

# 无线传感器网络冗余节点休眠调度算法

温涛<sup>1,2</sup>, 张冬青<sup>1,2</sup>, 郭权<sup>2</sup>, 宋晓莹<sup>1</sup>

(1. 东北大学 软件中心, 辽宁 沈阳 110819; 2. 大连东软信息学院 计算机科学与技术系, 辽宁 大连 116023)

**摘要:** 提出一种冗余节点休眠调度算法来延长网络生命周期。调度过程中重点考虑两方面问题: 一是采取策略防止大量节点同时从工作状态转入休眠状态以防止大量盲区同时产生; 二是根据邻居表中节点的工作邻居数量, 判定节点是否处于网络边界, 对边界节点和内部节点采用不同的调度策略, 防止边界收缩。仿真结果表明, 算法能有效延长无线传感器网络的生命周期。

**关键词:** 无线传感器网络; 冗余节点休眠; 监测盲区; 网络边界收缩; 网络生命周期

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2014)10-0067-14

## Wireless sensor network redundancy node sleep scheduling algorithm

WEN Tao<sup>1,2</sup>, ZHANG Dong-qing<sup>1,2</sup>, GUO Quan<sup>2</sup>, SONG Xiao-ying<sup>1</sup>

(1. Software Center, Northeastern University, Shenyang 110819, China;

2. Department of Computer Science and Technology, Dalian Neusoft University of Information, Dalian 116023, China)

**Abstract:** A scheduling algorithm for redundant nodes is proposed to effectively prolong the lifetime of WSN. The algorithm focuses on two problems. One is to prevent a large number of nodes shifting at the same time from working state into sleeping state to avoid the appearance of blind areas, the other is to employ different scheduling strategies on border nodes and internal nodes to avoid contraction of the network boundary by judging whether a node is on the network boundary according to the number of working nodes in its neighbor table. The simulation results show that the algorithm can effectively prolong the lifetime of WSN.

**Key words:** wireless sensor network; redundant nodes sleep; monitoring blind area; network boundary contraction; network lifecycle

## 1 引言

近些年, 随着物联网的提出和越来越广泛的应用, 无线传感器作为物联网的基础设施在军事和民用领域也得到越来越广泛的应用, 在深空探测、深海潜航、环境监测和保护、湿地监控、河流湖泊水文、城市规划建设等方面大量部署, 在远程监控、智能城市、智能家居、智能办公等多个领域体现出越来越高的实用价值<sup>[1]</sup>。

由于无线传感器网络的特殊应用领域, 使很多情况下传感器节点是一次性部署的, 不再回收, 在

整个工作过程中能量无法补充, 直到网络中所有节点的能量耗光, 整个网络失效。有线网络中的经典算法以及一些能量不受限制或者可以中途进行能量补充的无线网络中的算法都无法直接应用于能量受限的无线传感器网络中。对于能量受限的网络必须尽一切可能节省能量才能使生命周期延长。所以节能问题研究一直是无线传感器网络的一个重要研究分支。

目前, 无线传感器网络的节能问题主要从网络级和节点级 2 个层次进行研究<sup>[2]</sup>。在网络级可以设计良好的分簇算法和路由协议, 使数据在转发传递

收稿日期: 2013-06-08; 修回日期: 2013-10-13

基金项目: 国家自然科学基金资助项目(61170169, 61170168)

**Foundation Item:** The National Natural Science Foundation of China(61170169, 61170168)

时能耗降低,从而延长网络生命周期,而在节点级可以从开源和节流两方面进行,开源方面通过对电子元器件的研究使无线传感器电源的续航时间延长或者通过太阳能等方式进行中途充电,节流方面通过对节点的休眠调度算法来延长整个网络的生命周期。

在网络级,Stojmenovic<sup>[3]</sup>给出了已知节点位置信息的 Ad Hoc 网络的路由算法;Lochert<sup>[4]</sup>给出了移动节点组成的 Ad Hoc 网络的路由算法;Sunil<sup>[5]</sup>等提出了采用聚类分析的方法把事件的数据分组进行聚类,然后进行数据融合,从而减小网络上的开销;Saad<sup>[6]</sup>等提出了一种新型的体系结构,通过分布式自适应聚类算法来延长网络生命周期,并提高网络覆盖率,这种结构中起到中心节点作用的那些节点容易因为计算量增大而成为整个网络的瓶颈。在节点级,通过延长电池本身寿命或者对电池进行多次充电的方法不在本文研究范围之内,本文只研究如何通过调度算法来节省能量。节点级的节能调度算法也可以分成 2 类:一是知道每个节点的精确位置信息的情况,二是不知道节点的精确位置的情况。Tian 和 Georganas<sup>[7,8]</sup>提出了非执勤资格规则(off-duty eligibility rule),根据每个节点位置以及信号到达的角度,可以计算节点之间的覆盖关系,并且在文献[8]中对 NS-2 中的 LEACH 协议进行了扩展,对冗余节点进行休眠调度;Wang<sup>[9]</sup>等提出了覆盖配置协议,可以由应用提供不同程度的节点覆盖要求;Mihaela 和 Jie<sup>[10]</sup>对各种静态部署的无线传感器网络节点覆盖率进行了综述性研究和总结。大多数应用中无线传感器是随机部署的,需要配备定位模块才能获得每个节点的位置,这一方面增加了传感器成本,另一方面位置信息也受制于定位模块的精度。

因此,很多学者着力于研究节点精确位置未知的算法。Wan<sup>[11]</sup>等对于方形网络区域的节点服从泊松分布的情况提出了随机部署算法,能够应对复杂的边界情形;Liu<sup>[12]</sup>等提出了随机休眠算法,可以在不进行节点间通信的情况下实现休眠调度,但该算法由于随机性难以保证网络的覆盖率和覆盖均匀性,只适用于对监测覆盖率要求较低的场合;Lazos<sup>[13]</sup>等研究了平面异构传感器网络覆盖问题,给出了随机覆盖的解析表达式;Zou<sup>[14]</sup>等研究了基于集群的分布式传感器网络的部署策略,提出了虚拟强化算法,尝试最大限度地提高传感器的覆盖范

围;Tsai<sup>[15]</sup>等研究了传播路径中有障碍物的传感器网络中的阴影效果;Jiang<sup>[16]</sup>等提出了一个轻量级分布式随机部署的节点调度算法,让不相交的节点集交替工作,有效地延长了网络生命周期;Fan<sup>[17]</sup>等通过数学方式给出了传感器节点数量与所期望的部署质量之间的函数关系;Hu<sup>[18]</sup>等提出了一种融合遗传算法的安排转型操作的混合型算法,利用遗传算法的特性控制各个节点的演化,该算法节点间需要较多额外的通信量;Wu<sup>[19]</sup>等提出了一种轻量级部署调度算法,称为 LDAS,算法先计算一个节点被它的  $n$  个邻居节点完全覆盖的概率,随后计算一个节点被它的  $n$  个邻居覆盖的区域占自身监测区域的百分比,基于这 2 个结果设计了一种称为邻居表的数据结构,记录邻居节点的必要信息,然后在邻居节点之间定期传递邻居表消息,通过邻居表中的特定字段可以判断一个节点是否应休眠,该算法所需的额外通信量较小,而且能够保证较高的网络覆盖率,但没有区分网络边界节点和网络内部节点,运行后期会因为处于边界的节点较少获得休眠机会而首先耗尽能量,出现边界收缩现象;凡高娟<sup>[20]</sup>等在此基础上提出了基于容忍覆盖区域的节点调度算法,考虑了边界节点的特殊性,在进行调度之前首先判断节点是否为覆盖冗余节点,此时会通过一个积分表达式计算节点是否处于容忍覆盖区域(即网络的边界),该算法虽然在一定程度上缓解了边界收缩现象,但通过对不被所有邻居覆盖概率的积分来求解不被邻居集覆盖面积的数学期望的方法缺乏理论依据,并不严谨,另外在判定过程中需要用到节点间距离等信息,所需通信量也较大。

本文在文献[19]的基础上,分析了处于网络内部的节点与处于网络边界的节点所包含的邻居数量和覆盖率的不同,针对网络内部和边界的节点进行不同的调度,提出了一种冗余节点休眠调度算法(RNSSA, redundant nodes sleep scheduling algorithm),并在此基础上引出 2 个子算法:防止大量节点同时休眠算法和边界节点判定算法,用于解决 RNSSA 算法中的关键问题。

## 2 网络模型

本文研究的无线传感器网络是随机部署的高密度静态传感器网络,整个网络需要满足以下要求:

- 1) 节点部署有随机性, 事先不规定每个节点部署的位置;
- 2) 所有节点一经部署, 其位置就不再改变;
- 3) 所有节点是同质节点, 即各种规格参数都相同, 尤其是感知半径和通信半径均相同, 但不局限于特定的某种传感器节点;
- 4) 节点可以感知到所有位于其感知半径以内的处于工作状态 (见定义 3) 的节点;
- 5) 每个节点都有唯一的标识 ID, 彼此相互感知的节点可以获得对方的 ID;
- 6) 节点通信半径不小于感知半径;
- 7) 被监测网络区域不需要始终达到 100% 的监测覆盖率, 监测过程中允许出现少量区域没有被监测到的情况发生;
- 8) 节点部署的密度高, 平均每个节点的邻居 (见下文的定义 1) 数量不低于 11 个 (取该值的依据参考文献[19]);
- 9) 节点不知道自己的精确位置, 也不知道邻居的精确位置;
- 10) 整个网络区域是一个边缘没有特殊凸起或凹陷的多边形、圆或椭圆 (即边界没有特别尖锐的角);
- 11) 整个网络区域范围远大于节点的感知范围, 网络区域面积至少为一个节点感知区域面积的 100 倍。

