

物联网中基于多 reader 合作机制的 RFID-sensor 信息收集协议

李文秀^{1,2}, 郭亚红³, 李金宝^{1,2}, 郭龙江^{1,2}, 张绍斌⁴

(1. 黑龙江大学 计算机科学技术学院, 黑龙江 哈尔滨 150080; 2. 黑龙江省数据库与并行计算重点实验室, 黑龙江 哈尔滨 150080; 3. 黑龙江大学 信息科学与技术学院, 黑龙江 哈尔滨 150080; 4. 中国移动通信集团黑龙江有限公司 发展战略部, 黑龙江 哈尔滨 150080)

摘要: 为有效收集多 reader 环境下传感器节点的信息, 提出了一种高效率的数据收集协议-MRCIC。MRCIC 通过前次信息收集的时槽位置来分配本次 tag 回复信息的时槽位置, 缩短了检测区域范围内 tag 的时间。针对于 reader 冲突导致的无法检测到多 reader 相交区域内 tag 问题, MRCIC 采用没有相交区域的多个 reader 同时发送询问信息, 其他 reader 睡眠的方法来收集相交区域范围内 tag 信息, 提高了信息收集的准确率。理论分析与实验结果表明, MRCIC 比 AMDCU、ALOHA、BT 和 PIC 延迟更短, 准确率更高。

关键词: 多 reader; RFID; 传感器; 执行时间; 信息收集

中图分类号: TP212

文献标识码: A

文章编号: 1000-436X(2013)Z1-0292-11

RFID-sensor based multi-reader cooperation mechanism information collecting protocol in IOT

LI Wen-xiu^{1,2}, GUO Ya-hong³, LI Jin-bao^{1,2}, GUO Long-jiang^{1,2}, ZHANG Shao-bin⁴

(1.School of Computer Science and Technology, Heilongjiang University, Harbin 150080, China; 2.Key Laboratory of Database and Parallel Computing of Heilongjiang Province, Harbin 150080, China; 3.Information Science and Technology of Heilongjiang University College, Harbin 150080, China; 4. Strategy Development Department of China Mobile Communications Group, Heilongjiang Ltd, Harbin 150080, China)

Abstract: In order to collect the information of sensor nodes in multi-reader environment, the MRCIC information collection protocol was proposed. MRCIC information collection protocol uses the previous time slot position to assign the slot position where tags replied, and save the execution time of identifying the interrogated tags efficiently. Because of reader collisions problem, the intersecting regions tags couldn't be detected. Hence, in order to detecting them, MRCIC adopts the method that readers which did not have intersecting regions transmit interrogation information simultaneously and other readers sleep, to collect the intersecting regions tags' information. It improves the accuracy of information collection. Theoretical analysis and experimental results show that MRCIC gets shorter delay than AMDCU、ALOHA、BT and PIC, and it can reach the significantly accuracy rate.

Key words: multi-reader; RFID; sensor; execution time; information collection

1 引言

物联网成为继计算机、互联网之后, 世界信息产业的第三次革命, 无线射频识别(RFID, radio

frequency identification)技术是物联网应用的关键技术之一。RFID 是一种无接触、通过射频信号自动识别对象并获取相关数据的通信技术, 它以非接触、耗能少、抗污染能力强、存储数据量大、

收稿日期: 2013-06-20

基金项目: 国家自然科学基金资助项目(61070193); 黑龙江省杰出青年基金资助项目(JC201104); 黑龙江省高校科技创新团队建设计划基金资助项目(2013TD012); 教育部新世纪优秀人才支持计划基金资助项目(NCET-11-0955); 黑龙江省教育厅高校新世纪优秀人才支持计划基金资助项目(1252-NCET-011)

Foundation Items: The National Natural Science Foundation of China (61070193); Heilongjiang Province Funds for Distinguished Young Scientists (JC201104); Program for Group of Science and Technology Innovation of Heilongjiang Educational Committee (2013TD012); Program for New Century Excellent Talents in University (NCET-11-0955); Programs Foundation of Heilongjiang Educational Committee for New Century Excellent Talents in University (1252-NCET-011)

体积小、可重复使用等特性备受人们关注, 现已广泛应用于物流追踪、供应链管理、生产监控、食品安全、交通运输等众多领域^[1-4]。随着物联网的发展, 它将拥有更广泛的发展前景和应用领域。

典型的 RFID 系统包括若干个 reader 和若干个 tag, tag 绑定在产品上, 存储产品的信息, reader 识别 tag 上存储的信息。tag 分为 3 种, 分别为 passive tag、semi-active tag 和 active tag, 其中, active tag 传输距离最远, 但是需要电池供电, 它计算能力较强, 自身有能量可以主动发送数据。semi-active tag 传输距离次之, 也需配备一个小型电池。现在应用最广泛的是 passive tag, 它的特点是不需要电池, reader 发出射频信号, tag 反射 reader 所发出的能量, 并发送出存储在芯片中的产品信息。但它的传输距离是三者中最短的。在一个大型的 RFID 系统中, 由于阅读区域的限制, 一个 reader 一般无法读取到所有 tag 的信息, 所以需要有一个 reader 通过移动来读取所有 tag 的信息, 或者采用多个 reader 同时读取 tag 的信息。如果 tag 同时也在移动, 那么通过一个 reader 移动的方式类读取 tag 的信息就可能冗余或者漏读一些 tag 的信息, 因此在一个 tag 同时移动的大型 RFID 系统中, 布置多个 reader 同时读取 tag 信息就显得尤为重要。

近些年, 数据收集问题受到人们广泛关注^[5,6], 而单 reader 环境下将传感器与 RFID 相结合利用 tag 收集传感器节点信息的研究日益增多^[7-9], 而本文研究的是如何在多 reader 的环境下有效收集传感器节点信息。本文对三类单 reader 信息收集协议用在多 reader 的环境下的效率进行了分析, 这些协议都需要花费大量的时间去确定自己区域内的 tag, 它们在多 reader 的环境下执行时间长, 且效率很低。针对该问题, 本文提出了基于多 reader 合作机制的 RFID-SENSOR 信息收集协议 (MRCIC, multi-reader cooperation based information collection protocol), 该协议降低了数据收集的时间, 提高了准确率。

2 相关工作

现存的 RFID 技术的研究主要集中在如何快速有效地读取所有 tag 的 ID, 读取 tag 信息的方法分为 3 种: ALOHA-based^[10-12]、tree-based^[13,14]和 hybrid^[15,16]的方法。在 ALOHA-based 方法中, reader 产生多个

