

分组自适应分配时隙的 RFID 防碰撞算法研究

张小红, 胡应梦

(江西理工大学信息工程学院, 江西赣州 341000)

摘要: 为了解决射频识别(Radio Frequency Identification, RFID)系统中的多标签防碰撞问题,在分析帧时隙ALOHA算法的基础上,提出一种基于分组自适应分配时隙的RFID防碰撞算法(GAAS).首先让阅读器对标签随机所选的时隙进行扫描统计,并将其发送给每一个标签,标签再进行相应地时隙调整,使阅读器跳过空闲时隙和碰撞时隙,自适应地分配有效时隙,进而对标签进行快速识别.当未识别标签数比较大时,算法采用分组以及动态调整帧长等策略,以减少时隙处理的时间.仿真结果表明:GAAS算法提高了系统的识别效率和稳定性,降低了传输开销.特别是当标签数超过1000时,该算法的吞吐率仍保持在71%以上,比传统的帧时隙ALOHA-256算法和分组动态帧时隙ALOHA算法的系统效率分别提高了300%和97.2%.

关键词: 射频识别; ALOHA算法; 标签分组; 吞吐率; 自适应分配时隙

中图分类号: TN911.23

文献标识码: A

文章编号: 0372-2112(2016)06-1328-08

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

DOI: 10.3969/j.issn.0372-2112.2016.06.010

Research on a Grouped Adaptive Allocating Slot Anti-collision Algorithm in RFID System

ZHANG Xiao-hong, HU Ying-meng

(School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China)

Abstract: Based on frame slotted ALOHA algorithm, a grouped adaptive allocating slots (GAAS) anti-collision algorithm is presented to solve the problem of collision between the reader and multi-tag in radio frequency identification (RFID) system. First, the reader needs to obtain the time slots chosen randomly by tags and send the results to each tag; then the tags rectify the time according to the instruction; moreover, the reader skips free and collision time slots, and adaptively distributes valid ones; finally, the tags are quickly recognized in GAAS. When the number of unidentified tags is very large, the tags are grouped and the frame sizes are adjusted dynamically to reduce the processing time. The simulation results show that GAAS has higher identification efficiency and stability, and lower cost of communication. Particularly, when the number of tags is over 1000, the throughput rate still maintains above 71%. Compared with the framed slotted ALOHA-256 algorithm and the grouped dynamic framed slotted ALOHA algorithm, the proposed algorithm enhances the system efficiency by 300% and 97.2% respectively.

Key words: RFID; ALOHA algorithm; tags grouping; throughput rate; adaptively allocating slots

1 引言

射频识别(Radio Frequency Identification, RFID)是一种利用电磁波传播方式在阅读器和标签之间进行非接触双向数据传输,进而获取被标识物体信息的识别技术^[1],被公认为21世纪的最具发展前景和变革力的高新技术.该技术具有数据交换快、追踪物体及时、无空间限制、穿透能力强、多目标识别和抗污染等优点,在物流管理、交通运输、自动化生产、公共信息服务等行业有着广泛的应

用,可大幅提高管理与运作效率,降低成本^[2,3].

RFID系统由电子标签(tag)、读写器(reader)和后端数据库(database)三大部分组成.在多个阅读器与多个标签的射频识别系统中,存在着两种形式的碰撞,即阅读器碰撞和标签碰撞^[4].由于阅读器碰撞发生的概率较小且阅读器本身的处理能力较强,因此阅读器碰撞问题较容易解决.国内外的学者在多标签同时与读写器进行通信的碰撞问题方面已经做了大量的研究^[5-7],这些方法一般被分为四类:空分多路法、频分多

收稿日期:2014-09-08;修回日期:2014-12-01;责任编辑:梅志强

基金项目:国家自然科学基金(No. 61363076, 11062002);江西省自然科学基金(No. 20142BAB207020);江西省教育厅科技项目(No. GJJ14465);江西省研究生创新专项资金(No. YC2014-S370)

路法、码分多路法和时分多路法^[8]。由于标签的低功耗、低存储能力和有限的计算能力等特点,标签防碰撞方法主要采用时分多路法。

在时分多路法中,目前最常用的多标签防碰撞算法分为两种,一种是基于二进制树的确定型算法,其主要代表算法有二进制搜索算法^[9]、动态二进制搜索算法^[10]、跳跃式动态树算法^[11]、查询树算法^[12]等。该法是由阅读器根据标签 ID 的唯一性来选择标签进行通信,所以搜索算法的性能会随 ID 位数的增加而急剧恶化。另一种是基于 ALOHA 的统计型算法。其主要算法包括时隙 ALOHA (Slotted ALOHA, SA) 算法^[13]、帧时隙 ALOHA (Frame-slotted ALOHA, FSA) 算法^[14]、动态帧时隙 ALOHA (Dynamic Frame-slotted ALOHA, DFSA) 算法^[15], 另外还有 EPCglobal 提出的 EPC Class-1 Generation-2 (EPC C1G2) 标准^[16]采用一种基于 Q 值的随机帧时隙算法^[17,18]以及近期有文献提出基于碰撞预测的 ALOHA 防碰撞 (Collision Prediction-ALOHA, CP-ALOHA) 算法^[19]等。这些防碰撞算法实现过程相对比较简单,标签内部也不需要设计复杂的电路,因此标签的成本也较低。然而,随着标签的数目增大,甚至上千时,发生碰撞的概率也随之增大,系统识别的性能急剧下降。针对于大量标签的识别问题,研究者们又提出分组的概念,有增强的动态帧时隙算法^[20]、分组动态帧时隙 ALOHA (Grouped Dynamic Frame-slotted ALOHA, GDFSA) 算法^[21]等。与之前的算法相比,分组类的防碰撞法其吞性能比较稳定,但是与其他算法一样,其吞吐率均比较低,只有 40% ~ 50% 左右。

