

动态密集标签环境下 RFID 多路接入控制协议研究

杨 健¹ 王永华¹ 蔡庆玲²

(广东工业大学自动化学院 广州 510006)¹ (中山大学工学院 广州 510275)²

摘要 提出一种静态密集标签环境下基于 ALOHA 和查询树的多路接入控制协议(S-TMAC),分析了该协议的原理、步骤及性能参数。接着将其扩展为动态密集标签环境下的多路接入控制协议(M-TMAC),定义了协议结构,建立了动态系统模型,并对标签识别时间、入射角、最大速度等参数进行了分析。仿真表明,动态系统模型下的 M-TMAC 的标签识别时间与 S-TMAC 相当,仅为 ALOHA 算法的 1/2、树形算法的 1/3;且在满足预设识别率和干扰率时,允许的标签有较大的入射角和运行速度。

关键词 射频识别,动态密集标签环境,多路接入控制协议

中图分类号 TN911 **文献标识码** A

Research on RFID MAC Protocol under Dense Mobile Tag Environment

YANG Jian¹ WANG Yong-hua¹ CAI Qing-ling²

(School of Automation, Guangdong University of Technology, Guangzhou 510006, China)¹

(School of Engineering, Sun Yat-sen University, Guangzhou 510275, China)²

Abstract This paper firstly presented a RFID MAC protocol based on ALOHA and query tree for static tag environment (S-TMAC), then analyzed the fundamental, process and performance. Secondly, S-TMAC was improved to be M-TMAC which is suitable for mobile tag environment. Protocol structure was defined, and system model was built. Optional reader deployment strategies were put forward. The discussion of the tag identifying time, incidence angle and speed limit was followed. Lastly, simulations indicate that the tag identifying time in M-TMAC and S-TMAC are similar, which is half of ALOHA-based and one-third of tree-based. Furthermore, M-TMAC has low extra system cost, upper tag speed limit and tag incidence angle limit when satisfying default identifying rate and interference rate.

Keywords RFID, Mobile dense tag environment, MAC protocol

1 引言

标签 MAC 协议主要可分为频分复用、码分复用、空分复用和时分复用。由于被动标签对读写器发射频率的非选择性以及标签有限的硬件资源,频分复用和码分复用均不适合用于被动标签。空分复用 MAC 对标签的位置和天线的方向性要求较高,同样不适用于功能简单的被动标签。时分复用 MAC 协议主要分为基于时隙的 ALOHA 算法和基于二叉树搜索的树形算法两大类。ALOHA 算法^[4,5]中读写器与标签之间的通信负担虽轻,但可能发生标签漏读,且标签平均识别时间较长,系统吞吐率不高;树形算法^[6]在理想条件下可对标签进行无遗漏识别,但增加了通信负担,导致功耗上升,且较前者标签平均识别时间更长。

此外,传统时分复用标签 MAC 协议应用条件大都理想,标签位置相对固定,未考虑大量标签快速进出系统的情况以及实际信道干扰对识别率所造成的影响。而现实应用中,

标签在识别区域内的运动状态大多是随机的,运动轨迹及速度可变,因此针对此类问题建立的动态识别模型必须降低对标签数量、速度、位置等因素的要求。

由此可见,建立预设识别率、信道干扰率及动态密集标签环境下的标签识别模型,研究具有识别速度快、系统开销少等特点的标签 MAC 协议,是非常具有现实意义的。

2 问题描述

本节先提出静态密集标签环境下的标签 MAC 协议(Static Tag MAC, S-TMAC),并对其关键部分进行讨论。接着将 S-TMAC 扩展为动态密集标签环境 MAC 协议(Mobile Tag MAC, M-TMAC),并对其各层结构、动态模型及识别过程、性能指标等进行分析。

2.1 S-TMAC 协议

S-TMAC 的主要思想如下:读写器根据标签数量将其分为若干组,设置合适的帧长(帧内时隙数),并依组对标签进行

到稿日期:2012-06-13 返修日期:2012-10-22 本文受国家自然科学基金项目(61102034, 61071038, 61172156),广东省自然科学基金项目(9451009001002607),广东省教育部产学研项目(2011A090200128),广州市应用基础重点项目(11C42090780),广东工业大学博士启动基金项目(13002)资助。

