

基于分组机制的位仲裁查询树防碰撞算法

付钰¹, 钱志鸿¹, 程超², 刘晓慧¹

(1. 吉林大学通信工程学院, 吉林 长春 130012; 2. 长春工业大学计算机科学与工程学院, 吉林 长春 130012)

摘要: 提出了一种基于分组机制的位仲裁查询树(GBAQT, bit arbitration query tree based on grouping mechanism)算法。该算法根据标签 ID 自身特征分组, 采用 3 位仲裁位来取代传统 1 位仲裁识别标签的方式, 通过碰撞位信息得到传输数据, 从而能避免一些空闲时隙。算法的性能分析和仿真结果表明, GBAQT 防碰撞算法具有较少的总时隙数, 系统效率和时隙利用率也明显优于其他算法。

关键词: RFID; 标签防碰撞; 查询树; 分组机制

中图分类号: TN92

文献标识码: A

Bit arbitration query tree anti-collision algorithm based on grouping mechanism

FU Yu¹, QIAN Zhi-hong¹, CHENG Chao², LIU Xiao-hui¹

(1. College of Communication Engineering, Jilin University, Changchun 130012, China;

2. School of Computer Science & Engineering, Changchun University of Technology, Changchun 130012, China)

Abstract: A bit arbitration query tree anti-collision algorithm based on grouping mechanism was proposed. GBAQT divided tags into two groups according to the tag ID's feature and used three arbitration bits to identify tags instead of using one bit in traditional methods. The reader can obtain the transmit data based on information of collision bits and thus avoid some idle timeslots. The algorithm performance analysis and simulation results show that GBAQT anti-collision algorithm has fewer total number of timeslots. Timeslot utilization and system efficiency of GBAQT algorithm are significantly better than the other algorithms.

Key words: RFID, tag anti-collision, query tree, grouping mechanism

1 引言

物联网已经成为当下最具潜力的产业之一, 其概念最初是在 1999 年被提出, 通过射频识别技术和传感器等信息传感设备, 把任何物品与互联网连接起来, 实现监控、定位功能的智能化网络。现如今, 尽管“物联网”的概念得到了进一步的完善, 但 RFID 技术作为物联网的感知层, 在物联网的核心技术中仍起着至关重要的作用。RFID 是一种非接触式自动识别技术, 利用空间的射频信号实现双向数据的无线通信。在《中国物联网 RFID2013 年度报告》中显示, 2013 年中国物联网

市场规模达到 4 896 亿元, RFID 市场规模达到 318.4 亿元, 比上一年增长了 34.6%。RFID 技术的应用领域更加广泛, 如军事、航空、医疗、制造业和零售业等^[1]。通常, RFID 系统主要由阅读器、电子标签以及后台服务器构成^[2]。阅读器主要负责控制和标签的双向通信; 每个电子标签都具有能被识别的、全球唯一的电子编码; 后台服务器是提供服务和管理的控制中心。

由于标签本身无法知道其他标签的存在, 各标签只能按照阅读器的命令发送信息。单个阅读器只有一个通信信道, 当 2 个或更多的标签同时向阅读器发送它们的 ID 信息时, 就会产生信号干扰,

收稿日期: 2014-11-19; 修回日期: 2015-01-20

基金项目: 国家自然科学基金资助项目(No.61371092); 吉林省科技厅基金资助项目(No.20140204019GX)

Foundation Items: The National Natural Science Foundation of China (No.61371092); Research Project of Science and Technology Department of Jilin Province (No.20140204019GX)

即发生标签碰撞问题，碰撞问题会严重地影响标签数据的完整性，使标签无法被正确识别。标签碰撞问题是 RFID 系统中重要的难题之一，极大地制约了 RFID 技术的发展。

目前，有很多优秀的协议来解决标签碰撞问题，这些算法主要分为 2 大类：一类是基于 ALOHA 的不确定性算法^[3-5]；另一类是基于树的确定性算法^[6,7]。前者包括纯 ALOHA 算法^[8]、时隙 ALOHA 算法、帧时隙 ALOHA 算法、动态帧时隙 ALOHA 算法及其改进算法等。后者包括了二进制搜索树算法、查询树 (QT, query tree) 算法^[9]、碰撞树算法^[10]等多种基于图的树型算法。