满足以上要求的无线传感器网络有一定的实际应用价值, 如目前正在研究一种作战方案, 在进入战场之前由无人机先在战场区域部署大量传感器节点, 对战场情况进行数据采集和存储, 但并不需要立刻汇总并发送到指挥部, 而是等步兵或坦克等军种进入该区域后再收集这些节点<sup>[21]</sup>。此外, 这种无线传感器网络的部署方式还可以应用于一些不需要实时收集数据的监测场合。

### 3 相关定义

**定义 1** 邻居节点。部署在一个节点周围, 处于该节点感知半径范围内的节点, 称为该节点的邻居节点, 可表示为

$$N(i) = \{j \text{ 为节点} | d(i, j) \leq R, j \neq i\} \quad (1)$$

其中,  $R$  为节点  $i$  的感知半径,  $d(i, j)$  表示节点  $i$  和节点  $j$  之间的距离。

显然, 节点与其邻居节点之间的通信是一跳可以到达的, 并且, 若  $A$  是  $B$  的邻居, 则  $B$  也是  $A$

的邻居。

**定义 2** 完全冗余节点、部分冗余节点和孤立节点。设  $C_i$  为节点  $i$  的感知区域, 如果  $\bigcup_{j \in N(i)} C_j \supseteq C_i$ , 则称  $i$  为完全冗余节点, 换句话说, 如果某个节点能够被它的所有邻居节点完全覆盖住, 则是完全冗余节点。显然, 由于  $i$  已经被完全覆盖住了, 该节点的工作完全可以由其他那些覆盖住它的节点来代替进行, 该节点就可以休眠从而保存能量。

如果  $\bigcup_{j \in N(i)} C_j \cap C_i \neq \emptyset$  并且  $C_i \not\subset \bigcup_{j \in N(i)} C_j$ , 则称  $i$  为部分冗余节点, 即只能覆盖的一部分, 则是部分冗余节点。

如果节点  $i$  没有任何邻居, 则称节点  $i$  为孤立节点。对于孤立节点, 它无法感知到其他节点, 其他节点也无法感知到它, 也就无法与整个网络协同工作, 所以对于孤立节点本文不予研究。

**定义 3** 工作节点和休眠节点。在无线传感器网络中, 每个节点都可以处于工作状态或者休眠状态, 处于工作状态的节点可以正常进行感知、监测、通信等各种工作, 这种节点称为工作节点。处于休眠状态的节点以极低的功耗在运行, 不对周边进行感知和监测, 也不与其他节点进行通信, 这种节点称为休眠节点。工作节点和休眠节点可以相互转换, 当工作节点满足一定条件时, 可以进入休眠期, 而休眠节点经过一定的时间会被唤醒, 进入工作状态。

**定义 4** 无线传感器网络的服务质量(QoS)。本文将整个无线传感器网络在某个时刻的服务质量定义为该时刻所有工作节点对网络的实际监测的覆盖面积占整个网络的监测区域面积的百分比, 即

$$QoS(t) = \frac{W \cap \sum_{i=1}^k W_i}{W} \quad (2)$$

其中,  $k$  表示在时刻  $t$  处于工作状态的节点数量,  $W_i$  表示第  $i$  个节点的监测区域,  $W$  表示整个监测网络区域。对于大多数无线传感器网络来说, 并不需要在任何时刻  $QoS$  都达到 100%, 在网络中存在一些监控的盲点是允许的, 只要不低于特定阈值即可。本文讨论的算法允许存在监控的盲点, 但这些点不是一成不变的, 而是随着时间的推移在不断变化, 这个时刻的盲点到了下一个时刻可能就会被监测到 (因为不同的节点在不同的时期休眠), 反之这个时刻可以被监测的点, 下一个时刻有可能变成盲点。对于始终要求  $QoS$  达到 100% 的监控网络, 本

文的算法并不适用。

每个无线传感器网络都有其覆盖范围，总有一些节点位于整个网络的边界处，对于处于边界的节点不能与处于网络中心的节点同等对待，因为很显然位于边界的节点的邻居数量要少一些。所以对于边界节点要进行专门的考察。

**定义 5** 边界节点和网络内部节点。边界节点和网络内部节点。根据网络模型假设 10)，整个网络区域是一个有边界的封闭图形。定义感知区域经过整个无线传感器网络的边界的节点称为边界节点。显然边界节点有一个共同的特征，某一侧（网络外部）没有邻居节点，而另一侧（网络内部）有大量邻居节点。

不处于网络边界的节点称为网络内部节点。

**定义 6** 边界节点的内部和外部。根据网络模型假设 11)，网络覆盖范围远大于每个节点的感知范围，为研究方便，可以设网络边界通过边界节点监测区域的圆心，或者说把网络边界定义在通过边界节点圆心的线上。由于网络边界很大，则对于每个处于边界的节点来说，边界都可以近似的看成一条直线，也就是一条直径，从该直径划分，属于网络覆盖范围的那一半称为边界节点的内部，另外一半称为边界节点的外部。对于边界节点的内部，与普通节点一样，有自己的邻居节点，也会被足够多的邻居覆盖，而对于边界节点的外部这个半圆，邻居的数量和被覆盖范围都会很小甚至为 0。所以对于边界节点，要重点研究其内部的情况。

在本文的后续部分所有针对边界节点的研究均为邻居处于边界节点内部的情况。

**定义 7** 交点、相交扇形与有效角。如果节点处于网络内部，如图 1 所示，节点  $X_1$  是节点  $O$  的邻居， $P_1^1$  和  $P_1^2$  是节点  $X_1$  与  $O$  的 2 个交点，相交扇形为 2 个交点所夹内部弧与圆心  $O$  构成的扇形，即扇形  $P_1^1OP_1^2$ ，而有效角为 2 个交点与圆心  $O$  的夹角，即  $\angle P_1^1OP_1^2$ 。如果某个节点处于网络边界，分为 2 种情况：一是该节点与某个邻居的 2 个交点都处于网络内部，这时几个定义与处于网络内部的相同；二是该节点与某个邻居的 2 个交点中只有一个处于网络内部，此时相交扇形为处于边界节点内部的交点与边界半径所夹生成的扇形，而该扇形的圆心角为有效角，如图 2 所示， $O$  是位于网络边界的节点， $MN$  是网络边界，其右上方为该边界节点的内部，左下方为外部， $P_1^1$  和  $P_1^2$  是节点  $X_1$  与  $O$  的 2 个交点，

其中，只有  $P_1^2$  位于边界节点内部，此时相交扇形即为  $P_1^2ON$ ，而有效角为  $\angle P_1^2ON$ 。

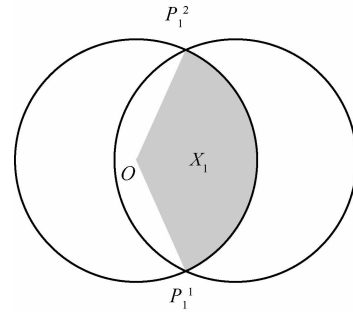


图 1 处于网络内部的节点相关定义

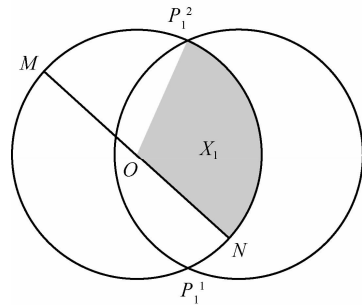


图 2 处于网络边界的节点相关定义

**定义 8** 邻居表。保存在每个节点中的一种数据结构，其结构如表 1 所示，其中每一行记录该节点的一个邻居节点的信息。每一行中的 ID 表示邻居节点的唯一标识；Mode 表示节点状态，可以取 3 种值：W 代表工作状态，P 代表预休眠状态，S 代表休眠状态；NoWN(number of working neighbors)，表示该邻居能够感知到的处于工作状态邻居的数量；UpdTime 表示该条信息更新的时间。

表 1		邻居表	
ID	Mode	NoWN	UpdTime/ms
1	W	4	15 283 056 840
2	W	5	15 283 445 090
3	P	6	15 283 356 400
4	S	5	15 283 648 320

**定义 9** 信标。节点以广播的形式向其邻居发送的一种特定格式的消息，每个信标消息包含 2 部分信息：节点自己的 ID，自己的邻居中处于工作状态的节点数量。

每个节点通过向外发送信标消息来标识自己的存在，当一个节点收到来自于邻居的信标消息，则根据其中的内容更新自己的邻居表。每个节点都

会定期检查自己的邻居表, 如果很长时间(可以设定一个阈值)没有收到某个邻居节点发送过来的信标消息, 则认为该邻居处于休眠状态, 将其在邻居表中的状态更新为 S, 并重置更新时间。

