移动传感网中基于定向角度路由的数据收集

梁俊斌, 邹绍军*, 李陶深

广西大学计算机与电子信息学院, 南宁 530004

* 通信作者. E-mail: zousj2014@163.com

收稿日期: 2014-09-07; 接受日期: 2014-12-01

国家自然科学基金 (批准号: 61103245, 61363067)、广西自然科学基金 (批准号: 2012GXNSFB053163)、香江学者计划项目和 2013 年广西高等学校中青年优秀骨干教师培养工程资助项目

摘要 在 Sink 和节点不断移动的多跳无线传感器网络中, 节点和 Sink 的位置不断变化. 如果节点采用传统的广播方式发送数据, 节点发送的数据在到达移动 Sink 之前, 很有可能被大量的节点不断转发. 这样不仅容易导致数据冗余过大, 而且容易造成节点的能量消耗过快从而缩短网络生命周期. 为了最大化网络生命周期, 提出一种节能的算法 ROAR 来解决这个问题. 首先, 节点执行类似于选取簇头的收集器选择算法, 能量高的节点以高概率决定是否成为收集器. 然后, 节点将数据发送给有效区域内的邻居节点. 所谓的有效区域是指过节点作连接节点和 Sink 的直线的垂线, 以垂线为边界, 与移动 Sink 同侧的区域. 最后, 收集器采用折半角度传输的方式将数据发送给移动 Sink. 仿真实验结果表明, 与目前已有的算法相比, 算法 ROAR 能够有效地缩短收集一轮数据所需的时间、减少能量开销、延长网络生命周期.

关键词 无线传感器网络 数据收集 路由 角度传输 最大化网络生命周期

1 引言

无线传感器网络 (wireless sensor network, 简称 WSN) 是一种由大量具有传感、数据处理、无线通信能力以及能量有限且无法补充的节点自组织而成的多跳网络. 近年来, 随着传感器技术和物联网技术的迅猛发展, 与传感器网络相关的应用在我们现实生活中不断的涌现, 如医学上对病人的健康监测、移动旅游、动物的追踪和监视等. 这些应用与人和动物息息相关且具有很大的现实意义和应用价值.

在现实生活中, 无论是人还是动物, 他们的移动是不可避免的. 随着人或动物的移动, 他们所携带的传感器节点形成网络拓扑不断变化的无线传感器网络. 以医院对病人的健康监测为例, 医护人员给病人发放小型可穿戴传感器, 穿戴在病人身上的传感器可以获取病人的血压、体温、肺活量、心脏跳动频率等生命体特征. 医护人员携带处理能力强的手持式移动数据收集终端 (Sink), 利用 Sink 即可收集到传感器所感知的人体重要数据并对收集到的数据进行处理分析. 根据 Sink 返回的处理结果, 医护人员可以随时监控到病人的身体状况并对有突发病情的病人进行及时有效的救治. 再如, 为了监测某个水域的渔业资源 (如鱼的数量、活动范围等信息), 在部分大鱼身上安装水下传感器节点并且把 Sink 安装在小船上. 收集者划动小船即可收集相关数据进行分析和研究. 随着人或动物的移动, 他们所携

引用格式: 梁俊斌, 邹绍军, 李陶深. 移动传感网中基于定向角度路由的数据收集. 中国科学: 信息科学, 2015, 45: 111–128, doi: 10.1360/N112014-00256

带的传感器节点形成网络拓扑不断变化的无线传感器网络, 节点发送的数据很有可能在网络中不断被转发, 使得节点能量消耗过快, 缩短网络生命周期.

为了能够在 Sink 和节点不断移动的多跳无线传感器网络中快速、准确的收集数据以及最大化网络生命周期, 本文提出随机定向角度路由的数据收集算法 (random orientation angle routing data collection algorithm, ROAR). 首先, 节点执行收集器选择算法, 能量高的节点以更高的概率决定是否成为收集器. 然后, 节点将数据发送给有效区域内的邻居节点. 最后, 收集器采用折半角度传输的方式将数据发送给移动 Sink. 仿真实验结果表明, 与目前已有的算法相比, 算法 ROAR 能够有效地缩短收集一轮数据所需的时间、减少能量开销、延长网络生命周期.

本文第 2 节介绍 WSN 中数据收集协议的研究现状和相关工作. 第 3 节是网络模型和问题描述. 第 4 节阐述算法 ROAR 并进行理论分析. 第 5 节通过仿真实验, 验证算法 ROAR 的有效性. 第 6 节进行总结.

2 相关工作

目前有很多工作采用移动节点对数据收集进行研究, 根据网络中 Sink 和节点的移动特性以及他们之间的协作关系, 主要分为以下几类:

(1) 面向 Sink 移动、节点静止的研究

(i) 路径固定的移动 Sink 的数据收集协议

对于路径固定的移动 Sink 的数据收集协议, 移动 Sink 的路径是预先确定好的. 为了均衡数据收集过程中节点的能量消耗. 首先, 将 Sink 移动的区域设置为缓冲区. 然后, 节点将数据沿最短路径传输到缓冲区内能量高的节点上存放. 最后, Sink 在移动过程中收集缓存区内节点的数据. 比较典型的基于路径固定的移动 Sink 的数据收集协议有 MSMA^[1], PMDGM^[2] 和传感器网络中具有负载平衡的移动协助数据收集模式^[3] 等. 此外, 还有部分工作采用地理由^[4~8] 和多径路由^[9~11] 传输数据. 但 Sink 移动轨迹固定时, Sink 由于受到有限的通信时间和节点随机分布的影响, 很难在降低总通信能耗的同时提高数据采集量.

(ii) 路径可变的移动 Sink 的数据收集协议

对于路径可变的移动 Sink 的数据收集协议, 移动 Sink 出发后将根据网络的状况不断调整自己的路线, 以尽量节省网络中所有节点能量耗费的方式来采集数据. 典型的路径可变的移动 Sink 的数据收集协议有 SenCar^[12], TECM^[13], PBRHC^[14] 和 BRH-MDG^[15] 等. 但是, 它们所针对的网络模型过于理想化, 导致得到的结果很难应用于实际环境.

(2) 面向 Sink 移动、节点随机移动的研究

在实际的应用中, 越来越多的各类传感器被安装到设备 (如移动电话、运动鞋)、人或动物等身上. 安装在它们身上的节点形成网络拓扑不断变化的传感器网络. 然后, 移动传感器通过多跳的方式将自身感知的数据和接收的数据发送给移动 Sink, 进而完成传感器网络中的数据收集. 传统的工作是利用广播^[16~20] 来分发数据. 但是, 广播协议难以控制数据传输冗余并且能量有效性较低. 因此, 近年来有工作提出采用角度传输来解决问题. 比较典型的无线传感器网络中 Sink 和节点随机移动的数据收集协议有 Sidewinder^[21] 和 SMITE^[22].

Sidewinder 是在网络拓扑不断变化的情况下进行数据收集, 网络中的节点在面向 Sink 成 60 度角的区域内选取邻居节点转发数据且为确保数据能够被 Sink 接收, 当转发数据的节点在 Sink 的一跳范

围内时, 节点采用广播的方式将数据发送两跳范围内的邻居节点. 但是, Sidewinder 在选取节点转发数据的时候没有考虑节点的能量且采用广播方式发送数据, 使得整个网络中的消息开销仍然较大.

