

中国科学技术大学

---

博士学位论文

---

无线传感网络拓扑控制若干问题的研究

---

姓名：李建波

---

申请学位级别：博士

---

专业：计算机软件与理论

---

指导教师：黄刘生

---

20090501

## 摘要

无线传感网络通常具有如下特点：大规模、自组织、传感器节点能量受限、部署环境恶劣等。这些特点决定了必须设计能量有效的协议来减少传感器节点的能耗以延长无线传感网络的寿命。拓扑控制问题作为无线传感网络中的核心问题之一，对于延长网络的生存时间、减小通信干扰、提高 MAC 协议和路由协议的效率以及促进数据融合、提高网络的可扩展性、可靠性、安全性等其他性能等具有重要意义。因此如何进行拓扑控制以优化拓扑结构并延长传感网络的寿命已经成为重要的研究方向。

本文主要研究能量有效的拓扑控制算法：(1) 应用于静态密集部署传感网络的一种基于完全图的分簇算法 (2) 一种在上述分簇算法基础上加入移动节点担当网关的算法 (3) 应用于稀疏移动传感网络的传输功率控制算法 (4) 应用于大规模部署移动传感网络的分簇算法。

无线传感网络的某些应用，例如生态监护、军事侦察等，为了能够准确的对数据进行收集并保持覆盖性，通常会在监测区域部署大量冗余的静态节点。在这种情况下，为了能够延长传感网络的寿命，通常选取一定的节点保持活动，由它们构成数据传输的主干网，而其他节点在没有任务时可以进入睡眠状态，这正是层次分簇算法所应用的场合。针对分簇算法过程中的重新分簇所带来的高负载问题，本文提出了一种基于完全图的能量有效的分簇算法 CGCA(Complete Graph based Clustering Algorithm)。CGCA 利用完全图中节点之间互相等价的性质，只在系统启动的时刻执行分簇算法，而在以后的重新分簇阶段，簇头只需要在每个簇的内部节点间进行轮换，而不像以前的分簇算法需要进行全局性的触发来选举新的簇头。因此 CGCA 从根本上消除了重新分簇所带来的“涟漪”效应。仿真试验表明：在节点密集部署的情况下，CGCA 产生的消息交换个数远小于 HEED(Hybrid Energy Efficient Distributed clustering) 分簇算法。最后簇头均匀分布方面，CGCA 也明显优于 LEACH(Low-Energy Adaptive Clustering Hierarchy) 分簇算法。

在一个分簇算法中，一个网关节点 (gateway) 除了要完成负责基本的数据之外，还要在相邻的簇之间承担起数据转发的任务。由于簇之间的数据流量相对较大，因此相对于普通节点来说，网关节点更容易耗尽自己的能量，因此这些网关节点容易成为系统的瓶颈(bottleneck)，从而严重影响传感网络系统的性能表现。本文在 CGCA 分簇算法的基础上加入了资源和配置丰富的移动节点来担当簇之间的网关，以把普通节点从转发簇间数据的繁重任务中解放出来，

籍此达到延长传感网络寿命的目的。除此之外，由移动节点来担当网关可以显著的提高系统的稳定性和容错性，这是因为由配置丰富的移动节点代替能量受限的普通节点来担当网关可以减少系统出现故障的概率，从而可以进一步的提升系统的整体性能表现。实验表明，在密集部署传感器节点的网络中，本算法产生的消息交换个数只是传统的基于最小标号分簇算法 LIC(Low Identifier Clustering)的20%左右。另外，本算法所取得的系统时间也是 LEACH 算法的两倍左右。

无线传感网络在军事和紧急搜救的应用促使了移动传感网络的兴起。由于这些被监控对象本身所具有的移动性，使得底层的拓扑在不断的发生变化，因此很难应用传统的拓扑控制算法或者简单的对它们改进使之适用于移动传感网络。本文提出了在给定移动网络模型 VRMN(Variant Rate Mobile Network)下的一种基于传输功率控制的拓扑控制算法，并分别给出了它的集中式和分布式版本。本算法首先求出每个节点的一跳邻居集合，然后利用类似于 XTC 的方法把可由较近邻居节点中继到达的最远邻居节点删除，这样在不损失连通性的前提下就可以减少节点的发射功率，从而达到节省节点能量以延长传感网络寿命的目的。理论证明，本算法是高效的，它具有  $O(n^3)$  的多项式复杂度时间。实验结果表明，本算法在减少节点传输功率和保持网络连通性方面也取得了较好的性能。

在一个大规模移动传感网络中，由于分簇算法能够形成一个层次结构以及能够较好的支持网络的可扩展性并在节点移动时能够较好的维护拓扑结构的稳定性，这就使得分簇算法比较适合于对这种大规模密集移动传感网络进行拓扑控制。本文分簇算法首先将部署区域划分成小的单元格，在每个单元格的中心位置事先指定一个节点担当簇头，其他的移动节点只需要监听簇头的“Hello”消息并通过比较消息的信号强弱就可以加入到相应的簇头中。在簇形成阶段，本文分簇算法的时间复杂度和消息复杂度均为  $O(N)$ ，从而较大的节省了节点的能耗并取得了较好的能量有效性。而在簇维护阶段，簇的维护是由事件触发并且异步进行，不会产生“涟漪”效应，因此本文算法能够取得一个较为稳定的簇结构以适应节点的移动和底层拓扑的变化。仿真实验表明，本文算法能够产生较少的消息个数和较长的系统寿命。

本文的主要贡献和创新点如下：

- 1) 针对重新分簇所带来的大量计算和通信负载以及“涟漪”效应，本文提出了一种基于完全图的分簇算法 CGCA，使得重新分簇只是异步的局部触发，从而维持了簇结构的稳定性以延长系统生命期。
- 2) 在上述分簇算法的基础上，通过加入资源丰富的移动节点来担当转发簇间数据的网关，从而使得普通节点能够集中到收集或者探测等任务上，这能够进一步的节省能量从而延长整个传感网络的工作寿命。

- 3) 在一个给定的移动网络模型 VRMN 下, 提出了一种多项式时间的基于功率控制的拓扑控制算法, 本算法能够在保持网络连通性的前提下, 减少了节点的发射功率, 从而达到节省节点能量的目的。
- 4) 在一个大规模移动传感网络中, 提出了一种能量有效的分簇算法, 它具有简单的簇形成和维护过程, 使得节点移动可能对簇结构的影响局限在一个较小的范围内, 从而形成了一个较为稳定的簇结构。

**关键词:** 无线传感网络 移动传感网络 传感器节点 拓扑控制 功率控制 分簇算法 移动模型 能量有效 系统生命期 时间复杂度

## Abstract

WSN (Wireless Sensor Networks) usually has the following features: large scale, self-organization, limited energy equipped sensor nodes and bad deployment environment etc, which determines that some energy efficient protocols must be designed to prolong the lifetime of WSN by reducing the energy consumption of sensor nodes. Topology control, as a core problem of WSN, has a fundamental impact on some important network parameters, such as prolonging the system lifetime of WSN, reducing signal interference, enhancing the energy efficiency of MAC and routing protocols, promoting data aggregation and enhancing the scalability, stability, and safety of WSN. As a result, how to perform topology control to optimize topology structure and prolong the WSN lifetime has been an important research topic recently.

This dissertation mainly focuses on investigating energy efficient topology control algorithms: (1) propose a complete graph-based clustering algorithm (CGCA) applied in static densely deployed sensor networks (2) introduce some resource and hardware rich mobile nodes acting as inter-cluster gateways so as to augment the aforementioned clustering algorithm (3) propose a power control based topology control algorithm used in sparse mobile sensor networks (4) propose a clustering algorithm used in large-scale deployed mobile sensor networks.

Some applications in WSN, such as ecosystem monitoring and military surveillance, often deploy large redundant sensor nodes on the monitored area to accurately perform data gathering and preserve coverage. In this situation to prolong the lifetime of WSN, some nodes keep active to form the backbone of forwarding data traffic, while other nodes can turn off their radios when having no tasks, and this is exactly where the hierarchical clustering algorithm handles. Aiming to solve the high overheads brought by re-clustering in a clustering algorithm, we propose an energy efficient complete graph-based algorithm (CGCA). By using the property that the nodes are of equivalence each other in a complete graph, CGCA is only executed at the system activation time and the cluster head role needs only to be rotated among the internal nodes in each cluster at the subsequent re-clustering phase, while the previous clustering algorithms need a global trigger to re-elect cluster heads, which incurs greatly reduced communication and computation overheads. Consequently our CGCA algorithm totally eliminates the ripple effect in re-clustering phase. The simulation

experiments demonstrate that the number of exchanged message produced by CGCA is much less than that of HEED clustering algorithm in the densely deployed case. Finally, CGCA significantly outperforms LEACH algorithm in terms of evenly distributing cluster heads.

Besides performing the basic sensing data task, a gateway node is also being responsible for the forwarding the data traffic among adjacent clusters in a typical clustering algorithm. Compared with the ordinary node, a gateway is much prone to deplete its energy due to the large data traffic among inter-clusters, thus being the bottlenecks of the system and in turn degrading the sensor network performance. This dissertation introduces some resource and configuration rich mobile nodes to act as inter-cluster gateways to for the purpose of liberating the ordinary nodes from the heavy forwarding inter-cluster data, thus achieving the aim of prolonging the lifetime of WSN. Furthermore, letting the mobile nodes act as gateways can significantly enhance the stability and fault-tolerance of the whole system, because substituting the configuration rich mobile nodes for the energy constrained ordinary nodes can reduce the probability of occurring system malfunction, thus further promoting the system overall performance. Extensive simulation experiments demonstrate that the number of exchanged messages produced by our algorithm is only about 20% that of the traditional identifier based clustering algorithm in a densely deployed case. Furthermore, our proposed achieves an improvement in system lifetime of factor 2 that of the LEACH in a dense sensor network.

The military and emergency rescuing applications of wireless sensor networks promote the proliferation of MSN (Mobile Sensor Networks). The changing underlying topology, due to the intrinsic mobility characteristic of monitored objects, makes it difficult to apply or adapt the traditional topology control algorithms to be applied in MSN. This dissertation proposes a power control based topology control algorithm under a given mobility model called VRMN (Variant Rate Mobile Network) and then presents its central and distributed versions respectively. In this algorithm each node first computes the one-hop neighbor set and then applies a XTC-like procedure to prune the furthest neighbor that can be reached by the relay of its closer neighbor, which incurs reducing transmitting power of each node and consequently saving power consumption to prolong the network lifetime on the basis of without impairing the network connectivity. Our algorithm obtains a polynomial time complexity of  $O(n^3)$  thus being efficient from the theoretic view and the experiments results show that our

algorithm achieves comparatively better performance from the perspective of reducing node transmitting power and maintaining network connectivity.

In a large scale mobile sensor network, a clustering algorithm can form a hierarchical structure, which can better support the scalability of network and maintain the stability of topology in the presence of node movement, thus making it well suitable for the topology control problem in a large-scale densely deployed mobile sensor network environment. Our clustering algorithm first partitioned the deployment region into some small grids, for each of which a mobile node was dispatched to act as cluster head. Subsequently, all the other mobile nodes can join their corresponding cluster by monitoring and comparing the signal strength of the “Hello” messages sent from cluster heads. In the cluster formation phase, our algorithm achieves both  $O(N)$  time and message complexity. In the subsequent cluster maintenance phase, the maintenance of the cluster structure is unsynchronously event driven, thus eliminating the ripple effect. So our algorithm achieves a comparatively stable cluster structure so as to be well adapt to the node’s movement and underlying topology’s change. Furthermore, the simulation results show that our algorithm can produce comparatively small number of messages and long system lifetime.

The contributions and novelties of this dissertation are as following:

- 1) Aiming to solve the large computation and communication overheads and “ripple” effect brought by re-clustering, this dissertation proposes a complete graph-based clustering algorithm, called CGCA, which makes re-clustering unsynchronized and locally triggered, for the purpose of maintaining a stable cluster structure and prolonging the system lifetime.
- 2) On the basis of the aforementioned algorithm, some resource rich mobile nodes are put into the sensor networks so as to act as gateways to relay the heavy inter-cluster data traffic, which can make the ordinary sensor nodes more devoted to the gathering or probing tasks and consequently further saving much energy to prolong the lifetime of the sensor networks.
- 3) Under a given mobility model called VRMN, this dissertation presents a power control based topology control algorithm with a polynomial time complexity, which can reduce transmitting power on a per-node basis, thus achieving the goal of saving energy, without impairing the network connectivity.
- 4) In a large scale mobile sensor networks, this dissertation presents a energy

eifficient clustering algorithm, which has simple cluster formation and maintenance procedure. Our algorithm achieved a much stable cluster structure by limiting impact of the movement of nodes on the cluster into a local area.

**Key words:** wireless sensor networks, mobile sensor networks, sensor nodes, topology control, power control, clustering algorithm, mobility model, energy efficiency, system lifetime, time complexity

## 图表目录

|                                      |    |
|--------------------------------------|----|
| 图 1.1 无线传感网络通信体系结构 .....             | 2  |
| 图 1.2 传感器节点示例 .....                  | 3  |
| 图 1.3 传感器节点基本结构 .....                | 3  |
| 图 1.4 功率控制算法分类 .....                 | 8  |
| 图 1.5 邻近图示例 .....                    | 9  |
| 图 1.6 最小支配集示例 .....                  | 11 |
| 图 2.1 随机路点模型下节点的移动模式 .....           | 24 |
| 图 2.2 无线传感网络的单位圆盘图 .....             | 25 |
| 图 3.1 GAF 算法中网格边长的确定 .....           | 28 |
| 图 3.2 GAF 算法中节点状态转换图 .....           | 29 |
| 图 3.3 系统启动时刻的分簇过程 .....              | 32 |
| 图 3.4 最邻近簇头之间的最大跳数 .....             | 34 |
| 图 3.5 CGCA 分簇示例 .....                | 35 |
| 图 3.6 簇头个数随传输半径的变化趋势 .....           | 38 |
| 图 3.7 单节点簇的比例随着节点个数的变化趋势 .....       | 38 |
| 图 3.8 CGCA 与 HEED 的消息交换个数比较 .....    | 40 |
| 图 3.9 CGCA 与 LEACH 协议的簇头分布比较 .....   | 41 |
| 图 4.1 传感网络中加入移动中继节点 .....            | 45 |
| 图 4.2 加入移动网关的 CGCA 算法 .....          | 48 |
| 图 4.3 簇的数目随着簇大小阀值的变化 .....           | 50 |
| 图 4.4 三种分簇算法所产生的消息个数 .....           | 51 |
| 图 4.5 本文算法与 LEACH 的网络寿命比较 .....      | 52 |
| 图 4.6 移动网关节点个数的变化趋势 .....            | 53 |
| 图 5.1 节点 $v_i$ 和 $v_j$ 之间的最大距离 ..... | 60 |
| 图 5.2 应用类 XTC 算法实例 .....             | 61 |
| 图 5.3 VRMN 下集中式拓扑控制算法流程 .....        | 62 |
| 图 5.4 VRMN 下分布式拓扑控制算法流程 .....        | 63 |
| 图 5.5 算法执行完例程 1 的初始拓扑 .....          | 65 |
| 图 5.6 算法执行完例程 2 的最终拓扑 .....          | 65 |
| 图 5.7 本文算法在减少节点发射半径方面的性能表现 .....     | 66 |

|                                   |    |
|-----------------------------------|----|
| 图 5.8 算法的连通度性能表现 .....            | 67 |
| 图 6.1 本文的初始簇结构 .....              | 70 |
| 图 6.2 分簇算法产生的消息个数随最大速度的变化趋势 ..... | 74 |
| 图 6.3 簇内成员平均寿命随最大移动速度的变化趋势 .....  | 75 |
| 图 6.4 系统寿命随着节点个数的变化趋势 .....       | 76 |
|                                   |    |
| 表 1.1 分簇算法的分类 .....               | 10 |
| 表 2.1 无线收发单元在不同状态下的能耗 .....       | 19 |
| 表 3.1 本文定义的一些符号和术语 .....          | 30 |
| 表 3.2 CGCA 簇形成过程示例 .....          | 35 |
| 表 6.1 簇头 1 的信息表 .....             | 72 |
| 表 6.2 仿真参数设置 .....                | 73 |

## 中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文,是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外,论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名: 李建波

签字日期: 2009.6.1

## 中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一,学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权,即:学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版,允许论文被查阅和借阅,可以将学位论文编入有关数据库进行检索,可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

公开 保密(\_\_\_\_年)

作者签名: 李建波

导师签名: 苏利华

签字日期: 2009.6.1

签字日期: 2009.6.1.

## 第1章 绪论

无线传感网络中的节点通常携带能量较少的电池，而节点很多时候都是部署在恶劣的环境里，这使得电池的替换或者充电是不实际的。因此传感器网络的一个很重要的挑战就是设计能量有效的协议以延长传感器网络的寿命。目前研究的热点主要集中在能量高效的 MAC 协议、路由协议以及应用层协议上，随着研究的不断深入，如何进行合理的拓扑控制以优化拓扑结构和延长 WSN 的寿命已成为 WSN 中研究的重要方向。

无线传感网络是一个比较新兴的领域，国际上对无线传感网络的研究仅有十余年的历史，但由于无线传感网络的巨大应用以及市场前景，自从它的诞生之日起，它就迅速成为研究的热点问题。随着科技的发展和应用的需要，无线传感网络的发展也日新月异，同时关于无线传感网络的理论以及应用研究成果也不断涌现。然而这些研究成果还处于起步阶段，距离成熟的应用需求还相差较远。目前无线传感网络拓扑控制问题的研究还不成熟、不系统、不能满足实际应用的要求，因此在诸多方面还有待改进。

本文系统地研究了无线传感网络能量有效的拓扑控制问题，针对不同的应用场景提出了不同了的解决方案，并对诸多方面提出了一些改进方案，从而达到节省节点能量和延长无线传感网络系统寿命的目的。

### 1.1 无线传感网络的背景知识

20世纪90年代以来，嵌入式系统、无线通信、分布式信息处理技术、网络、微机电系统(micro-electro mechanism system, MEMS)、低功耗的模拟和数字电路技术的快速发展使得开发小体积、低成本、低功耗的微型传感器成为可能<sup>[1-9]</sup>。这些传感器通常具有感知、数据处理、计算和无线网络通信能力。无线传感器网络（Wireless Sensor Network, WSN）就是由这些大量、廉价的传感器节点以无线、多跳、自组织方式构成的网络，它能够实时地监测、感知和采集部署区域中感兴趣的感知对象的各种信息（如光强、温度、湿度、噪音和磁场强度等物理现象），并对些信息进行处理后以无线的方式发送出去，最后通过卫星或者 Internet 网发送给任务管理者。因此通常的无线传感网络通信体系结构如图 1.1 所示：

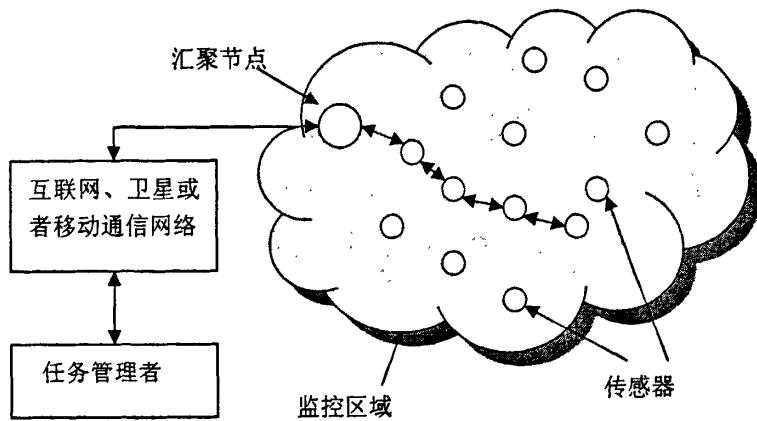


图 1.1 无线传感网络通信体系结构

如图 1.1 所示，无线传感网络能够将逻辑上的信息世界与真实的物理世界融合在一起，从根本上改变了人类与自然交互的方式，非常符合 Mark Weiser<sup>[10]</sup>提出的“普适”（Ubiquitous Computing, UC）计算的思想，也就是说用户和周围环境的很多个大大小小的计算设备在潜意识上进行交互。由于无线传感网络具有自组织、可快速部署和不依赖固定基础设施等优点，因此它具有广阔的应用前景，如国防军事、城市管理、工农业、生物医疗、环境监测、制造业、危险区域远程控制等领域都有潜在的巨大适用价值<sup>[11-15]</sup>，已经引起了许多国家工业界和学术界的高度重视，被认为是 21 世纪能够产生巨大影响的技术之一。

如图 1.1 所示，传感器节点作为一种微型的嵌入式设备，构成了无线传感网络的基础层支撑平台。因为无线传感网络大部分是采用电池供电，并且工作环境通常比较恶劣，因此更换电池或者对其充电都非常困难，所以低功耗是无线传感网络最重要的设计准则之一，从传感器节点的硬件设计到整个网络各层的协议设计都把节能作为最重要设计的目标，以尽可能延长无线传感网络的系统寿命。

由于具体的应用背景不同，目前国内外出现了多种无线传感网络节点的硬件平台。典型的节点包括美国的 CrossBow<sup>[16,17]</sup>公司开发的 Mote 系列节点 Mica2、MicaZ 以及 Mica2Dot，Infineon 公司开发的 EYES 传感器节点等等。如图 1.2 所

示。

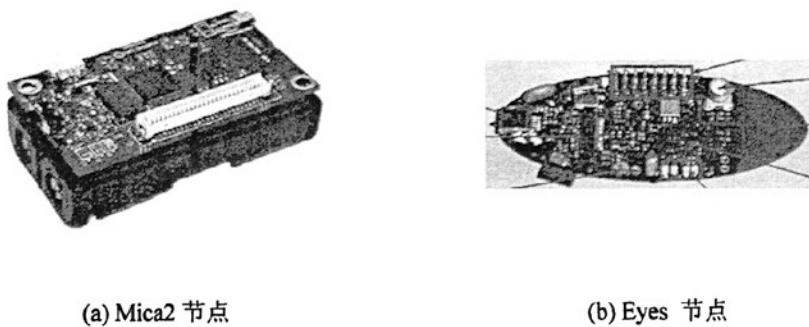


图 1.2 传感器节点示例

实际上各平台最主要的区别是采用了不同的处理器、无线通信协议以及与应用相关的不同的传感器。常用的处理器有 Intel StrongARM<sup>[18]</sup>、Texas Instrument MSP 430 和 Atmel Atmega，常用的无线通信协议有 802.11b、802.15.4/ZigBee、和 Bluetooth 等；与应用相关的传感器有光传感器、热传感器、压力传感器以及湿度传感器等。虽然具体应用不同，传感器节点的设计也不尽相同，但是其基本结构是一样的。传感器节点通常由处理器单元、无线传输单元、传感器单元和能量供应单元 4 部分组成，如图 1.3 所示。

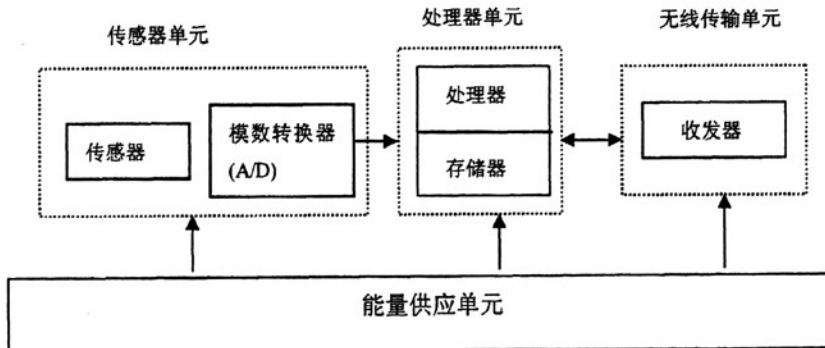


图 1.3 传感器节点基本结构

### 1.1.1 无线传感网络的特点

无线传感网络尽管与传统的无线自组网 Ad-hoc、MANET 具有一些相同的特点<sup>[19,20]</sup>，例如无中心和自组织性、动态变化的网络拓扑、受限的无线传输带宽、移动终端的能力有限、多跳路由以及安全性较差等。但是无线传感网络具有以下

一些区别于其他无线自组网的独有的特征：

- 1) 大规模部署：为了获取精确的信息，通常在监测区域部署大量的传感器节点。这样一方面可以提高收集信息的精度，降低对单个传感器节点的精度要求，另外大量冗余节点的存在，使得系统具有较好的容错性。
- 2) 以数据为中心的网络：目前的互联网是一个以地址为中心的网络，而无线传感网络通常用来监测或者采集一定的事件，然后将采集到的事件汇报给对事件感兴趣的管理者。管理者关心的是是否采集到事件以及事件的精确性，而不关心到底是某个或者某几个节点采集到的事件。这种以数据本身作为查询和传输线索的思想更接近于自然语言交流，因此通常说无线传感网络是一个以数据为中心的网络。
- 3) 移动性较弱：与 MANET(Mobile Ad-hoc NETwork)相比，通常移动性较弱。
- 4) 能量更受限：传感器节点由于自身体积的原因，配备能量十分有限的电池，例如 MicaZ 的电源供应为 2 节五号电池。另外，传感器通常部署在比较复杂和恶劣的环境，这就使得以更换电池或者以对其充电来进行能源补充是不现实的。因此在无线传感网络中，设法降低节点的能耗是系统首先考虑的第一要素。

### 1.1.2 无线传感网络的国内外研究现状

无线传感网络是继因特网 Internet 之后，将对 21 世纪人类生活方式产生重大影响的 IT 热点技术。由于无线传感网络的巨大应用前景，为此，2003 年 8 月，美国《商业周刊》在其“未来技术专版”中发表文章，将传感网络列为有可能掀起新的产业浪潮的未来四大高新技术之一<sup>[21]</sup>。2003 年，麻省理工大学技术评论杂志将传感网络总结为改变未来世界的十种新兴技术之一<sup>[22]</sup>。

最早开始新一代智能无线传感网络技术研究的是美国军方，美国国防高级研究计划局（DARPA）于 1978 年在美国的卡内基-梅隆大学召开了分布式传感器网络研讨会<sup>[23]</sup>。DARPA 在 1979 年提出了“分布式传感器网络计划—DSN”。20 世纪九十年代中期，开始了低功率无线集成微型传感器的研究，1998 年 DARPA 又提出了“传感器信息技术计划—SensIT”，这些计划的发起使得人们对无线传感网络系统的兴趣持续增长。同时美国军方在 C4ISR 的基础上提出了 C4KISR 计划，强调战场情报的感知能力、信息的综合能力和信息的利用能力，把无线传感网络作为一个重要研究领域。尤其是 2005 年试验成功的美军“狼群”地面无线传感器网络标志着电子战领域技术的最新突破。

在学术领域，美国自然科学基金委员会(NSF)一期资助 4000 万美元在 UCLA 成立了传感网络研究中心 CESN，联合周边大学展开“嵌入式智能网络传感器”

的研究项目。在 NSF 的推动下，美国加州大学伯克利分校、波士顿大学、加州大学洛杉矶分校、麻省理工学院等学校率先开始了无线传感网络的关键技术和基础理论的研究。例如加州大学伯克利分校成功的研制了传感器操作系统—TinyOS，目前是 2.1.0 release 版本。加州大学洛杉矶分校开发了一个无线传感网络和一个无线传感网络的模拟环境 Avrora，用于仿真模拟无线传感网络各方面的问题。

在工业界，美国英特尔公司（Intel）、微软公司（Microsoft）等信息工业界巨头也开始了无线传感网络方面的工作，并纷纷设立或启动相应的行动计划。例如英特尔公司与加州大学伯克利分校正领导着“智能微尘”（Smart Dust）技术的研究工作。在日本，日立制作所与 YRP 泛在网络研究所开发出了全球体积最小的有源传感器网络终端，它使用锂离子充电电池。此外，加拿大、德国、英国、意大利、韩国和印度等国家的研究机构也对无线传感网络表现出极大的兴趣，相继展开了该领域的研究工作。

我国真正意义上的无线传感网络研究始于 1999 年出现在中科院《知识创新工程试点领域方向研究》的信息与自动化领域研究报告中，并作为该领域提出的五个重大项目之一<sup>[23]</sup>。2001 年中科院依托上海微系统所成立微系统研究与发展中心，并通过该中心在无线传感网络的方向上部署了若干重大研究项目和方向性项目，初步的建立了传感网络系统研究平台，在无线传感网络通信技术、微型传感器、传感器节点等方面取得了很大的进展。2002 年，在国家自然科学基金委的引导下，国内掀起了一个理论研究的高潮，尤其是在 2006 年国家 973 基础研究计划“无线传感网络的基础理论及关键技术研究”的推动下，国内高校对无线传感网络的研究热情空前高涨，清华大学、中国科技大学、南京大学、上海交通大学、浙江大学等纷纷开展了有关无线传感网络方面的基础研究工作。2006 年初发布的《国家中长期科学与技术发展规划纲要》在为信息技术确定的前沿方向中，其中两个与无线传感网络的研究直接相关，即传感器网络及智能信息处理和下一代网络关键技术与服务，这足以看出我国对无线传感网络发展的重视程度。

## 1.2 拓扑控制问题的背景知识

无线传感网络作为近几年的研究热点向广大科研工作者提出了诸多挑战，其中拓扑控制是一个很重要的研究方向。拓扑控制的首要目标是节省节点的能耗以延长传感网络寿命；另外，拓扑控制能够提高 MAC 和路由协议的效率，减少干扰和提高系统吞吐量，并且它可为数据融合、目标定位奠定基础。因此拓扑控制是无线传感网络研究的核心技术之一。

拓扑控制主要的研究内容是在保证网络连通性和覆盖性的前提下，通过功率控制或者骨干节点选择，删除节点之间某些不必要的无线链路，形成一个高效的转发数据的骨干网。拓扑控制算法大体上可以分为功率控制和层次分簇两个方面。功率控制指的是在满足网络连通性的前提下，减少节点的发射功率，均衡节点的一跳邻居数目。层次分簇机制则是按照一定规则选取某些节点担当簇头，选取出的簇头形成转发数据的骨干网，其他非骨干节点在没有任务时就可以关闭其通信模块以节省能量。需要注意的是，由于无线传感网络是面向应用的，不存在某种拓扑控制技术对所有的应用场景都是最优的。因此不同的应用环境就需要不同的拓扑控制技术，这方面的问题会在 1.2.1 节中有详述。

### 1.2.1 拓扑控制问题的分类

不同于文献<sup>[24]</sup>将无线传感网络的拓扑控制按照节点的可移动与否（静止或者动态）和部署的可控（可控或者不可控）进行分类，本文根据节点的可移动与否（静止或者动态）和节点部署的密集程度（密集或者稀疏）分为如下四类：