杨 健(1982-),男,博士,讲师,主要研究方向为射频识别技术、无线认知网技术,E-mail:jimmyyoungyj@163.com(通信作者);王永华(1979-),男,博士,讲师,主要研究方向为通信与网络技术;蔡庆玲(1966-),女,博士,副教授,主要研究方向为 RFID 技术及其安全问题。

识别。每组标签在其随机选择的时隙内响应读写器,对于多标签响应的碰撞时隙,则采用查询树算法识别,并在完成当前时隙内标签的识别后才进入下一时隙。S-TMAC 的识别过程总体可分为:分组动态帧识别阶段和碰撞时隙的查询树识别阶段,具体又可分为以下几个部分:1. 估算未读标签数量;2. 根据未读标签数量确定最优帧长和分组;3. 每帧识别一组标签,帧内按时隙顺序识别,并对碰撞时隙采用查询树算法识别。

2.1.1 未读标签数量的估计

S-TMAC 采用与文献[5]类似的标签数量估算方法:设初始帧长为 N_{mi} , 初始分组数为 G_{mi} , 依次计算标签数量 n 在某个范围内变化时读写器统计到的实际时隙数向量 $c=(c_0 c_1 c_k)^T$ (c_0 :空时隙数, c_1 :单标签时隙数, c_k :碰撞时隙数)与理论时隙数向量 $i=(i_0 i_1 i_k)^T$ (i_0 :空时隙数, i_1 :单标签时隙数, i_k :碰撞时隙数)之间的距离 ϵ , 把 ϵ 最小值时 n 的取值看作是未读标签数量,即:

$$\epsilon = \min_n |i - c| \quad (1)$$

式中, n 的范围为 $[c_1 + 2c_k, 2(c_1 + 2c_k)]^{[5]}$, 式(1)中向量 i 中有 k 个标签的时隙数由下式计算得到:

$$i_k^{N_{mi}, n} = G_{mi} N_{mi} \binom{n/G_{mi}}{k} \left(\frac{1}{N_{mi}}\right)^k \left(1 - \frac{1}{N_{mi}}\right)^{n/G_{mi} - k} \quad (2)$$

$(k \leq \lceil n/G_{mi} \rceil)$

2.1.2 帧长和分组数的确定

分组动态帧识别阶段较高的帧识别率(单标签时隙数与帧长之比)可以减少碰撞时隙数和碰撞标签数,从而减少查询树识别阶段时间。此外,在保证帧识别率的前提下,通过增加分组数来减少每帧识别时间,从而减少总识别时间,但过多分组数又会增多读写器操作,进而增加总识别时间,因此需要合理选择帧长和分组数。在本项目组已取得的前期研究成果的基础上^[2], 采取以下分组分帧策略,即:若 $n \leq N_{max}$, 则 $N=n$, $G=1$;若 $n > N_{max}$, 则 $N=N_{max}$, $G > 1$ 。 G 的确定应遵循:当标签数量 n_k 的增加使分组数为 k 的系统效率(识别标签数与总时隙的比值)即将小于分组数为 $k+1$ 组后的系统效率时,则将分组数增加为 $k+1$ 组;反之亦然。可得分组变化临界点的标签数为:

$$\frac{i_1^{N_{max}, \frac{n_k}{k}}}{N_{max}} = \frac{i_1^{N_{max}, \frac{n_k}{k+1}}}{N_{max}} \quad (k=1, 2, \dots)$$

$$\Rightarrow n_k = (k^2 + k) \log_{\left(1 - \frac{1}{N_{max}}\right)} \frac{k}{k+1} \quad (k=1, 2, \dots) \quad (3)$$

由式(3)可确定任意标签数量时的最优帧长 N 和分组数 G 。

2.1.3 碰撞时隙内的查询树识别

对于碰撞时隙, S-TMAC 采用查询树算法加以识别,具体过程为:读写器初始化查询序列栈初始值为 '0', 并向碰撞标签广播, 标签从最高位起检查其 EPC, 若 EPC 中包含该查询序列, 则响应读写器, 若存在标签碰撞, 则读写器将原查询序列后增加 1 ('0' 或者 '1') 位产生两个新的查询序列, 并将其中最低位为 '1' 的入栈, 最低位为 '0' 的则作为下一个查询序列向标签广播。若仍导致标签碰撞, 则重复上一过程; 若成功识别标签或无标签响应, 则出栈查询序列用作下一查询周