**定义 10** 选票。节点向其邻居发送的一种特定格式的消息, 每个选票消息包含 3 个字段: 发送消息的节点 ID、接收消息的节点 ID、选票有效时间。其中的选票有效时间的单位也是 ms, 可以用于与邻居表中的 *UpdTime* 进行比较。每个选票的生命周期(选票生成时间与选票有效时间之间的时间差)是相同的。

每个节点都可以接收到来自于邻居的选票, 但只有当收到的选票中的接收消息节点 ID 与自身匹配时, 才会保留该选票, 其他时候均丢弃该选票, 从这个层面可以认为选票消息与信标消息的传播方式不同, 是点对点的。

## 4 算法描述

整个算法分为 3 部分, 一是冗余节点休眠调度算法; 二是防止大量节点同时休眠调度子算法; 三是边界节点判定子算法。

**算法 1** 冗余节点休眠调度算法。每个节点有 4 套并发的工作机制。

1) 如果节点处于工作状态或者预休眠状态, 则定期向邻居发送信标消息来标识自身的存在, 如果节点处于休眠状态则不发送信标消息。

2) 接收模块会随时收到来自不同邻居的信标消息, 收到之后就根据信标消息中的内容来更新自身的邻居表。

3) 定期对自身的邻居表做 2 项检查, 一是如果发现邻居表中某条记录长时间没有更新(与预先设定的阈值比较), 则认为该邻居处于休眠状态, 将邻居表中的 Mode 值改为 S, 并重置其更新时间; 二是如果邻居表中工作邻居数量  $m$  超过阈值  $T$ , 则随机向其中  $m-T$  个处于工作状态的邻居节点发送选票。

4) 定期检查自身收到选票的数量, 对边界节点和内部节点分别判断。对边界节点要比较其收到的有效选票数量与边界邻居数量阈值的大小, 对内部节点则要比较其收到的有效选票数量与内部邻居数量阈值的大小, 无论边界节点还是内部节点, 如果超过各自的阈值, 则节点会进入预休眠状态。预休眠状态可能转入休眠状态, 也可能恢复工作状态, 具体调度见算法 2。

网络中的所有节点都按照上述并发的工作机制运行, 就构成了冗余节点的休眠调度算法。

该调度算法中有以下几个关键问题。

1) 邻居表的更新时间与选票有效期限等与时间相关的信息, 如果像文献[19]那样以时分秒的形式记录时间, 遇到 2 个不同日期中的时间进行比较时就容易出错, 即使把日期考虑进来也会增加比较的复杂度, 如果以年月日时分秒的形式记录时间, 占用的存储空间较大, 故本文采用长整型数据存储时间, 既不会占用大量存储空间, 又可以高效地进行比较。

2) 当某个节点获得的有效选票数量超过阈值后, 转入预休眠状态而不是直接进入休眠状态, 原因是防止出现大量节点同时进入休眠状态而导致网络出现过多盲区, 因为整个网络中所有节点部署好之后同时开始运行, 算法对于所有节点都有效, 每个节点检测选票是否达到阈值的时间间隔也都相同, 这样显然会出现大批量节点的选票都超过阈值的情况, 如果都进入休眠状态, 就会产生过多盲区。对此提出“防止大量节点同时休眠算法”, 即后续算法 2。

3) 边界节点的判定

处于边界的节点和处于网络内部的节点, 其邻居数量、被邻居节点完全覆盖的概率以及被邻居节点覆盖住的百分比都有所不同。注意到处于边界的节点的邻居中, 至少有 2 个也同样处于网络的边界, 可以据此给出“边界节点判定算法”, 即算法 3, 该算法给出了在不知道每个节点具体位置的前提下, 判断节点是否处于网络边界的方法。

**算法 2** 防止大量节点同时休眠算法。有 2 种方式可以防止大量节点同时休眠, 第一种算法为: 进入预休眠状态的节点等待一个介于 0 和指定阈值  $T_{max}$  之间的随机时间, 重新检测有效选票的数量, 如果有效选票数量超过阈值, 则转入休眠状态, 如果有效选票数量没有达到阈值, 则转入工作状态。第二种算法为: 取消预休眠状态, 只保留工作和休眠两种状态进行相互转换, 在算法 1 的第 3) 套工作机制中的第二项检查, 即检查邻居表中工作邻居数量  $m$  是否超过阈值  $T$ , 若超过, 则随机向其中  $m-T$  个处于工作状态的邻居节点发送选票, 这个检查和发送选票的周期不再是固定值, 而是一个介于某个范围的随机值。

算法中另一个关键问题是如何判定节点是否处于网络边界, 处于网络边界的节点与处于网络内

部的节点最明显的区别在于：网络内部节点的周边有其他邻居节点，而且各个方向的邻居数量和密度基本相同（因为是随机均匀分布的）；而网络边界的节点只有处于内部的半边有邻居节点，处于网络外部的另外半边是没有邻居的，所以其邻居数量要少于网络内部节点。

但是仅凭节点的邻居数量并不足以确定节点是否处于边界，因为随着网络的工作时间延长，部分节点因能量耗尽而死亡，整个网络节点的密度降低，内部节点的邻居数量也会减少。所以还要在此基础上进一步分析边界节点的特性。如图 3 所示，节点 *A* 处于网络边界，则它一定至少有 2 个也处于边界的邻居节点 *B* 和 *C*，分别位于 *A* 的两侧，此外还有一些其他的邻居节点位于网络的内部，如 *D*、*E*、*F* 等。每个处于网络边界的节点都会至少有 2 个同样处于网络边界的邻居，这两个邻居的邻居节点数量也会比网络内部节点的邻居数量少，在邻居表中的 *NoWN* 项恰好记录了邻居节点中活动的邻居数量。根据以上这些特性，可以得到：如果一个节点（如图 3 中的 *A* 节点）的邻居数量与它其中 2 个邻居节点的邻居数量相当（这 2 个邻居也处于网络边界，如图 3 中的 *B*、*C* 节点），同时它的邻居总数量明显少于其他大部分邻居节点的邻居数量（其他邻居节点都处于网络内部，如图 3 中的 *D*、*E*、*F* 等节点），那么该节点自身就处于网络边界。其定量的描述见算法 3。

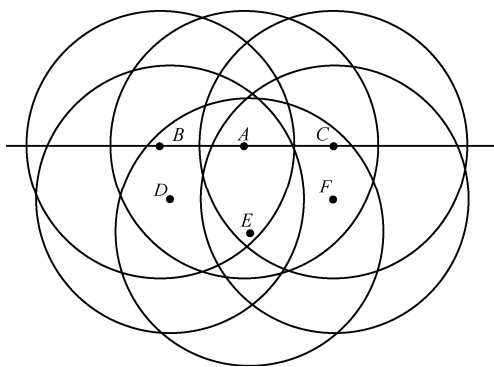


图 3 边界节点示意

**算法 3** 边界节点判定算法。某节点有 *n* 个邻居处于工作状态，且  $n \geq N$  (*N* 是使网络能够处于正常工作状态的阈值，后续的定理会计算 *N* 的值)，若 *n* 个邻居中至少有 2 个的 *NoWN* 项的值约等于 *n*（上下浮动不超过 1），其他邻居的 *NoWN* 项的值大于  $n+2$ ，则该节点处于网络边界。

算法的核心问题是求出算法 1 中的邻居数量阈值 *T* 以及处于边界的节点被邻居完全覆盖的可能性达到一定要求的阈值 *N*，在网络内部的节点邻居数量阈值 *T* 已经在文献[19]中计算得到，本文后续计算处于边界的节点所需的阈值，其求解过程详见下面的定理。

**定理 1** 当只有一个交点位于边界节点内部时，2 个邻居的有效角取值范围为  $(\pi/3, \pi)$ ；当节点位于网络内部，或者 2 个交点都在边界节点内部时，有效角取值范围为  $[2\pi/3, \pi)$ 。

**证明** 对于第一种情况，其上界趋近于  $\pi$  的情况与文献[19]的证明中的情况相同，不再赘述。如果邻居节点恰好位于边界上，可以计算出该邻居与边界节点相交后所构成的扇形，其圆心角至少为  $2\pi/3$ ，而边界节点内部所生成的夹角至少为  $\pi/3$ 。

而对于第二种情况，文献[19]中已经给出证明。综上定理 1 得证。

在文献[19]中已经给出了网络内部节点邻居之间的全覆盖概率以及覆盖百分比关系，本文不再赘述。本文重点针对边界节点的邻居覆盖关系提出定理 2 和定理 3。

**定理 2** 边界节点的内部被该节点的 *n* 个邻居完全覆盖的概率 *P*(*A*) 为

$$1 - \left( \frac{2n}{3} \left( \frac{2\sqrt{3}\pi - 5}{3\pi^2} \right)^{n-1} + \frac{n}{3} \left( \frac{5\pi - 3\sqrt{3}}{6\pi} \right)^{n-1} \right) \quad (3)$$

为了证明该定理，先证明相关的几个引理。设节点 *O* 为边界节点，*C* 为该节点的内部感知区域，*C<sub>i</sub>* 为节点 *i* 的感知区域，*C<sub>i</sub>* 与 *C* 的相交区域包括 2 个交点，这 2 个交点可能有一个，也可能有 2 个会位于边界节点 *O* 的内部。如果只有一个交点位于内部，则称该交点为第一交点，如果有 2 个交点位于内部，则称在 *O* 的内部逆时针方向上首先遇到的那个交点为第一交点。

