

密级： 保密期限：

北京邮电大学

硕士学位论文



题目：无线多跳网络高速路由研究与实现

学 号：2013140093

姓 名：江善桃

专 业：电子与通信工程

导 师：廖青

学 院：信息与通信工程

2015 年 11 月 24 日

独创性（或创新性）声明

本人声明所呈交的论文是本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中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他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。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在 年解密后适用本授权书。

非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名：_____ 日期：_____

导师签名：_____ 日期：_____

无线多跳网络高速路由研究与实现

摘 要

无线多跳网络是由带有无线收发装置的移动节点组成的,能够在节点间进行多跳通信的临时性自治网络。无线多跳网络具有灵活机动、无需基站支持、组网快速、适应环境能力强、多跳路由等优良特性,而且还具有顽健性、抗毁性、成本低等特点。凭借这些特性,让其在市场应用中占据一席之地,使得其广泛应用于未来军事竞争、灾后通信、传感网络、车载网络、无人机等领域。

路由协议是建立无线多跳网络的首要问题。学术界对无线多跳网络路由技术的研究已经开展多年,研究出多种适应各种应用需求的无线多跳网络路由协议。但是这些路由协议针对的无线多跳网络的中继节点移动速率较慢,每秒移动速率一般为每秒数米或者十几米。针对节点高速移动的场景,节点移动速度可达音速,需要考虑场景应用的特殊性,在原有的协议基础上进行改进,使路由协议能适应节点高速移动场景的需求。

因此,本文主要是基于高速移动的应用场景,进行路由协议设计、仿真和实现,使得设计的路由协议能够适应节点高速移动特性,保障组网能力和路由效率。论文研究的重点是路由协议的设计,结合链路状态协议和 GPS 地理定位技术的优点,改进传统 OLSR 协议,在路由寻找和路由表计算中引入链路稳定间隔 (Link Stability Duration, LSD) 预测因子,辅助路由控制消息的广播和最短路径的选择。在 NS2 (Network Simulator 2) 网络仿真器上实现路由协议模块,验证协议的可行性,并与传统 OLSR 协议进行性能比较。最终,将路由协议在 uC/OS-II 嵌入式系统上实现,和其他的模块协作,共同组成一个完整的高速移动无线多跳网络组网的通信系统。

关键字 无线多跳网络 高速路由协议 链路稳定间隔 NS2 uC/OS-II

RESEARCH AND IMPLEMENTATION OF AD-HOC HIGH-SPEED ROUTING PROTOCOL

ABSTRACT

Wireless multi-hop network(Ad-hoc) is composed of mobile nodes with wireless transmitters and receivers which can networking rapidly between nodes. Ad-hoc network has advantages of flexibility, no need of base support, rapid networking, strong ability to adapt to the environment and multi-hop routing. It is robust, invulnerable, and low-cost. So it is widely used in future military war, disaster communication, sensor networks, vehicle networks, UAVS and the like.

Routing protocol is the most important issue to establish a wireless multi-hop network. Academic study to routing technology of wireless multi-hop network has been carried out for many years, and has developed a variety of routing protocols. But these routing protocols just adapt to low speed scenarios, generally tens of meters per second. For high-speed scenarios, it is necessary to improve these routing protocols to meet the high-speed feature.

The main research of this thesis is routing protocol design, simulation and implementation which are based on high speed scenarios, making the routing protocol to meet the demands of high mobility characteristics, rapid networking capability and good routing efficiency. The designed routing protocol combines the advantages of link-state protocol and GPS positioning technology to improve the OLSR protocol. It introduces the predictor of Link Stability Duration (LSD) which predicts the survival time of the route according to the motion and position information provided by GPS. The LSD contributes to broadcasting routing control messages and Dijkstra's shortest path algorithm in the proceed of routing finding and routing table calculation.

This thesis confirms validity and feasibility of the designed protocol and compares the performance with the traditional OLSR protocol through the simulation experiments in the NS2 simulation platform. Finally, implement the routing protocol on the uC/OS-II embedded system, collaborate with other modules together to compose a complete high-speed mobile wireless multi-hop networking system.

KEY WORDS Wireless Multi-hop Network High-speed Routing
 Protocol Link Stability Duration NS2 uC/OS-II

目录

第一章 绪论.....	1
1.1 引言.....	1
1.2 论文的主要内容.....	2
第二章 无线多跳网络路由协议.....	3
2.1 路由协议概述.....	3
2.2 路由协议质量要求.....	3
2.3 路由协议的评价指标.....	5
2.4 路由协议的分类.....	6
2.4.1 主动路由协议和按需路由协议.....	7
2.4.2 平面路由协议和分级路由协议.....	8
2.4.3 地理定位辅助路由协议和非地理定位辅助路由协议.....	9
2.4.4 单路径型路由协议和多路径型路由协议.....	10
2.4.5 单播路由协议和多播路由协议.....	11
2.5 高速路由协议现状.....	12
2.6 小结.....	13
第三章 高速路由协议的设计和仿真.....	15
3.1 高速路由协议需求场景.....	15
3.2 高速路由协议的设计.....	16
3.2.1 GPS 辅助的 LSD 链路稳定间隔预测算法.....	17
3.2.2 链路稳定间隔 LSD 在寻路和路由算法中的应用.....	18
3.2.3 路由表算法.....	19
3.3 NS2 仿真平台.....	22
3.3.1 路由协议关键结构说明.....	23
3.3.2 路由协议的仿真分组格式.....	25
3.3.3 路由协议的操作.....	26
3.5 NS2 仿真验证场景设计.....	27
3.6 仿真结果分析.....	30
3.7 与传统 OLSR 协议的性能比较.....	31
3.7.1 仿真参数说明.....	32
3.7.2 仿真结果说明.....	33
3.8 小结.....	35
第四章 高速路由协议在 uC/OS-II 上的实现.....	37
4.1 uC/OS-II 嵌入式系统.....	37

4.2 多跳系统架构.....	37
4.3 网络层结构设计.....	39
4.4 网络层功能接口设计.....	40
4.5 网络层具体实现.....	43
4.5.1 列表结构.....	43
4.5.2 路由表结构.....	44
4.5.3 数据包缓存队列.....	44
4.5.4 拓扑消息的结构.....	44
4.5.5 全局变量.....	45
4.5.6 网络层的数据包格式.....	45
4.5.7 路由表的计算.....	45
4.5.8 处理应用层的数据包发送.....	45
4.5.9 收到来自物理层的消息处理.....	46
4.6 测试结果.....	46
4.6.1 测试环境.....	46
4.6.2 测试结果.....	46
4.7 小结.....	50
第五章 总结和展望.....	51
5.1 本文总结.....	51
5.2 研究展望.....	51
参考文献.....	53
致谢.....	57

第一章 绪论

1.1 引言

传统的通信网络对基础设施有很大的依赖性，建造和运行的成本很高，抗毁性差，在战争和自然灾害环境中很容易遭到破坏。而且随着移动产品的普及，针对终端设备的高度移动性、不需要有线基础设施的支持、能够快速组网和快速恢复等的要求，研究人员提出了一种新型的网络结构——无线多跳网络。

无线多跳网络，又叫移动自组织网络（Ad Hoc），是一种具有高度动态拓扑结构、节点能够任意移动、加入和退出的网络，它强调的是网络中过度依赖网络中继节点，有异于有线网络，通过移动节点的自组织功能来组网通信，且节点间进行分布式的自我创建，管理和组织。由于两个或多个需要通信的节点之间常没有直连的链路，需要其他节点的多次中继转发。

无线多跳网络具有灵活机动、无需基站支持、适应环境能力强、组网快速、多跳路由等优良特性，并且还具有顽健性、抗毁性、鲁棒性、成本低等特点。凭借这些优良特性，让其在市场应用中占据有一席之地，使得其广泛应用于未来军事竞争、移动漫游、灾后通信、传感网络、车载网络、无人机等领域。

正是由于无线多跳网络的这些特性，使得它的体系结构、服务质量保障、物理安全性等问题变得复杂，传统的各种性能良好的协议和技术都无法直接应用于无线多跳网络。其实现的关键技术和实现难点主要有网络体系结构、路由协议、MAC 协议、协议安全问题、无线网络的资源管理和网络的可扩展性等方面。考虑到无线多跳网络的拓扑变化快，节点可以任意接入和退出特点，设计良好的路由协议就成了组网的关键因素。

路由协议是建立无线多跳网络的核心技术，同时也是学术研究的重点。无线多跳网络的节点移动性高，网络短暂分裂概率高，无线带宽资源有限，传统的距离矢量路由协议和链路状态路由协议在拓扑变化快速无线多跳网络中，显示出其自身的缺陷，因此需要研究一种能够适应拓扑变化的动态路由协议^[1]。所以无线多跳网络的路由协议需要具备掌握全网拓扑变化，节点间的自由组网，高度自适应的能力及节点能量有效利用等功能，此外，还需要考虑多路径路由、多播路由、路由安全性，和稳定性路由等问题。学术界对无线多跳网络路由协议进行大量探讨，研究出了很多适应各种需求的相对成熟的路由协议^[2]。

1.2 论文的主要内容

但是这些路由协议针对的无线多跳网络的中继节点移动速率较慢,每秒移动速率一般为每秒数米或者十几米。针对节点高速移动的场景,如节点移动速度高达百米每秒,传统路由协议将不能良好的适应高速特性,需要考虑场景应用的特殊性,在原有的协议基础上进行改进,使路由协议能适应节点高速移动场景的需求。

本文主要是对无线多跳网络高速路由的研究和实现,针对高速移动的场景,假设每个节点都配有 GPS 设备,能够传递自身的位置,高度,运动姿态和速度等信息。参考现有成熟协议的设计思想,进行路由协议的改进和设计,使其满足高速移动的环境。结合状态路由协议和 GPS 地理定位技术,在传统 OLSR 协议的基础上,引入链路稳定间隔(Link Stability Duration, LSD)预测因子的计算,预测和保证协议的稳定性,辅助路由控制消息的广播和最短路径的选择。简化路由协议的消息结构,减小网络开销,设计一个适合中小型高速移动网络的路由协议。在 NS2(Network Simulator 2)网络仿真器上实现路由协议模块,验证协议的可行性,并与传统 OLSR 协议进行性能比较。最终,将路由协议在 uC/OS-II 嵌入式系统上实现,和其他的模块协作,构建一个完整的无线多跳网络通信系统,以期达到能够在小规模无人机等军事探测或灾情地理探测等领域的应用。

本文主要分为五个章节:

第一章对无线多跳网络进行介绍,包括无线多跳网络的定义,路由协议的研究现状和热点,明确高速移动路由的研究方向和研究内容。

第二章对现有无线多跳网络路由协议进行介绍,包括移动路由协议的质量要求,评价标准,协议的分类,优缺点比较分析,对路由协议有一个全面的把握。

第三章针对高速移动场景,结合链路状态协议和 GPS 定位技术的优点,改进 OLSR 路由协议,引入链路稳定间隔 LSD(Link Stability Duration)因子,辅助路由控制消息的广播和路由算法的决策,减少路由控制开销,增加路由稳定性。在 NS2 平台上,实现该路由协议模块,设计仿真场景对协议的可靠性进行验证,并传统的 OLSR 协议进行性能比较,结果表明引入 LSD 因子后,路由协议在高速场景中的路由断裂次数和路由广播分组数量上有更良好的性能,提高路由稳定性,减少路由开销。

第四章将引入 LSD 因子的 OLSR 路由协议移植到 uC/OS-II 轻量级嵌入式系统上,对网络层功能进行封装,提供统一的对应用层和 MAC 层的通信接口,与其他模块协作,构建适应高速场景的无线多跳网络系统。

第五章对研究课题的总结,并展望将来在该课题上的深入研究的方向和实际应用的实现。

第二章 无线多跳网络路由协议

2.1 路由协议概述

研究学者已经对路由协议、功率控制、MAC 协议、网络安全等课题做了多年的研究，取得丰硕成果。在所有课题中，无线多跳网络的路由协议问题最为重要。IETF 成立了 MANET 工作部门集中力量从事无线网络路由协议的研究和优化工作。

路由协议包含两个功能，一方面是确定最佳的传输路，另一方面是通过网络传输数据。后者称之为数据分组的交换，该功能思路简单，但是前者选择路由的难度较大。路由协议的跳数损耗是路由协议算法用来确定从源节点到达目的节点的最佳路径所经过的中间节点数量，是衡量路由协议最佳路径的计量标准。为了在全网中选择合适的路径，路由算法需要先初始化网络拓扑消息，更新路由表，收集路径信息，而路径信息的格式和内容根据使用的路由算法决定。路由表一般会包含很多信息，如目的节点的地址、下一跳节点的地址、总跳数等。数据分组交换的流程包括：某节点发起数据分组的请求，通过查找节点维护的路由信息表，根据目标节点的 IP 地址，确定下一跳节点的 IP 地址，然后，源节点填写下一跳节点的 MAC 地址，将数据分组发送到底层链路。如果当前节点的路由表中不存在去往目的节点的路由，则将数据包丢弃，并报告给网络层链路丢失。重复以上动作，直到下一跳就是最终的目的节点。

路由协议的功能是网络层的核心功能，任何网络需要发送数据分组，都需要路由协议的辅助，无线多跳网络也不例外。无线多跳网络的路由选择受很多因素的影响，如网络的规模，网络适应的业务类型，信道的特性等。无线网络的信道干扰和节点的移动性，导致网络拓扑结构快速变化，因此其路由问题比固定网络或传统的无线网络要复杂^[3]。

在无线多跳网络路由协议的设计过程中，需要克服的最大难题就是由节点的随机移动、加入和退出而带来的整个网络拓扑结构的随机性，因此路由协议需要适应于频繁变化的网络拓扑环境。无线多跳网络中的资源有限，无线环境多变，信道干扰，路径损耗，信道衰落等因素的影响，决定了路由协议不能过于复杂，计算量不能过大。

2.2 路由协议质量要求

路由协议为节点维护一张通向全网节点的路由表。由于无线多跳网络具有节

点随机移动性、网络拓扑高度动态、传输多跳、无线信道不稳定、带宽资源有限等特性,从技术上看,无线多跳网络的路由设计面临许多挑战。路由协议的基本思想是减少路由协议的能量消耗,控制消息的带宽消耗,有效提高网络的吞吐量,从有限的信息中推导出有关网络变化的重要信息,采用自适应技术,根据网络拓扑的变化,动态调整路由协议和参数。当前,对路由协议的研究主要集中在路由的安全性,可扩展性、节能路由等领域。

传统的有线路由协议主要有两种,一种是基于距离矢量的路由协议,一种是基于链路状态的路由协议^[4]。距离矢量协议的基本思想是,每个节点保存并维护到达所有目的节点的距离矢量信息,包括路径的开销,如跳数,距离损耗等,下一跳节点的地址。每个节点只将自身的路由信息传递给自己的邻居节点,节点将收到的最新路径信息和原来的数据进行对比,以此来更新自身的路由信息。如果原来的路由有效并且开销小,则保留原来的数据,否则用新的最佳路由信息去更新路径信息。距离矢量路由协议的收敛速度慢,路由搜索过程有可能产生环路,造成数据链路的死循环。而链路状态路由协议的基本思想是,通过让所有的节点各自维护一张全网拓扑结构图,根据全网拓扑结构计算出到达所有目的节点的路由表,可以有效解决路由环路问题。但是,路由控制信息是向全网广播洪泛,网络带宽资源消耗较大。

无线多跳网络是一个多跳的移动计算机网络,多跳是研究无线多跳网络路由协议的基础。一个主要的难点在于:一方面,节点需要了解可以到达其邻居节点的信息,用来确定分组的路由;另一方面,网络拓扑会频繁变化。当网络节点的数目很大时,节点间会频繁交换数据分组,所以更新路由的流量会很大。总的来说,在进行路由设计时,需要考虑以下几点:网络的自组织能力,拓扑的高度动态性,移动终端限制,网络安全,无线带宽资源限制,单向信道的存在,生存时间的短暂性和分布式网络控制。

针对无线多跳网络的特性,对路由协议提出了更高的要求,主要有以下几种:

1) 分布式计算:路由计算必须是分布式的,在一个动态的、无中心的网络中采用集中的方式选择路由是不现实的,即便是一个很小的动态网络。

2) 收敛速度:网络拓扑结构的快速变化需要路由协议拥有较快的反应能力,路由算法的收敛速度快,能够快速搜索通往全网节点的路径,避免节点不可达的结果。

3) 无环路由:无环或非循环路由是所有路由协议都应该遵循的原则,无环工作可避免带宽的浪费。

4) 支持单向链路:在设计无线多跳网络路由协议时,通常假设节点间的链路是双向的,因此很多路由算法在单向链路上不能工作。但是由于网络节点具有

异构化的特点，由传输的不对称性引起的单向链路时常也是可能的。因此，路由协议按特定场景的需求，考虑充分利用单向链路。

5) 网络安全：无线多跳网络如果没有任何形式的网络安全或链路级安全措施，则当面对攻击时会显得很脆弱，容易在传输分组信息中被窃听、修改路由信息和数据分组包头等。

6) 终端限制：无线多跳网络中的终端是用能量有限的移动电源，相比于固定有线终端，其 CPU 性能较低，电量和存储容量有限，所以路由协议不能对终端设备的性能要求过高，即路由协议的算法的计算量不宜太大，复杂的不宜过高。

