

摘 要

无线 mesh 网络是一种多跳网络, 因其覆盖范围大, 部署方便及健壮性等一系列优点在近年受到了极大的关注。无线个域网也是当今的研究热点, IEEE 802.15.4 标准定义了低速的小范围无线网络, 支持星型和对等结构, 采用 CSMA/CA 机制竞争信道。而 IEEE 802.15.5 草案对 15.4 进行了网络层的扩充, 使其支持 mesh 结构, 本论文将对低速无线 mesh 网络的关键技术进行研究, 重点关注节点的吞吐率和流量公平性, 并基于硬件实验平台实现了软件系统。

吞吐率和公平性是无线网络的重要性能指标。目前对吞吐率的研究主要集中在单跳网络, 也有一些文献对路径吞吐率和端到端吞吐率进行了研究, 但是它们一般都是基于 IEEE802.11 协议簇, 研究结果不能直接应用于采用 IEEE802.15.4 标准的低速网络。由于无线 mesh 网络是多跳网络, 网络中每个节点所处的位置不一样, 干扰和竞争状况也不一样, 因此对网络中任意节点的吞吐率进行研究具有实际意义。本文将对“接收优先”情况下的低速无线网络的节点饱和吞吐率进行建模并用 OPNET 进行仿真, 仿真结果表明该模型较好地预测了节点的吞吐率, 为评估网络性能提供了一种工具。

网络的公平性包括节点间的公平性和节点内的公平性, 对节点间的公平性主要采用公平选路方法解决, 而对节点内不同数据流间的公平性可以通过队列调度的方式处理。由于采用 CSMA/CA 算法随机接入网络的节点获得信道使用权的能力与其退避窗口的大小成反比, 因此可以通过调整不同节点的初始退避窗口使负荷大的节点获得更多的使用信道的机会, 相应地使网络边缘负荷小的节点获得更少的使用信道的机会, 从而提高节点的流量公平性。第一次发送时根据饱和情况下 IEEE802.15.4 标准的退避过程的 Markov 模型为不同负荷的节点设置不同的初始退避窗口, 以后每次发送时都对节点及其竞争节点实际获得的吞吐率进行估计, 然后根据这些估计值计算一个公平指数, 最后按照公平指数所在的范围采用不同的方法调整节点的初始退避窗口。采用这样的方案节点的流量公平性得到了很大的提高, 网络的丢包率也大幅下降且非饱和情况下节点的吞吐率也有一定的提高。

为了对不同的算法性能进行评估, 有必要建立实际的实验平台。基于由 TI 公司的 MSP430F149 微处理器和 Chipcon 公司的 CC2420 射频芯片搭建的低速无线 mesh 网络硬件平台, 本文介绍了符合 IEEE 802.15.4 标准和 IEEE 802.15.5 草案的 mesh 网络软件系统实现方法。该软件系统采用分层结构, 可扩展性较好, 可以方便地进行升级, 为实际评估算法性能提供了有效的平台。

【关键词】

无线 mesh 网络、IEEE 802.15.4 标准、IEEE 802.15.5 草案、接收优先、吞吐率、流量公平性、软件实现

ABSTRACT

Wireless mesh networks is a kind of multi-hop networks, which has attracted much concentration in recent years because of its large coverage, easy deployment and haleness. Wireless personal area network is also one of the hotspots of research now, while the IEEE 802.15.4 standard specifies low rate small area wireless networks. It supports star-topology and peer-to-peer topology. The IEEE 802.15.5 draft specifies the network layer for IEEE 802.15.4 standard to realize a mesh network. This thesis will make some research on some key technologies of low rate wireless mesh networks, which mainly focused on node throughput and flow fairness, and finally realized the software system based on our hardware testbed.

Throughput and fairness are two of the most important performance indexes. Current researchs on throughput are mainly focused on single-hop networks, and some others made research on path throughput or end-to-end throughput. But almost all of them are based on IEEE802.11s, the result of which can't be applied to low-rate networks adopting IEEE802.15.4 standard. As wireless mesh networks are multi-hop networks and each node in the network has different location, different interference and competing neighbors, it makes sense to analyse arbitrary node's throughput in network. This thesis will model node's throughput in "reception preference" situation of low-rate networks and make simulation using OPNET. The results show that our analysis well predicted node's throughput, which provides a good tool to evaluate network's performance.

Fairness of networks includes inter-node's and intra-node's fairness. And inter-node's fairness is solved by fair path selecting metrics generally, while intra-node's fairness is solved by queue dispatching approaches. As nodes adopting CSMA/CA algorithm has an ability which's inverse to its backoff window to use the channel, we can adjust different nodes' initial backoff windows to make nodes with heavier loads have more chance to use the channel and nodes with less loads have less chance to transmit, therefore improve their flow fairness. For first transmission, different initial backoff windows are set according to nodes' loads based on backoff procedure's Markov model in saturated situation of IEEE802.15.4 standard. From then on during each transmission node should first estimates its achieved throughput and its competing nodes' sum throughput, then calculates a fairness index based on them, and adopts different methods to adjust its initial backoff window according to the range of the fairness index at last. In this way, nodes' flow fairness get improved greatly, packet loss rate highly declined and throuput of node improved in unsaturated condition.

To evaluate different algorithms' performance, it's necessary to set up real testbed. Based on our hardware platform constituting of TI's MSP430F149 and Chipcon's CC2420, this thesis will introduce the method to realize software system which complies with IEEE802.15.4 standard and IEEE802.15.5 draft. The software system's constructed in layers and can be extended and upgraded easily, which provides an effective platform to evaluate different algorithms.

【Keywords】

Wireless mesh networks, IEEE 802.15.4 standard, IEEE 802.15.5 draft, reception preference, throughput, flow fairness, software realization

缩 略 词

WPAN, Wireless Personal Area Network, 无线个域网

AP, Access Point, 接入点

MAC, Media Access Control, 媒体控制接入

RTS, Request to Send, 发送请求

CTS, Clear to Send, 清除请求

FFD, Full function device, 全功能设备

RFD, reduced-function device, 精简功能设备

PAN, Personal area networks, 个域网

DCF, Distributed coordination function, 分布式协调功能

CSMA/CA, Carrier Sense Multiple Access with Collision Avoidance, 载波侦听多点接入/避免冲撞

GTS, Guaranteed slots, 有保证时隙

LQI, link quality indication, 检测接收数据包的链路质量指示

CCA, clear channel assessment, 空闲信道评估

MHR, MAC header, 帧头

MFR, MAC footer, 帧尾

FCS, Frame checking sequence, 帧校验序列

MSDU, MAC sub-layer data unit, MAC 子层数据单元

PSDU, PHY sub-layer data unit, 物理子层数据单元

SHR, Synchronization header, 同步头部

PHR, PHY header, 物理层头部

MLME, MAC layer management entity, MAC 层管理实体

TTL, Time to live, 生存时间

SPOF, Single point of failure, 单点故障

ACK, acknowlegment, 应答

列图索引

图 2-1 星型网络和点对点网络	8
图 2-2 MAC 层基本帧格式	10
图 2-3 帧控制字段结构	10
图 2-4 信标帧格式	11
图 2-5 数据帧的格式	12
图 2-6 确认帧的格式	12
图 2-7 命令帧的格式	13
图 2-8 非时隙 CSMA/CA 算法流程	14
图 2-9 mesh 层基本帧格式	17
图 2-10 mesh 帧控制字段结构	17
图 2-11 mesh 层数据帧格式	18
图 2-12 路由控制域结构	18
图 2-13 mesh 层命令帧格式	18
图 2-14 mesh 层 hello 帧结构	19
图 2-15 地址分配示意图	19
图 2-16 邻节点列表	20
图 2-17 连通性矩阵	21
图 2-18 寻找下一跳转发节点算法流程	22
图 2-19 参考 mesh 网络结构	23
图 3-1 节点发送模型	26
图 3-2 802.15.4 CSMA/CA 退避过程 Markov 模型	27
图 3-3 节点吞吐量: 理论分析和仿真结果	33
图 4-1 mesh 网络中节点流量示意图	36
图 4-2 改进算法和标准算法的系统吞吐量	42
图 4-3 改进算法和标准算法的系统丢包率	42
图 4-4 改进算法和标准算法的公平性指数 1	44
图 4-5 改进算法和标准算法的公平性指数 2	44
图 5-1 普通节点实物图	47
图 5-2 网关节点实物图	48
图 5-3 软件系统结构框图	48
图 5-4 存储单元结构	49
图 5-5 内存池结构图	50
图 5-6 层间原语调用示意图	52
图 5-7 网关设备状态转移图	53
图 5-8 普通设备状态转移图	54
图 5-9 CSMA/CA 算法流程图	55
图 5-10 路由算法流程图	56
图 5-11 网关设备工作流程图	57
图 5-12 普通设备工作流程图	58
图 5-13 接收中断处理流程图	59

列表索引

表格 2-1 载波信道特性一览表	9
表格 3-1 节点吞吐量分析模型仿真参数	32
表格 4-1 采用标准算法节点获得的吞吐量	43
表格 4-2 采用改进算法节点获得的吞吐量	43

东南大学学位论文独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名： 刘修齐 日期： 2008.1.20

东南大学学位论文使用授权声明

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学研究生院办理。

研究生签名： 刘修齐 导师签名： 徐子平 日期： 2008.1.20

第1章 绪论

1.1 引言

无线通信网络按覆盖范围可分为无线广域网,无线城域网,无线局域网和无线个域网,其中无线局域网最初定义为有线局域网的无线延伸,现在则作为高速无线互联网接入技术,应用于机场、工厂、大厦、图书馆、学校等人口密集的热点地方,至少能够支持大于 100 米的距离。目前无线局域网的相关标准为 IEEE 802.11^[1]系列。而无线个域网则是一种用于连接家庭各种家电、娱乐设备的无线网络,作用距离在 10 米范围内。WPAN^[2]分低速和高速两种,IEEE 802.15.4^[3]和蓝牙(Bluetooth)是低速 WPAN 标准,而 IEEE 802.15.3^[4]则是高速 WPAN 标准。

传统的无线网络只支持单跳方式接入网络,即客户端只能和接入点(AP)进行直接通信,即使多个客户端相邻。而无线 mesh 网络作为一种多跳网络解决了传统网络覆盖范围有限的缺点,并具有一系列其它优点。

本章将介绍无线 mesh 网络的优点,关键技术和本文所重点研究的无线 mesh 网络性能,包括节点吞吐率和公平性等相关研究背景,并提出本文的研究安排。

1.2 无线 mesh 网络简介

无线 mesh 网络也称为“多跳”网络^[5],它是一种与传统无线网络完全不同的新型无线网络技术,而 mesh 这个词的原本意思就是所有的节点互相连接。

在传统的无线局域网中,每个客户端均通过一条与 AP 相连的无线链路来访问网络,用户如果要进行相互通信的话,必须首先访问一个固定的接入点(AP),这种网络结构被称为单跳网络。而在无线 mesh 网络中,任何无线设备节点都具有路由功能^[6],网络中的每个节点都可以发送和接收信号,每个节点都可以与一个或者多个对等节点进行直接通信,也可以转发其它节点的数据。在无线 mesh 网络中,节点一般是静止的且拓扑结构比较稳定,其功能旨在为客户端提供接入服务^[7]。无线 mesh 网络采用分布式控制,且具有自配置,自愈合功能^[8]。

无线 mesh 网络有三种结构:骨干网型无线 mesh 网络,客户端型无线 mesh 网络和混合型无线 mesh 网络^[9]。骨干型无线 mesh 网络里多个无线路由器形成 mesh 结构,作为骨干网络为客户端提供接入服务;客户端型无线 mesh 网络中客户端形成 mesh 结构,没有中心节点,每个客户端都具有自动配置和路由功能,从而形成多跳网络;而混合型无线 mesh 网络则混合了以上两种网络结构,骨干网和客户端都形成 mesh 结构。

与传统的单跳无线网络相比,无线 mesh 网络具有几个无可比拟的优势:

- (1) 快速部署和易于安装

安装 mesh 节点非常简单, 由于网络具有自配置和自愈合功能, 只需要将设备从包装盒里取出来, 接上电源就行了。由于极大地简化了安装, 用户可以很容易增加新的节点来扩大无线网络的覆盖范围和网络容量。

(2) 覆盖范围大

由于无线 mesh 网络能实现多跳传输, 这样和接入点一跳范围内的邻近用户可以为距离接入点多跳的用户提供了无线宽带访问功能, 从而具有非视距传输的特性。无线 mesh 网络能够非视距传输的特性大大扩展了无线宽带的应用领域和覆盖范围。

(3) 健壮性

实现网络健壮性通常的方法是使用多路由器来传输数据。mesh 网络比单跳网络更加健壮, 因为它不依赖于某一个单一节点的性能。在单跳网络中, 如果某一个节点出现故障, 整个网络也就随之瘫痪。而在 mesh 网络结构中, 由于每个节点都有一条或几条传送数据的路径。如果最近的节点出现故障或者受到干扰, 数据包将自动路由到备用路径继续进行传输, 整个网络的运行不会受到影响。

(4) 结构灵活

在单跳网络中, 设备必须共享 AP。如果几个设备要同时访问网络, 就可能产生通信拥塞并导致系统的运行速度降低。而在多跳网络中, 设备可以通过不同的节点同时连接到网络, 因此不会导致系统性能的降低。Mesh 网络还提供了更大的冗余机制和通信负载均衡功能。在无线 mesh 网络中, 每个设备都有多个传输路径可用, 网络可以根据每个节点的通信负载情况动态地分配通信路由, 从而有效地避免了节点的通信拥塞。而目前单跳网络并不能动态地处理通信干扰和接入点的超载问题。

(5) 带宽利用率高

无线通信的物理特性决定了通信传输的距离越短就越容易获得高带宽利用率, 因为随着无线传输距离的增加, 各种干扰和其他导致数据丢失的因素随之增加。因此选择经多个短跳来传输数据将是获得更高网络带宽利用率的一种有效方法, 而这正是 mesh 网络的优势所在。

在 mesh 网络中, 一个节点不仅能传送和接收信息, 还能充当路由器对其附近节点转发信息, 随着更多节点的相互连接和可能的路径数量的增加, 总的带宽也大大增加。

此外, 因为每个单跳的传输距离短, 传输数据所需要的功率也较小。既然多跳网络通常使用较低功率将数据传输到邻近的节点, 节点之间的无线信号干扰也较小, 网络的信道质量和信道利用效率大大提高, 因而能够实现更高的网络容量。比如在高密度的城市网络环境中, Mesh 网络能够减少使用无线网络的相邻用户的相互干扰, 大大提高信道的利用效率。

1.3 无线 mesh 网络关键技术

虽然无线 mesh 网络相比传统无线网络有很多优点, 但是目前其本身也存在很多问题

有待解决,当前阶段主要的研究集中在以下几个方面:

(1) 天线技术

无线 mesh 网络中一个重要的问题就是天线的使用,因为每个节点必须和各个方向上的多个节点通信,很简单的一种方式就是采用全向天线,但是这样覆盖范围有限,并会带来干扰,导致频谱利用率下降,减小网络的容量。所以不建议采用全向天线,美国的 SkyPilot 公司将智能天线技术应用于无线 mesh 网络,允许频谱重复使用,从而大大提高了频谱的利用效率,同时减小了干扰。

(2) 路由选择技术

无线 mesh 网络中另外一个很重要的问题是路由选择问题,例如从某个节点 A 到达其它的节点 B,可以经过不同的用户站中转,于是就存在多条路径,选择哪条路径就成为一个关键问题,这将直接影响系统的性能。而且,当节点加入或者推出网络时,无线 mesh 网络的拓扑结构就会发生变化,路由选择问题变的更加复杂。无线 mesh 网络中还存在“隐藏终端”问题,这些都需要进一步研究解决。文献[10]对无线 ad_hoc 网络中采用的基本路由技术进行了很好的总结和对比。

(3) 网络扩展性

基于现有的 MAC,路由和传输层协议,吞吐率,端到端时延和公平性等衡量网络性能的关键指标随网络中的节点数和跳数变化都是不可扩展的。这些问题可以通过提高单个节点的容量得到缓解。典型的方法包括:采用多个信道^[11]或者多个收发机^[12],开发具有更高传输速率的无线收发机。然而这些方法都不能真正地提高网络的扩展性,因为相关的网络性能并没有随着网络容量的增加而提高,因此必须研究新的 MAC,路由和传输协议来增强网络的扩展性。

(4) 安全性

由于无线自组织的结构,无线 mesh 网络在不同的协议层次对安全攻击^[13]都比较脆弱。目前的安全措施只对某一层有效,缺乏一种能阻止或反击各层的安全攻击的综合机制。

1.4 无线 mesh 网络性能

吞吐率是衡量网络性能的一个重要指标,在无线 mesh 网络中,由于数据采用多跳传输,随着跳数的增加,网络吞吐率急剧下降。公平性是衡量无线网络性能的另一重要指标,由于 mesh 网络主要为客户端提供接入服务,所有的数据交换必须通过接入点(网关)进行转发,因此接入点附近流量较大,而网络中其它一些节点则可能比较空闲,从而形成业务的“热区”和“冷区”^[14]。另外,无线 mesh 网络采用的 MAC 机制也会对网络公平性造成影响。当前对吞吐率的分析建模主要集中在单跳网络且缺乏对 IEEE802.15.4 低速网络的精确分析,因此,如何对低速无线 mesh 网络进行精确的建模,对上述指标进行分析,仿真,从而研究出提高网络性能的方法是关键所在。

1.4.1 网络吞吐率

目前对无线网络吞吐率的分析建模主要集中在对 IEEE 802.11 系列标准协议的研究上。文献[15]中为所有采用指数退避方法的该类协议提供了计算单跳网络饱和吞吐率的简单模型。文中假设任意节点发送产生冲突的概率相互独立且为常数,在该条件下计算了指数退避的 DCF 饱和吞吐率,包括三种模式:基本模式,RTS/CTS 模式,混合模式。文中定义吞吐率为:一个时隙内发送的平均负载长度除以一个时隙的平均长度,根据马尔可夫模型推导出节点在任一时隙的发送概率,再计算网络中至少有一个节点发送的概率和发送成功的概率,这样就可以根据成功发送一帧数据的时间和平均冲突时间计算出吞吐率。虽然文献中的仿真结果表明该模型有较高的精确性,然而它没有考虑网络的拓扑结构且没有表现发送端的相邻活动节点造成的冲突的特点。