帧, 在每个帧开始时向 tag 广播这个帧的长度 f 和一个 0 到 $f-1$ 之间的随机数 r , tag 根据这些值将自己的 ID 散列到该帧的一个时槽, 并在该时槽到来时回复自己的 ID。在 tree-based 方法中, 每个 tag 都有一个初值为 0 的计数器。若某个时槽没有冲突发生, 所有 tag 计数器减一; 否则, reader 广播冲突信息, 发生冲突的 tag 随机选择 0 或 1 加入到自己的计数器中, 没有卷入冲突的 tag 将自己的计数器加一。当计数器为 0 时, tag 发送自己的 ID。在 hybrid 方法中, Chen Qian^[16]等人将 ALOHA-based 和 tree-based 方法进行结合去收集所有 tag 的 ID, 如果一个时槽内 tag 数量大于 32 就采用 tree-based 的方法, 否则使用 ALOHA-based 的方法去收集信息。

近些年, 有关 RFID 研究的新方向不断兴起, 在不读取所有 tag ID 的前提下估计总体数量的研究日益增多^[17,18], Hao Han^[17]等人利用 ALOHA-based 协议, 通过第一个非空时槽的位置来估算总体 tag 的数量。Yuanqing Zheng^[18]等人通过二叉树的随机路径上的第一个空时槽的位置来估算总体 tag 总数。Chen Qian^[16]等人则通过散列函数的方法让所有的 tag 的一半散列到第一个时槽中, 所有 tag 的四分之一散列到第 2 个时槽中, 那么只需要经过 $\lg n$ 时槽就可以估计出总体 tag 的数量。tag 的追踪以及隐私保护等问题也是研究的重点之一^[19,20]。

Shigang Chen^[7]等人第一个提出在单 reader 的环境下利用 tag 收集传感器节点信息的协议。该协议在已知所有 tag ID 的前提下, 利用 7 个散列函数的方法 (MIC) 收集与主动 tag 绑定的传感器节点信息。理论分析表明 MIC 的执行时间是下界时间的 1.61 倍, 但每个 tag 都需存储多个散列函数并进行多次计算。Li Wenxiu^[8]等人提出的 ATC 协议只使用一个散列函数, 该协议采用预先分配时槽的方法, 减少空时槽和冲突时槽的分配, 使得执行时间与时间下界基本相同。并且能够在几乎与下界相同的时间内收集所有信息。

然而, 这些方法大多数都适用于单 reader 环境。事实上, 单 reader 的应用范围有限, 一个 reader 很难收集到大量的 RFID tag 的全部信息, 因此在很多 RFID 场景中无法直接应用。在多 reader 的环境下利用基于单 reader 的信息收集协议来收集信息, 需要消耗大量时间去定义每个 reader 区域范围

内的 tag，针对该问题，本文提出了基于多 reader 合作机制的 RFID-SENSOR 信息收集协 MRCIC，该协议能够在很短的时间内收集所有传感器节点的信息。

3 系统模型

假设在某大型仓库中，有若干个 reader，每个 reader 的覆盖区域都有很多的物品，将物品的背面绑定一个结合传感器节点的 tag，周期地检测每个物品的信息(温度、保质期等)并传送给区域内的 reader。假设 reader 的覆盖区域的圆的直径是 s ，如果一个 reader 与另一个 reader 的距离不大于 s ，这 2 个 reader 就互为邻居。本节通过合理布置使得没有 reader 覆盖不到的地方并且最小化 2 个 reader 间的相交区域，由于信息收集的时间很短并且 2 个 reader 之间的相交区域很小，故一个 tag 在极短的时间内不可能从一个 reader 的区域范围内移动到其邻居 reader 区域范围外。

假设所有 RFID tag 数目的总和是 m ， t_{inf} 是 tag 向 reader 发送回复信息所消耗的时间，那么一个覆盖区域范围内有 n 个 tag 的 reader 收集信息的时间下界是 nt_{inf} 。在多 reader 的前提下，由于必须花费额外的时间去判定当前 reader 区域范围内有哪些 tag，要达到这个时间下界是很难的，因此如何减少信息收集时间使之与下界时间相近就是本文要研究的主要问题。

信号冲突是影响 tag 信息收集时间的瓶颈问题，信号冲突方式主要有 3 种，分别是 tag 冲突、reader 冲突、reader-tag 冲突，3 种方式如图 1 所示。在图 1(a)中，多个 tag 同时向 reader 回复信息，reader 无法辨别出每个 tag 回复的信息；在图 1(b)中，多个 reader 同时向 tag 发送信息，tag 无法解读每个 reader 的信息；在图 1(c)中，在 t_1 给 reader1 回复信息时，reader2 也在广播信息，此时 reader1 无法正确解读 t_1 的信息。本文通过合理布置 reader 的位置来避免 reader-tag 冲突，通过 MRCIC 算法来减少其他 2 种冲突。

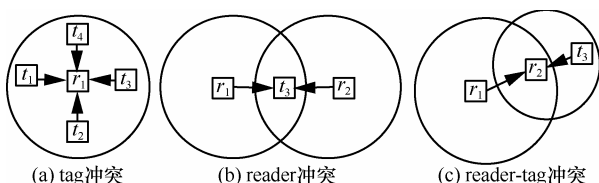


图 1 冲突图

本文假设所有 reader 都与同一个数据库相连接，数据库中存储每一个 reader 的覆盖范围内所有 tag 的信息，数据库的初始信息可由 Bo Sheng 等人提出的协议^[21,22]获得。每个 reader 都有自己的编号，每个 tag 的寄存器中都存储着自己区域范围内 reader 的编号。由于所有 tag 都有可能发生移动，每隔一段时间系统利用 MRCIC 对所有 tag 进行扫描以及信息收集，将最新的信息发送到数据库中。本文涉及的符号如表 1 所示。

符号	含义
m	系统中 tag 的总数
n_i	reader _{i} 覆盖范围内 tag 的总数
t_{est}	判断一个 tag 是否在自己区域范围内的等待时延
t_{id}	reader 广播 tag 的 ID 和 tag 的 crc 消耗的时间
t_{inf}	tag 将传感器节点信息回复给 reader 消耗的时间
t_{sum}	tag 发送自己 ID 以及传感器节点信息的总时间
t_{bit}	传输 1bit 数据消耗的时间
sn	tag 寄存器中存储的前次信息回复的时槽位置
T	FICP 算法的执行时间
T_0	flag 为 0 的 tag 信息收集时间
T_{sum}	MRCIC 算法的执行时间
S	reader 的覆盖区域的圆的直径
n_b	当前 reader 的邻居 reader 总数
n_{un}	从其他区域移入当前 reader 区域范围内的 tag 总数
n_{mis}	移出当前 reader 所在区域范围内的 tag
$Miss$	算法执行完没有被找到的 tag