SMITE 也是在网络拓扑不断变化的情况下进行数据收集, 网络中的 Sink 和节点都是在不断移动. SMITE 采用了三个阶段进行数据收集: (1) 收集器的选择; (2) 直接传输; (3) 角度传输. 虽然 SMITE 与其他同类的数据收集协议相比, 无论是在数据率、延迟, 还是在网络中的消息开销上都有很大的提升. 但是, 在角度传输阶段很有可能出现在角度传输范围内没有邻居节点的情形, 尤其是节点和 Sink 节点之间的距离较远的情况下, 这样势必造成角度传输的数据交付率下降, 甚至在规定一轮的时间内无法将数据发送 Sink.

本文借鉴 SMITE 的思想, 针对 Sink 和节点不断随机移动所形成网络拓扑不断变化的 WSN 中的数据收集问题, 提出改进 SMITE 的数据收集协议, 简称 ROAR. 改进后的协议不仅能够避免在 SMITE 所定义的传输角度范围内没有节点而无法进行数据传输的情况, 而且能够控制数据的传输方向和均衡节点的能量消耗从而有效缩短收集一轮数据所需要的时间、减少能量开销和延长网络生命周期.

3 系统模型和问题描述

文献 RPMA^[23], MLMME^[24] 等, 都是假设网络中只有单个或者多个节点在网络中移动, 这样的好处是能够有效地均衡网络中各节点的能量消耗. 但是在实际的应用中, 越来越多的各类节点和 Sink 被安装到人或动物身上. 在现实生活中, 无论是人还是动物, 他们的移动是不可避免的. 这些节点随着人或动物的移动, 形成网络拓扑不断变化的无线传感器网络, 节点发送的数据很有可能在网络中不断被转发, 使得节点的能量消耗过快, 缩短网络生命周期. 在该网络模型下, 上面的数据收集方案不再适用. 鉴于实际应用中的需要, 本论文研究的是在 Sink 和节点都是随机移动的无线传感器网络中进行数据收集, 最大化网络生命周期.

3.1 网络模型

在面积为 a 平方米的正方形区域内随机部署一个移动 Sink 和 n 个移动节点. 整个传感器网络组成一个无向图 $G(V, E)$, 其中: V 是节点集合, $V = v_0, v_1, \dots, v_{n-1}, v_n$, $|V| = n+1$. v_0 表示 Sink, $v_1, v_2, \dots, v_{n-1}, v_n$ 表示节点; E 是边的集合. 如果两个节点 v_i 和 v_j 相互处于对方的通信区域内, 则边 $(v_i, v_j) \in E$. $|E| = m$ 为边的数量. 网络具有如下性质:

- 网络中的节点的存储空间是有限的.
- 各节点的初始能量异构且不能补充.
- Sink 的能量充足.
- 移动 Sink 和节点的通信半径为 r .
- 网络中的 Sink 和节点随机移动.
- 网络拓扑不断变化.

为了简化问题和便于讨论, 假设节点采用固定发射和接收功率, 发射 1 bit 数据需要的能量是 E_t , 接收 1 bit 数据需要的能量是 E_r ^[25,26]. 假设 MAC 层有可靠的包传输机制, 以避免数据在传输过程中发生拥塞和重传.

3.2 相关定义

定义 1(时间轮) 将时间分割成大小相等的时间戳, 2 个连续的时间戳形成一时间轮, 时间轮的长度为 $L = 1/(1 - (1 - \pi r^2/a^2)^{n \times p})^{[22]}$. 其中, p 为节点决定是否成为收集器的概率.

定义 2(网络生命周期) 网络中第 1 个节点死亡时, 该节点存活的轮数^[27] 是网络生命周期.

定义 3(通信区域) 以节点所在位置为圆心, r 为半径的圆所覆盖的区域.

定义 4(预测区域) 以 Sink 所在位置为圆心, 预测半径 $r_p^{[22]}$ 为半径的圆所覆盖的区域.

定义 5(有效区域) 设过节点和 Sink 的直线为 l , 过节点做直线 l 的垂线 m , 以垂线 m 为边界, 与移动 Sink 同侧的区域.

定义 6(定向广播) 节点将数据发送到有效区域的过程.

定义 7(速度方向) 以 Sink 在 t_{i-1} 时刻所在位置为起点, 以 Sink 在 t_i 时刻所在位置为终点的矢量方向即为 Sink 在 t_i 的速度方向.

定义 8(传输区域) 设过节点和 Sink 的直线为 l , 分别过节点和 Sink 作直线 l 的垂线 m_1, m_2 , 两垂线 m_1, m_2 之间的区域.

定义 9(方向区域) 设过节点和 Sink 的直线为 l , 以直线 l 为界, 与 Sink 的速度方向同侧的区域.

定义 10(非方向区域) 设过节点和 Sink 的直线为 l , 以直线 l 为界, 与 Sink 的速度方向异侧的区域.

定义 11(角度传输区域) 过节点做预测区域的切线, 在传输区域内, 两切线所形成夹角的区域.

定义 12(方向传输区域) 方向区域与传输区域重叠的部分在减去角度传输区域所得区域.

定义 13(非方向传输区域) 非方向区域与传输区域重叠的部分在减去角度传输区域所得区域.

3.3 问题描述

在传统的基于移动 Sink 的数据收集协议中, 移动 Sink 的路径往往是固定的. 这样, 移动 Sink 只要按照固定路径移动即可收集到全网节点感知的数据. 但是在实际应用中, 越来越多的各类传感器被安装到设备 (如移动电话、运动鞋)、人或动物等身上. 随着设备、人或动物的移动, 安装在他们身上的节点形成网络拓扑不断变化的无线传感器网络, 造成在该网络模型下收集数据困难.

由于节点和 Sink 都是随机移动, 移动 Sink 无法采用传统的基于固定路径的数据收集协议访问网络中的每一个节点收集数据. 此外, Sink 的随机移动, 导致节点无法确定 Sink 的位置, 故无法将感知的数据准确的发送给移动 Sink.

假设移动 Sink 通过 GPS 的方式确定当前所处位置并将它的位置信息发送给节点, 如果节点采用传统的广播方式将数据发送给邻居节点, 数据也是无法准确的发送给 Sink. 因为 Sink 是不断随机移动, 如果节点无法在较短的时间内将数据发送给移动 Sink, 等到数据到达 Sink 所在位置的时候, Sink 已经移动到其他位置, 导致数据传输失败. 节点如何准确、快速、高效的将数据发送给移动 Sink 成为在该网络模型下收集数据的一大难点.

为此, 本文充分考虑 WSN 中网络拓扑不断变化、节点的缓存空间有限、能量小且不可补充等特点, 结合网络的实时性和收集路径的随机性的要求, 提出一种节能的算法 ROAR 来解决这个问题. 以下是算法 ROAR 的详细描述.

4 算法 ROAR 的设计

4.1 算法的基本思想

在无线传感器网络中, 算法 ROAR 分 3 个阶段完成数据收集. 第 1 阶段, 节点执行收集器选择算法, 以概率 p 决定是否成为收集器, 能量越高的节点成为收集器的概率越大. 第 2 阶段, 收集器 v_i 广播数据收集消息 $\text{Collect}(v_i)$, 收到数据收集消息且未发送数据的节点将数据发送给有效区域内的邻居节点. 第 3 阶段, 当收集器的存储容量已满或到一时间轮的结尾时, 收集器采用折半角度传输的方式将数据的发送给移动 Sink. 为了便于观察, 采用着色算法对不同类型的节点进行着色, 用红色标识普通节点、绿色标识收集器. 移动 Sink 属于收集器.

4.2 算法 ROAR 的描述

下面分三个阶段详细描述算法 ROAR 是如何从网络中选取节点作为收集器以及节点是如何实现定向广播数据和折半角度传输数据, 从而实现均衡节点的能量消耗, 最大化网络生命周期.

