

RFID 系统中停留标签的组策略防碰撞算法

王 勇^{*①②} 唐小虎^② 张莉涓^②

^①(西南交通大学物理学院 成都 610031)

^②(西南交通大学信息科学与技术学院 成都 610031)

摘 要: 在射频识别 RFID 系统中, 防碰撞算法对于标签的快速识别, 尤其在移动应用的场景中非常重要。该文针对移动场景中停留标签, 提出了组策略停留标签算法 GSRA(Group Strategy for Remaining tags Algorithm), 分为停留标签识别和新到标签识别两个阶段, 并将停留标签信息分组存放与更新, 从而提高停留标签的识别效率。理论分析证明该算法系统效率仅与标签动态时移动时的迁移速率和静态时的系统效率有关, 而与标签数量无关。仿真结果表明 GSRA 算法在标签迁移速率为 20% 时, 结合冲突树算法 CT, 系统效率可以达到 240%。

关键词: 射频识别; 标签识别; 标签估计; 防碰撞

中图分类号: TP391.45

文献标识码: A

文章编号: 1009-5896(2016)03-0594-06

DOI: 10.11999/JEIT150745

Group Strategy for Remaining Tags in Anti-collision Algorithm for RFID System

WANG Yong^{①②} TANG Xiaohu^② ZHANG Lijuan^②

^①(School of Physical Science and Technology, Southwest Jiaotong University, Chengdu 610031, China)

^②(School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China)

Abstract: In RFID systems, tag anti-collision algorithm is significantly important for fast tag identification, especially in mobile scenarios. A Group Strategy for Remaining tags Algorithm (GSRA) is proposed. It is divided into two phases, which are the identification of remaining tags and the identification of new arriving tags, and stay tag information grouped is stored and updated, so as to improve the identification efficiency of the stay tag. The theoretic analysis shows that the system efficiency only concerns the rate of migration and static system efficiency, and is not related with the number of tags. The simulation result demonstrates that the system efficiency of GSRA algorithm achieve 240% in Collision Tree (CT) algorithm and 20% with respect to the rate of tag migration.

Key words: Radio Frequency IDentification (RFID); Tag identification; Tag estimation; Anti-collision algorithm

1 引言

防碰撞算法要求高效可靠地识别标签, 这对于 RFID 系统工程应用是一个非常现实的问题。在 Reader-Talk-First 模式中, 阅读器首先发出查询命令, 其查询范围内的标签将返回存储的 ID 等信息。因为所有的标签是通过共享信道方式和阅读器通信, 所以不可避免地导致互相干扰(碰撞), 造成多标签传输时系统性能的下降。

防碰撞问题与经典的多址通信相似, 解决方案有树型协议、ALOHA 协议和载波检测多路访问(CSMA)等。然而防碰撞算法极大地受限于标签本身计算能力弱和内存小, 同时无源标签无法检测信道情况, 所以 CSMA 在 RFID 防碰撞算法中无法使

用。目前的防碰撞算法主要集中在基于时分的方式: 树型或者 ALOHA 算法^[1-7]。

在树型协议中, 阅读器发出查询请求, 标签基于 ID 进行响应。该经典的轮询方案在标签数量太多的情况下有延时过大的问题, 此外, 标签的分布及 ID 的长度也极大地影响着系统识别效率。例如传统的 QT 算法虽然简单, 但其最差时间复杂度可达 $n \cdot (k + 2 - \log n)$, 这里 n 为标签数量, k 为标签 ID 位长度^[8]。

对于 ALOHA 协议, 标签响应阅读器以随机时隙方式实现, 不受标签分布和 ID 长度的限制。其也有很多变种^[9], 在这些协议族中, DFSA 是理论研究和实际应用较多的防碰撞算法。然而系统效率有 36.8% 的理论上限^[10], 并且在标签数量未知的情况下, 不能保证所有标签被识别。

