

EERSFS: 高能效的混杂 RFID 数据冗余过滤方法^①

罗庆平^② 王永利 张功萱

(南京理工大学计算机科学与工程学院 南京 210094)

摘 要 为克服无线传感技术和射频识别(RFID)技术在物联网中应用上的缺陷,针对由无线传感和 RFID 结合所形成的混杂 RFID 网络,引入簇的概念,提出了一种基于簇结构的高能效的射频识别和传感数据过滤模型(EERSFS),此模型通过冗余的分层定义,使冗余数据在冗余源附近得到过滤处理。实验证明,该过滤方法能有效降低网络中的冗余数据量和网络节点的能量消耗,延长无线传感网络的寿命。

关键字 无线传感,混杂 RFID,冗余过滤

0 引 言

随着物联网(Internet of things)^[1]技术的发展,无线传感网络(WSN)和无线射频识别(radio frequency identification, RFID)作为联接物理世界和虚拟世界的典型感知技术,在智能交通、智能电网、远程监控、医疗、仓库管理中均得到了广泛应用。WSN 是由大量可进行感知处理的传感器节点相互通信所组成的网络,其无线节点将感知到的环境参数(如温度、湿度等)传递给网关接入点,由网关将收集到的信息通过网络层提交到后台进行处理。RFID 是一种利用射频信息自动识别目标对象并获取相关信息的技术,RFID 系统由电子标签、阅读器和天线组成。WSN 有感知功能,但不能有效地识别移动目标的运动轨迹。RFID 对节点标识有着得天独厚的优势,但不便于野外环境的现场部署。为了相互弥补对方的缺陷,研究人员整合了无线传感节点(WSN node)和 RFID 阅读器,创造了混杂 RFID 节点,即 WRFID。WRFID 节点主要由 RFID 射频模块、传感器模块、中央控制器和天线组成,其中 RFID 射频模块负责读取环境中的标签信息,传感器模块采集环境参数,中央控制器将读取到的标签数据和传感数据处理后,通过天线进行无线传输。由该类节点所组成的混杂 RFID 网络^[2-4]既能实现对周围环境参

数的检测,又可感知移动对象的运动轨迹,同时,还能利用无线传感网络动态搭建数据传输链路。由于网络节点的高密度部署和阅读器高频率侦听的特性,导致大量冗余数据的产生,网络负载随之增加。Carbunar 等人提出在服务器端进行冗余数据过滤^[5],但通过传感器网络传送这些冗余数据到基站进行处理,会大大消耗节点的能量,降低网络的生命周期。因此,如何在 WRFID 网内对混杂 RFID 冗余数据进行处理,是本课题的研究重点。

本文基于簇^[6]的概念,研究了低能耗过滤标签移动网络冗余数据的方法,提出了适合混杂 RFID 网络结构特性的高能效的射频识别和传感数据过滤模型(energy efficient RFID and sensor data filtering scheme, EERSFS)和标签移动网络的基于簇的冗余数据过滤算法。

1 相关工作

1.1 RFID 数据清洗

数据冗余一直是物联网环境下 RFID 应用中的一个重要问题。许多学者针对数据漏读、多读及数据冗余等情况进行了深入研究。Tran^[7]等利用概率推理的方法提出了一种推断移动环境下标签位置的方法,但该技术需要事先知道标签和阅读器的运动规律等先验知识,因此不能应用于没有规律可遵循

① 国家自然科学基金(61170035),江苏省自然科学基金重大专项(BK2011022),中国博士后科学基金(200902517),江苏省自然科学基金(BK2011702),中央高校基本科研业务费专项资金(30920130112006)和南京市科技计划(020142010)资助项目。

② 女,1988年生,硕士生;研究方向:物联网,数据融合,数据挖掘等;联系人,E-mail:catherineluo1988@gmail.com
(收稿日期:2013-05-06)

的物联网环境。Chen 等^[8]基于贝叶斯推论提出一种利用空间冗余的 RFID 原始数据清理方法,但这种方法同样需要大量先验知识。Jeffery 等提出了基于概率模型的自适应数据清洗方法^[9],利用滑动窗口的统计信息处理丢失的 RFID 读数,依据概率来自适应调整窗口大小,有效地保证了数据的准确性。文献[9,10]都提出了基于滑动窗口的清洗策略,然而,窗口大小的设定也是急需解决的问题。目前,由于受到阅读器电源、计算单元、内存等资源的限制,以上方法只能在基站得以实现,但冗余数据的传输又会增加网络开销。因此,如何进行网内冗余数据的处理是需要研究的重要内容。

1.2 数据融合

数据融合^[11,12]是无线传感网络通过减少原始数据中的数据冗余以减少通信量、延长网络生命周期的关键技术。数据级融合是直接采集到的原始数据层上,利用信息之间的相关性,对多次测量的数据进行融合处理。加权平均是最简单、直观的融合方法,它将传感器组提供的信息进行加权平均,并将其结果作为融合值。权的分配对融合效果的影响十分明显,如何分配融合时的权值是一个重要的问题。卡尔曼滤波^[13]作为数据级的实时融合方法,利用测量模型的统计特性递推决定统计意义下的最优融合数据估计。但该方法对于模型的创建具有较高的要求。若创建的模型与真实模型不相符,则会出现滤波发散现象。贝叶斯估计^[14]是融合静态环境中多传感器底层信息的常用方法,它将传感器信息按照概率原则进行组合,并以条件概率表示测量的不确定性。贝叶斯方法的理论基础基本趋于成熟,但由于其保证系统相关性和一致性的特性,当系统规则发生变更时,需要重新计算所有概率,运算量也随之增加。对于节点失效率比较高的实时处理来说,并不太适合。

