

单位代码: 10293 密 级:           



# 南京邮电大学

## 硕士学位论文

论文题目: 面向无线传感器网络的多路径路由协议研究

---

学	号	<u>Y004091317</u>
姓	名	<u>龚瑜</u>
导	师	<u>宗平</u>
学 科 专 业		<u>计算机应用技术</u>
研 究 方 向		<u>计算机网络与分布计算系统</u>
申请学位类别		<u>工学硕士</u>
论文提交日期		<u>二〇一二年三月</u>

## 南京邮电大学学位论文原创性声明

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，也不包含为获得南京邮电大学或其它教育机构的学位或证书而使用过的材料。与我一同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。本人学位论文及涉及相关资料若有不实，愿意承担一切相关的法律责任。

研究生签名：\_\_\_\_\_ 日期：\_\_\_\_\_

## 南京邮电大学学位论文使用授权声明

本人授权南京邮电大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档；允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索；可以采用影印、缩印或扫描等复制手段保存、汇编本学位论文。本文电子文档的内容和纸质论文的内容相一致。论文的公布（包括刊登）授权南京邮电大学研究生院（筹）办理。

涉密学位论文在解密后适用本授权书。

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

南 京 邮 电 大 学

硕士学位论文摘要

学科、专业：工 学 计算机应用技术

研 究 方 向：分布计算技术与应用

作 者：2009 级研究生 龚瑜

指 导 教 师：宗平 教授

题 目：面向无线传感器网络的多路径路由协议研究

英 文 题 目：Study of Multi-Routing Protocols for WSN

关 键 字：无线传感器网络，多路径路由，蚁群算法，信息素

Keywords: WSN, Multi-Path Routing, Ant Colony Algorithm, Pheromone

## 摘 要

本文根据 WSN 网络自身的特征，给出了路由协议所需要的设计原则，详细分析了多路径路由协议，基于这些多路径路由协议的特点，开展了针对 SMR 协议的进一步研究。同时针对基本的蚁群算法存在过快收敛的问题，提出了一个优化蚁群算法的设计方案，通过变更全局信息素更新规则、公式以及局部信息素更新公式，实现了只在有最优路径和最差路径时才进行信息素的更新，并且限定信息素的浓度范围，若不属于该范围则重置的优化策略。

本文将改进后的蚁群优化策略加入到 SMR 协议中，得到一种改进的多路径路由算法 AIS-SMR。该算法不仅保留了 SMR 算法的优点，还能通过对信息素浓度的大小判断出被选作下一跳对象的可能性，同时跳数、延时、剩余能量也将加入到相应的路由计算中。SMR 在无路由信息时是向全网广播，改进的路由协议只需向邻节点广播，并且加入了路由更新阶段，在数据包发送和定时两种情况下对路径信息进行更新，更好地实现稳定性，均衡负载，有效地利用资源，延长了生存周期。本论文对 AIS-SMR 进行了仿真实验，通过仿真结果分析可知，AIS-SMR 能很好地发现最优路径，达到节能增效、稳定路由和延长网络生命周期的目的。

**关键字：**无线传感器网络，多路径路由，蚁群算法，信息素

## Abstract

According to the characteristics of WSN, this thesis presents the design requirements of the routing protocols and analyzes the several typical multipath routing protocols in detail. Based on the characteristics of these multipath routing protocols, this thesis mainly studies SMR algorithm. The convergence of the basic ant colony algorithm is fast, so this thesis presents an improved ant colony optimization algorithm AIS, which not only updates global pheromone newer rules and formula, but also updates local pheromone newer formula. Only the best and worst path can update their formula. Moreover, if the concentration of formula is beyond its defined limits, its value will be reset.

By adding the improved biological simulation strategy to SMR algorithm, this thesis proposes an improved routing protocol algorithm called AIS-SMR. This algorithm retains the advantages of SMR algorithm and can also calculate the probability of a next node by the concentration of formula. Hop counts, time-delay and residual energy make up the calculation of routing. When there is no routing information, SMR will broadcast to the entire network, but AIS-SMR algorithm will only broadcast to neighbor nodes and adds a routing updated stage. In this stage, when data packets are being set and it is at regular time, the information of paths will be updated. This will achieve the selection of stable paths, balance load, improve the effective use of resources and extend lifetime of network. This thesis makes the simulation for AIS-SMR algorithm. Through analyzing the results of the simulation, AIS-SMR algorithm can find out the optimal path better, achieve the purposes of reducing energy consumption, increasing the efficiency, selecting the stable path and prolonging lifetime of network.

**Keywords:** WSN, Multi-Path Routing, Ant Colony Algorithm, Pheromone

# 目 录

<b>第一章 绪论</b>	<b>1</b>
1.1 研究背景	1
1.2 研究现状	3
1.3 本文的主要工作及组织结构	4
<b>第二章 无线传感器网络</b>	<b>6</b>
2.1 无线传感器网络的体系结构	6
2.2 无线传感器网络的协议栈结构	7
2.3 无线传感器网络路由协议概述	9
2.3.1 无线传感器网络路由协议的分类	9
2.3.2 典型多路径路由协议的分析与比较	11
2.4 本章小结	14
<b>第三章 蚁群算法概述</b>	<b>15</b>
3.1 蚁群算法简介	15
3.1.1 蚁群算法原理	15
3.1.2 蚁群算法的传统模型	16
3.2 蚁群算法的应用情况	18
3.3 典型的蚁群算法	19
3.4 在无线传感器网络路由算法中引入蚁群算法的优势	24
3.5 本章小结	25
<b>第四章 算法的改进</b>	<b>26</b>
4.1 基本蚁群算法的不足	26
4.2 改进的蚁群算法	27
4.3 AIS-SMR 多路径路由协议的实现	28
4.3.1 协议设计目标	28
4.3.2 协议改进思路	30
4.3.3 协议设计	31
4.3.4 算法流程	33
4.4 AIS-SMR 的结构设计	37
4.4.1 节点设计	37
4.4.2 报文设计	39
4.4.3 数据结构定义	41
4.5 AIS-SMR 算法的性能分析	44
4.6 本章小结	45
<b>第五章 仿真实验与分析</b>	<b>46</b>
5.1 NS2 仿真工具	46
5.2 NS2 仿真步骤	47
5.3 仿真参数设置	48

---

5.4 实验结果分析 .....	48
5.5 本章小结 .....	51
<b>第六章 总结与展望 .....</b>	<b>52</b>
6.1 总结 .....	52
6.2 展望 .....	53
<b>致 谢 .....</b>	<b>54</b>
<b>参考文献 .....</b>	<b>55</b>
<b>读研期间发表的论文 .....</b>	<b>59</b>

# 第一章 绪论

## 1.1 研究背景

无线传感器网络（Wireless Sensor Networks, WSN）拥有低功耗、低成本、分布式和自组织的特点<sup>[1-3]</sup>，它的出现引领了信息行业的一场革命风暴。散布在监控区域中的大量低成本的微型传感器节点构成了自组多跳、以无线方式进行通信的无线传感器网络。无线传感器网络的搭建运用了诸如低功耗设计技术、通信技术、检测与感知技术等多种技术。由于无线传感器网络中的节点具有自组织及多跳传输的特性，具有很大的市场潜力。无线传感器网络被《商业周刊》和《技术评论》两大杂志分别誉为对二十一世纪的发展最有影响的 21 项技术之一<sup>[4]</sup>和影响世界的 10 大技术之一<sup>[5]</sup>。由此可以看出无线传感器网络商业价值非常可观，引起各界人士举足轻重的关注也不足为奇。它的主要应用领域有：

### （1）军事领域

美国国防部以及其他相关部门非常的重视无线传感器网络，C4KISR 是基于 C4ISR（Command, Control, Communication, Computing, Intelligence, Surveillance and Reconnaissance）提出的，它突出了对信息的感知、综合和利用能力的强调。他们对该网络给予了极大的期望，并为此设立了关于军事方面的研究课题。

2001 年美国陆军提出一项“灵巧传感器网络通信计划”，它被授予为一项重要的科学技术研究计划，将用四年时间进行实施<sup>[6]</sup>。灵巧传感器网络通信计划是通过搭建一个完善的通信设施平台，给予前线以强力的支持，利用未来战斗的机器人系统、传感器和无人值守式弹药构成一个三位一体的网络，这样使传感器的能力得到了成倍的提高，作战指导员能够快速计划出更好的作战方针，使战斗系统更加灵活而富有生命。

同样地，美国海军也相当关注无线传感器网络，他们主要是开发和研究近海水域附近的水下无线传感器网络，通过其融合技术来探测敌方军舰情况，通过获得的信息来确定如何攻击敌方<sup>[7]</sup>。

无线传感器网络自组织、自恢复的特点，当部分节点受到攻击而损坏时，无线传感器网络也不会因此而崩溃，这种特性很适合环境十分恶劣的战场。军方可以使用传感器网络来监测战场中我方的兵力、物资和装备等情况；监测敌方的动态，能够确定敌方兵力的部署等



信息；通过生化方面的传感器，可以准确对敌方的生化袭击做出预警；可以通过对传感器感知到的信息进行整理和研究来协助判断整个战事的状态情况。

## （2）民事领域

传感器网络被广泛应用在医学、森林火灾、海洋开发、生命科学、环境监测、海底监测、行星探测、农业种植等多个领域中。比如动植物学家可以在对野生动植物及其周围环境监测的同时却不会影响他们的正常生活；城镇规划人员可以通过放置事先编制好程序的传感器在城市中来了解该地区的地质条件，这样他们就容易知道该如何设计该区域的建筑，使得这些建筑更具安全性。因为由于已经将检测地址信号的传感器放置于建筑周围，这样当地震即将来临时就会提前给出警报，居民就可以有时间来得及逃离；在森林中散布传感器节点，当火灾发生时，它就会立刻给出警报，消防人员可以通过传感器给出的信息迅速定位，大大减少了损失，甚至可以防止火灾的发生；通过在病人身上装置医疗传感器，即使医护人员不在病人身边，他们也可以随时知道病人的情况，收集病人的数据，当病人突然发生状况时，也可以及时地采取措施，拯救病人的生命。农民将传感器放置于土壤之中，可以实时监测土壤的情况，如湿度，酸碱度等等，有助于制定最合理的种植计划，也可以监测害虫的情况，防止害虫对农作物的伤害。

## （3）学术界

基于美国自然科学基金委员会的大力支持，美国的麻省理工大学、加州大学、康奈尔大学等高校都开始专研无线传感器网络的核心技术，在技术上做出了重大突破。英法德意等各个国家也开始着手无线传感器网络的研究工作。我国也开始启动了关于无线传感器网络的研发项目，已经从理论研究逐步转向协议的实现，并有不同的应用部署。

路由协议主要是用于构造源节点和目的节点之间传输数据的最优路径。传统的无线局域网和 Ad hoc 网络<sup>[9]</sup>为了使各终端的公平性得到保证，它提供的服务要求具有高质量和高效率。这些网络本来就具有计算源和目的节点之间最少时延路由的能力，使网络负载得以平衡。

传感器网络具有大量的传感器节点，如果构造一个关于全网的路由信息，节点就会因为地址信息带有大量负荷，这种方式往往不可取。由于节点的能量有限并且节点的计算能力也有限，传统无线网络和传感器无线网络不能共用同样的路由协议，就需要设计一个可靠的、能量高效的、能量消耗平衡的路由协议来适应无线传感器网络的特征。

无线传感器面对的应用是各种各样的，因此用于建立各种各样不同的最优路径的方法也会有所区别，国内和国外的研究人员为了适应各种场景，设计了很多不同的策略。由此看来，有效应用无线传感器网络的首要任务就是研究其路由协议。

国内外很多的研究人员已经为无线传感器网络的路由协议做了很多的实验和分析<sup>[8-10]</sup>。我们可以利用这些研究来得出无线传感器网络的特征，并且根据这些特征给予的信息、挑战和限制，对已有的路由协议进行分析和归纳，并且可利用已有算法的思想或将某些优化算法融入路由建立的策略之中，设计出性能更好、更适合无线传感器网络的路由算法，这一课题是十分有意义的。

## 1.2 研究现状

无线传感器网络可以应用于很多领域，可以对人们的生活产生一定的影响。现在已经有多个国家投入很大的资源和精力对它进行了一系列的研究，极其重视无线传感器网络的发展。

作为首个提出无线传感器网络概念的国家，美国军方希望利用该网络协助作战使用。二十世纪 70 年代后期，美国国防部高级研究所（DARPA）联合卡内基梅隆大学成立了分布式传感器网络研究组<sup>[11]</sup>，但是直到 1998 年才开始进行大量的研究工作，关于研究无线传感器网络的学术论文频繁出现在各大期刊上。美国的军事部门给予了无线传感器网络高度的重视，制定了大量关于无线传感器网络在军事领域方面的研究计划，比如：1993 年至 1999 年期间，DAPRA 赞助加州大学洛杉矶分校对 WINS<sup>[12]</sup>项目进行研究；1998 年至 2002 年期间，DAPRA 赞助二十多个研究机构一起对 Sense IT<sup>[13]</sup>项目进行研究；1999 年至 2001 年期间，DAPRA 赞助 UC Berkeley 对 Smart Datasets<sup>[14]</sup>进行研究；1999 年至 2004 年期间，海军研究所研究 Sea Web<sup>[15]</sup>项目。二十一世纪初期，美国自然科学基金委员会拿出三千多万美元对传感器网络的基础知识开展研究工作。一些高等院校比如麻省理工学院，加州大学伯克利分校都对其展开了研究。微软、英特尔等大公司慢慢开始研究传感网，开始制定一些相应的研究计划。2002 年英特尔计划实施研究微型传感器网络的计算；2003 年 NSF 公司也开始计划研究传感网。美国某些公司开发的“Mote”、“智能尘埃”等项目已经进行测试，即将投入应用。

日本将开发和利用传感器技术列为国家重点发展 6 大核心技术之一。日本文部科学省制定的 20 世纪 90 年代重点科研项目中有 70 个重点课题，其中有 18 项与传感器技术密切相关。德国 80 项优先资助的计划中，两项为传感器计划，一项为微型化传感器，另一项为生物传感器。代表欧洲国家在高新技术领域的整体研究趋向的计划有 29 个项目直接与传感器技术相关。欧盟已经把传感器技术作为带动各领域技术水平提升的关键性技术来看待，而且在传感器技术的研究中非常重视传感器技术与其他高新技术的交叉研究。其他国家也已经开始了多项无线传感器网络的应用研究。

我国无线传感器网络及其应用研究几乎与发达国家同步启动,首次正式出现于 1999 年中国科学院《知识创新工程试点领域方向研究》的“信息与自动化领域研究报告”中。2004 年中国国家自然科学基金委员会将一项无线传感器网络项目(面上传感器网络的分布自治系统关键技术及协调控制理论)列为重点研究项目;2005 年将无线传感器网络基础理论和关键技术列入计划;2006 年将水下移动传感器网络的关键技术列为重点研究项目。国家发改委下一代互联网(CNGI)示范工程中,也部署了无线传感器网络相关的课题。2006 年初发布的《国家中长期科学与技术发展规划纲要》为信息技术定义了 3 个前沿方向,其中 2 个与无线传感器网络的研究直接相关,即智能感知技术和自组织网络技术<sup>[16]</sup>。国内各研究院、各大高校等已经在这个领域进行了研究,相关研究成果取得了很大的进展。国内也有越来越多的企业开始关注无线传感器网络技术的发展,开始推出针对无线传感器网络及 ZigBee 的解决方案。目前 WSN 的研究热点主要有:网络体系结构路由及数据分发,链路层协议 WSN 和 IPv6 的结合,数据的采集捕获技术,融合处理技术和数据库技术,功耗管理,定位技术安全,适合 WSN 的软件开发环境和软件工具,模拟器和应用系统的开发等。其中路由协议占有十分重要的地位。

### 1.3 本文的主要工作及组织结构

本文的主要工作如下:

本文首先对无线传感器网络进行了介绍,包括应用领域、研究现状、体系结构等;然后给出了设计无线传感器网络的路由协议需要遵守的基本规则;讨论了无线传感器网络路由协议的分类,说明了多路径路由协议应用于无线传感器网络的优势,分析了多种典型的多路径路由协议,提出了一种一个多路径路由协议的优化策略。面对现有的多路径路由协议所面临的不足,本文结合无线传感网络的特征阐述了如何将蚁群算法应用于路由协议改进的设计思想,并给出了将蚁群算法应用于多路径路由协议的设计与实现方法,同时仿真和比较了改进前后的多路径路由协议的性能。

本文的章节安排如下:

第一章绪论。主要是说明了论文研究的背景和目的,概述了无线传感器网络的研究现状,同时简单地给出了论文的主要工作与组织结构。

第二章无线传感器网络。讨论了无线传感器网络的体系结构、协议栈、路由协议的分类和典型的多路径路由协议。

第三章蚁群算法概述。讨论了蚁群算法的基本原理和模型，典型的蚁群算法和蚁群算法应用于无线传感网络路由协议的优势。

第四章算法的改进。在分析蚁群算法特点的基础上，提出蚁群算法的改进策略，并将改进后的蚁群算法用于 SMR 多路径路由协议，得到一种新的 AIS-SMR 路由协议，详细说明了相关的设计思想，流程图与伪代码和报文设计等的实现方法。

第五章仿真实验与分析。应用 NS2 对改进的路由协议进行仿真，并和 SMR 协议进行了比较和分析。

第六章对全文进行总结，并对下一步工作进行了展望。

## 第二章 无线传感器网络

### 2.1 无线传感器网络的体系结构

随着嵌入式技术、现代网络、传感器、无线通信以及分布式信息处理等技术的飞速发展，以低功耗、低成本为特点的无线传感器网络也随之应运而生。由大量分散在监测区域内的、低成本的微型传感器节点构成了无线传感器网络，以节点与节点之间无线通信搭建一个自组织结构的网络。该网络规模大、自组织能力强、动态性好、可靠性高、能源消耗低且不需要固定网络支撑。而内置于节点中的各式各样的传感器可以采集不同对象的参数，如距离、加速度、声波、应力等，通过自组织网络收集并传输数据。无线传感器网络属于无线多跳网络，可以通过多个相续的节点传递数据，信息传输迅速，节省能量、容错率高、成本低。通过传感器网络对监测区域中对象信息的感知、采集和处理，观察者可以了解其中的情况<sup>[17]</sup>。

