

江苏大学

硕士学位论文

基于层次拓扑结构的无线传感器网络多路径路由算法的研究与
实现

姓名：张沛

申请学位级别：硕士

专业：计算机应用技术

指导教师：张建明

20080605

摘要

随着“普适计算”技术模式的出现,无线传感器网络作为一种新型的数据采集技术手段,在未来具有无限光明的应用前景。目前,传感器网络许多相关技术内容仍然处于探索阶段,比如:数据传输可靠性和安全性、网络协议设计、能源管理等问题。只有有效解决所面临的这些技术问题,传感器网络才能有机会真正应用到军事、医疗、商业、教育、环境等领域,发挥潜在的巨大价值,才能真正转化为生产力。因此,对其研究是大有裨益的。本文从解决无线传感器网络多路径数据可靠传输的角度进行探讨,作了一些有益的尝试,主要工作体现在以下四个方面:

第一,论述了传感器网络区别于传统有线网络和其它无线网络的一些特点,分析了一些传感器网络路由算法设计的影响因素。对现有的一些传感器网络路由算法和数据可靠传输方法进行分类的基础上,综述了它们的优缺点及其性能问题,为设计性能更好的路由算法奠定基础。

第二,为了延长传感器网络生存时间,提高能量使用效率,建立了传感器网络的层次拓扑结构。在选举簇首节点建立层次拓扑结构时,同时考虑了节点ID编号值和节点的剩余能量因素,选择剩余能量值较高的节点作为本轮的簇首,从而提高了能量使用效率,也有效的延长了网络生存时间。

第三,从多路径路由算法能够较好地提高传感器网络中数据传输的可靠性的角度考虑,在借鉴AODV路由算法思想的基础上,提出了一种基于层次型拓扑结构的多路径路由算法。该算法通过在数据源节点和汇聚节点之间建立多条节点不相关的路径,排除了路径之间的相关性,提高了路径的可靠性和数据传输的可靠性。将路径建立在层次拓扑结构之上,均衡了网络的能量消耗,满足无线传感器网络能量有效性的要求,延长网络的生存时间。

第四,在仿真平台 TOSSIM 中仿真了本文提出的基于层次拓扑结构的多路径路由算法。设计了层次拓扑结构生成、路径建立等相关功能模块,从能量有效性和传输可靠性两个角度,网络生存时间、汇聚节点接收数据分组数量和路由算法开销三个方面对本文提出的方法与现有的多路径路由算法进行了比较,验证了本文提出的可靠多路径路由算法的有效性和可靠性。

关键词: 可靠传输, 多路径, 节点不相关, 层次拓扑结构, 无线传感器网络

ABSTRACT

As a new technique of implementing Ubiquitous Computing, Wireless Sensor Networks(WSNs) which is a new technical method of data collection can be used in many aspects in the coming future. However, several technologies related to WSNs are still in investigation, such as security and reliability of transmission, network routing protocol, energy management, etc. Only if these issues can be solved successfully and effectively, WSNs will really be used in a wide range of applications in the military, health, commerce, education, environment, etc. And its potential and great value can be developed and transformed into productivity. Therefore, it is more meaningful to make scientific researches on WSNs. In this paper, mainly focusing on data reliable transmission by multi-path in WSNs, some researches are made. The main work is presented as follow:

First, WSNs' inherent characteristics are firstly described, which are different from traditional wireless networks. And we analyze some design challenge for routing protocol in WSNs. Then, based on a classification of routing protocols and data reliable transmission methods, a comprehensive survey about advantage, disadvantage and performance issues of them is done, that establish infrastructure for design nice behavior routing strategy.

Second, in order to postpone WSNs lifetime and enhance energy efficiency, we present a method to build the hierarchical structure for WSNs in this paper. Not only the ID value of sensor nodes and also the nodes' residual energy can be considered, the node with most residual energy become the cluster head by this method, so this method can get reasonable results.

Third, from the point of view of the multi-path routing algorithm which can better enhance data transmission reliability in WSNs and on the basis of drawing on the thinking of AODV algorithm, a multi-path routing algorithm based on hierarchical structure is presented. This algorithm establishes multiple paths which are node-disjoint between Source and Sink, excludes the correlation of paths, improves the reliability of paths and data transmission. Establishing paths on the basis of hierarchical structure can balance energy consumption of WSNs, meet with the requirement of energy efficiency in WSNs, postpone WSNs lifetime.

Fourth, we simulate the multi-path routing algorithm in TOSSIM, which is one of simulation tools for WSNs. We develop the modules of hierarchical topology structure and multi-path establishing. From the point of view of energy effectiveness and transmission reliability in WSNs, through comparison and analyze among other methods detailed in network lifetime, data packet number in Sink, routing algorithm cost, and we prove the effectiveness and validity of our method.

Key Words: reliable transmission, multi-path, node-disjoint, hierarchical structure, Wireless Sensor Networks(WSNs)

江苏大学学位论文版权使用授权书

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权江苏大学可以将本学位论文的全部内容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和汇编本学位论文。

保 密 ☐，在 年解密后适用本授权书。

本学位论文属于

不保密 ☒。

学位论文作者签名：

2008年6月10日

张伟

指导教师签名：

2008年6月10日

张明

独 创 性 声 明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已注明引用的内容以外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

张伟

日期：2008年6月10日

第一章 绪论

1.1 研究目的与意义

作为 21 世纪的计算模式的普适计算, Mark Weiser 在 1988 年首次提出普适计算(Pervasive Computing)思想以来, 日益受到人们的重视与关注。普适计算涉及的技术包括无线通信技术、微机电系统制造技术 MEMS(Micro-Electro-Mechanical Systems)、面向应用的嵌入式操作系统技术及相关的软件技术等。而无线传感器网络正是体现普适计算基本思想的一个典型应用, 人们可以通过无线传感器网络快捷地收集到所感兴趣的各種信息。

无线传感器网络是一种新兴的前沿技术, 目前尚处于研究阶段。国外研究人员针对无线传感器网络提出了不少研究成果, 在传感器网络数据传输、路由设计等方面的相关理论研究成果也很多。但是, 这些研究成果处于起步阶段, 距离实际需求还相差甚远。

美国的麻省理工学院、康奈尔大学、南加州大学等很多大学开展了无线传感器网络通信协议的研究, 先后提出了几种新的路由协议, 包括 SPIN^[1]、Directed Diffusion^[2]、LEACH^[3]、TEEN^[4] & APTEERN^[5]、PEGASIS^[6]等。

近年来, 国内一些科研院校也开展了无线传感器网络理论和应用的研究, 如清华大学计算机与科学系、解放军理工大学信息通信工程学院、西南交通大学计算机与通信工程学院、中山大学电子与通信工程系、哈尔滨工业大学计算机科学与技术学院和黑龙江大学计算机科学技术学院等, 但大部分工作仍处在仿真和试验阶段。

数据收集是无线传感器网络设计的初衷所在, 如何有效、可靠地将传感器节点监测的数据传输到观察点, 是研究无线传感器网络中数据传输的重要目的之一。应用于环境监测、智能家居、仓库管理等方面的传感器网络, 都应具备数据采集与传输功能。国内外研究人员在充分分析了影响无线传感器网络中数据可靠传输的因素后, 先后提出了多种解决方案, 从最初的利用链路重传机制, 到利用编码机制对监测数据进行处理以及发展到利用其他技术提高无线传感器网络中数据的可靠传输, 传感器网络中的数据传输成功率已比以往提高了许多, 但是随着传感器网络研究的日益深入, 在网络整体水平比以往有较大幅度的提高的基础上, 继续使用一成不变的数据可靠传输技术将阻碍无线传感器网络的发展, 因此传感器网络的研究发展迫切需要对其中的数据可靠传输方法进行研究与发展。

无线传感器网络与传统的无线网络相比,具有其特殊性,同时其应用领域与普通通信网络有着显著的区别^[7, 8],随着微机电系统 MEMS 和通信等技术的发展、计算成本的下降以及微处理器体积的微型化,无线传感器网络在越来越多的领域均表现出广阔的应用前景,主要表现在军事、环境、健康、家庭和商业等领域。

在军事领域,具有可快速部署、自组织性和具有较强容错能力的无线传感器网络可以用来建立一个集命令、控制、通信、计算、智能、监视、侦查和定位等功能于一体的战场指挥系统。

在商业领域,无线传感器网络可以实现家居环境和工作环境智能化。嵌入到家电和家具中的传感器节点与计算控制中心组成的无线网络与 Internet 连接在一起将会为人们提供更加舒适、方便和具有人性化的智能家居和办公环境。在其他商业领域,如工厂自动化生产线、仓储管理、智能交通等方面也引入了该技术。

在医疗及健康管理领域,可以实现远程实时监控病人的身体情况。传感器节点能够长时间的监测人体的各种生理状况,利用传感器网络可以随时了解病人的病情,一旦发生危急情况可在第一时间实施救助。

在环境监测和保护领域,无线传感器网络可用于实现诸如野生动物栖息地生态环境监控、火山爆发监控、森林火情预警监控等作用,另外,传感器网络还可以用于农、林业种植中,在空间探索和救灾抢险等特殊的环境监测领域,传感器网络也有其得天独厚的技术优势。

总体而言,我国在无线传感器网络方面的研究工作还较少。由于无线传感器网络是一门新兴技术,国内与国际水平的差距并不是很大,及时开展这项对人类未来生活影响深远的前沿技术的研究,将对整个社会和国家经济具有重大的战略意义。

1.2 国内外研究现状

无线传感器网络中数据可靠传输方法主要集中在链路重传机制、纠错码机制和多路径机制三个方面。通过对无线传感器网络中的数据可靠传输方法进行研究与分析:仅靠传统网络中的链路重传机制以及纠错码机制这类被动的提高传输成功率的方法是远远不够的,因为这两类方法对网络中的通信质量等客观环境具有很强的依赖性,不仅耗费了大量的能源,更重要的是不能够取得预期的效果,必须寻找一种能够主动解决问题的网络方案,而多路径方法正是满足需求的提高传输可靠性的方法之一,通过在源节点与目的节点之间建立多条传输通道,在前两类方法无法取得成功的条件下,多路径方法可以取得令人满意的效果,国内外一

些研究学者对利用多路径机制提高传输可靠性方面进行了研究。

Deepak Ganesan 等人在文献[9]中提出了一种新的多路径建立方法,称为缠绕多路径。该方法是为源节点到目的节点的主路径上的每个点设置一个备用节点,当主路径上的某个节点出现失误时,可立即使用备用节点进行传输,如图 1.1 所示^[10]。虽然缠绕多路径考虑了平衡多条路径之间能耗的差距,但是由于备用路径数量有限,缠绕多路径同样无法很好的满足数据可靠传输的需要,例如图 1.1 中,若节点 B、C 同时失效,则 Source 将无法传递数据到汇聚节点 Sink。

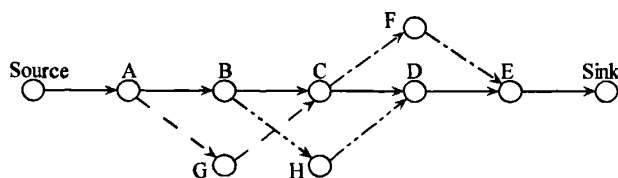


图 1.1 缠绕多路径示意图

Mahesh K. Marina 等人在文献[11]中提出了一种多路径的路由协议,该协议是在 Ad Hoc 网络中提出的一种路由协议,是通过在数据源节点和汇聚节点之间建立多条链路不相关的路径提高数据传输可靠性的一种路由机制,但是建立的多条路径之间存在关联节点,数据传输可靠性没有得到明显提高。

Rahul C 等人在文献[12]中提出了一种能量感知的路由协议,目的主要用于改善定向扩散协议的耗能情况。该方法在源节点与目的节点间建立多条路径,然后根据能量单位为每条路径分配一个被选择的概率,但是能量多路径需要提前预知全局的网络拓扑信息,这些传感器网络往往无法满足,导致其可靠性不如定向扩散协议。

De S 等人在文献[13]中提出了 MESH 路由协议,这是一种利用多路径提高数据传输可靠性的策略,它让数据在最初几跳快速展开到多条路径中,展开后的数据沿单一路径传向基站。MESH 虽然获得了较高的可靠性,但是消耗了较多的能量。

无线传感器网络的拓扑结构是设计和组建传感器网络的基础,层次型路由协议中的拓扑生成算法主要集中在两个方面:一是借助节点定位装置,对网络进行有效的划分,但是定位装置造价高,带来的能耗也比较大;二是利用层次算法进行分簇,可采用汇聚节点或基站集中控制式或分布式的方式进行,这类方法在实现过程中对网络的要求不高,不需要精确的定位机制等。目前,国内外一些学者对这方面的研究偏重于后者。

Xu Y 等人在文献[14]中提出了 GAF 算法,该算法是以节点地理位置为依据的分簇算法,GAF 是 Ad Hoc 网络中提出的一种路由算法,利用其虚拟单元格思

想为分簇机制提供了新思路。但是由于传感器节点资源的有限性,若要得到节点的地理位置,将涉及其他多方面的需求,如需配置 GPS 定位系统等,造价较高。

Gerla M 等人在文献[15]中提出了基于节点 ID 的分簇算法。该算法规定,相邻节点中具有最小 ID 的节点作为簇首,其一跳邻居节点成为该簇头所在簇的成员节点,并不再参与簇首选举过程。算法计算简单,但是倾向与选择 ID 较小的节点作为簇首节点,没有考虑剩余能量因素。

麻省理工学院的 Heinzelman 等人在文献[3]中为无线传感器网络设计了低功耗的自适应聚类路由算法 LEACH,通过随机选择聚类首领,平均分担中继通信业务,与一般的平面多跳路由协议和静态聚类算法相比,LEACH 可以将网络生命周期延长 15%。但是该算法没有考虑节点的剩余能量,同时选取的簇首在网络中的分布也不均匀,尽管如此,该方法仍然作为经典算法被后人引用。

Sonia Fahmy 等人提出了 HEED 算法^[16],该算法把节点剩余能量作为一个参量引入算法,使得选出的簇首更适合担当数据转发任务,形成的网络拓扑更趋合理,全网能量消耗更均匀,但是对于传感器节点的硬件要求较高,且需要网络中的所有节点都能够与基站进行直接通信,没有提出可行的路由建立机制,导致簇首节点能量过快消耗。

无线传感器网络中已有的数据可靠传输方法与技术均是针对平面型网络拓扑结构而设计的,无法直接运用到层次型拓扑结构中。在层次型无线传感器网络的数据可靠传输方法设计过程中,如何在资源受限的条件下,使设计的数据可靠传输方法既能够满足传感器网络中能量有效性的要求,又能够将网络监测到数据可靠地、有效地传递到汇聚节点,是研究基于层次型拓扑结构的无线传感器网络数据可靠传输问题的重要研究方面,也是本文研究的重点。

1.3 研究内容及主要工作

无线传感器网络是对21世纪人类生活方式产生重大影响的IT技术之一。它的出现引起了全世界范围内的广泛关注,其应用已经由军事国防领域扩展到环境监测、交通管理、医疗健康和工商服务等诸多领域。只有有效解决所面临的技术问题,传感器网络才能发挥潜在的巨大价值,才能真正转化为生产力。因此,对其研究是大有裨益的。本文从解决无线传感器网络多路径数据可靠传输的角度进行探讨,作了一些有益的尝试,主要工作体现在以下四个方面:

第一,深入研究无线传感器网络区别于传统有线网络和其它无线网络的一些特点,分析一些传感器网络路由算法设计的影响因素。对现有的一些传感器网络路由算法和数据可靠传输方法进行分类的基础上,总结分析它们的优缺点,为设

计性能更好的路由算法奠定基础。

第二,为了延长传感器网络生存时间,提高能量使用效率,提出一种能效较高的层次拓扑结构生成算法。该算法在选举簇首节点时,在考虑传感器节点ID编号值的同时,还考虑了节点的剩余能量因素,从而提高了能量使用效率,也有效延长了网络生存时间。

第三,从多路径路由算法能够较好地提高传感器网络中数据传输的可靠性的角度考虑,在借鉴AODV路由算法思想的基础上,提出一种基于层次型拓扑结构的多路径路由算法。该算法通过在数据源节点和汇聚节点之间建立多条节点不相关的路径,排除路径之间的相关性,提高路径的可靠性和数据传输的可靠性。将路径建立在层次拓扑结构之上,均衡网络的能量消耗,满足无线传感器网络能量有效性的要求,延长网络的生存时间。

第四,在仿真平台TOSSIM中仿真本文提出的基于层次拓扑结构的多路径路由算法。设计层次拓扑结构生成、路径建立和维护、数据传输三个模块,从能量有效性和网络可靠性两个角度,网络生存时间、汇聚节点接收数据分组数量以及路由算法开销三个方面对本文提出的方法与现有的多路径路由算法进行了比较,验证本文提出的可靠多路径路由算法的可靠性和有效性。

1.4 论文组织结构

本文共分为六章,其主要内容概要如下:

第一章主要介绍了课题的研究背景和研究意义,概述了无线传感器网络的研究现状,总结了本文的主要研究工作。

第二章阐述了传感器网络区别于传统有线网络和其它无线网络的一些特点,分析了一些传感器网络路由算法设计的影响因素。对现有的一些传感器网络路由算法和数据可靠传输方法进行分类的基础上,总结分析了它们的优缺点。

第三章分析了层次拓扑结构对路由协议的影响,定义了网络模型,并提出了具有能量意识的层次拓扑结构生成算法。

第四章提出了一种建立在层次拓扑结构之上的,在数据源节点与汇聚节点之间建立多条路径的多路径路由算法,设计并分别说明了该算法中的多路径路由的建立、多路径路由的维护和数据传输三个部分。