本文采用接近度矩阵对测量数据进行一致性检验,并利用文献[15]提出的平均值迭代方法对检验结果进行有效融合,这不仅有效地消除了噪音数据对冗余结果的影响,也在循环迭代的过程中降低了融合值与真实值之间的误差。此外,较低的融合复杂度也大大降低了融合处理的时间,对于数据的网内实时融合具有重要作用。

在混杂 RFID 环境中,网络节点在读取标签信息的同时还需要测量环境参数,因此,除了解决 RFID 数据冗余外,还需要对传感数据冗余进行处理。根据混杂 RFID 网络的以上特征,本文提出基

于簇的结构,在网内实现冗余数据的融合过滤。

2 模型与定义

在问题描述前,先进行概念的定义。

定义 1 混杂 RFID 网络。是由大量 WRFID 网络节点组成的网络。与 WSN 类似,借助自组织网络协议,通过网络自组织,实现数据从源头节点到 sink 节点的传输。不同的是,由于 WRFID 节点特殊的结构和功能,混杂 RFID 网络除了传送传感信息外,还能获取环境中的射频信息,并将环境参数与射频信号进行有效整合后统一传输。

定义 2 簇。是混杂 RFID 网络中节点的子集。每个簇由一个簇头节点和若干成员节点组成。

定义 3 簇头。也是一个 WRFID 节点。簇中的其他成员将采集到的数据传输给簇头,再由簇头进行统一处理,并将结果传给 sink 节点。簇头是网络中信息传递和信息处理的原子节点。

定义 4 成员节点。是簇中除簇头之外的其他节点。它负责环境信息的采集,将数据传输给簇头节点。成员节点与簇头节点进行单跳通信。成员节点是簇中信息采集和信息传递的原子节点。

定义 5 区域 $D \in$ 簇 C 。当区域 D 在簇 C 的阅读范围内时,称区域 D 属于簇 C 。

定义 6 重叠覆盖。当存在区域 $D \in$ 簇 C 且区域 $D \in$ 簇 E 时,称簇 C 与簇 E 重叠覆盖。

定义 7 簇内节点。节点 N 是簇 C 的成员节点,区域 D 是 N 的阅读区域,当区域 D 只 $\in$ 簇 C 时,称节点 N 为簇 C 的簇内节点。如图 1, A_1 是簇 A 的簇内节点, B_1 是簇 B 的簇内节点。

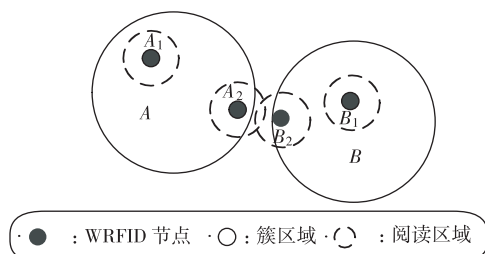


图 1 节点定义

定义 8 簇间节点。设有两个不同簇 C_0 和 C_1 , 节点 N 是簇 C_0 的成员节点,区域 D 是 N 的阅读区域,存在区域 $d \in D$,使得 $d \in$ 簇 C_0 且 $d \in C_1$,称节点 N 是簇 C_0 的簇间节点。如图 1, A_2 是簇 A 的簇间节

点, B_2 是簇 B 的簇间节点。

2.1 冗余定义

在混杂 RFID 网络中, 为了提高系统的精确度, 在环境中部署了大量的网络节点。由于节点部署的密度较高, 使得节点可感知的范围存在重叠区域。当目标在重叠区域移动时, 其 RFID 标识被多个网络节点获取。

定义 9 本地数据冗余。 在 WRFID 数据流 s 中, 数据 $x \in s$ 是冗余的, 当且仅当存在数据 $y \in s$, 使得 $y.tagId = x.tagId$, $y.readerId = x.readerId$, 且 $|x.timeStamp - y.timeStamp| < t$, 其中 t 大于零, 是应用相关参数。

定义 10 簇内数据冗余。 在 WRFID 数据流 s 中, 数据 $x \in s$ 是冗余的, 当且仅当存在数据 $y \in s$, 使 $y.tagId = x.tagId$, $y.readerId \neq x.readerId$, $y.clusterHead = x.clusterHead$, 且 $|x.timeStamp - y.timeStamp| < t$, 其中 t 大于零, 是应用相关参数。

定义 11 簇间数据冗余。 在 WRFID 数据流 s 中, 数据 $x \in s$ 是冗余的, 当且仅当存在数据 $y \in s$, 使得 $y.tagId = x.tagId$, $y.readerId \neq x.readerId$, $y.clusterHead \neq x.clusterHead$, 且 $|x.timeStamp - y.timeStamp| < t$, 其中 t 大于零, 是应用相关参数。

2.2 问题描述

在混杂 RFID 网络中, 冗余信息的传输不但消耗了大量的 WSN 带宽, 还大大消耗了传感器节点的电能, 降低了网络的寿命。CLIF^[16] 是一种基于簇结构的网内去冗方法, 当冗余信息来自于同一簇的不同节点时, 在该簇的簇头节点进行冗余过滤。当冗余信息来自于不同簇的节点时, 通过中间簇头节点进行冗余过滤。过滤的结果通过多跳路由传输到 sink 节点, 如图 2 所示。

