

密级: \_\_\_\_\_



中国科学院大学  
University of Chinese Academy of Sciences

# 博士学位论文

面向数据聚合的无线传感器网络传输调度方法研究

作者姓名: \_\_\_\_\_ 肖世良

指导教师: \_\_\_\_\_ 袁晓兵 研究员

中国科学院上海微系统与信息技术研究所

学位类别: \_\_\_\_\_ 工学博士

学科专业: \_\_\_\_\_ 通信与信息系统

培养单位: \_\_\_\_\_ 中国科学院上海微系统与信息技术研究所

二零一五年五月



**Research on TDMA-based Transmission Scheduling Methods**  
**for Data Aggregation in Wireless Sensor Networks**

**By**

**Xiao Shiliang**

**A Dissertation Submitted to  
University of Chinese Academy of Sciences  
In partial fulfillment of the requirement  
For the degree of  
Doctor of Engineering**

**Shanghai Institute of Microsystem and Information Technology,  
Chinese Academy of Sciences**

**May, 2015**



## 致谢

“暮春三月，江南草长，杂花生树，群莺乱飞”。在这万物生长的美好时节，我就要告别长达五年的硕博连读生涯，走向第一份正式的工作岗位了。在科大的一年和在微系统所的四年，是我人生中重要和难忘的一段历程。读博是一种磨练，是一种洗礼，是一种超越。在切实地努力奋斗过后，收获的不是几篇论文，一个学位，而是宝贵的精神财富。当然，我明白，自己能够顺利度过研究生阶段并有所收获，除了个人的努力之外，与许许多多人的指导和帮助是分不开的。这里，我有很多感谢的话要说。

感谢我的博士生导师袁晓兵研究员。在博士选题上，袁博给予了我们很大的自由度，提供了我们个性发展的空间。凭借着深厚的专业功底和宽阔的学术视野，袁博为本博士课题的架构和内容提供了很多有价值的意见和建议，让我受益匪浅。感谢参与指导我博士论文的李宝清研究员。李博具有敏锐的洞察力和丰富的工程经验，多次对本论文进行深入细致的点拨和指导，为论文的日渐完善付出了很多辛劳。虽然袁博和李博工作繁忙，但他们在有限时间里的言传与身教依然让我获益良多。袁博和李博勤勉务实的工作作风，正直厚道的为人处事方式，永远是我学习的榜样。

感谢实验室支部书记练敏英和行政部丁吉芸等人在这些年为我们四室学生所做的大量细致、繁琐的工作。你们的无私付出，使得我们有了舒适的学习环境，并能心无旁骛地进行学术研究。感谢人教处罗琦老师、邵天浩老师、余翔波老师和解佳老师。你们任劳任怨、艰苦耐心地为我们学生做了大量工作，让我们能够顺利、快乐地在微系统所学习和生活。

感谢实验室的同事刘建坡、钱汉旺、裴俊、吕建飞、钱文军、李强、柳军、王文彬、沙晓玲、欧阳泽，以及曾经的同事邱云周、宋恩亮、张婷、周磊磊、杨立朝等人的帮助，从你们身上所学到的工作能力和素养让我很受用。

感谢实验室已经毕业的覃荣华、孙启、谢东峰、周乾伟等师兄们的指导和帮助。感谢实验室在读的程小六、程勇博、吴光、陈昕麟等师弟师妹们的帮助。感谢同届的黄景昌、潘乐炳和张鑫，你们有太多的优点值得我学习，与你们一起奋斗的岁月令人难以忘怀。感谢四室这些年所有毕业和在读的同学们。在这个团结和进步的集体里，我们每一个人得以幸福、快乐地成长。

感谢我的父母和家人。父母的养育之恩，是我一生一世都无以为报的。在二十多年的漫长求学生涯中，我所取得的每一点进步都离不开父母的支持和帮助。感谢我的妻子史丹晖。感谢你的深情陪伴和鼎力支持。与你在一起的这些年是无比幸福的。

最后，感谢论文评审和论文答辩委员会的各位专家。感谢你们付出的辛劳和提出的宝贵意见与建议。



# 面向数据聚合的无线传感器网络传输调度方法研究

肖世良(通信与信息系统)

指导老师: 袁晓兵 研究员

## 摘要

无线传感器网络是由大量具备感知、存储、通信和计算功能的传感器节点采用多跳无线通信的方式自组织而成的网络。由于使用灵活、布设方便并且成本低廉, 无线传感器网络已经在军事侦察、环境监测、智能家居、工业控制等领域得到广泛应用。在纷繁复杂的应用场景中, 无线传感器网络有一项最基本的任务, 即从物理环境中收集感知数据。由于节点数量很大, 并且数据包一般需要经过多跳转发才能达到目的节点, 因此, 如果让基站直接收集所有源节点产生的原始数据包, 将严重加剧节点能量消耗, 从而缩短网络生命周期。

在实际应用中, 传感器节点通常密集分布, 并且为某种共同的应用提供数据感知服务。此时, 网络中各个节点产生的数据一般满足某种空间或时间上的相关性。在数据收集过程中, 我们可以充分挖掘和利用这种相关性, 让网络中的每一个中间节点对其收到的待转发数据包和本地生成的数据包进行融合, 然后将融合后的数据包发送给下一跳节点, 从而显著减轻节点的传输负载。无线传感器网络中的数据聚合, 指的正是这样一个“边收集边融合”的过程。一般来说, 设计一个高效的数据聚合方案需要考虑融合函数选择、数据聚合路由和数据聚合调度等重要功能模块。其中, 数据聚合调度控制节点进行数据融合和接入无线信道的时机, 是数据聚合的一项关键技术。

与传统的无线网络调度问题不同, 数据聚合调度本质上是一个广义传输调度问题。它不仅决定数据的发送时刻, 而且决定数据的传输路径和数据发送时采用的传输功率/速率等。设计高效的数据聚合调度方法, 对提高网络能量效率、降低网络传输时延等具有重要意义。然而, 数据聚合调度研究面临着若干突出的技术挑战, 包括克服无线干扰/丢包的不利影响、优化空间的复用度、权衡多种性能指标、便于分布式实现等。本博士论文主要针对干扰网络和丢包网络中的数据聚合调度问题展开深入研究, 目的是设计可行和高效的调度算法与协议, 以提升网络的时延、数据质量等性能。本

文的具体研究内容以及取得的创新点如下：

协议干扰模型是一种广泛使用的抽象无线干扰模型，具有简洁和易于建模的优势。针对协议干扰模型下的无线传感器网络最小时延数据聚合调度问题，本文提出了一种两阶段的数据聚合调度方案。在第一阶段，基于改进的连通支撑集，构造一颗数据聚合树作为数据聚合的路由；在第二阶段，基于分层贪心极大策略和网络分区与着色策略，分别提出了一种集中式和一种分布式的数据聚合传输调度算法。从理论上证明了所提出的调度算法的可行性，并且推导了数据聚合时延的上界。分析结果表明两种调度算法均是具有常量近似比的次优算法。基于 MATLAB 的仿真实验进一步验证了调度算法的性能。实验结果显示，相比文献中的最新相关算法，本文算法的平均网络时延在多种网络仿真场景下均有一定程度的改善。

由于协议模型无法刻画无线干扰的衰减和累加特性，研究者逐渐将视角转向物理干扰模型，即信号干扰噪声比(SINR)模型。虽然物理干扰模型能够更为真实地反映无线干扰对信号接收的影响，但是其非局部性和非二元性为分布式调度算法和协议的设计带来了严峻挑战。针对物理干扰模型下的无线传感器网络最小时延数据聚合调度问题，本文提出了一种完全分布式的数据聚合解决方案。该方案以迭代网络分区与着色技术为基础，由分布式数据聚合树构造子算法、分布式链路调度子算法和分布式功率控制子算法联合而成。通过合理配置调度和功率控制模块的参数，证明了该方案在物理干扰模型下的可行性。推导了该方案在最坏情况下的聚合时延。结果表明，最坏时延与网络规模无关，从而显示该方案具有良好的扩展性。通过大量的仿真实验评价了该方案在不同场景下的平均时延、能耗等性能。

传统的调度策略，不管是基于协议干扰模型还是基于物理干扰模型，其基本原理都是让潜在的相互冲突链路尽可能地在不同的时隙调度，从而避免或降低无线干扰对目的信号接收的危害。然而，类似被动和保守的干扰处理策略将导致较低的时隙利用率，从而使调度算法的时延性能下降。为了提高时间效率，本文基于主动的干扰消除理念，将物理层多包接收模型引入无线传感器网络数据聚合调度中，并提出了两种时延高效的启发式调度算法，分别为随机链路优先聚合调度算法和最短链路优先聚合调度算法。证明了两种调度算法的可行性，分析了两种调度算法的计算时间复杂度。实验结果表明，相比传统基于单包接收模型的调度方法，本文算法能够显著地降低数据

聚合平均时延和提高网络吞吐量。

当网络中出现由于传输错误而导致的数据包丢失时，传统的数据聚合协议的性能将急剧降低。这体现在汇聚节点此时收到的聚合数据包中的有效信息量将大大减少。为了克服网络丢包对数据聚合性能的损害，迫切需要设计一种传输错误容忍的数据聚合协议。针对低占空比且链路不可靠的无线传感器网络，本文提出了一种错误容忍数据聚合协议。该协议通过采用链路可靠性增强手段，来提高点到点的传输成功概率，最终改善网络链路损耗状况下的数据聚合信息质量。根据采用的链路可靠性增强技术的不同，我们为该协议设计了三个变种，分别为基于动态睡眠调度调整策略的协议，基于机会路由策略的协议和基于混合策略的协议。上述三种错误容忍数据聚合协议具有各自不同的特点。仿真实验表明，在链路不可靠的网络场景下，相比传统的未采用链路可靠性增强手段的数据聚合协议，上述三种协议均能够不同程度地提升汇聚节点所接收聚合包的信息质量。其中基于混合策略的协议较另外两种协议对数据质量性能的提升更为显著。

本文研究了不同干扰和丢包模型下的无线传感器网络数据聚合问题。由于传感器网络种类繁多，节点类型多样化，并且应用场景日趋复杂，下一步需要针对更加实际和精确的网络模型设计数据聚合调度算法。另外，由于多数情形下的调度问题都是极其复杂和难处理的，迫切需要设计理论性能和实际性能均更加良好的调度算法，这也是下一步的研究方向之一。

**关键字：** 无线传感器网络；数据聚合；调度；时分多址；无线干扰



# Research on TDMA-based Scheduling Methods for In-network Data Aggregation in Wireless Sensor Networks

Xiao Shiliang (communications and information system)

Directed by: Professor Yuan Xiaobing

## Abstract

Wireless sensor networks (WSNs) are composed of a large number of sensor nodes that are able to sense the surroundings, process the sensed data, and communicate the sensed data to the user center via multi-hop wireless communications. Due to their flexibility in use, simplicity in deployment and low in cost, WSNs succeed in a wide spectrum of applications, including military surveillance, environment monitoring, smart home, industrial control, etc. Among these numerous applications, one common, and perhaps the most important task of WSNs is to gather data from the physical world. Note that the number of sensor nodes is large, and it usually takes a few hops for each packet to reach the sink. Thus, gathering all the raw data packets from the network directly would be a rather energy-consuming process, especially for wireless sensor networks whose batteries are not only limited but also difficult to replenish.

Luckily, the sensor nodes in WSNs are usually deployed within a crowded area, and they cooperate with each other to accomplish a common application task. That means the data generated within the network are likely to be correlated in spatial or temporal dimensions. We can exploit such kind of correlation to enhance the energy efficiency of data gathering in WSNs. Specifically, the intermediate nodes are allowed to aggregate the received packets with its own packet by using some correlation-dependent fusion functions, and then transmit the aggregated packets to next-hop nodes. The aforementioned process is referred to as in-network data aggregation in the literature. In data aggregation, we need to appropriately schedule the transmission requests in the network to fulfill the aggregation task in a timely and energy-efficient manner. Such a research topic is referred to as data

aggregation scheduling, and has gained much attention in recent years.

In this thesis, we are concentrated on the data aggregation scheduling problem in WSNs. Unlike traditional scheduling problems in wireless networks, data aggregation scheduling determines not only the active slot of each node/link, but also the transmission path of each packet and the used transmission power/rate. In general, this thesis is conducted from the following aspects.

Protocol interference model is a widely used abstract model that captures the effect of physical-layer interference on the transmission reliability of MAC-layer packets, due to its simplicity and ease of description. We first study the data aggregation scheduling problem in WSNs under the protocol interference model, and propose a two-phase algorithmic framework for solving it. More specifically, in the first phase, a data aggregation tree construction algorithm based on connected dominating set is presented. In the second phase, both a centralized greedy timeslot allocation algorithm and a distributed network-partition-and-coloring based timeslot allocation algorithm are presented. We theoretically show that the proposed scheduling algorithms are feasible, and moreover, can achieve constant approximation ratios on the optimal latency. MATLAB-based simulations are conducted to evaluate the practical performance of the proposed algorithms and the results show that compared to the state-of-the-art algorithms, our algorithms can indeed improve the network latency to some extent under various settings.

Recently, the more accurate and practical physical interference model, also known as the signal-to-interference-plus-noise (SINR) model, has attracted more and more attention from the research community. The complexity of the SINR model brings extra challenges to the design of data aggregation scheduling algorithms. Previous work on this topic usually gives centralized solutions, which, however, cannot be applied to dynamic wireless sensor networks. Aware of this inefficiency, we propose a distributed data aggregation framework, which jointly implements data aggregation tree construction, link scheduling and physical-layer power control in a distributed manner. By carefully choosing the system parameters of the proposed framework, we show that the induced scheduling algorithm is

feasible under the physical interference model. Meanwhile, we rigorously derive the upper bound of the aggregation latency of the proposed framework, and conduct extensive simulations to evaluate the practical performance of the proposed algorithms under various network settings.

Traditional scheduling strategies, either based on the protocol interference model or on the physical interference model, are essentially interference-passive solutions that treat interference as something harmful and try to avoid or tolerate it while making scheduling plans. The outcome is that such scheduling strategies are inefficient in time latency. To improve the time efficiency of data aggregation scheduling, we introduce the idea of interference cancellation based scheduling, advocate the using of multi-packet reception model inspired by advanced physical layer techniques, and propose two polynomial-time scheduling algorithms that can fully utilize the interference cancellation capability to increase the number of scheduled links in each time slot. We theoretically show that both algorithms are feasible. Our simulations indicate that the proposed algorithms substantially reduce the data gathering latency in WSNs with comparison to traditional interference-passive scheduling solutions.

Finally, we investigate the data aggregation problem in a practical WSN where links are unreliable and duty-cycle technique is coupled with the MAC layer. When there exists packet loss in the network, traditional data aggregation schemes will suffer from significant performance degradation because the effective information among the aggregation packet received by the sink will diminish remarkably. To overcome the damage of packet loss on network performance, especially in low-duty-cycle wireless sensor networks, we propose a fault-tolerant data aggregation protocol (FODAP) which leverages several link reliability enhancement techniques to improve the quality-of-information (QoI) of data aggregation. These techniques include dynamic sleep schedule adjustment (DSSA), opportunistic routing (OR), and the combination of DSSA and OR. Accordingly, we present three variations of FODAP, namely FODAP-D, FODAP-R and FODAP-H. Through extensive simulations, we show that the above three

protocols can all improve the QoI of data aggregation in wireless sensor networks, and in most of the scenarios, FODAP-H outperforms the other two schemes.

In summary, we have studied the data aggregation scheduling problem in WSNs under various network models and application sceneries, and proposed a set of competing scheduling algorithms for solving it. Due to the diversity of sensor nodes, network topologies, and application requirements, the data aggregation scheduling problem under more involved models are still challenging problems that need further research. Besides, designing more efficient scheduling algorithms with provably performance bounds is also an interesting research direction.

**Keywords:** wireless sensor networks; data aggregation; scheduling; TDMA; wireless interference.

# 目录

致谢 .....	I
摘要 .....	III
Abstract.....	VII
目录 .....	XI
第1章 绪论 .....	1
1.1    课题背景及意义 .....	1
1.2    无线传感器网络概述 .....	2
1.2.1 无线传感器网络体系结构 .....	2
1.2.2 无线传感器网络通信协议栈 .....	4
1.2.3 无线传感器网络应用 .....	6
1.3    无线传感器网络数据聚合调度概述 .....	7
1.3.1 无线传感器网络数据聚合基本概念 .....	7
1.3.2 无线传感器网络数据聚合设计目标与关键技术 .....	8
1.3.3 无线传感器网络数据聚合调度基本概念与特点 .....	11
1.3.4 无线传感器网络数据聚合调度研究进展与研究挑战 .....	12
1.4    本文的研究内容与结构安排 .....	14
1.4.1 本文的研究内容与主要创新点 .....	14
1.4.2 本文的结构安排 .....	15
1.5    本章小结 .....	18
第2章 协议干扰模型下的无线传感器网络数据聚合调度 .....	19
2.1    引言 .....	19
2.2    相关工作 .....	20
2.3    网络模型与问题描述 .....	21
2.3.1 网络模型 .....	21
2.3.2 协议干扰模型 .....	21
2.3.3 问题描述 .....	22
2.4    协议干扰模型下的数据聚合调度算法 .....	22
2.4.1 基于连通支撑集的数据聚合树构造算法 .....	23
2.4.2 基于贪心极大策略的集中式调度算法 .....	25
2.4.3 基于网络分区与着色策略的分布式调度算法 .....	26
2.4.4 算法性能分析 .....	27

---

2.5	仿真实验 .....	30
2.5.1	实验场景与参数设置 .....	30
2.5.2	实验结果与分析 .....	31
2.6	本章小结 .....	34
第3章	物理干扰模型下的无线传感器网络分布式数据聚合调度 .....	35
3.1	引言 .....	35
3.2	相关工作 .....	36
3.3	网络模型与问题描述 .....	37
3.3.1	网络模型 .....	37
3.3.2	物理干扰模型 .....	37
3.3.3	问题描述 .....	38
3.4	物理干扰模型下的分布式数据聚合算法 .....	39
3.4.1	算法概述 .....	39
3.4.2	分布式数据聚合树构造子算法 .....	39
3.4.3	分布式链路调度子算法 .....	41
3.4.4	功率分配子算法 .....	42
3.4.5	算法性能分析 .....	43
3.5	算法性能评价 .....	46
3.6	本章小结 .....	50
第4章	基于多包接收技术的树状无线传感器网络数据聚合调度 .....	51
4.1	引言 .....	51
4.2	相关工作 .....	52
4.3	网络模型与问题描述 .....	53
4.3.1	网络模型 .....	53
4.3.2	物理层多包接收模型 .....	53
4.3.3	问题描述 .....	55
4.4	多包接收模型下的数据聚合调度算法 .....	55
4.4.1	链路相对干扰 .....	55
4.4.2	基于随机链路优先的贪心聚合调度算法 .....	56
4.4.3	基于最短链路优先的贪心聚合调度算法 .....	57
4.4.4	算法分析 .....	58
4.5	算法性能评价 .....	60
4.5.1	实验场景设置 .....	60
4.5.2	网络规模对算法性能的影响 .....	61

---

4.5.3 SINR 参数对算法性能的影响 .....	63
4.5.4 簇的数量对算法性能的影响 .....	65
4.5.5 与文献 <sup>[104]</sup> 进行对比 .....	66
4.6 本章小结 .....	68
<b>第 5 章 FODAP：面向低占空比无线传感器网络的错误容忍数据聚合协议 .....</b>	<b>69</b>
5.1 引言 .....	69
5.2 相关工作 .....	70
5.3 系统模型 .....	70
5.3.1 低占空比模型 .....	70
5.3.2 网络模型 .....	71
5.3.3 链路可靠性与数据聚合质量模型 .....	72
5.3.4 设计目标 .....	73
5.4 设计方案 .....	73
5.4.1 FODAP 总体框架 .....	73
5.4.2 FODAP 设计细节 .....	74
5.4.3 FODAP 实现方式 .....	77
5.5 方案性能评价 .....	78
5.5.1 实验设置与评价标准 .....	78
5.5.2 实验结果与分析 .....	78
5.6 本章小结 .....	82
<b>第 6 章 总结与展望 .....</b>	<b>83</b>
6.1 本文总结 .....	83
6.2 下一步研究计划 .....	84
参考文献 .....	87
攻读学位期间发表的学术论文与研究成果 .....	99
作者简介 .....	101
附录 .....	102



## 第1章 绪论

### 1.1 课题背景及意义

20世纪90年代以来，微机电系统(MEMS, Micro-Electro-Mechanical System)技术的快速发展促进了传感器的微型化，降低了传感器的成本，使得大批量地生产和利用传感器成为可能。同期，随着低功耗无线通信技术的进步，无线自组织网络应运而生。研究人员发现，通过将微传感器技术与无线自组织网络技术有机结合起来，会产生一种全新的网络形态，即无线传感器网络。所谓无线传感器网络(Wireless Sensor Network, WSN)，就是由大量具备了信息感知、信息存储、信息处理和信息传输能力的传感器节点通过多跳无线通信的方式自组织而成的网络<sup>[1]</sup>。与普通传感器不同，WSN中的传感器节点不仅能够进行数据采集，而且具备了无线数据通信的能力，同时还拥有一定的数据存储和处理功能。大量的传感器节点自主和智能地通过无线通信方式进行协作，在很多重要应用领域均展现出强大生命力。例如，在军事侦察领域，WSN能够被布设在偏远边境或危险战场，全天候地对可疑入侵目标进行监控，并及时地将探测信息上报给指控中心<sup>[2]</sup>。在环境保护领域，WSN能够被布设在湖泊、森林等人迹罕至的地方，对水质、空气污染物等进行实时、长期的监测<sup>[3]</sup>。在工业控制领域，WSN能够用来准确地收集工业生产过程中的产品状态信息，并实时地传输给控制中心，以辅助控制中心做出更加合理的决策与调度<sup>[4]</sup>。近十年来，随着学术研究的深入和工程化的进展，WSN逐渐从实验室走进人类的生产生活中。由WSN衍生出来的物联网方兴未艾，大规模的WSN应用系统也逐渐在全世界各地发展起来，如浙江省天目山森林碳汇监测系统<sup>[5]</sup>，美国大鸭岛生态环境监控系统<sup>[6]</sup>等。

在纷繁复杂的应用场景中，WSN有一项共同的、也是最基本的任务，那就是从物理环境之中收集感知数据。不同的应用领域，仅仅是数据类型和含义的差异，而数据收集这项操作是必须执行的。数据收集(data collection)是WSN研究中的一个重要课题。严格来讲，所谓数据收集，指的是网络中的传感器节点采集环境信息，并将生成的含有采集信息的数据包通过多跳转发方式发送给基站，最后数据包全部汇聚到基站的过程<sup>[7]</sup>。由于传感器节点数量较多，且距离基站较远的传感器节点产生的数据包因无法直达而需要经过多跳转发才能达到基站，因此在数据收集过程中，网络中的节点(尤其是中间节点)承载着大量的传输负荷。众所周知，无线传感器网络是能量受限的。低功耗传感器节点不仅电池容量有限，而且电池能量难以获得有效补充。另外，在节点的感知、存储、计算和通信等功能模块中，通信模块的能量消耗是最突出的<sup>[11]</sup>。因此，数据收集是一个严重消耗能量的过程。降低数据收集的能量消耗，对于提高

WSN 能量有效性、延长 WSN 生命周期至关重要。

在实际应用中，由于传感器节点分布通常较为密集，并且节点一般都是针对某种相同的物理状态进行感知，因此网络中不同节点产生的数据往往满足某种空间或时间上的相关性<sup>[8]</sup>。通过挖掘和利用这种空时相关性，在数据收集的过程中，可以让每个中间节点对其接收到的待转发数据包与自己产生的数据包进行压缩和融合，然后将融合后的数据包发送给下一跳节点。上述过程被称为无线传感器网络数据聚合(data aggregation)<sup>[9, 10]</sup>。与传统的数据聚合概念不同，无线传感器网络中的数据聚合并不是指节点级的数据融合操作，而是一个“边收集边融合”的过程。由于能够大大减轻网络传输负载和显著提升数据收集能量效率，无线传感器网络数据聚合吸引着越来越多的研究<sup>[9-11]</sup>。一般来说，为了设计一个高效的数据聚合方案，我们需要考虑融合函数选择、数据聚合路由和数据聚合调度等重要功能模块。其中，数据聚合调度(data aggregation scheduling)控制节点进行数据融合和接入无线信道的时机，是数据聚合的一项关键技术。

与传统的无线网络调度问题不同，数据聚合调度本质上是一个广义调度问题。它不仅决定数据包的发送时刻，而且决定数据包的传输路径，甚至包括数据包的传输功率/发送速率等<sup>[12]</sup>。设计高效的数据聚合调度方法，对于进一步提升网络性能，改善网络服务质量具有重要意义。然而，WSN 中的数据聚合调度方法设计面临诸多技术挑战，包括克服无线干扰/丢包的不利影响、优化空间的复用度、权衡多种性能指标、便于分布式实现等。

本博士论文主要针对干扰网络和丢包网络中的数据聚合调度问题展开深入研究，目的是设计可行和高效的调度算法与协议，以提升网络的时延、数据质量等性能，从而改善网络的用户体验。

## 1.2 无线传感器网络概述

本节讲述无线传感器网络(WSN)的基本概念，包括 WSN 体系结构、WSN 通信网络协议栈和 WSN 应用等。

### 1.2.1 无线传感器网络体系结构

由于应用场景的多样性，WSN 的体系结构并没有统一的标准。一种较为常见的 WSN 体系结构如图 1.1 所示。一般来说，数量较多的传感器节点通过固定或随机地方式布设在监测区域中。传感器节点既可以单独也可以通过与邻居节点协作来对某个环境物理状态进行感知。由于具备无线通信能力，传感器节点能通过自组织网络技术构成一个连通的网络架构，所有节点均可以通过构建好的网络基础设施进行信息交互。在监测区域外，有一个或多个汇聚节点(sink)。汇聚节点又称为基站，它与普通传感

器节点不同，其并不直接进行数据采集，而是充当所有传感器节点的上级管理节点。具体来说，汇聚节点负责以向全网广播控制命令的方式对网络进行管理，同时从网络中收集部分或者全部源节点所感知到的环境数据。汇聚节点既可以与网络用户通过物理线缆直接连接，也可以通过互联网、卫星通信等方式进行间接地远程连接。网络用户或控制中心一般是由一台功能完备的PC机和其上搭载运行的界面友好的应用程序组成，为网络的实际使用者所操作<sup>[1]</sup>。网络使用者通过桌面应用程序能够实时地监测传感器节点采集到的环境数据，也可以通过发送命令显式地控制网络的运行状态，如启动、休眠、关闭等。

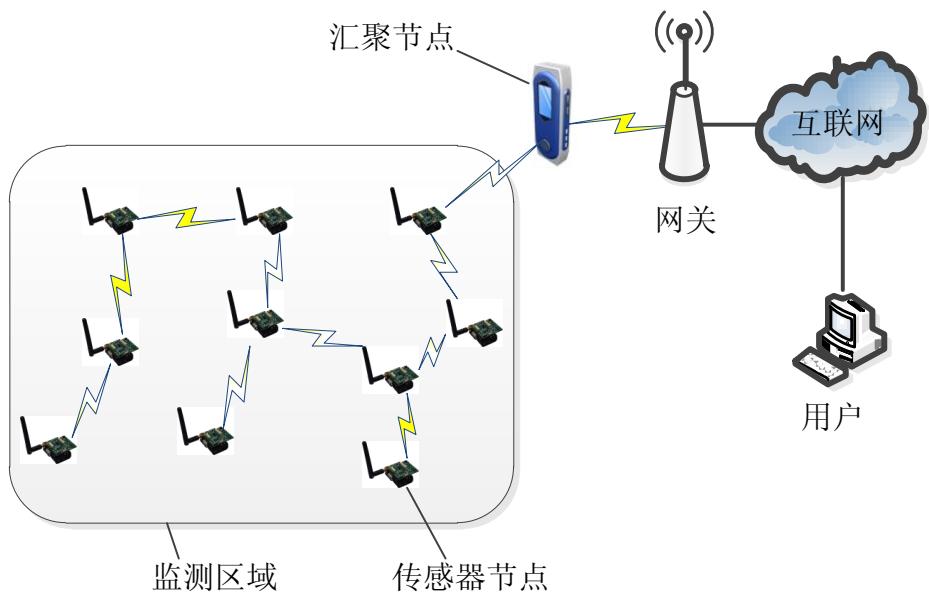


图 1.1 典型的无线传感器网络体系架构

传感器节点是 WSN 最基本的组成单元。典型的传感器节点体系结构如图 1.2 所示。由图 1.2 可知，WSN 中的传感器节点至少包括以下主要模块。

- **电源模块：**电源模块负责对传感器节点进行供电，一般由锂电池构成。由于传感器节点通常工作在野外环境，无论是对电源进行充电还是更换新的电池设备其代价都是非常高的并且难以实现的。因此，电源模块的容量以及节点的能量效率成为制约 WSN 长期工作的瓶颈。目前，人们正在寻求从环境当中捕获能量(energy harvesting)，例如将太阳能或车辆运动造成的路面震动所产生的动能转换成电能，并将其存储在电源模块中，从而加大 WSN 节点的能量储备，延长网络的生命周期<sup>[13, 14]</sup>。
- **感知模块：**感知模块由一种或者多种传感器组成，负责采集物理环境当中的某一种或者多种状态变量。常见的 WSN 传感器有温度传感器、声响传感器、红外传感器、震动传感器等。
- **存储模块：**存储模块负责存储传感器节点的程序和数据。程序包括传感器节

点所运行的系统程序和应用程序。数据包括传感器节点自身采集到的环境数据，其接收到的其他节点发送过来的环境数据，以及控制和命令信息等。

- **处理模块：**处理模块又称为控制模块，负责控制整个传感器节点的运行状态，一般由嵌入式的微控制器或微处理器组成。处理模块还会执行一些基本的计算任务，如对采集到的数据进行预处理、编码，对多个数据包进行融合等。
- **通信模块：**通信模块负责执行数据收发任务。传感器节点一般采用低功耗、低速率的无线通信技术。由于节点的体积较小，并且出于能量有效性的考虑，通信模块一般传输的距离比较短。因此，为了在较大物理范围内实现有效通信，需要借助其他节点进行中继或转发。这意味着，WSN 中的数据包通常需要通过多跳通信的方式才能从源节点到达目的节点。



图 1.2 典型的传感器节点模块组成图

除了上述模块，目前有一些 WSN 节点还包括了执行器模块。与感知模块只是被动地采集环境信息不同，执行器模块可以主动地通过控制和调节某些变量来影响和改变环境。既具有传感模块又具有执行器模块的节点组成的网络称为无线传感器/执行器网络(Wireless Sensor and Actor Network, WSAN)<sup>[15, 16]</sup>。

作为 WSN 的基本组成单元，已经有一些通用的传感器节点商业产品出现。国际上，加州大学伯克利分校、MIT、因特尔等研究机构纷纷发布了面向科研工作者和企业的通用传感器节点开发平台<sup>[11]</sup>。在国内，无锡儒安科技公司等也推出了类似产品<sup>[17]</sup>。

### 1.2.2 无线传感器网络通信协议栈

上一小节我们介绍了组成 WSN 的传感器节点、sink 节点和控制中心等，但是仅有这些设备还无法构成一个可以正常工作的无线传感器网络系统。和传统的计算机网络、蜂窝网一样，我们需要一套复杂、完整、功能强大的协议栈来辅助上述设备实现组网连接和数据传输。图 1.3(a)所示的是 OSI 组织制定的通信网络协议栈参考模型<sup>[18]</sup>，而图 1.3(b)给出的则是典型的 WSN 通信协议栈模型<sup>[1]</sup>。我们可以发现，WSN 的协议

栈相比 OSI 参考模型结构更为复杂，这是由 WSN 的独特性质所决定的<sup>[19]</sup>。

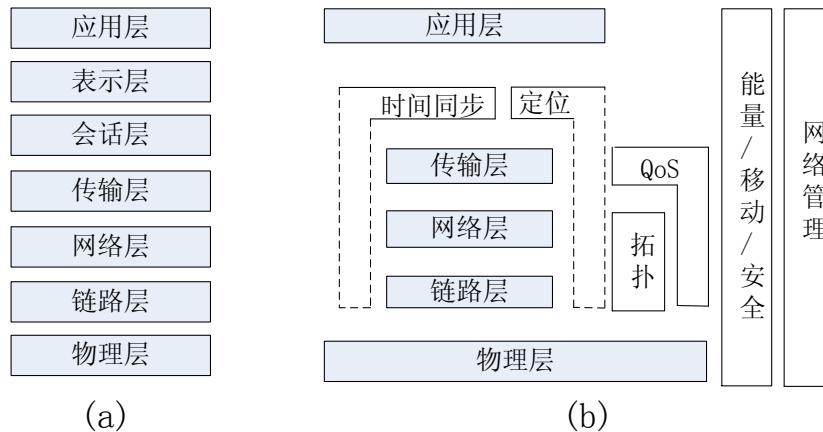


图 1.3 通信网络协议栈模型: (a)OSI 参考模型 (b)无线传感器网络通常采用的模型

下面逐一简要介绍 WSN 通信协议栈中的主要功能模块。

- **物理层:** 协议栈最基础的一层，实现点到点之间的无线数据收发功能。常用的物理层通信技术有窄带通信技术、扩频通信技术和正交频分复用技术等。
- **链路层:** 主要功能是控制和协调节点对无线信道的访问，又称为媒介接入控制(Medium Access Control, MAC)层，常见的 MAC 层技术有 TDMA、FDMA、CDMA、CSMA、ALOHA 等<sup>[20]</sup>。一般来说，链路层功能还包括帧校验、帧确认、差错控制等。
- **网络层:** 确定数据包的传输路径，由路由协议来实现。由于网络形态的多样性，WSN 的路由协议可以被划分为不同类型。常见的分类方法包括平面型路由和层次型路由、静态路由和动态路由、集中式路由和分布式路由等<sup>[21]</sup>。
- **传输层:** 实现端到端之间的差错控制、流量控制和拥塞控制。因特网中广泛采用的传输层协议有 TCP 和 UDP。针对 WSN 的特性，设计符合 WSN 需求的传输层协议是一个重要的研究方向。
- **应用层:** 为用户提供访问网络的接口，一般被各种各样的应用程序，如 Web 应用、FTP 应用等调用。
- **时间同步:** 为了使网络众多节点之间能相互通信和协作，需要实现节点之间的时间同步。一般由专门的同步硬件电路或软件协议来完成<sup>[22]</sup>。
- **定位:** 位置信息是重要的辅助信息，没有位置信息的传感器数据是没有意义的<sup>[23]</sup>。定位技术又分为节点的自定位技术和节点对目标的定位和跟踪技术<sup>[24]</sup>。
- **拓扑控制:** WSN 节点众多，网络结构复杂且易动态变化。拓扑控制模块负责生成和管理网络的拓扑结构，并应对旧节点失效和新节点加入等情况<sup>[25]</sup>。
- **能量管理:** 能量是 WSN 的瓶颈，是制约 WSN 长期运行的关键因素。能量

管理模块控制节点能耗状况，目的是提高节点能量效率，延长网络生命周期。

- **移动管理：**WSN 一般是静态的，但是在部分应用场景中，传感器节点或者 sink 可以运动。此时，需要移动管理模块来控制节点的移动速率、方向、轨迹等，以保障网络的连通性和低功耗<sup>[7]</sup>。
- **安全管理：**安全性不足是各种网络所面临的共同问题，WSN 也不例外。如何设计高效的网络安全和隐私保护协议，是 WSN 重要的研究方向之一<sup>[26]</sup>。
- **QoS 保障：**布设和运行网络的目的是为用户提供某种服务，而服务质量的好坏直接关系到用户对网络的评价和满意度。保障服务质量(Quality of Service, QoS)是一个系统工程，需要对协议栈的各个模块进行联合设计与优化<sup>[27]</sup>。

### 1.2.3 无线传感器网络应用

WSN 最突出的优势就是应用极其广泛。在人类生产生活的各个方面，WSN 均有用武之地。图 1.4 给出了若干种常见(但不完全)的 WSN 应用场景，下面逐一介绍之。

- **公共安全与国防：**在公共安全领域，WSN 能用来对危险物品、可疑目标等进行检测和监视，从而保障公共场所的安全；在军事国防领域，WSN 能用来对战场进行全天候监视，并收集情报信息，以辅助本方作战方案设计<sup>[2]</sup>。
- **环境监测：**进入 21 世纪以来，环境保护越来越受到人们的重视。如何实时地监测大气、湖泊、河流、森林等的环境质量状况日渐成为一个重要的课题，而 WSN 正好能够满足人类的这种需求<sup>[3]</sup>。
- **健康医疗：**健康是每个人幸福的基础。WSN 已被成功应用于健康医疗领域。无线体域网(Wireless Body Area Networks, WBAN)通过在患者体内植入极其微小的传感器芯片，可以实现对患者心率、血压、血糖等健康指标的监护，从而促进疾病的诊治，维护患者的健康<sup>[28]</sup>。
- **智能家居：**智能家居(smart home)是当前市场上最为火热的 WSN 应用之一。智能家居通过给房屋里的各种电器设备以及门窗等物体安装传感器，实时地获取这些设备和物体的工作状态信息，并将其远程发送给主人，同时接受主人的控制命令，从而实现家居的智能化<sup>[29]</sup>。
- **智能电网：**电网是维持人类生产生活的动力源。电网面临的突出问题是电力资源的损耗和浪费十分严重。智能电网(smart grid)通过在电力传输线路上布设电力传感器节点，能够获取大范围内的电力供应与消费状况，以便于调度中心进行合理的调度安排，从而减少电力损耗<sup>[30]</sup>。
- **移动感知系统：**近年来，智能手机逐渐普及起来。智能手机有个突出特点就是其上搭载了多种传感器，至少包括位置传感器(GPS)、声音传感器(麦克风)、图像传感器(camera)、重力传感器(陀螺仪)等。利用这些传感器，智能手机能

够感知环境中的很多物理信息。数量巨大且无处不在的智能手机如果能够协作地进行感知，将产生巨大的影响力。目前，针对多个智能手机组成的移动手机感知系统(Mobile Phone Sensing System, MPSS)得到了学术界和工业界的共同关注<sup>[31]</sup>。

- **工业自动化：**工业自动化是 WSN 的一个重要应用领域。在各种工业场合，如机械制造、化工制造、桥梁建造等，WSN 能够用来准确地收集生产和制造过程中的环境与物理状态信息，并实时地传输给控制中心，以辅助控制中心做出更加合理的决策安排<sup>[4]</sup>。



图 1.4 无线传感器网络部分典型应用场景示意图

### 1.3 无线传感器网络数据聚合调度概述

本节介绍面向 WSN 数据聚合传输调度研究的背景和现状。首先，介绍无线传感器网络数据聚合的基本概念、设计目标和关键技术，然后介绍数据聚合调度的含义，最后总结国内外学术界针对数据聚合调度所已经取得的研究成果以及数据聚合调度面临的研究挑战。

#### 1.3.1 无线传感器网络数据聚合基本概念

在前面已经指出过，WSN 的基本功能是从物理世界收集数据。由于 WSN 的能

量、带宽等资源都极为受限，而用户又对网络生命周期、传输时延等性能指标有着极为苛刻的要求，因此，进行大量数据的直接收集的可行性比较低。数据聚合技术的出现，为在资源受限的 WSN 中进行高效的数据收集开辟了一个有吸引力的方向。通过利用空间分布的不同传感器节点所采集到的数据之间的相关性，数据聚合允许中间节点(即转发或中继节点)在发送数据包之前，对已接收到的数据包和本地产生的数据包进行融合，然后将融合后的数据包发送给下一跳节点。图 1.5 以图示的方式给出了 WSN 数据聚合的基本含义。在物理状态监测区域中布设了许多传感器节点，汇聚节点(sink)需要收集这些节点所采集到的信息。如图 1.5(a)所示，直接将每个节点产生的数据包通过多跳通信方式转发给 sink 将给网络带来极大的传输负载，而且靠近 sink 的传感器节点由于有大量的数据转发任务而急剧消耗能量，过早地成为死亡节点，从而破坏网络连通性(connectivity)，造成所谓的“网络空洞”问题<sup>[32]</sup>。为了避免此类问题，在数据聚合中，中间节点可以采用某种数据融合方法，将其需要转发的数据包和本地产生的数据包进行融合，然后发送融合后的数据包，正如图 1.5(b)所示。此时，该节点又称为聚合节点。由于每个中间节点均可以进行数据融合，网络的传输负载将大大减少，并且节点之间也实现了一定程度的负载平衡。通过持续地消除冗余数据信息，数据聚合能够减轻节点通信用任务，从而有效节省网络的能量、带宽等资源，改善网络时延和生命周期<sup>[33]</sup>。尽管数据聚合具有很多优势，但是要获得这些优势并非轻而易举，而是需要克服一些关键的技术难题。下一小节将讨论 WSN 数据聚合所涉及到的若干关键技术。

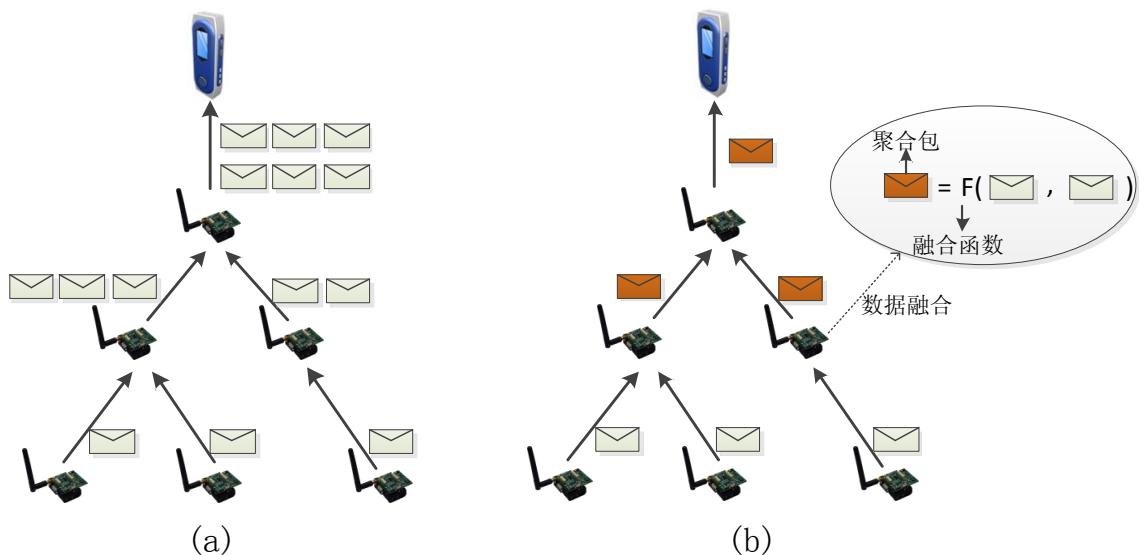


图 1.5 无线传感器网络数据收集与数据聚合示意图：(a)数据收集 (b)数据聚合

### 1.3.2 无线传感器网络数据聚合设计目标与关键技术

与路由、定位、同步等 WSN 中的常见功能模块一样，数据聚合的设计也需要满

足一个或多个网络性能指标。一方面，数据聚合方法的选择直接影响着网络性能；另一方面，数据聚合的各个功能模块之间相互作用，彼此耦合，共同对网络性能产生影响。图 1.6 以示意图的方式给出了数据聚合与网络性能之间的关系。