对于移动场景: 典型场景为传送带上物品的数

量统计，在每一轮查询中总有上一轮识别过的标签(停留标签)和上一轮中未出现的标签(新到标签)，主流的防碰撞算法是ABS(Adaptive Binary Splitting)和AQS^[11](Adaptive Query Splitting)以及它们的改进算法如CRB(Couple-Resolution Blocking protocol)和ECRB^[12-14](Enhanced Couple-Resolution Blocking protocol)。但是对于停留标签，上述算法同时只能识别两个标签，这是较低效的算法。因为是已经识别过的标签，可以采用高效的算法再次识别它们，由此本文提出了组策略停留标签识别算法(Group Strategy for Remaining tags Algorithm, GSRA)。通过适当设计信息标志的比特长度，比如64位，该算法可以在一个时隙同时识别64个停留标签。GSRA算法效率仅与识别新到达标签的防碰撞算法效率和标签迁移速率有关。如果识别新到达标签的算法系统效率为常数，如CT算法^[15-16]，那么在标签迁移速率为常数的情况下，GSRA算法系统效率也为常数。

2 GSRA 算法

变量：GroupID代表阅读器或者标签侧的组ID。tmpID代表标签的临时ID并且二进制位中仅有1位非零。MarkBit用来控制tmpID的产生及计算停留标

签的数量，例如：标签的ID长度为4位，那么tmpID的长度同为4位长，如表1所示。‘*’星号表示同组中该位为冲突状态。

本文提出组策略GSRA算法识别停留标签，该算法采用对停留标签分组与分配临时ID的方法有效解决了停留标签的识别问题。

表 1 阅读器侧 GroupID 和 MarkBit 值

GroupID	#1	#2	...
MarkBit	****	****	****

GSRA算法由停留标签识别阶段1和新到达标签识别阶段2两个阶段构成，形成识别过程中的1个轮次，其算法阅读器侧流程图如图1所示。

停留标签识别过程阶段1的伪代码如表2和表3所示。首先，阅读器按组GroupID由小到大查询标签，如果标签的GroupID与阅读器所发相等，则立即响应tmpID给阅读器。其次，利用曼彻斯特编码，如果阅读器侧的MaskBit某位出现‘*’，意味着标签处于停留状态，可统计出停留标签的数量。最后，阅读器应顺序更新和保存每组的MarkBit值。