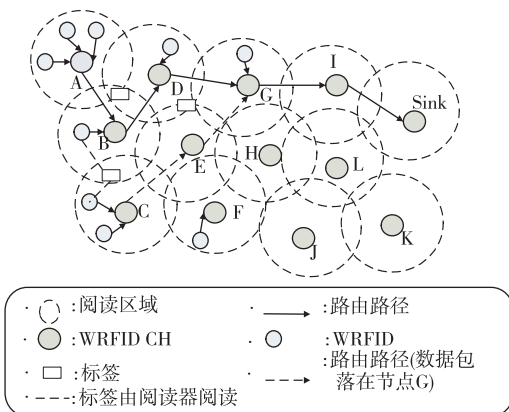


图 2 基于簇结构的冗余过滤网络

从图 2 可以看出, 当簇头 B 和簇头 C 发生冗余时, 并没有及时进行冗余处理, B 和 C 节点仍按照各自的路由进行信息传输, 直到 B 和 C 的信息在 G 点相遇时, 冗余问题才得以解决。尽管 CLIF 能够在每个簇的簇头节点对该簇内的冗余信息进行相应的处理, 但不能有效地处理簇与簇之间的冗余。为了解决这一问题, 文献[6]提出了一种更加有效的方法——EIFS, 它采用与 CLIF 类似的拓扑结构, 当簇间冗余发生时, 改变簇头的路由路径, 使得簇间冗余问题在冗余源附近得到处理。例如, 当图中的簇头 B 和簇头 C 发生冗余时, 簇头 C 将改变原有的路由, 将数据发送给簇头 B , 再由簇头 B 进行信息过滤。尽管如此, EIFS 方法的实现却只仅适用于标签固定的环境中, 对于移动标签的情况, 并不能得到有效的解决。此外, 上述方法仅仅解决了标签冗余的问题。在本文提到的混杂 RFID 环境中, 由于需要同时提取目标所在的环境参数, 还存在传感器数据冗余的情况。为了解决以上问题, 本文提出了 EERSFS, 基于簇实现混杂 RFID 数据的有效去冗。

2.3 系统模型与假设

首先, 混杂 RFID 网络由 N 个 WRFID 节点组成, 每个 WRFID 节点都具有阅读器模块和环境感知模块, 节点不能移动。网络中所有节点的环境感知模块属于同构传感器。该网络被划分为许多簇, 每个簇包含一个簇头节点和若干个成员节点。每个 WRFID 节点只能且必须属于一个簇。信息在网络中依靠簇头传递。只有簇头节点执行冗余检测和过滤算法。然后, 假设: (1) WRFID 节点传输距离大约为 2 倍的 RFID 阅读距离; (2) 网络中标签缓慢移动。

2.4 数据格式

表 1 描述的是混杂 RFID 网络中的数据包传输格式。其中, $readerId$ 用于存储最早读取到 RFID 标识的 WRFID 编号, 当 WRFID 节点阅读器与地理地图信息进行绑定时, 不仅可以识别标签信息, 还能通过阅读器编号进行目标定位。 $localId$ 指传输节点的编号。 $tagId$ 是环境中移动标签的编号, 即目标的绑定信息。 $sensorData$ 是指节点中传感模块所获取到的环境参数, 如温度、湿度、风力等。 $timeStamp$ 即节点采集到标签信息时的时间戳。 e 是指 WRFID 的剩余能量。 f 是用于记录该数据包是否已经过冗余处理, 初始值为 1, 其语义如下:

$$f = \begin{cases} 1: \text{默认值} \\ 2: \text{数据包已经过簇内冗余过滤} \\ 3: \text{数据包已经过簇间冗余过滤} \end{cases}$$

当 $f=3$ 时,表示该数据包不需要进行冗余过滤,通过无线路由最短路径传输到 sink 节点。

表 1 WRFID 数据包结构

分量	localId	readerId	tagId	sensor data	time stamp	e	f
字节数	4	4	2	2	4	4	1

混杂 RFID 网络在初始化时,节点通过信息交换获取邻居节点的相关信息 NM (neighbor message)。NM 包括两部分内容:节点 ID 和簇头 ID,如表 2 所示。成员节点检查所有收到的 NM,可判断自己是簇内节点还是簇间节点。当存在 NM 且 NM 中簇头 ID 信息与节点自身簇头 ID 不同时,该节点即为簇间节点。簇间节点维持一份 NM 列表,列表中的所有 NM 信息都来自于其他簇的簇间节点。

表 2 NM 数据结构

分量	readerId	clusterId
字节数	4	4

定义 12 相邻。具有重叠覆盖区域的簇头节点即为相邻。

同时,簇间节点将自身的编号和相邻簇的簇头信息发送给簇头节点。簇头节点保存簇间节点——邻居簇的对应信息 NC (neighbor cluster),如表 3 所示。当簇头节点要与相邻簇头通信时,查找 NC 列表,将数据发送给对应的簇间节点,再由簇间节点查询 NM 列表,并把数据传输给目标簇的簇间节点,最后,由后者将信息传递给目标簇头。

表 3 NC 数据结构

分量	readerId	Neighbor_clusterId
字节数	4	4

簇间节点接收到不同簇的簇间节点发送的数据时,需要检查一个名为 ND (no deliver) 的列表,列表内容如表 4 所示。*readerId* 表示一个簇的簇头节点,*tagId* 是标签标号,*timeStamp* 是时间戳。一个 ND 消息表示在时间 t 内 (t 大于零,是应用相关参数),不需要转发 *readerId* 和 *tagId* 与之匹配的数据。节点收到数据后与该列表进行匹配,若发现 *readerId* 和 *tagId* 已经存在于列表中,则将数据丢弃;若不存在,