4.2.1 第一阶段收集器的选择

假设每个节点都有一个类型属性 type 和随机属性 rand . 如果 $\text{type}=1$, 则节点为普通节点; 如果 $\text{type}=2$, 则节点为收集器. 每个节点 v_i 以概率 $p=\lceil a/r^2 \rceil/n + e_{\text{now}}/(n \times e_0)$ 决定是否成为收集器. 其中, a 是网络的面积, r 是节点的通信半径, n 是节点的数量, e_{now} 是节点的当前剩余能量, e_0 是节点的初始能量. 节点 v_i 产生一个 0 到 1 的随机数 rand , 如果 $0 \leq \text{rand} \leq p$, 则节点 v_i 成为收集器, 更新 type 的值为 2. 否则, 节点 v_i 为普通节点, 它的 type 值为 1. 节点执行收集器选择算法时无需与它的邻居节点进行通信, 这样可以有效节约能量. 收集器选择算法 $\text{CollectorSelection}(n)$ 的具体描述如算法 1 所示.

算法 1 CollectorSelection

Require: 节点的数量 n ;
Ensure: 收集器的集合 collectorsSet ;
 1: $p=\lceil a/r^2 \rceil/n + e_{\text{now}}/(n \times e_0)$;
 2: **for** 网络中的每一个节点 v_i **do**
 3: $\text{node}(v_i).\text{rand}=\text{rand}(1,1)$; // 节点 v_i 产生一个随机数;
 4: **if** $0 \leq \text{node}(v_i).\text{rand}$ and $\text{node}(v_i).\text{rand} \leq p$ **then**
 5: $\text{node}(v_i).\text{type}=2$;
 6: 将节点着成绿色;
 7: 将节点 v_i 加入收集器集合 collectorsSet ;
 8: **else** $\{\text{node}(v_i).\text{type}=1\}$
 9: 将节点着成红色;
 10: **end if**
 11: **end for**

定理 1 如果所有收集器的通信区域不重叠, 则所有收集器的通信区域能够覆盖整个网络区域.

证明 用随机变量 x_i 表示节点 v_i 的类型, 令 $x_i=1$ 表示节点 v_i 为收集器, $x_i=0$ 表示节点 v_i 为普通节点, 则 $p(x_i=0)=1-p$, $p(x_i=1)=p$. 由于 $x_1, x_2, \dots, x_n$ 相互独立且服从二项分布, 故收集器的数量 $c = \sum_{i=1}^n (x_i) = n \times p$. 如果所有收集器之间的通信区域不重叠, 则所有收集器的覆盖面积为

$n \times p \times \pi \times r^2$. 由于

$$\begin{aligned}
 n \times p \times \pi \times r^2 &= n \times \left(\lceil a/r^2 \rceil / n + e_{\text{now}} / (n \times e_0) \right) \times \pi \times r^2 \\
 &= \pi \times r^2 \times \lceil a/r^2 \rceil + \pi \times r^2 \times e_{\text{now}} / e_0 \\
 &> \pi \times r^2 \times \lceil a/r^2 \rceil \\
 &> r^2 \times (a/r^2) = a,
 \end{aligned}$$

其中, n 是节点的数量, p 是节点决定是否成为收集器的概率, r 是节点的通信半径, a 是网络的面积, e_{now} 是节点的当前剩余能量, e_0 是节点的初始能量.

故所有收集器的通信区域能够覆盖整个网络区域. 得证.

4.2.2 第二阶段普通节点采用定向广播的方式发送数据

假设每个节点 v_i 有一个发送属性 send 和一个接收集合属性 receiveSet . 如果 $\text{send}=0$, 则节点 v_i 未发送数据到有效区域内的邻居节点. 如果 $\text{send}=1$, 则节点 v_i 已经将数据发送给有效区域内的邻居节点. 网络中的节点执行完收集器选择算法后, 收集器 v_c 开始收集数据. 由于收集器 v_c 的邻居节点不断发生变化, 为了收集到更多的数据, 收集器 v_c 需要定期的广播数据收集消息 $\text{Collect}(v_c)$ 到它的所有邻居节点, 宣称自己是收集器.

收集器 v_m 收到收集器 v_c 广播的数据收集消息 $\text{Collect}(v_c)$ 时, 它将丢弃收到的数据收集消息 $\text{Collect}(v_c)$. 普通节点 v_i 收到收集器广播的数据收集消息 $\text{Collect}(v_c)$ 时, 按以下规则处理: (1) 如果普通节点 v_i 的 send 值为 0, 则普通节点 v_i 采用定向广播的方式将数据 $\text{Data}(v_i)$ 发送到有效区域内的邻居节点并且立即改变它的 send 值为 1; (2) 如果普通节点 v_i 的 send 值为 1, 则普通节点 v_i 已经将数据发送出去, 故普通节点 v_i 直接丢弃收集器 v_c 所发送的数据收集消息 $\text{Collect}(v_c)$. 这样可以防止节点多次发送同一数据, 有效节约节点的能量消耗进而延长网络的生命周期.

接收到数据收集消息 $\text{Collect}(v_c)$ 的普通节点 v_i 将数据 $\text{Data}(v_i)$ 发送给有效区域的邻居节点. 在有效区域内的邻居节点 v_j 收到数据 $\text{Data}(v_i)$ 时, 按以下规则处理: (1) 如果节点 v_j 是收集器且其能量大于节点 v_i 的能量, 则收集器 v_j 将数据 $\text{Data}(v_i)$ 存储到它的存储空间并且把节点 v_i 的接收集合添加到接收集合中, 表示收集器 v_j 已经接收到节点 v_i 所收集到的所有数据; (2) 如果节点 v_j 是普通节点、存储空间未满足且没有将自己的数据发送给有效区域内的邻居节点, 则节点 v_j 将数据 $\text{Data}(v_i)$ 存储到它的存储空间并且把节点 v_i 的接收集合添加到接收集合中, 表示节点 v_j 已经接收到节点 v_i 所收集到的所有数据; (3) 如果节点 v_j 是普通节点且已经将自己的数据发送给有效区域内的邻居节点, 则节点 v_j 将丢弃数据 $\text{Data}(v_i)$; (4) 如果节点 v_j 没有将自己的数据发送给有效区域内的邻居节点, 但存储空间已满, 则节点 v_j 将丢弃数据 $\text{Data}(v_i)$.

为详细描述普通节点是如何采用定向广播的方式发送数据的, 下面以图 1 为例. 在图 1 中, 收集器 v_1 广播数据收集消息 $\text{Collect}(v_1)$, 普通节点 v_4 收到广播消息后, 并不采用广播的方式发送数据, 而是采用定向广播的方式发送数据. 尽管普通节点 v_2 和 v_5 都是节点 v_4 的邻居节点, 但由于它们都不在普通节点 v_4 的有效区域内, 故它们不会接收普通节点 v_4 发送的数据包 $\text{Data}(v_4)$. 只有在普通节点 v_4 有效区域内的收集器 v_{11} 和普通节点 v_6, v_7 才会收到数据包 $\text{Data}(v_4)$. 因为收集器 v_{11} 的能量大于节点 v_4 的能量, 所以收集器 v_{11} 接收并存储数据 $\text{Data}(v_4)$. 因为普通节点 v_4 的邻居节点 v_6 的存储空间未满足且没有发送自身的数据 $\text{Data}(v_6)$ 给收集器 v_{11} , 所以普通节点 v_6 接收并存储数据 $\text{Data}(v_4)$.