第五章介绍了本文采用的仿真工具——TOSSIM,介绍了多路径路由协议的仿真实现过程,并就仿真结果给出了性能评价。

第六章对本文的工作进行总结,并就下一步工作做出了展望。

第二章 无线传感器网络路由协议及可靠传输方法研究

由于无线传感器网络与传统网络相比具有自身的特点,决定了其网络路由协议的设计会与传统网络有所不同,本章从无线传感器网络的体系结构和特点出发,分类并分析现有的路由协议,阐述无线传感器网络中的可靠传输方法,最后总结多路径路由机制的相关概念、分类及面临的问题。

2.1 无线传感器网概述

2.1.1 无线传感器网络体系结构

无线传感器网络的组成如图 2.1 所示^[10, 17]: 监测区域中随机部署着大量传感器节点,这些节点通过自组织的方式构成网络。每个节点既有数据采集又有路由功能,采集到的数据经过多跳转发传输至汇聚节点或者基站,并可以通过互联网或卫星到达管理节点。用户通过基站或者管理节点对传感器网络进行配置和管理,发布监测任务以及收集监测数据。无线传感器网络系统通常包括传感器节点(Sensor Node)、汇聚节点(Sink)和管理节点(Manager Node)。

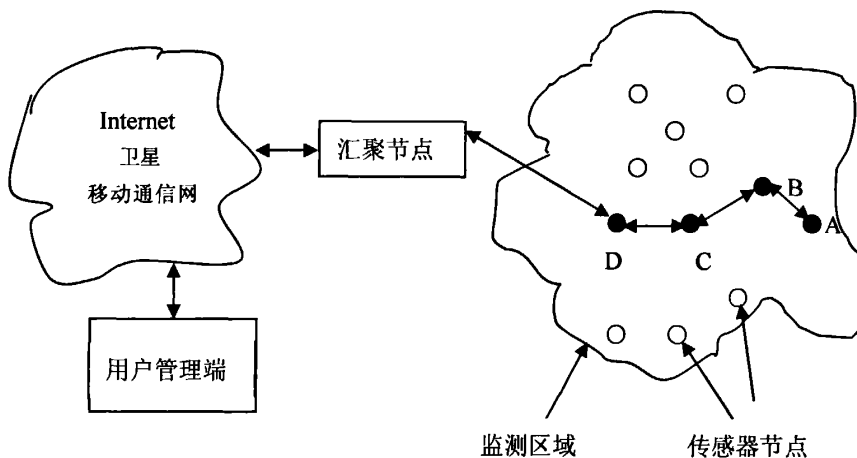


图 2.1 无线传感器网络的体系结构

在图 2.1 中,大量传感器节点随机部署于整个监测区域中,以自组织的方式构成网络。传感器节点在对所监测到的信息进行初步处理之后,以多跳中继的方式将其传送给汇聚节点,然后经过卫星、互联网或是移动通信网络等途径最终到达用户所在的管理节点。终端用户也可以通过管理节点对无线传感器网络进行管理和配置、发布监测任务以及收集监测数据。

传感器节点通常是一个嵌入式系统,由于受到体积、价格和电源供给等因素的限制,它的处理能力、存储能力相对较弱,通信距离也很有限,通常只与自身通信范围内的邻居节点交换数据。要访问通信范围以外的节点,必须使用多跳路由。为了保证监测到的数据信息能够通过多跳传输到汇聚节点,节点的分布要相当密集。从网络功能上看,每个传感器节点都具有信息监测和路由的双重功能,除了进行本地信息收集和数据处理外,还要存储、管理和融合其他节点转发过来的数据,同时与其他节点协作完成一些特定任务^[18]。

汇聚节点通常具有较强的处理能力、存储能力和通信能力,它既可以是一个具有足够能量供给和更多内存资源与计算能力的增强型传感器节点,也可以是一个带有无线通信接口的特殊网关设备。汇聚节点连接无线传感器网络与外部网络,通过协议转换实现管理节点与传感器网络之间的通信,把收集到的数据信息转发到外部网络上,同时发布管理节点提交的任务。

2.1.2 无线传感器网络特点

无线传感器网络的特点是节点的电源能量有限、通信能力有限、计算能力有限、与物理世界紧密耦合、大规模密集部署、网络动态性强。为了准确、及时的获取信息,必须依靠节点间的协作,大量传感器节点只有通过低功耗无线电通信技术连成网络才能够发挥其整体和综合作用。

传感器网络虽然与 Ad Hoc 网络有相似之处,但同时也存在很大的差别。Ad Hoc 网络是由几十个到上百个节点组成的、采用无线通信方式的、动态组网的多跳的移动性对等网络;其目的是通过动态路由和移动管理技术传输具有服务质量要求的多媒体信息流。通常节点具有持续的能量供给。

传感器网络是集成了监测、控制以及无线通信的网络系统,节点数目更为庞大(上千甚至上万),节点分布更为密集;由于环境影响和能量耗尽,节点更容易出现故障;环境干扰和节点故障易造成网络拓扑结构的变化;通常情况下,大多数传感器节点是固定不动的。另外,传感器节点具有的能量、处理能力、存储能力和通信能力等都十分有限。

在无线传感器网络的研究初期,人们一度认为成熟的 Internet 技术加上 Ad Hoc 路由机制是足够充分的,但后续的研究表明传感器网络 and 传统网络,如移动通信网、无线局域网、蓝牙网络、Ad Hoc 网络等相比,前者具有明显不同的特点及技术要求^[19]:

■ 大规模

为了获取精确信息,在监测区域通常部署大量传感器节点,传感器节点数量

可能达到成千上万,甚至更多。无线传感器网络的大规模性包括两方面的含义:一方面是传感器节点分布在很大的地理区域内;另一方面,传感器节点部署很密集,在一个面积不是很大的空间内,密集部署了大量的传感器节点。

■ 自组织

在无线传感器网络中,传感器节点的部署采用非人工、随机方式实施,节点的位置不能预知,节点之间的相互邻居关系预先也不知道,这样就要求传感器节点具有自组织的能力。在无线传感器网络使用过程中,经常有新节点加入或已有节点失效,网络的拓扑结构变化很快,无线传感器网络的自组织性要能够适应这种网络拓扑结构的变化。

■ 动态性

由于环境因素或电能耗尽,传感器节点容易出现故障或失效;环境条件也可能造成无线通信链路发生变化,甚至时断时通;无线传感器网络的传感器,感知对象和观察者都可能具有移动性;新节点的加入等等,这都要求无线传感器网络能够适应这种变化,具有动态的系统可重构性。

■ 可靠性

无线传感器网络通常部署在恶劣环境或人类不宜到达的区域,传感器节点可能工作在露天环境,非常容易遭到破坏,这都要求传感器节点非常坚固,不易损害,以适应各种恶劣环境条件。由于监测区域环境的限制,网络的维护工作十分困难,无线传感器网络的通信保密性和安全性也十分重要。因此,无线传感器网络的软硬件必须具有鲁棒性和容错性。

■ 应用相关

无线传感器网络用来感知客观物理世界,获取物理世界的信息量,不同的无线传感器网络应用,关心的物理量不同,因此对无线传感器的应用系统也有多种多样的要求,因而不可能像 Internet 一样,有统一的通信协议平台。

■ 以数据为中心

传统网络是以 IP 为中心的网络,每个节点拥有全网唯一的 IP 地址,数据转发以目的节点的转发为依赖。在无线传感器网络中,节点没有 IP 地址,是以数据为中心的网络,它关心的是数据本身,如事件、时间和地点,而并不关心数据是由哪个节点采集的。

2.2 无线传感器网络现有路由协议

针对不同的传感器网络应用,人们都在研究与之相适应的路由算法,目前还缺乏一种对于无线传感器网络来说普遍适用的路由算法。研究人员根据传感器网

络自身固有的特点来设计路由协议,从路由的表现形式、与应用的相关程度,无线传感器网络中路由协议分为以下两种类型^[20]:

■ 平面型路由协议

■ 层次型路由协议

在具体的路由协议设计时,可以综合几种策略,下面分别简要介绍。

2.2.1 平面型路由协议

平面型路由是指网络中各传感器节点在路由功能上地位相同,没有引入层次拓扑结构。平面型路由的优点是网络中没有特殊的节点,网络流量均匀地分散在网络中,路由算法易于实现;缺点是可扩展性小,在一定程度上限制了网络的规模。拓扑结构如图 2.2 所示^[21]。

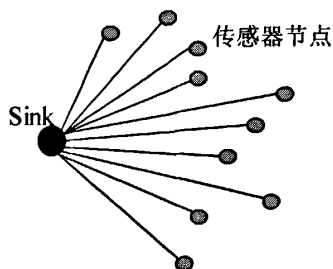


图 2.2a 平面单跳结构

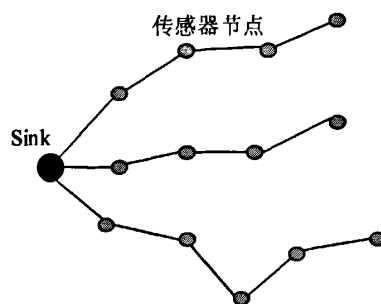


图 2.2b 平面多跳结构

下面介绍几种典型的平面路由协议。

1、SPIN (Sensor Protocol for Information via Negotiation, SPIN)

SPIN^[1]是一种以数据为中心的自适应路由协议。其目标是通过使用节点间的协商制度和资源自适应机制,解决洪泛法存在的不足之处。

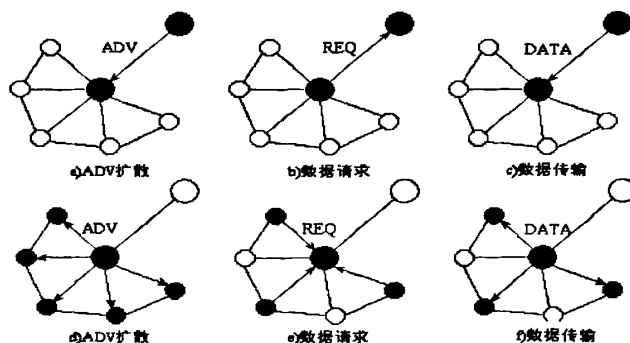


图 2.3 SPIN 协议路由建立和数据传输示意图

为了避免出现洪泛法的信息爆炸问题和部分重叠现象,传感器节点在传送数据之前彼此进行协商,协商制度可确保传输有用数据。节点间通过发送元数据(即

描述传感器节点采集的数据属性的数据, Meta-data) 进行协商, 而不是采集的整个数据。由于元数据大小远小于采集的数据, 所以, 传输元数据消耗的能量相对较少。

2、SAR (Sequential Assignment Routing)

在选择路径时, 有序分配路由策略 SAR^[22]充分考虑了功耗、QoS 和分组优先权等特殊要求, 采用局部路径恢复和多路径备份策略, 避免节点或链路失败时进行路由重计算需要的过量计算开销。为了在每个节点与汇聚节点 Sink 间生成多条路径, 需要维护多个树结构, 每个树以落在 Sink 节点有效传输半径内的节点为根向外生长, 枝干的选择需满足一定 QoS 要求并要有一定的能量储备。这一处理使大多数传感器节点可能同时属于多个树, 可任选其一将采集数据回传到 Sink 节点。

3、定向扩散(Directed Diffusion)

定向扩散^[2]是以数据为中心的路由协议发展过程的里程碑, 与已有的路由协议有着截然不同的实现机制, 其突出特点是引入了梯度来描述网络中间节点对该方向继续搜索获得匹配数据的可能性。该协议的一个关键步骤是汇聚节点 Sink 进行所谓的兴趣发布(Interest Dissemination), 也就是用一组属性描述它的查询请求, 并向全网广播以搜索匹配的数据。被称为“梯度”的度量与兴趣发布过程相关联, 以反映在不同搜索方向上找到匹配数据的可能性。这样就建立起一个与查询请求对应的“梯度场”, 传感器节点上的数据沿“梯度”最大的方向传输到 Sink 节点。其中, 源节点采集的数据首先在本地采用数据融合技术进行整合, 然后在网络中传输。该方法的缺点是没有形成到 Sink 节点的多条路径, 路由健壮性不够好。

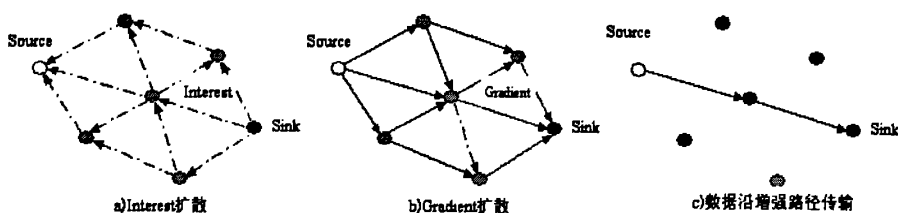


图 2.4 Directed Diffusion 路由协议路径建立过程

其他的以数据为中心的路由协议都是基于定向扩散改进或者采用类似的关键思想来提出的。定向扩散的数据命名机制, 定向扩散方式, “梯度”决定加强选择路径的原理等都包含着广阔的研究空间。

2.2.2 层次型路由协议

与平面型路由协议相对应的是层次型路由协议。层次型路由协议采用簇的概

念对传感器节点进行层次划分。若干个相邻节点构成一个簇，每一个簇有一个簇首。簇与簇之间可以直接或通过网关通信。网关可以是其他簇首节点。网关之间的连接构成上层骨干网，所有簇间通信都通过骨干网进行^[21]。拓扑结构如图 2.5 所示。

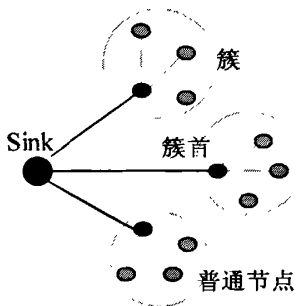


图 2.5 层次拓扑结构示意图

层次型路由协议包括成簇协议、簇维护协议、簇内路由协议和簇间路由协议四个部分。分簇协议解决如何在动态分布式网络环境下使移动节点高效地聚集成簇，它是层次路由协议的关键。簇维护协议要解决簇的产生和消亡以及在节点移动过程中的簇结构维护，其中包括移动节点退出和加入簇等功能。层次路由协议比较适合于无线传感器网络，但成簇过程会产生一定的能源消耗，如何有效的生成层次拓扑结构也正是各地学者深入研究的问题。

下面介绍几种典型的层次型路由协议。

1、LEACH (Low-Energy Adaptive Clustering Hierarchy)

LEACH^[3]是层次型路由协议中最具代表性的路由协议，其他层次式的路由协议如：TEEN、PEGASIS 等大都由 LEACH 发展而来。

LEACH 是由麻省理工学院的 Heinzelman 等人为无线传感器网络设计了自适应聚类路由算法。该协议分为两个阶段操作，即簇建立阶段 (Set-up Phase) 和就绪阶段 (Ready Phase)。为了使能耗最小化，就绪阶段持续的时间比簇准备阶段长。簇准备阶段和就绪阶段所持续的时间总和称为一轮 (Round)。在簇准备阶段，LEACH 随机选择一个传感器节点作为簇首节点 (Clusterhead Node)，随机性确保簇首与汇聚节点 Sink 之间数据传输的高能耗成本均匀地分摊到所有传感器节点。

在簇首节点选定后，该簇首节点对网络中所有节点进行广播，广播数据分组含有该节点成为簇首节点的信息。一旦传感器节点收到广播数据分组，根据接收到的各个簇首节点广播信号强度，该节点选择信号强度最大的簇首节点加入，向其发送成为其成员的数据分组。簇形成后，簇首节点采用 TDMA (Time Division Multiple Access, 时分多址) 策略分配信道使用权给簇内节点。一旦处于就绪阶段，簇首节点开始接收簇内各节点采集的数据，然后采用数据融合和数据压缩等技术

进行整合,将整合后的数据传输给汇聚节点 Sink。在就绪阶段持续了一段时间后,网络又进入了另一次的簇准备阶段。LEACH 拓扑结构如图 2.6 所示。

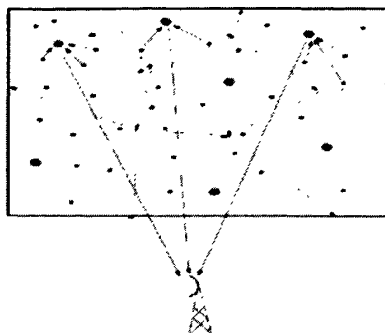


图 2.6 LEACH 拓扑结构示意图

2、TEEN (Threshold Sensitive Energy Efficient Sensor Network Protocol)

TEEN^[4]和 LEACH 的实现机制非常相似,只是前者是响应型路由协议,而后者属于主动型路由协议。在 TEEN 中定义了硬、软两个门限值,以确定是否需要发送监测数据。当监测数据第一次超过设定的硬门限时,节点用它作为新的硬门限,并在接着到来的时隙内发送它。在接下来的过程中,如果监测数据的变化幅度大于软门限界定的范围,则节点传送最新采集的数据,并将它设定为新的硬门限。通过调节软门限值的大小,可以在监测精度和系统能耗之间取得合理的平衡。

3、PEGASIS (Power-Efficient Gathering in Sensor Information System)