无线传感器网络中的各个节点被随意地分布在 WSN 感知区域内。各个节点之间构成自组织网络，并以多跳的方式将感知和收集的信息传递至节点，最后通过互联网或卫星将信息传递至任务管理节点进行信息的整合及处理。在此过程中，各节点之间不仅仅是简单的信息感知、收集和传递，而且需要相互协同工作，相互传递和整合信息并合理的反馈。无线传感器网络结构如图 2.1 所示。

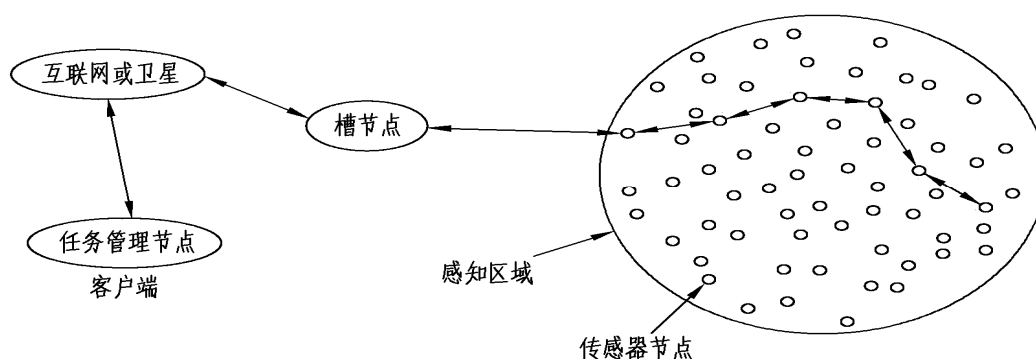


图 2.1 无线传感器网络结构

传感器节点以自组织的形式构成了无线传感器网络。通常情况下，传感器、处理器、无线通信和能量供应模块构成了传感器节点。其中，根据不同的感知对象，会使用不同类型的传感器，感知和收集的信息不同，因而节点的功能也会不同。一般而言，传感器将感知和收集到的信息通过 AC/DC 转换传递至处理器进行信息处理，之后再处理过的信息传递给无线

通信模块。能量供应模块则负责向传感器、处理器和无线通信模块提供能量。无线传感器节点结构如图 2.2 所示。

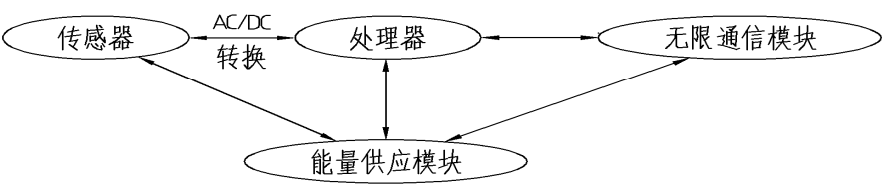


图 2.2 无线传感器节点结构

2.2 无线传感器网络的协议栈结构

图 2.3 是目前普遍使用的协议栈模型。由图 2.3 可见，水平方向上由物理层、数据链路层、网络层、传输层、应用层五层构成，用于网络的通信。垂直方向上由能量管理、移动管理、任务管理三大平台构成，用于网络的管理。

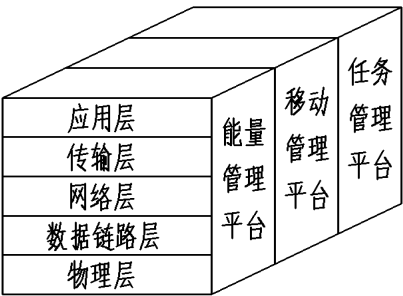


图 2.3 无线传感器网络协议栈结构

（1）物理层

作为无线传感器网络的物理基础，物理层借助数据通信技术，运用一些物理媒体来完成最基本的任务，即透明地传送比特流。

物理层需完成收集数据，对数据进行抽样处理，同时要对信号进行调制解调，并且发送接收信号等任务。由于无线传感器网络不同于一般的网络，传感器节点能量有限，节省能量、延长节点的生存周期十分重要，作为决定这些关键点的物理层必然成为无线传感器网络的研究的重中之重。

（2）数据链路层

数据链路层的协议数据单元是帧，其最根本的任务是将网络成交下来的数据发送到链路上，以及把接收到的帧中的数据取出并上交给网络层。

数据链路层负责对相邻节点之间的物理链路进行控制、差错控制、检测数据帧以及媒体访问控制，有利于节点对资源能够公平、有效、合理的共享，并保证信息能够准确的在各个网络节点中进行传递。MAC 协议是建立可靠通信链路的重要组成部分，因此链路层的设计重点要放在 MAC 协议上。由于无线传感器网络自身的特点，传统的 MAC 协议无法达到其要求，该协议的多次握手与节约能量的原则是相悖的。因此合理与高效的 MAC 协议的研究是无线传感器网络研究的核心之一。

### （3）网络层

与传统的网络层一样，无线传感器网络的网络层最主要的任务也是实现两个端系统之间数据的透明传输，因此无线传感器网络最主要的任务就是发现和维护路由。

由于无线传感器网络自身的特点，它不同于像 Ad hoc、无线局域网这类传统无线网络，它的首要目标不再仅仅是提供优异的服务和平等效率地分享网络资源，能耗问题也是它需要考虑的首要问题。因为传感器节点是无线传感器网络的核心，节点不仅仅用于收集数据，而且还扮演着路由器的角色，因此脱离节点，传感器网络就没有任何意义。一般而言，节点的能量有限而且无法补充，因此能量合理高效的使用是路由协议需要考虑的重点之一。多跳通信相比于直接通信更加节省能量，因此传感器节点通常使用多跳路由与其他节点连接起来，同时这也恰好满足了协同信号处理以及数据融合的要求。通常情况下，传感器网络需要大量的节点做支撑，节点只能感知和收集局部的拓扑信息，根据这些信息，路由协议需要找到一条合理而又节能的路由路径。

### （4）传输层

传输层是对数据流端对端层次的传输控制。基于网络层的基础上，传输层需确保其向应用层提供的数据传输的可靠性、节能性、透明性以及高质量性，需避免数据传输时发生丢失、乱序、重复、延迟时间紊乱等。同时，传输层还含带一定的缓冲能力。传输层是无线传感器网络的第四层，在网络层以及数据链路层之上。假如网络层和数据链路层的服务质量无法达标，那么传输层就对该服务进行提高，用以满足应用层的要求；反之，当网络层和数据链路层提供高可靠性、高质量性的服务时，那么传输层只需完成小额的工作任务。同时，一个网络连接可允许不同逻辑连接的同时发生。

目前，传感器网络节点的硬件条件有限，每个节点的信息存储量有限，无法像 Internet 上的服务器一样进行大量信息的储存，导致传输层的发展受到一定的限制，遭遇一定的困难。

在已有的成果中，通常会存在一个电源及硬件配置不同于普通节点的特殊节点，将其当作网关，使用卫星、Internet 或者其它链路传输数据。

#### （5）应用层

应用层作为无线传感器网络的最高层，直接为应用程序提供相应的服务。应用层使用面的广泛性导致应用层的不成熟性。应用层主要通过各种端上应用程序的接口为客户提供各种服务，例如时间同步、网络管理、应用服务以及节点定位等。其中，各节点单独能力的局限性，时间同步的作用就是调节各个节点，使之相互配合，以满足系统的要求。此外，时间同步还可控制节点的睡眠和唤醒。节点定位是指通过有限的已知的节点，确定感知区域中其余节点的相对或者绝对位置，从而保证对感知对象的精确定位。

#### （6）能量管理平台

能量管理平台主要负责对各传感器节点使用能量的调节、分配，使各协议层有效合理地使用能量，达到节能的目的。

#### （7）移动管理平台

移动管理平台主要负责对各传感器节点的移动进行检测和追踪，一直到传感器节点集中的路由。

#### （8）任务管理平台

任务管理平台主要负责平衡、调度和检测给定感知区域内的任务。

## 2.3 无线传感器网络路由协议概述

无线传感器网络属于任务型的网络，脱离节点讨论传感器网络毫无价值。网络中的传感器节点通过编号标识，网络通信协议的配置决定了节点编号在整个网络中的唯一性。用户通过传感器网络进行事件查询的时候，会直接向网络进行关注事件的报告，而非向某个固定节点进行报告，网络在收到事件的信息后通知用户。由此可见传感器网络的中心就是数据。传感器网络拥有大量的节点，导致它无法获得全局的地址；大部分的应用中，少量节点的位置会有所变动，大部分就不会再有什么变动；特定的应用会形成相应的路由协议。

### 2.3.1 无线传感器网络路由协议的分类

无线传感器网络路由协议的分类依据分类标准的不同而不同，其分类方法与传统 Ad hoc 网的类似。



(1) 依据网络管理的逻辑结构, 可划分成平面路由和分层路由两类。

平面路由: 所有节点收集和传递数据至汇聚节点, 各个节点的地位、功能一样, 相互间协作地完成对数据的感知和传递的任务。此种路由算法较为简单, 便于实现, 但是用于路由维护的能耗大, 数据传递跳数多, 扩展性差, 只可用于小型网络。大型网络中, 传感器节点的密度大, 汇聚节点任务加重, 又单个节点能量有限, 只适合短距离通信, 数据传递方式必为多跳方式。由此可见, 此种结构不适用于大型网络。

层次路由: 各个节点以不同的分簇规则进行分簇, 划分成相应的簇。各个簇挑选出簇头节点, 通过多跳的方式进行通信, 再将传递的数据融合, 以达到减少数据传递次数, 减少能量损耗, 有效延长生存周期的目的。此类路由的可扩展性较好, 适用于大型网络。但簇头节点能耗大, 且是整个网络的核心节点, 如何提高簇头节点的可靠性、节约簇头节点的能耗是研究的热点之一。

(2) 依据路由发现策略, 可划分为主动路由和按需路由两类。

主动路由: 节点按照一定规律广播并交换路由信息, 以此发现路由。其优点非常明显, 节点传递数据, 只要存在一条通往目的节点的路由, 需要的延时小。缺点亦很明显, 若当前网络拓扑结构产生变动, 网络会主动的使用较大的能量更新路由, 其中包含从未使用过的路由, 这样会造成能量的浪费。

按需路由: 与主动路由的工作原理相反, 此种路由只在工作的时候去“按需”地发现路由, 在非工作状态下不进行信息存储。因无需周期性地广播和交换路由信息, 所以相对地节约了能耗, 延长了生存周期, 但也会导致延时较大。

(3) 依据应用的不同, 可划分为能量感知路由、基于查询的路由、地理位置路由和可靠路由。

能量感知路由: 从能量损耗的角度出发, 研究降低路由能量损耗和延长网络的生存寿命。此类路由相较于传统路由, 具有优秀的可扩展性, 能量损耗也降低数倍。

基于查询的路由: 在某些环境监测的应用中, 需不间断地查取节点感知的数据。查询节点向传感器节点发出命令, 传感器节点接受命令并反馈数据, 与此同时, 数据在传递的路由上可能会进行融合, 以此来减少通信量, 从而达到节约能量的目的。

地理位置路由: 利用传感器节点的地里位置以及目的节点的地理位置作为挑选路由的参照, 数据通过既定的策略进行传递。此类路由可显著减少数据传递次数, 降低能量损耗。

可靠路由: 在诸如传递视频等应用时, 无线传感器网络会对通信质量提出高要求, 以保证数据传递的可靠性。在通常的无线传感器网络中, 拓扑变化多, 路由稳定性差, 通信质量

低。要保证通信质量，保证数据传递的稳定性、及时性、可靠性，则需有针对性的设计高通信质量的路由协议。

(4) 依据路径数量的不同，可划分为单路径路由和多路径路由<sup>[18-22]</sup>两类。

单路径路由：是一种简单易于实现的，且便于管理的路由。该类路由对资源的利用率低，开销大，数据传输效率低，无法保障网络的通畅。

多路径路由：因网络中的所有节点都有路由功能，所以源节点可通过多条路径到达目的节点，提高了网络的容错能力。多路径路由可发展出不同的多路径协议，以满足环境的需求，如平衡网络负载，延长网络寿命，缩短网络延时等。

### 2.3.2 典型多路径路由协议的分析与比较

#### (1) 泛洪 (Flooding) 协议

泛洪 (Flooding) 协议<sup>[23]</sup>作为传统而典型的路由协议，泛洪算法在无线传感器网络中的应用大同小异。传感器节点感知或收集数据后，无需挑选路由策略，而是直接向其邻节点广播该数据，邻节点重复广播数据的过程，直到该数据失效或者到达目的节点才会停止。泛洪算法简单易实现，无需存储维护路由信息，能耗小。但该算法很容易产生“内爆”（节点收到大量重复数据）、“重叠”（节点依次收到同一块区域内不同节点发送的完全相同的数据）和“资源利用盲目”（节点忽略自身资源，在所有情况下盲目发送数据）。

#### (2) 闲聊 (Gossiping) 协议

基于泛洪协议，闲聊协议<sup>[24]</sup>作了相对的改进从节约能量的角度考虑，闲聊协议在转发数据时使用随机的形式，而不是广播的形式。传感器节点随机的选择一个邻节点发送数据，有效规避了“内爆”的发生，但也会相应的增加网络延时。同时，泛洪协议的另两个缺点也没有得到解决。

#### (3) SPIN 协议

SPIN 协议<sup>[25]</sup>是以数据为中心自适应路由通信协议。针对泛洪协议和闲聊协议的缺陷，SPIN 协议重点考虑通过节点间的相互协议来保证传输数据的有效性，以此解决扩散法的缺陷。当节点间需传递数据时，彼此间会发送元数据来协商，元数据相较于采集数据很小，故发送元数据所需的能量很小。各个节点需监测各自可用的能量，在发送或收集数据时，若该节点的能量低于临界点，则会停止一些任务，以延长工作周期。

SPIN 协议有 ADV、REQ、DATA 三种数据包。节点通过 ADV 广播将有数据进行传输，通过 REQ 申请接收数据，通过 DATA 进行数据封存。ADV 用来广播新数据，当某节点共享

数据时, 可通过 ADV 数据包进行广播, 该数据包包含元数据。节点会发送 REQ 数据包用于申请接受 DATA 数据包。DATA 为传感器节点采集的数据形成的数据包, 元数据头亦包含其中。

SPIN 协议相对简单, 清楚邻节点的信息即可, 可有效解决泛洪协议和闲聊协议的缺陷。但其可靠性无法保证, 当某节点的全部邻节点都对数据无需求的时候, 数据的转发陷入停滞, 从而目的节点无法接收相关数据。

#### (4) 定向扩散 (Directed Diffusion) 协议

定向扩散协议<sup>[26-27]</sup>是以数据查询驱动为中心的平面路由协议。不同于已有的路由算法, 该协议引入“梯度”的概念。整个路由过程分为三部分: 兴趣的传播、梯度的建立和数据传输。

兴趣的传播: 基站向全部传感器节点传递“兴趣”, 传感器节点检测该“兴趣”是否存在于缓存中, 若无, 则进行保存。

梯度的建立: 同兴趣的传播同步建立, 在网络中的全部节点收到“兴趣”以后, 该“兴趣”梯度即可顺势建立。

数据传输: 源节点完成“兴趣”目标数据的收集后, 会对收集的数据进行筛选融合, 再将处理过的数据沿梯度传递至基站。

定向扩散协议在数据传输的过程中完成了数据融合, 减少了网络通信流量, 相应的也节约了节点能量。而多路径的使用, 使网络有较好的健壮性。但梯度的建立会消耗很大的能量, 不可用于多源节点网络。且节点数目越多, 开销和延时也越大。

#### (5) SMR 协议

SMR 协议<sup>[28-29]</sup>作为按需多路径路由协议, 重点构建、使用和维护最大数量的多条路由, 且路由间不可相交。数据通过各条路由进行传输。为合理而高效的使用能源, 节省开销, 保障网络通畅, 降低网络延迟, 数据由各路由分散传播。

SMR 的路由发现机制类似于 DSR 协议, 由路由发现和路由维护组成。源节点在整个网络中广播 RREQ 信息; 中间节点与 DSR 协议不同, 不可简单而随意地丢弃重复的 RREQ, 而是要对 RREQ 进行检查。若新的 RREQ 来自于不同的节点, 而且该 RREQ 内的跳数不大于之前节点接收到的 RREQ 的跳数, 那么就接收该 RREQ, 并保存到路由表内。同时, 该节点需在 RREQ 添加本身地址后再重新广播。反之, 则直接丢弃 RREQ。中间节点禁止反馈源节点任何路径。在最终选定路由的时候, 目的节点会先从最先收到的 RREQ 中挑选出一条路径, 再从此后设定的时间段内收到的 RREQ 中挑选出一条与之前所挑选路径相差最大的路径。

在检测到链路发生断裂时,通过替换路径再次传输数据,并且向源节点传输 RREP,删除该链路覆盖的路径。而对于源节点而言,在获取 RREP 之后,有两种路由策略供选择。当收到路由断裂的信息就重新进行路由发现,这种方式便于得到最新的路由信息;若要节省路由开销,则需在收到两条或多条路径均断裂的消息后,才能重新进行路由发现。在实际应用中,后一种路由策略的性能比前一种的性能好。图 2.4 为 SMR 协议建立路由的示意图。

