南开大学

硕士学位论文

ZigBee路由算法分析及改进

姓名：陈波

申请学位级别：硕士

专业：通信与信息系统

指导教师：韩毅刚

20090601

摘 要

ZigBee 技术基于 IEEE802.15.4 标准，主要面向低成本、低功耗、低数据传输率的网络应用，在环境监测、军事、医疗、工业控制和家庭智能监控等领域有着广泛的应用前景。因此，Zigbee 技术目前正成为研究热点，尤其是节能措施、路由算法是目前研究的重点方向。

虽然 ZigBee 规范中路由协议没有专门针对网络能量有效利用方面的机制，但它的技术特点和应用领域决定了节能问题对 ZigBee 网络的重要性。从单个网络节点的角度看，需降低其能耗且延长节点使用时间。从整个网络来看，单纯追求降低各节点能耗反而可能造成某些节点过多消耗能量，最终导致这些节点失效甚至造成网络分割。

本文对 Cluster-Tree、Z-AODV、MTPR 和 MMBCR 路由算法用 NS2 软件进行仿真研究，根据实验结果总结出这几种路由算法的优缺点，并提出一种能量均衡算法的改进方案。这种路由算法改进方案的主要思想是：尽可能地让网络各节点平均地消耗能量，在此基础上，选择累计能量消耗最少的路径作为转发路由。

仿真实验得出的结果表明，这种能量均衡算法的改进方案能够有效地降低网络的总体功耗，推迟节点失效时间，延长网络的工作时间，对降低 ZigBee 网络的维护成本是有益的。

关键词：Zigbee　路由算法　仿真　能量均衡

Abstract

ZigBee technology, based on the IEEE802.15.4 standard, is a low cost, low power and low data rate wireless network, which is widely used in environmental monitoring, military, medical and industrial control and home intelligent control system. Therefore, ZigBee technology attracts more attentions of researchers who are especially intreasted in studying energy-saving measures and routing algorithm.

Although the routing protocol in the ZigBee specification is not involved with the energy-efficient mechanism in networks. the characteristics and application fields of Zigbee technology determines the importance of energy saving for ZigBee networks. For a single nodes in network, its energy consumption must be reduced and its usage time must be extented. For whole network, only seeking to reduce each node energy consumption can not reduce the encergy consumption of whole network, but lead to some nodes cosuming excessive energy, and even to network partitioning caesed by the nodes failure.

In this paper, several routing algorithms such as Cluster-Tree, Z-AODV, MTPR and MMBCR were studied by simulation experiments with NS2 software. Based on the experimental results, the advantages and disadvantages of several routing algorithms above mentioned were concluded, and that an improving scheme on energy-balanced algorithm was presented. The main idea of the improving scheme was to cause each node in network consuming averagely energy as much as possible, and to choose the path cosuming the least accumulated energy as a data routing.

The results of simulation experiments show that the improving scheme on energy-balanced algorithm can effeciently reduce the overall power cosumption in network, delay network node failure, and extend working hours of network, which was benefit for reducing maintenance cost for ZigBee network.

Key Words: ZigBee, Routing algorithm, Simulation, Energy balance

南开大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年　　月　　日

第一章 绪论

近年来，各种无线通信技术迅猛发展，极大地提高了人们的工作效率和生活质量。然而，在日常生活中，人们仍然被各种电缆所束缚，如何在近距离范围内实现各种设备之间的无线通信，成为当前的研究热点。由于短距离无线通信和网络技术有广阔的应用前景和巨大的市场空间，因此得到了许多厂商的重视，取得了很大的发展。

第一节 研究背景

目前发展较成熟的几大无线通信技术，如红外的 IrDA（Infrared Data Association）、射频识别（Radio Frequency Identification，RFID）、蓝牙（Bluetooth）、超宽带（UWB）、Wi-Fi 技术等，它们各有自己的特点和应用领域，也有各自的局限性。这些技术协议比较复杂，耗费资源较多，成本也比较高，不太适用于要求低成本、低功耗的工业控制和家庭网络。

ZigBee 是一种新兴的短距离、低速率无线网络技术，是一组基于 IEEE802.15.4 无线标准研制开发的，关于组网、安全和应用软件的技术标准，适合于承载数据流量较小的业务。与其它无线通信协议相比，ZigBee 无线协议复杂性低、对资源要求少，主要特点有：功耗低、成本低、时延短、传输范围小。数据传输的可靠性由于 ZigBee 采用了碰撞避免机制，同时为需要固定带宽的通信业务预留了专用时隙，从而避免了发送数据时的竞争和冲突。MAC 层采用完全确认的数据传输机制，每个发送的数据包都必须等待接收方的确认信息，保证了节点之间传输信息的高可靠性。

ZigBee 技术一经出现就受到众多芯片生产厂商、软件开发商、OEM 厂商和系统集成厂商的注意。ZigBee 技术以其低功耗、低速率、低成本的技术优势得到迅猛发展[1]。它将给人们的工作和生活带来极大的方便和快捷。

第二节 ZigBee 技术的发展及应用

1999 年 3 月，成立了 IEEE802.15 工作组，主要是建立个人工作空间无线通讯的国际标准，实现和 IEEE802.11 协议族的融合。IEEE802.15 组制定了三种 WPAN 标准， IEEE802.15.1 标准，即蓝牙技术；IEEE802.15.3 主要针对高速图像和多媒体应用，有较高速率；对其物理层进行改进，形成 IEEE802.15.1a 标准，应用于超宽带技术（UWB）。

2000 年 12 月，IEEE 成立 IEEE802.15.4 工作组，致力于定义一种廉价设备使用的低复杂度、低成本和功耗的低速率短距离无线连接技术。2003 年 5 月通过了 IEEE802.15.4 标准[2]，定义了媒体访问控制层和物理层上的规范。ZigBee 联盟成立于 2002 年，由英国 Inversys 公司、日本三菱电气公司、美国摩托罗拉公司及荷兰飞利浦半导体公司发起，联盟在 IEEE802.15.4 的基础上定义了网络层和应用层，并负责高层应用、测试和市场推广等方面的工作。联盟于 2004 年 12 月发布了 1.0 版本规范，2006 年 11 月发布了 1.1 版本规范，2007 年发布了 1.6 版本规范，即 ZigBee Pro[3]。目前已有很多国外厂商推出遵循 ZigBee 规范的产品，市场上主流芯片供应商包括 TI、Ember、Freescale 及 Jennic，国内的华为等公司也加入了联盟[4]。目前，ZigBee 技术正是无线网络的研究热点。随着 ZigBee 技术的不断完善和发展，它必将有着广阔的应用前景，如楼宇自动化、工业控制、环境监测、智能家居等领域[5]。其典型的应用领域如下：

1．数字家庭领域

ZigBee 技术可以应用于家庭的照明、温度、安全、控制等。ZigBee 模块可以安装在电视、电灯、遥控器、儿童玩具、游戏机、门禁系统、空调系统和其他家电产品中，例如在电灯中装置 ZigBee 模块，人们要开灯就不需要走到墙壁开关处，直接通过遥控便可以开灯。当你打开电视机时，灯光会自动减弱；当电话铃响起时或你拿起电话机准备打电话时，电视机会自动静音。通过 ZigBee 终端设备可以收集家庭各种信息，传送到中央控制设备，或是通过遥控达到远程控制的目的，提供家居生活自动化、网络化与智能化。韩国移动手持设备制造商 Curitel Communications 公司已经开始研制 ZigBee 手机，该手机能够使手机用户在短距离内操纵电动开关和控制其他电子设备。

2．工业控制

通过 ZigBee 网络自动收集各种信息，并将信息回馈到系统进行数据处理与

分析，以利于工厂整体信息的掌握。例如火警的感测和通知，照明系统的感测，生产设备的流程控制等，都可以由 ZigBee 网络提供相关信息，已达到工业与环境控制的目的。在 ZigBee 技术产生前，已经有一些无线技术用于工业设备监控领域，但 ZigBee 技术有非常突出的优势[6]。韩国的 NURI Telecom 在基于 Atmel 和 Ember 的平台上成功研发出基于 ZigBee 技术的自动抄表系统，该系统无须人工手动读取电表、天然气表及水表，减少了人力需求，从而节省了公共开支[7]。

3．智能交通

如果沿街道、高速公路及其他地方分布式地装有大量 ZigBee 终端设备，车辆上安装 ZigBee 模块可以告诉司机交通信息，比如限速、前方交通路况、或事故信息。也可以跟踪公共交通情况，适时调度车辆[8]。

4．医学护理

在医学领域，利用 ZigBee 传感网络可以准确、实时监测每个病人的血压、体温和心率等信息，有助于医生快速做出反应，减少医生查房的工作负担，特别适用于对危重病人的监护和治疗[9]。

5．其他领域

在现代农业中，利用 ZigBee 传感网络可以监测和控制土壤湿度、pH 值等信息，有助于提高水资源利用率和农作物产量。在军事领域，可以用于战场侦查和监控。

第三节 论文的主要研究工作和内容安排

由于 ZigBee 技术规范推出的时间相对较晚，规范本身存在一些空白之处，如路由方式的选择，如何进一步降低功耗等。本文研究的主要目的是通过对一些 ZigBee 路由协议的研究和网络性能分析，对路由协议做出算法优化和改进，以达到降低网络功耗、延长网络寿命、提高网络性能的目的。本文主要的工作是：

1．分析 IEEE802.15.4 和 ZigBee 协议，理解协议的技术特性。

2．用 NS2 对 ZigBee 几种常见路由算法进行仿真实验，对结果深入分析。

3．在上述工作的基础上，改进能量均衡的路由算法，并用 NS2 进行仿真实验，仿真结果表明，改进的算法达到了预期的效果。

论文的内容安排：

第一章介绍 ZigBee 的技术背景和当前的研究现状，提出本课题的主要研究内容；

第二章介绍 IEEE802.15.4/ZigBee 协议的概念及架构；

第三章介绍 ZigBee 的几种基本路由算法；

第四章用 NS2 实现 ZigBee 仿真，测试路由算法效率；

第五章根据仿真结果的分析，对一种能量均衡路由算法进行改进，并做性能仿真；

第六章对研究工作做出总结，提出未来的研究方向。

第二章 IEEE802.15.4/ZigBee 协议架构

ZigBee 是近几年出现的无线网络技术，对设备性能要求低，网络结构灵活，有着鲜明的技术特点。它的物理层和媒体访问控制层采用 IEEE802.15.4 标准，网络层和应用层由 ZigBee 联盟定义。

第一节 ZigBee 网络概述

2.1.1 ZigBee 网络特点

ZigBee 是一种低速无线个域网（Wireless Personal Area Network，WPAN）技术，它适用于通信数据量不大，数据传输速率相对较低、分布范围小的场合。与其他几种常用的短距离无线通信技术相比，它有几个特点：

1．灵活的工作频段。ZigBee 选用工业科学医疗（ISM）免注册频段，为适应不同情况，定义了 2.4GHZ 频段和 868/915MHZ 频段。在 2.4GHZ 频段内分配了 16 个信道，868MHZ 频段内有一个信道；915MHZ 频段内有 10 个信道[10]。我国使用的设备应该工作在 2.4GHZ 频段内。

2. 低功耗。ZigBee 的发射功率一般仅为 10mW，网络可采用间接数据传输，由电池供电的节点发起数据传输，这样设备可以在没有数据传输的时候工作在睡眠状态，从而最大限度地降低功耗。

3．成本低廉。ZigBee 免收协议专利费，相对于其它的网络技术，ZigBee 网络协议较简单，可以在计算能力和存储能力非常低的设备上运行，对设备资源要求低，适合于低成本的应用场合。

4．网络结构灵活。ZigBee 网络既可以采用 16 位的网络短地址，又可以采用 64 位的 IEEE 地址，既支持星形网络结构、树形网络结构，也支持网形网络结构。而 Bluetooth 最多只能组成八个节点的星形网络[11]，WiFi 只能组成星形网络[12]。

5. 低传输速率。ZigBee 在 2.4GHZ 频段内信道数据传输速率为 250kb/s[13]，868MHZ 频段内信道数据传输速率为 20kb/s，915MHZ 频段内信道数据传输速率

为 40kb/s，满足低速率传输数据的应用需求。Bluetooth 最大传输速率为 750kb/s，WiFi 最大传输速率达到 54Mbp/s。

6．传输距离。符合 ZigBee 规范的无线传输距离是 10 至 75 米[14]，在增加发射功率后也可增加到 1 到 3 千米。Bluetooth 一般在 10 米左右，WiFi 在 100 米左右。

7．网络容量大。一个单独的 ZigBee 网络可以容纳最多 2^{16} 个设备节点。

2.1.2 ZigBee 网络设备类型

ZigBee 网络中的设备按照性能分为两类：全功能设备和精简功能设备。

全功能设备 (Full Function Device，FFD)，具有完整功能的全功能设备，支持协议标准定义的所有的功能和特性。

精简功能设备 (Reduce Function Device，RFD)，只具有部分功能的精简功能设备。RFD 的功能非常简单，存储容量要求很少，可以用最低端的微控制器实现，在网络里只能作为终端设备。

按照功能分为三类：协调器、路由器和终端设备。

协调器（ZigBee Coordinator，ZC）必须是 FFD，一个 ZigBee 网络有且仅有一个协调器，它的任务包括网络启动，信道选择，网络设备地址分配，发送时间信标，维护网络，具有最多的存储空间和计算能力。

路由器（ZigBee Router，ZR）必须是 FFD，任务包括数据存储转发，邻居发现，路由发现和维护。

终端设备（ZigBee End Device，ZE）可以是 FFD 也可以是 RFD[15]，只能发送接收数据。

2.1.3 ZigBee 网络拓扑结构

ZigBee 网络结构灵活，支持星形、树形和网形网络拓扑结构。

1．星形拓扑结构

星形拓扑结构的网络由一个协调器节点和若干从设备节点组成。协调器负责网络的建立维护和数据转发，从设备只能和协调器进行直接数据传输，而与其他终端设备之间数据传输必须经过网络协调器转发。从设备可以是 FFD 也可以是 RFD。星形结构通常用于小范围的场合。

例如图 2.1 所示，0 节点作为网络协调器，负责建立网络，1 到 8 节点作为从设备节点接入网络，这些节点都可以与 0 节点直接传输数据，而它们之间的数据传输必须由 0 节点转发。

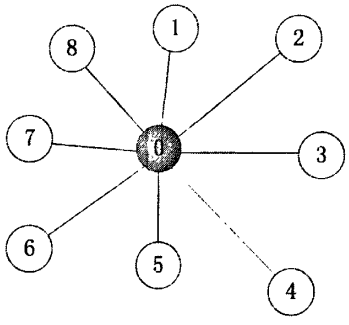

图 2.1 星形拓扑结构的网络

2．树形拓扑结构

树形拓扑结构的网络由一个根节点和若干子节点构成，这些子节点可以有自己的子节点。树的根节点是网络的协调器，因此必须是 FFD；既有子节点又有父节点的节点作为路由器，也必须是 FFD；只有父节点而没有子节点的节点叫做叶节点，既可以是 FFD 也可以是 RFD。显然，树形结构是由星形网络扩展而来。树形网络中只有父子节点之间可以进行数据传输[16]，数据沿树形结构向上或向下传输，从一个节点传输到相邻的节点称为“一跳”。如图 2.2 所示的网路中，节点 0 是网络的根节点，节点 1、节点 2 是其子节点。

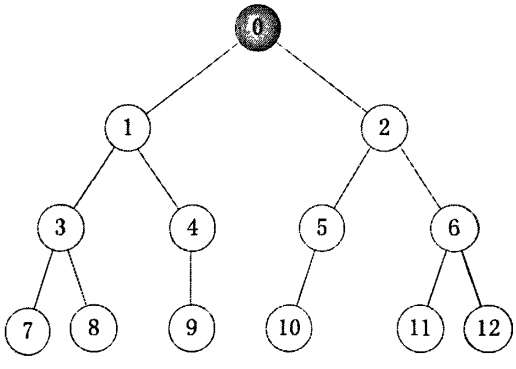

图 2.2 树形拓扑结构的网络

3．网形拓扑结构

在网形拓扑结构的网络中有一个网络协调器，通信范围内的全功能节点之

间可以相互通信，每个全功能节点都具有路由功能[17]。如图 2.3 所示。在这种网络结构中设备之间传输数据时，可以通过路由器转发，即多跳的传输方式，以增大网络的覆盖范围。网形拓扑结构具有强大功能[18]。