期。每个碰撞时隙内查询树识别过程的开始以 '0' 查询序列的发出为标志, 同时设置栈深寄存器。

跟踪当前栈深, 当其减为 0 时, 结束查询树识别过程。图 1 以第 1 帧为例给出了 S-TMAC 协议示意图, 其中 * 代表时隙内的标签, Φ 表示时隙为空, 虚线箭头指向查询栈表示入栈操作, 指出则表示出栈操作。图中第 1 帧的第 $i+1$ 时隙发生碰撞, 落入其内的 3 个标签 EPC 为: "10001000"、"01010000" 和 "01110000" (8 位仅为示意)。读写器初始化栈深寄存器 $SDepth$ 为 0, 用于跟踪栈的深度, 每次入栈操作时 $SDepth$ 增 1, 出栈操作时减 1, 查询树识别过程的结束以 $SDepth$ 减为 0 为标志, 图 1 中的 $SDepth$ 值是每次栈操作之后的值。

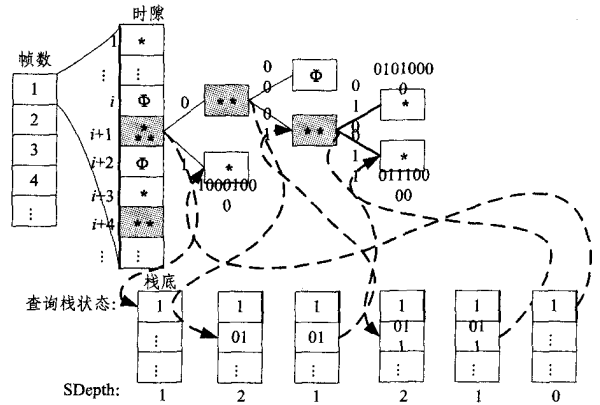


图 1 S-TMAC 协议示意图

2.2 M-TMAC 协议

2.2.1 协议设置

为使协议尽量简单易行且具有通用性, 参考了 Auto-ID 实验室的 900MHz C0 协议^[7] 的设置。读写器发出的数据符号分为 '0'、'1' 和 'Null' 3 种, 其中, '0'、'1' 用于构成读写器指令, 'Null' 用作标志读写器指令的起止及时隙的间隔; 数据所需时间为 $12.5 \mu\text{s}/\text{bit}$ 。标签发送的数据包括 64 位 EPC、16 位 CRC 以及少量的状态数据; 数据发送时间分别为: EPC 和 CRC 为 $4 \mu\text{s}/\text{bit}$, 状态数据为 $12.5 \mu\text{s}/\text{bit}$ 。

其他设置如下: 复位时间 T_{ri} 为 $800 \mu\text{s}$, 校准时间 T_{cal} 分为频率校准时间 (8 个周期为 $116 \mu\text{s}$ 的脉冲 + 8 个宽度为 $6 \mu\text{s}$ 的间隔低电平) 和数据校准时间 (3 个参考数据用于判断数据 '0'、'1' 和 'Null', 分别为 $10.5 \mu\text{s}$, $13.75 \mu\text{s}$ 和 $17.5 \mu\text{s}$), 因此 $T_{cal} = (116 + 6) \times 8 + 10.5 + 13.75 + 17.5 = 1017.75 \mu\text{s}$ 。

2.2.2 协议结构

与静态密集标签环境相比, 动态密集标签环境需考虑标签处于读写器覆盖区域内时间的限制, 以保证标签在离开时能以预设概率 P 被识别, 可通过提高帧识别率和增加标签经历的总帧数来实现。静态环境下, 整个识别过程只需对标签激活 (复位和校准) 一次; 而在动态密集标签环境下, 随时可能有新标签进入, 因此 M-TMAC 以一次激活 + 一帧识别为一个周期, 整个识别过程由若干个这样的周期组成, 如图 2 所示。每个周期开始后, 发送激活信号更新现存标签数量, 根据新的标签数量估计值重新确定最优帧长和分组数, 并通过读写器指令向标签广播, 标签接收后将在其范围内随机选择所属组和时隙, 其中第一组标签将在随后的时隙序列里响应读写器。该帧识别结束后, 剩余组内的标签和该周期内进入系

统的新标签构成新的现存标签,并进入下一周期,如图3所示。考虑到标签数量通常在相邻周期内相差不大,为了减少估计标签数量所用时间,用上一帧实际帧长和分组数来设置当前的帧初始帧长和分组数。此外,由于M-TMAC帧周期较短,新标签进入后处于未激活状态的等待时间最长不会超过一次激活时间+一个帧周期(下节将证明该等待时间小于150ms),之后便被激活并分组识别,因此读写器对新标签的反应速度是很快的。