- 1) 静态节点、密集部署：大量的静态节点部署在给定的区域，这是大部分无线传感网络的基本假设。此时由于分簇机制可以让大部分节点投入到睡眠模式中去以较大的节省节点能量，因此分簇机制是这种配置情况下最主要的拓扑控制机制。
- 2) 静态节点、稀疏部署：这种情况类似于传统的 Ad-Hoc 网络，此时如果采用分簇机制，则会导致产生的簇太小（即簇包含的节点个数太少）从而使分簇算法在数据聚合和能量节省方面的优势得不到发挥，因此，利用功率控制来优化节点邻居个数，减少节点到汇聚节点的时延显得更为重要。
- 3) 移动节点、密集部署：这种情况类似于密集部署的 MANET 网络。在采用功率控制的平面拓扑结构中，不论是采用主动或者被动路由，都会带来大量的计算和通信负载以及端到端时延，因此采用分簇算法以产生层次结构更适合这种情况。
- 4) 移动节点、稀疏部署：这种情况类似于稀疏部署的 MANET 网络。此时如果采用分簇算法，由于节点内在具有的移动性会导致簇结构的极其不稳定和簇头的频繁更换，这样在维护簇的过程中就会带来大量的计算和通讯负载，因此功率控制算法显得更为可行。

### 1.2.2 功率控制

如前所述，所谓的传输功率控制是指在满足一定的网络属性的前提下，例如连通性和网络冗余度，通过减少节点的发射功率以节省网络能耗。通过减少节点的发射功率，一是能够延长传感网络的寿命；二是可以减少节点访问无线信道的竞争。这是因为当节点的传输功率较小时，多个节点能够同时传输而不会发生干扰，这样网络容量也会随之提升。

功率控制通过降低节点的发射功率来延长网络的生存时间，但却没有考虑节点空闲侦听时的能量消耗和覆盖冗余。事实上，无线通信模块在空闲侦听时的能量消耗与收发状态时相当，覆盖冗余也造成了很大的能量浪费<sup>[25]</sup>。除此之外，基于功率控制的拓扑控制算法往往依赖于节点的地理位置信息、方向信息以及信号强度，这些信息的不稳定容易导致所产生的拓扑结构的不稳定。因此传输功率控制比较适合用于部署比较稀疏的无线传感网络。

希腊学者 Kirousis 等人<sup>[26]</sup>将功率控制问题简化为发射范围分配问题，简称 RA(Range Assignment)问题。Kirousis 等人假设网络的节点部署在  $d(d=1,2,3)$  维的边长为  $I$  的区域内。此时把网络  $M_d$  建模成  $M_d=(N,P)$ ，其中  $N$  是节点集合，而  $P$  是节点的位置函数。对于静态网络，定义  $P:N \rightarrow [0,I]^d$ 。此时 RA 问题就可以定义为： $N \rightarrow (0,r_{max}]$ ，也就是给每个节点的发射半径赋一个位于  $(0,r_{max}]$  的一个值，其中  $r_{max}$  是网络中所有节点的最大发射半径。假设赋给节点  $n_i$  的发射半径为  $r(n_i)$ ，此时的 RA 问题就是在保证网络连通性的前提下，使得各个节点的发射功率之和最小，即  $\min \sum (r(n_i))^\alpha$ ，其中  $5 \geq \alpha \geq 2$ 。

根据是否给所有节点赋值相同的发射距离 RA，基于功率控制的拓扑控制可以分为两类：均一功率控制方法（Homogeneous power control）和非均一功率控制方法(Nonhomogeneous power control)。其中前者又称为临界传输距离(Critical Transmitting Range,CTR) 问题，它的计算较为简单，本质上等同于计算最小生成树 MST(Minimum Spanning Tree)的最长边。而非均一的发射功率控制方法根据计算拓扑结构所依赖的信息不同，可进一步的划分为基于位置、基于方向和基于邻居的功率控制算法。如图 1.4 所示。

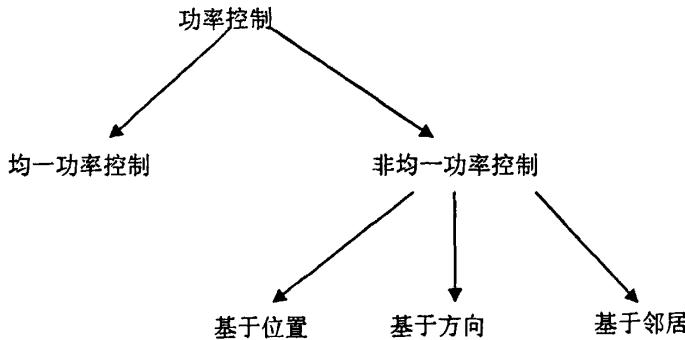


图 1.4 功率控制算法分类

1) 基于位置的功率控制: 这类拓扑控制方法需要事先知道节点的精确地理位置, Rodoplu<sup>[27]</sup>等人提出了一种分布式的依赖地理位置信息的功率控制算法, 以最小化到给定汇聚节点所需要的能量。另外, Li 等人提出了经典的 LMST<sup>[28]</sup>算法, 在 LMST 中, 每个节点首先利用 Prim 算法<sup>[29]</sup>求出其局部的最小生成树, 然后每个节点以到达最远邻居所需的功率作为它自己的最终发射功率。LMST 能够保证所产生的拓扑图是强连通的, 并且每个节点的最大度数不超过 6。

基于位置的功率控制算法所采用的一般过程是<sup>[30,31]</sup>: 给定当所有节点都以最大功率发射时形成的图  $G_R$ , 拓扑控制的过程本质就是裁剪  $G_R$  以形成一个子图  $G$ , (也称路由图), 并且使得  $G$  具有一些良好的拓扑特性, 例如具有常数的能量伸展因子(power stretch factor)和有界的节点度数等。

计算这种路由图  $G$  通常用到计算几何中邻近图(proximity graph)的理论知识, 常见的邻近图有 RNG, GG, Yao Graph, MST, DT 等。对于任意一个节点集合  $V$ , 都有  $MSN(V) \subseteq RNG(V) \subseteq GG(V) \subseteq DT(V)$ 。这几种邻近图的不同体现在计算子图  $G$  的过程中选择邻居的标准不同。例如, 在 RNG(Relative Neighbor Graph)中任意两个节点  $u$  和  $v$  如果存在一条链路, 那么在网络中不存在任意其他一个节点  $w$ , 使得  $\max(d_{uw}, d_{vw}) \leq d_{uv}$ 。而对于 GG(Gabriel Graph)图, 选择的标准是不存在任意一个节点  $w$ , 使得  $d_{uw}^2 + d_{vw}^2 \leq d_{uv}^2$ , 其中  $d_{ab}$  表示节点  $a$  和节点  $b$  之间的欧几里德距离, 如图 1.5 所示的是节点  $u$  和  $v$  之间不存在链路的情形:

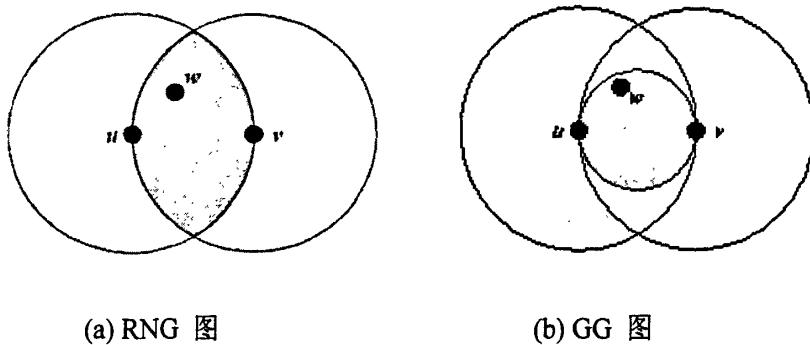


图 1.5 邻近图示例

2) 基于方向的功率控制: 这种方法通常需要节点配备多个有向天线, 以精确的获得可靠的方向信息来解决到达角度问题。微软亚洲研究院的 Wattenhofer 和康奈尔大学的 Li 等人提出了一种能够保证网络连通性的基于圆锥的拓扑控制算法(Cone Based Topology Control, CBTC)<sup>[32,33]</sup>。基本思想是: 节点  $u$  选择最小功率  $P_{u,\alpha}$ , 使得在任何以  $u$  为圆心且角度为  $\alpha$  的锥形区域内至少有一个邻居。并且理论证明了当  $\alpha \leq 5\pi/6$  时, 就可以保证网络的连通性。Borbash 等人<sup>[34]</sup>等人也提出了一种基于方向的分布式拓扑控制协议, 该协议通过计算 RNG 图来保证网络的连通性和其他网络性能。

3) 基于邻居的功率控制: 此类算法的基本思想是保持节点的邻居个数在给定的上限和下限之间, 每个节点动态调整自己的发射功率, 使得节点的度数落在上限和下限之间。这种方法的难点在于确定一个合适的邻居数目  $k$ , 使得产生的拓扑图是连通的。Xue 和 Kumar<sup>[35]</sup>证明了  $k \in \Theta(\log n)$  是保持网络连通性的充分必要条件, 其中  $n$  是节点个数。在 Xue 的基础上, Blough 等人提出了 k-Neighbor 协议<sup>[36]</sup>, 该协议使得节点的邻居个数在等于或者略小于  $k$  的情况下通过删除非对称边来保持网络的对称性。本地平均算法 LMA(Local Mean Algorithm)和本地邻居平均算法 LMN(Local Mean of Neighbor Algorithm)<sup>[37]</sup>也是此类算法的代表。

### 1.2.3 层次分簇

传感器节点的无线通信模块在空闲状态和收发状态下的能耗相当, 因此只有关闭其节点的无线通信模块才能真正有效的降低非工作能耗。层次分簇就是在这一背景下产生的, 分簇算法指的是按照一定规则选取某些节点担当簇头, 选取出的簇头节点形成转发数据的骨干网, 而其他的簇内节点只是完成数据感知的基本任务。分簇算法中除了簇头节点一直保持活动外, 簇内节点在没有任务时可以关闭其通信模块以节省能量。除此之外, 簇头通过对簇内成员所传输的数据进行

数据聚合(data aggregation)，大大减少了数据通信量，这样有利于延长传感网络的寿命。由于簇头节点要管理和协调簇内节点的工作，并且负责数据的融合和转发，能量消耗相对较大，因此分簇算法通常采用周期性重新选择簇头的方法来均衡网络中的节点能耗。分簇算法产生的层次型拓扑具有很多优点，可以减少通信负载和节点能耗、平衡负载、延长网络寿命和增强网络的可扩展性等，因此分簇算法非常适合于大规模部署的无线传感网络。

层次分簇算法的核心是如何选择簇头集合，并把剩余的节点划分到已经产生的簇头集合中。分簇算法有不同的分类标准，例如根据产生的簇是否有簇头可以划分为基于簇头的分簇算法和非基于簇头的分簇算法，根据簇内成员和簇头之间的跳数可以把分簇算法划分为一跳簇算法和多跳簇算法。本文根据<sup>[38]</sup>的分类方法，按照分簇算法要达到的目标对分簇算法进行分类，具体的分类和它们达到的目标见表 1.1。

表 1.1 分簇算法的分类

| 分类      | 分簇达到的目标                                   |
|---------|---|
| 基于支配集   | 建立一个连通支配集来减少参与转发数据的构成骨干网的节点个数             |
| 低维护费用分簇 | 生成一个稳定的簇结构以减少维护簇结构所需要的计算和通信负载             |
| 移动感知分簇  | 把移动节点的移动行为考虑进内，使得分簇算法适用于移动网络环境            |
| 能量有效分簇  | 避免不必要的能耗或者让所有节点轮流担当簇头                     |
| 负载平衡分簇  | 在构建簇的过程中，限制每个簇内节点个数，使得簇头消耗能量均匀            |
| 加权分簇    | 在选举簇头时考虑多种因素，例如节点度、剩余能耗等，并且根据不同的应用调整加权因子。 |

下面对表中所列分簇算法进行示例讲解：

1) 基于支配集的分簇算法：这类算法旨在求解给定网络的最小支配集

(Minimum Dominating Set, MDS) 或者最小连通支配集 (Minimum Connected Dominating Set, MCDS)。如图 1.6 所示, 图中标注的黑色节点构成了最小支配集。

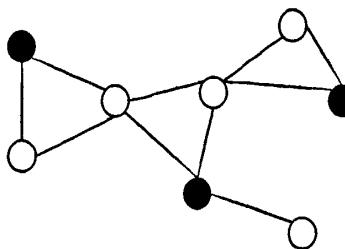


图 1.6 最小支配集示例

求解 MDS 和 MCDS 在理论上是 NP 难问题<sup>[39][40]</sup>, 因此往往借助于近似或者启发式算法来求解。Wu 等人<sup>[41]</sup>提出了一种分布式的计算连通支配集的方法。首先如果某个节点发现自己有至少两个不连通的邻居, 那么就把自己“标注”为支配集的节点; 其次, 如果某个支配集节点发现自己的邻居集合在其某个邻居支配集节点的一跳邻居集合之内, 那么就把自己从支配集中删除, 这样可以在不破坏支配集性质的前提下减少支配集的大小。连通支配集尽管能够很好的提供簇间路由, 但是它产生的簇头的个数往往过大并且簇和簇之间有很大的重叠 (overlapping), 为了解决此问题, Chen 等人<sup>[42]</sup>提出了一种弱的连通支配集 (Weakly Connected Dominating Set, WCDS) 算法, WCDS 包含的节点个数要少于 CDS, 因此更有利于进一步简化网络结构。

尽管基于 MDS 和 MCDS 的分簇算法能够在理论上达到某种最优的特性, 但是两点缺点: 一是所需要的计算和通讯负载较大; 二是当发生某些事件, 例如节点移动或者簇头需要轮换时, 需要重新计算 MDS 和 MCDS, 因此这种方法导致的簇的维护费用过高。

**2) 低维护费用的分簇算法:** 分簇算法带来的一个严重问题是在形成和维护簇结构的过程中, 节点和节点之间需要交换大量的跟分簇有关的消息, 这会消耗大量的节点能量和网络带宽, 从某种意义上会损害分簇算法对网络的可扩展性和性能所带来的优势。低维护费用的分簇算法旨在通过减少簇内成员的从属关系变化 (指的是簇内成员所隶属的簇头发生改变), 尤其是尽最大努力的减少簇头变更所带来的重新分簇行为, 以维护一个稳定的簇结构。

Chiang<sup>[43]</sup>等人提出了一种维护稳定簇结构的 LCC(Least Cluster Change)算法, 它是对传统的基于节点最小标号算法<sup>[44-46]</sup>和节点最大度数算法<sup>[47-49]</sup>的改进。在最小标号和最大度数分簇算法中, 为了维护簇头在其邻居节点集合内的特定属

性，分簇算法需要周期性的重新执行以选出符合条件的新的簇头集合，而这会带来大量的计算和通讯负载。在 LCC 算法中，重新选择簇头是事件驱动的，重新分簇只是发生在两种特定的情形下，因此维持了簇结构的稳定性，减少了维护簇的费用。

Yu<sup>[50]</sup>等人提出了一种称为 3hbac(3-hop Between Adjacent Clusterheads)的算法，不同于以往大部分的分簇算法把网络中的节点划分为簇头和簇内成员，该算法通过额外引入一种称为“Clusterguest”身份的节点来维持簇结构的稳定性。如果某个节点不是和簇头直接邻接，但是它和从属于某个簇头的簇内成员直接相邻，那么此节点就成为这个簇头的“Clusterguest”。3hbac 能够在节点移动的情况下很好的维护了簇的稳定性，例如当两个簇头 A 和 B 彼此进入对方的发射半径之内，此时一个簇头必须放弃自己的簇头身份。假设簇头 A 放弃了簇头身份，此时簇头 A 就变成了簇头 B 的簇内成员，而 A 的簇内成员就变成 B 的“Clusterguest”，这就意味着对其他簇内的所有节点没有任何影响。因此 3hbac 把重新分簇这一全局性的行为变成了局部化，从而维护了簇的稳定性。另外一个类似的通过引入“Clusterguest”身份来增强簇结构稳定性的算法是由 Yu 等人提出的 ECS 算法<sup>[51]</sup>。

**3) 移动感知的分簇算法：**目前的大部分分簇算法通常假设网络是静态部署的，移动感知的分簇算法在簇形成阶段把节点的移动性考虑进内，因此适合于移动的无线传感网络。Basu 等人<sup>[52]</sup>提出了一种 MOBIC 算法，该算法通过计算每个节点和其所有邻居节点的相对移动的方差值，具有方差最小的节点担当簇头，这是因为方差最小的节点意味着此节点相对于其邻居节点来说相对静止。从维护簇结构的稳定性角度来看，此节点适合担当簇头节点。因此 MOBIC 算法非常适合于具有“组移动模型”(group mobility model)的网络，在此移动模型下，一个组之内的所有节点具有相似的速度和方向。但是由于 MOBIC 算法在后续的簇维护阶段没有把节点的移动性考虑进内，因此当节点随机移动时，该算法的性能就会大大折扣。另外一个通过形成多跳簇来更好的适应节点移动所带来的变化，以期形成一个相对稳定簇结构的算法是由 Madonald 等人<sup>[53]</sup>提出的 DDCA(Distributed Dynamic Clustering Algorithm)。

**4) 能量有效分簇：**由于簇头相比普通簇内成员来说要承担更多的任务，例如数据聚合和簇内协作，因此消耗能量的速度也会更快，也更容易导致能量衰竭。因此能量有效的分簇算法旨在通过让每个节点都有担当簇头的机会，把能量消耗更均匀的分布在网络中的所有节点以延长传感网络的寿命。

在这方面，最著名的是由麻省理工学院的 Heizehman 等人提出的低功耗自适应分簇算法 (Low-Energy Adaptive Clustering Hierarchy, LEACH)<sup>[54,55]</sup>。LEACH

协议事先确定一个簇头节点占总节点的比例  $p$ , LEACH 协议提出用轮 (round) 的概念来轮转簇头节点, 一共进行  $1/p$  轮, 每进行一轮就从剩余的未担当过簇头的节点集合中选举出  $np$  个新的节点担当簇头, 其中  $n$  为网络中的总节点数。选举的方法是每个节点随机的生成一个 0~1 之间的一个值, 然后把此值和一个事先确定的阈值函数相比较, 如果小于阈值函数的值就担当簇头。由于 LEACH 协议非常巧妙的选择了阈值函数, 它能够保证经过  $1/p-1$  轮后, 所有仍未选举出来担当簇头的节点成为簇头的概率为 1, 这就保证了在循环中每个节点成为簇头的机会只有一次。在  $1/p$  轮后, 整个过程重新开始。但是 LEACH 协议的缺点是它不能保证簇头的均匀分布, 而且它要求所有的簇头必须经一跳能够到达汇聚节点, 因此 LEACH 只适合小规模的网络。

另外一个简单易行的轮转簇头的算法是由 Amis 等人提出的 IDLBC(ID Load Balancing Clustering)<sup>[56]</sup>。IDLBC 规定每个节点有一个虚拟标号(Virtual ID, VID), 初始情况下, VID 的值等于节点本身的标识符, 同时每个节点都维持一个虚拟标号的循环队列, 每过一个时间单元, VID 的值就左移一次。担当簇头的节点设置一个定时器, 当定时器的值到达事先设置的最长服务时间 max\_duration 时, 此节点不再担当簇头而变为普通的簇内成员, 与此同时把自己的 VID 设置为 0。同时在自己一跳邻居范围内拥有最大 VID 的节点被选举为新的簇头。IDLBC 通过限定簇头的最长服务时间来实现簇头的更换和负载平衡, 但是 IDLBC 有其固有的缺点: 一是为了收集精确的 VID 信息, 节点之间需要非常严格的时间同步。二是当重新选举簇头时, 会导致所谓的“涟漪效应”(ripple effect), 也就是说当一个簇头发生变化时, 网络中所有的节点都要重新参与选举新簇头的过程中去, 这很明显会带来大量的通讯负载, 不利于延长传感器网络的系统工作时间。

**5) 负载平衡分簇:** 在分簇算法中, 簇的大小也是一个很重要的考虑因素。簇太大时容易导致簇头过载、干扰加重以及影响底层 MAC 协议的效率。簇大小将会导致传输数据时较大的端到端时延, 并削弱分簇算法在数据聚合方面所带来的优势。负载平衡分簇尝试在簇大小方面寻求一个折衷, 规定一个簇所能拥有的最大和最小节点个数。

Chatterjee 等人提出了一种 WCA(Weighted Clustering Algorithm)<sup>[57]</sup>并较好的解决了簇之间的负载平衡。WCA 通过预先指定一个簇头所拥有的理想簇内成员个数  $\delta$ , 让每个簇的大小都接近于  $\delta$ , 这样所有的簇头都消耗大致相同的能量。并且本文定义了一个负载平衡因子 (Load Balancing Factor, LBF) 来量化负载平衡的表现能力。它的定义如下:

$$LBF = \frac{n_c}{\sum_i (x_i - \mu)^2} \quad (1.1)$$

其中  $n_c$  代表簇头的个数,  $x_i$  表示簇  $i$  所拥有的节点个数。 $\mu = \frac{(N - n_c)}{n_c}$ , 其

中  $N$  表示网络中的总节点个数,  $\mu$  表示的是一个簇头所能拥有的平均簇内成员个数。很明显, LBF 越大代表簇之间的负载平衡程度就越高。

Ohta 等人也提出了一种负载均衡的分簇算法 AMC(Adaptive Multi-hop Clustering)<sup>[58]</sup>, 该算法对一个簇头所能拥有的簇内成员个数设置了上下界, 分别是  $U$  和  $L$ 。当一个簇头  $CH_i$  发现它的簇内成员个数小于  $L$  时, 此时执行“合并”机制, 簇头  $CH_i$  寻找某个邻居簇头  $CH_j$ , 使得  $|CH_i| + |CH_j| \leq U$  并且使得二者的和最大化, 其中  $|CH_i|$  和  $|CH_j|$  分别表示簇头  $CH_i$  和  $CH_j$  所拥有的簇内成员个数。反之, 当簇头  $CH_i$  所拥有的簇内成员个数大于  $U$  时, 此时它执行“分割”机制, 把  $CH_i$  所隶属的簇分成两个大小近似相等的簇, 并按照规则为“分割”出的新簇选举出合适的新簇头。本算法的实现简单, 但是 AMC 算法并没有给出如何选择  $U$  和  $L$  的值。另外, 本文所提出的 CGCA 算法也可以归类为负载平衡簇, 详见本文第三章。

**6) 加权分簇:** 目前的大多数分簇算法在选举簇头的过程中, 为了达到某种目标往往只考虑一种属性, 例如选举出的簇头具有最小标号、最大节点度或者具有最优的簇内成员个数等。与此不同的是, 加权分簇算法在分簇的过程中, 往往同时考虑多个因素, 例如节点的移动性、剩余能量、簇的大小、簇头和簇内成员的距离等。因此选举出的节点更适合担当簇头, 并且能够根据不同的应用调整加权因子, 具有较好的可扩展性和适应性。

由 Younis 等人提出了著名的 HEED(Hybrid Energy-Efficient Distributed clustering)<sup>[59]</sup> 算法, 该算法在选举簇头时, 综合考虑节点的剩余能量和簇内通信代价, 周期性的通过迭代的方法实现分簇。HEED 用平均最小可达功率 (Average Minimum Reachability Power, AMRP) 作为当某个节点被选举为簇头时的簇内通信代价的度量。AMRP 指的是一个簇头和簇内所有其他节点通信所需要的最小功率的平均值。相比于 LEACH 协议<sup>[54,55]</sup>, HEED 协议由于把节点的剩余能量和簇内通信代价结合起来考虑, 因此具有更好的可扩展性和负载平衡能力, 更有利于延长传感网络的寿命。

前面提到的 WCA<sup>[57]</sup> 算法本质上也是一个加权分簇算法, 在选举簇头的时候综合考虑了簇内节点个数、节点的移动性、节点距其一跳邻居的距离和以及节点担当簇头的时间四个因素, 算法选举具有最小权重的节点担当簇头。Moussaoui 等人<sup>[60]</sup>提出的分簇算法, 在形成簇的过程中, 考虑了簇大小阈值、簇内成员之间的相互距离之和以及节点的剩余能量层级, 因此从本质上也是加权分簇算法。

### 1.2.4 拓扑控制问题的研究现状

随着科技的不断发展和无线传感网络应用的不断深入，有关无线传感网络的理论和应用成果不断涌现。拓扑控制作为重要的理论研究方向之一，国内外也有大量的文献报道了关于它的研究成果。但是这些研究成果所依赖的网络模型往往过于理想化，并且某些算法过于理论化以至于很难满足实用性的要求，因此拓扑控制算法还有诸多需要改进的地方。

当前，拓扑控制的研究主要以最大限度的延长网络的生命周期作为首要设计目标，并兼顾其他方面，比如降低干扰、提高容错性和增强可扩展性等。采用的手段集中于功率控制和层次分簇两种。

基于功率控制的拓扑控制算法侧重的是赋予每个节点一定的传输功率（相同或者不同），使得产生的拓扑图按照某个标准是最优的，它们重视的是产生拓扑图的质量而不是具体的构建拓扑结构的过程。这些拓扑控制算法往往有很强的前提假设，例如要求节点有精确的地理位置信息或者方向信息，而现实的传感器节点的硬件配置通常很难达到这些要求，因此这些算法通常具有较强的理论指导意义而缺乏实用性。而另一类拓扑控制算法，例如 XTC<sup>[61]</sup>，侧重于设计简单可行的完全分布式的协议来构建和维持一个合理的拓扑，代表了拓扑控制研究的新方向。

基于层次分簇的拓扑控制算法侧重的是怎样选择和优化簇头，目前的分簇算法往往过分强调簇头的属性或者分簇的过程，而忽略了更重要的簇结构的后期维护费用。因此将来的分簇算法应该把簇的形成和维护作为一个整体来考虑，不仅要使形成的簇满足给定的属性，而且要满足后期的簇维护费用要低。要到达此目标就必须要消除重新分簇所可能带来的“涟漪”效应，也就是说簇头的重新选举要异步的触发，以维护一个较为稳定的簇结构。

国内对于无线传感网络研究起步较晚，在拓扑控制算法方面的研究工作还较少。目前，已有若干科研机构和相关大学开展了该领域的研究工作，例如南京大学、东南大学、西安电子科技大学，并且已经取得了一定的研究成果<sup>[25,62-65]</sup>。

另外，目前的大部分关于拓扑控制的研究成果都是针对静态网络的，造成这一状况的原因是移动网络中节点本身所固有的移动性使得底层的拓扑结构时刻在发生变化。而确定性的基于功率控制或者层次分簇的拓扑控制算法很难适用于这种不确定性的拓扑。因此可以预见，应用随机算法或者近似算法来解决移动传感网络的拓扑控制问题会成为将来的研究热点。

### 1.3 本文的主要工作及安排

针对现有拓扑控制研究工作存在的缺陷，本文做了相应的工作，主要工作如下：

- 1) 研究了在重新分簇过程中怎样减少计算和通讯负载的问题，提出了基于完全图的分簇算法 CGCA，此算法能够消除重新分簇可能带来的“涟漪”效应并维护簇结构的稳定性，以延长传感网络寿命。
- 2) 研究了在无线传感网络中加入移动节点来承担负载较重的任务，通过利用移动节点的灵活性和能量充裕的特点，以延长传感网络系统时间的问题。本文提出了在上述 CGCA 分簇算法基础上通过加入移动节点来担当簇和簇之间的网关，以进一步的节省普通节点的能量并延长网络寿命。
- 3) 研究了在移动传感网络的拓扑控制问题，并在给定的移动网络模型 VRMN(Variant Rate Mobile Network)下提出了一种简单易行并且能量有效的基于功率控制的拓扑控制算法。
- 4) 研究了在大规模移动传感网络下的拓扑控制问题，并提出了一种能量有效的分簇算法来较好的解决了当节点移动时的簇维护问题。

本文的组织如下：

第一章为本文的绪论部分，介绍了本文的背景知识和主要工作。

第二章描述了拓扑控制问题的设计因素和基本模型。

第三章描述了一种在密集部署传感网络中基于完全图的分簇算法 CGCA，并对算法做了仿真实验。

第四章描述了在 CGCA 分簇算法基础上加入移动节点来担任簇和簇之间网关的问题，以进一步的节省节点能量并延长传感网络寿命，并对算法进行了仿真实验。

第五章描述了在给定的移动网络模型 VRMN 下，提出了一种基于功率控制的拓扑控制算法，并进行了相应的仿真实验。

第六章描述了在大规模密集部署的移动传感网络中，提出了一种能够保持簇结构相对稳定的分簇算法，并对算法性能进行了仿真验证。

第七章对本文做了总结，并对拓扑控制的前景进行了展望。

## 第2章 设计因素和基本模型

无线传感网络拓扑控制算法的设计与具体的应用和网络配置密切相关，针对不同的应用需求，要设计相应的拓扑控制协议，这样才能使得无线传感网络高效运行。

本章首先介绍了拓扑控制算法的设计因素，然后介绍了本文研究的拓扑控制算法所采用的基本模型。

### 2.1 设计因素

设计无线传感网络的拓扑控制算法时，需要考虑很多的制约因素，包括：网络寿命、连通性、覆盖、节点的能量消耗、容错性、可移动的鲁棒性、干扰和竞争、吞吐量、延迟、伸展因子、传输介质、算法特性和移动模型等，这些都是无线传感网络拓扑控制算法的性能衡量指标或影响因素。

#### 2.1.1 网络寿命

在许多的无线传感网络应用中，节点的工作依赖具有有限能量的电池供电，因此在工作环境中更换电池或者对其进行充电通常是不实际的。同时，无线传感网络的应用通常要求它必须保证至少工作一段时间或者尽可能的延长其系统寿命。因此，延长网络寿命是拓扑控制算法设计所需要考虑的第一要素。网络寿命有多种定义，有时把它定义为从开始运行到第一个节点能量耗尽所经历的时间，但通常将网络寿命定义为死亡节点的百分比低于某一指定阀值的持续时间。当考虑网络的服务质量(QoS)时，可以认为网络只有在满足一定的连通质量、覆盖质量或者某些其他的服务质量时才是存活的。功率控制和分簇算法是延长传感网络寿命的十分有效的技术。最大化的延长传感网络寿命是一项十分困难和复杂的任务，它一直是拓扑控制研究的首要目标。

#### 2.1.2 连通性

如前所述，无线传感网络是多跳的自组织网络，网络中的节点需要将感知到的数据经过多跳的方式路由至汇聚节点，因此连通性是拓扑控制算法必须保证的基本要求。拓扑控制一般要求生成的拓扑图是1连通的，也就是说在拓扑图中的任意两个节点 $u$ 和 $v$ 中都有一条路径。而个别的应用，还可能要求是 $k$ 连通的，这就意味着在生成的拓扑图中，对于任意两个节点 $u$ 和 $v$ 来说，当且仅当存在连

接  $u$  和  $v$  的  $k$  条路径，使得每对路径只有  $u$  和  $v$  两个公共点<sup>[66]</sup>。 $k$  连通主要是为了增强网络的容错性。在移动网络中，有时也讨论渐近意义上的连通，亦即当部署区域或者网络中的节点个数趋于无穷大时，网络连通的概率趋于 1。

### 2.1.3 覆盖性

覆盖控制作为无线传感网络中拓扑控制的一个基本问题，反映了网络所能提供的“感知”服务质量，可以使无线传感器网络的空间资源得到优化分配，进而更好地完成诸如环境感知、信息获取和有效传输等任务<sup>[25,67]</sup>。覆盖问题可以分为区域覆盖、点覆盖和栅栏覆盖(barrier coverage)等。跟拓扑控制相关的覆盖问题主要研究连通性覆盖，它同时考虑了无线传感网络的覆盖能力和网络连通性这两个相互联系的属性。连通覆盖问题所要解决的是如何同时满足网络的给定传感覆盖和通信连通性需求，这对于一些要求可靠通信的应用至关重要。根据具体的连通性要求，连通性覆盖又可具体分为两类：活跃节点集连通覆盖<sup>[68]</sup>和连通路径覆盖<sup>[69]</sup>。