在文献[16]中为节点具有相同传输范围的 802.11b 多跳自组织网络提出了单跳的饱和吞吐率分析模型。通过对由非邻近节点引起的干扰而导致的冲突和退避次数进行简单的启发式估计,该模型抓住了多跳环境下 CSMA/CA 协议的本质。仿真结果表明该模型很好地逼近了实际吞吐率。另外文献[17]提出了计算多跳无线网络路径吞吐率的方法。采用 IEEE 802.15.4 标准构建 mesh 网络,如果节点处于退避状态时不接收帧,那么在饱和的情况下通过多跳传输的节点只能获得很低的吞吐率,因此我们应该考虑对标准做一些扩展,即在退避时节点打开接收机并接收其它节点发送的帧,这也称为“接收优先^[18]”,我们将研究这种情况下 mesh 网络中任意节点的吞吐率。

1.4.2 无线 mesh 网络公平性

文献[19]认为无线网络中的“隐藏终端”和指数退避方法会导致网络的公平性问题,使部分节点相比其它节点获得更多的数据发送的机会。而文献[20]则认为 802.11 系列协议使节点获得了很好的 MAC 层公平性,通过采用相同的初始退避窗口,每个节点可以获得相似的接入信道的能力。网络公平性包括节点间的公平性和节点内的公平性,节点间的公平性是指网络中节点所处位置不一样,竞争和干扰状况也不一样,从而导致不同节点获得的吞吐率也不一样;节点内的公平性指多跳网络中,节点不仅要发送自身生成的数据,也要转发其它节点发送的数据,由于节点存储空间有限,如果调度方式不合理会导致自身数据和转发数据不能获得相同的发送机会,从而引起不公平。对节点间的公平性,一般采取选路算法去平衡负载;而对节点内部的公平性则通过队列调度去解决。对采用固定路由的 mesh 网络,节点间的公平性不能通过选路策略来解决,因此我们将研究采取根据网络层获取的信息来控制 MAC 层参数的方法去解决上述问题。

1.5 论文的主要研究内容与章节安排

本文主要进行了 WPAN mesh 网络的节点吞吐率建模分析和网络公平性的研究。本文从低速无线个域网 IEEE 802.15.4 出发,研究了低速无线 mesh 网络的节点饱和吞吐率,首

先对无线 mesh 网络的节点饱和吞吐率进行数学建模, 然后根据实际场景进行仿真, 验证了模型的准确性。此外本文还对无线 mesh 网络中节点的公平性进行了研究, 改进了标准 CSMA/CA 算法, 有效提高节点的负荷公平性。最后本文对无线 mesh 网络实验平台的软件实现进行了介绍。

第一章是绪论。首先介绍了无线 mesh 网络的基本概念, 网络架构及其相较于传统网络的优点, 其次简要介绍了当前无线 mesh 网络的关键技术, 进而介绍了无线 mesh 网络的性能研究背景, 主要包括吞吐率研究和网络公平性研究。最后简述了论文的研究安排。

在第二章, 介绍了低速 WPAN 的标准 IEEE802.15.4 及其网络层的扩展草案 IEEE802.15.5。对 IEEE802.15.4, 主要介绍了其物理层功能, MAC 层功能, CSMA/CA 算法以及帧结构等; 对 IEEE802.15.5, 主要介绍了其网络层基本功能和路由算法。在本章中, 我们将看到该标准如何对 mesh 结构进行支持, 这为后续章节的吞吐率建模和公平性研究提供了基础, 并为第五章的软件实现提供了协议规范。

第三章为全文的重点, 介绍了确定基于 IEEE802.15.4 和 IEEE802.15.5 的无线 mesh 网络的节点饱和吞吐率的数学模型。首先介绍了建立该模型的无线 mesh 网络结构, 然后在此基础上给出了计算节点饱和吞吐率的数学模型, 并通过仿真进行了验证。

第四章研究了网络公平性的问题, 针对网络中存在的业务“冷区”和“热区”问题以及节点负荷不均匀的状况进行了研究, 提出了优化解决方案, 从而使网络中节点的流量吞吐率尽量接近, 从而提高了网络性能。

第五章在实验室搭建的硬件实验平台的基础上, 进行了软件实现, 从而形成完整的基于 IEEE802.15.4 标准及 IEEE802.15.5 草案的无线 mesh 网络实验平台。该软件采用分层结构并对硬件的有限内存进行有效管理, 使网络便于升级, 可以在该平台的基础上试验不同的算法。

第六章是总结了全文的工作, 并对进一步工作做了展望。

第2章 IEEE 802.15.4 标准及 IEEE 802.15.5 草案概述

2.1 引言

本章主要对 IEEE802.15.4 标准和 IEEE802.15.5 草案作一个概述, 并对标准里如何对 mesh 结构进行支持作了简单介绍。IEEE802.15.4 标准定义了无线个域网 (WPAN) 中设备间通信的协议, 标准采用冲突避免的载波侦听(CSMA/CA)机制并支持星型和对等拓扑网络。标准规定了 868/915 MHz 和 2.4 GHz 两个频段的物理层和 MAC 层规范, 其传输速率分为: 20 kbps, 40 kbps 和 240 kbps, 是一种低速率无线网络标准。该标准主要用于低速率的无线固定, 手持或者移动设备间的通信, 通常设备采用电池供电且通信范围在 10 m 以内, 其应用范围主要为低复杂度, 低成本, 低功耗的智能家居, 无线玩具, 传感器网络和工业控制设备等。IEEE802.15.5 是 WPAN mesh 网络的候选草案, 支持高速和低速 WPAN 网络, 前者主要是对 IEEE802.15.3 标准的网络层扩展, 而后者是对 IEEE802.15.4 标准的网络层扩展。

2.2 IEEE 802.15.4 标准概述

IEEE802.15.4 标准定义了低速 WPAN 网络中无线设备间通信的物理层和 MAC 层规范, 通常这些设备都是电池供电的手持或者移动设备, 它们的通信范围较小, 一般不超过 10m。低速 WPAN 网络是一种简单的, 低成本的通信网络, 网络中允许低功耗的, 对吞吐率要求不高的无线应用。低速 WPAN 的目标是易于安装, 可靠的传输, 短距离通信, 极低的能耗, 合理的电池寿命且协议简单, 灵活。低速 WPAN 网络的一些特点包括:

- ✓ 传输速率为: 20 kbps, 40 kbps 和 250 kbps
- ✓ 星型或者对等网络
- ✓ 分配 16bit 短地址或 64bit 长地址
- ✓ 分配有保证时隙(GTS)
- ✓ 带冲突避免的载波侦听
- ✓ 采用完全应答机制以提高传输可靠性
- ✓ 低能耗
- ✓ 能量检测
- ✓ 链路质量指示
- ✓ 2.4 GHz 频段 16 个信道, 915 MHz 频段 10 个信道, 868 MHz 频段 1 个信道

低速 WPAN 网络中有两种设备：全功能设备(FFD)和精简功能设备(RFD)，FFD 可以工作在三种模式，分别作为网络协调器，协调器或者普通设备。FFD 可以和网络中所有设备进行通信，而 RFD 只能作为普通设备且只能和 FFD 通信。

2.2.1 IEEE 802.15.4 网络结构

符合 IEEE802.15.4 标准的 WPAN 一般包括至少两个设备，其中至少有一个全功能设备作为网络协调器。这些设备要在至少一个其它设备的通信范围以内以保持网络的连通性，网络中每个设备可以和通信范围内的其它设备通信或者只能和相邻的 FFD 通信，这样就分别形成了对等网络和星型网络，它们的结构如图 2-1 所示。

在星型网络中，所有设备都与中心控制设备 PAN 网络协调器通信。普通设备一般要通过关联过程加入到网络中，且一般为通信的发起者或结束者。而网络协调器一般有特别的用途，但一般被用作建立网络，路由数据等。在这种网络中，网络协调器一般使用持续电力系统供电，而其他设备采用电池供电。

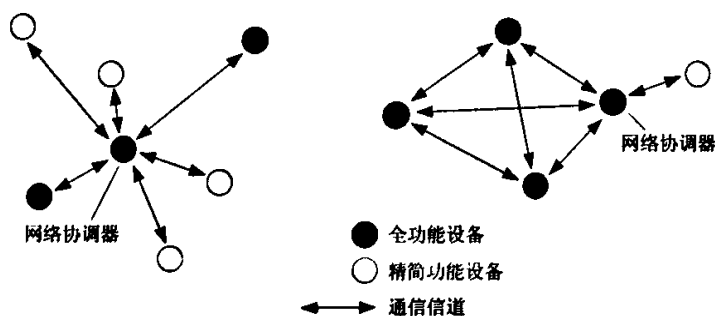


图 2-1 星型网络和点对点网络

对等网络中同样有一个网络协调器，但是和星型网络不同的是对等网络中任何两个设备之间都可以直接通信，只要彼此都在对方的无线辐射范围之内。对等网络允许复杂的网络形成过程，例如 mesh 网络拓扑。使用这种拓扑结构的应用场景很多，包括工业控制或监控，无线传感器网络，资产和库存跟踪，智能建筑和安全场合应用等。对等网络可以是 ad_hoc，自组织和自愈的，也允许数据在设备间通过多跳传输。

星型网络的形成过程如下：FFD 首次激活后将建立自己的网络且成为网络协调器，通过选择一个唯一的网络标识号，多个星型网络可以独立共存，当确定网络标识号后，网络协调器允许其它设备加入到网络中，包括 FFD，这样一个包含多个设备的星型网络就建立起来了。比较重要的是选择网络标识号的算法以及发现网络中冲突的标识号后如何处理，这将在网络层的协议中讨论。

对等网络中所有设备都能和它通信范围的其它设备通信，因此可以采用某种算法确立一个 FFD 为网络的协调器，例如最先启动的那个 FFD，之后其它设备可以通过关联加入到网络中。可以通过这种方式形成一个簇状树网络，网络协调器作为簇头形成了网络中的第一个簇，并在网络中向其它设备广播信标，其它设备收到信标后可以通过关联过程申请加入网络，RFD 只能作为树的叶子节点加入网络，即它不能允许其它设备关联到它。通过这样的方法可以形成一个多层的簇状网络。

2.2.2 IEEE802.15.4 物理层规范概述

IEEE802.15.4 标准物理层定义了物理无线信道和 MAC 子层之间的接口, 提供物理层数据服务和物理层管理服务。物理层数据服务提供物理层数据单元(PPDU)通过无线物理信道进行收发, 而物理层管理服务维护一个由物理层相关参数组成的数据库。

物理层提供的主要功能包括:

- 激活和关闭收发器
- 信道能量检测
- 链路质量指示(LQI)
- 信道选择
- 空闲信道侦听(CCA)
- 数据收发

IEEE802.15.4 物理层定义了三个频段用于收发数据, 它们分别为:

- 868-868.6 MHz(用于欧洲)
- 902-928 MHz(用于北美)
- 2400-2483.5 MHz(用于全球)

符合 IEEE802.15.4 标准的设备必须运行在以下一个或多个频段, 且各个频段有不同的调制方式和传输格式, 具体规定如表格 2-1 所示:

表格 2-1 载波信道特性一览表

PHY (MHz)	频段 (MHz)	序列扩频参数		数据参数		
		片速率 (kchip/s)	调制方式	比特速率 (kbps)	符号速率 (ksymbol/s)	符号 (symbol)
868/915	868~868.6	300	BPSK	20	20	二进制位
	902~928	600	BPSK	40	40	二进制位
2450	2400~2483.5	2000	O-QPSK	250	62.5	十六进制

IEEE802.15.4 物理层规定了 27 个通信信道, 编号分别为 0~26, 其中 2.4 GHz 频段有 16 个信道, 915 MHz 频段有 10 个信道, 868 MHz 频段有 1 个信道。各信道的中心频率按以下公式计算:

- $F_c = 868.3, k = 0$
- $F_c = 906 + 2(k - 1), k = 1, 2, \dots, 10$
- $F_c = 2405 + 5(k - 11), k = 11, 12, \dots, 26$

2.2.3 IEEE802.15.4 MAC 层规范概述

IEEE802.15.4 MAC 规范定义了 MAC 子层和上下层间接口, 提供两种服务: MAC 层数据服务和 MAC 层管理服务。MAC 层数据服务主要负责 MAC 协议数据单元在物理层数据服务中的正确收发, 而 MAC 层管理服务提供层管理功能并维护一个存储 MAC 子层被管

理对象信息的数据库。

MAC 子层提供的主要功能包括：

- 如果设备是协调器的话，负责生成网络信标帧
- 和信标同步
- 支持 PAN 关联和解关联
- 支持设备安全
- 利用 CSMA/CA 机制接入信道
- 处理并维护 GTS 机制
- 在对等 MAC 层实体间提供可靠链路

2.2.4 MAC 层帧结构

MAC 层帧由三个部分组成：

- MAC 层帧头(MHR)：包括帧控制字段，帧序号和地址信息
- MAC 层负载：长度不固定，包含特定帧类型的信息，应答帧没有负载
- MAC 层帧尾部(MFR)：包含帧校验序列(FCS)

基本的 MAC 层帧格式如图 2-2 所示：

字节数:2	1	0/2	0/2/8	0/2	0/2/8	可变	2
帧控制 信息	帧序 列号	目的设备 PAN 标识符	目标 地址	源设备 PAN 标识符	源设备 地址	帧数据 单元	FCS 校验码
		地址信息					
帧头						MAC 负载	MFR 帧尾

图 2-2 MAC 层基本帧格式

帧控制字段长度为 16bit，包含帧类型，地址域和其它控制标志，帧控制字段格式如图 2-3 所示：

比特位:0~2	3	4	5	6	7~9	10~11	12~13	14~15
帧类型	安全使能	有后续帧	应答请求	网络内部传输	保留	目的地址模式	保留	源地址模式

图 2-3 帧控制字段结构

各部分的含义如下：

- 帧类型：目前定义了四种帧类型，分别为信标帧(000)，数据帧(001)，应答帧(002)和命令帧(003)，其它的保留为以后扩展
- 安全使能：为 1 则需要对数据进行加密，为 0 则不需要
- 有后续帧：为 1 说明发送该帧的设备紧接着这帧数据还有其它的数据要发送给接

收端，为 0 则无后续数据发送，该标志位仅用于信标使能网络中的无竞争时段或者用于非信标使能网络中

- 应答请求：为 1 则要求接收方收到一帧数据或命令且确认帧有效后作出应答，为 0 则无需应答
- 网络内部传输：为 1 说明数据在 PAN 内部传输，如果帧头部给出了源地址和目的地址则无需再给出源网络标识号，否则说明数据在不同的 PAN 间传输，不能省略源网络标识号
- 目的地址模式：目前给出了三种模式，分别为无需网络标识号和地址(00)，地址域包含 16bit 短地址(10)和地址域包含 64bit 长地址(11)，模式 01 保留
- 源地址模式：同目的地址模式

帧序列号长度为一个字节，唯一地确定了一个帧，数据帧和命令帧使用统一的序列号，信标帧采用独立的序列号。系统启动时，帧序列号被初始化为一个随机值，每次生成一个帧时帧序列号在上帧的基础上加 1。如果帧要求应答，则应答帧将把收到帧的序列号拷贝到本身的序列号域，如果帧发送失败，重发帧将使用上次发送失败帧的序列号。

IEEE 802.15.4 标准定义的四种帧具体格式如下：

(1) 信标帧

信标帧格式符合基本的 MAC 层帧格式，其特别之处在于它的负载部分，信标帧的负载包括四部分：超帧描述字段、GTS 分配字段、待处理数据目的地址字段和信标帧负载数据，如图 2-4 所示。

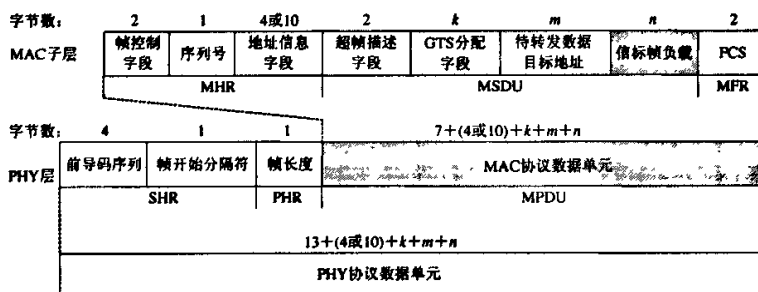


图 2-4 信标帧格式

信标帧中超帧描述字段规定了这个超帧的持续时间，活跃部分持续时间以及竞争访问时段持续时间等信息。

GTS 分配字段将无竞争时段划分为若干个 GTS，并把每个 GTS 具体分配给了某个设备。

转发数据目标地址列出了与协调器保存的数据相对应的设备地址。一个设备如果发现自己的地址出现在待转发数据目标地址字段里，则意味着协调器存有属于它的数据，所以它就会向协调器发出请求传送数据的 MAC 命令帧。

信标帧负载数据为上层协议提供数据传输接口。例如在使用安全机制的时候，这个负

载域将根据被通信设备设定的安全通信协议填入相应的信息。通常情况下这个字段可以忽略。

在非信标使能网络里，协调器在其他设备的请求下也会发送信标帧。此时信标帧的功能是辅助协调器向设备传输数据，整个帧只有待转发数据目标地址字段有意义。

(2) 数据帧

数据帧用来传输上层发到 MAC 子层的数据，它的负载字段包含了上层需要传送的数据。数据负载传送至 MAC 子层时，被称为 MAC 服务数据单元 (MSDU, MAC service data unit)。它的首尾被分别附加了 MHR 头信息和 MFR 尾信息后，就构成了 MAC 帧，如图 2-5 所示。

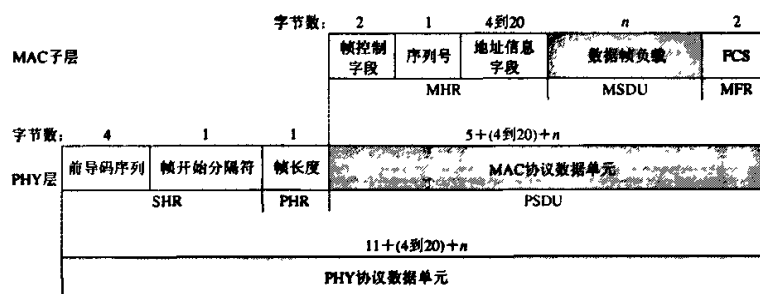


图 2-5 数据帧的格式

MAC 帧传送至物理层后，就成为了物理帧的负载 PSDU。PSDU 在物理层被“包装”，其首部增加了同步信息 SHR 和帧长度 PHR 字段。同步信息 SHR 包括用于同步的前导码和帧开始分隔符字段，它们都是固定值。帧长度字段 PHR 标识了 MAC 帧的长度，为一个字节长而且只有其中的低 7 位是有效位，所以 MAC 帧的长度不会超过 127 个字节。

(3) 确认帧

如果设备收到目的地址为其自身的数据帧或 MAC 命令帧，并且帧的控制信息字段的确认请求位被置 1，设备需要回应一个确认帧。确认帧的序列号应该与被确认的帧的序列号相同，并且负载长度应该为 0。确认帧紧接着被确认的帧发送，不需要使用 CSMA/CA 机制竞争信道，如图 2-6 所示。