1.1 曼彻斯特编码

基于树的算法中常常使用曼彻斯特编码作为检测碰撞位的方法^[11-13]。曼彻斯特编码将数据包含在数据流中，每位编码有一跳变，用下降沿表示比特“1”，上升沿表示比特“0”。在 RFID 系统中，如果有多个标签同时传输不同比特，那么上升沿和下降沿将被抵消。然而在数据传输过程中曼彻斯特编码不允许出现这种情况，因此使用曼彻斯特编码能判断该位为碰撞位。本文亦采用曼彻斯特编码，已达到阅读器能在接收数据中准确检测出碰撞位的目的。例如，假设系统中存在 2 个标签，分别为 Tag A(0101)和 Tag B(0011)。当 Tag A 和 Tag B 同时传送 ID 的数据信息时，阅读器通过曼彻斯特编码接收到的信号为“0XX1”(X 表示碰撞位)，即第 1 位和第 2 位均为碰撞位。图 1 所示为标签使用曼彻斯特编码的响应过程。

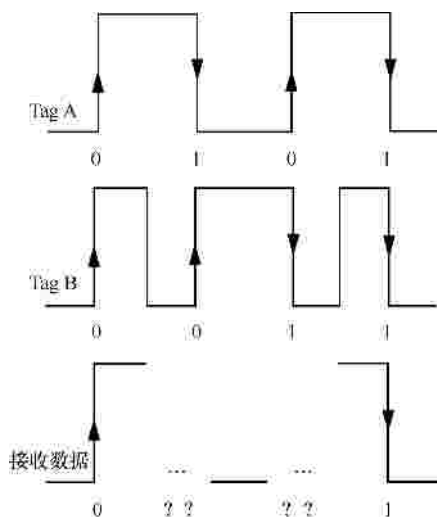


图 1 曼彻斯特编码响应过程

1.2 查询树算法

查询树算法主要由轮询和响应 2 个部分构成。在每一轮中，阅读器要发送一个字符串（即查询前缀），查询标签 ID 是否含有特定的前缀。只有 ID 与查询前缀相匹配的标签才能响应轮询命令。如果只有一个标签响应，那么该标签可以被成功识别，并将自己的 ID 发送给阅读器。如果有多个标签响应，那么就会发生标签碰撞，然后阅读器在该查询前缀后分别添加 0 和 1 构成了新的查询前缀。通过扩展前缀，直到只有一个标签的 ID 匹配。因此该算法可以识别所有的标签。

然而，在基本的查询树算法中如果没有标签的 ID 和查询前缀匹配，那么将会产生多余的空闲时隙。例如 2 个标签，它们的 ID 只有最后一位不同，其余各位均一致。那么为了识别这 2 个标签，将会产生许多的碰撞时隙和空闲时隙。同样，也会增加查询时间，使系统的查询效率下降。

1.3 基于查询树算法最近的研究成果

鉴于查询树算法的不足，研究学者们提出了许多基于查询树的改进算法。其中，Zhang 等^[14]提出了自适应四叉修剪查询树(A4PQT, adaptive 4-ary pruning query tree)算法。根据碰撞位的特点对四叉查询树进行改善，以减少不必要的空闲时隙。阅读器向读取范围内的标签发送查询前缀，匹配前缀的标签将除前缀外余下的 ID 发送给阅读器。当标签发生碰撞时，假设标签 ID 长度为 n bit，阅读器接收数据信息为 $p_1p_2...p_{c-1}p_cp_{c+1}...p_n$ p_c 为首碰撞位，那么可以分成 3 种情况。

- 1) 如果 $p_{c+1}=0$ ，那么阅读器将产生 2 个新的查询前缀，分别为 $p_1p_2...p_{c-1}00$ 和 $p_1p_2...p_{c-1}10$ ，并将原四叉树的“01”和“11”分支剔除。
- 2) 如果 $p_{c+1}=1$ ，那么阅读器将产生 2 个新的查询前缀，分别为 $p_1p_2...p_{c-1}01$ 和 $p_1p_2...p_{c-1}11$ ，并将原四叉树的“00”和“10”分支剔除。
- 3) 如果 p_{c+1} 也是碰撞位，那么阅读器将产生 4 个新的查询前缀，分别为 $p_1p_2...p_{c-1}00$ 、 $p_1p_2...p_{c-1}01$ 、 $p_1p_2...p_{c-1}10$ 和 $p_1p_2...p_{c-1}11$ ，不剔除四叉树任何分支。

因此，A4PQT 算法可以减少一些空闲时隙，所需要的总时隙为 $\frac{7n-1}{6}$ 至 $2n-3$ 之间。然而，如果 2 个标签的 ID 分别为“01”和“10”，那么阅读器通过曼彻斯特编码得到的数据为“XX”，根据 A4PQT