### 2.1.4 节点的能量消耗

传感网络与传统无线自组网络有着显著的区别。Ad-Hoc、MANET 和蓝牙等无线网络首要设计目标是提供高性能的服务质量(Quality of Service, QoS)，由于节点可以不断的获得电能补充，节点的能量考虑放在次要位置。而传感器节点是个小型的电子设备，目前基本上是由能量受限的电池来供电，因此存在严重的能量约束。如何高效使用节点的能量以延长网络系统的生存期成为传感网络的首要设计目标。因此能量管理和维护对传感网络十分重要，由于传感器节点是传感网络的基本构成组件，怎样设法减少单个节点的能耗就显得尤为重要。

一个传感器节点通常由传感单元、处理单元、收发单元和电池等四部分组成。传感单元通常有两部分组成：传感器和模数转化器(A/D)，传感器对环境进行感应，产生模拟信号，经模数转化器生成数字信号，再经过数字编码转到处理单元与来自其他节点的数据进行融合、安全加密处理，并计算路由且经收发单元发送到下一跳。传感单元的能耗与产生的数据成正比关系，因此对传感器节点产生的数据进行聚合和压缩显得非常重要。处理单元中用于数据存储和处理的能耗微乎其微，例如 Mica 节点在闪存上读取数据的能耗是 1.111nAh，而写入数据的消耗是 83.33nAh<sup>[14]</sup>。

随着集成电路工艺的飞速发展，低功耗的处理器技术日益成熟，使得传感器节点实际工作能耗主要集中在用于无线通信的收发单元上。在文献<sup>[70]</sup>叙述中，

当无线电传输模型服从四次方路径衰耗的瑞利衰落时，在100米距离上传输1000字节的数据所消耗的能量大概与在100MIPS/W处理器上执行三百万条指令所消耗的能量相当。

无线收发单元具有四种运行状态<sup>[71]</sup>:

- 1) 发送: 此时无线收发单元的发送部分处于工作状态, 天线辐射能量。
- 2) 接收: 此时无线收发单元的接收部分处于工作状态。
- 3) 空闲: 此时无线收发单元处于准备接收, 但是还没有接收到分组的状态。  
在空闲状态下, 接收电路的许多部分仍处于工作状态, 而其他部分可以断电。因此, 处于空闲状态的能量消耗和发送或者接收状态相差不大。
- 4) 休眠: 在此状态下, 无线收发单元的关键部分是断电的, 因此节点处于休眠状态时最节能。

表2.1列出了无线收发单元在各个状态下的能耗:

表 2.1 无线收发单元在不同状态下的能耗

| 无线收发单元的状态 | 功耗(mW) |
|-----------|--------|
| 发送        | 14.88  |
| 接收        | 12.50  |
| 空闲        | 12.36  |
| 休眠        | 0.016  |

从表2.1可以看出, 传感器节点在空闲状态下消耗的能量略小于其在发送和接收状态下的能耗。因此从节省节点能量和延长传感网络寿命的角度来说, 尽量要让传感器节点更多的时间投入到“休眠”状态, 这也就是分簇算法要把大部分的节点投入到休眠状态的事实依据。

假设节点的无线通信距离为 $d$ , 无线电的路径衰减指数为 $\beta$  ( $2 \leq \beta \leq 5$ ), 无线电发送设备的发射功率 $P_T = \alpha_1 + \alpha_2 d^\beta$ , 其中 $\alpha_1$ 和 $\alpha_2$ 是固定的常数。无线电接收设备的功率为 $P_R$ , 其它的无线电功率为 $P_0$ , 其中 $P_R$ 和 $P_0$ 是常数。所以总功率 $P = P_T + P_R + P_0 = \alpha_1 + \alpha_2 d^\beta + P_R + P_0$ , 单位距离的功率 $P/d = \alpha_2 d^{\beta-1} + (\alpha_1 P_R + P_0)/d$ , 当 $d = [(\alpha_1 + P_R + P_0)/(\alpha_2 \beta - \alpha_2)]^{1/\beta}$ 时,  $P/d$ 取最小值。也就是说在理论上, 无线通

信距离  $d = [(\alpha_1 + P_R + P_0) / (\alpha_2 \beta - \alpha_2)]^{1/\beta}$  时，进行长距离通信所消耗能量最小。

节点的无线通信距离的设计还取决于其它因素，如应用环境、连通度等。例如应用环境对  $\beta$  的值影响很大，当部署环境恶劣、节点贴近地面时，此时干扰较大从而导致  $\beta$  的值较大。当  $\beta$  取较大的值时，节点的发射功率随着距离  $d$  的增加而增加，节点的能耗也急剧增加。因此，应该在满足网络连通性的前提下，减少节点的单跳通信距离，这也就是基于功率控制的拓扑控制算法的基础。即通过减少节点的发射功率，进而减少节点的单跳邻居集合，最后实现了减少节点的单跳通信距离。当某些应用要求系统具有较大的连通度时，此时节点应该增大无线电发送功率，以提高连通度，但与此此时节点能耗也增加较快。

### 2.1.5 容错性

无线传感网络中的传感器节点可能由于电池耗尽、外界破坏或者环境影响等原因而失效，容错性就是指一个或者多个传感器节点的失效并不能影响整个无线传感网络的正常运行，从而维持传感网络的运行能力。

容错性的要求和具体的应用密切相关。如果传感节点布置在受外界影响比较小的区域内，那么对容错性的要求就较低。例如，在楼宇内布置传感节点，以监测室内温度、湿度和空气流通等物理参数的变化，对容错性的要求就比较低<sup>[8]</sup>。相反，如果在战场布置传感器节点用于军事监测，对容错性的要求就非常高。在实际应用中，通常通过部署大量冗余节点的方法来达到容错性。在理论上，容错性和连通性密切相关，本质上是通过维持网络的  $K$ -连通来解决( $K \geq 2$ )。

### 2.1.6 可移动的鲁棒性

可移动的鲁棒性(robustness)意味着当节点的移动、失效或者无线信道的变化导致节点的拓扑邻近关系发生时，原始的拓扑结构不需要经过大量的调整，也就是说节点之间只需要交换少量的信息就可以适应新的拓扑结构。例如，在基于分簇的拓扑控制算法中，消除重新分簇所带来的“涟漪”效应(ripple effect)是可移动鲁棒性的重要体现。而在基于功率控制的拓扑控制算法中，可移动的鲁棒性主要体现在两个方面：一是拓扑控制算法要能够适用于移动的无线传感网络；二是当节点移动时，拓扑控制算法体现在时间复杂度和消息复杂度方面应该是高效的。

### 2.1.7 干扰和竞争

减少干扰和竞争是拓扑控制算法所力求的重要设计目标。功率控制通过减少节点的发射功率来减少通信干扰和 MAC 层的竞争，这是因为减少功率可以减少节点的发射半径，而减少半径就可以减少竞争的区域大小。而层次分簇是通过将尽可能多的簇内成员节点投入到睡眠模式来减少干扰和竞争。这两种拓扑控制算法的本质都是通过减少节点的一跳邻居个数来减少干扰和竞争。近年来，基于干扰感知(interference-aware)的拓扑控制算法<sup>[72,73]</sup>已经成为该领域内的研究热点。

### 2.1.8 吞吐量

吞吐量是衡量无线传感网络工作能力在系统级别上一个很重要的指标，也是拓扑控制算法的一个重要设计目标。假设目标区域是一个凸区域，每个节点的吞吐率为  $\lambda$  bit/s，在理想情况下，有下面的关系式<sup>[25,74]</sup>：

$$\lambda \leq \frac{16AW}{\pi\Delta^2L} \cdot \frac{1}{nr} \quad (2.1)$$

其中， $A$  是目标区域的面积， $W$  是节点的最高传输速率， $\pi$  是圆周率， $\Delta$  是大于 0 的常数， $L$  是源节点到目的节点的平均距离， $n$  是节点数， $r$  是理想球状无线电发射模型的发射半径。从公式 (2.1) 可以看出，功率控制通过减少节点的发射半径  $r$  可以达到提高系统吞吐量的目标，而层次分簇通过减少活动节点的个数  $n$  也可以达到同样的目的。

### 2.1.9 延迟

在无线传感网络中，延迟与以下几个因素密切相关：应用环境、带宽、MAC 层协议以及具体的拓扑控制协议相关。应用环境对延迟的要求相差很大，对生态环境的监测，例如对岛上鸟类栖息地的观测<sup>[14]</sup>等应用只需要周期性的将监测数据传输至观测点，因此延迟并不是放在系统考虑的重要因素。而有的应用，例如智能交通或者灾难救援的应用则对延迟的要求很高，因为高延迟将会导致大的误差，此时延迟性是系统设计的首要因素。

其他的影响因素方面：在同等的条件下高的带宽往往意味着低的延迟。由于无线传感网络的能量受限特性，MAC 层的协议设计为了减少干扰和节点的能耗，通常在节点没有任务时让节点“休眠”而在有传输任务时“唤醒”节点，这就是所谓的 MAC 协议唤醒机制<sup>[75-78]</sup>。但是没有一个 MAC 协议能够保证每当节点有任务时都能够“恰好”的唤醒节点，因此当不能满足此条件时就会导致比较大的

延迟。

拓扑控制算法和延迟之间往往存在一个性能折衷：在功率控制算法中，大的传输功率往往意味着在源节点和汇聚节点之间只需要较少的跳数，因此延迟较小但能量消耗则较大。而降低传输功率在取得能量有效性的同时却增加了跳数，因而导致了较大的延迟。在分簇算法中，大的簇（含有簇内成员较多）意味着能够将更多的节点投入到睡眠状态，因此能够较大的节省节点能量。但与此同时大的簇意味着同一个簇内，参与信道竞争的节点个数增加，进而导致比较大的延迟。而小的簇虽然能够取得较小的延迟，但是分簇算法所带来的数据聚合等方面的优点却得不到体现。

### 2.1.10 伸展因子

在基于功率控制的拓扑控制算法中，伸展因子（stretch factor）是考察算法能量有效性的一个很重要的指标。其中能量伸展因子  $P_s$  定义为在原始图  $G$  和拓扑控制图  $G'$  之间，任意节点对  $u$  和  $v$  之间消耗能量最小的路径上消耗能量之比的最大值：

$$P_s = \max_{u,v \in N} \frac{E_{G'}(u,v)}{E_G(u,v)} \quad (2.2)$$

其中  $E_G(u,v)$  和  $E_{G'}(u,v)$  分别表示节点  $u$  和  $v$  在图  $G$  和  $G'$  中的消耗能量最小的路径上所消耗的能量。以此类推，跳伸展因子  $H_s$  被定义为在原始图  $G$  和拓扑控制图  $G'$  之间，任意节点对  $u$  和  $v$  之间最短路径上跳数的最大增长：

$$H_s = \max_{u,v \in N} \frac{H_{G'}(u,v)}{H_G(u,v)} \quad (2.3)$$

其中  $H_G(u,v)$  和  $H_{G'}(u,v)$  分别表示节点  $u$  和  $v$  在图  $G$  和  $G'$  中的最短路径所包含的跳数。

### 2.1.11 传输介质

传感器节点之间进行协作通信的前提条件就是选择合适的传输介质，连接节点的无线介质可能是无线电、超声波、可见光，以及适用于特殊场合的其他媒体（例如磁感应），选择哪种无线介质是由依据应用场合的特点而决定的。

到目前为止，多数传感节点都采用无线电作为传输介质，这是因为它满足大部分无线传感网络的应用要求：例如，远程和高数据率，以及在合理功耗下容许

的误码率，并且它不需要发射器和接收器之间的视距路径。对于实际应用的无线电通信，必须选择适合的载波频率，可以选择用作工业、科学及医疗的免费波段，目前无线传感网络的典型通信频率在 433MHz~2.4GHz 之间<sup>[79]</sup>。

### 2.1.12 算法特性

由于无线传感网络的自组织以及无人值守特性，因此拓扑控制算法应该用分布式来实现，也就是说每个节点应该根据自己以及与其一跳或者两跳远的邻居信息来独立的做出自己的决定。另外，拓扑控制算法的总开销也是一个必须考虑的方面，这主要体现在时间复杂度和计算复杂度上。时间复杂度是算法性能优劣的最直接体现。因此为了可实现性，一个好的拓扑控制算法应该是多项式时间复杂度。另外，计算复杂度通常体现在交换的消息个数，这在分簇算法中的重新分簇（re-clustering）阶段体现的尤其明显，因此应该极力消除由重新分簇所带来的“涟漪”效应。

### 2.1.13 移动模型

在设计移动传感网络（Mobile Sensor Networks, MSN）的拓扑控制算法时，移动模型是其考虑的首要前提，同一个拓扑控制算法，在不同的移动模型下的性能表现也会差距较大。下面，只简单的介绍实体移动模型（entity mobility model）<sup>[80]</sup>，而不考虑组移动模型（group mobility model）。

- 1) 随机路点模型（Random Waypoint Model）<sup>[81-84]</sup>: 由于这种模型的简单性以及能够较好的模仿节点的真实运动，因此它是一种应用的非常广泛的移动模型。在这种模型中，节点首先在部署区域中的某个位置停顿某个指定的时间，当此时间到期，此节点随机在部署区域选择一个目的节点。然后，此节点随机均匀的在区间[ $Minspeed, Maxspeed$ ]中选择一个移动速度并以此速度移动到目的节点。当节点移动到目的节点之后，再停顿某个指定的时间。当此时间期满，节点再重复同样的过程。
- 2) 随机方向模型（Random Direction Model）<sup>[85]</sup>: 随机路点模型有一个缺点，这就是它会带来“中心聚集效应”，即位于部署区域中间的节点较为密集。这是由于在目的地的选择中，位于中心位置的目的地或者需要穿越中心位置到达的目的地的概率非常高。为了克服这一缺点并且保持一个相对稳定的邻居数，Royer 等人提出了随机方向模型。在该模型中，节点不是随机选择一个目的地而是随机选择一个方向，然后沿此方向移动到边界。到达边界之后节点随机的停顿一段时间，然后随机在 0 到 180 度之间选

择一个新的方向重复上述的过程。

- 3) 随机走模型 (Random Walk Model)<sup>[86]</sup>: 在随机路点模型和随机方向模型中, 节点的前后两次移动均有一定的联系并且节点在新的移动之前需要一定的停顿时间。为了克服移动之间的关联性并且让节点的运动更为随机, 提出了随机走模型。在随机走模型中, 节点从当前的位置以随机的速度和方向到达一个新的位置, 新的速度和方向从预先定义的区间选择, 分别是 $[speedmin, speedmax]$ 和 $[0, 2\pi]$ 。当到达边界时, 它就以来时的角度所确定的方向反弹回去, 并沿着新的路径重复同样的过程。由于随机走模型在下一次的移动过程中并不考虑以往的位置和速度, 因此会出现一些不太现实的移动方式, 例如急转或者突然停止。

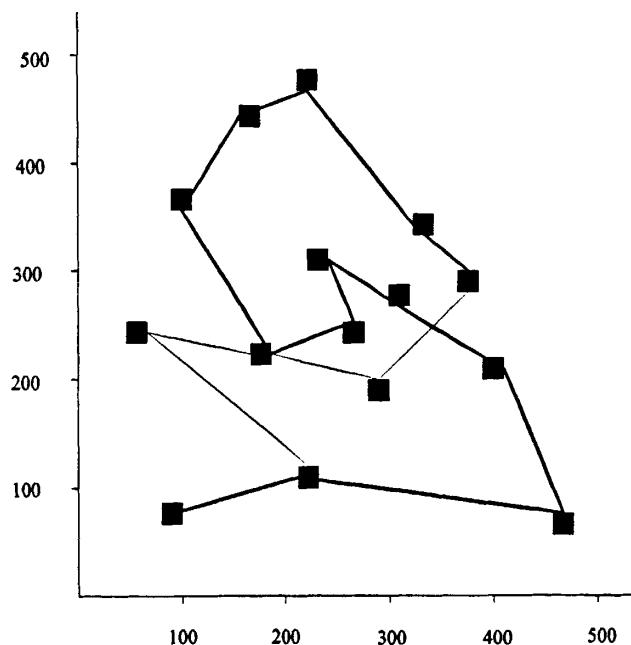


图 2.1 随机路点模型下节点的移动模式

## 2.2 基本模型

无线传感网络拓扑控制问题的设计与具体的应用密切相关, 针对不同的应用需要设计与之对应的协议, 这样才能使得拓扑控制发挥其优化拓扑、节能节点能耗和延长网络寿命的特长。下面简要介绍本文要研究的无线传感网络拓扑控制问题的基本模型:

- 1) 部署区域为二维平面区域;

大部分无线传感网络的应用场景都可抽象为二维平面区域，如公路、建筑、园林、战场等均可近似为二维的平面区域。本文的所有工作都是针对二维平面所作的研究，这是因为，针对二维平面模型的协议具有代表性和普遍性，本文所提出的基于平面区域的拓扑控制算法可以稍加修改就可应用于一维线性模型和三维空间模型上。

2) 节点有一个相同的、固定的、圆形的无线通信范围；

传感器节点的无线通信范围与无线电发送设备的发射信号强度、周围环境、干扰情况等有关。我们假设传感器节点的发送信号强度是各向同性（homogeneous）的，并忽略各区域间由干扰造成的差异，节点的无线通信区域是圆形，半径为节点的最大无线传输距离，并且假设所有的节点具有相同的最大传输距离。因此，我们将节点的通信区域抽象为单位圆盘图（Unit Disk Graph, UDG），如图 2.1 所示：

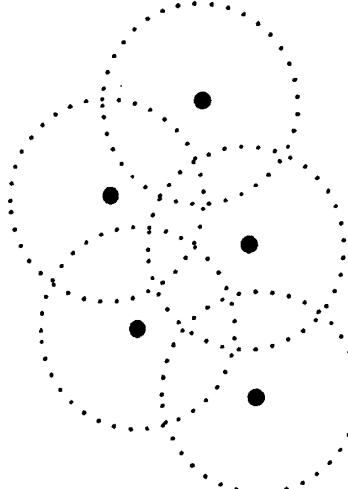


图 2.2 无线传感网络的单位圆盘图

3) 无线传感网络抽象为无向图；

拓扑控制的很多问题最终需要借助于图论的理论来解决，通常我们将无线传感网络抽象为无向图  $G=(V,E)$ ，其中  $V$  代表传感器节点的集合， $E$  代表节点之间链路的集合，也就是说，对于任意两个节点  $u$  和  $v$ ，如果他们彼此都在对方的通信半径之内，那么就将它们之间的链路加入  $E$  中。我们的模型只考虑无向边不考虑有向边，这是因为有向边不利于 MAC 层协议的设计，例如有向边使 RTS/CTS 确认机制变得非常困难。

4) 统一的仿真工具 OMNET++；

在对我们提出的拓扑控制算法进行性能验证时，采用了统一的仿真工具

OMNET++( Objective Modular Network Test bed in C++)<sup>[87]</sup>。它是一种开源的基于组件的模块化的开放网络仿真平台，是近年来在科学和工业领域里逐渐流行的一种优秀的网络仿真平台。OMNeT++作为离散事件仿真器，具备强大完善的图形界面接口和可嵌入式仿真内核，同 NS2<sup>[88]</sup>等其他仿真平台相比，OMNeT++可以简便定义网络拓扑结构，具备编程，调试和跟踪支持等功能。选择 OMNET++一个很重要的原因是它专门提供了一种描述网络拓扑的 NED 语言，采用它可以比较容易完成一个网络模型的描述。网络描述包括下列组件：输入声明、信道定义、网络定义、简单模块和复合模块定义。

5) 拓扑控制算法以节省节点能耗、延长传感网络寿命为第一目标：

本文提出的方案都是以优化网络拓扑、减少节点能量消耗并藉此延长系统生命周期为出发点的，而某些应用可能在此基础上还要求减少干扰等特性。某些特殊应用可能对系统的其他属性看得更为重要，例如智能交通系统对准确性、延迟性的要求比系统寿命的要求更高，本文对此不作研究。

## 第3章 基于完全图的分簇算法

为了消除分簇算法中的重新分簇所带来的“涟漪效应”，本文提出了在密集部署传感网络中一种基于完全图的分簇算法 CGCA(Complete Graph-based Clustering Algorithm)<sup>[89]</sup>。利用完全图内节点的等价性，在重新选举簇头阶段，簇头只需要在每个簇的内部节点之间进行轮换，从而使的重新分簇这一全局触发行为变为局部触发。因此 CGCA 的通信和计算负载可以大量减少，这样便可以减少无线传感网络中的能量消耗，从而延长网络的生命周期。

本章首先介绍本文算法的背景知识及相关工作，然后提出基于完全图的分簇算法，最后对此算法的性能进行仿真验证。

### 3.1 背景介绍

如绪论部分所述，执行层次分簇算法将节点集合划分为簇头集合和簇内成员集合。选取出的簇头形成转发数据的骨干网，因此在分簇机制中除了簇头节点一直保持活动外，簇内节点在没有任务时可以关闭其通信模快以节省能量。除此之外，由于簇头节点要管理和协调簇内节点的工作，并且负责数据的融合和转发，能量消耗比普通簇内成员节点大的多，因此分簇算法通常采用周期性重新选择簇头的方法来均衡网络中的节点能耗。由于分簇算法可以减少通信负载和节点能耗，延长网络寿命和增强网络的可扩展性，因此分簇算法非常适合于大规模部署的无线传感网络，本文的工作也正是以此为前提提出的。

重新分簇一方面平衡了网络中的能量负载，但是从另一方面却容易引起“涟漪效应”(ripple effect)中，也就是说，一个簇头的更换会导致网络中所有的节点都参与到重新选举簇头的过程中，从而导致了大量的计算和通信负载。目前的分簇算法大部分侧重的是簇头的产生和优化，而对后期的簇结构的维护则关注较少。例如传统的基于节点标号<sup>[45]</sup>和基于节点度<sup>[49]</sup>的分簇算法，尽管产生初始簇头的过程非常容易，但是在后期的重新选举簇头过程中，由于要严格维护簇头在其一跳邻居内所具有的属性，即具有最小标号或者最大节点度的特性，因此当一个簇头不满足此条件时，网络中的所有节点都会参与到重新选举新的簇头过程中去，尽管其他的簇头可能此时不需要更换。本文的工作就是以此为背景而设计的。

本文提出了一种在密集部署传感网络下的一种基于完全图的分簇算法 CGCA，算法形成的簇结构是完全图。由于完全图中节点的等价性，因此 CGCA 只需要在系统启动的时刻执行分簇算法，而在以后重新选举簇头时只需要在每个簇的内部局部进行，而不是像以往的分簇算法进行全局性的重新触发，因此这大

大减少计算和通讯负载。并且 CGCA 重新选举簇头不再象以往算法那样周期性的进行，而是由事件进行触发，这样可以极大的维护簇结构的稳定性。除此之外，CGCA 通过优先选取距离簇头较近的节点加入簇中使得簇头比较均匀的分布在部署区域。通过减少计算和通讯负载以及在部署区域内均匀分布簇头，CGCA 能够减少节点能耗并促进延长传感器网络寿命。最后，CGCA 通过限制簇的大小来均衡簇头负载以及提高底层 MAC 协议的效率。

### 3.2 相关工作

在我们的 CGCA 算法中，产生的每个簇都是一个完全图，而完全图内所有的节点都是等价的。实际上，最早提出“等价类”思想的是由 Xu<sup>[90]</sup>等人提出的基于地理位置的分簇算法 GAF (Geographic Adaptive Fidelity)。它的思想是把整个部署区域划分成多个虚拟网格，使得每个网格内的节点都能够与相邻网格内的节点直接通信。这样，从路由的分组转发角度来看，同一个单元格内的节点可以认为是等价的，因此在一个单元格内只需要选择出一个节点担当簇头并保持活动状态，其他节点就可以睡眠以节省能量。

GAF 算法的执行分为两步：一是虚拟单元格的划分；二是单元格内簇头的选择。假设每个单元格都是边长为  $r$  的正方形，而已知节点的最大传输半径为  $R$ ，第一步的关键就是由  $R$  来确定  $r$  的大小。由于前提是相邻单元格内的任意两个节点都能够直接通信，这就意味着相邻的两个单元格内距离最远的两个节点也都在彼此的通信范围之内，因此有下面的公式，如图 3.1 所示：

$$(2r)^2 + r^2 \leq R^2 \Rightarrow r \leq \frac{R}{\sqrt{5}} \quad (3.1)$$

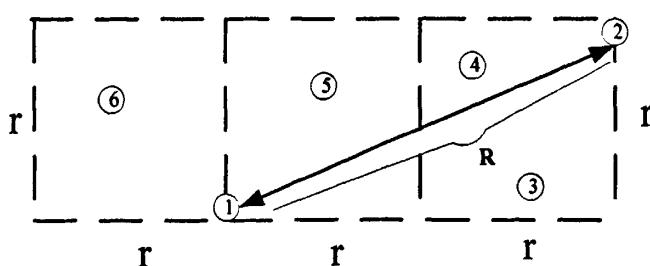


图 3.1 GAF 算法中网格边长的确定

GAF 算法中在第二阶段选举簇头的过程中采用的是随机选择簇头的方法。

每个节点可以处于发现（Discovery）、活动（Active）和睡眠（Sleeping）三种状态。起始情况下，每个节点都处于发现状态。节点和节点之间互相交换信息，包括节点的  $id$ ，网格  $id$ ，以及自己的地理位置信息等，经过这一阶段，每个节点都能够知道和其位于同一个单元格内部的其他节点的信息。为了竞争簇头，每个节点设置一个随机的定时器  $T_d$ ，一旦定时器超时，此节点发送消息宣布自己成为簇头节点并立刻进入活动状态。当节点在定时器超时之前收到其他节点的簇头声明消息时，证明竞争簇头失败并立刻进入睡眠状态。成为簇头的节点设置一个活动定时器  $T_a$ ，它代表簇头节点可以处于活动状态的最长时间，当  $T_a$  超时节点立刻进入发现状态。设置  $T_a$  主要是为了网络的负载平衡，不让某个节点长时间的担当簇头，而是把担当簇头的机会轮转到每个节点。处于睡眠状态的节点设置一个睡眠定时器  $T_s$ ，当  $T_s$  超时节点自动进入发现状态。处于活动或者发现状态的节点，如果按照某种节点的排序规则(ranking order)发现某个节点更适合担当簇头，则此节点自动进入睡眠状态。GAF 算法的节点状态转换图如图 3.2 所示：

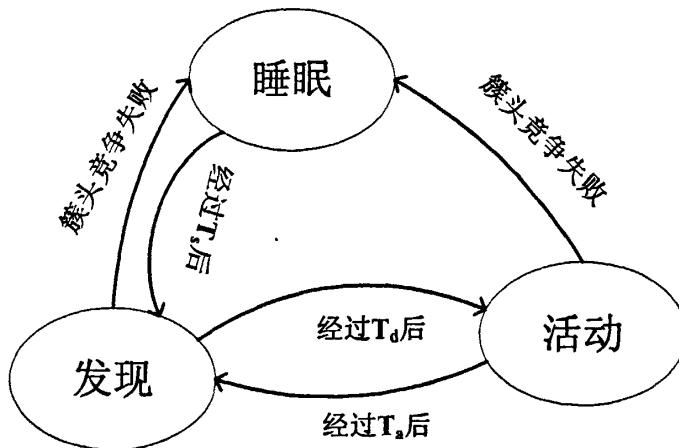


图 3.2 GAF 算法中节点状态转换图

从上面的论述可以看出，GAF 利用了等价类的特点能够让单元格内的大部分节点保持在睡眠状态，从而节省了能量。但是 GAF 也有其内在的不足：一是在选举簇头的时候没有考虑节点的剩余能量；二是在选举簇头和更换簇头的时候，需要节点之间交换大量的信息并且更换簇头的频率较高；三是没有考虑簇的大小，因此簇头的负载非常的不均衡。我们提出的 CGCA 能够克服上面的不足之处，详见 3.3.