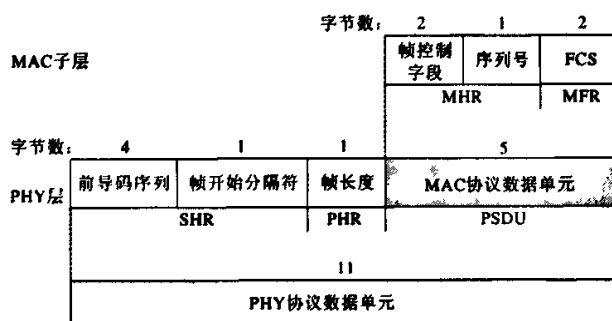


图 2-6 确认帧的格式

(4) 命令帧

MAC 命令帧用于组建 PAN 网络，传输同步数据等。目前定义好的命令帧有 9 种类型，主要完成三方面的功能：把设备关联到 PAN 网络，与协调器交换数据，分配 GTS。命令帧在格式上和其他类型的帧没有太多的区别，只是帧控制字段的帧类型位有所不同。帧头的帧控制字段的帧类型为 011，表示这是一个命令帧。命令帧的具体功能由帧的负载数据表示。负载数据是一个变长结构，所有命令帧负载的第一个字节是命令类型字节，后面的数据针对不同的命令类型有不同的含义，如图 2-7 所示。

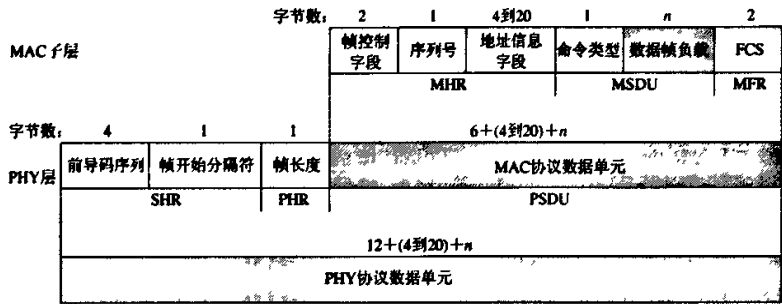


图 2-7 命令帧的格式

2.2.5 MAC 层功能概述

2.2.5.1 信道接入

IEEE802.15.4 标准提供了两种信道接入模式：基于竞争的和无竞争的。基于竞争的信道接入模式允许设备利用指数退避算法以分散的方式接入信道，而无竞争接入模式通过使用有保证时隙由网络协调器集中控制。由于使用无竞争接入模式比较复杂且不容易实现 mesh 结构网络，因此我们将重点讨论采用竞争方式接入信道的非信标使能网络。

非信标使能网络中的设备采用非时隙 CSMA/CA 算法竞争信道，每次发送数据帧或者命令帧前必须通过竞争获得信道的使用权。信标帧和应答帧的发送不采用竞争信道方式，而是直接进行发送。非时隙 CSMA/CA 算法中，设备退避的时间点不需要同步，可以在任意时刻进行退避。网络中每个竞争信道的设备必须为每次发送独立维护两个变量：退避次数(NB)和退避指数(BE)。NB 是设备进行发送前要进行退避的次数，在每次新的传输前要被置为 0，BE 关系到设备接入信道前要退避的时隙个数，在非时隙算法中 BE 要被初始化为 macMinBE。虽然算法中信道评估期间设备接收机被打开，但此时收到的所有帧都将被丢弃。具体的非时隙 CSMA/CA 算法流程如图 2-8 所示。

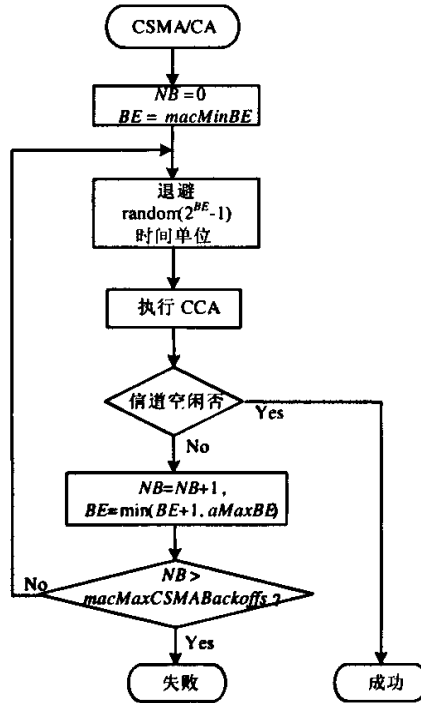


图 2-8 非时隙 CSMA/CA 算法流程

2.2.5.2 建立和维护网络

PAN 的建立只能由 FFD 完成，其过程如下：FFD 产生 MAC 层复位请求原语，请求复位 MAC 层，并执行主动信道扫描，选择合适的网络标识号和自身短地址，短地址的值必须小于 0xffff。选用网络标识号可以采用不同的算法，例如可以随机选择。PAN 建立后，FFD 就通过 MAC 层启动请求原语建立 PAN。MAC 子层收到该原语后，就更新信道参数等相关信息，然后向上层发布确认原语，且返回状态为成功，这样该 FFD 就作为 PAN 的协调器建立了网络。建立网络后，协调器向网络广播信标帧，告知其它设备网络的信息，其它设备收到信标帧后可以选择加入网络。

建立网络和设备加入网络都必须进行扫描，IEEE802.15.4 标准规定了四种扫描方式，分别为信道能量扫描，主动信道扫描，被动信道扫描和孤立设备扫描。所有设备都具备对指定信道进行被动扫描和孤立设备扫描的功能，FFD 还能对指定信道进行主动扫描和能量扫描，各种扫描方式基本原理如下：

(1) 信道能量扫描

MLME 首先通过设置 phyCurrentChannel 和 phyChannelPage 转移到该逻辑信道上，然后反复进行信道能量测量，持续时间是 $[aBaseSuperframeDuration * (2^n + 1)]$ ，在进入下一个信道扫描时，必须保存扫描过程中获得的最大 ED 度量值，当存取信道数目等于要求信道数或者已经完成每个指定逻辑信道的能量度量，停止 ED 扫描，扫描期间 MAC 子层丢弃所有通过物理层数据服务接收的帧。

(2) 主动信道扫描

设备通过主动信道扫描，能够搜索到自己通信范围内可以发送信标帧的所有协调器，预

期的网络协调器可利用这一结果在建立新的PAN前选择一个网络标识号,普通设备则利用扫描结果进行关联。首先通过设置`phyCurrentChannel`和`phyCurrentPage`的属性,转移到相应的逻辑信道,并发送信标请求命令,发送成功后,设备就激活接收机等待接收信标帧,使能状态的持续时间不能大于规定的最大时间,即 $[aBaseSuperframeDuration * (2^n + 1)]$,其中 n 是参数`ScanDuration`的属性值,即扫描的持续时间。扫描过程中,设备只接收信标帧,并记录PAN描述符结构中信标帧的信息,直至扫描结束,再利用扫描确认原语把它递交给上一高层,扫描期间只接收信标帧。

(3) 被动信道扫描

和主动信道扫描相似,区别在于被动信道扫描期间不发送信标请求帧,只是打开接收机接收网络中的信标。

(4) 孤立设备扫描

设备通过孤立信道扫描,重新定位失去同步的协调器,扫描期间只接收由物理层数据服务得到的协调器发出的重关联命令帧。设备首先对`phyCurrentChannel`和`phyCurrentPage`的属性进行设置,转移到相应的逻辑信道,然后发送孤立扫描的通知命令,成功发送后,设备激活接收机,保持到`macResponseWaitTime`结束。如果在这期间,设备成功接收到协调器的重关联命令,就中止扫描。

2.2.5.3 设备关联和解关联

设备通过发布原语 MAC 层复位原语对 MAC 子层复位,并进行主动或被动信道扫描后,才能和 PAN 关联,其中信道扫描的结果用于选择 PAN。关联过程为: 选择要关联的 PAN 后,高层就通过关联请求原语请求 MAC 层管理实体设置物理层和 MAC 层的 PIB 属性以便关联。之后设备向选定的网络协调器发送关联请求帧,如果协调器允许设备关联则给设备分配一个 16bit 短地址并通过关联响应帧传递该信息,否则不允许设备关联。设备收到关联响应帧后根据关联结果更新相应信息。

解关联过程和关联过程相似,设备通过解关联离开网络。解关联可以由网络协调器发起,通知已关联设备离开网络,也可以由设备自己发起。

2.2.5.4 事务处理

IEEE802.15.4 标准是为低成本设备制订的,即设备一般使用电池供电,因此事务的处理请求来自设备而不是协调器,且数据传输一般采用间接方式,即协调器必须在信标帧中指示设备有待处理的数据,或是设备询问协调器,是否有其尚未处理的事务。

当协调器的上层生成数据请求原语,或是 MAC 层管理实体请求发送 MAC 命令时,协调器开始处理事务。事务处理完毕,MAC 子层就给高层返回状态信息。如果请求原语发起间接传输,就要使用相应的确定原语返回状态值。间接数据传输请求里包含的信息构成了一个事务,协调器至少能储存一个事务。当协调器收到请求而没有足够的存储空间,MAC 子层就向上层发布恰当的原语,返回状态事务溢出。在收到间接数据传输请求后进行事务处理。

若协调器有足够的存储空间,则确保同一个设备的事务按照它们到达 MAC 子层的顺序进行传输。如果所传输的事务不是最后一个,MAC 层就把帧中的有后续帧域设为 1,指示还有需要处理的事务。

每个事务在协调器中的停留最长时间为 *macTransactionPersistenceTime*，若超过，MAC 层将丢弃该事务的信息，且向上层发布一个原语，返回事务过期状态。为确保每个事务都能成功提取，设备必须对接收的事务回复确认帧。

如果成功处理某个事务，MAC 子层就删除该事务的所有信息，并通过相应的原语向上层返回成功状态。

协调器在发送信标帧时，则应在地址列表中列出事务相关设备的地址，且在待处理地址域中说明地址的数目。若协调器能储存 7 个以上的事务，则按照先到先服务的原则在信标帧中指明，确保帧中包含的地址不超过 7 个。对用 GTS 进行传输的事务，信标帧不包含对应设备的地址，协调器只是在分配给设备的有保证时隙中进行传输。

2.2.5.5 数据传输

在非信标使能网络中，数据传输有两种方式：直接传输和间接传输。若采用直接传输方式，设备在使用 CSMA/CA 算法竞争信道后就直接发送帧给目的设备，而采用间接传输方式，帧被发送到协调器并存储在事务队列里，设备再通过发送数据请求帧定期从协调器取得数据。发送帧时，MAC 子层首先通过设置收发器状态请求原语请求物理层打开发射机，数据发送后再通过该原语将收发器状态改为关闭或接收机启动。

在空闲阶段，每个设备的 MAC 子层都要选择是否让接收机保持使能状态，因为 MAC 子层仍要处理来自高层的收发请求。收发任务可以是带有确认（如果要求）的发送请求，也可以是接收请求。每完成一个收发任务，MAC 子层将根据 *macBeaconOrder* 和 *macRxOnWhenIdle* 值来请求物理层激活或关闭接收机。设备收到一帧数据后将对数据进行多重过滤。第一层过滤将丢弃帧校验结果错误的帧，第二层过滤则取决于 MAC 子层是否工作在混合模式。若是，MAC 子层把所有通过第一层过滤的帧直接传送到上层，不需要再进行任何过滤或处理。当 *macPromiscuousMode* 设为 TRUE 时，MAC 子层的工作模式为混合模式。

如果 MAC 子层不处于混合模式，那么它将只接收通过第二层过滤的帧：

- 帧类型子域不含保留帧类型
- 如果帧中包含目的 PAN 标识符，那么它应和 *macPANId* 相匹配，或是广播 PAN 标识符 (0xffff)
- 如果帧中含有目的设备的短地址，那么它应和 *macShortAddress* 匹配，或是广播地址 (0xffff)。否则如果包含的是扩展地址，必须和 *aExtendedAddress* 匹配
- 如果帧是信标帧，源 PAN 标识符应和 *macPANId* 匹配，而当 *macPANId* 是 0xffff 时，忽略源 PAN 标识符并接收该信标帧
- 如果数据帧或 MAC 命令帧只包含源地址域，仅当设备是 PAN 协调器，并且源 PAN 标识符与 *macPANId* 匹配时，才接收帧

如果不满足第二层过滤的条件，MAC 子层就丢弃接收的帧。通过第二层过滤的帧被认为是有效帧，才能进一步处理。对于非广播的有效帧，如果是数据帧或 MAC 命令帧，且帧控制字段的确请求应答域置为 1，MAC 子层需要发送确认帧，且确认帧的序号和所收数据帧或 MAC 命令帧的序号应相同，这样做的目的在于使发送数据帧或 MAC 命令帧的设备收到确认帧后，可以判定该确认帧是发给自己的。

如果帧中包含目的地址和源地址的信息，且帧控制字段中的帧内传输域设为 1，MAC 子层就认为被忽略的源 PAN 标识符和目的 PAN 标识符相同。

在发送数据帧或 MAC 命令帧时, 将根据需要恰当地设置帧控制字段的请求应答域, 而对于信标帧或确认帧, 请求应答域总设为 0, 同样的, 所有广播帧的请求应答域也设为 0。如果发送帧的请求应答域设为 0, 接收方将不发送确认帧, 且发送设备也会认为帧已发送成功。如果帧控制域的请求应答域设为 1, 接收端在成功收到帧后就必须发送一个确认帧, 且确认帧的帧序号同接收的数据帧或 MAC 命令帧的序号相同。

如果发送的帧没有要求应答, 则总是认为帧发送成功, 即使实际上不成功也不需要重传。当设备发送的是数据帧或 MAC 命令帧且要求应答, 就要等待相应的应答帧。如果在规定的最大时间内, 即 *macAckWaitDuration* 个符号时间内收到确认帧, 且帧序号和所发送帧的序号相同, 发送设备判定发送成功, 如果在规定时间内没有接收到确认帧, 或是确认帧的序号和发送帧的序号不相同, 设备判定此次发送失败, 此时将重传帧直到达到最大重传次数 *macMaxFrameRetries*。重传帧的序号和上次发送失败帧的序号必须保持相同。

2.3 IEEE 802.15.5 草案概述

IEEE802.15.5 草案是对 IEEE802.15.3 标准和 IEEE802.15.4 标准的网络层扩充, 使其支持 mesh 结构网络。它包括高速和低速两个部分, 我们目前只讨论和低速相关部分, 主要包括帧结构, 地址分配和路由算法等, 下面将对各部分作简单介绍。

2.3.1 帧结构

基本的 Mesh 服务帧由几部分组成: 帧控制域, 目的地址域, 源地址域和帧负载域。具体结构如图 2-9 所示。

字节数 1	8/2	8/2	可变长度
帧控制	目的地址	源地址	Mesh 层负载
Mesh 层帧头			

图 2-9 mesh 层基本帧格式

帧控制域包含三个部分: 协议版本号, 帧类型和传输模式。协议版本号域长度为 4bit, 目前值为 0001, 草案中定义了两种帧类型, 分别为数据帧(00)和命令帧(01), 其它保留作为将来扩展, 传输模式有三种: 单播(00), 广播(01), 多播(10), (11)作为保留, 具体结果如图 2-10 所示。

比特位:7-4	3-2	1-0
协议版本号	帧类型	传输模式

图 2-10 mesh 帧控制字段结构

目的地址和源地址可以为 16bit 短地址或者 64bit 长地址, 且 mesh 层地址和 MAC 层地址相同。

2.3.1.1 数据帧格式

数据帧格式符合基本帧格式，具体结构如图 2-11 所示。

字节:1	8/2	8/2	1	1	可变
帧控制	目的地址	源地址	帧序号	路由控制	数据负载
Mesh层头部			Mesh层负载		

图 2-11 mesh 层数据帧格式

数据帧的 mesh 层负载部分包含 3 个子域，分别为帧序号域，路由控制域和数据负载域，路由控制域由上行-下行标志位和到目的节点邻居跳数域组成，如图 2-12 所示。上行-下行标志表明数据是否要沿树的上行或下行方向转发，到目的节点邻居跳数为当前转发节点到按规定算法确定的目的节点邻居的跳数。

比特位:7	6-2	1-0
上行-下行标志	到目的节点邻居跳数	保留

图 2-12 路由控制域结构

2.3.1.2 命令帧格式

IEEE802.15.5 草案目前定义了 15 种命令帧，其基本格式如图 2-13 所示。

字节:1	可变	1	可变
帧控制	路由域	命令帧子类型	命令负载
Mesh层头部		Mesh层负载	

图 2-13 mesh 层命令帧格式

命令帧子类型有子节点数目报告帧，地址分配帧，hello 帧，邻居信息请求帧，邻居信息应答帧，链路状态请求帧，链路状态应答帧，链路状态不匹配帧，探测帧，加入网络请求帧，加入网络应答帧，离开网络请求帧，离开网络应答帧，组协调器更新帧和多播组解散帧。其中 hello 帧对网络的形成和节点信息的收集至关重要，它的具体结构如图 2-14 所示。Mesh 层负载部分包括命令子类型，生存时间(TTL)，节点地址段起始地址，节点地址段结束地址，树层次，一跳邻居数目和一跳邻居地址。

字节:1	2	2	1	1	2	2	2	1	可变
帧控制	目的地址	源地址	命令帧子类型	生存时间	起始地址	结束地址	网层次	帧邻居数目	帧邻居地址
Mesh层头部			Mesh层负载						

图 2-14 mesh 层 hello 帧结构

2.3.2 Mesh 层功能概述

2.3.2.1 树形成和寻址

在低速 WPAN mesh 网络中, 为了寻址和路由网络中所有设备将形成一棵树。首先生成自适应树, 再生成 mesh 结构自适应树。自适应树的生成要经历两个阶段, 即初始化阶段和运行阶段。初始化阶段中, 设备先关联到网络, 然后逐层上报子节点数目, 当协调器判断所有的设备都关联到网络后进行地址分配, 这个时候自适应树就形成了。地址分配过程中, 可以为未来设备的加入预留部分地址, 具体的地址分配过程如图 2-15 所示, 图中方括号中的数字为子节点数目。