7) 避免无穷计算：经典的距离矢量算法在某条链路失效的情况下，可能会出现无穷计算的问题。无线多跳网络中链路断裂导致无穷计算的概率增大，所以要求路由协议能够避免无穷计算，即不采用会出现无穷计算的路由算法或改进这种算法。

8) 算法简单性：路由算法的简单有助于提高路由的可靠性，减少路由、控制、管理的开销。由于无线带宽资源的限制，减少各种非数据的传输，能更加有效地利用宝贵资源，因此要求在实现路由协议功能的基础上，尽可能使路由算法简单。

2.3 路由协议的评价指标

无线多跳网络的应用相当的广泛，在不同的应用场景中，对网络的性能要求指标也是不同的，如在以实时传输业务为主的应用场景下，要求路由建立的时间或端到端延时越短越好；而在高清图像的传输为主的应用场景中，要求路由开销尽可能小或端到端的数据吞吐量尽可能大。因此，无线多跳网络到底该采用何种路由协议的问题，是以应用场景或业务需求为导向的，不同应用场景会采用不同的路由评价指标。下面是几种最常见的无线多跳网络路由协议的性能评价指标 [5-6]。

1) 路由跳数：路由跳数是指数据分组从源节点出发到达目标节点的过程中经过的中间节点的数目。在路由协议的评价指标当中，路由跳数是一项重要的指标。数据分组在网络中经过的中间节点越多，被中间节点中转传输的次数就越多，而因此造成的分组丢失的可能性就越大，每个节点消耗的能量就越大，传递一个数据分组所需要的开销就越大，造成整个网络生存时间的降低。

2) 路由建立的时间：路由建立的时间是指从源节点发出路由请求到获取到完整的路由所需要耗费的时间。这个指标是专门针对按需路由协议设定的，对实时性业务影响很大。网络中很多的因素都会影响到路由建立的时间，包括无线链路的带宽、节点间的物理距离、网络的拥塞情况等等。

3) 路由开销: 路由开销是路由协议的效率问题, 有关路由开销的定量评价指标也有很多, 比如整个网络发送的路由控制信息比特平均数量占整网接收到的数据分组比特的平均数量的百分比、整网发送的路由控制分组的平均数量占全网接收到的数据分组的平均数量的百分比等。不同的定量指标, 在收集仿真数据时, 难易程度也不相同。无线多跳网络中无线带宽的有限性要求路由的控制开销尽可能小。

4) 端到端时延: 端到端时延是指源节点产生数据分组到目标节点接收到数据分组所经过的平均时间。忽略中间节点对数据分组的处理时间和数据分组在中间节点的等待时间, 端到端时延主要由路由建立时间、数据分组的发送时延和数据分组的传播时延三个部分组成。端到端时延会直接影响到通信服务的质量和用户的体验满意度, 对实时性业务的影响更是明显。影响路由建立时间的因素很多, 主要是网络中所选取的路由协议、网络的规模大小、网络的负载率等; 数据分组的发送时延则取决于数据分组的长度和无线信道带宽的比值, 而数据分组的传播时延则决定于节点间的物理距离和电磁波传播速度的比值。

5) 路由可靠性: 可靠性是指路由协议对网络连接的依赖程度, 一般是以误码率、误比特率、分组丢失率或分组投递率等指标来衡量, 以上指标能够反映出路由协议的完整性和正确性。由于网络拓扑结构的频繁变化, 因此很容易就使得原先的、甚至是刚刚建立起来的路由, 因中间节点链路的断裂而失效。可靠性高的路由算法会综合考虑节点的移动趋势、路由已经被使用的时间、接收信号能量、备用路由、路由抗噪声性能等方面的因素, 尽可能减少路由失效情况和路由重新建立的次数, 提高路由协议的感知能力和路由重建的效率, 从而充分利用已有的信息保障路由的可靠性。

目前已经存在的很多路由协议各有特点, 可是还是没有一种路由协议能够适用于所有网络, 所以在实际选择路由协议时, 仍需要根据实际应用场景的需求对路由协议进行决策和性能测定。

2.4 路由协议的分类

目前, 已经存在数十种适应无线多跳网络环境的路由协议。对于已经提出的路由协议, 可以从不同的角度进行不同的分类: 根据路由发现策略的不同, 可以分为主动路由协议和按需路由协议; 根据网络的逻辑结构不同, 可以分为平面路由协议和分级路由协议; 根据是否使用 GPS 地理定位系统作为路由辅助功能, 可以分为地理位置定位辅助路由协议和非地理位置定位辅助路由协议等。

2.4.1 主动路由协议和按需路由协议

根据路由发现的策略, 可以将路由协议分为主动路由协议和按需路由协议, 这是最常见的一种路由协议类型。主动路由协议也称作表驱动路由协议或先应式路由协议, 该协议在每一个节点上维护一个路由表, 包含了该节点到达其他节点最新的路由信息。每个节点周期性地向全网广播路由更新消息, 用于维护并更新全网的拓扑结构变化和其他节点路由的信息。这类路由协议是在现有的有线路由协议的基础上, 进行改进算法, 使其适应无线多跳网络的拓扑特性。

主动路由协议的优点主要在于当前节点需要发送分组数据时, 到目的节点的路由存在, 当前节点可根据目的节点的地址从路由表中获得路由信息, 不用重新进行路由搜索过程来寻找最佳路径, 这个有点能够满足实时性业务的质量要求。主动路由协议的不足在于维护路由表需要较高的代价, 比如带宽、能量、计算能力等, 以使得路由表中的信息尽量能跟得上当前整个网络拓扑结构的变化, 但是高度动态变化的网络拓扑结构可能会使花费较大代价获得的路由表信息变得无效, 路由协议的收敛性能较差。即使路由协议能够始终收敛, 由于路由控制消息的开销很大, 当网络中的节点数量达到很大规模, 路由控制信息会占用大量的网络带宽资源, 不能满足网络规模扩张的需求, 不适用与网络规模庞大, 节点众多的多跳无线网络。而且, 控制分组的全网洪泛还会消耗节点的能量, 降低全网的生存时间。

根据路由表的数量, 表项内容和广播策略的不同, 目前已经提出了不少具体的主动路由协议, 典型的有目的节点序列距离矢量路由 (Destination-Sequenced Distance-Vector Routing, DSDV)^[7]、优化的链路状态协议 (Optimized Link State Routing, OLSR)^[8,9]、无线路由协议 (Wireless Routing Protocol, WRP)、鱼眼状态路由 (Fisheye State Routing, FSR)^[10]、系统与业务关联的自适应路由算法 (System and Traffic Dependent Adaptive Routing Algorithm, STARA)、基于拓扑广播的反向路径转发 (Topology Broadcast Based On Reverse-Path Forwarding, TBRPF)^[11]等。而且, 针对上述的这些主动路由协议, 已提出了各种改善机制用以提高路由协议的性能。

按需路由协议是专门用于无线多跳网络环境的协议。在按需路由协议中, 网络中的节点不用维护全网的最新路由信息, 只要在源节点需要发送分组时, 根据网络的状况, 按照设计好的路由算法搜索有效的路由即可。按需路由协议相比于主动路由协议, 其优势主要在于节点不用周期性的发送路由控制分组, 网络带宽开销小, 节省了一定量的网络资源, 比如能耗, 带宽等, 能够延长网络的生存时间。而且, 因为路由是按照需求即时搜索的, 所以路由表中的路由信息可能只包含了全网拓扑结构中的一部分, 这就使得路由表项比较少, 路由表的内容也相对

简单。该类路由协议缺点就是当源节点有通信需求时，如果路由表中不包含到达目的节点的有效路径信息，节点就会启动路由的搜索和创建流程，数据分组的发送会因为路由表的创建搜索而被迫延迟，不是很适合用于实时性要求比较高的业务中。而且，按需路由协议寻找路由的方式通常也是通过向全网洪泛数据分组的机制进行路径的搜索，这会在一定程度上削弱按需路由协议带来的好处。

目前，按需路由协议主要有 Ad Hoc 按需距离矢量路由（Ad Hoc On-Demand Distance Vector Routing, AODV）协议^[12]、临时预订路由协议（Temporally-ordered Routing Algorithm, TORA）^[13]、动态源路由（Dynamic Source Routing, DSR）协议、基于信号稳定度的自适应路由算法协议（Signal Stability-based Adaptive Routing Protocol, SSR）^[14]、联合稳定度路由（Associability based Routing, ABR）协议等。

总体来说，主动路由协议和按需路由协议各有其自身的优势，分别应用于不同的需求场景中。但比较而言，按需路由协议更适合于无线多跳网络，因其更适应该网络的动态变化特性，对带宽、能量和移动性的要求。这两类路由协议的部分性能比较总结在表 2-1。

表 2-1 主动路由协议和按需路由协议的比较

参数性能	按需路由协议	主动路由协议
路由开销	小	大
拓扑变化的适应性	好	差
路由建立时延	大	小
移动处理性	使用本地路由信息转发	通知其他节点使路由保持一致
是否需周期性更新	否	是
路由信息的可用性	需要时有效	一直有效
产生的信令开销	随移动性增加而增加	比按需路由大

2.4.2 平面路由协议和分级路由协议

根据网络的逻辑拓扑结构，路由协议分为平面路由协议和分级路由协议。平面路由的网络逻辑视图是平面结构的，在网络内的所有节点的地位是平等的，具有完全相同的功能。拓扑结构是平面的网络组成简单，源节点与目的节点之间通信通常存在多条路径。多条路径的存在为不同业务提供了分流的功能。

分级路由的网络逻辑视图具有层次结构,网络节点根据不同的集群和分区算法,可以划分成不同的层次,在初级集群和区域概念上,又可以再次的细分出超群和超区,以此类推建立更加细致的层次结构。在这种分级结构的无线多跳网络中,主要依靠群首来传输群和群之间的数据分组。

平面路由协议由于节点的平等性,各个节点具有相同的功能地位,非常适用于分布式控制,易于网络的扩展,不存在网络瓶颈效应,流量可以相对均分到各个节点身上,起到分流的作用,稳定性较好。缺点在于当网络规模扩大时,网络复杂度增加,为了维护全网的连通性,需要耗费巨大的网络带宽来洪泛控制消息,不利于网络规模的扩展。而分级路由的优点是隐藏了网络拓扑结构的细节,降低大规模网络路由控制的开销,网络的路由表规模减小,通过分群得到比较稳定的小规模网络。但是,由于需要通过群首节点进行集群之间的数据分组交换,当群首出现问题,将对整个子网造成影响,存在明显的网络瓶颈效应。

分级路由协议主要包括区域路由协议 (Zone Routing Protocol, ZRP)^[15]、分群网关交换路由 (Cluster-head Gateway Switching Routing, CGSR)、基于分群结构的路由协议 (Cluster Based Routing Protocol, CBRP)^[16]、分级状态路由协议 (Hierarchical State Routing, HSR)、自组网路标路由协议 (Landmark Ad Hoc Routing Protocol, LANMAR)、基于区域的分级链路状态协议 (Zone-Based Hierarchical Link State Routing Protocol, ZHIS)^[17]等。平面路由协议和分级路由协议的比较参看表 2-2 所示。

表 2-2 平面路由协议和分级路由协议的比较

参数或性能	平面路由协议	分级路由协议
可扩展性	弱	强
复杂度	低	高
使用范围	小规模网络	大规模网络
健壮性	强	弱
安全性	高	低

2.4.3 地理定位辅助路由协议和非地理定位辅助路由协议

根据是否使用 GPS (Global Positioning System) 定位设备等提供的位置信息 (坐标位置和运动信息) 可以将路由协议分为地理定位辅助路由协议和非地理定位辅助型路由协议。在无线多跳网络中, GPS 定位设备提供的节点位置信息和运动数据, 可以用于预测节点运行状态, 预测网络拓扑的变化, 充分利用这些信息, 可以有效的提高路由协议的可靠性和效率。现存的无线多跳网络协议, 可以通过引入 GPS 设备提供的位置信息和运动数据, 进行有效的改进, 提升现有协议的

效率。但是目前利用 GPS 定位信息的路由协议还不多。常见的有地理辅助路由协议 (Location Aided Routing, LAR)^[20]、贪婪型转发和沿周边转发路由 (Greedy Perimeter Stateless Routing, GPSR)^[19]、移动距离效应路由算法 (Distance Routing Effect Algorithm For Mobility, DREAM)、基于移动预测的面向流的路由协议 (Flow Oriented Routing Protocol, FORP), 基于移动预测的距离向量路由协议 (Distance Vector with Mobility Prediction, DV-MP) 等。

由于无线多跳网络拓扑结构具有高度的动态变化的特性, 因此基于地理定位的路由协议得到广泛的应用。在地理定位辅助路由协议的基础上, 学者们提出了很多基于链路稳定性和基于链路均衡性的路由协议, 但是由于一些如实现成本高、路由协议复杂度很高等因素, 基于地理定位的路由协议在很大程度上还没有实用化。

2.4.4 单路径型路由协议和多路径型路由协议

根据源目的节点对之间保存的路径数目可以将无线多跳网络的路由协议分为单路径路由协议和多路径型路由协议。以上讨论的所有路由协议都是单路径型的协议, 下面将要以主流的按需路由协议为例, 来说明单路径路由协议的缺陷。按需路由协议只有当源节点有通信需求的时候, 才会发起路由更新的过程。因为按需路由协议依靠洪泛技术寻找、维护、更新路由, 因此当终端节点随机移动的时候, 网络中会有中等数量到大量的路由需要维护, 此时需要频繁的全网洪泛来对路由进行有效的控制, 如此将会降低按需路由协议的性能。

多路径型路由协议^[21]的基本思想是在一次的洪泛查询路由过程中, 探测或建立多跳可能的路由, 在数据分组转发的阶段, 将网络流量分配到多条路径中。源目的节点对之间的多条路径能够补偿无线多跳网络的动态特性和不可预测性。多路径型路由协议的优点是能够以低成本提供足够多的冗余度, 能够提高通信节点对带宽的有效利用率, 能够就平衡网络的负载, 能够更好的响应网络拥塞及突发传输, 提高分组的交付可靠性⁵⁴。此外, 研究表明: 在高密度的无线多跳网络中, 采用多路径型路由协议能够得到比单路径型路由协议更好的网络吞吐量。多路径路由协议的主要缺点在于它的复杂性和开销: 一方面, 在多路径扩展中, 维护多条到达目标节点的路径将导致中间节点更大的路由表; 另一方面, 在多路径路由中, 因为多路径可能导致报文乱序, 故必须考虑用何种方法将报文分配到多条路径上。

目前, 针对无线多跳网络已经提出了很多多路径型路由协议, 但其中大多数的工作方式与理论上的是不一样的。虽然在一次洪泛查询路由的过程中建立了多条的路径, 但在数据分组转发阶段只使用一条路由, 其他路由只有在主路由失效的时候才会使用。由于网络流量没有分配到多条路径中, 所以多路径型路由的潜

力并没有完全发挥出来。当前，典型的无线多跳网络多路径型路由协议主要有 Ad Hoc 按需多路径距离矢量路由 (Ad Hoc On-Demand Multipath Distance Vector, AOMDV) 协议^[25]、分离多路径路由 (Split Multipath Routing, SMR) 协议^[26]、多路径源路由 (Multipath source Routing, MSR) 协议、多路径源动态路由协议 (Multipath dynamic source Routing, MP-DSR)^[27]、TBRPF (Topology dissemination based on Reverse path forwarding)、CHAMP (Caching and multipath Routing Protocol)^[28]等。

2.4.5 单播路由协议和多播路由协议

根据数据分组中目的节点的个数的不同,可以将无线多跳网络的路由协议分为单播型路由协议和多播型路由协议。以上讨论的路由协议都是单播型路由协议。在无线多跳网络的众多应用领域当中,如灾难救助、战场指挥、临时会议等,通常都存在大量的一对多或多对多的通信。通常情况下,可以使用单播、多播和广播这三种模式进行通信。由于多播通信能充分利用无线信道的广播特性,因此有助于控制网络的流量,减少节点的处理量和占用的带宽资源,降低传输费用、能量的消耗和传输时延,是实现点对点或点对多点通信的最有效的方式^[29]。

多播的实现需要多播路由协议的支持,多播路由协议常见的思路是在群组成员间构造一颗分发树,数据分组在这棵树上传输,使得所有接受者都能收到该数据分组。在无线多跳网络中,也是采用这个思路。因为多播路由本身就是网络的一个难点,在加上无线多跳网络的无线传输、动态拓扑等特性,使得多播路由实现难度更大,除了要解决单播路由面临的问题外,还需要考虑一些新问题,如节点可以在任意时刻加入和退出组,组成员的数量和位置没有限制,节点可能是多个组的成员等^[30]。

鉴于无线多跳网络自身的特性,需要专门设计多播路由协议。目前,已经提出一些针对无线多跳网络的多播路由协议,如自组网多播路由协议 (Ad Hoc Multicasting Routing, AMRoute)^[31]、利用递增序号的自组网路由协议 (Ad Hoc Multicast Routing Protocol utilizing Increasing id-number, AMRIS)^[32]、前传组多播协议 (Forwarding Group Multicast Protocol, FGMP)^[33]、按需多播路由协议 (On-Demand Multicast Routing Protocol, ODMRP)^[34]、组播自组网按需距离矢量路由协议 (Multicast Ad Hoc on-demand Distance Vector Routing Protocol, MAODV)、借助核心的网格协议 (Core Assisted Mesh Protocol, CAMP) 等。