PEGASIS^[6]是由 LEACH 发展而来。它假定组成网络的传感器节点是同构且静止的。节点发送能量递减的测试信号,通过监测应答来确定离自己最近的相邻节点。通过这种方式,网络中的所有节点能够了解彼此的位置关系,进而每个节点依据自己的位置选择所属的聚类,聚首节点参照位置关系优化出到 Sink 节点的最佳链路。因为 PEGASIS 中每个节点都以最小功率发送数据分组,并有条件完成必要的的数据融合,减小数据流量。因此,整个网络的功耗较小。研究结果表明,PEGASIS 支持的传感器网络的生命周期是 LEACH 的近两倍。PEGASIS 协议的不足之处在于节点维护位置信息(相当于传统网络中的拓扑信息)需要额外的资源。

4、TTDD (Two-Tier Data Dissemination)

TTDD^[23]基于分布式架构,利用网络结构(Grid Structure)将网络划分成若干小区(Cell),只有处在小区边界上的传感器节点负责转发和融合数据,小区内节点只负责数据的监测和提交,不转发其他小区内节点的监测数据。在算法实现过程中,应用端驱动在整个网络区域建立网络结构,并设定离网格边界最近的传感器节点为分发节点 DN(Dissemination Nodes)。应用端发出查询请求分组,该

分组只在小区内进行泛洪，当最近 DN 节点收到查询分组时，它则向相邻的 DN 节点转发，这个过程持续到发现匹配数据源为止。在查询请求分组向整个网络进行广播的阶段，整个网络即可建立通向应用端的反向链路，使得能够监测数据能够顺利回传到应用端。图 2.7 描述了 TTDD 的网络结构。

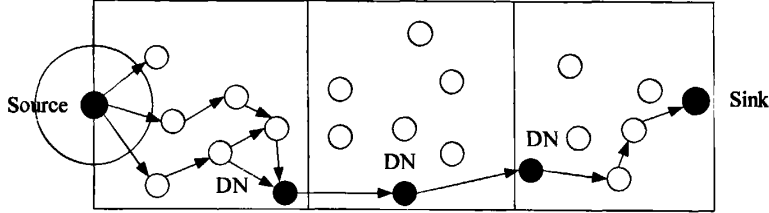


图 2.7 TTDD 协议拓扑结构示意图

2.3 无线传感器网络数据可靠传输方法

由于传感器节点能量有限，对于无线传感器网络可靠传输方法的设计极具挑战性。传感器网络的数据可靠传输方法必须以节省传感器节点的能量为出发点，并且能够最大限度的延长网络的生存时间。

考虑到无线传感器网络所具有的自身特点，目前数据可靠传输方法主要借助链路重传机制、纠错码机制和多路径机制这三种机制进行设计。

2.3.1 链路重传机制

在无线网络中，链路的失败率比有线网络中的情况严重的多，并且随着跳数的增加影响也就越大，假设每跳的丢失率为 10%，那么 15 跳以后，丢失率将高达 80%。如果一个消息在第 n 跳被丢失，那么此前所有的 $n-1$ 跳工作都浪费了，为了将数据分组再次传送到第 n 跳节点，需要在 $n-1$ 个节点中重新传输分组，假设这 $n-1$ 个节点在传送中不发生错误的话，使用链路重传机制以后，只需要一次重传就能够把消息带给需要的节点，基于无线网络通道有效性考虑，链路重传是一个比较有效的方法。

链路重传机制在传感器网络中的应用可以分为两类：第一类是指当基站或汇聚节点 Sink 收到数据分组后，发送确认信息，产生数据的源节点如果收到确认信息，就认为该数据已成功传到基站或 Sink 节点；否则，源节点在等待一段时间后将重传该数据。另一种链路重传机制是单跳确认重传机制，它通过每一跳传输时的多次重传和确认，提高了传输质量，满足数据传输成功率的要求，但是它没有考虑确认丢失的情况，在统计通信负载时也没有考虑确认的通信量。

如前例，距离基站 5 跳的节点，在各跳信道错误率为 30%（即相邻节点间的

传输成功率为 70%) 的情况下, 数据传到基站的成功率仅为 $(70\%)^5=16.8\%$, 而如果每一跳重传数据 3 次, 则一跳成功率可提高为 $1-30\%^3=97.3\%$, 那么数据传到基站的总成功率可上升到 $(97.3\%)^5=87.2\%$, 这种方法大大提高了数据传输的成功率, 但是消耗了较多的能量^[24]。

Chieh-Yih Wan 等人在文献[25]中提出了一种可靠传输协议, 该协议通过链路重传机制, 提供了逐跳修复的功能, 能够保障数据的可靠传输。

Jun Wang 等人采用单路径传输和路径修复机制^[26]。每个节点都有唯一的转发节点, 通过它可以把数据转发到基站或汇聚节点。每个节点在把数据传向转发节点后, 都要等待转发节点的确认信息, 如果收不到确认信息, 就认为传输失败。当由于信道错误或转发节点失效而导致传输失败时, 就启用路径修复机制, 查询新的转发节点, 并把数据沿新的转发节点传输。在每个数据的传输过程中, 最多允许三次修复, 当需要第四次修复时, 就放弃传输该数据。

重传和确认是解决网络信道错误问题的基本方法。通过多次重传可以提高数据的传输成功率, 但是, 考虑到无线传感器网络能源上的限制, 进行太多次的重传, 也是不可取的。此外, 传感器网络中产生的数据重要程度不同, 它们所要求的传输成功率也不同。比如在森林火灾监测中, 异常温度的数据显然要比常温的数据要求更高的传输成功率。因此, 在传感器网络中信道错误处理问题可定义为安排尽量少的确认和重传, 满足数据的传输成功率要求。

2.3.2 纠错码机制

用于提高无线传感器网络数据可靠传输性的另外一个重要机制是纠错码机制^[27], 这种机制的基本思想在于可以通过接收 n 位码字中的任意 m 位($n>m$), 就可以重建原始的 n 位的数据信息。

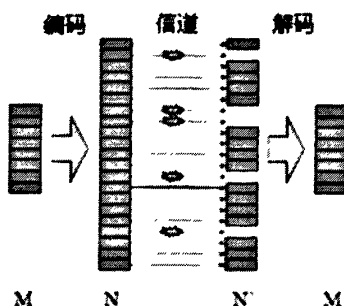


图 2.8 纠错码机制示意图

一般来说, 提高抗干扰能力(降低失真或错误概率)往往是以降低信息传输率为代价的; 反之, 要提高信息传输率又常常会使抗干扰能力减弱。编码实质上

是对信源的原始符号按照一定的数学规则进行的一种变换,纠错码的目的是引入剩余度,在传输的信息码后增加一些多余的码元(称为校验元),以使信息损失或错误后仍能在接收端被恢复。如果 n 与丢失率相比足够有效,那么就可以通过纠错码来获得可靠性,而不需要通过重传。如图 2.8 所示为高水平机制的纠错码机制示意图。

Qilian Liang 等人在文献[28]中提出了一种利用代码机制提高传输可靠性的方法 MTACCI,这种机制可以解决带宽和能量限制的问题,使无线传感器网络更具生存能力。MTACCI 方法使用多路径和编码机制相结合的方法,实现传感器网络中容忍错误的数据传输机制。该方法首先利用 Convolutional 代码对信息位进行加密,然后将码字进行交叉存取,消除由于噪音或者衰退而影响接收码字的其它相邻符号的现象,并确保在一条失败的链路上丢失的符号在解除交叉存取后,仍然能够被传播,这样就能够使 Viterbi 解码器取得更好的效果。对于 M 条路径的传输,MTACCI 方法将一个帧中的符号分成 M 个相等长度的脉冲,并且每个脉冲都在每条路上并行传输,接收节点调节每个成功到达的脉冲,并且提供软决策输出(Soft-Decision Output),接收节点根据每个脉冲传输的到达顺序将它们的软决策输出结合起来用于得到源节点传输的原始数据。

纠错码机制对于提高传感器网络中的数据可靠传输是非常有效的,但是这种编码方式在使用时存在多种限制,在编码过程中,消息的数量不能够超过用来代表消息的位数,而且码字的数量必须小于用来计算的外部空间容量^[29]。

2.3.3 多路径路由机制

多路径研究在 Ad Hoc 网络中已经有了一定的研究进展,主要以分离的,不相交的多路径路由算法为代表。Ad Hoc 网络中的多路径路由协议在有效使用带宽、降低堵塞方面起到了很大的作用,但传感器网络中的情况和 Ad Hoc 网络中的情况有所不同:传感器网络是一种处于不可靠环境下的网络,使用多路径路由协议可以大大提高网络的冗余度,同时由于构造的网络拓扑结构为层次型,当某些簇首的功能遭到破坏时可能导致网络瘫痪而引起传输数据的丢失,如果进行路由重建,则将导致大量能耗的浪费及网络延时,使用高冗余度的多路径路由协议可以有效的提高构造的层次型传感器网络的可靠性^[30]。

Deepak Ganesan 等人^[9]在定向扩散协议的基础上提出了一种在主路径之外建立若干备选路径的方案,并允许备选路径和主路径相交。通过在备选路径上低速率的传输数据来维护该路径。当主路径损坏时,就选用某一条备选路径。这种方法较好的处理了拓扑变化,却没有考虑信道质量的问题。

De S 等人在文献[13]中提出了另外一种利用多路径提高数据传输可靠性的策略 MESH, 它让数据在最初几跳快速展开到多条路径中, 避免还没有展开就传输失败的情况, 展开后的数据沿单一路径传向基站, 和 Braided 只是发送低速率数据维护备用路径不同, MESH 把数据沿多条路径传输, 因此 MESH 虽然达到了较高的可靠性, 但是也消耗了较多的能量。

Rahul C 等人在文献[12]中认为, 频繁地使用最优路径进行数据传输, 从延长网络生命时间以及网络长期连通性的角度来说并非最佳选择的, 因此文中提出了一种能量感知的路由协议。该协议的目的主要在于改善定向扩散协议的能耗情况, 采用地理位置和数据类型(即节点类型)标识节点。该方法在源节点与汇聚节点间建立多条路径, 并根据能量单位为每条路径分配一个被选择的概率, 通过偶尔使用次优路径来取得延长网络生命时间以及延长网络连通性的目的。实验结果显示, 该方法比传统的定向扩散路由延长了 40% 的网络生命时间。其缺点是汇聚节点需要周期性进行泛洪以维护路由信息, 需要进行节点间收发开销和剩余能量测量, 而且根据概率随机选择一条路径, 导致其可靠性不如定向扩散协议。

Wenjing Lou 等人在文献[31]中基于一个分布的多(多个节点)对一个(一个汇聚节点)的多路径发现协议的基础上, 提出了一种混合多路径机制 H-SPREAD 来提高安全性和可靠性。在数据源节点与汇聚节点之间存在多条不相交多路径的基础上, 再利用共享的密码进行组合, 增加了端到端的数据安全性。由于面对不可靠的无线链路以及不可靠的节点, 可选择的路由提高了每个数据分组的传输可靠性。仿真结果显示, 该混合多路径机制在提高网络安全性以及可靠性方面都比较有效。

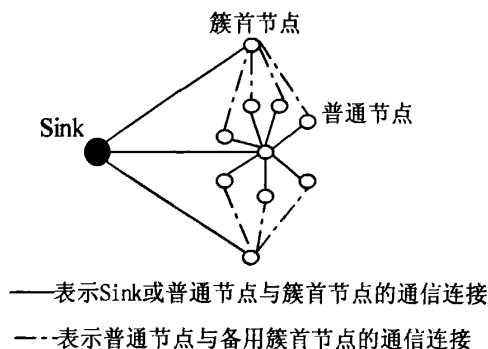


图 2.9 多路径模型

Gupta G 等人在文献[33]中, 为了满足传感器网络可靠性而提出的多路径模型, 如图 2.9 所示。为了保证路由的可靠性, 每个簇中的普通节点都维护着一个备用簇首节点的信息, 一旦本簇的簇首节点由于异常原因失效(这种情况在实际应用中很可能发生), 立即加入备用簇首所在的簇, 实现了到汇聚节点 Sink 的多

条路由，保证了路由的健壮性，但同时可以发现，由于缺乏簇首与 Sink 节点之间的可靠机制，该方法存在较高的安全隐患，如果簇首与 Sink 节点间发生通信错误，将会导致大量数据的丢失。

Intanagonwiwat C 等人在文献[2]中提出的是一种容忍错误的能量有效的无线传感器网络路由协议，它是根据洪泛路由发展而来的，但是比传统的洪泛路由更节省网络能量，而且也具有更高水平的容错性。该方法假设每个节点都有各自的地理位置信息以及基站或汇聚节点的地理位置信息，每个节点都有各自的传输角度范围，降低了泛洪方法的冗余，可以提高网络生存时间，但是该方法由于存在角度限制，节点只能在指定传输角度范围内寻找目的节点，因此该方法在数据传输可靠性方面还存在局限性。

2.4 无线传感器网络多路径路由研究

相对于单路径路由协议而言，多路径路由协议在路由可靠性、容错性、QoS 路由等方面有很多优势。因此，为了进一步提高传感器网络的路由质量，多路径路由问题逐渐成为了这几年的研究热点。

2.4.1 多路径路由相关概念

1、多路径路由定义

多路径路由是指为数据源节点和汇聚节点之间同时提供多条可用的路径，并允许节点选择如何使用这些路径。多路径路由算法为节点间提供多条路径，并确保发往其中一条路径的数据经由该路径到达汇聚节点。多路径路由网络是执行多路径路由算法的网络。

2、多路径路由使用

管理多条路径的网络协议层必须能够有效地使用多条路径，以便能够改善服务质量。适宜管理多条路径的网络协议层可以是网络层或应用层。比如，可以由应用程序甚至用户亲自决定使用哪些路径、如何使用；也可以由网络层透明地替用户提供路径的使用方法。究竟由哪个层次来管理多条路径的使用，涉及到灵活性、性能和软件工程之间的折衷。

使用模式说明了多路径如何被使用。有两种基本的使用模式：

- 同时使用多条路径传输监测数据；
- 先使用主路径传输监测数据，主路径失效后再使用其它替换路径。

称前一种方法为同时多径，后一种为替换多径。

3、多路径路由特点

- 保证了传感器节点之间数据传输的可靠性;
- 可以为不同的服务质量要求提供不同的路径;
- 可以为同一种类型的服务提供多条路径, 经聚集可实现更高的服务质量;
- 由于传感器节点对路径有自主的使用权, 它可以通过探测各路径的状况 (比如丢包率), 据此调整对各路径的使用, 从而在得到高质量服务的同时也提高了网络的利用率。因此, 多条路径的正确使用还可以提高网络的利用率。

2.4.2 多路径路由分类

多路径路由根据网络中节点之间和链路之间是否存在相关性可以分为如下三类^[34]:

- 节点不相关多路径路由(Node-Disjoint)
- 链路不相关多路路由(Link-Disjoint)
- 相关多路径路由(Non-Disjoint)

下面分别简要介绍。

1、节点不相关多路径路由(Node-Disjoint)

节点不相关路由, 也即可以称之为完全不相关路由, 路由之间没有共用的节点或者链路, 如下图 2.10 所示, 数据源与汇聚节点对 S 和 D 之间的路径 SXD, SYD, SZD 是节点不相关路由。

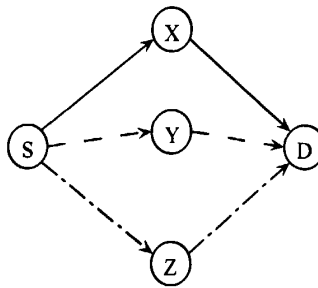


图 2.10 节点不相关路径示意图

2、链路不相关多路路由(Link-Disjoint)

链路不相关的路由之间没有共用的链路, 但是可能有共用的节点, 如下图 2.11 所示, 数据源与汇聚节点 S 和 D 之间的路径 SYD, SXYZD 是链路不相关路由, 节点 Y 为共用节点。

3、相关多路径路由(Non-Disjoint)

相关路由之间既有共用的节点, 也有共用的链路, 如下图 2.12 所示, 数据

源-汇聚节点对 S 和 D 之间的路径 SYD , $SXYD$ 是相关路由, 节点 Y 为共用节点, YD 为共用链路。

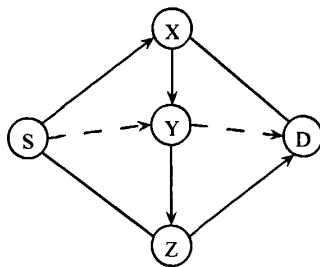


图 2.11 链路不相关路径示意图

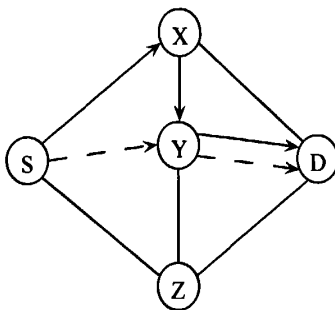


图 2.12 相关路径示意图

节点或链路相关路由同节点不相关路由相比, 它所占用的资源要少, 因为它既有共享的链路又有共享的节点, 因此资源是共享的。并且同等的网络分布密度下, 相关路由的搜索要容易的多, 这是因为不相关路由的搜索其约束性要强的多。但是正是因为相关路由有共享的节点或者链路, 其容错能力就差很多。在上述三种路由类型中, 节点不相关路由的容错能力最强, 链路不相关路由的容错能力次之, 相关路由的容错能力最差。在链路不相关多路径路由中, 如果共享的节点失效, 那么该节点所连接的所有路径便都失败了, 而节点不相关路由由于链路的独立性, 则不会产生这种连锁反应。一般在网络分布密度相对较大的情况下, 采用节点不相关多路径路由; 但在节点密度相对稀疏的网络环境中, 会采用链路不相关多路径路由。一般的话, 相关多路路由是不宜采用的。