算法将要产生 4 个查询前缀，由此将产生 2 个额外的空闲时隙。

针对上述研究所存在的问题，本文基于分组机制和查询树识别，提出一种基于分组机制的位仲裁查询树(GBAQT, bit arbitration query tree based on grouping mechanism)防碰撞算法。在 GBAQT 算法中标签根据 ID 自身特征，通过 3 位仲裁的方式将未识别标签分成 2 组，阅读器根据各组标签特点和接收数据的碰撞位信息能得到传输数据，减少了大量空闲时隙。仿真结果显示，与其他标签防碰撞算法相比，GBAQT 算法具有较高的系统效率和时隙利用率，并且在较多标签存在的环境下具有明显的优势。

2 算法描述

目前，基于树的确定性算法中通常使用前缀来识别标签，在识别树中一般包括 3 个时隙。

碰撞时隙：在该时隙中，多个标签同时响应阅读器的查询，阅读器将不能识别任何标签，称该时隙为碰撞时隙。

成功时隙：在该时隙中只有 1 个标签响应阅读器，则标签能被成功识别，称该时隙为成功时隙。

空闲时隙：在该时隙中没有标签响应，则为空闲时隙。

2.1 基本思想

本文采用 3 位仲裁的方式来代替传统的 1 位仲裁，尽管采用多叉查询树能有效地减少碰撞时隙，但会带来大量的空闲时隙。因此，GBAQT 算法对空闲时隙数进行了优化，通过标签 ID 的自身特征分组，由碰撞信息尽量准确地推断出传输数据的组合形式。

分组机制描述如下。假设标签 ID 长度为 n ，标签的第 s 位开始相邻 3 位为 $p_s p_{s+1} p_{s+2}$ ，定义 \otimes 为异或运算符，如果 $p_s \otimes p_{s+1} = p_{s+2}$ ，则 $p_s p_{s+1} p_{s+2}$ 为第 0 组，表示为 $G=0$ ；如果 $p_s \otimes p_{s+1} \neq p_{s+2}$ ，则 $p_s p_{s+1} p_{s+2}$ 为第 1 组，表示为 $G=1$ 。例如标签 i 的 ID 中 3 位连续数据为 $1_4 0_5 1_6$ ，符合 $1_4 \otimes 0_5 = 1_6$ ，那么 $G=0$ 。另外，还发现以下规则：对于 $G=0$ 组的标签 $p_s p_{s+1} p_{s+2}$ 3 位数据符合其中任意 2 位异或都等于其余一位，由此定义 $G=0$ 组标签具有异或特征；而 $G=1$ 组标签， $p_s p_{s+1} p_{s+2}$ 3 位数据符合其中任意 2 位同或等于其余一位，由此定义 $G=1$ 组标签具有同或特征。

例如，在 RFID 系统中存在 2 个标签，分别为 Tag A(101)和 Tag B(110)，标签发送 ID 与组号信息，阅读器得到“1XX”和 $G=0$ 。根据 $G=0$ 组标签的异

或特征，可得到传输数据的组合形式为 101 和 110，能准确地判断标签 ID。相比 A4PQT 算法，可以减少 2 个空闲时隙。

2.2 算法相关指令

REQUEST($q_1 q_2 \dots q_k, G$)请求命令， $q_1 q_2 \dots q_k$ 为查询前缀，其中， $q_j \in \{0,1\}, 0 < k \leq n$ 。G 表示组号， $G \in \{0,1\}$ 。阅读器发送 REQUEST($q_1 q_2 \dots q_k, G$)命令，符合查询前缀为 $q_1 q_2 \dots q_k$ 且组号为 G 的标签响应，处于激活状态。

PUSH(QUERY)读写指令，将 QUERY 作为新的查询前缀压入栈 S 的底部。

SELECT(ID)选择指令，选择符合此 ID 的标签。

READ-DATA 读取指令，读出标签 ID 的数据信息。

SILENCE 静默指令，将标签置于静默状态，不再参与识别过程。

“//”表示连接符，将 2 组数据连接组合成一组，如“00//01”，表示将“00”和“01”连接成“0001”。

2.3 算法流程

GBAQT 算法流程如图 2 所示。

第 1 步 初始化，阅读器置查询前缀堆栈 S 和组号存储器 G 为空。

第 2 步 阅读器发送 REQUEST($q_1 q_2 \dots q_k$)。首次发送该命令时置查询前缀为 e ，即读取范围内的标签全部响应。

第 3 步 与查询前缀 $q_1 q_2 \dots q_k$ 相匹配的标签响应。如果没有标签响应，则为空闲时隙，转到第 6 步；否则转到第 4 步。

第 4 步 阅读器查询标签是否发生碰撞。如果未发生碰撞，则该标签可以被成功识别，阅读器发送 SELECT 和 READ-DATA 指令，对标签读写操作后阅读器发送 SILENCE 指令，将标签置为静默状态，转到第 6 步；如果发生碰撞，则转到第 5 步。