表 2-3 常见几种多播路由协议关键特性比较

多播路由协议	AMRoute	ODMRP	CAMP	MAODV	AMRIS
路由结构	树状	网状	网状	树状	树状
是否无回路	否	是	是	是	是
是否依赖于单播协议	是	否	是	是	否
周期性控制分组	是	否	是	否	是
控制分组扩散	是	是	否	否	是

目前,无线多跳网络多播路由协议正朝着更高的顽健性、灵活性和高效性的方向发展,以使新的多播路由协议能够动态的适应网络环境的变化。此外,节省能量也是多播协议的研究方向之一。

2.5 高速路由协议现状

以上讨论的各类路由协议,使用与最常见的应用场景中,如车载网络,无线感知网络,救灾网络等,这些网络场景都有一个明显的特点,网络中节点的移动速度相对较低,一般的车载网络中,车辆的运行速度平均是每小时 60 至 120 公里,约合每秒 20 到 50 米。而在高速飞行领域,如无人机,导弹等,其节点移动速度高达秒数百米,这与普通的应用场景,在速度量级上就差了一到两个数量级。在如此高速的移动网络中,传统的无线多跳网络路由协议很难直接适用。为了适应高速移动场景,需要对现有的协议进行改进,提高协议的自适应性,增加硬件设备,如 GPS 定位系统,来增加网络的有效信息,帮助节点更加高效地收集和维持网络拓扑结构。

高速网络的特点是节点的运动速度很高,通信的范围增大,导致网络的拓扑结构变化更加的复杂和频繁,这就增加了路由的搜索、建立和维护的难度。由于其适用的场景不普及,因此这方面的研究相对较少。目前的研究主要是在原有的路由协议基础上,改进路由的算法,增加网络中的拓扑变化信息的采集和预测,来提高路由协议的效率,让路由算法的收敛速度更快。

在无人机研究领域,提出全局可动态重组战略网(GDSN)和无人机协同作战模式(UAS-COM),构建出可动态重定义的网络体系结构(DRNA),用以适应无人机高速移动过程中的通信组网建设。在路由协议方面,有基于源路由和多路径的OLSR 协议(SRMP-OLSR),基于负载度的自适应多路径选径(LAMP-DSR)和基于移动代理的混合多路径路由算法(MAH-MPR)等。这些算法都是在主动路由协议和按需路由协议的基础上,结合各协议自身的特点,引入多路径的方式,提升路由的稳定性和效率。在相关方面的研究还有待深入,这也是本文课题的研究点。

2.6 小结

本章通过对路由协议的基本功能，质量要求和评价指标的介绍，说明衡量路由协议的方法。介绍了常见的无线多跳网络路由协议的分类，通过对比，说明无线多跳路由协议没有绝对的优劣，路由协议的应用必须结合其应用的场景，业务的需求，根据这些环境特点，设计特定的路由协议来适配各种生产环境，以达到规定的质量要求。高速移动场景下，节点的运动速度高，网络拓扑变化更加快速频繁，对路由协议提出更高的要求，需要更加有效的路由协议解决方案。

第三章 高速路由协议的设计和仿真

尽管国际学者对无线多跳网络路由技术的研究已经开展多年,但是针对的是无线多跳网络的中继节点移动速率较慢的场景,每秒移动速率一般为每秒数米或者十几米。例如,步兵运兵车或普通交通工具的移动速度为每小时几十公里(约十几米每秒);而美国无人机的时速可以达到六百公里/小时(约 150 米/秒),这两类场景在移动速度上有一个数量级的差别。因此,针对节点高速移动的场景,需要考虑场景应用的特殊性,在原有的协议基础上进行改造,使路由协议能适应节点高速移动场景的需求。

本章的内容是基于节点高速移动的小型无线网络应用场景,进行路由协议设计、仿真和实现,使得设计的路由协议能够适应节点高速移动特性,保障组网能力和应对节点的高度可变化特性。结合链路状态协议和 GPS 定位技术的优点,改进 OLSR 路由协议,引入链路稳定间隔 LSD (Link Stability Duration) 因子,辅助路由控制消息的广播和路由算法的决策,减少路由控制开销,增加路由稳定性。在 NS2 平台上,实现该路由协议模块,设计仿真场景对协议的可靠性进行验证,并与传统的 OLSR 协议进行性能比较,结果表明引入 LSD 因子后,路由协议在高速场景中的路由断裂次数和路由广播分组数量上有更良好的性能,提高路由稳定性,减少路由开销。

3.1 高速路由协议需求场景

本章讨论的高速路由协议适应的场景是一个节点高速移动的小型无线多跳网络,节点的移动速度高,可以达到每秒几十米至几百米。在节点飞行的过程中,节点可以自由的加入和退出。节点的数目一般在十数个左右,属于小型无线多跳网络规模。每一个节点配备有 GPS 定位设备,可以提供自身的位置和运动信息数据。

网络层的总体目标是建立和维护一个能适应高速移动节点的稳定的无线多跳网络。它应具备以下几个特点:具有高抗毁性的网络结构;当网络遭到攻击或自身运动而被破坏时拥有自动愈合的功能;当转发用户的数据时能够解决隐藏终端和暴露终端的问题;在进行路由发现时能够抑制广播风暴的出现;接口结构合理,能为应用层和链路层提供通畅的数据传输转发功能,并支持多跳转发。

该路由协议需要完成的功能包括:一,初始组网阶段,周期性地广播入网握手帧,侦听握手帧和发送确认帧;二,搜集网络拓扑数据信息,根据数据计算网络拓扑结构表,形成全网连通矩阵;三,网络维护阶段,需要保证节点加入、退

出组网的正确性。

考虑到该场景下的无线多跳网络的规模比较小,可以采用链路状态协议的思想来设计该路由协议,在链路状态协议的思想,根据 GPS 定位设备提供的节点位置信息,辅助路由的选择和广播时机预测,因此,路由协议的复杂度相对可以降低,以提高稳定性和可靠性为主。

3.2 高速路由协议的设计

针对上述场景的特点,由于网络的规模较小,节点移动高速,节点具备 GPS 定位设备的辅助,考虑链路状态协议和 GPS 地理位置辅助方法相结合,根据链路协议周期性广播路由控制消息,控制消息中带上节点的位置信息,每个节点独自维护一张全网的拓扑结构,在收到控制消息后,更新拓扑表。利用收集到的节点位置消息,计算节点间的稳定链路间隔 (Link Stability Duration, LSD), 根据 LSD 来预测下次广播路由控制消息的时机,并在计算全局路由表的时候,根据预测的 LSD 来选取一条预测最稳定的通信链路,充分利用位置数据提供有效信息,优化传统 OLSR 状态路由协议。

每一个节点的路由层需要负责维护全网的拓扑结构的变化,网络的拓扑结构可以用连通矩阵这个结构来存储,连通矩阵中的元素代表了全网节点间两两互通情况,该连通矩阵由两部分组成:邻居列表和拓扑列表。邻居列表中包含的是当前节点所有有效的一跳范围内的邻居节点,即与当前节点直接连通的节点集合。邻居列表中元素的更新维护是通过节点发送一跳 Hello 消息相互通知对方的。而拓扑列表中的元素代表网络中其他节点的连通情况,它是通过节点周期广播拓扑 (TC) 消息来更新维护的。

邻居列表的维护主要靠广播一跳的 Hello 消息,携带自身的位置速度相关信息;当收到 Hello 消息,比较序列号来确定是否加入新的邻居;每个邻居表项都有一个定时器,维护邻居节点的有效时间,过期后则删除该邻居。

拓扑列表的维护主要靠全网广播 TC 消息,每个 TC 消息携带自己的邻居列表相关信息;当收到 TC 消息,比较序列号确定是否更新拓扑表;每个拓扑表项各自维护一个定时器,维护其有效性,过期则删除该拓扑表项。

路由表的计算就是通过查询这两张表,并按照最短路径算法来计算,每当处理完 Hello 消息或 TC 消息,重新计算路由。

每个节点还维护一张重复数据表,记录从每个节点收到的数据包의 源地址,序列号,以防止重复处理相同的数据包,消除数据包环路。

本协议的设计目标是能够适应节点高速移动的无线多跳网络,因此定义了多种稳定路由机制,即根据链路状态、位置移动等辅助信息来实现路由优化,选择

网络中相对稳定的移动节点组建路由，提高动态环境中的路由数据传输质量；采用目的序列号来确保路由不会出现环路。

3.2.1 GPS 辅助的 LSD 链路稳定间隔预测算法

在传统的 OLSR 链路状态协议思想的基础上，结合基于 GPS 定位系统提供的节点位置和运动信息，对协议进行优化。路由协议掌握各节点的位置运动等辅助数据，可以预测出与节点通信链路的稳定时间，预测节点的轨迹，由此来辅助路由寻找和路径选择的决策，减少周期性广播数据分组的洪泛，选择更加稳定的链路进行通信。

根据 GPS 定位设备提供的节点运动信息，通过各节点收集全网节点位置信息，可以预测两节点间的链路的稳定间隔（Link Stability Duration, LSD），然后依据预测的链路的稳定间隔做全网拓扑稳定性预测和周期性广播分组的间隔决断依据，进而为找到更加稳定的路由提供保障，减少网络开销。

具体的算法是根据节点的当前位置和运动信息，结合收集到的其他网络节点的 GPS 辅助信息，依据两点间的运行轨迹预测两节点链路断裂的时间，将此时间作为稳定链路选择和全网路由消息广播分组间隔的判断依据。

两个相邻节点位置及运动信息如下图所示：

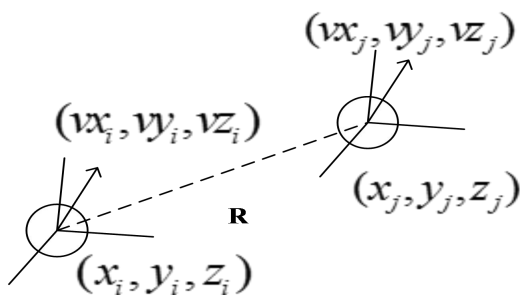


图 3-1 相邻节点链路稳定间隔 LSD 计算参数图

(vx_i, vy_i, vz_i) 及 (vx_j, vy_j, vz_j) 分别代表节点 i 及节点 j 的移动速度向量， (x_i, y_i, z_i) 和 (x_j, y_j, z_j) 分别表示节点 i 和节点 j 的位置坐标向量， R 为节点的有效通信半径。此处假设两个节点始终处于同一平面。

当两个节点间的距离超出有效通信半径 R 时，链路将会断裂，从节点开始连接到链路中断之间经过的时间间隔就是链路稳定间隔。

用相对运动的原理来计算，假设 x 轴上节点的相对运动速度是 a ，相对距离

是 b , y 轴上节点的相对运动速度是 b , 相对距离是 d , z 轴上节点的相对运动速度为 e , 相对距离为 f , 则:

$$\begin{aligned} a &= vx_i - vx_j \\ b &= x_i - x_j \\ c &= vy_i - vy_j \\ d &= y_i - y_j \\ e &= vz_i - vz_j \\ f &= z_i - z_j \end{aligned} \quad (3-1)$$

相对坐标 (b, d, f) , 经过时间 t 后, 两个节点的相对坐标, 即两节点间距离为 $(b+at, d+ct, f+et)$

维持链路稳定状态的两节点间距离的临界值是 R , 所以

$$(b+at)^2 + (d+ct)^2 + (f+et)^2 = R^2 \quad (3-2)$$

化简得

$$(a^2 + b^2 + c^2)t^2 + 2(ab + cd + ef)t + b^2 + d^2 + f^2 - R^2 = 0 \quad (3-3)$$

解一元二次方程得

$$t = \frac{-(ab + cd + ef) \pm \sqrt{(a^2 + c^2 + e^2)R^2 - [(ad - bc)^2 + (af - be)^2 + (cf - de)^2]}}{a^2 + c^2 + e^2}$$

所以得到公式

$$LSD = \frac{-(ab + cd + ef) \pm \sqrt{(a^2 + c^2 + e^2)R^2 - [(ad - bc)^2 + (af - be)^2 + (cf - de)^2]}}{a^2 + c^2 + e^2}$$

依据以上的推导过程, 可以得到两个节点按照当前的自身的运动信息状态进行运行, 则预测在 LSD 时间内, 两节点间的链路将是稳定的, LSD 值越大, 说明两节点间链路的局部稳定性就越好。

两节点间的预测链路稳定间隔可以根据上述公式求得, 得到的 LSD 值说明了两个节点按照当前获知的运动信息持续运动, 两节点间的链路能够保持连通的最大稳定间隔。因此, 可以根据预测值 LSD 来作为广播路由控制消息时间间隔的决定因素, 并且根据 LSD 的值来在做最短路由选择算法中的一个权值因子, 帮助算法找到一条更加稳定的路由。

3.2.2 链路稳定间隔 LSD 在寻路和路由算法中的应用

现在, 我们可以获取到两节点的链路稳定间隔, 然而要选取最佳的路径, 我

们需要知道整个路径的稳点间隔，也就是从源节点到目的节点的路由稳定间隔。

对于一条完整的路由来说，从源节点到目的节点之间任意一跳路径的失效都会引起整条路由的断裂。路由从某一跳节点发送数据分组的时候，若 MAC 层发现到达下一跳节点的链路已经断裂，就会向网络层报告该链路失效，网络层就会发起新的路由寻找和计算过程。因此，对于整条路由来说，路由中的每一跳链路的稳定间隔的最小值决定了该条路由的整体稳定间隔。完整路由的 LSD 可以由以下计算公式得到：

$$LSD_{total} = \min(LSD_{ij}) \quad (3-4)$$

上式中， LSD_{total} 表示整条路由的链路稳定间隔， LSD_{ij} 表示路由中两个相邻节点 i 和 j 的链路稳定间隔。

因此，在应用最短路由算法计算当前节点的路由表时，可以应用上述公式作为一个权重因子来决定最短路径，除了考虑节点间的距离损耗因子（跳数），还需要比较路由的整体稳定间隔，即确保 LSD 值总体最大的那条路径才是更加稳定的选择。Dijkstra's 最短路径算法中，是按照跳数递增方式来遍历查找最佳路由的，在算法执行中引入 LSD 因子，当计算到达同一目的节点的距离损耗相同的多条路径时，考虑当前多条路径的整体稳定间隔 LSD 值，选取 LSD 最大的那条作为最佳路径，继续算法执行。按照该算法，将得到全网的稳定路由表。

传统的链路状态协议的控制路由消息的广播是采用周期性的广播方式，固定的发送间隔在某些情况下可能会产生链路中断时间变长或太过频繁的广播造成无线带宽资源的浪费。因此，借鉴基于 GPS 地理位置预测的路由协议的特点，可以利用预测的链路稳定间隔 LSD 作为决定每个节点发送广播控制路由数据周期的因子。考虑到每个节点都掌握和邻居节点的链路稳定间隔，因此可以根据公式（3-4）的结论，找到一个最小的 LSD 值作为节点的发送周期，为了防止网络稳定间隔预测的延误情形，设置一个固定间隔作为发送周期的上限，整体的 LSD 值大于该上限时，将上限值作为发送周期，这样就可以结合固定周期和随机周期的优势，充分利用收集的位置信息，做出最佳决断，节约带宽资源。

路由协议的实现中会定一个固定的周期作为路由控制消息的更新间隔，将当前节点维护的到达全网其他节点路径的 LSD 值最小的那个，作为当前节点广播路由控制消息的间隔，比较该 LSD 值与固定周期的大小，如果 LSD 值大于固定周期，则折中将固定周期作为广播间隔，避免节点长时间不做路由更新动作。

3.2.3 路由表算法

每一个节点都会独立维护全网的拓扑结构，定期的更新网络拓扑的变化。本

路由协议主要是依靠节点广播一条的握手 Hello 消息和多跳拓扑 TC 消息控制信息来通知网络中其他节点网络的变化。因此，路由表的计算依赖于邻居列表和拓扑表提供的信息。掌握了全网的拓扑消息，为了找到当前节点到网络中其他节点的最短路径信息，可以基于经典的戴克斯特拉算法(Dijkstra's Algorithm)的思想，结合考虑路由总损耗和路由稳定间隔 LSD 两个权值进行决策最短路由。路由表的算法思想描述如下：

当邻居列表和拓扑列表发生变化后，清空原路由表，按照下列步骤重新计算新的路由表。

步骤一：清空路由表。

步骤二：将所有邻居节点添加到路由表中，目的节点和下一跳地址均是邻居节点的地址，跳数为 1，路由 LSD 值等于该链路的 LSD 值，即 $R_dest=nb_addr$, $R_next=nb_addr$, $R_hops=1$, $R_LSD=nb_LSD$ 。

步骤三：循环从 $h=1$ 开始，每一次循环 h 加 1，直到 h 大于网络中节点总数，停止循环，路由表算法结束。

循环操作的内容是，对每一个在拓扑列表 $topologyset$ 中的元组执行下列操作：

首先，如果 $T_last==R_dest_1 \ \&\& \ R_hops_1==h$ ，令 $LSD=\min(T_LSD, R_LSD_1)$

然后，如果 $T_dest!=R_dest_2$ ，添加新的路由项：

$R_dest_new=T_dest$;