2.4.3 多路径路由面临的主要问题

尽管无线传感器网络的多路径路由研究取得了一定的进展。但是, 还有很多问题有待于进一步探索和研究。主要有:

■ 数据的并行传输问题

如何充分利用多路径并行传输, 提高网络资源的利用率和信息传输流量仍有

很多问题待进一步研究。

■ 路由协议适应传感器网络环境问题

传感器网络的应用环境多种多样。应用环境的各不相同，造成路由协议也会有所不同。如何适应不同的应用环境，正引起关注。

■ 传输可靠性问题

备选多路径涉及路径的替换，这就存在主路径失效后备用路径是否有效的问题，必须提高有效的路径维护策略，以保证备用路径的有效性，提高多路径并行传输的可靠性。并行多路径涉及数据分组的分配和可能的分割和重组，这就存在分割优化问题。多路径并行传输还会引入分组传输乱序问题，必须进行差错控制和顺序控制，提高多路径并行传输的可靠性。

■ 路由可扩展性问题

随着应用的发展，传感器网络规模不断扩大。如何提高路由的可扩展性是多路径路由研究的又一重要问题。

■ 能量控制问题

传感器网络节点在硬件资源上受到限制，如何节约能量、减少开销并延长网络生存时间是传感器网络中的一个重要研究问题。

总之，传感器网络多路径路由协议应考虑如下因素：分布式运行、无环路由、按需进行路由操作、节能策略、可靠性、可扩展性等。

2.5 小结

本章首先概述了无线传感器网络的体系结构，阐述了无线传感器网络的特点，分类介绍了传感器网络的两类路由协议：平面型和层次型，通过分析与比较两种类型各自的优缺点，得出基于层次拓扑结构的层次型路由协议能够较好的符合传感器网络能量有效性的要求，适合大规模传感器网络发展的需要，然后介绍了无线传感器网络中的数据可靠传输的三种机制，指出多路径路由机制能够较好的适用于传感器网络，最后总结了多路径路由机制的相关概念、分类及面临的问题。

第三章 具有能量意识的层次拓扑结构生成算法

无线传感器网络的拓扑结构是设计和组建传感器网络的基础。布置在自然环境中且无人看守的大规模无线传感器网络,一般都采用层次型拓扑结构,目的是均衡网络中传感器节点的能量消耗,延长网络的生存时间。本章在分析层次拓扑结构对路由协议影响的基础上,提出了一种能效较高的无线传感器网络的层次拓扑结构生成算法。

3.1 层次拓扑结构对路由协议的影响

无线传感器网络的拓扑结构是设计和组建传感器网络的基础。在无线传感器网络中,网络的层次拓扑结构有着十分重要的意义,主要表现在以下几个方面:

■ 影响整个网络的生存时间

无线传感器网络的节点一般采用电池供电,节省能量是网络设计主要考虑的问题之一。网络层次拓扑结构的一个重要目标就是在保证网络连通性和覆盖度的情况下,尽量合理高效地使用网络节点能量,延长整个网络的生存时间。

■ 减少节点间通信干扰,提高网络通信效率

传感器网络中节点通常密集部署,如果每个节点都以大功率进行通信,会加剧节点之间的干扰,降低通信效率,并造成节点能量的浪费。另一方面,如果选择太小的发射功率,会影响网络的连通性。所以,层次拓扑结构控制中的功率控制技术是解决这个矛盾的重要途径之一。

■ 为路由协议提供基础

在传感器网络中,只有活动的节点才能够进行数据转发,而层次拓扑结构控制可以确定由哪些节点作为转发节点,同时确定节点之间的邻居关系。

■ 影响数据融合

传感器网络中的数据融合指传感器节点将采集的数据发送给骨干节点,骨干节点进行数据融合,并把融合结果发送给数据收集节点。而骨干节点的选择是层次结构控制的一项重要内容。

■ 弥补节点失效的影响

传感器节点可能部署在恶劣环境中,在军事应用中甚至部署在敌方区域内,所以很容易受到破坏而失效。这就要求网络层次拓扑结构具有鲁棒性以适应这种情况。

3.2 网络模型与拓扑结构参数

本文考虑了从数据源节点广播路径请求分组以及在源节点和汇聚节点之间建立路径并传输数据的整个网络运行过程,提出了一个具有能量意识的层次型拓扑结构生成算法,该算法是一种基于传感器节点 ID 编号值的分布式、具有能量意识的自协商成簇算法,通过执行该算法,可以使布置在观测区域中的传感器节点形成以簇为单位的层次拓扑结构,并为每个簇选择一个簇首。下一章介绍的多路径路由的建立是在此层次拓扑结构之上的。在本文提出的路由算法中,以每一次层次拓扑结构的建立过程为界限,将其称为“一轮(Round)”。每一轮分为两个阶段:第一阶段是层次拓扑结构的形成阶段,第二阶段是路径建立与数据传输阶段。本节详细介绍第一阶段层次拓扑结构生成算法,第二阶段在下一章中进行说明。

为了便于对簇内成员节点的管理,层次拓扑结构要求簇首节点与普通成员节点之间的通信链路是双向可达的,因此分簇算法一般不考虑网络中的单向链路,而假定网络是双向连通的。此时,无线传感器网络的拓扑结构的定义描述如下:

可以定义传感器网络的拓扑结构为无向图 $G=(V, E)$, 其中, V 代表网络中的传感器节点集合, E 是网络中的传感器节点之间所有通信链路的集合, $\forall u, v \in V$, 如果传感器节点 u, v 之间的距离小于最大通信距离 r , 则它们之间存在一条双向通信链路 $(u, v) \in E$ 。

考虑到网络模型的定义会直接影响到层次拓扑结构生成算法的性能,我们分别在节点移动性、节点位置信息、节点间距离、节点剩余能量和节点 ID 编号值五个方面给出了定义,具体如下:

■ 节点移动性

传感器节点是类似静止的,即节点在布置完成后基本保持静止,这在传感器网络的大多数应用中都是适用的。

■ 节点位置信息

传感器节点不需要了解自己的位置信息,因此,节点不需要配备 GPS 设备,这样可以提高传感器网络的通用性。

■ 节点间距离

层次拓扑结构生成过程中,节点应该尽可能加入离自己最近的簇,这样有利于提高簇的稳定性。

■ 节点剩余能量

在基于层次结构的路由协议中,簇首要作为路由节点转发分组,因此簇首

节点的能量消耗通常要大于成员节点。为了避免由于簇首节点的能量过度消耗而提前失效,层次拓扑生成算法必须能够对簇首节点进行轮换,使得网络中的各个节点依次充当簇首节点,均衡消耗各个节点的能量,延长网络的生存时间。测量节点的能量消耗有两种方式:一种是累计发送和接收的数据量并折合成消耗的能量,一种是直接测量节点的剩余能量。第一种方法较为方便,网络接口一般都带有计数器,能够对接收和发送的分组数量自动进行统计;第二种方法更加准确,不仅综合考虑了发送接收以及载波侦听消耗的能量,并且适用于初始电源能量不同的异构节点,因此本文采用第二种方式。所有传感器节点在部署之后,是在无人监管的状态下工作的,节点的电池无法更换。

■ 节点 ID 编号

网络中的每个传感器节点都有一个唯一的 ID 编号作为节点标识,利用节点 ID 可以保证簇首选择的唯一性。

3.3 具有能量意识的层次拓扑结构生成算法

3.3.1 主要分组结构定义

在层次拓扑结构生成阶段,网络中的节点通过周期性的广播 Hello 控制分组 Hello Packet 用以获得生成层次结构所必需的邻居节点信息,Hello 分组的格式以及各个字段的意义如下图 3.1 所示:

Node ID _u Number
CHNode ID _u Number
ER _u

图 3.1 Hello 分组格式示意图

其中:

- Node ID_u Number: 表示节点 u 的 ID 编号值;
- CHNode ID_u Number: 表示节点 u 所在簇的簇首节点 ID 编号,如果未加入任何簇则为 NULL;
- ER_u: 表示节点 u 的电源剩余能量。

根据 Node ID_u Number 和 CHNode ID_u Number 字段的的不同,我们可以将 Hello 控制分组分为三类:

1、邻居发现分组 Neighbor Detect Packet

该类分组的 Node ID_u Number 和 CHNode ID_u Number 字段需满足下列条件:

$$NodeID_u \neq NULL \quad \text{and} \quad CHNodeID_u = NULL \quad (3-1)$$

分组格式如下图 3.2 所示:

Node ID _u Number	$\neq NULL$
CHNode ID _u Number	$= NULL$
ER _u

图 3.2 邻居发现分组格式示意图

2、簇首声明分组 CHNode Report Packet

该类分组的 Node ID_u Number 和 CHNode ID_u Number 字段需要满足下列条件:

$$NodeID_u = CHNodeID_u \neq NULL \quad (3-2)$$

分组格式如下图 3.3 所示:

Node ID _u Number	$= CHNode ID_u$
CHNode ID _u Number	$\neq NULL$
ER _u

图 3.3 簇首声明分组格式示意图

3、加入分簇分组 Participation Cluster Packet

该类分组的 Node ID_u Number 和 CHNode ID_u Number 字段需要满足下列条件:

$$NodeID_u \neq NULL, CHNodeID_u = NULL \quad \text{and} \quad NodeID_u \neq CHNodeID_u \quad (3-3)$$

分组格式如下图 3.4 所示:

Node ID _u Number	$\neq NULL \text{ 且 } \neq CHNodeID_u$
CHNode ID _u Number	$= NULL$
ER _u

图 3.4 加入分簇分组格式示意图

节点通过周期性的广播 Hello 分组交换信息, 建立一个邻居节点信息表 NT_u (Neighbor Table), 表中记录的内容与 Hello 分组的内容相同。节点 *u* 的邻居节点信息表 NT_u 的格式如下图 3.5 所示。

Node ID _u Number	CHNode ID _u Number	ER _u
...
...

图 3.5 邻居节点信息表格式示意图

3.3.2 层次拓扑生成算法

层次拓扑结构生成过程分为网络初始化、簇结构建立、簇首节点重选三个步骤，具体如下：

1、网络初始化

网络初始化时，每个节点的 CHNode ID 都为空，节点每隔时间 T 周期性的广播 Hello 分组，并等待一段时间 $T_w(T_w > T)$ 使得节点获得完整的邻居节点信息表 NT_u 后，利用下面的簇首节点选择方法判断自己是否可以成为簇首节点，如果自己是簇首节点，则设置自己的 CHNode ID 并广播一个簇首声明分组通知所有的邻居节点。

根据 NT_u 中的信息，节点判断自己能否成为簇首节点 CHNode 的方法描述如下：

```

CHNode_Elect (  $NT_u$  )
{
    if( CHNode IDu != NULL) return NULL;
     $TCH_u = \{z | (z \in NT_u) \wedge CHNodeID_z = NULL\}$ ;
     $CHNode = \{x | \forall y \in TCH_u (ER_x > ER_y \vee (ER_x = ER_y \wedge NodeID_x < NodeID_y))\}$ ;
    return CHNode ID;
}

```

节点首先判断自己是否已经加入了某个簇，如果是则返回空，否则在 NT_u 中筛选出还未加入簇的节点得到集合 TCH_u ，然后在该集合中选择剩余能量最多的节点作为簇首节点，如果节点的剩余能量相同，则选择 ID 编号值最小的节点作为簇首节点，最后返回簇首节点 ID 编号 CHNode ID。

2、簇结构建立

节点收到一个 Hello 分组后，首先更新邻居节点信息表，然后根据 Hello 分组中的内容判断该分组的类型：

1. 如果是簇首声明分组，并且自己没有加入任何簇，则加入该簇并广播一个加入分簇分组；如果自己已经加入某个簇则根据监听到的节点信号功率，选择离自己更近的作为簇首节点；如果自己也是簇首则进行簇首竞争，剩余能量较小的节点自动放弃簇首资格，加入能量较大的簇首所在的簇，并广播

一个加入分簇分组。

II. 如果收到的是加入分簇分组，并且自己未加入任何簇，则重新判断自己能否成为簇首节点。

III. 如果是一个重新分簇分组，则将以该节点作为簇首的所有邻居节点的 CHNode ID 置为空，然后重新进行簇首节点的选举。

经过上述过程，每个节点都会加入某个簇或者成为簇首节点。簇首节点使用基于时分复用的通信方式 TDMA，为簇中的每个成员节点分配通信时间槽。在稳定工作阶段，簇内的成员节点将采集到的监测数据传输给簇首节点，进行必要的融合处理之后，通过其他簇首节点转发至汇聚节点。为了降低能量消耗，稳定的数据传输阶段时间要长于形成簇阶段的时间。持续一段时间以后，整个网络进入下一工作周期，重新选择簇首。

3、簇首节点重选

当网络一个运行周期结束后，所有节点进行簇首节点重选，簇首节点放弃簇首地位，将自己的 CHNode ID 设置为空，并广播一个重新分簇分组，这样就保证了簇首节点轮流由剩余能量最大的节点担当，均衡了因作为簇首节点而带来的能量消耗，延长了网络的生存时间。

层次拓扑生成算法处理流程图如下图 3.6 所示：

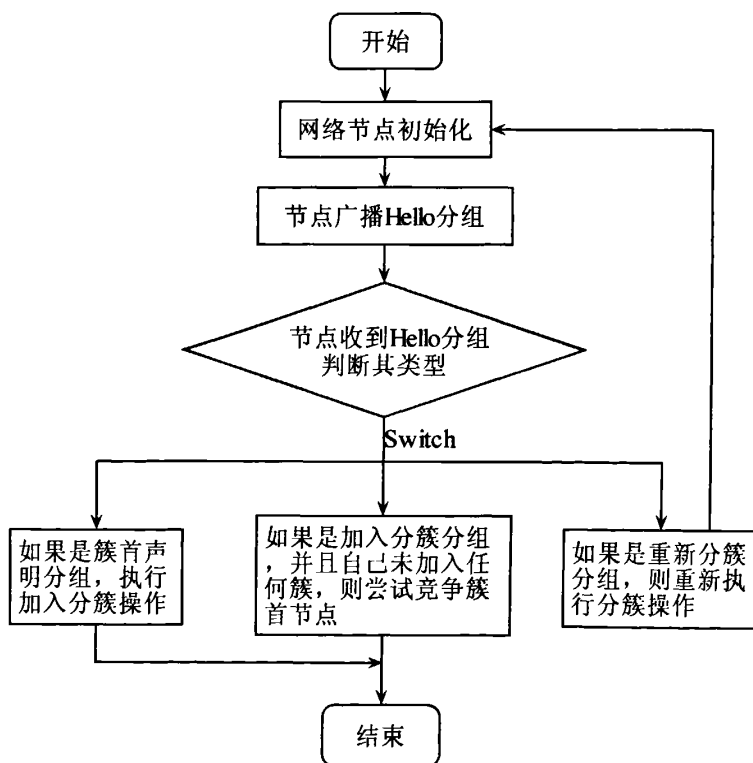


图 3.6 算法处理流程图

3.4 小结

本章首先阐述了层次拓扑结构对路由协议的影响，然后定义了一个网络模型，该模型概括了在无线传感器网络中研究传输方法、路由机制等问题的一般性要求，具有通用性。随后提出了一种能量有效的层次拓扑生成算法，该方法同时考虑了节点 ID 编号值和节点的剩余能量因素，提高了能量有效性，延长了网络生存时间。

第四章 基于层次结构的节点不相关可靠多路径路由算法

数据收集是无线传感器网络的重要任务,可靠的数据传输机制能够更有效的获取监测数据。

本章考虑了从数据源节点广播路径请求分组以及在源节点和汇聚节点之间建立路径并传输数据的整个网络运行过程,在借鉴以往路由算法思想 AODV^[35]的基础上,提出了一种基于层次拓扑结构的数据源-汇聚节点对之间的多路径建立算法,该算法利用传感器网络节点高度冗余的特点,可以在每个拓扑结构生成阶段后,为数据源寻找一系列可到达汇聚节点的节点不相关的路径。节点不相关路径降低了路径之间的关联性,避免了连锁失效的情况,提高了路径的可靠性。整个数据可靠传输机制利用冗余多路径确保数据传输的可靠性,利用能量有效的层次拓扑结构确保该可靠传输方法的能量有效性。

4.1 多路径数据可靠传输方法设计准则

由于无线传感器网络资源有限且与应用高度相关,研究人员采用多种策略来设计传输方法及路由协议,其中好的方法及协议必须具有以下特点^[36]:针对能量高度受限的特点,高效利用能量几乎是设计的第一策略;针对流量特征、通信耗能等特点,采用通信量负载平衡技术;针对通信耗能、节点有合作关系、数据有相关性、节点能量有限等特点,采用数据聚合、过滤等技术;针对节点少移动的特点,不维护其移动性;针对网络相对封闭、不提供计算等特点,只在汇聚节点 Sink 考虑与其他网络互联;针对网络节点不常编址的特点,采用基于数据或基于位置的通信机制;针对节点易失效的特点,采用多路径机制。通过对当前的各种可靠传输方法进行分析与总结,可以看出将来传感器网络路由协议采用的某些研究策略与发展趋势:

■ 容错性

由于传感器节点容易发生故障,因此应尽量利用节点易获得的网络信息计算路由,以确保在路由出现故障时能够尽快得到恢复;并可采用多路径机制传输数据来提高数据传输的可靠性。

■ 保持通信量负载平衡

通过更加灵活地使用路由策略让各个节点分担数据传输,平衡节点的剩余能量,提高整个网络的生存时间。例如,可在层次型路由中采用动态簇头;在路由选择中采用随机路由而非稳定路由;在路径选择中考虑节点的剩余能量。

