identification efficiency, $100 \le N \le 4\,000$, $p = 0.4$

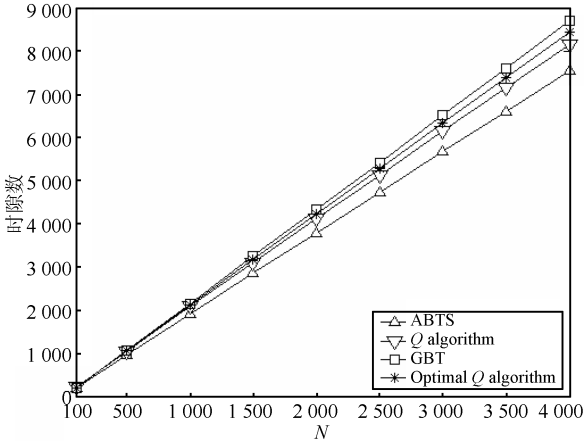

图 9 仿真结果: 识别时隙数, $100 \le N \le 4\,000$, $p = 0.4$

Fig. 9 Simulation results: the number of slots, $100 \le N \le 4\,000$, $p = 0.4$

图 11 分别给出了 ABTS 协议在标签数 100～1 000 之间变化时, 不同初始 Q_0 值下的系统效率曲线. 从图 11 中可以看到, 当 Q_0 值大于等于 8, 识别标签的效率在标签小于 100 时较低, 不到 0.4, 只有标签大于 100 时, 识别标签的吞吐量才逐渐增大, 超过了 0.4. 当 Q_0 值小于等于 2 时, 识别标签的吞吐量在标签小于 100 时有较高的值, 接近 0.40, 而标签大于 100 时, 识别标签的吞吐量却不到 0.40. 因此, 从以上结果可以看到, 取 Q_0 初值为 4.0 是一个折衷的结果, 能够确保标签数较大和较小时, 识别效率均不会得到一个较低值.

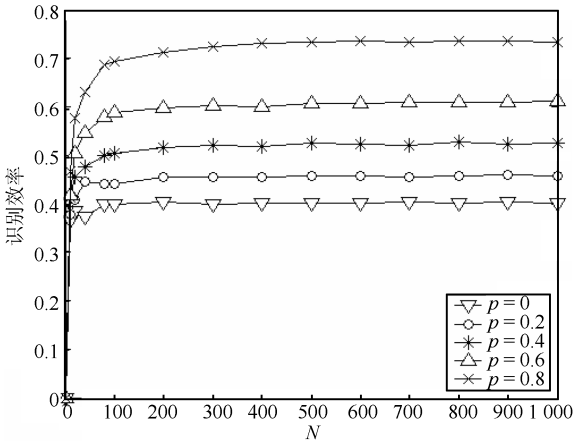

图 10 在不同 p 下的 ABTS 识别效率, $Q_0 = 4.0$

Fig. 10 Simulation results: ABTS efficiency of different p under $Q_0 = 4.0$

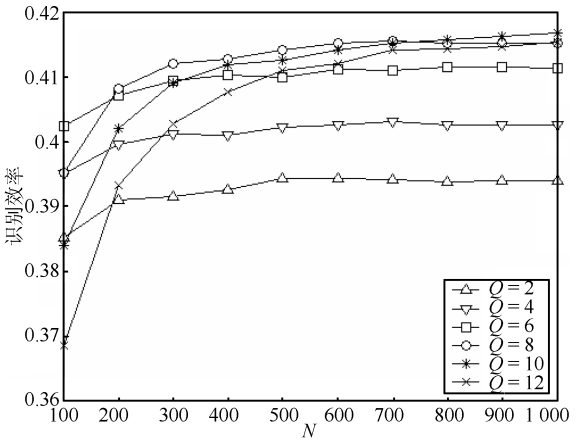

图 11 在不同 Q_0 下的 ABTS 识别效率, $100 \le N \le 1\,000$

Fig. 11 Simulation results: ABTS efficiency under different Q_0, $100 \le N \le 1\,000$

7 结论

本文提出 ABTS 协议在捕获效应发生概率小于 0.6 时, 其识别效率和识别时间性能要优于现存的 Q 算法、GBT 和 Optimal Q 算法. 虽然当捕获效应概率大于 0.6 时, ABTS 协议的效率并不比现存的协议高, 然而, 概率大于 0.6 在实际应用中并不常见, 因此, 本文没有考虑这种情况. 另外, ABTS 协议的识别效率在标签数少和多的情况下均有较好的稳定性. 最后, 相比于其他初始值, Q_0 的初始值取 4.0 可保证 ABTS 在较少和较多的标签数下均有不低的识别效率.