则需要把数据转发给簇头节点,同时,将相关信息写入该列表,*timeStamp* 取当前时间。

表 4 ND 数据结构

分量	readerId	tagId	timeStamp
字节数	4	2	4

2.5 EERSFS

基于以上冗余的定义,我们提出了一个冗余过滤模型——EERSFS,如图 3 所示。

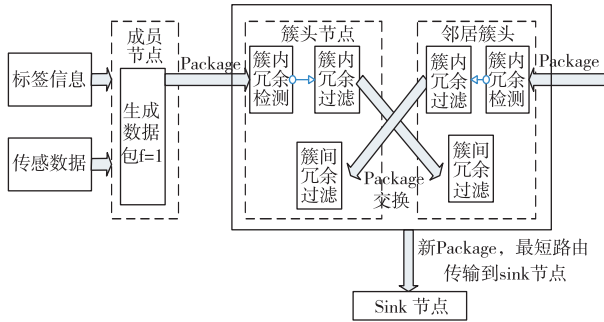


图 3 冗余过滤模型——EERSFS 模型

根据冗余的定义,冗余过滤模型 EERSFS 主要由簇内冗余过滤及簇间冗余过滤两部分组成。簇头节点收集成员节点采集的信息,并进行簇内冗余过滤,过滤的结果与相邻簇头进行交换,再处理簇间冗余过滤。

路由策略

- (1)成员节点采集到环境信息时生成数据包,令 $f=1$,将数据包传送给簇头节点。
- (2)簇头节点接收到簇内成员节点发送的数据时(即 $f=1$ 的数据包),运行冗余检测模块,检测是否存在簇内冗余,若存在,则执行簇内冗余过滤算法。令 $f=2$,并将数据包发送给所有生成冗余数据包的簇间节点。
- (3)簇间节点收到簇头发送且 $f=2$ 的数据包时,通过查询 NM 列表,把数据包发送给相邻簇的其中一个簇间节点。当簇间节点收到非簇头节点发送且 $f=2$ 的数据包时,检查 ND 列表,若发现 *readerId* 和 *tagId* 信息已经存在于列表中,则将数据包丢弃;否则,将数据包转发给本簇的簇头节点,同时,将 *readerId* 和 *tagId* 信息写入 ND 列表。
- (4)簇头节点接收时间窗口内所有 $f=2$ 的数据冗余包,每收到一个数据包向簇间节点组播一个

ND 消息,以通知簇间节点不需要再发送相关数据。当接收完所有数据后,若发现自身剩余能量最多时,执行簇间冗余算法,令 $f=3$,将数据包沿着最短路由方向传送。

(5) 节点接收到 $f=3$ 的数据包时,不进行任何冗余检测,将数据包沿着最短路由方向传送。

如图 4 所示, A 和 B 是两个重叠覆盖的簇头。 a_1 是 A 的簇间节点, b_1, b_2 是 B 的簇间节点。 a_1 同 b_1, b_2 都有重叠覆盖区域,因此 a_1 的 NM 列表中存储着 b_1, b_2 的相关信息。 t_0 时刻, a_1 和 b_1 都获取到标签 T 的射频信息,并将数据分别传输给 A 和 B 进行簇内冗余过滤。簇头 A 生成 $f=2$ 的数据包 *package_A*, 并将其发送给 a_1 , a_1 检查 NM 列表,选择 b_1, b_2 中其中一个节点进行包传输。图中 b_2 收到来自 a_1 的数据包,并将其转发给簇头 B 。依照此原理,簇头 B 也将簇内冗余过滤的结果经过 b_1 和 a_1 发送给簇头 A 。这样便完成了簇头之间的信息交换。

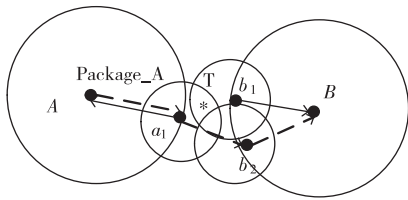


图 4 数据交换过程

在混杂 RFID 网络中,信息的传递是通过节点之间的无线通路进行的。在此,引入无向图的概念 $G(N, E)$ 表示 WRFID 网络, N 是 WRFID 节点的集合, E 是边的集合。边表示节点之间的通信关系,用 $e < N_i, N_j >$ 表示。当且仅当节点 N_i 和节点 N_j 单跳通信时, $e < N_i, N_j > \in E$ 。

定理 1: 图 $G(N, E)$ 中,任意具有重叠覆盖区域的节点之间都存在路径。

证明: 情况 1: 节点 a 和节点 b 是不同簇的簇间节点, a 和 b 存在重叠覆盖区域。如图 4 所示。 a_1 是 A 的簇间节点, b_1 是 B 的簇间节点。节点阅读半径为 r 。 a_1 和 b_1 存在重叠覆盖区域,则 a_1, b_1 之间的距离的小于等于 $2r$ 。由 2.3 中提到的 WRFID 节点传输距离大约为 2 倍的 RFID 阅读距离可知 a_1 和 b_1 可单跳通信,即 $e < a_1, b_1 > \in E$, a_1 和 b_1 之间存在路径成立。

情况 2: 节点 A 是簇头节点,节点 b 是另一簇的簇间节点, A 和 b 存在重叠覆盖区域。由于 A 和 b 存在重叠覆盖区域,则存在节点 a ,使得 a 是簇头 A

的簇间节点且 a 和 b 存在重叠覆盖区域。 a 和 A 之间单跳通信,因此, $e < a, A > \in E$ 。由情况 1 可知, $e < a, b > \in E$,故节点 A 和节点 b 之间存在路径。

情况 3: 节点 A 和节点 B 是不同簇的簇头节点, A 和 B 存在重叠覆盖区域。由于 A 和 B 存在重叠覆盖区域,故存在节点 a 和节点 b ,其中,节点 a 是簇头 A 的簇间节点,节点 b 是簇头 B 的簇间节点,使得 a 和 b 存在重叠覆盖区域。由于 $e < a, A > \in E$, $e < b, B > \in E$, $e < a, b > \in E$,因此, $e < A, B > \in E$,即 A 和 B 之间存在路径。