设 *A<sub>i</sub>* 表示的事件为：*C<sub>i</sub>* 与 *C* 相交的第一交点没有被 *O* 的其他邻居节点的感知区域 *C<sub>j</sub>* 覆盖。则与 *A<sub>i</sub>* 相反的  $\overline{A_i}$  表示的事件为：任何一个 *C<sub>i</sub>* 与 *C* 相交的第一个交点均会被 *O* 的其他邻居节点的感知区域 *C<sub>j</sub>* 覆盖，但各个 *C<sub>j</sub>* 不必相同。又设 *A* 表示的事件为：*C* 被 *C<sub>i</sub>* 完全覆盖。则事件 *A* 与事件  $\overline{A_i}$  之间的关系如下。

**引理 1**  $A = \bigcap_{i=1}^n \overline{A_i}$

**证明** 先证明  $A \Rightarrow \bigcap_{i=1}^n \overline{A_i}$ ，如果  $A$  成立，即  $C$  被  $C_i$  完全覆盖，则考虑  $C$  与  $C_i$  相交的 2 个交点中至少有一个在内部感知区域的情况，对于 2 个交点都在外部感知区域的情况不予考虑，因为其对于本命题没有影响。交点在  $C$  的内部感知区域的情况，该点必然还会被其他的  $C_j$  覆盖，这可以通过反证法来证明，假设存在某一  $C$  与  $C_i$  的交点，不被任何  $C_j$  覆盖，则必然可以在该点附近找到相邻的一点，既不在  $C_i$  中，也不被任何  $C_j$  覆盖(因为圆在二维空间中是连续的)，则该点没有被覆盖到，与已知条件  $C$  被  $C_i$  完全覆盖矛盾。

再证明  $\bigcap_{i=1}^n \overline{A_i} \Rightarrow A$ ，每一个  $C_i$  与  $C$  的交点均被其他某个邻居节点的感知区域  $C_j$  覆盖，那么圆周上的每一点到圆心  $C$  的连线也必然被该  $C_j$  覆盖，所以只需要考虑  $C$  的圆周即可。由于  $C$  所有的邻居都满足此条件，则  $C$  要么被完全覆盖，要么邻居数量为零，是一个孤立节点。用反证法，假设  $C$  的邻居数量不为零，并且没有被完全覆盖，考虑到圆周的连续性，没有被覆盖到的部分与某一个邻居是相连接的，则在该邻接点上就不满足已知条件，产生矛盾。

综上，引理 1 得证。

**引理 2**

$$P(A_1) = \frac{2}{3} \left( \frac{2\sqrt{3}\pi - 5}{3\pi^2} \right)^{n-1} + \frac{1}{3} \left( \frac{5\pi - 3\sqrt{3}}{6\pi} \right)^{n-1} \quad (4)$$

**证明** 如图 4 所示，设  $O$  为处于边界的节点， $L_1L_2$  为网络边界，其上方为网络外部，下方为网络内部，该边界通过  $O$  的圆心，该边界节点有一个邻居与之交于  $P_1^1$  和  $P_1^2$ ，现固定圆  $O$  不动，使该邻居

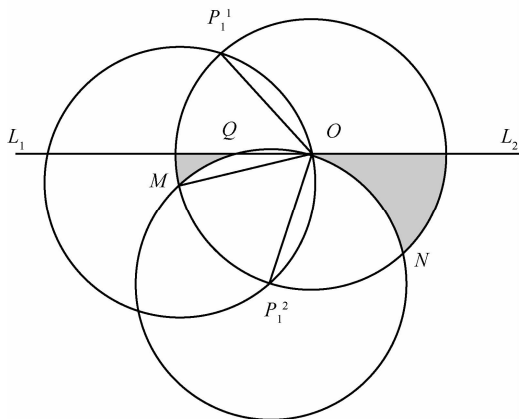


图 4 交点分别位于网络边界内外示意

沿着  $O$  的逆时针方向逐渐旋转，则  $P_1^1$  和  $P_1^2$  也沿着逆时针方向逐渐旋转，其旋转过程分为以下几个阶段。

第一阶段， $P_1^2$  处于边界节点的内部，而  $P_1^1$  处于边界节点的外部，即如图 4 所示的情形，此时的  $P(A_1)$  就是求另一个邻居不覆盖  $P_1^2$  的概率有多大，若要满足该条件，另一个邻居的圆心必须位于以  $P_1^1$  为圆心的圆之外，即图 4 中的阴影部分，否则另一个邻居一定会覆盖到  $P_1^2$ ，阴影部分的面积等于半圆面积减去 2 个互相过对方圆心的圆相交部分的面积，再加上圆  $P_1^2$  超过直线  $L_1L_2$  上方露出的那部分面积。设半径为  $R$ ，则半圆的面积为  $\pi R^2/2$ ，2 个互过对方圆心的圆相交部分的面积等于  $(2\pi - \sqrt{3})R^2/3$ ，而圆  $P_1^2$  在直线  $L_1L_2$  上方露出的那部分面积等于扇形  $P_1^2QO$  减去三角形  $P_1^2QO$  的面积，设  $\angle QOM = \alpha$ ，则可以计算得到  $\angle QP_1^2O = 2(\pi/6 - \alpha)$ ，从而扇形  $P_1^2OQ$  的面积为  $(\pi/6 - \alpha)R^2$ ，三角形  $P_1^2OQ$  的面积为  $(R^2/2) \sin(\pi/3 - 2\alpha)$ ，所以圆  $P_1^2$  在直线  $L_1L_2$  上方露出的那部分面积等于  $(\pi/6 - \alpha)R^2 - (R^2/2) \sin(\pi/3 - 2\alpha)$ ，其中， $\alpha$  的范围是从 0 到  $\pi/6$ ，故该阶段阴影部分的平均面积为

$$\frac{\int_0^{\pi/6} \left( \frac{\pi}{2} R^2 - \frac{2\pi - \sqrt{3}}{3} R^2 + \frac{\pi - 6\alpha}{6} R^2 - \frac{R^2}{2} \sin\left(\frac{\pi}{3} - 2\alpha\right) \right) d\alpha}{\int_0^{\pi/6} d\alpha} = \frac{2\sqrt{3}\pi - 5}{6\pi} R^2 \quad (5)$$

接下来，当  $\alpha$  的范围从  $\pi/6$  变到  $\pi/3$  时，情形与前一阶段类似，只是圆  $P_1^2$  在直线  $L_1L_2$  上方露出的那部分不在圆心  $O$  的左侧，而在右侧，其变化过程和面积与前一个阶段相同。

考虑  $\alpha$  的变化范围，从 0 到  $\pi/6$ ，又从  $\pi/6$  变到  $\pi/3$ ，都占了半圆圆心角的  $1/6$ ，故这个阶段

$$P(A_1) = \frac{n}{6} \left( \frac{2\sqrt{3}\pi - 5}{6\pi} \frac{R^2}{\frac{\pi}{2} R^2} \right)^{n-1} \times 2 = \frac{n}{3} \left( \frac{2\sqrt{3}\pi - 5}{3\pi^2} \right)^{n-1} \quad (6)$$

第二阶段， $P_1^1$  和  $P_1^2$  都处于边界节点的内部，如图 5 所示的情形，此时的  $P(A_1)$  也就是求另一个邻居不覆盖  $P_1^1$  的概率有多大，若要满足该条件，另一个邻居的圆心必须位于以  $P_1^1$  为圆心的圆之外，即

图 5 中的阴影部分(根据前面的假设, 不考虑邻居处于边界以外的情形), 此阴影部分的面积等于扇形  $L_1ON$  的面积加上  $ON$  所夹的小弓形的面积, 即

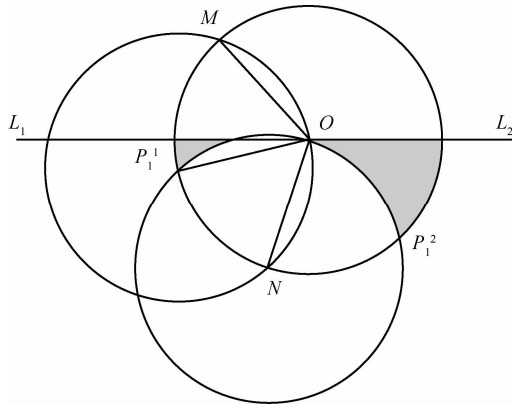


图 5 交点均位于网络内部示意

$$\frac{\int_{\pi/3}^{2\pi/3} \left( \frac{\alpha}{2\pi} \pi R^2 + \frac{2\pi - 3\sqrt{3}}{12} R^2 \right) d\alpha}{\int_{\pi/3}^{2\pi/3} d\alpha} = \frac{5\pi - 3\sqrt{3}}{12} R^2 \quad (7)$$

考虑  $\alpha$  的变化范围, 从  $\pi/3$  变到  $2\pi/3$ , 占了半圆圆心角的  $1/3$ , 故这个阶段

$$P(A_1) = \frac{n}{3} \left( \frac{5\pi - 3\sqrt{3}}{12} \frac{R^2}{\frac{\pi}{2} R^2} \right)^{n-1} = \frac{n}{3} \left( \frac{5\pi - 3\sqrt{3}}{6\pi} \right)^{n-1} \quad (8)$$