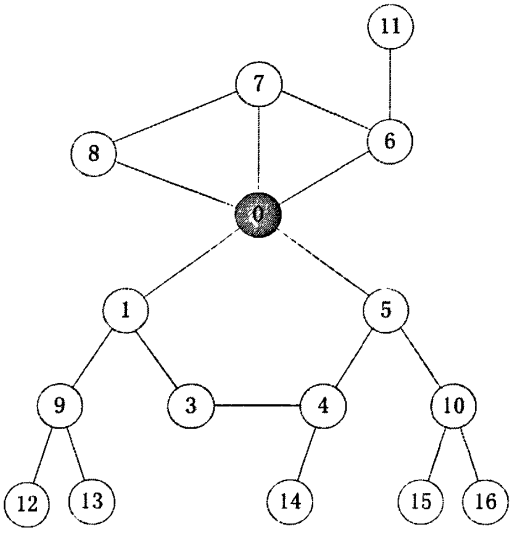

图 2.3 网形拓扑结构的网络

2.1.4 ZigBee 网络的两种工作模式

ZigBee 网络有信标（Beacon）和非信标（Non-beacon）两种工作模式。

在信标工作模式下，网络中所有设备都同步工作、同步休眠，以达到最大限度地减小功耗[19]。网络协调器负责以一定的时间间隔广播信标帧，两个信标帧之间有 16 个时隙，这些时隙分为休眠区和活动区两个部分，数据只能在网络活动区的各时隙内发送。

非信标模式下，只有 ZE 进行周期性休眠，ZC 和 ZR 设备一直处于工作状态[20]。网络中，父节点为 ZE 子节点缓存数据，ZE 主动向其父节点提取数据，这样 ZE 可以周期性休眠。

第二节 ZigBee 网络体系结构

对照 OSI 模型，ZigBee 网络分为 4 层，物理层(PHY)、媒体访问控制层(MAC)、网络层(NWK)和应用层(APL)，如图 2.4 所示。

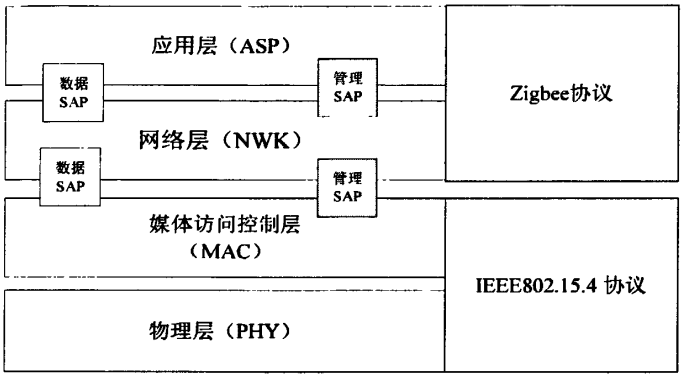

图 2.4 IEEE/802.15.4 的架构

ZigBee 的物理层和媒体访问控制层使用 IEEE 802.15.4 协议，而网络层和应用层由 ZigBee 联盟制定[21]。每一层向它的上层提供数据和管理服务。

1．物理层(Physical Layer，PHL)由半双工收发器及其接口组成，利用无线信道实现数据传输。

2．媒体访问控制层(Medium Access Control，MAC)提供节点自身和与其相邻节点之间可靠的数据传输链路。其主要任务是实现传输媒体的共享，提高通信的有效性。

3．网络层(Network Layer，NWK)提供数据通信、路由、多跳转发能力，网络实现和维护。对于一些简单的节点而言，其功能只是加入和离开一个网络；而路由器则需要完成数据转发、邻居发现、路由发现等任务。协调器的任务包括启动网络，为新加入网络的节点分配地址等。

4．应用层(Application Layer，APL)实现网络应用，具体分为三个部分：应用支持子层（Application Support Layer，APS）的任务是将网络信息转发到运行在节点上的不同应用端点；应用对象（Application Object）是运行在端点的应用软件，它具体实现节点的应用功能；应用框架是驻留在设备里的应用对象的环境，给应用对象提供数据服务。

IEEE802.15.4/ZigBee 协议定义了网络对等层之间的帧的格式、意义和交换的方式，各层实体利用协议来实现服务。对于数据帧在网络中各层之间的传输，上层向下层传输时，它会给帧附加上帧首部和尾部，以实现相应的功能；而下层向上层传输时，把本层的帧首部和尾部去掉，然后把帧载荷部分提交给上层。网络帧的结构如图 2.5 所示。

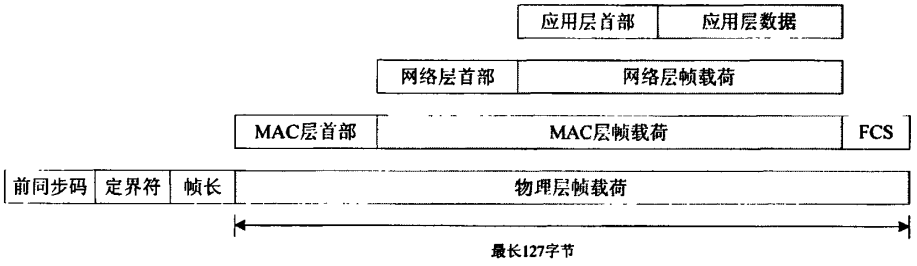

图 2.5 网络帧的结构示意图

2.2.1 物理层

物理层不仅规定了信号的工作频率范围、调制方式和传输速率，而且还规定了物理层的功能和为上层提供的服务。物理层的主要任务是通过无线信道进行安全有效的数据通信，为 MAC 层提供服务。它具备以下功能：

1. 信道选择（channel select）；
2. 信道能量监测（energy detect）；
3. 空闲信道评估（clear channel assessment）；
4. 无线信道收发数据(PHY Protocol Data Unit)；
5. 接收包链路质量检测；

ZigBee 物理层提供了一个从媒体访问控制层（MAC）到物理层的无线信道接口。在物理层中有数据服务接入点和物理层实体服务接入点。数据服务接入点支持在对等连接 MAC 层的实体之间传输 MAC 层数据单元，提供数据发送和接收服务；物理层实体服务接入点通过调用物理层的管理功能函数，为物理层管理服务提供接口。

ZigBee 采用了扩频通信技术，在 2.4GHZ 频带上使用偏移正交相移键控调制法（Offset Quadrature Phase Shift Keying，OQPSK），而在 868/915MHZ 频带使用二进制移相键控调制法（Binary Phase Shift Keying，BPSK）。

4 字节	1 字节	1 字节		可变
前同步码	帧定界符	帧长度（7 位）	保留位（1 位）	PSDU
同步包头		物理层包头		物理层有效载荷

图 2.6 物理层协议数据单元结构

物理层协议数据单元结构（PPDU）又称物理层数据包，是由附加的同步包头、物理层包头和物理层有效载荷（PSDU）组成，结构如图 2.6 所示。

1．前同步码由 32 个 0 组成，接收设备根据接收到的同步码获取同步信息识别每一位，从而进一步区分出“字符”。

2．帧定界符（SFD）为 11100101，一个字节，用来标示同步码的结束和数据包的开始。

3．物理层帧首部由一个字节组成，最高位保留，后七位用来表示有效载荷的数据长度。

4．PSDU 域是物理层携带的有效载荷，长度为 0 到 127 字节。长度为 5 字节时为 MPDU（确认帧），长度大于 7 字节时为 MAC 层的有效帧，其余长度的作为保留。

2.2.2 媒体访问控制层

ZigBee 的 MAC 层的任务是为两个 ZigBee 设备的 MAC 层实体之间提供可靠的数据链路，处理所有物理层无线信道的接入。它通过公共部分子层服务接入点提供数据服务，通过管理实体服务接入点提供管理服务。MAC 层的主要功能包括：

1．通过 CSMA-CA 机制解决信道访问时的冲突；

2．发送或者检测、跟踪信标；

3．处理、维护和保护时隙（GTS）；

4．连接的建立和断开；

5．为数据通信的安全性提供支持。

一个完整的 MAC 帧（MPDU）由帧首部、帧载荷和帧尾三部分构成，其通用格式如图 2.7 所示。

<table>
<tr><td>2字节</td><td>1字节</td><td>0/2字节</td><td>0/2/8字节</td><td>0/2字节</td><td>0/2/8字节</td><td>可变</td><td>2字节</td></tr>
<tr><td rowspan="2">控制域</td><td rowspan="2">序列号</td><td>目的 PAN 标识符</td><td>目的地址</td><td>源PAN标识符</td><td>源地址</td><td rowspan="2">帧载荷</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="4">地址域</td></tr>
<tr><td colspan="6">MAC 帧首部</td><td>MAC 载荷</td><td>帧尾</td></tr>
</table>

图 2.7 MAC 层协议数据单元结构

1．帧控制域 帧控制域的长度为 16 位，其结构如图 2.8。

位序	0-2	3	4	5	6	7-9	10-11	12-13	14-15
标示	帧类型	安全允许控制	未处理数据标记	请求确认	PAN 内部标记	保留	目的地址模式	保留	源地址模式

图 2.8 帧控制域结构

(1) 帧类型子域定义了四种帧类型：信标帧(000)、数据帧(001)、确认帧(010)、命令帧(011)。

(2) 安全允许控制子域为 1 位，该位置 1，则对该帧进行加密处理后再传送到物理层；该位置 0，则直接传送到物理层，不进行加密。

(3) 未处理数据标记子域长度为 1 位，该位置 1，则表示除该帧数据外，本设备还有应发送给对方的数据，因此，接收该帧的设备应向本设备再次发送请求数据命令，直到所有的数据传送完毕。如果发送设备中已经没有要发送给接收方的数据了，就把该位置 0。

(4) 请求确认子域的长度为 1 位，该位置 1 时，接收方接收到有效帧后应向发送方发送确认帧，该位置 0 时接收方不需要发送确认帧。

(5) PAN 内部标记子域的长度为 1 位，该位置 1 时，表示该 MAC 帧在本身所属的 PAN 内传输，这时帧的地址域中不包含源 PAN 标识符；为 0 时，表示该帧是传输到另一个 PAN，帧中必须包含源节点和目的节点的 PAN 标识符。

(6) 目的地址模式子域的长度 2 位，它表示的意义如下：

00 PAN 标识符和地址子域不存在

01 保留

10 表示 16 位短地址

11 表示 64 位物理地址

(7) 源地址模式子域的长度 2 位，它表示的意义如下：

00 PAN 标识符和地址子域不存在

01 保留

10 表示 16 位短地址

11 表示 64 位物理地址

2．序列号域

序列号域的长度为 8 位，它是帧的序列标识，由设备自己的帧序列号发生器产生，采用循环计数方式，范围 0 到 0xFF。接收方可以根据此序列号判断接收的帧是否为新帧。

3．地址域

地址域长度 0 到 20 字节，它有四个子域：目的 PAN 标识符子域、目的地址子域、源 PAN 标识符子域和源地址子域。

(1) 目的 PAN 标识符子域

目的 PAN 标识符子域的长度为 16 位，它是接收该帧的设备所在 PAN 的唯一标识符。当标识符值为 0xFFFF 时，表示该帧为广播帧，即在同一信道上的所有设备都可以接收该帧。仅在帧控制子域的目的地址模式为非 00 时，该子域才存在。

(2) 目的地址子域

该地址是接收帧设备的地址。根据帧地址控制子域不同的情况，目的地址为 16 位或 64 位。地址 0xFFFF 是广播地址。同样，仅在帧控制子域的目的地址模式为非 00 时，该子域才存在。

(3) 源 PAN 标识符子域

源 PAN 标识符子域的长度为 16 位，它是发送该帧的设备所在 PAN 的唯一标识符。仅在帧控制子域的源地址模式为非 00 且内部 PAN 标记位为 0 时，该子域才存在。

(4) 源地址子域

该地址是帧发送设备的地址。根据帧地址控制子域不同的情况，目的地址为 16 位或 64 位。同样，仅在帧控制子域的源地址模式为非 00 时，该子域才存在。

4．有效载荷域

帧有效载荷即帧传送的数据，若帧的安全控制域值为 1，则载荷采用相应的加密方式进行处理。

5．帧校验子域

帧校验子域包含一个 16 位的 ITU-T CRC 校验码。

2.2.3 网络层

ZigBee 网络层主要实现节点加入或离开网络、接收或抛弃其它节点、路由查找及传送数据等功能[22]，但没有给出组网的路由协议，这样就为用户的使用提供了更为灵活的组网方式[23]。网络层通过使用 MAC 层提供的各种功能，保证 MAC 层各种功能的正确执行，完成建立、维护网络的任务，并向应用层提供服务。网络层的功能包括：

1．节点加入、离开网络的管理；

2．帧的安全机制管理；

3．根据路由发送帧到目的节点；

4．发现和维护路由；

5．发现邻居节点和维护邻居节点信息。

网络层内部在逻辑上由两部分组成，即网络层数据实体（NLDE）和网络层管理实体（NLME）。网络层数据实体通过访问服务接口（NLDE-SAP）为上层提供数据服务，在两个或多个设备的对等层之间实现协议数据单元的传输。其内容如下：

1．通过为应用支持子层协议数据单元 PDU 增加适当的协议信息，构造网络层协议数据单元 NPDU。

2．拓扑结构的特定路由：将 NPDU 传输到某设备，该设备可以是数据传输的最终目的，也可以是到达最终目的设备路由上的下一个设备。

3．安全：有能力保证传输的真实性和保密性。

网络层管理实体通过访问服务接口 NLME-SAP 为上层提供网络层的管理服务，另外还负责维护网络层信息库。其内容如下：

1．配置和初始化设备，保证该设备有能力完成它在网络中的功能。

2．如果设备是协调器，则它应能初始化并启动一个新网络。

3．如果设备是协调器或路由器，则它应能支持其它设备的连接，也可以要求某设备离开网络。如果设备是路由器或终端设备，则它应能实现与协调器和其他路由器的连接。

4．协调器或路由器应能够为设备分配网络地址。

5．应具备发现、报告和记录邻居设备的信息。

6．协调器和路由器应具备发现、记录通过网络有效传送信息的路由的能力。

7．应能控制设备的接收电路处于接收状态持续时间，使 MAC 层实现同步或直接接收。

2.2.3.1 网络的形成和维护

1．启动网络协调器

网络协调器 ZC（必须是 FFD）的网络层向 MAC 层发出请求，对所指定的信道或默认信道进行能量检测，以避开可能的干扰。能量检测完毕后，MAC 层将结果返回给网络层，网络层对检测的结果进行排序，把能量值超过了允许水平的信道丢弃，然后对剩余信道进行主动扫描，以获得设备所在区域内已有的各 ZigBee 网络的网络标识符(PANID)，网络层根据这些信息选择自己的 PANID，设置网络地址 0x0000 等信息，设置物理信道，启动网络。

2．允许设备接入网络

ZigBee 网络中的 ZC 和 ZR 都可以作为一个父节点允许设备接入网络。网络中的设备具有父设备和子设备的从属关系。当一个新设备向网络中已经存在的设备发出连接请求并建立了连接之后，新设备称为子设备，原来的设备成为父设备。

子设备可以通过请求连接方式和直接方式建立连接。

3．地址分配

ZigBee2004 与 2006 版本的地址分配机制有两种，即分布式的地址分配方式和高层地址分配机制。ZigBee Pro 版本中用随机地址取代了高层地址分配方式。