图 1 定向广播

Figure 1 Directional broadcast

因为普通节点 v_7 已经发送自身的数据 $\text{Data}(v_7)$ 给收集器 v_{11} , 所以普通节点 v_7 丢弃普通节点 v_4 发送的数据 $\text{Data}(v_4)$. 这种定向广播的方式能够有效控制数据的传输方向, 避免数据向无效区域传播, 极大地减少了网络中的通信量, 节约节点能量开销. 定向广播算法 $\text{DirectionalBroadcast}(n, \text{TDe}, \text{RDe}, \text{EOverhead}, \text{collectorsSet})$ 的具体描述如算法 2 所示.

4.2.3 第三阶段收集器将数据以折半角度传输的方式发送给移动 Sink

收集器收集完数据后, 收集器将数据以折半角度传输的方式发送给移动 Sink. 通过以下几个步骤实现数据的折半角度传输.

首先, 移动 Sink 采用波段概率广播的方式局部泛洪它的状态信息.

假设 WSN 中的节点知道信号衰减的模型. 节点接收到 Sink 发送的状态信息时, 根据接收到信息信号的强弱, 利用波段映射函数 $\text{BMF}^{[28]}$ 计算自身的波段值即 $\text{BMF}(\text{SIS_Strength}) \rightarrow \text{Band_Number}$, 其定义如下:

$$\text{Band_Number} = \begin{cases} 1, & \lambda_1 \leq \text{SIS_Strength} < \lambda_0, \\ i, & \lambda_i \leq \text{SIS_Strength} < \lambda_{i-1}, \\ N, & \lambda_{N=0} \leq \text{SIS_Strength} < \lambda_{N-1}, \end{cases} \quad (1)$$

其中, SIS_Strength 表示信号的强度, Band_Number 表示波段值.

针对概率广播, 我们采用 APBBDP^[29] 模型, 其定义如下:

$$P_c^i = \begin{cases} (\pi \times \log n - 2) / (2|N(v_i)|), & |N(v_i)| > \pi \times \log n - 2, \\ 1, & \text{otherwise.} \end{cases} \quad (2)$$

算法 2 DirectionalBroadcast

Require: 节点的数量 n $\vee$ 节点发送数据的能耗 TDe $\vee$ 节点接收数据的能耗 RDe $\vee$ 能量开销 EOverhead $\vee$ 收集器集合 collectorsSet;

Ensure: 能量开销 EOverhead;

```

1: for 收集器集合 collectorsSet 中的每个收集器  $v_a$  do
2:   收集器  $v_a$  定期的广播数据收集消息 Collector( $v_a$ ) 到它的所有邻居节点, 宣称自己是收集器;
3:   for 收集器  $v_a$  的每个邻居节点  $v_b$  do
4:     if node( $v_b$ ).type==1 and node( $v_b$ ).send==0 then
5:       node( $v_b$ ).e = node( $v_b$ ).e - TDe;
6:       EOverhead=EOverhead+TDe;
7:       node( $v_b$ ).send=1;
8:       for 节点  $v_b$  的每个邻居节点  $v_c$  do
9:         if 节点  $v_c$  在有效区域内且未发送数据 then
10:          node( $v_c$ ).e= node( $v_c$ ).e - RDe;
11:          EOverhead=EOverhead+RDe;
12:          node( $v_c$ ).receiveSet=union(node( $v_c$ ).receiveSet, node( $v_b$ ).receiveSet);
13:        end if
14:      end for
15:    end if
16:  end for
17: end for

```

其中, v_i 表示网络中节点, $|N(v_i)|$ 表示节点 v_i 的邻居节点数量, n 表示无线传感器网络中全部节点的数量。

收集器节点完成折半角度传输需要获取移动 Sink 的位置、移动速度等信息。因此, 任意节点 v_i 保存移动 Sink 最近的 k 个速度信息 (简称速度序列): $v_{t_1}, v_{t_2}, \dots, v_{t_k}$ 以及最近两个位置 $(x_{t_{k-1}}, y_{t_{k-1}})$, (x_{t_k}, y_{t_k}) 。其中, t_{k-1}, t_k 是最近的两个时间。为了节约节点的能量和减少网络中的通信量以及数据冗余, 节点 v_i 采用波段概率广播的方式定期局部的广播移动 Sink 的状态信息 stateMessage(x_t, y_t, v_t, h_s , BMF, b_i); 其中, x_t, y_t, v_t 分别表示移动 Sink 在 t 时刻的 x 坐标, y 坐标, 移动速度; $h_s = \lceil \sqrt{n \times p} \rceil$ [22] 表示 Sink 波段概率广播的跳数; b_i 表示节点 v_i 的波段值。

节点 v_i 收到节点 v_l 转发的状态信息 stateMessage(x_t, y_t, v_t, h_s , BMF, b_l) 时, 节点 v_i 按下面的规则处理状态信息: (1) 对于任意的 $j \in [1, k]$, 如果 $t \neq t_j$, 说明状态信息 stateMessage(x_t, y_t, v_t, h_s , BMF, b_l) 是一个“新信息”, 则按以下情形处理: (a) 如果 $t < t_j$, 节点 v_i 直接丢弃状态信息 stateMessage(x_t, y_t, v_t, h_s , BMF, b_l); (b) 如果 $t_1 < t < t_j$, 则节点 v_i 删除最早的速度, 并将状态信息中的 v_t 按照时间递增的顺序插入到速度序中, 但此时并不更新移动 Sink 的位置; (c) 如果 $t > t_j$, 则该状态信息是最新的, 故用 v_t 取代 v_{t_1} 并且用 (x_t, y_t) 取代 $(x_{t_{k-1}}, y_{t_{k-1}})$ 。节点 v_i 利用 BMF 函数计算自身的波段值 b_i , 如果 $b_l < b_i < h_s$, 则节点 v_i 以 p_c^i 的概率转发状态信息 stateMessage(x_t, y_t, v_t, h_s , BMF, b_i)。否则, 节点 v_i 丢弃状态信息 stateMessage(x_t, y_t, v_t, h_s , BMF, b_l)。 (2) 如果 $t = t_j$, 说明节点已经接收到了状态信息 stateMessage(x_t, y_t, v_t, h_s , BMF, b_l), 故直接将其丢弃状态信息 stateMessage(x_t, y_t, v_t, h_s , BMF, b_l)。波段概率广播状态信息算法 stateMessageTransmission() 的具体描述如算法 3 所示。

为详细描述节点是如何采用波段概率广播的方式发送状态信息的, 下面以图 2 为例。例如, 在图 2 中, 节点 v_4 和 v_{15} 接收到节点 v_7 发送的状态信息 stateMessage(x_t, y_t, v_t, h_s , BMF, 2), 节点 v_4 和 v_{15} 通过 BMF 计算得到自身的波段值分别是 1, 3。因为节点 v_4 的波段值 1 小于节点 v_7 的波段值 2, 所以节点 v_4 丢弃节点 v_7 发送的状态信息 stateMessage(x_t, y_t, v_t, h_s , BMF, 2)。因为节点 v_{15} 的波段

算法 3 StateMessageTransmission

```

1: if 状态信息中的时间  $t$  大于  $t_1$  并且小于  $t_k$  then
2:   节点  $v_i$  删除最早的速度并将状态信息中的  $v_t$  按照时间递增的顺序插入到速度序中;
3: end if
4: if 状态信息中的时间  $t$  大于  $t_k$  then
5:   节点  $v_i$  删除最早的速度并将状态信息中的  $v_t$  按照时间递增的顺序插入到速度序中;
6:   节点  $v_i$  利用 BMF 函数计算自身的波段值  $b_i$ 
7:   if 状态信息中的时间  $t$  大于  $t_1$  并且小于  $t_k$  then
8:     节点  $v_i$  以  $p_c^i$  的概率转发状态信息;
9:   end if
10: end if

```

图 2 波段概率广播

Figure 2 Band probability broadcast

值 3 大于节点 v_7 的波段值 2, 所以节点 v_{15} 以 p_c^i 的概率进行转发状态信息 $\text{stateMessage}(x_t, y_t, v_t, h_s, \text{BMF}, 3)$. 这种波段概率广播机制有效防止状态信息回传, 不仅保证了在局部泛洪区域内的所有节点能够接收到状态信息, 而且减少数据冗余和网络中的通信量、极大地节约节点的能量消耗, 延长网络生命周期.