■ 减少通信量以节约能量

由于传感器网络中的数据通信最耗费能量,因此在协议中应尽量减少数据通信量。例如,可在数据查询或者数据上报中采用某种过滤机制,抑制节点上传不必要的数 据;采用数据聚合机制,在数据传输到汇聚节点前就完成可能的数据计算。

■ 安全机制

由于传感器网络具有的自身特点,其路由协议极易受到安全威胁,尤其是在军事应用中。目前的路由协议很少考虑安全问题,因此在一些应用中必须考虑设计具有安全机制的路由协议。

4.2 多路径路由建立算法设计

4.2.1 主要分组及路由表结构定义

1、路由请求分组 Path Request Packet, Preq_Pack

Preq_Pack 分组用于路由请求,主要由以下一些字段组成:

- Type
- Preq_Pack Sequence Number
- Destination ID Number
- Originator ID Number
- Repeater ID Number
- Firsthop ID Number
- Hop Count

其中,“Type”用于标识分组的类型;“Preq_Pack Sequence Number”和“Originator ID Number”用于唯一标识一个路由请求,Preq_Pack 分组序列号字段的值要在源节点维护的最新值的基础上加 1,同时“Preq_Pack Sequence Number”还用于在广播路由请求的过程中保证反向路由的有效性,该字段值为源节点维护的当前序列号值加 1;“Firsthop ID Number”用于记录首次转发 Preq_Pack 分组的节点 ID 号,也即转发 Preq_Pack 分组的源节点的一跳邻居节点的 ID 号,此字段只能由源节点的邻居节点设置值,在自此之后的转发过程中该字段值保持不变,这是保证路径之间节点不相关的关键字段;“Hop Count”用于计算 Preq_Pack 分组经过的节点数。

2、路由回复分组 Path Reply Packet, Prep_Pack

Prep_Pack 分组用于路由回复,主要由以下一些字段组成:

- Type
- Prep_Pack Sequence Number
- Destination ID Number
- Originator ID Number
- Repeater ID Number
- Firsthop ID Number
- Hop Count

其中,“Originator ID Number”用于记录路由回复分组 Prep_Pack 的源节点,即路由请求分组 Preq_Pack 的目的节点 ID 号;“Destination ID Number”用于记录路由回复分组 Prep_Pack 的目的节点 ID 号,即路由请求分组 Preq_Pack 的源节点 ID 号;“Prep_Pack Sequence Number”和“Originator ID Number”用于唯一标识一个路由回复, Prep_Pack 分组序列号字段的值要在源节点维护的最新值的基础上加 1,同时“Prep_Pack Sequence Number”还用于在广播路由请求的过程中保证反向路由的有效性,该字段值为源节点维护的当前序列号值加 1;“Hop Count”用于计算 Prep_Pack 分组经历过的跳数。

3、节点路由表

节点路由表主要记录以下字段:

- Firsthop ID Number
- Destination ID Number
- Preq/Prep_Pack Sequence Number
- Repeater ID Number
- Hop Count

节点路由表示意图如下图 4.1 所示:

	Firsthop ID	Destination ID	Rreq/Rrep Sequence	Hop Count	Repeater ID
反向路由

正向路由

图 4.1 节点路由表示意图

4.2.2 路由建立

由于 Ad Hoc 网络与无线传感器网络存在很多相似性,本文在路由建立阶段借鉴了 Ad Hoc 网络中的单路径路由建立算法 AODV 的一些思想。在网络中建立

多条路径,很可能存在环路问题,通过借鉴 AODV 中的目的序列号的思想可以保证无环路由。考虑到传感器网络中的节点具有冗余性,且无线通信具有广播特性,对 AODV 进行扩展,在路由请求分组中添加源节点的邻居节点 ID 字段,该字段只能由源节点的邻居节点写入自己的 ID 值,分组在此后的转发中该字段值不再变动,已处理过路由请求分组的非源节点邻居的中间节点对收到的具有不同邻居节点 ID 值的路由请求分组采取简单的丢弃处理,目的节点在收到路由请求分组时,只需要按照转发路由请求分组过程中建立的反向路由由发送至源节点,这样可以在数据源节点和汇聚节点之间建立多条节点不相关的路径。

当数据源节点需要与目的节点也即汇聚节点 Sink 进行通信时,它首先在本节点维护的路由表中查找是否有到达该目的节点的路由。若路由表中已包含了到达汇聚节点的有效路由,则立即使用此路由向汇聚节点发送监测数据分组,否则它将向所有邻居节点广播 Preq_Pack 分组,用以发起一个路由发现过程来建立多条节点不相关的到达汇聚节点的可用路由。

1、生成、处理并转发路由请求

Step 1: 数据源节点生成路由请求分组 Preq_Pack, 设置分组中各字段的值,然后继续下一步 Step 2;

路由请求分组 Preq_Pack 中的字段值设置如下:“Originator ID Number”设置为该数据源节点的 ID 编号值;“Preq_Pack Sequence Number”字段值要在源节点记录的序列号值的基础上加 1;“Firsthop ID Number”字段值设置为空,这是保证路径之间节点不相关的关键字段;“Hop Count”用于计算 Preq_Pack 分组经历的跳数,设置为 0。

Step 2: 节点广播该 Preq_Pack 分组,然后继续下一步 Step 3;

Step 3: 收到 Preq_Pack 分组的节点,首先查看该 Preq_Pack 分组中 Firsthop ID Number 字段值是否为空,是,转 Step 4; 否,则转 Step 5;

Step 4: 如果收到的 Preq_Pack 分组中, Firsthop ID Number 字段值为空,则该节点即为数据源节点的一跳邻居节点,将 Firsthop ID Number 字段值设置为本节点的 ID 编号值,然后转 Step 8;

Step 5: 如果收到的 Preq_Pack 分组中, Firsthop ID Number 字段值不为空,判断是否处理过相同 Firsthop ID Number 字段值的 Preq_Pack 分组,可以通过与节点路由表中记录的 Firsthop ID Number 字段值比较得到。这是用来保证路径不相关的关键。是,转 Step 6; 否,则转 Step 13;

Step 6: 判断是否需要进行路由更新,需要,转 Step 7, 否则,转 Step 13;

满足下列条件之一需要对现有路由表项进行更新: Preq_Pack 分组中携带的路由请求分组序列号值 Preq_Pack Sequence Number 比节点路由表中存储的值

大,或二者序列号值相等,但 **Preq_Pack** 中的跳数值比节点路由表记录的跳数值小。

Step 7: 更新节点中记录的路由信息,然后转 **Step 8**;

Step 8: 建立一条通往数据源节点的反向路由:

利用 **Preq_Pack** 分组中的“**Originator ID Number**”和“**Preq_Pack Sequence Number**”建立一条通往数据源节点的反向路由,这样该节点在此后处理返回给源节点的路由回复分组 **Prep_Pack** 时,就可直接使用该反向路由进行转发。反向路由的表项主要有如下设置:“目的节点 ID 号(**Destination ID Number**)”设为 **Preq_Pack** 分组中的源节点 ID 号(**Originator ID Number**);“**Preq_Pack/ Prep_Pack Sequence Number**”设为 **Preq_Pack** 分组中 **Preq_Pack** 分组序列号 (**Preq_Pack Sequence Number**);“下一跳节点 (**Next Hop**)”设为广播该 **Preq_Pack** 分组的邻居节点(**Repeater ID Number**);“跳数(**Hop Count**)”设为 **Preq_Pack** 中的跳数值,然后转 **Step 9**;

Step 9: 判断当前节点是否为目的节点,如果当前节点是中间节点,则转 **Step 10**;如果当前节点是目的节点,则转 **Step 11**;

Step 10: 当前节点是中间节点,判断当前节点的路由表中是否存在到达目的节点的有效路由,存在,转 **Step 11**;否则,转 **Step 12**;

Step 11: 当前节点是目的节点,或者当前节点的路由表中存在到达目的节点的有效路由,则生成一个路由回复分组 **Prep_Pack**,并将 **Preq_Pack** 的 **Destination ID Number** 和 **Originator ID Number** 复制到 **Prep_Pack** 分组中的相应区域,并沿建立好的反向路由向数据源节点发送 **Prep_Pack** 分组,然后转 **Step 13**;

目的节点生成并回复的 **Prep_Pack** 分组主要字段设置如下:“源节点的一跳邻居节点 ID 号(**Firsthop ID Number**)”设置为 **Preq_Pack** 分组中 **Firsthop ID Number** 字段的值,“目的节点 ID 号(**Destination ID Number**)”设置为目的节点 ID 号,“**Prep_Pack** 分组序列号(**Prep_Pack Sequence Number**)”设置为目的节点维护的 **Prep_Pack Sequence**,“源节点 ID 号(**Originator ID Number**)”设置为产生 **Preq_Pack** 的源节点 ID 号,“跳数(**Hop Count**)”置为 0;

中间节点回复的 **Prep_Pack** 主要字段设置如下:“目的节点 ID 号(**Destination ID Number**)”设为 **Preq_Pack** 分组中 **Destination ID Number** 字段值,“**Prep_Pack** 分组序列号(**Prep_Pack Sequence Number**)”设为该节点路由表中存储的 **Prep_Pack** 分组序列号,“源节点 ID 号(**Originator ID Number**)”设置为产生 **Preq_Pack** 的源节点 ID 号,“跳数(**Hop Count**)”设为该节点记录的到达目的节点的有效路由经历的跳数值;

Step 12: 当前节点是中间节点且不存在到达目的节点的有效路由,则继续转

发该 Preq_Pack 分组, 在转发之前需要更新节点路由表(如, 将路由表中的 Firsthop 设置成 Preq_Pack 分组中 Firsthop ID Number 字段值), 并对 Preq_Pack 分组的部分字段进行更新(如, 将 Preq_Pack 分组中的 Repeater ID Number 设置成当前节点 ID 号和 Hop Count 字段值加 1), 然后转 Step 2;

Step 13: 当前节点丢弃接收到的 Preq_Pack 分组, 然后转 Step 14;

Step 14: 处理结束。

路由请求分组 Preq_Pack 的处理流程如图 4.2 所示。

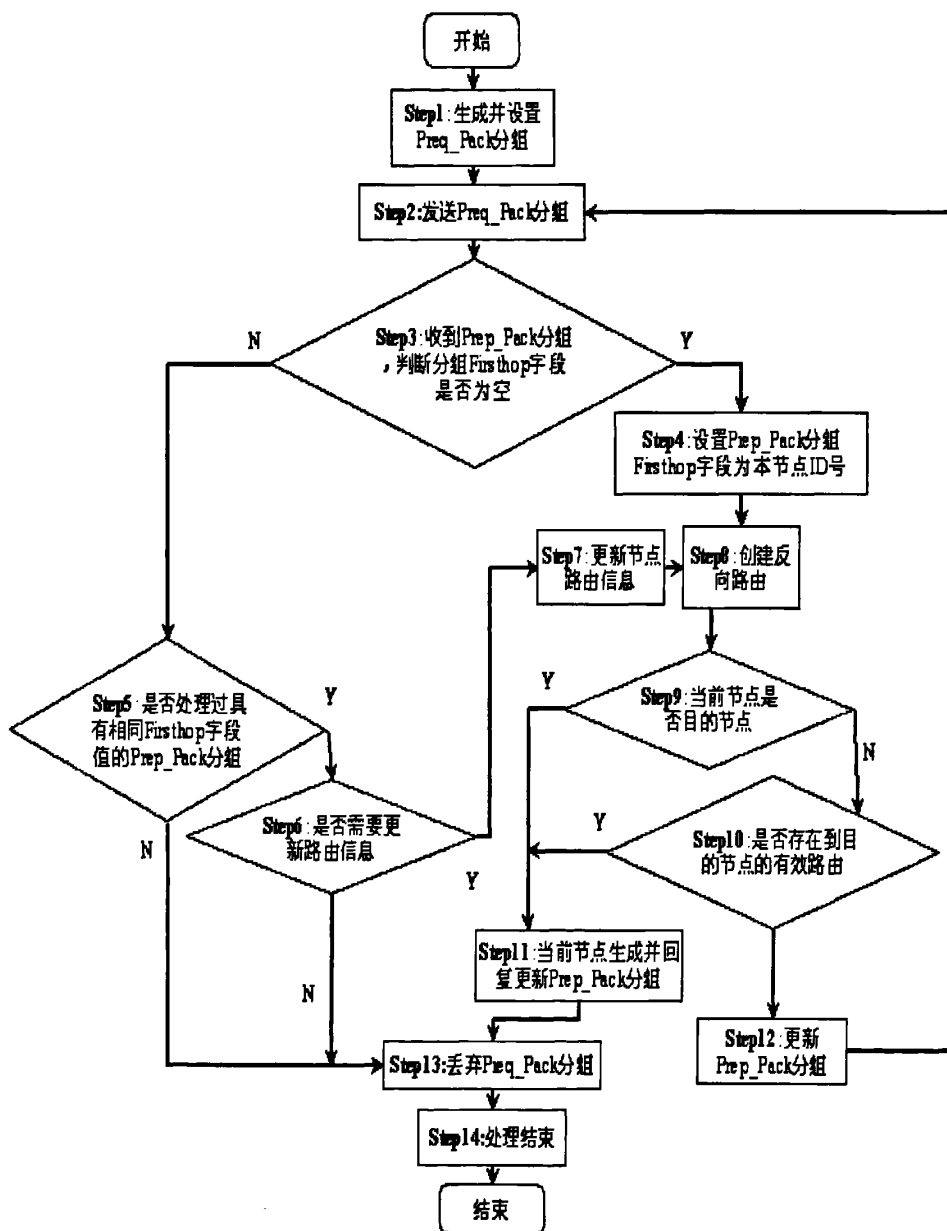


图 4.2 Preq_Pack 分组处理流程图

2、生成、处理并转发路由回复

Step 1: 目的节点或存在到达目的节点有效路由的中间节点生成 Prep_Pack 分组, 设置分组中各字段的值并广播该分组, 然后继续下一步 Step 2;

目的节点生成并回复的 Prep_Pack 分组主要字段设置如下: “源节点的一跳邻居节点 ID 号(Firsthop ID Number)” 设置为 Preq_Pack 分组中 Firsthop ID Number 字段的值, “目的节点 ID 号(Destination ID Number)” 设置为目的节点 ID 号, “Prep_Pack 分组序列号(Prep_Pack Sequence Number)” 设置为目的节点维护的 Prep_Pack Sequence, “源节点 ID 号(Originator ID Number)” 设置为产生 Preq_Pack 的源节点 ID 号, “跳数(Hop Count)” 置为 0。

中间节点回复的 Prep_Pack 主要字段设置如下: “目的节点 ID 号(Destination ID Number)” 设为 Preq_Pack 分组中 Destination ID Number 字段值, “Prep_Pack 分组序列号(Prep_Pack Sequence Number)” 设为该节点路由表项中存储的 Prep_Pack 分组序列号, “源节点 ID 号(Originator ID Number)” 设置为产生 Preq_Pack 的源节点 ID 号, “跳数(Hop Count)” 设为该节点记录的到达目的节点的有效路由经历的跳数值。

Step 2: 节点广播该 Prep_Pack 分组, 然后继续下一步 Step 3;

Step 3: 收到 Prep_Pack 分组的节点, 首先比较分组中 Firsthop 字段值和节点路由表中 Firsthop 值是否相等, 是, 转 Step 4; 否, 则转 Step 11;

Step 4: 判断当前节点是否已有到达上一跳节点的正向路由, 是, 转 Step 5; 否, 则转 Step 7;

Step 5: 当前节点已有到达上一跳节点的正向路由, 看节点路由表中该路由是否需要更新, 需要, 转 Step 6; 否则, 转 Step 8;

满足下列条件之一需要对现有路由表项进行更新: Prep_Pack 中的目的序列号大于现有表项的目的序列号; 序列号相同, 但新跳数值比现有表项中的跳数值小。

Step 6: 更新节点中的正向路由信息, 然后转 Step 8;

Step 7: 当前节点路由表中不存在到达上一跳节点的正向路由, 则创建一条到达目的节点的正向路由, 然后转 Step 8;

正向路由的表项主要设置如下: “目的节点 ID 号(Destination ID Number)” 设为 Prep_Pack 分组中的目的节点 ID 编号值; “Prep_Pack 分组序列号(Prep_Pack Sequence Number)” 设置为 Prep_Pack 分组中的 Prep_Pack 分组序列号; “上一跳转发节点(Repeater ID Number)” 设置为转发此 Prep_Pack 分组的上一跳节点; “跳数(Hop Count)” 设置为 Prep_Pack 分组中携带的跳数值。