4．路由发现

路由发现是建立到达某一目的设备的路由的过程。ZigBee 网络中的相关设备相互合作，实现路由的发现及建立路由表。

2.2.3.2 网络层帧结构

网络层是网络层的协议数据单元（NPDU），通用结构如图 2.9 所示，它由

<table>
<tr><td>2 字节</td><td>2 字节</td><td>2 字节</td><td>0/1 字节</td><td>0/1 字节</td><td>长度可变</td></tr>
<tr><td rowspan="2">帧控制域</td><td>目的地址</td><td>源地址</td><td>广播半径</td><td>广播序列号</td><td rowspan="2">帧载荷</td></tr>
<tr><td colspan="4">路由域</td></tr>
<tr><td colspan="5">网络层帧首部</td><td>有效载荷</td></tr>
</table>

图 2.9 网络层的协议数据单元结构

两部分组成：网络层帧首部，包括帧控制、地址和序列信息等。

1．帧控制域 帧控制域长度为 16 位，其中包含了帧的类型、地址、序列号及其他一些信息，结构如图 2.10 所示。

0－1	2－5	6－7	8	9	10－15
帧类型	协议版本	路由发现	保留	安全性	保留

图 2.10 帧控制域结构

(1) 帧类型子域占 2 位，0x00 代表数据帧，0x01 代表命令帧，其它的保留。

(2) 协议版本子域长 4 位，表示 ZigBee 协议的版本。

(3) 路由发现子域长 2 位，0x00 表示 抑制路由发现

0x01 表示 使能路由发现

0x02 表示 强迫路由发现

(4) 安全子域长 1 位，当需要对网络层帧进行安全处理时，应将该子域置 1。

2．路由域包括目的地址子域、源地址子域、广播半径子域和广播序列号子域。

(1) 目的地址子域

目的地址子域长度为 2 字节，其值为 16 位的目的设备网络地址，或者是广播地址 0xFFFF。

(2) 源地址子域

源地址子域长度为 2 字节，其值为 16 位的源设备网络地址。

(3) 广播半径子域

广播半径子域长度为 1 字节，其值规定了广播帧的传输范围。在传播时，每个设备接收一次广播帧，将该域的值减 1。

(4) 广播序列号子域

广播序列号子域在每一个帧中都存在，长度为 1 字节。设备每发送一个新的帧，该值加 1。通常帧的序列号与它的源地址一起用来识别一个帧，以避免 1 字节长的序列号会产生混淆。

3．帧载荷域

帧载荷域的长度可变，它是帧需要发送的数据。

2.2.4 应用层

应用层包括应用支持子层（Application Support Sub-Layer，APS）、应用框架（Application Framework，AF）、ZigBee 设备对象（ZigBee Device Object，ZDO）。

应用支持子层主要功能：APS 层协议数据单元 APDU（APS Protocol Data Unit）的处理；节点间的应用对象绑定；组播群寻址；APS 数据传输机制。应用支持子层提供了网络层与应用层间的接口，ZigBee 设备对象与厂商定义的应用对象可以通过 APS 数据实体和 APS 管理实体使用 APS 子层提供的一系列服务。

ZigBee 设备中应用对象驻留的环境称为应用框架（Application Framework）。在应用框架中，应用程序可以通过数据实体服务访问接口（APSDE-SAP）发送接收数据，通过设备对象（ZDO）公共接口实现应用对象的控制与管理。应用框架为各个用户自定义的应用对象提供了模板式的活动空间，主要提供了端点管理、发送和接收数据两种功能。应用框架用于对 APS 子层数据服务传输来的帧进行过滤，AF 接收来自 APS 子层的数据，并传输至指定的端点和指定的配置。

ZigBee 设备对象（ZDO）提供应用对象、模板和 SAP 之间的接口，表示一类基本的功能。它处在应用框架和应用支持子层之间，满足 ZigBee 协议栈中所有应用操作的公共需求。设备对象通过端点 0，利用数据实体服务访问接口实现数据实体服务，利用管理实体服务访问接口实现管理服务。这些公共接口在应用框架中提供设备地址管理、发现、和安全功能，可以看成是一种公共的应用，提供一个公共的功能集，供用户自己定义的应用对象调用 APS 层的服务及 NWK 层的服务。主要功能包括设备网络启动、设备发现和服务发现、终端网络层管理服务。

第三章 ZigBee 路由算法分析比较

ZigBee 路由协议指的是 ZigBee 规范中规定的与路由相关的功能和算法部分，主要包括不同网络拓扑结构下 ZigBee 协议数据单元的路由方式、路由发现和路由维护等内容。

第一节 星形网络的建立和数据传输

ZigBee 星形网络是简单的一对多通信的网络。树形网络、网形网络可以看作是对星形网络的扩展，网络的建立过程类似。

3.1.1 星形网络的建立

这小节将详细描述星形网络的建立过程。

1．初始化设备。

首先，每个设备的 IEEE 802.15.4 的协议栈必须要对其 PHY 层和 MAC 层进行初始化的工作。

2．创建 PAN Coordinator。

每个网络必须有一个且仅有一个 PAN Coordinator。建立网络的第一步就是需要选择并且初始化这个 PAN Coordinator。初始化 PAN Coordinator 的动作只在相应的被事先约定的设备上进行。

3．选择 PAN ID 和 Coordinator 的短地址。

PAN Coordinator 一旦初始化完成就必须为它的网络选定一个 PAN ID 作为网络标识，PAN ID 可以被人为的预定义。每一个设备已经具有了一个唯一固定的 64 位 IEEE MAC 地址，通常叫做扩展地址。但是作为组网的标识，还必须分配一个 16 位的网络地址，通常叫做短地址。使用短地址进行通讯可以使网络通讯更加高效。这个短地址是预先定义的。

4．选择射频信道。

PAN Coordinator 必须为网络选择一个射频通道。PAN Coordinator 可以进行一次能量扫描检测来找到一个相对安静的通道，用这个通道建立自己的无线网

络。通过能量扫描检测 API 将返回每一个射频通道的能量水平，能量水平高就意味着这个通道的无线信号活跃。

5．启动网络。

一个无线网络的启动过程是从初始化配置 PAN Coordinator 开始的，这个设备以 PAN Coordinator 的模式启动。然后 PAN Coordinator 就将开放对于加入网络的请求应答。

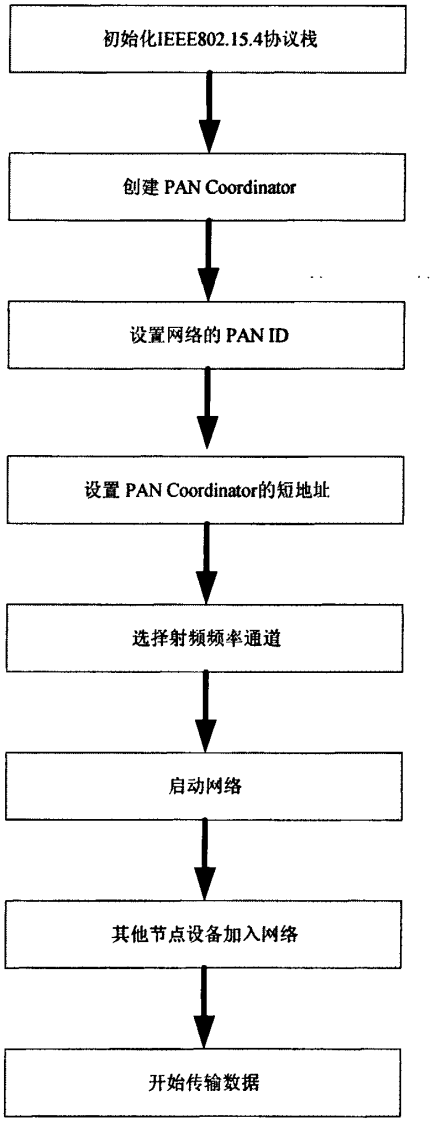

图 3.1　网络建立流程图

6、设备加入网络。

一旦网络中出现了可以利用的 Coordinator，其他的网络设备就可以作为 End Device 加入网络了。一个设备如果需要加入网络，首先要完成自己的初始化过程，然后他需要找到 PAN Coordinator。为了找到 PAN Coordinator，设备需要进行频道扫描，它将在特定的频率通道中发送信标请求， PAN Coordinator 检测到信标请求后，将回应相应的信标来向该设备标识自己。信标网络中，PAN Coordinator 将周期性的发送信标，需要加入网络的设备可以被动侦听来自 PAN Coordinator 的信标。

设备找到 PAN Coordinator 之后就发出加入网络的申请， Coordinator 决定接受或拒绝设备加入。如果接受了设备，它将发送一个 16 位的短地址给设备，作为该设备在网络中的标识。

网络建立过程流程如图 3.1 所示。

3.1.2 星形网络的数据传输

当网络中出现了 PAN Coordinator 和至少一个 End Device 后，网络就可以进行数据传输了。

Coordinator 向 End Device 传输数据有两种模式：直接传输模式和间接传输模式。

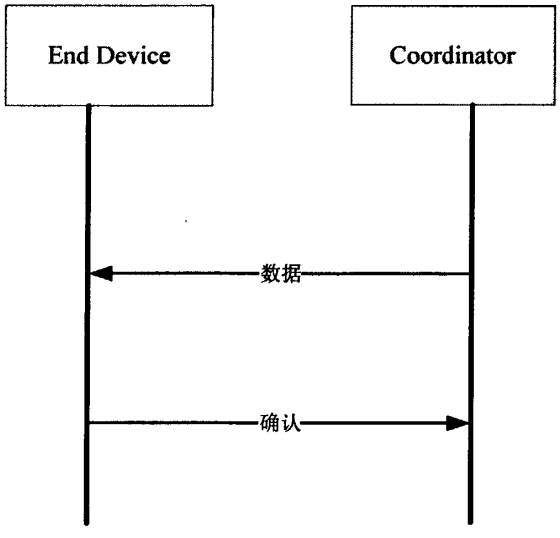

图 3.2 直接传输模式

直接传输模式：PAN Coordinator 可以将数据直接发送给 End Device。End Device 接收到数据后发送确认消息给 Coordinator。这种传输方式要求 End Device 随时都处于数据接收的状态，也就是要求其随时处于唤醒状态。过程如图 3.2 所示。

间接传输模式：Coordinator 将数据保存起来等待 End Device 请求读取数据。采用这种方式，End Device 为了获得数据必须先发送数据请求，收到数据请求后，Coordinator 判断是否有需要发送给这个设备的数据，如果有就发送相应的数据，End Device 收到数据后发送确认消息。这种方式适用于 End Device 需要降低功耗的情况，其大多数时候处于休眠状态以节省能量。过程如图 3.3 所示。

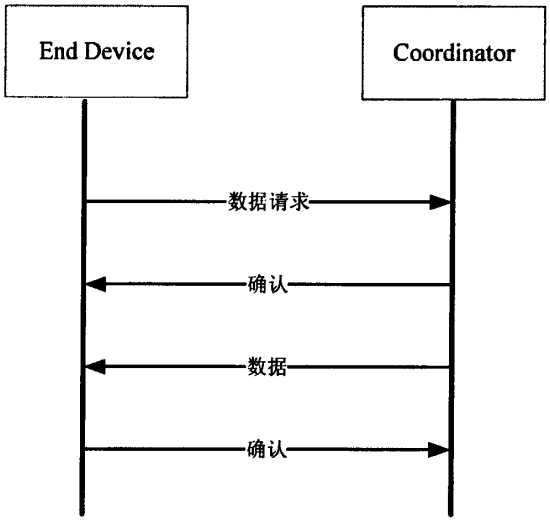

图 3.3 间接传输模式

End Device 向 Coordinator 传输数据通常是直接发送数据，Coordinator 接到数据后给 End Device 发送确认消息。

3.1.3 代码描述

网络协调器 Coordinator 代码描述：

AppColdStart()会调用 vStartEnergyScan()，这个函数将会开始在各个通道进行能量扫描以获得各个通道的能量级别。

AppColdStart()将通过调用 vProcessEventQueues()的方式等待 MLME 的回应。

vProcessEventQueues()函数将检查三个不同类型的事件队列并将接到的事件交给不同的事件处理函数处理。

AppColdStart 将循环调用 vProcessEventQueues()函数处理来自于 MCPS 的消息队列和来自于硬件的消息队列。

当数据到达 MCPS 队列后，vProcessEventQueues()首先调用函数 vProcessIncomingMcps()来接收到达的数据帧。

当硬件事件到达硬件队列后，vProcessEventQueues()将调用函数 vProcessIncomingHwEvent() 来接收到来的事件 。

End Device 的运行过程和 Coordinator 完全不同：

End Device 运行过程仍然从 AppColdStart 开始，AppColdStart 调用 vInitSystem，这个函数将初始化 IEEE 802.15.4 的协议栈。

AppColdStart 调用 vStartActiveScan()开始对于活动通道的扫描，End Device 将向扫描的通道发送信标请求，并接收 PAN Co-ordintor 的信标请求回应。扫描请求的初始化和发送的工作通过 MLME 请求的方式通过 MAC 层发送。

AppColdStart()将通过 vProcessEventQueues()来检查和处理 MLME 回应。这个函数将调用 vProcessIncomingMlme()来处理收到的 MLME 回应。vHandleActiveScanRespose()会被调用处理返回的活动通道扫描结果：

如果找到 PAN Coordinator，函数将保存相应的 Coordinator 信息，并调用 vStartAssociate()向 Coordinator 提交入网请求，这一请求将通过 MLME 请求的方式提交。

如果 PAN Coordinator 没有找到，这个函数将重新调用 vStartActiveScan()启动扫描。

AppColdStart 将循环的调用 vProcessEventQuenes()等待来自 Coordinator 的入网回复。收到回复后就将调用 vProcessIncomingMlme()，然后调用 vHandleAssociateRespose()来处理回复，接下来的函数将检查回复的状态：

如果 Coordinator 接受入网请求，将设备置于联网状态，如果 Coordinator 拒绝入网请求，函数调用 vStartActiveScan()搜索另外一个 PAN Coordinator。

AppColdStart()接下来将循环调用 vProcessEventQuenes()等待来自于 PAN Coordinator 的 MCPS 的信息或者硬件队列的信息。当数据到达 MCPS 队列，vProcessEventQueue ()首先使用函数 vProcessIncomingMcps()来接收数据帧，接着调用 vProcessRecedDataPackt()。

当硬件事件到达硬件事件队列，vProcessEventQuenes()将调用 vProcessIncomingHwEvent()来接收到达的事件。

第二节 树路由算法

树形网络中，每一个全功能设备都可以成为父节点，精简功能设备只能作为子节点。树形网络常采用树形路由机制。树形路由机制包括地址分配机制和树路由算法[24]。树路由算法原理来源于分布式地址分配策略。

3.2.1 分布式地址分配策略

在使用分布式地址分配的网络中，当协调器 Coordinator 建立一个新的网络，它将给自己分配网络地址 0，网络深度 Depth0=0。网络深度表示仅仅采用父子关系的网络中，一个传送帧传送到协调器所传递的最小跳数，协调器自身深度为 0，而它的子设备深度为 1。如果节点（i）想要加入网络，并且与节点（k）连接，那么节点（k）将称为节点（i）的父节点。根据自身的地址 A_k 和网络深度 Depthk，节点(k)将为节点(i)分配网络地址 A_i 和网络深度 Depthi=Depthk+1。

网络地址的分配与三个参数有关：参数 nwkMaxChildren(Cm)表示路由器或协调器在网络中允许拥有子设备数量的最大值。参数 nwkMaxRouters(Rm)表示子节点中路由器的最大个数，而剩下的设备数为终端设备数。参数 nwkMaxDepth (Lm)表示网络的最大深度。

根据 3.1 式和这三参数可以算出在每一级(也就是每个深度的)情况下，兄弟节点之间的地址间隔(Cskip)。

$$\mathrm{Cskip}(d)=\begin{cases}1+C_m\cdot(L_m-d-1) & (R_m=1)\\ \dfrac{1+C_m-R_m-C_m\cdot R_m^{L_m-d-1}}{1-R_m} & others\end{cases} \qquad (3.1)$$