综上所述,任意具有重叠覆盖区域的节点之间都存在路径。

由定理 1 可知,通过信息交换,簇头节点可以获得所有冗余信息,并对冗余数据进行有效处理。

基于 EERSFS 的二次过滤机制不仅保证了数据过滤的效率,而且,其相邻簇头数据包交换策略还能够使冗余数据在靠近冗余源的节点进行过滤,不仅避免了由标签多读造成的带宽浪费,还有效地处理了由传感数据带来的冗余问题,降低了网络的能量消耗,延长了网络的寿命。

3 EERSFS 的冗余过滤方法

冗余过滤模型 EERSFS 除了要避免标签多读的情况外,还要有效解决传感数据冗余问题。设 $T_1, T_2, T_3, \dots, T_n$ 表示 n 个 WRFID 节点在时刻 t 的目标环境温度测量结果。由于传感器自身精度和环境的影响,测量结果存在一定误差。文献[15]提出了一种平均值迭代的数据融合方法。该方法循环地用平均值更新测量数据中的最大值和最小值,在迭代更新的过程中降低了偏离真实值的误差。

3.1 数据检验

在混杂 RFID 系统中,当某些 WRFID 节点突然发生故障或受到环境干扰采集了虚假信息时,如果不进行数据一致性检查而直接将所有数据进行融合,将会大大降低融合的精度。本文采用测量数据的接近度矩阵来进行数据的一致性检验。将各 WRFID 节点的所测数据视为一个模糊集合,根据模糊数学理论,模糊集合之间的接近程度可用接近度来度量。 a_i 表示节点 WRFID _{i} 所测数据, a_j 表示节点 WRFID _{j} 所测数据, a_{ij} 表示节点 WRFID _{i} 和 WRFID _{j} 所测数据的接近度。 a_{ij} 的计算方法如下:

$$a_{ij} = \begin{cases} 1 & |a_i - a_j| \leq \xi \text{ 且 } i \neq j \\ 0 & |a_i - a_j| > \xi \text{ 或 } i = j \end{cases} \quad (1)$$

其中 ξ 根据传感模块的测量精度来确定。根据(1)式,可以建立接近度矩阵:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdot & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

对于 A 中的第 i 行元素,令

$$k_i = \frac{\sum_{j=1}^n a_{ij}}{(n-1)}, \quad i \neq j \quad (2)$$

k_i 越大,说明第 i 个 WRFID 节点所测数据与多数 WRFID 测得数据越接近;反之,第 i 个 WRFID 节点所测数据偏离多数 WRFID 测得数据。混杂 RFID 网络中仅仅对满足 $k_i \geq \lambda$ 的数据进行融合处理。其中, λ 是根据系统的融合精度来确定的。

3.2 迭代融合

$T: (T_1, T_2, T_3, \cdots, T_n)$ 表示测量数据集, $\bar{T}$ 表示测量数据的融合值。以下是平均值迭代的步骤:

(1) 计算样本 T 的平均值 $\bar{T}$, 找出样本中的最大值 $T_i = \max \{T_1, T_2, T_3, \cdots, T_n\}$, 令 $T_i = \bar{T}$ 。

(2) 重新计算样本更新后的平均值 $\bar{T}$, 找出样本最小值 $T_j = \min \{T_1, T_2, T_3, \cdots, T_n\}$, 令 $T_j = \bar{T}$ 。

(3) 再次计算样本更新后的平均值 $\bar{T}$ 。求出 $\bar{T}$ 与更新后的样本 $(T_1, T_2, T_3, \cdots, T_n)$ 差的绝对值之和

$$\delta, \text{ 即 } \delta = \sum_{i=1}^n |T - T_i|。$$

(4) 若 $\delta > \varepsilon$ ($\varepsilon > 0$), 则跳到步骤(1), 否则结束, $\bar{T}$ 为最后融合结果。其中 ε 为融合精度要求。

3.3 算法描述

设滑动窗口大小为 W , 窗口起始时间为 $window_startTime$, 结束时间是 $window_endTime$ 。冗余处理的时间间隔为 Δt , 其中 $W \geq 3\Delta t$ 。系统将 $window_startTime$ 到 $window_startTime + \Delta t$ 时间内采集到的冗余数据进行过滤处理, 生成新的数据包进行传输。冗余过滤的过程, 其实就是生成新的数据包的过程。

簇头节点对时间窗口内接收到的冗余数据包, 执行如下冗余过滤算法:

Algorithm 1. DataFiltering algorithm for cluster head based on EERSFS Model

输入: 数据流

输出: 新的数据包 $newData$

算法描述:

1: do

```

2:   get the first data D
3:   if D.f = 1
4:     get data whoses f = 1, tagId = D.tagId and arrives
        between window_startTime and
        window_startTime + Δt, and insert data into items[ ].
5:     data_temp ← Data_Validation(items, ξ, λ);
6:     sensorData ← Iterative_Average_Fusion(data_temp, ε);
7:     newData ← New_Package_Generation(items, sensorData);
8:     newData.f ← 2;
9:     deliver newData to inter-cluster nodes whose readerId
        appear in items;
10:  end if
11: if D.f = 2
12:   get data whoses f = 2, tagId = D.tagId and arrives
        between window_startTime and
        window_startTime + Δt, and insert data into
        clusterItems[ ].
13:   if the energy is the most
14:     data_temp ← Data_Validation(items, ξ, λ);
15:     sensorData ← Iterative_Average_Fusion(data_temp, ε);
16:     newData ← New_Package_Generation(items, sensorData);
17:     newData.f ← 3;
18:     deliver data_result to inter-cluster nodes whose readerId appear in
        items;
19:   end if
20: end if
21: if D.f = 3
22:   deliver D to sink node with the shortest-path router
23: end if
24: while not end of window
25: end do
    
```

Procedure 1. Data_Validation($packages[], \xi, \lambda$)