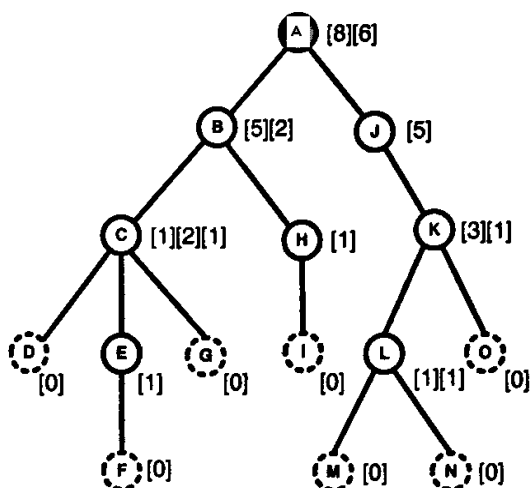


图 2-15 地址分配示意图

在运行阶段, 普通的数据传输过程为根据目的节点的地址块选择合适的路由并进行转发, 如果没有找到合适的路由则转发给父节点。在运行阶段, 仍然允许其它设备加入网络。Mesh 结构自适应树的生成不需要额外的动作, 只需要在前面生成的自适应树的基础上每个节点将它所有的邻居节点当成自己的子节点即可。这样数据转发可以有更短的路径且消除了单点故障(SPOF)造成的影响。

Mesh 结构自适应树形成后提供了基本的单播路由功能, 但此时的路由算法不够高效, 通过分布式地收集链路状态信息可以得到更有效地路由, 但节点间必须交换信息, 例如通过互相发送 hello 帧。总的来看, 单播路由不需要路由发现过程, 也不需要路由维护,

相比自适应树，mesh 结构自适应树提供了更短的路径和更好的鲁棒性，另外通过交换链路的状态信息，可以获得目的节点的多条路径且更易于扩展。

另外，IEEE802.15.5 草案还提供了多播的功能，设备可以选择加入或者离开一个多播组，本文中仅考虑单播问题。

2.3.2.2 邻居信息获取和维护

当设备加入网络且被分配地址后，它就向自己的邻节点广播几个 Hello 消息，通过交换 Hello 消息，每一个节点将建立邻居列表。节点不仅将 Hello 消息的源节点加入到邻节点列表，还将源节点的一跳邻节点加入到自己的邻节点的列表，除非 Hello 消息的 TTL 字段值为 1，邻节点表包括网络中离设备 TTL 范围内所有节点的信息，例如起始地址，结束地址，树层深度，和设备的关系及跳数，如图 2-16 所示。

起始地址	结束地址	树层深度	和设备关系	和设备跳数
BeginAddr1	EndAddr1	TreeLevel1	Parent/Child/Sibling	Hop1
...				
BeginAddrn	EndAddrn	TreeLeveln	Parent/Child/Sibling	Hopn

图 2-16 邻节点列表

根据收到的 Hello 帧设备将邻节点列表的所有信息填满，如果部分信息不全则设备向它的所有一跳邻居广播 hello 请求帧。

从 Hello 帧中的单跳邻节点信息中，一个节点可以为邻节点列表中的节点建立一个连通性矩阵，如图 2-17 所示。邻节点列表中的设备到每个邻节点的跳数可以通过连通性矩阵计算出来。首先，每个邻节点的跳数字段设置为无限，然后所有直接与设备相连的一跳邻节点的跳数设为 1，接着所有与单跳邻节点相连的节点跳数设为 2，这一过程不断进行下去直到所有的节点的跳数计算出来。

图中“+”和“-”标识直接连通和不直接连通，连接均为双向，所以矩阵为对称的，故只给出上半部分。

	me	nb1	nb2	nb3	...	Nbn-2	Nbn-1	nbn
me	-	-	+	-	...	-	+	-
nb1		-	+	-	...	+	-	-
nb2			-	+	...	-	-	-
nb3				-	...	+	-	-
...				
Nbn-2						-	-	+
Nbn-1							-	-
nbn								-

图 2-17 连通性矩阵

2.3.2.3 路由算法

IEEE802.15.5 草案中规定的路由算法实质上为最短路径算法，其寻址过程如下：首先从邻居列表的所有节点中寻找为目的节点的直系上层节点且其树层深度为最大，如果没有找到这样的节点且目的节点不是自己的直系子节点，则寻找是否有树层深度比自己小(更上层)且其树层深度与到本节点的跳数之和最小的节点，如果存在多个这样的节点则选择离自己最近的节点(到本节点跳数最小)，否则无法为该目的节点找到转发路由，可能要进行路由发现过程。

找到这样的节点后，首先把所有邻节点列表里所有节点的跳数设为无限大且当前跳数为本节点到找到的邻居节点的跳数，再将邻节点列表所有跳数等于当前跳数的一跳邻居的跳数设为当前跳数减 1，再将当前跳数减 1 重复上述过程直到当前跳数为 1，寻找下一跳节点的具体流程如图 2-18 所示。

如果通过算法找到多个满足条件的下一跳节点，则可以任意选择一个进行转发，为了平衡网络负载也可以在多次发送时使用不同的下一跳节点转发，但是为了避免接收端数据失序，同一个应用的所有数据必须用同一条路径发送。对找到的下一跳转发节点结果可以进行两种处理，即存储结果以便下次转发或者每次转发时重新进行上述过程。两种方法各有利弊，第一种方法可以节省处理器的计算时间，但是如果网络拓扑发生变化，例如有节点加入或离开现有网络，则可能需要重新寻找到目的节点的路由，第二种方法不会有这样的问题但是计算开销比较大，如果处理器速度较慢也可能会影响网络的吞吐率等性能。我们可以采用一种折中的方法，网络刚刚建立后进行一次上述路由发现过程并将结果写入节点中的路由表，每次有节点加入或离开网络时更新邻节点列表和连通性矩阵，并顺带更新路由。具体的路由维护方法本文不进行讨论。

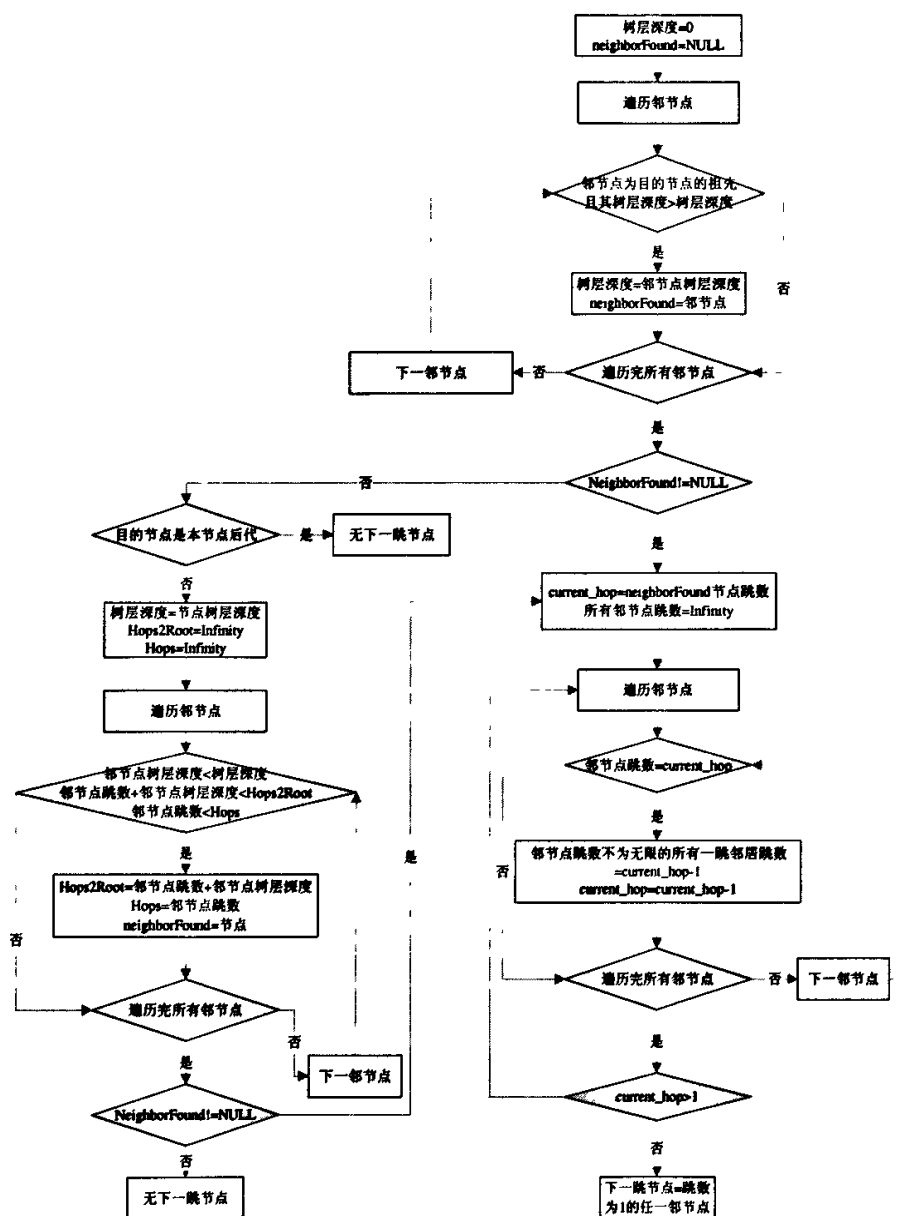


图 2-18 寻找下一跳转发节点算法流程

2.4 标准对 Mesh 网络的支持

IEEE802.15.4 标准里没有给出实现 mesh 网络的方法，而 IEEE802.15.5 草案也只是给出了网络层的协议，对如何与底层协议结合从而支持 mesh 网络没有给出详细的规定。标准中定义了两个结构的网络，星型网络和对等网络，利用对等方式可以形成 mesh 网络。网络协调器和所有的 FFD 相互连接，它们之间可以互相通信，形成骨干网络，RFD 关联到它们附近的 FFD 上。

网络采用非信标使能的方式, FFD 之间采用 CSMA/CA 机制竞争信道, 直接进行通信。根据 IEEE802.15.4 标准的规定, 为了降低能耗, RFD 只能和 FFD 通信, 采用两种传输方式, 即直接传输和间接传输。采用直接传输方式, RFD 有数据要发送时, 通过 CSMA/CA 算法和其它节点竞争信道, 获得信道使用权后把数据直接发送给和它关联的 FFD。一般 RFD 采用间接传输方式从 FFD 中取得其它节点发送给它的数据, RFD 定期向关联的 FFD 查询, 如果有发送给自己的数据则通过向 FFD 发送数据请求帧获得。

采用非时隙 CSMA/CA 算法, 节点有数据发送时首先进行退避, 如果这个时候其它节点给它发送数据, 标准没有规定如何处理。为了在节点饱和的情况下使通过多跳传输的节点获得较大的吞吐量, 借鉴 IEEE802.11 系列协议的退避机制, 节点将暂停退避, 直到接收的数据处理完后继续退避。

为了支持全 mesh 网络, 即网络中每个设备都能进行数据转发, 设备间都可以直接通信, 可以对标准里规定的传输机制进行适当改动, 这时 RFD 的接收机总是保持为开启状态, 也允许其它设备关联到它, 这时候 RFD 和 FFD 没有任何区别。但是由于 RFD 一般采用电池供电, 用这种方式 RFD 寿命比较短, 一般应用在设备能充电或者用主电源供电的场合。

网络的上层协议直接按照 IEEE802.15.5 草案规定, 组网过程中有一些算法没有具体规定, 例如选择关关节点的算法, 可以自己定义。草案中提出的路由算法实质上为最短路径算法, 没有利用链路状态等策略, 我们可以对它进行扩展, 采用高效算法从而提高网络性能。一个符合 IEEE802.15.4 标准和 IEEE802.15.5 草案的参考 mesh 网络结构如图 2-19 所示。

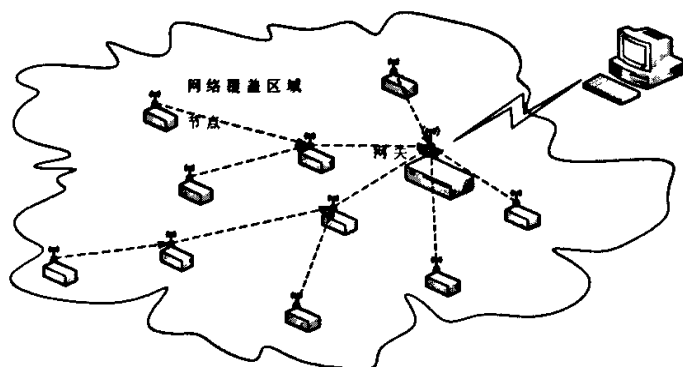


图 2-19 参考 mesh 网络结构

2.5 本章小节

本章对 IEEE802.15.4 标准和 IEEE802.15.5 草案进行了概述, 并对它们如何支持 mesh 网络进行了简单分析。IEEE802.15.4 标准规定了低速无线个域网的物理层和 MAC 层协议, 本章主要介绍了 MAC 层协议, 包括帧格式, 基本功能和 CSMA/CA 算法, IEEE802.15.5 草案是对无线个域网的网络层扩展, 包括高速和低速两个部分, 本文只介绍了低速部分的帧结构和基本功能以及网络形成过程, 路由算法等。最后对如何结合两个协议实现 mesh 网络进行了简单分析, 本章的内容是后续章节理论分析和软件实现的基础。

第3章 低速无线 mesh 网络节点吞吐率分析

3.1 引言

吞吐率是无线网络的重要性能指标,目前的研究主要集中在基于 IEEE802.11 系列协议的单跳网络上,也有部分研究关注了网络的端到端吞吐率和路径吞吐率,对基于 IEEE802.15.4 标准的低速 mesh 网络的吞吐率研究较少。另外,为了在饱和状态下使网络中多跳节点获得一定的吞吐率,必须让节点在退避时打开接收机,此时可以接收其它设备发送的数据帧,这也被称作“接收优先”。本章作为全文研究的重点,建立了一种在“接收优先”情况下确定基于 IEEE 802.15.4 标准和 IEEE 802.15.5 草案的 mesh 网络节点饱和吞吐率的分析模型,主要是根据 IEEE 802.15.5 草案中规定的方法首先确定每个节点一跳邻居节点数目和竞争范围内节点个数,再估计节点的发送概率和一个完整退避周期内平均接收到的数据帧的个数和平均退避时隙数从而得到每个节点的饱和吞吐率。

3.2 低速无线 mesh 网络节点吞吐率分析模型

在基于 IEEE 802.15.4 标准和 IEEE 802.15.5 草案的 mesh 网络中,每个节点都具有数据转发功能,所有数据通过多跳发送到网关节点,节点采用非时隙 CSMA/CA 机制竞争信道,为了实现全 mesh 网络,节点的接收机一直处于开启状态且退避期间可以接收帧。网络层采用 IEEE 802.15.5 草案规定的算法,在网络的形成阶段每个节点可以获得网络中所有其它节点的信息,包括距离自己的跳数和彼此间的相邻关系。

目前对无线 mesh 网络吞吐率的研究主要集中在 IEEE 802.11 系列网络上,由于 IEEE 802.15.4 标准中的 CSMA/CA 机制不采用 RTS/CTS 方式,因此原有的研究不能直接应用。另外对 IEEE 802.11 协议吞吐率的研究主要为单跳吞吐率^[21,22,23],所有节点只进行发送不能接收并转发其它节点的数据。有部分文献对节点端到端^[24]和路径吞吐率进行了分析,但是这些分析都是假设为线性拓扑或者路径中只有源节点产生数据,路径中其它节点只是单纯地转发数据,因此很多时候并不符合实际情况。

在多跳网络中节点任意分布,每个节点的邻居都不相同,干扰也不一样,因此研究网络中任意节点的吞吐率具有实际意义,本章将对 mesh 网络中每个节点的饱和吞吐率进行建模分析,力求比较准确地估计节点的吞吐率。

3.2.1 节点发送模型

在多跳网络中假设所有节点都是连通到网络的,即每个节点都有一跳邻居,且所有节点的传输范围和干扰范围都相等且为 R 。对任一节点 i 发送数据给节点 j ,如果要确保发送成功则此时节点 i 的一跳邻居不能进行发送,目的节点 j 的一跳邻居也不能进行发送,

节点 i 和节点 j 的传输范围之和为源节点 i 的干扰范围，即 $A+B+C$ ，其中 B 为节点 i 和节点 j 的传输范围的公共部分，如图 3-1 所示。

假设 A 区内节点个数为 n_i ， B 区节点个数为 n_j ， C 区节点个数为 n_j ，则对节点 i 发送产生冲突的节点总数为： $n_i + n_j + n_j$ 。如果源节点一跳范围内的节点进行发送，即 A 区和 B 区内的节点发送，则节点 i 将检测到信道忙，影响信道检测的节点总数为： $n_i + n_j$ 。

为了准确地估计节点的吞吐率就必须得到源节点和目的节点的邻居信息，IEEE 802.15.5 草案中的 hello 过程完成了网络中节点信息的收集工作，我们可以从最后形成的邻居列表中很容易得到上述信息。

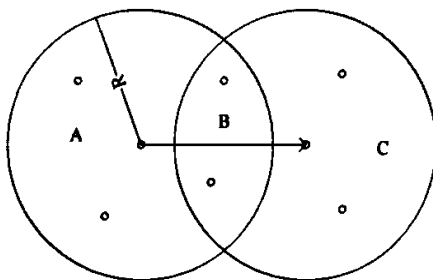


图 3-1 节点发送模型

3.2.2 无线 Mesh 网络模型

Mesh 网络中节点采用 IEEE 802.15.4 标准中规定的非时隙 CSMA/CA 协议。在网络中，每个节点不仅自己产生数据包，也要接收其它节点发送的数据包并进行转发，为了在饱和情况下形成多跳网络，对标准协议进行了简单扩展，即节点在退避期间打开接收机，如果在退避时隙内收到数据将暂时停止退避计数器并对接收的数据包进行处理，即所谓的接收优先方式，同时为了确保应答帧不被干扰，数据帧接收后要等待一段时间再重新启动退避计数器，等待时间为短帧间间隔加上发送应答帧所需的时间。

节点采用固定路由发送数据包，所有的数据都通过多跳发送到网关节点。节点形成树型结构，每个节点唯一的由其邻居节点和其发送概率确定。假设每个节点用来存放待发送数据帧的缓冲区无限大，这样接收的数据帧不会被丢弃。当且仅当干扰范围内没有其它节点发送数据时认为发送成功，否则发生冲突。发送数据后需要等待应答帧的到来，在标准规定的时间内如果收到发送帧的应答则数据传输成功，如果在规定时间内没有收到应答帧或者收到的不是发送帧的应答则认为产生了冲突，数据传输失败，等待应答过程中所有其它帧将被丢弃。

3.2.3 非时隙 CSMA/CA 分析模型

本节将对 IEEE 802.15.4 标准中规定的非时隙 CSMA/CA 协议进行详细介绍。

节点有数据包要发送时，首先进行退避，退避时间为 $[0, CW_0-1]$ 之间的随机数。和单

跳网络不同的是,在 mesh 网络中节点在退避期间保持接收机为打开状态,如果有数据包到达则先暂停退避计数器,然后对数据包进行处理,处理完数据包后还要等待一段时间以确保应答帧不被干扰地发送。