然后, 节点 v_i 利用接收到的状态信息, 节点 v_j 与移动 Sink 之间的距离计算出预测半径 r_p 、预测区域、角度传输区域、方向传输区域以及非方向传输区域.

最后, 收集器 v_i 采用集中式算法计算某个邻居节点 v_j 所处区域, 具体计算过程如下:

$$|v_i v_j| \leq r, \quad (3)$$

图 3 折半角度传输

Figure 3 Binary angle transmission

$$0 \leq \arccos((|v_i v_j|^2 + |v_0 v_i|^2 - |v_0 v_j|^2) / (2 \times |v_i v_j| \times |v_0 v_i|)) \leq \arcsin(r_p / |v_0 v_i|), \quad (4)$$

$$\arcsin(r_p / |v_0 v_i|) < \arccos((|v_i v_j|^2 + |v_0 v_i|^2 - |v_0 v_j|^2) / (2 \times |v_i v_j| \times |v_0 v_i|)) \leq \pi/2, \quad (5)$$

其中, $|v_i v_j|$, $|v_0 v_i|$, $|v_0 v_j|$ 分别表示收集器 v_i 和节点 v_j , 移动 Sink 和收集器 v_i , 移动 Sink 和节点 v_j 两点之间的欧几里得距离. 如果同时满足 (3) 和 (4) 两个不等式, 则说明邻居节点在角度传输区域内, 如图 3 所示, 节点 v_9 和 v_{17} 属于这种情况. 如果同时满足 (3) 和 (5) 两个不等式, 则说明邻居节点在方向传输区域或者非方向传输区域内. 如图 3 所示, 节点 v_1 和 v_4 属于这种情况. 为了进一步区分邻居节点是属于方向传输区域还是非方向传输区域内, 节点 v_i 根据本地保存的两个最近位置来确定移动 Sink 的速度方向. 如图 3 所示, 节点 v_4 在方向传输区域内, 节点 v_1 在非方向传输区域内. 计算节点所处区域的算法 $\text{CalculationArea}(v_i, r, r_p)$ 的具体描述如算法 4 所示.

节点根据以下规则选取邻居节点作为目标节点来完成角度传输: (1) 如果在角度传输区域内有邻居节点, 则选取角度传输区域内能量最高的节点作为目标节点进行角度传输; (2) 如果在角度传输区域内没有邻居节点, 则选取方向传输区域内能量最高的节点作为目标节点进行角度传输; (3) 如果在角度传输区域和方向传输区域内都没有邻居节点, 则选取非方向传输区域内能量最高的节点作为目标节点进行角度传输.

由于在短时间内 Sink 的移动范围变化不大, 收集器 v_i 将数据 $\text{Data}(v_i)$ 送给目标节点 v_j 时, 附加发送预测半径 r_p , Sink 的最新位置 (x_{s_t}, y_{s_t}) 以及折半距离跳数 $h_s = \lceil [d/r]/2 \rceil$. 其中, d 是收集器 v_i 与移动 Sink 之间的距离. 首先, 目标节点 v_j 将接收到数据包里面的位置 (x_{s_t}, y_{s_t}) 中的时间 t 与

算法 4 CalculationArea**Require:** 节点 v_i $\vee$ 节点的通信半径 r $\vee$ 预测半径 r_p ;**Ensure:** flag;

```

1: if 节点  $v_j$  在节点  $v_i$  的角度传输区域 then
2:   node( $v_i$ ).AngelSet=union( $v_j$ , node( $v_i$ ).AngelSet);
3:   flag=1;
4:   continue;
5: end if
6: if 节点  $v_j$  在节点  $v_i$  的方向传输区域 then
7:   node( $v_i$ ).DangelSet=union( $v_j$ , node( $v_i$ ).DangelSet);
8:   flag=2;
9:   continue;
10: end if
11: if 节点  $v_j$  在节点  $v_i$  的非方向传输区域 then
12:   node( $v_i$ ).AngelSet=union( $v_j$ , node( $v_i$ ).AngelSet);
13:   flag=3;
14:   continue;
15: end if
16: return flag.

```

本地保存的两个最近位置 $(x_{t_{k-1}}, y_{t_{k-1}})$, (x_{t_k}, y_{t_k}) 中的时间 t_{k-1} , t_k 进行比较: 如果 $t \leq t_{k-1}$, 则丢弃 (x_{s_t}, y_{s_t}) ; 如果 $t > t_k$, 则用 (x_{s_t}, y_{s_t}) 取代 $(x_{t_{k-1}}, y_{t_{k-1}})$. 这样做的目的是使目标节点 v_j 保存移动 Sink 的最新位置. 通过连接移动 Sink 在 t_{k-1} 和 t_k 两个时间点所在的位置可以确定移动 Sink 的移动方向, 目标节点 v_j 利用移动 Sink 的移动方向确定方向传输区域. 然后, 目标节点 v_j 将折半距离跳转数 h_s 减 1, 如果折半距离跳转数不小于 0, 则目标节点 v_j 不必重新计算预测半径, 而是以 (x_{t_k}, y_{t_k}) 为圆心, r_p 为半径的圆作为新的预测区域. 最后, 目标节点 v_j 继续以角度传输的方式将数据 $\text{Data}(v_j)$ 发送给下一个目标节点. 如果在角度传输的过程中发现在角度传输区域内没有邻居节点或者折半距离跳转数 h_s 小于 0, 则节点需要重新计算预测范围后在进行折半角度传输. 以上数据传输过程就是所谓的折半角度传输. 折半角度传输算法 BinaryAngleTransmission($n, v_i, r, r_p, \text{TDe}, \text{RDe}, \text{collectorsSet}, \text{MOverhead}$) 的具体描述如算法 5 所示.

如果节点 v_k 的邻居节点中没有收集器, 则节点 v_k 无法接收数据收集信息, 故节点 v_k 不会将数据发送给有效区域内的邻居节点. 这样导致移动 Sink 无法收集到节点 v_k 的数据. 为了避免这种情况, 算法 ROAR 采用如下方式处理: 如果节点在一时间轮的结尾时, 仍未将数据发送出去, 则节点 v_k 广播它的数据到邻居节点, 邻居节点采用折半角度传输的方式继续发送数据, 直到 Sink 接收到节点 v_k 的数据.

4.3 算法性能分析

本节将从能量消耗、时间复杂度等方面分析算法 ROAR 的有效性.