第 5 步 假设阅读器接收的数据为 $p_1 p_2 \dots p_{c-1} p_c p_{c+1} p_{c+2}$ ，其中 $p_i \in \{0,1\}$ ， p_c 表示最高碰撞位。从标签 ID 的 p_c 位开始的连续 3 位 $p_c p_{c+1} p_{c+2}$ 为一组，执行分组指令，相应的置标签内部组号存储器 G 为 0 或 1。标签发送数据“ $p_c p_{c+1} p_{c+2} // G$ ”。根据阅读器中组号存储器 G 接收数据是否碰撞可分为以下 2 种情况讨论。

1) 如果阅读器的组号存储器 G 发生碰撞，则先查询 $G=0$ 的标签，然后再查询 $G=1$ 的标签。若该组中没有标签响应，则为空闲时隙；若该组中只有一个标签响应，则可以直接识别，阅读器发送

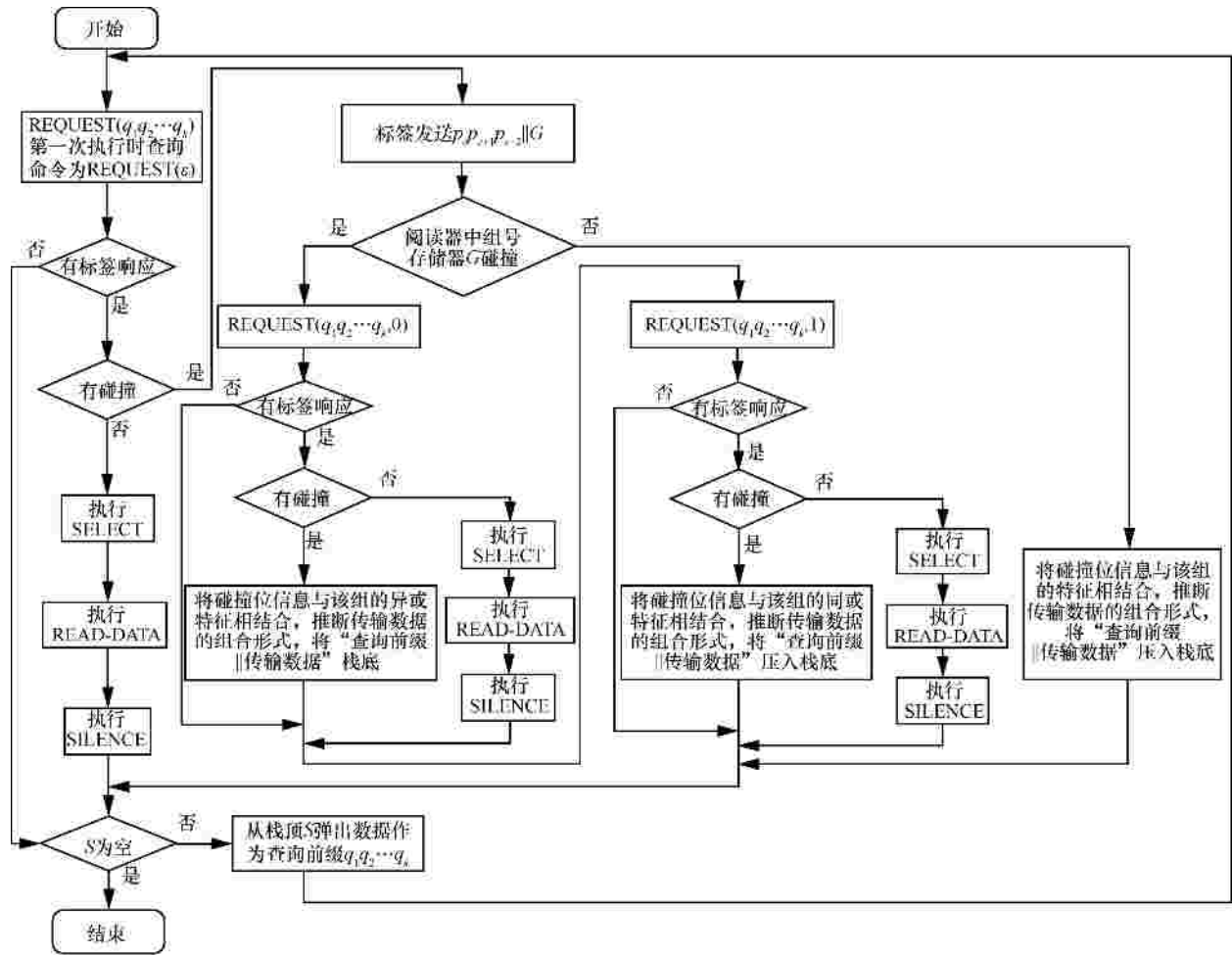


图 2 GBAQT 算法流程

SELECT 和 READ-DATA 指令,对标签读写操作后阅读器发送 SILENCE 指令,将标签置为静默状态;若该组发生碰撞,那么通过碰撞位信息,遵循其异或(或同或)特征可推断传输数据的组合形式,将“查询前缀//传输数据”压入栈底,转到第 6 步。