本文提出了自适应分配时隙 (Adaptive Allocating Slots, AAS) 算法,随着待识别标签数目的增加,又加入分组的概念,即分组自适应分配时隙 (Grouped Adaptive Allocating Slots, GAAS) 算法。该算法结合了传统 GDFSA 和 CP-ALOHA 算法的特点,在它们的基础上进行了优化和改进。首先估计标签数量,然后采用分组、动态调整帧长、时隙预约以及自适应分配时隙等策略对标签进行识别。仿真结果表明,新的算法降低了传输开销,简化了标签电路的设计,提高了系统的识别效率和稳定性。当标签数目大于 1000 时,其系统识别效率仍可达 71% 以上,与 EPC 标准协议相比,传输开销大约下降了 48.1%,且随着标签数目的增加,新算法的优越性更加明显。

2 GAAS 算法描述

2.1 基本定义

定义 1^[22] 设定一帧内的时隙数,即帧长为 L ,在对 n 个标签进行识别时,每一个标签都会随机选择一个时隙来发送自己的识别码信息,根据二项分布定理,某个标签占用帧内任意时隙的概率为 $p = 1/L$,则同一个

时隙里有 r 个标签的概率可以表示为:

$$P(L, n, r) = C_n^r \times \left(\frac{1}{L}\right)^r \times \left(1 - \frac{1}{L}\right)^{n-r} \quad (1)$$

当 $r=0$,即一个时隙里没有请求识别标签,该时隙称为空闲 (idle) 时隙,其概率为:

$$P_i = P(L, n, 0) = \left(1 - \frac{1}{L}\right)^n \quad (2)$$

当 $r=1$,即一个时隙里只有一个标签请求识别标签,该时隙称为成功 (success) 时隙,其概率为:

$$P_s = P(L, n, 1) = \frac{n}{L} \times \left(1 - \frac{1}{L}\right)^{n-1} \quad (3)$$

当 $r \geq 2$,即一个时隙里有两个及以上的标签请求识别标签,该时隙称为碰撞 (collision) 时隙,其概率为:

$$P_c = 1 - P_s - P_i \quad (4)$$

则在一帧的识别周期后,没有标签的时隙数的期望值 $a_i^{L,n}$,仅有一个标签的时隙数的期望值 $a_s^{L,n}$,产生碰撞的时隙数的期望值 $a_c^{L,n}$ 分别为:

$$a_i^{L,n} = L \times P_i = L \left(1 - \frac{1}{L}\right)^n \quad (5)$$

$$a_s^{L,n} = L \times P_s = n \left(1 - \frac{1}{L}\right)^{n-1} \quad (6)$$

$$a_c^{L,n} = L \times P_c = L - a_i^{L,n} - a_s^{L,n} \quad (7)$$

定义 2^[23] RFID 系统的吞吐率 S 是指阅读器在一个识别帧长的时间内成功传输信息的标签数目所占的比例,即:

$$S = \frac{\text{成功识别标签的数目}}{\text{帧长度}} = \frac{a_s^{L,n}}{L} = P_s \quad (8)$$

2.2 标签数目估计

GAAS 算法要求阅读器在开始识别前,需估计剩下标签的数量。帧长的大小和未识别标签的数目越接近,系统效率越高,因此该过程对于整个识别过程起着至关重要的作用。目前已经提出了多种标签数目估计方法,有 Vogt 算法^[24]、 C_{ratio} 算法^[25]、Schoute 算法^[26]和 Low Bound 算法^[27]。Vogt 算法可以做出误差相对较小且稳定的估计,因此本文采用 Vogt 算法来估计标签的数目。

$$d(L, c_i, c_s, c_c) = \min \left\| \begin{pmatrix} a_i^{L,n} \\ a_s^{L,n} \\ a_c^{L,n} \end{pmatrix} - \begin{pmatrix} c_i \\ c_s \\ c_c \end{pmatrix} \right\| \quad (9)$$

其中, L 表示当前帧长, n 为估算的标签数量。 c_s 、 c_i 、 c_c 分别表示上一识别周期内成功时隙数、空闲时隙数、碰撞时隙数。而成功时隙数 $a_s^{L,n}$ 、空闲时隙数 $a_i^{L,n}$ 、碰撞时隙数 $a_c^{L,n}$ 的期望值可分别由式 (5) ~ (7) 求得。

2.3 RFID 系统最优帧长分析

阅读器在其可读范围内的标签数目进行估计之后,需要根据标签数目动态的调整帧长,如果帧长取值

太大就会产生大量的空闲时隙,反之会导致碰撞时隙急剧上升,最终都将影响系统识别效率.因此要想取得较高的吞吐率效率,必须找出帧长与标签数目之间对应关系,即确定最优帧长^[28].对式(8)求导:

$$\frac{dS}{dn} = \frac{1}{L} \left(1 - \frac{1}{L}\right)^{n-1} + \frac{n}{L} \left(1 - \frac{1}{L}\right)^{n-1} \ln\left(1 - \frac{1}{L}\right) \quad (10)$$