4 3 种信息收集协议的比较

4.1 Polling 协议

在传统的 polling 协议(PIC)中，reader 依次广播每个 tag 的 ID，对应的 tag 接收到自己的 ID 后将传感器节点的信息回复给 reader。由于所有 reader 都不确定自己区域范围内对应的 tag，那么每个 reader 就需要消耗大量时间来广播所有 tag 的 ID。这里不考虑 reader 冲突造成的信息收集时间的延误，设一个 reader 的覆盖区域内 tag 总数为 n ，并且至少需要 $test$ 等待时延来判断一个 tag 是否在自己区域范围内，由于不知道区域范围内有哪些 tag，reader 需要广播所有 tag 的 ID 则 PIC 的时间是 $m tid + (m - n) test + nt_{inf}$ 。

4.2 ALOHA 协议

在传统的 ALOHA 协议中, reader 初始化一个帧, 并向 tag 广播这个帧的长度 f 和一个随机数 r , tag 通过 f 和 r 计算出自己 ID 散列的时槽位置, 当这个时槽到来时将自己 ID 和传感器节点的信息一起发送给 reader。如果当前帧中存在发生冲突的时槽, 那么 reader 再发送另一组 f 和 r , 重复上述过程, 直到该帧中没有冲突时槽的帧为止。由于 reader 不知道自己区域范围内有哪些 tag, 在该协议中, 每个 tag 都发送自己的 ID 以及传感器节点信息, 那么 reader 便可通过这些信息确定自己区域内的 tag 以及收集这些 tag 的信息。Shigang Chen^[7] 等人证明 ALOHA 的执行时间至少是 $ets_{\text{um}}n$, 其中, t_{sum} 是 tag 发送自己 ID 以及传感器节点信息的总时间。

4.3 BT 协议

在基于树的信息收集协议(BT)中, 每个 tag 都有一个初值为 0 的计数器。若某个时槽没有冲突发生, 所有 tag 计数器减一; 否则, reader 给所有 tag 广播冲突信息, 发生冲突的 tag 随机选择的 0 或 1 加入到自己的计数器中, 没有卷入冲突的 tag 将自己的计数器加一。当计数器值为 0 时, tag 回复信息给所在区域内的 reader。与 ALOHA 协议类似, 根据 D. R. Hush 等人提出的协议^[23], BT 的时间至少是 $2.885t_{\text{sum}}n$

前 3 种协议虽然能够解决多 reader 环境下信息收集问题, 但是时间消耗较长, 为了能够减少信息收集的时间, 本文提出了基于合作机制的 RFID-SENSOR 信息收集协议 MRCIC。

5 基于多 reader 合作机制的 RFID-sensor 信息收集协议

5.1 信息收集过程

第一次信息收集时, MRCIC 采用 Li Wenxiu 等人提出的 ATC 来收集传感器节点信息。在 ATC 中, reader 发送一个数组 vc 来分配每个 tag 回复信息的时槽位置, tag 计算自己在当前帧中回复信息的时槽位置 sn , 查看 $vc[sn]$ 是否为 1, 如果不是, tag 查看 vc 中 1 的总数, 并将其存储在初值为 0 的计数器中, 之后进入睡眠状态, 下一帧开始时再醒来; 否则, tag 计算 vc 的前 sn 比特中 0 的总数(记为 c), 并执行 $sn=sn-c$, 将 tag 回复信息的时槽位置 sn 加入到计数器中。在该帧中, 未睡眠的 tag

在 sn 时槽位置向 reader 回复信息, 回复信息后, tag 进入睡眠状态以保证不参与下一帧。也就是说, 如果在第 1 个帧 vc_1 中 1 的总数为 100, tag1 的时槽对应位置 $vc_1[sn_1]$ 为 0, tag1 将 100 加入到计数器 sn 中, 进入睡眠状态。第 2 个帧开始时醒来, 假设 sn_2 等于 10, $vc_2[sn_2]$ 为 1, 且 vc_2 中 sn_2 之前的 0 的个数 c 为 4, tag1 执行 $sn_2=sn_2-c=10-4=6$, 将 6 加入到 sn 中, 那么 tag1 的计数器中存储的值 sn 变为 $100+6=106$ 。reader 将本次信息收集的共花费的时槽总数存储在自己的计数器 Rsn 中。

下一次信息收集时, 每个 reader 首先广播自己的编号给区域内的 tag, tag 收到编号后, 与寄存器中存储的编号比较, 如果相同, 表示自己并未移出之前 reader 覆盖范围内; 否则, 表示自己移动到了新的 reader 的覆盖范围内。每个 reader 都有一个标志寄存器 $flag$, 初始化为 0, 移出之前 reader 覆盖范围内的 tag 将自己的 $flag$ 置 1, 所有 $flag$ 为 1 的 tag 将自己的计数器 sn 置为 0, $flag$ 为 0 的 tag 按照前次信息收集时计数器 sn 中存储值的时槽位置回复传感器节点的信息, 回复信息后进入睡眠状态, 待下一次信息收集时再醒来。Reader 只需按照事先存储在计数器 sn 中的值进行信息回复, 无需发送 vc 来重新分配 tag 回复信息的时槽位置, 节约了信息收集的时间。

由于有些 tag 可能已经移出 reader1 的覆盖范围内, 这样会导致某些时槽没有 tag 回复信息, 将这些 tag 定义为丢失的 tag。以 reader1 为例, 当这些时槽到来时, reader1 等待 t_{est} 时间未有 tag 回复信息, 则广播信息 0 来结束该时槽, 所有 $flag$ 为 0 的 tag 收到结束时槽信息 0 后提前结束该时槽。这样在信息收集过程中便能够检测出所有丢失的 tag, 而无需重新分配时槽。根据 Li Wenxiu 等人提出的 PTC, $T1$ 为 reader 广播信息的结束时刻到 tag 回复信息开始时刻的等待时延, $T2$ 为 tag 回复信息的结束时刻到 reader 再次广播信息开始时刻的等待时延, $T1=T2=t_{\text{min}}$ 。设丢失 tag 的总数为 n_{mis} , τ 为发送 1 bit 数据的时间, 则收集 $flag$ 为 0 的 tag 信息的时间 $T_0=(n-n_{\text{mis}}) \times t_{\text{inf}} + n_{\text{mis}} \times (t_{\text{est}} + t_{\text{min}} + \tau)$ 。

为了收集所有 $flag$ 为 1 的 tag 的信息, 首先需要确定这些 tag 的 ID, 传统的 ALOHA、BT 方法需要消耗大量的时间收集所有 tag 的 ID, 为了缩短时间, 本文提出 FICP(fast ID collection pro-

tocol), 使其能够在最短时间内收集所有 flag 为 1 的 tag 的 ID。

在 FICP 中, 首先利用 Chen Qian 等人提出的 ASAP 来估计所有 flag 为 1 的 tag 的总数。在 ASAP 中, reader 首先发送一个为 fa 长的帧, 让 flag 为 1 的 tag 根据 ASAP 中的几何散列函数方法 GSE 进行应答, 这个散列函数使得第 1 个时槽的标签数是整个标签数的 $1/2$, 第 2 个为 $1/4$, 第 3 个为 $1/8$, 以此类推。设 p_0 为第 1 个空时槽的位置, 则 tag 总数为 $1.2897 \times 2P_0$ 。因为收集信息的时间极短, 移动的 tag 极少, 那么未知 tag 的数量也极少, 所以将帧长 fa 设为 10 足够计算出总体 tag 的数量, 将计算后得到的值计为 n_{un} 。