第三阶段,  $P_1^2$  处于边界节点的外部, 而  $P_1^1$  处于边界节点的内部, 即如图 6 所示的情形, 此时的  $P(A_1)$  就是求另一个邻居不覆盖  $P_1^1$  的概率有多大, 若要满足该条件, 另一个邻居的圆心必须位于以  $P_1^1$  为圆心的圆之外, 即图 6 中的阴影部分, 否则, 另一个邻居一定会覆盖到  $P_1^1$ , 阴影部分面积的计算方法与第一阶段的相同。其概率也为

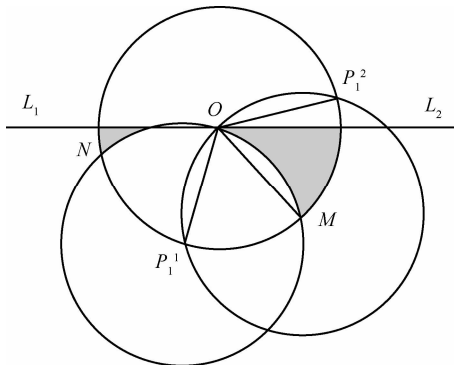


图 6 另一交点均位于网络外部示意

$$P(A_1) = \frac{n}{3} \left( \frac{2\sqrt{3}\pi - 5}{3\pi^2} \right)^{n-1} \quad (9)$$

综合以上 3 个阶段, 其总概率为各个阶段的概率和, 由式(6)、式(8)、式(9)最终得到

$$P(A_1) = \frac{2}{3} \left( \frac{2\sqrt{3}\pi - 5}{3\pi^2} \right)^{n-1} + \frac{1}{3} \left( \frac{5\pi - 3\sqrt{3}}{6\pi} \right)^{n-1} \quad (10)$$

**引理 3** 对于任意互不相等的  $i, j$ , 有  $P(A_i A_j) = 0$

**证明**  $P(A_i A_j)$  表示在边界节点上有 2 个互不相同的处于边界节点内部的邻居, 其与边界节点相交的第一个交点均不被其他邻居覆盖的概率, 根据引理 2 证明的过程, 边界节点的一个邻居与其相对位置可以有 3 个阶段, 在这 3 个阶段中, 阴影部分是第 2 个邻居的圆心可以出现的位置, 但无论哪一个阶段, 前 2 个邻居的位置固定之后, 都无法再容纳第 3 个邻居进入并且使前 2 个邻居与边界节点的第一个交点不被第 3 个邻居覆盖, 具体步骤可以采用图 4~图 6 进行穷举, 限于篇幅, 此处不再逐一验证。由于以上分析, 不存在能够不覆盖前 2 个邻居第一个交点的第 3 个邻居, 故所求概率  $P(A_i A_j) = 0$ 。

**引理 4**  $P(A) = 1 - nP(A_1)$  (11)

**证明** 根据引理 1,  $A = \bigcap_{i=1}^n \bar{A}_i$ ,

可得  $P(A) = P(\bigcap_{i=1}^n \bar{A}_i) = 1 - P(\bigcup_{i=1}^n A_i)$

$$= 1 - \left( \sum_{i=1}^n P(A_i) - \sum_{i<j} P(A_i A_j) + \sum_{i<j<k} P(A_i A_j A_k) - \dots + (-1)^{n-1} \sum_{i<j<\dots} P(A_i A_j \dots A_n) \right) \quad (12)$$

又由于引理 3 可得上式中第 3 项开始均为 0, 故可知式(12)为

$$P(A) = 1 - \left( \sum_{i=1}^n P(A_i) \right) \quad (13)$$

而节点是随机均匀分布的, 故  $P(A_1) = P(A_i)$ , 所以  $P(A) = 1 - nP(A_1)$ 。

由以上 4 个引理很容易得出定理 2。

**定理 3** 一个位于边界的节点的感知区域被它的  $n$  个邻居节点覆盖住的百分比不会低于

$$\frac{8\pi - 6n - 6\sqrt{3}}{9\pi} \left( \frac{\sqrt{3} - 2}{2\pi - 3} \right)^{n-1} + \frac{4}{3} \left( \frac{\sqrt{3} - 1}{2\pi - 6} \right)^n + \frac{1}{3} \left( \frac{\sqrt{3} + 1}{2\pi + 6} \right)^{n-1} \quad (14)$$

**证明** 设  $O$  为边界节点， $X_i$  为节点  $O$  的邻居节点， $X_i$  与  $O$  构成的相交扇形为  $P_i^1OP_i^2$ ， $(1 \leq i \leq n)$ ， $\forall i$ ，设  $I_i$  为逆时针方向上在相交扇形之前没有被覆盖住的面积（如图 7 所示），则分为以下 2 种情况：

- 1)  $P_i^1$  位于另一个相交扇形区域内；
- 2)  $P_i^1$  没有位于任何其他相交扇形之内。

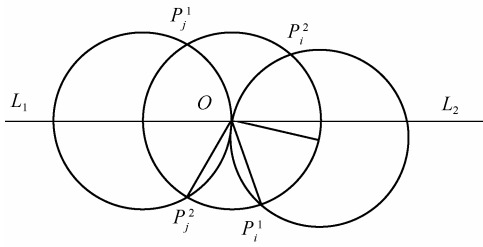


图 7 角  $\alpha$  定义

对于情况 1)，显然节点  $O$  已经被完全覆盖， $I_i = 0$ 。

对于情况 2)，设  $P_j^2$  是在顺时针方向上距离  $P_i^1$  最近的交点， $\alpha = \angle P_j^2OP_i^1$ ，如图 7 所示。设  $S$  是没有被处于边界节点  $O$  的随机分布的邻居覆盖的区域， $E(S)$  为  $S$  的数学期望。显然  $S = \sum_{i=1}^n I_i$ ， $E(S) = E(\sum_{i=1}^n I_i) = \sum_{i=1}^n E(I_i)$ 。考虑到邻居分布的随机和均匀性，只需计算  $E(I_1)$  即可。

$$E(I_1) = \int_{X_1} (\int_{X_2 \dots X_n} I_1 dX_2 \dots dX_n) dX_1 \quad (15)$$

设  $\Delta(X_1) = \int_{X_2 \dots X_n} I_1 dX_2 \dots dX_n$ ，显然有

$$\Delta(X_1) \leq \int_{X_2 \dots X_n} \pi R^2 \frac{\alpha}{2\pi} dX_2 \dots dX_n = \frac{R^2}{2} E(\alpha | X_1) \quad (16)$$

对于给定的  $X_1$ ，考虑角  $\alpha$  的分布概率，如图 8 所示，设  $M$  为弧  $P_1^1P_1^2$  之外的一点， $\angle MOP_1^1 = \theta$ ， $N$  为以  $P_1^1$  为圆心、半径为  $R$  的圆与  $O$  的第 2 个交点， $Q$  为以  $M$  为圆心、半径为  $R$  的圆与  $O$  的第一个交点，则  $0 \leq \theta \leq \pi$ ， $\alpha \geq \theta$  当且仅当除  $X_1$  之外的其他邻居  $X_2, \dots, X_n$  的圆心都位于图 8 的阴影区域，所以只需计算阴影部分的面积占整个圆面积的百分比，就可以算出相应概率。由于  $O$  位于网络边界，有效区域仅为半圆，所以对于阴影部分的面积计算分成 2 种情况：

1)  $P_1^1$  在边界节点内部，而  $P_1^2$  在边界节点外部（如图 8 所示）；

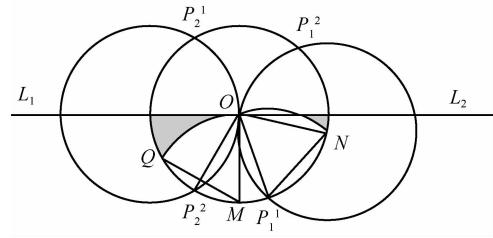


图 8 交点分列边界两边示意

2)  $P_1^1$  和  $P_1^2$  都在边界节点内部（如图 9 所示）。

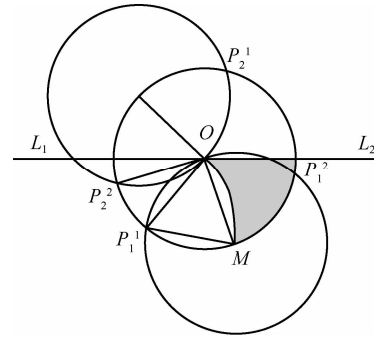


图 9 交点均位于边界节点内部示意

对于情况 1)，有

$$P(\alpha \geq \theta) = \left( \frac{\sqrt{3}}{2\pi} - \frac{1}{6} - \frac{\theta}{2\pi} \right)^{n-1}, \quad 0 \leq \theta < \pi \quad (17)$$

可以得到  $\alpha$  的分布函数为

$$F(\theta) = 1 - \left( \frac{\sqrt{3}}{2\pi} - \frac{1}{6} - \frac{\theta}{2\pi} \right)^{n-1} \quad (18)$$

由该分布函数可以计算

$$\begin{aligned} E(\alpha | X_1) &= \int_0^\pi \theta dF(\theta) = (\theta F(\theta))_0^\pi - \int_0^\pi F(\theta) d\theta \\ &= \frac{4\pi - 3n - 3\sqrt{3}}{3n} \left( \frac{\sqrt{3}}{2\pi} - \frac{2}{3} \right)^{n-1} + \frac{2\pi}{n} \left( \frac{\sqrt{3}}{2\pi} - \frac{1}{6} \right)^n \quad (19) \end{aligned}$$