令式(10)等于0,达到帧长 L 与标签的个数 n 应满足:

$$n = -\frac{1}{\ln\left(1 - \frac{1}{L}\right)} \quad (11)$$

再由泰勒级数展开得到

$$L \approx \frac{1 + \frac{1}{n}}{1 + \frac{1}{n} - 1} = n + 1 \quad (12)$$

由式(12)可知,当标签数 n 远大于1,并且 n 接近于帧长 L 时,系统的吞吐率达到最大.图1给出了 L 为不同固定值时系统吞吐率的仿真结果:

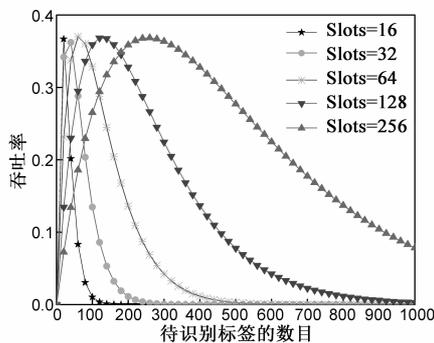


图1 不同固定帧长的吞吐率

根据式(8),相邻固定帧长 L_1 和 L_2 的吞吐率曲线交点处的标签的数目,即为调整帧长的临界点.

$$\frac{n}{L_1} \times \left(1 - \frac{1}{L_1}\right)^{n-1} = \frac{n}{L_2} \times \left(1 - \frac{1}{L_2}\right)^{n-1} \quad (13)$$

$$n = \left\lfloor 1 + \ln\left(\frac{L_2}{L_1}\right) / \ln\left(\frac{L_1}{L_2} \times \frac{L_2 - 1}{L_1 - 1}\right) \right\rfloor \quad (14)$$

其中 $\lfloor * \rfloor$ 表示向下取整运算,从而得到了标签数目 n 和帧长 L 的对应关系,如表1所示.由标签数目的取值范围决定帧长的大小,当标签数目大于354时,仍以最大帧长256分别进行识别.

表1 帧长与标签数目对应关系

帧长	8	16	32	64	128	256	256
最小标签的数目	1	12	23	45	89	178	355
最大标签的数目	11	22	44	88	177	354	≥ 355

2.4 标签分组原理

由于受到标签成本的限制,使得帧时隙数不能够随着标签数目的增加而无限地增加,所以针对于大量

的标签的情况,只有通过限制每次响应的标签的数目,才能使系统保持相对较高的吞吐率.根据式(8),选取对标签进行分组的临界值^[29],即两相邻帧的性能曲线交点 $P_a = P_b$ 处的标签数值.

$$\frac{n}{256 \times a} \times \left(1 - \frac{1}{256}\right)^{n/a-1} = \frac{n}{256 \times b} \times \left(1 - \frac{1}{256}\right)^{n/b-1} \quad (15)$$

其中, a, b 为标签的相邻分组数,例如当取 $a = 1, b = 2$ 代入式(15)可得:

$$\frac{n}{256 \times 1} \times \left(1 - \frac{1}{256}\right)^{n/1-1} = \frac{n}{256 \times 2} \times \left(1 - \frac{1}{256}\right)^{n/2-1} \quad (16)$$

$$n = 2 \frac{\ln \frac{1}{2}}{\ln \frac{255}{256}} \approx 354 \quad (17)$$

即 $n = 354$ 是将标签分为一组或两组的临界值,为了使系统保持较高的吞吐率效率,未识别标签数 n 不能大于354,当 n 大于354时,需要对未识别的标签进行分组.由式(15)可计算得到标签数量在2283以内的分组数,如表2所示.

表2 标签总数与分组数的对应关系

分组数	1	2	4	6	8
最小标签数	1	355	710	1246	1767
最大标签数	354	709	1245	1766	2283

表2中的分组数1,2,4,6,8是自主设计的,各组中的最小标签数与最大标签数是式(15)中通过 a, b 数值计算出来的门限值.随着标签数目的增加,分组数也不断增大.

2.5 分组自适应时隙分配算法

2.5.1 GAAS 算法约定

(1) Query(M)命令是碰撞时隙扫描的命令,其中的参数 M 是帧的时隙数.标签在这 M 个时隙中随机选取一个时隙,并加载到计数器中,同时向阅读器返回预约时隙数 k ,然后根据所选择的时隙进行相应的延时.

(2) Refresh_slot(Slots)命令,该命令是阅读器根据在碰撞时隙扫描的过程中,将时隙的选择情况发送给每一个标签,其中,Slots是一个数组.如果时隙能够完整读取就标记为0,反之则记为-1.标签接收到该命令之后,再进行时隙调整.

(3) Collection()命令,该命令是读取标签信息的命令.标签接收到这个命令之后,会根据上一个命令中更新的时隙,延迟响应时间,最后再将自身的数据发送至阅读器.

(4) Count()命令,该命令的作用是让标签时隙计数器减1,计数器为0的标签响应阅读器.