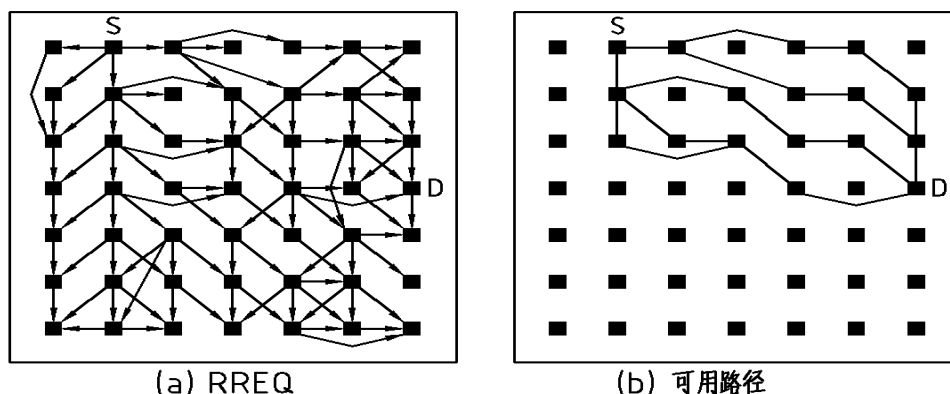


图 2.4 SMR 路由发现及可用多路径示意

#### (6) TBF 协议

TBF 协议<sup>[30]</sup>是一个基于源站和位置的路由协议。与 GPSR<sup>[31]</sup>协议有所不同, TBF 协议的传播路径并非最短路径。在数据包头中,该协议规划一条连续的

传输轨道。介于之间的节点在轨道参数的基础上,利用贪心算法计算出离轨道最近的邻节点作为下一跳目标。协议通过改变轨道参数,可轻松完成多路径以及特定区域的传播和广播。由于协议是基于源站的协议,所以节点省去了用于储存路由信息的缓存。数据包头不属于节点信息,其规模的大小与网络的大小无关,对网络变化有一定的适应能力。其缺点为节点用于计算的开销随网络的变大而增加。

#### (7) SAR 协议

SAR 协议<sup>[32]</sup>是首个在无线传感器网络中有 QoS 保障的路由协议。作为一种多路径路由,该协议在表驱动的基础上提高了能量使用的合理性,加强了网络的鲁棒性。该协议中,以所有属于汇聚节点有效传输范围内的邻节点为根生成树。树结构的创建离不开对能源、延时、包的优先级等参数的考虑。一个节点有概率属于多颗树,这就可以通过不同参数进行多路径的搭建。节点会参考能源、延时等参数选择路径进行数据传输。当数据传输发生错误时,该协议会主动重建路径,并强制使各个传感器节点路由表一致,完成错误纠正,实现问题解决。该协议合理有效地使用能量,延长了节点生存周期,但它并不适合规模大或者拓扑变化多的网络。

### (8) TORA 协议

TORA 协议有三个部分：路由发现、路由维护、路由删除。各个节点以各自的区域半径划分区域，区域半径为区域内各节点的最大跳数。每个节点只会随区域拓扑结构变化而变化，与整体网络无关。若跳数大于 1，各个节点的区域会大量的相交，数据通过这样的相交区域进行传输，同时设定跳数上限。当存在不同的路径完成同一个数据传输任务时，就形成了多路径路由。因对跳数设定了上限，则当超过跳数上限时，数据就会丢失，避免了数据冗余。而区域半径相对整个网络是较小的，所以区域内的拓扑开销也是有限的，相对较小的。

### (9) MSR 协议

MSR 协议<sup>[33]</sup>是对 DSR<sup>[34]</sup>协议的扩展。该协议通过节点间的反馈形成多路径，且以路径探测的方式来有效提高网络质量，减少网络堵塞和延迟。路由发现的步骤中，MSR 寻找并建立多条路径，同时将这些路径储存在高速路由缓冲中。一个序号仅对应一条路径，因此对于多路径的选择是非常简单便捷的。若某中间节点可找到路径连接目的节点，该节点在局部路径能量分布的基础上对传输数据的局部路径进行分组重排，构成分层的多路径。这样可以在负载不变的基础上提高资源的使用率。所有路径不能出现路由环路，因此在分组重排后需做仔细的检查。发现路径断开等问题时，需在断开的节点进行修复。若该节点负载相对较大，应选取另外的路径。在 MSR 中，独立性对于路由来说非常重要。独立性的强弱对路径间的影响有直接的关联。为满足高独立性的要求，需尽量选择相关性低的路径，该过程会增加额外的开销。

## 2.4 本章小结

本章主要是论述了无线传感器网络的体系结构、协议栈，并且介绍了传感网路由协议的分类，重点对相关的典型多路径路由协议进行了分析与比较，为后续路由算法的改进设计奠定了技术基础。

## 第三章 蚁群算法概述

### 3.1 蚁群算法简介

蚁群算法<sup>[35-36]</sup>是由意大利学者M. Dorigo, V. Maniezzo和A. Colomi等人于1992年提出来的,当初是针对商旅(TSP)问题设立的,随后其应用逐步地扩散,算法也随之逐步地完善起来,发展为各种各样的ACO算法。蚁群算法是一种基于模仿自然界中蚂蚁觅食的行为提出的仿生算法。生物学家通过观察和研究得出:蚂蚁总是能找出一条从蚁穴通往食物地点的最短路径<sup>[37-38]</sup>。蚂蚁通过散发一种特殊的化学物质来寻找食物,这种化学物质被称之为信息素。蚂蚁释放的信息被保留在所经过的路径上并且还会以一定的速率蒸发掉,后来的蚂蚁会根据路径上残留下来的信息素判断行进方向。蚂蚁沿着信息素浓度高的方向前进,并且它也会在该路径上留下新的信息素,使得该路径的信息素得以加强。这个过程不断的循环下去,经过同一条路径的蚂蚁越多,该路径的信息素浓度就越高,由于蚂蚁选择路径的概率与信息素浓度成正比,所以选取该路径的可能性也越高,这条路径上聚集的蚂蚁也就越多,从而形成一种正反馈机制,通过这种方式,蚂蚁就能找出一条从蚁穴通往食物地点的最短路径。

#### 3.1.1 蚁群算法原理

蚁群算法是用于探测空间中含有较高信息素浓度的路径,信息素随路径上不断增多的蚂蚁而逐渐的增强,搜索渐渐地开始变得有规律,逐步接近最优路径。

假设现在有一群蚂蚁向四面八方寻找食物,蚂蚁会在寻找食物的途中释放出一种激素,当某只蚂蚁找到食物后,就会沿原路返回,并在返回的途中也会释放出一种激素,但这种激素也会因为向四面八方散发导致浓度降低,现在假设有两只蚂蚁从不同的方向去寻找食物,如图 3.1 所示。

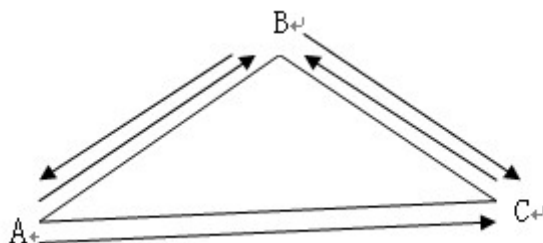


图 3.1 蚂蚁觅食示意

两只蚂蚁分别从 A 点出发，食物在 B 点，一只蚂蚁（甲）沿 A→B 路径，另一只蚂蚁（乙）沿 A→C→B 路径，当甲蚂蚁到达 B 点并找到食物按原路返回到 A 点时，蚂蚁乙才刚刚找到食物，由此可见，AB 路径上已经有了两次往返的激素，而路径 ACB 上只有一次去的激素，可见 AB 上的激素比 AC 上的激素来的高，后面的蚂蚁也会根据激素的浓度而去选择 AB 路径觅食（即探求到的最短路径），图 3.2 更好的说明了这个原理。

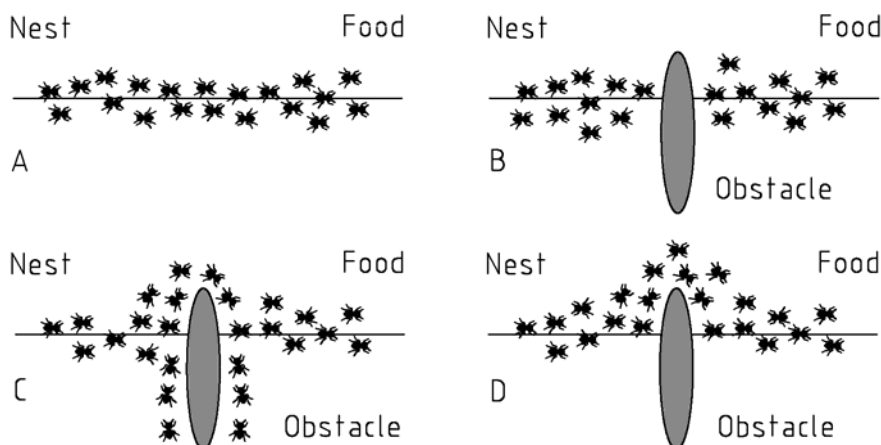


图 3.2 蚁群路径示意

由图 3.2 可见，蚂蚁不但能找到蚁穴与食物源之间的最短路径，而且当蚁穴与食物源之间出现障碍物的时候，它也能够很快的适应变化，从而快速的重新找到蚁穴与食物源之间的最短路径。

蚁群算法的本质在于：第一是选择机制，某条路径上拥有的信息素浓度越高，对蚂蚁的吸引力就越大，该路径被蚂蚁选取的几率就越高；第二是更新机制，路径上的信息素会因为蚂蚁走过次数的增长而增长，也会因为时间的延长，信息素的挥发而减少；第三是协同机制，蚂蚁之间是通过路径上的信息素含量进行相互的通信与工作的，由于这个原因，使得蚂蚁具有获得最优解的能力。

### 3.1.2 蚁群算法的传统模型

蚁群算法最开始是用于解决 TSP 问题的。因此我们可以通过 TSP 问题来说明蚁群算法最基本的模型，阐述蚁群算法是如何实现的。

现在假设有  $N$  只（人工）蚂蚁以随机的方式分配到  $M$  个城市中，在路径创建当中，蚂蚁是按照一定的规则来挑选下一个不曾到达过的城市，并且在经过一个循环（即访问过所有  $M$  个城市）或者走完一步后（从一个城市访问另一个城市），就要对路径上的信息素浓度进行更新，因此蚁群算法主要包含两个重点即路径创建和信息素更新。

在路径创建中，蚂蚁会按照一个概率来选择下一步要访问哪个城市，而这个概率是由两个要素决定的，一个是  $t$  时刻城市  $i$  和  $j$  之间路径上的剩余信息素浓度  $\tau_{i,j}(t)$ ，该值由算法提供，另一个是城市  $i$  访问城市  $j$  的启发式信息  $\eta_{i,j}$ ，该值由问题本身提供。由这两个要素得出在  $t$  时刻，城市  $i$  访问城市  $j$  的概率公式为：

$$p_{i,j}^k = \begin{cases} \frac{\tau_{i,j}^\alpha(t) \eta_{i,j}^\beta(t)}{\sum_{s \in M_i^k} \tau_{i,s}^\alpha(t) \eta_{i,s}^\beta(t)} & j \in \text{allowed}_k \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

其中： $p_{i,j}^k$  表示  $t$  时刻蚂蚁  $k$  从城市  $i$  选择城市  $j$  的概率； $\tau_{i,j}(t)$  表示  $t$  时刻该节点在路径  $(i,j)$  上的信息素浓度； $M_i^k$  表示在城市  $i$  上的蚂蚁  $k$  还未经过的相邻城市的集合； $\eta_{i,j}=1/d_{i,j}$  是先验知识，它代表一个预先给定的局部启发式信息，其中  $d_{i,j}$  为城市  $i$  与城市  $j$  之间的距离； $\alpha, \beta$  是两个参数，前者代表信息素浓度的影响力，后者代表启发式信息的影响力。

由公式 3.1 可以看出，当  $\alpha=0$  时，最靠近城市  $i$  的城市被选中的概率最大；当  $\beta=0$  时，只有信息素浓度起到了作用，这会给算法带来坏处，使算法产生停滞的现象。

随着时间的变化，各个路径上的信息素会有一定程度的挥发，挥发公式如下所示：

$$\tau_{ij} = (1 - \rho) \times \tau_{ij}, \quad \forall (i,j) \in L \quad (3.2)$$

其中  $\rho$  代表信息素的挥发率， $0 < \rho \leq 1$ 。引进  $\rho$  的作用是防止信息素不停的增大，同时也可以替代算法中之前不好的路径。

在蚂蚁完成一个循环访问和挥发步骤之后，必须处理路径上的剩余信息素，将最新的蚂蚁访问路径的信息加入原信息，由此得出：

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^n \Delta \tau_{ij}^k, \quad \forall (i,j) \in L \quad (3.3)$$

$\Delta \tau_{ij}^k$  表示第  $k$  只蚂蚁在它所经过的路径上所释放的信息素，如下所示：

$$\Delta \tau_{ij}^k = \begin{cases} 1/C^k, & \text{如果边 } (i, j) \text{ 在路径 } T^k \text{ 上} \\ 0, & \text{否则} \end{cases} \quad (3.4)$$

其中  $C^k$  表示路径  $T^k$  上所有边的长度之和。



由公式 (3.4) 可以看出, 一条边所在的路径越短, 并且被越多的蚂蚁选择, 那么这条边获得的信息素就越多, 那以后被选中的概率就越大。

通过以上的介绍, 蚁群算法的基本处理过程可归纳为三个步骤:

第一步: 初始化  $A(t)$ , 即对蚁群进行初始化;

第二步: 更新信息素, 即对蚂蚁经过的路径按照一定的公式进行信息素的增加或信息素的挥发减少;

第三步: 蚂蚁移动, 即根据蚂蚁留下的信息素和自己的判断计算出每个节点的下一个节点的概率, 选择概率最高的节点进行移动。

## 3.2 蚁群算法的应用情况

现在很多优化问题或者求解最优解的问题都会运用蚁群算法来解决, 最初蚁群算法大多被用于解决 NP 问题, 其中最典型的是 TSP 应用, 现在已经扩展到其他领域, 比如通信网络, 工业上的调度等。

### (1) NP 问题

我们经常比较已知算法和启发式算法的性能, 从而可以证明新的启发式算法是有效的。一开始蚁群算法在解决 TSP 的问题上获得了很大的成功, 迄今为止, 我们已经对上百种 NP 问题进行了大量的实验, 这些 NP 问题大概可以划分为以下几种: 集合问题、路径问题、调度问题、指派问题等。并且蚁群算法也成功应用在机器人控制及生物系统建设领域中。局部搜索策略可以用于改进蚂蚁寻找到的解, 而且由局部搜索策略一下子找到合适的初始解是不可能的, 这时蚁群算法的适应性解的迭代过程能够很好地解决这类问题。

### (2) 工业问题

蚁群算法在工业领域受到了很多的关注, 国外已经有公司将它应用于解决现实生活中不同的调度问题, 同时已经有许多工业领域的软件工具是以蚁群算法为基础的。这些软件工具解决了瑞士某个大型企业的车辆燃油的管理分配问题, 还解决了某些仓储超市的货车调度与路由问题。除此之外, 很多外国的企业还将蚁群算法应用到更广泛的实际领域当中, 如有规划平衡网络路由、挖掘数据、预测及分析金融现金流、构造水利设施、设计大型制造商的生产线等。实际取得成功的方案有 BiosGroup 为法国某公司成功设计车辆路线选择方案、Gange 和 Gravel Price 应用蚁群算法于铝制品制造中的工业调度等。在国内, 蚁群优化算法在工业领域中还需要进行更多的研究和探索。

### (3) 通信问题

由于蚁群算法自身的特点，使得它于通信网络中网络优化问题的特点匹配度很高。因此，蚁群算法很适合用于解决网络优化问题。英国电信公司和美国惠普公司在这个方面都展开了研究。被最广泛应用的例子便是 AntNet 算法，其实验是在各种网络，各种负载的情况下进行的，实验结果表明 AntNet 算法比其他同类算法具有更好的适用性和良好的鲁棒性。

### 3.3 典型的蚁群算法

二十世纪 90 年代，意大利学者 Marco Dorigo 在其博士论文中提出了蚁群优化，后来又与他人一起设计出了第一个蚁群算法——蚂蚁系统（Ant System），这个算法是通过观测现实中蚂蚁的觅食行为而提出来的，它是一创新性高的启发式算法。蚁群优化是一种仿生物学的算法，它对生物的进化进行了模拟，在很多领域都有很大的应用价值。蚁群优化是用于求解最优解问题，并且它还具有贪婪启发式搜索、分布式计算以及正反馈的特点，基于上述特点，蚁群优化已成为许多研究人员的研究热点。

迄今为止，人们对于蚁群优化提出了许多的改进，例如有精英策略蚂蚁系统、排序蚂蚁系统、最大最小蚂蚁系统和蚁群系统等<sup>[39-41]</sup>。

#### （1）精英策略蚂蚁系统

精英策略蚂蚁系统（Ant System with Elitist Strategy, ASWES）是首个由蚂蚁系统改进的算法。该算法之所以会有精英策略来概括，主要是因为它与遗传算法中的精英策略有相类的地方。遗传算法使用选择（selection），重组（recombination）和突变（mutation）这三个遗传算子，某一代中最好的个体（即某个循环中的最优解）有很大的可能不存在于下一代中。基于这种情况，那个最好个体的信息会因此而无法保存下来。所以遗传算法里就出现了精英策略，目的就是能够保存这个最好的个体，使得能够生存下来。同样，在蚂蚁系统中，为了使当前的最优解能够在后续循环中对蚂蚁产生更多的影响，所以在每次循环结束后，该循环的最优解需要增加额外的信息素量。该解叫做全局最优解（global-best solution），找出该解的蚂蚁就是精英蚂蚁。该算法的更新信息素的公式为：

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij} + \sigma\Delta T_{ij}^{bs}, \quad \forall (i,j) \in L \quad (3.5)$$

其中  $\Delta\tau_{ij}$  的定义公式为：

$$\Delta\tau_{ij} = \sum_{k=1}^n \Delta\tau_{ij}^k \quad (3.6)$$

$$\Delta \tau_{ij}^k = \begin{cases} 1/L_k, & \text{若蚂蚁} k \text{在本次循环中经过} ij \\ 0, & \text{否则} \end{cases} \quad (3.7)$$