退避时间到后节点进行信道状态检测,如果在规定的时间内信道一直为空闲则节点马上进行发送,否则进入到下一个退避节点,退避窗口加倍。数据包发送后节点要等待应答帧的到来,如果在规定时间内收到发送的数据包的应答则认为发送成功;如果在这段时间内没有收到相应的应答帧则认为发送失败,节点进入重传阶段,等待应答帧期间收到的所有其他帧都将被丢弃。

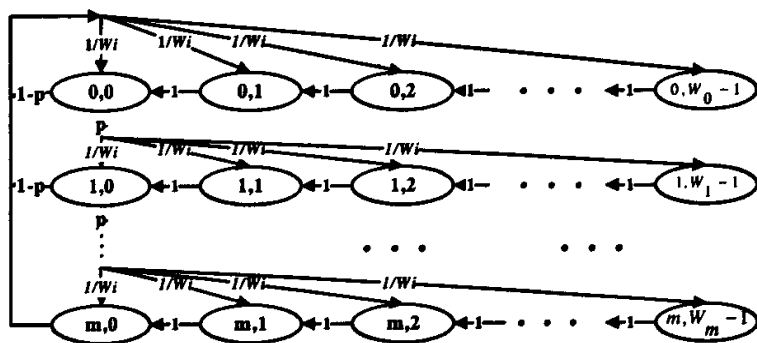


图 3-2 802.15.4 CSMA/CA 退避过程 Markov 模型

如图 3-2 所示, W_i 为当前退避计数最大值, m 为当前退避次数, $1-p$ 为检测到信道忙的概率, (i, j) 为退避状态。当退避计数值 j 为 0 时, 设备进行 CCA, 节点在退避阶段收到包后暂停退避计数器并处理包的过程没有在图中表示。

假设总共 $m+1$ 次退避为一个完整的退避周期, 我们将一个周期为单位分析节点的吞吐量。通过上面的 Markov 模型我们可以看到, 每次节点的退避计数器为 0 时, 节点可能会转移到两种状态: 如果 CCA 过程检测到信道忙, 则跳转到下一个退避状态; 否则进入到发送状态, 它们的概率由检测到信道忙的概率决定。并且进入发送状态后有两种结果: 发送成功和发送冲突, 这取决于节点干扰范围内的邻居节点是否进行发送, 这又和邻居节点的发送概率有关。对一个完整退避周期分析, 我们得到节点吞吐率的计算方法:

$$S = \frac{\text{avg_payload_transmitting_time}}{\text{avg_bk_time} + \text{avg_pkt_handle_time} + \text{avg_pkt_transmitting_time}} \quad (3-1)$$

其中 $\text{avg_payload_transmitting_time}$ 为节点在完整退避周期内平均成功发送负载所用时间, avg_bk_time 为节点在完整退避周期内平均退避时间, $\text{avg_pkt_handle_time}$ 为节点在完整退避周期内平均包处理时间, 这个时间包括帧间间隔和发送 ACK 所用时间, 而 $\text{avg_pkt_transmitting_time}$ 为节点在完整退避周期内平均发包时间, 这个时间为成功发送和发送冲突时间的均值, 包括帧发送时间, 帧间间隔和发送 ACK 的时间。

通过上面的分析我们可以知道, 影响节点吞吐率的因素有: 节点侦听到信道忙的概率,

退避期间节点收到的包个数以及节点的发送概率。其中节点侦听到信道忙的概率和网络拓扑密切相关，而退避期间节点收到的包个数也和节点的一跳邻居数日相关。假设节点 i 的发送概率为 τ_i ，则节点侦听到信道忙的概率的计算公式如下：

$$P_{busy} = 1 - \prod_{i=1}^{ns} (1 - \tau_i) \quad (3-2)$$

其中 ns 为节点的一跳邻居个数，下面我们将对影响节点吞吐率的各个因素进行分析。

3.2.3.1 节点平均成功发送概率

前面我们已经说明节点成功发送的条件为同一时刻节点干扰范围内没有其它节点发送，我们将研究在一个完整的退避周期，即最大退避次数内节点的平均成功发送概率。假设每次发送成功的条件概率为 p_{succ} ，则其计算公式为：

$$P_{succ} = \prod_{i=1}^{nc} (1 - \tau_i) \quad (3-3)$$

其中 nc 为源节点竞争范围内节点个数。为了发送要进行 k 次退避的概率为：

$$P_{tr} = (1 - p_{busy}) * p_{busy}^{k-1} \quad (3-4)$$

这样我们可以得到一个完整退避周期内节点成功发送的平均概率为：

$$\overline{p_s} = p_{succ} * \sum_{k=1}^m (1 - p_{busy}) * p_{busy}^{k-1} \quad (3-5)$$

经过简单运算可以得到更明了的关系式：

$$\overline{p_s} = p_{succ} * (1 - p_{busy}^m) \quad (3-6)$$

3.2.3.2 节点平均发送冲突概率

同理我们可以由上面的分析很容易得到节点平均发送冲突概率。假设节点每次发送冲突的概率为 p_{coll} ，则：

$$P_{coll} = 1 - \prod_{i=1}^{nc} (1 - \tau_i) \quad (3-7)$$

从而得到一个完整退避周期内节点发送冲突的平均概率为：

$$\overline{p_c} = p_{coll} * (1 - p_{busy}^m) \quad (3-8)$$

3.2.3.3 节点平均退避时隙数

在每次退避过程中, 节点将选择 $[0, w_i - 1]$ 内的随机个时隙进行退避, 因此, 在一次退避中平均的退避时隙数为:

$$b_i = \frac{1}{2} * (w_i - 1) \quad (3-9)$$

退避次数为 k 时节点的平均退避时隙数为:

$$\bar{b}_k = \sum_{i=1}^k \frac{1}{2} * (w_i - 1) \quad (3-10)$$

再由前面得到的要进行 k 次退避的概率为:

$$p_{ir} = (1 - p_{busy}) * p_{busy}^{k-1} \quad (3-11)$$

由此我们可以得到一个完整的退避周期内节点平均退避时隙数:

$$E(bk_slot) = \sum_{k=1}^{m-1} (1 - p_{busy}) * p_{busy}^{k-1} * \bar{b}_k + p_{busy}^{m-1} * \bar{b}_m \quad (3-12)$$

3.2.3.4 节点退避期间平均接收包数

为了计算节点吞吐率必须先得到节点在一个完整退避周期内在退避期间收到的平均包个数, 从而得到处理这些包所用的时间。假设退避期间每收到一个包, 为了使应答帧顺利被发送, 节点需要等待一定时间 t_f , 包括帧间间隔和发送 ACK 的时间:

$$t_f = SIFS + t_{ack} \quad (3-13)$$

由于节点在退避期间接收到的数据包个数和节点的一跳邻居数目 ns 有关, 因此它应该是 ns 的函数且以比随 ns 线形增长更快的速度增加, 因为节点可能因为一跳邻居的邻居的发送而退避, 另外它随着退避次数的增加而减小, 因为随着节点退避次数的增加它有更大的可能获得信道的使用权^[16], 因此我们使用下式来估计节点在第 k 次退避期间收到的包数目:

$$A_k = ns * \log ns / k \quad (3-14)$$

由此我们可以得到一个完整的退避周期内节点退避期间平均收到的包个数:

$$\bar{A}_k = \sum_{i=1}^{m-1} (1 - p_{busy}) * p_{busy}^{i-1} * A_i + p_{busy}^{m-1} * A_m \quad (3-15)$$

3.2.3.5 节点吞吐率模型

通过以上分析我们可以得到节点吞吐率的计算公式，假设节点发送每个包的负载所用的时间为 t_p ，成功发送一个包所用时间为 t_s ，发送产生冲突所用时间为 t_c ，退避期间接收到包后等待时间为 t_f ，每个退避时隙长度为 t_{slot} ，我们得到以下节点吞吐率计算公式：

$$S = \frac{t_p * \overline{p_s}}{E(bk_slot) * t_{slot} + \overline{A_k} * t_f + \overline{p_s} * t_s + \overline{p_c} * t_c} \quad (3-16)$$

式中各个参数的计算式如下：

$$E(bk_slot) = \sum_{k=1}^{m-1} (1 - p_{busy}) * p_{busy}^{k-1} * \overline{b_k} + p_{busy}^{m-1} * \overline{b_m} \quad (3-17)$$

$$\overline{A_k} = \sum_{i=1}^{m-1} (1 - p_{busy}) * p_{busy}^{i-1} * A_i + p_{busy}^{m-1} * A_m \quad (3-18)$$

$$\overline{p_s} = p_{succ} * (1 - p_{busy}^m) \quad (3-19)$$

$$\overline{p_c} = p_{coll} * (1 - p_{busy}^m) \quad (3-20)$$

$$t_f = SIFS + t_{ack} \quad (3-21)$$

$$t_s = t_p + SIFS + t_{ack} \quad (3-22)$$

$$t_c = t_p + t_{ack_wait_duration} \quad (3-23)$$

$t_{ack_wait_duration}$ 为标准规定的最大 ACK 帧等待时间，节点在发送完帧后一直等待直到收到正确的 ACK 帧或者超时。由于 p_{busy} ， $\overline{p_s}$ ， $\overline{p_c}$ 均和节点及其邻居节点的发送概率相关，而在 mesh 网络中各个节点的发送概率由于节点间的复杂依赖关系而难以确定。因此我们假设节点在任意时刻进行成功发送的绝对概率为 $p_{i \rightarrow j}$ ，且有：

$$p_{i \rightarrow j} = \tau_i * \prod_{k=1}^{N_c} (1 - \tau_k) \quad (3-24)$$

这里我们假设节点干扰范围内所有节点的发送概率都相等且为 τ ，即式中 $\tau_i = \tau$ ，使

$p_{i \rightarrow j}$ 最大我们得到：

$$\tau = 1/(nc+1) \quad (3-25)$$

我们将使用这个值来估计每个节点的发送概率，这样根据我们从节点的邻居列表里获得的节点邻居信息我们可以计算出上述所有参数从而得到节点吞吐率的估计值。值得注意的是由于 mesh 网络中节点不光要发送自己产生的包，还要接收其它节点发送的包并进行转发，而且网络拓扑复杂，节点之间存在很大的依赖关系，我们的模型中对节点发送概率和退避期间节点平均接收到的包个数不会是很精确的，但是这样的估计符合一定的实际情况，而且这样的估计值很容易获得，因此我们的模型具有现实意义。

3.3 分析模型验证和仿真结果

3.3.1 仿真环境和参数

为了验证本文前面提出的基于 IEEE 802.15.4 标准和 IEEE 802.15.5 草案的低速 mesh 网络的节点吞吐率模型的可行性和准确性，本节将对理论分析的结果和相应的仿真结果进行对比。本文使用 OPNET 进行仿真，根据标准规定的参数建立了 MAC 层和网络层仿真模型。

我们的仿真环境如下：

- ✓ 节点随机分布在一个 50m×50m 的矩形范围内；
- ✓ 网络中，所有的发送节点都是相似的，它们之间可以互相发送数据帧；
- ✓ 网络是对等的多跳网络，即网络中的每个节点都可以收到其相邻节点发送的数据；
- ✓ 网络中有 9 个普通节点，1 个网关节点；
- ✓ 所有的数据通过多跳发送到网关节点，网关节点只接收数据；
- ✓ 所有节点按照 IEEE 802.15.5 草案规定的方式形成树型的 mesh 结构网络；
- ✓ 所有的通信只能通过一个信道完成；
- ✓ 信道误比特率为零。所有的帧丢失都是由发生冲突或缓存溢出引起。

为了仿真饱和状态状态下节点的吞吐率，我们不采用使用较高的包到达率的方法，同时为了和非饱和时的模型保持一致，我们对空闲状态作了特别处理，节点一进入空闲状态马上跳到退避状态进行下一次发送，每次节点发送完一个包并等待接收 ACK 帧后根据发送成功与否分别进入到重发和下一次全新发送状态。另外为了在 CCA 期间准确感知信道的状态，我们在规定的时间内每隔一个最小时隙就收集一次信道的状态，如果期间信道一直空闲就认为信道空闲，否则一感知到信道忙就立即跳出该状态进入到下一次退避过程。所有的仿真参数都严格按照标准规定，为了更方便的分析结果，节点发送的包长固定。表格 3-1 列出了部分重要的仿真参数。

表格 3-1 节点吞吐率分析模型仿真参数

帧负载长度	512bits
应答帧长度	13 bytes
信道比特率	250 kbps
传播时延	0
时隙时间	4 us
符号时间	4*slot
短帧间间隔	12*symbol
单位退避时间	20*symbol
应答等待持续时间	54*symbol
最大退避次数	3
最小退避指数	3
最大退避指数	5

3.3.2 仿真结果及性能分析

如图 3-3 所示,本章建立的 mesh 网络中节点的饱和吞吐率分析模型较准确地估计了网络的性能。从图中可以看到,每个节点实际获得的饱和吞吐率和分析结果趋势一致,但是有部分节点相差较大,这是因为我们的分析模型中对节点的发送概率和退避期间节点平均收到的包个数进行了先验估计,一般情况下这个估计方法能较好地接近实际情况,但是由于多跳网络的复杂性,在某些情况下存在较大的偏差。

从图中我们也可以看到,网络中不同位置的节点获得的吞吐率有很大的差别,相差数倍。因为节点采用固定路由传输数据,靠近网络处节点不仅要发送自身生成的数据,还要转发其它节点的数据,成为业务的“热区”,例如节点 2, 4, 6。仿真结果显示节点 6 实际获得的吞吐率比它的子节点 8 获得的吞吐率小得多,这样就会造成节点 8 发送的数据在节点 6 处被大量丢弃,严重浪费了网络资源,给网络性能带来恶劣影响。下一章我们将考虑这种情形,并对标准的 CSMA/CA 算法进行改进,以提高节点 6 的吞吐率同时减小节点 8 的吞吐率,这样可以提高节点的负荷公平性,提升网络性能。

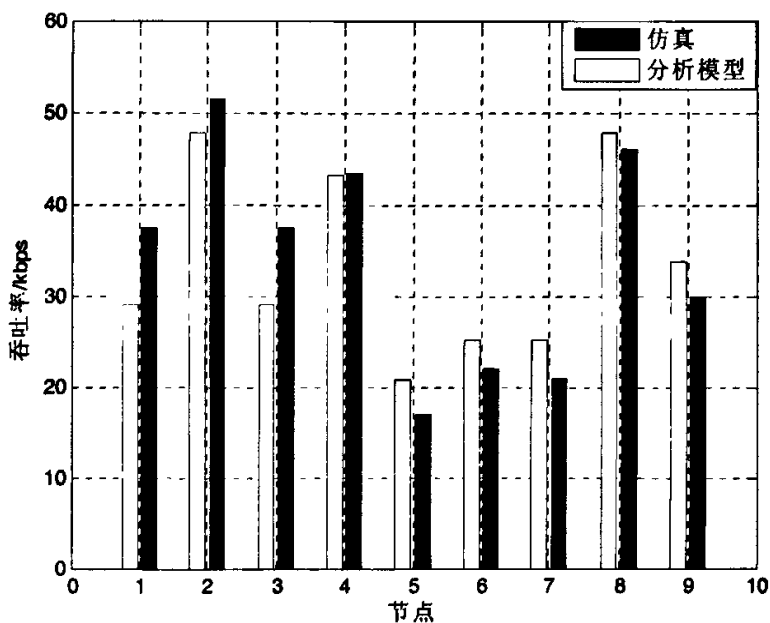


图 3-3 节点吞吐率：理论分析和仿真结果

3.4 本章小节

本章建立的基于 IEEE 802.15.4 标准和 IEEE 802.15.5 草案的低速 mesh 网络的节点吞吐率分析模型比较准确地估计了网络中每个特定节点的吞吐率，主要创新和贡献如下：

- 1) 建立了 mesh 网络中节点吞吐率的分析模型，并通过仿真验证了其有效性和准确性。
- 2) 根据通过 IEEE 802.15.5 草案中 hello 过程获得的邻居节点信息，估计了节点发送概率和退避期间节点平均接收的包数目。
- 3) 本分析模型对 IEEE 802.15.4 标准规定的 CSMA/CA 协议进行了部分扩展，为了使其更适合多跳网络提出节点在退避过程中可以接收其它节点发送的包，即接收优先。

第4章 低速无线 mesh 网络公平性研究

4.1 引言

网络的公平性是无线网络的重要特性之一^[25]，在网络层和 MAC 层中，网络的公平性包括节点间的公平性和节点内部不同数据流的公平性。节点间的公平性主要指网络中的每个节点，不论网络拓扑如何，也不论节点所处的位置如何，要有相似的获得信道使用权和成功发送数据的能力。而节点内部不同数据流的公平性是指在多跳网络中每个节点不仅要发送自己产生的数据，还要转发收到的其它节点发送的数据，一般情况下节点的自身数据到达速率较高且节点的存储空间有限，如果简单地对自身数据流和转发数据流作相同处理，且在队列满时丢弃新到达的数据会造成大量转发的数据被丢弃，尤其是节点负荷较大时，从而影响远离网关的节点吞吐率。对节点间的公平性一般通过网络层的路由选择算法使网络负载尽量均衡，而对节点间的公平性则通过节点内部的数据队列管理实现。当然也可以通过网络层和 MAC 层的跨层协作同时实现节点间和节点内部的公平性。

无线 mesh 网络是分布式系统，各节点负荷的业务量，节点面临的冲突情况都不一样，因此各节点的发送能力也不一样。通常，在靠近网关的地方，由于节点要发送自身数据和转发数据，从而形成地理上的业务“热区”，同时由于节点共享传输信道，使得网关附近冲突加剧，在网关附近的节点发送能力也受到限制，成为网络瓶颈。而在网络的边缘，由于节点需要转发的数据流较少，形成业务“冷区”，在这些地方，节点可以比较容易获得无线资源来传输数据。但是远端节点发送来的数据往往在这些瓶颈处被丢弃，这些被丢弃的数据包已经参与了网络无线资源的竞争并耗费了网络传输资源，造成了系统资源的浪费。

IEEE802.11 系列协议提供了很好的 MAC 层公平性，然而不能确保网络层的公平性，同样可以认为与之相似的 IEEE802.15.4 标准也提供了很好的 MAC 层公平性。CSMA/CA 算法中退避窗口的大小表明了节点获得信道使用权的能力^[26]，为了提高网络性能，可以动态调整节点的退避窗口^[27,28,29,30]。目前提高网络中节点公平性的主要方法有：最大-最小策略^[31]，比例公平策略^[32]等，本章采用的方法属于比例公平策略的一种。本章对 mesh 网络中节点公平性的研究就是通过以下方法实现的：第一次发送时按照负荷大小给每个节点设置一个权值，并按照权值设置节点的初始退避窗口；以后每次发送时动态估计节点自身获得的吞吐率以及网络中其它竞争节点获得的吞吐率，用吞吐率和负荷之比作为节点的公平性的度量，如果这个度量值超过规定的门限则动态地调整节点的初始竞争窗口，使其获得较好的流量比例公平。仿真结果表明了本章算法的有效性，不仅显著提高了节点的流量公平性，同时节点的丢包率和非饱和时的吞吐率也得到了很大的改善。