图2 M-TMAC识别过程总体结构

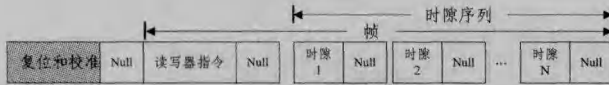


图3 M-TMAC识别帧结构

图4为单个时隙的内部结构,读写器在时隙开始后检测标签响应,不同的时隙状态将采取不同的操作:1. 单标签时隙,读写器识别该标签后发送ACK信号使该标签静默;2. 空时隙,读写器发送NACK和Null进入下一时隙;3. 碰撞时隙,读写器无法识别该时隙内所有标签,于是发送NACK后进入查询树识别过程。具体如下:读写器发送查询指令,其包含查询树中当前查询序列以及所要识别标签所属结点的深度;接着读写器检测标签响应状态,若单标签响应则发送ACK,无标签和多标签响应均发送NACK;然后等待满足查询条件的标签响应,根据新反馈的标签响应状态再发送新的查询指令,直到当前结点深度减为零,退出查询树过程,至此该碰撞时隙内的标签识别完毕。

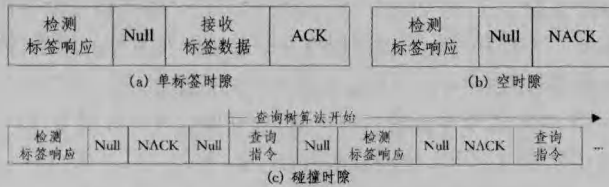


图4 M-TMAC帧内时隙结构

2.2.3 动态密集标签环境下的系统模型

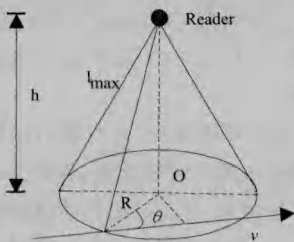


图5 动态环境下的系统模型

动态密集标签环境下,标签在读写器覆盖区域内的停留时间主要受两方面制约:1. 读写器部署方式;2. 标签运动方式。为此,我们在文献[8]模型的基础上加以改进。文献[8]将单读写器固定在流水线正上方,标签在流水线上做匀速直线运动。该模型对标签数量、运行速度及位置要求苛刻,实用场合有限。对其改进后的模型如图5所示,设标签均在同一平面内直线运动,运动方向任意,且最大速率为 v_{max} 。读写器到

标签位移平面距离 $h=5m$,最大作用距离 $l_{max}=10m$ 。读写器在标签位移平面上的作用区域可看作一个半径为 R 的圆,且圆心 O 即为读写器到该平面的垂心,并定义标签运动方向与进入点到圆心 O 连线的夹角为入射角 θ 。

标签处于读写器覆盖区域内的时间 t 可分为标签激活时间 t_a 和标签识别时间 t_i ,有 $t=t_a+t_i$ 。考虑到新标签进入后最长等待时间为一个识别周期时间,即 $T_{rst}+T_{cal}+T_f$,为保证标签在进入后至少被激活一次,应设置标签激活时间至少与该值相等,即 $t_a=T_{rst}+T_{cal}+T_f$ 。当标签数量大于最大帧长时,标签被分为 G 组识别,且标签平均需经过 G 帧才能参与识别,所以标签在读写器覆盖区域内经历的识别帧数 s 应满足:

$$s = \left\lfloor \frac{t-t_a}{(T_f+T_{rst}+T_{cal}+T_{null})} \right\rfloor \geq G \quad (4)$$

同时考虑到通信信道存在干扰,导致标签在该帧内无法识别的情况。设信道干扰率为 P' ,因S-TMAC在理想条件下帧识别率为100%,故此时M-TMAC的帧识别率应为 $1-P'$ 。此外, s, P 以及 P' 之间还应满足: $1-P \geq P'^s$,由此可得到标签在信道干扰率为 P' 时,为达到总识别率 P 需要参与的最小帧数 s_{min} 为:

$$s_{min} = \lfloor \log_{P'}(1-P) \rfloor \quad (5)$$

已知 $T_f = nT_{tag}/G, T_{rst} = 800\mu s, T_{cal} = 1017.75\mu s, t = 2\sqrt{l_{max}^2 - h^2} \cos\theta/v$,代入 $s \geq Gs_{min}$ 可得标签最大入射角 θ_{max} 和标签最大运行速率 v_{max} 的关系:

$$\theta_{max} \leq \arccos \left[\frac{G \lfloor \log_{P'}(1-P) \rfloor (T_f + 1830.25) + T_f + 1817.75}{2\sqrt{l_{max}^2 - h^2}} v_{max} \right] \quad (6)$$