$R_next_new=R_next_1$ (the route entry whose R_dest is equal to T_last) ;

$R_hops_new=h+1$;

$R_LSD_new=LSD$;

最后，如果 $T_dest==R_dest_2 \ \&\& \ R_hops_2 \geq h+1 \ \&\& \ LSD > R_LSD_2$ ，更新原有路由项：

$R_hops_2 = h + 1$;

$R_LSD_2 = LSD$;

其中： R_dist 代表路由表项中目的节点地址， R_next 代表下一跳节点地址， R_hops 代表路由， R_LSD 代表总跳数路由 LSD 值。 nb_addr 代表邻居列表中节点地址， nb_LSD 代表邻居链路 LSD 值。 T_dest 代表拓扑表中元素目的节点地址， T_last 代表元素上一跳节点地址， T_LSD 代表拓扑表元素链路 LSD 值。 R_dest_1 代表新建的路由表项中的元素， R_dest_2 代表原路由表项中的元素。

该算法采用的是广度优先搜索，对于当前节点而言，将所有的邻居节点作为一跳范围内的最短路径，其路径损耗为一跳间隔，路径的 LSD 值即该链路的 LSD

值。接着寻找两跳范围内的最短路径，遍历拓扑表结构中剩下的所有节点，在一跳路径的基础上，根据表中的数据，更新两跳路径的路径损耗和 LSD 值；当到达目的节点的新路径和原路径具有相同的跳数损耗，则比较两跳路径当前的 LSD

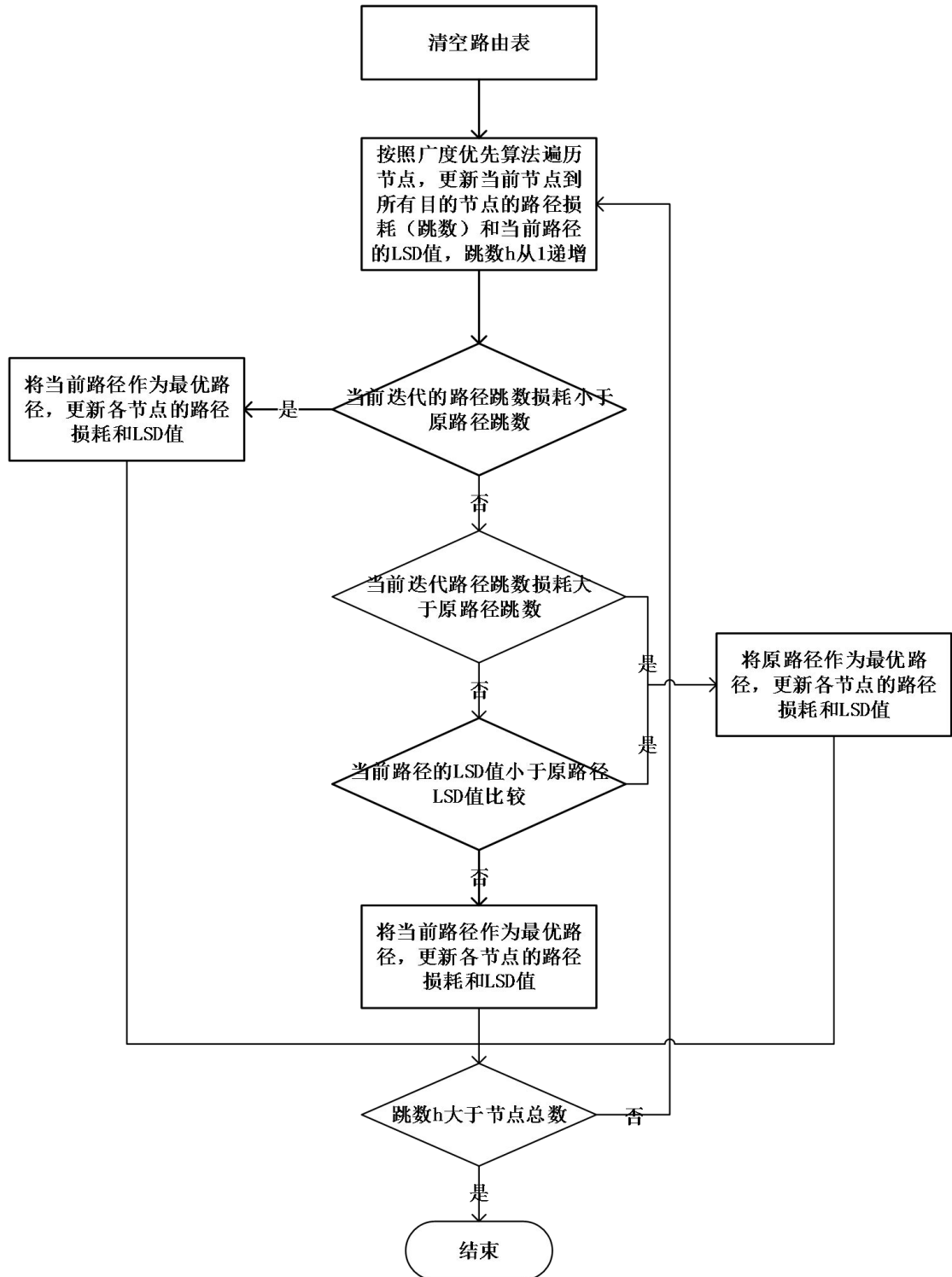


图 3-2 路由表算法流程图

值，当前 LSD 值大，说明该路径根据预测结果是更加稳定的路径，因此选择 LSD 值大的路径作为本次迭代的最佳路径。将路径损耗跳数继续增加，继续

上述的计算和比较更新，直到拓扑表结构中的所有节点都找到最佳的路径为止，路由算法结束。通过回溯结果可以得到当前节点到达网络中其他节点的准确的路由信息，形成路由表结构，包含到达的目的节点 IP 地址，下一跳节点的 IP 地址，总的跳数损耗及路径的 LSD 值。具体的流程如图 3-2 所示。

该算法主要是借鉴 Dijkstra 的算法思想，在选择路由的阶段，考虑了路径的 LSD 值和路径损耗（跳数），选择一条跳数最小并且 LSD 值最大，即预测最稳定的链路，路径损耗跳数值作为第一因子，在第一因子相同的多条路径中，比较路径的 LSD 值，选择 LSD 值最大的路径作为源节点到该目的节点的最佳路径。

3.3 NS2 仿真平台

为了验证路由协议的可行性，需要先在实验环境中进行路由协议的仿真，通过对仿真结果的具体分析，了解路由协议的性能和缺陷，不断的修改和优化协议设计，使其满足在高速移动场景的需求。在众多的仿真平台中，选择功能强大并且是开源的网络仿真器平台 NS2。

NS2（Network Simulator 2），即网络仿真器。NS2 是基于 REAL 网络仿真器的原型，最初是由伯克利加州大学的网络研究组开发的，后来由卡内基梅隆大学的 MONARCH 项目组进行扩展，提供了对无线自组织网络的支持。由于 NS2 是开源的仿真平台，而且强大的功能，因此被广泛的应用在网络仿真领域。

无线多跳网络路由协议模块是基于 NS2 平台的仿真。仿真的主要构件是系统平台提供的 MobileNode 类，即移动节点类。该类中包含了无线网络仿真的所有组成部分，结构如图 3-1 所示，详细说明如下：

MobileNode 类继承自 **Node** 类，因此包含了 **Node** 类的所有公共成员。

Node Entry：节点的入口，所有节点的处理逻辑都是从该入口开始，收到的数据包先通过该路口；

Classifier：分类器，作用是通过 IP 地址和端口号的分类规则决定收到的数据包该传给哪个组建，起到数据包的分发作用；

Agent：代理类，主要是应用层代理和路由层代理。应用层代理如 CBR 负责数据的产生，而路由层代理实现路由协议相关的策略和路由表计算和维护逻辑，本次仿真最重要的组成部件就是实现了一个 LS 路由协议代理类，负责处理路由表的计算和数据包的路由寻址转发业务。

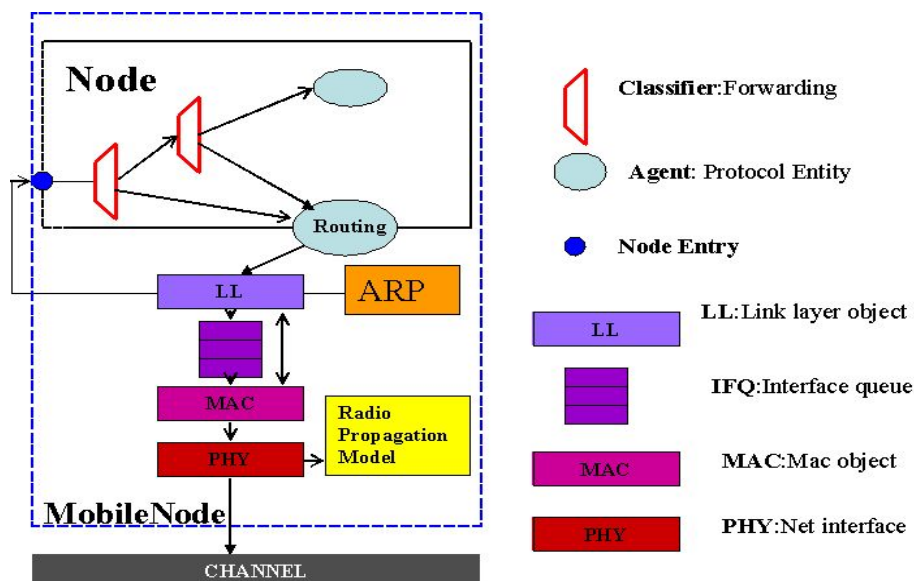


图 3-3 无线节点模块结构

LL：链路层类，基本上，所有下行的数据包都由路由层向下发给 LL 层，而所有的上行数据包都由 MAC 层向上发给 LL 层进行中转；

ARP：链路层的附加功能，地址解析协议类，用于根据 IP 地址映射物理地址；

IFQ：接口队列（Interface Queue），用于缓存 LL 层需要下发给 MAC 层的数据包，采用优先队列的策略，对数据包进行过滤和缓存；

MAC：媒体接入层类，主要负责实现物理层的相关协议，这里主要用到 TDMA 协议；

PHY：网络接口（Network Interface），作为接入信道的物理接口；包含碰撞处理和能量模型的策略实施；

Radio Propagation Model：能量传播模型，这里主要用到 TwoRayGround 能量模型策略；

Antenna：天线模型，这里主要用到全向天线模型；

路由协议模块的仿真需要在 NS2 平台的架构层次上，添加一个全新的路由模块。因此需要涉及到对 Otcl 层次的配置和 C++ 层次的路由模块的程序编写工作。主要实现的结构是：LS 路由层协议代理类，负责路由的发现，更新和维护工作，并计算每个节点的路由表，处理数据分组的接收、处理和转发。为上层和底层提供数据通信接口。

3.3.1 路由协议关键结构说明

路由表项是路由表中的路由单元，保存的是当前节点到达网络中其他节点的路径信息，包括目的节点地址，下一跳地址，总条数（路径损耗）和路径 LSD

值。该表项是路由协议中最关键的数据。如下表：

表 3-1 路由表项结构

目的节点地址 R_dist	下一跳节点地址 R_next	总跳数 R_hops	路由 LSD 值 R_LSD
------------------	-------------------	---------------	-------------------

目的节点地址是当前节点可达的网络中其他节点的 IP 地址；下一跳节点地址说明了要到达目的节点，从当前节点转发的数据包要传递的下一个节点的地址；总跳数表示从当前节点到达目的节点需要经过的中间节点的个数；路径 LSD 值是该条路径预测的稳定连通的时间间隔。每当节点确认需要转发数据分组，需要根据数据分组需要前往的目的节点，查询路由表中的信息，以确定数据的下一跳节点的地址，将数据转发出去。

邻居列表是维护与当前节点直接相连的节点地址和相关的链路信息的结构，表示当前节点的天线直接连通的邻居节点的信息，所有的数据分组都需要经由邻居节点的转发才能到达网络的其他部分。该结构是计算路由的一个关键数据结构，如下表所示：

表 3-2 节点邻居列表信息

邻居节点地址 nb_addr	链路 LSD 值 nb_LSD	有效时间 nb_time
----------------	-----------------	--------------

邻居节点地址表示和当前节点直接连通的节点 IP 地址；链路 LSD 值说明了当前节点和该邻居节点间的链路预测稳定间隔；有效时间表示的是该邻居节点信息的有效时间范围，超时后，该邻居自动失效。

拓扑结构元组是包含整个网络拓扑结构，各节点间连通性信息，及用于辅助路由计算的数据结构。该结构包含当前节点掌握的关于全网拓扑的信息，任意两点的连通性可以查询该表得到。表中包含连通的两节点的地址，两节点链路的 LSD 值，收到该消息的序列号及该表项的有效时间。如下表所示：

表 3-3 网络拓扑结构元组信息

目的地址 T_dest	上一跳地址 T_last	两节点间 T_LSD	序列号 T_seqnum	有效时间 T_time
----------------	-----------------	---------------	-----------------	----------------

这里的地址是网络中节点的 IP 地址，上一跳地址说明需要经过该节点到达目的地址 T_dest，即表明目标地址 T_dest 和上一跳地址 T_last 是连通的。T_LSD 是两节点间的链路稳定间隔。序列号表明最近一次收到的消息序号，用于防止重复或旧数据的污染。有效时间是这条数据的有效时间范围，过期后则失效。

3.3.2 路由协议的仿真分组格式

在该协议的设计中,对本协议所设计的所有数据结构采取统一的分组格式进行通信,如此更便于路由功能的扩展和强化,以及版本的升级和向后兼容性等。每个分组封装一条或多条消息,每个消息共享一个公共的分组头部,以便各节点能够正确的接收和转发未知类型的消息分组。消息可以洪泛到整个网络中,或者是某个直径范围内,直径可以由距离本消息源节点的跳数来定义。当洪泛任一路由控制消息时,由于每个节点都会维护一个用于防止对同一个路由控制消息发送两次或者两次以上的重复分组集合,因此可以排除掉重复消息的复制和转发,让路由控制消息在整个网络中的重传次数降到最低的程度。此外,一个节点可以通过检查消息头,以便获取有关到达该消息源节点的跳数信息。如表 3-4 是本协议中分组的统一格式。

公共分组头部结构包括分组长度和分组序列号两部分。分组长度 (Packet Length) 该域为 16 比特,表示的是整个分组的字节长度。分组序列号 (Packet Sequence Number, PSN) 该域有 16 个比特。当每发送一个新的路由控制消息分组时,会将分组的序列号 PSN 加 1,作为新旧重复分组的标识。

消息头部由消息类型,消息接收有效时间,消息的长度,消息的源节点地址,消息发送的有效生命期,跳数和消息序列号组成。

消息的类型 (Message Type) 该域长度为 8 比特,用来说明后面跟着的“消息”的类型信息,消息类型编号为 0-127。在本协议中只用到两种类型,握手 Hello 消息类型和拓扑 TC 消息类型。

表 3-4 路由协议通用分组格式结构

0Byte	1Byte	2Byte	3Byte
分组长度（Packet Length）		分组序列号（Packet Sequence Number）	
消息类型（Type）	接收有效时间	消息长度（Message Length）	
消息源节点地址（Originator Address）			
消息生命期 TTL	跳数（HOP）	消息序列号（MSN）	
消息具体的内容			
消息类型（Type）	接收有效时间	消息长度（Message Length）	
消息源节点地址（Originator Address）			
消息生命期 TTL	跳数（HOP）	消息序列号（MSN）	
消息具体的内容			

接收消息的有效时间 (Valid Time) 该域长度为 8 比特,说明一个节点必须考虑在收到该消息后,此消息的有效时间期限,除非在期限内收到一个更近的更新,否则过期该消息将不再有效。有效时间用该字节的高 4 比特和低 4 比特共同

表示，可以按照如下规则计算：

$$Vtime = C \left(1 + \frac{a}{16} \right) \times 2^b (s) \quad (3-5)$$

其中，a 和 b 分别表示该域高 4 比特和低 4 比特表示的整数，C 为一个比例因子，可以采用 1/16s，即 0.0625s。

消息的长度（Message Length）字段的长度为 16 比特，表示本条消息的字节总长度，从消息类型开始计算到下一条消息的开始位置。

消息的源节点地址（Originator Address）字段为 32 比特，表示产生该条消息的源节点的 IP 地址，在整个传输过程中是不变的。

消息发送的有效生命期（Time To Live, TTL）字段为 8 比特，说明本消息能够被网络转发的最大跳数。中间的节点收到该消息时，先检查其 TTL 域的值，如果是 0，则在任何条件下都不进行转发，直接丢弃该消息；否则的话，就将 TTL 值减 1，再转发该消息。TTL 值可以用来控制消息在网络中洪泛的范围，降低网络开销。因此可以根据网络的规模来调整该值。