2) 如果阅读器的组号存储器 G 未发生碰撞,即为单一组号时,只查询符合该组的标签即可,转到第 6 步。

第 6 步 查询堆栈 S 是否为空。若栈 S 不为空,则从栈顶 S 弹出数据作为新的查询前缀,转到第 2 步;若栈 S 为空,则查询过程结束。

阅读器根据碰撞位信息,利用各组的异或(或同或)特征推断出传输数据的组合形式,具体描述如下, $G=0$ 的标签中,若 $p_c p_{c+1} p_{c+2}$ 中有 2 位碰撞位,利用该组的异或特征,从传输数据中已知的一位数据信息推断出另 2 位的组合形式,例如 $p_c p_{c+1} p_{c+2} = 1XX$,则发送 $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 101)$ 和 $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 110)$;若 $p_c p_{c+1} p_{c+2}$ 中的各

位均为碰撞位,则推断为 $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 000)$, $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 011)$, $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 101)$, $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 110)$ 。同样, $G=1$ 的标签中,若 $p_c p_{c+1} p_{c+2}$ 只有 2 位碰撞位,利用该组的同或特征,亦可以推断出这 2 个碰撞位的组合形式,例如 $p_c p_{c+1} p_{c+2} = 1XX$,则发送 $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 100)$ 和 $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 111)$;若 3 位均为碰撞位则发送 $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 001)$, $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 010)$, $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 100)$, $PUSH(q_1 q_2 \dots q_k p_1 p_2 \dots p_{c-1} // 111)$ 。

下面举例说明 GBAQT 算法识别标签的步骤。设有 6 个待识别标签,标签 ID 分别为 Tag1(000110)、Tag2(000101)、Tag3(000011)、Tag4(011111)、Tag5(011010)、Tag6(111111)。GBAQT 算法阅读器查询和标签响应过程如表 1 所示,对应的查询树结构如图 3 所示。

表 1 阅读器查询和标签响应情况

阅读器查询命令	响应的标签	碰撞信息	查询堆栈 S
REQUEST(e)	Tag1~Tag 6	碰撞,XXX, G=X	S 为空
REQUEST(e,0)	Tag1~Tag 5	碰撞,0XX	S={000,011}
REQUEST(e,1)	Tag6	识别	S={000,011}
REQUEST(000)	Tag1~Tag 3	碰撞,XXX, G=0	S={011,000000,000011,000101,000110}
REQUEST(011)	Tag4, Tag5	碰撞,X1X, G=1	S={000000,000011,000101,000110,011010,011111}
REQUEST(000000)	-	空闲	S={000011,000101,000110,011010,011111}
REQUEST(000011)	Tag3	识别	S={000101,000110,011010,011111}
REQUEST(000101)	Tag2	识别	S={000110,011010,011111}
REQUEST(000110)	Tag1	识别	S={011010,011111}
REQUEST(011010)	Tag5	识别	S={011111}
REQUEST(011111)	Tag4	识别	空

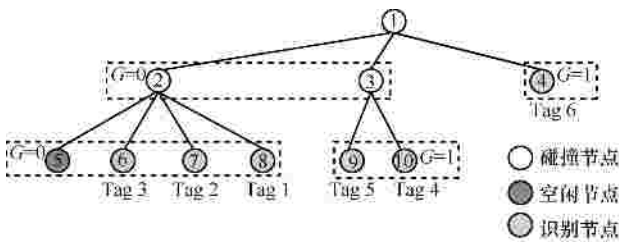


图 3 GBAQT 算法树结构

3 算法性能分析

在 RFID 系统中，衡量防碰撞算法的优劣主要有 2 个指标：总时隙数和系统效率。下面通过计算总时隙数和系统效率来分析 GBAQT 算法性能。假设系统中有 N 个标签，在第 L 层的时候，对于完全八叉树有 8^L 个节点，由于每个节点可分为 $G=0$ 组和 $G=1$ 组，那么该层有 $2 \cdot 8^L$ 组。由于每个标签的分配是相互独立的，其中有 k 个标签在第 L 层分配到同一组内的概率服从二项分布

$$P(N, K)_L = C_N^k p^k (1-p)^{N-k} \quad (1)$$

其中， $p = \frac{1}{2 \cdot 8^L}$ 表示在 $2 \cdot 8^L$ 组中选择一组的概率。

该组为空闲的概率为

$$P(N, 0)_L = (1-p)^N = \left(1 - \frac{1}{2 \cdot 8^L}\right)^N \quad (2)$$

该组能被成功识别的概率为

$$P(N, 1)_L = C_N^1 p (1-p)^{N-1} = Np \left(1 - \frac{1}{2 \cdot 8^L}\right)^{N-1} \quad (3)$$