4.3.1 能量耗费

在面积为 a 的正方形网络区域, 由于节点的感知数据能耗和计算能耗可以忽略不计^[30], 故 WSN 中数据包的发送和接收所消耗的能量成为主要的通信能耗. 节点发送一次大小为 b 位的数据包消耗的能量为 bE_t , 节点的某个邻居节点接收这个数据包消耗的能量为 bE_r . 在最坏的情况下, 节点和 Sink

算法 5 BinaryAngleTransmission

Require: 节点 v_i $\vee$ 节点的通信半径 r $\vee$ 预测半径 r_p $\vee$ 节点发送数据的能耗 TDe $\vee$ 节点接收数据的能耗 RDe $\vee$ 能量开销 EOverhead;

```

1: if 节点  $v_i$  不是 Sink 节点的邻居节点 then
2:   for 节点  $v_i$  的所有邻居节点  $v_j$  do
3:     CalculationArea( $v_i$ ,  $r$ ,  $r_p$ );
4:     switch(flag)
5:     {
6:       case 1: //节点  $v_i$  在角度传输区域内有邻居节点
7:         for 节点  $v_i$  的所有邻居节点  $v_j$  do
8:           选取能量最高的节点  $v_z$  作为目标节点;
9:           break;
10:        end for
11:       case 2: //节点  $v_i$  在方向传输区域内有邻居节点
12:         for 节点  $v_i$  的方向传输区域内的所有邻居节点 do
13:           选取能量最高的节点  $v_z$  作为目标节点;
14:           break;
15:        end for
16:       case 3: //节点  $v_i$  在非方向传输区域内有邻居节点
17:         for 节点  $v_i$  的非方向传输区域内的所有节点 do
18:           选取能量最高的节点  $v_z$  作为目标节点;
19:           break;
20:        end for
21:     }
22:     node( $v_i$ ).e = node( $v_i$ ).e - TDe;
23:     node( $v_z$ ).e = node( $v_z$ ).e - RDe;
24:     EOverhead=EOverhead+TDe+RDe;
25:     node( $v_z$ ).receiveSet=union(node( $v_z$ ).receiveSet, node( $v_i$ ).receiveSet);
26:     BinaryAngleTransmission( $v_z$ ,  $r$ ,  $r_p$ , TDe, RDe, EOverhead);
27:   end for
28: else
29:   if 节点  $v_i$  是 Sink 的邻居节点且收集到其他节点发送的数据 then
30:     node( $v_i$ ).e=node( $v_i$ ).e-TDe;
31:     node( $v_0$ ).e = node( $v_0$ ).e-RDe;
32:     EOverhead=EOverhead+TDe+RDe;
33:   end if
34:   if Sink 接收到所有节点发送的数据 then
35:     本轮数据收集结束;
36:   end if
37: end if

```

分别在网络区域的两个对角点上. 这样, 数据包最多被发送 $\sqrt{2a}/r$ 次后, 可以被 Sink 接收. 因此网络中的发送 k 个数据包到 Sink 的通信能量至多为 $(\sqrt{2a}/r) \times k \times b \times (E_t + E_r)$.

4.3.2 时间复杂度

定理 2 算法 ROAR 的时间复杂度为 $O(n^3)$.

证明 算法 ROAR 的第一阶段 (算法 1), 第 1 步到第 11 步是网络中的 n 个节点决定是否成为收集器的操作, 故该阶段的时间复杂度为 $O(n)$. 算法 ROAR 的第二阶段 (算法 2), 第 1 步到第 17 步

是节点采用定向广播发送数据. 其中, 第 2 步是收集器广播数据收集消息的操作; 第 3 步到第 7 步是未发送数据的普通节点接收数据收集消息并且发送数据的操作; 第 8 步到第 14 步是有效区域内未发送数据的邻居节点接收数据的操作, 故该阶段的时间复杂度为 $O(n^3)$. 算法 ROAR 的第三阶段 (算法 5), 第 2 步到第 27 步是节点在传输区域内根据选取目标节点的规则选择能量最高的邻居节点作为目标节点进行角度传输的操作. 其中, 第 3 步是调用 CalculationArea 函数计算收集器的邻居节点属于哪个传输区域的操作; 第 28 步到第 37 步是判断算法 ROAR 是否完成本轮数据收集的操作, 故该阶段的时间复杂度为 $O(n^2)$.

由以上分析, 算法 ROAR 第一阶段的时间复杂度为 $O(n)$; 算法 ROAR 第二阶段的时间复杂度为 $O(n^3)$; 算法 ROAR 第三阶段的时间复杂度为 $O(n^2)$. 因为 $O(n) < O(n^2) < O(n^3)$, 故整个算法的时间复杂度为 $O(n^3)$. 得证.

定理 3 在一个拥有 n 个节点的无线传感器网络中, 如果收集器采用折半角度传输的方式转发数据, 则当 n 充分大时, 收集器收集到的数据能够以接近 1 的概率发送给移动 Sink.

证明 由于所有节点和 Sink 都是随机移动的, 任意节点移动到节点 v_i 的传输区域的概率是 $\pi \times r^2/2a$, 因此至少有一个节点在节点 v_i 的传输区域内的概率为 $1 - (1 - \pi \times r^2/2a)^n$. 只要节点 v_i 的传输区域内有邻居节点, 则节点 v_i 就能够将数据发送给传输区域的邻居节点. 故在折半角度传输阶段, 节点 v_i 发送数据给传输区域的邻居节点的概率为 $1 - (1 - \pi \times r^2/2a)^n$. 通过以上分析得到当 n 充分大的时, 节点 v_i 能够以接近 1 的概率将数据发送给传输区域的邻居节点. 以此类推, 节点通过多跳的折半角度传输的方式能够以接近 1 的概率将数据发送到移动 Sink. 得证.

5 仿真实验与结果分析

本文所提出的算法 ROAR 主要关注于如何解决移动传感器网络中数据传输路由的选择和建立问题, 此外我们假设 MAC 层有可靠的包传输机制, 网络层的丢包和拥塞控制问题可以得到较好的解决. 由于 MATLAB 可以提供快速和可视化的原型系统构建和算法性能分析, 满足我们的实验仿真要求, 所以我们选择在 MATLAB 平台上进行仿真实验.

为了评估和分析算法 ROAR 的性能, 本节将在 MATLAB 平台对提出的算法 ROAR 与算法 SMITE 和算法 Sidewinder 进行一系列的仿真实验, 主要分两组进行实验: 第一组主要分析节点数量变化和节点最大移动速度变化对网络生命周期的影响; 第二组主要是对算法的执行时间和能量开销进行测试. 假设移动 Sink 和 n 个移动节点随机散布在面积 $a = 100 \times 100 \text{ m}^2$ 的正方形区域内, 每个节点的初始能量在 $[0.5 \text{ J}, 1 \text{ J}]$ 之间随机分配且移动节点和移动 Sink 从 $(0, v_{\max} \text{ m/s})$ 中随机选取一个速度在正方形区域内随机移动. 其中, $v_{\max}$ 是节点和 Sink 最大移动速度. 为使算法 ROAR 与算法 SMITE 和算法 Sidewinder 能够进行公平的对比, 节点采用固定发射功率. 以下各组实验所使用的实验参数如表 1 所示.

以下各组的实验结果均是算法执行 20 次后的平均结果.

5.1 网络生命周期

网络生命周期是评价网络性能的重要因素, 而均衡节点的能量法消耗有助于延迟网络的生命周期. 为了考察算法的可扩展性, 本小节将测试算法 ROAR、算法 SMITE 和算法 Sidewinder 在 2 种不同的网络场景下所获得的网络生命周期. 在第 1 种网络场景中, 节点最大移动速度为 3 m/s , 其他实验参数

表 1 实验参数
Table 1 Experimental parameters

Notation	Value	Description
a	$100 \times 100 \text{ m}^2$	The area of the network
r	25 m	The transmission radius of nodes
e	[0.5 J, 1 J]	The initial energy of nodes
d_s	512 bit	The size of data pack
h_s	4 hops	The band probability hops of Sink
E_t	100 nJ/bit	The energy consumption of sending data
E_r	50 nJ/bit	The energy consumption of receiving data

如表 1 所示, 分别测试传感器节点数量为 80, 90, 100, 110, 120 时, 各算法对网络生命周期的影响. 在第 2 中场景中, 节点的数量 $n=100$, 其他实验参数如表 1 所示, 分别测试节点的最大移动速度为 2, 4, 6, 8, 10 m/s 时, 各算法对网络生命周期的影响. 实验结果如图 4 所示.