$\Delta T_{ij}^{bs}$  的定义公式为：

$$\Delta T_{ij}^{bs} = \begin{cases} 1/C^{bs}, & \text{若边} (i, j) \text{是找出的最优解的一部分} \\ 0, & \text{否则} \end{cases} \quad (3.8)$$

$\Delta T_{ij}^{bs}$  代表精英蚂蚁所引起的在  $(i, j)$  路径上的信息素含量的增加， $\sigma$  代表精英蚂蚁的数量， $C^{bs}$  是最优解路径的总长度。文献[26]的实验结果显示：精英蚂蚁的数量需处于一个合适的范围，当精英蚂蚁的数量在这个范围内时，精英蚂蚁的数量和算法获取较好解的能力成正比，与获取较好解的时间成反比；但是如果精英蚂蚁的数量超出了这个范围，精英蚂蚁数量的增加会提高算法的性能。

## (2) 排序蚂蚁系统

精英策略蚂蚁系统和最初的蚂蚁系统一样，存在着一个相同的缺点，即在不断的进化中，由于解的总质量得到了提高，所以解和解之间的差异就变小了，从而对最优路径的选择也会变得困难，这样搜索无法集中到迄今为止所得到的最优解的附近，从而导致无法完全找到最优路径。许多蚂蚁会因为各个路径长度都差不多而沿着局部最优前进，这种情况会削弱短路径的作用。

基于上述的原因，二十世纪 90 年代末，Bullnheimer、Strauss 和 Hartl 等人提出了对蚂蚁系统另一种的改进方法——排序蚂蚁系统（Rank-Based Version of Ant System, RBVAS）。这套排序蚂蚁系统是引进了遗传算法中的排序机制。遗传算法通过排序的选择机制解决了如何选择的问题。该机制是这样的，首先通过蚂蚁的适应度对蚂蚁分类，然后将每个蚂蚁进行排序，适应度高则代表个体好，个体在队伍的位置越靠前，被选中的机会就越大。这种机制就被扩展运用到了蚂蚁系统当中，形成一个新的蚁群算法，即排序蚂蚁系统。

该算法在信息素更新前，先会对每只蚂蚁形成的路径按照其长度进行排序（ $L_1 \leq L_2 \leq L_3 \leq \dots \leq L_n$ ），路径越长排名越靠后，信息素修改时，蚂蚁释放的信息素的量要乘以该蚂蚁相对应的排名，但是只有排在  $w$  名之前（ $1 \leq r < w$ ）和精英蚂蚁才可以在路径上留下信息素。只有当前已获得的最优路径上的蚂蚁才能给予最强劲的反馈，它所释放的信息素的量要与权重  $w$  相乘，而排名在第  $r$  位的蚂蚁，它释放的信息素的量只是乘以  $w-r(>=0)$ ，因此得到以下公式：

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta \tau_{ij} + w \Delta \tau_{ij}^{bs} \quad (3.9)$$

其中  $\Delta\tau_{ij}$  的定义为：

$$\Delta\tau_{ij} = \sum_{r=1}^{w-1} (w-r)\Delta\tau_{ij}^r \quad (3.10)$$

该公式表示  $w-1$  只蚂蚁依据排名的名次在城市  $(i, j)$  信息素的更新量：

$$\Delta\tau_{ij}^r = \begin{cases} \frac{Q}{L^r}, & \text{排名为 } r \text{ 的蚂蚁经过路径 } (i, j) \\ 0, & \text{否则} \end{cases} \quad (3.11)$$

$$\Delta\tau_{ij}^{bs} = \begin{cases} \frac{Q}{C^{bs}}, & \text{若边 } (i, j) \text{ 是找出的最优解的一部分} \\ 0, & \text{否则} \end{cases} \quad (3.12)$$

其中， $r$  是蚂蚁的排名； $(w-r)\Delta\tau_{ij}^r$  是排名为  $r$  的蚂蚁所引起在路径  $(i, j)$  上信息素量的增加； $L^r$  表示排名为  $r$  的蚂蚁所经过的路径长度； $w\Delta\tau_{ij}^{bs}$  表示由精英蚂蚁所造成的在路径  $(i, j)$  上信息素的增加； $C^{bs}$  表示最优解的总长度。

实验结果显示， $AS_{rank}$  的性能对于 EAS 稍有提高，但对于 AS 而又有明显的提高。

### (3) 最大最小蚂蚁系统

最大最小蚂蚁系统 (MAX-MIN Ant System, MMAS) 是由德国学者 T. Stuetzle 和 H. Hoos 在二十世纪 90 年代后期提出的，该算法具有贪婪式寻优的特点，是迄今为止解决 TSP、QAP 这类问题最有效的蚁群算法之一，它的改进直接来自于蚂蚁系统 (AS)，改进最主要的任务就是避免算法过早的出现停滞。

通过一些实验发现，由于信息素过度集中到几条相对较优的路径上，导致其他路径上长期没有蚂蚁经过，没有新的信息量的增加，而只是一味的信息素的蒸发，最终导致这些路径上的信息素的浓度渐渐地接近 0 值，这样一来，这些路径就成为了蚂蚁必定不走的路径，这种情况会失去获取最佳路径的可能，算法就会过早的出现停滞的现象，基于这种原因，最大最小蚂蚁系统 (MMAS) 对最初的蚂蚁系统 (AS) 进行了 4 个方面的改进：

1) MMAS 算法中引入了信息素浓度的最大值和最小值，每条路径上的信息素浓度  $\tau_{ij}(t)$  必须满足  $0 < \tau_{\min} \leq \tau_{ij}(t) \leq \tau_{\max}$ ，如果  $\tau_{\max} < \tau_{ij}(t)$ ，则  $\tau_{ij}(t) = \tau_{\max}$ ，同样地， $\tau_{\min} > \tau_{ij}(t)$ ，则  $\tau_{ij}(t) = \tau_{\min}$ ，这样就可以避免信息素的无限叠加或者信息素挥发为 0 的情况，而蚂蚁系统没有对信息素量

加一限制，会出现某些路径上的信息素量远远高于其他路径，这样蚂蚁就会一直沿着某一路径进行，但这个解只是局部最优解，并非全局最优解，而降低了寻找全局最优解的可能。

2) 每次循环后，MMAS 中规定只有发现当前循环中最优解的蚂蚁或者发现已知最优解的精英蚂蚁才可以释放信息素，而在 AS 中，是所有蚂蚁都可以进行信息素的更新。

3) 初始时，各条路径上的信息素值被定义为  $\tau_{\max}$ ，这样定义的目的是一开始就扩大搜索的范围，是算法在最初阶段能够探索更多的路径。同时，也会定义一个挥发因子  $\rho$  ( $0 < \rho < 1$ )，将信息素的初始值定义为  $\tau_{\max}$ ，能够减小可行解之间的信息素值的差别，可以使算法能够对空间展开充分的搜索。

4) 当算法停滞或者长期没有找到最优解的时候，可以重新初始化路径上的信息素值。

MMAS 更新信息素值的公式为：

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best}, \quad 0 < \rho < 1 \quad (3.13)$$

其中  $\Delta\tau_{ij}^{best} = 1/C^{bs}$ ， $C^{bs}$  表示当前已知最优解路径的长度；或者  $\Delta\tau_{ij}^{best} = 1/C^{bi}$ ， $C^{bi}$  表示此次循环找到的最优解的路径长度。

#### (4) 蚁群系统

由于蚂蚁系统 (AS) 算法还存在不足，它的创始人 Marco Dorigo 和 Gambardella 又在 1996 年提出了一个新的改进算法——蚁群系统算法 (Ant Colony System, ACS)，这次改进主要是针对于信息素和构造解方面的改进，它包括三个方面：构造解的过程、全局信息素更新策略和局部信息素更新策略。

##### 1) 构造解的过程

由于在蚂蚁搜索初期，各个路径上的信息素的差别不是很大，这样就导致算法无法尽快收敛，因此信息素对蚂蚁寻找最优路径的指导性不强，就会出现蚂蚁盲目地搜索路径，因此蚁群系统 (ACS) 算法使用了一个新的选择规则来构造可行解，这个新的规则称为伪随机比例选择规则 (Pseudo random proportional action choice rule)。处于城市  $i$  的蚂蚁  $k$  会以概率  $q_0$  向城市  $u$  移动， $u$  代表为使  $[\tau_{iu}(t)]^\alpha [\eta_{iu}]^\beta$  获得最大值的城市，选择该方式表明蚂蚁将会以概率  $q_0$  将具有最大可能的城市选入可行性解；除了这种选择方式，蚂蚁还会以另一种概率选择下一个城市  $j$ ，其具体的状态转移公式为：

$$j = \begin{cases} \arg \max_{u \in allowed} \tau_{iu}^\alpha(t) \eta_{iu}^\beta(t), & \text{如果 } q \leq q_0 \\ S, & \text{否则} \end{cases} \quad (3.15)$$

其中  $q_0$  为常数且  $0 < q_0 < 1$ ,  $q$  为  $(0, 1)$  之间的随机数,  $\tau_{iu}(t)$  为  $t$  时刻城市  $i$  到城市  $u$  路径上的信息素值,  $\eta_{iu}$  为城市  $i$  和城市  $u$  之间的启发式因子,  $\alpha$  为信息素浓度的相对强弱,  $\beta$  为启发式因子的相对强弱。在选择下一个要移动的城市之前, 首先会产生一个随机数  $q$ , 如果  $q \leq q_0$ , 就会根据公式  $\arg \max_{u \in allowed} \tau_{iu}^\alpha(t) \eta_{iu}^\beta(t)$  来选择下一个要移动的城市; 如果  $q > q_0$ , 则就会按照以下公式进行选择:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \in M_i^k} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)} & j \in allowed_k \\ 0 & otherwise \end{cases} \quad (3.16)$$

通过这种选择方式, 增加了算法选择的多样性, 也有效地避免算法过早地出现局部停滞现象。

## 2) 全局信息素更新规则

为了加快蚂蚁的搜索速率, 采用了一种新的更新全局信息素的方法。在蚂蚁系统 (AS) 算法中, 所有蚂蚁在每次循环后都可以释放信息素, 但是蚁群系统 (ACS) 算法只有精英蚂蚁 (即每次循环中找到最优解的蚂蚁) 才可以释放信息素, 增加其外部信息素的值, 而且一次循环后, 不处于最优路径上的边也无需进行信息素挥发操作。更新公式如下:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}^{bs}, \quad \forall (i,j) \in T^{bs} \quad (3.17)$$

其中  $\rho$  为信息素挥发系数, 且  $0 < \rho < 1$ ;  $\Delta\tau_{ij}^{bs} = 1/C^{bs}$ 。

这样经过几次循环后, 属于最优路径上的边的信息素值就会明显超过其他边, 也就大大提高了搜索的目的性。

## 3) 局部信息素更新规则

蚂蚁系统 (AS) 算法很容易在局部优化中就陷入了搜索停滞, 因此蚁群系统 (ACS) 算法在这方面也进行了改进。ACS 算法规定在蚂蚁每搜索一条边后, 就会进行信息素的更新, 其调用规则的公式如下:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_0 \quad (3.18)$$

其中  $\rho$  表示信息素挥发系数,  $\Delta\tau_0$  表示信息素初始值。

这样蚂蚁就有更多的机会选择其他路径, 从而能更有效地解决由于过度集中而导致局部优化停滞的问题。

### 3.4 在无线传感器网络路由算法中引入蚁群算法的优势

蚁群优化算法是群体智能算法的一个方面，简单地可以理解为各个简单独立的个体通过相互的通信和对外界环境的适应，从而使整个群体的行为和模式能够慢慢地到达一致。下面将会结合无线传感器网络的特征<sup>[36]</sup>来说明蚁群算法运用于无线传感器网络路由算法<sup>[42]</sup>的优势：

#### （1）节点能量有限

无线传感器网络的节点一般是由电池或电源供电，而且无线传感器网络经常用于环境比较恶劣的情况下，比如沙漠、偏僻的山区等等，因此无法及时给节点供电，所以节点的能量是很有限的，而且，一般传感器节点的存储能力小和处理性低，这就要求设计简单的路由算法，以此减少存储量和计算量，降低能耗。蚁群算法利用蚂蚁按照简单的规则就能搜索到一条最优化的路径，将这种思想应用到无线传感器网络路由协议的设计中，可以简易实现算法，并且能有效地节约有限的网络资源。

#### （2）动态网络拓扑

无线传感器网络的节点可能因为自身能量的耗尽或者外界环境因素的影响而失效，因为某些情况需要临时增加新的节点，网络中的用户、节点和信息的位置产生了移动，这些情况都会使网络的拓扑结构随时随地产生变化。WSN 以前的路由协议因为这些原因而无法取得良好的效果。蚁群算法是群体智能的一个部分，每只蚂蚁都是独立的工作并且通过信息素相互之间进行通信，当群体中的某些蚂蚁因为某些原因不能工作或者路径中出现了障碍的时候，其他蚂蚁仍然能够按照之前约定好的规则继续运转，以便最及时地发现另一条最优路径。因此蚁群算法堆网络拓扑结构的动态变化有较好的适应能力。

#### （3）自组织网络

因为无线传感器网络通常是用在人类还未开拓的地方，所以通常使用飞机散播的方法来设置节点，导致节点的位置是无法提前预知的。在 WSN 中，节点的地位都是一样的，所以不存在网络中心。节点可以使用预先编写的程序来搭建网络的拓扑结构，这就是节点的自组织功能，节点通过对自身参数的预设，形成一个无人监管的网络。蚁群算法是利用正反馈的原则来找到最优化路径，该机制说明了最短路径的作用，这正好与满足节点的自组织特征，而且蚂蚁还可以将服务质量、能耗、延时作为搜索路径的一个指标，这样就可以找到一条高效的最优化路径。

#### （4）局部路由

因为传感器节点的存储力和计算力都有限，各个节点存储的路由信息只是邻节点的信息而不是所有的，所以普通的路由协议得不到很好的效果。如果是基于蚁群算法的路由协议中，节点只需要知道附近节点的路由信息就可以进行数据传输，这就可以解决传感器节点存在的缺点。

#### (5) 可扩展性

无线传感器网络对扩展性要求较高，由于蚁群算法不能完成集中控制，只能以各节点的协同工作来完成任务，使得扩展性很好，有利于分布式计算。

### 3.5 本章小结

本章主要论述了蚁群算法的基本原理，结合商旅问题讨论了基本蚁群算法的实现，并且通过基于典型 AS 算法改进的蚁群算法的分析，考虑无线传感器网络自身的特点，阐述了蚁群优化算法应用于传感网络的路由设计具有很强的优势。



## 第四章 算法的改进

### 4.1 基本蚁群算法的不足

最初的蚁群优化算法——蚂蚁系统(AS)是根据路径上的信息素浓度来选择最优路径的,但是它毕竟只是一种仿照生物觅食的算法,因此在解决实际问题时肯定存在不足与缺陷,它无法拥有和现实生活中的真实蚂蚁一样的能力和办事效率,它无法具有如遗传算法等那样完备的理论基础和实验基础,因此它还存在很多方面需要改进和完善。

(1) 算法在开始执行之时,算法求出问题最优解的速度非常缓慢,这是由于在初始阶段,网络中的各个路径上所表现出来的信息素浓度水平基本差不多,对蚂蚁搜索的指导性不强,这样就会存在蚂蚁盲目搜索现象,在经过一段时间后,信息素的作用才会慢慢体现出来。

(2) 根据蚁群算法的原理可以知道,蚁群算法其实是一种正反馈算法,虽然它具有较快的收敛速度,这一特性也很容易造成它陷入局部最优的尴尬境地。

例如,两个点之间的一条边,沿着这条边所花销的费用是最少的,因此,这条边一开始将获得最多的信息素(大量蚂蚁会沿着费用最少的路径行进),而具有较高信息素的路径又会吸引更多的蚂蚁聚集过来,这样其他路径被访问的概率就大大减少了,但是有一种情况是,最优路径并没有包含这条边。

蚁群算法中各个路径的信息素的初始值是一样的,蚁群算法有一点雷同于贪心算法,蚂蚁经过第一条路径,散发出信息素,但是该路径不一定就是最优路径。当蚂蚁数目较少或者路径选择更多时,蚂蚁找到的第一条可行路径并不一定有作为找寻最优解的指导因素。在第一次循环中,蚂蚁遗留的信息素可能会使找寻最优路径产生错误的指导,从而使后面的蚂蚁无法找到全局最优解。

(3) 基于蚁群算法的路由协议是平面路由协议的一种,平面路由协议存在一个网络节点得不到管理的缺点,能量消耗得不到有效的改善与控制。

(4) 算法在一开始收敛速度较快,经过一定的循环之后,就有可能在停滞在某些局部最优解,并且会不断地加强这个局部最优解附近的信息素,蚂蚁要离开这条路径另觅新的路径会比较困难。就算能够离开,那网络延迟会增大,循环次数也会增多。当网络的拓扑结构产生变化时,蚂蚁就无法迅速的找到新的路径来传输数据,性能就会明显降低了。

(5) 蚁群算法是根据信息素进行路径选择, 这会导致该路径上聚集的蚂蚁数量相当之多, 使得该路径上传感器节点的能量快速的减少, 缩短了网络的生存时间, 而且也有可能产生网络拥塞。

根据上面的阐述, 基本的蚁群算法仍有不足, 但是蚁群算法的思想对于 WSN 路由设计又有很好的借鉴作用, 文献[43]和文献[44]分别给出了蚁群算法应用无线传感器网络的示例。因此, 我们可以采用一定的方法对它进行改造, 比如信息素更新规则的改变等, 从而得出一个有效的、改进的蚁群算法。

## 4.2 改进的蚁群算法

由于基本的蚁群算法思想, 会使得路径的最优解增大, 很难在一定的时间内找到最优路径, 而且算法会使解停滞在局部最优解中。因此本文对基本的蚁群算法进行了以下几方面的改进, 得到了一个新的改进蚁群算法——AIS (Ant Improvement System)。