如果一个设备的 Cskip（d）大于 0，则允许新设备作为子节点接入网络，并为其分配地址。一个新的 RFD 节点（i），它没有路由能力，它与协调器连接作为协调器的第 n 个子节点。根据 3.2 式和它的深度 d，父节点（k）将为子节点（i）分配网络地址 ：

Ai = Ak + Cskip（d）· Rm + n 其中 1≤n≤（Cm-Rm） （3.2）

如果新的子节点是 FFD，它有路由能力，父节点（k）将根据 3.3 式给它分配网络地址：

$$Ai = Ak + 1 + Cskip(d) \cdot (n-1) \quad (3.3)$$

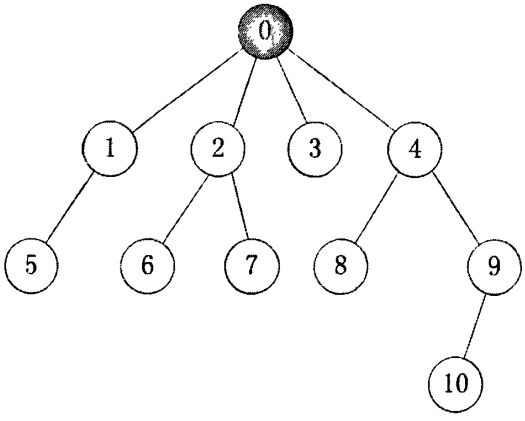

图 3.4 树形网络节点地址分配示意图

如图 3.4 所示的网络，其参数 nwkMaxChildren = 4，nwkMaxRouters= 4，nwkMaxDepth = 3，每个深度的 Cskip 为:

节点 0 的深度为 0， Cskip 为 21；节点 1、2、3、4 深度为 1， Cskip 为 5；节点 5、6、6、8、9 的深度为 2， Cskip 为 1；节点 10 的深度为 3，Cskip 为 0，因此，节点 10 不能允许新设备成为其子节点。根据公式 3.2、3.3 给各个节点分配地址：节点 1 地址为 1，节点 2 地址为 22，节点 3 地址为 43，节点 4 地址为 64，节点 5 地址为 2，节点 6 地址为 28。

3.2.2 树路由算法

树形拓扑结构的路由通常简化为上行路由(route up)或下行路由(route down)。假设一个路由器向网络地址为 DestAddr 的目的地址发送数据包，路由器的网络地址为 LocalAddr，网络深度为 d。

$$LocalAddr < DestAddr < LocalAddr + CSkip(d-1) \quad (3.4)$$

表达式成立，则表明目的节点是自己的后代，为下行路由；否则为上行路由。

如果能够确定目的节点是本设备的后代，则下一跳节点的地址 NextAddr 为：

$$NextAddr=\begin{cases} DestAddr & ,如果是终端设备 \\ LocalAddr+1+\left[\frac{DestAddr-(LocalAddr+1)}{Cskip(d)}\right]\times Cskip(d) & ,其它 \end{cases} \quad (3.5)$$

否则，下一跳节点是该路由器的父节点。

Cluster-Tree 路由适合于节点静止或者移动较少的场合，属于静态路由，不需要存储路由表。Cluster-Tree 路由对传输数据包的响应较快，因为 Cluster-Tree 路由不需要建立路由表，不需要进行路由发现[25]。其缺点是所选择的路由并非是最佳的路由，不能获得最小跳数的路由。因此，Cluster-Tree 路由适用于爆发型的数据传输。

第三节 Z-AODV 路由算法

网形网络采用一种按需距离矢量的路由算法 Z-AODV 与树路由相结合的混合路由方式。Z-AODV 路由算法由 AODV 路由算法改进的，提高了扩展性能[26]。

3.3.1 AODV 路由算法

AODV 路由算法是一种基于距离矢量的按需路由算法[27]，它的路由发现和路由维护过程中使用的命令帧有：路由请求命令帧（RREQ）、路由应答命令帧（RREP）、路由修复命令帧（RERR）、HELLO 帧，命令帧采用逐跳转发的方式，每个中间节点隐式保存了路由请求和回复的结果。HELLO 帧获取邻居节点的信息[28]，RREQ 帧发起路由请求进行路由发现，RREP 帧进行路由应答，返回路由信息，RERR 表示链路出现问题。

AODV 路由算法主要有 4 个基本过程[29]：

1．路由发现：路由的初始阶段，用来发现路由；

2．路由建立：通过这个过程，源节点和中间节点明确某条路由的下一跳节点；

3．路由维护：维护路由的时效性，及时地删除过时或是断开的路由；

4．邻居管理：通过周期地发送 Hello 消息，确定出在发射功率范围内的节点。

当一个节点发送数据时，如果在它的路由表中没有找到目的节点的路由，它就广播一个路由请求帧（RREQ），发起一个路由发现过程，因此数据分组的发送有一定时间的延迟。收到 RREQ 的节点会在自己的路由表中添加以源节点为目的节点的表项，并将 RREQ 广播出去，节点会抛弃收到的重复 RREQ。通过这种泛洪的方式，RREQ 会被广播到网络的每一个节点。目的节点或是拥有到达目的节点路由的中间节点，在收到 RREQ 后，会沿着广播路径逆向发送一个路由应答（RREP）给源节点。中间节点和源节点收到 RREP 后，就建立了到达目的节点的路由[30]。AODV 路由采用逐跳的方式转发分组，路由表中记录了到目的节点的下一跳地址，因此不需要在数据帧中携带完整的路由信息。

3.3.2 Z-AODV 路由特点

Z-AODV 算法是对 AODV 算法的改进，保持了 AODV 的原始功能[31]，但为了降低路由成本、节能性等因素，简化了 AODV 的一些功能[32]。

1．AODV 中节点周期性给邻居节点发送 Hello 分组做路由维护，Z-AODV 中节点不需要发送 Hello 分组，节省了控制开销。

2．Z-AODV 在路由发现的过程中，只有目的节点回复 RREP，减少了控制开销，简化了路由发现过程，在网络拓扑变化时可以快速收敛[33]。

3.3.3 路由表和路由发现表

ZigBee 协调器和路由器建立和维护自己的路由表。路由表中有若干个路由项，每个路由项包含的信息如表 3.1 所示。路由表应该保留一些空闲空间，作为增加路由或路由修复之用。

表 3.1 路由表

域名	大小	描述
Destination address	2 字节	该路由目的节点的 16 位网络地址
Status	3 位	该路由的状态
Next-hop address	2 字节	到目的节点路由的下一跳节点 16 位网络地址

表 3.2 种列出了路由状态的各种可能取值。

表 3.2 路由状态值表

状态	数值	说明
ACTIVE	0x00	该路由可用
DISCOVER_UNDERWAY	0x01	正在搜索发现路由
DISCOVER_FAILED	0x02	发现路由失败
INACTIVE	0x03	该路由不可用
RESERVED	0x04 – 0x07	保留

一个设备如果能借助于它的路由表建立起到达某一目的设备的路由，则该设备被称为具有路由能力，它应该具备以下特性：

1．是一个协调器或路由器；

2．维护有路由表；

3．路由表中有空闲的位置，或者已经建立有到达目的设备的路由；

4．可以进行路由修复，并且路由表中保留有为此目的的空闲空间。

此外，协调器和路由器中还有一个路由发现表，表中信息如表 3.3 所列。与路由表不同的是，路由表中的路由项是长期保留存在的，而路由发现表的表项仅在路由发现过程中存在，并且可以重新生成。

表 3.3 路由发现表

域名	长度/字节	描述
Route request ID	1	路由请求命令帧的序列号，设备启动一次路由请求，序列号加 1
Source address	2	路由请求发起者的 16 位网络地址
Sender address	2	对应于入口的路由请求标识符和源地址，发送最新的、最低的路由请求命令帧的设备 16 位网络地址。这个域常用来确定最终路由应答帧所经过的路由
Forward cost	1	从路由请求源设备到当前设备所积累的路由成本
Residual cost	1	下一跳设备到目的设备所需要的路由成本
Expiration time	2	减法计数器。它以 ms 为单位，其初始值为 nwkcRouteDiscoveryTime，计数到 0 表示时间耗尽

如果满足下面条件，则设备具有路由发现能力：

1．设备是一个协调器或路由器；

2．维护有一个路由发现表；

3．路由发现表中有空闲空间；

Z-AODV 路由算法把节点分成两类：具有路由能力和路由发现能力的 RN+

节点，没有路由发现能力的 RN-节点。Z-AODV 路由算法允许 RN+节点发起一个路由发现过程，找到一条最小路径，而 RN-节点只能执行沿树路由。

3.3.4 命令帧格式

网络层帧有数据帧和命令帧两种类型[34]，数据帧的结构和通用帧结构完全相同。其帧控制域的类型子域为 0x00，以表明这是个数据帧。

命令帧的帧控制域的帧类型子域为 0x01，以表明这是一个命令帧。而帧的有效载荷域的第一个字节是网络层命令标识符，如图 3.5 所示。

2 字节	6 字节	1 字节	可变长
帧控制域	路由域	命令标识符	命令载荷
网络层帧首部		网络层载荷	

图 3.5 命令帧的结构

命令标识符的取值和命令名称如表 3.4 所示：

表 3.4 命令标识符的取值和命令名称表

命令帧标识符	命令名称
0x01	路由请求
0x02	路由应答
0x03	路由错误
0x04	离开
0x00，0x05-0x0FF	保留

1．路由请求命令帧

路由请求命令帧用来发起一个路由发现过程，它的帧载荷部分如图 3.6 所示。

1 字节	1 字节	1 字节	2 字节	1 字节
命令帧标识符	命令选项	路由请求标识符	目的地址	路由成本
网络层载荷				

图 3.6 路由请求命令帧结构

为了发送路由请求帧，网络帧首部中的帧类型子域值设置为 0x01，必须把网络层帧首部的源地址设置为初始发送该帧的设备地址，目的地址设置为广播地址。命令帧标识符的值为 0x01。命令选项的低 7 位保留，最高位称为路由修复位，仅在网状拓扑结构中，产生路由请求命令时设置为 1。路由请求标识符是

一个 8 位的序列号，每发出一次路由请求命令，其值加 1。路由成本用来累计路由请求帧在网络中传输的成本信息。

2．路由应答命令帧

路由应答命令用来返回路由信息，它的帧载荷部分如图 3.7 所示。

1 字节	1 字节	1 字节	2 字节	2 字节	1 字节
命令帧标识符	命令选项	路由请求标识符	源地址	应答地址	路由成本
网络层载荷					

图 3.7 路由应答命令帧结构

网络帧首部中的帧类型子域值设置为 0x01，目的地址设置为初始发出路由请求帧的设备地址，源地址设置为传送该应答帧的设备地址。命令帧标识符值为 0x02，命令选项的低 7 位保留，最高位称为路由修复位，仅在网状拓扑结构中，产生路由应答时设置为 1。路由请求标识符是所应答的路由请求命令的标识符。路由成本用来标示路由请求帧到达目的设备所用的传输成本信息。

3．路由错误命令帧

当设备无法继续向前传送数据时，发出路由错误命令帧通知发送数据帧的设备。路由错误命令帧载荷结构如图 3.8 所示。

1 字节	1 字节	2 字节
命令帧标识符	错误代码	目的地址

图 3.8 路由错误命令帧结构

网络帧首部的目的地址项设置为初始发出路由请求帧的设备地址，源地址设置为传送此应答帧设备的地址。错误代码的取值及涵义如表 3.5 所示。

表 3.5 路由错误代码取值表

错误代码	描述
0x00	无有效路由
0x01	树形链路失败
0x02	非树形链路失败
0x03	电池电压低
0x04	无路由能力
0x05 -0xFF	保留

3.3.5 路由基本算法

当节点网络层收到一个数据帧之后，对其进行处理。如果网络层从更高层接收到数据帧并且数据帧的目的地址和广播地址一致，那么节点将数据帧广播发送。如果给节点是路由器或者协调器，同时数据帧的目的节点是一个终端设备，并且正是该节点的子节点，那么这个数据帧将直接传送到目的地址，并且设置下一跳目的地址和最终目的地址一致。一个有路由能力的节点会首先检查路由表中，如果存在到目的地址的路径则按该路径发送数据帧；如果没有到达目的地址路径，那么没有路由发现能力的节点会将数据帧沿着树形路径发送，有路由能力的节点将发起一个路由发现过程，找到到达目的地址的路由，整个处理过程如图 3.9。

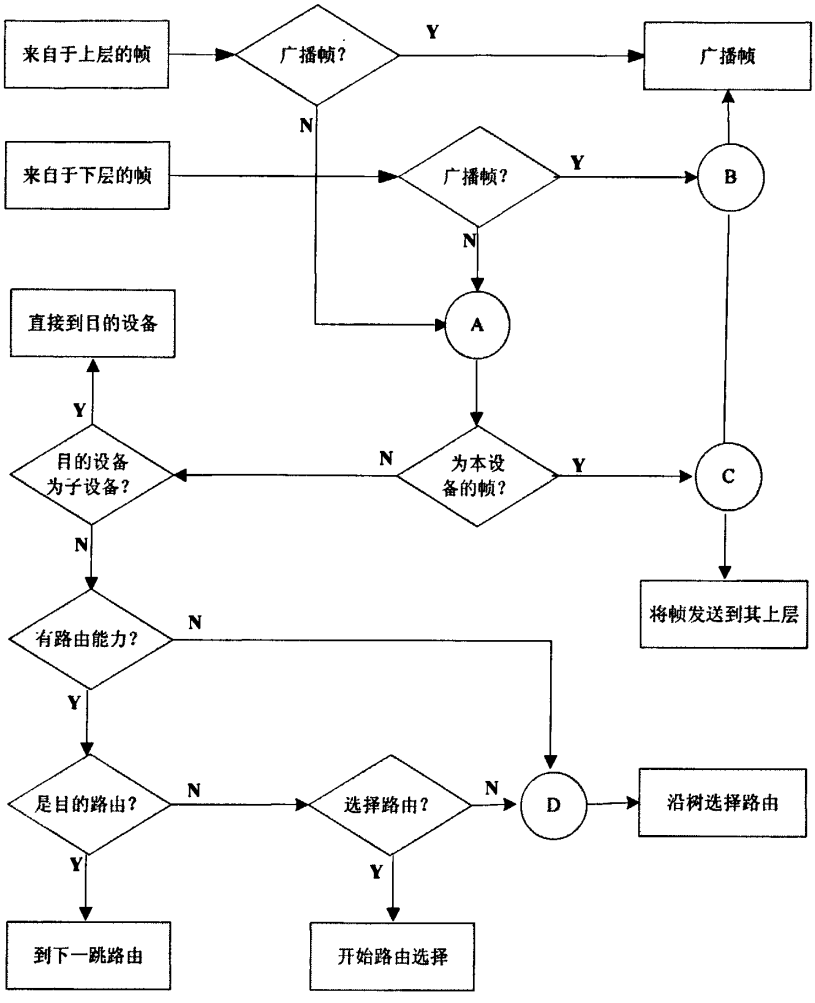

图 3.9　基本路由算法

3.3.6　路由发现过程

当一个 ZigBee 协调器或者路由器需要传输数据，而它的路由表中没有到达目的设备的路径时，设备需要启动一个路由发现过程来建立一个到达目的设备的路径。路由发现过程是网络设备通过网络层的合作来发现并建立路由的过程，而且在被执行时总是和一对特定的发起节点、目的节点相关。

路由发现过程如下：

1．路由发现过程的发起

对于一个具有路由能力的节点，在符合以下条件时，网络层应当发起路由发现过程：接收到一个从更高层发出的发送数据帧的请求，而路由表中没有和目的地址对应的条目；或者接收到一个来自媒体接入控制层的帧，帧头部的目的地址域中包含的目的地址并非本节点地址或广播地址，并且路由表中没有和目的地址对应的条目。

如果一个节点发起了路由发现过程，它就应当建立相应的路由表条目和路由发现表条目，状态设置为路由发现中，并且路由发现过程发起节点将广播路由请求命令帧。该命令帧中包含发起节点的地址、目的节点地址等信息。