(5) Sleep()命令,该命令的作用是让标签暂时处

于静默状态,使其不参与后面的查询。

2.5.2 GAAS 防碰撞协议

GAAS 算法的核心就是在进行标签识别之前,首先进行时隙扫描操作,记录下标签所预约时隙的情况,然后在标签识别阶段,让阅读器跳过碰撞时隙和空闲时隙,而直接分配有效时隙,对成功预约的标签直接进行识别,从而提高了时隙的利用率。其协议流程如图 2 所示,协议过程可分为三个阶段:

(1) 标签数目估计及分组阶段

①在识别开始阶段,用 Vogt 算法估计待识别标签的数目 n 。

②当标签数目小于 354 个时,采用动态帧时隙策略,动态调整识别帧的长度 M ,直接进入时隙处理阶段。当 n 大于 354 时,则需要对标签进行分组,由表 2 求得分组数 g 。

③标签在 1 到 g 之间随机选择一个数 i ,作为自己的组编号,同时把 $s[i]$ 的值增加 1,记录该组的标签数。

④初始化当前识别的组编号 $t = 1$,开始对第 t 组进

行识别。

(2) 时隙处理阶段

①在进行数据读取之前,首先进行时隙扫描,阅读器以广播的形式发送 Query(M) 命令给每个标签。标签收到该命令之后,再向阅读器返回各自所预约的时隙数。

②阅读器再根据所接收到的数据信息,来判断哪些时隙可以成功识别,哪些时隙将会产生碰撞或空闲时隙。如果能成功识别标签,就将相应的时隙标志位 Flag 设为 0,其他情况就将 Flag 设为 -1。

③阅读器根据标签所预约时隙的情况,将其通过命令 Refresh_slot(Slots) 发送给每个标签,让标签也能够知道其他标签选择时隙的情况,然后调整相应的时隙。Slots 里面的元素就是在前面的时隙扫描过程中记录下的标记值,标签根据这个数组来调整相应所选择的时隙数。

(3) 标签识别阶段

①由于阅读器在上一个阶段已经知道了各时隙的选择情况,在识别阶段直接分配有效时隙。接下来阅读器发送 Collection() 命令,标签接收到这个命令之后,就按照调整后的时隙依次向阅读器发送数据。

②阅读器发送 Count() 命令,让该组中各个标签的时隙计数器减 1。

③最后阅读器发送 Sleep() 命令,在这一轮查询过程中被正确读取到的标签就会进入静默状态,即不参与接下来的查询。

④在每一轮识别结束以后,再估算剩余未识别标签的数目,若标签数目不为 0,则重复上面②,③两个阶段,直至将该组剩下的标签全部识别完。

⑤组编号加 1,即 $t = t + 1$,然后执行下述两种情况之一。

(a) 如果 $t \leq g$,则表示存在没有识别的组,继续进行下一组的识别。

(b) 如果 $t > g$,即所有的组都识别完成,识别结束。

3 算法对比分析

3.1 DFSA 算法分析

动态帧时隙 DFSA 算法会根据前一帧的情况,估算出标签的数量,动态的调整帧的长度,这相对于 FSA 算法提高了效率。图 3 为 DFSA 算法的示意图,假设有 8 个标签,帧长为 8,即每帧中含有 8 个时隙,各标签在每帧中随机选取一个时隙发送信息。DFSA 算法的识别过程是:首先由阅读器向周围标签广播发送一个命令,通知标签该帧包含有 8 个时隙,然后标签从 1~8 时隙中随机选择一个作为自己的发送时隙,如果所选时隙上没有产生碰撞,则标签就会成功识别,接着退出系统。否

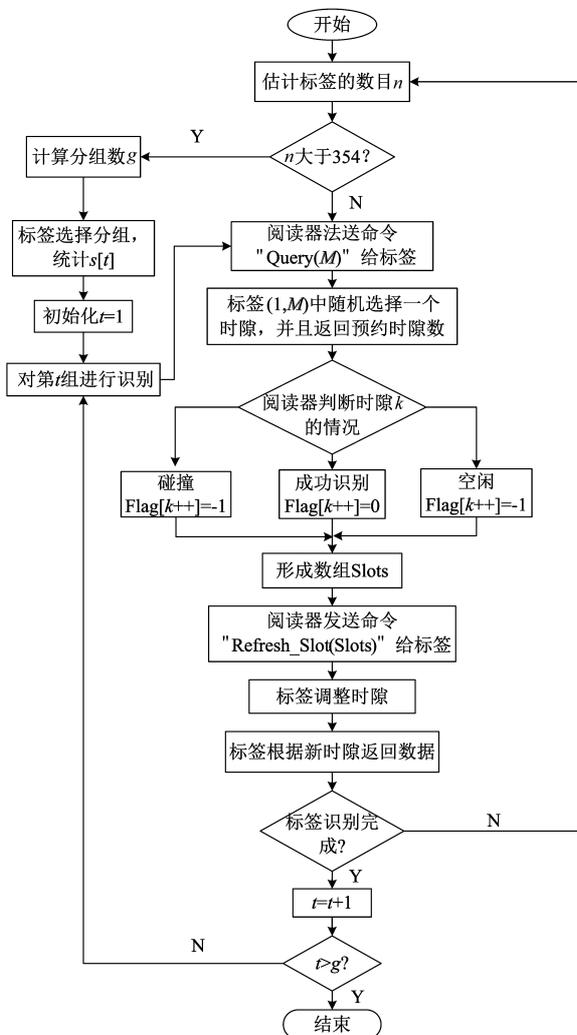


图2 GAAS算法的流程图

则,阅读器将通知剩下未识别标签,在下一轮识别中重新选择发送时隙,直到所有的标签均被正确识别.