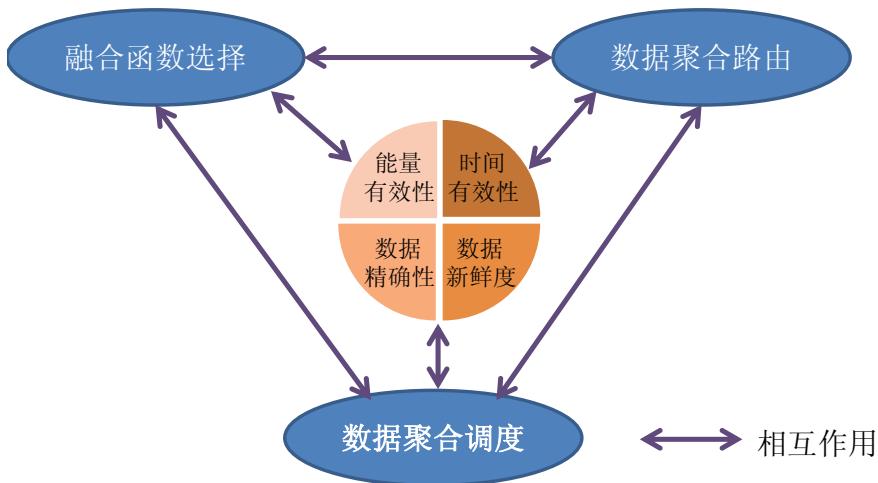


图 1.6 数据聚合与网络性能之间的作用关系示意图

一般来说，WSN 数据聚合的设计目标包括但不限于：

- **能量有效性：**相比原始数据收集，数据聚合所消耗的能量显著降低。但是由于 WSN 能量受限的本质并没改变，因此需要设计更加高效的数据聚合算法以进一步减少网络的能量消耗。另外，由于网络生命周期一般是由最早死亡节点的生命周期来决定，因此如何实现不同节点之间能量消耗的平衡也是数据聚合算法的重要设计目标之一<sup>[34]</sup>。
- **时间有效性：**WSN 的许多应用，如工业控制，灾害检测等，对数据的时效性有着越来越高的要求。这些应用场合往往要求数据聚合操作能够在尽可能短的时间内完成。在数据聚合中，为了进行高效率的数据融合，中间节点通常需要等待比较长的时间来接收其他节点的数据包。因此数据聚合的时延成为一个突出问题。如何降低数据聚合时延，成为一个重要的设计考量<sup>[12]</sup>。
- **数据精确性：**在数据聚合中，通过进行数据压缩，节点的传输负载能够显著降低，但是其代价是数据的有用信息也会有一定程度的损失，后果是数据的精确性(precision)有所降低<sup>[35]</sup>。为了提高聚合数据的精确性，通常可以增加参与数据聚合的传感器节点的数量，也可以优先选择数据精度更高的传感器节点作为源节点。
- **聚合新鲜度：**在数据聚合中，尽管来自不同传感器节点的数据包能够进行融合，但是对不同时间段产生的数据包进行融合将会导致不同的聚合新鲜度(aggregation freshness)。一般来说，数据包产生时间越接近，聚合新鲜度越

高，反之越低<sup>[11]</sup>。不同的应用对数据聚合的新鲜度有不同要求。

- **干扰感知度：**作为一种无线网络数据业务模式，与数据多播、数据广播等一样，数据融合也会受到无线冲突或干扰的影响。由于无线信道的开放特性，空间中同时进行的数据传输之间会相互干扰，严重时将导致丢包。在数据聚合中，由于传输的数据包通常是融合后的数据包，其蕴含的有效信息较普通数据包更多，包的丢失将产生更为严重的信息损失。因此，数据聚合算法需要尽可能克服干扰的不利影响，提高数据传输的成功率<sup>[12]</sup>。

为了达到上述设计目标，需要合理设计数据聚合方案。具体来说，一个完整的数据聚合解决方案至少包括了以下关键组成部分：

- **数据聚合函数：**数据聚合函数(aggregation function)决定数据包以何种方式进行压缩，其分类标准不一。具体来说，根据聚合后的数据量大小，分为完全(perfect)数据聚合和不完全(imperfect)数据聚合；根据在聚合操作前后数据所蕴含的信息含量有无损失，分为无损失(lossless)聚合和有损失(lossy)聚合；根据数据聚合是否基于应用数据的语义，分为应用依赖的(application-dependent)数据聚合、独立于应用的(application-independent)数据聚合和混合式(hybrid)数据聚合；根据传感器数据的操作级别，可以分为数据级聚合、特征级聚合和决策级聚合<sup>[9]</sup>。数据聚合函数直接决定了节点数据传输负载的大小、数据收集的精度等，从而进一步影响着网络性能的方方面面。选择何种数据聚合函数，需结合具体应用需求和网络设计目标来统筹决定<sup>[9]</sup>。
- **数据聚合路由：**给定网络连通图，数据聚合路由决定聚合后的数据包到基站之间的传输路径。一般来说，数据聚合路由分类两大类，即簇状(cluster-based)路由<sup>[36]</sup>和树状(tree-based)路由<sup>[37]</sup>。在簇状路由中，具有相同功能或聚集在相邻区域里的传感器节点形成一个簇，由随机产生的簇头节点负责收集簇内的数据包，并在簇头处进行数据融合；在树状路由中，由每一个非叶子节点负责收集自己的子节点的数据包，并进行数据融合，然后将融合后的数据包发送给父节点。注意到，无论是簇状路由还是树状路由，数据包传输的目的节点都是确定的，从而可以采用单播(unicast)技术实现，不需要采用能耗比较严重的广播(broadcast)技术<sup>[38]</sup>。
- **数据聚合调度：**数据聚合中，每个聚合节点通常需要完成三项操作，即等待并接收数据包、聚合数据包和发送数据包。聚合数据包的时间极短，一般可以忽略不计。数据聚合时延通常主要由等待并接收数据包的时间来决定。数据聚合调度确定每个聚合节点等待进行数据聚合的时间，并间接确定每个聚合包的传输时刻。一般的，数据聚合调度分为两大类，即非时隙(unslotted)数据聚合调度和时隙(slotted)数据聚合调度<sup>[11]</sup>。在非时隙数据聚合调度中，

节点之间无须进行时间同步，数据包的传输采用基于竞争的协议，如 ALOHA、CSMA(Carrier Sensing Multiple Access, 载波侦听多址接入)来实现。在时隙数据聚合调度中，时间被划分为连续的等长时隙，由专门的时间同步协议负责实现网络同步，并采用 TDMA(Time Division Multiple Access, 时分多址接入)机制给每个聚合节点分配确定的等待时隙数和具体的发送时隙。时隙数据聚合调度通过为节点分配固定和冲突感知的传输时隙，能够有效规避信道竞争产生的额外开销，适用于拓扑相对固定且对能耗极为敏感的 WSN 应用<sup>[39]</sup>。因此，本文聚焦于时隙数据聚合调度。在没有特别指出的情况下，本文接下来出现的调度均指基于时分多址接入(TDMA)的数据聚合调度。

### 1.3.3 无线传感器网络数据聚合调度基本概念与特点

数据聚合调度模块是 WSN 网络协议栈的重要组成部分。下面以一个简单的网络示例，来说明数据聚合调度的相关基本概念。图 1.7 给出了一个包含 6 个节点和 1 个基站的小规模 WSN，基站负责收集来自节点的聚合数据。图 1.7(a)给出了一种数据聚合路由的示意图，其中实现表示所选择的传输路径。由图可知，其采用的是以基站为根节点的树状聚合路由。图 1.7(b)给出的是一种具体的数据聚合调度方案示意图。图中，网络的生命周期被划分为等长的连续时隙，单个时隙的长度为一次数据传输所需的时间，包括发送数据包的时间和接收确认包的时间。在初始时刻，每个节点产生一个包含感知信息的数据包。接下来的每个时隙里，我们需要合理安排哪个节点发送数据，哪个节点接收数据。节点的发送时机受到诸多因素影响。比如，为了进行数据融合，节点 4 需要等待接收完节点 1 和节点 2 的数据包之后才能传输。具体来说，在图 1.7(b)中，节点 4 等待了 4 个时隙才发送数据。这段时间称为节点 4 的等待时间；又比如，为了避免干扰危害，节点 1 和节点 2 不能在同一个时隙发送，否则将产生包冲突。由此可见，调度需要考量的因素很多，具体的调度算法设计通常是比较复杂的。图 1.7(b)给出的是一种可行的(feasible)调度方案。采用该调度方案，数据聚合能够在 6 个时隙里完成。也就是说，6 个时隙之后，基站能够收到包含全网所有节点数据信息的聚合包。此时，这 6 个连续时隙称为该调度方案的调度周期，其周期长度为 6。调度周期长度是衡量数据聚合调度的重要性能指标，其直接影响到数据聚合的新鲜度和网络的延迟<sup>[40]</sup>。通过在时间上重复数据聚合调度过程，可以实现周期性的数据聚合。在每个具体的时隙，节点可能的状态有发送、接收和休眠等。在每个时隙，对于没有数据收发任务的节点，可以让其彻底关闭通信模块，进入深度休眠状态，从而有效节省节点的能耗。

上述示例中，我们假设网络所有节点共享一个通信信道。实际系统中，网络可能有多个可用的非重叠(non-overlapping)信道。此时，节点对信道的访问不仅需要考虑

接入时机，也需要考虑接入哪个信道。数据聚集调度将不仅进行时隙分配，还要进行信道分配<sup>[41]</sup>。在假设数据聚合路由固定、节点的传输速率和传输功率也是固定的情况下，数据聚合调度负责给每个节点分配传输时隙。此时的数据聚合调度属于狭义调度。与狭义调度相对应的是广义调度。所谓广义调度，就是综合考虑路由、时隙分配、传输速率、传输功率等，以进行调度方法的联合设计<sup>[42]</sup>。之所以需要进行广义调度，是因为数据聚合调度的性能不仅受到时隙分配算法的影响，而且与路径选择算法、速率控制算法和功率控制算法密切相关<sup>[11]</sup>。广义数据聚合调度已经成为 WSN 的一个重要的研究方向。

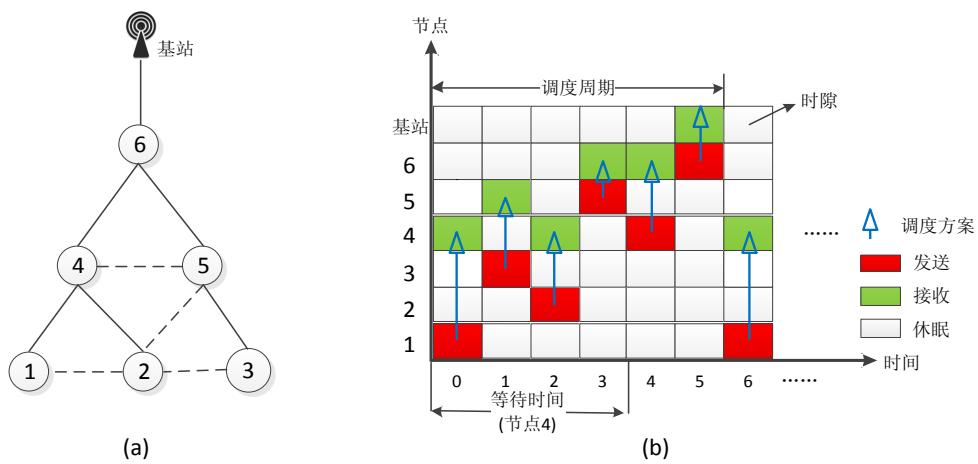


图 1.7 一个说明数据聚合调度基本概念的简单例子: (a)数据聚合路由示意图 (b)一种可行的聚合调度方案示意图

### 1.3.4 无线传感器网络数据聚合调度研究进展与研究挑战

相比融合函数选择和数据聚合路由，数据聚合调度研究的历史并不长。通过分析和整理现有文献，我们将已有的数据聚合调度研究工作依据节点访问信道方式的不同划分为时分复用类和混合复用类<sup>[11]</sup>。

- **时隙复用类:** 这类数据聚合调度方法采用时分多址接入，即 TDMA 作为接入无线媒介的手段。文献<sup>[43-57]</sup>中提出的数据聚合调度方法属于这一类。时隙复用类调度的主要任务是为网络中的节点分配传输时隙。一般来说，为了降低网络时延、提升网络吞吐量，我们需要尽可能地减少所花费的时隙数量。也就是说，我们需要在每个时隙调度尽可能多的传输节点。为了达到这一目的，除了时隙分配方法，数据传输的路径选择策略、节点的传输功率控制策略等也需要统筹考虑。
- **混合复用类:** 所谓混合复用，就是除了 TDMA，还综合使用另外的无线媒介接入技术，例如 CDMA 和 FDMA 等，来实现对无线信道的访问。具体来说，

混合复用类数据聚合调度又分为基于 TDMA&CDMA 的数据聚合调度<sup>[58, 59]</sup>和基于 TDMA&FDMA 的数据聚合调度<sup>[41, 59, 60]</sup>。

在本文第 2、3 和 4 章的相关工作部分，我们将更详细地介绍文献中数据聚合调度相关的具体研究工作。尽管目前已经有了许多关于聚合调度的研究，但是由于该问题本身的复杂性，以及网络模型和应用场景的多样性，WSN 中的数据聚合调度研究依然面临着诸多技术挑战。具体来说，数据聚合调度研究至少面临着四大难题，分别为：克服无线干扰对调度的不利影响，提高调度的时间效率，权衡多种互相冲突的网络性能指标，和设计易于分布式实现与工程化的调度协议。

- **克服干扰危害：**干扰(interference)是制约无线网络性能的瓶颈。干扰会造成数据包的冲突和丢失，从而浪费网络能量，增加数据传输时延。在任一时隙，哪些链路能够同时传输，本质上是由这些链路相互之间的干扰关系决定的。由于实际干扰情况非常复杂，如何用抽象的模型准确地刻画链路之间的干扰关系是一个具有挑战性的课题。当前学术界广泛使用的主流干扰模型有协议干扰模型<sup>[61]</sup>和物理干扰模型<sup>[62]</sup>等。这两种模型含义不同，但是其背后的共同思想是把干扰看作某种有害的东西，并通过阻止相对较近的链路同时传输来避免或减轻干扰对数据包正常传输的潜在危害。这类算法可以被视为被动或保守的调度方法。然而，随着物理层无线通信技术的进步，人们发现干扰并不一定是绝对有害的。与噪声不一样，干扰属于结构化的信号。通过首先将较强的干扰信号解码出来并消除之，我们可以进一步解码相对较弱的有用信号。这种技术称为干扰消除技术，或物理层多包接收技术<sup>[63]</sup>。干扰消除的思想为数据聚合调度算法的设计提供了一种新的思路。实际上，更进一步，人们发现无线干扰也有可能为我所用。通过故意制造特定的干扰，可以在不影响正常数据接收的同时，顺便在其上搭载一些控制命令，从而实现数据和控制信息的同时、同频传输<sup>[63]</sup>。
- **提高时间效率：**数据聚合调度有很多设计目标，其中最直接、也是最重要的一种目标是最小化调度周期长度，或最小化数据聚合时延。已有的相关研究工作中的绝大部分都聚焦于此。事实上，该问题是极其复杂和难处理的(*intractable*)。在理论上已经证明，即使是在最简单的网络模型和干扰模型下，最小数据聚合时延问题也是 **NP**-难的<sup>[12]</sup>。为了提高时间效率，虽然人们提出了各种各样的启发式算法和近似算法，但其实际性能远远未达到最优。在 TDMA 网络中，提高时间效率的关键在于实现更大程度上的时隙复用。由于单纯地通过时间复用改善网络时延难以再有大的突破，研究者逐渐考虑将 TDMA 技术与其他相关技术结合起来，通过设计广义或跨层的调度协议与算法，以进一步改善网络时延性能。一般来说，TDMA 与 FDMA 的结合，TDMA

与 CDMA 的结合, TDMA 与功率控制的结合, TDMA 与路由的结合等, 都是值得进一步探索的方向<sup>[11]</sup>。

- **多种性能折中:** 除了时间效率, 数据聚集调度也直接影响着众多其他的网络性能指标, 包括能量有效性、数据精确性、聚合新鲜度等。在设计数据聚合调度时, 需要统筹考虑这些性能指标。理想的状况是所有这些指标均同时达到最优。然而, 事实上这些性能指标两两之间往往是互相冲突的。例如数据聚合时延与数据精确性、能量有效性与数据精确性、数据聚合时延与能量有效性等均是互相影响和冲突的<sup>[35, 64, 65]</sup>。如何在多个设计目标之间达到平衡, 是数据聚合调度面临的重要问题。在处理这种多目标设计问题时, 我们既可以寻求多个设计目标的折中, 也可以寻求其他变量约束下的单一变量最优化。当前的调度研究基本还是以单一设计目标为主, 考虑多个设计目标的数据聚合调度方法是一个极具挑战性的研究方向。
- **可分布式实现:** 从实际实现方式来讲, 数据聚合调度分为集中式调度和分布式调度。在集中式调度中, 基站收集全网的拓扑信息、干扰信息等, 然后计算调度方案, 再将调度方案以多跳广播(multi-hop broadcasting)的方式发送给全网节点。而在分布式调度中, 通过让每个节点与邻居节点以消息传递的方式进行协商, 来产生每个节点各自的调度方案。集中式调度的优点是能得到很好的调度方案, 其缺点是不适用于动态网络情形(需频繁地重新计算调度)。分布式调度的优点是能适应网络动态性, 但是很难得到性能较好的调度方案<sup>[66]</sup>。当前文献中的调度方案<sup>[43-57]</sup>大多是集中式的。针对网络动态性, 如时变信道条件、不可靠的传输链路、节点频繁加入/失效/移动等情形, 设计简单、快速、有较好性能, 并且易于实现的分布式聚合调度方法是一个重要的研究方向。

## 1.4 本文的研究内容与结构安排

### 1.4.1 本文的研究内容与主要创新点

数据聚合是 WSN 中一种特殊的业务模式。虽然数据聚合为在资源极为受限的 WSN 中进行能量高效的数据收集开辟了一条新的道路, 但是为了等待进行数据融合, 数据聚合将为数据收集带来额外的包延时。为了更好地满足时延敏感应用的需求, 需要设计高效的数据聚合调度算法。数据聚合调度不仅关系到网络的时延性能, 而且与能量消耗、数据精度等也密切相关。本文主要研究无线传感器网络中的数据聚合调度算法和技术。本文从理论和实验两方面, 针对复杂网络场景, 如多用户干扰、链路丢包等情形下的数据聚合调度问题展开深入研究。本文所取得的创新点如下:

- 1) 针对协议干扰模型下的 WSN 数据聚合调度问题, 提出了一种改进的两阶段

数据聚合方案。在第一阶段，基于连通支撑集构造数据聚合树；在第二阶段，分别基于分层贪心策略和网络分区与着色策略进行数据聚合调度。论证了方案的可行性，推导了算法的时延上界，并证明了所提出的两种调度算法均是具有常量近似比的次优算法。仿真实验表明，调度算法的平均性能远好于理论上界，并且与文献中的代表性算法相比，本文算法在多种网络场景下均能一定程度上降低数据聚合的总时延。

- 2) 针对物理干扰模型下的 WSN 数据聚合调度问题，本文提出了一种完全分布式的数据聚合解决方案。该方案以迭代网络分区与着色技术为基础，由分布式数据聚合树构造子算法、分布式链路调度子算法和分布式功率控制子算法联合而成。通过合理配置调度和功率控制模块的参数，证明了该方案在物理干扰模型下的可行性。推导了该方案在最坏情况下的聚合时延上界。理论结果表明，该方案的最坏时延与网络规模无关，从而显示算法具有良好的扩展性。最后通过大量的仿真实验评价了该方案在不同场景下的平均时延、能耗等性能。
- 3) 基于主动的干扰消除理念，将物理层多包接收模型引入 WSN 数据聚合调度中。分别基于随机链路优先策略和最短链路优先策略，提出了两种时延高效的启发式聚合调度算法。证明了两种调度算法的可行性，分析了两种调度算法的计算复杂度。仿真实验结果表明，相比传统的基于单包接收模型的调度方法，本文算法能够显著地降低数据聚合平均时延，提高网络吞吐量。
- 4) 针对低占空比且链路不可靠的无线传感器网络，提出了一种错误容忍的数据聚合协议。该协议通过采用链路可靠性增强手段，来提高点到点的传输成功率，最终改善网络链路损耗状况下的数据聚合信息质量。依据采用的链路可靠性增强技术的不同，我们分别基于动态睡眠调度调整策略，基于机会路由策略和基于混合策略，为该协议设计了三个变种。仿真实验表明，在链路不可靠的网络场景下，上述三种协议均能够不同程度地提升数据聚合过程中汇聚节点所接收聚合包的信息质量。其中基于混合策略的协议较另外两种协议对数据质量性能的提升更为显著。

#### 1.4.2 本文的结构安排

本文的结构安排如图 1.8 所示。具体来说，本文包括如下章节：

- 1) 第一章，绪论。本章概述了论文的研究背景、研究动机、研究现状和研究框架。首先简要介绍了无线传感器网络的背景，包括网络体系结构、通信协议栈和应用场景等，然后介绍无线传感器网络数据聚合的基本概念、设计目标、关键技术，以及数据聚合调度的基本概念、研究进展和研究挑战，接着简要

介绍本文的主要研究内容以及所取得的创新点。

- 2) 第二章, 协议干扰模型下的无线传感器网络数据聚合调度。协议干扰模型是一种重要的描述无线网络节点相互之间干扰关系的抽象模型。由于该模型相对比较简单, 易于描述和处理, 因而受到学术界的广泛关注。本章研究了协议模型下的无线传感器网络数据聚合调度问题。首先简要介绍研究背景和相关研究工作, 然后给出网络模型与问题的数学描述。接着介绍所提出的两阶段调度方案, 包括一种改进的数据聚合树构造算法 **IDATC**, 一种基于分层贪心策略的调度算法 **LGAS**, 和一种基于网络分区与着色策略的调度算法 **NPCS**。我们详细给出了算法的框架、伪代码、可行性分析、时延性能分析等, 并通过仿真实验评价了算法的性能。
- 3) 第三章, 物理干扰模型下无线传感器网络分布式数据聚合调度。物理干扰模型是一种更为实际、更为精确的干扰模型。已有的物理干扰模型下的聚合调度算法大都是集中式的, 存在其应用的局限性, 只适用于具有静态拓扑的无线传感器网络。当网络拓扑动态变化, 例如新节点随机加入、旧节点失效时, 集中式算法的效率将很低, 迫切需要设计分布式的数据聚合调度方案。为此, 本章提出了一种全分布式的数据聚合解决方案 **DDA**。在该方案中, 我们首先基于迭代分区策略, 提出了一种分布式数据聚合树构造算法 **DDATC**。然后, 基于迭代分区与着色策略, 提出了一种分布式链路调度算法 **DLS**。在物理层, 我们采用两种可选的传输功率控制策略, 即均匀功率分配 **PA-U** 和线性功率分配 **PA-L**。通过合理配置调度和功率控制模块的参数, 证明了 **DDA** 方案在物理干扰模型下的可行性。同时, 推导了 **DDA** 在最坏情况下的聚合时延。最后, 通过大量的仿真实验评价了 **DDA** 方案在不同物理层功率控制策略和不同网络参数设置下的时延、能耗等性能。
- 4) 第四章, 基于多包接收技术的树状无线传感器网络数据聚合调度。传统的聚合调度方法, 不管是基于协议干扰模型还是基于物理干扰模型, 其基本原理都是让潜在的相互冲突链路尽可能地在不同的时隙调度, 从而避免或降低无线干扰对目的信号接收的危害。然而, 类似被动和保守的干扰处理策略将导致较低的时隙利用率, 从而使网络的时延性能下降。为了提高时隙利用率, 本章基于主动和积极的干扰消除理念, 将物理层多包接收模型引入数据聚合调度问题中, 并研究此模型下的调度算法。本章首先介绍了问题背景和相关工作, 然后给出网络模型和问题描述, 接着详细介绍提出的两种聚合调度算法, 即随机链路优先贪心调度算法 **RFGS** 和最短链路优先贪心调度算法 **SFGS**, 并分析了两种算法的可行性和计算时间复杂度。最后通过仿真实验评价了算法在各种网络场景下的平均时延和吞吐量性能。

- 5) 第五章，面向低占空比无线传感器网络的错误容忍数据聚合协议。当网络中出现由于传输错误而导致的数据包丢失时，传统的数据聚合协议的性能将急剧降低。这体现在此时 sink 节点收到的聚合数据包中的有效信息量将大大减少。为了克服网络丢包对数据聚合性能的损害，迫切需要设计一种传输错误容忍的数据聚合协议。针对低占空比且链路不可靠的无线传感器网络，本章提出了一种错误容忍数据聚合协议 **FODAP**。依据采用的链路可靠性增强技术的不同，我们分别基于动态睡眠调度调整策略，基于机会路由策略和基于混合策略，为该协议设计了三个变种，即 **FODAP-D**, **FODAP-R** 和 **FODAP-H**。本章首先介绍了研究背景和相关工作，接着给出了系统模型描述，包括低占空比模型、网络模型、链路模型等，然后给出了详细的协议设计，包括总体框架、设计细节和实现方案等，最后通过模拟实验评价了协议的性能。
- 6) 第六章，总结与展望。本章首先总结了全文的研究内容，包括所研究的具体问题、提出的创新性解决方法和所取得的成果，然后给出了下一步的研究方向。

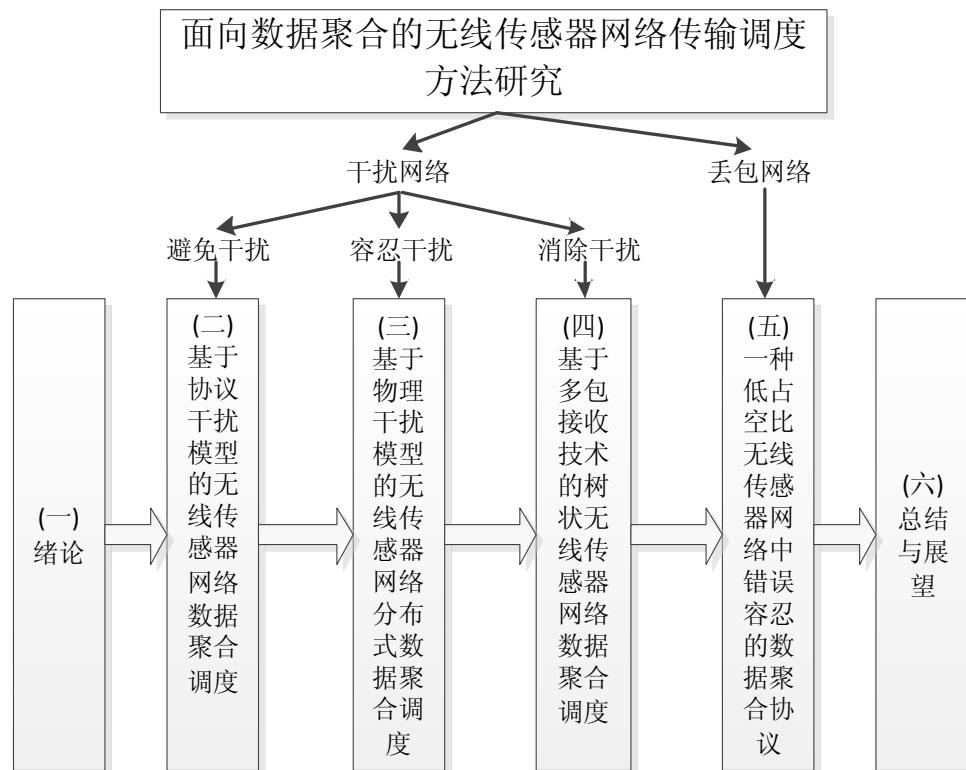


图 1.8 本博士论文的研究框架与章节安排示意图

### 1.5 本章小结

传输调度是无线传感器网络数据聚合的一项关键技术。本博士课题研究了面向数据聚合的传输调度算法和协议。作为统揽全文的绪论章节，本章概述了该课题研究的背景、现状与挑战，以及论文的总体框架和具体的研究子课题及取得的创新点。接下来将分章节地介绍每一个子课题的研究内容。

## 第2章 协议干扰模型下的无线传感器网络数据聚合调度

### 2.1 引言

数据聚合(data aggregation)是无线传感器网络(Wireless Sensor Network, WSN)一项最基本的业务模式之一<sup>[9]</sup>。对于许多实时和时延敏感的应用，例如灾难预警、入侵检测等，数据的时效性非常重要。在这种情形下，数据聚合业务需要在严格的时延约束下完成。在实际应用场景中，数据聚合的时延受到诸多因素的影响，例如聚合路径的选择、节点传输时机的调度等。这些因素均属于广义传输调度的范畴<sup>[10]</sup>。在文献中，探寻无线传感器网络数据聚合时延的理论最小值并设计高效的聚合调度算法/协议以达到或逼近这一理论最小时延成为了重要的研究课题，获得了学术界的广泛关注<sup>[43-57]</sup>。这个研究课题不仅具有理论意义，即获得数据聚合这一 WSN 基本业务的理论最优时延，同时也具有重要的现实意义，即可以用来衡量实际数据聚合协议的性能优劣和指导实际系统、尤其是对时延敏感的系统中的网络协议设计。

在最小时延数据聚合调度问题研究中，一个重要的技术挑战就是对网络中存在的各种传输请求进行合理的调度和安排。和无线网络中的一些经典调度问题，如链路调度<sup>[67, 68]</sup>、广播调度<sup>[69-71]</sup>一样，聚合调度也面临着一个相同的挑战，那就是克服无线干扰的影响。为了处理干扰，首先需要寻找一种有效的手段对无线干扰进行建模，以准确刻画干扰对数据传输的影响。然而，在实际系统中，物理环境中的无线干扰往往是复杂多变和难以建模的。为了简化网络协议的设计，研究者采用了一些简单的抽象模型(Abstract Models)来刻画无线干扰，其中包括主干扰模型<sup>[72]</sup>、单位圆盘图干扰模型<sup>[73]</sup>等。近年来，基于主模型的聚合调度<sup>[40, 41, 74]</sup>和基于单位圆盘图模型的聚合调度<sup>[12, 43, 46, 48-54, 56, 66]</sup>均得到了深入的研究。这两种干扰模型尽管简洁，但是均将无线干扰的作用范围限定在节点的传输范围内。然而，干扰的实际作用范围可能会超出传输范围。为此，人们提出一种更为一般化的干扰模型，即协议干扰模型<sup>[75]</sup>。协议干扰模型是一种更具有普适性的模型。事实上，上述主模型和单位圆盘图模型均可看作协议干扰模型的特殊形式。然而，针对这种一般化的协议干扰模型下的数据聚合调度问题，当前的相关研究还比较匮乏。

本章研究协议干扰模型下的无线传感器网络数据聚合调度问题，目的是设计干扰感知的数据聚合方案，以最小化网络级的数据聚合时延。本章首先对该问题进行了建模，并说明了其为 **NP**-难的。接着提出了一种两阶段的数据聚合调度方案。在第一阶段，提出了一种基于连通支撑集的数据聚合路由树构造算法 **IDATC**；在第二阶段，分别基于分层贪心策略设计了一种集中式的聚合调度算法 **LGAS**，和基于网络分区与

着色策略，设计了一种分布式的聚合调度算法 **NPCS**。从理论角度证明了所提出的方案在协议干扰模型下的可行性，并分析和推导了 **LGAS** 和 **NPCS** 算法在最坏情形下的聚合时延。具体来说，证明了两种算法的时延上界为  $\mathcal{O}(R + \Delta)$ ，其中  $R$  和  $\Delta$  分别为网络跳数半径和最大节点度。注意到  $\max(R, \Delta)$  是数据聚合调度问题一个理论下界<sup>[46]</sup>，因此，两种算法均是具有常量近似比  $\mathcal{O}(1)$  的近似算法。最后通过大量的仿真实验，对所提出的数据聚合方案的平均性能进行了评价。实验结果表明，本章算法的实际性能远远好于理论上界，并且与文献中的代表性算法相比，能够在一定程度上降低数据聚合的总时延。

## 2.2 相关工作

近年来，无线传感器网络中的最小时延数据聚合调度问题受到学术界的广泛关注。通过采用单位圆盘图模型对干扰进行建模，文献<sup>[12]</sup>首次从理论上证明了该问题是 **NP**-难的，并提出了一种时延上界为  $(\Delta - 1) \cdot R$  个时隙的聚合调度算法。自此之后，一系列旨在改善该时延上界的研究工作逐渐开展起来。基于极大独立集，文献<sup>[46]</sup>提出了一种时延上界为  $23R + \Delta - 18$  的聚合调度算法。由于  $\Delta$  此时不再是时延上界中的乘数因子而只是加性因子，因此相比文献<sup>[12]</sup>，文献<sup>[46]</sup>所提出的算法对网络时延有较大的改进，尤其对于最大节点度  $\Delta$  比较大的网络场景。接着，基于连通支撑集，文献<sup>[53]</sup>提出了三种近似算法，其时延上界分别为  $15R + \Delta - 4$ ， $2R + \Delta + \mathcal{O}(\log R)$  和  $(1 + \mathcal{O}(\frac{\log R}{\sqrt[3]{R}}))R + \Delta$ 。最近，采用一种新颖的基于邻接支配节点的调度策略，文献<sup>[52]</sup>将数据聚合时延上界降低到了  $12R + \Delta - 11$ 。另外，通过并行地执行数据聚合路由构造算法和聚合传输调度算法，文献<sup>[43]</sup>提出了另外一种改进的、时延上界为  $(\lfloor \frac{2\pi}{\arccos \frac{1}{1+\epsilon}} \rfloor + 4)R + \Delta - 4$  的调度策略，这里  $\epsilon$  是一个介于 0.05 到 1 之间的常数。针对多信道无线传感器网络，文献<sup>[48]</sup>也提出了一种具有理论性能保证的聚合调度算法。注意到，上述聚合调度算法基本都是集中式的，因此不是非常适用于拓扑频繁动态变化的网络场景。为此，文献<sup>[66]</sup>提出了一种分布式的、时延上界为  $48R + 6\Delta + 16$  的聚合调度算法。此后，文献<sup>[56]</sup>提出了一种改进的、时延上界仅为  $16R + \Delta - 14$  的分布式聚合调度算法。除了单位圆盘图干扰模型，基于主干扰模型的最小时延数据聚合调度问题也受到了关注。文献<sup>[60]</sup>针对树状无线传感器网络，提出了时延高效的数据收集调度和数据聚合调度方案，并证明了在某些特殊情形下，所提出的调度方案是时延最优的。

干扰建模是无线网络研究中的一个基础问题<sup>[61]</sup>。主干扰模型又称为节点排斥干扰模型(Node-exclusive Interference Model)，由于只考虑节点自身发射的射频信号的干扰，该模型与半双工约束本质上是一致的。单位圆盘图干扰模型将有用信号和干扰的

作用范围均限定在具有单位半径的圆盘之内。该模型的优势在于可以采用数学图论中工具，如边着色技术、顶点着色技术等进行网络算法和协议的设计与性能分析<sup>[73]</sup>。上述模型均将干扰作用范围进行了严格约束，然而干扰的实际作用范围可能会超出传输范围。为此，人们提出一种更为一般和普适的干扰模型，即协议干扰模型<sup>[75]</sup>。在该模型下，节点的干扰范围可以达到传输范围的若干倍。另外，近年来，物理干扰模型(Physical Interference Model)<sup>[76, 77]</sup>、高斯信道模型(Gaussian Channel Model)<sup>[78]</sup>、基于测量的干扰模型(Measurement-based Interference Model)<sup>[79]</sup>也被纷纷提出。通常来说，这些模型比协议干扰模型更为精确，但是相应地其复杂度也更高。

## 2.3 网络模型与问题描述

### 2.3.1 网络模型

考虑一种常用的无线传感器网络模型。在该模型中，传感器集合表示为 $\mathcal{V}$ ， $\mathcal{V}$ 中的节点任意地布设在二维平面里。 $\mathcal{V}$ 中有一个节点 $v_s$ 充当 sink 节点，其担任 $\mathcal{V}$ 中所有其他普通传感器节点的上级节点。每个节点装备有一个半双工的全向天线。采用该天线，节点在某一时刻能够发送或者接收数据，但无法同时进行数据收发。所有节点使用一个共享的无线信道。将集合 $\mathcal{V}$ 中的节点形成的无线网络建模成图 $G(\mathcal{V}, \mathcal{E})$ ，其中 $\mathcal{E}$ 表示网络中的通信链路集合。对于 $\mathcal{V}$ 中的任意两个节点 $u$ 和 $v$ ，链路 $(u, v)$ 存在，即 $(u, v) \in \mathcal{E}$ ，的充分必要条件是节点 $u$ 和 $v$ 在彼此的通信范围之内。对于任意两个节点 $u$ 和 $v$ ，用 $d(u, v)$ 表示它们之间的欧式距离。所有节点均具有相同的传输半径 $r$ 。假设网络中所有节点之间是时间同步的，并且时间被划分为连续的等长时隙。每个时隙的长度设定为足够发送或者接收一个数据包的长度。

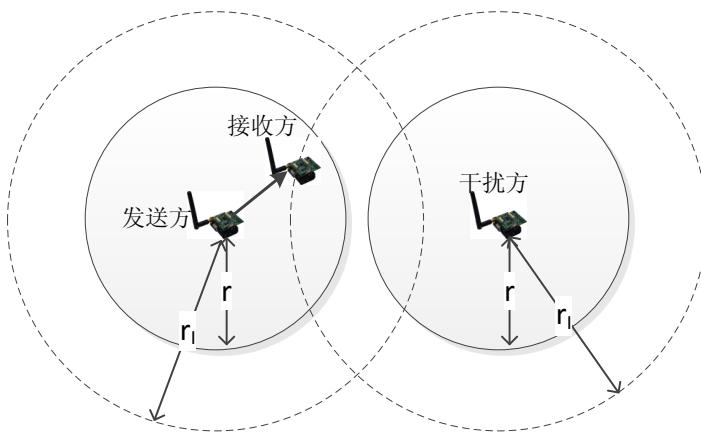


图 2.1 无线网络中的协议干扰模型示意图

### 2.3.2 协议干扰模型

无线信道的一个基本特性就是开放。在无线环境下，某条链路的发送方发射的信

号对于该链路的接收方来说，是有用的信号，但是对于其他的传输链路来说，却是干扰。在同一时刻，采用相同信道的多条无线链路相互之间有可能因干扰导致某条甚至全部链路传输失败。在本章的研究中，我们采用协议干扰模型(Protocol Interference Model, PrIM)<sup>[75]</sup>来刻画无线干扰对链路层传输的影响。图 2.1 以图示的形式直观地给出了该模型的含义。具体来说，在协议干扰模型中，假设发送方的传输范围和干扰范围均为以自身为中心的同心圆，且其传输半径和干扰半径分别为 $r$ 和 $r_I$ 。链路 $(u, v)$ 传输成功的充分必要条件是：接收方 $v$ 在发送方 $u$ 的传输范围之内且在所有其他同时活跃的发送方的干扰范围之外，即链路 $(u, v)$ 满足 $d(u, v) \leq r$ 且 $d(w, v) > r_I, \forall w \in \mathcal{S}_u \setminus \{u\}$ 。其中 $\mathcal{S}_u$ 表示与 $u$ 同时发射的节点集合。在 PrIM 中，定义干扰半径与传输半径的比值为干扰比 $\rho$ ，即 $\rho = r_I/r$ 。在本章中，假设所有节点均有相同的干扰半径和传输半径，并且干扰比 $\rho$ 也是固定的常数，一般来说 $\rho \geq 1$ 。

### 2.3.3 问题描述

假设数据聚合是按照轮次进行的。每一轮中，所有节点产生一个相同大小的数据包。经过数据聚合，sink 节点 $v_s$ 得到源自网络所有节点的数据聚合包。在中间节点进行数据融合时，假设完全聚合(perfect aggregation)<sup>[46]</sup>，即多个相同大小的数据包能够融合成一个与原始数据包等长的聚合包。让 $\mathcal{A}$ 和 $\mathcal{B}$ 表示 $\mathcal{V}$ 的两个不相交的非空子集。如果在某个时隙， $\mathcal{A}$ 中的全部节点同时发送数据包，且所有发送的数据包都能被 $\mathcal{B}$ 中的节点成功接收，则我们称数据包在该时隙内从 $\mathcal{A}$ 聚合到 $\mathcal{B}$ 。数据聚合调度定义为满足下列条件的发送节点集合序列 $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_l$ ：(1)  $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset, \forall i \neq j$ ，(2)  $\bigcup_{i=1}^l = \mathcal{V} \setminus \{v_s\}$ ，(3) 在时隙 $k (k = 1, 2, \dots, l)$ ，数据包从 $\mathcal{S}_k$ 聚合到 $\mathcal{V} \setminus \bigcup_{i=1}^k \mathcal{S}_i$ ，且经过 $l$ 个时隙，所有节点的数据包都被聚合到 sink 节点 $v_s$ 。这里条件(1)确保每个节点至多传输一次，条件(2)确保所有节点都有机会传输。这里 $l$ 即是数据聚合调度长度，也称为数据聚合时延。

据此，定义最小时延数据聚合调度(Minimum Latency Data Aggregation Scheduling, MLDAS)问题为：给定网络 $G(\mathcal{V}, \mathcal{E})$ ，寻找一个数据聚合调度序列 $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_l$ ，使得数据聚合时延 $l$ 最小。文献<sup>[12]</sup>已经证明，当采用单位圆盘图干扰模型时，MLDAS 问题是 NP-难的。考虑协议干扰模型，如果我们令 $\rho = 1$ ，则该模型简化为单位圆盘图模型。因此，协议干扰模型下的 MLDAS 问题，作为原始 MLDAS 问题的一般情形，也是 NP-难的。针对 NP-难类型的问题，目前还不存在多项式时间复杂度的最优(Optimal)算法。因此，本文将致力于寻找次优的(Sub-optimal)、计算复杂度较低，且具有理论性能保证的有效算法。