此时

$$\begin{aligned} E(S) &= E(\sum_{i=1}^n I_i) = \sum_{i=1}^n E(I_i) \leq \frac{nR^2}{2} E(\alpha | X_1) \\ &= \frac{nR^2}{2} \left( \frac{4\pi - 3n - 3\sqrt{3}}{3n} \left( \frac{\sqrt{3}}{2\pi} - \frac{2}{3} \right)^{n-1} + \frac{2\pi}{n} \left( \frac{\sqrt{3}}{2\pi} - \frac{1}{6} \right)^n \right) \quad (20) \end{aligned}$$

该平均面积占边界节点内部面积（半圆）的比例为该值再除以半圆的面积，即

$$\frac{n}{\pi} \left( \frac{4\pi - 3n - 3\sqrt{3}}{3n} \left( \frac{\sqrt{3}}{2\pi} - \frac{2}{3} \right)^{n-1} + \frac{2\pi}{n} \left( \frac{\sqrt{3}}{2\pi} - \frac{1}{6} \right)^n \right) \quad (21)$$

对于情况 2), 可直接求得平均面积占边界节点内部面积的比例为

$$\left(\frac{\sqrt{3}}{2\pi} + \frac{1}{6}\right)^{n-1} \quad (22)$$

情况 1) 的可能性为总的 2/3, 情况 2) 的可能性占总的 1/3, 故由式(21)和式(22)最终得到一个位于边界的节点的感知区域被它的  $n$  个邻居节点覆盖住的百分比不会低于

$$\frac{8\pi - 6n - 6\sqrt{3}}{9\pi} \left(\frac{\sqrt{3}}{2\pi} - \frac{2}{3}\right)^{n-1} + \frac{4}{3} \left(\frac{\sqrt{3}}{2\pi} - \frac{1}{6}\right)^n + \frac{1}{3} \left(\frac{\sqrt{3}}{2\pi} + \frac{1}{6}\right)^{n-1} \quad (23)$$

定理 3 得证。

综合定理 2 和定理 3, 可以计算得到如表 1 所示的结论, 在表 1 中,  $n$  代表邻居数量,  $PR_{CB}$  代表边界节点被  $n$  个邻居节点完全覆盖的概率,  $PCT_{CB}$  代表边界节点被  $n$  个邻居节点覆盖住的区域占内部区域的百分比,  $PR_{CI}$  代表内部节点被  $n$  个邻居节点完全覆盖的概率,  $PCT_{CI}$  代表内部节点被  $n$  个邻居节点覆盖住的区域占内部区域的百分比。

表 2 边界节点、内部节点与邻居的关系

$n$	$PR_{CB}$	$PCT_{CB}$	$PR_{CI}$	$PCT_{CI}$
4	74.78%	$\geq 95.14\%$	9.65~22.22%	$\geq 86.22\%$
5	83.36%	$\geq 98.68\%$	31.22~37.00%	$\geq 91.62\%$
6	89.09%	$\geq 98.75\%$	49.74~52.13%	$\geq 94.90\%$
7	<u>92.95%</u>	<u><math>\geq 99.69\%</math></u>	64.29~65.21%	$\geq 96.89\%$
8	95.52%	$\geq 99.73\%$	75.15~75.48%	$\geq 98.11\%$
9	97.19%	$\geq 99.92\%$	82.97~83.09%	$\geq 98.85\%$
10	98.26%	$\geq 99.95\%$	88.48~88.52%	$\geq 99.30\%$
11	98.93%	$\geq 99.96\%$	<u>92.28~92.30%</u>	$\geq 99.57\%$

对比表 2 的数据, 可以看到内部节点被  $n$  个邻居节点完全覆盖的概率达到 90% 以上并且被  $n$  个邻居节点覆盖住的区域占内部区域的百分比达到 99% 以上时需要的邻居数量为 11 个, 而边界节点只要有 7 个邻居节点就会产生类似的效果。可见网络边界的调度策略与网络内部的调度策略不同。

从表 2 中可以得到算法 3 中  $N$  的值应该取 7。

## 5 算法的能量消耗分析

设一个节点的总能耗为  $E_{Total}$ , 在工作状态的总

能耗为  $E_w$ , 在预休眠状态的总能耗为  $E_p$ , 在休眠状态的总能耗为  $E_s$ , 显然有

$$E_{Total} = E_w + E_p + E_s \quad (24)$$

节点在工作状态完成数据的采集、处理和节点之间的通信, 节点的采集模块和微处理单元分别负责采集和处理, 设单位时间内数据采集的能耗为  $U_{WG}$ , 用于采集的时长为  $T_{WG}$ , 单位时间内用于处理的能耗为  $U_{WP}$ , 用于采集的时长为  $T_{WP}$ , 则采集和处理的能耗为

$$U_{WG}T_{WG} + U_{WP}T_{WP} \quad (25)$$

按照算法的模型, 有两种消息在邻居节点之间通信, 分别是信标消息和选票消息。设发送一个字节的消息的能耗为  $U_{MS}$ , 接收一个字节的能耗为  $U_{MR}$ , 信标消息的字节数为 8 (设自身 ID 占 4 byte, 活动邻居数量占 4 byte), 选票消息的字节数为 12 (设自身 ID 占 4 byte, 目标节点 ID 占 4 byte, 选票有效时间占 4 byte)。根据算法 1, 设发送和接收信标消息的时间间隔为 BTI (beacon time interval), 发送选票消息的时间间隔为 TSTI (ticket send time interval), 接收选票消息的时间间隔为 TRTI (ticket receive time interval), 另设节点的总生命周期为  $T_{Total}$ , 则用于通信的能耗为发送和接收信标消息的能耗、发送选票消息的能耗、接收选票消息的能耗等几部分之和, 即

$$\frac{T_{Total}}{BTI} 8U_{MS} + \frac{T_{Total}}{BTI} 8U_{MR} + \frac{T_{Total}}{TSTI} 12U_{MS} + \frac{T_{Total}}{TRTI} 12U_{MR} \quad (26)$$

所以工作状态的总能耗为采集、处理和通信的能耗之和, 即

$$E_w = U_{WG}T_{WG} + U_{WP}T_{WP} + \frac{T_{Total}}{BTI} 8U_{MS} + \frac{T_{Total}}{BTI} 8U_{MR} + \frac{T_{Total}}{TSTI} 12U_{MS} + \frac{T_{Total}}{TRTI} 12U_{MR} \quad (27)$$

而预休眠状态和工作状态相比只是在处理方面增加了一些判断(参见算法 1 的第 4)部分), 其他都与工作状态相同, 可以认为其能耗与工作状态近似相等, 有

$$E_p = E_w \quad (28)$$

设休眠时单位时间内的能耗为  $U_s$ , 用于采集的时长为  $T_s$ , 则

$$E_S = U_S T_S \quad (29)$$

将式(27)、式(28)、式(29)代入式(24), 可得

$$E_{\text{Total}} = 2U_{\text{WG}}T_{\text{WG}} + 2U_{\text{WP}}T_{\text{WP}} + \frac{16T_{\text{Total}}(U_{\text{MS}} + U_{\text{MR}})}{BTI} + \frac{24T_{\text{Total}}U_{\text{MS}}}{TSTI} + \frac{24T_{\text{Total}}U_{\text{MR}}}{TRTI} + U_S T_S \quad (30)$$

注意到式(30)中表示工作状态、预休眠状态和休眠状态的时间段彼此是没有交集的, 即同一时刻一个节点只能处于 3 种状态当中的一种, 在每种状态下能耗不同, 根据调度情况每种状态会各自持续一定时间。虽然随着具体每种不同传感器节点类型参数不同, 以及节点活动邻居的数量不同, 会导致式(30)中各个参数的取值有一定差别, 但节点能耗与时间的总体趋势是体现为在 3 种状态之间切换, 差别就在于每次工作和预休眠状态持续的时间长短。通过这种调度会使单个节点在某些时间段内处于休眠状态, 能耗减少, 总生命周期延长。

从网络的整体情况看, 网络的总能耗是各个节点总能耗之和, 无论采用何种调度方法, 最终各个节点都死亡时网络的生命周期也就结束了, 所以网络的总能耗是一定的。在这个前提下, 争取做到网络的生命周期尽量长。如果不进行调度, 所有节点一直处于工作状态, 网络的生命周期等于每个节点的生命周期, 而采用本文提出的 RNSSA 算法进行调度, 增加了一定的通信成本(发送和接收信标消息、选票消息)和处理成本(对选票进行判断), 但为大多数节点提供了一定时长的休眠机会, 只要用于通信和处理的能耗小于通过休眠节省的能耗, 对于网络整体来说就是合算的, 即如果式(31)成立, 对于网络整体就是合算的。

$$2U_{\text{WP}}T_{\text{WP}} + \frac{16T_{\text{Total}}(U_{\text{MS}} + U_{\text{MR}})}{BTI} + \frac{24T_{\text{Total}}U_{\text{MS}}}{TSTI} + \frac{24T_{\text{Total}}U_{\text{MR}}}{TRTI} < (U_{\text{WG}} - U_S)T_S \quad (31)$$