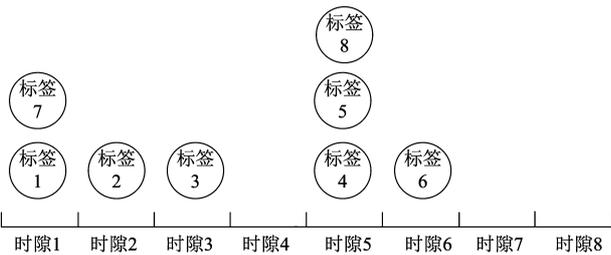


图3 DFSA算法的示意图

这种算法虽然简单,但是在其识别过程中会产生过多的碰撞时隙或空闲时隙,进而导致系统吞吐率不高.

3.2 GAAS 算法分析

仍以上面的例子,我们以 GAAS 算法进行分析. 首先进行时隙扫描,阅读器以广播的形式发送 Query (8) 命令给每个标签,每个标签就从 8 个时隙中随机选择一个作为自己的时隙,并按照所选时隙进行相应的延时,然后向阅读器返回一个时长非常短的预约帧. 由于受到标签成本的限制,所预约的时隙数不会超过 256 个,即 8bit,为了尽量避免预约帧产生碰撞,后面的协议都将预约帧规定为 16bit,即 2 个字节,按照 ISO18000 - 6 协议规定,数据速率为 40Kbps,每个普通数据帧为 16 个字节,可知发送预约帧只需 1/8 个时隙,所以发送预约帧非常快. 假设有 8 个标签,帧长为 8,即 $M = 8$. 如表 3 所示,其中 φ 表示没有标签.

表 3 时隙扫描过程中时隙的选择情况

时隙号	1	2	3	4	5	6	7	8
标签号	1,7	2	3	φ	4,5,8	6	φ	φ
状态	碰撞	成功	成功	空闲	碰撞	成功	空闲	空闲
标记值	-1	0	0	-1	-1	0	-1	-1

然后阅读器发送 Refresh_slot (Slots) 命令,其中数组 Slots 为 $[-1,0,0,-1,-1,0,-1,-1]$,标签就会根据这个数组来调整将要发送数据的时隙. 例如标签 1,7 开始所预约的时隙为 1,但是该时隙上产生了碰撞,故将时隙数调整为 0,所以标签 1,7 就不再参加后面的查询过程. 标签 2 开始所选时隙为 2,则根据数组的前 2 个数值来调整,即 $2 - 1 = 1$,所以标签 2 将在时隙 1 中上传数据. 同理,可以求得其他标签最终所选择的时隙,并且将进行相应的延时. 标签调整之后的时隙数如表 4 所示.

接下来阅读器发送 Collection() 命令,标签接收到这个命令之后,就会按照调整后的时隙依次发送数据. 只有标签 2,3 和 6 号标签选择了有效时隙,分别为新时

隙 1,2,3. 如图 4 所示. 阅读器跳过了碰撞时隙以及空闲时隙,自适应地分配有效时隙,因此就提高了信道的利用率. 假设每个时隙为 5ms,可以看到在这一轮查询过程中总的时间消耗为 8 个预约帧为 5ms,然后是 15ms 的读取时隙,所以一共消耗了 20ms. 相对于前面 DFSA 算法而言,减少了 20ms 的时间,当标签的数目增加,帧长值更大时,该算法将会节省更多的时隙.

表 4 调整后的时隙选择情况

标签号	调整的方法	调整后的时隙
1	设为 0	0
2	$2 - 1 = 1$	1
3	$3 - 1 = 2$	2
4	设为 0	0
5	设为 0	0
6	$6 - 1 - 1 - 1 = 3$	3
7	设为 0	0
8	设为 0	0

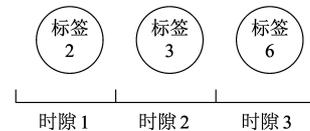


图4 GAAS算法的示意图

以上仅是一个简单的范例. 由 2.5.2 节可知整个算法可分为三个阶段:标签数目估计及分组阶段,时隙处理阶段和标签识别阶段. 由于第一阶段所消耗的时隙很少,所以本文主要讨论后面两个阶段.

阅读器首先根据在其作用范围内标签的数目分配一定数量的时隙(如表 5 所示,与识别帧的长度有关),以便获得标签预约时隙的情况,然后再根据协议,让标签重新调整时隙,进入标签识别阶段. 阅读器再根据标签成功预约的数目,自适应地分配相应数量的时隙,最后将此次成功预约的标签全部识别出来.

表 5 标签数与时隙预测阶段消耗的时隙数对应关系

识别帧长	8	16	32	64	128	256	256
最小标签的数目	1	2	23	45	89	178	355
最大标签的数目	11	22	44	88	177	354	≥ 355
预约阶段消耗的时隙数	1	2	4	8	16	32	32

从图 5 可知,当标签数目不断增大时,系统的性能急剧下降,因此需要对其进行分组. 但是,分组越多,其消耗的时隙会相应的增加,调整时隙的时间也会越长. 当标签的数目在 0 ~ 600 之间时,分四组反而会降低系统的性能. 所以 GAAS 算法采取自适应分组的方法,根据标签的数目自主进行分组. 该策略能够解决标签数目较大而致识别效率过低的问题,同时也能使系统性能保持相对稳定,减少了标签的识别时间.