Step 8: 判断当前节点是否为数据源节点, 这可以通过比较 Prep_Pack 分组

中的 Destination ID Number 字段值和本节点的 ID 编号值得到，是，转 Step 9；否，则转 Step 10；

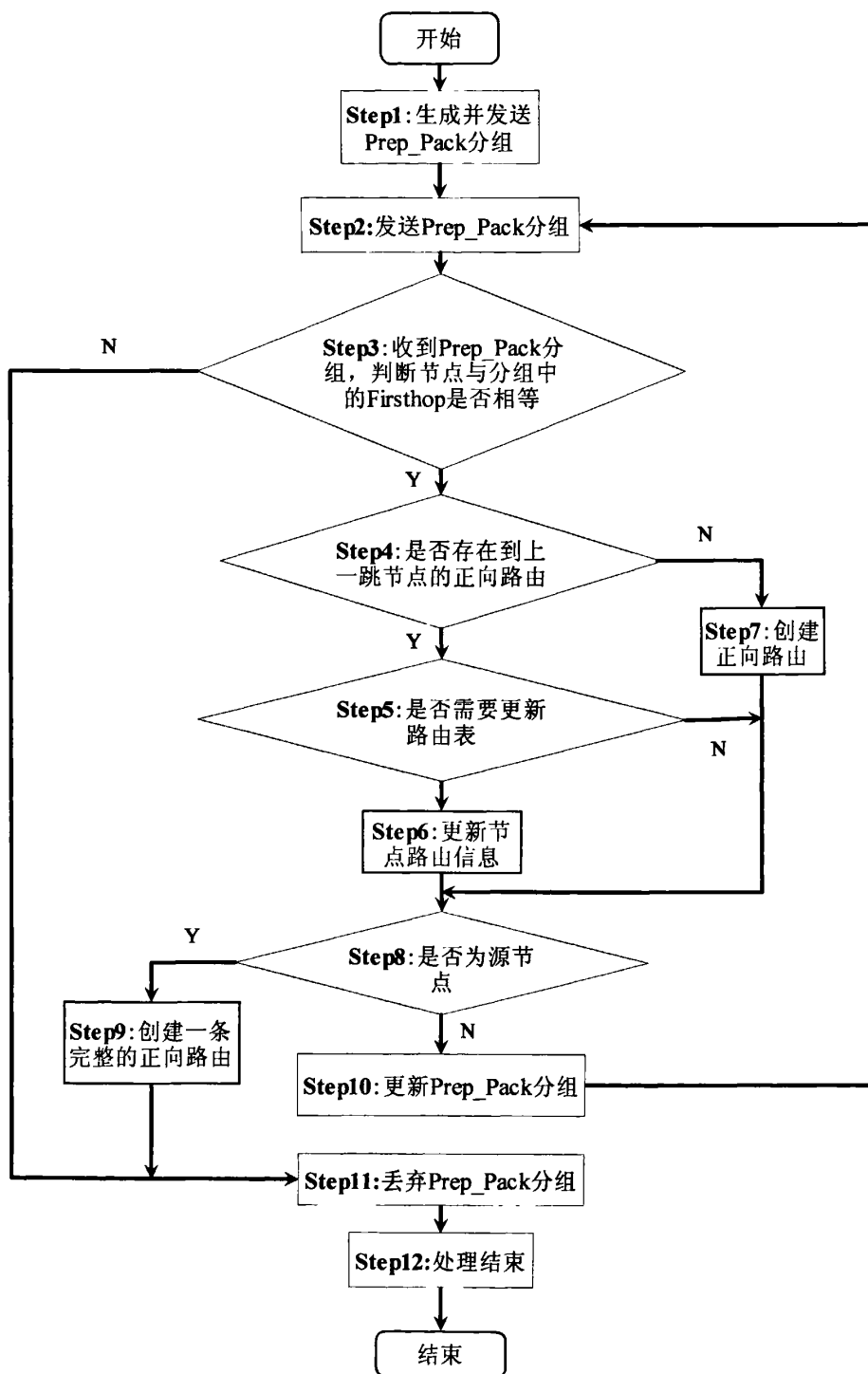


图 4.3 Prep_Pack 分组处理流程图

Step 9: 当前节点是数据源节点，它就可立即使用该正向路由来进行数据分组的发送，然后转 **Step 11**；

Step 10: 当前节点不是数据源节点，则将收到的 **Prep_Pack** 分组更新，然后转 **Step 2**；

路由回复分组 **Prep_Pack** 中字段值的具体更新如下：“**Repeater ID Number**”设置为该节点的 ID 编号值；“**Hop Count**”字段值在原来的基础上做加一运算，其余字段保持不变。

Step 11: 当前节点丢弃接收到的 **Prep_Pack** 分组，然后转 **Step 12**；

Step 12: 处理结束。

路由回复分组 **Prep_Pack** 的处理流程如图 4.3 所示。

4.3 多路径路由维护算法设计

由于无线传感器网络节点硬件资源受限而且工作环境恶劣，具有拓扑变化频繁的特点，网络中任何节点的失效都会导致部分现有链路的断裂，对现有路由造成一定影响。因此传感器网络的路由协议需具有维持路由、修复破损路由、拆除无效路由等机制。

4.3.1 主要分组结构定义

1、路由探查分组 Path Check Packet, Pchk_Pack

Pchk_Pack 分组用于链路探查，主要由以下一些字段组成：

- Type
- Repeater ID Number
- NextHop ID Number
- TTL Hop

其中，“**Type**”用于标识分组的类型；“**Repeater ID Number**”用于记录转发该 **Pchk_Pack** 分组的节点 ID 编号值；“**NextHop ID Number**”用于记录该 **Pchk_Pack** 分组的下一跳节点 ID 编号值；“**TTL Hop**”用于记录 **Pchk_Pack** 分组的生存周期，通常设置为 1。

2、路由重建请求分组 Path Rebuild Requist Packet, Prbdq_Pack

Prbdq_Pack 分组用于链路修复请求，主要由以下一些字段组成：

- Type
- Err_Node ID Number
- Firsthop ID Number

- Originator ID Number
- Repeater ID Number
- TTL Hop

其中,“Type”用于标识分组的类型;“Err_Node ID Number”记录失效链路的下一跳节点 ID 标号值,用于唯一标识一次路由修复;“Firsthop ID Number”用于记录数据源节点的一跳邻居节点 ID 编号值;“Originator ID Number”用于记录发起路由修复的节点 ID 值;“Repeater ID Number”用于记录转发该 Prbdq_Pack 分组的节点 ID 编号值;“TTL Hop”用于记录 Pchk_Pack 分组的生存周期,通常设置为 3。

3、路由重建回复分组 Path Rebuild Reply Packet, Prbdp_Pack

Prbdp_Pack 分组用于链路修复回复,主要由以下一些字段组成:

- Type
- Err_Node ID Number
- Firsthop ID Number
- Destination ID Number
- Repeater ID Number

其中,“Type”、“Err_Node ID Number”、“Firsthop ID Number”和“Repeater ID Number”字段的用途相同于 Prbdq_Pack 分组中的相应字段;“Destination ID Number”用于记录发起路由修复的节点 ID 值。

4、路由失效通知分组 Path Error Inform Packet, Perrifm_Pack

Perrifm_Pack 分组用于路由失效通告,主要由以下一些字段组成:

- Type
- Originator ID Number
- Repeater ID Number

其中,“Originator ID Number”用于记录源节点 ID 值;“Repeater ID Number”用于记录转发节点 ID 值。

4.3.2 路由维护

本章路由协议中,备用路由上的传感器节点通过广播 Pchk_Pack 分组来检查该路由上各段链路的连接情况。将 Pchk_Pack 分组中的“TTL Hop”字段值设为 1,则其就是一个生存周期为 1 跳的特殊分组。由于 TTL Hop 值为 1,因此该 Pchk_Pack 分组只能传播一跳的距离,到达邻居节点后便因 TTL Hop 等于 0 而被丢弃。

节点通过将 **Pchk_Pack** 分组中“**Repeater ID Number**”和“**NextHop ID Number**”字段分别设为本节点 ID 值和本节点所在的正向路由的下一跳节点 ID 值，广播该 **Pchk_Pack** 分组来确保该条链路是可用的。收到 **Pchk_Pack** 分组的节点按上面规则修改该分组中的“**Repeater ID Number**”和“**NextHop ID Number**”字段值，“**TTL Hop**”字段值重置为 1，然后广播。上一跳节点收到 **Pchk_Pack** 分组后比较分组中“**Repeater ID Number**”字段和街道上缓存的“**NextHop ID Number**”字段用以确定链路是可用的。若一段时间后，节点没有收到下一跳节点广播的 **Pchk_Pack** 分组，就认为它与该节点的链路已断裂，于是进行以下的局部链路修复。

节点发现其与某邻居节点间的链路断裂后，则发起局部链路修复。发起修复的节点广播一个 **Prbdq_Pack** 分组，试图寻找到目的节点的其他路由以代替当前破损路由。替代路由存在的情况下，目的节点或某个拥有到达目的节点有效路由的中间节点会返回一个 **Prbdp_Pack** 分组。发起修复的节点收到该 **Prbdp_Pack** 分组后便知道路由修复成功，该节点处缓冲的分组便可被继续发往目的节点。若没有 **Prbdp_Pack** 分组返回，则表示修复失败，此时再进行以下的路由拆除。

发起路由修复的节点在没有收到 **Prbdp_Pack** 分组后，广播 **Perr_Pack** 分组通知该正向路由上的其他上游节点，直到数据源节点。此后，数据源节点则不会使用该条路径传输数据。

4.4 数据传输过程

采用上一章的层次拓扑生成算法生成网络的层次拓扑结构，通过本章的多路径路由建立算法建立多路径路由，随后是稳定的簇内数据传输和簇间数据传输，为了能够使能量耗费最小化，稳定的数据传输阶段时间要长于拓扑结构生成阶段的时间。数据传输的整个过程可以细分为两个阶段：

I. 簇首节点建立 **TDMA** 调度表阶段：在簇首确定本簇所有非簇首成员节点后，簇首创建 **TDMA** 的时间表，为每个成员节点分配一个时间槽并将时间槽的分配结果通告簇内的成员节点，告知每个成员节点“你可以在什么时段发送数据”。在簇形成时，每个簇中包含的节点数可能并不相等，所以每个簇内的时槽大小也并不相等，其大小根据簇内每个普通节点的多少而决定。

II. 数据传输阶段：非簇首成员节点仅在自己的时间槽内才打开数据发送装置发送监测数据，其它时隙可以关闭发送装置以减小该节点的能量消耗。簇首节点应始终开启接收装置，接收所有成员节点发来的数据，之后通过数据融合对数据进行压缩处理，然后再通过路由算法建立的路径将监测数据发往汇聚节点。

4.5 小结

路由的可靠性是任何无线通信网络中最重要的研究热点之一。在已有的数据可靠传输方法中，由于多路径方法可以显著的提高网络在数据传输方面的可靠性，在借鉴以往路由算法思想的基础上，本文给出了一种建立在层次拓扑结构之上的，在数据源节点与汇聚节点之间建立多条路径的算法，该算法考虑了无线传感器网络及网络节点高度冗余的特点，能够快速地为源节点-目的节点之间建立多条可供数据传输的节点不相关路径，降低了路径之间的关联性，周期性的簇首重新选举，在确保数据传输可靠性的基础上，均衡了网络中节点的能量消耗，延长了网络的生存时间。

第五章 仿真实验与结果分析

网络仿真工具中, 最具代表性的是 OPNET^[37], QualNet^[38], OMNET++^[39] 和 NS-2^[40]。其中, OPNET 涵盖网络设计、仿真、分析和优化等领域, 提供庞杂的开发库, 功能强大, 是市场上最成功的商业网络仿真软件之一。QualNet 基于 GloMoSim^[41], 包括网络设计、仿真、分析和动画等功能模块, 与 OPNET 相比, 易于掌握, 仿真效率较高。OMNET++ 是基于组件的开源网络仿真框架, 具有较友好的图形用户界面。NS-2 是基于分裂对象模型的开源网络仿真工具, 能较快反映当前网络仿真需求的热点, 提供了较完整的协议栈和大量已有协议的实现, 易于进行协议间的横向性能比较, 其仿真结果得到学术界认可, 在国内外高校科研院所得到广泛应用。

但是以上仿真工具目前都不直接提供对传感器网络仿真的支持, 需做相应扩展。考虑到 TOSSIM^[42, 43] 仿真软件直接提供对传感器网络仿真的支持, 因此本文在实验阶段采用了 TOSSIM 仿真工具。

5.1 实验仿真平台——TOSSIM

TinyOS^[44] 是美国加州大学 Berkeley 分校开发的一个开源的嵌入式操作系统, 它是专门针对现代无线传感器节点的特性及应用需求设计的。TinyOS 是一个基于组件(Component-Based)的操作系统, 结构简单, 能够快速实现各种应用。TinyOS 的程序采用模块化设计, 程序核心都很小(一般来说核心代码和数据大概在 400Bytes 左右), 克服了传感器节点存储资源受的不足, 这使得 TinyOS 能够有效的运行在无线传感器网络上并去执行相应的应用管理任务。

nesC 语言^[44] 是由 C 语言扩展而来的, 具有类似 C 语言的语法, 意在把组件化/模块化思想和基于事件驱动的执行模型结合起来。nesC 提供了比较完善的组件机制和事件驱动机制, 从而降低了面向传感器网络的操作系统和应用程序实现的复杂度。目前 nesC 已被用作 Berkeley 大学 TinyOS 的编程语言。现有的一些重要传感器网络应用程序使用 nesC 实现。

TOSSIM 是一个可以在 PC 机上运行的 TinyOS 的模拟器。TOSSIM 能够同时模拟成百上千个节点, 并捕获节点的网络行为。TOSSIM 提供运行时调试输出信息, 允许用户从不同的角度分析和观察程序的执行过程。TOSSIM 的特点是建立了 TinyOS 的底层部分硬件的软件抽象, 并增加了仿真必须的事件模型和外部通信机制。TOSSIM 已经使用在 TinyOS 节点的性能分析中, 并通过仿真运行发

现解决了 TinyOS 系统内部的部分缺陷。TOSSIM 是高度面向 TinyOS 系统的仿真工具, 可以进行比特级的仿真, 而这点是目前大部分仿真工具所不具备的。TinyViz 是基于 Java 的 TOSSIM 可视化工具, 允许仿真的可视、可控及可分析, 展示了 TOSSIM 的通信服务能力。TinyViz 如图 5.1 所示。

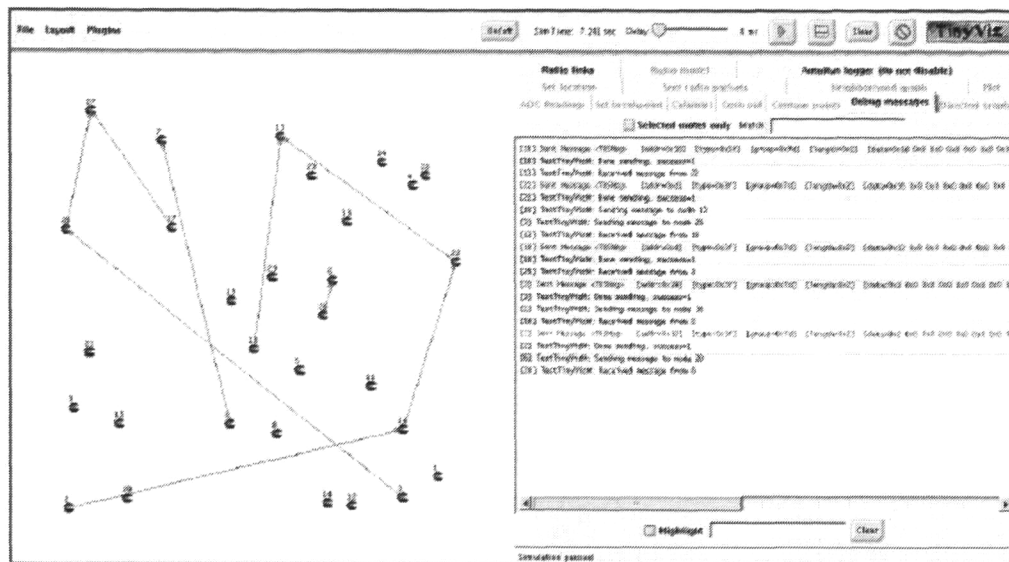


图 5.1 可视化工具 TinyViz 的示意图

5.2 仿真实现

在仿真工具 TOSSIM 中模拟了本文提出的路由协议, 下面详细介绍本文路由协议的仿真实现过程。

5.2.1 仿真运行环境

TinyOS 的安装运行文件从 <http://www.tinyos.net> 处获得, 软件包括两个安装文件:

■ TinyOS 自动安装程序版本: TinyOS Version 2.0

■ JDK 版本: JDK Version 1.4.1_02

安装完成后在 cygwin shell 窗口中的命令行中, 键入 `toscheck` 命令能够检查软件安装是否成功。

运行 TinyOS 的 PC 配置如下:

■ CPU: Pentium (R) 4 CPU 2.40GHz

■ 内存: 512MB

■ 操作系统版本: Microsoft Windows XP Professional 2002 Service Pack 2

5.2.2 组件及模块介绍

LEPS(Link Estimation and Parent Selection)协议是 TinyOS 的多跳路由协议，适用于数据收集应用。本文协议在仿真实现上借鉴了 TinyOS 中的 LEPS 接口连接实现代码，该代码较好地定义了 TOSSIM 的各项接口，我们以这些接口为基础，设计仿真环境来测试本程序是否能完成预定动作。在实现过程中主要涉及的 TinyOS 组件包括：

- **MultiHopEngineM**：该模块是路由协议的引擎，通过 **RouteControl** 和 **RouteSelect** 两个接口连接到路由协议的实现模块上，并和路由协议实现模块交互得到当前的路由状况。**MultiHopEngineM** 只是通过两个接口和路由协议模块交互，模块的实现独立于任何路由协议实现，这种实现方式非常有利于基于 TinyOS 平台的第三方路由协议的开发。
- **MultiHopLEPSM**：提供链路质量评估和路径选择机制，它监控所有从 **Snoop** 端口接收到的拥塞控制消息以及可能从单跳邻居处直接接收到的单跳路由更新信息。该模块管理着最近的可用邻居并且根据最短路径法则选取下一跳节点。
- **MultiHopRoute**：用于连接 **MultiHopEngineM**，**MultiHopLEPSM** 的组件，通过 **Receive**，**Send**，**Intercept** 以及 **Snoop** 接口来实现连接。