下面将对基于流量比例公平的动态退避窗口调整算法作详细介绍。

4.2 基于流量比例公平的动态退避窗口调整算法模型

4.2.1 初始退避窗口设置算法

如前文所述, mesh 网络中存在节点流量不均匀的情况, 靠近网关的节点相对于网络边缘的节点要承担更多的转发数据流。

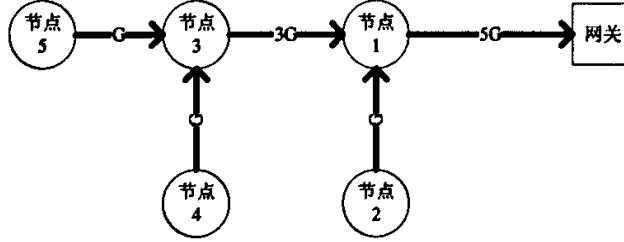


图 4-1 mesh 网络中节点流量示意图

因为采用 CSMA/CA 接入算法, 节点竞争到信道的能力近似和节点的退避窗口大小成正比, 如果对所有节点采用相同的初始退避窗口则节点近似公平地获得接入信道的能力。然而在 mesh 网络中, 数据通过多跳传输到网关节点, 靠近网关的节点不仅要发送自身的数据, 还要转发别的节点的数据, 这样同一个竞争区域里节点的负荷就不相同, 形成业务的“冷区”和“热区”。如图 4-1 所示, 假设每个节点的自身数据量为 G , 则节点 1 的负荷为 $5G$, 节点 2, 3, 4, 5 的负荷分别为 $1G$, $3G$, $1G$ 和 $1G$, 如果节点的初始退避窗口相同, 则可以认为它们的 MAC 层吞吐率之比为 $1: 1: 1: 1: 1$, 网络层要求的吞吐率之比为 $5: 1: 3: 1: 1$, 当网络能够承受这些容量时, 可以满足节点的发送要求, 但是当节点负荷超出网络容量时, 负荷大的节点由于不能得到更多的发送机会, 导致包的大量丢弃, 成为网络的瓶颈, 从而影响网络整体的吞吐率。如图中所示, 节点 1 和节点 2 公平的获得信道的使用权, 网络的吞吐率被节点 2 “拉低”。

在非饱和时, 理想状况下节点 1, 2, 3, 4, 5 取得的吞吐率分别为: $5G$, $1G$, $3G$, $1G$ 和 $1G$, 因此我们的目标就是让节点获得的吞吐率和其负荷成正比。使用 CSMA/CA 算法, 节点获得信道使用的能力近似和节点的初始退避窗口成正比, 因此我们将通过为不同负荷的节点设置不同的初始退避窗口来达到上述目标。

为了方便起见, 我们考虑节点饱和时的状况, 下面我们通过分析节点饱和状态下的 Markov 模型获得所需的节点初始退避窗口。

假设节点在 CCA 阶段检测到信道忙的概率为 p , 最大退避次数为 NB , 对退避状态 $\{s(t), b(t)\}$, 其中 $s(t)$ 为节点退避次数, $b(t)$ 为退避计数器值根据图 3.2 我们可以得到以下的一步转移概率。

$$p\{i, j | i-1, 0\} = \begin{cases} p/w_i, i \in [1, m], j \in [0, w_i - 1] \\ p/w_m, i \in [m+1, NB], j \in [0, w_m - 1] \end{cases} \quad (4-1)$$

$$p\{0, j|i, 0\} = \begin{cases} (1-p)/w_0, i \in [0, NB-1], j \in [0, w_0-1] \\ 1/w_0, i = NB, j \in [0, w_0-1] \end{cases} \quad (4-2)$$

假设稳态概率为 $b_{i,j} = \lim_{t \rightarrow \infty} p\{s(t) = i, b(t) = j\}$, $i \in [0, NB]$, $j \in [0, w_i-1]$, 我们很容易从 Markov 模型中得到:

$$b_{i,j} = \frac{w_i - j}{w_i} \begin{cases} (1-p) * \sum_{j=0}^{NB} b_{j,0} + p * b_{NB,0}, i = 0 \\ p * b_{i-1,0}, i \in [1, NB] \end{cases} \quad (4-3)$$

又有 $b_{i-1,0} * p = b_{i,0}$, $i \in [1, NB]$, 可得 $b_{i,0} = p^i * b_{0,0}$, $i \in [1, NB]$, 利用此式我们可以得

到 $b_{i,j} = \frac{w_i - j}{w_i} * b_{i,0}$, $i \in [1, NB]$, $j \in [0, w_i-1]$. 由于节点在饱和状态时在上述状态间循环

跳转, 因此这些状态的概率之和应该为 1, 即有:

$$\begin{aligned} 1 &= \sum_{i=0}^{NB} \sum_{k=0}^{w_i-1} b_{i,k} = \sum_{i=0}^{NB} b_{i,0} \sum_{k=0}^{w_i-1} \frac{w_i - k}{w_i} = \sum_{i=0}^{NB} b_{i,0} * \frac{w_i + 1}{2} \\ &= \sum_{i=0}^m b_{i,0} * \frac{w_i + 1}{2} + \sum_{i=m+1}^{NB} b_{i,0} * \frac{w_m + 1}{2} \\ &= \frac{b_{0,0}}{2} \left[w_0 * \left(\sum_{i=0}^m 2p^i + 2^m * \sum_{i=m+1}^{NB} p^i \right) + \frac{1 - p^{NB+1}}{1 - p} \right] \end{aligned} \quad (4-4)$$

为了简化取 $NB = m$, 这样从上式可以得到:

$$b_{0,0} = \frac{2 * (1-p) * (1-2p)}{(1-p) * (1-2p^{m+1}) * w_0 + (1-2p) * (1-p^{m+1})} \quad (4-5)$$

由于只有在退避计数器值为 0 且检测到信道为空闲时节点才会发送, 得到在任意时隙的发送概率:

$$\tau = (1-p) * \sum_{i=0}^m b_{i,0} = \frac{2 * (1-p) * (1-2p) * (1-p^{m+1})}{(1-p) * (1-2p^{m+1}) * w_0 + (1-2p) * (1-p^{m+1})} \quad (4-6)$$

为了让节点获得与负荷大小成正比的接入信道的能力, 我们取节点在任意时隙的最优发送概率为:

$$\tau_i^* = \frac{load_i}{\sum_{j=1}^{NC} load_j + load_i} \quad (4-7)$$

式中 NC 为节点干扰范围内其它节点的个数, $load_i$ 为节点 i 的负荷。 NC 和 $load_i$ 的值可以由网络层的 hello 过程和路由过程获得, 在此不作描述。令节点发送概率 $\tau_i = \tau_i^*$ 可以得到初始退避窗口 w_0 和发送概率及检测到信道忙概率 p 的关系。

$$w_0 = \frac{2 * (1-p) * (1-2p) * (1-p^{m+1}) - \tau^* * (1-2p) * (1-p^{m+1})}{(1-p) * (1-2p^{m+1})} \quad (4-8)$$

为了确定 w_0 的值我们还要知道每个节点感知信道忙的概率 p_i , 这可以通过下式计算:

$$p_i = 1 - \prod_{j=1}^{NC_i} (1 - \tau_j) \quad (4-9)$$

其中 NC_i 为节点 i 的一跳邻居个数, 这同样可以在 hello 过程中得到。由上述分析, 我们可以得到基于节点负荷公平的退避窗口初始值。

4.2.2 退避窗口动态调整算法

设置初始退避窗口后, 节点不能很精确地按负荷比例获得吞吐率, 因此我们必须对初始窗口值进行动态调整。

假设节点获得的吞吐率为 W_i , 节点的负荷为 G_i , 节点干扰范围内其它节点获得的吞吐率之和为 W_o , 它们的负荷之和为 G_o , 为了实现吞吐率的负荷比例公平, 必须满足以下条件:

$$PF_{index} = \frac{W_i/G_i}{W_o/G_o} \quad (4-10)$$

假设每个节点自身产生的数据流量相等, 节点的总负荷 G_i 我们可以从路由过程得到。如果采用树型路由, 即节点向其父节点发送数据, 那么节点的负荷就等于其自身数据负荷加上所有子节点的负荷之和。如果采用其它的路由算法, 可以在帧中携带负荷信息, 值得注意的是本文提出的方法仅限于节点使用固定路由的网络。为了计算比例公平指数 PF_{index} 还需要知道节点和干扰范围内其它节点实际获得的吞吐率, 下面我们介绍对它们进行估计的具体方法。

由于节点的吞吐率是由它成功发送的数据包个数决定的, 而这和节点实际发送的次数密切相关, 因此我们用节点实际发送包的次数来表示节点实际获得的吞吐率。具体的吞吐率估计算法如下:

if (节点 i 发送一个数据包)

W_i++ ;

if (节点 i 接收到一个数据包)

$W_o++;$

并对 W_i 和 W_o 在时间上不断累积, 我们的目的是通过动态调整节点的初始竞争窗口值使流量比例公平指数在时间轴上趋近于门限。窗口调整算法是一个分段函数, 对于比例公平指数超过门限的情况, 我们需要增大节点退避窗口; 而对于比例公平指数小于门限的情况, 我们需要减小退避窗口; 对门限范围内的情况不需要进行调整。这里所说的门限是一个范围, 假设上限值为 $threshold_{up}$, 那么下限为其倒数, 即:

$$threshold_{down} = 1 / threshold_{up}$$

具体调整算法如下:

```
switch(PFindex) {
case > thresholdup :
    initialCW = min(CWmax, CWorg * ad_increase);
case [thresholddown, thresholdup]:
    initialCW = CWorg;
case < thresholddown :
    initialCW = max(CWmin, CWorg * ad_decrease);
}
```

$ad_increase$ 和 $ad_decrease$ 的取值关系到窗口调整的速度和精度, 如果取值很小, 那么调整的精度较高但是速度较慢; 如果取值太大则调整的速度很快但是不能达到很高的精度。为了使调整的步伐适应公平性指数的变化, $ad_increase$ 和 $ad_decrease$ 的取值分别为:

$$ad_increase = 1.0 + PF_{index} / 10.0 \quad (4-11)$$

$$ad_decrease = 0.8 \quad (4-12)$$

通过仿真表明参考门限的取值为 $threshold_{up}=1.1$ 时可以取得比较好的效果。结合上述的初始退避窗口设置和调整算法, 通过先验地根据节点的负荷为其设置退避窗口初始值并实时估计节点实际获得的吞吐率从而调整退避窗口值, 节点的流量比例公平性得到了很大的提高, 另外节点的丢包率和非饱和状态下的吞吐率也得到了较大提高。相对与 IEEE 802.15.4 标准中规定的 CSMA/CA 算法, 本章提出了动态窗口调整算法在 mesh 网络中提供了更好的性能, 且该算法计算复杂度低, 对原有的标准算法改动很小, 可以适用于采用随机接入机制的低速多跳网络。

4.3 节点公平性评价准则

为了对比改进算法和标准算法性能的优劣, 我们采用了以下两种准则对它们进行评估。

4.3.1 评价准则 1

对网络中任意节点 i 实际获得的吞吐率为 W_i ，节点的负荷为 G_i ，那么我们很容易想到公平性指数 FI 为^[33]：

$$FI = \max \left\{ \forall i, j : \max \left(\frac{W_i}{G_i}, \frac{W_j}{G_j} \right) / \min \left(\frac{W_i}{G_i}, \frac{W_j}{G_j} \right) \right\} \quad (4-13)$$

很直观的， FI 的取值越接近于 1 说明网络中节点的流量公平性越好。但是当 FI 的取值远大于 1 时我们不能很好地判断几种算法性能的优劣。考虑到存在这样的情况，即网络中有几个节点的吞吐率和负荷之比差别很大，但是从总体来看它们的差距较小。造成这种现象的原因可能为个别节点所处的环境相差很大，例如节点 i 干扰范围内的竞争节点很少而且它只需发送自身产生的数据，这样它的吞吐率和负荷之比就比较大；而节点 j 干扰范围内竞争节点很多且负荷较大，另外节点 j 和其目的节点之间的通信链路质量很差，这样就导致吞吐率和负荷比很小。如果我们的算法强行要求这样的节点 i 和 j 有相近的吞吐率负荷比显然是不合理的，这样将导致全网的性能受到“拖累”而大幅下降，因此我们提出另一种评价准则。

4.3.2 评价准则 2

为了避免准则 1 的缺陷，本章提出了一种新的评判节点流量公平性的准则。该准则通过对每个节点吞吐率负荷比和它们的均值进行差方并累加以避免出现准则 1 中的情况，具体的准则如下：

$$FI = \frac{\sum_{i=1}^N (W_i/G_i - \overline{W_i/G_i})^2}{\sum_{i=1}^N (W_i/G_i)^2} \quad (4-14)$$

式中 N 为网络中除网关外节点的数目。节点的吞吐率负荷比与均值之差表明了节点偏离的程度，而再对所有节点的这个值进行累加并归一化则抵消了不同算法的节点平均吞吐率不同的影响。这种评价准则将网络中所有节点的吞吐率负荷比都考虑进去，抵消了网络中少数“特殊”节点吞吐率负荷比差距很大的影响，因而较准则 1 更具有意义。

为了表明本章提出的算法较标准算法在节点的流量公平性方面有很大提高，下面我们将分别利用上述两种准则对仿真结果进行比较，结果表明不论采用哪种评价准则，改进算法都取得了很好的效果。

4.4 动态退避窗口调整算法性能和仿真结果对比

4.4.1 仿真环境和参数

为了验证本文提出的基于流量比例公平的 CSMA/CA 改进算法的有效性,本章基于 OPNET 的仿真软件建立了动态窗口调整算法的仿真模型。我们采取了第 3.3.1 小节的仿真环境和参数以便和标准算法的性能比较。对每个节点采用相同的包生成速率,它们都服从泊松分布。为了仿真不同包到达率下算法的性能,本章的仿真模型和第三章对应的模型有些不同,节点开始进入空闲状态,如果在这个状态里接收到其它节点发送的包或者收到上层的包则待发送的包个数加 1 且进入退避状态,发送完一个包后无条件地重新返回空闲状态并判断待发送的包个数是否为 0,如果不是则进行新的发送,否则节点将一直处于空闲状态,直到收到下一个要求发送的包。

4.4.2 仿真结果

本节将从系统吞吐率,丢包率和节点流量公平性方面对改进算法和标准算法的仿真结果进行分析比较。

4.4.2.1 系统吞吐率

在 mesh 网络中,所有的数据都通过多跳发送到网关节点,因此系统的吞吐率即为网关节点的吞吐率^[34]。

图 4-2 是在不同包生成速率下本章提出的改进算法和标准算法所获得的网关节点的吞吐率。如图 4-2 所示,当包生成速率很小时,例如包生成速率服从参数为 $\lambda=0.1$ 的泊松分布时,网络很空闲,因此对两种算法都能取得要求的吞吐率,即基本上所有包都能成功发送;当包生成速率逐渐增大且网络没有达到饱和时,改进算法体现了更好的性能。因为虽然使用改进算法时节点的初始退避窗口要大于使用标准算法时的窗口值,但是因为改进算法中每个节点的退避窗口值不一样,负荷小的节点有较大的窗口,而负荷较大的节点相对有较小的窗口,这样它们的发送概率就相应不一样,导致节点的碰撞概率较小,因此反而取得了较大的系统吞吐率;当节点趋近于饱和时,标准算法获得了更好的性能,这是因为采用标准算法节点的退避时间相对较短,因此实际的包发送速率更快。

从图中也可以看到,随着包生成速率的增大,系统吞吐率呈非线性增长,包生成速率越大,系统吞吐率增加的越小,当网络饱和后,吞吐率不再增加,趋近于一个极限值。这是因为随着包生成速率的增加,节点之间的竞争也变得更激烈,这样它们需要退避更长的时间才能竞争到信道,而且由于隐藏终端的影响,节点发送产生冲突的概率也随着增加。比较有意思的是仿真结果结果表明,系统吞吐率取最大值的时候并不是网络饱和后,而是在网络达到饱和前的一个临界状态,这和文献[35]中通过仿真得到的结果相同,这也从另

一方面说明我们仿真模型的正确性。

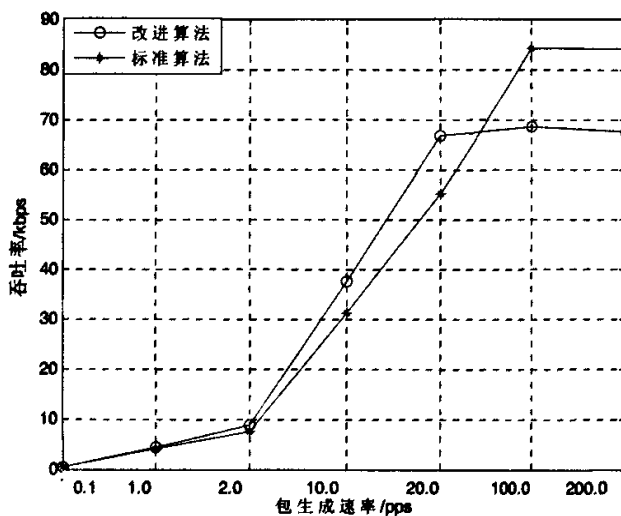


图 4-2 改进算法和标准算法的系统吞吐量

4.4.2.2 系统丢包率

IEEE 802.15.4 标准规定，如果节点在发送包后在规定的最大等待时间内没有收到相应的 ACK 帧，则认为发送失败，这时候需要重传，同时标准规定最大重传次数为 3，即如果重传次数达到规定值后该包将被丢弃。