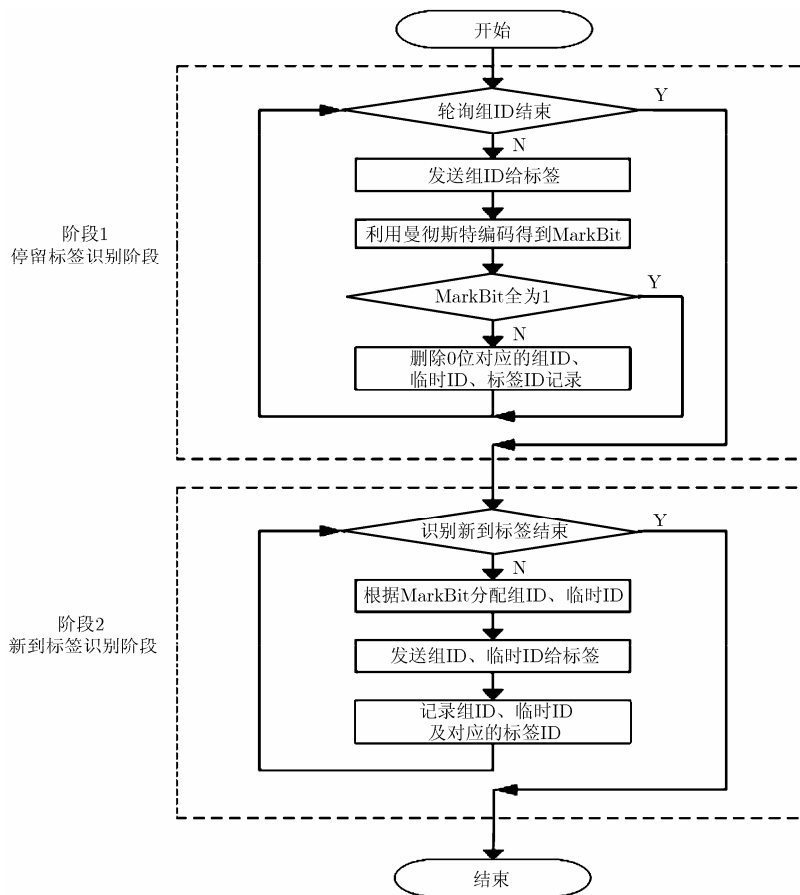


图 1 GSRA算法阅读器侧流程图

表2 阅读器侧识别停留标签算法伪代码

Algorithm1 update table in reader side

```

1: procedure
2: for( $i=0; i < \text{MaxGroupNum}; i++$ )           // MaxGroupNum 为最大存放停留标签的组数量
3:   if( $\text{GroupID} \neq 0$ )                          //如果组 GroupID 不为 0, 则用 send 命令发送 GroupID
4:     send( $\text{GroupID}$ );                          //令该组号的标签发送 tmpID 给阅读器
5:     get( $\text{MarkBit}$ );                          //阅读器利用曼彻斯特码获取到 MarkBit
6:     for( $j=0; j < \text{IDlength}; j++$ )          //对 MarkBit 逐位查询是否为 0
7:       if( $0 == \text{getbit}(\text{MarkBit}, j)$ )      //如果某位为 0, 则表示所对应的标签已离开
8:         delete( $\text{GroupID}, \text{tmpID}, \text{tagID}$ ); //从数据库中删除离开标签的 GroupID, tmpID, tagID
9:       else
10:        count++;                             //统计停留标签的数量
11:     end
12: end
13: end procedure

```

表3 标签侧识别停留标签算法伪代码

Algorithm1 in tag side

```

1: procedure
2: if( $\text{GroupID} == \text{get}(\text{GroupID})$ ) //标签如果收到 GroupID 与
   标签的 GroupID 一致
3:   send( $\text{tmpID}$ ) //则发送 tmpID 给阅读器
4: else
5:   no respond //不一致则不响应
6: end
7: end procedure

```

对于新到达的标签识别过程阶段2的伪代码如表4和表5所示。首先,阅读器维护表4,包括创建与更新。如果标签的GroupID等于0,则意味着是新到标签,应立即响应阅读器。标签被识别后,阅读器分配给其一个GroupID与tmpID,分配规则为:(1)如阅读器侧GroupID为0,则先按组分配序号给GroupID,然后依次从最低位到最高位分配临时ID给tmpID,直到该组可分配的标签数目达到最大值,再顺次在下一组分配临时标签ID;(2)如阅读器侧GroupID不为0,则检查MarkBit是否有零位,如果有则表明该组中有临时ID可分配给新到标签使用,此时应将组号赋给GroupID,同时将该零位对应的ID分配给tmpID,将MarkBit的该位置为*。阅读器侧需要记录标签组GroupID与临时tmpID以及标签的ID。

最后,为了说明GSRA算法的识别过程,我们给了一个简单例子说明该算法的工作流程。假设第1

轮待识别的标签有4个,分别是A(0000), B(0010), C(0011), D(0111),在第2轮,标签E(0001)和F(1111)为新到标签,标签A(0000), D(0111)为离开标签。识别过程如表6所示。

在第1轮(1#),因为没有停留标签,所以阅读器直接进入阶段2。阅读器首先广播查询命令给所有标签,该轮查询开始。一旦接收到这个查询命令,标签开始组GroupID匹配过程。因为所有的标签的组ID等于0,它们均进入活动状态,并开始响应阅读器,这个识别过程可以采用当前的主流算法,如CT或AQS算法等。在阅读器识别一个标签后,将创建一个表,并且分配GroupID和tmpID给这个标签。