该组发生碰撞的概率为

$$P(N, k)_L = 1 - P(N, 0)_L - P(N, 1)_L \quad (4)$$

若该组为碰撞时隙，那么可以将树结构分为以

下 3 种情况。

- 1) 只有 2 个子节点时，则一定没有空闲时隙；
- 2) 有 3 个子节点时，则必然产生 3 个非空闲时隙（其状态为碰撞还是成功时隙需有 $L+1$ 层进行判断）和一个空闲时隙；
- 3) 有 4 个子节点，则 4 个都是非空闲时隙。

因此，最佳树的情况应该没有空闲时隙，最差树的情况则是每个碰撞组有一个空闲时隙。

在一个碰撞组中，一个子节点为空闲的概率为 $\left(1 - \frac{1}{4}\right)^{N_0}$ ，一个子节点为非空闲的概率为 $1 - \left(1 - \frac{1}{4}\right)^{N_0}$ 。那么根据碰撞情况在该碰撞组中，有 2 个节点的概率为

$$p_2 = C_4^2 \left[\left(1 - \frac{1}{4}\right)^{N_0} \right]^2 \left[1 - \left(1 - \frac{1}{4}\right)^{N_0} \right]^2 \quad (5)$$

有 3 个节点的概率为

$$p_3 = C_4^3 \left(1 - \frac{1}{4}\right)^{N_0} \left[1 - \left(1 - \frac{1}{4}\right)^{N_0} \right]^3 \quad (6)$$

有 4 个节点的概率为

$$p_4 = \left[1 - \left(1 - \frac{1}{4}\right)^{N_0} \right]^4 \quad (7)$$

对于每组都只有 2 个节点的树结构，若只需要 L_1 层即可识别所有标签，那么总时隙

$$Q_2 = \sum_{l=0}^{L_1-1} 4^l + \sum_{l=0}^{L_1-1} 2 \cdot 4^l \quad (8)$$

每组都有 3 个节点的树结构，若需要 L_2 层即可识别所有标签，那么总时隙

$$Q_3 = \sum_{l=0}^{L_2-1} 6^l + \sum_{l=0}^{L_2-1} 2 \cdot 6^l + \sum_{l=0}^{L_2-1} 2 \cdot 6^{l-1} \quad (9)$$

每组都有 4 个节点的树结构,若需要 L_3 层即可识别所有标签,那么总时隙

$$Q_4 = \sum_{i=0}^{L_3-1} 8^i + \sum_{i=0}^{L_3-1} 2 \cdot 8^i \quad (10)$$

在 GBAQT 算法树结构中是 2 个节点、3 个节点和 4 个节点混合的情况,为了便于分析,这里按各个情况发生的概率占总体发生概率的百分比得到:发生 2 个节点的概率 $w_2 = \frac{p_2}{p_2 + p_3 + p_4}$; 发生 3

个节点的概率 $w_3 = \frac{p_3}{p_2 + p_3 + p_4}$; 发生 2 个节点的概率

率 $w_4 = \frac{p_4}{p_2 + p_3 + p_4}$ 。那么得到总时隙

$$Q(N) = Q_2 w_2 + Q_3 w_3 + Q_4 w_4 \quad (11)$$

空闲时隙数

$$Q_0(N) = Q_3 w_3 \quad (12)$$

系统效率为

$$h = \frac{N}{Q(N)} \quad (13)$$

4 仿真及分析

以 Matlab2012a 为仿真平台对算法进行仿真,阅读器读取范围内的标签数量取值范围为 100~1 000,且标签 ID 随机产生, ID 长度为 96 bit 进行仿真。下列结果仿真 20 次取平均值,下面将 GBAQT 算法与 A4PQT 算法、QT 算法、八叉树算法进行仿真,分别从空闲时隙数、总时隙数、系统效率和时隙利用率 4 个方面的性能进行对比。