### 2.4 协议干扰模型下的数据聚合调度算法

本文采用一种两阶段的方案来解决协议干扰模型下的 MLDAS 问题，方案的架构如图 2.2 所示。在第一阶段，基于精简的连通支撑集，构造一棵数据聚合树；在第

二阶段，进行数据聚合调度设计。具体来说，我们基于分层的贪心策略设计了一种集中式调度算法，并且基于网络分区与着色策略设计了一种分布式调度算法。

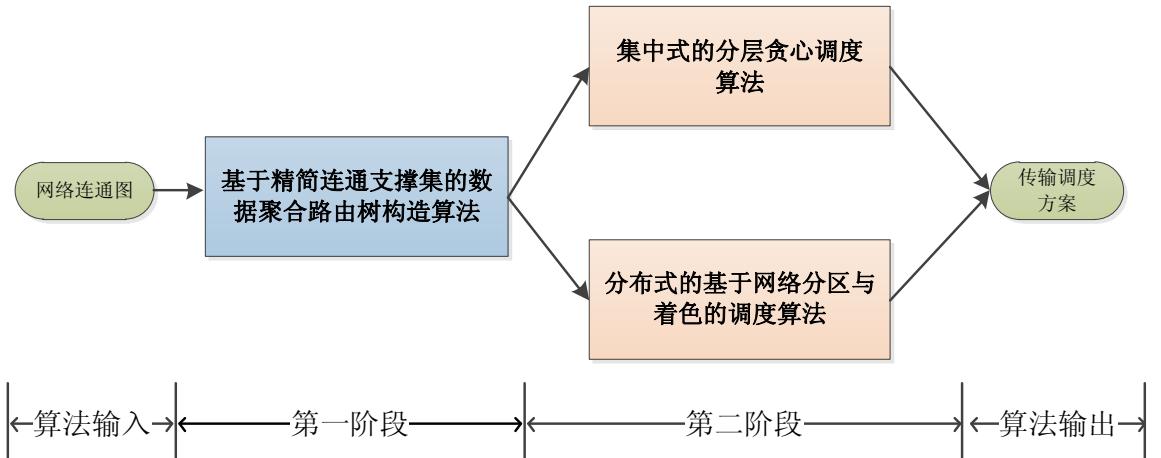


图 2.2 一种包含路由构造和调度设计的两阶段数据聚合方案

#### 2.4.1 基于连通支撑集的数据聚合树构造算法

通过改进文献<sup>[66]</sup>中的工作，我们提出一种新的数据聚合树构造算法，以在图  $G(\mathcal{V}, \mathcal{E})$  中生成树  $T(\mathcal{V}_T, \mathcal{E}_T)$ 。在文献<sup>[66]</sup>中，作者首先从节点集合  $\mathcal{V}$  中选择一个连通支撑集(Connected Dominator Set, CDS)作为网络的骨架(backbone)。CDS 的选择方法包括两个步骤。首先，从  $\mathcal{V}$  中选取一个极大独立集(Maximal Independent Set, MIS)；然后，选择一些连接节点(connectors)，将 MIS 中的节点连接起来，构成连通的子图。我们采用相同的方法构造 MIS，但是在选择连接节点时，我们尽量选择更少的连接节点以构造 CDS。也就是说，我们构造的 CDS 相比文献<sup>[66]</sup>中的 CDS 包含的节点数量更少，更为精简(reduced)，这将使得我们的调度算法的时延更加高效。图 2.3 给出了本文提出的改进型数据聚合树构造算法(Improved Data Aggregation Tree Construction, IDATC)的伪代码。

具体来说，IDATC 算法包括以下主要步骤：

(1) 初始时，图  $G$  中的节点全部被标记为白色。从  $v_s$  开始执行广度优先搜索(Breadth First Search, BFS)，得到一棵以  $v_s$  为根节点的 BFS 树  $T_{bfs}$ 。然后从  $v_s$  开始，按照节点在树中的层次，从高到低地选择独立节点，构成极大独立集  $\mathcal{U}$ 。 $\mathcal{U}$  中的节点称为支配节点(dominators)，并全部标记为黑色。用  $\mathcal{U}_i$  表示树  $T_{bfs}$  中位于第  $i$  层的支配节点集合，这里  $1 \leq i \leq R$ ，其中  $R$  表示图  $G$  的半径。

(2) 接下来，从集合  $\mathcal{V} \setminus \mathcal{U}$  中选择一部分节点，来互连(interconnect)  $\mathcal{U}$  中的支配节点。这部分被选择的节点构成连接节点集合  $\mathcal{W}$ 。具体来说，从第 1 层到第  $R$  层，对于  $\mathcal{U}$  中的每个节点  $u$ ， $u$  在树  $T_{bfs}$  中的父节点  $p(u)$  被选择为连接节点，加入到  $\mathcal{W}$  中，并被标记为灰色。然后从  $p(u)$  的同一层或上一层节点中选择一个支配节点  $d(p(u))$ 。显然，

$p(u)$ 负责连接支配节点 $u$ 和 $d(p(u))$ 。边 $(u, p(u))$ 和边 $(p(u), d(p(u)))$ 被加入到 $\mathcal{E}_T$ 中。

(3) 然后, 从 $\mathcal{W}$ 中移除冗余连接节点。一个连接节点 $w$ 是冗余的(redundant), 当且仅当移除 $w$ 不会使得任何一个 $w$ 在树 $T_{bfs}$ 中的孩子支配节点与 $w$ 的支配节点 $d(w)$ 中断连接。图 2.4 中给出了判定一个连接节点是否冗余的算法, 供 **IDATC** 算法调用。当 $w$ 被判定为冗余并被移除时,  $w$ 的每个孩子支配节点 $c(w)$ 通过另外一个连接节点 $w^*$ 连接到 $d(w)$ 。所有被移除的连接节点被重新标记为白色。剩余的连接节点集合 $\mathcal{W}$ 与支配节点集合 $\mathcal{U}$ 共同构成一个精简的 CDS。

(4) 最后, 随机地将每个白色节点(又称被支配节点, dominatee)连接到一个相邻的支配节点, 从而得到一棵最终的数据聚合树 $T$ 。很容易获得 $T$ 的若干性质, 包括: (i) 每个白色节点的父节点均为黑色节点, (ii) 除了 sink 外的每个黑色节点的父节点均为灰色节点, (iii) 每个灰色节点的父节点均为黑色节点。

---

**Algorithm** :改进型数据聚合树构造算法(**IDATC**)

---

**输入:** 图  $G(\mathcal{V}, \mathcal{E})$

**输出:** 树  $T(\mathcal{V}_T, \mathcal{E}_T)$

```

1: 在图 $G$ 中, 以 $v_s$ 为根节点, 执行宽度优先搜索(BFS), 构造一颗BFS树 $T_{bfs}$ ;
2: 以 $v_s$ 为出发点, 在 $G$ 中一层一层地构造极大独立集(MIS)  $\mathcal{U}$ ;
3:  $\mathcal{V}_T \leftarrow \mathcal{V}$ ,  $\mathcal{E}_T \leftarrow \emptyset$ ;
4: for  $i$ 从1到 $R$  do
5:   for  $\mathcal{U}_i$ 中的每一个支配节点 $u$  do
6:     将节点 $p(u)$ 加入集合 $\mathcal{W}$ , 并将其标记为灰色(GREY);
7:     将 $(u, p(u))$ 和 $(p(u), d(p(u)))$ 加入 $\mathcal{E}_T$ ;
8:   end for
9: end for
10: for  $\mathcal{W}$ 中的每一个连接节点 $w$  do
11:   if IsRedundantConnector( $w$ ) then
12:     将 $w$ 标记为白色(WHITE), 并将边 $(w, d(w))$ 从 $\mathcal{E}_T$ 中移除;
13:     for  $w$ 在树 $T_{bfs}$ 中的每一个孩子支配节点 $c(w)$  do
14:       将 $(c(w), w)$ 从 $\mathcal{E}_T$ 中移除, 并将 $(c(w), w^*)$ 和 $(w^*, d(w))$ 添加到 $\mathcal{E}_T$ ;
15:     end for
16:   end if
17: end for
18: for  $\mathcal{V}$ 中的每一个被支配节点 $v$  do
19:   将 $(v, d(v))$ 中加入 $\mathcal{E}_T$ ;
20: end for
21: return  $T(\mathcal{V}_T, \mathcal{E}_T)$ 

```

---

图 2.3 改进型数据聚合树构造算法(**IDATC**)伪代码示意图

图 2.5 以一个简单的网络示例展示了 **IDATC** 算法的工作流程。具体来说, 图 2.5(a)给出了作为算法输入的网络连通图 $G$ , 图 2.5(b)显示的是从 $G$ 构造而得的一棵 BFS 树

$T_{bfs}$ 。图 2.5(c)给出了所选择的极大独立集 MIS，包括 0、3、4 和 8 号节点。图 2.5(d)给出了所选择的连接节点，这里之所以 1、2 和 5 号节点被选择，是因为它们分别是节点 4、3 和 8 在树  $T_{bfs}$  中的父节点。连接节点 1、2 和 5 分别选择 0、0 和 3 作为各自的支配节点。注意到 1 号连接节点是冗余的，因为 4 号节点可以通过 2 号节点连接到 0 号节点，因此 1 号连接节点被移除，得到图 2.5(e)。最后，白色节点 1、6 和 7 分别随机地连接到一个相邻的黑色节点，得到最终的数据聚合树，如图 2.5(f)所示。

---

**Algorithm :冗余连接节点判定算法(IsRedundantConnector)**


---

输入: 图  $G(\mathcal{V}, \mathcal{E})$ , BFS 树  $T_{bfs}$ , 连接节点  $w$

输出: 如果  $w$  为冗余连接节点, 则返回 TRUE; 否则, 返回 FALSE

```

1: for  $w$  在树  $T_{bfs}$  中的每一个孩子支配节点  $c(w)$  do
2:   if  $(\mathcal{W}_{c(w)} \setminus \{w\}) \cap (\mathcal{W}_{d(w)} \setminus \{w\}) = \emptyset$  then
3:     return FALSE (注: 这里  $\mathcal{W}_i$  表示节点  $i$  的相邻连接节点集合);
4:   end if
5: end for
6: return TRUE

```

---

图 2.4 供 IDATC 算法调用的冗余连接节点判定子算法伪代码示意图

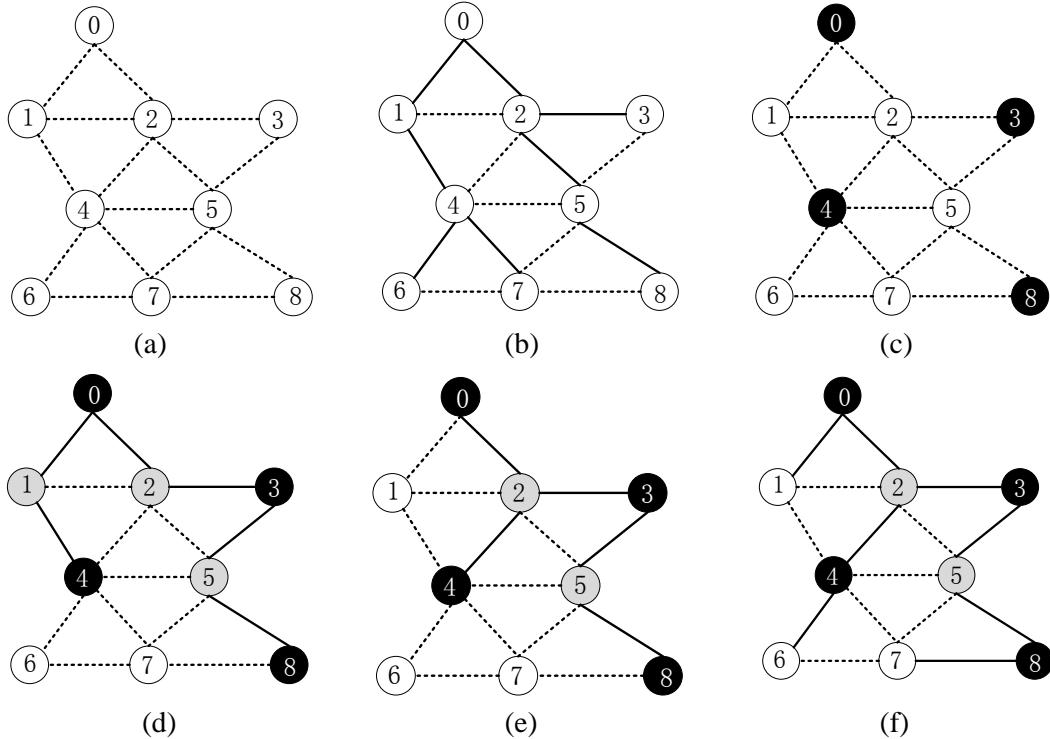


图 2.5 展示 IDATC 算法工作流程的一个简单网络示例

#### 2.4.2 基于贪心极大策略的集中式调度算法

本节提出一种集中式的数据聚合调度算法, 即分层贪心聚合调度算法(Layered

Greedy Aggregation Scheduling, LGAS)。LGAS 算法的工作原理如下：给定数据聚合树  $T$ ，首先调度白色节点，然后自底向上交替调度黑色节点和灰色节点；每个过程均采用相同的贪心调度子程序(Greedy Scheduling Sub-Procedure, GSSP)。GSSP 基于贪心策略，在不破坏各种约束条件的前提下，尽可能地并行调度更多的链路。具体来说，在每个时隙，GSSP 首先随机地从待调度发送节点中选择一个加入到当前时隙已调度集合，然后遍历所有待调度节点，当且仅当某个发送节点与当前时隙已调度集合中的所有节点均不冲突时，才对其进行调度；否则，将其从当前时隙移除，使其等待下一个时隙。图 2.6 详细地给出了算法 LGAS 的工作过程。我们将在 2.4.4 节对 LGAS 的性能进行理论分析时，进一步阐述其工作细节。

注意到 LGAS 是一种集中式的算法。在实际系统中，通常需要经过三个步骤实现 LGAS。第一步，由 sink 节点通过数据收集的方式，获取网络中各个节点的身份和位置信息；第二步，由 sink 节点依据 IDATC 算法构造数据聚合树，并依据 LGAS 算法计算调度方案；第三部，通过数据广播或洪泛(flooding)<sup>[80]</sup>方式，sink 节点将计算得到的调度方案发送给网络中的每个节点。

---

**Algorithm :分层贪心聚合调度算法(LGAS)**


---

**输入：** 数据聚合树  $T(\mathcal{V}_T, \mathcal{E}_T)$ , 图  $G(\mathcal{V}, \mathcal{E})$

- 1: 调用  $\text{GSSP}(\mathcal{V} \setminus (\mathcal{U} \cup \mathcal{W}), G, T)$ ;
- 2: 将  $\mathcal{V} \setminus (\mathcal{U} \cup \mathcal{W})$  中的所有节点从  $T$  中移除，得到骨架树  $T_B$ ;
- 3: **for**  $i$  从  $d_B$  到 1 **do**
- 4:     调用  $\text{GSSP}(B_i, G, T_B)$ ;
- 5: **end for**

▷ 贪心调度子算法(GSSP)

**输入：** 发送节点集合  $\mathcal{A}$ , 图  $G$ , 树  $T$

**输出：** 调度序列  $\mathcal{A}_1, \mathcal{A}_2, \dots$

- 1: 初始化:  $h \leftarrow 0$ ;
  - 2: **while**  $\mathcal{A} \neq \emptyset$  **do**
  - 3:     从  $\mathcal{A}$  中随机选择一条链路，并移动到  $\mathcal{A}_h$  中，同时让  $h \leftarrow h + 1$ ,  $\mathcal{D} \leftarrow \emptyset$ ;
  - 4:     **for** 集合  $\mathcal{A} \setminus \mathcal{D}$  中的每个发送节点  $v$  **do**
  - 5:         **if**  $\forall u \in \mathcal{A}_h$ , 满足  $d(v, p(u)) > \rho \cdot r$  且  $d(u, p(v)) > \rho \cdot r$  **then**
  - 6:             将节点  $v$  从  $\mathcal{A}$  移动到  $\mathcal{A}_h$  中;
  - 7:         **else**
  - 8:             将节点  $v$  从  $\mathcal{A}$  移动到  $\mathcal{D}$  中;
  - 9:         **end if**
  - 10:     **end for**
  - 11: **end while**
- 

图 2.6 分层贪心聚合调度算法(LGAS)伪代码示意图

### 2.4.3 基于网络分区与着色策略的分布式调度算法

注意到 LGAS 是一种集中式算法。当网络动态性加剧，如因节点频繁加入/失效

导致网络拓扑变动时，需要重新计算调度方案，这将带来巨大的额外开销。为了适应动态传感器网络的需要，本节提出一种基于网络分区与着色技术的调度算法(Network Partition and Coloring based Scheduling, NPCS)。**NPCS** 算法是分布式的，每个节点只需要局部信息即可以自治地计算各自的调度方案。

在 **NPCS** 算法中，我们首先对网络进行分区与着色。具体来说，采用一族垂直线  $x = i \cdot a, i \in \mathbb{Z}$  和一族水平线  $y = j \cdot a, j \in \mathbb{Z}$  将二维平面划分为半开半闭、大小为  $a \times a$  的正方形小区。接下来，我们用  $k^2$  种不同的颜色对这些小区进行着色。使用的颜色用序列数  $0, 1, \dots, k^2 - 1$  进行编号。节点  $u$  所在小区的颜色表示为  $Color[u]$ 。图 2.7 给出了一种分区与着色方案的示意图，该图中采用了 9 种颜色对小区进行着色。

在上述网络分区与着色方法的基础上，**NPCS** 采用如下基本思想进行调度：在某个时隙，任一小区最多只允许一个节点发送或者接收数据包；并且，只允许处于相同颜色小区的节点同时进行数据包发送和接收。与 **LGAS** 算法一样，**NPCS** 算法也是先调度被支配节点，然后交替调度支配节点和连接节点。

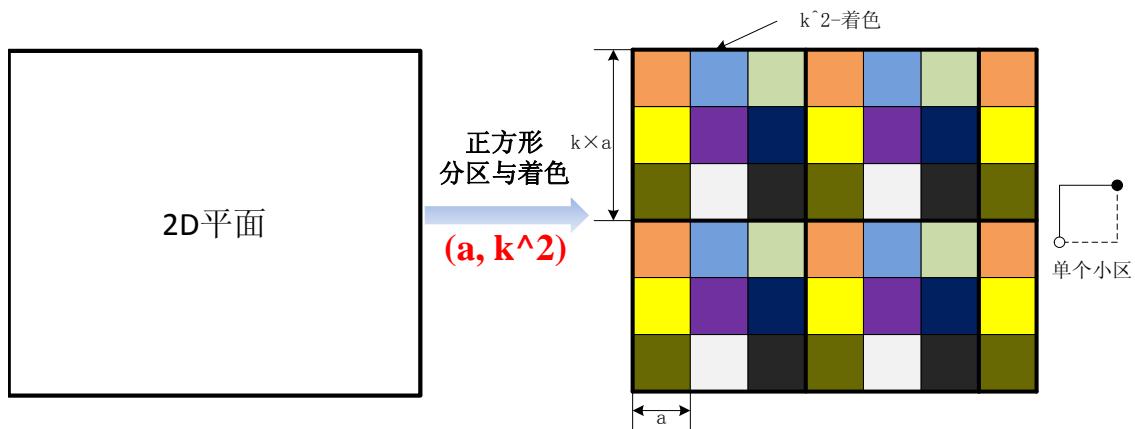


图 2.7 采用正方形进行二维平面分区与着色示意图，该图中  $k = 3$

图 2.8 给出了 **NPCS** 算法的伪代码。由图可知，算法包括三个阶段：初始化阶段，消息传递阶段和调度计算阶段。在 **NPCS** 算法中，假设节点具有自定位能力，这可以通过装备 GPS 定位模块或采用网络定位算法<sup>[23]</sup>来实现。依据自身位置信息，节点  $u$  可以获取所在小区的颜色  $Color[u]$ 。另外，假设节点  $u$  已知其在树中的层数，以及孩子支配节点数量  $N_{DoC}(u)$ 、孩子连接节点数量  $N_{CoC}(u)$  和孩子被支配节点数量  $N_{DeC}(u)$ 。为了确保 **NPCS** 算法产生的调度方案可行，小区大小  $a$  和颜色数  $k^2$  需满足： $a \leq r/\sqrt{2}$ ， $k \geq \lceil (\rho + 1) \cdot \frac{r}{a} \rceil + 1$ (见定理 2.7)。为了降低 **NPCS** 的时延，我们取  $a$  的上界和  $k$  的下界，即  $a = r/\sqrt{2}$ ,  $k = \lceil (\rho + 1) \cdot \frac{r}{a} \rceil + 1$ 。

#### 2.4.4 算法性能分析

在本节，我们首先分析算法 **IDATC** 所构造的数据聚合树  $T$  的性质，然后从理论

角度论证 **LGAS** 和 **NPCS** 的可行性并推导其时延性能上界。

---

**Algorithm :**基于网络分区与着色的调度算法(**NPCS**)

---

**输入:** 图 $G$ , 数据聚合树 $T$

**输出:** 每个节点 $u$ 的调度(传输时隙序号):  $Sche(u)$

```

/*初始化阶段*/
1: 初始化:  $a = r/\sqrt{2}$ ,  $k = \lceil (\rho + 1) \cdot \frac{r}{a} \rceil + 1$ ;
2: 支配节点根据预设或定位技术获得自己在二维平面中的位置坐标 $(x, y)$ ;
3: 支配节点计算自己所在小区的颜色序号:  $Color[\cdot] = \lfloor \frac{x\%ka}{a} \rfloor + \lfloor \frac{y\%ka}{a} \rfloor \times (k - 1)$ ;
/*消息传递阶段*/
4: 对于支配节点 $u$ : (1)将 $Color[u]$ 和集合 $\{1, 2, \dots, N_{DeC}(u)\}$ 中的某一个指定数发送给 $T$ 中的每个孩子被支配节点, (2)将 $Color[u]$ 和集合 $\{1, 2, \dots, N_{CoC}(u)\}$ 中的某一个指定数发送给 $T$ 中的每一个孩子连接节点;
/*调度计算阶段*/
5: 对于被支配节点 $v$ : 设置 $Sche(v) = Color[p(v)] + (N - 1) \times k^2$ ;
6: 对于支配节点 $u$ : 设置 $Sche(u) = Color[u] + \frac{d_B - l_B(v)}{2} \times 13k^2 + (\Delta - 1) \times k^2$ ;
7: 对于连接节点 $w$ : 设置 $Sche(w) = Color[p(w)] + (N - 1) \times k^2 + \frac{d_B - l_B(v) - 1}{2} \times 13k^2 + (\Delta - 1) \times k^2$ 

```

---

图 2.8 基于网络分区与着色的调度算法(**NPCS**)伪代码示意图

**引理 2.1:** 让 $T_B$ 表示树 $T$ 中只包含黑色节点和灰色节点的子树。设 $d_B$ 和 $R$ 分别表示树 $T_B$ 的深度和图 $G$ 的半径, 则我们有:  $d_B \leq 2(R - 1)$ 。

证明. 由于 sink 节点 $v_s$ 是支配节点, 因此其在树 $T_B$ 中的所有邻居节点均为连接节点。故在树 $T_B$ 中, 至少有一个支配节点的层的数量不超过 $R$ 。另外, 对于每两个相邻的支配节点所在的层, 存在一连接节点。故在树 $T_B$ 中, 至少有一个连接节点的层的数量不超过 $R - 1$ 。综上, 树 $T_B$ 总的层数不超过 $2R - 1$ 。也就是说, 树的深度 $d_B$ 不超过 $2R - 2$ , 即 $d_B \leq 2(R - 1)$ 。引理证毕。  $\square$

**引理 2.2:** 在图 $G$ 的连通支撑集中, 每个连接节点最多与 5 个支配节点相邻<sup>[46]</sup>。

**引理 2.3:** 设 $u$ 、 $v$ 和 $w$ 分别表示图 $G$ 的连通支撑集中的三个支配节点, 且 $v$ 和 $w$ 是 $u$ 的两跳邻居节点。让 $v^*$ 和 $w^*$ 分别表示连接 $v$ 和 $w$ 与 $u$ 的连接节点。如果 $\widehat{vuw} \leq 2 \arcsin \frac{1}{4}$ , 则 $v^*$ 与 $w$ 相邻或者 $w^*$ 与 $v$ 相邻<sup>[46]</sup>。

**引理 2.4:** 让 $T_B$ 表示树 $T$ 中只包含黑色节点和灰色节点的子树,  $d_B$ 表示树 $T_B$ 的深度。对于树 $T_B$ 的第 $i - 1$ 层中的任一节点 $u$ ,  $u$ 位于第 $i$ 层的邻居节点的个数 $\Delta_i(u)$ 满足

$$\Delta_i(u) \leq \begin{cases} 4 & i \text{ 为偶数} \\ 11 & i \text{ 为奇数, 且 } i > 1 \\ 12 & i = 1 \end{cases} \quad \begin{array}{l} (1a) \\ (1b) \\ (1c) \end{array}$$

证明. 由引理 2.2 可知, 每个连接节点最多与 5 个支配节点相邻。注意到对于每个连接节点, 存在一个相邻且位于上一层的支配节点充当其父节点, 因此, 与该连接

节点相邻且处于下一层的支配节点最多为 4 个。当  $i$  是偶数时, 第  $i - 1$  层和第  $i$  层分别为连接节点和支配节点。因此, 当  $i$  为偶数时, 有  $\Delta_i(u) \leq 4$ 。另外, 通过引理 2.3, 我们可以证明每个支配节点最多与 12 个连接节点相邻。我们采用反证法来验证这一判断。假设存在一个支配节点  $u$ , 其与  $n$  个连接节点相邻, 这里  $n \geq 13$ 。对于每个连接节点, 由算法 **IDATC** 可知, 其之所以没有被移除, 是由于至少存在一个支配节点只能通过该节点与  $u$  连接。也就是说,  $u$  的两跳范围内, 存在  $n$  个支配节点。根据抽屉原理, 在该范围内, 存在两个支配节点  $v$  和  $w$ , 满足  $\widehat{vw} \leq \frac{2\pi}{n} < 2 \arcsin \frac{1}{4}$ 。根据引理 2.3 可知,  $v^*$  与  $w$  相邻或者  $w^*$  与  $v$  相邻, 这里  $v^*$  与  $w^*$  分别是  $v$  与  $w$  唯一的连接节点。这里产生了矛盾, 因此  $n < 13$ , 即每个支配节点最多与 12 个连接节点相邻。当  $i$  为奇数时, 层  $i - 1$  中的每一个支配节点  $u$  至多与层  $i$  中的 12 个连接节点相邻, 即  $\Delta_i(u) \leq 12$ 。另外, 当为  $i$  奇数且  $i > 1$  时, 节点  $u$  必然有一个父亲连接节点在其上一层, 此时  $\Delta_i(u) \leq 11$ 。引理证毕。  $\square$

**定理 2.5:** 算法 **LGAS** 产生的调度方案是可行(feasible)的。

证明. 为了证明 **LGAS** 算法产生的调度是可行的, 我们需要证明在每个时隙所调度的发送节点两两之间均互不冲突。在子算法 **GSSP** 中, 在两种情形下会进行调度。第一种情形如 **GSSP** 第 3 行所示, 我们随机选择一个节点调度。此时, 其为唯一调度节点, 故必然成功; 第二种情形如 **GSSP** 第 6 行所示, 我们将满足第 5 行所示条件的节点加入当前调度时隙。该条件依据协议干扰模型的定义, 能够判定节点之间是否存在冲突。综上, **LGAS** 算法产生的调度是可行的。定理证毕。  $\square$

**定理 2.6:** 算法 **LGAS** 产生的调度方案在最坏情形(worst-case)下的时延不超过  $\lfloor \frac{\pi}{\sqrt{3}}(\rho + 1)^2 + (\frac{\pi}{2} + 1)(\rho + 1) + 1 \rfloor \cdot (15R + \Delta - 15)$ , 其中  $\rho = r_I/r \geq 1$  为干扰比,  $R$  和  $\Delta$  分别为图  $G$  的半径和最大节点度。

证明. 算法 **LGAS** 的时延由三部分组成, 分别为被支配节点的发送时延, 支配节点的发送时延和连接节点的发送时延。首先讨论第一部分时延。当两个被支配节点分别向各自的支配节点发送数据时, 在协议干扰模型下, 如果对应支配节点之间的距离大于  $(\rho + 1)r$ , 则两条传输链路必能共存。因此, 对于任何一个支配节点, 其半径为  $(\rho + 1)r$  的圆盘内的其他支配节点是潜在的无法同时调度的冲突节点。注意到任意两个支配节点之间的距离大于 1, 根据 Wegner Theorem<sup>[56]</sup>, 上述范围内的支配节点个数不超过  $\lambda(\rho)$  个, 这里  $\lambda(\rho) = \lfloor \frac{\pi}{\sqrt{3}}(\rho + 1)^2 + (\frac{\pi}{2} + 1)(\rho + 1) + 1 \rfloor$ 。另外, 注意到每个支配节点最多与  $\Delta - 1$  个被支配节点相邻, 而其每次只能接收一个被支配节点的数据。综上, 第一部分时延在极端情况下为  $\lambda(\rho) \cdot (\Delta - 1)$  个时隙。同理, 我们可以推导出第二和第三部分的最坏时延, 分别为  $\lambda(\rho) \cdot 4(R - 1)$  和  $\lambda(\rho) \cdot (11R - 10)$  个时隙。综上,

**LGAS** 在最坏情形下的时延不超过 $\lambda(\rho) \cdot (15R + \Delta - 15)$ 个时隙。证毕。  $\square$

**定理 2.7:** 算法 **NPCS** 产生的调度方案是可行(feasible)的。

证明. 算法 **NPCS** 的可行性, 是由参数 $a$ 和 $k$ 的选择来决定的。通过将 $a$ 设置为 $r/\sqrt{2}$ , 确保了每个小区至多只包含一个支配节点; 通过将 $k$ 设为 $\lceil (\rho + 1) \cdot \frac{r}{a} \rceil + 1$ , 确保了任意两个所在小区具有相同颜色的支配节点之间的距离大于 $(\rho + 1)r$ 。这进一步确保了具有相同颜色小区内的支配节点同时接收或者发送数据时, 必然不会产生冲突。定理证毕。  $\square$

**定理 2.8:** 算法 **NPCS** 产生的调度方案在最坏情形下的时延不超过 $(\lceil \sqrt{2}(\rho + 1) \rceil + 1)^2 \cdot (16R + \Delta - 17)$ 。其中 $\rho = r_I/r \geq 1$ 为干扰比,  $R$ 和 $\Delta$ 分别为图 $G$ 的半径和最大节点度。

证明. 由于具有相同颜色小区内的支配节点同时接收或者发送数据时, 必然不会产生冲突。那么经过 $k^2$ 个时隙, 全部的支配节点均可至少成功调度一次。采用类似于**定理 2.6** 证明中的方法, 我们可以依次分析被支配节点, 支配节点和连接节点最坏情形下的发送时延。为了节省篇幅, 这里省略详细的计算步骤。最终, 我们可以计算出 **NPCS** 算法的最坏时延不超过 $(\lceil \sqrt{2}(\rho + 1) \rceil + 1)^2 \cdot (16R + \Delta - 17)$ 。定理证毕。  $\square$

注意到两种调度算法 **LGAS** 和 **NPCS** 在最坏情形下的时延不超过 $\mathcal{O}(R + \Delta)$ 这个上界。而对于 **MLDAS** 问题来说,  $\max(R, \Delta)$ 是其的一个下界<sup>[46]</sup>。因此, 两种算法均是具有常量近似比 $\mathcal{O}(1)$ 的近似算法。也就是说, 两种算法在极端情形下的时延性能从理论上讲是有较好保证的。

## 2.5 仿真实验

### 2.5.1 实验场景与参数设置

在本节, 我们通过 MATLAB 仿真实验对 **LGAS** 和 **NPCS** 的性能进行评价。

表 2.1 仿真参数设置

参数	值
网络监测区域大小	150m × 150m
Sink 节点位置	随机
节点数量	200~1000
节点传输半径	15m~40m
干扰比 $\rho$	1~5
单个时隙长度	1
每次实验重复次数	50

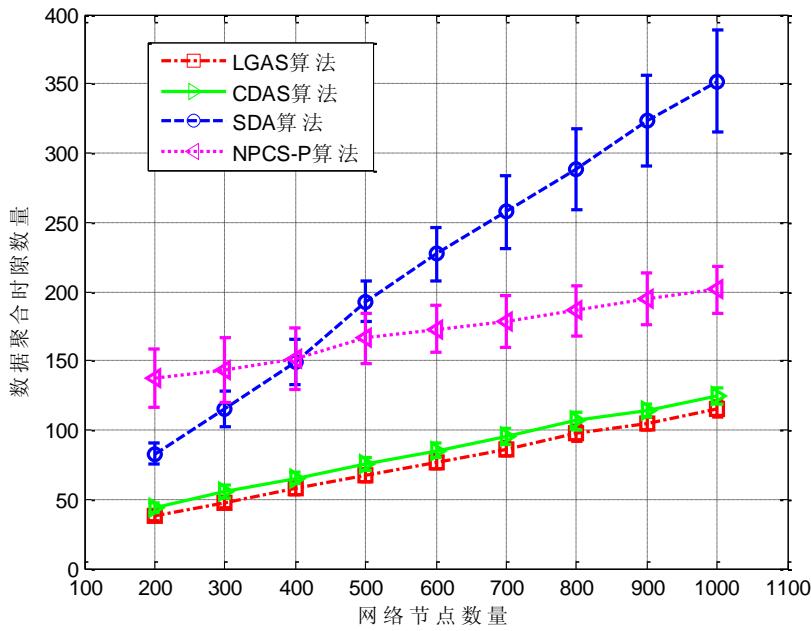
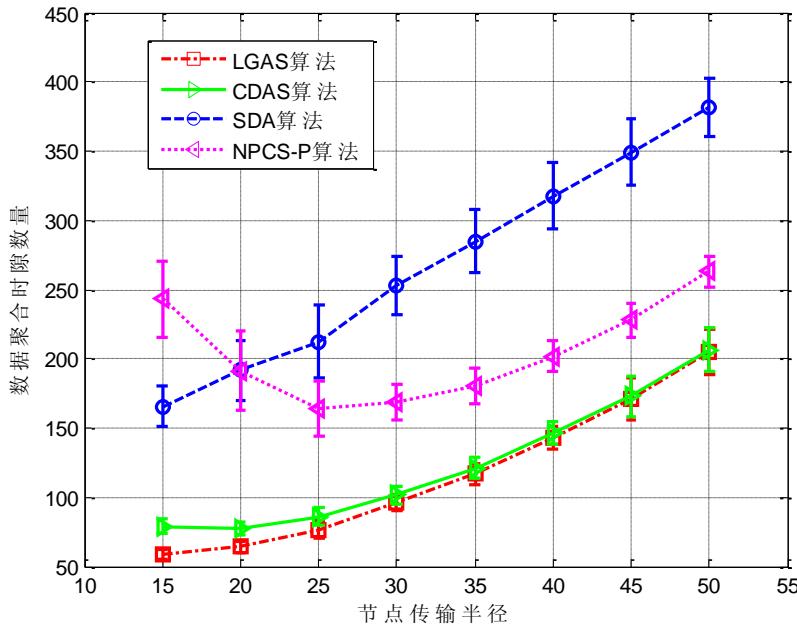
在我们的仿真环境中, 网络监测区域的范围被固定为 $150m \times 150m$ , 并且监测中心, 即 sink 节点随机地布设在该区域之中。网络中的节点数量(包括 sink 节点)在 200

到 1000 的范围内变化。节点的传输范围为以自己为中心的半径为  $r$  的圆盘，这里  $r$  的范围为 15m 到 40m。发送节点的干扰半径为传输半径的  $\rho$  倍，这里  $\rho$  的范围为 1~5。网络中的节点保持时间同步。我们将单个时隙的长度归一化为 1，这样数据聚合总的时延就是数据聚合所消耗的时隙数量。由于网络的几何拓扑会影响到调度算法 **LGAS** 和 **NPCS** 的性能，因此对于每次实验，我们重复进行 50 次，并取 50 次实验的平均结果作为最终实验结果。表 2.1 总结了仿真实验中主要的参数及其取值(或取值范围)。

### 2.5.2 实验结果与分析

我们首先研究网络节点数量对数据聚合时延的影响。我们与文献中的两种代表性数据聚合调度算法进行性能对比。这两种算法分别为 **SDA**<sup>[12]</sup> 和 **CDAS**<sup>[56]</sup>。由于这两种算法只适用于单位圆盘图干扰模型，即干扰比  $\rho = 1$  的情形。因此，我们暂时固定  $\rho = 1$ 。另外，我们将节点传输半径固定为 25m。图 2.9 给出了四种聚合调度算法 **LGAS**、**NPCS**、**SDA** 和 **CDAS** 在不同网络规模下，所获得的调度方案的时延对比示意图。这里我们同时画出了数据聚合的时延平均值和时延标准差。我们可以发现，当节点数量增加时，所有算法的时延均单调增长。这是由于节点数量的增加导致传输请求量的增加，对于每种算法，均需要花费更多时隙来调度这些传输请求。当网络节点数量从 200 变化到 1000 时，**LGAS** 的时延性能比其他三种更好。具体来说，与文献中最好的算法 **CDAS** 相比，**LGAS** 将数据聚合的平均时延降低了 8%(当  $n = 1000$  时)到 14%(当  $n = 200$  时)。在这四种算法中，**SDA** 和 **NPCS** 的时延性能相对来说要差一些。注意到由于 **NPCS** 算法为了实现分布式操作，采用了较为保守的调度策略，导致调度方案中有很多的空闲时隙。为此我们采用去除了空闲时隙的调度方案，表示为 **NPCS-P**(P 代表 pruned，表示经过裁剪的)进行性能对比。观察图 2.9，当网络规模较小时，**NPCS-P** 的时延比 **SDA** 要大；然而当规模较大( $n > 400$ )时，**SDA** 的时延明显大于 **NPCS-P**。

接下来，我们研究节点传输半径对数据聚合时延的影响。同上，我们与 **SDA** 和 **CDAS** 进行性能对比，并取  $\rho = 1$ 。另外，我们将网络节点数量固定为 600。图 2.10 给出了当传输半径从 15 增加到 50 时，**LGAS**、**SDA**、**CDAS** 和 **NPCS-P** 的时延变化示意图。注意到，当传输半径增加时，**LGAS**、**SDA**、**CDAS** 和 **NPCS-P** 的时延均单调增长。这是由于，当传输半径增加时，节点的度会增大，也就是说更多的链路将拥挤在一起，这将阻止链路的并发传输。当传输半径大于 35 时，**LGAS** 和 **CDAS** 的时延相差无几，且均比另外两种算法要低。而当传输半径在 15 到 30 之间时，**LGAS** 的时延比 **CDAS** 要低。尤其是当传输半径为 15 时，**LGAS** 的时延只有 **CDAS** 时延的 74% 左右。我们也可以发现，在大多数情况下，**SDA** 的时延性能是所有算法中最差的。

图 2.9 节点数量变化时，调度算法时延性能对比( $\rho = 1$ )图 2.10 节点传输半径变化时，调度算法时延性能对比( $\rho = 1$ )

在上述实验中，我们均将干扰比固定为  $\rho = 1$ ，也就是说，我们只考虑节点的干扰范围与传输范围相同的情形。接下来，我们将研究干扰比  $\rho (\rho \geq 1)$  对数据聚合时延的影响。由于 **SDA** 和 **CDAS** 无法适用于  $\rho > 1$  的情形，因此我们不与其进行性能对比。取而代之的是，我们与 **LGAS** 的理论上界(表示为 **LGAS-UB**)和 **NPCS** 的理论上界(表示为 **NPCS-UB**)进行对比。这有利于衡量 **LGAS** 和 **NPCS** 的实际性能好坏。将节点传输半径固定为 15，节点数量固定为 2000。图 2.11 给出了当干扰比  $\rho$  从 1 增大到 5

时, **LGAS**、**NPCS-P**、**LGAS-UB** 和 **NPCS-UB** 的时延变化趋势图。我们可以发现, 当干扰比  $\rho$  增大时, 所有四种算法的时延均快速增加。这是由于, 当干扰范围增大时, 更多的链路无法在同一时隙传输, 因而需要更多的时隙来调度这些链路。我们还可以发现无论  $\rho$  如何变化, **LGAS** 和 **NPCS-P** 均没有超过各自的理论界, 这说明我们推导出来的理论界是有效的。同时, **LGAS** 和 **NPCS-P** 的时延显著低于各自的理论界, 这说明调度算法的实际性能较理论推导的性能界要好。

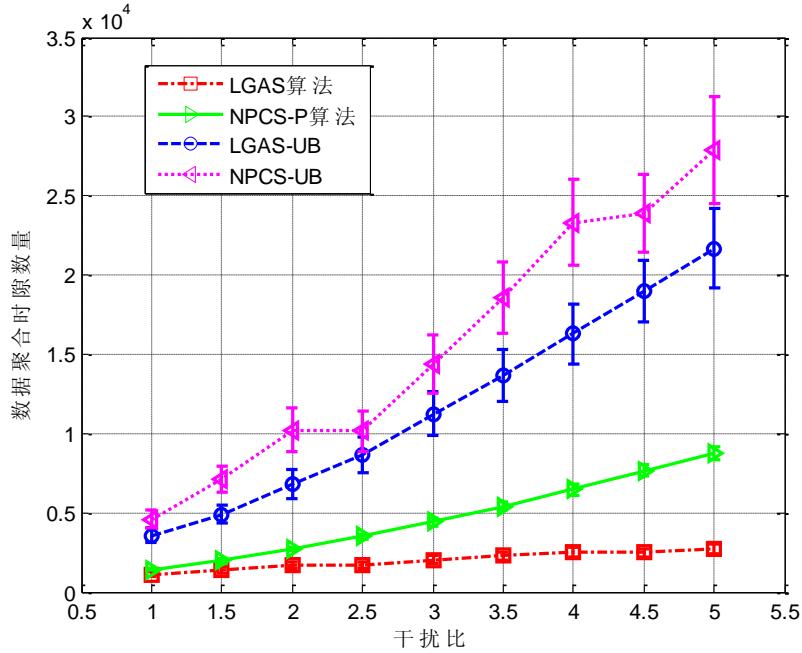


图 2.11 干扰比  $\rho \geq 1$  时, 调度算法时延性能与理论界对比

在上述实验中, sink 节点的位置是随机分布的。自然, 我们希望知道 sink 节点的位置会对数据聚合时延产生何种影响。我们考虑三种布设 sink 节点的策略, 即随机布设(random deployment, RD), 中心布设(center deployment, CT)和边缘布设(corner deployment, CN)。在 RD 中, sink 节点的位置在监测区域中随机分布; 在 CT 中, sink 节点固定在监测区域中心; 在 CN 中, sink 节点固定在监测区域中的某一个角落。图 2.12 给出调度算法 **LGAS** 和 **NPCS-P** 在三种 sink 节点布设策略下的数据聚合时延随网络节点数量变化的示意图。这里我们将节点传输半径固定为  $15m$ , 将干扰比  $\rho$  固定为 1。我们可以发现, 在这三种策略中, 以 CT 方式布设时, 调度算法的时延最低。这是由于当 sink 布设在区域中心时, 网络的跳数半径较小, 数据传输到 sink 的路径更短, 因而聚合时延较低。而当 sink 位于区域边缘时, 跳数半径较大, 因而数据聚合时延也就较高。