跳数（HOP Count）字段为 8 比特，说明该消息已经被转发的跳数。中间节点每转发一次该消息，就将该字段加 1，再转发。

消息序列号（Message Sequence Number, MSN）字段为 16 比特。节点每产生一个消息时，就会给该消息分配一个唯一的标识码，并将这个号码插入到本条消息的消息序列号字段中，此后，每产生一个消息，就将序列号加 1。消息序列号可以用来保证一条消息不会被其他节点转发两次以上，避免不必要的开销。

3.3.3 路由协议的操作

中间节点收到一个数据分组后，先检查消息的头部，根据消息类型的取值决定该消息的最终结果。因为分组的洪泛传播，一个节点可能会多次接收到同一条消息，为了避免消息被重复接收或处理，每个节点都需要维护一个最近所收到的消息的记录到重复列表中。

当一个节点收到一个数据分组后，对其中封装的每一条消息需要做如下的处理：一、如果该分组没有包含任何消息，即分组的长度小于或等于分组头部的长度，则丢弃该分组。二、如果该消息的 TTL 域的值为 0，或者本条消息的源节点地址等于当前节点的主地址，说明是本节点产生的数据，直接丢弃。三、消息处理分为以下步骤：1) 如果当前节点的重复分组集合中存在一个元素，它的源节点地址和本消息的源节点地址相同，并且他们的消息序列号也相等，说明该消息已经被处理过了，无需在进行处理。2) 如果当前节点认可本条消息的消息类型，则按照所属类型的规范来处理本消息。

当前节点处理消息转发流程为：1) 如果当前节点的重复分组集合中存在一个分组，它的源地址和本条消息的源节点地址相同并且消息序列号也一样，说明分组已经被转发过，无需再进行转发。2) 否则如果当前节点认可本消息的消息类型，则按照消息类型规定的处理流程处理对应的消息。3) 如果当前节点不认可本条消息的类型，说明是其他应用数据，当前节点只需按照默认的转发算法来处理该消息即可。

默认转发算法的思想是：1) 如果在当前节点的路由表中没有找到该消息需要发往的目标地址的路由信息，则算法终止，消息不必转发。2) 如果当前节点的重复分组集合里不存在源地址和本条消息的源节点地址相同并且消息序列号也一样的元素，则考虑对分组进行转发。3) 转发分组时，需要更新重复分组集合的数据，将本条消息的 TTL 减 1，将本条消息的跳数加 1，最后将本条消息广播到路由出口队列里。

以上是网络中的节点对接收到的分组进行数据的检查，处理和转发的过程。此外，该路由协议模块自身也会产生两种类型的消息，一种是用于完成链路探测、相邻节点探测的握手 Hello 消息，另一种是用于完成拓扑结构声明的拓扑控制 TC 消息。当前节点收到 Hello 握手消息时，根据 Hello 消息中带有的源节点地址和运动信息，更新当前的邻居列表数据。当前节点收到 TC 拓扑控制消息时，需要根据消息中包含的节点地址信息和位置信息更新拓扑列表。

3.5 NS2 仿真验证场景设计

在完成路由协议在 NS2 网络仿真平台上的模块实现后，需要通过设计仿真验证场景来测试实现的路由模块的功能和性能。通过设计不同的场景，节点数由少增多，通信方式由简到繁，运用于实现的路由协议之上，根据得到的 trace 文件，分析仿真的结果，根据路由协议质量评价指标丢包率和端到端时延两个参数来衡量路由协议的可行性和可靠性。

丢包率是指从源节点发送的数据分组（本文仿真中即 CBR 数据分组）与在传输过程中丢失的分组数的比的倒数，该指标针对应用层，因此能够有效反映路由有效性和网络服务质量。

平均端到端时延是指数据分组从源节点到达目的节点经过的总时间，

$$\text{平均端到端时延} = \frac{1}{N} \sum_{i=0}^N (rt_i - st_i) \quad i=0,1,2,\dots,N$$

可以检验路由建立是否满足项目需求。

公共仿真参数的设置如下：

表 3-5 仿真场景通用参数

参数类型	参数值
传输信道类型	Channel/WirelessChannel
能量传播模型	Propagation/TwoRayGround
物理信道模型	Phy/WirelessPhy
物理层协议类型	Mac/Tdma
IFQ 队列类型	Queue/DropTail/PriQueue
IFQ 队列长度	50
路由协议类型	LS
链路层类型	LL
天线类型	Antenna/OmniAntenna
模拟空间大小	500km
节点有效通信半径	10km
节点载波监听有效半径	12km

仿真场景一：两个节点相对运动，验证高速移动场景下节点能够进行通信；

场景参数：节点 0 和节点 1 同向运行，节点 0 向节点 1 发送 CBR 数据包；在这过程中，两节点脱离通信半径 17 秒中后重新进入相互的通信半径。仿真时间为 200 秒。

表 3-6 仿真场景一具体参数

参数	节点 0	节点 1
运行速率	5km/s	7km/s
数据包大小	512Byte	0Byte
发送间隔	0.1s	0.1s

通过对 Trace 文件的分析可以看出：在通信过程中，节点 0 CBR 总发送数据包为 1899 个，节点 1 收到 CBR 数据包 1714，丢包 185 个，仿真十次，取平均丢包率为 9.7420%。可以看出完全符合场景设置，两个节点断开通信 17 秒，按照 CBR 速率，理论应丢失 170 个数据包，分析 Trace 的结论和理论相近。还计算得到通信的端到端平均时延为 0.043949 秒。说明该协议能够在高速移动场景下正常通信。

仿真场景二：三个节点相对运动，验证多跳传输及节点脱离后再次连入网络仍能自组；

场景参数：起初节点 2 通过节点 1 向节点 0 发送 CBR 数据包，然后节点 1 断开与节点 0 的连接，节点 2 与节点 0 失联 31 秒；接着节点 2 移动到节点 0 的通信半径内，节点 2 与节点 0 重新建立连接；节点 0 通过节点 2 向节点 1 发送 CBR 数据。

表 3-7 仿真场景二具体参数

参数	节点 0	节点 1	节点 2
运行速率	0km	5km	7km
数据包大小	512Byte	0Byte	512Byte
发送间隔	0.1s	0s	0.1s

通过对 Trace 文件的分析可以看出：在通信过程中节点 2 向节点 0 发送 CBR 数据包共 1982 个，节点 0 收到数据包共 1655 个，丢失 327 个，仿真十次取平均得到平均丢包率为 16.4985%，由于失联 30 秒，理论丢包 300 个，实际仿真数据和理论相近，端到端平均时延 0.555476 秒。节点 0 向节点 1 发送 CBR 数据包 304 个，节点 1 收到 208，丢失 96 个，由于失联 2 秒加上中间节点 2 的缓冲溢出 7 秒，理论上丢包 90 个，实际仿真和理论相近，端到端平均时延为 2.390935 秒。可以说明本协议支持多跳传输，并且节点离开或重新加入网络都能够使网络自组织成功。

仿真场景三：4 个节点相互形成多跳自组网，然后一个节点脱离，验证剩下的节点仍能重新自组网络。

仿真参数：4 个节点相对运动，节点 0 经过中间节点 1，节点 2 向节点 3 发送 CBR 数据包。然后节点 2 脱离网络，节点 0，节点 1，节点 3 重组网络并继续数据传输。

表 3-8 仿真场景三具体参数

参数	节点 0	节点 1	节点 2	节点 3
运行速率	5km/s	7km/s	3km/s	5km/s
数据包大小	512Byte	0Byte	0Byte	0Byte
发送间隔	0.1s	0.1s	0.1s	0.1s

通过 Trace 文件的分析可以看出：通信过程中，节点 0 向节点 3 发送 CBR 数据包共 1979 个，节点 3 收到数据包 1972 个，丢失 7 个，丢失原因是中间节点 2 离开后，重新组网过程中因找不到路由而丢失，当网络重新组建，数据继续传输。丢包率为 0.45477%，符合正常稳定网络的要求。端到端平均时延为 0.125885 秒。可以说明该协议能满足节点脱离后剩余节点重新快速自组网络的需求。

仿真场景四：分四批次发送十个节点，验证网络的组网，以及多跳传输性能。

场景参数：其中四号节点和六号节点会在中途停止，模拟节点的退出。

表 3-9 仿真场景四具体参数

批次	第 1 批	第 2 批		第 3 批			第 4 批		第 5 批	
节点	节点 0	节点 1	节点 2	节点 3	节点 4	节点 5	节点 6	节点 7	节点 8	节点 9
发送	0s	75s	75s	150s	150s	150s	225s	225s	300s	300s
速率 km/s	5	5	5	5	5	5	5	5	5	5
数据	512	512	512	512	512	512	512	512	512	512
间隔	0.1s	0.1s	0.1s	0.1s	0.1s	0.1s	0.1s	0.1s	0.1s	0.1s

3.6 仿真结果分析

图 1 所示的是场景四下的丢包率。横坐标表示节点间通信的跳数，纵坐标表示的是 cbr 包的丢包率。在一跳的情况下，收到 5236 个数据包，丢失 1 个；两跳的情况下，收到 4504 个数据包，丢失 1 个，丢包率 0.022%；三跳的情况下，收到 3739 个数据包，丢失 2 个，丢包率 0.053%；四跳的情况下，收到 2960 个数据包，丢失 27 个，丢包率 0.90%。丢包率基本随着跳数增加而增加。

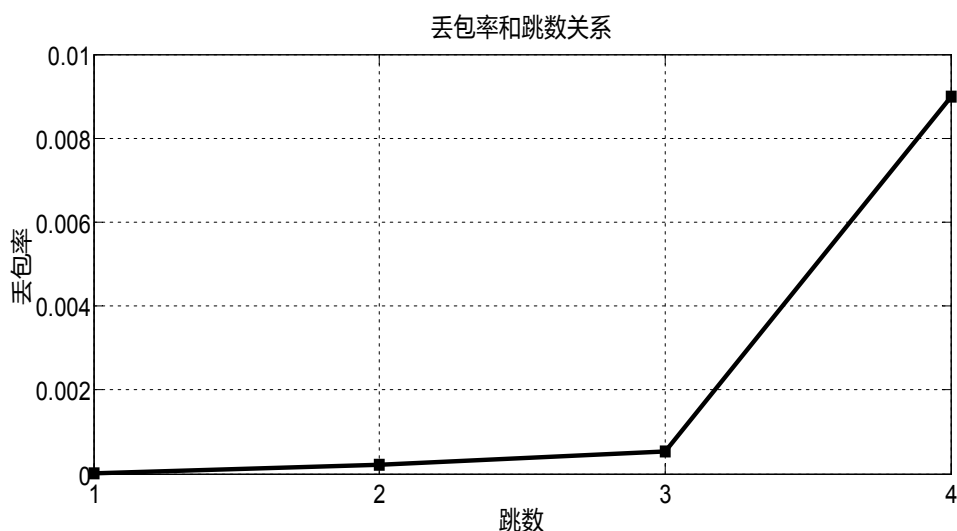


图 3-4 丢包率与跳数的关系

多跳情况下，端到端时延如下图所示。其中一跳的平均时延 0.044664s，两跳的平均时延 0.120218s，三跳的平均时延 0.200751s，四跳的平均时延 0.332521s。

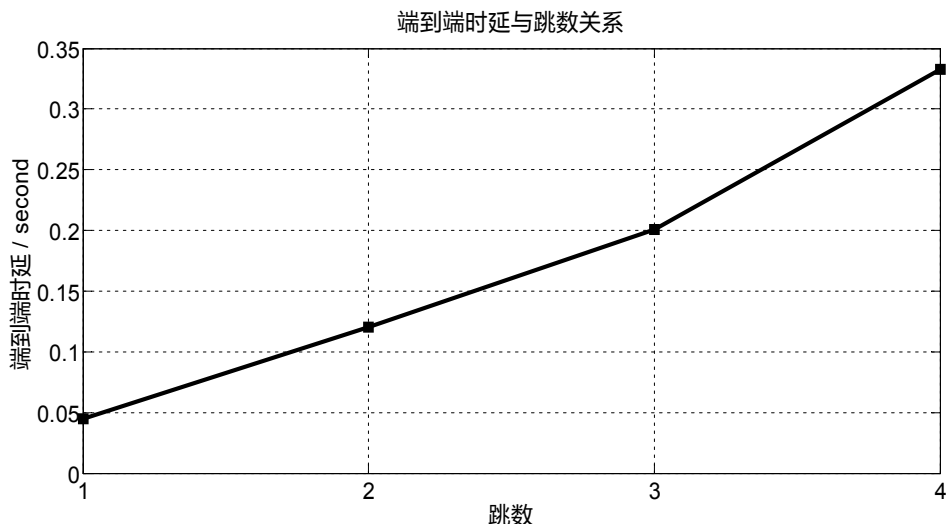


图 3-5 端到端延时与跳数关系

从上图中可以看出，端到端的延时随着数据分组经过的中转节点数的增加而增加，一跳情况下得端到端平均时延为 45 毫秒，满足协议的性能要求。

通过仿真验证，本方案能够满足项目设计要求，在场景四下，丢包率为 0.182%，端到端平均时延是 0.045s。

通过仿真验证，该方案在高速移动场景下节点能够进行组网，多跳传输及节点脱离后再次连入网络，满足节点脱离后剩余节点重新快速自组网络的需求，网络稳定后的平均丢包率和通信的端到端时延能够满足需求。

3.7 与传统 OLSR 协议的性能比较

传统的 OLSR 路由协议是在链路状态（Link State）协议的基础上改进而来的，它通过在全网范围内周期性的广播链路状态和拓扑结构变化，试图让所有的节点掌握全网的连通信息。当有数据分组到来，需要寻找路由，节点会根据自身掌握的关于全网的拓扑图结构，利用 Dijkstra 算法，来获取到达目的节点的最佳路径。OLSR 路由协议具有如下特点：一是缩减链路状态更新消息的分组大小，只包含某节点的邻居节点子集；二是限制了链路状态更新消息的重传范围，只有选中的邻居节点才会担当重传链路更新分组的任务。OLSR 采用上述两种方式来提供洪泛消息的有效机制，明显地减少了网络的路由开销，尤其适用于大型密集网络结构。OLSR 使用多点中继来进行洪泛，而且洪泛的节点又利用各自的洪泛节点进行重传，只有多点中继（MultiPoint Relay, MPR）节点作为中间路由节点。

而本文设计的路由协议是结合了链路状态协议和 GPS 辅助定位技术的优点，在 OLSR 路由协议的基础上进行的改进，使路由协议能够满足高速移动的小型无线多跳网络的场景需求。本文充分利用 GPS 定位设备提供的节点位置和运动数

据，引入链路稳定间隔时间（LSD）的预测，利用 LSD 来辅助寻找稳定链路和计算广播分组的周期间隔，提高全网拓扑的预测能力和减少路由开销。

在控制网络路由开销方面，每个节点的路由控制消息的广播周期间隔，将根据节点计算的与邻居节点间的 LSD 值来决定；当前节点预测到与邻居节点发生链路中断，即链路维持时间超出 LSD 预测值时，当前节点将把自身的最新的拓扑消息广播到网络上，以便其他节点更新自身的拓扑结构；为了防止长时间的稳定链路预测的可靠性问题，设置一个定期广播间隔，当预测的 LSD 大于定期广播间隔时，以定期广播间隔作为广播路由控制信息的间隔。这种方案比传统的 OLSR 中的定期广播方式，更加灵活，自适应能力强，根据网络拓扑结构的变化情况，预测链路消息的广播间隔，控制网络路由开销。

在辅助稳定链路决策方面，传统的 OLSR 协议利用 Dijkstra 算法来计算最短路由，只考虑跳数作为路径损耗的因子，在计算中出现多条相同路径时，一般只能选取第一条作为最佳路径。而本文引入了 LSD 作为路径损耗因子的一部分，整个算法以跳数损耗作为主因子，LSD 作为辅助因子。在路由计算过程中，遇到同时存在到目的节点的多跳路径时，将 LSD 作为决策因子，比较每条路由的 LSD 值，由于 LSD 值代表了整条路由的预测稳定时间，因此可以选择 LSD 最大的那条路由作为最佳路径存入节点的路由表中。这种方案可以基于预测 LSD 因子，决策出最稳定的路由，比传统 OLSR 采用的 Dijkstra 算法，更加灵活、可靠。

上一节通过场景的设计及仿真结果的分析，验证了路由协议的可靠性和可行性。本节将对引入 LSD 后的改进 OLSR 协议和传统 OLSR 协议在高速移动场景下进行仿真对比，比较两个协议的性能差别。本节主要比较两个协议在不同移动速度和不同节点数目的场景下，控制路由消息的开销和路由稳定性方面的差异。

路由控制消息的开销可以通过统计仿真过程中，所有节点广播的路由控制消息的数量来衡量。而路由稳定性方面，可以通过统计链路断裂次数来衡量。

3.7.1 仿真参数说明

本次仿真主要是针对在不同的移动速度和不同节点数目两个场景下，两个路由协议在网络控制开销和路由稳定性方面的比较。NS2 平台无线多跳网络仿真的通用参数同表 3-1 所示，仿真时间均为 1000 秒，节点移动速度比较高，以随机的方向运行。具体仿真参数如下：