图 4 为 GBAQT 算法、A4PQT 算法、QT 算法和八叉树算法空闲时隙数比较,可以看出 GBAQT 算法的空闲时隙数远远小于 A4PQT 算法、QT 算法和八叉树算法,并且随着 N 值的增加,其空闲时隙数的增加趋势缓慢而稳定。然而八叉树的空闲时隙数增加的速度最快,在标签数量为 1 000 时,八叉树的空闲时隙数几乎是 GBAQT 算法的 28 倍。因此,当存在大量标签时,GBAQT 算法的优势更加明显。

图 5 为 GBAQT 算法、A4PQT 算法、QT 算法和八叉树算法总时隙数的比较。识别 N 个标签时,GBAQT 算法所需的总时隙数明显小于 A4PQT 算法、QT 算法和八叉树算法,并且在标签数量较多时,可有效地减少搜索的总时隙数。

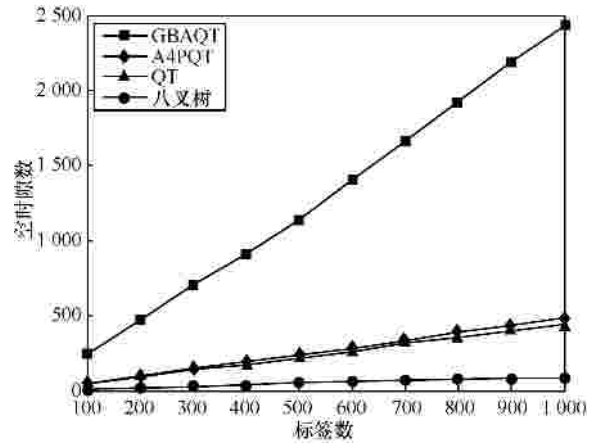


图 4 空闲时隙数对比

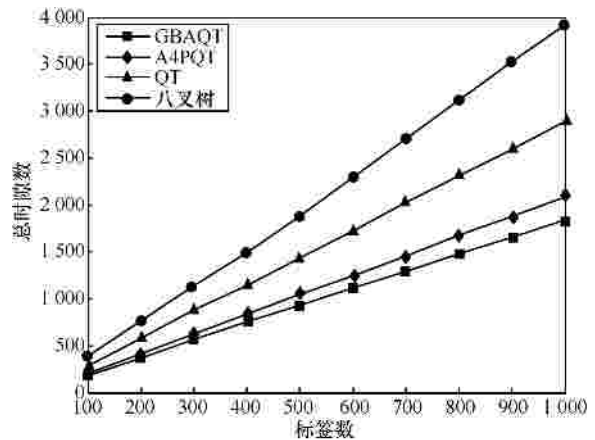


图 5 总时隙数对比

图 6 对 GBAQT 算法、A4PQT 算法、QT 算法和八叉树算法的系统效率进行了比较,可以看出 GBAQT 算法的系统效率明显要优于 A4PQT 算法、QT 算法和八叉树算法,可维持在 0.54 左右。其次是 A4PQT 算法效率,不超过 0.49。四叉树算法系统效率次之,保持在 0.35 左右。八叉树算法的系统效率最低,在 0.25 至 0.28 之间浮动。

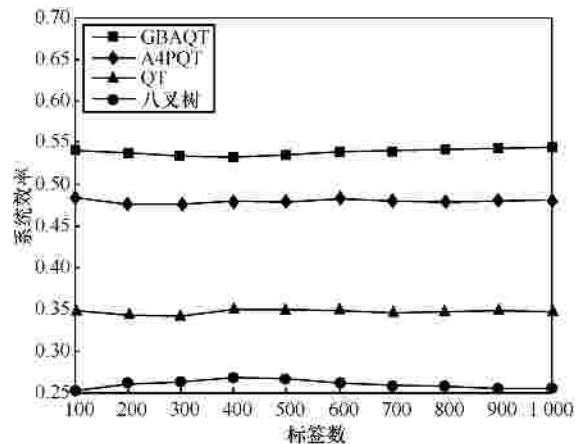


图 6 系统效率对比

图 7 表示 GBAQT 算法、A4PQT 算法、QT 算法和八叉树算法的时隙利用率，其中时隙利用率 $= \frac{\text{非空时隙}}{\text{总时隙}}$ 。由于 GBAQT 算法可以利用碰撞时隙

反演出标签，因此碰撞时隙和成功时隙都是有用时隙。可以看出 GBAQT 的时隙利用率要优于其他 3 种算法，达到了 0.95。其次是 QT 算法，时隙利用率维持在 0.85 左右。A4PQT 算法次之，在 0.77 附近浮动。尽管八叉树算法能减少碰撞时隙，但同时也伴随着大量空闲时隙的产生，因此其系统利用率最低，不超过 0.4。