(1) 首先设定了信息素浓度的范围, 每条路径上的信息素浓度  $\tau_{ij}(t)$  必须满足  $0 < \tau_{\min} \leq \tau_{ij}(t) \leq \tau_{\max}$ , 如果  $\tau_{\max} < \tau_{ij}(t)$ , 则  $\tau_{ij}(t) = \tau_{\max}$ , 同样地,  $\tau_{\min} > \tau_{ij}(t)$ , 则  $\tau_{ij}(t) = \tau_{\min}$ , 这样就可以避免信息素的无限叠加或者信息素挥发为 0 的情况, 而蚂蚁系统没有对信息素量加一限制, 会出现某些路径上的信息素量远远高于其他路径, 这样蚂蚁就会一直沿着某一路径进行, 但这个解只是局部最优解, 并非全局最优解, 从而减少了寻找全局最优解的可能。

(2) 在全局更新信息素浓度规则上, 蚂蚁遍历完一次循环之后, 只有找出最优路径或最差路径上的蚂蚁才可以释放信息素, 同时其它路径上的边也无需进行信息的挥发操作。(i, j) 如果是最差路径上的边, 它必须同时保证该路径不处于最优路径上, 信息素更新规则为:

$$\tau_{ij}(t+1) = \begin{cases} (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}^{bs}, & (i,j) \in T^{bs} \\ (1-\rho)\tau_{ij}(t) - \rho\Delta\tau_{ij}^{ws}, & (i,j) \in T^{ws} \text{ 且 } (i,j) \notin T^{bs} \end{cases} \quad (4.1)$$

其中  $\rho$  为信息素挥发系数,  $0 < \rho < 1$ ;  $\Delta\tau_{ij}^{bs} = Q/C^{bs}$ ,  $C^{bs}$  表示最优路径的长度;  $\Delta\tau_{ij}^{ws} = Q/C^{ws}$ ,  $C^{ws}$  表示最差路径的长度; 进行多次循环之后, 就能使得最优解得到增强, 最差路径得到减弱, 有效地拉开了最优解与最差解之间的距离, 使算法能够集中于最优解的附近找出更优解。

(3) 在局部更新信息素浓度上, 蚂蚁在经过一条边之后就进行该边信息素的更新, 公式如下:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_0 \quad (4.2)$$

其中  $\rho$  表示信息素挥发系数,  $\Delta\tau_0$  表示信息素初始值。这样, 当有蚂蚁经过  $(i, j)$  路径上时, 信息素就会挥发比一部分, 那剩下的蚂蚁选择该路径的可能性就会变小, 更大可能地选择新的路径, 使得算法不会收敛于局部最优解, 陷入停滞现象, 缩短网络生存周期。

### 4.3 AIS-SMR 多路径路由协议的实现

通过对典型传感器多路径路由协议的分析, 我们不能简单地判定这些协议哪种最好, 因为每种协议都有自己的优缺点, 都是为特定环境或应用而设计的。下表对各种协议做了比较和总结, 如表 4.1 所示。

表 4.1 多路径路由协议比较

路由协议	策略	能耗	数据融合	QoS	不相交多路径	并发多路径传输
Flooding	主动	高	无	无	无	无
Gossiping	按需	高	无	无	无	无
SPIN	按需	中	有	无	无	无
DD	按需	中	有	可能	无	无
SMR	按需	低	无	有	有	无
TBF	按需	中	无	无	无	可能
SAR	按需	低	无	有	无	有
TORA	按需	低	无	无	可能	可能
MSR	按需	中	无	无	可能	无

SMR 协议能够建立最大不相交的多条路径, 能有效地防止网络阻塞, 但是 SMR 协议未能有效地利用路由的稳定性来发现路径。虽然 SMR 能够发现多条路由, 但长期只使用一条路由来进行数据传输, 一旦该路由断裂, 就要重新启动路由发现机制, 不仅没有充分的利用资源, 生命期也不长, 而且 SMR 协议只考虑了跳数这一性能, 未能很好地综合反映该路由的质量, 于是本文将改进的 AIS 蚁群算法用于 SMR 协议, 给出了一种改进算法——AIS-SMR 算法, 利用 AIS 的来弥补上述缺点。

#### 4.3.1 协议设计目标

设计无线传感网络的路由协议时需要考虑一些相关的约束要求。本文考虑的主要协议设计目标有:

### （1）能量高效性

无线传感网络是由大量传感器节点构成的，而节点都是由电源或电池进行供电，能源很难得到及时补充，因此它们的能量都有限。而无线传感网络又会应用于条件比较艰苦的地方，如沙漠、外太空、森林等等，电能一耗尽是无法进行补充的，而且以目前的技术，节点的电量很难得到明显的提高。因此设计的路由策略不但要选取能耗小的传输路径，而且要能够均衡网络中所有节点的能耗，避免某些节点能量消耗过快以至于无效，从而实现延长整个传感网的生存时间，提高网络稳定性的目的。

### （2）可靠性

大多数无线传感网络多用于艰苦恶劣的环境中，会因为各种各样的原因令节点失效，比如节点电量耗尽、环境的变化、硬件设备的损坏。此时网络的服务质量就会降低，这时就需要具有一定容错能力的路由协议，能够确保在网络发生故障时能迅速恢复，使得部分节点失效时不会对整个网络的运作造成不利的影响，以此提高网络的可靠性。

### （3）收敛性

节点失效、节点加入和节点位置的改变都会改变无线传感网络的拓扑结构，导致路由构造、节点能量以及网络服务质量产生影响，需要路由协议迅速收敛到最优路径上，适应新的网络拓扑结构，从而削减通信开销，提高数据传输速率。

### （4）安全性

因为无线传感网络自身的特征，无线传感网络容易受到外界的攻击，并且任何网络都会对安全性提出要求，因此无线传感网络设计路由策略时也要考虑这一要素。

### （5）扩展性

无线传感网络规模一般都比较小，因此不能为每一个节点都分配一个独立的 ID，也不可能获得并维护全局信息，所以需要有一个依赖于局部信息的分布式算法。由于各种原因使得节点失效、节点新增、节点移动，导致网络拓扑变化，这就需要设计出具有可扩展性的路由机制，以匹配网络结构的变化。

### （6）简易性

无线传感网络与传统的无线网络不同，节点要扮演几重角色：感知数据、收集数据和路由器，但传感器节点的存储能力和运算能力有限，因此我们要为其设计特殊的路由协议。该路由协议要尽可能的减少开销，提供高效的路由选择。相当于要求路由协议的算法和报文要尽量简单高效，路由表项要尽可能的少。

### （7）实时性

观察者一般都要要求能够实时地监测传感网，因此该算法要考虑减少时延。

当然，除去上述的因素之外，还有其他相关指标，如鲁棒性，数据融合等。

### 4.3.2 协议改进思路

AIS-SMR 算法的主要策略是能够使数据传输时，不再长期使用一条路由来传输数据，而且能够选择代价最小且最稳定的路径，不仅达到节约网络资源而且能有效地延长网络的生存周期，AIS-SMR 算法能够动态更新路径信息，并且较好地处理了节点失效和节点新增的问题，该算法不仅能对网络环境的各种变化采取快速的解决办法，并且引入的开销并不大，增强了算法的适应能力和运行能力。

AIS-SMR 算法是一种链路不相关的多路径路由算法，为了尽可能多地找到到达目的节点多条链路不相关的路径，前向报文蚂蚁在到达目的节点之后，就会被抛弃，重新产生后向更新蚂蚁，这样就会找到多条链路不相关路径，并且在后向更新蚂蚁返回源节点的途中，只有最优路径上和最差路径上的边才能通过后向更新蚂蚁携带的信息进行信息素的更新，其他路径上的边无需任何操作，这样就能使得最优路径得以加强，最差路径得以减弱，增加下一次循环最优路径对蚂蚁的吸引力，增强算法在最优路径附近的搜索能力。最优路径建立的同时，也会有很多的备用路径被发现，备用路径的存在可以增强网络的抗毁性。同时在正向搜索路径时，蚂蚁每经过一条边就会更新该边的信息素值，防止蚂蚁一直选择同一条路径，从而使算法陷入局部停滞状态，这样可以使蚂蚁有更多的机会选择其他路径，而不会像 SMR 算法一样长期使用一条路径来传输数据，导致节点能量快速的消耗。该算法相对于 SMR 算法而已，路径选择时，不仅考虑了跳数，而且还加入了剩余能量和延时的考虑，使得数据能在一个更稳定的路径上进行传输。基于上述种种改进，网络的生存能力得到了显著的提高。

AIS-SMR 算法是在对 SMR 协议运用改进的蚁群算法策略进行改进后的协议，它们两者有以下的区别：

(1) SMR 协议是通过广播 RREQ 报文，回馈的是 RREP 报文，而 AIS-SMR 算法则是通过发送 FANT 报文，BANT 报文和 QANT 报文。

(2) 当路由表中没有源节点到目的节点的路由信息时，SMR 协议则是广播 RREP 报文，而 AIS-SMR 算法则只是向邻居节点广播 FANT 报文，节省了资源。

(3) SMR 协议的参数性能只包含跳数，而 AIS-SMR 算法则根据信息素选择下一跳，信息素包含时延，跳数，剩余能量等。

(4) SMR 协议没有路由更新部分, 大部分时间只使用一条路由, 并且对于删除或增加节点没有任何处理, 而 AIS-SMR 算法则增加了路由更新部分, 能更好的实现稳定性, 均衡负载, 有效的利用资源, 延长了网络生命。

由于大部分多路径路由协议只考虑了最短路径, 未考虑路径的稳定性, AIS-SMR 利用蚁群算法的特性, 结合各个参考因素, 在路径的稳定性方面优于其他多路径路由协议, 也更好地延长了网络的生存周期。

### 4.3.3 协议设计

AIS-SMR算法中的路由发现, 路由更新和路由维护等三个重要处理过程的设计方法具体说明如下。

#### (1) 路由发现

首先在算法中构造了如下三类人工蚂蚁:

1) 前向蚂蚁 (FANT): 表示从源节点到目的节点的蚂蚁报文, 用来收集在通往目的节点的路径中经过的各节点的信息, 如时延, 跳数, 剩余能量等。

2) 后向蚂蚁 (BANT): 表示从目的节点返回源节点的蚂蚁报文, 在归途中根据FANT收集到的信息来形成源节点到目的节点的正向路径。

3) 全局信息素更新蚂蚁 (QANT): 表示从目的节点返回源节点的蚂蚁报文, 该报文只包含两个路径信息, 全局最优和最差路径全部信息, 用于全局信息素的更新和最优最差路径的路由形成。

假设初始时每个节点的信息素值都为  $\tau_{i,j}(0) = \tau_{\max}$ , 每个节点初始能量都已知。当源节点向目的节点传输数据时, 如果该节点的路由表中有到目的节点的路由信息, 可根据公式 (1) 计算出概率来选择下一跳节点, 概率大的选择几率越大。

$$P_{i,j}(t) = \frac{[\tau_{i,j}(t)]^\alpha}{\sum_{k \in NT^i} [\tau_{i,k}(t)]^\alpha} \quad (4.3)$$

其中  $P_{i,j}(t)$  代表在时间为  $t$  的时候节点  $i$  到目的节点的路由中, 选取节点  $j$  作为下一跳节点的概率;  $\tau_{i,j}(t)$  代表在时间为  $t$  的时候节点  $i$  选择节点  $j$  为下一跳节点的路径上的信息素的值;  $\alpha$  表示信息素启发因子,  $0 < \alpha < 1$ ;  $NT^i$  表示节点  $i$  的邻居表, 是由节点  $i$  所有一跳邻居节点构成的集合。

若没有路由信息, 则生成FANT报文, 广播给所有邻居节点, 该报文中包含该节点到邻居节点的时延, 跳数, 剩余能量, 信息素, 等等, 中间节点收到FANT报文, 若该FANT报文不

是来自同一个节点或者该报文的信息素大于之前收到的FANT报文的信息素的值,则转发该报文,否则丢弃。FANT报文到达目的节点后,则会根据情况生成BANT,QANT报文,该BANT报文包含非最优和最差路径的节点选择内容,但QANT报文要包含FANT所收集集到的最优和最差路径的全部信息, FANT报文被丢弃。该BANT报文根据路径信息返回源节点,但在返回途中不会修改各节点的信息,但是QANT在返回途中会根据收集到的信息修改经过的各节点的信息素,可根据以下公式(4.1)来修改信息素的值。其中的 $\Delta\tau_{ij}^{bs}$ 和 $\Delta\tau_{ij}^{ws}$ 相当于对于最优路径和最差路径所要增加和减少的额外信息素值,计算方法统一由下面的公式给出:

$$\Delta\tau_{i,j} = \left( \frac{T_{\max}(D) - T_i^j(D)}{T_i^j(D)} + \frac{1}{HopCount} + \frac{1}{E_{ini} - E_i} \right)^\beta, \quad (4.4)$$

其中 $T_i^j(D) < T_{\max}(D), E_i > E_{\min}, HopCount \geq 1$

$$T_i^j(D) = T_i^j(j) + T_j^j(j_1) + T_{j_1}^j(j_2) + \dots + T_{j_k}^j(D) \quad (4.5)$$

其中: $T_{\max}(D)$ 表示允许的最大时延, $T_i^j(D)$ 表示节点i通过节点j到达目的节点D的时延, Hop Count表示节点i通过节点j到达目的节点的路由跳数, $E_i$ 表示节点i的剩余能量, $E_{ini}$ 表示初始能量, $\beta$ 用来调节信息素增量的速度, $0 < \beta < 1$ 。当BANT报文和QANT报文到达源节点时,正向路由形成并丢弃BANT报文和QANT报文,就可以发送数据包了。

## (2) 路由更新

路由更新分为两种情况下需要更新信息素的值,一种是发送数据时,另一种是定时更新。

1) 发送数据时:当源节点需要向目的节点传输数据时,根据邻居表中与各邻居节点的信息素浓度值,利用公式(4.3)计算出被作为下一节点的概率值,中间节点i收到这个数据包,该数据包的头部包含转发此数据包需要消耗的能量 $E_{cost}$ ,若节点i的剩余能量小于 $E_{cost}$ ,则将通过该节点转发的所有路径从路由表中删除,否则根据概率值转发该数据包,然后根据公式(4.2)更新该路径上的信息素,其中 $\Delta\tau_{ij}$ 的计算公式为:

$$\Delta\tau_{ij} = \left( \frac{1}{E_{ini} - (E_i - E_{cost})} \right)^\sigma \quad (4.6)$$

其中: $\sigma$ 用来调整信息素增量的速度, $0 < \sigma < 1$

2) 定时更新时:每隔一段时间 $\Delta t$ ,各个路径上的信息素浓度按照公式(4.7)进行挥发:

$$\tau_{ij}(t + \Delta t) = (1 - \rho)\tau_{ij}(t) \quad (4.7)$$

当路径上的信息素浓度 $\tau_{ij}(t)$ 必须满足 $0 < \tau_{\min} \leq \tau_{ij}(t) \leq \tau_{\max}$ ,如果 $\tau_{\max} < \tau_{ij}(t)$ ,则 $\tau_{ij}(t) = \tau_{\max}$ ,

同样地, $\tau_{\min} > \tau_{ij}(t)$ ,则 $\tau_{ij}(t) = \tau_{\min}$ ,确保信息素的浓度控制在 $\tau_{\min}$ 和 $\tau_{\max}$ 之间。

### (3) 路由维护

由于无线传感器网络中各节点能量的消耗情况不一样,或者某节点突然失效,或者链路发生中断,在一段规定的时间内某节点无法成功地发送数据包,则需要向该邻节点发送询问报文,查询该邻节点是否存在,当该邻节点已经死亡,若存在备用路径,切换到备用路径继续发送,若无备用路径,则将数据包保存在该节点的缓存空间中,这里有两种方式进行修复:一种是本地修复,该节点发动一次到目的节点的路由建立过程,建立恰当的路由再进行发送;另一种是发送错误消息到源节点,重新进行完整的路由建立。

#### 4.3.4 算法流程

AIS-SMR 算法是通过改进基本的 ACO 算法的前提下结合无线传感网络中已有的 SMR 多路径路由协议修改形成的,该新算法不仅保留了 SMR 协议的优点,并且在此基础上做了相关的修改,能够更好地选择最佳路径,减少节点的开销,有效地提高了网络的生存能力。在 AIS-SMR 中,路由的建立会产生三种报文,一种是 FANT 前向报文,通过该报文发现路由,另两种是 BANT 和 QANT 后向报文,通过该报文建立和更新路由。图 4.1 描述了给出 AIS-SMR 算法总处理流程。

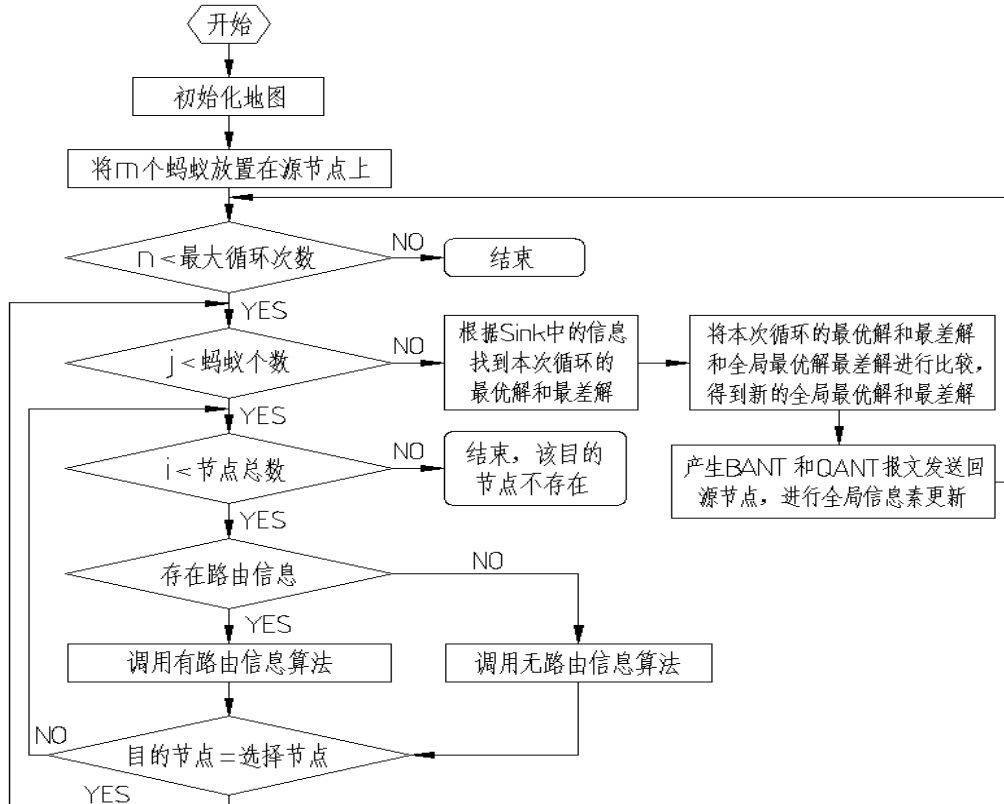


图 4.1 AIS-SMR 算法流程总流程



如当需要发送数据时，节点存在路由信息，AIS-SMR 算法通过之前计算好的概率值来选择下一个节点，并且通过局部更新信息素原则来更新该边的信息素，如图 4.2 所示。