本节说明改进算法在提高非饱和时系统吞吐率的同时极大地降低了系统的丢包率。这里丢包率定义为：节点在单位时间内丢弃的包个数。

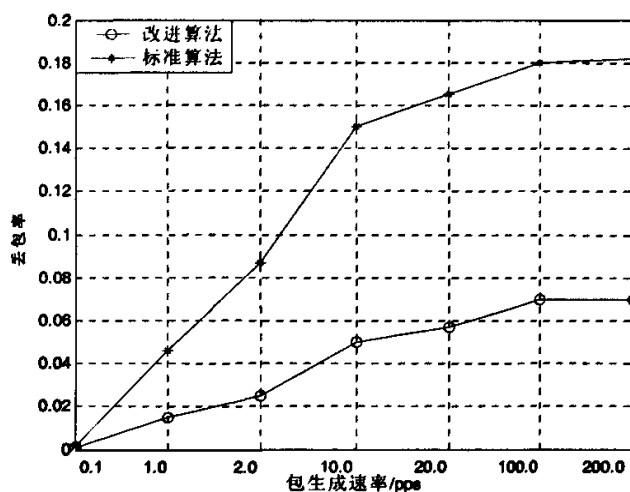


图 4-3 改进算法和标准算法的系统丢包率

如图 4-3 所示, 当节点的包生成速率较小时, 系统能够满足节点的发送要求, 所有的负荷都能成功得到发送, 这时的系统丢包率也相对很小; 随着包生成速率的增大, 网络中节点间的竞争也变得更激烈, 因此系统丢包率也随之变大; 随着系统趋向饱和, 系统丢包率也趋向于稳定。另外, 在节点包生成速率很小时, 系统丢包率的增长超过包生成速率的增长; 但是当节点包生成速率达到一定值后, 系统丢包率逐渐趋缓, 最终趋近一个固定值。

4.4.2.3 节点每流量吞吐率

为了考察网络中节点间的流量公平性, 本节将讨论节点获得的每流量吞吐率。节点的每流量吞吐率的定义为: W_i/G_i , W_i 为节点的总吞吐率, G_i 为节点的负荷。

如表格 4-1, 表格 4-2 所示, 节点的包生成速率很低时, 网络中的每个节点都能获得要求的吞吐率, 此时改进算法和标准算法的性能没有明显的差距; 当节点的包生成速率逐渐增大时, 随着信道竞争变激烈, 碰撞也随之增多, 采用改进算法节点取得了更大的吞吐率, 这也是因为不同负荷的节点采用不同的初始退避窗口, 它们侦听到信道忙的概率更小且发送产生碰撞的概率也要小; 节点趋于饱和时, 采用标准算法网络中不同位置的节点获得的吞吐率差距变大, 因为在多跳网络中, 吞吐率和公平性是两个矛盾的目标^[36], 改进算法通过牺牲吞吐率来增强节点的流量公平性。

表格 4-1 采用标准算法节点获得的吞吐率

包生成速率(pps)/ 节点吞吐率 (kbps)	1	2	3	4	5	6	7	8	9
0.1	0.10	0.10	0.10	0.10	0.10	0.1	0.10	0.10	0.10
1.0	0.91	0.98	0.93	0.95	0.90	0.90	0.95	1.00	1.00
2.0	1.67	1.93	1.79	1.75	1.62	1.70	1.73	1.85	1.85
10.0	6.60	9.10	8.30	8.00	6.75	7.50	8.00	9.30	8.40
20.0	11.00	19.00	17.00	16.00	11.50	15.00	15.20	18.20	16.50
100.0	13.60	46.00	36.20	44.70	9.70	43.20	20.00	61.50	57.00
200.0	13.70	46.00	36.00	44.50	9.70	43.50	20.00	62.50	57.00

表格 4-2 采用改进算法节点获得的吞吐率

包生成速率(pps)/ 节点吞吐率 (kbps)	1	2	3	4	5	6	7	8	9
0.1	0.10	0.10	0.10	0.10	0.10	0.1	0.10	0.10	0.10
1.0	0.97	1.00	0.97	0.95	1.00	1.00	1.00	1.00	1.00
2.0	1.95	2.00	1.90	1.85	2.00	2.00	2.00	2.00	2.00
10.0	8.00	10.00	9.00	8.50	8.50	10.00	9.00	9.70	10.00
20.0	14.00	19.00	18.00	17.50	14.50	18.50	16.00	19.00	20.00
100.0	14.20	25.50	19.00	19.00	14.70	22.00	19.00	39.30	37.70
200.0	14.10	25.50	18.50	18.20	14.70	22.00	18.50	40.50	39.00

4.4.2.4 网络公平性指数

本节将分别采用 4.3 节描述的两个准则对两种算法的公平性进行比较。准则 1 说明了节点获得的吞吐率负荷比的最大值和最小值的比值，这个值越接近于 1 说明节点之间的流量公平性越好；准则 2 说明了网络中所有节点取得的吞吐率负荷比与均值的归一化差距，这个值越小说明节点的流量公平性越好，当这个值为 0 时，网络中任意节点都获得了相同的吞吐率负荷比。

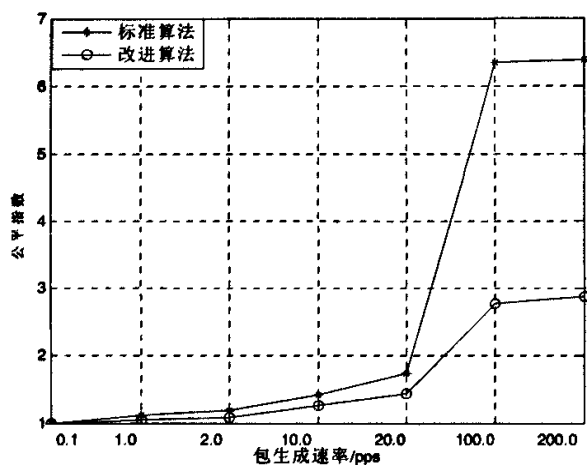


图 4-4 改进算法和标准算法的公平性指数 1

如图 4-4 所示，采用改进算法，利用公平性准则 1 判断的节点流量公平性指数要好于采用标准算法。节点的公平性质数随着包生成速率的增大而逐渐远离 1，这表明公平性变差，当网络饱和时，公平性指数最终趋于一个稳定值。

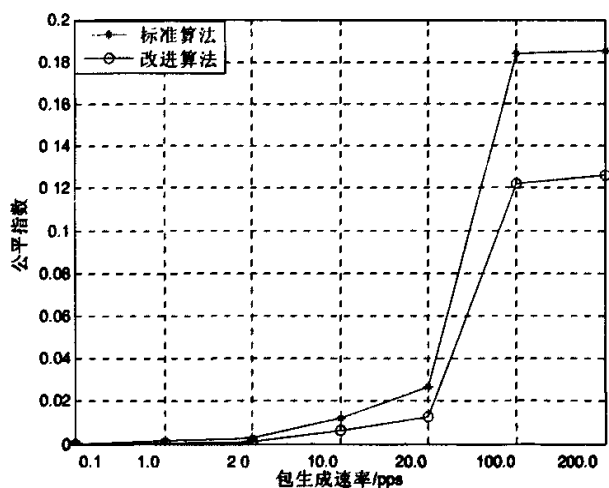


图 4-5 改进算法和标准算法的公平性指数 2

同样,如图 4-5 所示,采用改进算法,利用公平性准则 2 判断的节点流量公平性指数要好于采用标准算法。节点的公平性质数随着包生成速率的增大而逐渐远离 0,这表明公平性变差,当网络饱和时,公平性指数最终趋于一个稳定值。

4.5 本章小节

本章针对 mesh 网络中节点负荷不同的情况基于节点的流量公平性,改进了标准 CSMA/CA 算法,通过为每个节点设置退避窗口值并根据节点实际获得的吞吐率动态调整初始退避窗口,在极大地提高节点流量公平性的同时还大大改善了系统的丢包率和非饱和情况下的节点和系统吞吐率,主要的创新和贡献如下:

- 1) 采用了根据节点的负荷设置第一次发送时节点初始退避窗口值的方案,该方案只需要从网络层获得邻居节点信息,简单可行;
- 2) 以后每次发送时,采用了根据节点实际获得的吞吐率动态调整其初始竞争窗口以提高节点流量公平性的方法,该方法对标准算法改动很小,对流量的估计也简单易行;

按照两种准则对节点的流量公平性进行了评判,结果表明在两种准则下,改进算法相比标准算法都极大地提高了节点的流量公平性。

第5章 无线 Mesh 网络软件实现

5.1 引言

为了对研究的算法进行验证评估,需要搭建实际的实验平台,该实验平台包括硬件部分和软件部分,软件部分要求较高的扩展性以方便加载不同的算法。整个实验系统基于 IEEE802.15.4 标准和 IEEE802.15.5 草案,完成了标准规范的基本框架,硬件平台采用了 TI 公司的 MSP430F149^[37,38,39]单片机和 Chipcon 公司生产的 CC2420^[40]收发芯片。MSP430F149 是一款功耗极小的微处理器,片内集成了 2K 的 EEROM 和 60K 的 flash,这使得在片内实现标准协议成为可能;CC2420 收发芯片则实现了 IEEE802.15.4 物理层的功能,处于 2.4G 无许可证频段,无需授权即可使用,另外该芯片的接口也比较简单,通过 SPI 口与处理器进行通信。

软件的开发环境为 MSP430 单片机的专用开发环境 IAR Embedded Workbench for MSP430 V3^[41],它集成了调试和仿真等一系列功能,程序可进行 PC 端模拟,也可以通过 JTAG 口直接下载到单片机中进行在线调试,这使得开发非常方便。本章将对软件系统的整体实现进行详尽的介绍,包括软件架构,分层模型,内存管理和算法实现等。

5.2 硬件平台介绍

我们设计并实现了一个完整的低速 mesh 网络硬件实验系统,包括普通节点和网关。节点和网关设备采用相同的处理器和射频芯片,分别为 TI 公司的超低功耗 MSP430F149 和 Chipcon 公司的 CC2420,数据采集部分采用 sensirion 公司的 SHT71 温湿度传感器,它是一种数字传感器,低功耗,温湿度一体,使用方便。为了减小体积,方便调试和替换,我们将射频模块,串口通信模块,外接电源模块,传感器模块与主板分离,设计成可插拔式的。普通节点和网关设备的实物如图 5-1 和图 5-2 所示。

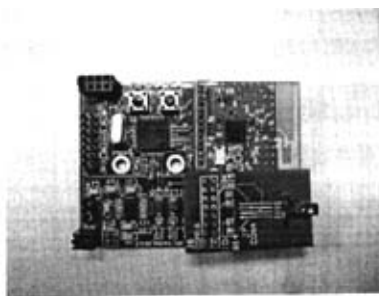


图 5-1 普通节点实物图

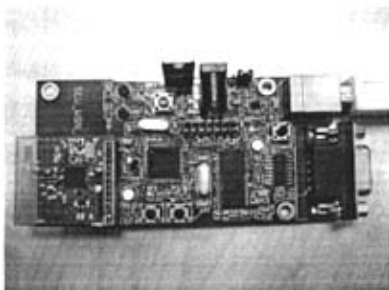


图 5-2 网关节点实物图

5.3 软件系统分层模型

IEEE802.15.4 标准规定了网络的物理层和 MAC 层规范，而 IEEE802.15.5 草案在此基础上进行了网络层的扩充，基于“功能分层，对等互连”的原则，软件系统也分为三层以便和标准相对应。但是为了简单方便，物理层和 MAC 之间不采用消息传递的方式，因为 CC2420 芯片已经实现了物理层的基本功能，因此在这里物理层只提供对物理设备的接口和驱动。整个软件系统的框架如图 5-3 所示：

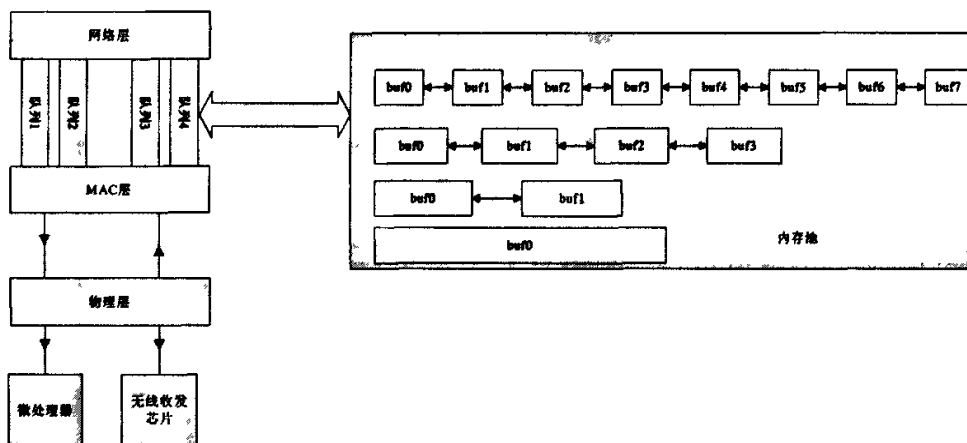


图 5-3 软件系统结构框图

物理层直接对收发芯片和单片机进行初始化并为上层提供函数接口，MAC 层直接对物理层的函数进行调用以实现标准规定的功能。MAC 层和网络层之间采用消息机制进行通信，网络层和 MAC 层间维护四个队列，MAC 层管理实体和 MAC 层公共服务子层各拥有两个方向相反的队列，一个方向为网络层到 MAC 层，另一个为 MAC 层到网络层，分别用于存放网络层到 MAC 层的消息和 MAC 层到网络层的消息。MAC 层管理实体主要为网络层提供以下服务：设备关联，解关联，信标管理，信息库管理，孤立设备管理，信道扫描，设备同步。MAC 层公共服务子层主要为网络层提供数据相关服务，包括：数据发送，收到数据通知等。网络层要 MAC 层提供服务时首先产生一个请求原语（消息），然后向内存池中申请一个合适的空间，申请成功后将消息内容写入该存储空间并将消息加入网络层

和 MAC 间相应的队列中等待处理，MAC 层和网络层定期查询队列，有消息到来时，将消息取出并执行相应的处理。

5.4 内存管理

5.4.1 内存空间基本结构

由于微处理器内存空间有限，频繁的内存分配会造成内存碎片，导致分配效率低下甚至不能得到合适的内存空间，因此对有限的内存进行有效的管理是软件系统的重要任务之一。系统以状态机的方式运行，每个状态层与层之间进行信息交换，每次信息交换都必须进行内存分配，所以事先为系统申请了一段连续的内存空间。内存空间的大小和消息的大小和分配的频繁程度相关，由于不同消息的大小不一致，内存空间也被分成大小不一的小块。每个大小相等的小块组成单向链表，并从链表的头部分配空间，释放的空间按大小插入到对应的链表尾部。链表的每个节点结构如图 5-4 所示：

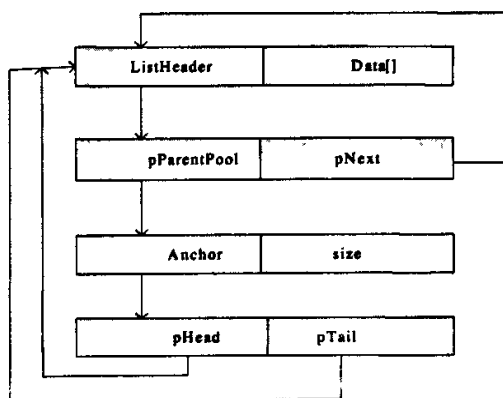


图 5-4 存储单元结构

其中 ListHeader 是一个结构体，包含两个部分：pParentPool 和 pNext，它们两个都是指针变量，pNext 指向链表上的下一个存储单元，而 pParentPool 指向由该链表构成的存储池的信息，存储池信息包括链表指针和链表中存储单元的大小。它们的具体数据结构如下所示：

```

typedef struct block0_tag{
    listHeader_t listHeader;    //链表信息
    uint8 data[bufSize0];      //数据存放区域
}block0_t;

typedef struct listHeader_tag {
    struct listHeader_tag *pNext;    //指向下一个节点指针
    struct pools_tag *pParentPool;   //指向存储池信息指针
}listHeader_t;
  
```

```

typedef struct pools_tag {
    anchor_t    anchor;           //链表信息
    uint8       blockSize;       //存储池中节点单元的大小
}pools_t;

typedef struct anchor_tag {
    listHeader_t *pHead;         //指向链表头部指针
    listHeader_t *pTail;         //指向链表尾部指针
}anchor_t;

```

具体的内存管理包括：内存池空间初始化，分配给定大小空间，释放空间，协议层间队列初始化，消息入队列，消息入队列头部，从队列尾部取出消息，以及检测队列中有无消息。内存池空间初始化比较简单，它所要做的事情就是填充存储池的信息，包括空间大小以及链表指针等。而分配给定大小的空间，我们采用了一种“奢侈”的方法：对每个请求都给它分配满足最小条件的固定空间。因为我们设定的内存池里的存储块大小都是固定的，每个链表都由相同大小的存储块组成，考虑到标准里定义的层间消息大小不一样，我们设定了四个大小不同的链表，其存储空间大小和个数分别倍增和倍增。内存池结构如图 5-5 所示：

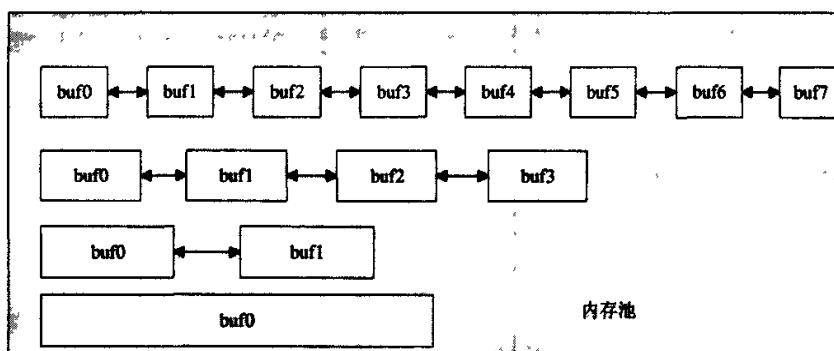


图 5-5 内存池结构图

5.4.2 内存管理方法

分配空间时，先检查链表 1 空间大小是否满足要求，如果是且链表中存在未分配空间则分配链表头部节点空间，如果空间大小不符合要求或者没有可用的空间则检查下一个链表并重复上述过程，如果最终没有合适空间则分配失败。释放空间则比较简单，只需要将它加入到所在链表的尾部即可。对内存空间的管理没有考虑内存的利用率以及内存空间耗尽的问题，根据我们的系统设计不会有开发的空间申请，而且内存块的使用时间也比较短，所以可以满足要求。内存块大小固定的设计虽然空间利用率不高，但是实现起来比较方便，而且也因为内存空间目前来说完全能够满足要求，所以也是可行的。当然我们也可以使用

另外一种内存管理方式,即采用非固定大小内存块,需要是可以精确分配,必要时也可以合并相邻空间,这样内存的利用率高,但是实现起来相对麻烦,这可以作为以后改进的方向。

内存管理还包括对协议层间队列的管理,主要功能为队列初始化,消息入队列,从队列中取出消息和判断队列中是否有消息。队列初始化比较简单,开始的时候将队列的头部和尾部指针都置为 NULL,因为系统初始化时没有消息生成。对消息入队列我们设计了两种方式,即将消息加入到队列尾部或将消息加入到队列头部。协议栈对消息队列处理时,默认从队列头部取得消息,提供将消息加入到队列头部的操作可以为某些消息提供较高的优先级。从队列中取出消息时首先将队列头部节点从队列中断开,然后返回该节点的头部指针。判断队列中是否有消息只需判断队列头部指针是否为 NULL。