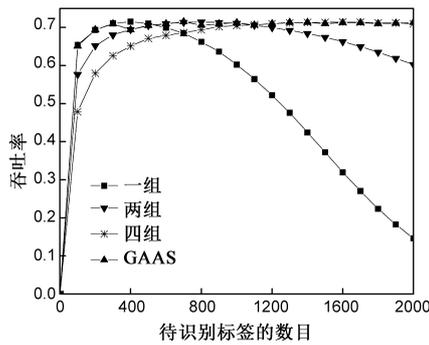


图5 不同分组对应的吞吐率

4 算法性能分析

为了从实验角度证实 GAAS 算法的有效性,在 Windows7 操作系统,2G 内存的环境下,利用仿真软件 MATLAB 分别对 FSA-256 算法、DFSA 算法、EPC 标准算法、GDFSA 算法以及 GAAS 算法进行仿真.假设系统的标签分布是均匀的,标签个数取 1500 以上,为了与分析相对应,仿真的结果是取相同条件下 100 次实验的平均值.

4.1 标签电路复杂度分析

假设标签随机选取时隙的随机数为 R ,则需 $\lceil \log_2 R \rceil$ 位随机数,由于本算法采取分组的方法, R 最大值设为 256,仅需 8 位即可,所以标签只需 8 位随机数产生电路.而 EPC C1G2 防碰撞协议除了产生选取时隙的随机数需 4 位外,还要求生成 16 位 RN16 随机数,一共需要 $16 + 4 = 20$ 位随机数产生电路.因此,本算法对标签设计随机数电路复杂性明显低于 EPC C1G2 标签.此外,标签中还有基于状态机的控制器来执行命令,EPC C1G2 标签需执行主要命令有 5 个(Query, QueryAdjust, QueryRep, Ack, RN16),本文协议中标签也需执行 5 个(Query, Refresh-slot, Count, Collection, Sleep),故 GAAS 协议标签的控制器电路复杂性也与 EPC C1G2 协议相当.综上所述,GAAS 协议标签电路设计比 EPC C1G2 协议简单,从而降低了标签的成本.

4.2 传输开销分析

传输开销是评估一个算法的重要指标,其包括阅读器开销和标签开销两部分.分别对 AAS 算法、GAAS 算法以及 EPC 标准中的算法在标签识别过程中的传输开销进行仿真,仿真中用到的 GAAS 命令和参数长度如表 6 所示.

4.2.1 阅读器的开销

设标签数量在区间 $[0, 2000]$ 内变化,读写器传输的比特数如图 6 所示,随着标签数目的增加,阅读器的开销也不断增大.当标签的数目少于 1300 时,AAS 算法与 GAAS 算法阅读器开销相当,而当标签的数目超过

1300 时,GAAS 算法的优势开始凸显出来.这是因为 AAS 算法没有将标签进行分组处理,随着标签数目的增加,标签碰撞的概率急剧上升,阅读器发送的指令也会随之增加.当标签数目为 2000 时,AAS 算法中读写器比特数为 66790,EPC 标准算法分别为 60527,而 GAAS 算法为 54245,比 AAS 算法的开销下降了 18.8%,比 EPC 标准算法下降了 10.4%.

表 6 GAAS 算法中涉及的命令和参数长度

命令/参数	功能	长度/bit
Query	调整帧长	8
Refresh-slot	调整时隙数	32
Count	时隙计数	8
Collection	确认通信	18
Sleep	进入静默状态	2

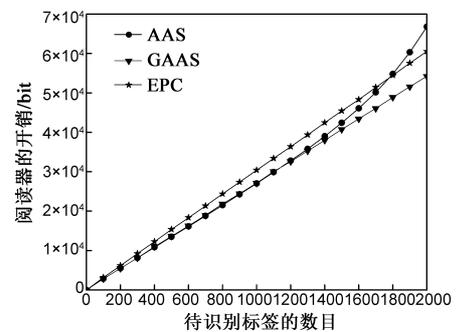


图6 阅读器的开销

4.2.2 标签的开销

图 7 显示了三种算法标签的开销,随着标签数目的增加,AAS 算法中标签传送比特数接近指数增长,而 GAAS 算法和 EPC 标准算法近似线性增长,其中 GAAS 增长最为缓慢.当标签的数目为 2000 时,AAS 算法中读写器比特数为 422135,EPC 标准算法分别为 121545,而 GAAS 算法仅为 39601,比 AAS 算法标签的开销下降了 96.1%,比 EPC 标准算法下降了 67.4%.

4.3 总时隙数分析

总时隙是决定系统效率的一个关键因素,总时隙数越少,系统的性能越好.从上面分析可知,整个算法分

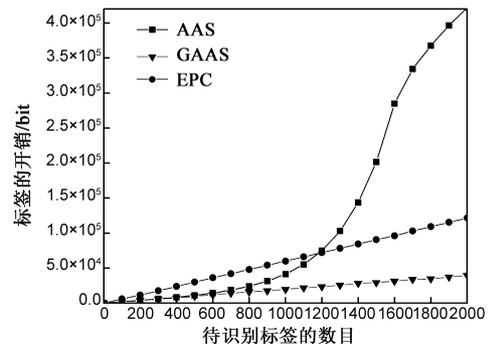


图7 标签的开销

为两个阶段:时隙扫描阶段和标签识别阶段,所以查询总时隙也是这两个阶段消耗时隙之和。

对 FSA-256 算法、DFSA 算法、GDFSA 算法以及 GAAS 算法的总时隙数进行仿真,仿真结果如图 8 所示。标签数量从 0 变化到 1500 的过程中,FSA-256 算法和 DFSA 算法所需要的总时隙数随标签数量的增加呈指数增长,GDFSA 算法和 GAAS 算法呈线性增长,其中 FSA-256 算法增长速度最快,而 GAAS 算法最慢,其次是 GDFSA 算法。特别是当标签数目较大时,GAAS 算法的优势更加明显。当标签数目为 1000 时,GAAS 算法只需要约 1400 个时隙,比 FSA-256 减少约 4165,比 DFSA 减少约 3727,比 GDFSA 减少约 1366。