输入: 冗余的数据包数组 $packages[]$

输出: 进行数据融合的测量值数组 $sensorData[]$

算法描述:

```

1: n = packages.length;
2: for i = 1, ..., n do
3:   for j = i, ..., n do
4:     if |packages[i].sensorData - packages[j].sensorData| ≤ ξ
5:       A[i][j], A[j][i] ← 1;
6:     else
7:       A[i][j], A[j][i] ← 0;
8:     end if
9:   end for
10: end for
11: for d = 1, ..., n do
12:   k, count ← 0;
13:   for m = 1, ..., n do
14:     if d! = m
    
```

```

15:          $k = k + A[d][m]$ 
16:     end if
17: end for
18:  $k = k / (n - 1)$ ;
19: if  $k \geq \lambda$ 
20:      $sensorData[count] = package.s[d].sensorData$ ;
21:      $count++$ ;
22: end if
23: end for
24: return  $sensorData$ ;
    
```

Procedure2. *Iterative_Average_Fusion*($sensorData[]$, ε)

输入: 传感数据组 $sensorData[]$

输出: 数据融合结果 $resultData$

算法描述:

```

1: do
2:      $n = sensorData.length$ ;
3:     get the maximum  $sensorData[i]$  in  $sensorData$ ;
4:      $sensorData[i] = \frac{1}{n} \sum_{i=1}^n sensorData[i]$ ;
5:     get the minimum  $sensorData[j]$  in  $sensorData$ ;
6:      $sensorData[j] = \frac{1}{n} \sum_{i=1}^n sensorData[i]$ ;
7:      $resultData = \frac{1}{n} \sum_{i=1}^n sensorData[i]$ ;
8:     for  $k = 1, \dots, n$  do
9:          $\delta = |resultData - sensorData[k]|$ ;
10:    end for
11: while  $\delta > \varepsilon$ 
12: end do
13: return  $resultData$ 
    
```

proceduer 3. *New_Package_Generation*($data[]$, $resultData$)

输入: 冗余的数据包数组 $data[]$, 传感数据融合结果 $resultData$

输出: 新的数据包 $newData$

算法描述:

```

1: package ← find the package from data whose timestamp is the minimum;
2: newData.readerId ← package.readerId;
3: newData.tagId ← package.tagId;
4: newData.timeStamp ← package.timeStamp;
5: newData.e ← remaining energy;
6: newData.sensorData ← resultData;
7: newData.localId ← id of node
8: return newData
    
```

簇间节点接收到数据包时, 执行路由算法如下:

Algorithm 2. *DataFiltering_Routing* algorithm for inter-cluster node based on EERSFS Model

输入: 数据包 $package$

输出: 下一跳传输节点 $node$

算法描述:

```

1: if the package is from cluster head and package.f = 2
2:     check NM list and choose one node in every neighbor cluster to transport the package;
3: end if
4: if the package is from node in NM list and package.f = 2
5:     if ND exists which package.readerId = ND.readerId and package.tagId = ND.tagId
6:         abandon the package
7: end if
8: else
9:     deliver the package to cluster head;
10:    create a ND message whose readerId = package.readerId and tagId = package.tagId
11:    add message to ND LIST
12: end else
13: if package.f = 3
14:     deliver data to sink node with the shortest-path router;
15: end if
    
```

3.4 算法分析

冗余过滤算法的目的是对时间窗口内接收到的冗余数据进行数据融合。设冗余数据包个数为 n , *DataFiltering* 包含三个过程: *Data_Validation*, *Iterative_Average_Fusion*, *New_Package_Generation*。

(1) *Data_Validation* 过程: 对冗余数据包组进行数据检验的过程, 首先计算出接近度矩阵 A , 由式(1)可知 A 是一个对称矩阵, 其时间复杂度为 $O(n^2)$; 然后利用接近度矩阵获得需要进行数据融合的测量数据组, 时间复杂度为 $O(n^2)$ 。因此, 该过程的时间复杂度为 $O(n^2)$ 。

(2) *Iterative_Average_Fusion* 过程: 测量数据个数为 k , 利用平均值迭代方法对测量数据进行融合, 设要使满足条件 $\delta > \varepsilon$, 需要执行 e 次融合。每次融合的时间复杂度为 $O(k)$ 。那么, 总的复杂度应该为 $O(e * k)$ 。

(3) *New_Package_Generation* 过程: 需在原始数据包组中进行最小值查找, 其时间复杂度为 $O(n)$ 。

综上所述, 算法 *DataFiltering* 的时间复杂度为 $O(n^2) + O(e * k) + O(n)$, 即 $O(n^2 + e * k)$ 。

4 仿真实验及结果分析

4.1 实验环境

本节中对 EERSFS 以及在此模型上的算法进行实验分析。仿真实验采用 NS2 作为实验平台。NS2 底层的仿真引擎主要由 C++ 编写,同时利用 Tcl 语言作为仿真命令和配置的接口语言,通过调用引擎中的属性、方法,定义网络的拓扑结构,配置节点,运行并跟踪仿真结果,以及对结果进行统计处理等。实验采用的操作系统是 Ubuntu 10.04。硬件配置是 CPU: Intel(R) Xeon(R) CPU E5620 2.40GHz; 内存: 12GB; 硬盘: 500GB; 网卡: Intel(R) PRO/1000MT。实验模拟一个 WRFID 网络以及若干标签节点。其中,标签节点在网络覆盖区域内按一定轨迹缓慢移动,网络节点除了能够获取标签射频信息,还能获取环境温度值。网络中节点均匀分布,具体的模拟环境如表 5 所示。

表 5 模拟环境

网络参数	默认值
网络节点个数	320
簇头节点个数	20
每个簇中成员节点个数	15
网络中标签个数	100
节点阅读距离	5m
节点传输距离	10m
冗余过滤时间 Δt	2s