（1）移动速度

节点个数和通信半径固定，移动的速度进行设置

节点个数：10 个

通信有效半径：10km

节点移动速度：50m/s, 100m/s, 150m/s, 200m/s, 250m/s, 300m/s

(2) 节点数目

节点间通信半径和节点的移动速度固定，对节点数目进行设置

节点移动速度：200m/s

通信有效半径：10km

节点个数：10, 20, 30, 40, 50, 60

传统的 OLSR 协议的拓扑消息广播间隔为 5 秒，而引入 LSD-OLSR 协议的最大定期广播间隔设置为 15 秒。节点的通信半径比较大，主要为了适应飞行器在空中的通信方式和飞行器的高速移动特性。节点的运动速度很高，满足对高速路由性能的测试。节点的数目控制在中小型规模。在移动速度不断升高和节点数目不断增加两种情况下，考察两个协议的性能差异。

3.7.2 仿真结果说明

对仿真结果 trace 文件进行分析，统计在每个场景下路由控制消息的数量和链路断裂的数量。路由控制消息的数量可以说明在广播间隔引入 LSD 因子后的 OLSR 协议与传统的 OLSR 协议，在网络控制消息的开销方面的差异。链路断裂的数量可以说明在计算路由的 Dijkstra 算法中引入 LSD 辅助因子后，路由稳定性方面的差异，统计结果如下图所示。

图中展示了 OLSR 路由协议和 LSD-OLSR 路由协议在 NS2 仿真平台上结果的分析，分别说明了路由控制消息数量和链路断裂次数在不同移动速度和不同节点数量情况下的比较。从图中可以看出，随着移动速度的加快，网络中节点的运动的变化率增加，从而增加了网络拓扑的变化速度。传统的 OLSR 路由协议广播控制消息的间隔是设定好的，因此，传统 OLSR 协议的路由控制消息数量随速度的变化不大，基本维持在一个平均水平。而 LSD-OLSR 协议路由控制消息的广播间隔是根据实时计算的 LSD 作为预测因子来动态决策的，在速度较小或节点相对位置变化不剧烈的情况下，可以有效的减少控制分组的洪泛，比传统 OLSR 协议效果好。在链路断裂次数方面，由于 LSD-OLSR 在 Dijkstra 最短路径算法中引入 LSD 因子，有效的帮助寻找稳定路由，从而降低了链路的断裂次数。

1. 移动速度变化

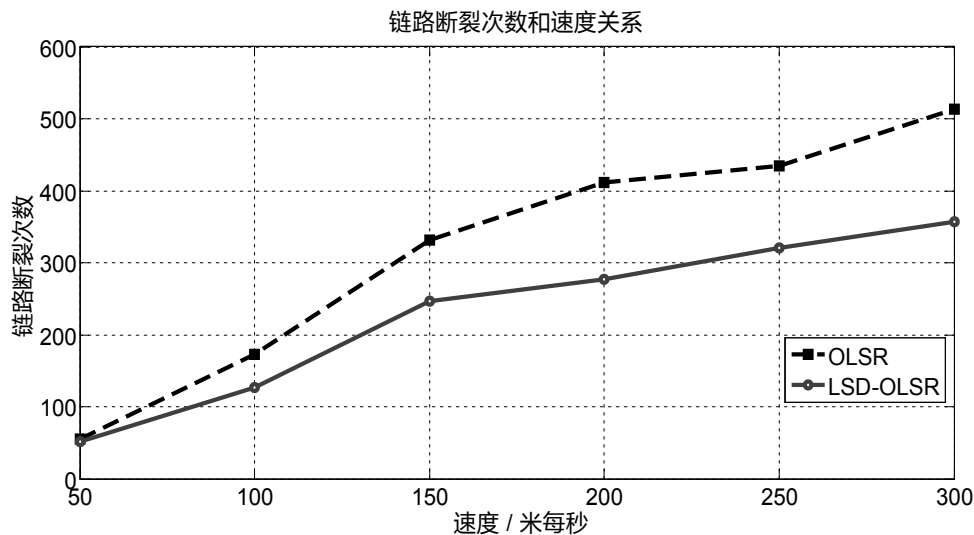


图 3-6 链路断裂次数与速度相关关系

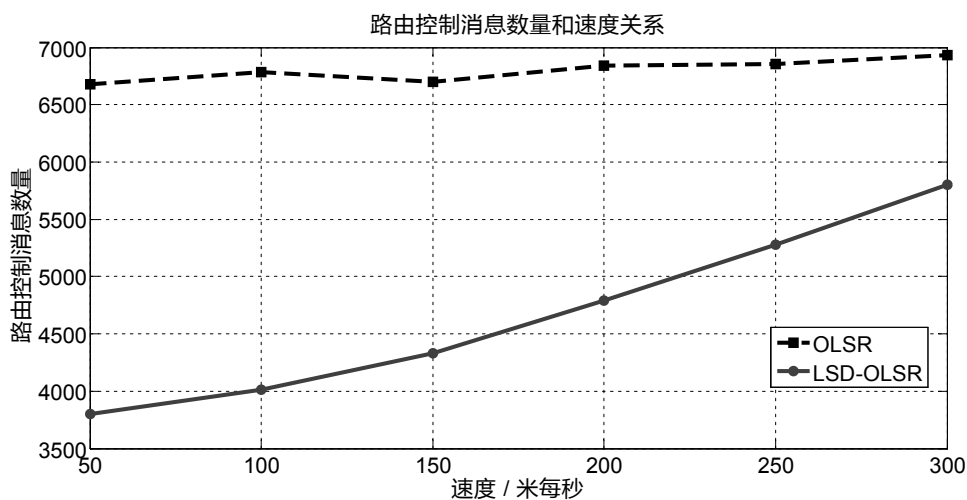


图 3-7 路由控制消息数量与速度相关关系

当网络中的节点数量增多时，分组经过的中间路由节点的数量和可能性都会增加，这导致了链路断裂次数的增加。由于 LSD-OLSR 协议引入了链路稳定间隔因子，在选择路由时比传统 OLSR 路由算法有所改进，从下图中可以看到 LSD-OLSR 的链路断裂次数少于传统的 OLSR。在路由控制消息数量方面，随着节点的增加，传统 OLSR 的路由控制消息的数量也随着成倍的增长，而 LSD-OLSR 协议的路由控制消息数量增长速度小于传统 OLSR 协议。

2. 节点数量变化

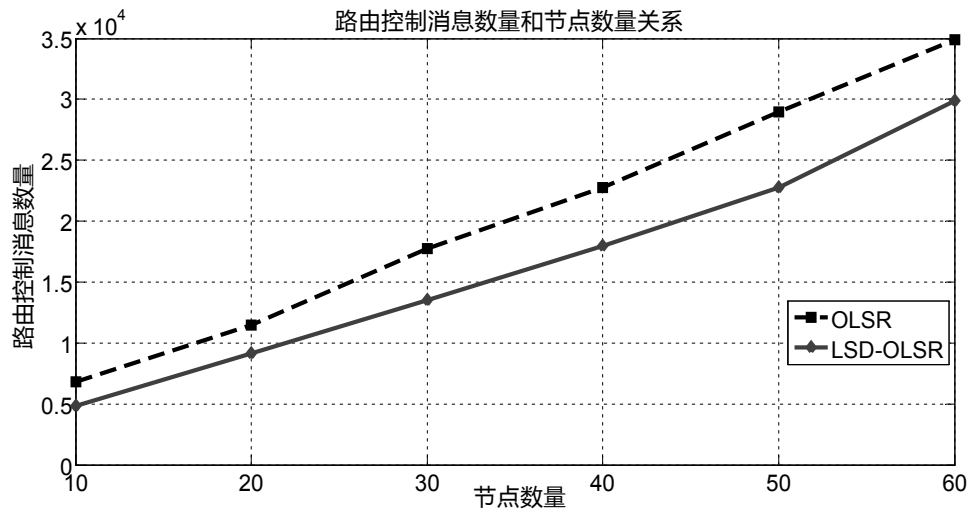


图 3-8 路由控制消息与节点数量相关关系

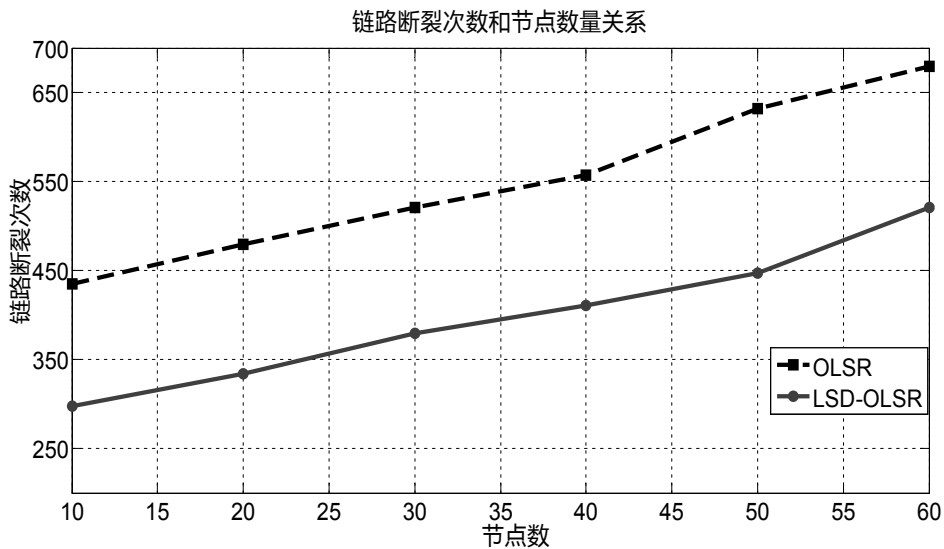


图 3-9 链路断裂次数与节点数量相关关系

从上述的结果分析可以看出，引入链路稳定间隔 LSD 因子的 OLSR 路由协议在链路断裂次数和路由控制消息数量上的表现都优于传统的 OLSR 协议，说明了改进后的协议更加适合高速移动的应用场景，增强链路稳定性，减少路由控制消息的开销。

3.8 小结

本章为小型高速移动的无线多跳网络场景设计了一种可行的路由协议，该协议结合了链路状态协议和 GPS 辅助定位的优点，对传统的 OLSR 协议进行改进。引入链路稳定间隔时间 LSD 因子，作为路由控制消息广播间隔和稳定路由选择

的决策因子，充分利用 GPS 设备提供的节点位置和运动数据，提高路由的稳定性和减少网络路由开销。在 NS2 仿真平台上，实现该路由模块，通过在不同场景下的验证，证明了该路由协议的可行性和可靠性。并通过在移动速度和节点数目变化两种情况下，比较 LSD-OLSR 协议和 OLSR 协议在路由开销和路由稳定方面的差异。

第四章 高速路由协议在 uC/OS-II 上的实现

上一章设计了适应高速移动场景的高速路由协议 LSD-OLSR, 并验证了该路由协议的可靠性, 和传统 OLSR 比较了性能的差异。本章需要在 NS2 仿真的基础上, 将该路由协议方案移植到 uC/OS-II 嵌入式操作系统上, 实现与其它模块的协调, 共同组成一个完整的无线多跳网络系统。路由模块的实现基于 uC/OS-II 操作系统, 主要负责系统网络层的功能, 包括路由的发现、维护和更新, 提供与 MAC 层的通信接口, 数据包的接收和处理, 及与上层应用层的通信接口。

4.1 uC/OS-II 嵌入式系统

uC/OS-II 是由 Micrium 公司开源的轻量级嵌入式操作系统, 是一个可移植的、可裁剪的、可固化、抢占式多任务的实时内核, 它适用于多种微型处理器、数字处理芯片与微型控制器。

μ C/OS-II 是专门为嵌入式应用设计的, 绝大部分代码是基于 C 语言编写的。CPU 硬件相关部分的代码是用汇编语言编写的, 长度约 200 行的汇编语言部分被压缩到最低限度, 目的就是方便于移植到各种不同的 CPU 上。用户只需要有标准的 ANSI 的 C 语言交叉编译器、汇编器、连接器等编译工具, 就能将 μ C/OS-II 系统嵌入到开发的产品中。 μ C/OS-II 具有执行效率高、实时性能好、占用的空间小和可扩展性很强等特点, 最小的内核可以编译至 2KB 左右。现今, μ C/OS-II 操作系统已经可以移植到几乎所有知名的 CPU 上了。

uC/OS-II 系统的目标是实现基于优先级调度的抢占式的实时内核, 并在这个内核的基础之上提供最基本的系统服务, 比如内存管理, 中断管理, 邮箱, 消息队列, 信号量等。

路由协议模块的实现就是基于该操作系统, 利用系统提供的消息队列, 内存管理, 邮箱等多种通信方式, 实现网络层功能在系统上的移植。

4.2 多跳系统架构

无线多跳网络系统是构建在 uC/OS-II 嵌入式系统之上的, 由操作系统构成的基础服务模块, 数据收发队列模块, 任务调度中心模块, 应用层, 网络层, MAC 层, 天线模块七个模块共同协作完成, 系统架构如图 3-1 所示。模块之间提供了相互访问的通信接口, 用以在上下层之间的服务和信息交换, 各个模块的功能相对独立, 完成自身的任务, 为其他模块的正确运行提供保障。

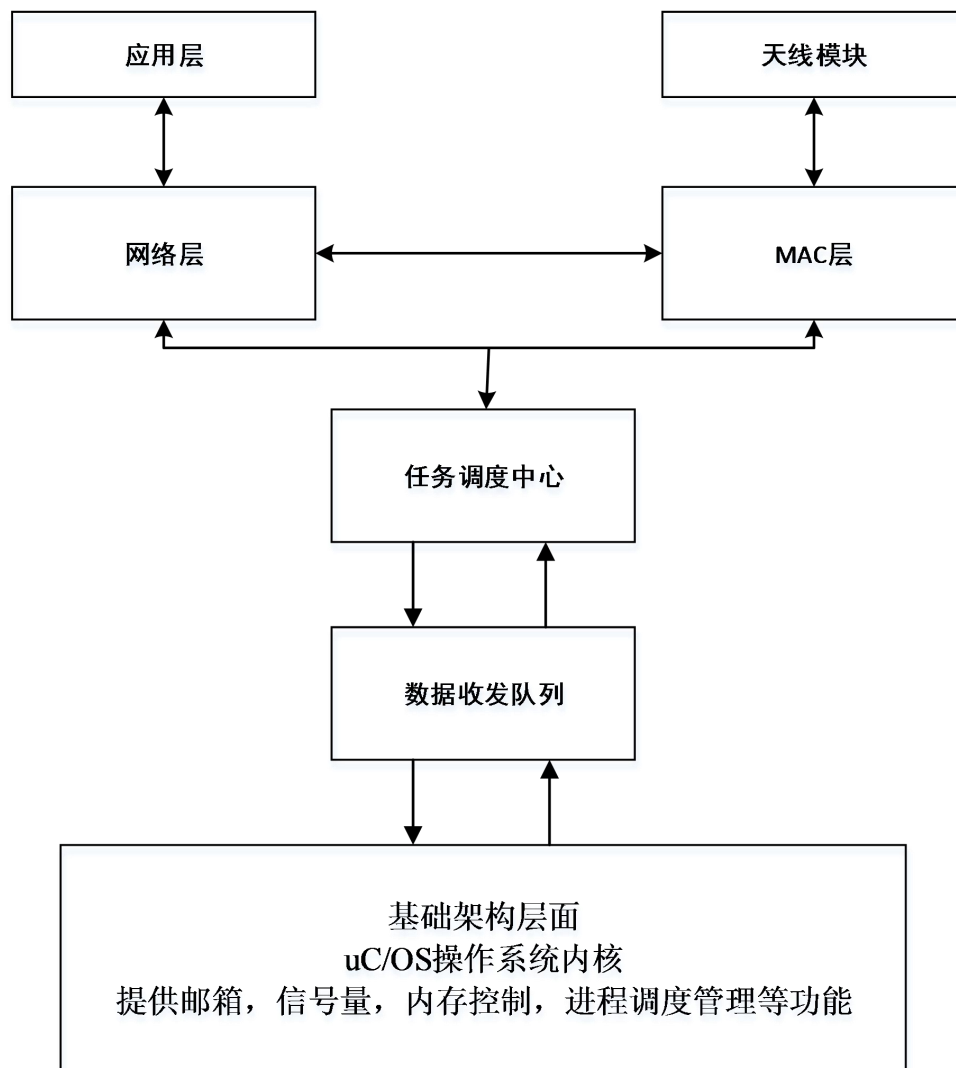


图 4-1 无线多跳网络系统架构图

基础服务模块的功能主要是对 uC/OS-II 嵌入式内核提供的基本服务的封装，嵌入式系统提供的邮箱，信号量，内存管理和进程调度等机制，都是底层运行的核心保障，对这些底层服务进行抽象封装，提供符合本系统其他模块调用的更加友好安全的应用程序编程接口，提供统一的基础服务功能接口，方便其他模块的使用，节省学习成本。

数据收发队列模块的功能主要是处理对网卡收到的数据包和从 MAC 层转发的数据分组的缓存消息队列，提供对分组的封装，排队和分类，供任务调度中心模块提取消息分组进行调度。

任务调度中心模块是整个系统的调度中心，所有的任务执行，包括网络层和 MAC 层的处理，都是通过任务调度中心通过对消息队列中的消息分组的解析，根据具体的事件类型，将分组分派到其他模块进行特定的处理。