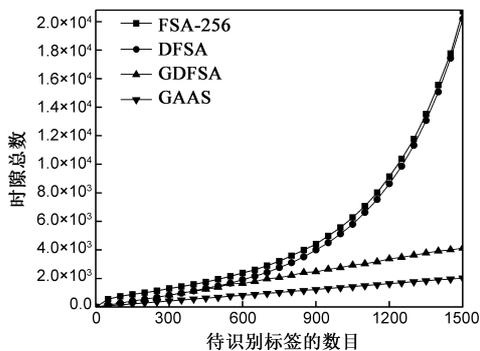


图8 四种算法所需的总时隙数

4.4 吞吐率分析

系统吞吐率也是衡量系统性能的一个重要指标。从图 9 可以看出,当标签少于 354 时,GDFSA 与 DFSA 吞吐率相同,FSA-256 算法最低,仅有 0.2 左右,而 GAAS 算法最高,可达 0.7 以上。GDFSA、DFSA、GAAS 算法都能根据实际标签数目,自适应地分配有效时隙进行识别,而 FSA-256 算法采用固定帧长 256。当标签数大于 354 时,FSA-256、DFSA 算法的吞吐率均急剧下降,而 GDFSA、GAAS 算法则将标签分为多组,通过动态调整帧长对各组标签进行识别,使得吞吐率稳定在一定范围内。本文算法的系统吞吐率比其他三种算法显然要大得多,FSA-256 算法的吞吐率在 0.1 ~ 0.25 之

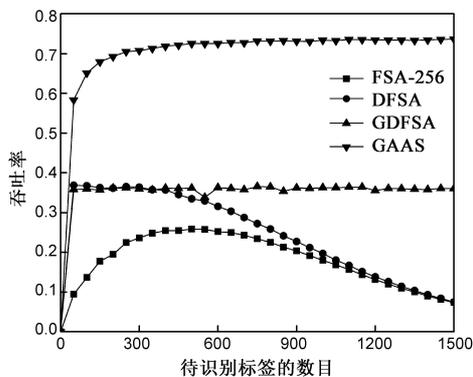


图9 四种算法的吞吐率比较

间,DFSA 算法则在 0.2 ~ 0.36 之间,GDFSA 算法仅维持在 0.36 左右,而 GAAS 算法比它们都要高。当标签的数目达到 1000 时,与 FSA-256 和 GDFSA 相比,GAAS 算法的系统效率分别提高了 300% 和 97.2%。

5 结论

本文提出了一种分组自适应分配时隙 RFID 防碰撞(GAAS)算法,通过对标签数量的估计和分组、时隙预约以及自适应分配时隙等策略对标签进行快速识别。仿真结果表明,随着标签数量的不断增加,特别是当标签的数目超过 1000 时,GAAS 算法的吞吐率维持在 0.71 以上,比基于 ALOHA 的传统算法的吞吐率都较大幅度的提高,整个识别过程所需要的时隙总数和传输开销几乎保持了线性增加。能有效地提高 RFID 系统的工作效率,增加了系统吞吐率的稳定性,降低了标签的成本。针对于大量的标签的识别,GAAS 算法的优势尤为显著,具有广阔的应用前景。

参考文献

- [1] Want R. An introduction to RFID technology [J]. IEEE Pervasive Computing, 2006, 5(1): 25-33.
- [2] 宁焕生,张瑜,刘芳丽,等. 中国物联网信息服务系统研究 [J]. 电子学报, 2006, 34(12A): 2514-2517.
- [3] NING H S, ZHANG Y, LIU F L, et al. Research on china internet of things' services and management [J]. Acta Electronica Sinica, 2006, 34(12A): 2514-2517. (in Chinese)
- [4] Hunt V D, Puglia A, Puglia M. RFID: A Guide to Radio Frequency Identification [M]. New Jersey: John Wiley & Sons, 2007.
- [5] Yoon W, Vaidya N H. RFID reader collision problem: performance analysis and medium access [J]. Wireless Communications and Mobile Computing, 2012, 12(5): 420-430.
- [6] 李萌,钱志鸿,张旭,等. 基于时隙预测的 RFID 防碰撞 ALOHA 算法 [J]. 通信学报, 2012, 32(12): 43-50.
- [7] Li Meng, Qian Zhi-hong, Zhang Xu, et al. Slot-predicting based ALOHA algorithm for RFID anti-collision [J]. Journal on Communications, 2012, 32(12): 43-50. (in Chinese)
- [8] Wu H, Zeng Y. Bayesian tag estimate and optimal frame length for anti-collision ALOHA RFID system [J]. IEEE Transactions on Automation Science and Engineering, 2010, 7(4): 963-969.
- [9] Deng D J, Tsao H W. Optimal dynamic framed slotted ALOHA based anti-collision algorithm for RFID systems [J]. Wireless Personal Communications, 2011, 59(1): 109-122.
- [10] Küpper A. Front Matter [M]. New Jersey: John Wiley & Sons, Ltd, 2005.
- [11] Finkenzerler K. Example Applications [M]. New Jersey: John Wiley & Sons, Ltd, 2003.