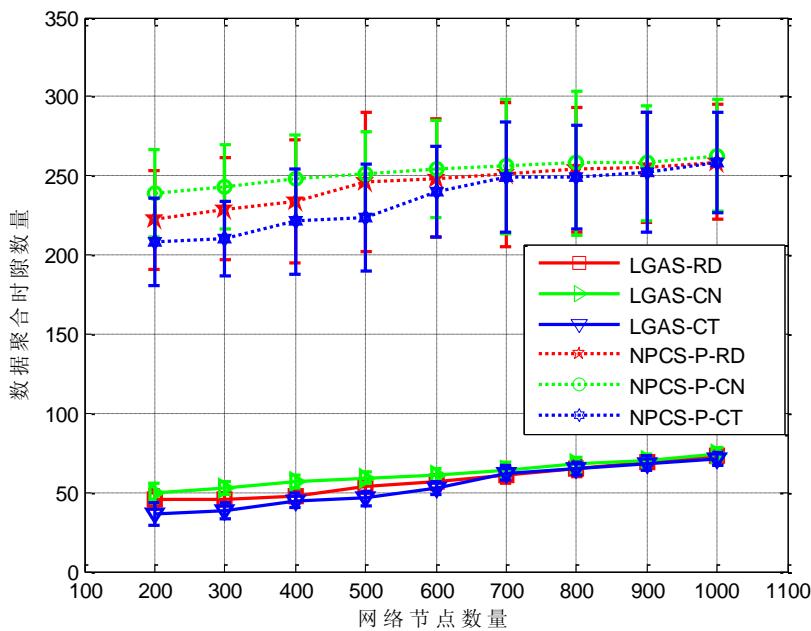


图 2.12 sink 节点位置对调度算法时延性能的影响

## 2.6 本章小结

本章研究了协议干扰模型下的无线传感器网络数据聚合调度问题，目的是最小化网络级的数据聚合时延。我们对该问题进行了建模，并说明其为 **NP**-难的。针对该问题，我们提出了一种两阶段的数据聚合方案。在第一阶段，我们提出了一种基于精简连通支撑集的数据聚合路由树构造算法 **IDATC**。在第二阶段，我们分别提出了一种基于分层贪心策略的聚合调度算法 **LGAS** 和基于网络分区与着色策略的聚合调度算法 **NPCS**。通过理论分析，我们证明了所提出的调度方案的可行性，并推导了 **LGAS** 和 **NPCS** 算法在最坏情形下的聚合时延。分析结果表明两种算法均是具有常量近似比  $\mathcal{O}(1)$  的近似算法。最后我们通过大量的仿真实验评价了上述调度方案的平均性能。实验结果表明，调度算法的实际性能远远好于理论上界，并且与文献中的代表性算法相比，能够在一定程度上降低数据聚合的总时延。

## 第3章 物理干扰模型下的无线传感器网络分布式数据聚合调度

### 3.1 引言

与经典的链路调度<sup>[67, 68]</sup>问题一样，无线传感器网络中的数据聚合调度也面临着一个突出的挑战，那就是多用户干扰。干扰制约了网络在每个时隙的容量的提升，从而也制约了网络总时延的改善。实际系统中，干扰的情况极为复杂多变和难以描述。为了刻画物理层的干扰对网络层的协议和算法性能的影响，研究者对无线干扰进行了抽象，由此产生了各种理论干扰模型。文献中的众多干扰模型，包括单位圆盘图模型、802.11 模型<sup>[81]</sup>和本论文上一章采用的协议干扰模型，本质上均属于基于图的干扰模型 (graph-based models)<sup>[61]</sup>。这类模型之所以得到广泛的关注，是由于其简洁性非常利于从数学上进行描述，并基于图论的理论知识进行复杂的协议和算法设计。这类图干扰模型将干扰的影响限定在一个特定范围，并且将干扰关系固定为二元关系，即任意两条链路之间是否干扰与其他链路无关。

事实上，越来越多的研究发现，上述图干扰模型存在局限性。这种局限性体现在两方面。首先，在实际系统中，干扰强度与信息强度一样，是随着距离衰减的。也就是说，尽管超出一定距离之外的干扰强度会很小，但是并不能完全忽略掉。其次，在实际系统中，干扰具有累积效应，即节点受到的干扰是来自当前所有活跃干扰节点的干扰信号的叠加。此时，干扰不再是一对一的关系。基于上述观察，研究者提出了一种更为实际、更为精确的干扰模型，即物理干扰模型(physical interference model)<sup>[76, 77]</sup>。在该模型下，我们约定，对于每条待传输的链路，当且仅当接收方处的 SINR 大于或等于一个硬件依赖的(hardware-dependent)解码门限值时，链路的传输才能成功。这里 SINR，即 Signal-to-Interference-plus-Noise-Ratio，表示信号强度与干扰强度和噪声强度之和的比值。

当采用物理干扰模型时，数据聚合调度问题将变得更为复杂和难处理。这是因为，对于任何一条链路，即使是来自距离很远的链路的干扰也不能完全被忽略；而且，即使两条链路原本能够共存，但是由于另一条链路的加入，它们的干扰关系可能发生改变，从而变得无法共存。针对物理干扰模型下的数据聚合调度，文献中已经有一些初步的研究<sup>[44, 45, 49, 54, 55, 57, 82]</sup>。但是，文献中提出的算法一般都是集中式的，存在其应用的局限性。通常来说，集中式算法只适用于具有静态拓扑的无线传感器网络。当网络拓扑动态变化，例如新节点随机加入、旧节点失效时，集中式算法的效率将变得很低，迫切需要设计分布式的数据聚合调度方案。

针对物理干扰模型下的无线传感器网络数据聚合问题，本文提出了一种分布式的

数据聚合解决方案 **DDA**。在该方案中，我们首先基于迭代分区策略，提出了一种分布式数据聚合树构造算法 **DDATC**。然后，基于迭代分区与着色策略，提出了一种分布式链路调度算法 **DLS**。在物理层，我们采用两种可选的传输功率控制策略，即均匀功率分配 **PA-U** 和线性功率分配 **PA-L**。通过合理配置功率控制和链路调度模块的参数，我们从理论上证明了 **DDA** 方案在物理干扰模型下的可行性。同时，我们还推导了 **DDA** 在最坏情况下的聚合时延。分析结果表明，**DDA** 的时延上界与网络规模无关，从而显示 **DDA** 具有良好的扩展性。最后，我们通过仿真实验评价了 **DDA** 方案在不同功率控制策略和不同网络参数配置下的时延、能耗等性能。

### 3.2 相关工作

如何对物理层无线干扰进行建模，是无线网络协议设计，尤其是调度/MAC 协议设计的基础性问题。传统的干扰建模方法大多是基于冲突图的，例如主干扰模型、协议干扰模型、802.11 干扰模型等。近年来，一种更加精确、更加实际，但也更加复杂的干扰模型，即物理干扰模型逐渐引起人们的重视。基于物理干扰模型的网络层算法与协议设计成为无线网络学术界前沿的研究方向。文献<sup>[61]</sup>研究了各种干扰模型对无线网络性能的影响。作者发现，无论是在调度接入还是在随机接入网络中，不同的干扰模型会导致显著不同的网络性能。并且，物理干扰模型相对来说更为客观地刻画了无线干扰的实际影响。文献<sup>[76]</sup>研究了物理干扰模型中的网络容量问题，并且提出了一种近似比独立于网络拓扑的调度算法。同时，作者为物理干扰模型下的最小长度链路调度问题设计了近似比为  $\mathcal{O}(\log n)$  的调度算法。这里  $n$  为网络中待调度的链路数量。针对物理干扰模型下的网络容量最大化问题，文献<sup>[83]</sup>基于博奕论提出了分布式的、且具有理论性能保证的链路调度算法。为了改善物理干扰模型下无线节点的信道利用率和能量有效性，文献<sup>[84]</sup>设计了一种功率控制型 MAC 协议。在物理干扰模型下，除了容量、调度和 MAC 问题，无线网络中其他的基础性问题，例如广播<sup>[71]</sup>、信标 (beaconing)<sup>[85, 86]</sup>和数据收集<sup>[87, 88]</sup>等，也得到了研究者的广泛关注。

当采用物理干扰模型来建模无线干扰时，无线传感器网络中的数据聚合调度问题将更具有挑战性。传统的针对协议模型设计的聚合调度算法不再适用。为此，研究者提出了若干新的数据聚合调度算法。文献<sup>[57]</sup>基于连通支撑集，并考虑节点采用均匀发射功率的情形，设计了两种最坏时延为  $\mathcal{O}(R + \Delta)$  的聚合调度算法，这里  $R$  和  $\Delta$  分别为网络的跳数半径和最大节点度。注意到文献<sup>[57]</sup>中的调度方法只适用于均匀随机网络。针对任意拓扑网络，文献<sup>[49]</sup>基于最近邻居节点优先准则构造数据聚合树，并在此基础上提出了两种消耗的时隙数量分别为  $\mathcal{O}(\log n)$  和  $\mathcal{O}(\log^3 n)$  的集中式调度算法。这里  $n$  为网络中的节点数量。此后，针对物理干扰模型下的聚合调度问题，并考虑功率控制，文献<sup>[44]</sup>提出了一种最坏时延为  $\mathcal{O}(\log n + R)$  的调度算法，并证明该调度算法

的近似比为常数。当数据聚合树作为调度算法的输入时，文献<sup>[55]</sup>证明了此时的数据聚合调度问题仍然为 **NP**-难的。并且，作者经过论证发现，上述几种调度算法在该模型下的时延最坏情况下将高达最优时延的  $\sqrt{n}$  倍。

### 3.3 网络模型与问题描述

#### 3.3.1 网络模型

考虑二维平面上的无线传感器网络。集合  $\mathcal{V}$  中的节点随机地布设在大小为  $L \times L$  的监测区域中，其中  $v_s \in \mathcal{V}$  为汇聚节点， $\mathcal{V} \setminus \{v_s\}$  为传感器节点集合。汇聚节点负责管理传感器节点，并接收来自传感器节点的聚合数据。所有节点均配备有一根全向天线，可以从(向)各个方向接收(发送)数据。节点工作在同一共享信道。由于半双工的约束，节点不能在任一时刻同时进行数据的发送和接收。对于任意两个节点  $u$  和  $v$ ，我们用  $d(u, v)$  表示它们之间的欧式距离。由于实际物理条件的约束，节点之间的距离不可能无限小。为此，与文献<sup>[89]</sup>类似，我们假设任意两个节点之间的距离不小于 1。注意到在监测区域  $L \times L$  中，节点最大距离为  $\sqrt{2}L$ ，因此我们有： $1 \leq d(u, v) \leq \sqrt{2}L$ 。另外，我们假设网络中的节点满足时间同步。

#### 3.3.2 物理干扰模型

任给节点  $v \in \mathcal{V}$ ，我们用  $Q_v$  表示其发射功率。当节点  $v$  发送时，另一节点  $u \in \mathcal{V}$  所接收到的来自  $v$  的信号强度为：

$$Q(u, v) = \frac{Q_v}{d^\alpha(u, v)} \quad (3.3.1)$$

注意到，如果  $u$  是  $v$  的目的接收节点，则  $Q(u, v)$  表示目的接收信号强度。否则， $Q(u, v)$  表示来自节点  $v$  的干扰信号强度。式(3.3.1)中， $\alpha \in (2, 6]$  是路径损耗因子(path loss exponent)。由式(3.3.1)可知，信号强度是随着距离衰减的，这符合无线信道的传播特性。图 3.1 形象地给出了物理干扰模型衰减特性的示意图。

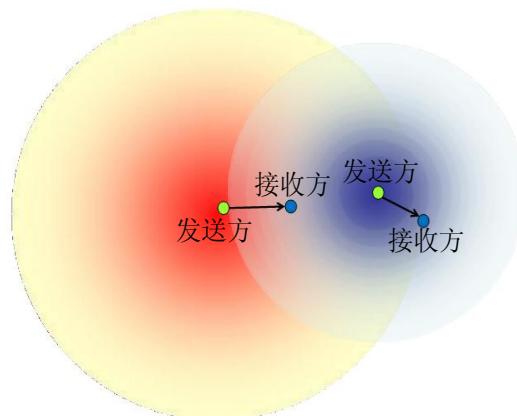


图 3.1 无线网络中的物理干扰模型示意图

在物理干扰模型下，如果 $u$ 是 $v$ 的目的接收节点，则 $u$ 能够成功接收(解码)来自 $v$ 的数据包(信号)的充分必要条件是：节点 $u$ 处的 SINR 值(即信号干扰噪声比)大于或等于一个解码门限<sup>[77]</sup>，即：

$$SINR(u, v, \mathcal{S}_t) = \frac{Q(u, v)}{\xi + \sum_{w \in \mathcal{S}_t} Q(u, w)} \geq \beta \quad (3.3.2)$$

这里， $\xi \geq 0$ 表示背景噪声强度， $\beta \geq 1$ 表示解码门限值， $\mathcal{S}_t$ 表示与 $v$ 在同一时刻 $t$ 发射的节点集合(不包括 $v$ 自身)。

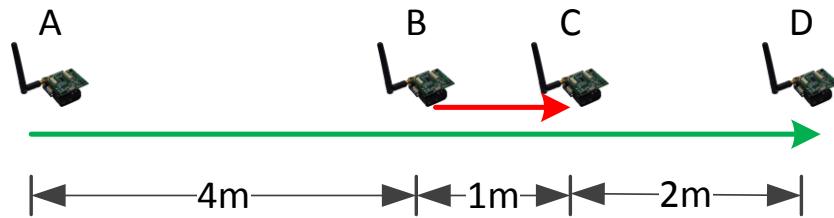


图 3.2 说明物理干扰模型准确性的一个简单例子

相比图干扰模型，物理干扰模型能够更为准确地描述系统实际的干扰情况。图 3.2 以一个简单的示例说明这一判断<sup>[90]</sup>。在该图中，参数配置如下： $\alpha = 3$ ， $\beta = 3$ ， $\xi = 10nW$ ， $Q_A = 1dBm$  和  $Q_B = -15dBm$ 。在图干扰模型下，由于节点 $C$ 在 $A$ 的干扰范围内，链路 $A \rightarrow D$ 和链路 $B \rightarrow C$ 必然不能并发传输。然而，在物理干扰模型下，接收方 $C$ 处的 SINR 值为： $\frac{31.6\mu W/(1m)^3}{0.01\mu W+1.26mW/(5m)^3} \approx 3.11 \geq \beta$ ，接收方 $D$ 处的 SINR 值为：

$\frac{1.26mW/(7m)^3}{0.01\mu W+31.6\mu W/(3m)^3} \approx 3.13 \geq \beta$ 。均能满足 SINR 解码条件，也就是说链路 $A \rightarrow D$ 和链路 $B \rightarrow C$ 能够并发传输。

### 3.3.3 问题描述

我们给出物理干扰模型下的无线传感器网络最小时延数据聚合调度问题的准确描述。我们将该问题简记为：**MLDAS-PHY**，表示 Minimum Latency Data Aggregation Scheduling under Physical Interference Model。类似于第二章中 **MLDAS** 问题的定义，我们将时间划分为连续的等长时隙。每个时隙的长度设定为足够发送或者接收一个数据包的长度。我们寻找发送节点集合序列  $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_l$ ，其满足如下条件：(1)  $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset, \forall i \neq j$ ，(2)  $\bigcup_{i=1}^l = \mathcal{V} \setminus \{v_s\}$ ，(3) 在时隙 $k$ ( $k = 1, 2, \dots, l$ )，数据包从 $\mathcal{S}_k$ 成功聚合到 $\mathcal{V} \setminus \bigcup_{i=1}^k \mathcal{S}_i$ ，且经过 $l$ 个时隙，所有节点的数据包都被聚合到 sink 节点 $v_s$ 。这里条件(1)确保每个节点至多传输一次，条件(2)确保所有节点都有机会传输，条件(3)确保了每条链路传输的成功性。我们希望能寻找一种集合序列使得 $l$ 最小，这里 $l$ 表示数据聚合时延。在文献<sup>[82]</sup>中，**MLDAS-PHY** 已经被证明为 **NP**-难的。本文将针对该

问题提出一种分布式数据聚合算法。

### 3.4 物理干扰模型下的分布式数据聚合算法

#### 3.4.1 算法概述

为了解决 **MLDAS-PHY** 问题, 我们需要求解三个子问题, 即传输路径选择问题、传输时序调度问题和功率控制问题。具体来说, 给定传感器节点  $v \in \mathcal{V} \setminus \{v_s\}$ , 我们需要决定  $v$  的下一跳节点  $P_v$ ,  $v$  的传输时隙  $H_v$  和  $v$  的传输功率  $Q_v$ 。在本节中, 针对物理干扰模型下的数据聚合问题, 我们提出一种分布式数据聚合(Distributed Data Aggregation, DDA)解决方案。该方案包括三个核心子算法, 即分布式数据聚合树构造(Distributed Data Aggregation Tree Construction, DDATC)子算法, 分布式链路调度(Distributed Link Scheduling, DLS)子算法和功率分配(Power Allocation, PA)子算法, 分别用于解决上述路径选择、时序调度和功率控制问题。图 3.3 给出了 **DDA** 的整体框架示意图。接下来, 我们首先分别介绍各个子算法的设计细节, 然后对算法的可行性和最坏时延性能进行理论分析。

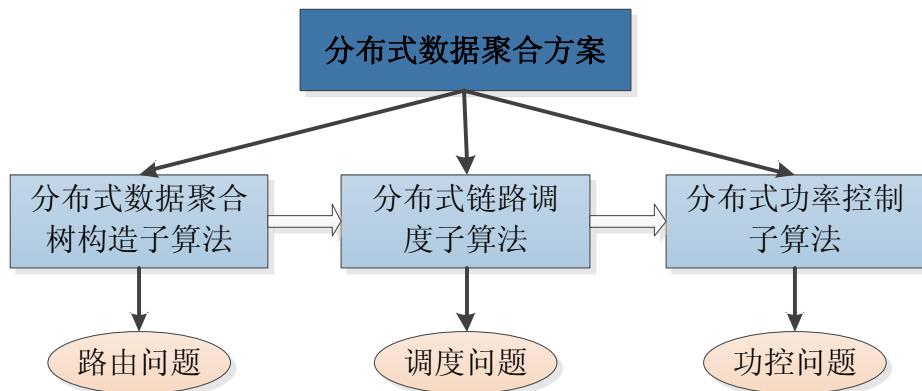


图 3.3 物理干扰模型下的无线传感器网络分布式数据聚合方案 DDA 总体框架

#### 3.4.2 分布式数据聚合树构造子算法

在本节, 我们介绍一种分布式数据聚合树构造(**DDATC**)算法。**DDATC** 算法借鉴了本文 2.4.3 节中的网络分区方法, 不同的是该算法是以迭代(iterative)的方式工作的。在第  $m$  次迭代( $m = 1, 2, \dots, m^*$ ), 我们采用一族垂直线  $x = i \cdot 2^{m-1}a, i \in \mathbb{Z}$  和一族水平线  $y = j \cdot 2^{m-1}a, j \in \mathbb{Z}$  将二维平面划分为半开半闭、大小为  $2^{m-1}a \times 2^{m-1}a$  的正方形小区。这里  $a$  为最小分区边长。在第 1 次迭代时, 每个节点根据预先确定的位置信息, 或者根据定位算法<sup>[23]</sup>, 可以获知自己所在小区身份。每个小区的身份可以由其中心位置坐标唯一标定。接下来, 小区内的每个节点运行本地广播(local broadcasting)算法<sup>[85]</sup>, 告知所在小区其他邻居节点自己的位置和信息。我们约定每个小区距离 sink 节

点 $v_s$ 最近的传感器节点充当所在小区的首部节点。 $v_s$ 的位置信息可以在算法初始化时,由 $v_s$ 广播给全网所有节点。小区中其他节点将该小区首部节点选定为自己的下一跳节点(即父节点)。只有当前迭代被选择为小区首部的传感器节点才参与算法的下一次迭代。当前迭代已经确定父节点的传感器节点将在下一次迭代中保持沉默。重复上述过程,直到第 $m^*$ 次迭代,此时小区大小为 $2^{m^*-1}a \times 2^{m^*-1}a$ 。约定 $m^* \geq \log_2(L/a) + 1$ ,则此时网络中只有一个小区,该小区中离 $v_s$ 最近的节点充当小区首部,成为小区中其他节点的下一跳节点,并且该节点将 $v_s$ 选择为父节点。由于 $m$ 为整数,故我们可以取 $m^* = \lceil \log_2(L/a) + 1 \rceil$ 。至此,一颗以 $v_s$ 为根节点的数据聚合树 $T$ 被成功构造。图 3.4 以伪代码的形式给出 **DDATC** 算法的工作流程。

---

**Algorithm : 分布式数据聚合树构造算法(DDATC)**


---

**输入:** 节点集合 $\mathcal{V}$ , sink 节点 $v_s$

**输出:** 数据聚合树 $T$ , 每个传感器 $v \in \mathcal{V} \setminus \{v_s\}$ 的父节点 $P_v$

- 1: sink 节点 $v_s$ 向全网广播其位置信息;
  - 2: **for**  $m = 1, 2 \dots, m^*$  **do**
  - 3:    $\mathcal{V} \setminus \{v_s\}$ 中的每个节点 $v$ 根据预设或定位技术获取自己在平面中的地理位置;
  - 4:   采用一族垂直线 $x = i \cdot 2^{m-1}a, i \in \mathbb{Z}$ 和一族水平线 $y = j \cdot 2^{m-1}a, j \in \mathbb{Z}$ 将二维平面划分为半开半闭、大小为 $2^{m-1}a \times 2^{m-1}a$ 的正方形小区;
  - 5:   节点 $v$ 根据自身位置信息判断所处的小区 $\text{Cell}(v)$ , 并采用本地广播算法向半径为 $\sqrt{2} \cdot 2^{m-1}a$ 的范围广播自身位置和身份信息;
  - 6:   节点 $v$ 计算自身及 $\text{Cell}(v)$ 中其他节点到 $v_s$ 的距离, 如果 $v$ 距离 $v_s$ 最近, 则 $v$ 称为所在小区首部节点, 否则 $v$ 将距离 $v_s$ 最近的邻居节点确定为父节点 $P_v$ ;
  - 7:   将 $\mathcal{V}$ 所有父节点已经确定的节点移除, 并开始下一次迭代;
  - 8: **end for**
  - 9: 唯一的小区首部节点将 $v_s$ 确定为父节点;
- 

图 3.4 分布式数据聚合树构造算法(DDATC)伪代码

在所得到的数据聚合树 $T$ 中, 节点 $v$ 的父节点表示为 $P_v$ , 孩子节点集合表示为 $\mathcal{C}_v$ , 在树 $T$ 中的深度表示为 $D_v$ 。我们将第 $m$ 次迭代时被选择为小区首部的节点集合表示为 $\mathcal{H}_m$ , 我们有 $\mathcal{H}_1 \supseteq \mathcal{H}_2 \supseteq \dots \supseteq \mathcal{H}_{m^*}$ 。当 $m = 1, 2, \dots, m^* - 1$ 时, 集合 $\mathcal{H}_m \setminus \mathcal{H}_{m+1}$ 中的节点的深度为 $1 + m^* - m$ ; 当 $m = m^*$ 时, 集合 $\mathcal{H}_m$ 中的元素只有一个, 其深度为 1。树的高度 $d(T)$ , 即树中节点的最大深度, 为 $1 + m^*$ 。

图 3.5 以一个网络示例直观地描述了 **DDTAC** 算法的迭代过程。其中图 3.5 (a)、(b)和(c)分别表示第 1、2 和 3 次迭代的结果, 图 3.5 (d)给出了算法构造的数据聚合树。

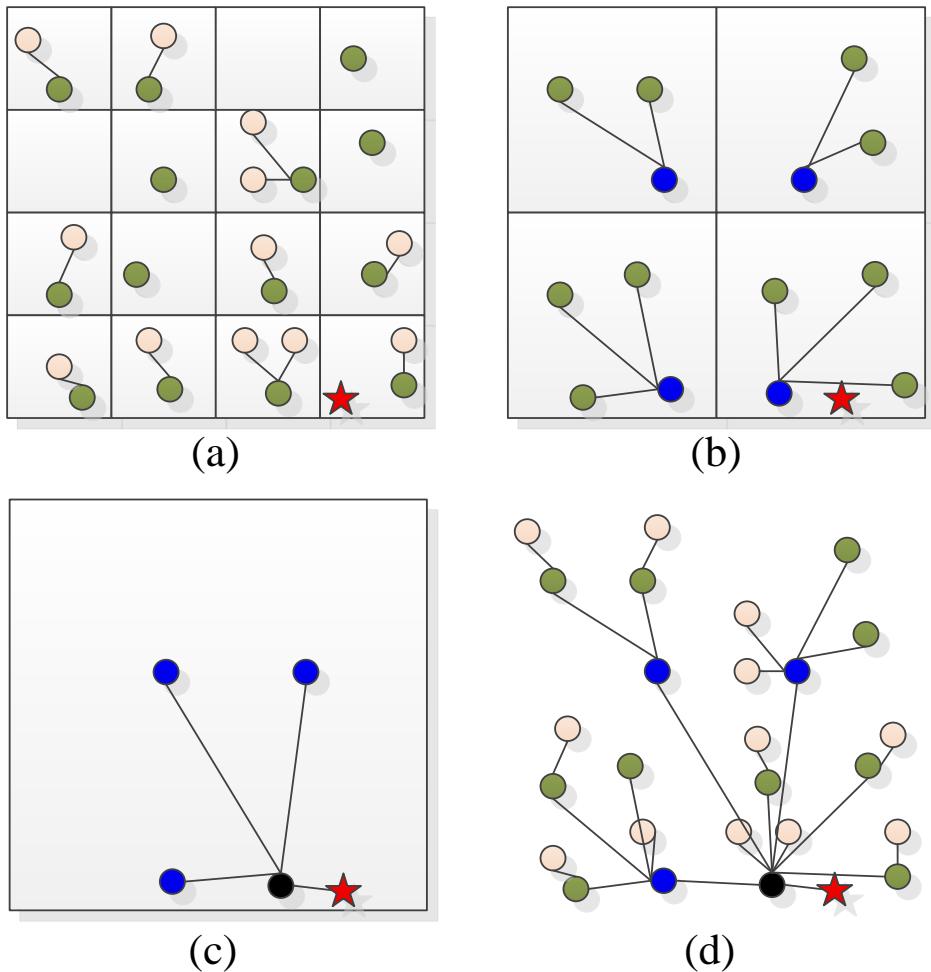


图 3.5 说明分布式数据聚合树构造算法工作过程的一个网络示例: (a) 迭代序号  $m=1$ , (b) 迭代序号  $m=2$ , (c) 迭代序号  $m=3$  (d) 生成的数据聚合树  $T$

### 3.4.3 分布式链路调度子算法

当数据聚合树由 **DDATC** 算法构造完成后, 我们需要对树中的链路进行调度, 即确定每个节点  $v$  的传输时隙  $H_v$ 。本节给出用于完成此项任务的分布式链路调度算法 **DLS**。**DLS** 算法也是一种迭代算法。**DLS** 的迭代次序与 **DDTAC** 一样。在第  $m$  次迭代时, 我们调度当前迭代中未被选为头部的节点(表示为集合  $\mathcal{N}_m$ )。小区的头部节点  $v \in \mathcal{H}_m$  负责为其在集合  $\mathcal{N}_m$  中的孩子节点分配传输时隙。具体来说, 每个小区的头部节点  $v$  向集合  $\mathcal{C}_v \cap \mathcal{N}_m$  中的每个节点发送一个来自  $\{1, 2, \dots, |\mathcal{C}_v \cap \mathcal{N}_m|\}$  的各不相同的整数。每个小区的非头部节点  $v$  在当前迭代轮次的传输时序确定为:  $H_v = (\text{Num}[v] - 1) \cdot k^2(m) + \text{Color}[v]$ , 其中  $\text{Num}[v]$  为  $v$  收到的来自  $P_v$  的整数。这里, 我们采用了类似于 2.4.3 节中的小区着色方法。在第  $m$  次迭代时, 我们用  $k^2(m)$  种颜色对小区进行着色。在进行链路调度时, 我们让具有相同颜色的小区中的链路同时传输。同时, 我们确保每个小区在任何一个时隙最多只有一条链路传输, 这可以通过让首部

节点给每个成员节点发送不同的数字 $\text{Num}[v]$ 来实现。 $\text{Num}[v]$ 代表了成员节点的传输次序。在定理3.2中，我们将证明，通过合理地选择 $k(m)$ 的值，并且精心地控制每个节点的传输功率，我们可以确保在物理干扰模型下，上述调度策略的可行性。在完成 $m^*$ 次迭代后，网络中将只剩下第 $m^*$ 次迭代的首部节点未进行传输。此时，给其分配一个单独的时隙将数据包发送给 sink 节点 $v_s$ 。

---

**Algorithm :分布式链路调度算法(DLS)**


---

**输入:** 数据聚合路由树  $T$

**输出:** 每个传感器 $v \in \mathcal{V} \setminus \{v_s\}$ 的传输时隙 $H_v$

- 1: **for**  $m = 1, 2 \cdots, m^*$  **do**
  - 2:   采用一族垂直线 $x = i \cdot 2^{m-1}a, i \in \mathbb{Z}$ 和一族水平线 $y = j \cdot 2^{m-1}a, j \in \mathbb{Z}$ 将二维平面划分为半开半闭、大小为 $2^{m-1}a \times 2^{m-1}a$ 的正方形小区；
  - 3:   节点 $v$ 根据自身位置坐标 $(x, y)$ 计算所在小区颜色序号： $\text{Color}[v] = \lfloor \frac{x\%(k(m)2^{m-1})}{2^{m-1}a} \rfloor + \lfloor \frac{y\%(k(m)2^{m-1}a)}{2^{m-1}a} \rfloor \times (k(m) - 1)$ ；
  - 4:   每个小区的首部节点 $v$ 向集合 $\mathcal{C}_v \cap \mathcal{N}_m$ 中的每个节点发送一个来自集合 $\{1, 2, \cdots, |\mathcal{C}_v \cap \mathcal{N}_m|\}$ 各异的整数；
  - 5:   每个小区的非首部节点 $v$ 在当前迭代轮次的传输时序确定为： $H_v = (\text{Num}[v] - 1) \cdot k^2 + \text{Color}[v]$ ，其中 $\text{Num}[v]$ 为 $v$ 收到的来自 $P_v$ 的整数；
  - 6: **end for**
  - 7:  $\mathcal{H}_{m^*}$ 中的节点向 $v_s$ 发送数据；
- 

图 3.6 分布式链路调度算法(DLS)伪代码

为了方便描述，我们将整个调度过程分为 $m^* + 1$ 个调度阶段。在第 $m$ ( $1 \leq m \leq m^*$ )次迭代中传输的节点对应于调度过程的第 $m$ 个调度阶段；在第 $m^* + 1$ 个调度阶段，网络中只有一个节点传输，并且目的节点为 $v_s$ 。在前 $m^*$ 个调度阶段，我们基于小区着色技术进行调度时序安排。具体来说，在第 $m$ 个调度阶段，我们采用 $k^2(m)$ 种颜色对小区进行着色，并且让相同颜色小区中的链路同时传输。在第 3.4.5 节，我们将发现，这里 $k(m)$ 的取值不仅将决定调度算法在物理干扰模型下的可行性，而且也会影响调度方案在最坏情况下的时延。另外， $k(m)$ 的取值与物理层采用的功率控制策略也直接相关。在第 3.4.5 节中，我们将给出不同功率控制技术下参数 $k(m)$ 需要满足的条件。

#### 3.4.4 功率分配子算法

为了保证调度策略的正确性，我们需要给每个节点 $v \in \mathcal{V} \setminus \{v_s\}$ 分配合适的传输功率 $Q_v$ ，这可以通过功率分配(Power Allocation, PA)子算法实现。我们考虑以下两种易于分布式实现的功率分配策略：

(1) 均匀功率分配(Uniform Power Allocation, PA-U): 在均匀功率分配下，在第 $m$ 个调度阶段进行传输的所有节点采用相同的传输功率 $Q_0(m)$ 。 $Q_0(m)$ 的选取需满足一定的条件。注意到，在第 $m$ ( $1 \leq m \leq m^*$ )个调度阶段，链路最大长度为 $\sqrt{2} \cdot 2^{m-1}a$ 。

为了保证物理干扰模型下的成功传输，需要满足： $\frac{Q_0(m) \cdot (\sqrt{2} \cdot 2^{m-1}a)^{-\alpha}}{\xi} \geq \beta$ ，我们有：

$Q_0(m) \geq \beta\xi \cdot (\sqrt{2} \cdot 2^{m-1}a)^\alpha$ 。我们将 $\beta\xi \cdot (\sqrt{2} \cdot 2^{m-1}a)^\alpha$ 确定为基准发射功率。在实际情形中，我们取 $Q_0(m) = \delta \cdot [\beta\xi \cdot (\sqrt{2} \cdot 2^{m-1}a)^\alpha]$ 。这里， $\delta \geq 1$ 为可调参数。在第 $m^* + 1$ 个调度阶段，由于此时网络中只有一个小区，链路的最大长度为 $\sqrt{2}L$ ，对应的发射功率不小于 $\beta\xi \cdot (\sqrt{2}L)^\alpha$ 。注意到此时，网络中只有一条活跃链路，因此发射功率可以取最小值 $\beta\xi \cdot (\sqrt{2}L)^\alpha$ 。

(2) 线性功率分配(Linear Power Allocation, PA-L): 线性功率分配下，在任何一个调度阶段，链路的传输功率均与链路的能量损耗(loss)成正比。具体来说，对于长度为 $d(v, P_v)$ 的链路 $(v, P_v)$ ，发送方 $v$ 的传输功率 $Q_v$ 与链路 $(v, P_v)$ 的能量损耗，即 $d^\alpha(v, P_v)$ 成正比，即 $Q_v = \epsilon \cdot d^\alpha(v, P_v)$ 。这里参数 $\epsilon$ 需满足条件： $\frac{\epsilon \cdot d^\alpha(v, P_v) \cdot d^{-\alpha}(v, P_v)}{\xi} \geq \beta$ ，等价于 $\epsilon \geq \beta\xi$ 。

### 3.4.5 算法性能分析

在本小节，我们从理论角度分析上述算法的性质。分析的重点在于：(1)说明在物理干扰模型下，当满足某些条件时，所提出的调度算法具有可行性，(2)调度算法具有较好的理论时间效率。

**引理 3.1:** 在 **DDATC** 算法构造的数据聚合树 $T$ 中，设 $\mathcal{C}_v$ 表示节点 $v$ 的孩子节点集合， $\mathcal{N}_m$ 表示在算法在第 $m$ ( $1 \leq m \leq m^*$ )次迭代中的成员节点集合，则我们有：

$$|\mathcal{C}_v \cap \mathcal{N}_m| \leq \begin{cases} \lfloor \frac{2a^2}{\sqrt{3}} + 2a + 1 \rfloor & m = 1 \\ 3 & m = 2, \dots, m^* \end{cases} \quad (1a)$$

$$(1b)$$

证明. 首先考虑 $m = 1$ 的情形，此时平面被划分为边长为 $a$ 的正方形小区。注意到任意两个节点之间的距离不小于 1。根据 Wegner Theorem<sup>[56]</sup>，在边长为 $a$ 的正方形中，两两之间距离不小于 1 的点的个数不超过 $\lfloor \frac{2a^2}{\sqrt{3}} + 2a \rfloor + 1$ 。另外，每个小区有一个节点成为首部节点。因此，第 1 次迭代时，每个小区的成员节点个数不超过 $\lfloor \frac{2a^2}{\sqrt{3}} + 2a \rfloor$ 个。对于节点 $v$ ，在每一轮的迭代中都有可能有其他节点成为 $v$ 的子节点，表示为集合 $\mathcal{C}_v \cap \mathcal{N}_m$ 。在 $m = 1$ 时，如果 $v$ 为首部节点，则 $v$ 的子节点为 $v$ 所在小区的成员节点，我们有 $|\mathcal{C}_v \cap \mathcal{N}_m| \leq \lfloor \frac{2a^2}{\sqrt{3}} + 2a \rfloor$ ；如果 $v$ 为成员节点，则 $\mathcal{C}_v \cap \mathcal{N}_m = \emptyset$ ，上述条件同样满足。接着我们考虑 $m \geq 2$ 的情形。此时，平面被划分为边长为 $2^{m-1}a$ 的小区。每个小区最多有 4 个节点，其中一个充当首部节点，另外 3 个充当成员节点。因此，节点 $v$ 在第 $m$ 次迭代时的子节点的个数不超过 3 个，即 $|\mathcal{C}_v \cap \mathcal{N}_m| \leq 3$ 。综合上面两种情形所述，引理得证。  $\square$

**定理 3.2:** 在物理干扰模型下, 当物理层采用均匀功率(或者线性)分配策略时, 如果  $k(m) \geq (\frac{4\delta}{\delta-1}(1+2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha))^{\frac{1}{\alpha}} + 1$ (或者  $k(m) \geq (\frac{4\epsilon}{\epsilon-\beta\xi}(1+2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha))^{\frac{1}{\alpha}} + 1$ ), 则算法

**DLS** 产生的调度方案是可行的。也就是说, 在任何一个时隙所调度的每一条链路的接收方均符合 SINR 解码条件。

证明. 我们分两种情况讨论 **DLS** 算法的可行性, 即物理层采用均匀功率分配的情形和物理层采用线性功率分配的情形。首先, 我们考虑均匀功率分配, 即 **PA-U** 的情形。为了证明调度算法在物理干扰模型下的可行性, 我们需要证明在任意一个时隙  $t$  传输的任意一条链路  $(v, P_v)$  均能成功, 即接收方  $P_v$  处的 SINR 不小于解码门限  $\beta$ 。由于在第  $m^* + 1$  个调度阶段, 只有一条长度不大于  $\sqrt{2}L$  的链路活跃, 且传输功率不小于  $\beta\xi \cdot (\sqrt{2}L)^\alpha$ , 因此该链路的传输必然满足 SINR 解码条件。于是, 我们只需要考虑前  $m^*$  个调度阶段的情况。我们需要计算接收方  $P_v$  处的信号强度和总的干扰强度。

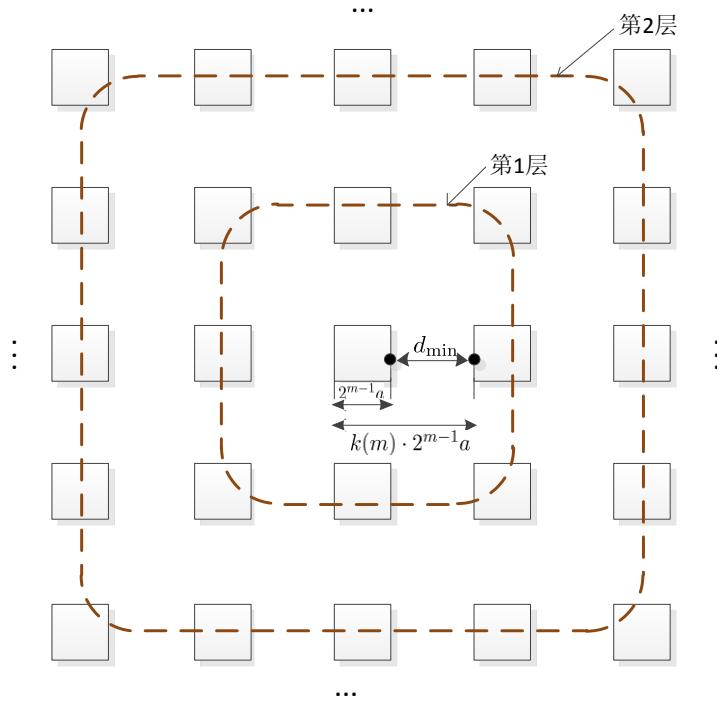


图 3.7 对节点  $P_v$  的干扰来自所有方向具有相同颜色小区中的发送节点

首先, 我们给出总的干扰强度  $\mathcal{I}$  的上界。对于链路  $(v, P_v)$ , 干扰来自于与该链路所在小区颜色相同的其他小区中的并发链路。对于这样一条干扰链路的发送方, 其与  $P_v$  的最短距离  $d_{\min}$  满足条件:  $d_{\min} \geq (k(m) - 1) \cdot 2^{m-1}a$ , 相应的干扰强度不大于  $Q_0(m) \cdot ((k(m) - 1)2^{m-1}a)^{-\alpha}$ 。实际上, 对链路  $(v, P_v)$  的干扰来自于该链路的各个方向。如图 3.7 所示, 第 1 层, 第 2 层, ..., 等小区中的并发链路均会对  $(v, P_v)$  造成干扰。所有这些干扰的强度之和  $\mathcal{I}$  不超过:

$$\begin{aligned}
\mathcal{I} &\leq \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} Q_0(m) \cdot ((k(m)-1)2^{m-1}a \cdot \sqrt{i^2+j^2})^{-\alpha} \\
&= 4 \sum_{i=1}^{\infty} Q_0(m) \cdot ((k(m)-1)2^{m-1}a \cdot i)^{-\alpha} + 4 \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} Q_0(m) \cdot ((k(m)-1)2^{m-1}a \cdot \sqrt{i^2+j^2})^{-\alpha} \\
&= 4Q_0(m) \cdot ((k(m)-1)2^{m-1}a)^{-\alpha} \left( \sum_{i=1}^{\infty} i^{-\alpha} + \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} (\sqrt{i^2+j^2})^{-\alpha} \right) \\
&\leq 4Q_0(m) \cdot ((k(m)-1)2^{m-1}a)^{-\alpha} (\zeta(\alpha) + 2^{-\frac{\alpha}{2}} \cdot \sum_{i=1}^{\infty} i^{-\frac{\alpha}{2}} \cdot \sum_{j=1}^{\infty} j^{-\frac{\alpha}{2}}) \\
&= 4Q_0(m) \cdot ((k(m)-1)2^{m-1}a)^{-\alpha} (\zeta(\alpha) + 2^{-\frac{\alpha}{2}} \cdot \sum_{i=1}^{\infty} i^{-\alpha}) \\
&= 4Q_0(m) \cdot ((k(m)-1)2^{m-1}a)^{-\alpha} (1 + 2^{-\frac{\alpha}{2}}) \cdot \zeta(\alpha)
\end{aligned}$$

这里  $\zeta(\cdot)$  是 Riemann Zeta Function<sup>[91]</sup>。于是，节点  $P_v$  处的 SINR 为：

$$\begin{aligned}
SINR &= \frac{Q_0(m) \cdot d^{-\alpha}(v, P_v)}{\mathcal{I} + \xi} \\
&\geq \frac{\delta \cdot [\beta \xi \cdot (\sqrt{2} \cdot 2^{m-1}a)^\alpha] \cdot (\sqrt{2} \cdot 2^{m-1}a)^{-\alpha}}{4\delta \cdot [\beta \xi \cdot (\sqrt{2} \cdot 2^{m-1}a)^\alpha] \cdot ((k(m)-1)2^{m-1}a)^{-\alpha} (1 + 2^{-\frac{\alpha}{2}}) \cdot \zeta(\alpha) + \xi} \\
&= \frac{\delta \beta \xi}{4\delta \beta \xi \cdot (1 + 2^{-\frac{\alpha}{2}}) \cdot \zeta(\alpha) \cdot (\sqrt{2})^\alpha \cdot [k(m)-1]^{-\alpha} + \xi} \\
&\geq \beta
\end{aligned}$$

最后一个不等式成立，是因为  $k(m) \geq (\frac{4\delta}{\delta-1} (1 + 2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha))^{\frac{1}{\alpha}} + 1$ 。

接下来，我们考虑线性功率分配，即 **PA-L** 的情形。此时，对于在第  $m$  个调度阶段传输的节点  $v$ ，其发射功率  $Q_v$  满足  $Q_v = \epsilon \cdot d^\alpha(v, P_v) \leq \epsilon \cdot (\sqrt{2} \cdot 2^{m-1}a)^\alpha$ 。采用类似的方法，我们可以计算来自所有其他节点的对接收方  $P_v$  总的干扰强度  $\mathcal{I}$  不超过： $\mathcal{I} \leq \epsilon \cdot (1 + 2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha) \cdot (k(m)-1)^{-\alpha}$ 。据此，节点  $P_v$  处的 SINR 为：

$$\begin{aligned}
SINR &= \frac{Q_v \cdot d^{-\alpha}(v, P_v)}{\mathcal{I} + \xi} \\
&\geq \frac{\epsilon}{4\epsilon \cdot (1 + 2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha) \cdot (k(m)-1)^{-\alpha} + \xi} \\
&\geq \beta
\end{aligned}$$

最后一个不等式成立，是因为  $k(m) \geq (\frac{4\epsilon}{\epsilon-\beta\xi} (1 + 2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha))^{\frac{1}{\alpha}} + 1$ 。定理得证。  $\square$