3 性能分析

本节对所提出的动态识别模型中单标签平均识别时间 T_{tag} 等展开分析。无论是动态还是静态密集标签环境,其都满足帧长等于标签数的条件。若标签的入射角和速率满足要求,则相同标签数量时,动态密集标签环境下每帧的碰撞时隙数和单个碰撞时隙内的标签数应和静态密集标签环境下的相当,因此它们的单标签平均识别时间和查询栈平均深度应相当。我们将通过分析静态密集标签环境来得到动态密集标签环境下的 T_{tag} 和 $SDepth$ 。

由图2知,标签识别时间 T =复位时间 T_{rst} +校准时间 T_{cal} +分组数 $G \times$ 帧识别时间 T_f 。又由图4可知,单标签时隙仅比空时隙多出传输标签数据部分,碰撞时隙仅比空时隙多出查询树识别指令和传输标签数据部分,因此有: T_f =单个读写器指令时间 T_{cmd} +帧内时隙数 $N \times$ 单个空时隙时间 T_e +帧内碰撞时隙数 $Num_c \times$ 平均单个碰撞时隙内查询树算法指令时间 T_{query} +帧内标签数 $n/G \times$ 单标签传送数据时间 T_{data} 。所以 T 可表示为:

$$T = T_{rst} + T_{cal} + G(T_{cmd} + NT_e + Num_c T_{query} + \frac{nT_{data}}{G}) \quad (7)$$

式中, T_{cmd} 长度为19位(14位表示帧长+5位表示分组),ACK/NACK为1位,由协议设置可知传输时间均为 $12.5\mu s$;EPC为64位,CRC为16位,传输时间为 $4\mu s/bit$;由文献[8]知,检测标签响应接收的标签状态数据为2位时最佳,故检测

时间 $T_{chk} = 25\mu\text{s}$ 。因此式(7)可确定的参数有: $T_{cmd} = 12.5\mu\text{s} \times 20 = 250\mu\text{s}$, $T_e = T_{chk} + T_{Null} + T_{NACK} = 50\mu\text{s}$, $T_{data} = 4\mu\text{s} \times (64+16) = 320\mu\text{s}$, G, N 也可由分组和帧长选取策略得到, Num_c 可由概率知识得到, 为 $N[1-n/G(1-1/N)^{n/G-1} - (1-1/N)^{n/G}]$ 。因此式(7)可化简为:

$$T = 1817.75 + G(250 + 50N + Num_c T_{query}) + 320n \quad (8)$$

式(8)中只有 T_{query} 未知。设单个查询树平均指令数为 Num_{avg} , 读写器发出的连续两个查询序列之间的指令时间为一个基本查询周期, 记为 T_{cycle} , 由图 4(c) 可知 $T_{query} = Num_{avg} T_{cycle}$ 。所以应分别求解 Num_{avg} 和 T_{cycle} 。

3.1 单个查询树平均指令数 Num_{avg}

借鉴通信系统中计算冲突数据包所需重传次数的思想, 设标签数量为 n 时查询树识别单个碰撞时隙内标签所需的平均指令数为 Num_n , 考察其概率母函数 $Q_n(z)$:

$$Q_n(z) \approx \sum_{i=0}^n P\{Num_n = i | n\} z^i \quad (9)$$

当碰撞发生后, 新的查询序列使原 n 个标签以概率 $B_{n,i}$ 分裂为有 i 个标签的子组和 $n-i$ 个标签的子组, 其中 $B_{n,i} = \binom{n}{i} \left(\frac{1}{2}\right)^n$ ($0 \leq i \leq n$)。设该两个子组的 Num_i 和 Num_{n-i} 的概率母函数分别为 $Q_i(z)$ 和 $Q_{n-i}(z)$, 由概率母函数的性质, 有下式成立:

$$Q_n(z) = \sum_{i=0}^n B_{n,i} Q_i(z) Q_{n-i}(z) \quad (n=2, 3, \dots) \quad (10)$$