- [10] Crain T, Gramoli V, Raynal M. A speculation-friendly binary search tree[J]. *Acm Sigplan Notices*, 2012, 47(8):161–170.
- [11] 王亚奇, 蒋国平. 基于分组机制的跳跃式动态二进制防碰撞算法[J]. *自动化学报*, 2010, 36(10):1390–1400.
Wang Ya-qi, Jiang Guo-ping. Anti-collision algorithm based on grouping mechanism and jumping dynamic binary[J]. *Acta Automatica Sinica*, 2010, 36(10):1390–1400. (in Chinese)
- [12] Yang C N, He J Y. An effective 16-bit random number aided query tree algorithm for RFID tag anti-collision[J]. *IEEE Communications Letters*, 2011, 15(5):539–541.
- [13] Liva G. Graph-based analysis and optimization of contention resolution diversity slotted ALOHA[J]. *IEEE Transactions on Communications*, 2011, 59(2):477–487.
- [14] Wu H, Zeng Y. Efficient framed slotted ALOHA protocol for RFID tag anti-collision[J]. *IEEE Transactions on Automation Science and Engineering*, 2011, 8(3):581–588.
- [15] Deng D J, Tsao H W. Optimal dynamic framed slotted ALOHA based anti-collision algorithm for RFID systems[J]. *Wireless Personal Communications*, 2011, 59(1):109–122.
- [16] Chien H Y, Chen C H. Mutual authentication protocol for RFID conforming to EPC Class-1 Generation-2 standards[J]. *Computer Standards & Interfaces*, 2007, 29(2):254–259.
- [17] Chen W T. A feasible and easy-to-implement anti-collision algorithm for the EPC global UHF Class-1 Generation-2 RFID protocol[J]. *IEEE Transactions on Automation Science and Engineering*, 2014, 11(2):485–491.
- [18] 张小红, 张留洋. RFID 防碰撞时隙应变协处理算法研究[J]. *电子学报*, 2013, 42(6):1139–1146.
ZHANG Xiao-hong, ZHANG Liu-yang. Research on RFID anti-collision algorithm of slot responding in real-time and co-processing[J]. *Acta Electronica Sinica*, 2013, 42(6):1139–1146. (in Chinese)
- [19] 龚冰青. 一种 433MHz 有源 RFID 标签的设计与实现[D]. 成都:电子科技大学, 2013.
Gong Bing-qing. Design and implementation of active RFID tag with 433MHz [D]. Chengdu: University of Electronic Science and Technology of China, 2013. (in Chinese)
- [20] Wang C Y, Lee C C, Lee M C. An enhanced dynamic framed slotted ALOHA anti-collision method for mobile RFID tag identification[J]. *Journal of Convergence Information Technology*, 2011, 6(4):340–351.
- [21] Lin C F, Lin F Y S. Efficient estimation and collision-group-based anti-collision algorithms for dynamic frame-slotted ALOHA in RFID networks[J]. *IEEE Transactions on Automation Science and Engineering*, 2010, 7(4):840–848.
- [22] Park J, 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.
- [23] 庞宇, 彭琦, 林金朝, 等. 基于分组动态帧时隙的射频识别防碰撞算法[J]. *物理学报*, 2013, 62(14):148401–148401.
Pang Yu, Peng Qi, Lin Jin-zhao, et al. Reducing tag collision in radio frequency identification systems by using a grouped dynamic frame slotted ALOHA algorithm[J]. *Acta Phys Sin*, 2013, 62(14):148401–148401. (in Chinese)
- [24] 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[J]. *自动化学报*, 2010, 36(4):620–624.
Wu Hai-feng, Zeng Yu. Tag estimate and frame length for dynamic frame slotted ALOHA anti-collision RFID system[J]. *Acta Automatica Sinica*, 2010, 36(4):620–624. (in Chinese)
- [25] Eom J B, Lee T J. Accurate tag estimation for dynamic framed-slotted ALOHA in RFID systems[J]. *IEEE Communications Letters*, 2010, 14(1):60–62.
- [26] Schoute F C. Dynamic frame length ALOHA [J]. *IEEE Transactions on Communications*, 1983, 31(4):565–568.
- [27] Chen W T. An accurate tag estimate method for improving the performance of an RFID anti-collision algorithm based on dynamic frame length ALOHA [J]. *IEEE Transactions on Automation Science and Engineering*, 2009, 6(1):9–15.
- [28] Cha J R, Kim J H. Novel anti-collision algorithms for fast object identification in RFID system [A]. *Proceedings of the 2005 Eleventh International Conference on Parallel and Distributed Systems [C]*. Fukuoka: IEEE, 2005. 63–67.
- [29] Lee S R, Joo S D, Lee C W. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification [A]. *Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services [C]*. California: IEEE, 2005. 166–172.

作者简介



张小红 女, 1966 年 8 月出生, 河北昌黎人, 现为江西理工大学信息工程学院教授、博士、硕士生导师, 研究方向: 无线传感器网络、非线性动力学理论、混沌保密通信。
E-mail: xiaohongzh@263.net



胡应梦 男, 1989 年 11 月出生, 湖南娄底人, 现为江西理工大学信息工程学院硕士研究生, 研究方向: RFID 防碰撞算法、可信认证协议。
E-mail: huyingmeng89@163.com