注意，在实际调度算法中，为了减少数据聚合的时延，我们希望采用尽可能少的颜色数量。因此，我们通常取  $k(m)$  的最小值。具体来说，当物理层采用均匀功率控制 **PA-U** 时，我们取  $k(m) \geq \lceil (\frac{4\delta}{\delta-1} (1 + 2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha))^{\frac{1}{\alpha}} \rceil + 1$ ；而当物理层采用线性功率控制 **PA-L** 时，我们取  $k(m) \geq \lceil (\frac{4\epsilon}{\epsilon-\beta\xi} (1 + 2^{\frac{\alpha}{2}}) \cdot \zeta(\alpha))^{\frac{1}{\alpha}} \rceil + 1$ 。这里  $\delta$  和  $\epsilon$  分别是 **PA-U** 和 **PA-L**

可配置的系统参数。接下来，我们用 $K$ 表示最小颜色数量，即 $K = (k^2(m))_{\min}$ 。

**定理 3.3：**在最坏情况下，数据聚合调度算法 **DLS** 所消耗的时隙数量不超过 $\lfloor \frac{2a^2}{\sqrt{3}} + 2a + 1 \rfloor \cdot K + 3K \cdot m^* - 3K + 1$ 。

证明. 调度算法 **DLS** 所消耗的时隙数量，表示为 $Time(DLS)$ ，等于 $m^* + 1$ 个调度阶段各自消耗的时隙数量之和。我们用 $Time(m)$ 表示第 $m$ 个调度阶段花费的时隙数量，这里 $m = 1, 2, \dots, m^* + 1$ 。则我们有：

$$Time(DLS) = \sum_{m=1}^{m^*+1} Time(m)$$

注意到，当 $m = m^* + 1$ 时，只有一条活跃链路， $Time(m) = 1$ 。我们考虑 $m < m^* + 1$ 的情形。在每个调度阶段的每个时隙，具有相同颜色的小区中的链路允许并发，而我们一共有 $k^2(m)$ 种不同的颜色。因此，经过 $k^2(m)$ 个时隙，每个小区至少有一条链路被调度。根据引理 3.1，每个小区有最多有 $|\mathcal{C}_v \cap \mathcal{N}_m|$ 条链路需要调度。于是，我们有：

$$Time(m) \leq k^2(m) \cdot |\mathcal{C}_v \cap \mathcal{N}_m|$$

注意到，引理 3.1 给出了 $|\mathcal{C}_v \cap \mathcal{N}_m|$ 的上界。因此，我们有：

$$\begin{aligned} Time(DLS) &= \sum_{m=1}^{m^*} Time(m) + 1 \\ &\leq \lfloor \frac{2a^2}{\sqrt{3}} + 2a + 1 \rfloor \cdot k^2(1) + 3 \sum_{m=2}^{m^*} k^2(m) + 1 \\ &= \lfloor \frac{2a^2}{\sqrt{3}} + 2a + 1 \rfloor \cdot K + 3K \cdot m^* - 3K + 1 \end{aligned}$$

由上述式子可知，**DLS** 的时延上界与网络规模无关，因而其具有良好的可扩展性。注意到 $K$ 是常数，因此调度总时隙数 $Time(DLS)$ 与分区大小 $a$ 密切相关。另外， $K$ 的值与物理层功率控制参数 $\delta$ 和 $\epsilon$ 也密切相关。在下一章，我们将通过仿真实验来探索参数 $K$ 、 $\delta$ 和 $\epsilon$ 对网络时延、能耗等性能指标的影响。定理得证。  $\square$

### 3.5 算法性能评价

在本节，我们通过仿真实验评价分布式数据聚合调度方案 **DDA** 的性能。我们将采用均匀功率控制策略 **PA-U** 和线性功率控制策略 **PA-L** 的 **DDA** 方案分别表示为 **DDA-U** 和 **DDA-L**。我们考虑二维平面上的传感器网络，网络监测区域大小固定为 $100m \times 100m$ ，sink 节点位于监测区域中心位置。我们评价的性能指标有数据聚合时延和数据聚合过程中节点的平均能量消耗。为了方便统计，我们将单个时隙的长度归一化为 1。因此数据聚合时延就是数据聚合所消耗的时隙数量。由于网络的几何拓扑

会影响到 **DDA** 的性能，因此对于每次实验，我们重复进行 50 次独立实验，并给出 50 次实验结果的平均值和标准差作为实验结果。

我们首先研究网络节点数量对数据聚合时延的影响。我们将最小分区边长  $a$  设置为 5，SINR 参数  $\alpha$ ,  $\beta$  和  $\xi$  分别设置为 3, 1 和 0.0001。功率控制参数，即 **PA-U** 中的  $\delta$  和 **PA-L** 中的  $\epsilon$  分别为 3 和 0.004。图 3.8 给出了此时的实验结果。这里网络节点数量的取值为 500, 1000, 1500, 2000, 2500, 5000 和 10000。我们比较 **DDA-L**, **DDA-U** 以及各自的理论界 **DDA-L-UB** 和 **DDA-U-UB**。观察图 3.8 可知，当节点数量增加时，调度方案 **DDA-L** 和 **DDA-U** 的时延均单调增加。当节点数量变化时，**DDA-L-UB** 和 **DDA-U-UB** 的时延并无变化。这是因为调度方案的理论最坏时延与网络规模无关，显示了调度方案具有良好的扩展性。在整个节点数量变化范围中，采用线性功率分配的 **DDA-L** 的时延均显著低于采用均匀功率分配的 **DDA-U**。具体来说，当节点数量为 500 时，前者时延仅为后者的 64%；当节点数量为 10000 时，前者时延为后者的 71%。另外，我们可以发现，无论网络规模如何变化，**DDA-U** 和 **DDA-L** 的时延均没有超过各自的上界。并且，当网络规模足够大时，它们的时延将逐渐逼近各自的理论界。这一方面说明了我们推导出的理论界的正确性，另一方面也表明在低密度网络场景中，调度方案的实际性能要远远好于理论上界。

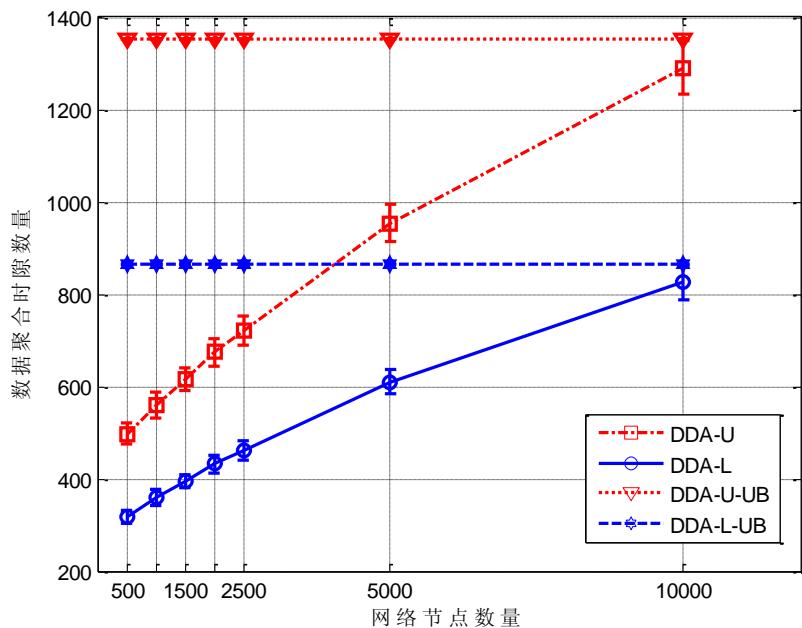


图 3.8 网络节点数量对数据聚合时延的影响

接下来，我们研究 SINR 参数，即路径衰减指数  $\alpha$  和解码门限  $\beta$  对数据聚合时延的影响。我们将最小分区边长固定为 5，参数  $\xi$ ,  $\delta$  和  $\epsilon$  分别设置为 0.0001, 3 和 0.0004。另外，将节点数量固定为 2000。在研究路径衰减指数  $\alpha$  对聚合时延的影响时，我们将 SINR 解码门限  $\beta$  固定为 1。图 3.9 给出了此时的实验结果，包括 **DDA-U**, **DDA-L**,

**DDA-U-UB** 和 **DDA-L-UB** 四种算法在不同 $\alpha$ 下的平均时延。观察图 3.9 可以发现，当路径损耗指数 $\alpha$ 增大时，四种算法的平均时延均单调减少。例如，当 $\alpha$ 从 2 增大到 6 时，**DDA-U** 的时延减少到原来的 18%。这是由于当 $\alpha$ 增大时，信号随着距离衰减的速度加快。彼此距离一定范围内的两条链路之间的相互干扰程度将减弱。这意味着我们可以实现更大程度的空间复用，因此调度算法所需要的时隙数量也就减少了。我们还可以发现，当 $\alpha$ 变化时，**DDA-L** 的时延不高于 **DDA-U**；并且，**DDA-L** 和 **DDA-U** 的时延均低于各自的理论界。接着探讨 SINR 解码门限 $\beta$ 对数据聚合时延的影响，此时我们将 $\alpha$ 固定为 3。图 3.10 给出了当 $\beta$ 从 1 增加到 2 时，算法数据聚合时延的变化趋势图。观察图 3.10，我们可以发现，无论是在均匀网络还是在分区网络中， $\beta$ 的值对各个调度算法的时延性能的影响并不显著。这是由于 $\beta$ 只是一个比值(ratio)，其值的变化并不会对调度算法的性能产生实质影响。

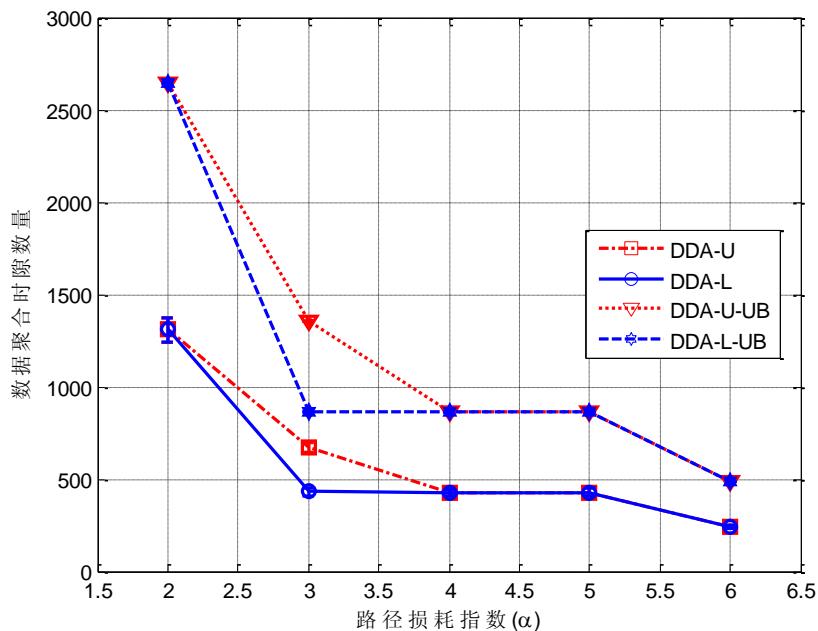


图 3.9 路径损耗因子 $\alpha$ 对数据聚合时延的影响

在第 3.4.5 节中的分析可知，网络最小分区边长会对 **DDA-L** 和 **DDA-U** 的时延上界产生影响。这里我们从实验角度评价这种影响的程度。我们将 SINR 参数 $\alpha$ ,  $\beta$ 和 $\xi$ 分别设置为 3, 1 和 0.0001，将功率控制参数 $\delta$ 和 $\epsilon$ 分别设置为 3 和 0.004。网络中的节点数量为 2000。图 3.11 给出了当网络最小分区边长取值为{1, 2, 4, 8, 10, 20}时，调度算法的时延性能对比图。我们可以发现，当最小分区大小增大时，算法 **DDA-U** 和 **DDA-L** 的时延均呈单调增长趋势。然而，相对各自的理论界，算法的增长趋势较为缓慢。具体来说，当最小分区大小为 1 时，**DDA-U** 和 **DDA-L** 的时延为各自理论界的 93.2% 和 93.3%，而当最小分区大小为 20 时，**DDA-U** 和 **DDA-L** 的时延只有各自理论界的 21.0% 和 21.0%。可见，**DDA-U** 和 **DDA-L** 的实际时延性能要远远

优于各自的理论界，尤其当最小分区边长比较大时。

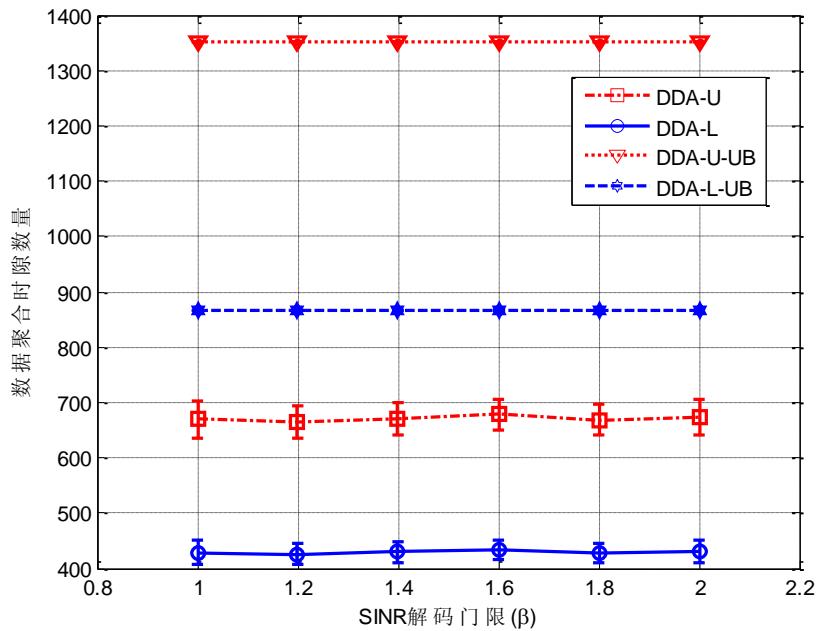


图 3.10 SINR 解码门限  $\beta$  对数据聚合时延的影响

最后，我们研究采用均匀功率分配和线性功率分配时，网络中节点的平均能量消耗情况。我们将最小分区边长设置为 5，SINR 参数  $\alpha$ ,  $\beta$  和  $\xi$  分别设置为 3, 1 和 0.0001。图 3.12 给出了不同网络规模和不同功率控制策略下的节点平均能量消耗。我们可以发现，采用均匀功率分配策略时，系统参数  $\delta$  越大，节点的平均能耗越高；采用线性功率分配策略时，系统参数  $\epsilon$  越大时，节点平均能耗越高。这说明，功率控制模块中的系统参数会直接影响到网络的能量消耗。我们还可以发现，采用线性功率分配策略情形(DDA-L)的节点平均能量消耗要远远小于采用均匀功率分配的情形(DDA-U)。

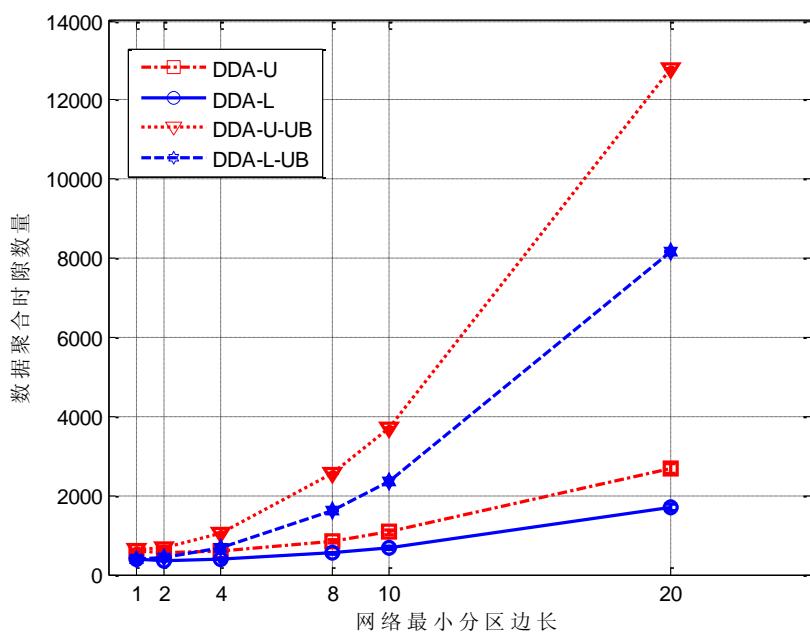


图 3.11 网络最小分区边长对数据聚合时延的影响

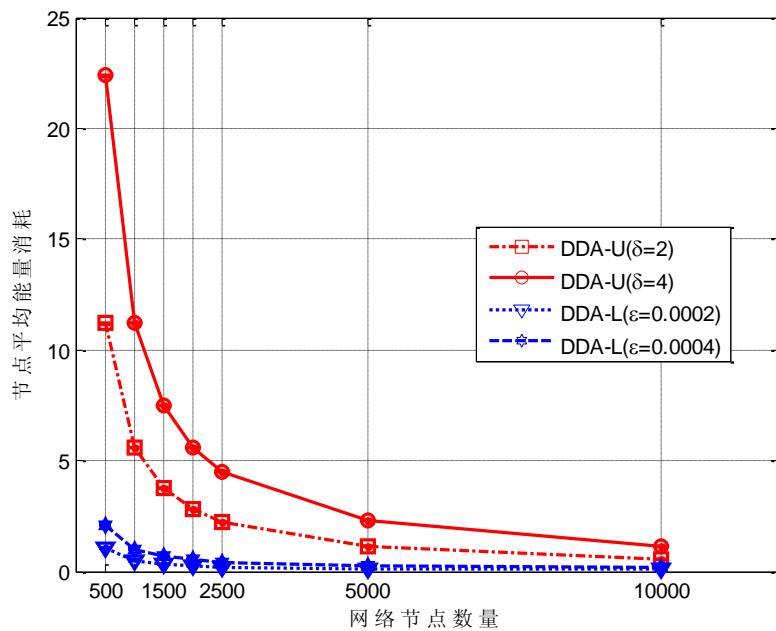


图 3.12 不同功率控制策略下的节点平均能量消耗对比

### 3.6 本章小结

本文研究了物理干扰模型下的无线传感器网络数据聚合问题，目的是设计分布式的跨层(Cross-layer)算法，以最小化数据聚合的总时延。为此，本文提出了一种分布式数据聚合解决方案 **DDA**。该方案包含一种分布式数据聚合树构造算法 **DDATC** 和一种分布式的链路调度算法 **DLS**。另外，在物理层，我们采用两种可选的传输功率控制策略，即均匀功率分配 **PA-U** 和线性功率分配 **PA-L**。通过理论分析，我们证明可以通过合理配置功率控制和链路调度模块的参数来保证 **DDA** 方案在物理干扰模型下的可行性。同时，我们还推导了 **DDA** 在最坏情况下的聚合时延。分析结果显示 **DDA** 具有良好的扩展性。最后，我们通过大量的仿真实验评价了 **DDA** 方案在不同功率控制策略和不同网络参数配置下的时延、能耗等性能。

## 第4章 基于多包接收技术的树状无线传感器网络数据聚合调度

### 4.1 引言

干扰是制约无线网络调度算法性能提升的瓶颈。在第2章和第3章，我们分别研究了协议干扰模型和物理干扰模型下的无线传感器网络最小时延数据聚合调度问题。在协议干扰模型下，无线干扰被认为只存在于彼此距离在一定范围之内的链路之间。设计协议模型下的调度算法时，通常让在彼此干扰范围之内的链路在不同时隙传输，从而实现干扰避免。在物理干扰模型下，来自当前所有活跃链路的干扰均被考虑进来，并且一个信号能被成功接收的充分必要条件是接收方处的信号干扰噪声比 SINR 不小于解码门限。这意味着，接收方在解码目的信号时，将非目的信号(即干扰)与背景噪声同等看待。因而在设计物理模型下的调度算法时，只要每条链路的接收方均能够忍受来自其他并发链路的干扰，那么这些链路就能够并发传输。我们可以认为物理干扰模型下的调度方法是基于干扰容忍的。

无论是基于协议干扰模型还是基于物理干扰模型的调度算法，其基本原理都是让潜在的相互冲突链路尽可能地在不同的时隙调度，从而避免或降低无线干扰对目的信号接收的危害。然而，无线通信物理层技术的发展表明，通过使用多包接收(Multi-Packet Reception, MPR)技术，接收方有能力同时解码多个接收信号<sup>[63]</sup>。相继干扰消除，即 Successive Interference Cancellation (SIC)是多包接收的一种代表性、且十分有前景的技术<sup>[92]</sup>。在文献<sup>[92, 93]</sup>中，相关的理论和实验研究已经表明，采用 SIC 技术能够显著提升无线网络的性能。由于接收方被允许同时接收多个信号，因此在同一时隙，将有更多的链路并行传输，这有望提升网络吞吐量，从而降低数据聚合的时延。

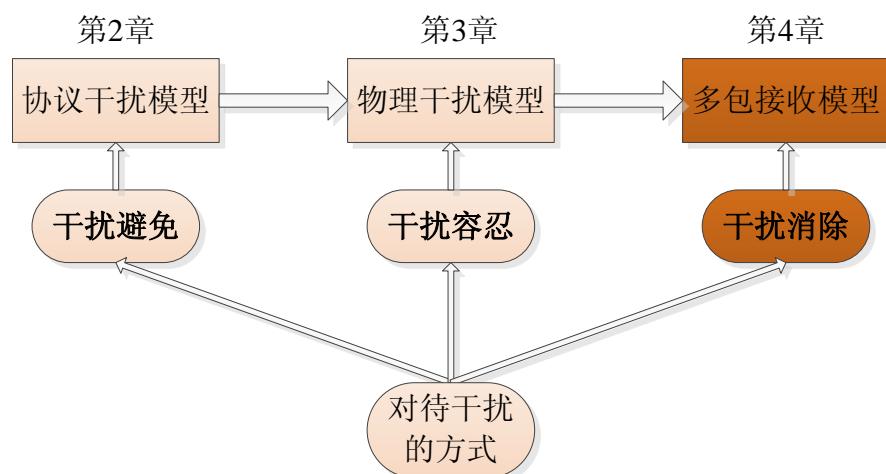


图 4.1 对待无线干扰的方式

本章研究当节点物理层具备多包接收能力时, 无线传感器网络中的数据聚合调度问题。本章与前面两章的关系如图 4.1 所示。根据对待干扰的方式, 第 2 章我们侧重设计基于干扰避免的调度算法, 在第 3 章我们侧重设计基于干扰容忍的调度算法, 而在本章, 我们将聚焦基于干扰消除的调度算法。另外, 在本章中, 我们考虑路由已经由上层协议确定的情形。特别的, 我们考虑一种生成树(spanning tree)形式的数据收集路由。我们对这一新场景下的数据聚合调度问题进行了描述, 并证明了该问题为 **NP**-难问题。为此, 我们提出两种启发式贪心调度算法, 分别为基于随机链路优先的贪心调度(Random-link-First Greedy Scheduling, RFGS)和基于最短链路优先的贪心调度(Shortest-link-First Greedy Scheduling, SFGS)。两种算法均具有多项式时间复杂度, 并且通过在每个时隙构造极大的可行链路集合, 以充分利用 SIC 带来的链路并发机会, 从而较少调度所需的时隙数量。针对大规模网络的仿真实验表明, 在各种网络场景下, 相比传统的未采用多包接收技术的聚合调度算法, 本文算法的聚合时延显著降低。

## 4.2 相关工作

一般来说, 通过挖掘多用户干扰的结构特性, 而不是简单地将其看作噪声, 多用户接收方能够获得显著的性能增益<sup>[92]</sup>。文献<sup>[94]</sup>首先提出以相继干扰消除 SIC 作为多用户检测的一种有效手段。SIC 的核心思想是让接收方连续地消除干扰, 使多个用户能够同时被解码。文献<sup>[95]</sup>通过实际系统设计, 证明了 SIC 的简洁性和可行性。为了进一步理解 SIC 给自组织网络传输容量带来的提升, 文献<sup>[92]</sup>研究了当自组织网络中的接收节点具有完全(perfect)SIC 能力和不完全(imperfect)SIC 能力时, 网络传输容量下界的闭合表达式。基于随机几何框架, 文献<sup>[96]</sup>研究了具备 SIC 能力的 $d$ -维随机无线网络的累积吞吐量问题。考虑具备任意衰减分布, 指数路径损耗和泊松点过程的几何网络, 作者展示了 SIC 的性能如何随着网络几何特性、衰减分布和路径损耗法则变化。考虑 SIC, 文献<sup>[97]</sup>研究了随机接入系统中的分布式功率控制问题。

当物理层具有 SIC 能力时, 无线网络中的链路调度问题将更具有挑战性。文献<sup>[98-102]</sup>研究了具有 SIC 能力的无线网络调度问题。其中, 文献<sup>[101]</sup>研究了蜂窝网络中的上行链路调度问题, 其目的是最大化系统吞吐量或者最大化用户公平性。文献<sup>[100]</sup>证明了基于 SIC 的调度问题为 **NP**-难的, 并且提出了基于独立集策略的贪心极大调度算法。另外, 文献<sup>[103]</sup>研究了如何利用 SIC 来改善多跳无线网络的性能。作者发现, 单纯地使用 SIC 不足以克服多跳无线网络中的干扰影响, 因而提倡采用联合的干扰消除和干扰避免机制。为此, 作者提出了一个联合考虑物理层、链路层和网络层的跨层优化框架, 以最大化网络吞吐量。文献<sup>[93]</sup>从 MAC 协议的角度研究了 SIC 给网络性能带来的增益。通过理论分析和仿真实验, 作者发现, 多对一的网络传输场景能够获得比一对传输场景更多的 SIC 增益。

当物理层具备多包接收能力时，面向数据聚合的传输调度方法需要重新设计。相关的研究工作目前还比较匮乏。最近，文献<sup>[104]</sup>研究了当物理层具备 SIC 能力时，无线传感器网络数据聚合的能耗和时延联合优化问题。具体来说，作者给出了时延和能耗的理论下界，并且提出了一种有效的调度算法。该算法能够同时获得渐进最优的聚合时延和时延-能耗折中(tradeoff)。本文与文献<sup>[104]</sup>的不同之处在于：首先，文献<sup>[104]</sup>研究的问题是与物理层功率控制问题高度耦合的，而本文假设节点采用均匀功率分配，并聚焦于调度问题本身；其次，文献<sup>[104]</sup>的设计目标是聚合时延和能量消耗的联合优化，而为了降低能量消耗，在数据聚合的过程中，SIC 带来的链路并发机会仅被部分利用；最后，文献<sup>[104]</sup>对网络拓扑有特殊要求，即对网络中节点之间的最短距离有严格限制，而本文不对网络拓扑做任何约束。也就是说，本文提出的聚合调度算法适用于具有任意拓扑形式的无线传感器网络。

### 4.3 网络模型与问题描述

#### 4.3.1 网络模型

我们考虑二维平面上的树状无线传感器网络  $T(\mathcal{V}, \mathcal{E})$ ，其中  $\mathcal{V}$  为节点集合， $\mathcal{E}$  为链路集合。 $v_s \in \mathcal{V}$  为 sink 节点，也是树  $T$  的根节点。对于节点  $v \in \mathcal{V}$ ，我们用  $P_v$  表示其在树  $T$  中的父节点， $l_v = (v, P_v) \in \mathcal{E}$  表示节点  $v$  和其父节点  $P_v$  之间的链路。每个节点均装备有一根全向天线，能向各个方向发送信号，并能接收来自各个方向的信号。所有节点工作在同一信道。由于自干扰(self-interference)的约束，节点不能同时进行数据的收发。在数据聚合中，父节点  $P_v$  作为接收节点，负责接收来自发送节点  $v$  的数据。对于链路  $l_v$ ，其长度，即  $v$  到  $P_v$  的欧式距离，表示为  $d(l_v) \triangleq d(v, P_v)$ 。对于两条链路  $l_u$  和  $l_v$ ， $l_u$  到  $l_v$  的距离定义为  $l_u$  的发送节点  $u$  到  $l_v$  的接收节点  $P_v$  之间的欧式距离，表示为  $d(l_u, l_v) \triangleq d(u, P_v)$ 。假设所有节点采用固定和一致的发射功率  $Q$ 。对于链路  $l_v$ ，其接收信号强度为  $Q(l_v) = Q/d^\alpha(l_v)$ ，这里  $\alpha \in (2, 6]$  是路径损耗因子。对于链路  $l_v$ ，来自于另外一条同时活跃的链路  $l_u$  的干扰强度，表示为  $Q(l_u, l_v) = Q/d^\alpha(l_u, l_v)$ 。用  $\xi$  表示背景噪声强度。假设网络中的所有节点满足时间同步，并且时间被划分为连续的等长时隙。每个时隙的长度设定为足够发送或者接收一个数据包的长度。

#### 4.3.2 物理层多包接收模型

在传统的物理层模型中，链路  $l_v$  成功传输的条件是，该链路接收方的信号干扰噪声强度比 SINR 不小于解码门限。即：

$$\frac{Q(l_v)}{\xi + \sum_{\substack{l_u \neq l_v \\ l_u \in \mathcal{S}_t}} Q(l_u, l_v)} \geq \beta \quad (4.3.1)$$

该条件又称为 SINR 解码条件。这里  $\beta (\beta \geq 1)$  为解码门限， $\mathcal{S}_t$  表示与  $l_v$  同时活跃的

链路集合。注意到在物理模型下，如果存在某条链路 $l_u$ ，其对 $l_v$ 的干扰强度 $Q(l_u, l_v)$ 大于 $Q(l_v)$ ，则根据式(4.3.1)，链路 $l_v$ 必然传输失败。然而，在SIC模型下，我们有可能从强干扰中解码出较弱的信号 $Q(l_v)$ 。图4.2给出了SIC的工作原理示意图<sup>[92]</sup>。SIC采用一种迭代的方法。在每一次迭代中，我们尝试解码当前最强的一路信号，并将其他较弱信号视作干扰。如果满足SINR解码条件，则当前最强信号可被解码出来，然后对其进行重构，并在混叠信号中将其消除。之后，开始下一次迭代，尝试解码次强信号。这个过程持续进行，直到目的信号被解码出，或者某个信号解码失败。

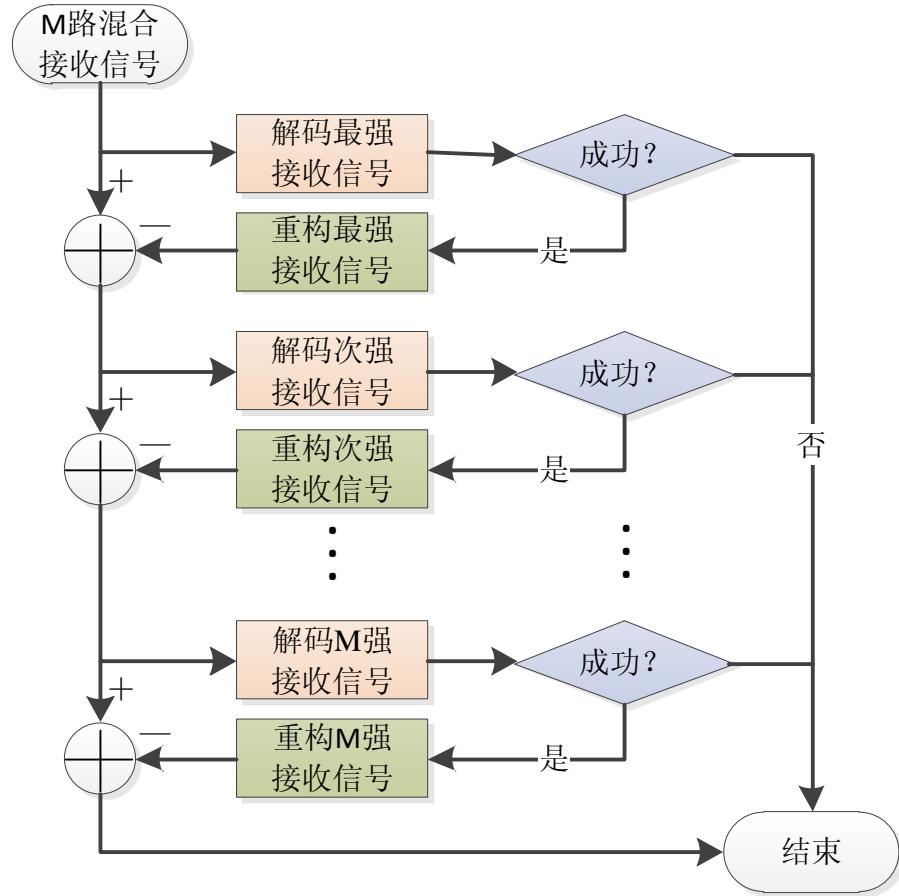


图 4.2 多包接收代表性技术之相继干扰消除的工作流程示意图

在SIC模型下，链路 $l_v$ 成功传输的条件是：(a)对于每一条满足 $Q(l_u, l_v) > Q(l_v)$ ,  $l_u \in \mathcal{S}_t$ 的链路 $l_u$ :

$$\frac{Q(l_u, l_v)}{\xi + \sum_{\substack{l_w \in \mathcal{S}_t, l_w \neq l_v \\ Q(l_w, l_v) \leq Q(l_u, l_v)}} Q(l_w, l_v)} \geq \beta \quad (4.3.2)$$

(b)对于链路 $l_v$ ，满足：

$$\frac{Q(l_v)}{\xi + \sum_{\substack{l_u \in \mathcal{S}_t, l_u \neq l_v \\ Q(l_u, l_v) \leq Q(l_v)}} Q(l_u, l_v)} \geq \beta \quad (4.3.3)$$

### 4.3.3 问题描述

我们定义当网络中的节点具有 SIC 能力时, 树状无线传感器网络中的最小时延数据聚合调度问题。我们将这个问题简记为 **MLDAS-SIC**, 表示 Minimum Latency Data Aggregation Scheduling with Successive Interference Cancellation。在本章的网络模型下, 一个时延为  $K$  个时隙的有效数据聚合调度指的是满足下列条件的发送节点集合序列  $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K$  : (a)  $\bigcup_{i=1}^K \mathcal{S}_i = \mathcal{V} \setminus \{v_s\}$ , (b)  $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset, \forall i \neq j$ , (c) 对于  $i(i = 1, 2, \dots, K)$ , 集合  $\mathcal{S}_i$  中的数据被成功聚合到集合  $\{P_v | \forall v \in \mathcal{S}_i\} \subseteq \mathcal{V} \setminus \bigcup_{j=1}^i \mathcal{S}_j$  中, 即对于集合  $\mathcal{S}_i$  中的任何一个节点  $v$ , 链路  $(v, P_v)$  均满足(4.3.2)和(4.3.3)。其中条件(a)确保网络中除 sink 之外的节点均至少发送一次数据; 条件(b)确保任何一个节点只发送一次数据; 条件(c)确保每个节点均将数据发送给其父节点, 并且在第  $K$  个时隙, 来自所有节点的聚合数据成功发送到 sink 节点。在 **MLDAS-SIC** 问题中, 我们希望寻找一个满足上述(a)、(b)和(c)三个条件的可行调度, 并且该调度的时延  $K$  能达到最小值。关于 **MLDAS-SIC** 问题, 我们有下面的定理说明其复杂度。

**定理 4.1:** 问题 **MLDAS-SIC** 是 **NP**-难的。

证明. 让  $l_v$  表示一条待调度的链路。在 SIC 模型下, 我们须检验链路  $l_u(Q(l_u, l_v) > Q(l_v))$  是否遵从条件(4.3.2)。让式(4.3.2)中的 SINR 解码门限  $\beta$  趋向于无穷大。这种情况下, 来自链路  $l_u$  的干扰信号必然无法在  $l_v$  的接收方  $P_v$  处解码。此时, 只有链路  $l_v$  的目的信号是最强的, 且遵从条件(4.3.3)时,  $l_v$  才能传输成功。也就是说, 此时问题 **MLDAS-SIC** 规约为物理干扰模型下的 **MLDAS** 问题, 而该问题已经被证明为 **NP**-难的<sup>[82]</sup>。因此, 我们研究的新问题, 作为原始 **MLDAS** 问题的更一般化的情形, 也是 **NP**-难的。定理证毕。  $\square$

## 4.4 多包接收模型下的数据聚合调度算法

在本节, 我们给出两个用于解决 **MLDAS-SIC** 问题的具有多项式时间复杂度的启发式调度算法 **RFGS** 和 **SFGS**。我们首先给出链路相对干扰的定义, 然后据此分别给出上述两种调度算法的设计细节, 最后对所提出的调度算法进行可行性和计算复杂度分析。

### 4.4.1 链路相对干扰

为了方便算法的设计与分析, 这里我们给出一些重要的定义。我们定义链路  $l_u$  对链路  $l_v$  造成的相对干扰(Relative Interference, RI)为:

$$RI(l_u, l_v) = \begin{cases} 0 & l_u = l_v \\ \infty & d(l_u, l_v) = 0 \ \&\& l_u \neq l_v \\ c(l_v) \cdot \frac{d^\alpha(l_v)}{d^\alpha(l_u, l_v)} & \text{others} \end{cases}$$

这里  $c(l_v) = \frac{Q\beta}{Q-d^\alpha(l_v)\cdot\beta\xi}$  是一个取值只依赖于链路  $l_v$  的性质和若干常量的系数。RI

也称为影响度(Affectance)，在文献<sup>[83]</sup>中有类似、但是具有不同表达式的定义。让  $\mathcal{S}_t$  表示与  $l_v$  同时调度的链路集合。类似的，我们定义链路集合  $\mathcal{S}_t$  对  $l_v$  造成的相对干扰为：

$$RI(\mathcal{S}_t, l_v) = \sum_{l_u \in \mathcal{S}_t} RI(l_u, l_v) \quad (4.4.1)$$

通过定义相对干扰这个概念，我们可以很容易得到如下结论：

**引理 4.2：** 物理干扰模型下的 SINR 解码条件(4.3.1)等价于  $RI(\mathcal{S}_t, l_v) \leq 1$ 。

在 SIC 模型下，为了解码链路  $l_v$  的信号，我们首先需要在接收节点  $P_v$  处解码来自链路集合  $\mathcal{S}_t$  的较强干扰信号。我们定义这样一条干扰较强的链路为  $l_v$  的关联链路(Associated Link, AL)，并且  $\mathcal{S}_t$  中所有  $l_v$  的关联链路构成了关联链路集合(Associated Link Set, ALS)，表示为：

$$ALS(\mathcal{S}_t, l_v) = \{l_u | l_u \in \mathcal{S}_t, Q(l_u, l_v) > Q(l_v)\} \quad (4.4.2)$$

类似的，我们可以定义链路  $l_u \in ALS(\mathcal{S}_t, l_v)$  的关联链路集合为：

$$ALS(\mathcal{S}_t, l_v, l_u) = \{l_w | l_w \in \mathcal{S}_t, Q(l_w, l_v) > Q(l_u, l_v) > Q(l_v)\} \quad (4.4.3)$$

集合  $ALS(\mathcal{S}_t, l_v, l_u)$  表示接收节点  $P_v$  处比  $l_u$  更强的干扰链路。欲解码  $l_u$ ，须首先解码集合  $ALS(\mathcal{S}_t, l_v, l_u)$  中的所有链路。基于上述定义，我们可以很容易得到如下结论：

**引理 4.3：** 在 SIC 模型下，链路  $l_v$  传输成功的充要条件(4.3.2)和(4.3.3)等价于：任给

$$l_u \in ALS(\mathcal{S}_t, l_v) \cup \{l_v\} \text{，有: } RI(l_u, l_v) \geq \frac{\beta}{\beta+1} \cdot RI(\mathcal{S}_t \setminus ALS(\mathcal{S}_t, l_v, l_u), l_v) + \frac{c(l)-\beta}{\beta+1}。$$

#### 4.4.2 基于随机链路优先的贪心聚合调度算法

在本节，我们给出基于随机链路优先的贪心调度算法，即 Random-link-First Greedy Scheduling (RFGS) 算法。**RFGS** 算法的基本思想是在每个时隙，采用首次适合(First-Fit)策略，随机选择一条链路进行调度。然后基于贪心法，从候选的待调度链路集合中选取一个可行的极大(Maximal)链路子集进行调度。所谓可行，就是说调度任何一条链路不会影响已经调度的链路的正确传输；所谓极大，就是说不可能在不破坏调度集合可行性的前提下，再添加任何一条链路到当前调度链路集合中。图 4.3 给出了 **RFGS** 算法的伪代码。初始时，**RFGS** 算法的待调度链路集合  $\mathcal{L}$  为树  $T$  中所有叶节点(leaf nodes)对应的发送链路。在从集合  $\mathcal{L}$  中随机地选取一条链路  $l_v$  并加入到当前调度链路集合  $\mathcal{S}_t$  后，算法遍历集合  $\mathcal{L} \setminus \mathcal{M}$  中的每一条链路。这里， $\mathcal{M}$  表示经确认不能在当前时隙调度的链路集合。对于  $\mathcal{L} \setminus \mathcal{M}$  中的每一条链路  $l_u$ ，我们采用干扰判定

(Interference Checking)子算法 **RFGS-IC** 来决定  $l_u$  是否能够在当前时隙调度。如果可行，那么  $l_u$  被加入  $\mathcal{S}_t$  中，否则  $l_u$  被加入  $\mathcal{M}$  中。干扰判定算法 **RFGS-IC** 的正确性决定了 **RFGS** 的可行性，在定理 4.4 中，我们将证明该算法的判定方法在 SIC 模型下是正确的。在遍历完集合  $\mathcal{L} \setminus \mathcal{M}$  后，我们将所有已经调度的链路从树中移除，然后更新待调度链路集合  $\mathcal{L}$ ，并将时间  $t$  加 1。通过这种链路遍历操作，我们确保当前时隙已调度链路集合是极大的。重复上述过程，直到树  $T$  中只包含根节点，此时树中所有传感器节点均已经被调度一次，数据聚合操作因此能确保完成。

---

**Algorithm** : 基于随机链路优先的贪心调度算法(**RFGS**)

---

**输入:** 树  $T(\mathcal{V}, \mathcal{E})$ , 叶节点集合  $\mathcal{V}_L$

**输出:** 调度序列  $\{\mathcal{S}_t\}$ ,  $t = 1, 2, \dots$ ,  $\mathcal{S}_t$  包含  $t$  时隙调度的链路集合

```

1:  $t \leftarrow 1, \mathcal{S}_t \leftarrow \emptyset, \mathcal{M} \leftarrow \emptyset;$ 
2:  $\mathcal{L} \leftarrow \{(v, P_v) | v \in \mathcal{V}_L\};$ 
3: while  $\mathcal{V} \neq \{v_s\}$  do
4:   从集合  $\mathcal{L}$  中随机选取一条链路  $l_v$ , 并将其移动到  $\mathcal{S}_t$  中;
5:   for 集合  $\mathcal{L} \setminus \mathcal{M}$  中的每一条链路  $l_u$  do
6:     if RFGS-IC( $\mathcal{S}_t, l_u$ ) == TRUE then
7:        $\mathcal{M} \leftarrow \mathcal{M} \cup \{l_u\};$ 
8:     else
9:        $\mathcal{S}_t \leftarrow \mathcal{S}_t \cup \{l_u\}, \mathcal{L} \leftarrow \mathcal{L} \setminus \{l_u\};$ 
10:    end if
11:   end for
12:    $\mathcal{E} \leftarrow \mathcal{E} \setminus \mathcal{S}_t$ , 更新  $\mathcal{V}$ 、 $\mathcal{V}_L$ ,  $\mathcal{L} \leftarrow \{(v, P_v) | v \in \mathcal{V}_L\}$ ,  $\mathcal{M} \leftarrow \emptyset, t \leftarrow t + 1$ ;
13: end while