获得 n_{un} 后, reader 向所有 tag 发送帧长 f (等于剩余的 tag 的数量) 和随机数 r , 在第一个帧中 f 等于 n_{un} , 未睡眠的 tag 根据这 2 个信息和自己的 ID, 通过散列函数 $h(ID \oplus r) \% f$ 计算自己当前帧中回复信息的时槽位置 sn , 将其存储在计数器中。当仅有一个 tag 散列到时槽 i 时, 将 i 称为单时槽; 当没有 tag 散列到时槽 i 中, 将 i 称为空时槽, 除了单时槽和空时槽以外其他的时槽统称为冲突时槽。若 j 个 tag 同时散列到时槽 i 时, 将 i 称为 j 冲突时槽。当一个时槽结束后, reader 向 tag 发送一个时槽结束信息 0。tag 收到这个信息后, 将 sn 减 1 并判断 sn 是否为 0。如果 sn 为 0, 则 tag 向 reader 发送一个 6 bit^[24] 长的随机数, 而无需回复每个 tag 的 64 bit 长的 ID, 节省了时间。

当 reader 等待 t 时间并未有 tag 进行信息回复时, 说明这个时槽为空, 则 reader 发送信息 0 来结束该时槽; 当一个时槽是单时槽时, reader 发送信息 1 通知散列到这个时槽的 tag 回复自己的 ID, 回复 ID 后 reader 广播信息 0 来结束该时槽; 当冲突时槽发生时, reader 发送信息 2 通知散列到该时槽的 tag 进入睡眠状态, 等下一个帧开始时再醒来, 没有散列到该时槽的 tag 将 sn 减 1。下一个帧也执行相同操作, 重复上述过程, 直到没有冲突时槽的帧为止。有些 tag 在本次信息收集时 flag 为 1, 在下次信息收集时 flag 为 0, 为了能让这些 tag 下次信息收集时按照 sn 中存储的值主动进行信息回复, 需要发送数组 vc 来安排它们回复信息的时间。

由于本次信息收集时 flag 为 0 的 tag 在下次信息收集回复信息的时槽位置已经被安排好, 那么 flag 为 1 的 tag 在下次信息收集时需要等所有 flag

为 0 的 tag 回复信息后再进行回复, 为了保证 tag 回复信息的顺序, reader 将 flag 为 0 的 tag 回复信息的总时间 Rsn 发送所有的 flag 为 1 的 tag, 并且利用 ATC 协议发送 vc 来分配这些 flag 为 1 的 tag 相互间回复信息的顺序。以 reader1 为例, 设 tag1 是本次信息收集时 flag 为 1 的 tag 之一, 并且在 vc 中被分配的时槽位置是 5, 而 flag 为 0 的 tag 回复信息的时槽总数 Rsn 是 100, 则在下次信息收集时 tag1 在第 105 个时槽位置进行信息回复。发送 Rsn 时间很短可忽略不计, 由 ATC 协议可知, 发送 vc 的时间为

$$\frac{n_{un} t_{id}}{96} \left(\frac{1}{1-\varepsilon} \right) \quad (2)$$

其中, $\varepsilon = 1 - e^{-1}$ 。FICP 协议如算法 1 所示。

算法 1 FICP 算法

输入: n_{un}

输出: flag 为 1 的 tag 的 ID

- 1) $flags \leftarrow 0$;
- 2) While $flags \neq 1$
- 3) $flags \leftarrow 0$, $f \leftarrow n_{un}$, reader 广播 f 、随机数 r ;
- 4) 每个 tag 计算时槽位置 $sn \leftarrow h_f(ID, r)$
- 5) for slot $i \leftarrow 0$ to $f-1$ do
- 6) If 某些 tag 的 sn 为 0 then
- 7) 这些 tag 发送 6 bit 长的随机数;
- 8) reader 通过随机数获得该时槽的状态;
- 9) If 该时槽为单时槽 then
- 10) reader 广播信息 1, 散列到该时槽的 tag 回复 ID 并进入睡眠状态, reader 广播信息 0, 未睡眠的 tag 将 sn 减 1, $n_{un}--$;
- 11) else
- 12) reader 广播信息 2, 散列到该时槽的 tag 进入睡眠状态, 未睡眠的 tag 将 sn 减 1, $flags \leftarrow 1$;
- 13) end if
- 14) else
- 15) if 经过 t_{est} 时间没有 tag 回复信息 then
- 16) reader 广播信息 0, 未睡眠的 tag 将 sn 减 1;
- 17) end if
- 18) end if
- 19) end for
- 20) end while

FICP 不仅解决了传统的 ALOHA 方法中空时槽的等待时间, 同时也弥补了冲突时槽回复 tag ID 却无法解析每个 ID 的时间消耗。由 FICP 协议可知, 当剩余 tag 数是 n' 时, k 个 tag 散列到某个时槽(sn' 值相同)的概率为

$$C_n^k \times \frac{1}{f} \times \left(1 - \frac{1}{f}\right)^{n-k} \quad (3)$$

当 $k=1$ 时, 即单时槽的概率为

$$q_1 = C_n^1 \times \frac{1}{f} \times \left(1 - \frac{1}{f}\right)^{n-1} \approx \frac{n'}{f} \times e^{-\frac{n'}{f}}$$

$$\frac{\partial q_1}{\partial f} = -\frac{n'}{f^2} e^{-\frac{n'}{f}} + \frac{n'^2}{f^3} e^{-\frac{n'}{f}}$$

当 $n'=f$ 时, $\frac{\partial q_1}{\partial f} = 0$, 因此单时槽概率的最大

值为 36.8%。

由式(2)可知, 空时槽($k=0$)的概率为

$$q_0 = C_n^0 \times \left(1 - \frac{1}{f}\right)^n \approx e^{-\frac{n'}{f}}$$

当 $n'=f$ 空时, 槽的概率为 36.8%。

综上所述, 冲突时槽的概率为

$$q_c = 1 - 36.8\% - 36.8\% = 26.4\%$$

则 FICP 协议需要的总帧长为 $n_{un}/36.8\% = e \cdot n_{un}$, 其中, 单时槽和空时槽的总帧长为 n_{un} , 而冲突时槽的总帧长为 $26.4\% \cdot e \cdot n_{un}$ 。

FICP 协议的信息收集阶段时间机制如图(2)所示, FICP 的总时间 T 为信息收集时间加发送 vc 时间, flag 为 1 的 tag 的总数为 n_{un} , 则

$$T = (t_{est} + \tau + t_{min})n_{un} + (8\tau + t_{id} + t_{inf} + 4t_{min}) \cdot$$

$$n_{un} + (8\tau + 2t_{min})26.4\%en_{un} + \frac{n_{un}t_{id}}{96} \left(\frac{1}{1-\varepsilon}\right) \quad (4)$$