4.2 实验分析

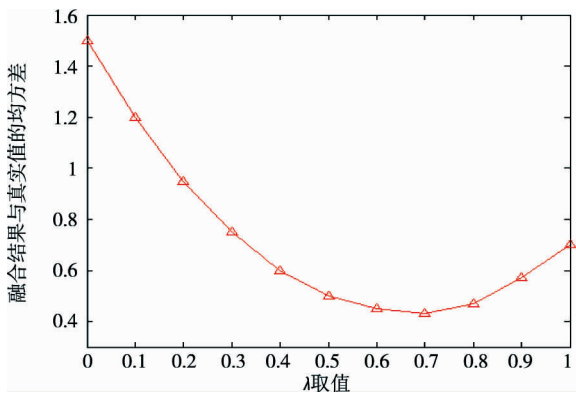
在野外环境温度测量时,测量值可能随着海拔等因素的变化而变化。节点由于部署在不同的位置,测出的温度值存在一定的差异,尤其在海拔变化明显的区域,测量值的差异也会更加明显。通过判断温度的变化范围确定 ξ 的大小,能够得到更为精确的数据一致性检验结果,为数据融合提供更加真实的融合样本。考虑到野外环境温度值的变化情况,实验中 ξ 取值为 3.0。

实验分为三个部分:(1)分别测评在不同 λ 、 ε 下融合算法的融合精度(融合结果与目标真实值的均方差);(2)测量不同网络规模下 EERSFS 的去冗余量;(3)与 CLIF 方法及 EIFS 方法进行对比,测量网络的数据传输总量。

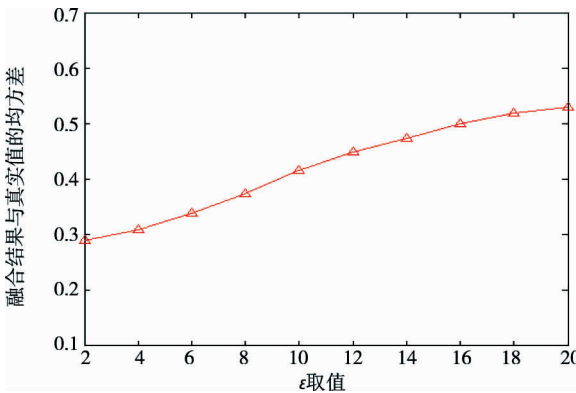
实验 1:不同 λ 、 ε 值对融合结果精度的影响。利用 20 个 WRFID 节点进行 50 次环境温度采样。

λ 值的大小代表了数据一致性检验的精度。值越小,参加融合处理的测量值越多。实验中 ε 、 ξ 取值固定,通过改变 λ ,观察融合结果与目标真实值之间的均方差。实验结果如图 5(a)所示。实验结果表明,当 λ 取 0.7 左右时,融合结果与真实值的误差最小,数据融合结果的真实度最高。

ε 值的大小代表了数据融合的精度。实验中 λ 、 ξ 取值固定,通过改变 ε ,观察融合结果与目标真实值之间的均方差。实验结果如图 5(b)所示。实验结果表明, ε 值越小,融合迭代的次数越多,融合结果与真实值的误差越小,数据融合结果的真实度越高。考虑到 ε 的取值会影响融合迭代次数,这对节点的计算能力和网络的实时性是一种挑战。因此,在以下实验中,取 $\varepsilon = 10$, $\lambda = 0.7$ 。



(a) 固定 ε 、 ξ 的实验结果



(b) 固定 λ 、 ξ 的实验结果

图 5 λ 、 ε 对融合结果的影响

实验 2:网络节点密度对冗余数据过滤量的影响。网络的节点密度可以表示为每个簇中的节点个数。节点密度越大,标签被多个节点同时获取的概率越大,生成的冗余数据就越多。实验在网络中随

机分布 100 个标签节点, 改变节点密度, 观察网络的数据过滤比例 $\beta = \frac{NFPN - FPN}{NFPN}$, 其中 $NFPN$ 表示不经过冗余过滤处理时, 网络传输的数据包总量, FPN 表示利用本文提出的方法进行过滤后, 网络传输的数据包总量。实验结果如图 6 所示。可以看出, 随着网络节点密度的增加, 网络数据过滤比例也随之增加。

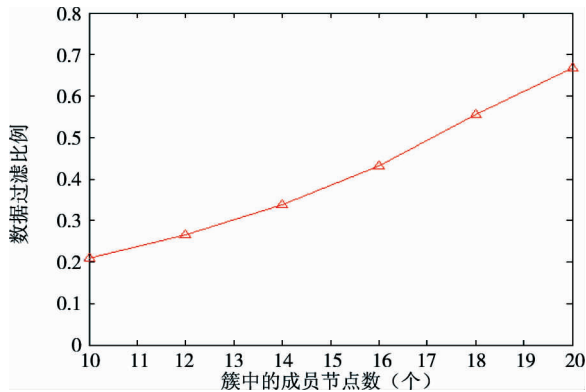


图 6 节点密度对冗余过滤量的影响

实验 3: 测量网络传输的数据包总量。本文提出的 EERSFS 方法是采用冗余分层过滤的思想, 通过簇头间信息交换, 在冗余源簇头进行过滤处理, 从而降低网络中的冗余数据传输量, 减少数据传输造成的能量开销。本实验通过增加网络中移动标签的个数, 就网络数据传输总量这一指标, 进行了 EERSFS 方法与文献[16]提出的 CLIF 方法及文献[6]提出的 EIFS 方法的对比, 实验结果如图 7 所示。结果表明, EERSFS 方法在降低冗余数据量上具有较明显的优势。