References

1 Welbourne E, Battle L, Cole G, Gould K, Rector K, Raymer S, Balazinska M, Borriello G. Building the internet of things using RFID: the RFID ecosystem experience. *IEEE Internet Computing*, 2009, **13**(3): 48−55

2 Finkenzeller K. *RFID Handbook: Radio-frequency Identification Fundamentals and Applications* (2nd edition). England: John Wiley and Sons, 2003

3 Shin D H, Sun P L, Yen D C, Huang S M. Taxonomy and survey of RFID anti-collision protocols. *Computer Communications*, 2006, **29**(11): 2150−2166

4 Lai Y C, Hsiao L Y. General binary tree protocol for coping with the capture effect in RFID tag identification. *IEEE Communications Letters*, 2010, **14**(3): 208−210

5 Schoute F C. Dynamic frame length ALOHA. *IEEE Transactions on Communications*, 1983, **31**(4): 565−568

6 Vogt H. Efficient object identification with passive RFID tags. In: Proceedings of the 1st International Conference on Pervasive Computing. Berlin Heidelberg, Germany: Springer-Verlag, 2002. 98−113

7 Lee S R, Joo S D, Lee C W. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification. In: Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. San Diego, CA, USA: IEEE, 2005. 166−172

8 Chen W T. An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA. *IEEE Transactions on Automation Science and Engineering*, 2009, **6**(1): 9−15

9 Wu Hai-Feng, Zeng Yu. Tag estimate and fame length for dynamic frame slotted ALOHA anti-collision RFID system. *Acta Automatica Sinica*, 2010, **36**(4): 620−624
(吴海锋, 曾玉. RFID 动态帧时隙 ALOHA 防冲突中的标签估计和帧长确定. 自动化学报, 2010, **36**(4): 620−624)

10 Bonuccelli M A, Lonetti F, Martelli F. Tree slotted ALOHA: a new protocol for tag identification in RFID networks. In: Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks. Buffalo-Niagara Falls, NY: IEEE, 2006. 1−6

11 Wu H, Zeng Y. Efficient framed slotted ALPHA protocol for RFID tag anticollision. *IEEE Transactions on Automation Science and Engineering*, 2011, **8**(3): 581−588

12 Capetanakis J I. Tree algorithms for packet broadcast channels. *IEEE Transactions on Information Theory*, 1979, **25**(5): 505−515

13 Hush D R, Wood C. Analysis of tree algorithms for RFID arbitration. In: Proceedings of the 1988 IEEE International Symposium on Information Theory. Cambridge, USA: IEEE, 1988. 107

14 Feng Bo, Li Jin-Tao, Zheng Wei-Min, Zhang Ping, Ding Zhen-Hua. A novel anti-collision algorithm for tag identification in RFID systems. *Acta Automatica Sinica*, 2008, **34**(6): 632−638
(冯波, 李锦涛, 郑为民, 张平, 丁振华. 一种新的 RFID 标签识别防冲突算法. 自动化学报, 2008, **34**(6): 632−638)

15 Myung J, Lee W, Srivastava J, Shih T K. Tag-splitting: adaptive collision arbitration protocols for RFID tag identification. *EEE Transactions on Parallel and Distributed Systems*, 2007, **18**(6): 763−775

16 Qian C, Ngan H, Liu Y H, Ni L M. Cardinality estimation for large-scale RFID systems. *IEEE Transactions on Parallel and Distributed Systems*, 2011, **22**(9): 1441−1454

17 Wu H F, Zeng Y, Feng J H, Gu Y. Binary tree slotted ALOHA for passive RFID tag anticollision. *IEEE Transactions on Parallel and Distributed Systems*, 2013, **24**(1): 19−31

18 La Porta T F, Maselli G, Petrioli C. Anticollision protocols for single-reader RFID systems: temporal analysis and optimization. *IEEE Transactions on Mobile Computing*, 2011, **10**(2): 267−279

19 Maguire Y, Pappu R. An optimal *Q*-algorithm for the ISO 18000-6C RFID protocol. *IEEE Transactions on Automation Science and Engineering*, 2009, **6**(1): 16−24

20 Information technology — Radio frequency identification (RFID) for item management — Part 6: Parameters for air interface communications at 860 MHz-960 MHz — Amendment 1: extension with type C and update of types A and B, ISO 18000-6C, 2006

21 EPC radio-frequency identification protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz-960MHz Version 1.1.0 Draft1, EPC C1 Gen2, 2005

22 Khandelwal G, Yener A, Lee K, Serbetli S. ASAP: a MAC protocol for dense and time constrained RFID systems. In: Proceedings of the 2006 IEEE International Conference on Communications. Istanbul, Turkey: IEEE, 2006. 4028−4033

23 Information technology — Radio frequency identification (RFID) for item management — Part 6: Parameters for air interface communications at 860 MHz-960 MHz, ISO 18000-6A/B, 2004

24 Li B, Wang J Y. Efficient anti-collision algorithm utilizing the capture effect for ISO18000-6C RFID protocol. *IEEE Communications Letters*, 2011, **15**(3): 352−354

25 Liu Yu, Zhu Zhi-Yuan, Guan Qiang, Yang Yi-Ping. Research on experimental-design-based RFID application combinatorial testing optimization. *Acta Automatica Sinica*, 2010, **36**(12): 1674−1680
(刘禹, 朱智源, 关强, 杨一平. 基于试验设计的 RFID 应用组合测试优化研究. 自动化学报, 2010, **36**(12): 1674−1680)

吴海锋 云南民族大学教授. 主要研究方向为 RFID 技术.
E-mail: whf5469@gmail.com
(**WU Hai-Feng** Professor at Yunnan Nationalities University. His main research interest is RFID technology.)

曾 玉 云南民族大学讲师. 主要研究方向为 RFID 技术. 本文通信作者.
E-mail: yv.zeng@gmail.com
(**ZENG Yu** Lecturer at Yunnan Nationalities University. Her main research interest is RFID technology. Corresponding author of this paper.)

凌永发 云南民族大学教授. 主要研究方向为无线随机多址系统.
E-mail: yfing73@163.com
(**LIN Yong-Fa** Professor at Yunnan Nationalities University. His main research interest is random-access system.)