```

▷ 干扰判定子算法(**RFGS-IC**)

**输入:** 调度链路集合  $\mathcal{S}$ , 待调度链路  $l_u$

**输出:** 如果链路  $l_u$  破坏集合  $\mathcal{S}$  中的干扰规则，则返回 TRUE；否则，返回 FALSE；

```

1: for 集合  $\mathcal{S} \cup \{l_u\}$  中的每条链路  $l_w$  do
2:   for 集合  $ALS(\mathcal{S}, l_w) \cup \{l_w\}$  中的每条链路  $l_y$  do
3:     if  $RI(l_y, l_w) < \frac{\beta}{\beta+1} \cdot RI(\mathcal{S} \setminus ALS(\mathcal{S}, l_w, l_y), l_u) + \frac{c(l_w)-\beta}{\beta+1}$  then
4:       return TRUE;
5:     end if
6:   end for
7: end for
8: return FALSE;

```

图 4.3 基于随机链路优先策略的贪心调度算法 RFGS 伪代码

#### 4.4.3 基于最短链路优先的贪心聚合调度算法

在本小节，我们提出另外一种调度算法，即 Shortest-link-First Greedy Scheduling (**SFGS**)，即基于最短链路优先的贪心调度算法。与 **RFGS** 随机地选择链路进行调度不同，**SFGS** 算法总是从当前候选链路集合中选择长度最短的链路进行调度。具体来

说，在每个时隙，我们从集合  $\mathcal{L}$  中选取长度最短的链路  $l_v = \arg \min_{l \in \mathcal{L}} |d(l)|$ ，并将其移动到  $\mathcal{S}_t$  中。对于  $\mathcal{L}$  中剩余的每一条待调度链路，我们将采用干扰判定子算法 **SFGS-IC** 以检查其是否会能够在当前时隙调度。观察图 4.4 中 **SFGS-IC** 子算法的伪代码，我们可以发现，有两种情况下，链路  $l_u$  将被认为是无法在当前时隙调度的。情形(1)：链路  $l_u$  对最短长度链路  $l_v$  的相对干扰  $RI(l_u, l_v)$  超过一个门限值  $\lambda \cdot c(l_v)$ ，这里  $\lambda$  是一个精心选择的常数，由  $\alpha, \beta$  来决定(见 **SFGS-IC** 子算法第 1 行)。情形(2)：链路  $l_u$  的接收方由于当前时隙已调度链路造成的干扰，无法在一定程度上解码目的信号。具体来说，如果集合  $ALS(\mathcal{S}, l_u) \cup \{l_u\}$  中的任意一条链路  $l_w$  对  $l_u$  的相对干扰小于一个如 **SFGS-IC** 子算法第 6 行所示的确定值，则链路  $l_u$  被认为无法在当前时隙调度。当前时隙所有经历过干扰判定子算法 **SFGS-IC**，并被认为无法调度的链路构成集合  $\mathcal{M}$ 。在 **定理 4.5** 中，我们将证明上述这种干扰判定策略确保无论何时我们从集合  $\mathcal{L} \setminus \mathcal{M}$  中选取长度最短的一条链路进行调度并加入到  $\mathcal{S}_t$ ，所有  $\mathcal{S}_t$  中的链路在 SIC 模型下均能成功接收。在每个时隙，我们将重复上述步骤，直到候选链路集合  $\mathcal{L}$  中的每条链路要么被调度，要么因无法调度而被移动到  $\mathcal{M}$  中。接下来，我们将所有已经调度的链路  $\mathcal{S}_t$  从树中移除，然后更新待调度链路集合  $\mathcal{L}$ ，并将时间  $t$  加 1。重复上述过程，直到树  $T$  中只包含根节点。此时树中所有的传感器节点均已经被调度一次，数据聚合操作完成。

#### 4.4.4 算法分析

在本小节，我们给出上述两种调度算法 **RFGS** 和 **SFGS** 的理论分析结果。我们将证明两种算法是可行的(feasible)，并且都具有多项式时间计算复杂度。这里，调度称为可行的，当且仅当调度方案中每个时隙内活跃链路的信号在 SIC 模型下均能成功解码。

**定理 4.4：** 在 SIC 模型下，算法 **RFGS** 产生的调度是可行的。

证明. 为了证明 **RFGS** 算法的调度是可行的，我们需要证明每个时隙  $t$  内加入到调度链路集合  $\mathcal{S}_t$  中的每条链路均能成功被解码。在 **RFGS** 算法中，在两种情形下，会有新链路加入到  $\mathcal{S}_t$  中。第一种情形如算法第 4 行所示，我们随机地选择一条链路  $l_v$  加入到  $\mathcal{S}_t$  中。由于此时  $l_v$  是集合  $\mathcal{S}_t$  中唯一的链路，因而其传输必然成功；第二种情形如算法第 9 行所示，我们将经干扰判定子算法 **RFGS-IC** 检验为与  $\mathcal{S}_t$  中所有链路均兼容的链路  $l_u$  加入  $\mathcal{S}_t$  中。根据引理 4.3，**RFGS-IC** 算法实际上是 SIC 模型下链路传输成功充分必要条件的直接表达，因此  $l_u$  的传输也是成功的。综上，所有加入到集合  $\mathcal{S}_t$  的链路在 SIC 模型下均能成功解码，也就是说 **RFGS** 产生的调度是可行的。定理得证。  $\square$

**Algorithm** :基于最短链路优先的贪心调度算法(**SFGS**)

**输入:** 树  $T(\mathcal{V}, \mathcal{E})$ , 叶节点集合  $\mathcal{V}_L$

**输出:** 调度序列  $\{\mathcal{S}_t\}$ ,  $t = 1, 2, \dots$ ,  $\mathcal{S}_t$  包含  $t$  时隙调度的链路集合

```

1:  $t \leftarrow 1, \mathcal{S}_t \leftarrow \emptyset, \mathcal{M} \leftarrow \emptyset;$ 
2:  $\mathcal{L} \leftarrow \{(v, P_v) | v \in \mathcal{V}_L\};$ 
3: while  $\mathcal{V} \neq \{v_s\}$  do
4:   while  $\mathcal{L} \setminus \mathcal{M} \neq \emptyset$  do
5:     从集合  $\mathcal{L}$  中选取一条长度最短的链路  $l_v$ , 并将其移动到  $\mathcal{S}_t$  中;
6:     for 集合  $\mathcal{L} \setminus \mathcal{M}$  中的每一条链路  $l_u$  do
7:       if SFGS-IC( $\mathcal{S}_t, l_u, l_v$ ) == TRUE then
8:          $\mathcal{M} \leftarrow \mathcal{M} \cup \{l_u\};$ 
9:       end if
10:      end for
11:    end while
12:     $\mathcal{E} \leftarrow \mathcal{E} \setminus \mathcal{S}_t$ , 更新  $\mathcal{V}$ 、 $\mathcal{V}_L$ ,  $\mathcal{L} \leftarrow \{(v, P_v) | v \in \mathcal{V}_L\}$ ,  $\mathcal{M} \leftarrow \emptyset, t \leftarrow t + 1$ ;
13: end while

```

▷ 干扰判定子算法(SFGS-IC)

**输入:** 调度链路集合  $\mathcal{S}$ , 待调度链路  $l_u$ , 最短链路  $l_v$

**输出:** 如果链路  $l_u$  破坏集合  $\mathcal{S}$  中的干扰规则, 则返回TRUE; 否则, 返回FALSE;

```

1:  $\lambda \leftarrow \frac{2^{\alpha}-2-1}{72\cdot\beta\cdot2^{\alpha}}$  if  $RI(l_u, l_v) \geq \lambda \cdot c(l_v)$  then
2:   return TRUE
4: else
5:   for 集合  $ALS(\mathcal{S}, l_u) \cup \{l_u\}$  中的每一条链路  $l_w$  do
6:     if  $RI(l_w, l_u) < \frac{3\beta}{2+3\beta} \cdot RI(\mathcal{S} \setminus ALS(\mathcal{S}, l_u, l_w), l_u) + \frac{3c(l_u)-3\beta}{3+3\beta}$  then
7:       return TRUE
8:     end if
9:   end for
10: end if
11: return FALSE

```

图 4.4 基于最短链路优先策略的贪心调度算法 SFGS 伪代码

**定理 4.5:** 在 SIC 模型下, 算法 **SFGS** 产生的调度是可行的。

**证明.** 类似的, 我们需要证明在算法 **SFGS** 中, 所有加入到  $\mathcal{S}_t$  的链路  $l_v$  均能成功传输。让  $\mathcal{P}$  表示早于  $l_v$  加入到  $\mathcal{S}_t$  (也就是长度不大于  $d(l_v)$ ) 的链路集合,  $\mathcal{Q}$  表示迟于  $l_v$  加入到  $\mathcal{S}_t$  (也就是长度不小于  $d(l_v)$ ) 的链路集合。我们有  $\mathcal{S}_t = \mathcal{P} \cup \{l_v\} \cup \mathcal{Q}$ 。对于  $\mathcal{S}_t$  中的每条链路  $l_u$ , 根据干扰检查子算法 **SFGS-IC**, 我们有:

(1) 若  $l_u \in \mathcal{Q}$ , 则我们有  $RI(l_u, l_v) < \lambda \cdot c(l_v)$ ;

(2) 若  $l_u \in ALS(\mathcal{P}, l_v) \cup \{l_v\}$ , 则我们有:  $RI(l_u, l_v) \geq \frac{3\beta}{2+3\beta} \cdot RI(\mathcal{P} \setminus ALS(\mathcal{P}, l_v, l_u), l_v)$

$$+ \frac{3c(l_v)-3\beta}{3+3\beta}$$

根据结论(1)和(2), 我们可以给出集合  $\mathcal{Q}$  中的所有链路对  $l_v$  的相对干扰之和的上界, 即

$RI(\mathcal{Q}, l_v) \leq \frac{c(l_v)}{3\beta}$ , 详细证明略, 读者可以参见文献<sup>[105]</sup>。另外, 根据(1), 我们可知对

于 $\mathcal{Q}$ 中任意链路 $l_u$ , 有 $\frac{d(l_u, l_v)}{d(l_v)} > (\frac{1}{\lambda})^{\frac{1}{\alpha}}$ 。注意到 $\lambda > 1$ 且 $2 < \alpha \leq 6$ , 我们有 $d(l_u, l_v) > d(l_v)$ 。

因此, 集合 $\mathcal{Q}$ 不包含链路 $l_v$ 的关联链路, 也就是说 $ALS(\mathcal{S}_t, l_v) \cap \mathcal{Q} = \emptyset$ , 且 $ALS(\mathcal{S}_t, l_v) = ALS(\mathcal{P}, l_v)$ 。同理, 我们有:  $ALS(\mathcal{S}_t, l_v, l_u) \cap \mathcal{Q} = \emptyset$ , 且 $ALS(\mathcal{S}_t, l_v, l_u) = ALS(\mathcal{P}, l_v, l_u)$ 。让 $l_u$ 表示 $ALS(\mathcal{S}_t, l_v)$ 中的任何一条链路, 根据(2), 我们有:

$$\begin{aligned} (2 + 3\beta)RI(l_u, l_v) &\geq 3\beta \cdot RI(\mathcal{P} \setminus ALS(\mathcal{P}, l_v, l_u), l_v) + 3c(l_v) - 3\beta \\ &= 3\beta \cdot RI(\mathcal{S}_t \setminus (\mathcal{Q} \cup \{l\}) \setminus ALS(\mathcal{S}_t, l_v, l_u), l_v) + 3c(l_v) - 3\beta \\ &= 3\beta \cdot [RI(\mathcal{S}_t \setminus ALS(\mathcal{S}_t, l_v, l_u), l_v) - RI(\mathcal{Q}, l_v)] + 3c(l_v) - 3\beta \\ &\geq 3\beta \cdot RI(\mathcal{S}_t \setminus ALS(\mathcal{S}_t, l_v, l_u), l_v) + 3c(l_v) - 3\beta - c(l_v) \\ &> 3\beta \cdot RI(\mathcal{S}_t \setminus ALS(\mathcal{S}_t, l_v, l_u), l_v) + 3c(l_v) - 3\beta - RI(l_u, l_v) \end{aligned}$$

上式最后一个不等式成立是由于 $RI(l_u, l_v) = c(l_v) \cdot \frac{d^\alpha(l_v)}{d^\alpha(l_u, l_v)} > c(l_v)$ 。因此, 我们有

$RI(l_u, l_v) > \frac{\beta}{\beta+1} \cdot RI(\mathcal{S}_t \setminus ALS(\mathcal{S}_t, l_v, l_u), l_v) + \frac{c(l_v) - \beta}{\beta+1}$ , 即引理 4.3 中的 SIC 解码条件得到满足。因此, 在 SIC 模型下, 链路 $l_v$ 能够传输成功。定理得证。  $\square$

**定理 4.6:** 算法 **RFGS** 和 **SFGS** 具有多项式时间计算复杂度。

证明. 首先, 我们分析 **RFGS** 算法的计算复杂度。在 **RFGS** 算法中, 为了检验链路 $l_u$ 是否会破坏已调度链路集合 $\mathcal{S}$ 的成功传输, 需要消耗子算法 **RFGS-IC** 最多 $\mathcal{O}(|\mathcal{S}| \cdot |ALS(\mathcal{S}, l_u)|) \leq \mathcal{O}(|\mathcal{S}|^2)$ 的运行时间(running time)。另一方面, **RFGS** 算法中的 *While*-循环和 *For*-循环在最坏情况下都各自需要花费 $\mathcal{O}(|\mathcal{L}|)$ 的运行时间。因此算法 **RFGS** 总的时间复杂度在最坏情形下为 $\mathcal{O}(|\mathcal{L}|^2 \cdot |\mathcal{S}|^2) \leq \mathcal{O}(|\mathcal{L}|^4) \leq \mathcal{O}(|\mathcal{E}|^4)$ 。该复杂度为多项式时间复杂度。接下来, 我们分析算法 **SFGS** 的计算复杂度。类似的, 我们可以发现干扰判定算法 **SFGS-IC** 的复杂度为 $\mathcal{O}(|\mathcal{S}|)$ 。除此之外, **SFGS** 中的三个循环(2个 *While*-循环和 1 个 *For*-循环)各自需要消耗的运行时间为 $\mathcal{O}(|\mathcal{L}|)$ 。因此, **SFGS** 算法总的计算复杂度为 $\mathcal{O}(|\mathcal{L}|^3 \cdot |\mathcal{S}|) \leq \mathcal{O}(|\mathcal{L}|^4) \leq \mathcal{O}(|\mathcal{E}|^4)$ , 也属于多项式时间复杂度。定理得证。  $\square$

## 4.5 算法性能评价

### 4.5.1 实验场景设置

在本节, 我们通过 MATLAB 仿真实验评价 **RFGS** 和 **SFGS** 的性能。我们与文献中的 **IAS**<sup>[57]</sup> 算法和文献<sup>[89]</sup>中的 **Cell-AS** 算法进行性能对比。这两种算法是当前物理干扰模型下有代表性的算法, 但是均没有假设物理层具有 SIC 能力。在实验中, 我们把

$n$ 个节点布设在大小为 $h \times h$ 的网络监测区域中。考虑两种网络拓扑形式，即均匀网络(uniform networks)和分簇网络(clustered networks)。在均匀网络中，节点在监测区域中的位置坐标服从均匀随机分布。在分簇网络中， $n_C$ 个簇的中心节点(即簇头)随机均匀地分布在平面上；同时在每个簇中，成员节点均匀随机地分布在以簇头节点为中心、半径为 $r_C$ 的圆盘上。均匀网络有利于数学建模与分析，在 WSNs 学术研究中广泛采用；分簇网络更有利于描述实际的、尤其是具有异构节点密度的网络场景。图 4.5 给出了包含 200 个节点的均匀和分簇网络拓扑的样本示例。在接下来的实验中，我们将簇的半径 $r_C$ 设置为节点最大传输半径 $r$ 。当某个节点同时处于多个簇的范围时，考虑能量有效性，其选择加入距离自己最近的簇。在这两种网络场景中，我们均将随机生成的、以 sink 节点 $v_s$ 为根节点的广度优先搜索(BFS)树作为数据聚合的路由。为了科学起见，对于每次实验，我们重复进行 50 次，并比较各个算法实验结果的平均值和标准差。

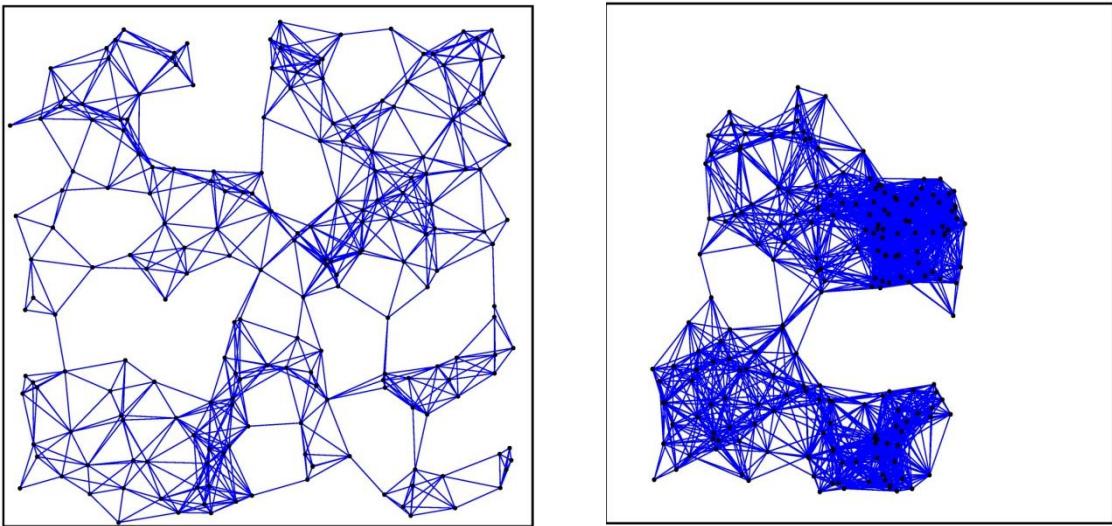
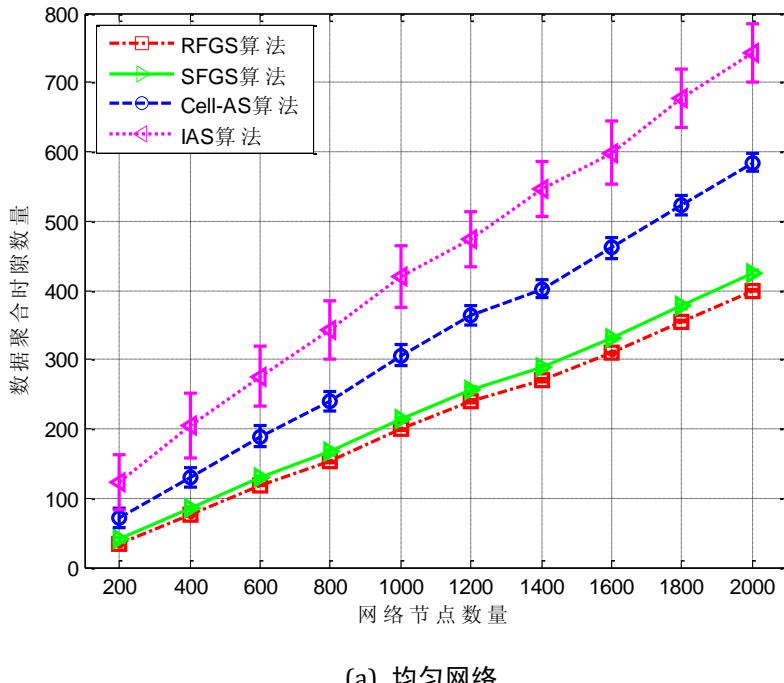


图 4.5 包含 200 个节点的网络样本示意图：(a)均匀网络 (b)分簇网络

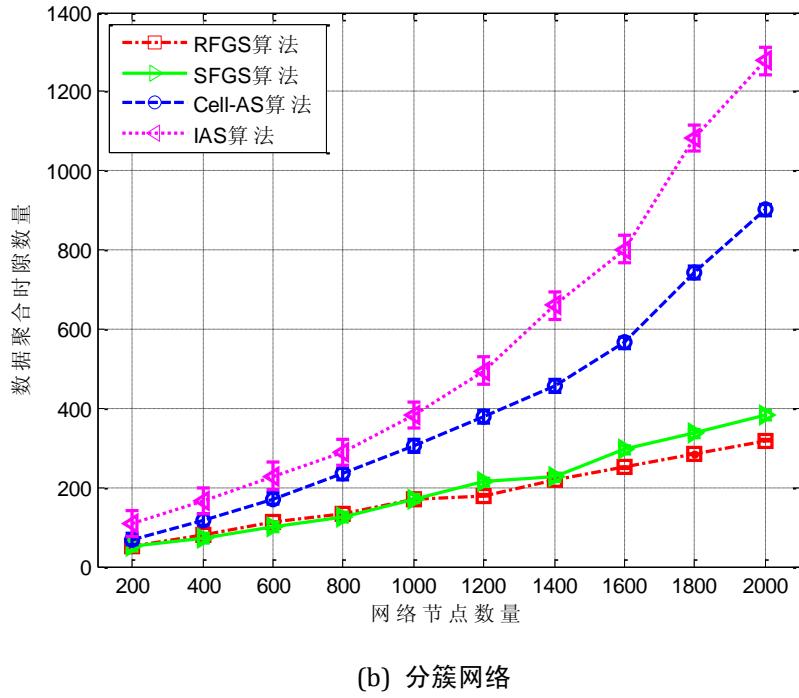
#### 4.5.2 网络规模对算法性能的影响

我们首先研究网络节点数量对数据聚合时延的影响。与文献<sup>[57]</sup>一致，我们将节点发射功率 $P$ ，路径损耗指数 $\alpha$ ，解码门限 $\beta$ 和背景噪声强度 $\xi$ 分别设置为 15, 2.5, 1.0 和 0.1。给定上述参数，我们可以计算出节点最大传输半径 $r=(\frac{P}{\xi\beta})^{\frac{1}{\alpha}}=7.4$ 。我们将区域半径 $h$ 固定为 100m，并以步长 200 逐步将节点数量 $n$ 从 200 增大到 2000。对于分簇网络，我们将簇半径 $r_C$ 设置为与 $r$ 相等的值 7.4，并将簇数量 $n_C$ 设置为 30。图 4.6(a)和图 4.6(b)分别给出了均匀网络和分簇网络下的仿真结果。我们可以发现，**RFGS** 和 **SFGS** 的性能均比 **IAS** 和 **Cell-AS** 要好。具体来说，在均匀网络中，**IAS** 的时延和 **Cell-AS** 的时延分别比 **SFGS** 时延平均高 30% 和 50% 左右。而在分簇网络中，**IAS** 的时延和

**Cell-AS** 的时延分别是 **SFGS** 时延的 3.6 倍和 4.7 倍。也就是说，在分簇网络中，我们的算法性能相比 **IAS** 和 **Cell-AS** 的改进程度要远大于在均匀网络中的情形。这是由于，在均匀网络中，不同链路的长度相差不大，因而从 **SIC** 中获得的并发传输增益不够显著。而在分簇网络，链路长度的多样性更为明显，因此调度算法可以从 **SIC** 中获得更多的并发传输增益。另外，观察图 4.6(b)，我们可以发现，随着节点密度的增加，**RFGS** 和 **SFGS** 的相对性能提升程度变得越来越好。这说明 **RFGS** 和 **SFGS** 能够同时调度更多簇中的链路，但是 **Cell-AS** 和 **IAS** 无法做到这一点。而在均匀网络中，如图 4.6(a) 所示，当节点密度增加时，**RFGS** 和 **SFGS** 的相对性能提升程度并无明显变化。这是由于在均匀网络中，尽管链路密度增大了，但是链路长度的多样性并无明显增加，因而来自 **SIC** 的并发链路增益也无明显提高。对于同时从 **SIC** 中获得增益的两种算法 **RFGS** 和 **SFGS**，我们发现无论是在均匀网络还是分簇网络中，其性能都无明显差异。也就是说，其从 **SIC** 中获得的并发链路的增益是相当的。



(a) 均匀网络

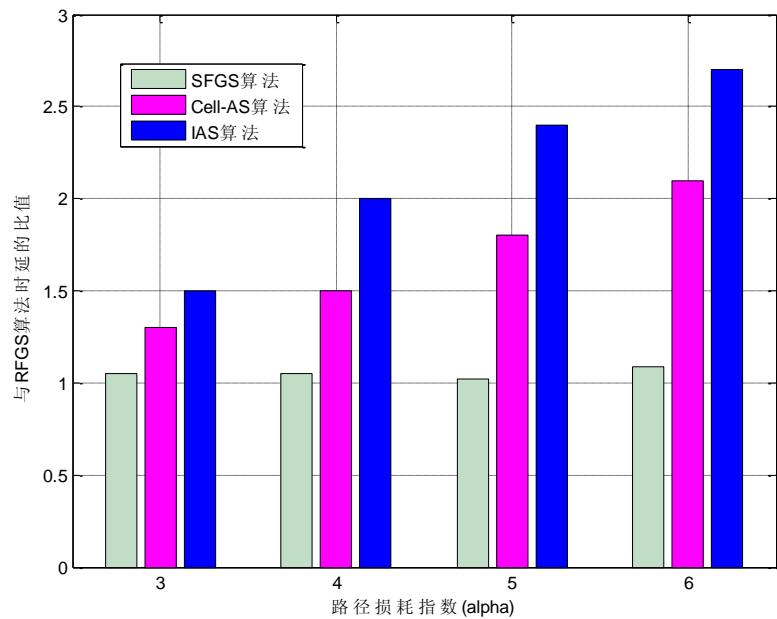


(b) 分簇网络

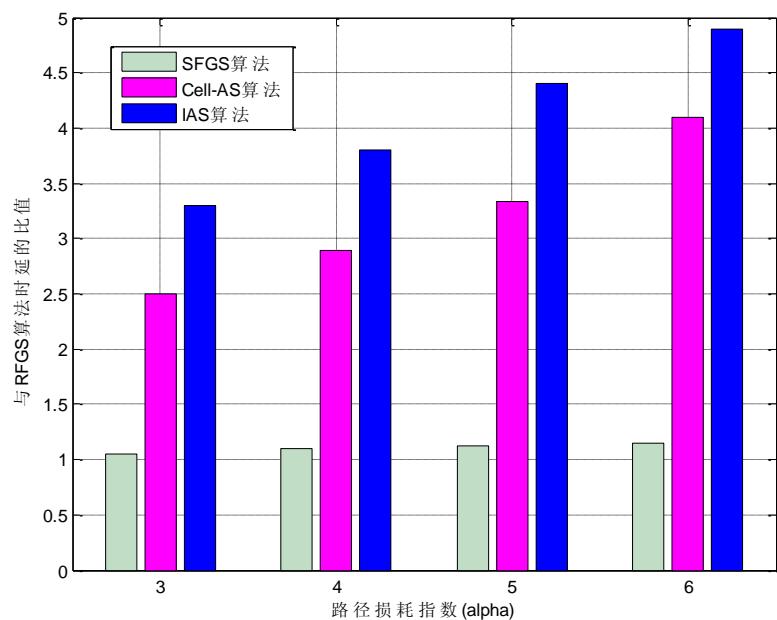
图 4.6 节点数量变化时，调度算法的时延性能比较

#### 4.5.3 SINR 参数对算法性能的影响

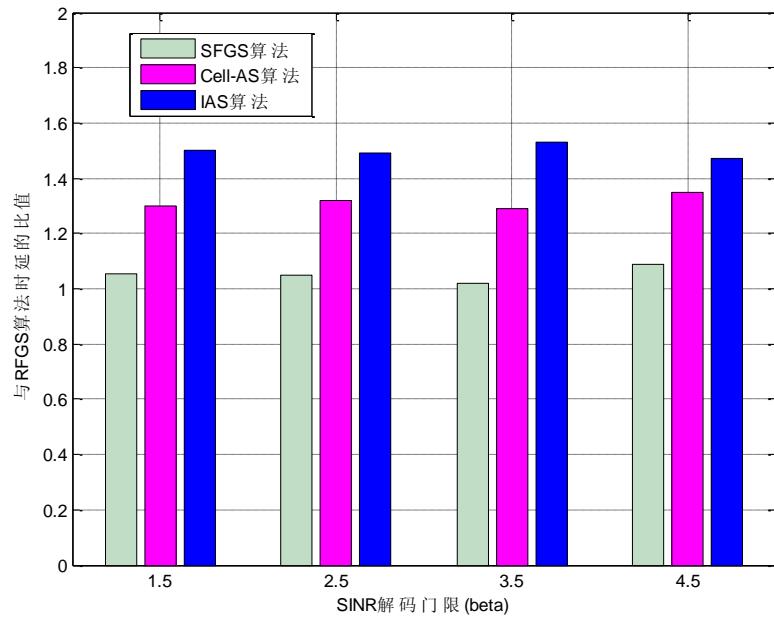
接下来，我们评价 SINR 参数  $\alpha$  和  $\beta$  对数据聚合时延的影响。我们将  $\xi$ ,  $P$  和  $h$  分别设置为 0.1, 15 和 100，并且考虑包含 1000 个节点的、密度适中的网络。对于分簇网络，我们将  $n_C$  设置为 30，并让  $r_C = r$ 。注意到  $r_C$  的值将随着  $\alpha$  和  $\beta$  的变化而变化。图 4.7(a) 和图 4.7(b) 给出了当路径衰减指数  $\alpha$  变化时，各种算法数据聚合时延的性能对比。这里，我们以算法 **RFGS** 为基准，比较其他三种算法，即 **SFGS**、**IAS** 和 **Cell-AS** 的时延与 **RFGS** 时延的比值。我们观察到，无论是在均匀网络还是在分簇网络，随着  $\alpha$  的增加，**IAS** 和 **Cell-AS** 的时延与 **SFGS** 时延的比值逐渐增加，并且都大于 1。这表明，**RFGS** 的相对性能提升程度随着  $\alpha$  的增加变得更加高。这是由于，当  $\alpha$  增大时，各条链路的接收功率之间的差异将更为显著，也就等价于链路多样性增加。因此，来自 SIC 的并发链路增益将更为显著。另外，图 4.7(c) 和图 4.7(d) 给出了当 SINR 解码门限  $\beta$  变化时，各种算法数据聚合时延的性能对比。我们可以发现，无论是在均匀网络还是在分区网络中， $\beta$  的值对各个算法时延的影响并不显著。这是由于  $\beta$  只是一个比值，其值的改变不会对调度算法的性能产生实质影响。



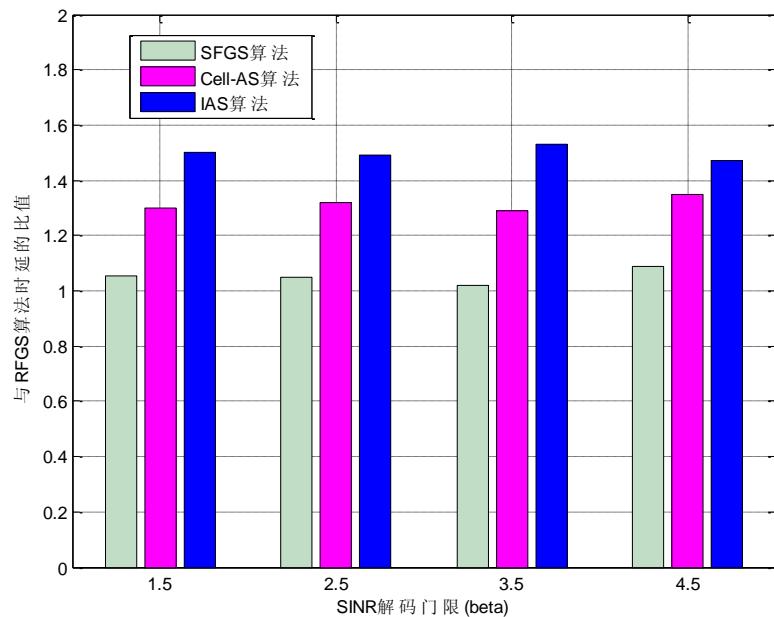
(a) 均匀网络



(b) 分簇网络



(c) 均匀网络



(d) 分簇网络

图 4.7 SINR 参数 $\alpha, \beta$ 变化时, 调度算法的时延性能比较

#### 4.5.4 簇的数量对算法性能的影响

从之前的实验结果中, 我们可以发现在分簇网络中, 来自 SIC 的并发传输机会较多, **RFGS** 和 **SFGS** 算法的性能也更好。然而我们一直将簇数量  $n_C$  设置为一个中等大小的值 30。由于  $n_C$  会影响分簇网络的几何特性, 从而也会影响聚合调度算法的性能,

因此在本节中，我们将研究簇数量 $n_C$ 对数据聚合时延的影响。注意到**RFGS**和**SFGS**性能接近，因此这里我们选择**RFGS**来进行对比。考虑具有1000个节点的网络，并将参数 $\alpha$ ， $\beta$ ， $\xi$ 和 $P$ 分别设置为2.5，1.0，0.1和15。图4.8给出了当 $n_C$ 变化时，三种调度算法的时延变化示意图。注意到为了保证网络的连通性， $n_C$ 的值不能取得太小。从图4.8可知，当网络中只有10个簇时，所有算法的时延性能均最差。这是因为簇的数量太少，导致大量的链路拥挤在一起，阻止了链路的并发传输。当 $n_C$ 增大时，各个算法的时延性能逐步改善。由于SIC带来的增益，**RFGS**的时延比**IAS**和**Cell-AS**更低。具体来说，当 $n_C=40$ 时，**IAS**和**Cell-AS**的时延分别是**RFGS**的2.6倍和2.1倍。我们同样也观察到，当 $n_C$ 变得很大，例如超过90时，各个算法的时延变得越来越接近。也就是说，来自SIC的并发传输增益将变得比较小。这是由于，当 $n_C$ 变得很大时，分簇网络将接近于均匀网络，因而链路多样性将不够显著，来自SIC的增益也就不是很显著了。

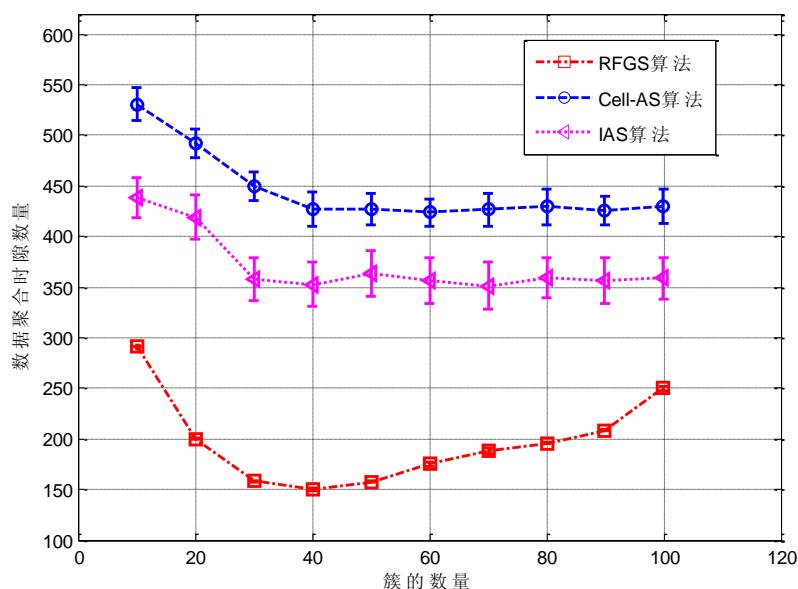
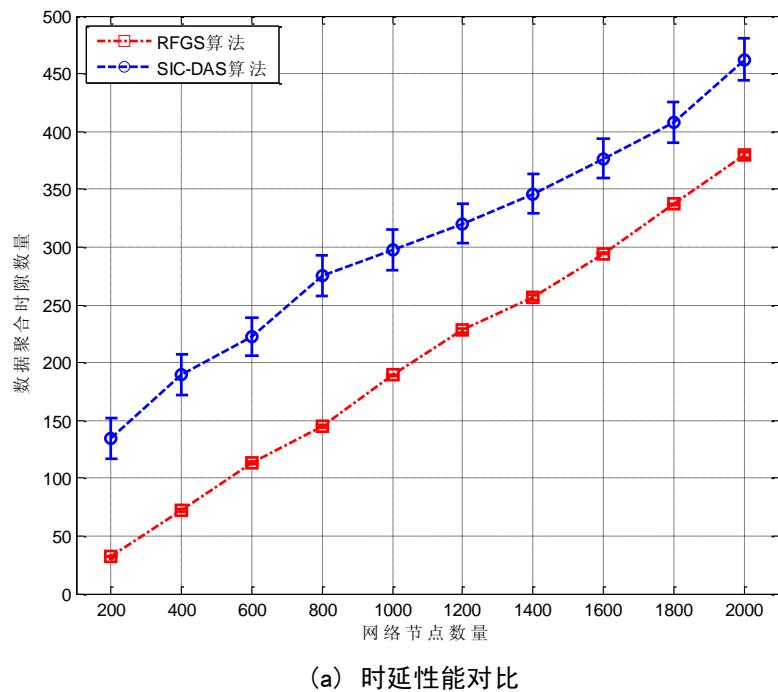


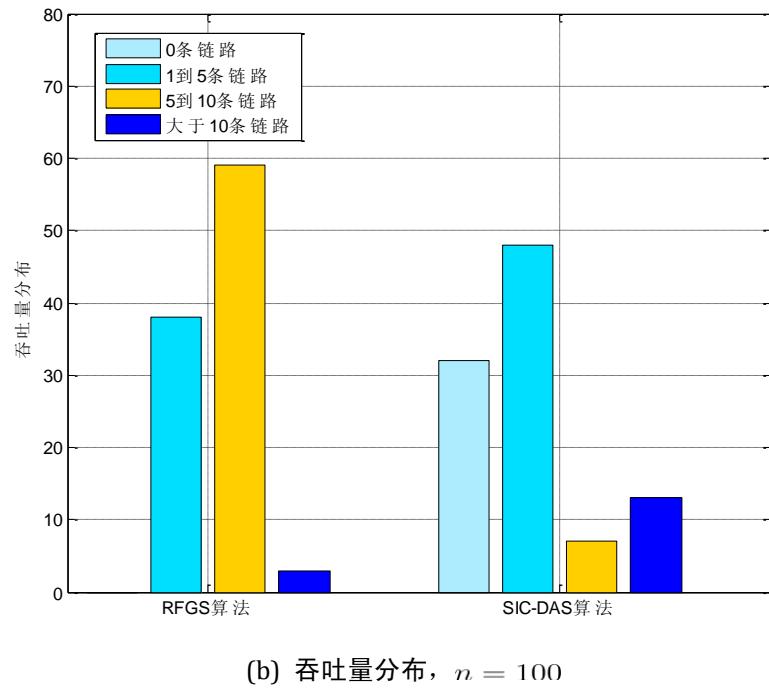
图4.8 分簇网络中的簇数量变化时，调度算法的时延性能比较

#### 4.5.5 与文献<sup>[104]</sup>进行对比

最后，我们与最近发表的一篇文献<sup>[104]</sup>中的**SIC-DAS**算法进行性能对比。由于**RFGS**和**SFGS**性能接近，因此选择**RFGS**为代表来与**SIC-DAS**算法进行对比。**SIC-DAS**算法也假设物理层具有SIC能力，并且也是面向数据聚合调度问题设计的。根据**SIC-DAS**的需要，我们产生这样一种网络拓扑，该拓扑中任意两个节点之间的距离均不小于1m。另外，参数 $\alpha$ ， $\beta$ ， $\xi$ ， $P$ 和 $h$ 分别设置为2.5，1.0，0.1，15和100。对于**RFGS**来说， $P$ 是所有节点都采用的均匀发射功率；对于**SIC-DAS**来说， $P$ 是节点在功率控制下能够采用的最大发射功率。图4.9(a)给出了当网络节点数量 $n$ 从200

变化到 2000 时, 两种算法的时延曲线。我们可以发现, **RFGS** 的时延均比 **SIC-DAS** 的时延小很多。这是由于, **SIC-DAS** 算法尽管也利用了 SIC 带来的链路并发机会, 但是其仅仅在数据聚合的第一阶段, 即局部头节点聚合来自邻居节点的数据时才利用了 SIC 能力。而在后续的调度阶段, 即局部头结点向骨干节点、骨干节点向 sink 节点发送数据时, 仍然使用的是传统的基于干扰避免的调度策略<sup>[104]</sup>。在本文调度算法 **RFGS** 中, 每个调度阶段均充分利用了 SIC 带来的链路并发机会。进一步的, 我们探索了调度算法 **RFGS** 和 **SIC-DAS** 的细节。考虑 1000 个节点的网络, 图 4.9(b)给出了此时 **RFGS** 和 **SIC-DAS** 的吞吐量分布对比。这里, 吞吐量分布(throughput distribution)指的是在整个调度周期, 单个时隙内同时传输的链路数量的分布。我们注意到, **SIC-DAS** 算法中, 10 条或以上链路同时传输的时隙数量相比 **RFGS** 更多。然而, 在绝大多数情况, **RFGS** 在单个时隙内能够调度 5~10 条链路, 而 **SIC-DAS** 只能调度 1~5 条链路。并且, 我们可以发现, 在 **SIC-DAS** 中, 很多情况下, 单个时隙内调度的时隙数量为 0。而 **RFGS** 在任意一个时隙, 至少能够调度 1 条链路。公平地讲, 虽然 **SIC-DAS** 算法的时延性能相比 **RFGS** 并不突出, 但是其能够分布式的实现, 而 **RFGS** 不能。对于网络拓扑动态变化, 例如新节点加入、旧节点失效较为频繁的情形, **SIC-DAS** 算法更为适用。在下一步的研究中, 我们希望能够结合 **RFGS** 和 **SIC-DAS** 各自的优势, 寻找一种既能够分布式地实现, 又能够有较好时延性能, 且支持 SIC 的数据聚合调度新算法。



(b) 吞吐量分布,  $n = 100$ 图 4.9 RFGS 算法与文献<sup>[104]</sup>的性能对比

#### 4.6 本章小结

在本节，我们研究了物理层具备相继干扰消除(SIC)能力时树状无线传感器网络中的最小时延数据聚合调度问题。我们描述了该问题的数学含义，并证明了其为 **NP**-难的。我们提出了两种启发式的调度算法，即基于随机链路优先策略的贪心调度算法 **RFGS** 和基于最短链路优先策略的贪心调度算法 **SFGS**，以充分利用 SIC 带来的链路并发传输机会。通过数学分析，我们证明了两种调度算法可行并且均具有多项式时间计算复杂度。通过大量的仿真实验，并通过与传统的基于干扰避免策略的调度算法进行对比，我们发现 **RFGS** 和 **SFGS** 确实能够显著改善无线传感器网络数据聚合的时延性能。本节的研究工作揭示了 SIC 在调度算法设计时的潜力。也就是说，当物理层具有 SIC 能力时，通过精心设计相应的 MAC 层和网络层算法/协议，无线网络的性能有望得到显著提升。