2．接收到路由请求命令帧

目的节点收到路由请求命令帧(RREQ)之后的具体处理过程如图 3.10 所示：

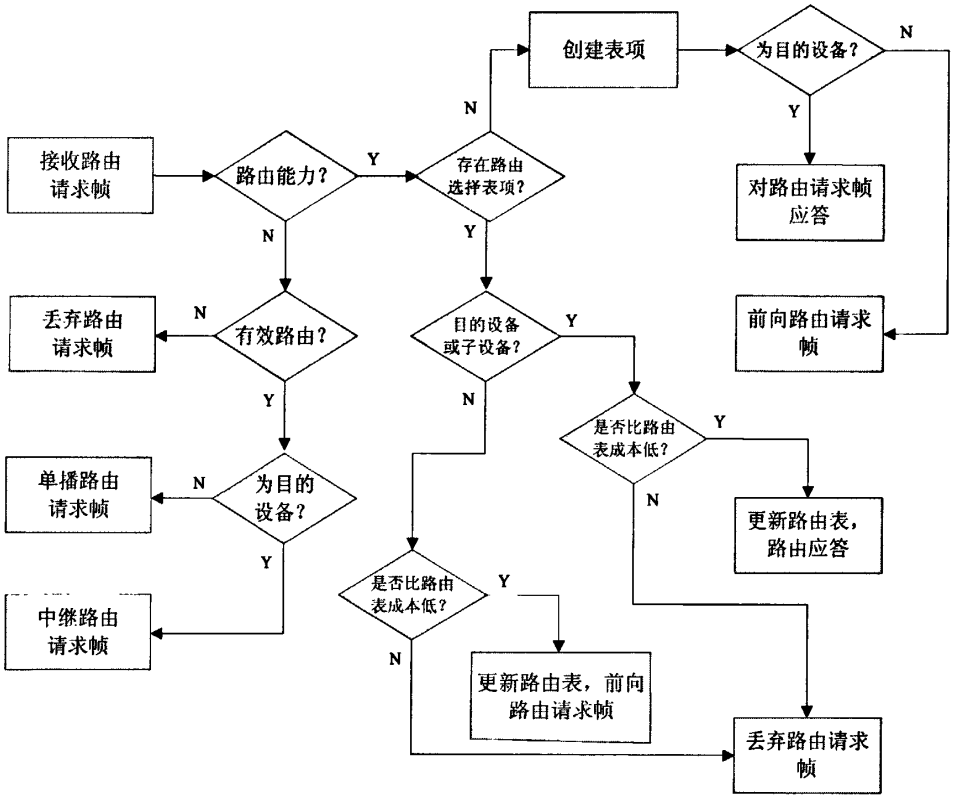

图 3.10　路由请求命令的处理流程

3．接收到路由应答命令帧

中间节点或者路由发现过程的发起节点接收到路由应答命令帧（RREP）的处理过程如图 3.11 所示。路由发起过程的发起节点接收到来自多条路径的 RREP 时，该节点将选择累积路径损耗最小的路径作为到目的节点的路由。累积路径损耗最小的路径不唯一时，路由发起节点将选择最早接收到的 RREP 所对应的路径。

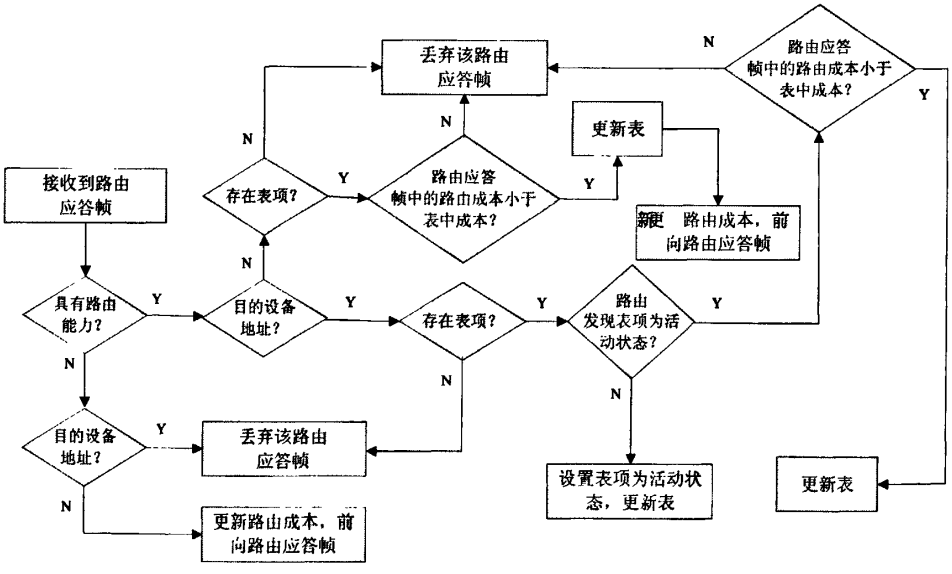

图 3.11　对路由应答命令帧的处理流程

另外，除了发起和完成路由发现过程之外，网络层的路由功能还包括路由维护过程。每个节点都要为它的每个邻居节点维护一个失败计数器，当计数器超过一定数值或者路由失败时，节点就要发起路由修复过程。路由修复过程与路由发现过程类似。为了避免过多的网络开销，往往在实际应用中不会频繁地发起路由修复过程。

第四章 ZigBee 路由算法仿真

网络仿真是进行网络研究的一个基本手段。在网络新技术的研究过程中，由于各种原因，实际网络系统的实现往往是代价较高或者是不现实的。在这种情况下，仿真就成了最佳可供选择的测试、评估和验证手段了。网络仿真有周期小、成本低等优点，使研究者更容易获得研究成果。

当前应用比较多的几种网络仿真工具包括 NS2（Network Simulator 2）、OPNET、QualNet 等。其中 OPNET、QualNet 支持图形用户界面，具有使用简单等优点，但由于属于商业软件，价格比较昂贵。NS2 是由美国加州大学伯克利分校 LBL、Xerox PARE UCB 和 USC/ISI 共同开发的免费网络仿真工具，并且源代码开放。

第一节 NS 仿真软件介绍

NS2 是一种针对网络技术的源代码公开、免费的软件模拟平台，研究人员使用它可以很容易地进行网络技术开发和研究[35]，而它所包含的模块几乎涉及到网络技术的所有方面。因此，NS2 成为目前学术界广泛使用的网络模拟软件。通过这种方法获得的研究结果也是被学术界普遍认可。

4.1.1 NS 仿真软件介绍与原理概述

本文在对 ZigBee 网络路由算法的研究中，选择 NS2 网络仿真作为主要研究手段。

1．离散事件模拟器

NS2 是一个离散事件模拟器。事件规定了系统状态的改变、状态的修改仅在事件发生时进行。模拟时钟的推进由事件发生的时间量确定。模拟处理过程的速率不直接对应着实际时间。一个事件的处理可能又会产生后续的事件，模拟器所做的就是不停地处理一个个事件，直到所有的事件都被处理完或者某一特定的事件发生为止。典型的事件包括分组到达、时钟超时等。

NS2 的核心部分是一个离散事件模拟引擎。NS2 中有一个“调度器”

（Scheduler）类，负责记录当前时间，调度网络事件队列中的事件，并提供函数产生事件，指定事件发生的时间。

NS2 是一个事件（event）驱动的模拟器，目前 NS2 支持两种类型的事件调度器：非实时的（none real-time）和实时的（real-time）。事件调度器的主要功能是处理分组（packet）的延迟以及充当定时器。一个事件调度器的执行过程是这样的：从所有事件中选择一个发生时刻最早的事件，调用它的 handler 函数，把该事件执行完毕，然后从剩余的所有事件中选择发生时刻最早的事件执行，如此反复执行。NS2 只支持单线程，故在某一时刻只能有一个事件在执行，如果有多个事件被安排在同一时刻，那么会按照事件代码插入的先后次序执行。

2．NS2 构件库

NS2 已经预先进行了大量的模型化工作，对网络系统中的一些通用实体进行了建模，并用对象来实现这些实体的特性和功能，这些对象组成了 NS2 的构件库。NS2 的构件库十分丰富，并且支持广域网、局域网、移动通信网、无线自组织网等各种类型网络以及各种路由方式。

NS2 的基本构件包括节点、链路、代理、包和队列等。

(1) 节点（Node）是由 TclObject 对象组成的复合组件，在 NS2 中可以表示端节点和路由器。节点的属性包括节点的地址类型、移动节点各个网络构件的类型、移动节点路由协议类型、是否打开各层的 Trace 功能等。

(2) 链路（Link）由多个组件复合而成，用来连接网络节点。所有的链路都是以队列的形式来管理分组的到达、离开和丢弃。

(3) 代理（Agent）负责网络层分组的产生和接收，也可以用在各个层次的协议实现中。每个代理连接到一个网络节点上，由该节点给它分配一个端口号。

(4) 包（Packet）由头部和数据两部分组成。一般情况下，包只有头部、没有数据部分。

(5) 队列（Queue）是用于保存或者丢弃包的存储空间。

3．NS2 仿真结果的输出

NS2 利用 Trace 文件输出仿真结果。通过编程设定，Trace 文件可以记录模拟过程中事件发生的时间、相关参数以及节点状态等信息，是对仿真结果进行研究的主要数据基础。

另外，NS2 还提供了一些实用工具，例如 NAM，可以对模拟过程进行图形动画演示。

4.1.2 用 NS2 进行网络仿真的过程

使用 NS2 网络仿真工具进行网络研究的一般方法包含两个层次：一个是基于 Otcl 编程的层次，利用 NS2 已有的网络元素实现仿真，无需对 NS2 本身进行修改，只有编写 Otcl 脚本，就可以完成网络仿真实验；另一个层次是基于 C++和 Otcl 编程的层次，需要对 NS2 进行扩展，添加新的 C++类和 Otcl 类，然后再编写 Otcl 脚本，完成仿真实验。

用 NS2 进行网络仿真的一般过程：

1．编写 Otcl 脚本。首先配置仿真网络拓扑结构，此时可以确定链路的基本特性，如延迟、带宽和丢失策略等。

2．建立协议代理，包括端设备的协议绑定和通信业务量模型的建立。

3．配置业务量模型的参数，从而确定网络上的业务量分布。

4．设置 Trace 对象。Trace 对象能够把仿真过程中发生的特定类型的事件记录在 trace 文件中。NS2 通过 trace 文件保存整个仿真过程。仿真完成后，用户可以对 trace 文件进行分析研究。

5．编写其他的辅助过程，设定仿真结束时间，至此 Otcl 脚本编写完成。

6．用 NS2 解释执行刚才编写的 Otcl 脚本。

7．对 trace 文件进行分析，得出有用的数据。也可以用 NAM 等工具观看网络仿真运行过程。

第二节 路由算法仿真分析

用 NS2 软件做仿真测试，仿真中提取的各项性能指标定义如下：

1．分组平均递交率=目的节点接收到的数据包个数/源节点发送的数据包个数；

2．平均跳数=仿真中所有数据包被转发的次数和/接收到的数据包个数；

3．平均端到端时延=所有成功递交的数据分组的端到端时延的平均值。

ZigBee 星形网络是一种最简单的树形网络。而在 ZigBee 网形网络中，由于可以通过对节点进行参数设备来使得网络采用单纯的树路由或者 Z-AODV 路由，因此树形网络是网形网络的一个子集。本文选用三种情况的网络作为仿真场景：星形网络、协调器在网络边缘的网形网络、协调器在网络中心的网形网络。在

这三个场景中分别对树路由算法和 Z-AODV 路由算法进行仿真。

4.2.1　星形网络的仿真

此次仿真采用的星形网络拓扑如图 4.1 所示。

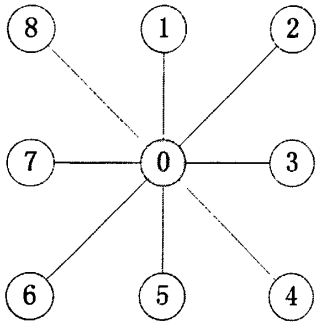

图 4.1　星形网络

图中连线表示父子关系。节点 0 为 ZigBee 网络的协调器，所有节点都是 RN+。设置仿真场景为 50m×50m 的正方形区域，仿真时间为 100 秒，采用 CBR 业务源，每个数据源发包速率为 10Packet/s，每个数据包 100 字节，节点的有效传输距离为 15m。根据 NS2 仿真数据得到图 4.2、图 4.3 和图 4.4。

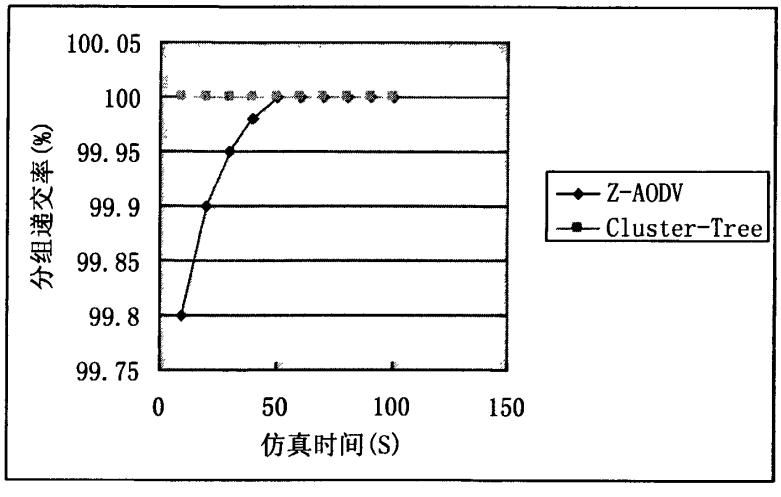

图 4.2　星形网络分组递交率

从图 4.2 中可以看出，Z-AODV 路由算法在仿真初期分组递交率低于

Cluster-Tree 路由算法，随后逐渐提高。这是因为在初始阶段，Z-AODV 需要进行路由发现，大量的路由发现帧可能产生碰撞，导致数据丢帧，路由建立以后，不再需要路由发现，因此分组递交率提高了。

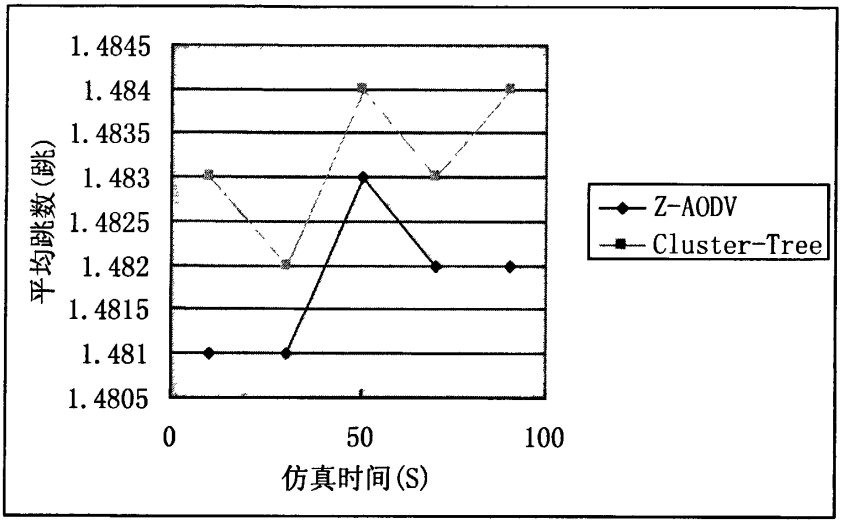

图 4.3　星形网络平均跳数

从图 4.3 中可以看出，Z-AODV 路由算法的平均跳数稍低于 Cluster-Tree 算法。由于仿真的星形网络结构简单，所以相差很小。

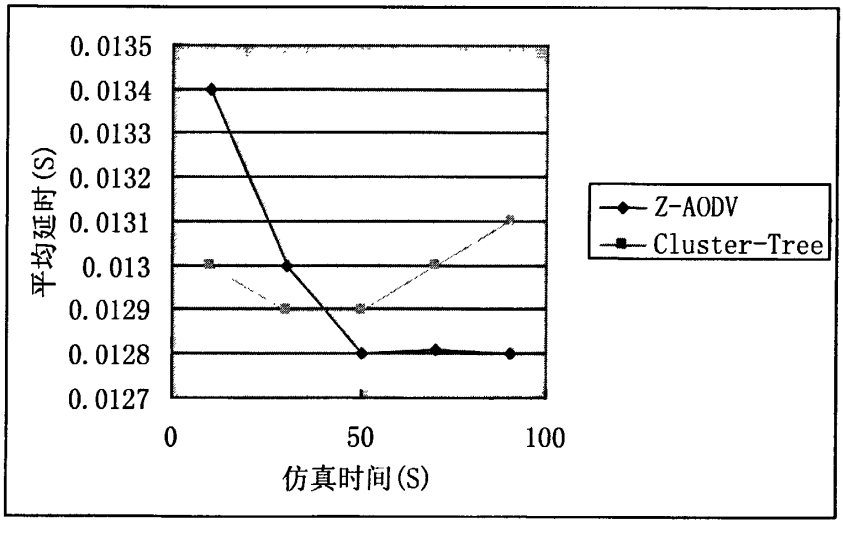

图 4.4　星形网络平均延时

从图 4.4 中可以看到，在仿真初期，Z-AODV 路由算法的延时要大于 Cluster-Tree 路由算法，而后期减少，小于 Cluster-Tree 算法。这是因为在仿真初期，Z-AODV 路由算法要进行路由发现，因而有较多的延时，当路由建立后，平均跳数稍低于 Cluster-Tree 算法，因而延时较低。