(1)查询命令发出后,识别过程自动开始。此时标签A, B, C和D的GroupID为0,所以他们立刻响应阅读器,标签A, B, C和D依次被识别。

(2)标签A被识别后,阅读器分配GroupID=1和tmpID=01给标签A,此时MarkBit=01。对于标签B,阅读器分配GroupID=1和tmpID=10给标签B,此时MarkBit=11。

(3)因为MarkBit长度为2,一个组只能记录2个标签,所以标签C和D分配进第2组。对于标签C,阅读器分配GroupID=2和tmpID=01给标签C。同理,对于标签D,阅读器分配GroupID=2和tmpID=10给标签D。

在第2轮(2#),阅读器通过组GroupID查询停留和离开的标签,一旦接收到GroupID,标签首先做GroupID匹配过程,此时匹配的标签进入活动状态,并且立即反馈tmpID给阅读器。阅读器根据标签的响应更新MarkBit。如果MarkBit的某位为0,则

表4 阅读器识别新到标签伪代码

Algorithm 2 Create table in reader side

```

1: procedure
2: init(); //reset all MarkBit only once           //初始化, 所有 MarkBit 仅复位一次
3: get(tagID);                                     //通过识别算法获取到标签 ID
4: for(i=0;i<MaxGroupNum;i++)                     // MaxGroupNum 为最大存放停留标签的组数量
5:   if(0==GroupID)                               //如果 GroupID 等于 0, 表示该组号没有存放过标签
6:     setbit(MarkBit,0);                         //设置该组的 MarkBit 的第 0 位为 1
7:     GroupID= i;                               //设置组号 GroupID 为 i
8:     tmpID=bitshift(1,0);                       //分配临时 tmpID 最低位为 1
9:     send(GroupID,tmpID,tagID);                 //发送分配的 GroupID, tmpID 给标签 tagID
10:    record(GroupID,tmpID,tagID);                //同时阅读器记录标签 tagID 及分配的 GroupID 和 tmpID
11:   else
12:     for(j=0;j<IDlength;j++)                   //如果 GroupID 不等于 0, 表示该组号存放过标签
13:       if(0==getbit(MarkBit,j))               //但该组的 MarkBit 的第 j 位为 0, 表示该位对应标签已离开
14:         setbit(MarkBit,j);                   //重新置位 MarkBit 的第 j 位为 1
15:         tmpID=bitshift(1,j);                 //分配第 j 位为 1 的 tmpID
16:         send(GroupID,tmpID,tagID);           //发送分配的 GroupID, tmpID 给标签 tagID
17:         record(GroupID,tmpID,tagID);         //同时阅读器记录标签 tagID 及分配的 GroupID 和 tmpID
18:       break;                                  //终止本次 for 循环
19:     end
20:   end
21: end procedure

```

表5 新到标签识别伪代码

Algorithm 2 in tag side

```

1: procedure
2: if(0==GroupID) //如果 GroupID 为 0 表示标签为新到标签
3:   send(tagID); //发送 tagID 给阅读器
4:   set(GroupID,tmpID); //记录阅读器分配的 GroupID 和 tmpID
5: else
6:   no respond //不是新到标签则不响应
7: end
8: end procedure

```

意味着标签已经离开。利用阅读器侧记录的 GroupID 和 tmpID, 可以知道已离开标签的 ID。

(1)在阶段 1, 阅读器按 GroupID 轮询标签状态。第 1 组 MarkBit=10, 可知标签 A 已经离开。第 2 组 MarkBit=10, 可知标签 D 已经离开。

(2)在阶段 2, 在阅读器发出查询命令后, 识别过程自动开始。因为标签 E 与 F 的 GroupID=0, 表明它们为新到标签, 因此立即响应查询命令, 标

签 E 和 F 依次被识别。

(3)识别标签 E 后, 因为第 1 组中 MarkBit=10, 所以阅读器分配 GroupID=1 和改写 MarkBit=11, 并且广播参数 GroupID=1 和 tmpID=01 给标签 E。同理, 对于标签 F, 阅读器分配 GroupID=2 和改写 MarkBit=11, 同时广播参数 GroupID=2 和 tmpID=10 给标签 F。