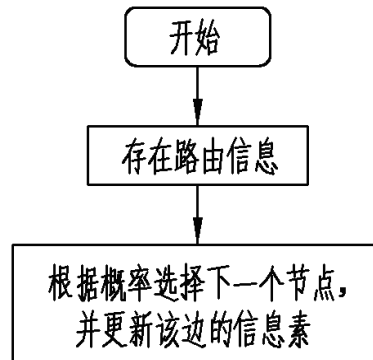


图 4.2 有路由信息时的 AIS-SMR 算法流程

如当需要发送数据时，节点不存在路由信息，AIS-SMR 算法就会进入路由发现阶段，生成蚂蚁前向报文 FANT，该报文携带某些信息被广播到相邻节点，如果该报文不是来自同一个节点或者它携带的信息素大于前一个节点，则该相邻节点就会转发该报文并更新上该边的信息素值，反之则弃之，如图 4.3 所示。

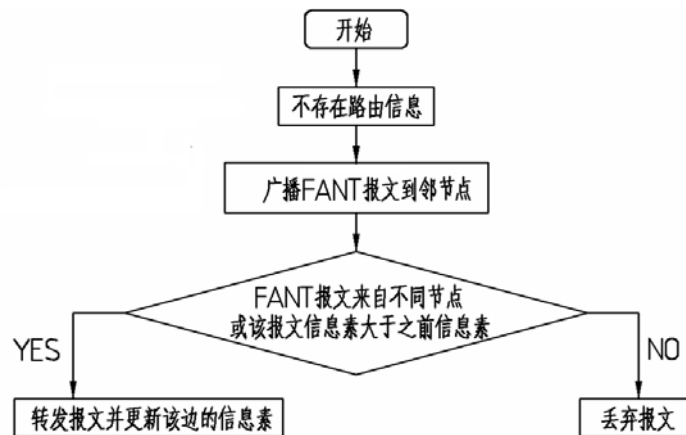


图 4.3 无路由信息时的 AIS-SMR 算法流程

AIS-SMR 算法实现的主要伪代码如下：

Procedure: //路由算法流程

BestTour; //全局最优路径

WorstTour; //全局最差路径

BTour; //本次循环最优路径

WTour; //本次循环最差路径

N; //节点个数

M; //蚂蚁个数

int i;

```

i=0;
while(i<N)
{
    放置 M 只蚂蚁在源节点上，并将蚂蚁的  $M_k$  表置空，并初始化各值;
    i++;
}End while
i=0;
int j=0;
While(i<N)
{
    While(j<M)
    {
        产生 FANT 报文;
        If（没有路由信息） then
        {
            广播该 FANT 报文到各个邻居节点;
            If（FANT 报文不是来自同一节点或者新的 FANT 报文携带的信息素大于
            之前收到的 FANT 报文携带的信息素）
            {
                加入缓存;
                中间节点转发该报文;
            }
        }
        Else 丢弃该报文
    }
    Else
        通过公式（4.3）计算  $P_{rs}^j$ , 蚂蚁 j 根据  $P_{rs}^j$  来确定 FANT 报文要发送给的下一跳
        节点 s;
        加入缓存
        If（s! =Sink）//节点 s 不是目的节点
        将  $M_k$  中的节点 r 更新为节点 s;

```

```

Else
    将节点 s 更新为 Sink 节点;
Endif
    利用公式 (4.2) 和 (4.6) 更新 (r, s) 上的信息素浓度;
    j++;
}
Endwhile
i++;
}
Endwhile
For (j=0;j<m;j++)
{
    分析 sink 节点中的各个数据, 找出本次循环中的最优路径 BTour 和最差路径 WTour;
    If BTour 比 BestTour (全局最优路径) 更优, WTour 比 WorstTour (全局最差路径) 更差
        将 BestTour 的内容更改为 BTour 的内容, WorstTour 的内容更改为 WTour 的内容
    Else
        无需修改 BestTour 和 WorstTour 的内容;
        此时会根据情况产生 BANT 报文和 QANT 报文, BANT 报文包含非最优和最差路径的节点选择的信息, 但是 QANT 报文要包含全局最优和最差路径的全部信息内容, 将这两类报文发送给源节点, BANT 报文无需修改节点的信息素值, 而 QANT 报文要根据公式 (4.1), (4.5) 和 (4.6) 进行全局信息素的修改。
    }
}
EndProcedure

```

上面说明的算法是蚂蚁进行一次循环的算法流程, 一次循环之后, 全部的蚂蚁又放回源节点, 进行第二次循环, 循环控制线程伪代码如下:

```

Procedure:    //循环控制线程
CurrNum=1;    //当前循环次数
TotalNum=N;   //总共循环次数

```

```

For (CurrNum<TotalNum; CurrNum++)
{
    进行路由算法;
    将蚂蚁重新放到源节点;
}
Endfor
EndProcedure

```

## 4.4 AIS-SMR 的结构设计

本文对 AIS-SMR 算法的结构设计分为两部分进行设计：节点设计和报文设计，并且给出了具体数据结构的定义。

### 4.4.1 节点设计

无线传感器网络中的节点需要保存该节点的属性信息，与邻节点之间的联系信息和到达目的节点的路由表，这样才能完成数据的传输。

(1) 节点属性表。节点属性表用来保存各个节点自身的信息，包括以下信息：

ID	Site	State	Hop	Energy
----	------	-------	-----	--------

**ID:** 用于标识本节点的节点 id，具有唯一性；

**Site:** 本节点的坐标；

**State:** 本节点的当前节点状态；Y 代表节点存在,N 代表节点删除，具体状态根据当时情况而定；

**Hop:** 本节点到目的节点的跳数；

**Energy:** 本节点的剩余节点能量

说明：如果本节点的剩余节点能量低于发送数据所需要的最低能量的话，该节点将会被删除，State 的状态将置为 N，并且要将包含该节点的邻节点链表中的那一条记录和该节点的所有邻节点链表删除。

(2) 邻节点链表。节点路由表是用于记录该节点的邻居节点的 id 以及对应的一些信息，包括以下信息：

NID	Nsite	Nenegy	NInfo	P	Time
-----	-------	--------	-------	---	------

**NID:** 用于表示邻节点的 id;

**NSite:** 邻节点的坐标;

**NEnergy:** 邻节点的剩余能量;

**NInfo:** 邻节点的信息素值;

**P:** 选择该邻节点的概率, 其值可以通过公式 (4.3) 得出;

**Time:** 选择该邻节点的延迟时间。

说明: 当蚂蚁报文从节点  $s$  能够向节点  $d$  传播信息, 表明节点  $d$  是节点  $s$  的一个邻居节点, 则节点  $s$  的邻节点链表中要保存节点  $d$  的信息。若节点  $s$  的邻节点链表中存在节点  $d$  的信息, 先判断该节点  $d$  的剩余能量是否超过算法中设定的最小能量, 如果大于最小能量值, 则更新节点  $s$  的节点属性表中的节点  $s$  的剩余能量值, 并且更新邻节点链表中的邻节点的信息素值, 选择该邻节点的延迟时间和概率; 如果小于最小能量值, 则该邻节点相当于一个死亡节点, 将该邻节点信息从表中删除。若节点  $s$  的邻节点链表中不存在节点  $d$  的信息, 则添加一条关于节点  $d$  的信息, 包括节点  $d$  的 id, 坐标, 节点剩余能量, 信息素值, 延迟时间。

### (3) 节点路由表

当源节点  $r$  向目的节点  $\text{sink}$  传输数据时, 如果节点  $r$  存在到目的节点  $\text{sink}$  的路由信息, 则根据概率选择某一条路径进行传递, 如果没有则需要为节点  $r$  和目的节点  $\text{sink}$  建立路由信息, 包括以下内容:

PID	NextID	SSite	NSite	Info	P	Time
-----	--------	-------	-------	------	---	------

**PID:** 目的节点的 ID;

**NextID:** 下一跳节点的 ID;

**SSite:** 目的节点的坐标;

**NSite:** 下一跳节点的坐标;

**Info:** 选择该路径时目的节点的信息素值;

**P:** 选择该路径的概率;

**Time:** 选择该路径时的延迟时间。

说明: 当源节点  $r$  向目的节点  $\text{sink}$  传输数据时, 如果存在节点  $r$  到目的节点的路由信息, 则根据路由表中的信息找出最优路径, 进行传输, 并在其传输数据的同时更新路由信息; 如果不存在该路由信息, 则需要进行路由发现操作, 生成蚂蚁报文。

### (4) 蚂蚁记录表。记录蚂蚁所经过的信息, 为蚂蚁返回的时候提供线路。

AID	BID	NID
-----	-----	-----

**AID:** 蚂蚁 ID

**BID:** 该节点前一跳的节点 id;

**NID:** 该节点后一跳的节点 id。

#### 4.4.2 报文设计

在路由发现阶段,本文构造了三类人工蚂蚁,分别是前向蚂蚁(FANT),后向蚂蚁(BANT)和全局信息素更新蚂蚁(QANT),前向蚂蚁用于发送 FANT 报文,建立通往源节点的反向路径,同时需保证对各个节点信息素值的更新;后向蚂蚁用于发送 BANT 报文,仅建立到源节点到目的节点的正向路径;全局信息素更新蚂蚁用于发送 QANT 报文,不仅用于建立源节点到目的节点的正向路径,还用于全局信息素的更新。

(1) 前向蚂蚁报文(FANT)。承载某些数据向 sink 节点传输,该报文要包含以下信息:

Type	AID	PID	Energy	Time	Hop	LID	NextID	Data
------	-----	-----	--------	------	-----	-----	--------	------

**Type:** 报文类型,FANT 报文类型值设为 1;

**AID:** 蚂蚁 id;

**PID:** 目的节点 id;

**Energy:** 节点剩余能量;

**Time:** 该蚂蚁所经过的路径的总延迟;

**Hop:** 该蚂蚁所经过路径的总跳数;

**LID:** 蚂蚁传输数据的当前节点的 id;

**NextID:** 蚂蚁传输数据的后一跳节点的 id;

**Data:** 蚂蚁传输的数据;

说明: 需要进行路由建立时会产生一个 FANT 报文,包含以上信息,广播给所有邻居节点,如果该报文不是来自同一个节点或者该报文的信息素大于之前收到的报文的信息素,根据 FANT 报文包含的信息更新邻居节点的信息素值,根据信息素值、时延、跳数来计算选择该邻居节点的概率,并把上一跳节点和下一跳节点的 id 添加到对应的节点信息表中。如果广播到的邻居节点正好是目的节点,那一条源节点到目的节点的路由路径就产生了,根据这种方法会产生多条路径,根据路径的信息素值找到一条最优路径和一条最差路径,并产生 BANT 报文,包含最优路径和最差路径的信息。

(2) 后向蚂蚁报文(BANT)。后向蚂蚁报文是用来建立源节点到目的节点的正向路径和更新节点的跳数。

Type	AID	SID	Hop	LID	NextID
------	-----	-----	-----	-----	--------

**Type:** 报文类型, BANT 报文类型值设为 2;

**AID:** 蚂蚁的 id;

**SID:** 源节点的 id;

**Hop:** 该蚂蚁经过的跳数, 将节点跳数加以更新;

**LID:** 当前节点的 id;

**NextID:** 下一跳节点的 id;

说明: 发送该报文的路径无需对该路径上的节点进行信息素的更新。

(3) 全局信息素更新蚂蚁报文 (QANT)。该报文不仅要建立源节点到目的节点的正向路径, 也要对最优和最差路径进行全局信息素的更新。

Type	Rtype	AID	SID	LID	NextID	Hop	Time
------	-------	-----	-----	-----	--------	-----	------

**Type:** 报文类型, QANT 报文类型值设为 3;

**RType:** 路径类型, 最优路径类型值设为 1, 最差路径类型值设为 0;

**AID:** 蚂蚁的 id;

**SID:** 源节点的 id;

**LID:** 当前节点的 id

**NextID:** 下一跳节点的 id;

**Hop:** 该蚂蚁经过的跳数;

**Time:** 到目的节点的延迟。

(4) 询问报文 (Ask Package, ASP)。该报文用于检验网络中的某个节点是否还存在, 包含以下信息:

Type	ID	Site
------	----	------

**Type:** 报文类型, ASP 报文类型值设为 4;

**ID:** 要询问的报文 id, 若该 id 为 1, 则表示为广播;

**Site:** 要询问的报文的坐标。

说明: 当数据包发生丢失时, 当预定的时间超时, 可向该节点发送询问报文, 查询该节点是否还存在, 如果该节点已经不存在的话, 就删除和该节点有关的所有信息, 当在网络中增加新节点时, 可以将该节点广播发送到所有邻居节点, 建立该节点的邻节点链接表。

(5) 回答报文 (Answer Package, ANP)。用于对询问报文的回应, 该报文包含以下信息:

Type	State	NID	NSite	Nenergy	Ninfo	P	Time
------	-------	-----	-------	---------	-------	---	------

**Type:** 报文类型, ANP 报文类型设为 5;  
**State:** 查询节点的状态, 1 表示存在, 0 表示不存在;  
**NID:** 用于表示邻节点的 id;  
**NSite:** 邻节点的坐标;  
**NEnergy:** 邻节点的剩余能量;  
**NInfo:** 邻节点的信息素值;  
**P:** 选择该邻节点的概率, 其值可以通过公式 (4.3) 得出;  
**Time:** 选择该邻节点的延迟时间。

说明: 若查询报文的 ID 值为非 0, 则回答报文只需返回该 ID 的状态值 **State**, 其他可以为空; 若查询报文的 ID 值为 0, 说明是新节点的加入, 广播给所有邻居节点, 回答报文要提供 NID, NSite, NEnergy, NInfo 和 Time 这五个方面的值, 使该节点可以建立其邻节点链接表。

#### 4.4.3 数据结构定义

##### (1) 蚂蚁报文数据结构定义

蚂蚁报文包括前向蚂蚁报文, 后向蚂蚁报文和全局信息素更新蚂蚁报文, 对其定义如下:

//蚂蚁报文宏定义

```
#define AIS_SMR_TYPE_FANT    0x03    //前向蚂蚁报文
#define AIS_SMR_TYPE_BANT    0x04    //后向蚂蚁报文
#define AIS_SMR_TYPE_QANT    0x05    //全局信息素更新蚂蚁报文
```

//具体蚂蚁类报文数据结构定义

```
struct hdr_AIS_SMR_ant{
    u_int8_t Ant_Type;           //蚂蚁报文类型
    u_int32_t  Ant_ID;           //蚂蚁 id 号
    u_int32_t  Ant_Pur_ID;       //sink 节点 id
    double     Ant_Energy;       //节点剩余能量
    double     Ant_Time;         //节点总延迟
    u_int8_t Ant_Hop;           //总跳数
    u_int32_t  Ant_LID;          //蚂蚁当前节点 id
    u_int32_t  Ant_NextID;       //蚂蚁下一跳节点 id
```



```

double    Ant_Length;    //经过节点是所需发送数据包的总长度
double    Ant_Link_Sum;   //报文所经过所有节点的总和
AIS_SMR_ANT_ROUTE Ant_Route_List; //报文所经过的节点的排列
};

```

### (2) 回答询问类报文数据结构定义

该报文分查询类和回答类报文，对其定义如下：

//控制报文宏定义

```
#define AIS_SMR_TYPE_ASK    0x0a;    //询问报文
```

```
#define AIS_SMR_TYPE_ANSWER 0x0b;    //回答报文
```

//具体数据结构定义

```

Struct hdr_AIS_SMR_CONTROL {
u_int8_t Control_Type;           //报文类型
u_int8_t Control_Sink_State;     //节点状态
u_int32_t Control_ID;           //节点 id
u_int32_t Control_Next_ID;       //邻节点 id
double    Control_Next_Energy   //邻节点剩余能量
double    Control_Next_Info     //邻节点信息素值
double    Control_Time          //选择该邻节点的延时
};

```

### (3) 路由表数据结构定义

每个节点都会保存一个路由表，该路由表有一定的时限，时间一到就会失效，具体的数据结构定义如下：

路由表项数据结构定义

```

Class AIS_SMR_ROUTE_ENTRY{
    friend class AIS_SMR_ROUTE_TABLE;
    friend class AIS_SMR;
public:
    AIS_SMR_ROUTE_ENTRY();
    ~AIS_SMR_ROUTE_ENTRY();
    void    nb_insert(addr_t id);           //节点插入