### 3.3 CGCA 算法

本节我们提出 CGCA 算法来解决重新分簇所带来的高负载问题，并给出了算法详细的过程、性能理论证明以及具体的实例等。

#### 3.3.1 网络模型

首先定义一些本文使用的记号和术语如表 3.1 所示：

表 3.1 本文定义的一些符号和术语

| 符号       | 定义                   | 符号              | 定义              |
|----------|----------------------|-----------------|-----------------|
| $N(i)$   | 节点 $i$ 的邻居集合         | $\delta$        | 簇的大小阀值          |
| $N_m(i)$ | 节点 $i$ 的第 $m$ 个最近邻居  | $tx_{range}(i)$ | 节点 $i$ 的传输半径    |
| $V$      | 网络的节点集合              | $Num(ch_i)$     | 被簇头 $i$ 覆盖的节点个数 |
| $C$      | 一个簇的节点集合             | $\alpha$        | 与节点临近性相关的权重     |
| $d(i,j)$ | 节点 $i$ 和 $j$ 的欧几里德距离 | $\beta$         | 与节点剩余能量相关的权重    |
| $E(i)$   | 节点 $i$ 的剩余能量         | $ S $           | 集合 $S$ 的势       |

下面简要对其中一些参数进行说明：其中  $\delta$  是簇内成员个数的阀值， $\delta$  的作用是限制簇的大小，以防止簇内成员过多时簇头出现过载并影响底层 MAC(Medium Access Control)协议的效率<sup>[57]</sup>。 $N_m(i)$  表示节点  $i$  按照欧几里德距离第  $m$  近的邻居。 $Num(ch_i)$  代表以  $i$  为簇头的簇所包含的节点个数。 $\alpha$  和  $\beta$  是用于重新选举簇头的权重。

CGCA 假定传感器网络模型具有下列的属性：

- (1) 网络中的传感器节点都是静止的，这是一般传感器网络的典型配置，并且网络中的节点部署要比较密集。
- (2) 节点具有相同的初始电池能量和相同的感知、通信和计算能力，并且网络中的所有节点都具有同等的重要性。

#### 3.3.2 簇设计原则

CGCA 满足下面的簇设计原则，其证明见 3.3.3。

- 1) 分簇算法必须完全是分布式的，每个节点根据局部信息独立的做出自己

的决定。

- 2) 簇头在部署区域应该是均匀分布。
- 3) 在每次分簇算法结束时，每个节点或者是簇头或者是仅仅属于一个簇头的簇内成员。
- 4) 簇头形成的覆盖图(overlay)必须是连通的，这与簇头的簇间通信能量(inter-cluster communication power)有很大的关系，如果各个簇头使用最大通信能量也不能使覆盖图连通，那么需要选择网关(gateways)来连接簇头以满足连通性要求。
- 5) 分簇算法体现在处理和消息交换复杂度来说应该是高效的。

### 3.3.3 CGCA 算法描述

CGCA 主要分为三个部分：( i )形成簇(ii)连通簇头(iii)重新选举簇头。( ii )和(iii)将分别在 3.3.4 和 3.3.5 中具体阐述，系统启动时刻 CGCA 形成簇的具体过程如图 3.3 所示。

下面证明算法是正确的并且满足 3.3.2 所列举的簇设计原则。

引理 1：CGCA 产生的每个簇都是一个完全图

证明：上述算法执行一个循环，每一次执行步骤 1 中的 while 循环产生一个簇集合  $C$ ，因此只需要证明初始情况下以  $j$  为簇头(步骤 3)的簇是一个完全图。当  $j$  的邻居节点非空时(步骤 5)， $j$  对它的邻居节点按照它们与  $j$  的距离从小到大排序(步骤 6)，簇头优先选择距离其近的邻居加入到簇中。下面用数学归纳法对  $C$  按照节点的个数进行证明：当  $C$  包含一个节点即节点  $j$  本身时， $C$  显然是一个完全图。假设包含  $i$  个节点的  $C$  是一个完全图，即  $C$  包含  $j$  和它的  $1 \sim i-1$  个邻居。下面证明当加入  $j$  的第  $i$  个邻近邻居节点时， $C$  仍然构成一个完全图。当第  $i$  个节点成功加入簇集合  $C$  时，说明满足步骤 15 中的判别条件，即第  $i$  个节点同时位于  $j$  的第 1 至第  $i-1$  个节点的邻居列表中(步骤 9 中的  $k$  从 1 到  $i-1$ )。因此加入  $j$  的第  $i$  个邻近邻居节点仍然维持  $C$  的完全图属性。注意当  $C$  中节点个数大于  $\delta$  时算法中止(步骤 20)，这是为了防止簇内成员过多而导致干扰加重和影响底层 MAC(Medium Access Control)协议的效率。每当形成一个簇，如果还有其他的节点没有分簇(步骤 24 和步骤 1)，则继续同样的分簇过程。证毕。

```

1. while(  $V$  is not empty)
2. begin randomly select a node  $j$  in  $V$ ;
3.    $j \leftarrow clusterhead$  ;
4.   compute node  $j$ 's neighbor set  $N(j)$ 
5.   if(  $N(j)$  is not empty)
6.     Sort the neighbor list according to the distance to  $j$  in an increasing order;
7.     Initialize cluster set  $C$  to  $j$  and  $N_1(j)$ , that is  $C \leftarrow \{j, N_1(j)\}$ ,  $N_1(j)$  is  $j$ 's nearest neighbor;
8.     for  $i \leftarrow 2$  to  $|N(j)|$ 
9.       begin for  $k \leftarrow 1$  to  $i-1$ 
10.          begin if(  $N_i(j)$  is in the neighbor list of  $N_k(j)$ )
11.             $k \leftarrow k+1$ ;
12.          else
13.            break;
14.          end
15.          if(  $k = i$ )
16.             $C \leftarrow C \cup \{N_i(j)\}$ ;
17.          else
18.            break;
19.           $i \leftarrow i+1$ ;
20.          if(  $|C| > \delta$ )
21.            break;
22.          end
23.        set all the nodes in  $C$  except  $j$  to cluster members and set their cluster head to  $j$ ;
24.         $V \leftarrow V - C$ ;
25.      end

```

图 3.3 系统启动时刻的分簇过程

引理 2：CGCA 产生的簇头在部署区域是均匀分布

证明：CGCA 是通过减少簇头和簇内成员的距离之和来保证簇头的均匀分布，如果距离之和较小，可以认为簇头的分布是均匀的。CGCA 首先对簇头节点的邻居根据它们到簇头的距离按从小到大排序，然后在算法的每次迭代过程中，邻居节点中距离簇头近的节点优先加入到簇中。这不仅能够使得簇头在部署区域

均匀分布，而且由于簇头和其簇内成员距离较近，簇头和簇内成员之间相互通信时，只需要较小的簇内通信能量(intra-cluster communication power)，从而减少了能量消耗并延长了网络寿命。

引理 3：CGCA 在处理和消息交换复杂度方面是高效的

证明：CGCA 的处理复杂度主要体现在簇头节点上，因为由它们负责进行分簇，而簇内成员主要的动作就是按照簇头节点的要求查询某些节点是否在其邻居列表中。由于邻居列表的长度是常数，并且查询的次数最多是  $\delta-1$ (发生在簇的大小为  $\delta$  时)，因此簇内成员的处理复杂度为  $O(1)$ 。簇头节点的处理复杂度主要是对其邻居节点按照它们到它的距离进行排序，由于其邻居的个数为  $O(1)$ ，因此其处理复杂度也为  $O(1)$ 。下面分析形成一个簇所需要交换的消息个数，从 3.3.3 的算法描述可以看出，簇头和簇内成员产生的消息个数相等，因为簇头每产生一个查询邻居消息，其簇内成员就返回一个关于邻居查询结果的消息。因此只需要计算簇头产生的消息个数。簇头产生的最大消息个数发生在产生的簇大小为  $\delta$  时，此时产生的消息个数  $M$  为  $\sum_{i=2}^{\delta-1} (i-1) = \sum_{i=1}^{\delta-2} i = (\delta-2) \cdot (\delta-1) / 2$ ，由于  $\delta$  为常数，

因此  $M = O(1)$ ，即消息复杂度为常数。上式中的  $i$  从 2 开始计数代表簇头从它的第二个邻近邻居开始验证，一直到其第  $\delta-1$  个邻近邻居。由于验证第  $i$  个邻居是否位于完全图中，簇头需要查看第  $i$  个邻居是否位于其前面  $1 \sim i-1$  节点的邻居列表中，因此循环中的计数变量为  $i-1$ 。CGCA 在单个簇的处理复杂度和消息复杂度是高效的，均为  $O(1)$ 。从全网角度来说，由于簇头的个数从理论上说最小为  $\lceil N/\delta \rceil$ ，最大为  $N$ (发生在  $\delta=1$  时)。因此 CGCA 在整个网络的处理和消息交换复杂度均为  $O(N)$ 。相比 HEED<sup>[59]</sup>在整个网络的  $O(N)$  消息复杂度和  $O(N^2)$  处理复杂度，CGCA 是高效的。问题得证。

### 3.3.4 连通簇头

分簇算法的优点之一是，分簇之后路由是在簇头所形成的覆盖图上进行，而覆盖图相比原来由所有节点所形成的无向图来说具有较小的网络直径，因此整个网络所需要的路由维护信息就大大的减少<sup>[91]</sup>。但是路由的前提是覆盖图必须是连通的，因此必须保证簇头之间是互相连通的。本文提出了一个启发式算法，它建立在下面的引理之上。

引理 4：分簇之后，最邻近的簇头之间的跳数最多为 3

证明：用反证法证明，如图 3.4 所示，假设与簇头  $C_1$  最邻近的簇头是  $C_2$ ，它们之间的跳数为 4，中间经过节点 A, B, C。根据 3.3.2 节簇设计原则中的(3)，

B 节点受距离其一跳远的某个簇头节点控制，在图中以  $C_3$  来表示，但  $C_3$  和  $C_1$  之间的跳数为 3，与假设的  $C_2$  是  $C_1$  的最近簇头相矛盾。由于 CGCA 形成的簇是一个完全图，即产生的簇结构是一个一跳网络。因此当重新分簇之后，如果新产生的簇头是上次分簇的网关节点，此时它与邻近的簇头最大跳数为 2，否则继续选择以前的网关节点和相邻的簇头进行通信，因此不破坏最邻近簇头之间的最大跳数是 3 的性质。证毕。

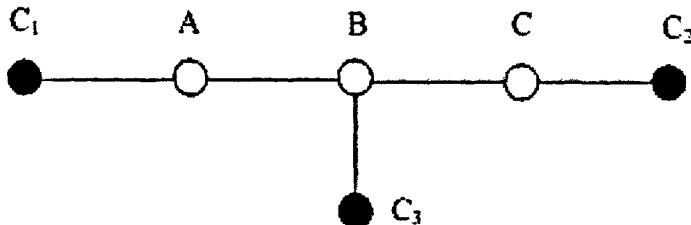


图 3.4 最邻近簇头之间的最大跳数

启发式方法通过连通邻近的簇头来保证所有的簇头之间是连通的。首先每个簇头节点广播一个消息，消息中包含自己的 ID 和一个初始化为 0 的跳数值。如果其他的簇头可以直接接收到此消息，这意味着它们彼此可以直达对方，此时在双方的路由表中加对方为直接到达。否则，其他的节点接收到簇头的消息后，存储此消息并将跳数值加 1 然后转发出去，从表面上看这类似于一个洪泛过程，但是由引理 4 可以得知，最邻近簇头之间的最大跳数为 3。因此当一个普通节点接收到一个跳数为 3 的消息时，就舍弃此消息不再对消息进行转发。当一个簇头接收到一个包含跳数值为 2 的消息时，代表它和发送此消息的簇头之间的跳数为 3，它们之间是邻近的。此时转发路径上的普通节点用来担当网关(gateways)，它们用来连通不直接相邻但是邻近的簇头。

### 3.3.5 CGCA 示例

利用 CGCA 对 10 个节点的集合进行分簇，其中  $\delta=4$ ，数字代表节点的标号，如图 3.5 所示。

当开始分簇阶段随机选取的节点为 1( $j=1$ )，即 1 为簇头时的簇形成过程如表 3.2 所示，表 3.2 显示了节点 1 是怎样把其邻居节点加入到簇集合  $C$  中的过程。另外两个以 5 和 8 为簇头的簇形成过程与此类似，不再详述。

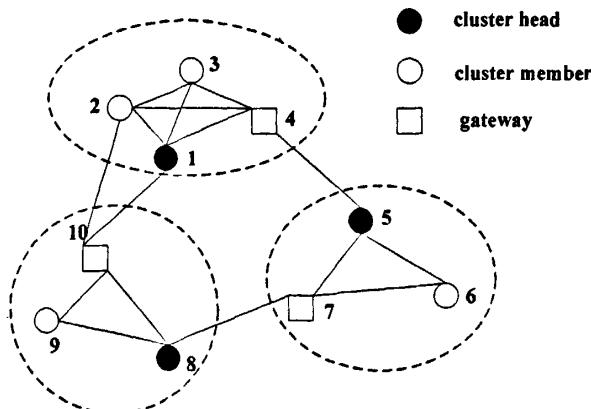


图 3.5 CGCA 分簇示例

表 3.2 CGCA 簇形成过程示例

| Step | Action  | Cluster set C |
|------|---|---------------|
| 1    | compute $N(1)$ and sort, $N(1)=\{2,3,4,10\}$                  | {1}           |
| 2    | $N(1) \neq \emptyset$ , add 2 to C                            | {1,2}         |
| 3    | if 3 is a neighbor of 2? Yes, then add 3 to C                 | {1,2,3}       |
| 4    | if 4 is a neighbor of 2? Yes                                  | {1,2,3}       |
| 5    | if 4 is a neighbor of 3? Yes, then add 4 to C                 | {1,2,3,4}     |
| 6    | if 10 is a neighbor of 2? Yes                                 | {1,2,3,4}     |
| 7    | if 10 is a neighbor of 3? No, terminate and one cluster forms | {1,2,3,4}     |

### 3.3.6 重新选举簇头

相比普通节点，簇头要承担更多的任务，例如簇内协作，簇内资源分配，数据聚合等，这导致它消耗更多的能量。因此为了负载平衡和延长传感器网络寿命，经过一阶段时间必须就要重新选举簇头以把担当簇头的机会分配给所有的节点。另外簇头节点的失效或者能量耗尽也是重新选举簇头的原因。

与以往的重新选举簇头意味着重新执行分簇算法不同, CGCA 的分簇协议只在系统启动阶段运行一次。这是因为 CGCA 形成的簇结构是一个完全图, 每个节点和簇中的其他节点都是等价的。因此重新选举簇头只在簇内进行, 这大大减少了重新选举簇头的复杂性和计算负载。由于系统启动阶段所有的节点都具有相同的初始能量和同等的重要性, 因此 CGCA 初始的簇头选择是随机的并通过优先选择邻近的邻居加入簇中以减少通讯能量耗费。这里的关键在于随后的阶段何时重新选择簇头以及如何选择新的簇头。

由于产生的簇的大小并不相同, 因此不同的簇头消耗能量的速度也并不相同, 一般情况下, 拥有簇内成员较多的簇头其消耗能量的速度也较快。因此为了达到较好的负载平衡, 这里采用一个简单的策略, 当簇头消耗的能量占其总能量的  $\frac{1}{Num(ch_i)}$  时重新选举新的簇头, 这里  $Num(ch_i)$  代表簇头  $i$  的簇内成员个数。

选举新的簇头按照下面的加权公式选择权重最小的。

$$W_i = \alpha \sum_{j \in N(i)} d(i, j) + \beta \frac{1}{E_i} \quad (3.2)$$

其中  $W_i$  代表节点  $i$  的复合权重,  $d(i, j)$  是节点  $i, j$  之间的距离,  $E_i$  是节点  $i$  的剩余能量,  $\alpha, \beta$  是大于 0 的权重因子。 $W_i$  综合两点来考虑: 一是节点和其邻居的距离之和; 二是节点的剩余能量。通常选择剩余能量大并且距离之和小的节点担当簇头, 即在一个簇内部选择有最小  $W_i$  的节点  $i$  担当新簇头。

如上所述, CGCA 的簇头轮换是异步进行的, 所带来优点一是克服了很多分簇算法需要非常严格的时间同步的问题; 二是重新选举簇头只需要在局部的单个簇内触发而不需要所有节点参与, 因此大大的减少了计算和通信负载。但是带来的问题是新的簇头产生之后, 应该如何通知邻近簇头, 否则将无法实现路由。由于 CGCA 产生的簇结构是一个一跳网络, 这一步比较简单。具体的过程是: 旧的簇头给新选举出的簇头发送其邻近簇头以及对应网关的 ID, 如果新的簇头检查自己的 ID 不同于任何网关的 ID, 则新的簇头只需要把接收到的信息添加到其路由表中(由于是完全图结构, 因此以前的网关现在仍然可以继续担当网关)。如果新簇头的 ID 等于某个网关的 ID, 则新的簇头需要首先将接收到的消息删除自身的 ID, 然后将此消息添加到路由表中即可。因此簇头轮换选举出新的簇头时, 为了和相邻的簇头通信, 只需要额外的一个消息和极少的信息处理, 因此消耗的能量可以忽略。因此这种异步簇头轮换带来的另一好处是产生的簇具有很好的容错性(fault tolerance)。即当某个簇头出现故障时, 新的簇头可以很快的选出并和邻近的簇头重新建立路由, 而不像大多已有的算法所有的节点重新执行一次分簇

算法以产生新的簇头集合。

### 3.4 仿真验证

利用 OMNET++<sup>[87]</sup>仿真平台对 CGCA 进行仿真模拟，实验模拟了  $N$  个节点部署在  $100m \times 100m$  下的算法性能，并与其他分簇算法进行了性能比较。本文确定了四个度量来衡量算法的性能：(i) 簇头个数 (ii) 单节点簇的比例 (iii) 每次分簇交换的消息个数 (iv) 簇头的均匀分布程度。研究这些度量在节点个数( $N$ )和节点发射半径 ( $tx_{range}$ ) 下的变化趋势。为了实验结果的准确性，实验结果取 10 次实验的平均值，每次实验都是把  $N$  个节点随机的散播在  $100m \times 100m$  的区域。相比于节点与其邻居的邻近程度，节点的剩余能量在决定节点是否有资格担当簇头中起着更为重要的作用，因此在节点的权重公式中，取  $\alpha=0.7$ ,  $\beta=0.3$ 。

CGCA 在节点传输半径  $tx_{range}$  下的簇头个数变化趋势，如图 3.6 所示，其中  $\delta=8$ ，节点的个数分别设置为 50, 100, 200。从图 3.6 可以看出两点规律：第一，簇头个数随着节点传输半径  $tx_{range}$  的增加而减少，这是显而易见的。因为  $tx_{range}$  越大代表着簇头可以覆盖更多的簇内成员，在同样的节点个数的情况下簇内成员个数越多意味着簇头的个数越少。第二，在  $tx_{range}$  较小时 ( $tx_{range} \leq 20$ ) 时，三条曲线下降明显，而当  $tx_{range}$  较大 ( $tx_{range} \geq 25$ ) 时，曲线趋向水平变化不明显。这是因为当  $tx_{range}$  较小时，CGCA 产生的簇(完全图)大小远小于  $\delta$ ，因此随着  $tx_{range}$  的增加，簇的大小也明显随着增加导致簇头个数明显减少。而随着  $tx_{range}$  继续增加到一个较大值时，产生的簇的大小为接近  $\delta$  的一个值，由于簇的大小增加缓慢，因此簇头的个数也变化不明显。因此 CGCA 所产生的簇头个数主要受节点传输半径  $tx_{range}$  的影响，而  $\delta$  的作用是限制簇的大小，防止簇头出现过载。注意当  $tx_{range}$  较小时，CGCA 产生较多的簇头。这一方面减少了分簇算法所带来的数据聚合等方面将的优势，另一方面消息传递需要经过更多的跳数，因此带来较大的延迟和能量消耗。注意在其他分簇算法中，即使节点的传输半径很大，也不意味着簇头可以拥有任意多的簇内成员。这是因为有的技术限制簇的大小，比如蓝牙(Bluetooth)规定一个簇头最多有七个簇内成员。另外，簇内成员太多将会导致簇头过载、干扰加重以及影响底层 MAC 协议的效率。CGCA 中的参数  $\delta$  可以在簇大小方面取得一个比较好的折衷。现在普通 Micaz 节点的传输半径通常为  $30m \sim 50m$ ，因此 CGCA 算法的性能不会受到较大的影响。

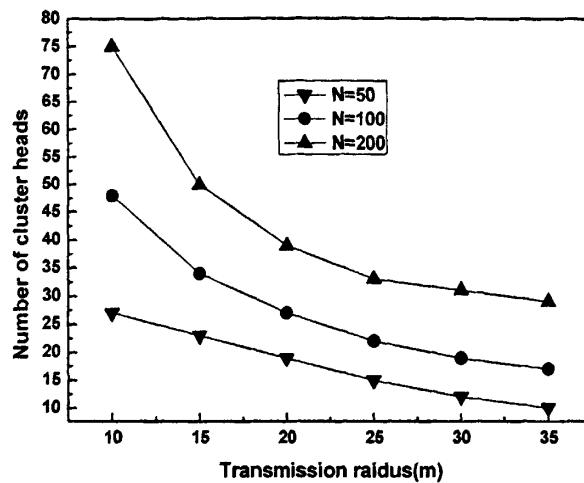


图 3.6 簇头个数随传输半径的变化趋势

在分簇算法中，可能产生具有单个节点的簇，即只包含簇头的簇。单节点簇的产生通常是由于某个簇头的邻居节点由于某种原因(例如邻近程度)加入到其他的簇中，此时此簇头要独立成簇。单节点的簇是分簇算法应该尽力避免的，这是因为单簇头削弱了分簇算法在数据聚合方面的优势，从某种程度上失去了分簇的意义。CGCA 在单节点簇方面的表现如图 3.7 所示，图 3.7 显示了单节点簇的比例随着节点个数变化的趋势。其中  $\delta=8$ ，节点的发射半径分别设置为 25m, 35m。

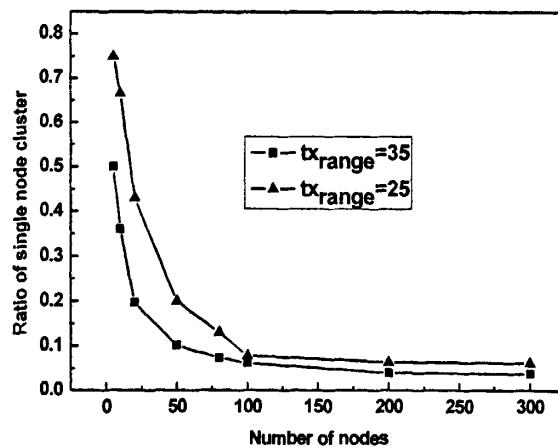


图 3.7 单节点簇的比例随着节点个数的变化趋势

图3.7可以看出，单节点簇比例随着节点个数的增加而减少。这是由于当节点个数较少时，节点的分布比较稀疏，节点之间的距离也比较远，节点的邻居个数就会较少，这样形成单个簇头的几率就会较大。而随着节点个数的增加，同样的发射半径能够使得簇头覆盖更多的簇内邻居成员，因此单节点簇头的比例就会大大减少。CGCA在 $N=5$ ,  $tx_{range}=25$ 时，单节点簇的比例为75%，而在 $N=200$ ,  $tx_{range}=25$ 时，比例仅为0.06%，这也从另一方面证明了CGCA不适合稀疏部署的网络，因为在稀疏部署的网络，节点之间很难形成完全图。另外，从图3.7中可以看出，在同样的情况下，节点的发射半径越大，单簇头的比例就越低。这是因为发射半径越大，节点的邻居个数就会越多，这减少了使节点成为单簇头的可能性。

图3.8显示了在执行分簇算法过程中消息交换个数方面，CGCA与HEED分簇算法的比较(HEED采用AMRP作为耗费定义)。其中节点的发射半径 $tx_{range}(i)=35$ ，簇大小阀值 $\delta=8$ 。

图3.8可以看出，在节点个数 $N$ 较小( $N \leq 50$ )时，CGCA在消息交换个数方面优势并不明显，当 $N$ 较大时( $N \geq 100$ )，CGCA要明显的优于HEED分簇算法。而且随着 $N$ 的继续增大，这种优势就越来越明显。例如当 $N=300$ 时，CGCA产生的消息个数只是HEED所产生消息个数的17.7%。这是因为如果HEED算法的耗费定义采用AMRP(Average Minimum Reachability Power)，每个节点都需要向其邻居节点发送一个消息指明到此邻居所需的信号功率。假设网络拓扑为无向图 $G=(V, E)$ ，节点个数即 $|V|=N$ ，仅在全网内广播AMRP所需要的消息个数就为 $2|E|$ 。当网络稀疏时， $|E| \sim \Theta(N)$ ，而当节点密集部署时， $|E| \sim \Theta(N^2)$ 。另外HEED中临时簇头(tentative cluster heads)集合也产生大量的冗余信息。CGCA分簇产生较少消息个数的根本原因是：HEED以及其他加权分簇算法例如WCA<sup>[57]</sup>，在分簇过程中，所有节点独立同步的发送消息来决定自己担当簇头还是簇内成员，因此产生大量的消息负载；而CGCA利用完全图的等价性，首先选举出簇头，然后由簇头异步的决定其各自的簇内成员，产生的消息负载主要集中在占节点比例少量的簇头节点，这导致CGCA在分簇过程中产生少量的消息负载。除此之外，如3.3.5所述，在重新分簇阶段，只需要很少的消息量就可以选举出新的簇头并且建立新的路由。而目前大多分簇算法，重新分簇需要相关节点重新执行一次分簇算法以选举出新的簇头集合，这将会带来较大的消息负载。在传感器节点的能量消耗中，发送和接收消息的能耗最大，因此较少的消息量通常意味着传感器网络有较长的寿命。由于缺乏一个统一的能耗模型以及网络寿命的定义，因此这里采用分簇过程所产生的消息交换个数来近似的表示分簇算法的

能量有效性。

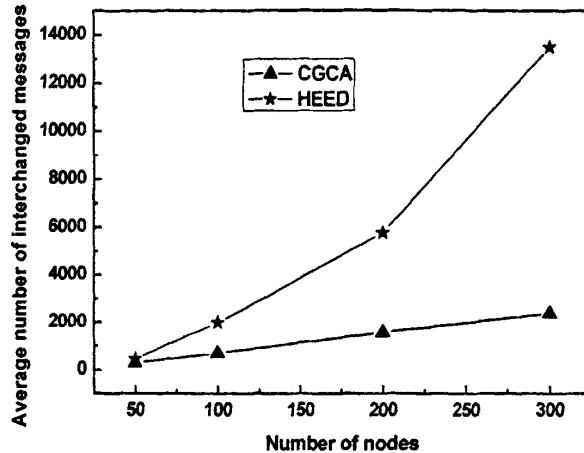


图 3.8 CGCA 与 HEED 的消息交换个数比较

CGCA 与 LEACH<sup>[55]</sup>协议在簇头均匀分布方面的比较, 如图 3.9 所示。其中节点的个数  $N=200$ , 簇大小阀值  $\delta=8$ 。簇头的均匀分布情况用所有簇头和簇内成员距离的平均值来衡量。从图 3.9 可以看出, 在簇头均匀分布方面, CGCA 明显优于 LEACH 协议。这是由于 LEACH 协议在选择簇头时只是根据节点随机产生的一个阀值来判断, 而并没有考虑它和簇内成员的邻近程度。而 CGCA 在计算初始簇头和后续簇头的过程中, 把簇头和簇内成员的邻近程度考虑在内, 优先选择距离较近的簇内成员。另外从图 3.9 可以观察得到, CGCA 中簇头和簇内成员的平均距离随着传输半径的增大而增大。这是因为当传输半径较小时, 簇的大小远小于  $\delta$ , 按照算法规定, 簇头优先选择距离最近的邻居加入到簇中, 因此簇头与邻居之间的平均距离较小。随着传输半径的增大, 簇的大小接近于  $\delta$ , 此时簇头就会选择一些距离其较远的邻居加入簇中, 从而导致平均距离较大。虽然传输半径较小时, 簇头分布更均匀, 但是会导致产生很多小的簇, 使分簇算法在数据聚合方面的优势得不到充分体现。因此为了系统的整体效率, 通常折衷选择适当大小的传输半径。

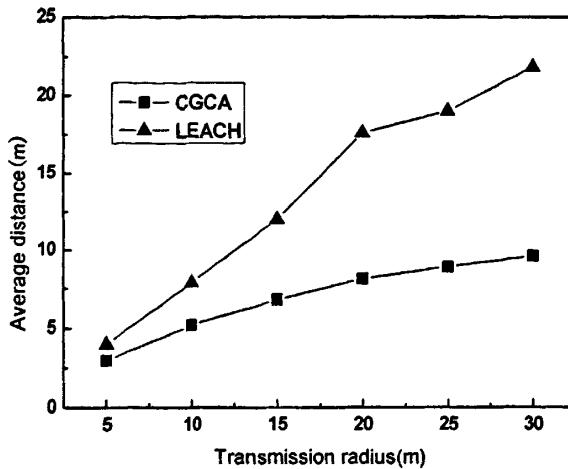


图 3.9 CGCA 与 LEACH 协议的簇头分布比较

### 3.5 本章小结

在这一章中，我们主要研究了怎样减少重新分簇所带来的高负载以及维护一个稳定簇结构的问题，并提出了一种基于完全图的分簇算法。最后通过仿真实验，对 CGCA 和目前性能较好的协议进行了分析比较。

由于 CGCA 把重新分簇这一全局性、周期性的行为转化为局部性和由事件触发，这使得 CGCA 的通信和计算负载可以大量减少，它在单个节点的处理复杂度和消息复杂度均为  $O(1)$ 。另外，通过优先选择距离簇头近的节点加入簇内，CGCA 不仅减少了簇头和簇内成员的簇内通信能量，而且使得簇头比较均匀的分布在部署区域。仿真实验表明：在节点密集部署的情况下，CGCA 产生的消息交换个数远小于 HEED 分簇算法。最后簇头均匀分布方面。CGCA 也明显优于 LEACH 分簇算法。

由于在节点个数较少时，节点分布稀疏使得节点之间很难构成完全图，因此 CGCA 不适合稀疏部署的传感器网络。另外 CGCA 没有考虑节点的移动性，在将来的研究中考虑调整 CGCA 使其适合于有少量节点移动的传感网络。

## 第4章 加入移动网关的分簇算法

在一个分簇算法中，一个网关节点（gateway）除了要完成基本的感知数据之外，还要在相邻的簇之间承担起数据转发的任务。而由于簇之间的数据流量比较大，因此相对于普通节点来说，网关节点更容易耗尽自己的能量而成为瓶颈节点，从而导致传感网络系统的寿命减少并进一步的削弱系统的吞吐量。

本章首先介绍了在无线传感网络中加入移动节点的背景和相关工作，其次在第三章所介绍的 CGCA 分簇算法基础上通过加入资源丰富、硬件配置高的移动节点来担当簇与簇之间的网关以延长传感网络寿命，最后通过仿真模拟来验证所提出算法的性能。

### 4.1 背景介绍

在目前有关无线传感网络的大部分文献中，提出了大量的有关拓扑控制、MAC 和路由等方面的协议和算法旨在减少节点能耗和延长传感网络寿命。但这些算法通常都有一个基本的假设，那就是传感网络中所有的节点都是静态部署的。而目前一个研究的趋势和热点是通过加入某些移动节点来担当中继或者汇聚节点，以更有利于感知数据顺利到达汇聚节点并同时提高网络寿命。

最早通过加入移动节点来提高网络性能出现在 Ad-hoc 网络中，例如<sup>[92]</sup>提出了一种算法，该算法借助移动节点的移动性以提高一个不连通的 Ad-hoc 网络中的网络连通性并转发消息。文献<sup>[93]</sup>论证了通过使得某些节点具有移动性，可以较大的提高 Ad-hoc 网络的吞吐量。而文献<sup>[94]</sup>提出了一种算法，使得一部分节点担当移动代理（mobile agents）并相互协作，在其他两个移动节点之间取得了能量有效的数据通信。

在无线传感网络中加入移动节点的背后机制有两点：一是在某些传感网络的某些应用中，例如在战场跟踪敌方部队，此时节点的部署环境恶劣并且系统要求网络要达到一定的性能保证，例如要达到指定的连通性和覆盖性。此时由于传感器节点的易错性，通常需要随机布置大量的冗余静态节点来满足指定的要求，系统的总耗费比较大。这种情况下，可以加入一些资源丰富和硬件配置能力高的移动节点和相对少量的静态节点也可以满足同样的系统要求，但此时系统总的耗费反而更低<sup>[95]</sup>。

二是无线传感网络是一个多跳自组织结构，通常节点采集到数据之后需要经过多个节点的中继才能到达汇聚节点（sink node）。因此，普通节点除了要完成基本的感知数据之外，还要承担起转发其他节点数据的任务，造成了较大的能量

浪费。尤其是靠近汇聚节点的那些节点，由于所有其他节点的数据都要经过它们的转发才能到达汇聚节点，因此这些节点的能量消耗非常快，导致它们成为系统的瓶颈节点。这种情况下在传感网络中加入移动节点就会出现两种可能性：一是移动节点担任汇聚节点；二是移动节点担任中继节点。

移动节点担当汇聚节点的本质就是把上述瓶颈节点平均分布到网络中的每个节点而不是仅仅局限在离其较近的那些节点，因此能够显著的提高网络寿命。根据移动汇聚节点的运动轨迹的不同可进一步的划分为两种情况：一是移动汇聚节点只能在事先确定好的某些“离散”点进行“跳跃”移动；二是移动汇聚节点可以沿着某条轨迹“连续”的移动<sup>[95]</sup>。前一种情况下，需要在满足节点初始能量限制的前提下，解决移动节点在那些点逗留并且要确定逗留的时间以使网络的寿命达到最大化，这可以最终转化为线性规划问题（Linear Programming）<sup>[96, 97]</sup>，后一种情况可以通过建立一个地理的流量模型来求得最优移动路径<sup>[98]</sup>。但是移动节点担当汇聚节点的缺点有以下几个方面：一是汇聚节点是传感网络与主干网的网关，当汇聚节点移动时，很难保证传感网络和主干网的持续连通性。二是所有节点都需要获知移动汇聚节点的当前位置以便把感知信息路由至它。