3 系统效率分析

对于 GSRA 算法, 每轮均使用相同的原则识别在其通信范围内停留与新到达的标签, 不考虑分组计算时延消耗, 分组匹配计算等因素影响。

本文假设标签 ID 均匀分布, 并且长度为 64 位, S_{new} 代表识别新到 $N \cdot \eta_{mob}$ 个标签平均使用的时隙数, 这里 N 为标签总数, 包括新到和停留的标签, η_{mob} 代表标签的迁移比例, η_{new} 表示识别新到标签的效率。

$$S_{new} = \frac{N \cdot \eta_{mob}}{\eta_{new}} \quad (1)$$

令 S_{stay} 代表识别停留标签平均所用时隙 $N \cdot (1 - \eta_{mob})$ 。

表 6 GSRA 算法示例

识别轮次		阅读器		标签		标签 ID
		GroupID	MarkBit	GroupID	tmpID	
1#	阶段 2	1#	01	1#	01	0000
		1#	11	1#	10	0010
		2#	01	2#	01	0011
		2#	11	2#	10	0111
2#	阶段 1	1#	10		leaved	0000
		2#	01		leaved	0111
	阶段 2	1#	11	1#	01	0001
		2#	11	2#	10	1111

$$S_{\text{stay}} = \left\lceil \frac{N \cdot (1 - \eta_{\text{mob}})}{64} \right\rceil \quad (2)$$

所以，识别所有标签 N 的总时隙 S_{total} ：

$$S_{\text{total}} = S_{\text{new}} + S_{\text{stay}} = \frac{N \cdot \eta_{\text{mob}}}{\eta_{\text{new}}} + \left\lceil \frac{N \cdot (1 - \eta_{\text{mob}})}{64} \right\rceil \quad (3)$$

总的系统效率 η_{total} 定义为 N / S_{total} ，有

$$\eta_{\text{total}} = \frac{N}{S_{\text{total}}} = N / \left(\frac{N \cdot \eta_{\text{mob}}}{\eta_{\text{new}}} + \left\lceil \frac{N \cdot (1 - \eta_{\text{mob}})}{64} \right\rceil \right) = 1 / \left(\frac{\eta_{\text{mob}}}{\eta_{\text{new}}} + \left\lceil \frac{(1 - \eta_{\text{mob}})}{64} \right\rceil \right) \quad (4)$$

显而易见， η_{total} 仅与 η_{mob} 和 η_{new} 有关。对于 CT 算法， η_{new} 近似为常数，所以如果 η_{mob} 为常数，那么 η_{total} 一定也为常数，从而不受标签数量的影响。下一节仿真部分进一步证明该结论。

4 性能评估

仿真中，依据 ISO18000-6C(EPC C1G2)标准，设定标签 ID 长度 $k = 64$ 。标签迁移比例 $\eta_{\text{mob}} = 0.2, 0.4, 0.6$ ，每一个仿真点代表 1000 次仿真的平均值。为评估效率，标签数量在 100~5000 范围内变化。我们比较了不同迁移率 η_{mob} 下的系统效率，如图 2 所示。对于结合 CT 的 GSRA 算法，系统效率近似为常数。当 $\eta_{\text{mob}} = 0.2$ ，系统效率可以达到 240%。也应注意到，随着移动率上升，组策略算法的系统效率也随之下降。在图 3 中，我们比较了独立的 AQS 算法和 GSRA 结合 AQS 的组策略算法。可以看到，AQS 的系统效率低于 50%，而 GSRA 结合 AQS 算法在大多数情况下系统效率超过 50%，特别当 $\eta_{\text{mob}} = 0.2$ 时，系统效率可达 170%。