对 z 求导且令 $z=1$, 得到 Num_n , 即:

$$\begin{aligned} Num_n &= \sum_{i=0}^n B_{n,i} (1 + Num_i + Num_{n-i}) \\ &= 1 + 2 \sum_{i=0}^n B_{n,i} Num_i \quad (n=2, 3, \dots) \end{aligned} \quad (11)$$

当 n 为 0 或 1 时, 只需一个查询周期, 所以有 $Num_0 = Num_1 = 1$, 代入式(11)分别可得标签数为 2 和 3 时的查询周期数: $Num_2 = 5$, $Num_3 = 23/3$ 。又由文献[1]可知, 满足标签数和帧长相等条件时单个碰撞时隙内平均标签数为 2.3922, 所以单个查询树平均指令数 Num_{avg} 满足 $Num_2 < Num_{avg} < Num_3$, 即:

$$5 < Num_{avg} < \frac{23}{3} \quad (12)$$

3.2 单个查询树基本查询周期 T_{cycle}

设单个查询树指令平均长度为 Q_{bits} , 标签所处结点在树中的深度用 4 位表示(可满足 $2^4 = 65536$ 个标签), 由图 4(c) 有:

$$\begin{aligned} T_{cycle} &= (Q_{bits} + 4) \times 12.5 + T_{chk} + T_{ACK/NACK} + 3T_{Null} \\ &= 12.5Q_{bits} + 125 \end{aligned} \quad (13)$$

因此需求解 Q_{bits} 。设 $Q_{bits}(n)$ 为碰撞标签数为 n 时查询树指令的平均长度, 因单碰撞时隙内平均标签数为 2.3922, 则有 $Q_{bits}(2) < Q_{bits} < Q_{bits}(3)$ 。若能求出碰撞标签数量为 2 和 3 时的查询树指令的平均长度, 则可求出 Q_{bits} 的范围。若两个标签 EPC 的 $k-1$ 均相同而第 k 位不同, 则查询树指令为 k 位时标签必被识别。设 $p_2(k, m)$ 为碰撞标签数量为 2、EPC 为 m 位且第 k 位为第一个不相同位的概率, 则 $p_2(k, m)$ 满足:

$$p_2(k, m) = \left(\frac{1}{2}\right)^k \quad (k \leq m) \quad (14)$$

由式(14)可得 $Q_{bits}(2)$ 为:

$$\begin{aligned} Q_{bits}(2) &= \sum_{k=1}^m k p_2(k, m) = \sum_{k=1}^m k \left(\frac{1}{2}\right)^k \\ &= 2 + (m-2) \left(\frac{1}{2}\right)^m \end{aligned} \quad (15)$$

由于 m 一般较大, 因此近似可得 $Q_{bits}(2) = 2$ 。

碰撞标签数量为 3 时较复杂, 分两步求解 $Q_{bits}(3)$: 先求出 EPC 首次不同时的平均位数 $Q_{bits}(3, 1)$, 即一个标签被识别时的平均位数; 再求解另两个标签在剩余 EPC 位中被识别时的平均位数 $Q_{bits}(3, 2)$, 两者之和即为 $Q_{bits}(3)$ 。

设 $p_3(k, m)$ 为碰撞标签数量为 3、EPC 为 m 位且第 k 位为第一个不同位的概率, 易知 $p_3(k, m)$ 为:

$$\begin{aligned} p_3(k, m) &= 2 \times 3 \times \left(\frac{1}{2}\right)^3 \times \left[2 \times \left(\frac{1}{2}\right)^3\right]^{k-1} \\ &= 3 \times \left(\frac{1}{4}\right)^k \end{aligned} \quad (16)$$

进而 $Q_{bits}(3, 1)$ 可表示为:

$$\begin{aligned} Q_{bits}(3, 1) &= \sum_{k=1}^m k p_3(k, m) = 3 \sum_{k=1}^m k \left(\frac{1}{4}\right)^k \\ &= \frac{4}{3} + (m - \frac{1}{4}) \left(\frac{1}{4}\right)^m \end{aligned} \quad (17)$$

$Q_{bits}(3, 2)$ 可由式(15)求得:

$$Q_{bits}(3, 2) = 2 + (m-2) \left(\frac{1}{2}\right)^{m-Q_{bits}(3, 1)} \quad (18)$$