各个模块连接示意图如下图 5.2 所示。

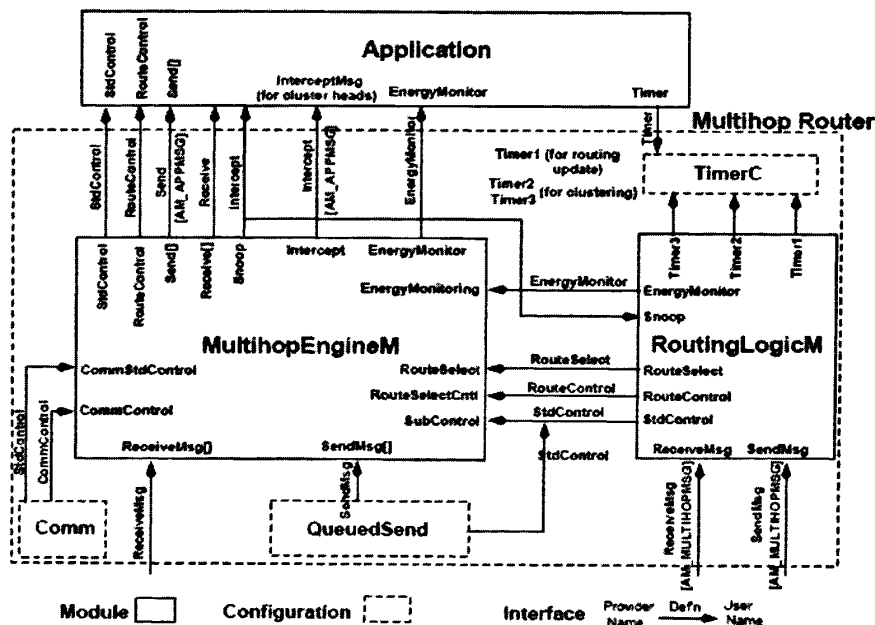


图 5.2 模块连接关系示意图

5.2.3 算法主要功能模块设计

1、网络节点初始化

在网络运行初期进行簇首选举或网络运行一段时间进行新一轮的簇首选举之前，网络中的节点首先要初始化。利用 **BcastMsg**、**Sense** 接口中的命令，处理过程如下：

```
command void clusterInit( u )
{
    传感器节点  $u$  的  $CHNodeID$  置为空:  $CHNodeID = NULL$ ;

    每隔时间  $T$  广播 Hello 分组  $Hello[NodeID_u, CHNodeID_u, ER_u]$ ;

    等待时间  $T_w$ ，邻居节点交换 Hello 分组信息;

    if( 节点  $u$  未加入分簇并且经簇首选举方法选择自己作为簇首节点:
         $CHNodeID_u = NULL \ \&\& CHNode\_Elect(NT_u) = NodeID_u$  )
    {
        节点  $u$  将  $CHNodeID_u$  置为自己的 ID 编号:  $CHNodeID_u = NodeID_u$ ;

        广播簇首声明分组: Broadcast CHNode Report Packet;
    }
}
```

2、网络分簇

在簇首节点选举过程中，未能成为簇首节点的普通节点选择加入簇中。利用 **Sense**、**BcastMsg** 和 **EnergyControl** 接口中的命令，处理过程如下：

```
task void myClusterTask( )
{
    while( 节点  $u$  监听邻居节点发送的 Hello 分组信息，如果监听到信息，
        则执行下面操作:  $ReceiveHello[NodeID_w, CHNodeID_w, ER_w]$  )
    {
        根据 Hello 分组更新本节点的邻居节点信息表  $NT$ :  $update(NT)$ ;

        if( 如果收到的 Hello 分组为簇首声明分组:  $CHNodeID_w = NodeID_w$  )
        {
            if( 如果本节点  $u$  尚未加入分簇:  $CHNodeID_u = NULL$  )
            {
```


节点 u 加入该分簇，将自己的 $CHNodeID_u$ 置为 $NodeID_w$ ：

$CHNodeID_u = NodeID_w$ ；

广播加入分簇分组：Broadcast Participation Cluster Packet;

}

elseif(如果本节点 u 为簇首节点： $CHNodeID_u = NodeID_u$)

{

进行簇首节点竞争；

if($ER_u < ER_w \parallel (ER_u = ER_w \ \&\& \ NodeID_u > NodeID_w)$)

{

节点 u 放弃成为簇首节点，并加入分簇：

$CHNodeID_u = NodeID_w$ ；

广播加入分簇分组：Broadcast Participation Cluster Packet;

}

}

else

{//节点 u 已经加入分簇

调用 EnergyControl, 比较监听到的两个簇首声明分组信号强度；

加入离自己更近的分簇；

广播加入分簇分组：Broadcast Participation Cluster Packet;

}

}

if(如果收到的 Hello 分组为加入分簇分组：

$CHNodeID_w \neq NodeID_w \ \&\& \ CHNodeID_w \neq NULL$)

{

if(节点 u 未加入分簇并且经簇首选举方法选举自己作为簇首节点：

$CHNodeID_u = NULL \ \&\& \ CHNode_Elect(NT_u) = NodeID_u$)

{

节点 u 将 $CHNodeID_u$ 置为自己的 ID 编号： $CHNodeID_u = NodeID_u$ ；

广播簇首声明分组：Broadcast CHNode Report Packet;

}

}

}

}

3、广播并处理路由请求

对于数据源节点，在没有到汇聚节点 Sink 的可用路由的时候，需要广播路由请求分组。利用组件 **Bcast Module** 实现路由请求分组的转发与处理，处理过程如下：

```

module BcastM
{
    provides
    {
        interface StdControl;
        interface Receive[uint8_t id];
    }
    uses
    {
        interface StdControl as SubControl;
        interface ReceiveMsg[uint8_t id];
        interface SendMsg[uint8_t id];
    }
}

implementation
{
    ///广播路由请求分组
    command void FwdBcast(Path_Request_Pack *pRcvPack, uint8_t Len,
                          uint8_t id)
    {
        广播路由请求分组 Preq_Pack;
    }
    //节点接收、处理并转发路由请求分组
    task void ReceiveMsg.procREQ [uint8_t id](Path_Request_PackPtr pPack)
    {
        if( 监听到路由请求分组 )
            接收路由请求分组 Preq_Pack;

        if( 转发 Preq_Pack 分组的节点是源节点的一跳邻居节点，

```

```

ReadDate.read(Firsthop)==NULL)
{
    设置分组中 Firsthop 字段值为 TOS_LOCAL_ADDR
    ReadDate.set(Firsthop)= TOS_LOCAL_ADDR;
    读取分组 Preq_Pack 中的路由信息 ReadDate.read(path), 并创建反
        向路由;

    if( 当前节点是汇聚节点 TOS_LOCAL_ADDR == SINK_ADDR )
    {
        发布路由回复分组处理任务 post ReceiveMsg.procREP;
        丢弃 Preq_Pack 分组;
    }
    elseif( 当前节点存在到达汇聚节点的路由 )
    {
        跳至 FLAG1 执行语句
        JUMP FLAG1;
    }
    else
    {
        更新 Preq_Pack 分组
        ReadDate.upDate();
        转发更新后的 Preq_Pack 分组
        call FwdBcast();
    }
}
elseif( 处理过具有相同的 Preq_Pack 分组 )
{
    if( 需要更新路由 )
    {
        读取分组 Preq_Pack 中的路由信息 ReadDate.read(path), 并创
            建反向路由;
    }
    else
    {

```

```

        跳至 FLAG1 处执行语句
        JUMP FLAG1;
    }
}
else
{
    跳至 FLAG1 处执行语句
    JUMP FLAG1;
}
FLAG1: 丢弃 Preq_Pack 分组;
}
}

```

4、广播并处理路由回复

汇聚节点 Sink 或记录有到达汇聚节点路由的中间节点在收到路由请求分组的时候，广播路由回复分组，建立正向路由。利用组件 **Bcast Module** 实现路由回复分组的转发与处理，处理过程如下：

```

module BcastM
{
    provides
    {
        interface StdControl;
        interface Receive[uint8_t id];
    }
    uses
    {
        interface StdControl as SubControl;
        interface ReceiveMsg[uint8_t id];
        interface SendMsg[uint8_t id];
    }
}

implementation
{
    //广播路由回复分组

```

```

command void FwdBcast(Path_Reply_Pack *pRcvPack, uint8_t Len,
                      uint8_t id)
{
    广播路由回复分组 Prep_Pack;
}
//节点接收、处理并转发路由请求分组
task void ReceiveMsg.procREP [uint8_t id](Path_Reply_PackPtr pPack)
{
    if( 监听到路由回复分组 )
        接收路由回复分组 Prep_Pack;
    if( Prep_Pack 分组与节点路由表中的 Firsthop 字段值相等 )
    {
        if( 当前节点没有到达上一跳节点的正向路由 )
        {
            读取分组 Prep_Pack 中的路由信息 ReadDate.read(path), 并创
                建正向路由;
        }
        if( 需要更新当前节点的正向路由信息 )
        {
            更新当前节点的正向路由信息 updatePathTable;
        }
        if( 当前节点源节点 TOS_LOCAL_ADDR == SOURCE_ADDR )
        {
            创建了一条源节点到汇聚节点的可用路由;
            丢弃 Prep_Pack 分组;
        }
        else
        {
            更新 Prep_Pack 分组: ReadDate.upDate();
            转发更新后的 Prep_Pack 分组: call FwdBcast();
        }
    }
}
}

```

5.2.4 仿真程序运行

采用 nesC 语言设计本文路由协议，并在 TOSSIM 中进行了仿真：

1. 在路由协议程序目录下运行命令：`make pc`，将程序代码编译成可以在 PC 端仿真工具 TOSSIM 上运行的二进制格式，生成可执行文件 `main.exe`。
2. 通过 gdb 调试程序，使用命令：`gdb build/pc/main.exe`，将程序的调试结果显示出来。
3. 使用 TinyViz 命令：`tinyviz -run build/pc/main.exe -num`，跟踪程序的运行，显示程序运行过程中节点间的通信以及节点的输出信息。
4. 利用命令：`o file-name`，将仿真结果输出到某个文件中，以便数据分析。
5. 统计数据，利用 Matlab 做出结果分析图。

5.3 仿真结果与性能分析

无线传感器网络是通过其网络中的有效传感器节点执行环境监测与数据采集等任务，而一个传感器节点的有效性，主要取决于节点的活动性（即能否监测采集数据）以及和网络中其它节点的连通性（即能否发送、转发监测数据至汇聚节点）。随着网络运行时间的增加，传感器节点由于能量耗尽或连通性等原因由有效节点变为失效节点，当网络中有效节点的数量占全部节点的比例低于某一百分比时，则认为传感器网络的生存时间已经结束^[32]。

由于目前尚没有针对无线传感器网络中路由协议的统一的衡量标准，根据文献[7, 9]中对传感器网络路由协议性能的评测方法，本节我们对本文算法以及缠绕多路径路由算法 Braid 和链路不相关多路径路由算法 AOMDV 进行能量有效性和传输可靠性的验证，主要在网络生存时间、汇聚节点接收数据分组数量、路由算法开销三个方面上进行比较分析。其中：

■网络生存时间

该性能评测标准是体现路由协议是否符合能量有效性要求的重要指标，主要包括节点存活个数百分比和平均剩余能量两个方面。

■汇聚节点接收数据分组数量

该性能评测标准是指汇聚节点在网络运行过程中接收到的由数据源节点发来的数据分组总数，是反映数据传输可靠性的主要衡量指标。

■路由算法开销

该性能评测标准是指用于路由建立和路由维护的控制分组的数量。该指标反映了路由算法的效率。

为了验证本文算法的能量有效性和传输可靠性, 缠绕多路径路由算法 **Braid** 和链路不相关多路径路由算法 **AOMDV** 运行在由 **LEACH** 算法生成的层次拓扑结构之上。仿真中, 我们在 200×200 的矩形区域中均匀布置 100 个节点, 每个传感器节点具有 500 点(points)的初始化能量。

5.3.1 网络生存时间

该仿真中, 主要衡量不同算法下, 网络生存时间的变化趋势, 主要从节点存活个数百分比和平均剩余能量两个方面进行比较分析。

图 5.3 是存活节点个数百分比随时间变化图。图中纵坐标是网络中生存节点的数目, 横坐标是网络运行周期数。网络初始化后节点定期发送数据, 当一个节点能量耗尽就退出网络, 仿真运行直到所有节点全部死亡。

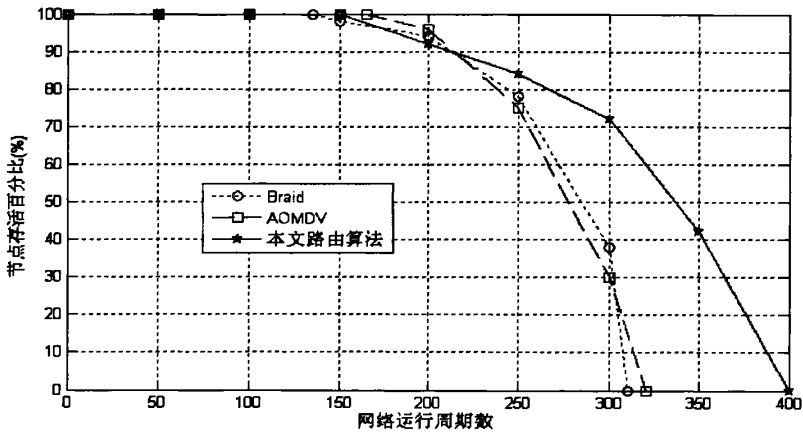


图 5.3 存活节点个数百分比

如图 5.3 所示, 在网络运行初期 200 个周期之前, 运行三种算法的网络节点存活百分比没有差别, 但是随着网络运行时间的增加, 这方面的差别出现并逐渐明显, 这可以解释为运行 **Braid** 和 **AOMDV** 网络的拓扑结构是由 **LEACH** 算法生成的, **LEACH** 通过随机数选取簇首节点, 没有考虑节点的剩余能量因素, 随着网络的运行, 节点能量逐渐消耗, 这种不足就体现出来了, 而本文提出的层次拓扑生成算法考虑了节点的剩余能量, 簇首节点选择剩余能量较高的节点充当, 因此, 采用本文算法的网络节点存活百分比随网络运行时间的变化曲线比较平缓, 具有较高的网络生存时间。如果将网络生命期定义为从初始布置网络后到还有 40% 节点存活的周期数, 则从图 5.3 可以看出, 采用缠绕多路径路由算法 **Braid** 和链路不相关多路径路由算法 **AOMDV** 的网络生存时间与采用本文算法的网络的生存时间相比, 相差将近 100 个周期, 因此, 本文所提出的路由算法能够获得

很好的延长网路生存时间的性能。

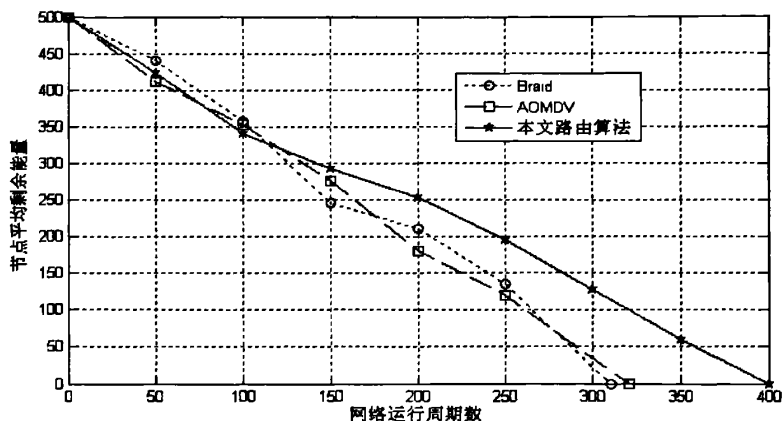


图 5.4 节点平均剩余能量

图 5.4 是平均剩余能量随时间变化图。图中纵坐标是平均剩余能量，横坐标是网络运行周期数。如该图所示，在网络运行初期 150 个周期之前，本文算法与另外两种算法在节点平均剩余能量方面的表现没有明显区别，但是随着网络运行时间的增加这方面的表现区别逐渐明显，从网络运行的整个时期来看，本文算法平均剩余能量变化较另外两种算法相对平缓，其原因与图 5.3 的解释相同，与其他两个算法相比具有较高的能量有效性，因此从节点平均剩余能量的角度进一步验证了运行本文算法的网络能够获得较长的网络生存时间。

5.3.2 汇聚节点接收数据分组数量

该仿真中，主要是衡量不同算法下，汇聚节点在相同时间内接收到的数据分组数量，该衡量标准是用于比较三种算法在提高传感器网络的数据传输可靠性方面的性能。

图 5.5 是汇聚节点接收数据分组数量随时间变化图，图中纵坐标是汇聚节点接收到数据分组的数量，横坐标是完成网络运行的周期数。如图中所示：三种算法在相同的实验环境下，汇聚节点获取的数据分组数量不同，从上图中可以看出三种算法中汇聚节点接收到数据分组数量随时间的增长而增加，并且可以明显看出采用本文算法的网络其汇聚节点收到的数据分组数量最多，这是由于 Braid 算法和 AOMDV 算法建立的多条路径之间均存在共用节点，在网络运行初期 150 周期之前，失效节点不多的情况下，并不会影响数据分组的传输，但是随着网络的运行，在 150 周期之后，节点能量逐渐消耗殆尽，失效节点增多，共用节点失效的情况也会频繁起来，共用节点的失效会导致多条路径断裂，这样对传输数据