从图 4(a) 可以看到, 各算法在场景 1 中所获得的网络生命周期. 算法 Sidewinder 获得的网络生命周期最小且随着节点数量的增大而不断减小. 算法 SMITE 的网络生命周期大于算法 Sidewinder, 小于算法 ROAR. 随着节点数量的增加, 算法 SMITE 和算法 ROAR 所获得的网络生命周期是先增后减. 主要的原因是各个算法的网络生命周期不仅与网络中的节点密度有关, 而且与选取节点转发数据的策略密切相关. 对于算法 Sidewinder, 节点是在面向 Sink 成 60 度角的区域内选取邻居节点转发数据且为确保数据能够被 Sink 接收, 当转发数据的节点在 Sink 的一跳范围内时, 节点采用广播的方式将数据发送两跳范围内的邻居节点. 这种数据转发机制在选取邻居节点转发数的时候没有考虑节点的剩余能量且节点发送的数据被多个节点转发, 使得能量低的节点仍有可能不断转发数据从而缩短网络生命周期. 对于算法 SMITE, 节点能够根据所预测的 Sink 所在区域动态计算角度传输区域并且选取角度传输区域内的邻居节点转发数据. 这样能够在较为合适的角度传输区域内选取邻居节点转发数据从而延长网络的生命周期. 对于算法 ROAR, 节点不仅在第二阶段中采用定向广播的方式发送数据, 使得节点发送数据时, 只有靠近 Sink 的邻居节点才能接收到数据, 而且在第三阶段中采用折半角度传输的方式发送数据, 节点根据选取目标节点的规则在相应的传输区域内选取了能量最高的邻居节点作为目标节点转发数据, 均衡了节点的能量消耗, 有效延长了网络生命周期.

从图 4(b) 可以看到, 在场景 2 中算法 ROAR 的网络生命周期最大, 算法 SMITE 的网络生命周期第二, 算法 Sidewinder 的网络生命周期最小. 随着节点最大移动速度的不断增大, 各个算法的网络生命周期出现了一定程度的波动. 主要有两方面的原因. 一方面, 各个算法的网络生命周期与选取节点转发数据的策略密切相关. 根据场景 1 中的分析, 算法 ROAR 能够在各阶段选取能量高的邻居节点转发数据, 均衡节点的能量消耗从而使得算法 ROAR 获得比算法 SMITE 和算法 Sidewinder 更大的网络生命周期. 另一方面, 各算法的网络生命周期与在传输区域内可被选为转发数据的节点数量密切相关且是成反比的关系. 当节点的最大移动速度变化时, 可被选为转发数据的节点数量可能增加, 也有可能减少. 这样就容易导致各算法的网络生命周期出现波动的情况.

5.2 算法执行时间和能量开销

本小节将在目标监视区域的面积 ($100 \text{ m} \times 100 \text{ m}$)、节点的通信半径 (25 m)、数据包的大小 (512 bit)、节点的最大移动速度 (5 m/s)、节点接收数据的能量消耗 (50 nJ/bit) 和节点发送数据的

图 4 网络生命周期对比

Figure 4 Network lifetime comparison of Sidewinder, SMITE and ROAR. (a) Scene 1; (b) scene 2

图 5 算法执行时间以及能量消耗的对比

Figure 5 Execution time and energy overhead comparison of Sidewinder, SMITE and ROAR. (a) Execution time and the number of nodes changed; (b) energy overhead and the number of nodes changed

能量消耗 (100 nJ/bit) 固定的条件下, 分别测试传感器节点数量为 80, 90, 100, 110, 120 时, 算法 Sidewinder、算法 SMITE 和算法 ROAR 收集一轮数据的所需要的执行时间和能量开销. 实验结果如图 5 所示.

从图 5(a) 可以看到, 无论是算法 Sidewinder、算法 SMITE 还是算法 ROAR, 它们的算法执行时间与节点的数量成正比, 并且算法 ROAR 获得了比算法 SMITE 和算法 Sidewinder 更低的算法执行时间. 主要有两个原因: 一是节点数量越多, 移动 Sink 要收集到全网节点发送的数据需要越多的时间; 二是算法 ROAR 采用了折半角度传输的方式发送数据, 节点转发数据时不必每次重新计算预测区域, 缩短数据转发所需时间, 从而缩短了算法的执行时间.

从图 5(b) 可以看到, 随着节点数量的增加, 算法 Sidewinder、算法 SMITE 和算法 ROAR 的能量开销也在不断增加, 并且算法 ROAR 获得了比算法 SMITE 和算法 Sidewinder 更低的能量开销. 主要有两个原因: 一是随着节点数量的增加, 节点不仅要发送数据给 Sink 且发送的数据很有可能被更多的节点转发从而增加网络中节点的能量消耗; 二是算法 ROAR 对算法 SMITE 中的角度传输进行

了改进, 即使在角度传输区域内没有邻居节点, 只要在方向传输区域内或非方向传输区域内有邻居节点, 收集器仍然能够在相应的传输区域内选取能量最高的邻居节点转发数据. 这样不仅可以减少转发数据的节点数量而且可以有效避免收集器的存储容量已满或到一时间轮结尾时, 如果在角度传输区域内没有邻居节点, 则需要将数据广播到所有邻居节点进行转发的情况, 减少消息的开销进而节省节点的能量消耗.

6 结论

在本文中, 我们针对 Sink 和节点随机移动所形成网络拓扑不断变化的 WSN 中的数据收集问题, 研究如何设计数据收集协议, 使得移动 Sink 能够快速、准确收集到全网节点感知的数据, 并且能够有效均衡各节点的能量消耗从而最大化网络生命周期. 我们通过对数据收集模型进行分析, 提出一种节能的算法 ROAR. 算法 ROAR 能够较好适用于网络拓扑不断变化的无线传感器网络, 有效控制数据收集消息和数据的传输方向, 均衡节点的能量消耗. 仿真实验表明, 算法 ROAR 有效缩短数据收集延迟、减少能量消耗、大幅度提高网络生命周期.

由于算法 ROAR 采用概率的方式从网络中选取部分节点作为收集器, 收集器的数量和分布并不是最优的. 因此, 下一步我们将研究如何优化算法 ROAR, 使得收集器的数量和分布更加合理, 并且我们将在更接近实际应用的硬件测试床或功能更强大的网络模拟工具 (如: NS3) 上测试我们的算法.