移动中继节点的背后机理是通过资源丰富的移动节点来承担在普通节点和汇聚节点之间转发数据的任务，从而让普通节点只需要专注于进行数据收集。此时节点不再需要过多跳将数据传输至汇聚节点，而是直接就将数据传送给距离其一跳远的移动中继节点，由移动中继再将接收到的数据进行缓存并传输至汇聚节点。综上所述，移动中继节点能够将原来的长距离的多跳通信变为现在的短距离一跳通信，因此能够较大的节省节点能量和延长传感网络寿命。例如文献<sup>[95]</sup>通过加入一个移动中继节点，并让此节点在汇聚节点的两跳范围之内担任数据转发任务，就可以将系统的寿命达到原来系统（指的是没有加入移动中继节点）的四倍左右。

相比移动汇聚节点，移动中继节点会带来较大的延迟。这是因为当一个节点有数据要发送时，只能等到移动中继节点靠近它时才能发送数据至移动中继节点，中继节点对数据进行缓存，等中继节点移动到汇聚节点的附近范围内才能最终将数据传送至汇聚节点，这将会带来较大的延迟。因此移动中继节点适合于对延迟性要求不是很高的应用，例如延迟容忍网络 DTN（Delay Tolerant Network）。除此之外，在延长传感网络寿命方面，基于移动汇聚节点的算法通常优于基于移动中继节点的算法。

除了移动节点担任汇聚节点和中继节点之外，在传感网络中加入移动性还有如下潜在的好处：

- 1) 网络修复：无线传感网络正常工作的基本前提就是网络必须是连通的，

当网络出现不连通时，可以通过加入一个或者几个移动节点移动到指定的位置来修复网络以维持网络的连通性。

- 2) 能量获取：文献<sup>[99]</sup>提出了一种方法，通过移动节点把能量从能量充裕的地方“搬运”到能量稀缺的地方，以建立一个自维持网络。
- 3) 事件监测：事件监测是无线传感网络最根本的功能之一。初始的网络部署可能并不是进行事件监测的最优部署，此时可以通过加入移动节点并使其移动到高概率的事件发生地对此问题进行修正。并且当被监测对象在动态变化时，可以自适应的调整移动节点的路线以随时对事件进行监测<sup>[100]</sup>。

## 4.2 相关工作

本节就在无线传感网络中加入移动中继和移动汇聚节点分别给出了一个典型的算法实例。

### 4.2.1 移动中继节点算法

最早提出移动中继节点思想的是 Intel 研究院的 Shah 等人<sup>[101]</sup>提出的 称为 Data Mule 的数据转发单元，Data Mule 相对于普通的节点来说，具有可更换电池的能力并且具有更强的信息处理能力和更大的内存供应。Shah 提出了一个三层的网络架构来提高转发数据成功率。位于中间层的 Data Mule 承担在位于第一层的普通静态节点集合和位于第三层的汇聚节点集合之间的数据转发任务。如图 4.1 所示，(a)表示了普通的节点 A 担任普通节点集合和汇聚节点集合之间的数据转发任务，可以看出节点 A 是系统中的瓶颈节点。一旦节点 A 的能量耗尽，传感网络的寿命也会随之终结。也就是说传感网络的寿命主要有节点 A 来决定。而在(b)中通过加入一个硬件水平配置更高 Data Mule 来担当中继节点以替换节点 A，可以很明显的提高网络的系统工作时间。

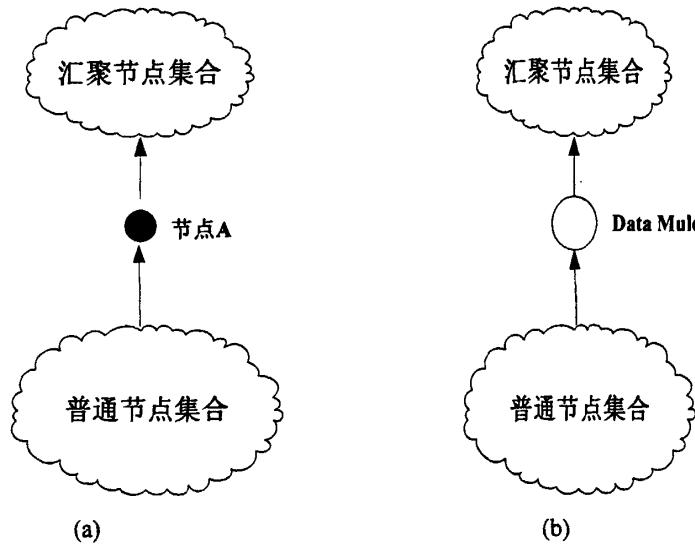


图 4.1 传感网络中加入移动中继节点

#### 4.2.2 移动汇聚节点算法

为了最大化网络寿命，Wang 等人<sup>[97]</sup>提出了利用线性规划来解决汇聚节点的移动轨迹并且计算出移动汇聚节点在指定位置的逗留时间。该算法巧妙的把最大化系统时间转化为最大化移动汇聚节点的在指定位置的逗留时间之和来建立线性规划模型，为了能够更好的理解该模型，定义如下参数和变量：

参数：

- $e_0$ : 节点的初始能量减去节点能够正常工作的能量阈值，单位是 J。
- $e$ : 节点发送或者接收一个比特的能耗系数，单位是 J/bit。
- $N$ : 网络的节点集合。
- $n$ : 网络中的节点个数。
- $L$ : 网络部署的正方形网格区域的边长。
- $r$ : 节点的数据包产生率，假设  $r$  对每个节点均相同，单位是 bits/sec。
- $f_{ij}^k$ : 当移动汇聚节点停留在节点  $k$  的位置时，从节点  $i$  到节点  $j$  的数据发送率，单位是 bits/sec。
- $c_i^k$ : 当移动汇聚节点停留在节点  $k$  的位置时，节点  $i$  用于发送和接收数据包的总能耗，单位是 J/sec。
- $S_i$ : 节点  $i$  的邻居集合。

变量：

- $t_k$ : 移动汇聚节点在节点  $k$  的逗留时间，单位是 s。

$z$ :网络寿命, 定义为从开始到第一个节点能量耗尽的时间, 单位是 s。

由此, 可以建立如下的线性规划模型:

$$\text{Max } z = \sum_{k \in N} t_k \quad (4.1)$$

$$\text{such that } \sum c_i^k t_k \leq e_0, i \in N \quad (4.2)$$

$$\text{and } t_k \geq 0, k \in N \quad (4.3)$$

公式(4.1)描述了目标函数用以最大化网络寿命, 并将其转化为最大化移动汇聚节点在所有可能位置的逗留时间之和。公式(4.2)和(4.3)描述了约束条件, 其中(4.2)刻画了任意一个节点  $i$  在移动汇聚节点的逗留时间之内的所有能量之和应该小于其初始能量限制  $e_0$ 。而(4.3)仅仅约束了逗留时间的非负性。上面公式的难点在于  $c_i^k$  的求解, 如下面公式所示:

$$c_i^k = e \sum_{j \in S_i} f_{ij}^k + e \sum_{j \in S_j} f_{ji}^k \quad i, k \in N \text{ and } i \neq k \quad (4.4)$$

$$c_i^k = er, \quad i, k \in N \text{ and } i = k \quad (4.5)$$

公式(4.4)描述了移动汇聚节点  $k$  和节点  $i$  在不重叠情况 (即位置不相同) 下的  $c_i^k$  的值。此时, 节点  $i$  消耗的能量分为两部分, 其中公式右侧第一部分表示节点  $i$  发送数据的能耗, 而第二部分表示接收数据的能耗。而公式(4.5)描述了汇聚节点  $k$  和节点  $i$  在重叠情况 (即位置相同) 下的  $c_i^k$  的值。在这种情况下, 节点  $i$  消耗的能量仅仅是其发送数据的能耗即  $er$ 。

### 4.3 本文工作

在本文中加入移动节点既不是担任移动汇聚节点也不是移动中继节点, 而是担当分簇算法中簇与簇之间的移动网关。本文在第三章所提的基于完全图的分簇算法 CGCA 的基础上, 加入了某些移动节点来担当网关以转发簇与簇之间的数据流, 并且能够保持簇头与簇头之间的持久连通性<sup>[102]</sup>。

在我们的算法中采用移动节点代替普通节点来担当网关有以下优点: 一是普通的网关节点除了要完成数据采集任务之外, 而且还要转发簇与簇之间繁重的数据流, 导致消耗大量的能量。因此加入资源丰富, 硬件配置高的移动节点担当网关就可以将普通节点从沉重的数据转发的任务中解放出来, 让这些节点只是专注于数据收集或者其他有用的业务, 这样就可以节省了节点能量以延长传感网络的寿命。二是由移动节点来担当网关可以很明显的提高系统的稳定性和容错性, 这

是因为普通的网关节点由于能量受限并且容易出现故障，一旦网关出现故障就会使网络出现不连通的情况，从而严重削弱系统的整体性能。三是加入移动节点担当网关之后，在重新分簇阶段，只需要在每个簇内选出新的簇头，而不用考虑重新选举网关的问题。加入移动节点担当网关的缺点就是引入新的移动节点增加了硬件耗费，但是我们认为在取得系统寿命明显增长的同时，只需要承担相对较低的硬件耗费是值得的。

### 4.3.1 移动网关算法流程

在 CGCA 分簇算法中，加入移动网关来连通簇的算法大体流程是：在执行 CGCA 算法之后，首先一个移动节点沿着事先制定好的路线移动，在移动的过程中此节点需要通过某种方法计算出在部署区域中的那些位置簇头之间是不连通的，也就是说在那些位置需要加入移动网关，此移动节点对这些位置进行存储。待移动节点回到出发点之后，就派出相应个数的移动节点到指定的位置担任移动网关。具体算法分为两个步骤：

**步骤 1：**每个簇头广播一个包含其自身 ID 的“Hello”消息，当簇头  $A$  收到从簇头  $B$  发送的此消息时，簇头  $A$  回复相应一个消息给  $B$ ，其中消息中包含  $A$  的 ID 和其簇内所有成员的地理位置信息。根据此消息，簇头  $B$  能够很容易的计算位于簇头  $A$  和  $B$  所在簇内任意两个节点的最大距离，以  $Max_D$  来表示此最大距离。然后  $B$  检查是否  $Max_D$  比节点的最大发射半径  $TX_{range}$  小，如果“是”，则表示簇头  $A$  和  $B$  所在簇内任意两个节点可以直接通信。这就意味着，即使在重新分簇阶段，新选举的簇头在不需要任何移动网关的前提下就可以相互直接通信。此时，簇头  $A$  和  $B$  在其各自路由表中加对方为直接到达。然后，每个簇头节点把自己的路由表发送至它的直接邻居，并从其他的直接邻居节点接收路由表同时把重复项删除。经过一段时间，这个过程的本质就相当于在网络中产生几个连通分量。在每个连通分量内，不同的簇头和簇头之间要么是直接相邻，要么可以通过别的簇头节点彼此到达对方。

**步骤 2：**假设每个移动节点都配有 GPS 定位设备。这一步骤我们借助于移动节点来担当网关，把产生的几个连通分量合并成一个连通分量，以给网络中的任意两个簇头节点提供连通性并且在簇与簇之间承担数据转发任务。移动节点采用的是 Robomote 平台<sup>[103]</sup>，首先由一个移动节点沿着部署区域“漫游”来发现区域内存在哪些位置需要对不同的连通分量进行连通并对这些位置进行记忆保存。因此一个问题自然就会产生，那就是此移动节点需要沿着一条什么样的移动轨迹进行移动？假设部署区域是一个矩形区域，我们把此区域划分成小的边长为  $TX_{range}$

$\sqrt{2}/2 \times \sqrt{2}/2$  的正方形网格，这就保证了在同一个网格内的簇头节点的连通性。因此此节点只需要沿着这些网格的边界进行移动而不需要遍历部署区域的每个点。需要注意的是，在移动节点进行移动的时间内，所有的簇头节点都要定期广播“Hello”消息。当此移动节点接收到从两个不同的簇头节点发送过来的两个消息之后，此移动节点需要停下来一段时间并发送一个询问消息至这两个簇头节点中的任意一个来确认是否这两个簇头节点属于同一个连通分量。如果收到的回复是“是”，那么代表这两个簇头节点在同一个连通分量内，因此不需要移动节点在它们之间担任网关。如果回复是“否”，那么此移动节点记住此位置以表示在这里需要派出一个移动节点来连通两个不相同的连通分量。除此之外，相关的两个簇头需要彼此交换路由表并对其进行合并，这样做的目的是把两个连通分量合并成一个并防止在将来部署多余的移动网关节点。这样当此移动节点沿着边界移动一周，就知道了在那些位置需要部署移动节点，并且派出相应数目的移动节点去那些相应的位置担当网关以转发数据，算法完毕。如图 4.2 所示，初始的移动节点只需要沿着红色标注的线进行移动，并且最终派出两个移动节点来连接三个不同的连通分量。

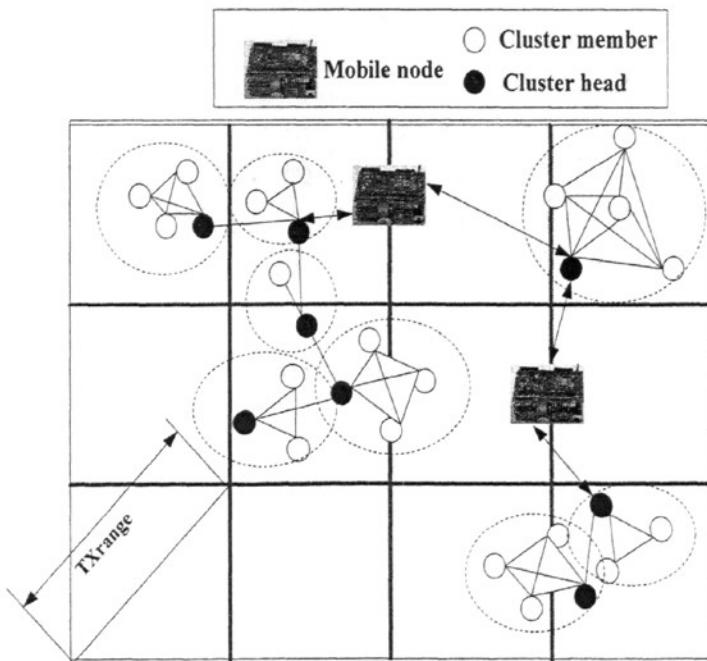


图 4.2 加入移动网关的 CGCA 算法

### 4.3.2 能耗模型

由于实验中需要测定传感网络的寿命，因此需要首先定义节点的能耗模型。公式(4.6)表示了节点发送 1bit 至距离  $d$  的能耗：

$$E_{tx} = \alpha_1 + \alpha_2 d^n \quad (4.6)$$

其中  $n$  为能量衰减因子并取值为 2。  $\alpha_1$  和  $\alpha_2$  取值为常数，并且在本文中取值如下：  $\alpha_1 = 50nJ/bit$ ，  $\alpha_2 = 100pJ/bit/m^2$ 。从其他节点接收一个 bit 的能量消耗如下：

$$E_{rx} = \varepsilon \quad (4.7)$$

在本文中  $\varepsilon$  的取值为  $50nJ/bit$ 。

在一个分簇算法中，一个簇内成员只需要将其产生的数据传送至它的簇头节点，然而一个簇头节点还要额外负责从其所有的簇内成员接收数据，并对数据进行聚合，然后把聚合后的数据传送至路由表的下一跳邻居。假设所有的节点都具有相同的初始能量和数据产生率，分别用  $e$  和  $r_g$  来表示。因此一个簇内成员的寿命可以用下面的公式来表示：

$$L_{cm} = \frac{e}{r_g(\alpha_1 + \alpha_2 d^n)} \quad (4.8)$$

其中  $d$  是到其相应簇头的距离，然而一个簇头的寿命表示如下：

$$L_{ch} = \frac{e}{((Num(ch_i)-1)r_g\varepsilon + (Num(ch_i)r_g + r_{ci})r_f \cdot (\alpha_1 + \alpha_2 d_i^n))} \quad (4.9)$$

其中  $r_{ci}$  是簇头  $i$  控制包的数据产生率， $r_f$  代表簇头  $i$  的数据融合率， $d_i$  表示簇头和其一跳邻居之间的距离。

注意公式(4.9)的分母的第一部分代表簇头从其簇内成员接收数据所消耗的能量，而第二部分表示把融合之后的数据发送至其邻居所消耗的能量。网络寿命定义为从系统启动到第一个节点耗尽能量所经历的时间。

### 4.3.3 仿真模拟

由于本文是对第三章所提出的 CGCA 分簇算法的进一步改进，因此本文所用到的参数和变量请参考第三章，这里不加赘述。除非另加说明，本文假设  $N$

个传感器节点随机均匀的散布在  $100m \times 100m$  的二维方形区域。为了衡量算法的性能，我们确定了三个度量标准：(I)产生簇的个数(II)分簇过程中产生的消息个数(III)传感网络的寿命(IV)移动网关节点个数。本文研究这些度量标准随着网络节点个数  $N$  和簇大小阀值  $\delta$  的变化趋势。对每次仿真场景，算法都被随机执行十次并对结果取平均值。仿真实验中用到的其他参数取值如下： $\alpha = 0.7$ ， $\beta = 0.3$ 。

图 4.3 显示了簇的个数在  $TX_{range}=25$  的前提下随着簇大小阀值  $\delta$  的变化趋势。从图中可以观察到两种现象：一方面随着  $\delta$  的增长，簇的个数在减少。这是由于大的  $\delta$  会使一个簇内拥有更多的节点个数，从而产生较少数目的簇。另一方面，当  $\delta$  取较小的值( $\delta \leq 6$ )时，三条曲线下降迅速。而随着  $\delta$  的进一步增大( $\delta > 8$ )时，三条曲线接近水平而变化不明显。这是由于  $\delta$  较小时，产生的簇（完全图）的大小接近于  $\delta$ ，因此随着  $\delta$  的增大，簇的个数明显的下降。而随着  $\delta$  的进一步增大，由于算法要满足完全图中节点相互临近的属性，此时簇的大小保持一个相对稳定的水平，因此簇的个数下降缓慢。

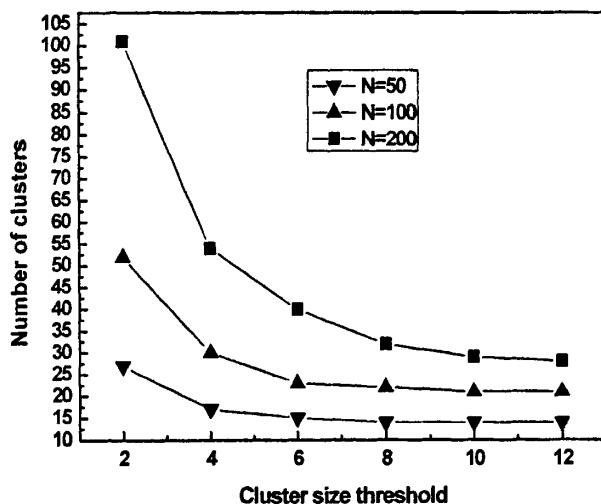


图 4.3 簇的数目随着簇大小阀值的变化

分簇过程中所产生的消息个数的多少是衡量分簇算法优劣的一个很重要的指标。产生的消息个数越少，表示分簇算法越高效，并且会导致网络总的能耗较小，从而间接的会延长传感网络寿命。图 4.4 研究了加入移动网关的 CGCA 算法和基于最小节点标号以及基于最大节点度的分簇算法在分簇过程中所产生的交换消息个数方面的性能比较，其中  $TX_{range}=30$ ,  $\delta=8$ 。当网络的节点个数较少( $N<50$ )

时,从图中可以看出,相比于另外两种启发式分簇算法,本文算法在交换消息个数方面的优势并不明显。但是随着  $N$  的进一步增大 ( $N > 100$ ), 本文算法所产生的消息个数明显的小于基于最小节点标号和最大节点度的分簇算法所产生的消息数。例如,在  $N=300$  的情况下,本文所产生的消息个数分别只是基于最小节点标号和最大节点度的分簇算法的 12% 和 13%。这里主要有两方面的原因:一是本文所产生的簇结构是一个完全图,因此簇头的选举只需要在系统启动时刻进行,而重新选举簇头也只是在每个局部的单个簇内部异步的触发选举,这将大量的减少所产生的消息负载。二是加入移动网关之后,使得重新选举簇头阶段网关不再需要重新选举,进一步的减少了消息负载。而基于最小节点标号和最大节点度的分簇算法的重新分簇过程是一个全局触发的类似于洪泛的过程来选举出新的簇头,因此会产生非常大的消息负载。图 4.4 还可以发现一个现象,那就是基于最大节点度的分簇算法所产生的消息个数略大于基于最小节点标号分簇算法所产生的。这是因为节点的标号是唯一的,而当在一个簇内有多个节点的最大度数相同的情况下,基于最大节点度的分簇算法需要额外的信息来解决此冲突以选举出唯一的簇头。

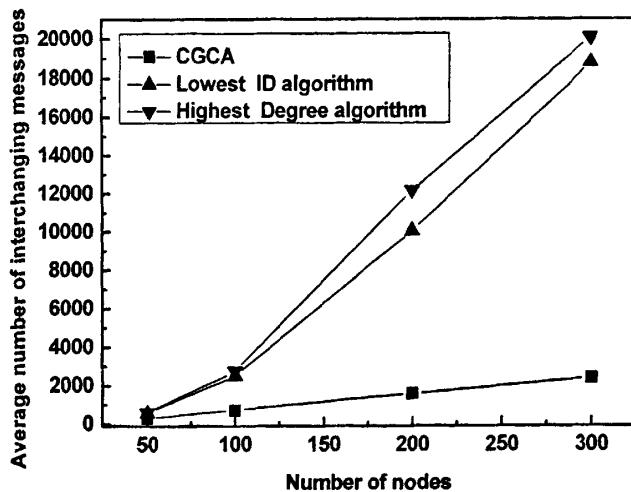


图 4.4 三种分簇算法所产生的消息个数

加入移动网关节点在延长系统寿命方面的作用,如图 4.5 所示。随着网络节点个数  $N$  的变化,本文算法和经典的 LEACH<sup>[55]</sup>在网络寿命方面进行了性能比较。在仿真环境中,  $TX_{range}$ ,  $\delta$ ,  $e$ ,  $r_g$ ,  $r_{ci}$ , and  $r_f$  分别取值为 30, 8, 2J, 100bit/s, 10bit/s 和 0.4。汇聚节点位于(50,50), 移动节点的速度设置为 10m/s, 并且假定

移动节点的能量不受任何限制。

从图 4.5 可以看出，在一个比较稀疏的网络 ( $N < 60$ ) 的情况下，LEACH 比本文算法能够取得更长的系统寿命。这是由于在节点个数较少的情况下，节点与节点之间较远的距离使得产生的簇的大小远小于簇大小阀值  $\delta$ ，因此本文的分簇算法在数据聚合方面的优势就得不到体现，以至于会产生较大的消息负载，从而减少了网络寿命。但是随着网络节点个数的增加，本文算法能够取得比 LEACH 算法更长的系统寿命，并且网络部署越密集，本文算法的性能提升就越明显。这里有两方面的原因：一是局部触发的重新分簇阶段大大的减少了消息负载，从而有利于延长网络寿命。二是本文加入的资源丰富的移动节点代替普通能量受限的节点来担当网关以承担繁重的簇与簇之间的数据流量，从而普通节点能够节省大量能耗，进一步的延长了传感网络寿命。图 4.5 从另一个侧面也验证了本文算法更适合于一个密集部署的传感网络。

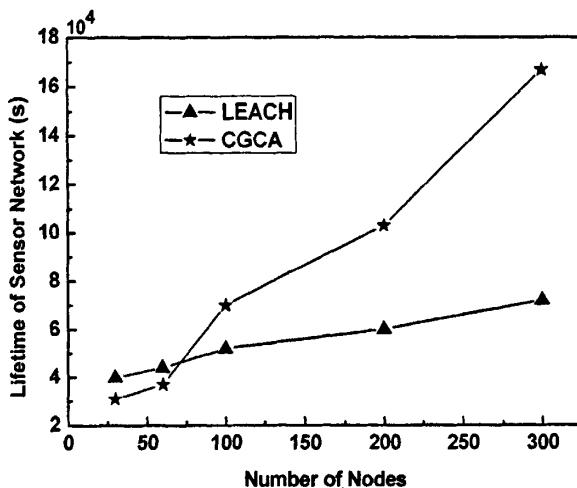


图 4.5 本文算法与 LEACH 的网络寿命比较

移动网关节点个数随着网络中传感器节点个数  $N$  的变化趋势，如图 4.6 所示。其中参数设置如下： $TX_{range}=30$ ,  $\delta=8$ 。从图 4.6 可以看出，移动网关节点个数的变化曲线随着传感器节点个数的增长呈先上升后下降的趋势。例如，当  $N=30$  时，移动网关节点个数为 8；当  $N=50$  时，移动网关节点个数变为 9；而当  $N=100$  时，移动节点个数却变为 5。造成这一现象的原因是：当  $N$  较小时，节点分布非常稀疏，簇与簇之间很难形成上述所谓的“连通分量”。因此随着  $N$  的增加，网络中形成的不连通簇的个数也随着增加，这就需要更多的移动网关来连通这些簇。这种情况下，移动网关所带来的优势会被随之增加的大量硬件耗费而抵消。随着  $N$

的进一步增加，虽然网络中簇的个数仍然继续增加，但是由于大量节点的密集部署，网络中形成的不连通簇的个数却反而减少，因此需要在这些不连通簇之间负责转发数据的移动网关个数也会减少。这也从另一方面验证了本文算法适合于密集部署的传感网络。

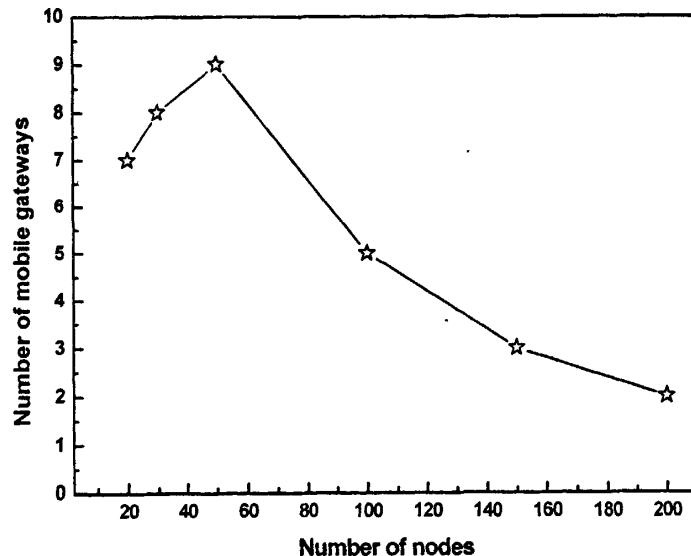


图 4.6 移动网关节点个数的变化趋势

#### 4.4 本章小结

本章我们通过在 CGCA 分簇算法的基础上加入少量资源和配置丰富的移动节点来担当簇与簇之间的网关，从而可以让普通节点从繁重的数据转发的任务中解放出来，达到节省能量和延长网络寿命的目的。除此之外，加入的移动网关节点可以进一步的提高簇的稳定性，这意味着重新分簇阶段不需要重新选举出新的网关节点。最后，由功能较强的移动节点代替容易出错的普通节点来担任网关可以提高系统的稳定性和容错性。

实验表明，在密集部署传感器节点的网络中，本算法产生的消息交换个数只是传统的基于最小标号分簇算法的 20% 左右。另外，在密集部署的传感网络中，本算法所取得的系统时间也是 LEACH 算法的两倍左右。但是本文算法的缺点是加入移动节点增加了额外的硬件耗费，因此在将来的研究中考虑在静态网络中加入移动节点所带来的性能折衷问题。

## 第五章 可变速率移动模型下的拓扑控制算法

移动传感网络（Mobile Sensor Networks,MSN）下的拓扑控制是一个难题。由于传感器节点的不断移动导致网络的底层拓扑时刻发生变化，因此拓扑控制算法通常需要周期性的执行来维持所需要的拓扑属性。

本章首先介绍了在移动传感网络下拓扑控制的背景和相关工作，其次本文提出了一个称为可变速率移动网络模型并在此模型的前提下，提出了一种类似于XTC的基于功率控制的拓扑控制算法，最后通过仿真模拟来验证所提出算法的性能。

### 5.1 背景介绍

最近由于一些智能移动设备的引入，见证了移动传感网络的兴起，例如可以将普通的传感器节点安装在这些移动设备上（例如 Robomote<sup>[103]</sup>）组成移动传感器网络 MSN。另外在军事和紧急搜救等应用中，由于被监视对象本身所具有的移动性也促使了移动传感网络的诞生和兴起。

如前所述，关于静态传感网络下的拓扑控制问题，目前有大量的文献提出了相应的解决方案，主要集中于功率控制和层次分簇两种方法。除此之外，在静态网络中拓扑控制算法通常是在系统启动时刻执行一次，以优化拓扑并为其他协议服务。但是由于移动传感网络中的节点本身所具有的移动性，也就是说节点的位置随着时间的推移是不断变化的，这种不确定性导致底层的拓扑结构也在不断变化，因此给拓扑控制问题提出了巨大的挑战。

一方面，为了达到想要的拓扑属性需要周期性的执行拓扑控制算法，而这会带来的大量的消息负载；而另一方面，节点的移动更有利与节点之间的负载平衡，这是因为在静态网络中，当节点  $u$  的发射半径是节点  $v$  的两倍时，节点  $u$  将比节点  $v$  更快的耗尽能量。而在移动网络中，由于节点的不断移动，因此节点的发射半径也在不断的发生变化，在这种情况下，节点之间取得负载平衡的概率会更大。因此，节点的移动对网络寿命的影响应该从整体的角度进行研究。

移动传感网络下的拓扑控制所带来的另外一个很重要的问题就是在拓扑控制算法所产生的消息负载和产生的拓扑质量之间存在一个性能折衷<sup>[104]</sup>。通常，移动传感网络下的拓扑控制算法需要周期性的执行。一方面为了取得较好的网络拓扑，例如一个连通的网络结构，需要更频繁的执行相应的拓扑控制重配置协议。但是每重新执行一次拓扑控制协议，就会带来大量的消息负载，这样就会带来大

量的能耗并降低系统寿命。

除此之外，移动模型的选择对移动传感网络下的拓扑控制问题也起着至关重要的作用。因为移动模型对网络中的某些属性，例如系统吞吐量和连通性，有着本质性的影响。而目前的相关移动模型通常过于理想化和理论化，导致相应的研究成果过于偏重理论结果，而导致在现实中缺少实用性。

最后，拓扑控制算法的正确执行一个最基本的前提就是所交换信息的一致性。但是最近的研究发现，节点的移动使得节点之间的时间同步变得非常困难，因此容易导致两种结果：一是节点之间信息的不一致性（view inconsistency），这会导致节点会选择过少的邻居节点；二是过时的信息（outdated information）会使节点选择过小的传输半径。Wu 等人<sup>[105,106]</sup>提出了局部视角一致性（local view consistency）和延迟移动管理（delay and mobility management）的方法分别对上述两个问题进行解决以保持网络的全局连通性。