分组的影响也会越来越严重,本文路由算法建立的多条路径是节点不相关的,不存在共用节点,因此,数据分组的传输不会受到上述情况的影响。在网络运行的最后阶段 350 周期之后,由于失效节点越来越多,三种算法均可能找不到数据源节点与汇聚节点之间的路径用来传输数据,因此汇聚节点收到的数据分组数量不再增加。从网络运行的整个时期来看,在三种算法中本文算法还是获得了较优的结果,体现出了较高的传输可靠性。

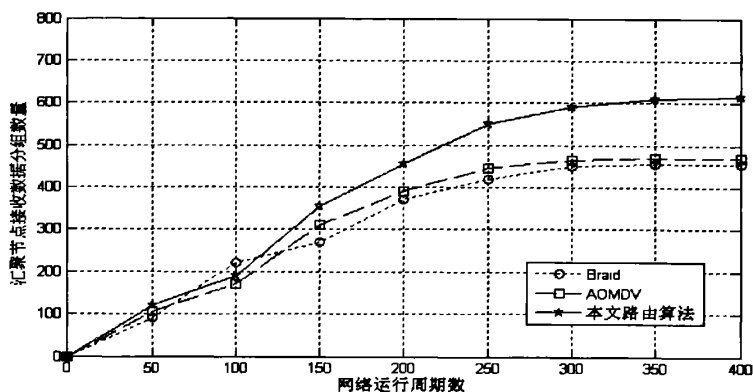


图 5.5 汇聚节点接收数据分组数量

5.3.3 路由算法开销

图 5.5 是路由算法开销随时间变化图,图中纵坐标是路由算法开销,横坐标是网络运行的周期数。

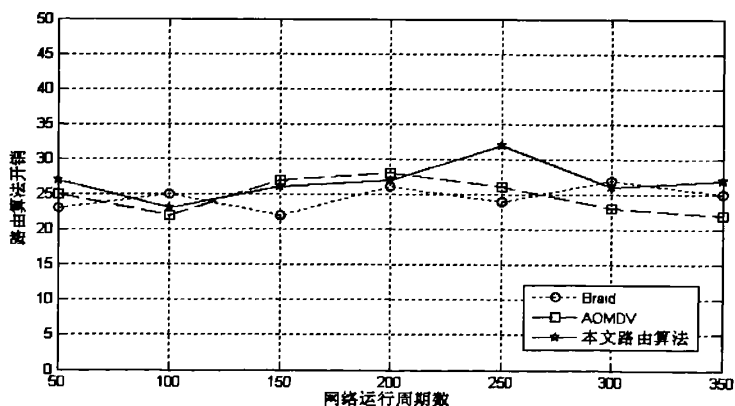


图 5.6 路由算法开销

本文路由算法是对以往单路径路由算法的改进,改进之后算法能够在数据源节点和汇聚节点之间建立多条节点不相关路径,这样势必增加路由算法的开销,但是,如图 5.6 中所示,本文算法的开销与其他两个算法相比,在 200 至 300 周

期之间,本文算法开销稍高于另外两种算法,其他周期内的情况是三者之间的开销相差不大。从整体来看,本文算法的开销同另外两种算法相比不算明显,综合考虑上面的衡量标准,我们可以看出,同缠绕多路径路由算法 **Braid** 和链路不相关多路径路由算法 **AOMDV** 相比,本文算法在没有明显增加开销的情况下,却比较明显的提高了数据传输的效率和网络生存时间,因此体现了数据传输可靠性和能量有效性。

5.4 小结

本章借助仿真工具——**TOSSIM** 对本文所提出的算法进行了仿真,主要测试了能量有效性和传输可靠性两个方面,具体包括了三种性能评测标准:网络生存时间、汇聚节点接收数据分组数量和路由算法开销。仿真结果显示本文提出的路由算法是可行的,在每条路径建立时需要满足节点不相关的苛刻条件,增加了路径建立延时,但相对于该方法在提高路径传输可靠性以及能量有效性方面所表现出的有效性是值得的。通过分析比较,在能量有效性和传输可靠性两个方面,本文提出的算法都具有较优的性能表现,符合无线传感器网络能量有效、路径可靠的数据传输路由的要求。

第六章 总结与展望

6.1 总结

作为 21 世纪最重要技术之一的无线传感器网络,是传感和计算功能有机的无缝结合,因此在资源受限环境下如何最有效、最长期的获取监测对象的特征信息,并有效可靠的传输到数据处理节点,是传感器网络路由研究的主要问题。

多路径路由问题在传感器网络的可靠性传输研究和应用中占据举足轻重的地位。本文在广泛阅读国内外有关数据可靠性传输方法的资料文献的基础之上,通过对国内外现有路由协议的分类、比较和分析,提出了一种基于层次拓扑结构的多路径路由协议,实现了多路径路由的建立方法和维护机制,并在仿真工具中仿真实现了多路径路由协议。本文具体工作主要体现在以下几个方面:

(1)深入研究了传感器网络区别于传统有线网络和其它无线网络的一些特点,分析了一些传感器网络路由算法设计的影响因素。对现有的一些传感器网络路由算法和数据可靠传输方法进行分类的基础上,总结分析了它们的优缺点,为设计性能更好的路由算法奠定基础。

(2)通过对传感器网络中的现有路由协议分类并分析比较,得出层次型拓扑结构较其他的拓扑结构更有利于节省传感器网络中节点的能量,能够较程度的延长网络的生存时间。在此基础上,提出了基于节点剩余能量和节点 ID 编号值的分簇算法,该算法综合考虑了节点剩余能量和节点 ID 编号值,能够生成能量有效的层次拓扑结构,延长了网络的生存时间。

(3)分类并研究了传感器网络中数据的可靠传输方法,得出多路径方法在提高传感器网络数据的可靠传输方面具有显著的效果,在借鉴以往路由算法思想的基础上,提出了基于层次型拓扑结构的传感器网络多路径路由算法。该算法在网络层次拓扑结构之上,在数据源节点与汇聚节点之间建立多条路径,进行数据传输。该算法建立的多条路径具有节点不相关性,可以避免路径之间的关联性和弥补传统多路径在网络突发状况下的不足,从而提高了网络的数据传输可靠性。

(4)借助传感器网络仿真工具——TOSSIM,对本文提出的基于层次拓扑结构的多路径路由算法进行了仿真,设计了基于层次型拓扑结构的传感器网络原型系统,并开发了相应的功能模块。通过将本文提出的算法与现有的多路径路由算法在能量有效性和网络可靠性两方面进行了比较。仿真结果表明本文提出的算法具有较高的数据传输可靠性和能量有效性。

6.2 展望

基于层次拓扑结构的无线传感器网络多路径路由算法在本文中已经进行了详细的论述，并通过仿真实验对其进行了相关验证，在本文研究工作的基础上，还可以从以下几个方面对此系统作出一些探索：

■ 跨层设计

在文中所提出的多路径路由算法的基础上，可以综合考虑运行在数据链路层的纠错码等编码机制和链路重传机制，以跨层设计的方式来进一步提高数据传输的可靠性。

■ 节点覆盖范围控制

传感器网络中的节点具有高度冗余性的特点，在保证监测区域被最大面积的覆盖的前提下，通过控制冗余节点睡眠，降低节点之间的通信干扰，也能够提高数据传输的可靠性，并可以延长网络的生存时间。

■ 网络安全问题

不同于有线网络传输媒介具有物理保护，而传感器网络用于数据传输的无线电波完全暴露于外界，如何保证网络的安全问题，避免网络的数据报文被盗听窃取、修改重要信息等将成为数据可靠传输研究的重要部分，同时这这也是一个刻不容缓的难题。

参考文献

- [1] Kulik J, Heinzelman W R, Balakrishnan H. Negotiation-based Protocols for disseminating information in wireless sensor networks In: Wireless Networks, 2002,(8): 169-185.
- [2] Intanagonwiwat C, Govindan R, Estrin D. Directed Diffusion: a scalable and robust communication paradigm for sensor networks In: Proceedings of the ACM MobiCom'00, Boston, MA, 2000. 56-67.
- [3] Heinzelman W, Chandrakasan A, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks In: IEEE Transaction on Wireless Communications, 2002, (10):660-670.
- [4] Manjeshwar A, Agrawal DP. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In: Proceedings of the 15th Parallel and Distributed Processing Symposium. San Francisco: IEEE Computer Society, 2001. 2009-2015.
- [5] A. Manjeshwar and D. P. Agarwal, APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks In: Parallel and Distributed Processing Symposium, Proceedings International, IPDPS 2002, 195-202.
- [6] S. Lindsey, C.S. Raghavendra, PEGASIS: power efficient gathering in sensor information systems, In: Proceedings of IEEE Aerospace Conference, 2002. 1125-1130.
- [7] Shijin Dai, Xiaorong Jing, Lemin Li. Research and Analysis on Routing Protocol for Wireless Sensor Networks. IEEE. 2005. 0-7803-9015-6/05:407-412.
- [8] Akyildiz I F, Su W, Sankarasubramanian Y, Cayirci E. A survey on sensor networks. IEEE Communications Magazine. 2002. 40(8):102-114.
- [9] Deepak Ganesan, Ramesh Govindan, Scott Shenker, Deborah Estrin. Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks. Mobile Computing and Communications Review. 2002. Vol.1. Number 2.
- [10] 任丰原, 黄海宁, 林闯. 无线传感器网络. 软件学报. 2003, 14(7):1282-1291.
- [11] Mahesh K. Marina, Samir R. Das. On-demand Multipath Distance Vector Routing in Ad Hoc Networks. IEEE, 2001, 1092-1658/01:14-23.
- [12] Rahul C, Jan M. Energy Aware Routing for Low Energy Ad Hoc Sensor Networks. IEEE. WCNC2002.:1-6.
- [13] De S, Qiao C, Wu H. Meshed multipath routing: An efficient strategy in sensor networks. Wireless Communications and Networking (WCNC 2003). IEEE. Vol 3:16-20.
- [14] Xu Y, Heidemann J, Estrin D. Geography-informed energy conservation for ad hoc routing, In: Proc 7th Annual Int'l Conf on Mobile Computing and Networking (MobiCOM), Rome, Italy. July 2001:70-84.
- [15] Gerla M, Tsai J T C. Multicluster, Mobile, Multimedia Radio Networks. Wireless Networks, 1995, 1(3): 255-265.

- [16] Ossama Yonniss, Sonia Fahmy. Hybird, energy-efficient, distributed clustering approach for Ad Hoc Sensor Networks. IEEE TRANSACTIONS ON MOBILE COMPUTING. VOL.3. NO.4:366-378.
- [17] 孙利民等著. 无线传感器网络. 北京: 清华大学出版社, 2005.
- [18] Pister K, Hohlt B, Jeong J, Doherty L, Vainio J P. Ivy. A sensor network infrastructure. 2003.<http://www.bsac.eecs.berkeley.edu/projects/ivy>.
- [19] Shijin Dai, Xiaorong Jing, Lemin Li. Research and Analysin on Routing Protocol for Wireless Sensor Networks. IEEE. 2005. 0-7803-9015-6/05:407-412.
- [20] 沈波, 张世永, 钟亦平. 无线传感器网络分簇路由协议. Journal of Software. Vol 17. No.7.July 2006:1588-1600.
- [21] Jamil Ibric, Imad Mahgoub. Cluster-Based Routing in Wireless Sensor Networks: Issues and Challenges. SPECTS'04. 1-56555-284-9:759-767.
- [22] Sohrabi K, Gao J, Ailawadhi V, Pottie G J. Protocols for self-organization of a wireless sensor network. IEEE Personal Communications, 2000,7(5): 16-27.
- [23] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "Two-tier data dissemination model for large-scale wireless sensors networks," ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2002), Atlanta, Georgia, USA, September 2002, pp:148-159.
- [24] Ozgur B. Akan, Ian F. Akyildiz. Event-to-Sink Reliable Transport in Wireless Sensor Networks. IEEE/ACM TRANSACTIONS ON NETWORKING. VOL.13, NO.5. OCTOBER 2005:1003-1017.
- [25] Michael Hempel, Hamid Sharif, Prasad Raviraj. HEAR-SN: A New Hierarchical Energy-Aware Routing Protocol for Sensor Networks. Proceedings of the 38th Hawaii International Conference on System Sciences. IEEE. 2005. 0-7695-2268-8/05.
- [26] Jun Wang, Haibin Yu, Zhijun Shang. Research on Reliable Link Layer Communication in Wireless Sensor Networks. IEEE. 2005. 0-7803-9015-6/05:417-421.
- [27] Sukun K, Rodrigo F, David C. Reliable Transfer on Wireless Sensor Networks. IEEE. 2004:449-459.
- [28] Qilian Liang, Lingming Wang. Fault-Tolerant Multipath Transportation Aided with Channel Coding and Interleaver for Wireless Sensor Networks.2004.IEEE.0-7803-8523-3:3679-3683.
- [29] 董梅, 杨曾, 张健, 王能. 基于信号强度的无线局域网定位技术.计算机应用.2004.12. CNKI:ISSN:1001-9081.0.2004-12-017
- [30] Deb B, Bhatnagar S, Nath B. ReInForM: Reliable Information Forwarding using Multiple paths in sensor networks. Proc 28th Annual IEEE Conf on Local Computer Networks(LCN), October 2003.
- [31] Wenjing Lou, Younggoo Kwon. H-SPREAD: A Hybrid Multipath Scheme for Secure and Reliable Data Collection in Wireless Sensor Networks. IEEE Transactions on vehicular technology. Vol 55, No.4. July 2006: 1320-1330.
- [32] 郭宗宝. 无线传感器网络数据传输可靠性研究.哈尔滨工业大学硕士学位论文.20050601.

- [33] Gupta G, Younis M. Fault-tolerant clustering of wireless sensor networks. Proc IEEE Wireless Communications and Networking(WCNC 2003).vol.3:16-20.
- [34] Stephen Mueller, Rose P. Tsaag, Dipak Ghosal. "Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges".2004.
- [35] Charles E. Perkins, Elizabeth M.Belding-Royer, Samir R.Das. "Ad Hoc On-Demand Distance Vector(AODV) Routing". IETF Internet Draft Feb. 2003.
- [36] Dennis Baker, M. Scott Corson, Paul Sass, S. Ramanathan. Flat vs. Hierarchical Network Control Architectures. ARO/DARPA Workshop on Mobile Ad Hoc Networking, Univ. of Maryland, Maryland, March 1997.
- [37] OPNET. OPNET Technologies, Inc. <http://www.opnet.com>.
- [38] QualNet. Scalable Network Technologies, Inc. <http://www.scalable-networks.com/>.
- [39] OMNET++. András Varga. Department of Tele-communications in Technical University of Budapest. <http://www.omnetpp.org/index.php>.
- [40] NS-2. The University of Southern California's Information Sciences Institute. <http://www.isi.edu/nsnam/dist/>.
- [41] GloMoSim. UCLA Computer Science Department. <http://pcl.cs.ucla.edu/projects/glomosim/>.
- [42] Philip Levis, Nelson Lee, Matt Welsh, David Culler Philip Levis TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. SenSys'03, Los Angeles, California, USA. November 5-7. ACM 1-58113-707-9/03.
- [43] TOSSIM. UC Berkeley. <http://www.tinyos.net>.
- [44] Philip Levis. UC Berkeley. TinyOS Programming. June 28, 2006.

致 谢

值此论文完成之际，我要衷心感谢所有关心、支持和帮助过我的老师和朋友们！

首先要向我的导师张建明教授致以最深切的谢意，感谢张老师在这三年来对我耐心的教育和指导。从张老师身上学到的不仅仅是专业知识，同时也学到了不少为人处世的道理。张老师广博的学识、敏锐的思维、严谨的态度、宽广的胸怀已经并将继续地影响我今后的学习、工作和生活，使我受益终身。

感谢詹永照教授，在课题研究过程中得到詹老师的悉心指导。作为一名计算机科学领域的专家，詹老师治学严谨，知识渊博，工作作风勤恳，对我的成长产生很大的影响。

感谢王良民老师，王老师对课题研究具有独到的见解，对我的研究课题提出了很多启发性的建议。同时，还要感谢网络课题组的所有老师及组员们，感谢他们对我的帮助。

感谢师兄史益斌、田建华，师姐顾丽芳、余群；感谢同门周玲玲、缪少军、杨丽瑞、陈君、牛险峰；感谢师妹樊丽静、房芳，师弟胡耀峰、江长勇、刘育、杨忠、许博文，感谢他们在学习和生活上对我的帮助和支持。还要感谢师兄文传军，他豁达的谈吐是我望尘莫及的。

感谢球队成员狮子、大伟、小明、小祝、小晔、老胡、老左等等，感谢他们和我一同渡过了三年的充实且健康的研究生生活，留下了很多快乐和难忘的回忆，这将会成为我人生历程中深刻的一页。

借此机会，还要感谢我的父母和哥哥，感谢他们多年来对我无私的关爱与鼓励，他们是我鼓起勇气克服困难，继续向前的源泉。

最后，感谢培育我的江苏大学，感谢计算机学院的所有老师们！

硕士期间发表的论文

在攻读硕士学位期间，作者发表论文如下：

1. 张建明, 张沛, 王良民. 基于移动代理的无线传感器网络可靠路由协议. 通信学报（增刊）. 2007, 11, No.11A, Vol.28: 93-97.
2. 张沛, 张建明. 基于蚁群算法的能量有效的传感器网络可靠路由协议. 计算机工程与设计. 2008, 11.