另外，本文还仿真测试了识别所有标签需要的冲突时隙数，如图 4 所示。可知，冲突时隙数量与

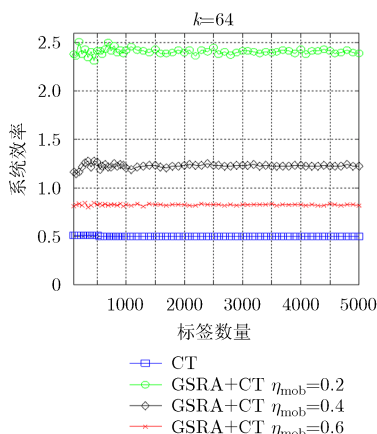


图 2 不同 η_{mob} 下单独 CT 算法和 GSRA 结合 CT 算法的系统效率

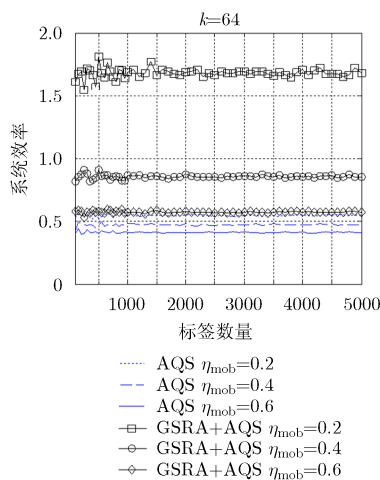


图 3 独立 AQS 算法与 GSRA 结合 AQS 算法的系统效率

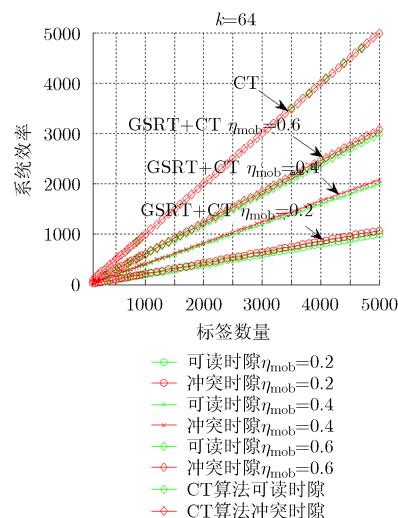


图 4 CT 算法与不同 η_{mob} 下 GSRA 结合 CT 算法所用冲突时隙数(红色)与可读时隙数(绿色)

可读时隙数相近, 冲突时隙数远低于标签数, 而传统 CT 算法冲突时隙数与标签数目相近, 所以本文的识别模型可以有效地降低冲突时隙。

5 结论

本文提出了一个射频识别 RFID 系统中解决冲突问题的 GSRA 协议。针对移动场景我们研究了该算法识别停留标签的系统效率。通过理论分析和仿真, 结果表明, 在标签数量从稀疏到密集的情景下, GSRA 协议处理停留标签在冲突时隙数量和系统识别效率两方面均优于当前主流防碰撞算法。