目前对移动传感网络拓扑控制问题的解决，通常采用以下四种方法<sup>[107]</sup>：(1) 研究移动概率模型下的节点分布，然后转化为静态网络的拓扑控制算法。例如 Santi 等人<sup>[108-110]</sup>研究了不同移动模型下的节点静态分布，然后理论上求得了使得移动网络取得连通性的临界传输距离，也就是说赋给所有节点一个统一的传输半径使得网络取得一个高概率的连通性<sup>[111]</sup>。(2) 为了保证网络的连通性，对网络进行周期性的采样并执行相应的拓扑控制算法。(3) 执行一个在线算法（on-line algorithm）来动态的维护网络的连通性。(4) 对每个节点的发射半径增加一个称之为冗余传输距离 RTR（Redundant Transmission Range）的因子，把节点在采样时间之内的所有可能移动考虑进内，以保持网络的连通性。例如，文献<sup>[112]</sup>提出了 RTR 与中继区域相结合的方法，另外<sup>[113,114]</sup>所提出的缓冲区域（Buffer Zone）的概念在本质上也等同于采用 RTR。从本质上来说，移动传感网络的拓扑控制问题最终都需要转化为一个静态网络的拓扑控制来解决，区别只是设置的采样时间以及 RTR 等参数的不同。缺点是很难确定这些参数所采用的合适的值，目前大多是通过大量实验来评估确定。

## 5.2 相关工作

本节首先简要介绍一个移动传感网络的拓扑控制框架，然后详述一个跟本文工作有密切关系的移动传感网络拓扑控制算法，即由 Zhao 等人<sup>[115]</sup>提出的一个在固定速率移动网络模型 CRMN（Constant Rate Mobile Network）下的拓扑控制算法。

### 5.2.1 移动传感网络拓扑控制框架

Ling 等人<sup>[104]</sup>提出了一个在移动传感网络下的拓扑控制框架，该框架具体分为以下三个步骤：

- 1) 以  $T_s$  为采样周期定期的对网络进行采样。
- 2) 在采样点实施指定的拓扑控制算法，求出每个节点  $v_i$  的传输半径  $r_i$ 。
- 3) 重新设置节点的传输半径为  $r_i + RTR$ ，其中  $RTR$  为冗余传输半径。

在相邻采样点之间对系统达到完全连通的要求是造成移动传感网络拓扑控制问题困难的主要原因。这是因为小的采样周期意味着网络频繁的执行拓扑控制算法，导致大量的消息负载。而大的采样周期，为了达到完全连通性则需要一个大的  $RTR$ ，而这样会降低拓扑控制算法的能量有效性。因此文献<sup>[104]</sup>提出缓和对系统连通性的严格约束，即使得系统连通性  $P_{con}$  大于某个指定的阀值即可，并通过大量的实验研究在不同的应用环境中，三个重要的系统参数  $T_s$ ,  $RTR$  以及  $P_{con}$  之间的相互关系。

实验采用的是随机走（Random Walk）移动模型，而内嵌的拓扑控制算法是经典的 LMST 算法<sup>[28]</sup>。通过实验得出的系统参数设置规则如下：

- 1) 对某些要求完全连通性的应用，必须选择一个小的采样周期  $T_s$  以及大的  $RTR$  并满足  $RTR \geq 2T_s v_{max}$ ，其中  $v_{max}$  为节点的最大移动速度。
- 2) 对要求高概率连通的应用，大的  $RTR$  是必需的。此外在不损害连通性要求的前提下，通常选择一个大的  $T_s$  来抑制通信负载。
- 3) 对要求中等概率连通的应用，需要选择一个大的  $T_s$ 。另外必须仔细的选择  $RTR$  来平衡系统连通性和能量耗费。
- 4) 对连通性要求很低的应用来说，为了能量节省通常选择一个小的  $RTR$ 。此外对采样周期  $T_s$  的选择依赖于系统设计的其他要素。

### 5.2.2 CRMN 模型下的拓扑控制算法

Zhao 等人<sup>[115]</sup>提出了一种在 CRMN(Constant Rate Mobile Networks)移动模型下的拓扑控制算法。为了简化拓扑控制算法的复杂性，文献<sup>[115]</sup>所提出的算法并不是维持网络在整个系统生命期间的连通性，而是首先把系统生命期划分成若干个单位时间并维持网络在各个单位时间内的连通性。在 CRMN 模型下有  $n$  个移动节点，跟每个节点相关的是在单位时间间隔内的初始位置和终止位置，每个节点在由初始位置和终止位置所定义的线段上移动，并且在单位时间间隔内每个节点保持恒定的速度和方向。拓扑控制算法的目标是最小化节点的最大能量，使得此能量赋给所有的节点之后网络是移动连通的。

文献<sup>[115]</sup>中的算法所需要的基本符号如下：

- $\pi(u, v)$ : 节点  $u$  和节点  $v$  之间的能量阀值，即当节点  $u$  以此功率发射信号时，节点  $v$  刚好接收到，并且  $\pi(u, v) = d(u, v)^\alpha$ ，其中  $d(u, v)$  为节点  $u$  和节点  $v$  之间的欧几里德距离，而  $\alpha$  为位于[2,4]之间的路径衰减指数。
- $G^p(N)$ : 代表在网络  $N$  中给每个节点赋予功率  $p$  之后的导出子图。
- $G'(N)$ : 代表网络  $N$  的阀值图，它是一个无向的完全图，其中每条边  $(u, v)$  都有一个权重并且等于  $\pi(u, v)$ 。

Zhao 的算法利用了一个称为  $MINMAXGRAPH(N)$  的子例程<sup>[116]</sup>，该子例程求出了功率  $p$ ，使得给每个节点赋予功率  $p$  之后网络  $N$  是连通的并且满足  $p$  是最小的。该子例程利用了  $p$  一定为取自于  $G'(N)$  中某条边阀值的性质，因此可以首先对  $G'(N)$  中所有边的权重进行排序，然后利用二叉搜索求出  $p$  的值。

文献<sup>[115]</sup>提出了两种版本的拓扑控制算法，分别是决策版本和最优化版本，并且它们的时间复杂度分别是  $O(n^2(\log(n))^2)$  和  $O(n^4(\log(n))^2)$ 。其中决策版本指的是给定一个功率  $p$ ，然后判断给定网络  $N$  在每个节点都赋予功率  $p$  的前提下  $N$  是否在单位时间间隔内是移动连通的？而最优化版本则是指给定网络  $N$ ，求出功率  $p$  使得在每个节点都赋予  $p$  的情况下， $N$  在单位时间间隔内是移动连通的并且要求使得求出的  $p$  最小化。

两种版本的拓扑控制算法的本质都是利用了时间切割 (time slicing) 的概念，把单位时间间隔划分成更小的时间片 (time slots)，使得在两种拓扑控制算法下产生的时间片分别成为常连通时间片 (constant-connectivity time slot) 和常次序时间片 (constant-order time slot)。常连通时间片指的是给定一个功率  $p$  和网络  $N$ ，使得  $G^p(N)$  的边集合在给定时间片内保持不变。而常次序时间片指的是给定一个网络  $N$ ，在给定的时间片内  $N$  的所有边之间的权大小关系保持不变。两种算法的主要区别在于划分时间片的方法以及最终产生的时间片的个数的不同，这也是导致两种算法时间复杂度相差较大的原因。

从上面的论述可以看出，文献<sup>[115]</sup>提出的算法主要有以下缺点：(1): 整个算法是集中式算法，因此缺少可行性。并且由于算法在时间片切割过程中对时间同步的严格要求，很难把该算法修改为一个分布式算法。(2): CRMN 移动模型缺少通用性和实用性，因为在现实中假定所有的节点在单位时间间隔内匀速移动是不现实的，使得该算法的价值主要体现在理论研究方面。(3): 赋予所有的节点一个统一的发射功率对那些有临近一跳邻居的节点来说，能量有效性不高。

### 5.3 本文基础

为了解决 5.2.2 节中算法的不足之处，本文提出了一个在变速率移动网络模型下（Variant Rate Mobile sensor Network,VRMN）的拓扑控制算法。与文献<sup>[115]</sup>相同的是，本文算法也是维持网络在单位时间间隔内的移动连通性。在 VRMN 模型中，有  $n$  个移动节点并且跟每个节点相关的是它在单位时间间隔内的起始点和终止点。与之不同的是，在单位时间间隔之内，每个节点在由起始点和终止点所定义的线段上移动的速度是可变的，目标是在保持网络移动连通的前提下，为每个节点  $v_i$  赋予一个发射功率  $p_i$ ，使得  $p_i$  要尽可能的小。

本文首先提出了拓扑控制算法在 VRMN 模型下的集中式版本。该集中式算法包含三个例程：(1)：对每个节点  $v_i$ ，求出它的一跳邻居列表。这里只把在单位时间间隔内与  $v_i$  最大距离小于  $v_i$  最大发射半径的那些节点放入  $v_i$  的一跳邻居中。采用节点之间最大距离的原因是：在 VRMN 模型下，在单位时间间隔内的任一瞬间，由于节点位置的不确定性导致计算不出节点之间精确的距离。(2)：对每个节点  $v_i$ ，应用一个类 XTC 算法把那些最远的邻居节点删除，前提是这些最远节点可经  $v_i$  较近的邻居节点由一跳中继到达。这样在不损失网络连通性的前提下，减少了每个节点的发射功率。(3)：应用一个广度优先搜索算法检查产生的网络拓扑是否是不连通的，如果是，那么仅仅需要把最大发射功率赋予每个节点。这是因为，可能网络中节点的移动或者节点的个数过少而导致网络拓扑图中节点分布十分的分散，即使赋予每个节点最大发射功率也不能保证网络在单位时间间隔内是连通的。此时赋予每个节点最大的发射功率可以使得网络具有尽可能大的连通度。

最后为了实用性，本文利用异步的“Hello”消息交换，把上述的集中式算法修改为相应的分布式算法。两种版本的拓扑控制算法具有同样的例程，区别体现在实现细节的不同。除此之外，两种算法都是前摄的（proactive），因此它们都能够保证给定的移动传感网络实例在单位时间间隔之内都是移动连通的。

为了讨论的方便，本文假定移动传感网络中所有的节点都具有相同的最大发射半径，以  $TX_{range}$  来表示，并且很容易把我们的算法扩展到节点具有不同发射半径的情形。本文用  $NL(i)$  表示节点  $v_i$  的一跳邻居并用  $len(p_1, p_2)$  表示以  $p_1$  和  $p_2$  为端点的线段的长度。节点  $v_i$  在单位时间内移动的起始位置和终止位置分别用  $sp(i)$  和  $ep(i)$  来表示。

### 5.3.1 网络模型

本文在 VRMN 移动模型下的拓扑控制算法对基本的网络模型假定如下：

- 1) 节点随机均匀的部署在边长为  $L$  的正方形区域内。
- 2) 为了便于路由和 MAC 协议的设计，只考虑网络中的无向边。
- 3) 在执行拓扑控制算法之前，节点的位置事先知道，这可以通过执行定位算法或者让传感器节点配备 GPS 定位设备来做到。

### 5.3.2 能耗模型

节点  $v_i$  的传输功率用  $pw_i$  来表示，它与节点的发射半径  $r_i$  直接相关。这里本文采用一个被广泛采用的模型<sup>[104]</sup>：

$$pw_i = C \times r_i^\alpha \quad (5.1)$$

其中  $C$  代表常数， $\alpha$  表示位于区间[2,4]的能量衰减因子。 $C$  和  $\alpha$  与具体的传输环境有关。

基于公式(5.1)，在本文中一个称为  $PW(r)$  的函数用于把节点的传输半径  $r$  映射为相应的传输功率。

## 5.4 VRMN 下的拓扑控制算法

在本节，本文首先给出拓扑控制算法的集中式版本所采用的两个例程，紧接着给出集中式拓扑控制算法的流程，最后把它修改为相应的分布式拓扑控制算法。如前所述，移动传感网络系统生命期被划分为若干单位时间间隔，本文只需要保证在 VRMN 模型下的给定移动传感网络实例在某个特定的单位时间间隔内保持连通，然后对其他的单位时间间隔应用相同的拓扑控制算法。

### 5.4.1 计算每个节点的一跳邻居列表

本文所提出的拓扑控制算法在本质上是一个基于地理位置的算法，这通常需要节点的一跳邻居信息。由于在 VRMN 中节点位置的不确定性，因此节点之间的链路也不像静态传感网络那样稳定。除此之外，由于节点的移动速度是可变的，在某个时间点任意两个节点之间的精确距离不能够像文献<sup>[115]</sup>那样求出，因此必须事先把节点所有可能的移动位置考虑进去。对任意一个节点  $v_i$ ，只把与  $v_i$  的最大距离小于  $v_i$  的最大发射半径的那些节点放入  $v_i$  的一跳邻居  $NL(i)$  中。这样做的

目的是为了让  $v_i$  和  $NL(i)$  中的邻居之间形成的链路在整个单位时间间隔内是稳定的。因此问题的关键在于怎样计算任意两个节点  $v_i$  和  $v_j$  在单位时间间隔之内的最大距离  $dist(v_i, v_j)$ 。由于每个节点的移动路径是由它的起始点和终止点所决定的线段，因此  $dist(v_i, v_j)$  在本质上等同于分别属于  $v_i$  和  $v_j$  的移动路径（即两条线段）上任意两点的最大距离。文献<sup>[117]</sup>给出了一种简单的方法来计算，以引理 1 来表示：

**引理 1：**两条线段之间的最大距离等于分别属于两条线段的端点之间的距离最大值。

例如，如图 5.1 所示，给定任意两个节点  $v_i$  和  $v_j$  以及它们各自的起始点和终止点位置  $sp(i), ep(i), sp(j), ep(j)$ ，那么  $v_i$  和  $v_j$  在单位时间间隔内的最大距离  $dist(v_i, v_j)$  如下所示：

$$dist(v_i, v_j) = \max(len(sp(i), sp(j)), len(sp(i), ep(j)), len(ep(i), sp(j)), len(ep(i), ep(j))) \quad (5.2)$$

从图 5.1 可以明显的看出， $dist(v_i, v_j)$  等于  $len(ep(i), sp(j))$ ，它是属于  $v_i$  和  $v_j$  移动路径上端点之间距离的最大值。

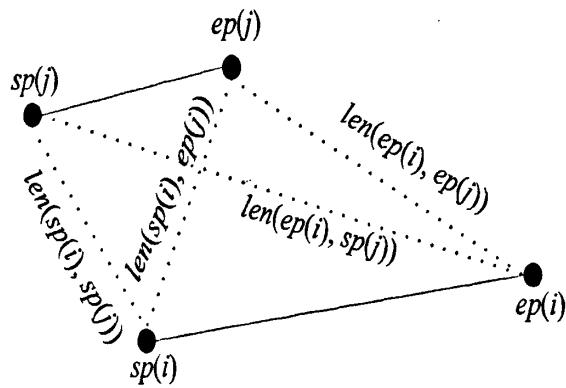


图 5.1 节点  $v_i$  和  $v_j$  之间的最大距离

#### 5.4.2 对每个节点应用类 XTC 算法

在计算出它的一跳邻居列表  $NL(i)$  之后，节点  $v_i$  以最大传输半径广播  $NL(i)$  至  $NL(i)$  中的所有邻居节点。节点  $v_i$  也从  $NL(i)$  中的一跳邻居节点中接收类似的邻居列表。紧接着，节点  $v_i$  通过删除  $NL(i)$  中最远的那些节点来简化  $NL(i)$  的大小，这些删除的最远节点可以通过  $v_i$  更为临近的邻居节点以每跳距离更短的方式中

继到达。也就是说，如果节点  $v_j$  和  $v_k$  都是节点  $v_i$  的一跳邻居，并且  $v_j$  也是  $v_k$  的一跳邻居，同时  $dist(v_b, v_k) < dist(v_b, v_j)$ ,  $dist(v_k, v_j) < dist(v_b, v_j)$ ，那么节点  $v_j$  就可以从  $v_i$  的一跳邻居列表集合  $NL(i)$  中删除掉。最后，通过调用函数  $PW(r)$  把节点  $v_i$  与其目前  $NL(i)$  中最远的邻居节点的距离  $r$  映射成发射功率，并用此功率作为节点  $v_i$  的最终传输功率。

通过此例程，本文能够在不损害网络原始连通性的前提下，尽最大可能的减少每个节点的发射功率。本文应用距离而不用像 XTC<sup>[61]</sup> 算法中的链路质量作为标准以删除最远的邻居节点，这是因为根据公式(5.1)，短的距离（也就是短的发射半径）通常意味着一个小的发射功率。应用类 XTC 算法删除  $v_i$  的一跳邻居节点  $v_j$  的示例见图 5.2。

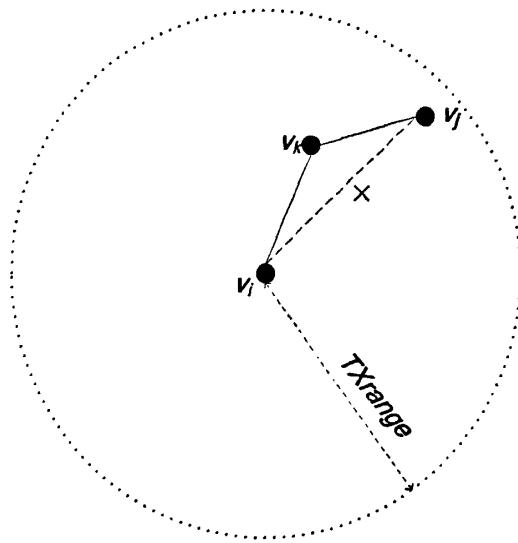


图 5.2 应用类 XTC 算法实例

### 5.4.3 VRMN 下集中式的拓扑控制算法

如 5.3 节所述，集中式拓扑控制算法包含三个例程，具体的流程图如图 5.3 所示。这里有一点需要注意，为了便于执行类 XTC 算法本文在计算每个节点一跳邻居列表时，对每个节点  $v_i$  邻居列表  $NL(i)$  中的节点按照它们到节点  $v_i$  的距离进行由大到小排序。下面我们证明此集中式算法是能量有效的。

**定理 1：**此集中式算法的时间复杂度是  $O(n^3)$

证明：步骤 7 中执行深度优先搜索算法只在某个初始节点执行一次，用邻接表表示它的时间复杂度是  $O(n^2)$ 。对每个节点  $v_i$ ，步骤 1 耗费  $O(n)$  的时间，而步骤 2 需要耗费  $O(n \log n)$  的时间对它的一跳邻居节点进行排序。难点在于步骤 4 的时间分析，对位于节点  $v_i$  的一跳邻居集合  $NL(i)$  中的每个节点，算法都需要遍历

在排序集合  $NL(i)$  中位于此节点之后的所有节点，因此步骤 4 的总时间复杂度为： $(n-1)+(n-2)+\dots+1=O(n^2)$ 。其它步骤的时间耗费明显是  $O(1)$ 。因此整个网络的时间复杂度是  $O(n^3)$ 。因此定理成立。

---

Input: An instance  $N$  of VRMN model with  $n$  moving nodes

Output: The power assignments to nodes  $\{p_1, p_2, \dots, p_n\}$  such that  $N$  is continuously connected in the unit time interval

Steps: Each node  $v_i$  independently executes the following steps except that step 7 is executed only once

- 1) Computes the initial one-hop neighbor list  $NL(i)$ .
- 2) Sorts the list  $NL(i)$  into  $(v_{i1}, v_{i2}, \dots, v_{im})$  in descending order by the distance  $dist(v_i, v_{ij})$ .
- 3) Broadcasts  $NL(i)$  to each node in  $NL(i)$ .
- 4) For  $v_{ij} \in NL(i)$  in ascending order do
  - a) If  $\{\exists k > j: v_{kj} \in NL(k), v_{ij} = v_{kj}, dist(v_k, v_{kj}) < dist(v_i, v_{ij})\}$  then  
Deletes  $v_{ij}$  from  $NL(i)$ .
  - Else  
Break out of the loop.
- 5) Let  $r_i$  be the distance between  $v_i$  and the first node in  $NL(i)$ .
- 6)  $p_i \leftarrow PW(r_i)$
- 7) Invokes breadth first search with  $v_i$  as the source to determine whether the topology induced by these powers is disconnected.
  - a) If the topology is disconnected then all the nodes are assigned to the full power

$$p \leftarrow PW(TX_{range})$$


---

图 5.3 VRMN 下集中式拓扑控制算法流程

#### 5.4.4 VRMN 下分布式的拓扑控制算法

在一个移动传感网络中，选择一个中央主机（Central Host）来执行拓扑控制算法并且把最终产生的发射功率赋于每个节点的做法往往是不现实的。一个好的并且可行的拓扑控制算法应该让每个节点根据其收集到的信息独立的作出自己的决定。

同对应的集中式算法一样，分布式的拓扑控制算法也具有相同的三个例程，只是它们的实现过程是不相同的。例如，在例程 1 中，每个节点  $v_i$  需要独立的计

算自己的一跳邻居列表。同时，例程 3 中需要一个分布式版本的深度优先搜索来检查最终的网络拓扑是否连通。VRMN 模型下分布式拓扑控制算法的流程见图 5.4。

需要指出的是，当节点  $v_i$  没有收到来自节点  $v_j$  的“Hello”消息时，节点  $v_j$  不能成为  $v_i$  的邻居节点。即使在单位时间内的某个时间点节点  $v_i$  和  $v_j$  之间的距离小于最大发射半径  $TX_{range}$ ，这是因为没有收到“Hello”消息说明两个节点之间的链路不是稳定的，也就是说它们之间的链路并不是在整个单位时间间隔内都存在。

---

Input: An instance  $N$  of VRMN model with  $n$  moving nodes

Output: The power assignments to nodes  $\{p_1, p_2, \dots, p_n\}$  such that  $N$  is continuously connected in the unit time interval

Steps: Each node  $v_i$  independently executes the following steps except that step 8 is executed only once

- 1) Broadcasts a “hello” message containing its own  $id$ , starting position  $sp(i)$  and ending position  $ep(i)$  by using maximum transmission radius  $TX_{range}$ .
- 2) When receiving a “hello” message from  $v_j$ , then  $v_i$  computes  $dist(v_i, v_j)$  and if  $dist(v_i, v_j)$  is less than or equal to  $TX_{range}$ , adds  $v_j$  to  $v_i$ ’s neighbor list  $NL(i)$ .
- 3) Sorts the list  $NL(i)$  into  $(v_{i1}, v_{i2}, \dots, v_{im})$  in descending order by the distance  $dist(v_i, v_j)$ .
- 4) Broadcasts  $NL(i)$ , receives analogous lists from its neighbors in  $NL(i)$  and disregards lists from other nodes.
- 5) For  $v_j \in NL(i)$  in ascending order do
  - a)  $v_i$  broadcasts a “probe” message containing the ids of  $v_i$ ,  $v_j$  and  $dist(v_i, v_j)$ , then it waits for the reply.
  - b) If  $\{\exists v_k \text{ receives such a message: } v_k \in NL(i), dist(v_i, v_k) < dist(v_i, v_j), dist(v_k, v_j) < dist(v_i, v_j)\}$ , then node  $v_k$  responds a “yes” reply and at the same time  $v_i$  also does the same process when receiving similar “probe” messages from other nodes
  - c) If  $v_i$  receives a “yes” reply, then
    - Delete  $v_j$  from  $NL(i)$ .
- 6) Let  $r_i$  be the distance between  $v_i$  and the first node in  $NL(i)$ .
- 7)  $p_i \leftarrow PW(r_i)$
- 8) Invokes the distributed layered breadth first search algorithm in [12] to determine whether the topology induced by these powers is disconnected.
  - a) If the topology is disconnected then simply assign full powers to all nodes

$$p \leftarrow PW(TX_{range})$$


---

图 5.4 VRMN 下分布式拓扑控制算法流程

下面来看一下分布式拓扑控制算法的性能表现。

**定理 2：**分布式拓扑控制算法的时间复杂度同样为  $O(n^3)$

证明：步骤 8 在整个算法中只是执行一次，采用<sup>[118]</sup>中提出的分布式广度优先搜索算法，它的时间复杂度为  $O(n^3)$ 。对每个节点  $v_i$ ，步骤 1, 2, 3, 4, 6 和 7 一共耗时  $O(n \log n)$ 。下面集中分析步骤 5 的时间复杂度。对每个节点  $v_i$ , 5.a 很明显只有  $O(1)$  复杂度，然而 5.b 和 5.c 都具有  $O(n)$  的时间复杂度，这是因为节点  $v_i$  最多接收  $n$  个“回复”信息和最多  $n$  个“应答”信息。这样，步骤 5 “贡献”  $O(n^2)$  的时间复杂度。因此系统总的运行时间复杂度为  $O(n^3)$ 。定理成立。

分布式版本的拓扑控制算法具有如下优点：

- 1) 简单：不需要复杂的硬件和软件，这与移动传感器节点只具有简单的计算和通信能力的特征相吻合
- 2) 前摄：在每个单位时间间隔的最开始就已经把每个节点的所有可能移动考虑进内，因此能够保证移动传感网络在单位时间间隔内的每个瞬间都是连通的

#### 5.4.5 性能评估

本节将评估 VRMN 模型下的分布式拓扑控制算法的性能表现，采用的仿真工具为 OMNET++<sup>[87]</sup>。并且由于没有在 VRMN 模型下的相关拓扑控制算法被提出，因此只提供本文算法的性能度量。除非另外说明，假定  $n$  个传感器节点随机均匀的分布在边长  $L$  为 100 的正方形区域中。为了防止传感器节点移动到部署区域之外而变得不可控制，本文对所有节点移动路径的长度（也就是由节点的起始点和终止点定义的线段的长度）赋予一个统一的上界，称之为  $MR$ ，很明显它的值应该严格的小于  $L$ 。为了测量算法的性能，确立了两个度量标准：( i ) 节点的平均传输半径 ( ii ) 连通度，被定义为最大连通分量的大小占整个网络大小（也就是  $n$ ）的比例。本文研究这些度量标准相对于节点个数  $n$  以及移动路径最大长度  $MR$  的变化趋势。对每次仿真场景，算法都被随机执行十次并对结果取平均值。

在 VRMN 模型下，图 5.5 和 5.6 分别显示了应用例程 1（计算一跳邻居并排序）和例程 2（应用类 XTC 算法删除最远节点）对某个网络实例  $N$  执行之后的拓扑图，其中  $n=10$ ,  $MR=25$ ,  $TX_{range}=40$ 。在  $N$  中，红色的点划线代表传感器节点的移动路线。我们可以明显的看出，图 5.6 中显示的拓扑图要比 5.5 中对应的拓扑图稀疏的多，这是因为在 5.6 中应用了一个类 XTC 算法来“裁剪”依附在每个节点上的最远链路边。

拓扑控制算法的首要目标就是减少节点的发射半径，这直接反映了拓扑控

制算法的质量和有效性。图 5.7 显示了本文拓扑控制算法在减少传输半径方面随着节点个数的变化趋势。具体的参数设置如下： $TX_{range}=40$ ,  $MR=25$ 。图 5.7 中的初始平均半径定义为执行完例程 1 之后的所有节点的发射半径的平均值，而最终平均半径定义为执行完整个拓扑控制算法之后的平均节点发射半径。

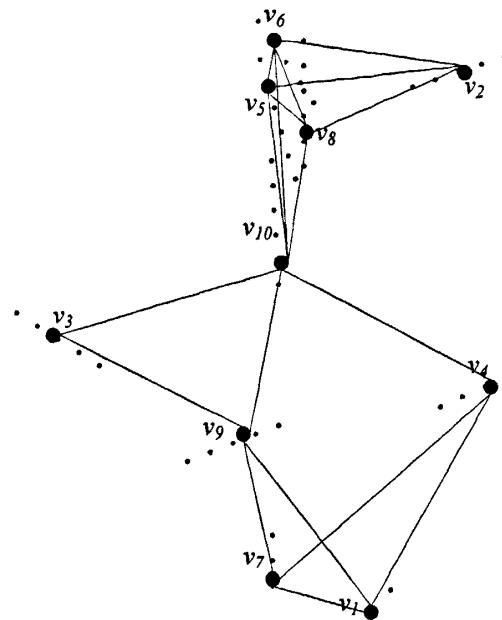


图 5.5 算法执行完例程 1 的初始拓扑

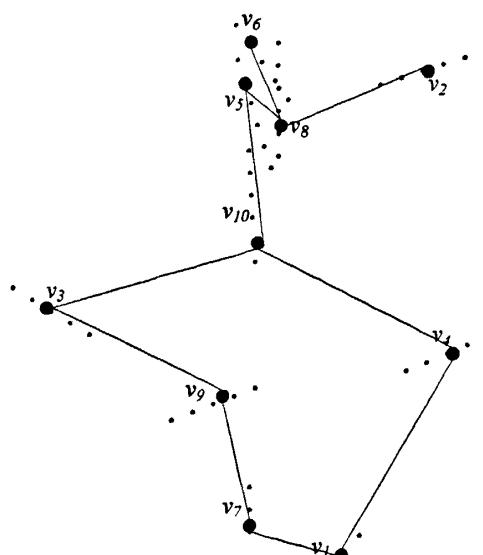


图 5.6 算法执行完例程 2 的最终拓扑

从图 5.7 可以看出，随着节点个数  $n$  的增加，两条曲线均呈下降趋势。这是因为随着节点个数的增加，节点的部署更为密集导致节点之间更为邻近，因此导致更小的发射半径。另一方面，随着  $n$  的增加，最终平均半径的曲线相比于初始平均半径下降的更为明显。这是由于节点个数的增加将会导致一个节点有更多的邻居节点，此时本文算法的例程 2，也就是类 XTC 算法将会有更多的选择机会来删除每个节点最远的那些邻居节点。

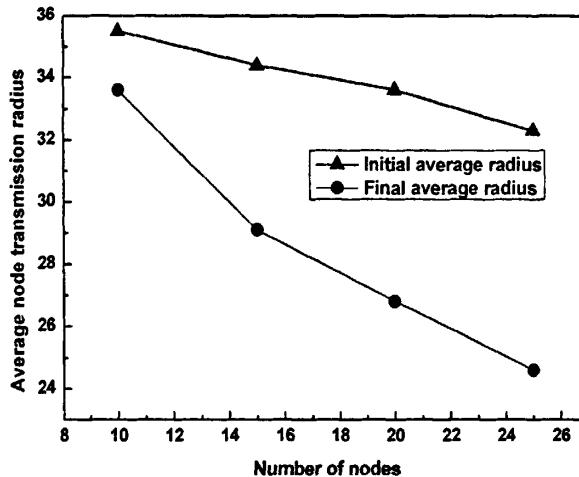


图 5.7 本文算法在减少节点发射半径方面的性能表现

如前所述，拓扑控制的一个主要目标就是维持网络的连通性。本文算法利用一个称为连通度的度量标准来评估算法随着  $MR$  的变化在连通性方面的性能表现，如图 5.8 所示。很明显，在一个连通的网络中，连通度的值为 1，而在一个非连通的网络中，连通度为位于区间(0,1)的一个实数值。仿真实验的参数设置如下： $TX_{range}=35$ ,  $n=15$ 。