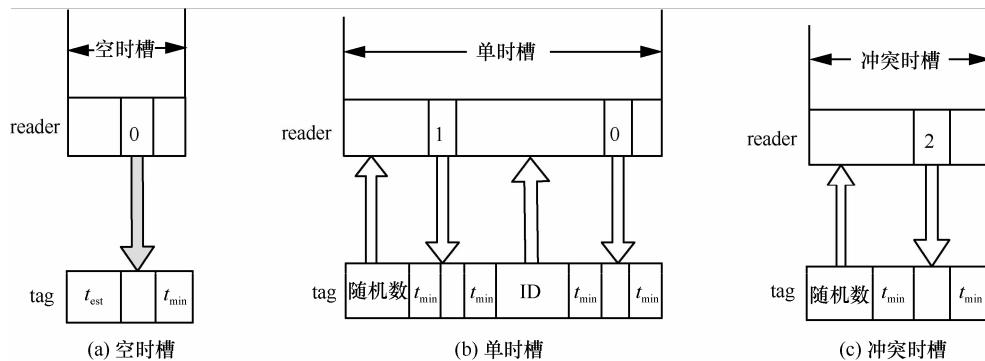


图 2 FICP 时间机制

由于 reader 冲突, 移动到多个 reader 相交区域内的 tag 无法被检测到。以 reader1 为例, 由系统模型部分分析可知, 一个 tag 在极短时间内从 reader1 的区域范围内移动到其邻居 reader 区域范围外的概率很小忽略不计, 那么这些 tag 只可能在 reader1 与其邻居的相交区域范围内或者 reader1 的邻居 reader 的相交区域内, 为了收集到这些 tag 的信息, 首先需要确定是哪些 tag 在多个 reader 的相交区域范围内。根据 ATC 协议可确定没有移动出 reader1 区域范围内的 tag, 由 FICP 协议可获得从其他区域范围内移动过来的 tagID, 而中央处理器的数据库中又存有 reader1 之前区域范围内所有 tag 的 ID, 假设中央处理器有极强的处理能力, 所有 reader 与中央处理器通信后, reader1 就会确定之前在自己区域范围内而移动到多个 reader 相交范围内的 tag, 将这些 tag 加入到集合 M_1 中, 如何与中央处理器进行通信不在本文的研究范围内。想要收集这些 tag 的信息, 仅仅获得这些 tag 的 ID 是不够的, 为了确定它们的位置以及收集信息, 本文提出了联合检测相交区域内 tag 算法(AJDT, algorithm to joint detect tags which in the intersection area)。

算法 2 AJDT 算法

输入: idi、集合 X

输出: idi 是否找到

- 1) while $X! = \emptyset$ then
- 2) 找出 X 中相交区域个数最多的 reader, 如果找出多于一个, 任选其一加入 S , 将与 S 没有相交区域的 reader 加入 S , $X \leftarrow X \setminus S$;
- 3) S 中所有的 reader 广播丢失 tag 的 id ;
- 4) 该 tag 收到自己 id 后回复传感器节点信息;
- 5) If S 中任意一个 $reader_i$ 收到回复 then
- 6) $reader_i$ 发送信息 1 给 $reader_1$, 发送数

字使该 tag 在下次信息收集时听到该数字回复信息, *break*;

7) end if

8) end while

9) 最后一个广播信息的 reader 发送 0 给 reader1。

5.2 检测相交区域 tag

在 AJDT 中, 同样以 reader1 为例, 假设 tag4 是集合 M_1 中的一个, 为了收集到它的信息, 将 reader1 与其邻居 reader 加入集合 X 中, 所有 X 中的 reader 相互通信, 找到与其他 reader 相交区域的个数最多的 reader 加入集合 S 中, 如果找到多于一个 reader, 任选其一加入集合 S , 找到与 S 中的 reader 没有相交区域的所有 reader 也加入集合 S 中, 集合 S 中的所有 reader 广播 id_4 , 收到回复信息的 reader(reader_{*i*})向 reader1 发送信息 1 告诉它已找到 tag4, 并且发送数字给该 tag 使其在信息收集时向 reader_{*i*} 回复信息, 由于 3 之前的数字已经被使用, 则发送 3 或者 3 以上的数字给该 tag, 下次信息收集时, 收到该数字后 tag 便向 reader_{*i*} 回复信息; 如果集合 S 中的所有 reader 都没有收到回复信息, 就从集合 X 中除去集合 S 中所有的 reader, 重新执行上述算法, 直到集合 X 中没有 reader。如果最终还是没有找到 tag4, 由集合 S 中最后一个广播 id_4 信息的 reader 向 reader1 发送信息 0 告诉其未找到 tag4。

如果最终没有找到 tag4 的位置, 则将其加入 *Miss*, MRCIC 执行三遍后如果还是没有找到 *Miss* 中 tag, 说明 *Miss* 中的 tag 发生损坏或者被人偷走, 则由 reader1 进行报警提示 *Miss* 丢失。每个 reader 都执行 MRCIC 协议, 现由 reader1 来描述该协议, 协议算法如算法 3。

算法 3 MRCIC 协议

输入: reader1 所在区域内先前收集所有 tag 的 ID

输出: reader1 所在区域内当前时刻所有 tag 的 ID, n_{mis} , *Miss*

1) reader1 广播自己的编号;

2) 收到信息的 tag 将存储的编号与收到的相比较

3) If 相同 then

4) 将标志寄存器 flag 置 1

5) end if

6) reader1 利用 ATC 收集 flag 为 0 的 tag

信息, 并确定 tag 寄存器 sn 的值

7) reader1 利用 FICP 收集 flag 为 1 的 tag 信息, 并发送 vc 分配这些 tag 在下次协议执行时回复信息的时槽位置

8) reader1 与中央处理器通信, 确定从自己区域范围内移动到多个 reader 相交区域内 tag 的 ID, 将它们加入集合 M

9) while $M \neq \emptyset$ then

10) reader1 广播 M 中任意一个 tag 的 id_i 给它邻居 reader

11) 将 reader1 与其邻居加入 X , 利用 AJDT 算法寻找 id_i 的位置以及收集 id_i 的信息

12) If reader1 收到信息 1 then

13) $M \leftarrow M \setminus id_i$

14) else

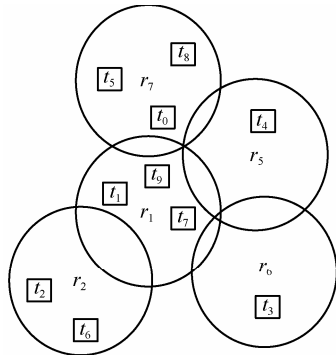
15) $M \leftarrow M \cup id_i, Miss \leftarrow id_i$