参 考 文 献

- [1] WONG C P and FENG Quanyuan. Grouping based bit-slot aloha protocol for tag anti-collision in RFID systems[J]. *IEEE Communications Letters*, 2007, 11(12): 946-948. doi: 10.1109/LCOMM.2007.071252.
- [2] SHIN Jongmin, JEON Byeongchan, and YANG Dongmin. Multiple RFID tags Identification with M -ary query tree scheme[J]. *IEEE Communications Letters*, 2013, 17(3): 604-607. doi: 10.1109/LCOMM.2013.012313.122094.
- [3] 李青青, 刘洪武, 张小林. 一种基于不等长时隙的射频识别防碰撞算法[J]. *电子与信息学报*, 2011, 33(11): 2628-2633. doi: 10.3724/SP.J.1146.2011.00303.
LI Qingqing, LIU Hongwu, and ZHANG Xiaolin. An anti-collision algorithm based on unequal timeslots in radio frequency identification system[J]. *Journal of Electronics & Information Technology*, 2011, 33(11): 2628-2633, doi: 10.3724/SP.J.1146.2011.00303.
- [4] 李志坚, 赖顺桥. 一种基于碰撞位指示的射频识别标签防碰撞算法[J]. *电子与信息学报*, 2014, 36(12): 2842-2847. doi: 10.3724/SP.J.1146.2013.01759.
LI Zhijian and LAI Shunqiao. An anti-collision algorithm based on collided bits indicator in radio frequency identification systems[J]. *Journal of Electronics & Information Technology*, 2014, 36(12): 2842-2847. doi: 10.3724/SP.J.1146.2013.01759.
- [5] LAI Yuancheng, HSIAO Lingyen, and LIN Borshen. Optimal slot assignment for binary tracking tree protocol in RFID tag identification[J]. *IEEE/ACM Transactions on Networking*, 2015, 23(1): 255-268. doi: 10.1109/TNET.2013.2295839.
- [6] WU Haifeng and YU Zeng. Passive RFID tag anticollision algorithm for capture effect[J]. *IEEE Sensors Journal*, 2015, 15(1): 218-226. doi: 10.1109/JSEN.2014.2339653.
- [7] LANDALUCE H, PERALLOS A, BENGTTSSON L, et al. Simplified computation in memoryless anticollision RFID identification protocols[J]. *Electronics Letters*, 2014, 50(17): 1250-1252. doi: 10.1049/el.2014.1552.
- [8] LAW Ching, LEE Kayi, and SIU Kaiyeung. Efficient memoryless protocol for tag identification[C]. Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Boston, Massachusetts, USA, 2000: 75-84.
- [9] LI Bo and WANG Junyu. Efficient anti-collision algorithm utilizing the capture effect for ISO 18000-6C RFID protocol[J]. *IEEE Communications Letters*, 2011, 15(3): 352-354. doi: 10.1109/LCOMM.2011.011311.101332.
- [10] SHIN Woojin and KIM Jeonggeun. A capture-aware access control method for enhanced RFID anti-collision performance[J]. *IEEE Communications Letters*, 2009, 13(5): 354-356. doi: 10.1109/LCOMM.2009.081970.
- [11] MYUNG Jihoon and LEE Wonjun. Adaptive binary splitting: a RFID tag collision arbitration protocol for tag identification[C]. IEEE 2nd International Conference on Broadband Networks, Boston, Massachusetts, USA, 2005, 1: 375-383. doi: 10.1109/ICBN.2005.1589637.
- [12] LAI Yuancheng and LIN Chihchung. Two couple-resolution blocking protocols on adaptive query splitting for RFID tag identification[J]. *IEEE Transactions on Mobile Computing*, 2012, 11(10): 1450-1463. doi: 10.1109/TMC.2011.171.
- [13] ZHU Weiping, CAO Jiamong, CHAN Henry C B, et al. Mobile RFID with a high identification rate[J]. *IEEE Transactions on Computers*, 2014, 63(7): 1778-1792. doi: 10.1109/TC.2013.41.
- [14] KIM Taekyung, SHAO Chenglong, and LEE Wonjun. Promptly pinpointing mobile RFID tags for large-scale internet-of-things[C]. 2015 International Conference, IEEE Big Data and Smart Computing (BigComp), Jeju Island, Korea, 2015: 118-123. doi: 10.1109/35021BIGCOMP.2015.7072820.
- [15] JIA Xiaolin, FENG Quanyuan, and YU Lishan. Stability analysis of an efficient anti-collision protocol for RFID tag identification[J]. *IEEE Transactions on Communications*, 2012, 60(8): 2285-2294. doi:10.1109/TCOMM.2012.051512.110448.
- [16] JIA Xiaolin, FENG Quanyuan, and LEI Quanshui. Multi-cycle collision tree algorithm for RFID tag identification[J]. *Journal of Southwest University of Science and Technology*, 2014, 29(1): 39-44.

王 勇: 男, 1974 年生, 讲师, 研究方向为射频识别 RFID 防碰撞算法及嵌入式系统。

唐小虎: 男, 1972 年生, 教授, 博士生导师, 研究方向为编码理论、网络信息安全和射频识别 RFID 系统。

张莉涓: 女, 1987 年生, 博士生, 研究方向为射频识别 RFID 标签防碰撞算法。