## 第5章 FODAP：面向低占空比无线传感器网络的错误容忍数据聚合协议

### 5.1 引言

无线传感器网络的节点一般是电池供电的，其能量不仅受限而且难以补充。网络的生存周期严重地受到节点能量的制约。近年来，为了延长无线传感器网络的生命周期，低占空比技术(low-duty-cycle technique)被广泛采用<sup>[80, 106]</sup>。所谓低占空比技术，就是让传感器节点以周期轮转的模式进行工作，并且在每个周期的绝大部分时间彻底休眠，只在极短时间内进行数据收发。由于每个节点的生命周期均得到了延长，整个网络的生存性能也就得到了提升<sup>[107]</sup>。本章研究低占空比无线传感器网络中的数据聚合协议。在前面几章中，我们假设链路质量总是可靠的，在此理想情形下进行调度算法的设计。事实上，在实际系统中，传感器网络的布设环境一般较为复杂。这种复杂性既体现在地理环境的复杂上，如障碍、遮挡等，也体现在电磁环境的复杂上<sup>[108]</sup>。此时，无线链路的传输并不总是可靠的，即网络中存在丢包的情况发生<sup>[109-111]</sup>。相比传统的业务模式，数据聚合中的丢包将造成更为严重的后果。这是因为在数据聚合中，网络中传输的数据包通常是融合了多个数据包信息的聚合包(aggregated packet)，其信息含量(amount of information)较大。我们将聚合包蕴含的信息含量定义为数据聚合的信息质量(Quality-of-Information, QoI)。这里的信息质量 QoI 是一个比较抽象的概念。在不同的应用场景中，QoI 可能会有不同的具体含义<sup>[35]</sup>。

当网络中出现由传输错误导致的数据包丢失时，传统数据聚合调度协议的性能将急剧降低。这种性能的衰退体现在 sink 节点最终收到的聚合包中的有效信息量大大减少。为了克服网络丢包对数据聚合性能的损害，迫切需要设计一种错误容忍的数据聚合协议。本文针对低占空比且链路不可靠的无线传感器网络，设计了一种错误容忍数据聚合协议(Fault-Tolerance Data Aggregation Protocol, FODAP)。FODAP 的设计目的是改善链路传输错误(又称链路损耗)情况下的数据聚合信息质量。为了实现这一目的，FODAP 利用了链路可靠性增强技术。根据采用的链路可靠性增强技术的不同，我们为 FODAP 设计了三个变种，分别为基于动态睡眠调度调整(Dynamic Sleep Scheduling Adjustment)策略的错误容忍数据聚合协议 **FODAP-D**，基于机会路由(Opportunistic Routing)策略的错误容忍数据聚合协议 **FODAP-R** 和基于混合(Hybrid)策略的错误容忍数据聚合协议 **FODAP-H**。上述三种错误容忍数据聚合协议具有各自不同的特点。仿真实验表明，在链路不可靠的网络场景下，相比文献中传统的未采用链路可靠性增强技术的数据聚合协议，本章提出的上述三种协议均能够提升数据聚合中 sink 节点获取的信息质量 QoI，其中基于混合策略的 **FODAP-H** 较另外两种协议对

QoI 性能的提升更为显著。

## 5.2 相关工作

近年来，低占空比无线传感器网络受到越来越多研究者的关注。文献<sup>[112]</sup>综述了学术界面向低占空比无线传感器网络组播通信(group communication)所提出的算法和协议。文献<sup>[113]</sup>则对低占空比网络中的路由协议和技术进行了总结与归纳。在许多实际布设的传感器网络系统中，低占空比技术成为延长网络生命周期不可或缺的关键手段<sup>[114-116]</sup>。针对低占空比网络中的媒介接入控制问题，文献<sup>[108]</sup>提出了 X-MAC 协议，该协议采用包长较短的序言包(preamble)，并保留了低功耗侦听(low-power listening)技术的若干优点，包括低功耗的通信模式、协议简洁性和与接收方/发送方睡眠调度的去耦合(decoupled)。文献<sup>[107]</sup>提出了一种动态的、流量感知(traffic-aware)的 MAC 协议，以进一步地减少低占空比网络的能量消耗。文献<sup>[106]</sup>研究了低占空比网络中的数据广播问题，并提出了一种启发式的广播调度算法。以最小化广播能量消耗为目标，文献<sup>[117]</sup>提出了具有近似性能保证的数据广播树构造和广播调度算法。以最小化广播时延为设计目标，文献<sup>[118]</sup>提出了具有常量近似比的广播调度算法。

近年来，低占空比无线传感器网络中的数据聚合也吸引了越来越多研究者的目光。文献<sup>[47, 119]</sup>研究了低占空比无线传感器网络中的最小时延数据聚合调度问题。考虑协议干扰模型，文献<sup>[119]</sup>证明了该问题是 **NP**-难的，并基于贪心策略提出了一种集中式的聚合调度方法，同时基于网络分区与迁移策略提出了一种分布式的聚合调度策略。针对异步低占空比传感器网络，并且考虑应用具有端到端时延约束的情形，文献<sup>[120]</sup>提出了一种生命周期平衡的数据聚合机制 LBA。为了以分布式方式实现上述目标，LBA 自适应地调整相邻节点的聚合等待时间(aggregation holding time)，从而平衡节点级的生命周期。为了进一步提高网络生命周期，文献<sup>[121]</sup>提出了一种基于联合低占空比和网络编码技术的数据聚合机制。该机制有效地结合了网络编码所带来的传输负载降低的优势。理论分析和仿真实验表明，该机制不仅能够延长网络生命周期，而且能够改善网络的包传输成功率和平均时延。

## 5.3 系统模型

### 5.3.1 低占空比模型

在低占空比传感器网络中，节点周期性地工作，并在一个周期内的大部分时间休眠。具体来说，时间被划分为连续的、重复出现的周期(period)，长度为  $W$ ，每个周期又被划分为连续的时隙(slot)。图 5.1(a)所示的调度周期长度  $W = 10$ 。单个时隙的长度被设定为不小于数据传输的往返时间(round-trip time)，即数据包(data packet)的传输时间和确认包(ACK packet)的传输时间之和。图 5.1(b)给出了单个时隙的结构图。

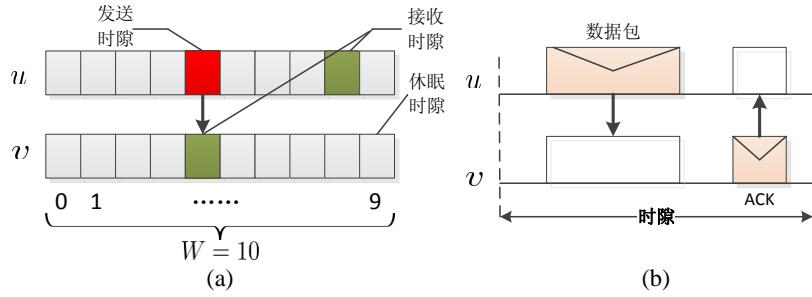


图 5.1 低占空比模型示意图: (a)单个周期内节点 $u$ 和 $v$ 的睡眠调度示意图 (b)单个时隙的结构图

在每个时隙, 节点执行**睡眠调度**(sleep scheduling)。睡眠调度决定节点在每个时隙是处于接收状态(receiving state)还是休眠状态(dominant state)。在接收状态, 节点开启射频模块, 并感知信道环境, 以准备从邻居节点接收数据。在休眠状态, 节点彻底关闭收发器, 只保留一个定时器, 以实现低功耗操作。在低占空比网络中, 节点在大部分时隙休眠。对于节点 $v$ , 其睡眠调度 $\Phi_v$ 表示在每个周期内, 节点处于接收状态的时隙序号集合, 我们有 $\Phi_v \subseteq \{0, 1, \dots, W - 1\}$ 。例如, 在图 5.1(a)中, 节点 $u$ 的睡眠调度 $\Phi_u = \{8\}$ , 节点 $v$ 的睡眠调度 $\Phi_v = \{4\}$ 。

我们假设节点之间保持时间同步, 并且为了保证相邻节点之间的正常通信, 每个节点均知道其一跳邻居节点(one-hop neighbours)的睡眠调度。节点与邻居节点之间的数据收发是由**MAC 调度**(MAC scheduling)决定的。由于睡眠调度的约束, 节点只有处于接收状态时才能接收来自邻居节点的数据。因此, 在 MAC 调度中, 为了保证发送方 $u$ 到接收方 $v$ 的成功传输,  $u$ 只能在 $v$ 处于接收状态时发送数据。如图 5.1(a)所示,  $u$ 欲向 $v$ 发送数据, 只能在每个周期的 4 号时隙进行。在低占空比网络中, 每个节点还有可能出于发送状态(transmitting state)。由上述讨论可知, 节点的发送状态是由 MAC 调度决定的, 并且要受到接收方的睡眠调度的约束。

### 5.3.2 网络模型

我们依然考虑数据聚合这种业务模式。网络中的节点集合表示为 $\mathcal{V}$ , 其中 $v_s \in \mathcal{V}$ 为 sink 节点, 其余为传感器节点。数据聚合按照连续的轮次进行。在每一轮,  $v_s$ 负责收集来自其他 $|\mathcal{V}| - 1$ 个节点的聚合数据。采用树形结构 $T$ 作为数据聚合的路由, 树的根节点为 $v_s$ 。图 5.2(a)给出了一颗数据聚合树的示意图。节点 $v$ 接收来自子节点集合 $\mathcal{C}(v)$ 的数据, 并用自己的数据包进行数据融合, 然后将融合后的包发送给其父节点 $P_v$ 。在低占空比模型下, 节点 $v$ 必须知道其父节点的睡眠调度。图 5.2(b)给出了树中每个节点的父节点睡眠调度。例如, 节点 $b$ 的睡眠调度为 $\{2, 6\}$ , 因此 $b$ 的子节点 $d$ 和 $e$ 只能在每个周期的 2 号时隙和 6 号时隙才能向 $b$ 发送数据。在本地, 节点需要存储的信息至少包括: 父节点 ID, 子节点 ID 集合, 父节点睡眠调度等。

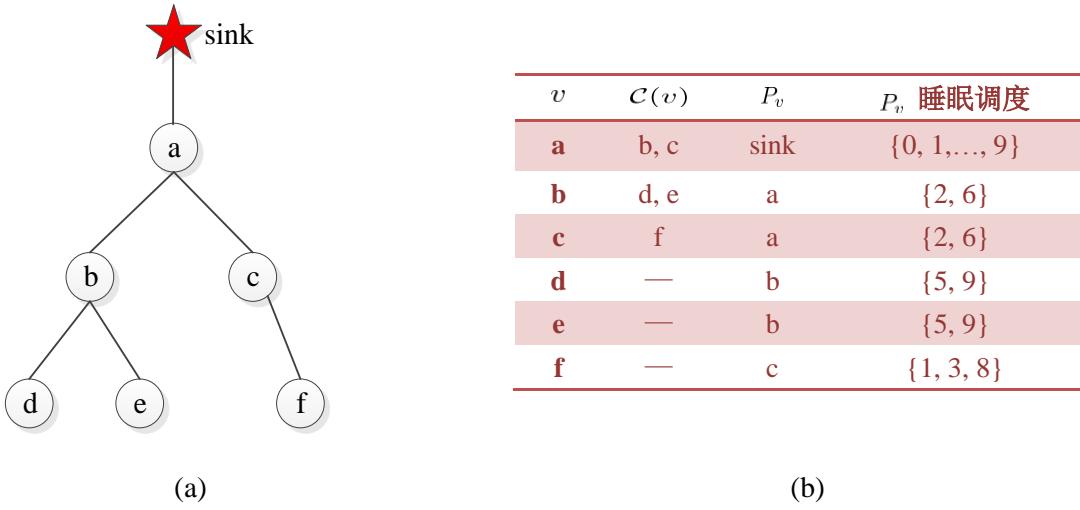


图 5.2 (a) 数据聚合拓扑示意图 (b) 每个节点本地存储信息列表(部分)

### 5.3.3 链路可靠性与数据聚合质量模型

在实际无线系统中，链路传输由于受到环境因素和信道状况的影响，会存在丢包的情况。在网络研究中，采用概率网络模型(Probabilistic Network Model)<sup>[122]</sup>来描述网络链路的可靠性。给定链路 $(u, v)$ ，用 $\rho(u, v) \in [0, 1]$ 表示其丢包概率。因此，链路 $(u, v)$ 传输的成功概率 $\eta(u, v)$ 为 $1 - \rho(u, v)$ 。我们考虑链路丢包对数据聚合性能的影响。在数据聚合中，sink 节点收到的聚合包中包含参与了数据聚合过程的各个数据包的信息。在同构无线传感器网络模型下，我们用成功地参与了数据聚合过程的节点个数来衡量 sink 收到的聚合包的质量，即数据聚合信息质量(Quality-of-Information, QoI)。如果不存在链路损耗，即所有链路都是 100% 可靠的，则数据聚合 QoI 等于 $|\mathcal{V}| - 1$ ；当链路存在损耗时，数据聚合质量将呈现一定的概率分布。给定树 $T$ 和树中每条链路 $(v, P_v)$ 的损耗概率 $\rho(v, P_v)$ ，数据聚合 QoI 的平均值(期望值) $\mathbb{E}(QoI)$ 为：

$$\mathbb{E}(QoI) = \sum_{v \in \mathcal{V} \setminus \{v_s\}} \prod_{u \in PATH_v} (1 - \rho(u, P_u)). \quad (5.3.1)$$

这里 $PATH_v = \{v, P_v, \dots\}$ 包含节点 $v$ 到 sink 节点 $v_s$ 之间的路径上的所有节点，且我们规定 $v_s \notin PATH_v$ 。由式(5.3.1)可知，每条链路的传输损耗概率都直接影响着数据聚合的期望 QoI。但是，不同链路的损耗对总的期望 QoI 的影响程度是不一样的。以图 5.3 为例说明。在该图中，我们给出了图 5.2(a)所示网络中，当链路 $(b, a)$ 和链路 $(f, c)$ 是不可靠链路，并且各自的链路损耗概率变化时，sink 节点收到的聚合包的期望信息质量变化趋势图。由图 5.3 可知，链路 $(b, a)$ 的损耗对 QoI 的影响程度更大。这是因为链路 $(b, a)$ 传输的数据包的信息质量更高，从而该链路丢包所造成的信息质量损失也就更为严重。

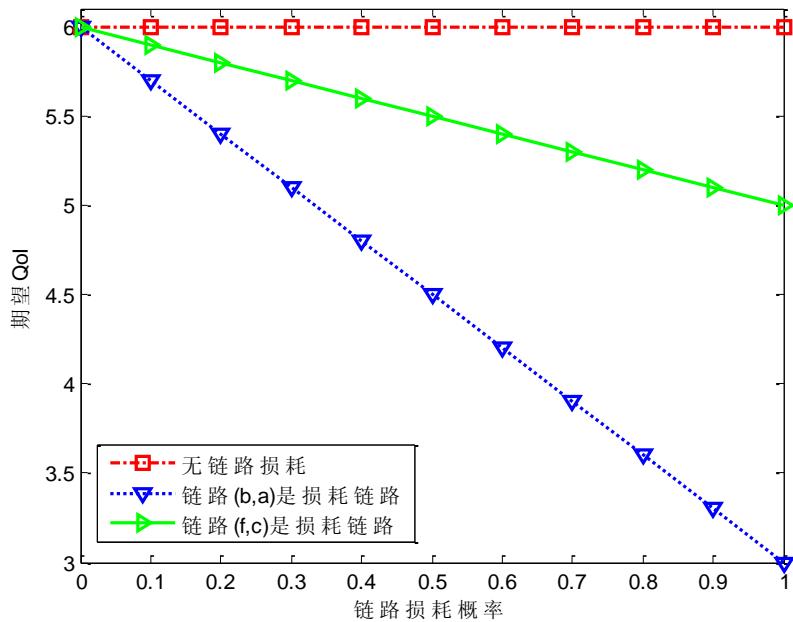


图 5.3 当图 5.2(a)中的部分链路损耗时, sink 节点的期望 QoI 对比

### 5.3.4 设计目标

由上一节的分析可知, 链路的传输损耗会降低无线传感器网络数据聚合的信息质量。为了克服链路传输损耗的不利影响, 我们需要设计传输错误(transmission failure)容忍的数据聚合协议。当链路质量不可靠时, 协议需要采用一定的链路质量增强机制, 来提高聚合包的传输成功概率, 从而提高无线传感器网络数据聚合的信息质量。本章的目标正是为低占空比且链路不可靠的无线传感器网络设计这样一种错误容忍的数据聚合协议。

## 5.4 设计方案

在本节, 我们给出错误容忍数据聚合协议(Fault-Tolerance Data Aggregation Protocol, FODAP)的详细设计方案。我们首先给出 **FODAP** 的总体架构, 然后介绍 **FODAP** 的设计细节, 包括主要步骤和关键算法, 最后给出 **FODAP** 的实现方式。

### 5.4.1 FODAP 总体框架

图 5.4 给出了 **FODAP** 的总体设计方案。**FODAP** 是周期运行的。即每隔一段时间, 会执行一次 **FODAP** 过程。单次 **FODAP** 执行完毕后, 会进行多轮次的数据聚合操作。当网络状态信息(链路损耗概率、拓扑结构等)显著变化时, 需要重新执行 **FODAP**。**FODAP** 的基本思路是采用链路可靠性增强手段来提高链路传输的成功概率, 从而增强数据聚合的信息质量。我们考虑三种不同的链路可靠性增强策略, 即基于动态睡眠周期调整(DSSA)的策略, 基于机会路由(OR)的策略和混合(HYB)策略。采

用上述三种链路可靠性增强策略的错误容忍数据聚合协议分别表示为 **FODAP-D**, **FODAP-R** 和 **FODAP-H**。在进行链路可靠性增强时，我们需要考虑链路损耗概率的多样性和节点级数据聚合信息质量的多样性。并且，在选择采用何种链路增强技术时，我们还需要考虑该策略带来的额外的控制负载。总的来说，**FODAP** 包括若干主要步骤，即节点级聚合信息质量计算、链路可靠性估计和链路可靠性增强策略选择等。接下来，我们将详细介绍每个步骤的实现方法。

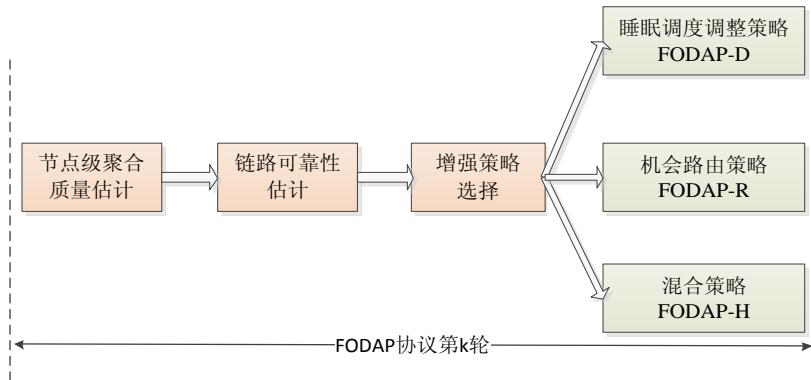


图 5.4 FODAP 总体框架示意图

## 5.4.2 FODAP 设计细节

### 5.4.2.1 节点级聚合质量估计

在 5.3.3 节中，我们已经讨论过，不同链路的损耗对数据聚合 QoI 期望值的影响程度是不一样的。链路传输的数据包信息质量越高，其丢包对最终 sink 节点体验到 (experienced) 的 QoI 的影响越大。为了克服某条链路损耗的不利影响，我们需要知道该链路传输的数据包的信息质量。对于节点  $v$ ，其聚合信息质量  $\text{QoI}(v)$  定义为节点  $v$  产生的数据聚合包的信息质量。具体来说， $\text{QoI}(v)$  与以  $v$  为根的子树  $T(v)$  的大小和子树中每条链路的损耗概率有关。也就是说， $\text{QoI}(v)$  是一个随机变量，其期望值可以采用类似式(5.3.1)的形式表达。注意到，子树  $T(v)$  的大小  $(\text{size})|T(v)|$  越大， $v$  的聚合信息质量  $\text{QoI}(v)$  也就越高。并且，相对来说，树大小  $|T(v)|$  是影响  $\text{QoI}(v)$  的主要因素。因此，我们采用  $|T(v)|$  作为  $\text{QoI}(v)$  的估计值。即，我们有  $\widetilde{\text{QoI}}(v) = |T(v)|$ 。

### 5.4.2.2 链路可靠性估计

考虑实际应用场景，相邻节点之间的链路是不可靠的，这意味着在数据聚合过程中，会发生丢包。为了获得每条链路的质量状况，即丢包的概率，我们需要采用某种手段对链路的可靠性进行估计。我们采用文献<sup>[123]</sup>的 4-比特估计方法对链路进行质量估计。该方法通过使用数据包和周期性的信标对链路质量进行测量和估计，具体的细节请读者参见文献<sup>[123]</sup>。

### 5.4.2.3 链路增强策略选择

当链路不可靠时，为了提升数据聚合的信息质量，我们需要采用一种技术手段来增强链路传输的成功概率。在本节我们将介绍三种用于提升链路传输成功概率的策略，分别为动态睡眠调度调整策略、机会路由策略和混合策略。三种策略具有各自不同的特点。在实际协议设计中，需要根据应用场景的不同和网络性能的需求，自适应地选择其中一种策略作为错误容忍数据聚合协议的基础。

(a) **动态睡眠调度调整策略：**为了克服链路损耗给数据聚合信息质量带来的损失，我们需要增强链路传输的可靠性。在本节，我们介绍一种可用于增强链路可靠性的策略，即动态睡眠调度调整(Dynamic Sleep Scheduling Adjustment, DSSA)策略。DSSA 策略的基本思路是根据数据聚合信息质量的大小和发射链路损耗的严重程度，动态地调整接收方的睡眠调度方案，以利于发送方进行聚合包的重传。图 5.5 给出了睡眠调度调整策略的示意图。在图 5.5(a)中，未进行链路可靠性增强时，节点 $u$ 只在每个工作/睡眠周期的 4 号时隙给 $v$ 发送数据；而在图 5.5(b)中，通过调整接收方 $v$ 的睡眠调度，节点 $u$ 此时能够在每个周期的 4、6 和 8 号时隙给 $v$ 发送数据。在具体实现上，对于链路 $(u, v)$ ，我们根据发送方 $u$ 的聚合信息质量 $QoI(u)$ ，链路 $(u, v)$ 的损耗概率 $\rho(u, v)$ 来决定接收方 $v$ 的睡眠调度 $\Phi(v)$ 。定义在单个周期内，接收节点 $v$ 能用于从链路 $(u, v)$ 接收数据的活跃时隙的数量为 $\mathcal{N}_v(u)$ 。 $\mathcal{N}_v(u)$ 也表示发送方 $u$ 允许的重传次数。对于损耗概率为 $\rho(u, v)$ 的链路，在最大重传次数 $\mathcal{N}_v(u)$ 下，其链路传输成功率为：

$$\eta(u, v) = 1 - [\rho(u, v)]^{\mathcal{N}_v(u)} \quad (5.4.1)$$

因此，被允许的重传次数越大，链路的传输成功概率也就越高。如何为每条链路 $(u, v)$ 选取合适的 $\mathcal{N}_v(u)$ ，是 DSSA 策略的关键。一般来说，对于 $\rho(u, v)$ 较小的链路，由于传输成功率已经很高，从节省能量的角度出发，我们希望 $\mathcal{N}_v(u)$ 的值比较小。对于 $\rho(u, v)$ 较大的链路， $\mathcal{N}_v(u)$ 的取值取决于 $QoI(u)$ 的大小。此时，如果 $QoI(u)$ 的值较大，为了提升数据聚合信息质量， $\mathcal{N}_v(u)$ 的值应该较大；如果 $QoI(u)$ 的值较小，从节省能量的角度， $\mathcal{N}_v(u)$ 的取值应比较小。总之，我们希望在增强数据聚合信息质量的同时，尽可能地节省网络的能耗。在 DSSA 策略中，给定 $\rho(u, v)$ 和 $QoI(u)$ ，我们采用下面的公式计算最大重传次数 $\mathcal{N}_v(u)$ ：

$$\mathcal{N}_v(u) = w \cdot \frac{QoI(u)}{1 - \rho(u, v)} \quad (5.4.2)$$

这里 $w$ 是一个可配置的参数。在确定了链路 $(u, v)$ 的最大重传次数 $\mathcal{N}_v(u)$ 后，我们需要调整节点 $v$ 的睡眠调度。对于接收方 $v$ 来说，需要增加其在每个周期内处于接收状态的时隙数量。睡眠调度调整策略如下：我们从 $\{0, 1, \dots, W - 1\}$ 中随机选择 $\mathcal{N}_v(u)$ 个时隙，作为 $v$ 的接收时隙，并将其反馈给发送方 $u$ 。这样， $u$ 便将这 $\mathcal{N}_v(u)$ 个时隙作为自己潜在的发送时隙。在数据聚合过程中，当链路 $(u, v)$ 发生包丢失时，如果还未达到

最大重传次数 $\mathcal{N}_v(u)$ , 则 $u$ 在下一个发送时隙重传(retransmit)聚合数据包, 否则,  $u$ 放弃传输自己产生的数据(聚合)包。

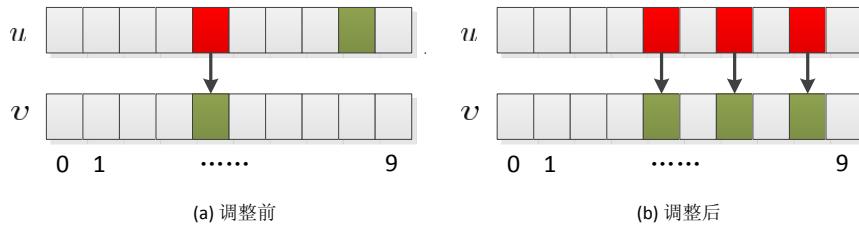


图 5.5 一种睡眠调度调整策略示意图

(b) 机会路由策略: 无线网络的一个基本特点是多径(multi-path)。注意到传输路径具有冗余性(redundancy), 也就是说, 每个节点通常存在多个候选的下一跳转发节点。这意味着, 节点 $v$ 并不是一定要将聚合数据包发送给其父节点 $P_v$ , 还存在其他的邻居节点也能充当 $v$ 的数据聚合包的转发节点。机会路由(Opportunistic Routing, OR)策略正是基于这一观察提出的。具体来说, 机会路由策略为每个节点 $v$ 构造一个候选的转发节点集合 $\mathcal{F}_v$ , 并在数据聚合过程中, 根据 $v$ 到 $\mathcal{F}_v$ 中的每个节点的链路损耗状况, 动态地决定选择哪一条链路作为节点 $v$ 的传输链路。图 5.6 给出了机会路由策略的示意图。在该图中, 节点 $b$ 需要向节点 $a$ 发送数据包。链路 $(b, a)$ 是有损链路, 其丢包概率为 0.3。此时, 由于节点 $b$ 到候选转发节点 $c$ 之间的链路损耗概率更小, 为 0.1。因此, 节点 $b$ 可以基于机会路由策略, 用节点 $c$ 替代节点 $a$ 作为自己的目的节点, 以提高数据包的传输成功概率, 如图 5.5(b)所示。

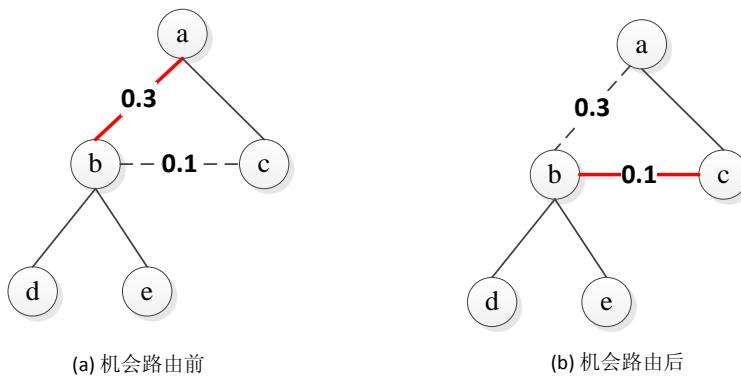


图 5.6 机会路由策略示意图

为了避免网络中出现循环路径(cyclic path), 我们规定转发节点集合 $\mathcal{F}_v$ 选取为 $v$ 的邻居节点集合 $\mathcal{B}_v$ 中除其孩子节点集合 $\mathcal{C}_v$ 之外的节点, 即:  $\mathcal{F}_v = \mathcal{B}_v \setminus \mathcal{C}_v$ 。以何种标准从集合 $\mathcal{F}_v$ 中选择节点 $v$ 的下一跳转发节点, 是机会路由策略的关键。为了提高传输的成功概率, 我们总是从集合 $\mathcal{F}_v$ 中选取到节点 $v$ 的链路质量最好(best link)的节点作为 $v$ 的下一跳转发节点 $P_v$ 。具体来说, 我们有:

$$P_v = \arg \max_{u \in \mathcal{F}_v} \rho(v, u) \quad (5.4.3)$$

由于重新选择候选集合会带来一定的开销，因此何时运行机会路由策略很重要。一般来说，当链路( $v, P_v$ )的损耗概率较高时，我们通过 OR 策略选取一个替代父节点  $P'_v$ ，使得替代后的链路损耗概率  $\rho(v, P'_v)$  较低。另外，当原来的父节点  $P_v$  的睡眠调度已经接近饱和，即此时无法通过简单的重传方法增强链路( $v, P_v$ )的可靠性时，需要进行机会路由。此时，进行机会路由不仅能够增强数据聚合的信息质量，而且有望使得节点之间的负载更加均衡，从而进一步改善网络生命周期。

(c) **混合策略：**在前面的章节，我们介绍了两种用于增强聚合包传输成功概率的策略，即动态睡眠调度调整(DSSA)策略和机会路由(OR)策略。这两种策略有各自的特点。DSSA 策略是以接收方的能耗增加为代价的。为了增强传输可靠性，接收方需要调整自己的睡眠/工作周期，从而在每个周期里休眠的时隙数量减少，而活跃的时隙数量增加。特别的，对于路由树中拥有多个子节点的中间节点，如果其需要为每个子节点都扩展相应的接收时隙范围，则其总的占空比(duty-cycle ratio)将急剧变大，从而缩短自身的生命周期。对于 OR 策略来说，由于发送节点需要知道到候选转发节点集合  $\mathcal{F}_v$  中每个节点的链路传输成功概率，因此将加重链路可靠性估计算法的负载。当链路损耗概率频繁波动时，采用 OR 策略将导致严重的控制包能量消耗。因此，一般来说 OR 策略适用于链路损耗概率变化比较平稳的网络场景。由于 DSSA 策略和 OR 策略是相互独立的，因此我们结合这两种策略的优势，提出一种混合(hybrid, HYB)策略。该策略同时进行动态睡眠调度调整和机会路由，以联合地增强链路传输可靠性，从而提升数据聚合信息质量。

#### 5.4.3 FODAP 实现方式

我们将采用了动态睡眠调度调整策略、机会路由策略和混合策略的 **FODAP** 分别表示为 **FODAP-D**, **FODAP-R** 和 **FODAP-H**。这三个协议变种的实现方式是类似的。在实际系统中，链路质量变化的频率通常比数据聚合操作变化的频率要低。每经过一段时间的数据聚合，我们执行一次 **FODAP** 过程。在每次执行 **FODAP** 时，我们首先计算每个节点的聚合信息质量，然后让每个节点执行链路估计算法，测量其与各个邻居节点之间的链路的损耗概率，然后根据系统预设或应用场景需求，从 DSSA、OR 和 HYB 三种策略中选取一种，作为链路可靠性增强手段，以提升数据聚合信息质量 QoI。一般来说，为了减轻协议运行造成的能力消耗，只有当链路质量变化幅度较大时，我们需要重新运行 **FODAP**。因此，为了降低 **FODAP** 带来的额外控制开销，我们可以自适应地根据链路质量变化的幅度和频率，来动态调整 **FODAP** 运行的周期间隔。

## 5.5 方案性能评价

### 5.5.1 实验设置与评价标准

在本节，我们通过仿真实验对所提出来的三种错误容忍数据聚合协议 **FODAP-D**、**FODAP-R** 和 **FODAP-H** 的性能进行评价。我们的仿真实验建立在 Intel Lab Data<sup>[124]</sup> 的网络场景和试验数据集之上。如图 5.7 所示，在该实验室布设的无线传感器网络中，一共有 54 个传感器节点。每个传感器节点获取环境中的温度、湿度等信息，并将其发送给汇聚节点(1 号节点)。网络中的每个节点的通信范围设置为 10m。以 sink 节点为根节点，构造一颗最短路径树(shorest path tree, SPT)作为数据聚合的路由。我们通过统计成功参与数据聚合过程的节点数量作为数据聚合的信息质量 QoI。如果不存在链路损耗，则数据聚合信息质量 QoI 达到最大值 54。为了简单起见，我们不考虑协议运行本身所造成的开销，只将关注重点放在数据聚合信息质量 QoI 这一性能指标上。

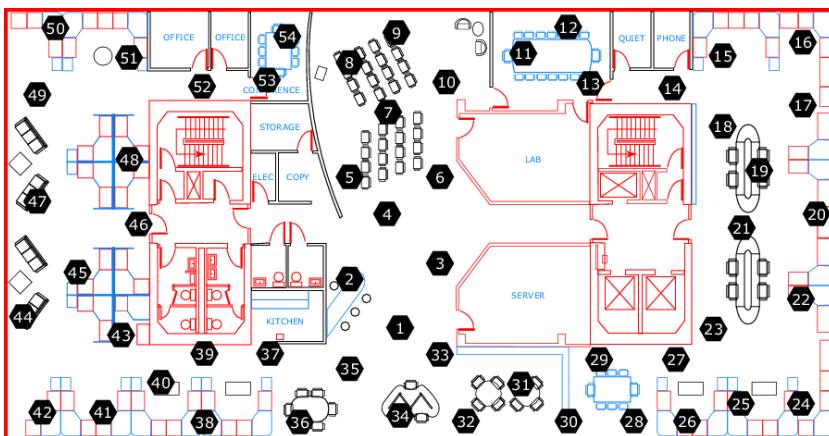


图 5.7 拥有 54 个传感器节点的 Intel Berkeley Research lab 传感器网络平面图

### 5.5.2 实验结果与分析

我们进行连续 3000 次的数据聚合，在第 1~1000 轮数据聚合时，网络的损耗程度较低，具体来说，链路的丢包概率在(0, 0.1)的范围内取值；在第 1001~2000 轮数据聚合时，网络的损耗程度中等，链路的丢包概率在(0.1, 0.4)的范围内取值；在第 2001~3000 轮数据聚合时，网络的损耗程度较高，链路的丢包概率在(0.4, 0.8)的范围内取值。我们统计在每一轮数据聚合时，三种错误容忍数据聚合协议 **FODAP-D**、**FODAP-R** 和 **FODAP-H** 的数据聚合信息质量。同时，我们与文献<sup>[125]</sup>中的 **OPD** 协议进行性能对比。**OPD** 协议没有考虑链路损耗，也就是说，无论传输成功与否，每个数据包均传输且只传输一次。图 5.8 给出了在上述 3000 轮的数据聚合过程中，四种协议各自的数据聚合信息质量 QoI 的分布图。可以直观地发现，当链路损耗程度加剧

时，四种协议的数据聚合质量均显著降低。并且，相比 QoI 性能最差的 **OPD** 协议，本文提出的三种错误容忍数据聚合协议 **FODAP-D**, **FODAP-R** 和 **FODAP-H** 均能一定程度地提升网络的数据聚合 QoI。具体来说，**FODAP-R** 对 QoI 的提升程度较低，而 **FODAP-D** 和 **FODAP-H** 均能显著提升数据聚合 QoI。注意到，对于采用了动态睡眠调度调整策略的错误容忍协议 **FODAP-D** 和 **FODAP-H**，为了实验方便，我们此时将每个节点的最大传输次数均固定为 3 次。

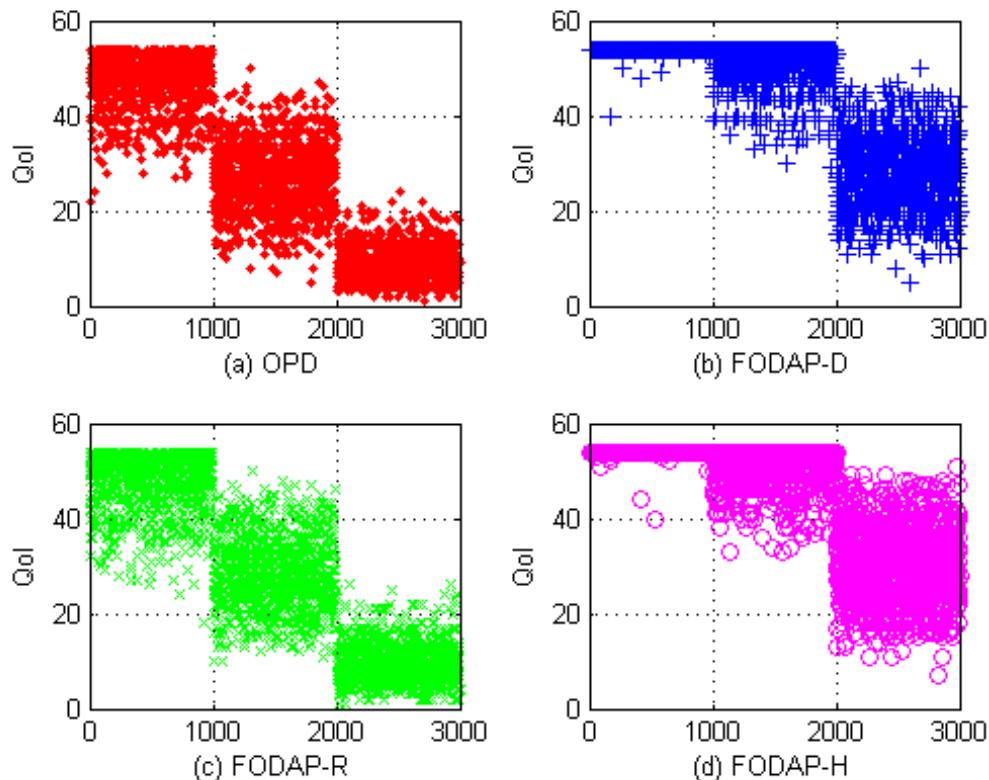


图 5.8 3000 轮数据聚合过程中各种协议数据聚合信息质量 QoI 的分布图

为了进一步量化地评价三种错误容忍数据聚合协议对数据聚合 QoI 的提升程度，我们统计出各种网络损耗程度下，1000 轮数据聚合中四种协议的 QoI 平均值，所获得的结果如图 5.9 所示。由图 5.9 可知，在低损耗网络场景下，没有考虑链路损耗的 **OPD** 协议的平均信息质量 QoI 为 47.16，而三种错误容忍数据聚合协议 **FODAP-D**, **FODAP-R** 和 **FODAP-H** 的平均 QoI 分别为 53.96, 47.94 和 53.97，分别较 **OPD** 协议提高 14.4%，0.02% 和 14.5%；在中等损耗网络场景下，**OPD** 的平均信息质量 QoI 显著下降为 26.78，而 **FODAP-D**, **FODAP-R** 和 **FODAP-H** 的平均 QoI 分别为 51.12, 28.31 和 51.78，分别较 **OPD** 提高 90.9%，5.7% 和 93.4%；在高损耗网络场景下，**OPD** 的平均信息质量 QoI 下降到仅为 8.91，而 **FODAP-D**, **FODAP-R** 和 **FODAP-H** 的平均 QoI 分别为 28.07, 9.75 和 30.2，分别较 **OPD** 提高 215%，9.4% 和 239%。根据这组数据，我们可知，本章提出的三种错误容忍协议均能一定程度上克服网络丢包的不

利影响，从而提升数据聚合的信息质量。并且，当网络链路的损耗加剧时，协议对数据聚合信息质量的提升程度将更为显著。另外，在各种损耗场景下，采用混合策略的协议 **FODAP-H** 的性能是最好的，而仅采用机会路由策略的协议 **FODAP-R** 相对性能改善程度较低。

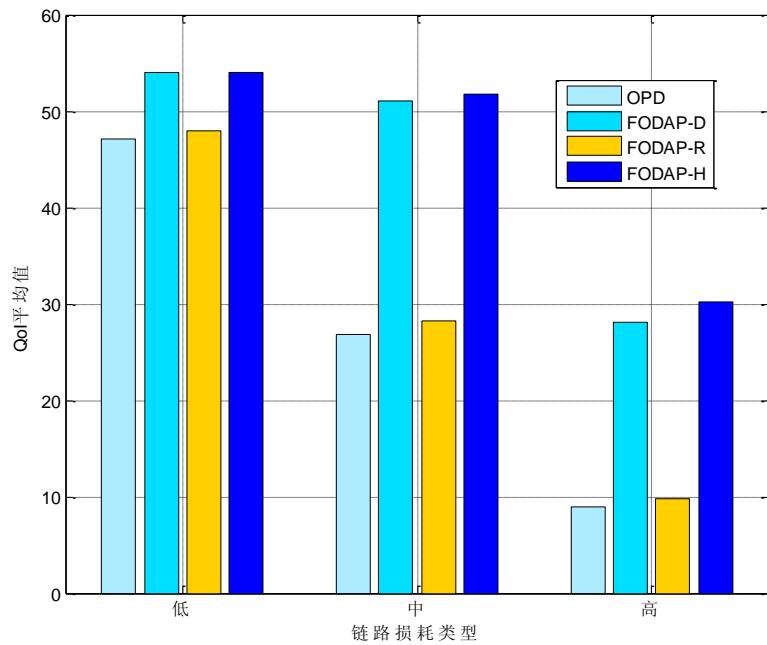


图 5.9 三种不同损耗场景下数据聚合协议的平均信息质量 QoI 对比

为了进一步的比较三种错误容忍协议的性能，我们统计了中等网络损耗程度下，1000 轮数据聚合中，**FODAP-D**，**FODAP-R** 和 **FODAP-H** 的信息质量 QoI 的累积分布函数(Cumulative Distribution Function, CDF)，如图 5.10 所示。由图 5.10 可知，三种协议的数据聚合 QoI 性能从低到高依次为 **FODAP-R**，**FODAP-D** 和 **FODAP-H**。举例来说，在 1000 轮数据聚合过程中，sink 节点获得的信息质量小于或等于 40 的概率分别为 93%(**FODAP-R**)，29%(**FODAP-D**)和 22%(**FODAP-H**)。

在上述实验中，当采用动态睡眠调度调整策略进行链路可靠性增强时，我们将节点最大传输次数设置为 3 次。接下来，我们研究节点最大传输次数  $k$  对数据聚合信息质量 QoI 的影响。我们考虑中等损耗程度的网络模型，即链路的丢包概率在(0.1, 0.4)的范围内取值。数据聚合的轮次为 1000 次，图 5.11 给出了当采用不同重传次数时，**FODAP-D** 所获得的数据聚合信息质量 QoI 的累积分布函数 CDF。由图可知，当最大传输次数增加时，网络的数据聚合质量 QoI 性能显著提升。具体来说，当最大传输次数为 2 次时，QoI 的值不大于 50 的概率为 24.5%，不大于 52 的概率为 40%；而当最大传输次数为 6 次时，QoI 的值不大于 50 的概率下降为 0.5%，不大于 52 的概率下降为 1%左右。可见，通过增大发送节点最大重传次数，数据聚合的信息质量能够进