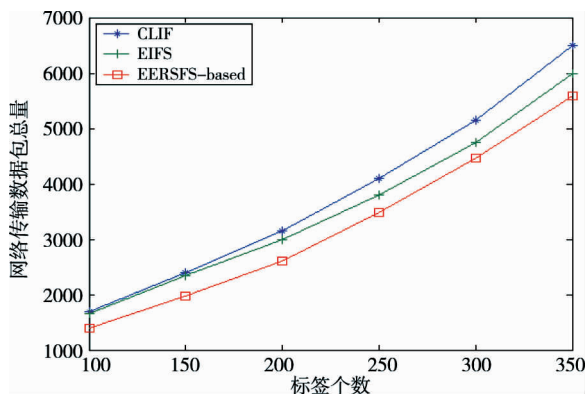


图 7 网络传输的数据包总量

4.3 实验结论

通过上述实验分析可知, 本文提出的分层过滤模型降低了混杂 RFID 网络中的冗余数据量, 同时, 引入的测量数据接近度矩阵, 有效地消除了虚假数据对融合精度的影响。实验表明, EERSFS 模型大大减少了由于冗余数据传输造成的能量消耗, 达到了延长无线传感网络寿命的目的。

5 结 论

本文提出的混杂 RFID 网络, 通过无线传感技术实现了 RFID 的多跳通信。同时, 针对混杂 RFID 网络的特点, 提出了一种基于簇结构的冗余过滤模型。该模型采用数据交换机制, 通过分层冗余过滤, 有效地减少了混杂 RFID 网络中的冗余数据, 避免了由于冗余数据传输带来的能量消耗, 延长了无线传感网络的寿命。下一步工作将在本文现有成果的基础上, 进一步研究在节点考虑自身剩余能量以及路由距离情况下的冗余过滤方法, 以达到网络内部能量平衡的目的。

参考文献

- [1] ITU Internet Reports 2005: The Internet of Things, Genève, ITU, 2005. <http://www.itu.int/osg/spu/publications/internetofthings/>, 2010
- [2] Lopez T S, Kim D, Canepa G H, et al. Integrating wireless sensors and RFID tags into energy-efficient and dynamic context networks. *The Computer Journal*, 2008, 52: 240-267
- [3] McKelvin M L Jr, Williams M L, Berry N M. Integrated Radio Frequency identification and Wireless Sensor Network architecture for automated inventory management and tracking applications. In: *Proceedings of the 2005 Conference on Diversity in Computing*, Albuquerque, USA, 2005. 44-47
- [4] Zhang L, Wang Z. Integration of RFID into wireless sensor networks: architectures, opportunities and challenging problems. In: *Proceedings of the 5th International Conference on Grid and Cooperative Computing Workshops (GCCW 06)*, Changsha, China, 2006. 463-469
- [5] Carbutar B, Ramanathan M K, Koyuturk M, et al. Redundant reader elimination in RFID systems. In: *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, Santa Clara, USA, 2005. 176-184
- [6] Bashir A K, Lim S J, Hussain C S, et al. Energy efficient

- in-network RFID data filtering scheme in Wireless Sensor Networks. *Sensors*, 2011, 11: 7004-7021
- [7] Tran T, SuRon C, Cocci R, et al. Probabilistic Inference over RFID streams in mobile environments. In: Proceedings of IEEE 25th International Conference on Data Engineering, Shanghai, China, 2008. 1096-1107
- [8] Chen H Q, Ku W S, Wang H X, et al. Leveraging spatio-temporal redundancy for RFID data cleansing. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, New York, USA, 2010. 51-62
- [9] Jeffery S R, Garofalakis M, Franklin M J. Adaptive cleaning for RFID data streams. In: Proceedings of the 32nd International Conference on Very Large Data Bases VLDB, Seoul, Korea, 2006. 163-174
- [10] Wang F, Liu P. Temporal management of RFID data. In: Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, 2005. 1128-1139
- [11] Kadayif I, Kandemir M. Tuning in-sensor data filtering to reduce energy consumption in wireless sensor networks. In: Proceedings of Design, Automation and Test in Europe Conference and Exhibition, Paris, France, 2004. 1530-1539
- [12] Choi W, Park M S. In-network phased filtering mechanism for a large-scale RFID inventory application. In: Proceedings of the 4th International Conference on IT & Applications (ICITA), Harbin, China, 2007. 401-405
- [13] Sasiadek J Z, Hartana P. Sensor data fusion using Kalman filter. In: Proceedings of the Third International Conference on Information Fusion, 2000. WED5/19-WED5/25, Vol. 2
- [14] 胡玉胜等. 基于贝叶斯网络的不确定性知识的推理方法. 计算机集成制造系统, 2001, 7(12): 652-658
- [15] 毛关利华, 杜昌平. 基于平均值迭代的多目标数据融合算法. 计算机应用, 2012, 32(6): 1759-1759, 1765
- [16] Kim D S, Kashif A, Ming X, et al. Energy efficient in-network phase Rfid data filtering scheme. In: Proceedings of the 5th International Conference on Ubiquitous Intelligence and Computing, Oslo, Norway, 2008. 311-322

EERSFS: an energy efficient hybrid-RFID data filtering method

Luo Qingping, Wang Yongli, Zhang Gongxuan

(Department of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094)

Abstract

To improve the applications of the wireless sensing and radio frequency identification (RFID) in the Internet of things, the concept of cluster was introduced into the hybrid RFID network formed by the combination of wireless sensing and RFID, and an energy-efficient RFID and sensor data filtering scheme (EERSFS) for the hybrid RFID network was put forward. The EERSFS defines the redundancy hierarchically, and makes redundant data filtered near by the redundant sources. The results of the simulation experiments show that the proposed method can reduce the amount of redundant data and the energy consumption in the network, and prolong the life cycle of wireless sensor networks.

Key words: wireless sensing, hybrid RFID, redundant filtration