由式(17)、式(18)以及 $m=64$, 可近似得到 $Q_{bits}(3) = 4/3 + 2 = 10/3$ 。

所以 Q_{bits} 满足: $2 < Q_{bits} < 10/3$, 代入式(13)得 $150 < T_{cycle} < 166.7$, 结合式(12)得 $750 < T_{query} < 1277.8$, 再将 T_{query} 和 Num_{avg} 代入式(8)可得 T , 与 n 相除后即得到 T_{tag} 的最终表达式:

$$\begin{aligned} &\frac{1817.75 + G(250 + 50N + 750Num_c)}{n} + 320 < T_{tag} \\ &< \frac{1817.75 + G(250 + 50N + 1277.8Num_c)}{n} + 320 \end{aligned} \quad (19)$$

式中, n, N, G 分别为标签数量、帧长和分组数, 且 $Num_c = N[1-n/G(1-1/N)^{n/G-1} - (1-1/N)^{n/G}]$ 。

4 仿真验证

在 Matlab6.5 上建立了动态密集标签环境下的系统模型, 为了考察 M-TMAC 协议理论的正确性, 相对于传统静态 MAC 协议的优势, 并根据标签数量、入射角和最大速度的关系, 仿真内容包括: 1) T_{tag} 理论值和仿真值的比较; 2) 传统静态 MAC 协议 (ALOHA 算法中 DFSA^[1] 和 EDFSA^[2]、树形算法中 BT^[6] 和 QT^[6]) 与 M-TMAC 协议 T_{tag} 的比较; 3) θ_{max} 、 v_{max} 和 n 的关系。仿真数据取运行 50 次后的平均值。

图 6 为 T_{tag} 仿真值与理论值的比较。标签 EPC 为 64 位、CRC 为 16 位时, T_{tag} 程序仿真值大致处于 500~600ms 之间, 理论值上下限能较好地反映出 T_{tag} 的范围。分组策略导致理论值波形呈锯齿状变化, 跃变点即为分组变化点, 因为该点两侧的标签数量相差不大, 但分组数增加后组内的标签数急剧下降, 导致每帧的碰撞时隙数下降, 使得总识别时间减少。

图 7 为标签相对静止时 M-TMAC 与传统静态 MAC 协议 T_{tag} 的比较。设置静态环境的原因是传统静态 MAC 协议

均只对标签激活一次,不能用于动态标签环境,一旦标签以较快速度运动,新进入的标签均会被漏读。但即使是在静态标签环境下,M-TMAC的 T_{tag} 也是最小的,仅为ALOHA算法的1/2、为树形算法的1/3,这是由于最优分组和帧长的选取降低了碰撞时隙和空时隙的发生概率,碰撞时隙内的标签平均数量也达到最小,这保证了查询树算法较小的识别时间;发现一个碰撞时隙即采用查询树算法解决一个,降低了将不同碰撞时隙内标签全放在一起再次识别时原先不同时隙内的标签间发生碰撞的可能性,减少了总识别时间。

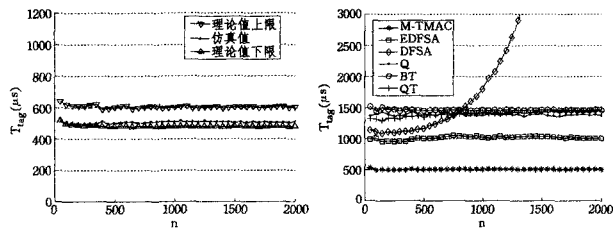


图6 T_{tag} 理论值与仿真值的比较 图7 传统静态MAC与M-MAC的 T_{tag} 比较

图8为单读写器部署, $P=99.9\%$, $P'=1\%$,M-TMAC协议 n 、 θ_{max} 和 v_{max} 的关系。可见, n 相同时, θ_{max} 与 v_{max} 成反比,且 n 越小, θ_{max} 相同时允许的 v_{max} 越大。当 n 较小时($n=100$), θ_{max} 为0和60°时允许的 v_{max} 分别高达110m/s和60m/s以上;即使在 n 很大的极端情况($n=2000$), θ_{max} 为0和60°时 v_{max} 也分别达到8m/s和4m/s左右,满足工程要求。

图9反映了占用读写器资源的情况,SDepth数据是将仿真平均值向右取整得到。易见,即使标签数量为2000时,实际查询栈栈深也不大于8,因此M-TMAC占用的额外读写器资源是很少的。