4.2.2 协调器在网络边缘的仿真

图 4.5 是一个 15 个节点的网络，图中的连线表示父子关系。节点 0 为 ZigBee 网络的协调器，偶数节点都是 RN+，奇数节点是 RN-。仿真时间为 100 秒，采用 CBR 业务源，每个数据源发包速率为 10Packet/s，每个数据包 100 字节，节点的有效传输距离为 15m。

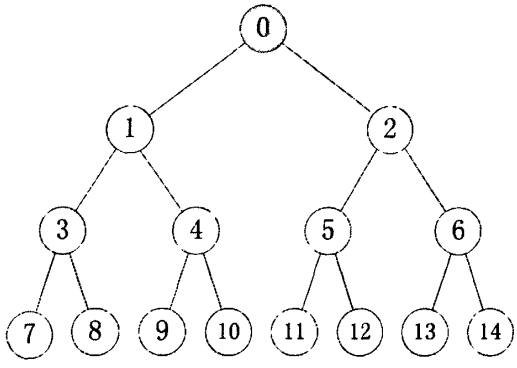

图 4.5 协调器在网络边缘的树形网络

根据仿真数据得到图 4.6、图 4.7 和图 4.8。

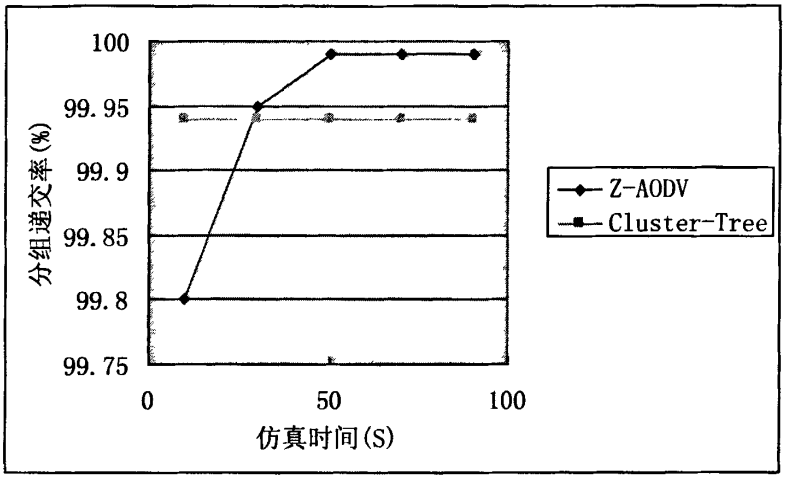

图 4.6 协调器在网络边缘的树形网络分组递交率

从图 4.6 可以看出，Z-AODV 路由算法在仿真初期分组递交率低于 Cluster-Tree 路由算法，而在后期高于 Cluster-Tree 路由算法。这是因为在初始阶段，Z-AODV 需要进行路由发现，大量的路由发现帧可能产生碰撞，导致数据丢帧，路由建立以后，不再需要路由发现，因此分组递交率提高了。而 Cluster-Tree 路由算法中数据传输只能在父子节点间进行，深度低的节点需要处理较多的数据分组，产生的碰撞可能性增加，导致丢包。

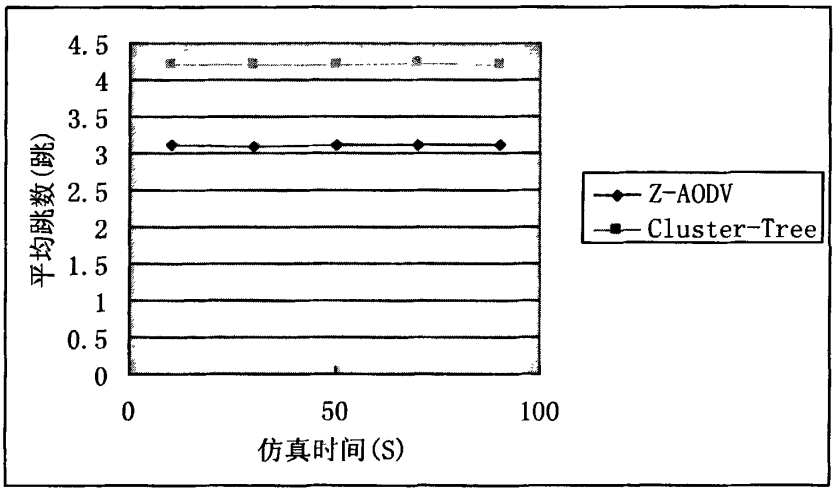

图 4.7 协调器在网络边缘的树形网络平均跳数

从图 4.7 可以看出，与 Cluster-Tree 路由算法相比，Z-AODV 路由算法有较低的平均跳数。这是因为 Cluster-Tree 路由算法的数据传输只能沿树形父子传输，多数情况下所经过的路径都不是最短路径。而 Z-AODV 路由算法可以选择最小跳数的路径，大大降低了路径的平均跳数。

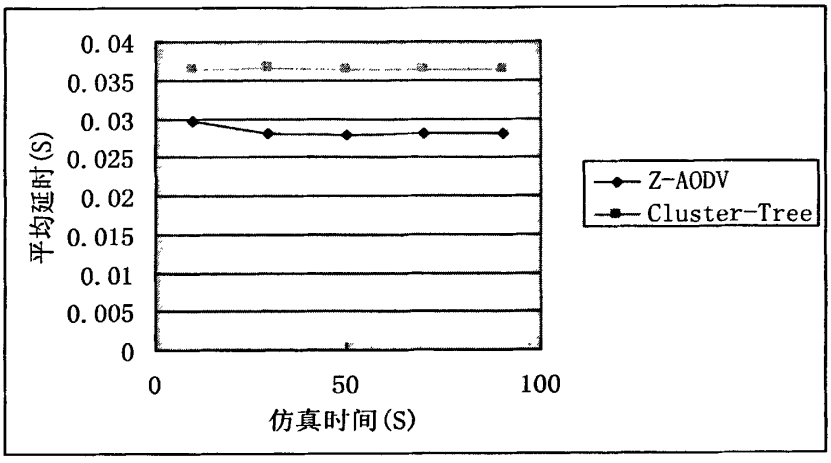

图 4.8 协调器在网络边缘的树形网络平均延时

从图 4.8 可以看出，与 Cluster-Tree 路由算法相比，Z-AODV 路由算法有较低的平均延时。这是因为 Cluster-Tree 路由算法多数情况下所经过的路径都不是最短路径，较多跳数导致更多的平均延时。而 Z-AODV 路由算法可以选择最小跳数的路径，大大降低了路径的平均延时。

4.2.3 协调器在网络中心的仿真

图 4.9 是一个 25 个节点的网络，图中连线表示父子关系。节点 0 为 ZigBee 网络的协调器，偶数节点都是 RN+，奇数节点是 RN-。仿真时间为 100 秒，采用 CBR 业务源，每个数据源发包速率为 10Packet/s，每个数据包 100 字节，节点的有效传输距离为 15m。

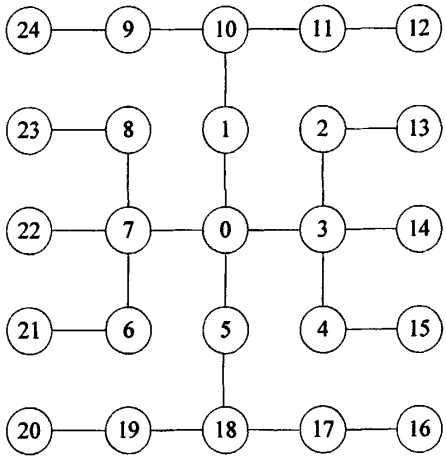

图 4.9　协调器在网络中心的树形网络

根据仿真数据得到图 4.10、图 4.11 和图 4.12。

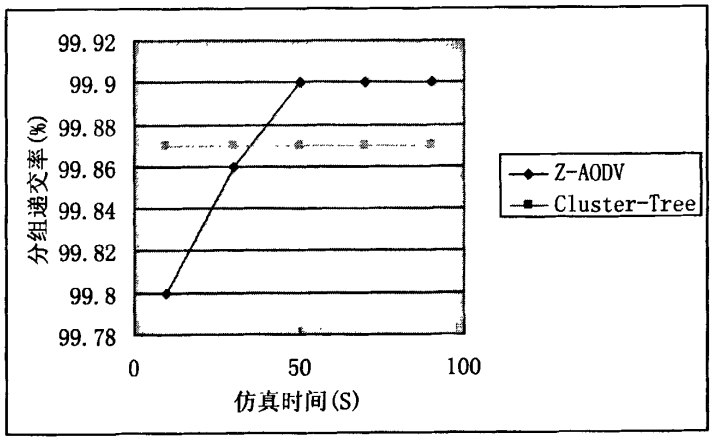

图 4.10　协调器在网络中心的树形网络的分组递交率

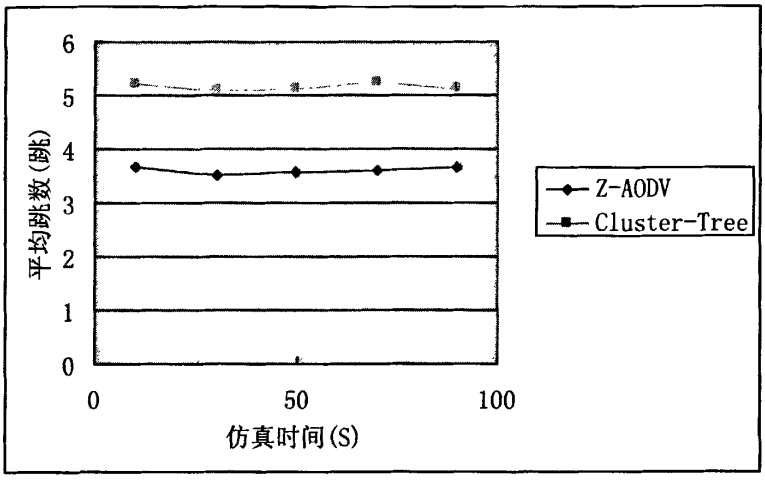

图 4.11 协调器在网络中心的树形网络的平均跳数

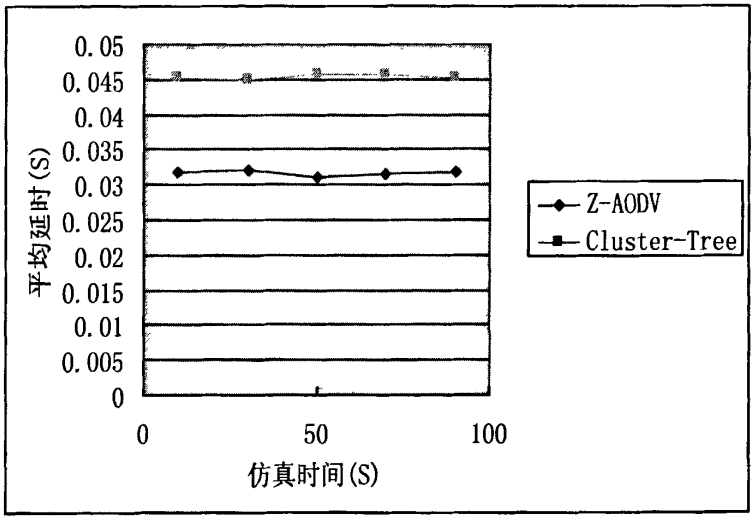

图 4.12 协调器在网络中心的树形网络的平均延时

从图 4.10、图 4.11 和图 4.12 可以看出，协调器在网络中心的情况下仿真的结果和协调器在网络边缘的情况基本一样，分组递交率稍降低，平均跳数和平均延时都相应增加，这是主要是由于网络节点增加、网络规模扩大引起的。

4.2.4 小结

从上述分析结果得出结论，在网络节点较少、结构简单的网络中，Z-AODV 路由算法和 Cluster-Tree 路由算法的性能差别不大。在网络节点多、结构复杂的网络中，Z-AODV 路由算法的性能明显比 Cluster-Tree 路由算法优越，能够有效降低路由跳数、减少数据传输延时。因为较大规模的网络中，Cluster-Tree 路由算法往往不能使用最短的路径传输数据，而 Z-AODV 路由算法能够选择最佳路径传输数据。

例如图 4.5 中节点 11 需要发送数据到节点 6。使用树路由的时候，节点 11 发送数据帧到其父节点 5，节点 5 判断目的节点不是自己的子节点，因此转发数据给父节点 2，2 节点判断目的节点是自己的子节点，于是把数据发送给节点 6。数据帧走过的路径是：11-5-2-6。

使用 Z-AODV 路由时，节点 11 首先广播路由请求命令帧发起路由发现过程，并且等待返回路由应答命令帧，然后根据路由应答命令帧获得的路由发送数据帧。在此这里，节点 11 的通信距离内包括节点 10 节点、节点 5 和节点 12，节点 6 的通信距离内包括节点 2、节点 5、节点 12、节 13 和节点 14。节点 11 发起路由发现过程，最短路径为两跳的路径有两个：11-5-6 和 11-12-6。节点 11 将选择最先返回应答的路径作为到节点 6 的路由。

第五章 能量均衡路由算法

ZigBee 网络往往拥有很多的节点，并且采用电池供电，虽然 ZigBee 规范中路由协议没有专门针对网络能量有效利用方面的机制，但 ZigBee 网络的特性和应用领域决定了节能问题在 ZigBee 协议栈各层设计中的重要性。从单个网络节点的角度来看，需要降低其能量消耗从而延长节点使用时间。ZigBee 技术采用了大量的措施来降低网络功耗，例如合理地降低发射功率，从而降低功耗和增加空间利用率；对空闲节点合理地切换到节能模式；通过信道监听和预测合理地避免数据帧碰撞。但是这些低功耗的措施主要集中在物理层和媒体访问控制层上，网络层路由算法中对能量控制的考虑比较少。虽然 ZigBee 网络层使用了 AODV 与 Cluster-Tree 算法相结合的路由协议，实现了两种算法的优势互补，较低的控制开销对节省电池能量消耗方面有一定的贡献，然而如果在 ZigBee 路由协议中加入专门的节能机制，将能显著的节省节点的能量消耗，延长 ZigBee 网络的寿命。另一方面，对于全网来说，单纯地追求降低各个节点能量消耗有可能造成某些节点被频繁地选做数据转发路径，过多的消耗能量造成这些网络节点失效，甚至导致网络寿命缩短[36]。