16) end if

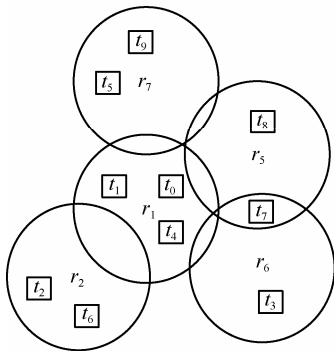
17) end while

举例说明 MRCIC 协议, 图 3 是仓库的一角, reader1 首先广播自己的编号 1, tag 收到编号后与存储在自己寄存器中的编号比较, 相同的话将自己的标志寄存器中 flag 置 1, flag 为 0 的 tag 按照寄存器中 sn 的值主动发送传感器节点信息。某一个时槽 t_{est} 时间内没有 tag 回复信息, 说明这个时槽的 tag 发生移动, reader1 检测到 tag7、tag9 发生了移动。Reader1 利用 ASAP 检测 flag 为 1 的 tag 总体数量 n_{un} , 发现从其他 reader 区域范围内移动过来的 tag 数量 $n_{\text{un}}=2$, 之后利用 FICP 来收集到 tag0 和 tag4 的 ID 以及它们的信息。信息收集后, reader1 与中央处理器进行通信, 发现 tag7 移动到了多个 reader 相交区域范围内, 利用 AJDT 方法收集 tag7 的信息, Reader1 广播 id_7 给其邻居, 由于 reader 间通信信道与 reader tag 间通信信道并不相同, 所有 tag 无法听到 reader1 广播的信息, reader1 将自己和它的邻居加入集合 X 中。5 个 reader 相互通信, 发现 reader1 是与其他 reader 相交区域最多的 reader, 将其加入 S , 其他的 reader 与 reader1 都有相交区域, 则 S 中只有 reader1。reader1 广播 id_7 , 没有 reader 收到回复, 将 S 中 reader 从 X 中除去, S 置空, 剩余的 4 个 reader 相互通信, 发现 reader5 是与其他 reader 相交区域最多的 reader, 将其加入 S , reader2 与 reader5 没有与相交区, 将 reader2 加入 S , S 中的 reader 广播 id_7 , reader5 收到回复, 也就是说 tag7 在 reader5 的区域范围内, reader5

将找到 tag7 的消息发送给 reader1，并且发送信息 3 告诉 tag7 在信息收集阶段时收到信息 3 后向其回复信息。



(a)前次检测时tag的位置



(b)当前检测时tag的位置

图 3 MRCIC 算法举例

5.3 协议执行时间

MRCIC 协议的期望时间是收集 flag 为 0、1 的 tag 信息时间与收集在 reader 相交区域范围内的 tag 信息时间总和。以 reader1 举例，设 reader1 有 n_b 个邻居，从 reader1 的区域范围内移动到多个 reader 相交区域范围内 tag 的总数为 n_{in} ，reader1 执行 AJDT 算法的最长时间是最后一个 M 中的 reader 广播丢失 tagID 信息后 tag 回复信息，而其他时间很短可忽略不计，则 AJDT 协议执行的最长时间为

$$n_{in}[t_{inf}+n_b t_{id}]$$

最短时间为

$$n_{in}[t_{inf}+t_{id}]$$

期望时间为

$$n_{in}[t_{inf}+t_{id}(n_b+1)/2] \tag{5}$$

由式(1)、式(4)和式(5)可知，MRCIC 的期望时间是

$$T_{sum} = (n - n_{mis})t_{inf} + n_{mis}(t_{est} + t_{min} + \tau) + (t_{est} + \tau + t_{min})n_{un} + (8\tau + t_{id} + t_{inf} + 4t_{min})n_{un} + (8\tau + 2t_{min})26.4\%en_{un} + \frac{n_{un}t_{id}}{96} \left(\frac{1}{1-\epsilon} \right) + n_{in} \cdot \left[t_{inf} + t_{id} \left(\frac{n_b+1}{2} \right) \right] \tag{6}$$

6 模拟实验

模拟环境设置如下：若干个节点随机地部署在 200 m×200 m 网络中。本文通过 100 组实验取其平均值来验证 MRCIC 的时间性能。

根据 ATC 协议可知，等待时槽 t_{min} 为 302 us， t_{est} 等于 321 us，reader 向 tag 发送数据速率是 26.5 kbit/s，则传送 1 bit 需要 37.76us，tag 向 reader 回复信息的速率是 53kbit/s，则传输 1 bit 数据需要 18.88 us，即 τ 等于 18.88 us。如果 tag 需要向 reader 发送 1 bit 信息，那么 t_{inf} 等于 321 us(包括等待时槽、不包含 CRC)。因为 t_{id} 是发送 64 bit 的 ID 和 32 bit 的 CRC 时间之和，所以 reader 向 tag 发送 96 bit 的信息所需要的时间 t_{id} 等于 3 927us(包括等待时槽)，则 t_{sum} 等于 $t_{id}+t_{inf}-t_{min}$ 。

为了验证 MRCIC 的时间性能，将 MRCIC 与 Li Wen Xiu 提出的 ADMCU+ATC 和 ALOHA、Polling、BT 协议进行对比，其中，ADMCU 是检查丢失 tag 以及未知 tag 的协议，而 ATC 是收集信息的协议，在多 reader 环境下先执行 ADMCU 确定区域范围内的 tag 后执行 ATC 则可以有效收集所有 tag 消息，将该方法简称为 ADMCU。表 2~表 7 表示当没有 tag 处于多个 reader 相交区域范围内， m 等于 10 000 且丢失的 tag 占 tag 总量的 5%，未知的 tag 占 tag 总量的 5%时，5 个协议的不同执行时间以及执行时间的下界。其中表 2~表 4 分别给出了当 n 等于 500、1 000 和 5 000 时，5 个协议的不同执行时间以及执行时间的下界， NB 表示每个 tag 需要回复信息的数据量(bit)大小。

表 2 当 $n=500$ 时的执行时间/s

协议	$NB=1$	$NB=32$	$NB=64$	$NB=96$	$NB=128$
PIC	42.5	42.8	43.1	43.4	43.7
BT	3.1	3.9	4.8	5.7	6.5
ALOHA	2.9	3.7	4.5	5.3	6.2
ADMCU+ATC	1.1	1.4	1.7	2.1	2.4
MRCIC	0.4	0.7	1.0	1.3	1.6
Lower bound	0.2	0.5	0.8	1.1	1.4

表 3 当 $n=1\ 000$ 时的执行时间/s

协议	$NB=1$	$NB=32$	$NB=64$	$NB=96$	$NB=128$
PIC	42.5	43.1	43.7	44.3	44.9
BT	6.2	7.84	9.6	11.3	13.1
ALOHA	5.8	7.39	9.0	10.6	12.3
ADMCU+ATC	2.2	2.8	3.3	3.8	4.3
MRCIC	0.7	1.3	1.9	2.5	3.1
Lower bound	0.3	0.9	1.5	2.1	2.7