网络层模块的功能主要是实现路由协议，为节点感知全网的拓扑结构的变

化,路由表的计算和维护,提供数据分组转发的路由信息,是数据转发中的最关键环节。该模块也是本文重点实现的模块,将设计的适应小型高速无线多跳网络环境的 LSD-OLSR 路由协议在 uC/OS-II 系统上实现。

MAC 层模块的功能负责底层链路控制功能,实现 TDMA 协议,掌握数据分组的接收和发送时机,控制天线模块的正确运行。

天线模块的功能主要是完成天线的接收和发送,控制天线的对齐,广播和点对点通信的不同功能模式的切换,完成分组的最终传输。

应用层模块的功能是根据不同的业务需求,处理数据分组,根据数据业务产生数据流等业务功能,提供端到端的可靠性等要求。

网络层主要提供三个接口,一个是 packetProcess()接口,负责处理应用层的数据分组;一个是 receiveFromMac()接口,负责处理来自 MAC 层的数据分组;一个是 computeRoutetable()接口,负责路由表数据的更新和计算。网络层通过这三个接口封装了 LSD-OLSR 路由协议,对外提供路由功能,为数据分组的转发服务。

4.3 网络层结构设计

在路由协议的仿真方案基础上,将路由协议的关键模块精简,考虑嵌入式系统的特性和其他模块之间的通信接口需求,封装路由协议的具体实现,提供一套完整的通信接口,供上层应用和 MAC 层调用。

网络层采用基于改进 IP 协议的无线多跳网络体系结构。改进点主要在 IP 数据报头的简化方面,减少网络开销,以节省无线环境带宽资源。无线多跳网络可以有平面式和分级式两种结构。平面式结构中,网络中所有节点的功能和地位相等,不存在瓶颈问题,网络比较健壮,并且节点的覆盖范围比较小,相对比较安全,主要适用于中小型网络,适用于本文讨论的高速小型网络场景。

网络层仅负责处理路由的建立、维护、更新和数据包的转发和传递功能,对于数据分组的检验和处理都交于应用层进行处理。因此简化后的 IP 报文头部结构如表 4-1 所示。

目标节点地址和源节点地址,该域各占 8 比特,由于网络规模较小,不需要传统 IP 协议 32bit 的地址空间。

数据长度,表示分组数据载荷的长度,从分组数据的起始位置算起,不包括 IP 数据固定头部大小。

生存时间,表示该分组数据在网络中的有效生存时间,这里是以跳数作为计算,分组被节点转发一次,该域的值减少一,当 TTL 等于 0 时,分组不在需要

被转发。该域可以有效的防止分组在网络中的泛滥。

表 4-1 简化后的 IP 头部结构

0(byte)	1(byte)
目标节点地址 (Destination Address)	源节点地址 (Originator Address)
数据长度 (Packet Length)	
生存时间 (Time To Live, TTL)	协议类型 (Protocol Type)
分组数据 (Packet Data) ...	

协议类型，该域占用 8 比特，用于区分上层接收的协议类型，起到区分不同协议或应用的业务的作用。

由于系统是在小型规模的高速无线多跳网络中应用，注重系统的稳定性，高效性。因此，传统 IP 协议的其他字段，在该应用场景中都不需要涉及，删除这些字段，可以有效的减少网络报文头部的开销，提高无线带宽资源的利用率。

4.4 网络层功能接口设计

网络层主要的任务是完成对路由协议的封装，提供统一的通信接口。而网络层的基本功能和接口主要可以分为以下三个：

- 1) 负责更新维护节点的路由表；
- 2) 负责从应用层接收数据，根据路由表确定下一跳地址，发送给 MAC 层处理；
- 3) 负责从 MAC 层接收数据，根据目标地址，如果目标地址是本节点，则上传给应用层，如果非本节点，则根据路由表确定下一跳地址，发送给 MAC 层处理转发；

1. 更新维护路由表

根据第三章中的路由协议方案，负责路由的发现、建立、更新和计算，结合链路状态协议和 GPS 辅助定位技术的优点，监测网络拓扑的变化。路由表的计算利用引入链路稳定间隔 LSD 因子的 Dijkstra 最短路径算法。

路由表项是这个接口的核心数据结构，这个结构存储了网络层最重要的数

据，所有分组的出口都是从路由表中查询得到的。路由表项的结构如下：

```
typedef struct rt_entry {
    uint8_t  destID;           //目标 IP 地址
    uint8_t  nextID;          //下一跳 IP 地址
    uint8_t  hops;             //总跳数
    uint32_t LSD;              //链路 LSD 值
} rt_entry;
```

目标地址（destID）代表去往节点的地址信息。

下一跳节点地址（nextID）代表将分组转发至目标地址需要经过的下一个节点的地址，指示当前节点下一个转发的节点地址。

总跳数（hops）代表从当前节点将数据分组转发到目的节点，需要经过的中间节点的个数，用于衡量分组的路径损耗。

路由稳定间隔（LSD）代表整条路由的一个稳定间隔预测值，说明了该路由能够维持的最长时间间隔，它是组成路由的所有链路中 LSD 的最小值，任意一个链路的断裂，都将导致整条路由的断裂。

路由表计算接口定义：

```
void computeRoutetable( void );
```

该接口中维护全连通矩阵和路由表，全连通矩阵的更新按照路由协议的方案执行，通过处理 Hello 消息和 TC 拓扑控制消息来更新连通矩阵的内容，然后路由层根据全连通矩阵计算路由表。路由表更新时机是 MAC 层在接收到全量和增量位置消息后，更新全连通矩阵和邻居列表，调用该计算路由的接口，完成路由表的计算，或者当路由不存在时，将数据包封装成消息存入消息队列，触发路由寻找发现的过程。

2. 应用层数据接收

该接口负责从应用层接收数据，处理应用程序数据，根据应用层提供的目标地址信息，在路由表中确定数据转发的下一跳节点地址，完成 IP 数据头部的填充，并发送给 MAC 层进行处理。

应用层接收数据的接口：

```
void packetProcess( uint8_t destID, void* point2data, uint32_t length);
```

参数 destID 代表数据发往的目标地址；参数 point2data 代表指向数据内存的地址；参数 length 代表数据块的长度。

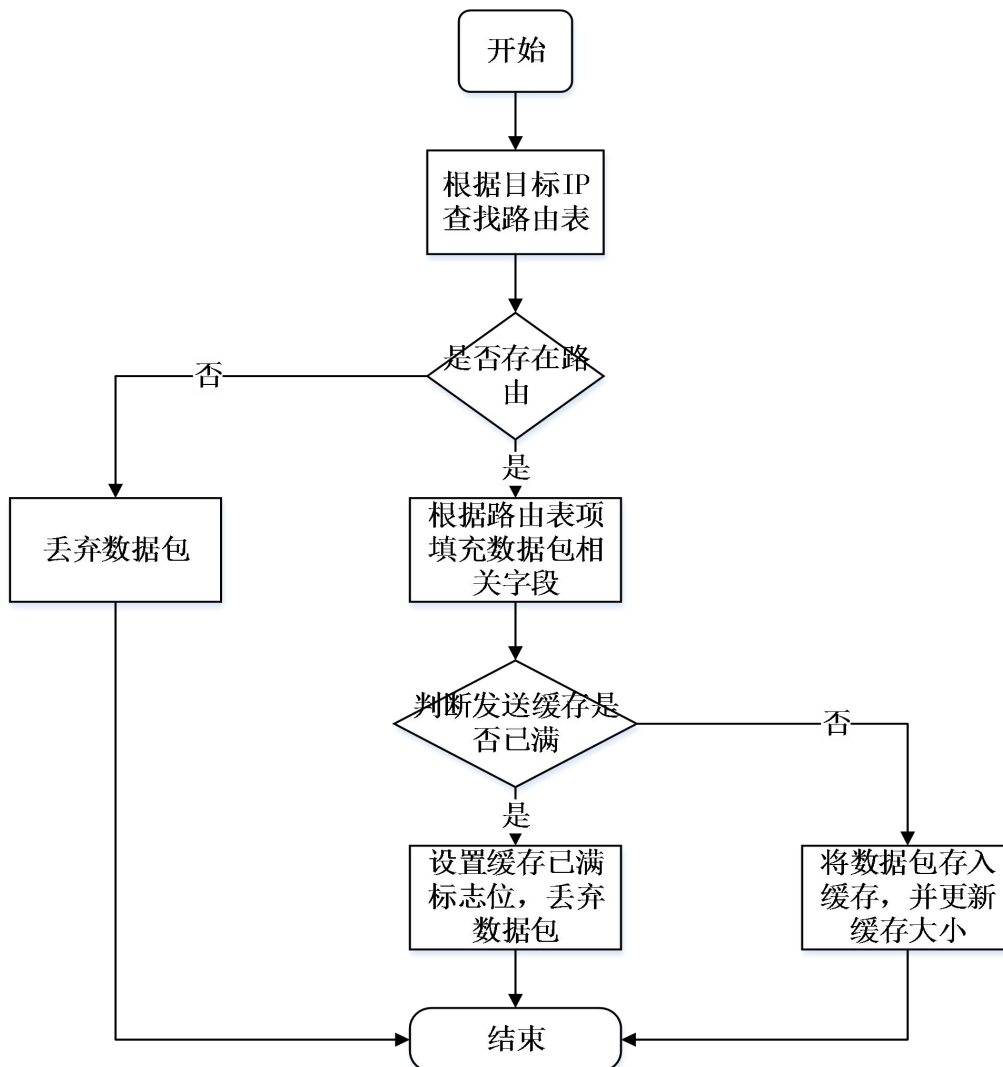


图 4-2 应用层接收数据流程图

该接口从应用层收到数据，根据目标地址，在路由表中查找下一跳地址，如果查到，则添加相应的 IP 报头，然后调用 MAC 层提供的接口，将数据交接给 MAC 层进行处理；如果没有相应路由表项，则丢弃数据包，并通知 MAC 层启动全连通矩阵更新流程。

3. 从 MAC 层接收数据

该接口负责从 MAC 层接收数据，根据目标地址，如果目标地址是本节点，则上传给应用层，如果非本节点，则根据路由表确定下一跳地址，发送给 MAC

层处理转发。

从 MAC 层接收数据接口定义：

```
receiveFromMac( void * point2data, uint32_t length);
```

参数 point2data 代表指向数据包内存地址的指针；参数 length 代表数据包的总长度。

该接口从 MAC 层收到数据后，根据 IP 报文头部信息中的目标地址判断，如果目标地址是本节点的 IP 地址，则将数据包上传给应用层；否则，处理转发流程，根据目标地址查找路由表，如果找到路由，则调用 MAC 层接口处理转发；如果没有找到路由，则丢弃数据包。

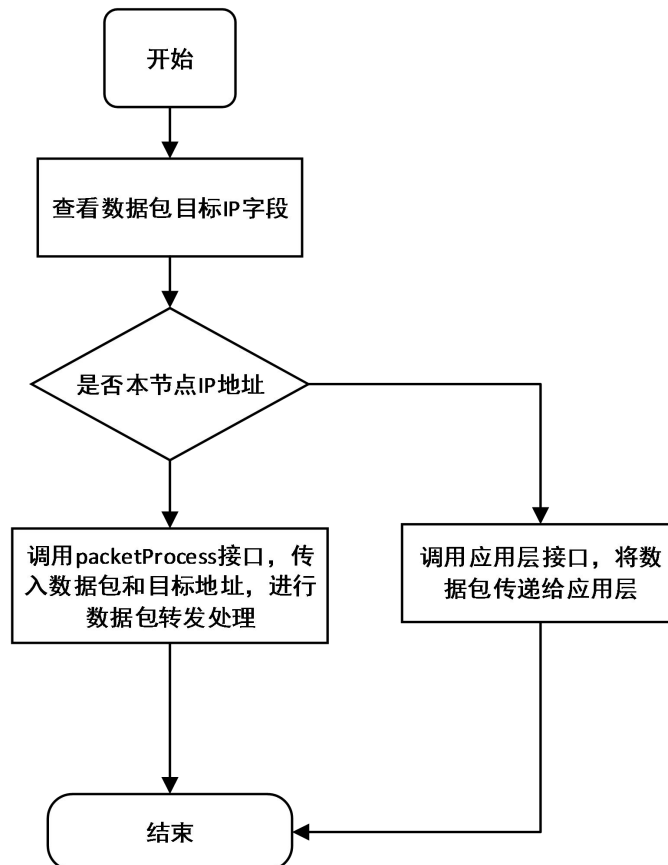


图 4-3 MAC 层接收数据流程图

4.5 网络层具体实现

4.5.1 列表结构

该数据结构是基于连续存储的线性列表，有固定的存储大小，默认为 256。

该数据结构主要用于在路由表计算的程序中用于存储未访问节点。列表提供的函数如下所示：初始化列表，添加元素，删除元素，返回列表当前大小，根据索引返回对应元素值，判断列表是否为空等；列表的使用前必须初始化。

4.5.2 路由表结构

该数据结构主要用于存储路由表项，是基于 Key-Value 的一个映射结构。这里采用的是连续存储的数组来实现，数组的下表代表 KEY（键值），数组的内容代表 Value（值），数组的默认长度为 256，足够容纳所有可能的 KEY。这里的 KEY 对应的是目标节点的 ID，VALUE 对应去往相应目的节点的路由表项。

路由表项包含了目的节点 ID，下一跳节点 ID，总跳数 HOPS，路由的 LSD 值。

路由表包含一个 256 大小的路由表项数组结构，256 大小的标识对应目的节点路由是否存在的标志数组，及当前表大小。提供基本路由表操作：初始化路由表，添加路由，删除路由，清空路由表，根据目的节点 ID 查找对应路由，根据目的节点查找发送路由等。

4.5.3 数据包缓存队列

该数据结构主要用于缓存要发送的数据包结构，采用的是基于链表的队列的实现。有新数据需要发送插入队尾，取数据，则队头数据出队。队列的元素均是动态分配的内存块，当出队后，数据结构需要调用者负责释放相应的内存块。本队列的使用必须先初始化，入队需要调用者判断队列是否已满，入队操作默认队列仍有空间；出队操作需要调用者自行判断队列是否为空，出队默认队列不空。

数据包的封装：该数据包的下一跳地址 ID，数据包起始指针，数据包长度。

队列包含：头指针，尾指针，队列容量，队列当前大小，设置阻塞标志的队列因子，清除阻塞标志的队列因子。

提供了基本的队列操作：初始化队列，入队，出队，读取队列头元素，判断队列是否阻塞，是否为空，是否已满，是否需要清空阻塞标志位等操作。

缓存中的元素是动态内存分配的，在删除元素，或者元素出队后，需要负责释放所占用的内存。

4.5.4 拓扑消息的结构

用于计算路由需要的全网拓扑结构，表示两个节点之间的连接关系和相应链路的 LSD 值；作为一个 10*10 的全局二维数组存在。

4.5.5 全局变量

全局的路由表和全局的数据包缓存队列在使用时需要在主函数中初始化。

全局全连通矩阵是 MAC 层负责更新，网络层根据此来计算全网路由。

阻塞标志符是用于表示全局数据包缓存队列是否处于阻塞的标识。

4.5.6 网络层的数据包格式

网络层的数据包的报文头部占用 6 个字节（BYTE），具体定义如下：

```
typedef struct netlayer_hdr {  
    INT8U source_ID;           //源地址  
    INT8U dest_ID;             //目标地址  
    INT16U dataLength;         //数据段的长度  
    INT8U  TTL;                //数据包的生存期，Time_To_Live（跳数计）  
    INT8U  protoType;          //协议类型  
} netlayer_hdr;
```

4.5.7 路由表的计算

根据全连通矩阵计算路由表，主要是依据 Dijkstra 最短路由算法，以跳数作为主损耗因子，LSD 作为辅助因子。根据网络节点数建立未访问节点列表，初始化到达目的节点对应的路径损耗 dist 数组，及到达目的节点的前一跳节点的 ID 数组 prev_nodeID[]，到达目的节点链路 LSD 值数组 min_LSD_entry[]；循环计算：在列表中选出损耗最小的的那个节点（第一次必然是本节点 ID），将该节点从未访问节点列表中删除，遍历列表中与该节点相邻的节点，更新他们的路径损耗值。直到列表为空，所有节点都访问了，说明最短路径已经计算完毕。最后，根据 dist 和 prev_nodeID 回溯计算路由表项，并插入路由表中，完成路由表的计算。

LSD 值代表两点间稳定保持在通信半径内的估计最长时间，即节点间能相互通信的估算时间，主要是通过各自节点的当前位置坐标和相应的速度分量估算出来的，用于在选择路由时，帮助选择一条期望最稳定，LSD 值最大的路径，提高路由稳定性。一整条链路的 LSD 值取每每两点间 LSD 的最小值（短板原理）。

4.5.8 处理应用层的数据包发送

收到应用层的数据包后，让指针指向网络层的首地址，强制转换成网络层头部结构，根据目标地址 ID 查询路由表，如果有路由：填充网络层头部，源地址，目标地址，数据包长度，TTL（默认 32），根据表项中下一跳地址封装数据包 packet 结构。然后查询数据包缓存是否满，如果满了则丢弃数据包；没有满，则将数据包入队，查看是否队列阻塞，如果阻塞，设置阻塞标志位。如果没有路由，