从图 5.8 可以看出，连通度随着  $MR$  的增长而下降。这是因为在给定的  $TX_{range}$  下，较大的  $MR$  将会在节点之间产生不稳定的链路，从而容易导致网络出现不连通的情形。特别应该注意的是， $TX_{range}/2$  是  $MR$  的一个临界值，这是由于如果两个节点沿着同一条直线做相向运动，此时二者在单位时间之内的最远距离恰好等于  $TX_{range}$ 。在这种情形下，很难维持一个移动连通的网络。例如，当  $MR=15$  时，连通度为 0.97，而当  $MR=20$  时，连通度仅仅为 0.75。因此本文的仿真实验也为实际的移动节点部署提供一个背景和参考。

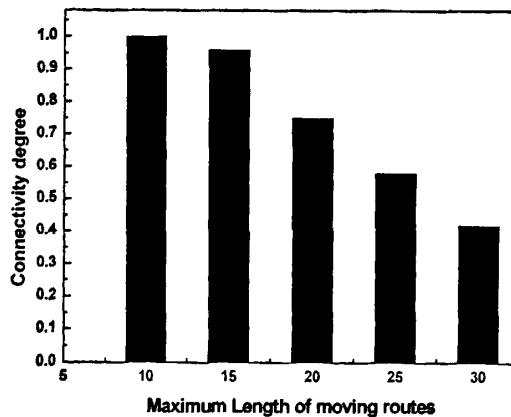


图 5.8 算法的连通度性能表现

## 5.5 本章总结

在本章中，我们主要研究了移动传感网络中的拓扑控制算法。本文首先阐述了一个可变速率的移动网络模型 VRMN，然后在此模型的基础上，我们提出了一种能量有效的拓扑控制算法，并分别给出了它的集中式和分布式版本，它们都具有  $O(n^3)$  的多项式时间复杂度。并且通过仿真实验对算法的性能进行了验证。

由于算法在单位时间间隔的最开始就考虑了节点的所有可能移动，因此算法能够保证网络在整个单位时间间隔内都是移动连通的。在将来的研究中，需要研究怎样将网络系统生命期划分为单位时间间隔。最后，由于移动模型对移动传感网络拓扑控制问题的重要性，需要提出更为灵活、现实的移动模型并对这些移动模型对拓扑控制问题的影响作进一步的研究。

## 第六章 大规模移动传感网络下一种能量有效的分簇算法

在一个大规模密集部署的移动传感网络中，如果采用基于功率控制的拓扑控制算法，那么在其产生的平面拓扑结构中，不论是采用主动或者被动路由，都会带来大量的计算和通信负载以及端到端时延。因此为了网络的可扩展性以及取得较好的性能保证，通常采用分簇算法来对这种大规模移动传感网络进行拓扑控制。

本章首先介绍了大规模移动传感网络下分簇算法的基本背景知识，然后提出了一种能够在节点移动时稳定簇结构的分簇算法，最后通过仿真实验对本文算法进行了性能验证。

### 6.1 背景知识

如前所述，在一个大规模移动传感网络中由于传感器节点的不断移动导致其底层的拓扑结构不断发生变化，因此在大规模移动传感网络下的拓扑控制是一个难题。由于分簇算法能够形成一个层次结构以及能够较好的支持网络的可扩展性和较好的维护拓扑结构的稳定性，这就使得分簇算法比较适合于对这种大规模密集移动传感网络进行拓扑控制。

一个分簇算法通常分为两个阶段：簇形成和簇维护。簇形成关注的是如何按照某种规则选择簇头以及簇内成员以形成初始的簇结构。簇维护指的是如何在分簇算法的后续阶段维护簇结构以达到某种目标或者适应拓扑的变化。在一个静态的传感网络中，簇维护指的通常是为了达到负载平衡或者当某个簇头失效时重新选举簇头。而在一个移动的传感网络中，簇维护的含义则是及时的调整簇结构以适应由节点移动造成的底层拓扑的不断变化，因此在一个移动传感网络的分簇算法中，如何减少簇维护的费用就显得至关重要。

在簇形成阶段，节点和节点之间通常需要交换一些信息来使得选举出的簇头在其一跳邻居范围内具有特定的属性，例如最小标号 LIC<sup>[45]</sup>或者最大节点度 HCC<sup>[47]</sup>。在静态网络中，由于传感器节点的位置是固定不变的，因此这种基于簇头特定属性的分簇算法在簇形成阶段不需要任何附加的条件。然而，在一个移动网络中，由于节点位置的不断变化，为了节点之间交换信息的正确性以及选举出的簇头满足预先定义的属性，这种基于簇头特定属性的分簇算法<sup>[50]</sup>在簇形成阶段通常假设一个“冰冻”时间 (frozen time)，也就是说在簇形成阶段节点之间互相交换信息的这一段时间，节点是静止的或者相对静止。然而，这种假设往往

是不现实的，因为在现实环境中，例如在随机走移动模型中<sup>[86]</sup>，节点时时刻刻都在随机的移动。为了消除对“冰冻时间”的依赖，RCC(Random Competition-based Clustering)<sup>[119]</sup>算法提出了“随机声明”(random claim)机制。随机声明机制规定每个节点都随机设置一个回退时间(backoff time)，在回退时间到达之后，任何一个节点都可以声明为簇头。然而，如果在回退时间到来之前就收到其他节点的“簇头声明消息”，那么此节点就变成相关簇头节点的簇内成员。本文算法也不依赖于“冰冻时间”，详见6.2节。

在移动传感网络分簇算法的簇维护阶段，由于节点的移动导致底层拓扑不断发生变化，因此簇维护指的是在重新分簇阶段，如何及时调整簇结构以适应这种变化。同时为了分簇算法的能量有效性和延长传感网络寿命，簇维护的费用要尽可能的低。要使簇的维护费用低，在本质上就必须消除重新分簇所带来的“涟漪”效应，也就是说在重新分簇阶段只需要做局部的簇结构调整就可以适应拓扑的变化，而不是让所有的簇结构都全局调整，这样就可以把节点的移动给分簇算法带来的影响达到最小化。簇维护的费用跟簇结构的内在属性有着本质的联系，例如，在节点移动的前提下，相比一跳簇的分簇算法，K跳簇分簇算法的簇结构由于簇头和簇内成员之间的距离更远，为了调整拓扑结构就会涉及到更多的节点，以至于簇结构难以维护，因此本文所提出的分簇算法也是基于一跳簇。此外，在簇维护的过程中，频繁的簇头和簇内成员更换将会产生大量的计算和消息负载，因此一个好的分簇算法应该最小化这种簇头和簇内成员更换，本文能够较好的解决这一问题，详见6.2节。

## 6.2 本文算法

为了解决大规模移动传感网络分簇算法所带来的高负载问题，本文提出了一种能量有效的分簇算法，并详细阐述了分簇算法的具体簇形成和维护过程、性能分析以及仿真模拟。

### 6.2.1 簇形成过程

下面首先给出本文分簇算法的基本假设和前提：

- 1) 传感器节点的部署区域为边长为  $L$  的正方形。
- 2) 移动节点分为两类，一种是配备 GPS 定位设备的移动节点，另外一种是不配备任何定位设备的移动节点。
- 3) 所有的节点都具有相同的初始能量供应  $e$  和最大传输半径  $TX_{range}$ 。

本文的分簇算法采用了同 GAF 算法<sup>[90]</sup>相同的划分单元格方法，即首先将部

署区域划分为小的边长为  $r$  的正方形单元格，由图 3.1 可以得出  $r \leq \frac{TX_{range}}{\sqrt{5}}$ ，

本文  $r$  选择接近与此值的一个整数值。随后的分簇过程比较简单：首先配备有 GPS 定位设备的移动节点移动到每个小的正方形单元格的中心位置担当簇头节点，选择中心位置是为了让簇头和其内部的簇内成员之间有比较近的簇内传输距离。这些簇头节点随后周期性的发送“Hello”消息，消息中包含自己的 ID 以及一个简单的簇头声明。随后其他的没有配备 GPS 定位设备的移动节点就可以随机的或者按照某种移动模型在部署区域内移动，在移动的过程中，由于划分单元格的方式会使每个单元格中的任意节点都能够与其相邻单元格内的任意节点直接通信，因此每个节点会收到多个簇头节点的“Hello”消息，此时节点选择“Hello”消息信号最强的节点作为自己的簇头节点，同时给相应的簇头节点发送一个包含自己 ID 以及 CID（簇头的节点标号）的“Hello”消息，确认自己成为相应簇头的簇内成员。同时，簇头根据接收到的簇内成员“Hello”消息将其所有的簇内成员加入到自己的信息表中，信息表的结构见 6.2.2。经过一段时间，初始的簇结构如图 6.1 所示。

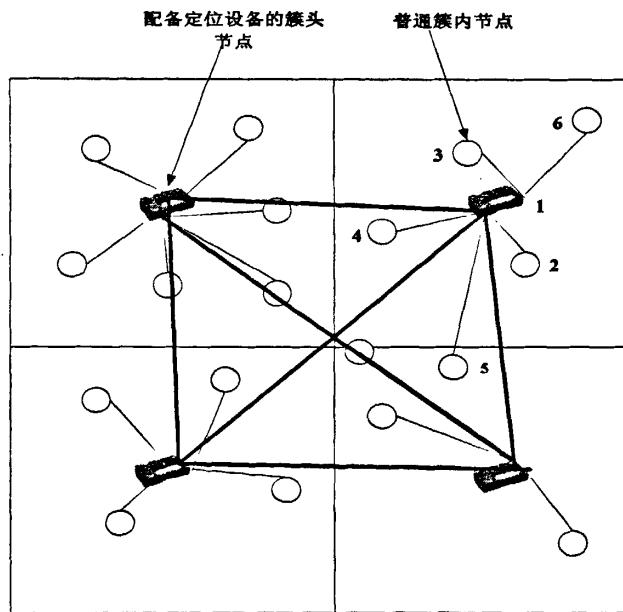


图 6.1 本文的初始簇结构

由上面的论述以及图 6.1 可以看出，本文的分簇算法具有下面的优势：

- 1) 本文中只要求少量的移动节点具有定位能力，而大部分的移动节点只需要具有感知信号强度的能力即可，因此从整体上说对节点的硬件能力要

求不高。

- 2) 本文形成的簇与簇之间不重叠 (Non Overlapping)，也就是说一个簇内成员只属于一个簇头，这对简化网络拓扑结构十分有利。不重叠簇的好处有二：一是不需要网关来进行簇与簇之间的连接，从而减少了能量负载；二是有利于后续的簇维护，这是因为此时一个簇内成员不会被多个簇头节点“覆盖”，因此一个簇内成员的改变不会同时引起多个簇结构的调整。也就是说把一个节点移动可能对簇结构的影响局限在一个小的范围内，从而形成了一个较为稳定的簇结构。
- 3) 由于本文算法的簇头节点是事先确定好的，并且不需要簇头节点在其一条邻居内具有某种特殊的属性，因此本文从根本上消除了在移动网络中的簇形成阶段对“frozen time”的依赖。
- 4) 本文分簇算法在初始的簇形成阶段实现过程简单，每个移动节点只需要监听簇头的“Hello”消息并通过比较消息的信号强弱，然后发送自己的一条消息加入到相应的簇头就可以形成初始的簇结构。由于每个节点只需要发送和接收常数个消息，因此在簇形成阶段，本文分簇算法的时间复杂度和消息复杂度均为  $O(N)$ ，从而较大的节省了节点的能耗并取得了较好的能量有效性。
- 5) 本文分簇算法在经过初始簇形成阶段之后的由簇头节点组成的骨干网是连通的。由于采用的划分单元格的方式使得相邻的两个单元格内部的任意两个节点都可以直接进行通信，因此相邻单元格的任意两个簇头是彼此连通的，这样整个由簇头节点构成的骨干网也是连通的。

值得注意的是，本文的分簇算法并不是依赖于节点的地理位置，而是依赖于接收到的信号强弱，这在现实中往往更容易做到。因此本文算法不像 GAF 算法那样，每个小的单元格构成一个簇，例如图 6.1 中位于区域右下角的节点 5 属于右上角单元格簇头的簇内成员，这是因为节点之间位置的远近并不能一定代表信号的强弱。

### 6.2.2 簇维护过程

在一个移动传感网络中，随着节点的不断移动，底层的拓扑也在不断发生变化，因此形成的簇结构应该不断更新以适应这种变化。这就使得在一个分簇算法中，簇维护显得比簇形成更为重要。簇维护阶段，每个移动节点需要一个称为“信息表”的内存空间来记录相应的信息。其中，普通簇内成员的信息表比较简单，只需要包含自己的 ID, CID(其簇头的 ID)，以及自己的身份（也就是簇内成员）。而簇头节点通常需要对整个簇内所有成员进行协作，它的信息表包含了更多的信

息,图 6.1 中位于右上方单元格内簇头 1 的信息表如表 6.1 所示,其中 Entry Update Time 对于簇头节点指的是上次扫描信息表的时间,而对于簇内成员它指的是它的簇头节点最近一次收到它发送”Hello”消息的时间。

表 6.1 簇头 1 的信息表

| Node Id | Status        | Entry Update Time(s) |
|---------|---------------|----------------------|
| 1       | clusterhead   | 100                  |
| 2       | clustermember | 99.5                 |
| 3       | clustermember | 100.2                |
| 4       | clustermember | 100.6                |
| 5       | clustermember | 99.8                 |
| 6       | clustermember | 99.2                 |

为了能够及时的更新簇结构,每个移动节点都每隔  $T_s$  发送一个”Hello”消息以及时通报节点的移动所可能带来的变化,在本文中  $T_s=1s$ 。其中簇内成员的”Hello”消息包括自己的 ID 以及 CID,而簇头的消息包含自己的 ID 以及簇头身份声明。在移动传感网络中,当簇内节点移动到新的位置但不更换簇头时,簇头只需要在其信息表中更新相关簇内成员节点的 Entry Update Time,但不需要对簇结构进行更新维护。引起分簇算法进行簇维护是事件触发的,可能引起簇维护的有以下四种事件:簇头失效、簇内成员失效、加入新的移动节点以及簇内节点更换簇头。下面分别就这几种事件的簇维护过程进行介绍。

1) 簇头失效:当某个簇内成员  $A$  发现自从上次收到簇头  $B$  的”Hello”消息时刻算起,又经过预先定义的阀值  $T_c$  之后还没有收到  $B$  的簇头声明消息时,它认为簇头节点  $B$  失效并把  $B$  从信息表中删除。由于单元格的划分方式,此簇内节点肯定会收到其他簇头节点的簇头声明消息,它选择目前具有信号最强的相应节点  $C$  作为自己的簇头。节点  $A$  在随后的”Hello”消息声明中额外的把簇头  $B$  失效的信息包含在内,当簇头节点  $C$  收到  $A$  的消息时需要做两件事情:一是把簇头  $B$  失效的信息经过骨干网传递至汇聚节点,然后汇聚节点需要另外补充一个配备 GPS 的移动节点到相应的单元格去担当簇头。二是当它收到节点  $A$  的消息时,发现自己信息表中并没有节点  $A$  的信息,但是簇头  $C$  发现节点  $A$  的消息中的 CID 等同于自己的 ID,因此这表示节点  $A$  选择了自己作为它的簇头,于是簇头  $C$  在信息表中添加一项并把  $A$  的相关信息填入在内。

2) 簇内成员失效:簇头节点每隔时间  $T_i$  对信息表进行扫描,当某个簇头节点  $A$  发现某个簇内成员  $B$  的 Entry Update Time 经过预先定义的时间间隔  $T_m$  还没有更新时,此时簇头  $A$  就认为节点  $B$  失效,并将其所对应的信息表项从信息表中删除。

3) 加入新的移动节点: 当网络中加入新的移动节点  $A$  时, 它根据接收到的簇头声明消息的信号强弱选择信号最强的节点  $B$  作为自己的簇头, 并在随后的”Hello”消息中声明自己的 ID 以及 CID, 簇头  $B$  接收到此消息时, 检查自己的信息表发现没有记录节点  $A$  的相关信息项, 但是发现消息中的 CID 等同于自己的 ID, 这表示有新的节点加入到自己的簇中, 于是簇头节点  $B$  在信息表中添加一项并把  $A$  的相关信息填入在内。由于单元格的临近关系, 可能与簇头  $B$  相邻的簇头  $C$  也接收到  $A$  发送的消息, 但是  $C$  发现消息中的 CID 与自己的 ID 并不相等, 因此簇头  $C$  舍弃此消息不做任何处理。

4) 簇内节点更换簇头: 这种情况是由簇内节点的移动而引起的, 当某个簇内成员节点  $A$  移动到某个位置发现从簇头  $C$  接收到的簇头声明信号强于其原来簇头节点  $B$  的信号时, 节点  $A$  立刻发送”Hello”消息并把相应的 CID 改为节点  $C$  的 ID。这样簇头  $B$  在接收到此消息后发现 CID 与自己的 ID 并不相等, 于是簇头  $B$  将节点  $A$  的相关信息项删除, 而簇头  $C$  则将  $A$  的相关信息作为新的一项添加到自己的信息表中。

### 6.2.3 性能评估

本节将评估本文分簇算法的性能表现, 采用的仿真工具为 OMNET++<sup>[87]</sup>。为了测量算法的性能, 确立了两个度量标准: (i) 仿真期间分簇算法所产生的消息个数 (ii) 系统寿命, 定义为从系统开始运行到第一个移动节点失效所经历的时间 (III) 簇内成员平均寿命, 定义为一个簇内节点被同一个簇头节点覆盖所经历时间的平均值, 它可以较好的刻画了簇结构的稳定性。本文研究这些度量标准相对于节点个数  $n$  以及节点最大移动速度  $V_{max}$  的变化趋势。对每次仿真场景, 算法都被随机执行十次并对结果取平均值。仿真实验所用的参数设置如表 6.2 所示。

表 6.2 仿真参数设置

| System Parameter | Value |
|------------------|-------|
| $e$              | 2J    |
| $TX_{range}$     | 100m  |
| $L$              | 1000m |
| Simulation Time  | 2000s |
| $T_c$            | 2s    |
| $T_i$            | 0.2s  |
| $T_m$            | 2s    |

一个分簇算法在簇的形成和维护时不可避免的产生跟分簇有关的消息，因此在分簇算法运行过程中所产生的消息个数是衡量算法性能和有效性的一个十分重要的因素。本文算法和 HCC 算法<sup>[47]</sup>在节点最大移动速度  $V_{max}$  下所产生的消息个数变化趋势如图 6.2 所示，其中节点个数  $N=500$ ，采用的是随机走移动模型。

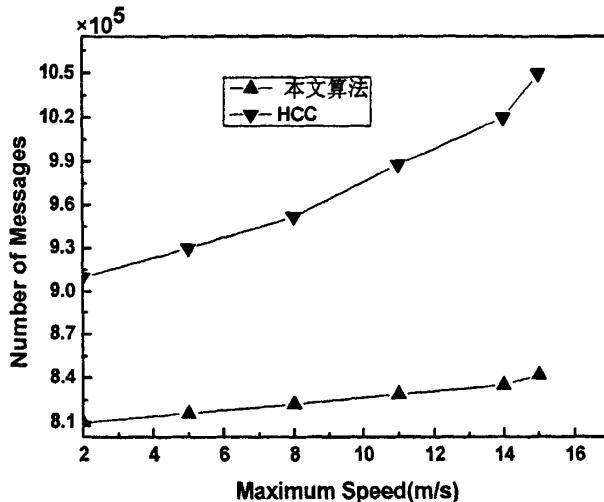


图 6.2 分簇算法产生的消息个数随最大速度的变化趋势

从图 6.2 可以看出，两种分簇算法所产生的消息个数曲线随着节点最大移动速度的增加呈上升趋势。这是显而易见的，因为随着节点的移动速度提高，簇内成员和簇头之间的隶属关系也变得更不稳定，因此为了维护簇结构，所需要的交换消息个数也随之增加。从图中还可以看出，HCC 算法所产生的消息个数要远大于本文算法并且随着节点移动速度的增加，本文算法产生的消息个数缓慢增长而 HCC 算法呈急剧增长。这是因为在 HCC 算法中，当节点的移动速度较快时，节点度数的变化也非常快，为了维持 HCC 算法中簇头在其一跳邻居内的最大节点度属性，簇结构更新的速度就必须非常快，而这会在整个网络中产生“涟漪”效应，从而导致产生过量的消息个数。而本文算法随着节点的移动，产生的消息个数大多来自于移动节点的更换簇头，这种情况下，相应的移动节点只需要额外发送一个“Hello”消息并修改相应的 CID 就可以解决，因此簇结构的更新只是局部进行的，从而导致产生的消息个数随着节点移动速度的提高而缓慢增长。最后值得注意的是，在节点移动速度较慢时，本文所产生的消息个数也远远小于 HCC 算法。这是因为本文的分簇算法的簇头是事先确定好的，而 HCC 算法需要在节点之间发送大量的消息并通过竞争来选举出合适的簇头。

本文算法和 HCC 算法在簇内成员平均寿命方面随着节点最大移动速度的变

化趋势如图 6.3 所示，其中节点个数  $N=500$ ，采用的是随机走移动模型。

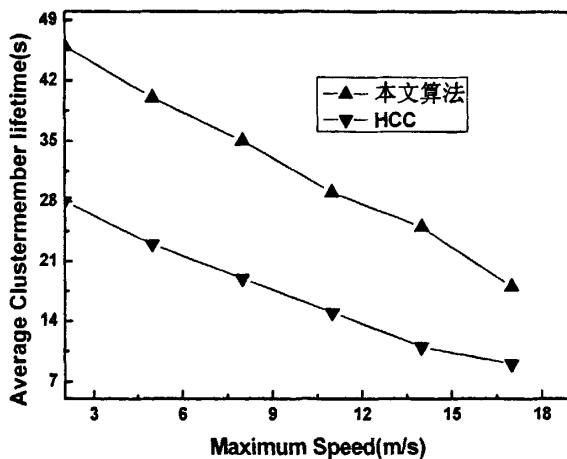


图 6.3 簇内成员平均寿命随最大移动速度的变化趋势

从图 6.3 可以看出，本文算法和 HCC 算法的簇内成员平均寿命随着节点移动速度的提高而减少，这是显而易见的，因为节点的移动速度越快，簇内成员更换簇头的频率就越快，因此簇内成员的平均寿命就越短。另外，在相同的节点移动速度下，本文算法的簇内成员平均寿命要明显长于 HCC 算法。这是因为在 HCC 算法中，节点的移动导致节点的度数变化比较快，从而导致簇内成员频繁的更换簇头，因此 HCC 算法的簇内成员平均寿命比较短。而在本文算法中，当一个簇内节点移动时，它只对相邻的几个簇有影响，也就是说簇的结构比较稳定，从而产生较长的簇内成员平均寿命。

本文算法在系统寿命方面在两种移动模型下随着节点个数的变化趋势如图 6.4 所示。本文采用了 4.3.2 节的能耗模型，其中  $V_{max}=7\text{m/s}$ ，随机路点模型下的停顿时间为 3s。由于簇头节点比簇内成员承担更多的任务，它的能量更容易耗尽。因此这里的系统寿命指的是所有单元格中的簇头节点从开始运行到耗尽能量所经历时间的最小值，并且本次实验不受仿真时间的约束。

从图 6.4 可以看出，随着节点个数的增加，本文算法在两种移动模型下的系统寿命都明显下降，这是因为节点个数的增加使得簇头拥有更多的簇内成员，从而消耗更多的能量，因此产生更短的系统寿命。另外从图 6.4 还可以看出，在相同的节点个数条件下，随机走模型比随机路点模型能够取得更长的系统寿命。这是由于随机路点模型能够带来节点的“中心聚集”效应，因此在区域中心位置附近单元格中的簇头节点会覆盖比随机走模型更多的簇内成员，导致在随机路点模型下的簇头节点更容易耗尽能量，从而使得系统寿命更短。

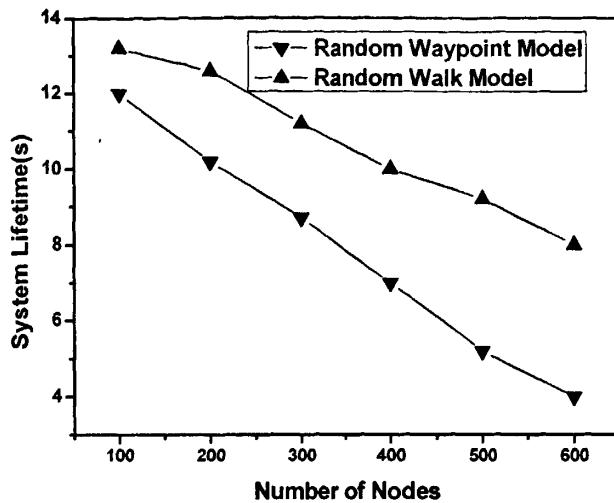


图 6.4 系统寿命随着节点个数的变化趋势

### 6.3 本章总结

在本章中，我们主要研究了大规模部署移动传感网络下的拓扑控制问题。本文提出了一种能量有效的分簇算法。本文的初始簇形成阶段比较简单，具有  $O(N)$  的时间复杂度和消息复杂度，并且在分簇过程中不依赖于节点的地理位置信息。而在簇维护阶段，簇的维护是事件触发并且异步进行，不会产生“涟漪”效应，因此本文算法能够取得一个较为稳定的簇结构以适应节点的移动和底层拓扑的变化。

除此之外，由于采用的单元格划分方式能够使得相邻的簇头可以直接通信，这样当某个簇头失效时，它的簇内成员可以非常容易的划分到别的相邻簇头中，因此本文算法具有较好的容错性。将来的研究侧重于两个方面：一是考虑将本文算法与数据收集算法相结合，也就是研究怎样在簇头节点形成的骨干网上选择路径将感知数据路由至汇聚节点；二是在某个簇头节点失效时，怎样及时快速的更换簇头。

## 第七章 总结与展望

本章对全文的研究内容、研究方法以及所做的贡献和创新进行全面的总结，同时展望拓扑控制研究所需进一步的工作。

### 7.1 本文的工作

本文系统详细地研究了无线传感网络拓扑控制问题，并针对不同的应用场景提出了三个相关的拓扑控制算法：基于完全图的分簇算法 CGCA、在 CGCA 的基础上加入移动节点担当网关以及在特定 VRMN 移动模型下的基于功率控制的拓扑控制算法。

针对重新分簇所带来的“涟漪”效应以及维护一个稳定簇结构的问题，本文提出了一种基于完全图的分簇算法 CGCA。CGCA 利用完全图中节点是等价的特性，使得重新分簇只是发生在系统启动时刻，而以后的重新分簇过程只需要在每个簇的内部节点进行轮换。由于 CGCA 把重新分簇这一全局性、周期性的行为转化为局部性和由事件触发，这使得 CGCA 的通信和计算负载可以大量减少，它在单个节点的处理复杂度和消息复杂度均为  $O(1)$ 。另外，通过优先选择距离簇头近的节点加入簇内，CGCA 不仅减少了簇头和簇内成员的簇内通信能量，而且使得簇头比较均匀的分布在部署区域。仿真试验表明：在节点密集部署的情况下，CGCA 产生的消息交换个数远小于 HEED 分簇算法。最后簇头均匀分布方面，CGCA 也明显优于 LEACH 分簇算法。

在分簇算法中，相对于普通节点来说，网关节点由于要承担簇与簇之间数据转发的任务，因此更容易耗尽自己的能量而成为瓶颈节点。为了解决此问题，本文在 CGCA 分簇算法的基础上加入少量资源和配置丰富的移动节点来担当簇与簇之间的网关，从而可以让普通节点从繁重的数据转发的任务中解放出来，达到节省能量和延长网络寿命的目的。除此之外，加入的移动网关节点可以进一步的提高簇的稳定性，这意味着重新分簇阶段不需要重新选举出新的网关节点。最后，由功能较强的移动节点代替容易出错的普通节点来担任网关可以提高系统的稳定性和容错性。

实验表明，在密集部署传感器节点的网络中，本算法产生的消息交换个数只是传统的基于最小标号分簇算法的 20% 左右。另外，在密集部署的网络环境中，本算法所取得的系统时间也是 LEACH 算法的两倍左右。但是本文算法的缺点是加入移动节点增加了额外的硬件耗费，因此在将来的研究中考虑在静态网络中加

入移动节点所带来的性能折衷问题。

移动传感网络下的拓扑控制既是一个热点又是一个难题，由于传感器节点的不断移动导致网络的底层拓扑时刻发生变化，因此拓扑控制算法通常需要周期性的执行来维持所需要的拓扑属性，这就会导致算法所产生的消息负载和产生的拓扑质量之间存在一个性能折衷。本文首先介绍了可变速率移动网络模型 VRMN，然后在此模型的基础上提出了一种类似于 XTC 算法的基于功率控制的拓扑控制算法，并分别给出了它的集中式和分布式版本，它们都具有  $O(n^3)$  的多项式时间复杂度。

本文在 VRMN 下的拓扑控制算法具有如下优点：(1) 简单：不需要复杂的硬件和软件，这与移动传感器节点只具有简单的计算和通信能力的特征相吻合 (2) 前摄：在每个单位时间间隔的最开始就已经把每个节点的所有可能移动考虑进内，因此能够保证移动传感网络在单位时间间隔内的每个瞬间都是连通的。仿真实验表明，本算法在维持网络连通性和减少节点发射半径方面表现出了较好的性能。

在一个大规模移动传感网络中，由于分簇算法能够形成一个层次结构以及能够较好的支持网络的可扩展性并在节点移动时能够较好的维护拓扑结构的稳定性，这就使得分簇算法比较适合于对这种大规模密集移动传感网络进行拓扑控制。本文提出了一种能量有效的分簇算法，它的初始簇形成阶段比较简单，具有  $O(N)$  的时间复杂度和消息复杂度，并且在分簇过程中不依赖于节点的地理位置信息。而在随后的对簇结构的维护由事件触发并且异步进行，当事件发生时节点之间只需要交换少量的信息就可以更新簇结构，因此本文算法能够取得一个较为稳定的簇结构以适应节点的移动和底层拓扑的变化。

除此之外，本文的分簇算法还具有如下的优点：(1) 本文算法的簇头节点是事先确定好的，并且不要求簇头节点在其一条邻居内具有某种特殊的属性，因此本文算法从根本上消除了在移动网络中的簇形成阶段对“frozen time”的依赖 (2) 本文采用的单元格划分方式能够使得相邻的簇头可以直接通信，当某个簇头失效时，它的簇内成员可以非常容易的划分到别的相邻簇头中，因此本文算法具有较好的容错性。

## 7.2 存在的问题和进一步的工作

尽管本文以及其他大量的学术资料对无线传感网络中的拓扑控制问题从多个角度进行了探讨，然而还有很多与拓扑控制相关的问题并没有得到仔细的研究和解决，就本文涉及的研究内容来看，无线传感网络拓扑控制存在的问题和未来

需要进行的工作可以概括如下<sup>[25,65,120]</sup>:

### 1) 更为现实的信道模型

例如，前面所述的无线电信道模型都基于一个基本的假设：那就是节点在平面上的覆盖范围是一个以节点为圆心、以发射距离为半径的圆形区域。这对应着采用的是对数距离路径信道模型，而以此建立起来的通信图的本质是一个点图。但现实中，由于无线传感器网络的部署环境比较复杂，由于建筑物、障碍物或者来自其他方面的干扰使得节点的信号传输可能是极其不规律的。因此为了使得到的结果具有实用性，一个更为现实的信道模型需要充分考虑阴影和衰减效应（shadowing or fading effect）。例如，文献<sup>[121]</sup>引入了随机顶点模型（Random Vertex Model），在此模型中，两个节点之间的链路不再是要么出现要么不出现，而是以一定的概率出现。另外一种使得信道模型更为现实的措施是把节点之间的干扰考虑进去。例如，文献<sup>[122]</sup>指出只有接收方的信噪比高于某个给定的阈值时，才认为发送方和接收方之间存在一条链路。

### 2) 更为现实的能耗模型

另外，目前的拓扑控制算法在建立能量有效的拓扑结构的过程中所考虑的能耗模型过于简单，它往往只考虑节点的发射信号的功耗，而忽略无线电接口中其他电路的能耗。例如从能量有效的角度来看，现有的拓扑控制算法认为多跳的短距离通信要优于单跳的长距离通信。但是从更为现实的能耗模型看，这种观点往往是错误的。因此，更为理想的节点能耗模型应该把节点的接收能耗以及电路的其他能耗综合考虑进去。

### 3) 更为现实的部署模型

当前关于拓扑控制的理论研究与算法设计通常假设传感网络的部署区域是一个二维的平面，然而事实上传感器节点的部署具有很强的随机性和地域受限性，因此三维立体空间更符合传感网络的实际部署环境。分析与研究三维空间中无线传感网络的拓扑控制算法以及拓扑属性，必将成为未来拓扑控制研究领域的趋势之一。

### 4) 考虑干扰的拓扑控制算法

如前所述，尽管拓扑控制的两大目标是减少能耗和减少干扰。但是，目前的大多数拓扑控制算法的研究都以减少能耗作为它们的研究目标。一个自然的问题就是，什么是干扰最优的网络拓扑？进一步来说，干扰最优的网络拓扑和能量最优的网络拓扑的相同点和不同点是什么？当所有的节点都采用统一的传输距离时，采用能使网络取得连通性的临界传输距离能够同时产生干扰最优的网络拓扑和能量最优的网络拓扑<sup>[123]</sup>。但是，当允许节点采用不同的传输功率时，问题变得非常的复杂。Burkhart 等人<sup>[124]</sup>首先引入了边覆盖的概念并对传感网络中的信

号干扰进行建模，并得出了一条结论，那就是在某些场合，减少节点能耗和减少干扰是互相冲突的，这也与<sup>[125]</sup>得出的结论相一致。然而，这些工作都只是对考虑干扰拓扑控制算法的初步工作，将来还有大量的工作需要对此方面进行研究，例如考虑多跳的通信干扰。

#### 5) 对移动性的支持

目前的大部分拓扑控制算法通常假设传感器节点静止或者弱移动，随着移动传感网络的兴起，拓扑控制算法必须能够适应节点的移动。但是，在这方面还有很多的问题没有得到深入研究。一是节点的移动对网络的综合效果还有待考察，一方面节点的移动带来了大量的消息负载；另一方面节点的移动有利于网络中的负载均衡和吞吐量提升。二是目前的网络移动模型过于理想化和理论化，因此一个更为现实可用的移动模型也是将来移动传感网络拓扑控制研究中的一项重要内容。

#### 6) 考虑 QoS 的拓扑控制

目前，QoS 在传感网络研究领域处于薄弱环节，它与具体的应用密切相关。例如，突发性的 WSN 应用通常需要生成一个延迟小和丢包率低的拓扑，而生态监护之类的传感网络应用则需要生成一个连通性强以及网络寿命长的拓扑。因此，结合 QoS 的拓扑控制算法会成为将来的一个研究方向。

综上所述，既深入研究拓扑控制相关的基础理论，又注重考虑拓扑控制算法的实用性将是我们未来的工作目标与内容。

## 参考文献

- [1] Cullar D, Estrin D, Srivastava M. Overview of sensor network[J]. Computer, 2004, 37(8): 41-49.
- [2] Pottie G and Kaiser W. Wireless integrated network sensors[J]. Communications of ACM, 2000-05, 43(5).
- [3] Kyildiz I, Su W, Sankasubramaniam Y, et al. Wireless Sensor networks:A Survey[J]. Computer Networks, 2002,(8):393-422.
- [4] Chandrakasan A, Min R, Bhardwaj M, et al. Power Aware Wireless Microsensor Systems[C]. In Proceedings of the ESSCIRC 2002, Florence, Italy, 2002-09.
- [5] Estrin D, Girod L, Pottie G, Srivastava M. Instrumenting the World with Wireless Sensor Networks[C]:ICASSP 2001:In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, UT, 2001-05.
- [6] Hill J, Culler D. MICA: A Wireless Platform for Deeply Embedded Networks[J]. IEEE Micro, 2002,22(6):12-24.
- [7] Pottie G J, Kaiser W J. Embedding the Internet: Wireless Integrated Network Sensors[J]. Communications of the ACM, 2000, 43(5): 51-58.
- [8] Rabaey J M, Ammer M J, Silva L,, et al. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking[J]. IEEE Computer, 2000,33(7): 42-48.
- [9] Steere D C, Baptista A, McNamee D, et al. Research Challenges in Environmental Observation and Forecasting Systems[C]:Mobicom 2000:In Proceedings of the 6th International Conference on Mobile Computing and Networking, Boston, MA, 2000.
- [10] Weiser M. Some Computer Science Issues in Ubiquitous Computing [J]. Communications of ACM, 1993, 36:75-85
- [11] Boriello G and Want R. Embedded Computation Meets the World Wide Web[J]. Communications of the ACM , 2000,43(5): 59-66.
- [12] Burrell J, Brooke T, Beckwith R. Vineyard Computing: Sensor Networks in Agricultural Production[J].IEEE Pervasive Computing,2004, 3(1): 38-45.
- [13] Estrin D, Govindan R, Heidemann J,et al. Next Century Challenges: Scalable Coordination in Sensor Networks[C]. In Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks, Seattle, Washington, DC, 1999.
- [14] Mainwaring A, Polastre J, Szewczyk R. Wireless Sensor Networks for Habitat Monitoring[C]. In Proceedings of the 1st ACM Workshop on Wireless Sensor Networks

- and Applications,Atlanta, GA, 2002-09.
- [15] Szewczyk R, Osterweil E, Polastre J, et al. Habitat Monitoring with Sensor Networks[J]. Communication of the ACM, 2004,47(6): 34-40.
- [16] Levis P, Culler D. Mate: A Tiny Virtual Machine for Sensor Networks[C]. In Proceedings of the 10<sup>th</sup> International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, CA, 2002-10.
- [17] Hill J, Szewczyk R, Woo A, et al. System Architecture Directions for Networked Sensors[C]. In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, Cambridge, MA, 2000:93-104.
- [18] Intel StrongARM SA-1100 Microprocessor Brief Data Sheet[R]. Intel product documentation, 2000-08.
- [19] 于海斌,曾鹏, 等. 智能无线传感器网络系统[M]. 1 版.北京: 科学出版社,2006: 6-7.
- [20] 孙利民,李建中,陈渝,等.无线传感器网络[M]. 1 版.北京: 清华大学出版社,2005:9-11.
- [21] Green.H. Technical wave 2: the sensor revolution[EB/OL]:Business week online.[2003-8-25].[http://www.businessweek.com/magazine/content/03\\_34/b3846622.html](http://www.businessweek.com/magazine/content/03_34/b3846622.html)
- [22] Werff. T. J. Ten emerging technologies that will change the world[R]. MIT's Technology review.[2003-2].[http://www.princeton.edu/~chouweb/choubio\\_papers/MIT\\_review.pdf](http://www.princeton.edu/~chouweb/choubio_papers/MIT_review.pdf).
- [23] 李善仓,张克旺. 无线传感器网络原理与应用 [M]. 1 版. 北京:机械工业出版社,2008-3.
- [24] Poduri S, Pattem S, Krishnamachari B, and et al. A unifying framework for tunable topology control in sensor networks[R]. CRES-05-004, University of Southern California, 2005:1-15.
- [25] 张学, 陆桑璐, 陈贵海,等. 无线传感器网络的拓扑控制[J]. 软件学报, 2007,18(4): 943~954.
- [26] KIROUSIS L, KRANAKIS E, KRIZANC D, and et al. Power consumption in packet radio networks[J]. Theoret. Comput. Sci. 243, 2000:289-305.
- [27] Rodoplu V, Meng T. Minimum energy mobile wireless networks[J]. IEEE Journal on Selected Areas in Communications, 1999,17:1333-1344.
- [28] LI N,HOU J,SHA L. Design and analysis of an MST-based topology control algorithm[C].In Proceedings of the IEEE Infocom. 2003:1702-1712.
- [29] Cormen T,Leiserson C, Rivest R, et al. 算法导论[M].2nd ed. 潘金贵,顾铁成,李成法,等.译.北京:机械工业出版社,2006-9:350-352.
- [30] Li L, Halpern J, Bahl P, et al. A cone-based distributed topology control algorithm for

- wireless multi-hop networks[J]. IEEE/ACM Trans. on Networking, 2005,13(1):147–159.
- [31] Santi P. Topology control in wireless ad hoc and sensor networks[J]. ACM Computing Surveys, 2005-6, 37(2):164 – 194.
- [32] Li L, Halpern J, Bahl P, et al. Analysis of Cone-Based Distributed TopologyControl Algorithm for Wireless Multi-hop Networks[C]. In Proceedings of the 20th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Newport, RI, 2001-08.
- [33] Wattenhofer R, Li L, Bahl P, et al. Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In Proceedings of IEEE Infocom, Anchorage, AK,2001-04.
- [34] BORBASH S,JENNINGS E. Distributed topology control algorithm for multihop wireless networks[C]. In Proceedings of the IEEE International Joint Conference on Neural Networks,2002:355–360.
- [35] XUE F,KUMAR P. The number of neighbors needed for connectivity of wireless networks[J]. Wireless Networks. The Netherlands: Kluwer Academic Publishers, 2004,(10) :169–181.
- [36] BLOUGH D., LEONCINI M., RESTA G., et al. The k-neighbors protocol for symmetric topology control in ad hoc networks[C]. In Proceedings of the ACM MobiHoc'03,2003: 141–152.
- [37] Kubisch M, Karl H, Wolisz A , et al. Distributed algorithms for transmission power control in wireless sensor networks[C]. IEEE WCNC 2003, New Orleans, Louisiana, 2003-3-16 –2003-3-20.
- [38] Yu J and Chong P. A survey of clustering schemes for mobile ad hoc networks[J]. IEEE Communications First Quarter 2005, 7(1):32-48.
- [39] Basagni S, Chlamtac,I-Farago A. A generalized clustering algorithm for peer-to-peer networks[C]. In Proceedings of Workshop on Algorithmic Aspects of Communication (satellite workshop of ICALP), 1997-07.
- [40] Amis A, Prakash R,Vuong T, et al. MaxMin D-Cluster Formation in Wireless Ad Hoc Networks[C]. In Proceedings of IEEE Conference on Computer Communications, 1999-03.
- [41] Wu J, Li H. On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks[C]. In Proc.3rd Int'l. Wksp. Discrete Algorithms and Methods for Mobile Computing and Communication, 1999:7–14.
- [42] Chen Y, Liestman A. Approximating Minimum Size Weakly-Connected Dominating Sets

- for Clustering Mobile Ad Hoc Networks[C].In Proc 3rd ACM Int'l. Symp. Mobile Ad Hoc Net. & Comp., 2002-06:165–172.
- [43] Chiang C, Wu H, Liu W, et al. Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel[C].In Proc IEEE SICON'97,1997-04:197–211.
- [44] Baker D, Ephremides A. A distributed algorithm for organizing mobile radio telecommunication networks[C] In: Proceedings of the 2nd International Conference on Distributed Computer Systems, 1981-04:476–483.
- [45] Baker D, Ephremides A. The architectural organization of a mobile radio network via a distributed algorithm[J]. IEEE Transactions on Communications.1981-11, 29(11):1694–1701.
- [46] Ephremides A, Wieselthier J, Baker D. A design concept for reliable mobile radio networks with frequency hopping signaling[C].In Proceedings of IEEE, 1987,75(1):56–73
- [47] Gerla M, Tsai J. Multicluster, mobile, multimedia radio network[J]. Wireless Networks, 1995,1(3): 255–265.
- [48] Lin C, Gerla M. A distributed control scheme in multi-hop packet radio networks for voice/data traffic support[C]. In Proceedings of IEEE GLOBECOM, 1995:1238–1242.
- [49] Lin C, Gerla M. A distributed architecture for multimedia in dynamic wireless networks[C].In: Proceedings of IEEE GLOBECOM, 1995:1468–1472.
- [50] Yu J, Chong P. 3hBAC (3-hop Between Adjacent Clusterheads): a Novel Non-overlapping Clustering Algorithm for Mobile Ad Hoc Networks[C]. In Proc. IEEE Pacrim'03, 2003-08, 1:318–321.
- [51] Yu J, Chang P. An efficient clustering scheme for large and dense mobile ad hoc networks(MANETs)[J]. Computer Communications ,2006, 30:5-16.
- [52] Basu P, Khan N, Little T. A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks[C]. In Proc IEEE ICDCSW' 01, 2001-04:413–418.
- [53] MacDonald A, Znati T. A Mobility-based Frame Work for Adaptive Clustering in Wireless Ad Hoc Networks[J].IEEE Journal on Selected Areas in Communications,1999-08, 17(8):1466–1487.
- [54] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks[C]. In Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii, HI, 2000-01:174-185.
- [55] Heinzelman W, Chandrakasan A, Balakrishnan H. An Application-Specific Protocol Architecture for Wireless Microsensor Networks[J]. IEEE Transactions on Wireless

- Networking, 2002,1(4): 660–670.
- [56] Atmis A, Prakash R. Load-Balancing Clusters in Wireless Ad Hoc Networks[C]. In Proc 3rd IEEE ASSET'00, 2000-03:25–32.
- [57] Chatterjee M, Das S, Turgut D. WCA:A Weighted Clustering Algorithm for Mobile Ad hoc Networks[J]. Journal of Cluster Computing, 2005, 5:193–204.
- [58] Ohta T, Inoue S, Kakuda Y. An Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks[C]. In Proc. 6th ISADS'03, 2003-04:293–300.
- [59] Younis O, Fahmy S. HEED:A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-Hoc Sensor Networks[J]. IEEE TRANSACTIONS ON MOBILE COMPUTING, 2004, 3(4):366–379.
- [60] Moussaoui O, Ksentini A, Naimi M, et al. A novel clustering algorithm for efficient energy saving in wireless sensor networks[C].In Proc of 2006 International Symposium on Computer Networks,2006-07-31:66–72.
- [61] Wattenhofer R, Zollinger A. XTC: A Practical Topology Control Algorithm for Ad-Hoc Networks[C]. In Proc of 18th International Parallel and Distributed Processing Symposium, 2004-04:216–223.
- [62] 刘林峰, 刘业. 一种无线传感器网络拓扑的启发式分簇控制算法[J].计算机研究与发展, 2008, 45(7):1099–1105.
- [63] 陈力军, 毛莺池, 陈道蓄, 等.平均度约束的无线传感器网络拓扑控制[J].计算机学报, 2007,30(9):1544–1550.
- [64] 王良民, 马建峰, 王超.无线传感器网络拓扑的容错度与入侵度[J].电子学报, 2006, 34(8):1446–1451.
- [65] 刘林峰, 金杉. 无线传感器网络的拓扑控制算法综述 [J]. 计算机科学,2008, 35(3):6–12.
- [66] Brualdi R. 组合数学[M].4 版.冯舜玺,罗平,裴伟东,译.北京:机械工业出版社,2006-05: 371–373.
- [67] 任彦, 张思东, 张宏科. 无线传感器网络中覆盖控制理论与算法.软件学报[J], 2006, 17(3):422–433.
- [68] Wang X, Xing G, Zhang Y, et al. Integrated coverage and connectivity configuration in wireless sensor networks. In Proceedings of the 1st international conference on Embedded networked sensor systems. New York: ACM Press, 2003:28–39.
- [69] Huang C, Tseng Y, Lo L. The coverage problem in three-dimensional wireless sensor networks. In: Shah R, ed. Proc. of the GLOBECOM. Dallas: IEEE Press, 2004,5:3182–3186.

- 
- [70] Pottie G, Kaiser W. Wireless integrated network sensors[J]. Communications of ACM, 2000-05, 43(5):51–58.
  - [71] Raghunathan V, Schurgers C, Sung P, et al. Energy-Aware Wireless Microsensor Networks[J]. IEEE Signal Processing Magazine, 2002-03,19(2): 40–50.
  - [72] Tang J, Xue G, Zhang W. Interference-Aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks[C]. MobiHoc'05, Illinois, USA, 2005-5:68–77.
  - [73] He Y, Zeng Y. Interference-Aware Topology Control Problem in Wireless Sensor Networks[C]. In proceedings of 6th International Conference on ITS Telecommunications, 2006:969–972.
  - [74] Gupta P, Kumar P. The capacity of wireless networks[J]. IEEE Transaction on Information Theory, 2000, 46(2):388–404.
  - [75] Lin E, Rabaey J, Wolisz A. Power-Efficient Rendez-vous Schemes for Dense Wireless Sensor Networks[C]. In Proceedings of IEEE International Conference on Communications , Paris,France, 2004-6, 7: 3769 –3776.
  - [76] Schurgers C, Tsatsis V, Ganeriwal S, et al. Optimizing Sensor Networks in the Energy-Latency-Density Design Space[J]. IEEE Transactions on Mobile Computing, 2002, 1(1): 70–80.
  - [77] Ye W, Heidemann J, Estrin D. An Energy-Efficient MAC Protocol for Wireless Sensor Networks[C]. In Proceedings of INFOCOM 2002. New York:IEEE Press, 2002-6, 3:1567–1576.
  - [78] Polastre J, Hill J, Culler D. Versatile low power media access for wireless sensor networks[C]. In Proceedings of 2nd international conference on Embedded Networked Sensor Systems, Baltimore, MD, 2004-11:95–107.
  - [79] Karl H, Willig A. 无线传感器网络协议与体系结构[M].邱天爽, 唐洪,李婷,等译.北京:电子工业出版社, 2007-1:18–19.
  - [80] Camp T, Boleng J, Davies V. A Survey of Mobility Models for Ad Hoc Network Research[J]. Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2002, 2(5):483–502.
  - [81] Johnson D, Maltz D. Dynamic source routing in ad hoc wireless networks[J]. Mobile Computing. The Netherlands :Kluwer Academic Publishers, 1996: 153–181.
  - [82] Johansson P, Larsson T, Hedman N, et al. Routing protocols for mobile ad-hoc networks - a comparative performance analysis[C]. In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, 1999: 195–206.
  - [83] Bettstetter C, Wagner C. The Spatial Node Distribution of the Random Waypoint

- Model[C]. Proc of 1st German Workshop on Mobile Ad-Hoc Networks(WMAN'02), 2002:1–18.
- [84] Broch J, Maltz D, Johnson D, et al. Multi-hop wireless ad hoc network routing protocols[C]. In Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, 1998:85–97.
- [85] Royer E, Smith P, Moser L. An analysis of the optimum node density for ad hoc mobile networks[C]. In Proceedings of the IEEE International Conference on Communications (ICC), 2001, 3:857–861.
- [86] V. Davies. Evaluating mobility models within an ad hoc network [D]. Golden, Colorado, Colorado School of Mines, 2000.
- [87] Varge A. The OMNET++ discrete event simulation system[C]. In Proceeding of the European Simulation Multiconference. Pargue, 2001:319-324.
- [88] NS2 [EB/OL]. <http://www.isi.edu/nsnam/ns/.2009>
- [89] 李建波, 黄刘生, 徐宏力, 等. 一种密集部署传感器网络的分簇算法[J]. 计算机研究与发展, 2008, 45(7): 1106–1114.
- [90] Xu Y, Heidemann J, Estrin D. Geography-Informed Energy Conservation for Ad Hoc Routing[C]. In Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy, 2001-7:70–84.
- [91] 安辉耀, 卢锡城, 彭伟, 等. MANET 中基于簇的多路径动态源路由(CMDSR)[J]. 计算机研究与发展, 2006, 43(3):381–388.
- [92] Li Q, Rus D. Sending messages to mobile users in disconnected ad-hoc wireless networks[C]. In Proceedings of 6th ACM Annual International Conference on Mobile Computing and Networking, Boston, MA, 2000-8:44–55.
- [93] Grossglauser M, Tse D. Mobility increases the capacity of ad-hoc wireless networks[J]. IEEE/ACM Transactions on Networking, 2002-8, 10(4):477–486.
- [94] Chatzigiannakis I, Nikoletseas S. An adaptive compulsory protocol for basic communications in highly changing ad-hoc mobile networks[C]. In Proceedings of the International Parallel and Distributed Processing Symposium, Fort Lauderdale, FL, 2002-4:193–202.
- [95] Wang W, Srinivasan V, Chua K. Using mobile relays to prolong the lifetime of wireless sensor networks[C]. In proceedings of Mobicom'05, 2005-8:270-283.
- [96] Gandham S, Dawande M, Prakash R, et al. Energy efficient schemes for wireless sensor networks with multiple mobile base stations[C]. In Proceedings of IEEE Globecom 2003, San Francisco, CA, 2003-12, 1:377–381.

- [97] Wang Z, Basagni S, Melachrinoudis E, et al. Exploiting sink mobility for maximizing sensor networks lifetime[C]. In Proceedings of the 38<sup>th</sup> Hawaii International Conference on System Sciences (HICSS), 2005, 9:287a-287a.
- [98] Luo J, Hubaux J. Joint mobility and routing for lifetime elongation in wireless sensor networks[C]. In Proceedings of the 24th IEEE INFOCOM, 2005-03, 3:1735–1746.
- [99] Rahimi M, Shah H, Sukhatme G, et al. Energy harvesting in mobile sensor networks[C]. In Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, September 2003.
- [100] Howard A, Mataric M, Sukhatme G. Self-deployment algorithm for mobile sensor networks[J]. Autonomous Robots Special Issue on Intelligent Embedded Systems, 2002, 13(2): 113–126.
- [101] Shah R, Roy S, Jain S, et al. Data MULEs: Modeling a three-tier architecture for sparse sensor networks[C]. In Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, SNPA 2003, Anchorage, AK, 2003-5-11: 30–41.
- [102] Li J, Huang L, Wang G. A Novel Clustering Algorithm by Using Mobile Gateways in Densely Deployed Sensor Networks[C]. In proceedings of International Conference on Information and Automation, Zhangjiajie, China, 2008-06:1553–1559.
- [103] Dantu K, Rahimi M, Shah H, et al. Robomote: Enabling Mobility In Sensor Networks[C]. In Proceedings of IEEE IPSN'05, 2005-04:404–409.
- [104] Ling Qing and Tian Zhi. Impact of mobility on topology control of wireless sensor networks[C]. In proceedings of Wireless Communications, Networking and Mobile Computing, 2007:2483 – 2486.
- [105] Wu J,Dai F. Mobility control and its applications in mobile ad hoc networks[J]. IEEE Network ,2004-07,18(4): 30–35.
- [106] Wu J, Dai F. Mobility-Sensitive Topology Control in Mobile Ad Hoc Networks[J]. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2006-06, 17(6):522–535.
- [107] Zhao L. Topology control for mobile ad hoc networks[D]. Newark, Delaware :University of Delaware, 2007:9–10.
- [108] Bettstetter C, Resta G, Santi P : The node distribution of the random waypoint mobility model for wirelss ad hoc networks[J]. IEEE Transactions on Mobile Computing,2003, 2:257–269.
- [109] Blough D, Resta G, Santi P. A statical analysis of the long-run node spatial distribution in mobile ad hoc networks [J]. Wireless Networks, 2004, 10:543–554.

- [110] Navidi W, Camp T. Stationary distributions for the random waypoint mobility model[J]. IEEE Transactions on Mobile Computing, 2004, 3:99–108.
- [111] Santi P. The critical transmitting range for connectivity in mobile ad hoc networks [J]. IEEE Transactions on Mobile Computing, 2005, 4:310–317
- [112] Zhao L, Lloyd E. Distributed Topology Control for Stationary and Mobile Ad Hoc Networks[C]. In Proc. of 3rd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'06), 2006-10:521-524.
- [113] Wu J, Dai F. Mobility sensitive topology control in mobile ad hoc networks[J], IEEE Transactions on Parallel and Distributed Systems, 2006-06, 17(6): 522-535.
- [114] Siripongwutikorn P, Thipakorn B. Mobility-aware topology control in mobile ad hoc networks [J]. Computer Communications, 2008, 31:3521-3532.
- [115] Zhao L, Lloyd E, Ravi S. Topology Control for Constant Rate Mobile Networks[C]. In Proceeding of IEEE GLOBECOM, 2006-11:1-6.
- [116] Ramanathan R, Rosales-Hain R. Topology control of multihop wireless networks using transmit power adjustment[C]. In Proceedings of IEEE INFOCOM 2000, 2000-03:404–413.
- [117] Okabe, Atsuyuki ,Miller, et al. Exact Computational Methods for Calculating Distances Between Objects in a Cartographic Database[J]. Cartography and Geographic Information Science, 1996-10, 23(4):180-195.
- [118] Robert G, Gallager. Distributed minimum hop algorithms[R]. Technical Report LIDS-P-1175, Laborator for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, 1982-01.
- [119] Xu.K, Hong.X and Gerla M. An ad hoc network with mobile backbones[C]. In proceedings of ICC'02, 2002, 5:3138-3143.
- [120] Santi P. Topology control in wireless ad hoc and sensor networks [M]. 1st ed. United States:John Wiley&Sons, 2005.
- [121] Farag'o A. Scalable analysis and design of ad hoc networks via random graph theory[C]. In Proceeding of ACM DIAL-M 02, Atlanta, GA, 2002:43–50.
- [122] Dousse O, Baccelli F, Thiran P. Impact of interferences on connectivity in ad hoc networks[C]. In Proceeding of IEEE Infocom 2003, San Francisco, CA, 2003: 1724–1733.
- [123] Narayanaswamy S, Kawadia V, Sreenivas R, et al. Power control in ad hoc networks: theory, architecture, algorithm and implementation of the compow protocol[C]. In Proceeding of European Wireless 2002, Florence,2002: 156–162.

- [124] Burkhart M, Rickenbach V, Wattenhofer R, et al. Does topology control reduce interference?[C]. In Proceeding of ACM MobiHoc '04, Lausanne, 2004: 9–19.
- [125] MeyerAufDerHeide F, Schindelhauer C, Grunewald M. Congestion, dilation, and energy in radio networks[C]. In Proceeding of ACM Symposium on Parallel Algorithms and Architectures (SPAA), Winnipeg, Manitoba, Canada, 2002:10–13.

## 致 谢

伴随着快乐、痛苦和兴奋的博士三年时光里，在我尊敬的导师黄刘生教授的悉心指导下，我完成了相关科研活动、公开发表文章和博士学位论文等充满艰辛的任务，在所有的工作中，都浓缩和凝聚着导师的心血和汗水！在博士生学习期间，黄老师渊博的知识，严谨求实的治学风格，勤奋的工作态度，扎实的科研精神以及对新技术的敏感性和思维的开放性，使我树立了宏伟的学术目标，掌握了通用的研究方法，为我以后在高校从事科研活动打下了坚实的基础。导师朴实无华、平易近人的人格魅力，乐观向上、积极主动的生活态度，也使我明白了许多待人接物与为人处世的道理，更时刻提醒我应该如何做人，这一切的教诲和熏陶必将使我终身受益。在此，向黄老师表示崇高的敬意和衷心的感谢。同时，也感谢师母张春花老师在学习和生活上给予我的关心。

衷心感谢肖明君老师、徐宏力和汪炀师兄的教导、鼓励与帮助。他们对待学术的一丝不苟的态度，努力拼搏、健康向上的人生态度，严谨的治学态度和平易近人的生活作风以及爱岗敬业的精神必将对我今后的学习、工作与生活产生深远的影响。在此，谨向他们表示我内心由衷的感谢和敬意。

感谢本人所在课题组的一群风华正茂的有志青年，尤其感谢徐犇博士、王继春博士、王刚博士、王培博士等在学术上所给予的莫大帮助，也要感谢实验室和我朝夕相处的师弟们，你们带给我的快乐和帮助将会是我一生非常美好的回忆。

感谢父母的养育之恩，你们的鼓励和教导是我一生前进的最大动力。在将来的日子里，我将继续努力，以更大的成绩回报父母，以及所有关心过我成长的亲人们。在这里尤其要感谢我的岳父和岳母，他们通情达理、宽容无私，在我读博期间给予我莫大的支持，你们对我的帮助我会终生铭记。在这里祝愿几位老人永远健康快乐！

最后，需要特别感谢我的温柔善良、善解人意的妻子姜山。在我读博期间，她一人独自撑起家庭的重担而毫无怨言，在我背后一直默默的支持着我。执子之手，与子偕老，是我论文搁笔之际的最大心愿。

李建波  
2009.5

## 在读期间完成的学术论文

[1] 李建波, 黄刘生, 徐宏力, 等. 一种密集部署传感器网络的分簇算法[J]. 计算机研究与发展, 2008, 45(7):1106-1114.

**EI, Accession number: 084111636138**

[2] Li Jianbo, Huang Liusheng, Wang Gang. A Novel Clustering Algorithm by Using Mobile Gateways in Densely Deployed Sensor Networks[C]. In proceedings of International Conference on Information and Automation, Zhangjiajie, China, 2008-06:1553-1559.

**EI, Accession number: 084411662982**

[3] Li Jianbo, Huang Liusheng, Xiao Mingjun. Energy Efficient Topology Control Algorithms for Variant Rate Mobile Sensor Networks[C]. In proceeding of the 4th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN'08), Wuhan, China, 2008-12:23-30.

**EI, Accession number: 20091111954054**

[4] Li Jianbo, Huang Liusheng, Xiao Mingjun, et al. A Hybrid Range-Free Localization Scheme in Wireless Sensor Networks[C]. In Proceedings of the 2nd international conference on Scalable information systems, Suzhou, China, 2007-07, Vol. 304.

[5] Wang Jichun, Huang Liusheng, Li Xiang, Huang He, Li Jianbo. A Collaborative Localization Scheme from Connectivity in Wireless Sensor Networks[C]. In 6th International Conference on Wired/Wireless Internet Communication, Tampere, Finland, 2008-05: 213-223.

**EI, Accession number: 20082411308178**

[6] Wang Gang, Huang Liusheng, Xu Hongli, Li Jianbo. Relay node placement for maximizing network lifetime in wireless sensor networks[C]. In proceeding of Wicom '08, Dalian, 2008-10: 1-5.

**EI, Accession number: 20090111833803**

## 在读期间参加的科研项目

- 1 2006.05 ~ 2011.05: 无线传感网络应用示范系统的研究, 973 一级子课题, 260 万;
- 2 2006.10 ~ 2008.07: 具有高可靠信息收集能力的 WSN 体系结构研究, 863 项目, 25 万;
- 3 2007.03 ~ 2009.08: 无线传感网络节点定位和目标跟踪技术研究, 中国科学院知识创新工程重要方向项目, 60 万。
- 4 2007.06 ~ 2008.7: 具有高可靠信息收集能力的 WSN 体系结构研究, 解放军某部, 50 万