```

```

    AIS_SMR_Next* nb_look_up(addr_t id);           //节点查询

    AIS_SMR_Path* path_insert(addr_t hh, u_int32_t h, double delay_time,
    addr_fh, double q, double cs, double ph, double ps);           //路径插入

    AIS_SMR_Path* path_look_up(addr_t next_id,addr_t last_id);     //路径查询
AIS_SMR_Path* misjoint_path_look_up(addr_t next_id,addr_t last_id); //不存在节点查询
bool    new_misjoint_path(addr_t next_id,addr_t last_id);
void    path_delete(addr_t id);           //删除路径
void    path_delete();                   //删除所有路径
void    path_delete_worst();             //删除质量最差路径
bool    path_empty();                   //路径是否为空
AIS_SMR_Path* path_find();
AIS_SMR_Path* path_find(int n);
AIS_SMR_Path* path_find_AntBest();     //找出最优路径
void    p_insert(addr_t id);
void    p_delete(addr_t id);
void    p_delete();
bool    p_empty();
double fant_timeout;
double  fant_count;
protected:
LIST_ENTRY(AIS_SMR_ROUTE_ENTRY) rout_link;
addr_t   rout_dst;
int      rout_last_hop;
AIS_SMR_PATH  rout_path_list;
int      rout_path_number;
bool     path_error;
double  rout_delay_time;
u_int8_t rout_type;
double  rout_energy;
};

```

路由表数据结构定义：

```
class AIS_SMR_ROUTE_TABLE{
public:
    AIS_SMR_ROUTE_TABLE();
    ~AIS_SMR_ROUTE_TABLE();
    AIS_SMR_ROUTE_ENTRY*      Head();
    AIS_SMR_ROUTE_ENTRY*      route_add(addr_t id);
    AIS_SMR_ROUTE_ENTRY*      route_look_up(addr_t id);
    void      route_delete(addr_t id);
    void      route_dump_table();
    bool      route_active();
    void      route_ant_choose();
private:
    List_Head(AIS_SMR_Head, AIS_SMR_ROUTE_ENTRY)route_head;
};
```

## 4.5 AIS-SMR 算法的性能分析

本文对算法改进前后的有关性能进行比较，来分析说明 AIS-SMR 算法的合理性和有效性。在 AIS-SMR 算法中，先是将基础的蚁群优化算法 AS 算法进行了改进，主要是对信息素更新规则的改进，一方面能够使蚂蚁能够聚集在最优路径附近进行搜索，另一方面也使蚂蚁能够探索到更多的路径，而不至于陷于局部优化停滞状态。SMR 算法本身就是一种多路径路由协议算法，它能够找到多条可用路径，但是由于该算法的局限性，使得经常只是在使用一条路由，导致该路由上的节点过早的能量耗尽，而其备用路由慢慢老化，并且该算法只考虑了节点间跳数的因素，这样找到的路由不一定是一条最优的路由。在 SMR 中引入 AIS 的想法，将跳数、剩余能量和时延综合考虑，映射为蚂蚁寻食过程释放的信息素，通过信息素值来计算选择概率，通过概率值来控制路径。这样不仅能找到多条路径，而且找到的路径相对来说更稳定，更优。由于会对传输路径上的信息进行更新，这样就不会长期使用一条路径，不仅解决了节点能耗的问题，而且也防止了备用路径的老化，从而达到了延长网络的生存周期。

AIS-SMR 是一种链路不相关路由协议算法, 链路不相关路由与节点不相关路由相比较而言, 节点不相关路由带有的约束条件相对较多。同相关路由相比, 链路不相关路由稳定性更好, 而且。AIS-SMR 考虑了多个性能参数从而得到的多条链路不相干路径, AIS-SMR 协议建立的路由这具有更好的稳定性, 由于其较好的稳定性, 有效地降低了路径断裂的可能, 减少了重新进行路由发现的次数。

AIS-SMR 算法是通过广播前向蚂蚁报文建立路由, 并且只是向邻居节点广播蚂蚁报文, 减少了资源的开销。路由建立过程中, AIS-SMR 算法是采用按需路由策略, 只在源节点发送数据而无路由信息的情况下, 才会生成前向蚂蚁报文, 减小了路由的控制开销。在 SMR 的基础上, 加入了路由更新策略, 在发送数据时, 会去更新节点的剩余能量, 利用公式 (4.2) 和公式 (4.6) 进行节点信息素的更新, 这时无需产生蚂蚁报文, 而只要在数据报文中的控制报文加入剩余能量的信息, 这样能够实时监测到节点的状态。

在复杂度方面, AS 算法会较早的收敛于路径解, 因此 AIS 基于 AS 算法进行了更新, 修改了信息素的更新规则, AIS-SMR 是 SMR 基于 AIS 改进的多路径路由协议。在 AIS-SMR 算法中, 存在  $n$  个传感器节点,  $m$  个蚂蚁一开始被放置在源节点上, 每个蚂蚁都要进行  $N$  次从源节点到目的节点的循环才能得出最终的路径, 因此可以得出以下结论。则有: 一次循环中需要对 AIS-SMR 初始化次数的复杂度为  $O(n)$ ; 一次循环中所有路径上的边需要进行信息素更新次数的复杂度为  $O(m \cdot n)$ ; 一次循环中寻找本次循环最优和最差路径次数的复杂度为  $O(m)$ ; 更新全局最优和最差路径上的节点的信息素浓度次数的复杂度为  $O(n)$ ; 一共要进行  $N$  次循环, 所以 AIS-SMR 算法的总时间复杂度为  $O(N \cdot m \cdot n)$ ; 因为有  $n$  个节点,  $m$  个蚂蚁, 每个节点都需要存储访问本节点的蚂蚁及其他信息, 所以需要的空间复杂度为  $O(m \cdot n)$ 。

## 4.6 本章小结

本章具体论述了算法改进的策略与实现方法。针对无线传感器网络的特点, AIS-SMR 算法中除了考虑跳数以外还加入了时延和剩余能量, 通过蚂蚁在路径上信息素浓度的释放来选择路径, 这样建立的路由具有更好的稳定性, 使得本来长期使用一条路径传输的 SMR 算法可以动态的选择最优的路径进行传输, 而不会一直集中于一条路径, 避免了某些节点因为长期使用能量消耗过快而失效, 并且 AIS-SMR 算法添加了对节点失效和新增的处理。同时, 在 AIS-SMR 算法中, 更改了信息素的更新规则, 使得蚂蚁能更多地在最优解周围探索并且能发现更多的路径信息, 也有利于延长了网络的生存时间。通过 AIS-SMR 算法的性能分析可知, AIS-SMR 算法有效地提高了路由的质量, 延长了网络的生命周期。

## 第五章 仿真实验与分析

本章将对改进的多路径路由协议 AIS-SMR 算法进行仿真,一开始会对仿真工具 NS2 进行简要介绍,介绍一般仿真的步骤。本章将会给出本文所需要的仿真参数,通过从节点生存个数、平均能量消耗、路由发现频率、平均时延四个方面来与 SMR 算法进行比较,并对其结果作出了性能评价。

### 5.1 NS2 仿真工具

计算机网络技术的发展需要研究人员进行不断的实验,但是由于种种原因,有些实验难于在真实的环境下进行设计和测试。NS2<sup>[45]</sup> (Network Simulator Version 2) 是由 LBL、加州大学伯克利分校和 USC/ISI 联合实验室开发,为了给新的网络协议提供一个虚拟的网络实验平台,利用该平台的仿真工具设计并开发新的网络协议。NS2 为有线网或无线网上的路由协议、多播协议、TCP/IP 协议以及其他各种协议进行模拟实验。NS2 主要使用 C++ 来编写底层的模拟引擎,模拟时用的命令和用于配置接口的语言都是 OTcl 语言,OTcl 语言是面向对象的语言;NS2 使用 TCL 脚本语言来帮助用户来开发工具 Tk,Tk 工具是用于开发图形界面的;NS2 拥有属于自己的类层次结构并且在 Tcl/Tk 基础上扩展了 OTcl;用户还可以使用 NS2 的可选 X graph 和 Nam 构造出图形化的结构,便于用户观察和分析仿真结果;NS2 编译类对象使用 OTcl 与相对应的解释类对象建立连接,该方法能够方便用户修改 C++ 对象函数,仿真器的灵活性和一致性得到了充分的体现。

在 NS2 中经常用到链路、代理、节点、应用等等网络构件。链路是指用来连接网络中各个节点的一条线路。真实网络中数据包的缓冲、数据传输的延时以及其他各种情况都可以通过链路上的队列、延迟等情况进行模拟。NS2 的链路不仅具有 TCP/IP 体系结构中数据链路层的作用,而且含扮演了物理层和部分网络层的角色。代理在构建分组的时候,它相当于网络层分组的开始,在销毁分组的时候,它相当于网络层分组的结束。通过代理可以满足不属于一个层的网络协议。比如 NS2 中的 Agent/TCP 实现了 TCP 协议,Agent/UDP 实现了 UDP 协议。在实际网络的不同位置分布着主机、路由器、交换机、传感器等各种网络设备,在 NS2 中将这此网络设备统称为节点。NS2 中的应用对象被用来模拟如 FTP、HTTP 等应用层的各种对象。

## 5.2 NS2 仿真步骤

进行网络仿真一般先要分析仿真的内容,是两个层次都使用 OTcl 编程层次还是使用基于 OTcl 和 C++两者的编程层次。OTcl 编程是使用 NS 中固有的元素和固有库进行网络仿真,此时用户只需要对仿真环境进行数据的配置,对 NS 无需任何修改,也无需自己编写 OTcl 脚本;基于 OTcl 和 C++编程需要用户自己开发仿真中所需要但是现有的 NS 中不存的网络元素,此时可以通过新建 C++类 OTcl 类并且编写自己的 OTcl 脚本来完成研究方案中所需要的网络仿真。用户进行仿真的基本步骤如下:

- (1) 首先编写 OTcl 脚本是用来配置所需要模拟的网络拓扑结构,编写 OTcl 脚本时可以设置链路的延迟、带宽等特性。
- (2) 将端设备之间的协议进行绑定并且建立通信业务量模型。
- (3) 通过对业务量模型所需要的参数进行配置,可以确定如何分配网络上的业务量。
- (4) 通过配置 Trace 对象,整个模拟过程就被 NS 保存了下来。完成仿真后,Trace 文件就可以被用户用来进行分析。
- (5) 编写其余辅助过程,规定何时模拟结束,如此就实现了 OTcl 脚本的编写。
- (6) 利用 NS 仿真工具解释并执行 OTcl 脚本。
- (7) 分析 Trace 文件,获取有用数据。
- (8) 重新对网络拓扑结构和业务模型进行调整,重复上述模拟过程。

具体的流程如图 5.1 所示。

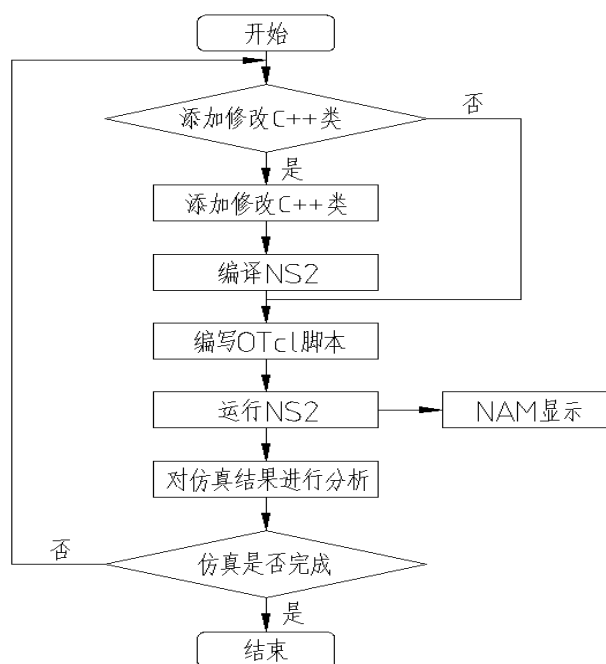


图 5.1 NS 一般仿真流程

5.3 仿真参数设置

本文使用 NS2 仿真工具对改进的算法进行仿真实验。为保证有效性和一般性，在 1200m×900m 的矩形区域中将放置 100 个移动节点，具体如图 5.2 所示：

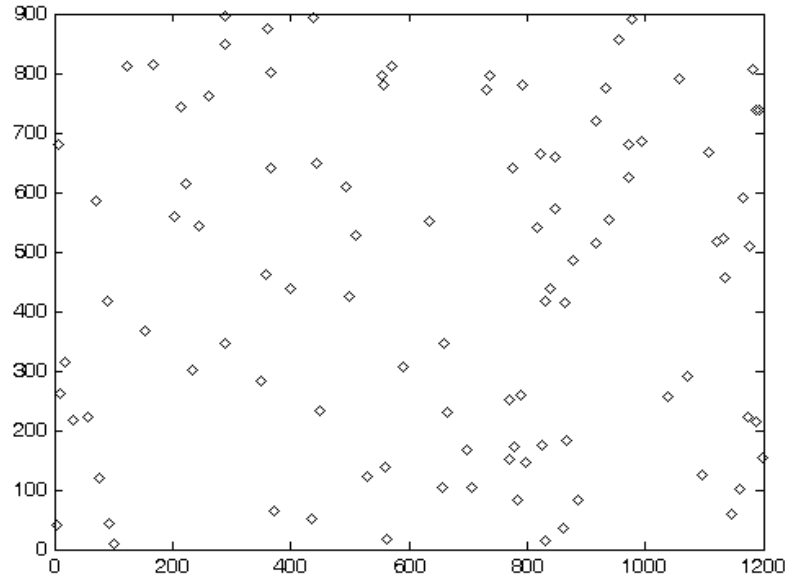


图 5.2 节点分布图

实验用到的具体参数如表 5.1 所示。在改进的算法中取  $\alpha=0.4$ ,  $\beta=0.15$ ,  $\sigma=0.45$ 。

表 5.1 仿真主要参数

参数项	值
路由协议	SMR/AIS-SMR
MAC协议	802.11
节点个数	100
节点初始能量	5J
信息素最大值	3
信息素最小值	0.001
发送功率消耗	0.1J
接收功率消耗	0.05J
信息素衰减系数	0.1
数据传输速率	1.5Mbit/s

5.4 实验结果分析

本文从节点生存个数、平均能量消耗、路由发现频率和平均时延四个方面对 AIS-SMR 算法和 SMR 算法进行了比较。

(1) 节点生存个数

图 5.3 表示 SMR 协议和 AIS-SMR 协议的节点生存数目随着时间变化的情况。由图 5.3 可知, AIS-SMR 协议比 SMR 协议的网络生存时间长, 这是因为 SMR 协议会长期使用一条主路径去传输数据, 某些节点因为长期使用而某些节点长期空置而导致网络生存时间不长, 而 AIS-SMR 协议则进行了改进, 使用概率来选择节点传输数据, 节点不会因为负责过重而死亡, 有效地延长了网络的生存时间。

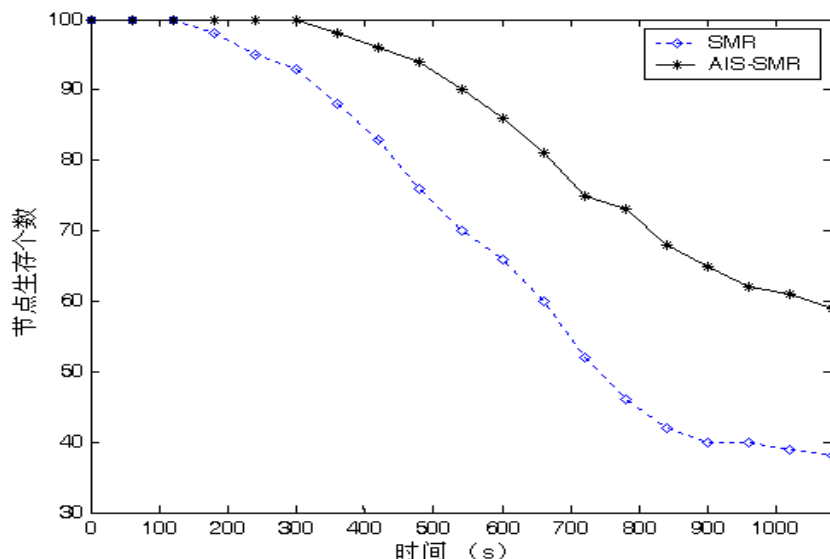


图5.3 节点生存数目随时间的变化

## (2) 平均能量消耗

图 5.4 表示 SMR 协议和 AIS-SMR 算法中时间和平均能量消耗的关系。由图 5.4 可以看出能量消耗和时间变化的关系。在开始时, 由于两种算法都要进行消息的广播, 所以消耗能量都会比较多, 但是又因为 SMR 算法是向全网广播而 AIS-SMR 算法只是向邻节点进行广播, 所以一开始用于 SMR 算法的能量就要比 AIS-SMR 算法有明显的减少。当找到优化路径以后, 两个算法的能量消耗就会慢慢趋于稳定, 但是 SMR 算法的能量消耗还是要比 AIS-SMR 算法来的多。SMR 算法虽然存在多条路径, 但是由于它长期使用一条路径进行传输, 这样会导致该路径因为负荷太重而断裂, 虽然存在备用路由, 但是也会因为长期使用而断裂, 而 AIS-SMR 算法不会长时间集中于一条路径传输, 这样路径发生断裂的概率就相对来说很小, 所以 SMR 相对于 AIS-SMR 而言, 需要重新建立路由的情况就多, 这样消耗的能量就会相对增加。