对于一个网络来说，某些节点能量耗尽而失效，其他节点在发送数据选择路由时，可以通过新的路由发现过程找到另外的路径来发送数据。但如果失效的是非常重要的节点，则网络的完整性就受到损害。而有些节点失效，会导致网络的分割，从而使整个网络失效。因此，如何使整个网络耗能均衡，保持网络的完整性，是个路由算法的重要研究方向。

第一节 几种节能路由算法介绍

目前，已有大量研究针对无线网络能量问题提出各种节能路由算法，主要有以下几类：

1．最小路径能量消耗路由策略。这种策略通过寻找总传送功率最小的路径来降低网络功耗，典型的算法有最小传输功率路由算法、PARO 算法和 LAPAR 算法。

2．基于节点能量感知的路由策略。这种策略把路径上节点剩余能量作为路径选择的依据，尽量避开低能量的节点，典型的算法包括 MBCR 和 MMBCR 算法[37]。

5.1.1 最小传输功率路由算法

最小传输功率路由算法 MTPR（Minimum Total Transmission Power Routing），是按需机制的路由技术增加了对于能量部分的考虑进行改进的。其在能量消耗方面的考虑，是在传送信息前，计算路径上每个点所需消耗能量的总和。最小传输功率路由算法在发起路由发现时发送RREQ帧，中间节点在收到RREQ时，计算自己传送此帧所需消耗的能量并累加写入 RREQ 后广播出去。目的节点接收到各个路径转发过来的 RREQ 后，在这些路径中选择一条累计所耗能量最少的路径回复 RREP，完成路由发现过程。

也就是说，最小传输功率路由算法总是选择网络消耗能量最少的路径[38]，也即选择路径电池开销最小的一条路径来传输数据分组，保持网络剩余总能量最高，以延长网络的使用时间。但是，这个路由算法有可能使网络中某些节点过多地参与数据转发，造成这些节点能量消耗过快，从而导致网络分割，反而减少了网络的使用时间。

5.1.2 最小电池开销路由算法

最小电池开销路由算法（Minimum Battery Cost Routing，MBCR）也是建立在按需机制的基础上，与最小传输功率路由算法不同的是，MBCR 算法把路径上所有节点的剩余电池能量总和作为路由选择的度量指标，选择的路径是源节点到目的节点总的剩余电池能量最多的路径。

节点电池剩余能量越大，其被选择作为转发数据路径的可能性越高，相反，剩余能量越小的节点，其被选择作为转发数据路径的可能性越小。如果所有节点的电池剩余能量一样多，则该算法相当于选择一个最小跳数路由。

MBCR 选择总的电池剩余能量最多的路由[39]，防止了某些节点的过度使用，在一定程度上延长了网络寿命，剩余能量充足的节点更多地参与数据分组的转发，使剩余能量匮乏的节点维持更多的工作时间，从而推迟了网络分割的时间。但是，此度量只考虑了电池开销函数值的总和，也有可能选择包含剩余电池能

量小的节点的路径。这样就会更快地耗尽原本剩余电池能量已不足的节点，从而无法达到均衡全网电池能量消耗、延长网络寿命的目的。

5.1.3　最小最多电池耗费路由

最小最多电池耗费路由（Min-Max Battery Cost Routing，MMBCR）是对MBCR 路由算法的改进设计，也是以剩余能量作为主要参考，但是 RREQ 报文中除了记载广播路径外，还记载了此路径中剩余能量最少的节点的剩余能量，因此目的节点可以对各条路径中剩余能量最少节点的能量进行比较，选择剩余能量较多的一条路径作为传输路由[40]。这样就可以避免某些节点被过度使用而造成网络被分割的情况出现，延长了网络的使用寿命。但是 MMBCR 不能保证选择是总传输功率最小的路径。

第二节　能量均衡路由算法的改进

本节在分析总结 MTPR、MBCR 和 MMBCR 路由算法后，提出一种能量均衡路由算法的改进方案，并对 MPTR、MMBCR 和改进方案做 NS2 仿真实验，进行性能比较。

5.2.1　能量均衡路由算法的改进

ZigBee 网络中如果选择 MMBCR 路由算法，则可以尽量避开电池剩余能量最少的一些节点，以期达到整个网络电池能量的均衡消耗。但是，这种算法在路由选择的时候可能会为了避开剩余能量低的节点而选择了较多跳数、较多能耗的路径。在 ZigBee 网络中，数据传输的所需能耗和路径选择紧密相关，数据转发的跳数增多、路径变长，整个网络总体消耗的能量就会增多，随路径跳数增加而带来的控制开销和时延也在增加。从这点看来，MMBCR 路由算法不是在任何情况下都能延长网络的使用寿命，反而可能会因为增加了网络总的能耗，而导致网络节点的总体寿命缩短。

如果在选择路由的时候，仅仅考虑数据转发路径的能耗，如 MTPR 路由算法，虽然能够使每次数据传送的消耗的能量最少，从而保持网络总的剩余能量最多。这样的路由选择策略有可能会造成某些节点频繁地被选作数据转发路径，

这些节点因为过多参与转发数据电池能量很快被耗尽，而造成这些节点的过早失效，导致网络性能下降，甚至整个网络失效。

因此，本文把这两种算法的优点结合起来，提出一种综合考虑路径的能量消耗和节点的剩余能量的能量均衡路由算法（CEER 算法）。其主要思想是：在保持各节点剩余能量均衡的基础上选择耗能最小的路径发送数据。

ZigBee 应用层的节点电源描述符给出了节点电源的动态状况。其中当前电源容量等级域指出了当前设备电源的剩余电量，如表 5.1 所示。

表 5.1　电源容量等级表

当前电源容量等级域	电量等级
0000	紧急
0100	33%
1000	66%
1100	100%
其他	保留

把电源的剩余电量划分为三个区域：

1．充足区：节点设备当前剩余能量在 100%到 66%之间。

2．中间区：节点设备当前剩余能量在 66%到 33%之间。

3．匮乏区：节点设备当前剩余能量在 33%以下。

把节电电池的剩余能量和路径总能耗作为主要参考参数，RN+节点在路由发现过程中，RREQ 报文中除了记载广播路径和路径累积能耗外，还把此路径中各节点的当前电源容量等级写入报文中，目的节点收到各条路径的 RREQ 后，对各条路径中各节点的剩余能量进行比较。如果路径中有节点处于匮乏区，则在路由选择的时候避免选择此路径。如果各条路径中各节点的剩余能量处于同一个区域，则按照 MTPR 算法选择路径电池开销最小的一条路径。如果某些路径中有节点处于较低能量区域，而某些路径中的节点处于较高能量区域，则尽量选择节点处于较高能量区域的路径。否则，尽量选择处于剩余能量多的路径。这样，进行路由选择的时候，能够尽量均衡的消耗节点能量，避免选择包含低电的节点，同时又尽可能地选择最小耗能的路径。

假设源节点到目的节点有 n 条路径，第 i 条路径上有 m 个节点，E_{ij} 表示第 i 条路径上的第 j 个节点的剩余能量，E_1 表示节点处于匮乏区，E_2 表示节点处于中间区，E_3 表示节点处于充足区。第 i 条路径上节点的最小剩余能量：

$$E^{i}_{\min} = \min\left\{E_{ij}, (1 \le j \le m)\right\} \tag{5.1}$$

C_{ij} 表示第 i 条路径上的第 j 个节点发送数据的能耗，路径的总能耗：

$$COST_i = \sum_{j=1}^{m} C_{ij} \tag{5.2}$$

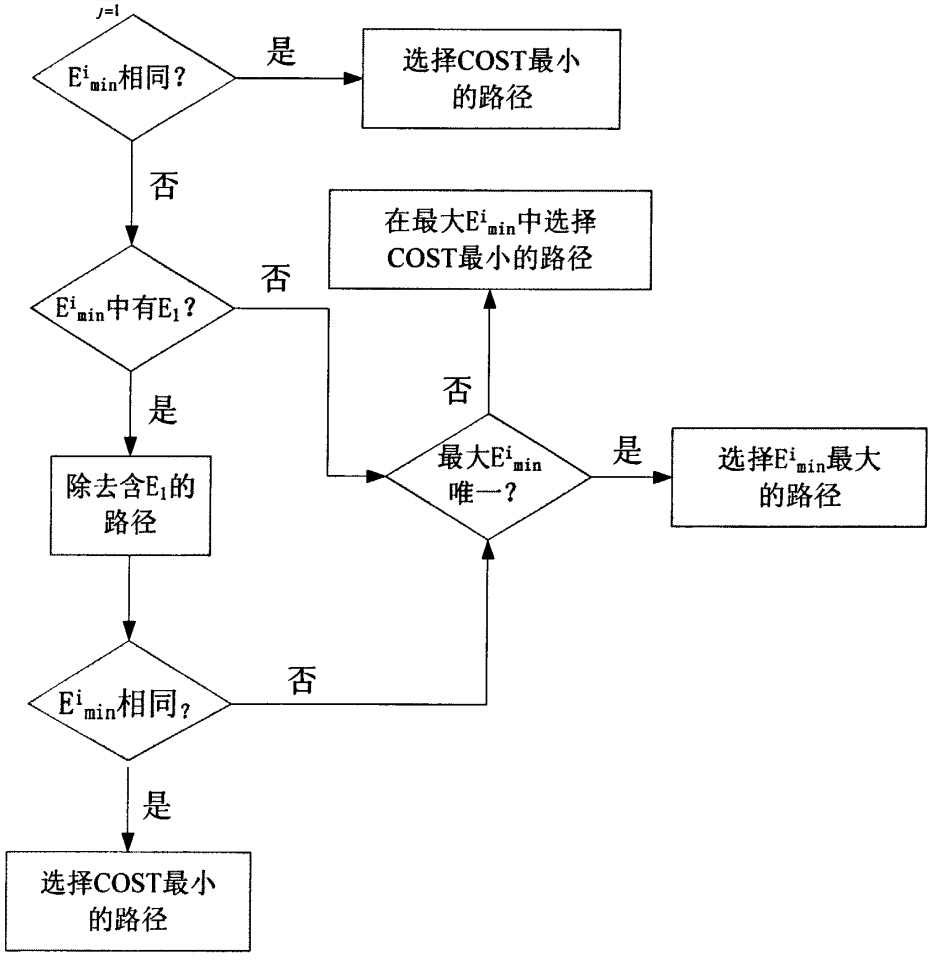

图 5.1　能量均衡算法改进方案流程

以图 5.2 所示的网络为例，说明该算法的路由选择过程。节点 0 是源节点，向目的节点 9 发送数据，从 0 节点到 9 节点有三条路径。节点 0 发出路由请求命令帧（RREQ）发起路由发现过程，节点 9 收到三个 RREQ 帧，这三个 RREQ 中分别携带了这三条路径的总的能耗 COST 和路径上各个节点剩余能量级的信息，节点 9 将根据这些信息选择一条路径返回路由应答（RREP）给源节点 0，假设 $COST_1 < COST_2 < COST_3 < COST_4$。

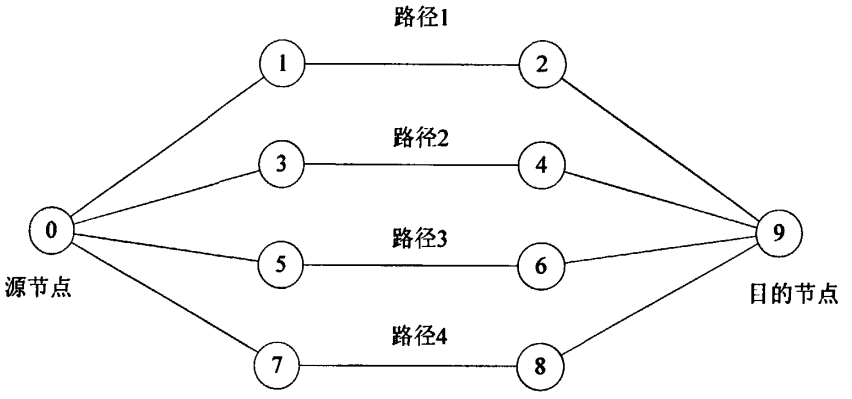

图 5.2　示例网络

表 5.2 为假设各种条件下各个节点的剩余能量级别及算法的路由选择。

表 5.2　不同条件的路由选择表

条件	节点 1 能量级	节点 2 能量级	节点 3 能量级	节点 4 能量级	节点 5 能量级	节点 6 能量级	节点 7 能量级	节点 8 能量级	路由选择
条件一	E_1	E_2	E_1	E_2	E_1	E_2	E_1	E_2	路径 1
条件二	E_3	E_3	E_3	E_3	E_3	E_2	E_2	E_3	路径 1
条件三	E_3	E_2	E_3	E_3	E_3	E_2	E_2	E_3	路径 2
条件四	E_1	E_2	E_2	E_3	E_2	E_3	E_2	E_3	路径 2
条件五	E_1	E_2	E_2	E_3	E_3	E_3	E_3	E_3	路径 3
条件六	E_1	E_2	E_2	E_3	E_2	E_3	E_3	E_3	路径 4

各种条件下该算法的路由选择过程：

条件一：四条路径上的 E_{min}，都是 E1，根据算法，选择 COST 最小的路径 1；

条件二：路径上的 E_{min} 不同，但是没有 E1，最大的 E_{min} 是 E3 不唯一，于是在路径 1 和路径 2 中选择 COST 最小的路径 1；

条件三：路径上的 E_{min} 不同，没有 E1，最大的 E_{min} 是 E3，于是选择 E_{min} 最大的路径 2；

条件四：路径上的 E_{min} 不同，最小的是 E1，除去路径 1 后，路径 2、3、4 上的 E_{min} 相同，都是 E2，于是选择 COST 最小的路径 2；

条件五：路径上的 E_{min} 不同，最小的是 E1，除去路径 1 后，路径 2、3、4 上的 E_{min} 不同，在 E_{min} 为 E3 的路径 3、4 中选择 COST 最小的路径 3；

条件六：路径上的 E_{min} 不同，最小的是 E1，除去路径 1 后，路径 2、3、4 上的 E_{min} 不同，选择在 E_{min} 为 E3 的路径 4。

5.2.2　仿真实验与结果分析

在此次仿真中，需要经过多次实验来确定合适的节点初始能量，以体现不同路由算法之间的性能差异。如果节点初始能量太高，则在仿真时间内网络中不会出现失效节点，不能检验能量平衡路由算法的优势。如果初始能量太低，则网络节点因能量很快耗尽而失效，一些性能指标会失去统计意义。仿真实验的网络拓扑结构如图 5.3 所示。

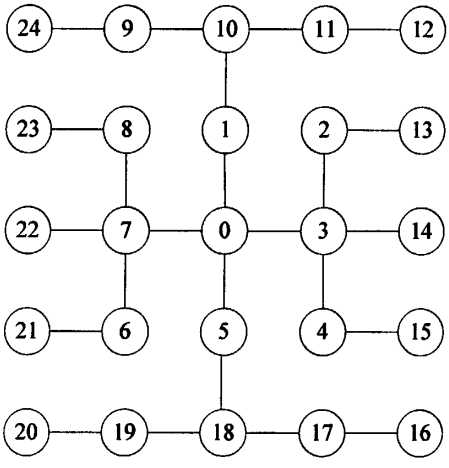

图 5.3　仿真网络场景

仿真时间为 1000 秒，采用 CBR 业务源，每个数据源发包速率为 10Packet/s，每个数据包 100 字节。节点初始能量为 100J，节点发送一个数据包耗能 0.4J，接收一个数据包耗能 0.1J。Cluster-Tree 的参数为 Rm=4，Cm=4，Lm=4。RN+节点有 15 个，RN-节点有 6 个。仿真中，称综合考虑路径能耗和节点剩余能量的路由算法为条件能量均衡路由算法，标记为 CEER，根据仿真数据得到图 5.4、图 5.5 和图 5.6，分析这三种路由算法的性能。