参考文献

- 1 Zhao M, Ma M, Yang Y Y. Mobile data gathering with space-division multiple access in wireless sensor networks. In: Proceedings of the IEEE International Conference on Computer Communications, Phoenix, 2008. 1957–1965
- 2 Ma M, Yang Y Y, Zhao M. Tour planning for mobile data-gathering mechanisms in wireless sensor networks. IEEE Trans Veh Technol, 2013, 62: 1472–1483
- 3 Shi G T, Liao M H. Movement-assisted data gathering scheme with load-balancing for sensor networks. J Softw, 2007, 18: 2235–2244 [石高涛, 廖明宏. 传感器网络中具有负载平衡的移动协助数据收集模式. 软件学报, 2007, 18: 2235–2244]
- 4 Kermarrec A M, Tan G. Greedy geographic routing in large-scale sensor networks: a minimum network decomposition approach. In: Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing, New York, 2010. 161–170
- 5 Cao Y, Sun Z L, Cruickshank H, et al. Approach-and-roam (AaR): a geographic routing scheme for delay/disruption tolerant networks. IEEE Trans Veh Technol, 2013, 63: 266–281
- 6 Biomo J M M, Kunz T, St-Hilaire M. Routing in unmanned aerial ad hoc networks: a recovery strategy for greedy geographic forwarding failure. In: Proceedings of IEEE Conference on Wireless Communications and Networking, Istanbul, 2014. 2236–2241
- 7 Jin X C, Zhang R, Sun J C, et al. TIGHT: a geographic routing protocol for cognitive radio mobile ad hoc networks. IEEE Trans Wirel Commun, 2014, 13: 4670–4681
- 8 Torrieri D, Talarico S, Valenti M C. Routing in unmanned aerial ad hoc networks: a recovery strategy for greedy geographic forwarding failure. In: Proceedings of IEEE Conference on Military Communications, Istanbul, 2014. 1079–1084
- 9 Thaalbi M, Tabbane N, Bejaoui T, et al. A weighted QoS aware multipath routing process in mobile ad hoc networks. In: Proceedings of IEEE Symposium on Computers and Communications, Split, 2013. 336–341
- 10 Duan Q, Virendra M, Upadhyaya S, et al. Minimum cost blocking problem in multi-path wireless routing protocols. IEEE Trans Comput, 2014, 63: 1765–1777
- 11 Zhang J, Ren F Y, Gao S, et al. Dynamic routing for data integrity and delay differentiated services in wireless sensor networks. IEEE Trans Mobile Comput, 2014, 1–14

- 12 Tang B, Ye B L, Lu S L. Latency-optimized broadcast in mobile ad hoc networks without node coordination. In: Proceedings of IEEE International Conference on Communications, Dresden, 2009. 1–6
- 13 Tang B, Ye B L, Lu S L. Latency-optimized broadcast in mobile ad hoc networks without node coordination. In: Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing, New York, 2014. 317–326
- 14 Mohsen G, Erez K, Nancy L, et al. Multi-message broadcast with abstract MAC layers and unreliable links. In: Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing, New York, 2014. 56–65
- 15 Nikhil B, Moses C, Ravishankar K, et al. Better algorithms and hardness for broadcast scheduling via a discrepancy approach. In: Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing, New York, 2014. 55–71
- 16 Wang S, Kim S M, Liu Y H, et al. CorLayer: a transparent link correlation layer for energy efficient broadcast. In: Proceedings of the 19th Annual International Conference on Mobile Computing and Networking, New York, 2014. 51–62
- 17 Ma M, Ma Y Y. SenCar: an energy-efficient data gathering mechanism for large-scale multi-hop sensor networks. *IEEE Trans Parall Distr Syst*, 2007, 18: 1476–1488
- 18 Xu X, Luo J, Zhang Q. Delay tolerant event collection in sensor networks with mobile sink. In: Proceedings of the IEEE International Conference on Computer Communications, San Diego, 2010. 1–9
- 19 Chowdhury S, Giri C. Data collection point based mobile data gathering scheme with relay hop constraint. In: Proceedings of the IEEE International Conference on Advances in Computing, Communications, Mysore, 2013. 282–287
- 20 Zhao M, Yang Y Y. Bounded relay hop mobile data gathering in wireless sensor networks. *IEEE Trans Comput*, 2012, 61: 265–277
- 21 Keally M, Zhou G, Xing G L. Sidewinder: a predictive data forwarding protocol for mobile wireless sensor networks. In: Proceedings of the IEEE International Conference on Sensor, Mesh and Ad Hoc Communications and Networks, Rome, 2009. 1–9
- 22 Guo L J, Beyah R, Li Y S. SMITE: a stochastic compressive data collection protocol for mobile wireless sensor networks. In: Proceedings of the IEEE International Conference on Computer Communications, Shanghai, 2011. 1611–1619
- 23 Alomari A, Aslam N, Comeau F. Data collection using rendezvous points and mobile actor in wireless sensor networks. In: Proceedings of the IEEE International Conference on Communications, Ottawa, 2012. 7115–7119
- 24 Donghyun K, Abay B H, Uma R N, et al. Minimizing data collection latency in wireless sensor network with multiple mobile elements. In: Proceedings of the IEEE International Conference on Computer Communications, Orlando, 2012. 504–512
- 25 Buragohain C, Agrawal D, Suri S. Power aware routing for sensor databases. In: Proceedings of the IEEE International Conference on Computer Communications, Piscataway, 2005. 1747–1757
- 26 Wu Y, Fahmy S, Shroff N B. On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm. In: Proceedings of the IEEE International Conference on Computer Communications, Phoenix, 2008. 1013–1021
- 27 Liang W F, Liu Y Z. Online data gathering for maximizing network lifetime in sensor networks. *IEEE Trans Mobile Comput*, 2007, 6: 2–11
- 28 Li J Z, Shatz S M, Kshemkalyani A D. Mobile sampling of sensor field data using controlled broadcast. *IEEE Trans Mobile Comput*, 2011, 10: 881–896
- 29 Liang J B, Li T S. An adaptive probability broadcast-based data preservation in wireless sensor networks. *J Comput Res Dev*, 2012, 49: 2229–2240 [梁俊斌, 李陶深. 无线传感网中基于自适应概率广播的数据保存. *计算机研究与发展*, 2012, 49: 2229–2240]
- 30 Heinzelman W R, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the Annual Hawaii International Conference on System Sciences, Maui, 2000. 3005–3014

Data collection based on orientation angle routing in mobile sensor networks

LIANG JunBin, ZOU ShaoJun* & LI TaoShen

School of Computer and Electronic Information, Guangxi University, Nanning 530004, China

*E-mail: zousj2014@163.com

Abstract In a multi-hop wireless sensor network in which the Sink and all nodes are constantly moving, an energy-efficient and latency-minimized algorithm ROAR is proposed. Firstly, all nodes perform a collector selection algorithm, which is similar to cluster-head selection algorithms, such that nodes with higher energy have a higher probability of becoming collectors. Then, each node sends its data to its neighbors in an effective zone. This zone is determined by drawing a horizontal line between the node and the Sink, and then a vertical line through the middle of this horizontal line; the effective zone is the field that locates in the same side as the Sink. Finally, each collector uses a binary angle transmission mechanism to send its data to the mobile Sink. Simulation results show that, compared with existing algorithms, ROAR can effectively control the direction of data transmission so as to minimize the data collection delay and balance the energy consumption of nodes, thus achieve a longer network lifetime. As a result, ROAR has better performance than existing algorithms, and is suitable for use in mobile wireless sensor networks with frequently changing topologies.

Keywords wireless sensor network, data collection, routing, angle transmission, maximum the network lifetime

LIANG JunBin was born in 1979. He received the Ph.D. degree in computer application technology from the Central South University, Changsha, in 2010. Currently, he is an associate professor at Guangxi University. His research interests include wireless sensor networks and network optimization theory.

ZOU ShaoJun was born in 1987. He received the B.E. degree in computer science and technology from Minnan Normal University, Zhangzhou, in 2012. Currently, he is a postgraduate student in Guangxi University. His research interest is wireless sensor networks.

LI TaoShen was born in 1957. He received the Ph.D. degree in computer application technology from the Central South University, Changsha, in 2008. Currently, he is a professor at Guangxi University. His research interests include wireless sensor networks and wireless mesh networks.