直接丢弃数据包。

4.5.9 收到来自物理层的消息处理

收到物理层的数据包后，指针跳转到网络层首地址，强制转换成网络层数据报头结构，然后查看目标地址 ID，如果 ID 是本节点，则调用应用层接口，发送给应用层；如果非本节点 ID，说明是转发的数据包，先查看 TTL 是否为 0，如果为 0，说明数据包过期，直接丢弃；如果 TTL 非 0，再根据目标地址 ID，查询路由表，根据表项的下一跳地址，封装数据包 packet 结构。然后查询数据包缓存是否满，如果满了则丢弃数据包；没有满，则将数据包入队，查看是否队列阻塞，如果阻塞，设置阻塞标志位。如果没有路由，直接丢弃数据包。

4.6 测试结果

4.6.1 测试环境

表 4-2 测试环境配置说明表

序号	用途	硬件环境	软件环境
1	模拟节点	内存：4G 硬盘：512G	操作系统：uc/os -II
2	仿真测试平台	内存：4G 磁盘：512G	操作系统：Windows 7

4.6.2 测试结果

场景一：测试 compute_route_table 函数能否正确计算出路由表

假设连通矩阵如下图所示

	0	1	2	3	4	5	6	7	8	9
0	0	1	0	0	0	0	0	0	0	0
1		0	1	0	0	0	0	0	0	0
2			0	1	0	0	0	0	0	0
3				0	1	0	0	0	0	0
4					0	1	0	0	0	0
5						0	1	0	0	0
6							0	1	0	0
7								0	1	0
8									0	1
9										0

图 4-4 预设的连通矩阵图

预期结果：1 号节点和 0 号，2 号节点直接连通。
1 号节点与 3，4，5，6，7，8，9 号节点通过 2 号节点连通。
测试结果：

```
the route table is:
destID->0      nextID->0      hops->1
destID->2      nextID->2      hops->1
destID->3      nextID->2      hops->2
destID->4      nextID->2      hops->3
destID->5      nextID->2      hops->4
destID->6      nextID->2      hops->5
destID->7      nextID->2      hops->6
destID->8      nextID->2      hops->7
destID->9      nextID->2      hops->8
```

图 4-5 路由表计算结果图

结论：符合预期结果，测试通过。
假设连通矩阵如下图所示

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1		0	0	1	1	0	0	1	0	0
2			0	0	0	0	0	0	0	0
3				0	0	1	0	0	0	0
4					0	0	1	0	0	0
5						0	0	1	0	0
6							0	0	0	0
7								0	0	0
8									0	0
9										0

图 4-6 预设的连通矩阵图

预期结果：1 号和 3，4，7 号节点直接连通。
1 号和 5 号节点通过 3 号节点连通。
1 号和 6 号节点通过 4 号节点连通。
测试结果：

```
the route table is:
destID->3      nextID->3      hops->1
destID->4      nextID->4      hops->1
destID->5      nextID->3      hops->2
destID->6      nextID->4      hops->2
destID->7      nextID->7      hops->1
```

图 4-7 路由表计算结果图

结论：符合预期结果，测试通过。
场景三：应用层发来数据，且路由表存在路由，但是发送缓存已满
假设本节点 ID 为 1，应用层发来向 5 号节点发送数据的请求，数据内容为

Hello!。

预期结果：应用层数据包丢弃，不加入缓冲队列中。

测试结果：

```
the route table is:
destID->3      nextID->3      hops->1
destID->4      nextID->4      hops->1
destID->5      nextID->3      hops->2
destID->6      nextID->4      hops->2
destID->7      nextID->7      hops->1
Buffer queue's capacity is 10. Buffer queue current size is 10
DestID is 5.Route nextID is 3.Buffer is full!
```

图 4-8 缓存满，丢弃数据包

结论：符合预期结果，测试通过。

场景四：应用层发送数据，路由表存在路由，但数据包缓存未滿

假设本节点 ID 为 1，应用层发来向 5 号节点发送数据的请求，数据内容为

Hello!。

预期结果：应用层数据包加入缓冲队列，更新缓冲队列状态。

测试结果：

```
the route table is:
destID->3      nextID->3      hops->1
destID->4      nextID->4      hops->1
destID->5      nextID->3      hops->2
destID->6      nextID->4      hops->2
destID->7      nextID->7      hops->1
Buffer queue's capacity is 10. Buffer queue current size is 0
Buffer queue's capacity is 10. Buffer queue current size is 1
Packet enter buffer.Packet data is Hello!,packet nextID is 3.
```

图 4-9 缓存未滿，数据包存入缓存区

结论：符合预期结果，测试通过。

场景五：应用层发送数据，不存在路由

假设本节点 ID 为 1，应用层发来向 2 号节点发送数据的请求，数据内容为

Hello!。

预期结果：找不到路由，应用层数据包丢弃。

测试结果：

```

the route table is:
destID->3      nextID->3      hops->1
destID->4      nextID->4      hops->1
destID->5      nextID->3      hops->2
destID->6      nextID->4      hops->2
destID->7      nextID->7      hops->1
No route!

```

图 4-10 路由表不存在到达目的节点的路由

结论：符合预期结果，测试通过。

场景六：物理层向网络层发送数据包，该数据来自其他节点

假设收到 5 号节点向 6 号节点发送的数据包，本节点 ID 为 1。

预期结果：调用 packetProcess 接口。

测试结果：

```

the route table is:
destID->3      nextID->3      hops->1
destID->4      nextID->4      hops->1
destID->5      nextID->3      hops->2
destID->6      nextID->4      hops->2
destID->7      nextID->7      hops->1
Receive a packet from mac layer.Source ID is 5, Destination ID is 6.
Call packetProcess!

```

图 4-11 处理转发给其它节点的数据包

结论：符合预期结果，测试通过。

场景七：物理层向网络层发送数据包，该数据包目的节点为当前节点

假设收到 5 号节点向 1 号节点发送的数据包，本节点 ID 为 1。

预期结果：将数据包传递给应用层。

测试结果：

```

the route table is:
destID->3      nextID->3      hops->1
destID->4      nextID->4      hops->1
destID->5      nextID->3      hops->2
destID->6      nextID->4      hops->2
destID->7      nextID->7      hops->1
Receive a packet from mac layer.Source ID is 5, Destination ID is 1.
Send to application layer!

```

图 4-12 处理发送给自身节点的数据包

结论：符合预期结果，测试通过。

4.7 小结

本章内容介绍了高速路由协议基于 uC/OS-II 嵌入式系统的设计及实现方案。在仿真验证的基础上，通过对传统 IP 协议的改进，简化 IP 协议的报文，设计适应于高速移动网络的路由。详细的介绍了路由协议的设计思路，路由模块和应用层及物理层的交互接口，重要的处理流程和关键数据结构，并通过实验测试，验证了该系统的各接口的可用性和正确性。

第五章 总结和展望

5.1 本文总结

随着移动无线网络的普及，对移动设备的自组网能力的需求越来越高。国际学者的多年研究，无线多跳网络路由协议日趋成熟。这些协议在低速移动节点的场景中应用效果良好，节点速率在十几米每秒到几十米每秒的水平。但是应对越来越多的高速移动应用场景，如飞行器，无人机等，节点移动速率可达到音速级别，比普通场景高了一个数量级，传统的路由协议的适用性效果就下降了。因此，本文针对高速移动的小型无线多跳网络场景，设计一种路由协议，使其能适应节点高速移动的特性。全文的主要工作包括：

1. 对无线多跳网络进行介绍，包括无线多跳网络的定义，网络的特点，路由协议的研究现状和研究热点，明确高速移动路由的研究方向和研究内容。
2. 对现有无线多跳网络路由协议进行介绍，包括移动路由协议的质量要求，评价标准，协议的分类，优缺点比较分析，对路由协议有一个全面的把握。
3. 针对高速移动场景，结合链路状态协议和 GPS 定位技术的优点，改进 OLSR 路由协议，引入链路稳定间隔 LSD 因子，辅助路由控制消息的广播和路由算法的决策，减少路由开销，增加路由稳定性。在 NS2 平台上，实现该路由协议模块，设计仿真场景对协议的可靠性进行验证，并传统的 OLSR 协议进行性能比较，结果表明引入 LSD 因子后，路由协议在高速场景中的路由断裂次数和路由广播分组数量上有更良好的性能，提高路由稳定性，减少路由开销。
4. 将引入 LSD 因子的 OLSR 路由协议移植到 uC/OS-II 轻量级嵌入式系统上，对网络层功能进行封装，提供统一的对应用层和 MAC 层的通信接口，与其他模块协作，组成适应高速场景的无线多跳网络系统。

5.2 研究展望

本文在适应小型高速移动场景的路由协议设计和改进上，取得了上述阶段成果。改进后的路由协议能够满足在小型高速移动飞行器上的自由组网功能，基本满足场景应用的需求。今后的研究还需要进一步探讨的问题有：

1. GPS 定位设备提供的节点位置和运动信息等数据的应用还不够充分。本文只利用到预测链路稳定间隔 LSD 因子，应用在路由决策和广播分组间隔设置两方面。这些数据还可以深入应用，比如在洪泛分组的范围，拓扑变化预测等方面。

2. 当前的路由协议是针对小型高速网络，当网络规模扩大后，可以考虑在主动路由 OLSR 的基础上，引入按需路由的某些策略及多路径路由方式，提高整体网络的可靠性和高效性。

3. 在 uC/OS-II 嵌入式系统上的实现，还可以考虑更多的安全策略，应急机制方案，让路由模块更好更高效地和其他模块协作。

这些值得进一步改进和探讨的问题，将在今后的研究过程中，深入了解和思考，不断完善该路由协议。

参考文献

- [1] DRAVES R, PADHYE J, ZALL B. Comparison of routing metrics for static multi-hop wireless network[A]. Proc of 2004 Conference on Computer Communications[C]. New York: Assoc Computing Machinery, 2004, 21 (6) : 351-360.
- [2] 史美林, 英春. 自组网路由协议综述[J]. 通信学报, 2001, 22 (11) : 93-101.
- [3] 于宏毅. 无线移动自组织网[M]. 北京: 人民邮电出版社, 2005.
- [4] 郑相全. 无线自组网技术使用教程[M]. 北京: 清华大学出版社. 2004.
- [5] 陈林星, 曾曦. 移动 Ad Hoc 网络——自组网分组无线网络技术[M]. 北京: 电子工业出版社, 2006
- [6] 王金龙. Ad Hoc 网络技术[M]. 北京: 国防工业出版社, 2004
- [7] PERKINS C E, BHAGWAT P. Highly dynamic destination sequenced distance vector routing(DSDV) for mobile computers[A]. The ACM SIGCOMM Conf. On Communications Architectures[C]. London, 1994
- [8] JACQUET P, MUHLSDHALAR P, QAYYUM A. Optimized Link State Routing Protocol[S]. Internet Draft(work in progress), INRIA Rocquencourt, 2000. Draft-ietf-manet-olsr-01.txt.
- [9] JACQUET P, MUHLSDHALAR P, QAYYUM A. Optimized Link State Routing Protocol[S]. IETF Internet Draft, draft-ietf-manet-olsr-10.txt, 2002.
- [10] PEI G, GERLA M, CHEN T W. Fish eye state routing: a routing schema for ad hoc wireless network[A]. The IEEE Int'l Conf. On Communications(ICC)[C]. New Orleans, LA, 2000.
- [11] BELLUR B. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)[S]. IETF Internet Draft, 2003. Draft-ietf-manet-tbrpf-08.txt.
- [12] PERKINS C, ROYER E M, DAS S R. Ad Hoc on-demand distance vector (AODV) Routing[EB/OL]. <http://www.ietf.org/rfc/rfc3561.txt>.
- [13] PARK V, CORSON S. Temporally-ordered Routing Algorithm (TORA) Version 1-Functional Specification[S]. Internet Draft (working in progress), Naval Research Lab and Univ. Of Maryland, 1997. Draft-ietf-manet-tora-spec-00.txt.
- [14] DUBE R, RAIS C D, WANG K Y, et al. Signal stability based Adaptive routing (SSA) for ad hoc mobile network[J]. IEEE Personal Communications Magazine, 1997, 4 (1) : 36-45
- [15] HAAS Z J, PEARLMAN M R. The performance of query control schemes for the

- zone routing Protocol[J]. The IEEE/ACM Transactions on Networking, 2001, 9 (4) : 427-438.
- [16]JIANG M L, LI J Y, TAY Y C. Cluster Based Routing Protocol (CBRP) [S]. Internet Draft (work in progress), National University Of Singapore, 1999. Draft-ietf-manet-cbrp-spec-01.txt.
- [17]SIVAKUMAR R, SINHA O, BHARGHAVAN V. Cedar: a Core-extraction distributed ad hoc Routing Algorithm[J]. IEEE Journal on Selected Areas in Communications, Special Issue on Ad Hoc Networks, 1999,17 (8) : 1454-1465.
- [18]BASAGNI S, CHLAMTAC I, SYROTIUK V R, et al. A Distance Routing Effect Algorithm for Mobility (DREAM) [A]. The ACM/IEEE Int'l Conf. On Mobile Computing and Networking (MOBICOM) .Dallas, 1998.
- [19]KARP B, KUNG H T, GPSR: Greedy perimeter stateless routing for wireless networks[A]. Proceeding 6th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM'98) [C]. Dallas, 1998: 66-75
- [20]KO Y B, VAIDYA N H. Location-Aided Routing (LAR) in Mobile Ad Hoc Networks[A]. The ACM/IEEE Int'l Conf. On Mobile Computing and Networking (MOBICOM) .Dallas, 1998.
- [21]MUELLER S, TSANG R P, GHOSAL D. Multipath routing in mobile ad hoc networks: issues and challenges[A]. Lectures notes in Computer Science[C]. New York: Springer, 2004: 209-234
- [22]PEARLMAN M R. On the impact of alternate path routing for load balancing in mobile ad hoc networks[A]. MobiHoc[C]. 2000: 3-10.
- [23]PHAM P P, PERREAU S. Performance analysis of reactive shortest path and Multipath routing mechanism with load balance[A]. Proceeding of the IEEE INFOCOM[C]. 2003.
- [24]PAPADIMITRATOS P, HAAS Z J, SIRER E G. Path set selection in mobile ad hoc networks[A]. Proceeding of the Third ACM Symposium on Mobile Ad Hoc Networking & computing (MobiHoc 2002) [C]. Lausanne, Switzerland, 2002.
- [25]MARINA M K, DAS S R. On-demand Multipath Distance vector routing in ad hoc networks[A]. Proceedings of the International Conference for network Protocol (ICNP) [C]. Riverside, 2001.
- [26]LEE S J, GERLA M. Split Multipath routing with maximally disjoint paths in ad hoc networks[A]. ICC'01, Helsinki[C]. Finland, 2001.
- [27]LEUNG R. MP-DSR: a QoS-Aware Multipath DSR Protocol for Wireless ad hoc

- Networks[A]. Proceedings of the 26th LCN[C].2001, 3: 132-142.
- [28] VALERA A, SEAH W, RAO S. CHAMP: a highly-resilient and energy-efficient routing Protocol for mobile ad hoc networks[A]. Proceedings of the fourth IEEE Conference on mobile and wireless Communications Networks (MWCN 2002) [C]. Stockholm, Sweden, 2002.
- [29] CORDEIRO C, GOSSAIN A, AGRAWAL D. Multicast over wireless mobile ad hoc networks : present and future directions[J]. IEEE Network Magazine , January/February 2003, 17 (1) : 52-59.
- [30] CHIANG C C, GERLA M. Routing and Multicast in Multihop, mobile wireless Networks[A]. Proceedings of IEEE International Conference on University Personal Communications (ICUPC'97) [C]. San Diego, USA, 1997.
- [31] BOMMAIAH F. AMRoute : Ad Hoc Multicast Routing Protocol[S]. Internet Draft, 1998.
- [32] WU C W, TAY Y C, TOH C K. Ad Hoc Multicast Routing Protocol Utilizing Increasing Id-Numbers (AMRIS) Functional Specification[EB/OL]. Internet Draft, 1998.
- [33] CHIANG C C, GERLA M. Forwarding group Multicast Protocol (FGMP) for Multihop[J]. Mobile Wireless Networks , ACM-Baltzer Journal of Cluster Computing: Special Issue on mobile Computing, 1998, 1 (2) : 187-196.
- [34] LEE S J, SU W, GERLA M. On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks[S]. Internet Draft (work in progress) , University of California, 2000. Draft-ietf-manet-odmrp-02.txt.

致谢

谨此论文完成之际，向所有曾经帮助过我的老师、同学、家人和朋友表示最诚挚的感谢。

首先，非常感谢我敬爱的导师廖青老师。本文从选题到写作的各个环节都在廖老师的悉心指导和严格要求下完成，廖老师严谨的治学态度、为人师表的高尚品德以及教人不倦的导师精神都给我很大的启发，对我在做人、做事、学习、生活等各个方面都产生了积极的影响，使我终生受益。

在论文完成的过程中，实验室同学们的帮助和支持也不可或缺，他们给了我很多的宝贵意见，让我不断成长，并陪伴我度过了愉快并难忘的三年研究生生活。

最后，要感谢我的家人，是他们不断的支持和温暖的关怀让我能够专心于课题的研究和论文的顺利完成，谢谢！

再次向所有的老师，同学，家人和朋友表示最真诚的感谢。