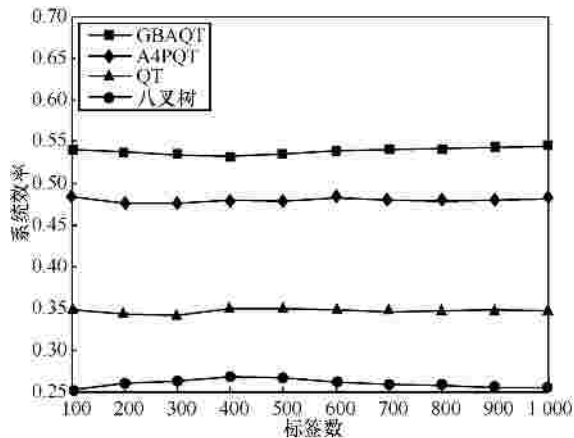


图 7 时隙利用率对比

5 结束语

本文基于分组机制和查询树识别，提出一种基于分组机制的位仲裁查询树防碰撞算法，该算法采用 3 位仲裁位的方式减少碰撞时隙，并根据标签 ID 自身特征分组，通过碰撞位信息得到的传输数据从而能避免一些空闲时隙。理论和仿真分析表明：GBAQT 算法能有效地减少空闲时隙数，具有较低的总时隙数，并且能有效地提高系统效率和时隙利用率。在标签数量较多的情况下，GBAQT 性能明显优于其他算法。

参考文献：

- [1] BLETSAS A, KIMIONIS J, DIMITRIOUS A G, et al. Single-antenna coherent detection of collided FM0 RFID signals[J]. IEEE Transactions on Communications, 2012, 60(3):756-766.
- [2] HSU C H, CHAO H C, PARK J H. Threshold jumping and wrap-around scan techniques toward efficient tag identification in high density RFID systems[J]. Information Systems Frontiers, 2011, 13(4):471-480.
- [3] 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.

- [4] LI M, QIAN Z H, ZHANG X, et al. Slot-predicting based ALOHA algorithm for RFID anti-collision[J]. Journal on Communications, 2012, 32(12):43-50.
- [5] ZANELLA A. Adaptive batch resolution algorithm with deferred feedback for wireless systems[J]. IEEE Transactions on Wireless Communications, 2012, 11(10):3528-3539.
- [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] JIANG Y, ZHANG R N. An adaptive combination query tree protocol for tag identification in RFID systems[J]. IEEE Communications Letters, 2012, 16(8):1192-1195.
- [8] YANG C N, HU L J, LAI J B. Query tree algorithm for RFID tag with binary-coded decimal EPC[J]. IEEE Communications Letters, 2010, 16(10):1616-1619.
- [9] LAW C, LEE K, SIU K Y. Efficient memory-less protocol for tag identification[C]//The 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. Boston, c2000: 75-84.
- [10] HUSH D R, WOOD C. Analysis of tree algorithms for RFID arbitration[C]//IEEE International Symposium on Information Theory. Cambridge, c1998: 107.
- [11] JIA X L, FENG Q Y, YU L S. Stability analysis of an efficient anti-collision protocol for RFID tag identification[J]. IEEE Transactions on Communications, 2012, 60(8):2285-2294.
- [12] LIU X H, QIAN Z H, ZHAO Y H, et al. An adaptive tag anti-collision protocol in RFID wireless systems[J]. China Communications, 2014, 11(7):117-127.
- [13] GAO J, WANG J, HE J, et al. QSA: query splitting-based anticollision for mobile RFID-based internet-of-things[J]. International Journal of Distributed Sensor Networks, 2013, DOI:10.1155/2013/674698.
- [14] CHEN Y H, HORNG S J, RUN R S, et al. A novel anti-collision algorithm in RFID systems for identifying passive tags IEEE Transactions on Industrial Informatics, 2010, 6(1):105-121.
- [15] ZHANG W, GUO Y J, TANG X M, et al. An efficient adaptive anti-collision algorithm based on 4-ary pruning query tree[J]. International Journal of Distributed Sensor Networks, 2013, 14(2): 135-143.

作者简介：



付钰 (1990-), 女, 吉林省吉林市人, 吉林大学博士生, 主要研究方向为 RFID 和物联网。

钱志鸿 (1957-), 男, 吉林长春人, 吉林大学教授、博士生导师, 主要研究方向为物联网、RFID、Wi-Fi、无线传感器网络和无线定位等。

程超 (1984-), 男, 吉林长春人, 博士, 长春工业大学讲师, 主要研究方向为短距离无线通信理论。

刘晓慧 (1989-), 女, 吉林长春人, 吉林大学硕士生, 主要研究方向为无线通信理论。