式(31)给出了设置传感器节点各种参数值的理论指导, 应确保式子右边足够大。单位时间内数据采集的能耗  $U_{\text{WG}}$  远大于单位时间内休眠的能耗  $U_S$  (对于大多数传感器来说, 二者都相差至少 2 个数量级), 故只需保证休眠时长  $T_S$  足够大即可, 根据文献[19,20]以及通过仿真实验, 将式中各部分

参数设置为如表 3 所示的值, 会得到比较理想的结果(网络总生命周期延长的同时保证一定的监控质量)。

表 3 算法参数的取值

参数名	占总生命周期的百分比
发送、接收信标消息的时间间隔 $BTI$	3%~5%
发送选票消息的时间间隔 $TSTI$	3%~4%
接收选票消息的时间间隔 $TRTI$	3%~4%
休眠时长 $T_S$	10%~20%

注: 此处的总生命周期指的是没有调度之前一个节点一直处于工作状态能够生存的时长。

## 6 系统仿真

采用 MATLAB 进行仿真。构建一个半径 50 m 的圆形区域, 在其中随机部署 600 个节点, 其密度达到了  $600/(50^2 \pi)$ , 设每个节点的监测半径为 8 m, 可以计算出平均每个节点的邻居数量为 15.36, 满足本文算法的阈值要求。为了快速计算得到仿真结果, 将节点的生命周期设置为 100 s, 每隔 5 s 发送一次信标消息, 每隔 3 s 进行一次节点检查, 节点的休眠时长为 10 s/次, 节点进入预休眠状态后等待时长为一个不超过 2 s 的随机数, 内部节点的选票阈值为 11, 边界节点的选票阈值为 7。不同类型的传感器以及参数设定的不同会导致节点工作方式的变化, 本文仿真实验中采用文献[22]提供的能量模型, 即节点在工作状态和预休眠状态下, 有 10% 的时间在节点之间进行数据传输, 有 70% 的时间在进行数据采集, 20% 的时间处于空闲状态, 节点在传输、采集、空闲和休眠时能量消耗比率为 20:8:4:0.01。

仿真实验进行两方面的比较。首先对比文献[19]中提出的 LDAS 算法与本文提出的冗余节点休眠调度算法(RNSSA 算法)在上述条件下运行时工作节点数量随时间的变化曲线, 每间隔 10 s 进行一次网络覆盖率的采样, 进行 5 次实验, 取平均值得到的运行结果如图 10 所示。从图 10 中可以看出, 2 种算法在前 800 s 的网络覆盖率都可以达到 95% 以上, LDAS 算法在 800 s 左右出现第一次覆盖率明显下降, 随后到了 1 300 s 左右时, 又出现第二次比较明显的覆盖率下降, 到 1 700 s 左右时, 几乎所有节点都处于死亡状态, 网络的生命周期结束。初步分析可能是由于大多数边界节点一直处于工作状态,

能量耗尽而死亡，出现了边界收缩现象，从而导致网络覆盖率下降，其具体产生的原因通过后续第二个仿真实验研究。而 RNSSA 算法调度的效果显示，在整个网络生命周期中没有明显突发的网络覆盖率下降的情况，即图中没有哪个时刻的网络覆盖率突变，从开始运行到 1 800 s 左右覆盖率都在 90% 以上，随后以近似于一条抛物线的形式下降，这说明该算法在调度过程中没有大批量节点同时死亡的情况发生，而且网络生命周期明显长于 LDAS 算法。

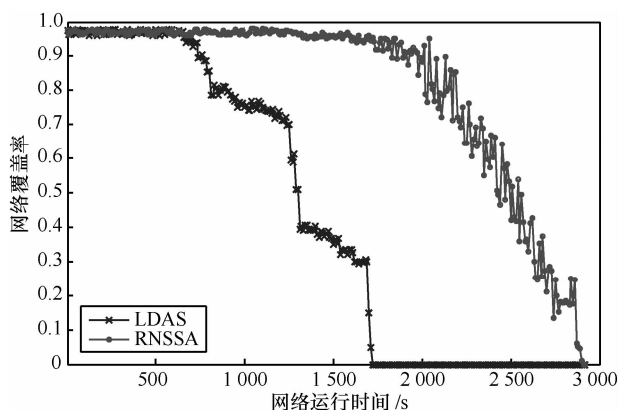


图 10 2 种算法网络运行时间与覆盖率关系

另一方面，对 600 个节点的实际运行情况进行监测，得到结果如图 11~图 16 所示。图 11~图 13 为 LDAS 算法在 200 s、600 s 和 1 000 s 时的采样，图 14~图 16 为 RNSSA 算法在 200 s、600 s 和 1 000 s 时的采样，图中的方块表示工作状态，倒三角表示休眠状态，实心黑色圆点表示死亡状态。