一步得到提升。当然，这是以网络能量消耗的增加为代价的。在实际应用中，不能仅仅为了提升数据聚合信息质量而大幅度提高节点的最大重传次数，而应在网络能耗与聚合信息质量之间取得平衡。

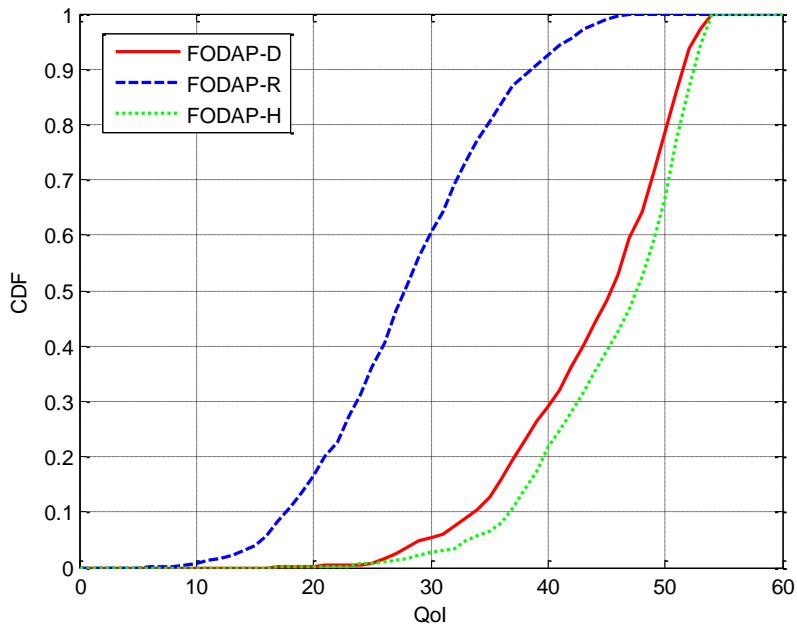


图 5.10 三种错误容忍数据聚合协议 QoI 的累积分布函数 CDF

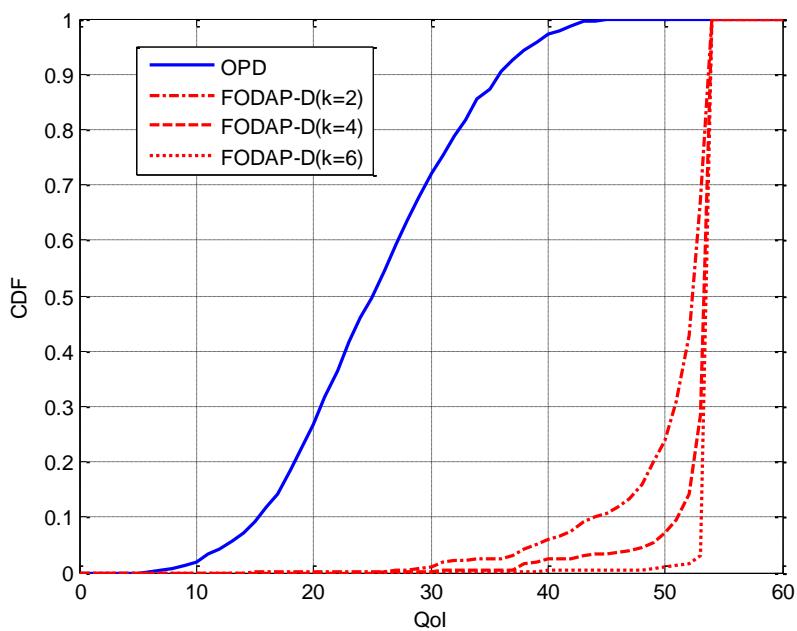


图 5.11 采用不同最大传输次数时，FODAP-D 协议 QoI 的累积分布

在之前的实验中，我们发现采用机会路由策略的协议 **FODAP-R** 对网络信息质量有一定的提升，但是提升幅度不大。在 **FODAP-R** 协议中，我们采用的机会策略是从候选链路集合中选择链路质量最好的链路进行传输。接下来，我们研究当采用不同的

链路选择策略时，**FODAP-R** 协议的性能变化情况。我们考虑三种策略，即 worst 策略、random 策略和 best 策略，分别表示从候选链路集合中选择链路质量最差的链路、随机选择链路，和从候选链路集合中选择链路质量最好的链路，作为传输路径。图 5.12 给出了中等损耗网络中，三种策略的 QoI 累积分布函数示意图。由图 5.12 可知，相比 worst 策略和 random 策略，采用质量最佳链路优先的机会路由策略所获得的 QoI 性能最好。

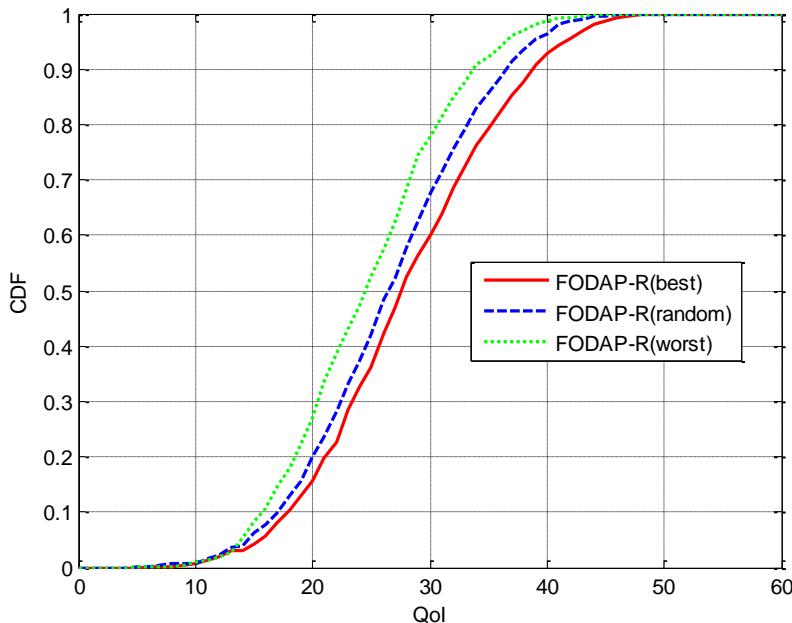


图 5.12 不同链路选择策略下 FODAP-R 协议 QoI 的累积分布

## 5.6 本章小结

本文针对低占空比且链路不可靠的无线传感器网络，设计了一种错误容忍数据聚合协议 **FODAP**。**FODAP** 的设计目的是改善链路传输错误情况下的数据聚合信息质量 QoI。为此，**FODAP** 充分利用了链路可靠性增强技术。根据采用的链路可靠性增强技术的不同，我们为 **FODAP** 设计了三个变种，分别为基于动态睡眠调度调整策略的错误容忍数据聚合协议 **FODAP-D**，基于机会路由策略的错误容忍数据聚合协议 **FODAP-R** 和基于混合策略的错误容忍数据聚合协议 **FODAP-H**。我们给出了协议的总体框架、设计细节和实现方式。仿真实验表明，在链路不可靠的网络场景下，相比文献中传统的未采用链路可靠性增强技术的数据聚合协议，本章提出的上述三种协议均能够提升数据聚合中 sink 节点获取的信息质量 QoI，其中基于混合策略的 **FODAP-H** 较另外两种协议对 QoI 性能的提升更为显著。

## 第6章 总结与展望

### 6.1 本文总结

数据聚合是无线传感器网络(WSN)中一种重要的业务模式(traffic pattern)。数据聚合为在资源严重受限的 WSN 中进行高效数据收集开辟了一个新的途径。然而，为了等待并进行数据融合操作，数据聚合将为网络带来额外的包延时。为了更好地满足时延敏感应用的需求，迫切需要设计高效的数据聚合调度算法，以改善网络时延性能。事实上，数据聚合调度不仅关系到网络的时延性能，而且也与生命周期、能量、数据精度等性能因子密切相关。本博士论文针对各种复杂网络场景，如多用户干扰、链路丢包等情形下的无线传感器网络数据聚合调度问题展开了深入研究。本文所取得的创新点如下：

- 1) 针对协议干扰模型下的 WSN 数据聚合调度问题，本文提出了一种改进的两阶段(two-phase)数据聚合方案。在第一阶段，提出了基于连通支撑集的改进型数据聚合树构造算法 **IDATC**；在第二阶段，分别基于分层贪心策略和网络分区与着色策略提出了集中式数据聚合调度算法 **LGAS** 和分布式数据聚合调度算法 **NPCS**。通过分析，论证了方案的可行性，推导了 **LGAS** 算法和 **NPCS** 算法的时延上界。理论推导结果表明所提出的两种调度算法均是具有常量近似比的次优算法(costant approximation algorithms)。仿真实验表明，本文调度方案的平均时延性能远好于理论上界。并且与文献中的代表性算法相比，本文算法在多种网络场景下均能一定程度上降低数据聚合的总时延。
- 2) 针对物理干扰模型下的 WSN 数据聚合调度问题，本文提出了一种完全分布式的(fully distributed)数据聚合解决方案 **DDA**。该方案以迭代网络分区技术为基础，由分布式数据聚合树构造子算法 **DDATC**、分布式链路调度子算法 **DLS** 和两种功率控制子算法 **PA-U** 与 **PA-L** 联合而成。通过合理配置调度和功率控制模块的参数，我们证明了该方案在物理干扰模型下的可行性。同时，我们推导了该方案的理论聚合时延上界。理论结果表明，**DDA** 的时延上界与网络规模无关，从而显示 **DDA** 具有良好的扩展性(scability)。我们还通过大量的仿真实验评价了该方案在不同场景下的平均时延、能耗等性能。
- 3) 基于主动的干扰消除理念，我们将物理层多包接收模型(Physical-layer Multi-packet Reception)引入了 WSN 数据聚合调度中。我们分别基于随机链路优先策略和最短链路优先策略，提出了两种时延高效的启发式聚合调度算法 **RFGS** 和 **SFGS**，以充分利用干扰消除和多包接收能力所带来的链路并发传输机会。证

明了两种调度算法的可行性，并且分析了两种调度算法的计算时间复杂度。仿真实验结果表明，相比传统的基于单包接收模型的调度方法，本文算法能够显著地降低数据聚合平均时延，提高网络吞吐量。本章研究工作揭示了干扰消除技术在调度算法设计时的潜力。也就是说，当物理层具有干扰消除能力时，通过精心设计相应的MAC层与网络层算法/协议，无线网络的性能有望得到显著提升。

- 4) 针对低占空比(low-duty-cycle)且链路不可靠(unreliable)的无线传感器网络，提出了一种错误容忍的数据聚合协议**FODAP**。该协议通过采用链路可靠性增强手段，来提高点到点的传输成功率，最终改善网络链路损耗状况下的数据聚合信息质量QoI。依据采用的链路可靠性增强技术的不同，我们基于动态睡眠调度调整策略，机会路由策略和混合策略，为该协议设计了三个变种，分别为**FODAP-D**，**FODAP-R** 和 **FODAP-H**。仿真实验表明，在链路不可靠的网络场景下，相比文献中传统的未采用链路可靠性增强技术的数据聚合协议，上述三种协议均能够不同程度地提升数据聚合过程中汇聚节点所接收聚合包的信息质量。其中基于混合策略的**FODAP-H**较另外两种协议对QoI性能的提升更为显著。

## 6.2 下一步研究计划

尽管无线传感器网络中的数据聚合调度得到了深入研究，但是依然存在很多值得改进和进一步探索的地方，主要体现在：

- 1) **改善已有数据聚合调度方法的理论上界**：相比启发式算法(heuristic algorithms)，具有理论性能保证的近似算法(approximation algorithms)更具有吸引力。这是因为，实际应用场景变化多端，启发式算法可能在某种场景下性能较好，但是当应用场景变化时，其性能可能变得很差。而对于近似算法来说，无论应用场景如何变化，其性能均是有保证的。近似算法的最坏性能不会超过其理论上界。近似算法的理论上界(upper bound)的大小是衡量近似算法性能的重要指标。如何进一步降低数据聚合调度算法时延的理论上界，是一个重要的、理论意义与实际意义兼具的研究方向。并且，这方面的探索是难以穷尽的。因为一般来说，对于这些通常都是**NP**-难的数据聚合调度问题，我们只能接近而无法达到最优的理论时延<sup>[126]</sup>。
- 2) **高级干扰模型下的数据聚合调度**：本文考虑了若干常见的干扰模型，如协议干扰模型、物理干扰模型等。然而，无线干扰的实际情况是极为复杂的。上述两种模型本质上都是抽象模型(abstract models)，尽管有利于进行建模和分析，却无法真实地刻画无线干扰的实际面貌。在文献中，研究人员陆陆续续提出了更为准确、更为实际的干扰模型，如高斯信道模型(Gaussian Channel Model)<sup>[78]</sup>、基于测量的干扰模型(Measurement-based Interference Model)<sup>[79]</sup>等。目前，基于这些高级干扰模型的WSN数据聚合调度研究基本上还处于空白状态。当采用这些高级干扰模

型时，数据聚合调度问题将变得更为复杂。这种复杂体现在多方面，如问题建模的复杂性(用标准的数学建模方法对调度问题进行建模)，求解方法的复杂性(提出可行的、并且有理论性能保证的新型聚合调度算法)等。

- 3) **易于工程化的分布式数据聚合调度：**对于算法设计来说，算法的实现方式是必须考虑的重要因素。从实现方式来讲，数据聚合调度分为集中式调度和分布式调度。在集中式调度中，由基站收集全网的拓扑信息、干扰信息等，然后计算调度方案，再将调度方案以多跳广播的方式发送给全网节点。在分布式调度中，通过每个节点与邻居节点之间以消息传递(message passing)的方式进行协商，来产生每个节点自身的调度方案。相比集中式调度，分布式调度的优点是能适应网络动态性，并且易于在自组织网络中实现。但是由于信息量不充分，分布式调度很难得到性能较好的调度方案。当前文献中的调度方案大多是集中式的。本文虽然对分布式数据聚合调度进行了初步的研究，但是总体来说仍不充分。针对网络动态性，如时变信道条件、不可靠的传输链路、节点频繁加入/失效/移动等情形，设计简单、快速、有较好性能，并且易于实现的分布式数据聚合调度方法仍然是一个重要的研究方向。
- 4) **满足多种性能指标的数据聚合调度：**除了时间效率，数据聚集调度也直接影响着其他的一些重要网络性能指标，包括能量有效性、数据精确性、聚合新鲜度等。在设计数据聚合调度方案时，需要综合考虑这些性能指标。理想的状况是所有这些指标均同时达到最优(optimal)。然后，通常这些性能指标相互之间往往是互相冲突的。例如数据聚合时延与数据精确性之间、能量有效性与数据精确性之间、数据聚合时延与能量有效性之间均是互相影响和冲突的。如何在多个设计目标之间达到平衡，是数据聚合调度研究面临的重要课题。当然，在处理这种多目标优化问题时，可以寻求多个设计目标的平衡(Pareto optimal)，也可以寻求其他变量约束下的单一变量最优化。当前的无线传感器网络数据聚合调度研究基本还是以单一设计目标为主，考虑多个设计目标的聚合调度是一个极具挑战性的研究方向。



## 参考文献

- [1] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [2] M. P. Durisic, Z. Tafa, G. Dimic, and V. Milutinovic, “A survey of military applications of wireless sensor networks,” in *Embedded Computing (MECO), 2012 Mediterranean Conference on*. IEEE, 2012, pp. 196–199.
- [3] L. M. Oliveira and J. J. Rodrigues, “Wireless sensor networks: a survey on environmental monitoring,” *Journal of communications*, vol. 6, no. 2, pp. 143–151, 2011.
- [4] V. C. Gungor and G. P. Hancke, “Industrial wireless sensor networks: Challenges, design principles, and technical approaches,” *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 10, pp. 4258–4265, 2009.
- [5] Y. Liu, G. Zhou, J. Zhao, G. Dai, X.-Y. Li, M. Gu, H. Ma, L. Mo, Y. He, J. Wang *et al.*, “Long-term large-scale sensing in the forest: recent advances and future directions of greenorbs,” *Frontiers of Computer Science in China*, vol. 4, no. 3, pp. 334–338, 2010.
- [6] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 88–97.
- [7] M. Di Francesco, S. K. Das, and G. Anastasi, “Data collection in wireless sensor networks with mobile elements: A survey,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 1, p. 7, 2011.
- [8] S. Pattem, B. Krishnamachari, and R. Govindan, “The impact of spatial correlation on routing with compression in wireless sensor networks,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 4, p. 24, 2008.
- [9] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “In-network aggregation techniques for wireless sensor networks: a survey,” *Wireless Communications, IEEE*, vol. 14, no. 2, pp. 70–87, 2007.
- [10] R. Rajagopalan and P. Varshney, “Data-aggregation techniques in sensor networks: A survey,” *Communications Surveys & Tutorials, IEEE*, vol. 8, no. 4, pp. 48–63.
- [11] M. Bagaa, Y. Challal, A. Ksentini, A. Derhab, and N. Badache, “Data aggregation scheduling algorithms in wireless sensor networks: Solutions and challenges,” 2014.

- [12] X. Chen, X. Hu, and J. Zhu, “Minimum data aggregation time problem in wireless sensor networks,” in *Mobile Ad-hoc and Sensor Networks*. Springer, 2005, pp. 133–142.
- [13] X. Fafoutis, D. Vuckovic, A. Di Mauro, N. Dragoni, and J. Madsen, “Energy-harvesting wireless sensor networks,” in *9th European Conference on Wireless Sensor Networks (EWSN 2012)*, 2012, pp. 84–85.
- [14] W. K. Seah, Z. A. Eu, and H.-P. Tan, “Wireless sensor networks powered by ambient energy harvesting (wsn-heap)-survey and challenges,” in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*. Ieee, 2009, pp. 1–5.
- [15] J. Akerberg, M. Gidlund, and M. Bjorkman, “Future research challenges in wireless sensor and actuator networks targeting industrial automation,” in *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*. IEEE, 2011, pp. 410–415.
- [16] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, *Wireless sensor and actuator networks: technologies, analysis and design*. Academic Press, 2010.
- [17] Homepage, <http://www.greenorbs.com/index-1.html>.
- [18] D. E. Comer, *Computer networks and internets*. Prentice Hall Press, 2008.
- [19] L. Van Hoesel, T. Nieberg, J. Wu, and P. J. Havinga, “Prolonging the lifetime of wireless sensor networks by cross-layer interaction,” *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 78–86, 2004.
- [20] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, “The evolution of mac protocols in wireless sensor networks: A survey,” *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 1, pp. 101–120, 2013.
- [21] S. K. Singh, M. Singh, D. Singh *et al.*, “Routing protocols in wireless sensor networks—a survey,” *International Journal of Computer Science & Engineering Survey (IJCSES) Vol*, vol. 1, pp. 63–83, 2010.
- [22] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, “Clock synchronization for wireless sensor networks: a survey,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.
- [23] Y. Liu, Z. Yang, X. Wang, and L. Jian, “Location, localization, and localizability,” *Journal of Computer Science and Technology*, vol. 25, no. 2, pp. 274–297, 2010.
- [24] M. Rudafshani and S. Datta, “Localization in wireless sensor networks,” in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International*

- Symposium on. IEEE, 2007, pp. 51–60.
- [25] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, “Topology control for wireless sensor networks,” in *Proceedings of the 9th annual international conference on Mobile computing and networking*. ACM, 2003, pp. 286–299.
- [26] A. Perrig, J. Stankovic, and D. Wagner, “Security in wireless sensor networks,” *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, 2004.
- [27] D. Chen and P. K. Varshney, “Qos support in wireless sensor networks: A survey.” in *International Conference on Wireless Networks*, vol. 233, 2004, pp. 1–7.
- [28] B. Latre, B. Braem, I. Moerman, C. Blondia, and P. Demeester, “A survey on wireless body area networks,” *Wireless Networks*, vol. 17, no. 1, pp. 1–18, 2011.
- [29] K. Bing, L. Fu, Y. Zhuo, and L. Yanlei, “Design of an internet of things-based smart home system,” in *Intelligent Control and Information Processing (ICICIP), 2011 2nd International Conference on*, vol. 2. IEEE, 2011, pp. 921–924.
- [30] X. Fang, S. Misra, G. Xue, and D. Yang, “Smart grid—the new and improved power grid: A survey,” *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 4, pp. 944–980, 2012.
- [31] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, “A survey of mobile phone sensing,” *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140–150, 2010.
- [32] X. Wu, G. Chen, and S. K. Das, “Avoiding energy holes in wireless sensor networks with nonuniform node distribution,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 5, pp. 710–720, 2008.
- [33] B. Krishnamachari, D. Estrin, and S. Wicker, “The impact of data aggregation in wireless sensor networks,” in *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*. IEEE, 2002, pp. 575–578.
- [34] H. Tan and I. Korpeoglu, “Power efficient data gathering and aggregation in wireless sensor networks,” *ACM Sigmod Record*, vol. 32, no. 4, pp. 66–71, 2003.
- [35] S. Xiao, B. Li, and X. Yuan, “Maximizing precision for energy-efficient data aggregation in wireless sensor networks with lossy links,” *Ad Hoc Networks*, vol. 26, pp. 103–113, 2015.
- [36] A. A. Abbasi and M. Younis, “A survey on clustering algorithms for wireless sensor networks,” *Computer communications*, vol. 30, no. 14, pp. 2826–2841, 2007.
- [37] D. C. Hoang, R. Kumar, and S. K. Panda, “Optimal data aggregation tree in wireless sensor networks based on intelligent water drops algorithm,” *IET wireless sensor*

- systems*, vol. 2, no. 3, pp. 282–292, 2012.
- [38] C. Joo and N. B. Shroff, “On the delay performance of in-network aggregation in lossy wireless sensor networks,” *Networking, IEEE/ACM Transactions on*, vol. 22, no. 2, pp. 662–673, 2014.
- [39] A. Sgora, D. J. Vergados, and D. D. Vergados, “A survey of tdma scheduling schemes in wireless multihop networks,” *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, p. 53, 2015.
- [40] C. Joo, J.-G. Choi, and N. B. Shroff, “Delay performance of scheduling with data aggregation in wireless sensor networks,” in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.
- [41] A. Ghosh, O. D. Incel, V. A. Kumar, and B. Krishnamachari, “Multi-channel scheduling algorithms for fast aggregated convergecast in sensor networks,” in *Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on*. IEEE, 2009, pp. 363–372.
- [42] D. Chafekar, V. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, “Cross-layer latency minimization in wireless networks with sinr constraints,” in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2007, pp. 110–119.
- [43] M. Bagaa, A. Derhab, N. Lasla, A. Ouadjaout, and N. Badache, “Semi-structured and unstructured data aggregation scheduling in wireless sensor networks,” in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2671–2675.
- [44] H. Du, Z. Zhang, W. Wu, L. Wu, and K. Xing, “Constant-approximation for optimal data aggregation with physical interference,” *Journal of Global Optimization*, pp. 1–14, 2012.
- [45] N. Hobbs, Y. Wang, Q.-S. Hua, D. Yu, and F. C. Lau, “Deterministic distributed data aggregation under the sinr model,” in *Theory and Applications of Models of Computation*. Springer, 2012, pp. 385–399.
- [46] S.-H. Huang, P.-J. Wan, C. T. Vu, Y. Li, and F. Yao, “Nearly constant approximation for data aggregation scheduling in wireless sensor networks,” in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE. IEEE, 2007, pp. 366–372.
- [47] X. Jiao, W. Lou, X. Wang, J. Cao, M. Xu, and X. Zhou, “Data aggregation scheduling in uncoordinated duty-cycled wireless sensor networks under protocol interference model,” *Ad Hoc and Sensor Wireless Networks*.

- [48] D. Li, Q. Zhu, H. Du, W. Wu, H. Chen, and W. Chen, “Conflict-free many-to-one data aggregation scheduling in multi-channel multi-hop wireless sensor networks,” in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.
- [49] H. Li, Q. S. Hua, C. Wu, and F. C. M. Lau, “Minimum-latency aggregation scheduling in wireless sensor networks under physical interference model,” in *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems*. ACM, 2010, pp. 360–367.
- [50] Y. Li, L. Guo, and S. K. Prasad, “An energy-efficient distributed algorithm for minimum-latency aggregation scheduling in wireless sensor networks,” in *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*. IEEE, 2010, pp. 827–836.
- [51] B. Malhotra, I. Nikolaidis, and M. A. Nascimento, “Aggregation convergecast scheduling in wireless sensor networks,” *Wireless Networks*, vol. 17, no. 2, pp. 319–335, 2011.
- [52] T. D. Nguyen, V. Zalyubovskiy, and H. Choo, “Efficient time latency of data aggregation based on neighboring dominators in wsns,” in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, 2011, pp. 1–6.
- [53] P.-J. Wan, S. C.-H. Huang, L. Wang, Z. Wan, and X. Jia, “Minimum-latency aggregation scheduling in multihop wireless networks,” in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2009, pp. 185–194.
- [54] P.-J. Wan, L. Wang, and O. Frieder, “Fast group communications in multihop wireless networks subject to physical interference,” in *Mobile Adhoc and Sensor Systems, 2009. MASS’09. IEEE 6th International Conference on*. IEEE, 2009, pp. 526–533.
- [55] G. Wang, Q.-S. Hua, and Y. Wang, “Minimum latency aggregation scheduling for arbitrary tree topologies under the sinr model,” in *Ad-hoc, Mobile, and Wireless Networks*. Springer, 2012, pp. 139–152.
- [56] X. Xu, X.-Y. Li, X. Mao, S. Tang, and S. Wang, “A delay-efficient algorithm for data aggregation in multihop wireless sensor networks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 1, pp. 163–175, 2011.
- [57] X. Xu, X.-Y. Li, and M. Song, “Efficient aggregation scheduling in multihop wireless sensor networks with sinr constraints,” *Mobile Computing, IEEE Transactions on*, vol. 12, no. 12, pp. 2518–2528, 2013.
- [58] S. Upadhyayula and S. K. Gupta, “Spanning tree based algorithms for low latency and

- energy efficient data aggregation enhanced convergecast (dac) in wireless sensor networks,” *Ad Hoc Networks*, vol. 5, no. 5, pp. 626–648, 2007.
- [59] S. Upadhyayula, V. Annamalai, and S. K. Gupta, “A low-latency and energy-efficient algorithm for convergecast in wireless sensor networks,” in *Global Telecommunications Conference, 2003. GLOBECOM’03. IEEE*, vol. 6. IEEE, 2003, pp. 3525–3530.
- [60] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, “Fast data collection in tree-based wireless sensor networks,” *Mobile Computing, IEEE Transactions on*, vol. 11, no. 1, pp. 86–99, 2012.
- [61] A. Iyer, C. Rosenberg, and A. Karnik, “What is the right model for wireless channel interference?” *Wireless Communications, IEEE Transactions on*, vol. 8, no. 5, pp. 2662–2671, 2009.
- [62] G. Brar, D. M. Blough, and P. Santi, “Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks,” in *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 2006, pp. 2–13.
- [63] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, “Successive interference cancellation: a back-of-the-envelope perspective,” in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, 2010, p. 17.
- [64] X. Tang and J. Xu, “Optimizing lifetime for continuous data aggregation with precision guarantees in wireless sensor networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 4, pp. 904–917, 2008.
- [65] J. Zhu, S. Papavassiliou, and J. Yang, “Adaptive localized qos-constrained data aggregation and processing in distributed sensor networks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 17, no. 9, pp. 923–933, 2006.
- [66] B. Yu, J. Li, and Y. Li, “Distributed data aggregation scheduling in wireless sensor networks,” in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 2159–2167.
- [67] P. Djukic and S. Valaee, “Delay aware link scheduling for multi-hop tdma wireless networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 3, pp. 870–883, 2009.
- [68] S. Gandham, M. Dawande, and R. Prakash, “Link scheduling in wireless sensor networks: distributed edge-coloring revisited,” *Journal of Parallel and Distributed Computing*, vol. 68, no. 8, pp. 1122–1134, 2008.
- [69] S. C.-H. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang, “Minimum-latency broadcast

- scheduling in wireless ad hoc networks,” in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE. IEEE, 2007, pp. 733–739.
- [70] R. Mahjourian, F. Chen, R. Tiwari, H. Zhai, and Y. Fang, “An approximation algorithm for conflict-aware broadcast scheduling in wireless ad hoc networks,” in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2008, pp. 331–340.
- [71] S. XIAO, J. PEI, X. CHEN, and W. WANG, “Minimum latency broadcast in the sinr model: A parallel routing and scheduling approach,” *IEEE communications letters*, vol. 18, no. 6, pp. 1027–1030, 2014.
- [72] P. Marbach, A. Eryilmaz, and A. Ozdaglar, “Achievable rate region of csma schedulers in wireless networks with primary interference constraints,” in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 1156–1161.
- [73] F. Kuhn, R. Wattenhofer, and A. Zollinger, “Ad-hoc networks beyond unit disk graphs,” in *Proceedings of the 2003 joint workshop on Foundations of mobile computing*. ACM, 2003, pp. 69–78.
- [74] J. Mao, Z. Wu, and X. Wu, “A tdma scheduling scheme for many-to-one communications in wireless sensor networks,” *Computer Communications*, vol. 30, no. 4, pp. 863–872, 2007.
- [75] Y. Shi, Y. T. Hou, J. Liu, and S. Kompella, “How to correctly use the protocol interference model for multi-hop wireless networks,” in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2009, pp. 239–248.
- [76] O. Goussevskaia, R. Wattenhofer, M. M. Halldórsson, and E. Welzl, “Capacity of arbitrary wireless networks,” in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 1872–1880.
- [77] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, 2000.
- [78] S. Li, Y. Liu, and X.-Y. Li, “Capacity of large scale wireless networks under gaussian channel model,” in *Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 2008, pp. 140–151.
- [79] H. Gudmundsdottir, E. I. Asgeirsson, M. H. Bodlaender, J. T. Foley, M. M. Halldórsson, G. M. Järvälä, H. Ulfarsson, and Y. Vigfusson, “Measurement based interference models for wireless scheduling algorithms,” *arXiv preprint arXiv:1401.1723*, 2014.
- [80] S. Guo, Y. Gu, B. Jiang, and T. He, “Opportunistic flooding in low-duty-cycle wireless

- sensor networks with unreliable links,” in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 133–144.
- [81] S. Y. Shin, H. S. Park, and W. H. Kwon, “Mutual interference analysis of ieee 802.15. 4 and ieee 802.11 b,” *Computer Networks*, vol. 51, no. 12, pp. 3338–3353, 2007.
- [82] M. K. An, N. X. Lam, D. T. Huynh, and T. N. Nguyen, “Minimum latency data aggregation in the physical interference model,” *Computer Communications*, vol. 35, no. 18, pp. 2175 – 2186, 2012.
- [83] M. Andrews and M. Dinitz, “Maximizing capacity in arbitrary wireless networks in the sinr model: Complexity and game theory,” in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 1332–1340.
- [84] D. N. M. Dang, C. S. Hong, S. Lee, and J. Lee, “A sinr-based mac protocol for wireless ad hoc networks,” *Communications Letters, IEEE*, vol. 16, no. 12, pp. 2016–2019, 2012.
- [85] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer, “Local broadcasting in the physical interference model,” in *Proceedings of the fifth international workshop on Foundations of mobile computing*. ACM, 2008, pp. 35–44.
- [86] D. Yu, Y. Wang, Q.-S. Hua, and F. C. Lau, “Distributed local broadcasting algorithms in the physical interference model,” in *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*. IEEE, 2011, pp. 1–8.
- [87] S. Chen, M. Huang, S. Tang, and Y. Wang, “Capacity of data collection in arbitrary wireless sensor networks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 1, pp. 52–60, 2012.
- [88] S. Ji, R. Beyah, and Y. Li, “Continuous data collection capacity of wireless sensor networks under physical interference model,” in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*. IEEE, 2011, pp. 222–231.
- [89] H. Li, C. Wu, Q.-S. Hua, and F. Lau, “Latency-minimizing data aggregation in wireless sensor networks under physical interference model,” *Ad Hoc Networks*, vol. 23, no. 2, pp. 123–135, 2011.
- [90] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, “Complexity in geometric sinr,” in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2007, pp. 100–109.
- [91] E. C. Titchmarsh, *The theory of the Riemann zeta-function*. Oxford Oxford University Press, 1951, vol. 196.
- [92] S. P. Weber, J. G. Andrews, X. Yang, and G. De Veciana, “Transmission capacity of

- wireless ad hoc networks with successive interference cancellation,” *Information Theory, IEEE Transactions on*, vol. 53, no. 8, pp. 2799–2814, 2007.
- [93] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi, “Successive interference cancellation: Carving out mac layer opportunities,” *Mobile Computing, IEEE Transactions on*, vol. 12, no. 2, pp. 346–357, 2013.
- [94] T. Cover, “Broadcast channels,” *Information Theory, IEEE Transactions on*, vol. 18, no. 1, pp. 2–14, 1972.
- [95] J. G. Andrews, “Interference cancellation for cellular systems: a contemporary overview,” *Wireless Communications, IEEE*, vol. 12, no. 2, pp. 19–29, 2005.
- [96] X. Zhang and M. Haenggi, “The aggregate throughput in random wireless networks with successive interference cancellation,” in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 251–255.
- [97] C. Xu, L. Ping, P. Wang, S. Chan, and X. Lin, “Decentralized power control for random access with successive interference cancellation,” *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 11, pp. 2387–2396, 2013.
- [98] O. Goussevskaia and R. Wattenhofer, “Scheduling wireless links with successive interference cancellation,” in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*. IEEE, 2012, pp. 1–7.
- [99] S. Lv, W. Zhuang, X. Wang, C. Liu, X. Hu, Y. Sun, and X. Zhou, “A performance study of csma in wireless networks with successive interference cancellation,” in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 466–471.
- [100] S. Lv, W. Zhuang, M. Xu, X. Wang, C. Liu, and X. Zhou, “Understanding the scheduling performance in wireless networks with successive interference cancellation,” *Mobile Computing, IEEE Transactions on*, vol. 12, no. 8, pp. 1625–1639, 2013.
- [101] M. Mollanoori and M. Ghaderi, “Uplink scheduling in wireless networks with successive interference cancellation,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1132–1144, 2014.
- [102] D. Yuan, V. Angelakis, L. Chen, E. Karipidis, and E. G. Larsson, “On optimal link activation with interference cancelation in wireless networking,” *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 2, pp. 939–945, 2013.
- [103] C. Jiang, Y. Shi, Y. T. Hou, W. Lou, S. Kompella, and S. F. Midkiff, “Squeezing the most out of interference: An optimization framework for joint interference exploitation and avoidance,” in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 424–432.

- [104]H. Li, C. Wu, D. Yu, Q.-S. Hua, and F. C. Lau, “Aggregation latency-energy tradeoff in wireless sensor networks with successive interference cancellation,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 11, pp. 2160–2170, 2013.
- [105]S. Xiao, L. Pan, Y. Qiu, and Y. Xiaobing, “Minimum latency aggregation scheduling in wireless sensor networks with successive interference cancellation,” *Ad Hoc & Sensor Wireless Networks*, vol. 26, no. 1-4, pp. 233–257, 2015.
- [106]F. Wang and J. Liu, “Duty-cycle-aware broadcast in wireless sensor networks,” in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 468–476.
- [107]T.-H. Hsu, T.-H. Kim, C.-C. Chen, and J.-S. Wu, “A dynamic traffic-aware duty cycle adjustment mac protocol for energy conserving in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.
- [108]M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks,” in *Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006, pp. 307–320.
- [109]D. Gong, Y. Yang, and Z. Pan, “Energy-efficient clustering in lossy wireless sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 73, no. 9, pp. 1323–1336, 2013.
- [110]S. Ji, R. Beyah, and Z. Cai, “Snapshot/continuous data collection capacity for large-scale probabilistic wireless sensor networks,” in *Proceedings of IEEE INFOCOM*, 2012, pp. 1035–1043.
- [111]G. Xing, C. Lu, X. Jia, and R. Pless, “Localized and configurable topology control in lossy wireless sensor networks,” *Ad Hoc Networks*, vol. 11, no. 4, pp. 1345–1358, 2013.
- [112]K. Han, J. Luo, Y. Liu, and A. Vasilakos, “Algorithm design for data communications in duty-cycled wireless sensor networks: A survey,” *Communications Magazine, IEEE*, vol. 51, no. 7, pp. 103–116, 2013.
- [113]J. Hao, B. Zhang, and H. T. Mouftah, “Routing protocols for duty cycled wireless sensor networks: A survey,” *Communications Magazine, IEEE*, vol. 50, no. 12, pp. 116–123, 2012.
- [114]Y. Cao, S. Guo, and T. He, “Robust multi-pipeline scheduling in low-duty-cycle wireless sensor networks,” in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 361–369.
- [115]Z. Cao, Y. He, and Y. Liu, “L2: Lazy forwarding in low duty cycle wireless sensor

- networks,” in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1323–1331.
- [116]Y. Gu and T. He, “Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 12, pp. 1741–1754, 2011.
- [117]K. Han, Y. Liu, and J. Luo, “Duty-cycle-aware minimum-energy multicasting in wireless sensor networks,” *Networking, IEEE/ACM Transactions on*, vol. 21, no. 3, pp. 910–923, 2013.
- [118]D. Zhao and K.-W. Chin, “Approximation algorithm for data broadcasting in duty cycled multi-hop wireless networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, no. 1, pp. 1–14, 2013.
- [119]S. Xiao, J. Huang, L. Pan, Y. Cheng, and J. Liu, “On centralized and distributed algorithms for minimizing data aggregation time in duty-cycled wireless sensor networks,” *Wireless Networks*, vol. 20, no. 7, pp. 1729–1741, 2014.
- [120]Z. Li, Y. Peng, D. Qiao, and W. Zhang, “Lba: Lifetime balanced data aggregation in low duty cycle sensor networks,” in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1844–1852.
- [121]R. R. Rout and S. K. Ghosh, “Enhancement of lifetime using duty cycle and network coding in wireless sensor networks,” *Wireless Communications, IEEE Transactions on*, vol. 12, no. 2, pp. 656–667, 2013.
- [122]H. Yousefi, M. H. Yeganeh, and A. Movaghar, “Long lifetime routing in unreliable wireless sensor networks,” in *IEEE International Conference on Networking, Sensing and Control*, 2011, pp. 457–462.
- [123]R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, “Four-bit wireless link estimation.” in *HotNets*, 2007.
- [124]S. Madden, “Intel berkeley research lab data,” USA: Intel Corporation, <http://berkeley.intel-research.net/labdata.html>, 2003.
- [125]W. Qu, K. Li, M. Kitsuregawa, and T. Nanya, “An efficient method for improving data collection precision in lifetime-adaptive wireless sensor networks,” in *Communications, 2007. ICC'07. IEEE International Conference on*. IEEE, 2007, pp. 3161–3166.
- [126]D.-Z. Du, K.-I. Ko, and X. Hu, *Design and analysis of approximation algorithms*. Springer Science & Business Media, 2011, vol. 62.



## 攻读学位期间发表的学术论文与研究成果

### 学术论文：

- [J1] **Shiliang Xiao\***, Baoqing Li, and Xiaobing Yuan, “Maximizing Precision for Energy-Efficient Data Aggregation in Wireless Sensor Networks with Lossy Links”, **Ad Hoc Networks**, vol. 26, no. 3, pp. 103-113, 2015. (SCI, IF=1.943)
- [J2] **Shiliang Xiao\***, Lebing Pan, Yunzhou Qiu, and Xiaobing Yuan, “Minimum Latency Aggregation Scheduling in Wireless Sensor Networks with Successive Interference Cancellation”, **Ad Hoc & Sensor Wireless Networks**, vol. 26, no. 1, pp. 233-257, 2015. (SCI, IF=0.478)
- [J3] **Shiliang Xiao\***, Jingchang Huang, et al., “On centralized and distributed algorithms for minimizing data aggregation time in duty-cycled wireless sensor networks”, **ACM/Springer Wireless Networks**, vol. 20, no. 7, pp. 1729-1741, 2014. (SCI, IF=1.055)
- [J4] **Shiliang Xiao\***, Jun Pei, et al., “Minimum Latency Broadcast in the SINR model: A Parallel Routing and Scheduling Approach”, **IEEE Communications Letters**, vol. 18, no. 6, pp. 1027-1030, 2014. (SCI, IF=1.463)
- [J5] Jingchang Huang\*, **Shiliang Xiao**, Qianwei Zhou, et al., “A Robust Feature Extraction Algorithm for Automatic Classification of Acoustic Target in Wild Environment”, accepted, **Springer Circuits, Systems, and Signal Processing**, 2014. (SCI)
- [J6] Lebing Pan\*, **Shiliang Xiao**, Xiaobing Yuan, et al., “Joint Multiband Signal Detection and Cyclic Spectrum Estimation From Compressive Samples”, accepted, **EURASIP Journal on Wireless Communications and Networking**, 2014. (SCI)
- [J7] **Shiliang Xiao\***, Lebing Pan, Jianpo Liu, et al., “Distributed Broadcast with Minimum Latency in Asynchronous Wireless Sensor Networks under SINR-based Interference”, **International Journal of Distributed Sensor Networks**, vol. 2013, pp.1-12, 2013. (SCI, IF=0.923)

\*代表通信作者

### 专利：

- [P1] 肖世良、程小六、吴光等, 面向无线传感器网络数据收集的半结构化传输调度方法, 发明专利, 申请号: 201410550746.x
- [P2] 程勇博、肖世良、钱汉旺等, 一种无线多媒体传感器网络接入 IP 网络的方法,

发明专利，申请号: 201410486749.1

[P3] 潘乐炳、**肖世良**、赵龙慧等，一种基于多通道压缩采样的宽带功率谱估计方法，  
发明专利，申请号: 2014103905674

[P4] 程小六、**肖世良**、王艳等，一种微气象环境及物理信息实时监测通用平台，发明  
专利，申请号: 201410394810x

[P5] 潘乐炳、**肖世良**、刘建坡等，一种基于欠乃奎斯特采样的宽带信号检测与识别  
方法，发明专利，申请号: 201410466563x

[P6] 邱云周、秦文霞、**肖世良**等，脉搏监控设备及系统，发明专利，申请号：  
201310196430.0

## 作者简介

姓名：肖世良

性别：男

出生年月：1988.07

籍贯：湖北省黄冈市

研究方向：无线网络、物联网、算法设计与分析

个人简历：

2010.09-2015.06 中国科学院上海微系统与信息技术研究所无线传感网实验室通信与信息系统专业(硕博连读)

2010.09-2011.06 中国科学技术大学，通信与信息系统专业(硕士代培)

2006.09-2010.07 浙江大学，电子信息技术及仪器专业(本科)

# 附录

---

中国科学院上海微系统与信息技术研究所

### 学位论文独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中国科学院上海微系统与信息技术研究所或其它教育机构的学位或证书而使用过的材料。与我一起工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名：\_\_\_\_\_日期：\_\_\_\_\_

中国科学院上海微系统与信息技术研究所

### 学位论文使用授权声明

本人完全了解中国科学院上海微系统与信息技术研究所有关保留、使用学位论文的规定，即研究所有权保留递交论文的复印件，允许论文被查阅和借阅；可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。保密的论文在解密后遵守此规定。论文的公布（包括刊登）授权中国科学院上海微系统与信息技术研究所人才教育处。

研究生签名：\_\_\_\_\_ 导师签名：\_\_\_\_\_ 日期：\_\_\_\_\_