表 4 当 $n=5\ 000$ 时的执行时间/s

协议	$NB=1$	$NB=32$	$NB=64$	$NB=96$	$NB=128$
PIC	42.4	45.4	48.4	51.4	54.5
BT	30.8	39.2	47.9	56.3	65.4
ALOHA	29.0	36.9	45.2	53.4	61.6
ADMCU+ATC	10.9	13.9	16.8	19.7	22.6
MRCIC	3.7	6.6	9.6	12.7	15.7
Lower bound	1.6	4.5	7.6	10.5	13.6

由表 2~表 4 可以发现, 无论回复信息的数据量如何变化, MRCIC 的执行时间都小于其他几个协议, 并与执行时间下界非常接近。这是因为相比于前其他协议, MRCIC 用于检测区域范围内 tag 时间最少。

表 5~表 7 给出了当 NB 等于 1、32 和 96 时, 5 个协议的不同执行时间以及执行时间的下界。不难发现, 由于 n 的增加使得传输数据量增大, 协议的执行时间就会随之增加。其中, BT 和 ALOHA 执行时间较为接近, PIC 的执行时间最长, ADMCU 和 MRCIC 的执行时间均比 BT、ALOHA 和 PIC 短, 而 MRCIC 执行时间始终最短, 并且与下界时间几乎相同, 这与式(6)的结论吻合。

表 5 $NB=1$ 时的执行时间/s

协议	$n=1\ 000$	$n=3\ 000$	$n=5\ 000$	$n=7\ 000$	$n=9\ 000$
PIC	42.5	42.5	42.4	42.4	42.4
BT	6.2	18.5	30.8	43.1	55.4
ALOHA	5.8	17.4	29.0	40.6	52.2
ADMCU+ATC	1.6	6.2	10.7	15.0	18.2
MRCIC	0.5	1.6	2.7	3.7	4.8
Lower bound	0.3	1.0	1.6	2.2	2.9

表 6 $NB=32$ 时的执行时间/s

协议	$n=1\ 000$	$n=3\ 000$	$n=5\ 000$	$n=7\ 000$	$n=9\ 000$
PIC	43.1	44.2	45.4	46.6	47.7
BT	7.84	23.5	39.2	54.9	70.6
ALOHA	7.39	22.2	36.9	51.7	66.5
ADMCU+ATC	2.2	7.9	13.6	19.1	23.4
MRCIC	1.1	3.3	5.6	7.9	10.0
Lower Bound	0.9	2.7	4.5	6.2	8.2

表 7 $NB=96$ 时的执行时间/s

协议	$n=1\ 000$	$n=3\ 000$	$n=5\ 000$	$n=7\ 000$	$n=9\ 000$
PIC	44.3	47.8	51.4	55.1	58.5
BT	11.3	33.8	56.3	78.8	101.3
ALOHA	10.6	32.0	53.4	74.7	96.1
ADMCU+ATC	3.3	11.6	19.6	17.5	34.3
MRCIC	2.3	7.0	11.6	16.3	21.0
Lower bound	2.1	6.3	10.5	14.8	19.0

当有 tag 处于多个 reader 相交区域范围内, 前面 4 个协议均无法检测到这些 tag。当 m 等于 10 000, 处于相交区域范围内的 tag 为 10%, 图 4 表示当 $n=2\ 000$ 、 $NB=48$, 丢失的 tag 占 tag 总量的 5%、未知的 tag 占 tag 总量的 10% 时和 $n=4\ 000$ 、 $NB=64$, 丢失的 tag 占 tag 总量的 10%、未知的 tag 占 tag 总量的 5% 时, 协议的不同执行时间。

由图 4 可以看出, 随着准确率提高, 由于帧长的增加, 协议的执行时间增大, 而 MRCIC 的执行时间最短并且变化最小, ADMCU 执行时间次之, BT 和 ALOHA 的执行时间较长, PTC 的执行时间最长。由于其他 4 个协议无法检测出区域范围内 10% 的 tag, 所以准确率最高只能达到 90%, MRCIC 却可以达到 100% 的准确率。

7 结束语

本文提出了在多 reader 的环境中有效收集传感器节点的协议, 传统的基于单 reader 的信息收集协议用得多 reader 环境下时需要消耗大量时间检测各自区域范围内的 tag, 而本文提出的 MRCIC 通过前次信息收集的时槽位置来分配本次 tag 回复信息的时槽位置, 并且无需重新分配时槽来检测丢失 tag, 缩短了检测区域范围内 tag 的时间。处在多个 reader 相交区域范围内的 tag 在单 reader 信息收集协议中无法被检测到, 本文

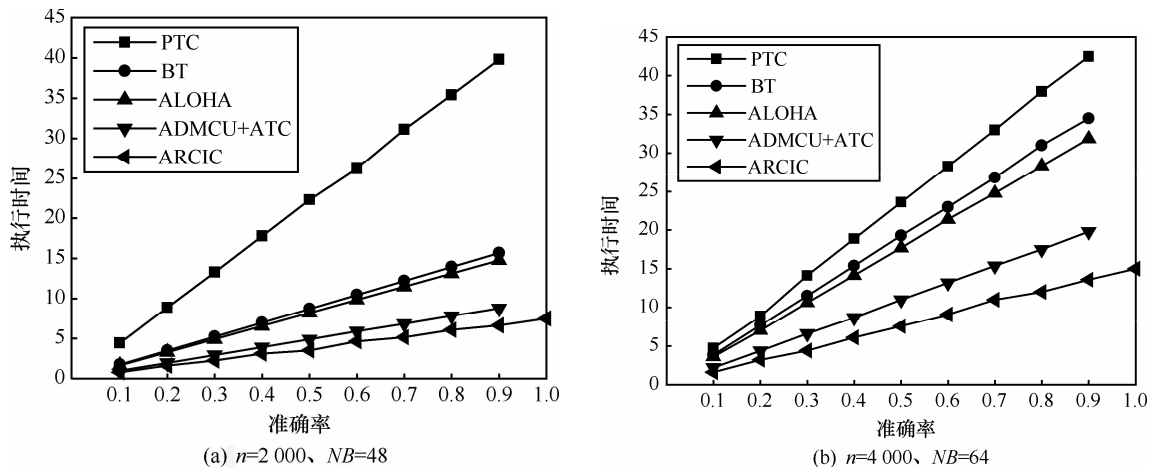


图 4 执行时间和准确率

提出的 AJDT 能够有效检测到这些 tag, 提高了协议的准确率。理论分析与实验结果表明, MRCIC 比 AMDCU、ALOHA、BT 和 PIC 延迟更短, 准确率更高。

参考文献:

- [1] PARK Y, KIM H. Tagging cities?:reading RFID tags from inside brick and mortar[A]. Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems(SENSYS 2012)[C]. NY: ACM, 2012. 331-332.
- [2] LEE E, YOO Y M, PARK C G. Installation and evaluation of rfid readers on moving vehicles[A]. Proceedings of the 6th ACM International Workshop on Vehicular Inter-NETworking(VANET 2009)[C]. NY: ACM, 2009. 99-108.
- [3] CHENG S, LI J Z, CAI Z P. $O(\epsilon)$ -approximation to physical world by sensor networks[A]. Proceedings of the 32th IEEE International Conference on Computer Communications(INFOCOM 2013)[C]. NJ: IEEE, 2013.
- [4] JI S, CAI Z. Distributed data collection and its capacity in asynchronous wireless sensor networks[A]. Proceedings of the 31th IEEE International Conference on Computer Communications(INFOCOM 2012)[C]. 2012. 2113-2121.
- [5] JI S, BEYAH R, CAI Z. Snapshot/continuous data collection capacity for large-scale probabilistic wireless sensor networks[A]. Proceedings of the 31th IEEE International Conference on Computer Communications(INFOCOM 2012)[C]. 2012. 1035-1043.
- [6] CHEN P Y, CHEN W T, WU C T, *et al.* A group tour guide system with rfids and wireless sensor networks[A]. Proceedings of the 6th IEEE International Conference on Information Processing in Sensor Networks(IPSNS 2007)[C]. 2007. 561-562.
- [7] CHEN S G, ZHANG M, XIAO B. Efficient information collection protocols for sensor-augmented rfid networks[A]. Proceedings of the 30th IEEE International Conference on Computer Communications(INFOCOM 2011)[C]. 2011. 3101-3109.
- [8] 刘文秀, 李金宝等物联网中基于 RFID-sensor 的信息收集协议[A]. 第六届中国传感器网络学术会议[C]. 2012.
- [9] LI W X, LI J B, *et al.* RFID-sensor based in formation collecting protocol in IoT[A]. The sixth Chinese sensor network Conference[C]. 2012.
- [9] QIAO Y, CHEN S, LI T. Energy-efficient polling protocols in RFID systems[A]. Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing(MOBIHOC 2011)[C]. 2011.
- [10] CHA J R, KIM J H. Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID systems[A]. Proceedings of the 3th IEEE International Conference on Consumer Communications and Networking Conference (CCNC)[C]. 2006. 768-772.
- [11] TAN C C, SHENG B, LI Q. How to monitor for missing rfid tags[A]. Proceedings of the 28th IEEE International Conference on Distributed Computing Systems (ICDCS 2008)[C]. 2008. 295-302.
- [12] XIE L, SHENG B, TAN C C. Efficient tag identification in mobile rfid systems[A]. Proceedings of the 29th IEEE International Conference on Computer Communications(INFOCOM 2010)[C]. 2010. 1-9.
- [13] MYUNG J, LEE W. Adaptive splitting protocols for RFID tag collision arbitration[A]. Proceedings of the 07th ACM International Symposium on Mobile Ad Hoc Networking and Computing(MOBIHOC 2006)[C]. 2006. 202-213.
- [14] PAN L, WU H Y. Smart trend-traversal: a low delay and energy tag arbitration protocol for large RFID systems[A]. Proceedings of the 28th IEEE International Conference on Computer Communications(INFOCOM 2009)[C]. 2009. 2571-2575.
- [15] RYU J, LEE H, SEOK Y. A hybrid query tree protocol for tag collision arbitration in RFID systems[A]. Proceedings of the 07th IEEE International Conference on The International Chamber of Commerce(ICC2007)[C]. 2007. 5981-5986.
- [16] CHEN Q, LIU Y H, NGAN H, *et al.* Asap: scalable identification and counting for contactless rfid systems[A]. Proceedings of the 30th IEEE International Conference on Distributed Computing Systems(ICDCS 2010)[C]. 2010. 52-61.
- [17] HAN H, SHENG B, TAN C. Counting RFID tags Efficiently and Anonymously[A]. Proceedings of the 29th IEEE International Conference on Computer Communications(INFOCOM 2010)[C]. 2010. 1-9.
- [18] ZHENG Y Q, LI M, CHEN Q. PET: probabilistic estimating tree for large-scale RFID estimation[A]. Proceedings of the 31th IEEE Inter-

national Conference on Distributed Computing Systems (ICDCS 2011)[C]. 2011. 37-46.

- [19] MISRA S, VERMA M, HUANG D. A secure and efficient anonymity scheme for low-cost RFID tags[A]. Proceedings of the 09th IEEE International Conference on The International Chamber of Commerce(ICC 2009)[C]. 2009. 1-6.
- [20] YANG L, HAN J, QI Y, *et al.* Identification-free batch authentication for RFID tags[A]. Proceedings of the 18th IEEE International Conference on Networks Protocols (ICNP 2010)[C]. 2010: 1092-1648.
- [21] KODIALAMM, NANDAGOPAL T. Fast and reliable estimation schemes in RFID systems[A]. The 12th Annual International Conference on Mobile Computing and Networking 2006(MOBICOM 2006)[C]. 2006. 322-333.
- [22] SHENG B, LI Q, MAO W Z. Efficient continuous scanning in rfid systems[A]. Proceedings of the 29th IEEE International Conference on Computer Communications(INFOCOM 2010)[C]. 2010. 1-9.
- [23] HUSHU. R., WOODC. Analysis of tree algorithms for RFID arbitration[A]. Proceedings of IEEE Symposium on Information Theory[C]. 1998.
- [24] XIAO Q J, BU K, XIAO B. Efficient monitoring of dynamic tag populations in RFID systems[A]. Proceedings of 9th IEEE International Conference on Embedded and Ubiquitous Computing (EUC 2011)[C]. 2011. 106-113.

作者简介:



李文秀 (1987-), 女, 黑龙江牡丹江人, 黑龙江大学硕士生, 主要研究方向为物联网。



郭亚红[通信作者] (1972-), 女, 黑龙江双鸭山人, 黑龙江大学讲师, 主要研究方向为无线传感器网络研究。E-mail: Guoyahong7225@163.com。



李金宝 (1969-), 男, 黑龙江庆安人, 黑龙江大学教授, 主要研究方向为无线传感器网络、数据库原理、移动计算和并行计算。



郭龙江 (1973-), 男, 河北丰宁人, 黑龙江大学教授, 主要研究方向为无线传感器网络和数据挖掘。



张绍斌 (1970-), 男, 安徽宿州人, 中国移动通信集团黑龙江有限公司工程师, 主要研究方向为通信管理。

ISSN 1000-436X



9 771000 436137

发行代号: 国内2-676
国外M395

(2013)京新出报刊增准字第(310)号

2013年8月25日出版 定价: 58.00元