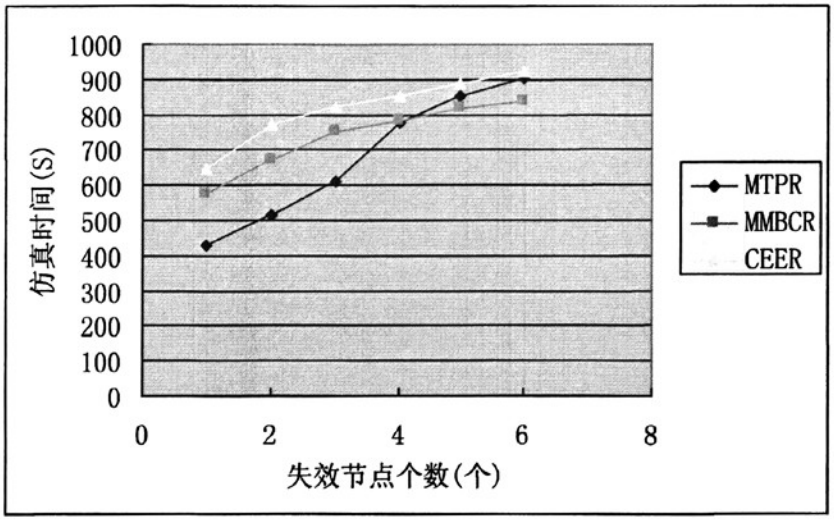

图 5.4　失效节点和仿真时间的关系

从图 5.4 可以看出来，MTPR 算法最先于 430 秒时出现第一个能量耗尽的失效节点，MMCBR 算法在 577 秒时出现第一个能量耗尽的失效节点，而 CEER 算法在 645 秒时出现第一个能量耗尽的失效节点。而在 780 秒后，MMCBR 算法中失效的节点个数多于 MTPR 算法中失效个数。这是因为，MMCBR 路由算法在路由选择的时候避开了网络中剩余能量最少的节点，而选择了能耗较高的路径，虽然推迟了第一个失效节点出现的时间，但是缩短了更多节点的存活时间。而 CEER 算法在延长了网络最大存活时间的同时，能够有效减少网络的总能耗，延长网络节点的存活时间。

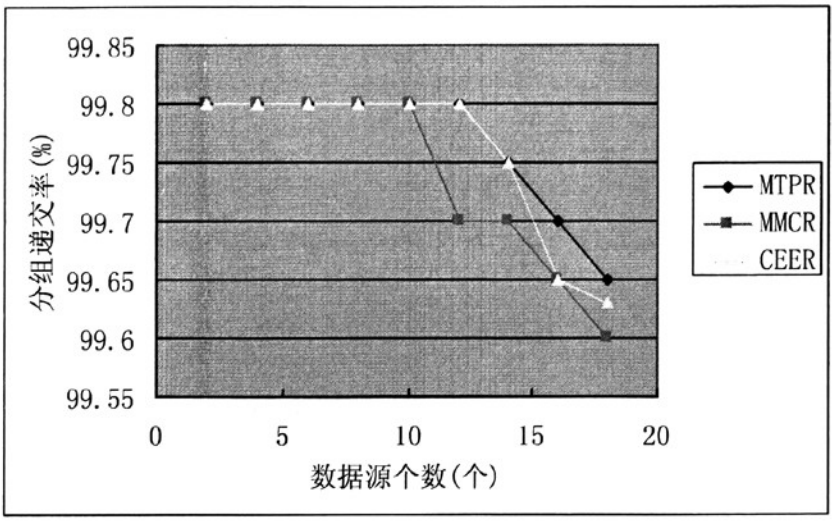

图 5.5　三种路由算法的分组递交率

从图 5.5 可以看出，三种路由算法都有较高的分组递交率，保持在 99.6%以上。当数据源个数增加到 10 个以上时，分组递交率稍稍有所下降，但差异不大。这是因为，随着数据源个数的增多，数据分组产生碰撞的可能性增大，导致丢包现象增多，分组递交率下降。

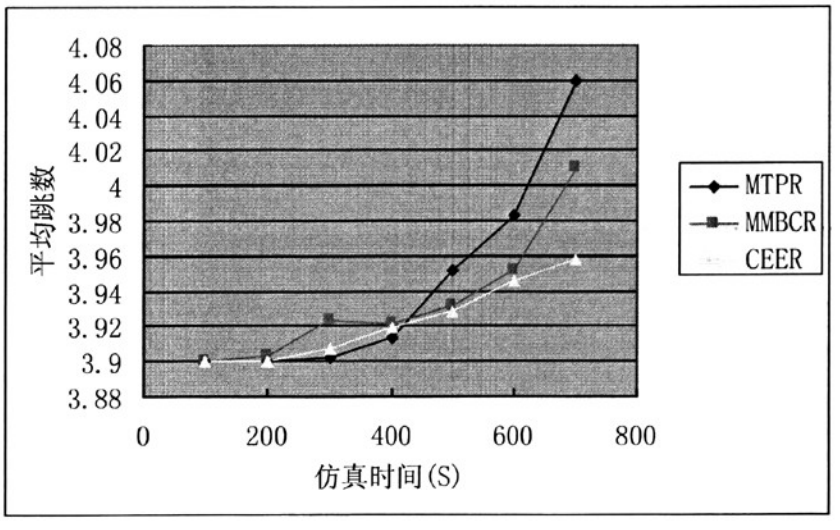

图 5.6　平均跳数

从图 5.6 可以看出，在仿真前 200 秒内，三种路由算法性能几乎一致。而在 200 秒到 400 秒之间，MTPR 算法的平均跳数较低，之后迅速增加，而 CEER 算法的平均跳数缓慢增加，但都低于其他两种路由算法。这是因为在仿真初期，节点的能量比较充足，三种路由算法选择的路径差别不大。在 200 秒到 400 之间，MTPR 算法选择功耗最小的路径，在这里也即是最少跳数路径，因此它的跳数最少，而在此之后，由于网络中出现失效节点而导致平均跳数迅速增加。而 MMBCR 算法避开剩余能量较少的节点而选择了较多跳数的路径，所以平均跳数会稍高，但因为它的失效节点出现晚，因而后面的平均跳数小于 MTPR 算法。而 CEER 算法在前期选择能耗最少的路径，后面选择剩余能量最多的路径，因而失效节点出现得最晚，平均跳数在前期处于中间水平，而后期为最低。

从以上分析可以看出，MTPR 算法在路由选择的时候仅考虑路由能耗最少，虽然在前期可以节省网络能耗，但会使一些节点过早地耗尽能量而导致节点失效。而 MMBCR 算法仅考虑避开剩余能量最少的节点，虽然推迟了失效节点的出现时间，但它增加了平均跳数和能耗，使后期网络的失效节点迅速增加，没有达到延长网络使用寿命的效果。而 CEER 算法能够尽量平衡节点能量消耗，并同时尽量选择能耗最小的路径，避开能量匮乏的节点，从仿真的结果来看，这样能够有效推迟失效节点的出现，降低路由平均跳数，延长了网络的使用寿命。

第六章 结论与展望

第一节 结论

本文简要介绍了 Zigbee 技术的产生和发展历史、国内外的研究进展和应用情况，详细介绍了 zigbee 规范的架构体系。

本文基于 NS2 仿真平台建立 ZigBee 路由模型，对 Cluster-Tree、Z-AODV、MTPR、MMBCR 路由算法进行仿真实验。仿真结果的分析表明：

1．树路由具有简单、响应速度快的特点，适合于网络结构简单、突发性的数据传输业务。

2．而 Z-AODV 路由在长期稳定的数据传输中则可以获得较少的转发次数。

3．单纯地考虑最小路由成本的路由方式，容易造成某些网络节点因耗能过多而过早死亡，导致网络分割，甚至整个网络失效。

在对 MTPR、MMBCR 能量路由算法仿真结果分析后，提出了一种能量均衡路由算法的改进方案，并进行了仿真实验和性能分析。改进方案进行路由选择的时候，尽量使节点能量均衡地消耗，避免选择包含低电的节点，同时又尽可能地选择最小耗能的路径。实验结果表明，这种路由算法能够有效地降低网络的总体功耗，推迟网络出现失效节点的时间，延长网络的工作时间。

综上所述，本文在 ZigBee 网络路由协议研究中取得的成果对提高 Zigbee 网络性能是有益的。

第二节 展望

ZigBee 是近几年提出的新技术，其相关机制尚不成熟，需进行更多研究和改进。作者认为，在以下几个方面值得做进一步研究：

1．本文提出的改进方案中，能量等级的划分是采用了应用节点电源描述符的级别划分，如何划分节点剩余能量等级更有效，可以做进一步的研究。

2．如何适应网络拓扑结构的变化，使算法的适用范围更加广泛。

参考文献

[1] 吕治安. ZigBee 网络原理与应用开发. 北京:北京航空航天大学出版社, 2008.10～12

[2] IEEE Std. 802.15.4 IEEE Standard for Information Technology. 2003

[3] 刘丽钧. ZigBee 技术网络层的路由算法分析. 计算机与信息技术,2008,Vol.1:70～72

[4] 颜艳华. 基于 ZIgBee 技术的无线网络芯片 SN250 及其应用. 中国集成电路,2008,Vol.17:73～76

[5] 杜焕军. ZigBee 网络的路由协议研究. 合肥工业大学学报,2008,Vol.31(10):1617～1621

[6] 伍琛尧,刘建煌. 基于 ZigBee 的工业设备监控系统. 计算机与数字工程,2007,Vol.35(9):155～158

[7] 熊邦毛. 基于 433M 路由算法在抄表系统中的应用与研究. 计算机工程,2008,34(8):281～282

[8] 朱开宇. 基于 ZigBee 的城市智能公交网络系统. 单片机与嵌入式系统应用,2008 (3):17～20

[9] Pietro Valdastri,Stefano Rossi.An implantable ZigBee ready telemetric platform for in vivo monitoring of physiological parameters.Sensors and Actuators, 2008,Vol.142 :369～378

[10] Ed Callaway, P. Gorday.Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks.IEEE Communication Magazine,2002,Vol.40(8): 70～77

[11] 王锐华, 益晓新. ZigBee 与 Bluetooth 的比较及共存分析. 测控技术, 2005,24(6):50～56

[12] 陈聪, 冯玉林, 施惠昌. ZigBee 与 Wi-Fi 的干扰与共存. 计算机工程与设计, 2006,27(18):34～41

[13] Jung N. Lee, Jong K.Park. A Novel Dual-Band Patch Antenna for ZigBee System. Microwave and Optical Technology, Vol.49(8):1864～1867

[14] 朱向庆, 王建明. ZigBee 网络路由算法测试方案. 电子测量技术, 2006, 29(5):142～145

[15] Barontip, Pllaip.Wireless sensor networks:A survey on the state of the art and the 802.15.4 and ZigBee standards. Computer Communications,2007,Vol.30(7):1655～1695.

[16] Wanzhi Qiu,Efstratios Skafidas.Enhanced tree routing for wireless sensor networks.Ad Hoc Networks,2009,Vol.7(3): 638～650

[17] 应瑛, 章坚武. ZigBee 网状网络路由协议的可扩展性研究. 杭州电子科技大学学报, 2008,Vol.28(4):41～44

[18] Akyilidiz I.F.,Wang X.A. Survey on Wireless Mesh Networks.IEEE Communications Magazine,2005, 43(9):23～30

[19] Kaushik R. Chowdhury,Nagesh Nandiraju.Channel allocation and medium access control for wireless sensor networks. Ad Hoc Networks,2009,Vol.7(2):307～321

[20] E.Susilo,P.Valdastri. A Miniaturized Wireless Control Platform for Robotic Capsular Endoscopy Using Advanced Pseudokernel Approch. Sensors and Actuators,2009,

Vol.151:26～32

[21] Edgar H, Callawy J. Wireless sensor networks: architectures and protocols.New York: Auerbach publication, 2003.260～300.

[22] 刘市生,张贤华. ZigBee 网络层的设计与实现.无线电工程,2008,Vol.38(11):7～9

[23] 蒋挺, 赵成林. 紫蜂技术及其应用. 北京:北京邮电大学出版社,2006.50～60

[24] Ran Peng,Sun Mao-heng.ZigBee Routing Selection Strategy Based on Data Services and Energy-balanced ZigBee Routing. Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing.Washington D.C:IEEE Computer Society, 2006.400～406

[25] 郭壮辉.降低路由开销的 ZigBee 路由算法研究. 电脑知识与技术,2008(6):1043～1048

[26] C.E. Perkins, E.M. Royer, Ad hoc on-demand distance vector routing.IEEE Workshop Mobile Computing Systems and Applications, 1999:90～100

[27] 宗怡. ZigBee 网络路由算法的研究及实现. 黑龙江信息科学, 2007(23):83～84

[28] Li V, Park HS.A Cluster-Label-Based Mechanism for Backbone on Mobile Ad hoc Network. Switzerland: The 4th Wired/Wireless Internet Communications, 2006:26～36

[29] 陈稼婴, 杨震. Adhoc 网络中基于节能的 AODV 路由算法改进. 南京邮电学院学报, 2004,Vol.24(3):21～25

[30] 王芳.一种改进的 ZigBee mesh 网络路由算法.计算机应用,2008,Vol.28(11):2788～2790

[31] 顾瑞红, 张宏科. 低速无线个域网中的 IPV6 路由实现. 北京交通大学学报, 2005,29(5):36～39.

[32] 黄双华.ZigBee 无线传感器网络路由研究与实现.电子测量技术,2007,30(2):59～64

[33] 刘军,王桂棠.ZigBee 技术中的 mesh 网络研究与实现.信息技术,2008(1):20～22

[34] 任秀丽,于海斌.ZigBee 无线通信协议实现技术的研究.计算机工程与应用,2007,43(6):143～145

[35] 徐雷鸣, 庞博, 赵耀. NS 与网络模拟. 北京:人名邮电出版社, 2003.1～27

[36] 班艳丽,柴乔林.基于能量均衡的 ZigBee 网络树路由算法.计算机应用,2008,Vol.28(11):2791～2794

[37] 冯嵩. 网络路由协议中的能量均衡机制. 电脑知识与技术,2008(3):15～19

[38] Scott K, Bambos N.Routing and Channel Assignment for Low Power Transmission in PCS.ICUPC'96,1996.498～502

[39] SNGH s, WOO M. Power Aware Routing in Mobile Ad Hoc Networks. Proceedings of Mobicom'98 Conference ,1998.181～189

[40]谭长庚,张芝华.MANET 能量与其他网络性能平衡路由协议.计算机应用,2007,Vol.27(5):1073～1076

致 谢

本文是在导师韩毅刚副教授的悉心指导下完成的。韩老师虽然有繁重的教学和科研工作，但是仍然抽出宝贵的时间给予我在学习上的指导和帮助，从课题选择到研究直至论文的写作，整个过程无不渗透着韩老师的心血。韩老师的真知灼见、敬业精神和严谨治学给我留下了极为深刻的印象，并使我在研究生的学习中受益良多。韩老师不但在学习和课题研究上给予指导，提出了很多宝贵的意见，在生活方面也给我以温暖的关怀和帮助。在此，我衷心地表示感谢。

真诚地感谢我的任课教师吴岳教授、吴虹教授、吴功宜教授、郭君益教授、李庆诚教授，他们每个人的讲课风格都给我留下了很深的印象，现在想起仍历历在目，他们传授的知识使我终生受益。

感谢同实验室王宝坤、冯建业、李学文、夏爱民、孟凡亮、唐海波、宁建军、魏震和蔡航俊，他们在学习上给予我很多帮助。

在课题的完成过程中还得到了其他老师和同学的帮助和指导，在此表示诚挚的谢意。

感谢南开大学，在这个全国闻名的大学里，我听过很多优秀教师的精彩讲课和讲座，参加了丰富的课外活动，结识了一些好朋友，体验了南开美好的校园生活。

感谢我的家人，感谢他们对我在学习和生活中的支持，他们是我学习和生活的动力，也是我最坚强的后盾！

最后，再次感谢所有给予我关心和帮助的人们！

个人简历

陈波，1977年11月16日出生，江苏淮安人。2000年7月，毕业于解放军信息工程大学，获得工学学士学位。

ZigBee路由算法分析及改进

作者：陈波
学位授予单位：南开大学