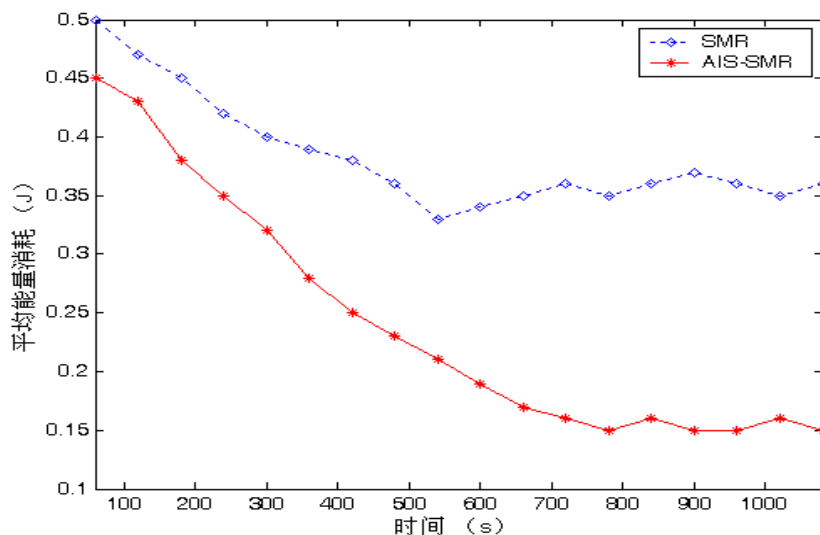


图 5.4 平均能量消耗随时间的变化

### (3) 路由发现频率

图 5.5 表示 SMR 算法和 AIS-SMR 算法的节点移动速度和路由发现频率的关系。由图 5.4 可知, AIS-SMR 算法路由发现频率低于 SMR 协议, 这是因为 AIS-SMR 算法路由选择时不仅考虑了路由跳数, 而且还考虑了时延和节点的剩余能量, 并且增加了路由更新过程, 而 SMR 协议仅仅只考虑了跳数这一个因素, 未能充分利用网络资源, 当节点快速移动, 网络经常产生改变时, SMR 协议发现的路由容易超时或中断, 从而需要启动路由发现过程, 而 AIS-SMR 算法利用蚁群算法动态地选择比较稳定的路径进行数据传输, 需要重新发现路由的次数有效的得到了抑制。

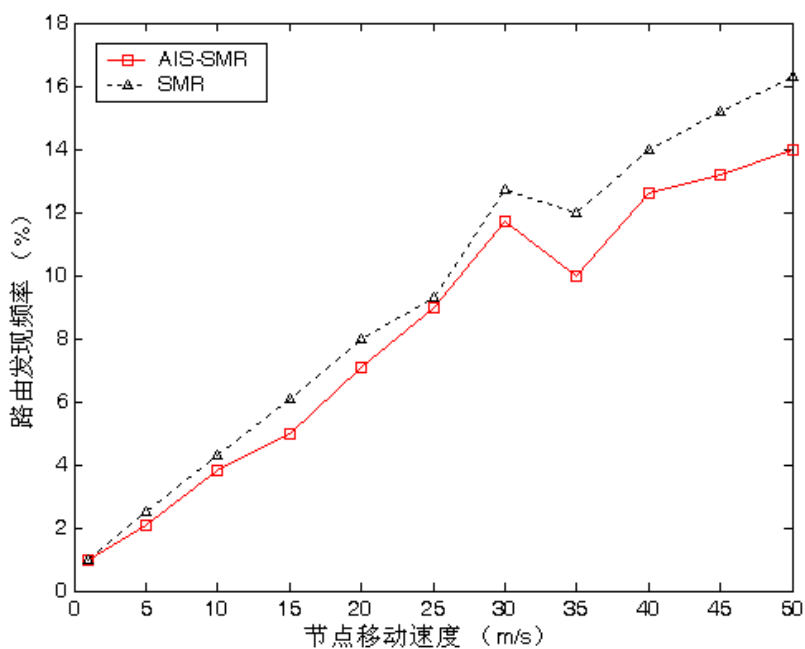


图 5.5 节点移动速度与路由发现频率的关系

### (4) 平均时延

图 5.6 表示 SMR 算法和 AIS-SMR 算法的节点移动速度和平均时延的关系。由图 5.6 可知, 当节点移动速度慢慢变大时, 两种协议的平均时延都会慢慢增大, 但是 SMR 算法时延增大的幅度要大于 AIS-SMR 算法, 换句话说, AIS-SMR 有效地降低了平时时延, 主要是因为 AIS-SMR 算法结合了改进的蚁群算法 AIS, 路径状态按照一定周期进行更新, 在跳数的基础上加入了剩余能量, 时延这两个因素, 使得节点会选择更稳定更优的路径进行传输, 降低了路径断裂的可能, 从而减少了路由重构的次数, 进而节省了路由修复和重新发送数据的时间。

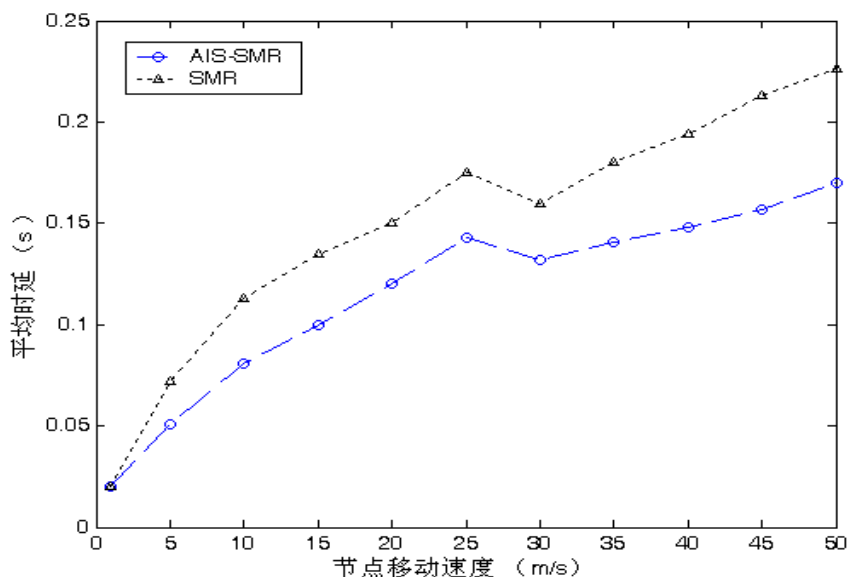


图 5.6 平均时延与节点移动速度的关系

## 5.5 本章小结

本章介绍了 NS2 仿真工具, 运用该仿真工具分别从节点生存个数、平均能量消耗、路由发现频率和平均时延四个方面对 SMR 算法和 AIS-SMR 算法进行了分析和比较, 通过仿真得到的结果可知, AIS-SMR 算法使其原来的 SMR 算法的性能得到了显著的提高。AIS-SMR 引入了改进后的蚁群优化算法思想并且综合考虑了跳数、剩余能量和延时等要素, 所以建立的路径具有更好的稳定性, 而且使网络的生存时间获得了有效地延长。

## 第六章 总结与展望

### 6.1 总结

无线传感器网络具有传统网络所没有的特有优势，所以它为许多特殊领域带来了很大量的全新应用价值。无线传感器网络在很多技术上的难点有一部分是由传感网自身的特征、节点能量有限或节点的处理计算能力有限造成的，由于这些难点的存在，导致发展无线传感器网络的技术需要更多更大的挑战。

本文的主要工作有：

(1) 本文论述了无线传感器网络的基本概念、国内外研究现状、体系结构和协议栈，详细介绍了几种典型的多路径路由协议，着重分析了 SMR 协议的优缺点，提出了本文改进路由协议的基本设计要求。

(2) 介绍了基本的蚁群算法，具体研究了精英策略蚂蚁系统、排序蚂蚁系统、最大最小蚂蚁系统和蚁群系统等典型的几种蚁群算法，通过分析这些蚁群算法的思想和优缺点，本文提出了一个改进的蚁群优化算法 AIS，加强最优解，减弱最差解，使得蚂蚁能够更多地在全局最优解的附近查找，不会过早地停滞于局部最优解。

(3) 在分析将蚁群算法加入路由协议的优势的基础上，本文设计了一种改进的多路径路由算法 AIS-SMR 算法。该算法根据信息素浓度值计算概率，然后通过概率值来选择下一跳的节点。由于 SMR 协议只考虑跳数来选择数据传输的路径，AIS-SMR 算法则要将跳数、时延和剩余能量这三部分加入到信息素浓度中进行综合考虑，这样使得找出的路径更稳定、更优。AIS-SMR 算法还加入了路由更新阶段，在数据包发送时，也能实时地更新网络中的链路情况，使得网络的能量消耗得以均衡，有助于网络的生存时间得以有效地延长，并且可以使网络的节点不会产生过劳死的情况。

(4) 本文对 AIS-SMR 算法进行了仿真，将基本的 SMR 算法和改进后的 AIS-SM 算法进行了比较，分别从节点生存个数、平均能量消耗、路由发现频率和平均时延这四个方面展开比较。仿真结果表明，通过 AIS-SMR 算法建立的路由更具稳定性，更好地延长了网络的生存时间。

## 6.2 展望

本文的研究工作还有值得进一步深入研究的地方，下一步工作可以在以下方面开展：

（1）AIS-SMR 算法中所使用的计算公式中用到的一些参数值大多是根据经验进行设置的，因此对于参数的选择需要通过更多仿真实验来进行比较分析并最终确定，这样设定的参数将更理想，需要我们进一步地研究。

（2）在 AIS-SMR 协议中，本文考虑了跳数、时延和剩余能量这三个方面的因素，以后还可以加入 QoS、丢包率等更多因素，进一步提高改进算法的有效性和适应性。

## 致 谢

一转眼，两年半的硕士生活即将结束了，在此论文完成之际，我要给予我帮助的老师、同学和亲人表示衷心的感谢，由于你们的支持和帮助，使我的研究生生涯充满了快乐和满足。

首先，我要感谢我的导师宗平教授，是他给了我机会能在南京邮电大学攻读我的硕士学位。宗老师渊博的知识、崇高的品德、敬业的精神和严谨的治学态度深深地影响了我。他对我而言既是教导我的老师，又是关心我的长辈。作为老师，刚入学时，他就对我们提出了严厉的要求，当我在论文上遇到迷茫的时候，他给予我建议和指导，帮助我完成论文，没有宗老师，我的论文是无法顺利完成的；作为长辈，他平易近人，关心我们的日常生活。无论是严师的他还是和蔼可亲的他都是我学习的榜样。

感谢我的同学陆颖颖、刘柳、徐鸽、黄力、陆虎、郭剑、张静静，谢谢你们的帮助，我才能顺利的完成我的毕业论文，更怀念我们在一起充满笑声的快乐时光。

感谢我的父母多年来辛苦的付出，没有他们的支持和关心就不会有现在的我，谢谢他们给予我这么美好的生活，谢谢他们这么多年来对我的养育之恩，我会用我的实际行动报答你们的恩情。

感谢参考文献的作者们，他们的学术见解和理论研究使我受益匪浅。

## 参考文献

- [1] 颜振亚, 郑宝玉. 无线传感器网络研究[J]. 计算机工程与应用, 2005,15:20-23.
- [2] 孙利民, 李建中, 陈渝等. 无线传感器网络[M]. 北京: 清华大学出版社, 2005:28-45.
- [3] 陈丹, 郑增威, 李际军. 无线传感器网络研究综述[J]. 计算机测与控制, 2004(8):701-704.
- [4] Byrne J. 21 Ideas for the 21st Century [J]. Business Week, 1999:78-167
- [5] Ten emerging technologies that will change the world[N]. Technology Review. Feb. 2003, 1.106(1):22-49.
- [6] LI Shan-shan, LIAO Xiang-ke, TAN Yu-song, LIU Jin-yuan. Dynamic Thread Management in Kernel Pipeline Web Server[C]. IFIP NPC 2005, Beijing, China, LNCS 3779, 2005:113-122
- [7] Robert Szewczyk, Eric Ostrweil. Habitat monitoring with sensor networks[J]. Communications of the ACM. 2004(6):20-26.
- [8] Shang Fengjun, Lei Yang. Energy efficient clustering algorithm for wireless sensor networks[J]. Journal of Chinese Computer Systems, 2009, 30(5):839-842.
- [9] Wen Yaofeng, Yang Hao, Chen Yuquan, et al. Clusters structure algorithm based on energy model in wireless sensor network[J]. Journal of Zhejiang University: Engineering Science, 2009, 43(4): 676-681.
- [10] Dong Fang, Qiu Peiliang. Angle random routing protocol for wireless sensor network[J]. Journal of Zhejiang University: Engineering Science, 2009, 43(4):621-625.
- [11] Lacoss R, Walton R. ProcDistributed SensorNets Conf, 1978:41-52.
- [12] Asada G, Dong M, Lin TS, et al. wireless integrated network sensors (WINS) for tactical information systems[C]. Proceeding of 1998 European Solid State Circuits Conference. New York: ACM Press, 1998:15~20.
- [13] DARPA Sensor Inform Action Technology Program[DB/OL].  
<http://www.darpa.mil/ito/resresearch/sensit/index.html>.
- [14] Smart Dust. <http://robotics.eecs.berkeley.edu/~pister/SmartDust/>
- [15] 王雪. 无线传感器网络测量系统[M]. 北京: 机械工业出版社, 2007.
- [16] 司海飞, 杨忠, 王珺. 无线传感器网络研究现状与应用[J]. 机电工程, 2011, 28(1):16-20+37.

- [17] 李建中,李金石,石胜飞. 传感器网络及其数据管理的概念问题与进展[J]. 软件学报, 2003,10:1717-1727.
- [18] TANG Yong, ZHOU Mingtian, ZHANG Xin. Over view of routing protocols in wireless sensor networks[J]. Journal of Software, 2006, 17(3):410-421.
- [19] Ramasubramanian S, Krishnamoorthy H, Krunz M, et al. Disjoint Multipath Routing Using Colored Tress[J]. Computer Networks, 2007, 51(8):2163-2180.
- [20] Jung SR, Lee JH, Roh BH, et al. An Optimized Node-Disjoint Multipath Routing Protocol for Multimedia Data Transmission Over Wireless Sensor Networks[C]. 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA-08). Sydney, Australia, Dec. 10-12, 2008:958-963.
- [21] Lu YM, Wong VWS. An Eneary-Efficient Multipath Routing Protocol for Wireless Sensor Networks[J]. International Journal of Communication Systems, 2007, 20:747-766.
- [22] 于继明, 卢先领, 杨余旺等. 无线传感器网络多路径路由协议研究进展[J]. 计算机应用研究, 2007, 24(6):1-3+6.
- [23] E M. Royer, C E. Perkins, An implementation study of the AODV routing protocol, Wireless Communication Conference[C]. WCNC IEEE, 2000, 3(23-28):1003-1008.
- [24] Hedeniemi S, Liestman A. A survey of gossiping and broadcasting in communication networks[J]. Networks. 1988, 18(4):319-349.
- [25] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. A survey on sensor networks[J]. IEEE Communications Magazine, August 2002, 40(8):102-114.
- [26] DORIGO M, Optimization, learning and natural algorithms[D]. Milano: Politecnico di Milano, 1992:33-35.
- [27] 马华东,陶丹. 多媒体传感器网络及其研究进展[J]. 系统仿真学报, 2002, 14(1):31-33.
- [28] Lee S J, Gerla M. Split multipath routing with maximally disjoint paths in Ad hoc networks[C]. Proceedings of IEEE ICC 2001. 2001(10):3201-3205.
- [29] Pham P, Perreau S. Performance analysis of reactive shortest path and multi-path routing mechanism with load balance[C]. IEEE/ACM Transactions on Networking. California. 2003: 724-728.
- [30] TRANH, LUX A, NGUYEN L T, et al. A novel approach for text detection in images using structural features[C]. Proceedings of ICAPR'05. 2005:627-635.

- [31] Hedeniemi S, Liestman A. A survey of gossiping and broadcasting in communication networks[J]. Networks. 1988, 18(4):319-349.
- [32] Sohrabik, Gao J, Ailawadhiv, etal. Protocols for self-organization of a wireless sensor network[J]. IEEE Personal Communications, 2000, 7(5):16-27.
- [33] YU M R, SONG Jiqiang, CAI Min. A comprehensive method for multilingual video text detection, localization, and extraction[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2005, 15(2):243-255.
- [34] Lee S J, Gerla M. Split multipath routing with maximally disjoint paths in Ad hoc networks[C]. Proceedings of IEEE ICC 2001. 2001.
- [35] Colomi A, Dorigo M, Maniezzo V, etal. Distributed optimization by ant colonies[C]. Proceedings of the 1st European Conference on Artificial Life. 1991:134-142
- [36] 王颖,谢剑英. 一种自适应蚁群算法及其仿真研究[J]. 系统仿真学报, 2002, 14(1):31-33.
- [37] Marco Dorigo, Vittorio Maniezzo, Alberto Colomi. The Ant System: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man and Cybernetics. 1996(26):29-41
- [38] M. Dorigo, E. Bonabeau, G. Theraulaz. Ant Algorithms and Stigmergy. Future Generation Computer System[J]. 2000, 16(8):851-871
- [39] B. Bullnheimer, R.F. Hartl, C. Strauss. A new rank-based version of the ant system: A computational study[J]. Central European Journal for Operations Research and Economics. 1999, 7(1): 25-38.
- [40] T. Stutzle, H. Hoos. Improvements on the ant system: Introducing MAX-MIN ant system[C]. Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms, Wien:Springer Verlag, 1997:245-249.
- [41] T. Stutzle, H. Hoos, MAX-MIN ant system and local search for combinatorial optimization problems[M]. Meta-Heuristics:Advances and Trends in Local Search Paradigms for Optmization, Kluwer Boston,1998137-154.
- [42] 段海滨. 蚁群算法原理及其应用[M]. 北京:科学出版社, 2005.
- [43] 冯跃喜,金心宇,蔡文郁. 基于改进型蚁群算法的无线传感路由协议[J]. 传感技术学报, 2007, 11(20):2461-2464.



- [44] Wei-Ming Chen, Chung-Sheng Li, Fu-Yu Chiang, et al. Jumping ant routing algorithm for sensor networks[J]. Computer Communications (S0140-3664), Elsevier B.V. 2007, 30(15):2892-2903.
- [45] Kevin Fall, Kanman Varadhan. The NS Manual, the VINT Project, A Collaboration between reaserchers at US Berkeley, LBL, USC/ISI, and Xerox PARC. <http://www.isi.edu/nsnam/ns/ns-documentation.html>.

## 读研期间发表的论文

- [1] WSN 中多路径路由协议算法的改进研究. 计算机科学与技术. 已录用.