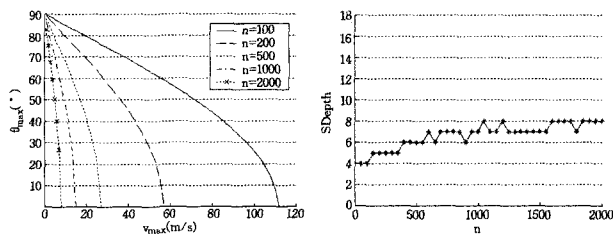


图8 M-TMAC v_{max} 和 θ_{max} 的关系 图9 查询栈深度变化情况

结束语 由此可见,无论是动态或是静态密集标签环境,M-TMAC协议的标签识别时间均较短,即使在静态密集标签环境下,与传统静态MAC协议相比,M-TMAC协议标签识别时间也仅为ALOHA算法的1/2、树形算法的1/3;在指定总识别率、信道干扰率的条件下,所支持的标签入射角及最大运行速度能满足绝大多数的应用场合。如 $P=99.9\%$ 、 $P'=1\%$ 、标签数量为500时,0°和60°入射角支持的标签最大速度分别达到60m/s和30m/s左右。将来的工作可在分析标签到达率与主要性能指标的关系、标签非直线运动情况及多读写器部署策略等方面展开。

参考文献

- [1] 杨健,王永华,蔡庆玲,等. EHiQ:一种基于增强型HiQ的RFID读写器MAC协议[J]. 计算机科学,2011,38(7):85-87,112
- [2] 杨健,詹宜巨,王永华,等. 基于分组动态帧和查询栈的射频识别反碰撞算法[J]. 系统仿真学报,2010,22(12):2920-2924
- [3] 孙文胜,金陈敏. 新型的RFID动态帧时隙ALOHA防碰撞算法[J]. 信息与控制,2012,41(2):233-237
- [4] Cha J-R, Kim J-H. Dynamic Framed slotted ALOHA algorithms using fast tag estimation method for RFID system[C]// Proceedings of the 3rd Consumer Communications and Networking Conference, 2006. USA; IEEE, 2006:768-772
- [5] Lee S-R, Joo S-D, Lee C-W. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification[C]// Proceeding of the 2nd Annual International Conference on Mobile and Ubiquitous Systems; Networks and Services, 2005. USA; IEEE, 2005:166-174
- [6] Myung J, Lee W, Srivastava J, et al. Tag-splitting: adaptive collision arbitration protocols for RFID tag identification [J]. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(6): 763-775
- [7] Auto-ID Center. Draft protocol specification for a 900MHz Class 0 Radio Frequency Identification Tag[OL]. <http://www.autoidlabs.org>, 2009-07-02
- [8] Venkatesh S, Mallareddy D, Sridhar R-S. A framework for fast RFID tag reading in static and mobile environments[J]. Computer Networks, 2008, 52(5):1058-1073

(上接第72页)

- [5] Xiao H. Towards parallel and distributed computing in large-scale data mining: A survey[R]. Technical University of Munich, 2010:1-30
- [6] http://developer.download.nvidia.com/compute/cuda/1_0/NV-VIDIA_CUDA_Programming_Guide_1.0.pdf
- [7] Almasi G S, Gottlieb A. Highly Parallel Computing[M]. Benjamin-Cummings publishers Co., Inc. Redwood City, CA, USA, 1989
- [8] Banko M, Brill E. Scaling to very very large corpora for natural language disambiguation[C]// ACL '01 Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA, USA, 2001:26-33

- [9] Hu W J, Song Q. An accelerated decomposition algorithm for robust support vector Machines[J]. IEEE Transactions on Circuits and Systems II; Express Briefs, 2004, 51(5):234-240
- [10] Do T-N, Nguyen V-H. A novel speed-up SVM algorithm for massive classification tasks[C]// Research, Innovation and Vision for the Future, 2008. RIVF 2008. IEEE International Conference, July 2008:215-220
- [11] Li Qi, Salman R, Test E, et al. Parallel multitask cross validation for Support Vector Machine using GPU[J]. Journal of Parallel and Distributed Computing, 2013, 73(3):293-302
- [12] Nickolls J, Buck I, Garland M, et al. Scalable Parallel Programming with CUDA[J]. Queue-GPU Computing, 2008, 6(2)
- [13] <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>