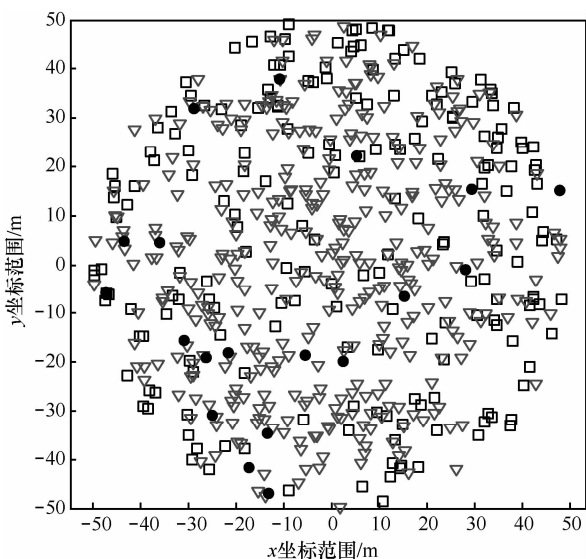


图 11 LDAS 算法 200 s 抽样

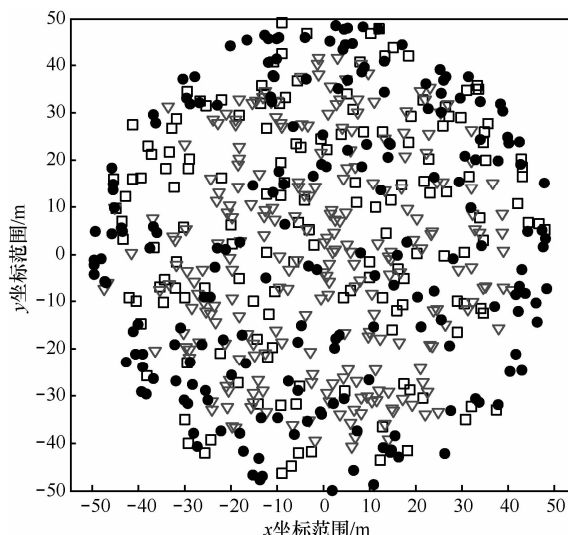


图 12 LDAS 算法 600 s 抽样

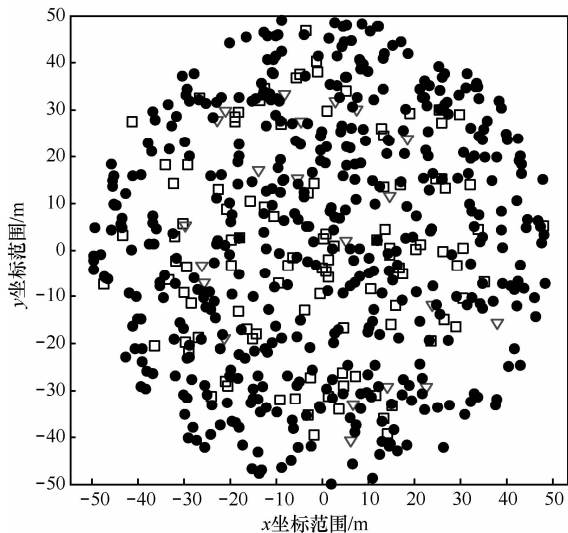


图 13 LDAS 算法 1 000 s 抽样

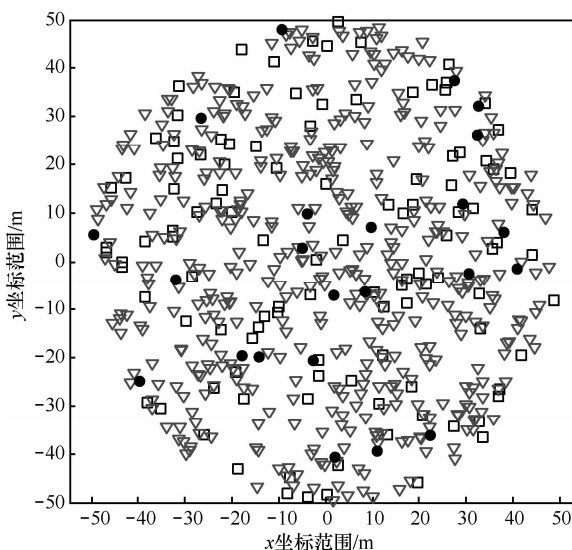


图 14 RNSSA 算法 200 s 抽样

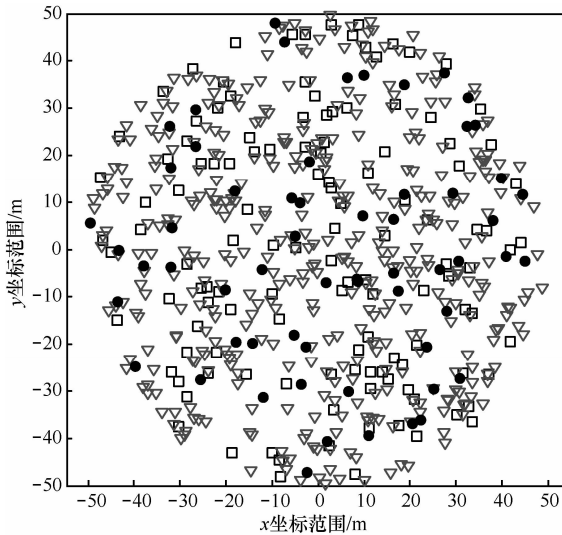


图 15 RNSSA 算法 600 s 抽样

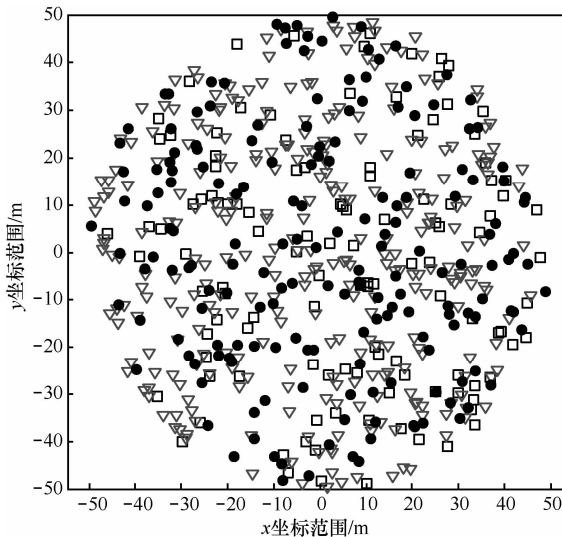


图 16 RNSSA 算法 1 000 s 抽样

从实验结果来看, 采用 RNSSA 算法的调度结果比采用 LDAS 算法调度能让边界节点有更多机会休眠, 可以有效防止出现边界收缩问题。在 200 s 时, 采用 LDAS 算法边界节点工作的概率明显高于内部节点, 而 RNSSA 算法在边界和内部休眠节点的概率几乎差不多。在 600 s 时, 采用 LDAS 算法边界能量耗尽而死亡的节点很多, 将要产生边界收缩现象, 而 RNSSA 算法死亡节点几乎均匀分布, 没有产生明显的边界收缩现象。在 1 000 s 时, 采用 LDAS 算法大多数节点已经死亡, 网络覆盖率明显降低, 而 RNSSA 算法中死亡的节点数量明显少很多, 网络仍然能够正常工作。

通过以上实验可以得出结论, RNSSA 算法能够有效地避免边界收缩问题, 并且能够显著延长整

个网络的生命周期。

## 7 结束语

本文针对高密度随机部署的无线传感器网络, 提出了 RNSSA 算法, 通过对节点休眠策略的调度研究, 可以得到以下结论:

- 1) 在文献[19]的算法基础上, 提出了无线传感器网络冗余节点的休眠调度算法;
- 2) 针对算法中的难点, 分别提出了防止大量节点同时休眠的调度子算法和边界节点判定子算法;
- 3) 通过计算得出边界节点被  $n$  个邻居完全覆盖的概率, 与内部节点被覆盖的概率进行了比较;
- 4) 通过计算得出边界节点被  $n$  个邻居覆盖的面积所占百分比的范围, 与内部节点的百分比进行了比较。

实验结果表明, 本文提出的算法能够通过让冗余节点休眠的调度算法来延长网络的生命周期, 并且能够在不增加通信成本的前提下, 判定网络边界, 对边界节点和内部节点采用不同调度策略, 来有效缓解边界收缩问题。

该模型采用的计算方法还不适用于三维空间, 下一步可以针对三维空间的无线传感器网络进行分析, 得出相应模型和调度算法。

## 参考文献:

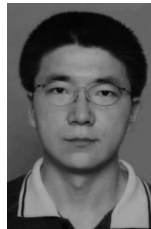
- [1] MAINWARING A, CULLER D, POLASTRE J, *et al.* Wireless sensor networks for habitat monitoring[A]. Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications[C]. New York, USA, 2002. 88-97.
- [2] POPA M, GIRBAN G. On the energy constraint in wireless sensor networks[J]. Computer Communications, 2003, 26(11): 1131-1144.
- [3] STOJMENOVIC I. Position-based routing in ad hoc networks[J]. Communications Magazine, IEEE, 2002, 40(7): 128-134.
- [4] LOCHERT C, HARTENSTEIN H, TIAN J, *et al.* A routing strategy for vehicular ad hoc networks in city environments[A]. Proceedings of IEEE[C]. Intelligent Vehicles Symposium. 2003.156-161.
- [5] SUNIL K, KASHYAP K, ZAN B, *et al.* An energy-aware and intelligent cluster-based event detection scheme in wireless sensor networks[J]. International Journal of Sensor Networks, 2008, 3(2): 123-133.
- [6] SAAD E M, AWADALLA M H, SALEH M A, *et al.* Adaptive and energy efficient clustering architecture for dynamic sensor networks[A]. 2nd International Workshop on Soft Computing Applications SOFA 2007[C]. Oradea, 2007. 221-225.

- [7] TIAN D, GEORGANAS N D. A node scheduling scheme for energy conservation in large wireless sensor networks[J]. *Mobile Computing*, 2003, 3(2): 271-290.
- [8] TIAN D, GEORGANAS N D. A coverage-preserving node scheduling scheme for large wireless sensor networks[A]. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*[C]. New York, USA, 2002. 32-41.
- [9] WANG X, XING G, ZHANG Y, *et al.* Integrated coverage and connectivity configuration in wireless sensor networks[A]. *Proceedings of the 1st international conference on Embedded networked sensor systems*[C]. New York, USA, 2003. 28-39.
- [10] MIHAELA C, WU J. Energy-efficient coverage problems in wireless ad-hoc sensor networks[J]. *Computer Communications*, 2006, 29(4): 413-420.
- [11] WAN P J, YI C W. Coverage by randomly deployed wireless sensor networks[J]. *IEEE/ACM Transactions on Networking (TON)*, 2006, 14(S1): 2658-2669.
- [12] LIU C, WU K, XIAO Y, *et al.* Random coverage with guaranteed connectivity: joint scheduling for wireless sensor networks[J]. *IEEE Transactions on Parallel and Distributed System*, 2006, 17(6): 562-575.
- [13] LAZOS L, POOVENDRAN R. Coverage in heterogeneous sensor networks[A]. *2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*[C]. 2006.1-10.
- [14] ZOU Y, CHAKRABARTY K. Sensor deployment and target localization based on virtual forces[A]. *INFOCOM 2003 Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*[C]. 2003.1293-1303.
- [15] TSAI Y R. Sensing coverage for randomly distributed wireless sensor networks in shadowed environments[J]. *IEEE Transactions on Vehicular Technology*, 2008, 57(1): 556-564.
- [16] JIANG J, FANG L, WEN J, *et al.* Random scheduling for wireless sensor networks[A]. *2009 IEEE International Symposium on Parallel and Distributed Processing with Applications*[C]. 2009.324-332.
- [17] FAN G J, WANG R C, HUANG H P, *et al.* Coverage-guaranteed sensor node deployment strategies for wireless sensor networks[J]. *Sensors*, 2010, 10(3): 2064-2087.
- [18] HU X M, ZHANG J, YU Y, *et al.* Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks[J]. *IEEE Transactions on Evolutionary Computation*, 2010, 14(5): 766-781.
- [19] WU K, GAO Y, LI F L, *et al.* Lightweight deployment-aware scheduling for wireless sensor networks[J]. *Mobile Networks and Application*, 2005, 10(6): 837-852.
- [20] 凡高娟, 王汝传, 黄海平等. 基于容忍覆盖区域的无线传感器网络节点调度算法. *电子学报*, 2011, 1(1): 89-94.
- FAN G J, WANG R C, HUANG H P, *et al.* Tolerable coverage area based node scheduling algorithm in wireless sensor networks[J]. *Acta Electronica Sinica*, 2011, 1(1): 89-94.
- [21] Information processing technology office (IPTO) defense advanced research projects agency (DARPA), bba 07-46 landroids broad agency announcement[EB/OL]. <http://www.darpa.mil/ipto/solicit/baa/BAA-07-46 PIP.pdf>, 2007.
- [22] HILL J, SZEWCZYK R, WOO A, *et al.* System architecture directions for networked sensors[J]. *ACM SIGOPS Operating Systems Review*, 2000, 34(5): 93-104.

#### 作者简介:



温涛 (1962-), 男, 山西宝鸡人, 东北大学教授、博士生导师, 主要研究方向为网络安全、知识组织。



张冬青 (1976-), 男, 辽宁大连人, 东北大学博士生, 主要研究方向为无线传感器网络。

郭权 (1973-), 男, 辽宁大连人, 博士, 大连东软信息学院教授, 主要研究方向为计算机网络、网格计算。

宋晓莹 (1984-), 女, 辽宁朝阳人, 东北大学博士生, 主要研究方向为无线传感器网络。