5.5 设备状态机

从驱动的角度来看,协议的实现方法主要有事件驱动,过程驱动和自动实现^[42]。事件驱动的基本思想是将连续的代码离散化,它与本文采用的方法类似,但其对事件的定义比较狭隘,一般仅指硬事件。过程驱动是指协议的实现基于传统的过程形式,这种方法的缺点是结构不清晰,不利于复杂控制协议的实现。自动实现是指从协议文本或通信需求通过某些工具直接导出实现,这种方法的抽象程度较高,但目前还停留在理论阶段。

通过参考其它的协议实现方法,我们采用事件驱动的方式驱动协议的执行。每一层为其上一层提供服务,并请求下一层为其提供服务,所谓的服务是指本层通过下一层提供的服务建立自己的功能,为其上一层提供的能力。服务可以通过描述第 N 层用户和第 N 层之间的信息流来说明。数据流通过离散的即时事件来建模,每个事件通过由一层向另外一层传递一个服务原语组成,传递过程通过和 N 层用户关联的服务接入点完成。服务原语通过提供特别的服务来传达需要的信息。这些服务原语只是一个抽象,因为它们只是指定了提供的服务而没有指出实现服务的手段。服务通过描述服务原语及其参数来指定。一个服务可能会由多个相关服务原语,而每个服务原语可能会包含 0 个或多个参数来传达提供服务所需的信息。原语包括一下四种:

- 请求原语(Request): 请求原语从第 N 层用户传递到第 N 层来请求启动一个服务
- 指示原语(Indication): 只是原语从第 N 层传递到第 N 层用户来指示对第 N 层用户很重要的内部事件。这个事件逻辑上可能和远程服务请求相关,或者由第 N 层的内部事件触发。
- 响应原语(Response): 响应原语从第 N 层用户传递到第 N 层去完成由先前指示原语引发的处理过程。
- 确认原语(Confirm): 确认原语从第 N 层传递到第 N 层用户来传达一个或多个先前相关的请求原语的结果。

IEEE802.15.4 标准中一个典型的层间交互过程如下: 节点设备的网络层调用关联请求

原语发起关联过程，MAC 层收到该消息后进行解析，判断是关联请求，根据设备日前所处的状态决定要执行的操作。如果允许关联则向目的节点发送关联请求帧，目的节点收到此帧后将向其上层发送一个关联 Indication 原语，上层对该原语进行解析处理并决定是否同意节点关联，然后向下层发送一个 Response 原语，最终由下层向源节点发送关联响应帧。源节点收到关联响应帧后调用 Confirm 原语向上层报告关联请求的结果。完整的层间交互如图 5-6 所示：

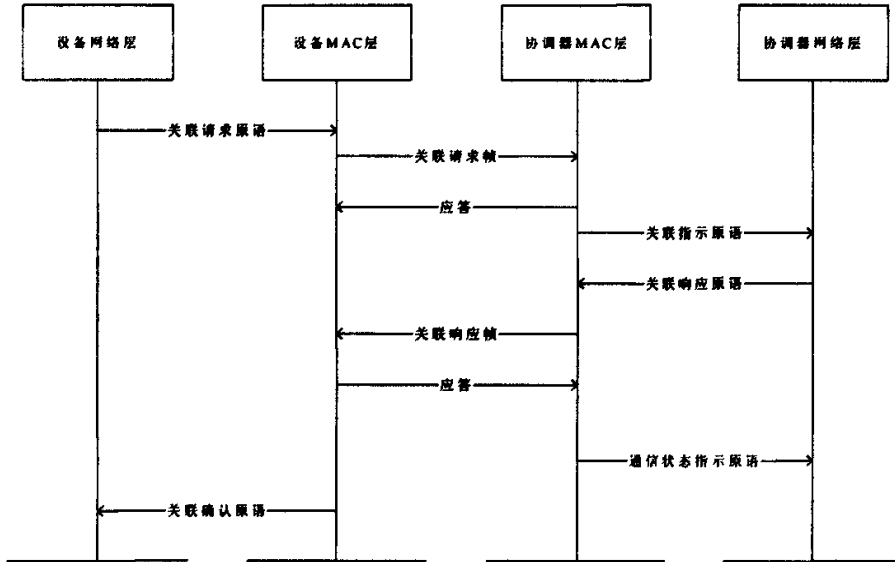


图 5-6 层间原语调用示意图

基于事件的驱动模型中的“事件”实际上是一个通信原语，也即消息。消息是一个结构体，包含消息类型和具体的消息实体。MAC 层到网络层的消息定义如下：

```
typedef struct nwkMessage_tag {
    primMlmeToNwk msgType;
    union {
        nwkAssociateInd_t      associateInd;
        nwkAssociateCnf_t      associateCnf;
        nwkDisassociateInd_t    disassociateInd;
        nwkDisassociateCnf_t    disassociateCnf;
        nwkBeaconNotifyInd_t    beaconNotifyInd;
        nwkGetCnf_t             getCnf;          // Not used
        nwkGtsInd_t             gtsInd;
        nwkGtsCnf_t             gtsCnf;
        nwkOrphanInd_t          orphanInd;
    };
};
```

```

nwkResetCnf_t      resetCnf;      // Not used
nwkRxEnableCnf_t   rxEnableCnf;
nwkScanCnf_t       scanCnf;
nwkCommStatusInd_t commStatusInd;
nwkSetCnf_t        setCnf;        // Not used
nwkStartCnf_t      startCnf;
nwkSyncLossInd_t   syncLossInd;
nwkPollCnf_t       pollCnf;

} msgData;

} nwkMessage_t;

```

msgType 是消息的类型，一共有 17 种，为枚举类型，而 msgData 包含具体的消息内容，定义为一个共同体，里面的每个变量均为一个结构体。

网关设备和普通设备在事件的驱动下在不同状态间转移并生成新的事件，每个设备的状态机包含 6 个状态，初始状态为 StateInit，网关设备的状态跳转如图 5-7 所示：

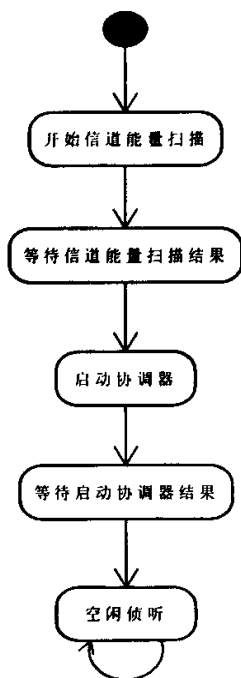


图 5-7 网关设备状态转移图

在原语的驱动下，网关设备从初始状态转到能量扫描状态，完成扫描后再跳转到等待扫描确认状态，如果网络层收到 MAC 层的扫描结果则对结果进行处理，选择一个通信信道并跳转到启动网络协调器状态，在该状态设备选择网络标识号和短地址，将自身状态改为活

动且允许其它节点关联，之后跳转到等待确认状态，最终进入侦听状态循环。值得注意的是，在每个状态中，处理完网络层的原语后都要调用 MAC 层的主函数处理网络层到 MAC 的 MAC 层自身生成的原语，这样构成一个闭环。

普通设备的状态转移图和网关设备类似，不过在每个状态进行的操作不同，为了使设备能顺利找到关联网络，增加了一个状态跳转，即当主动信道扫描后没有获得合适的关联设备信息时，将重新进行主动信道扫描，这样提高了网络的可靠性，避免了孤立设备的产生。普通设备状态转移如图 5-8 所示：

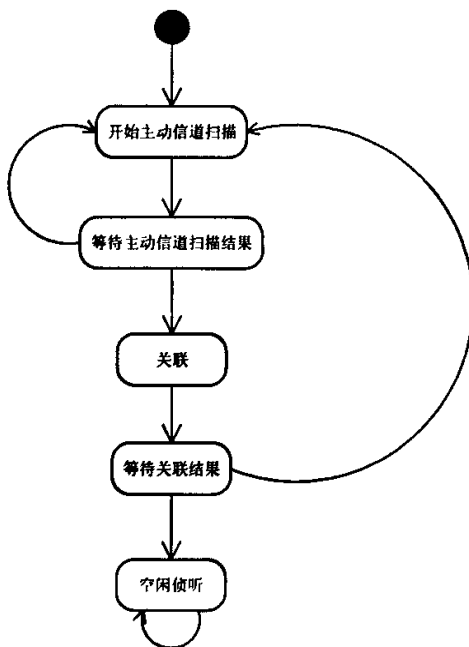


图 5-8 普通设备状态转移图

5.6 算法实现

整个软件系统是基于 IEEE802.15.4 标准和 IEEE802.15.5 草案开发的，因此网络层和 MAC 也采用了标准规定的算法，不过为了实现方便对路由算法作了适当简化，对所有的节点都采用树型算法，没有进行路由发现和路由维护。

5.6.1 MAC 层 CSMA/CA 算法实现

MAC 层采用非时隙 CSMA/CA 算法，随机数利用定时计数器寄存器的值获得，算法流程如图 5-9 所示。其中 NB 为退避次数，BE 为退避指数，它们的初始值分别为 0 和 macMinBE，标准规定 macMinBE=3，aMaxBE=5，maxCSMABackoffs=4。在非时隙 CSMA/CA 中，首先需对 NB 和 BE 进行初始化。然后 MAC 子层会在 $[0, 2^{BE-1}]$ 随机选取一个值完成一次退避延迟。也就是等待一个随机的时间，等待时间为随机取的那个值乘以退避时间单位 320us。之后 MAC 会请求物理层执行 CCA，就是空闲信道评估，这一过程 CC2420 芯片可

以自动完成，只需读其一个引脚即可得到 CCA 的结果。空闲信道评估的作用就是看信道是否空闲，是否有节点在发数据。

如果信道检测结果为忙，MAC 子层将变量 NB 和 BE 自动加 1，并保证 BE 值不超过 aMaxBE 且 $NB < \text{MACMaxCSMABackoffs}$ ，则重问随机退避那一步，如果 NB 的值大于 MACMaxCSMABackoffs，则终止此次 CSMA/CA 算法，表明此次信道接入过程失败。

如果信道检测为空闲，则表示竞争信道成功，该节点获得了信道的使用权。接着 MAC 子层可以立即发送数据帧。为简化设计，在检测信道空闲时没有再次进行退避，这样可能会出现多节点同时检测到空闲而发帧的情况，如果发生，则会碰撞，在此我们用多次重传来解决。

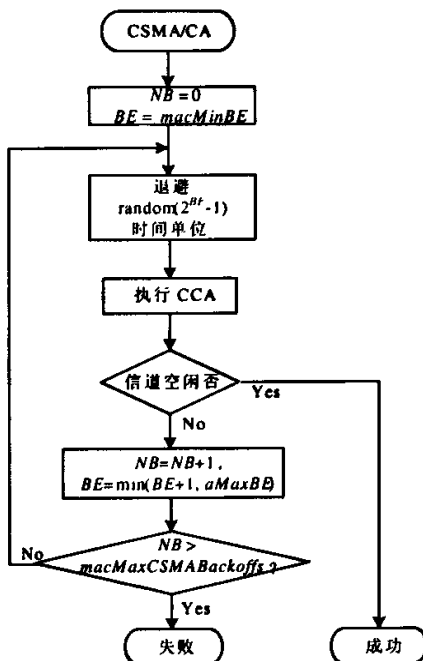


图 5-9 CSMA/CA 算法流程图

CSMA/CA 算法的代码实现比较简单，我们用一个子函数来返回算法的结果：竞争信道成功或者失败，然而这样做会带来一定的问题，因为竞争信道成功后不是马上发送数据，在这段延时中有可能其它设备会认为信道空闲从而进行数据发送，导致碰撞的产生。对于这个问题，我们采用了应答机制来解决，如果数据帧发送后规定的时间内没有收到应答帧，则认为发送失败，将进行最多三次重传。CSMA/CA 算法的核心代码如下：

```

uint8 mlmeCSMA()
{
    uint8 NB, BE, i; //NB 为退避次数, BE 为退避指数
    //按照标准规定的参数初始化退避次数和退避指数
    NB = 0;
    BE = macPIB.MinBE;
    //执行 CSMA
    do{
        //随机退避一段时间

```

```

i = ((uint8)random16)&((0x01<<BE)-1);
time=getTime(); //得到当前时间
while(timeDiff(getTime(), time)<aUnitBackoffPeriod*i);
//时间到, 开接收机
//执行 CCA
PHY_SetTRXState(PHY_RX_ON);
// 等待 RSSI 有效
while(PHY_RssiValid());
if(PHY_CCA()==PHY_IDLE) //信道空闲, 返回 TRUE, 可以马上发送数据
    return TRUE;
else{ //信道忙, 再次执行上述过程
    NB = NB + 1;
    if(BE < aMaxBE)
        BE++;
}
}while(NB<macPIB.MaxCSMABackoffs);
return FALSE; //退避规定的最大次数后信道仍然忙, 竞争信道失败
}

```

5.6.2 网络层路由算法实现

网络层采用树型路由算法, 即收到一帧数据后先判断目的节点是否为自己, 若是则进行处理, 否则进行转发。转发数据时直接将目的节点地址和一跳邻居地址进行比较, 并将数据转发给地址段包含目的节点地址的邻居, 如果没有这样的邻居则把数据发送给自己的父节点。这样的实现比较简单, 不需要建立路由表, 但是仅能够达到数据转发的目标, 没有考虑网络的性能, 不过采用新的算法对系统进行升级比较方便。路由算法的流程如图 5-10 所示:

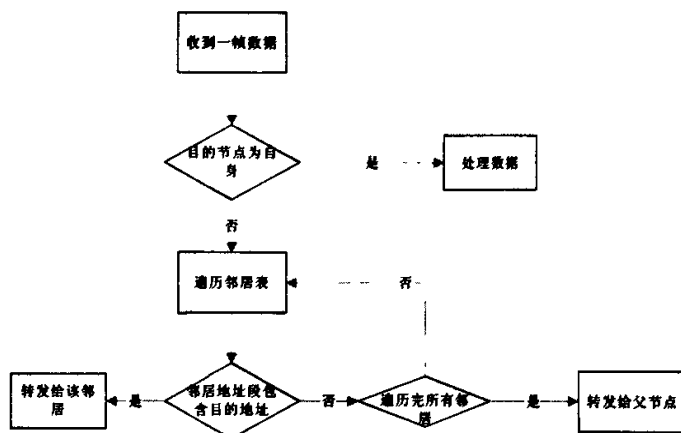


图 5-10 路由算法流程图

5.7 设备流程图

5.7.1 网关设备流程图

IEEE802.15.4 标准规定,网关节点将承担起网络建立和网络维护的工作。网关节点首先进行初始化,主要是对硬件初始化以及对标准中规定的一些参数和相应的变量初始化。初始化完成后进行信道能量扫描,为了降低通信过程中的外部干扰,根据扫描结果选取能量最低的信道作为通信信道。根据我们的地址分配算法,主节点将选取 0x0000 作为自己的短地址。这一切完成后网关节点将发送一个信标帧向周围设备告知它的信息,并设置自己的状态为活跃同时允许其它设备关联到自身,这样一个网络就建立起来了。网关节点工作流程如图 5-11 所示:

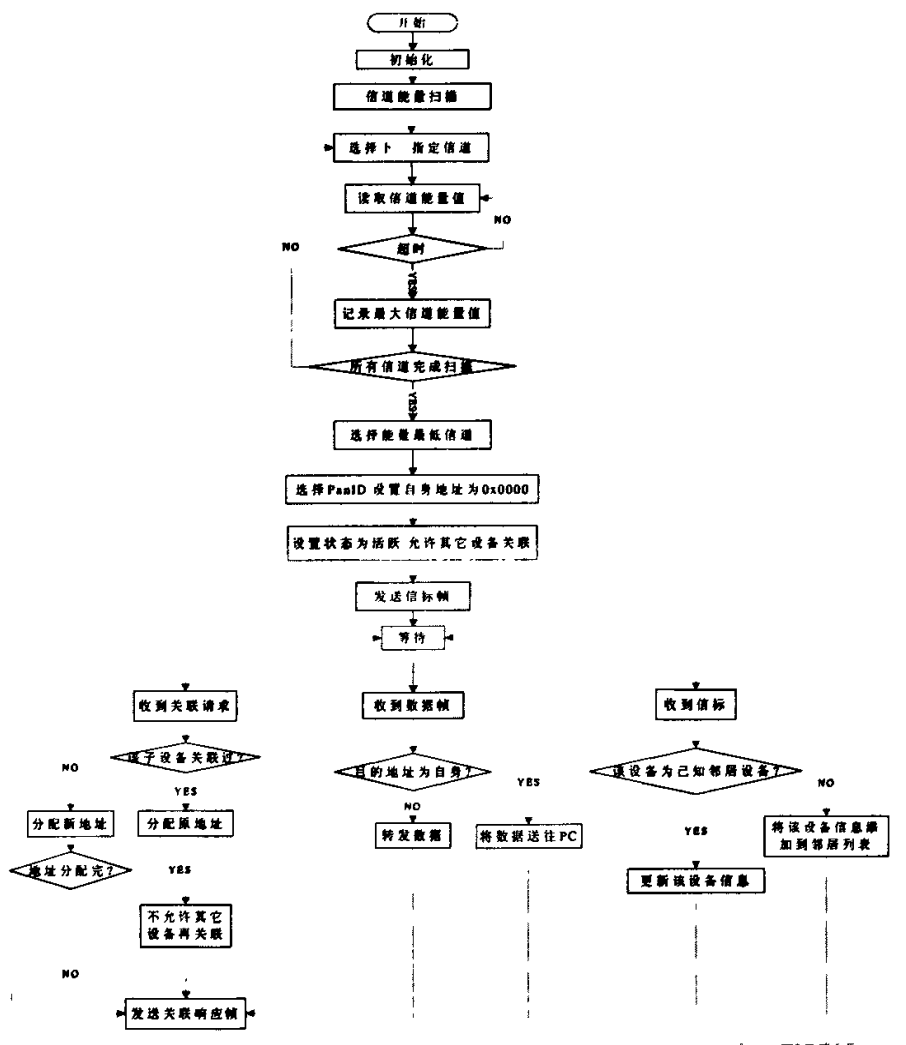


图 5-11 网关设备工作流程图

5.7.2 普通设备流程图

普通节点的工作流程和网关节点基本相同，差别在扫描过程，另外普通节点还负责采集数据并发送给网关节点或其它节点，其工作流程如图 5-12 所示：

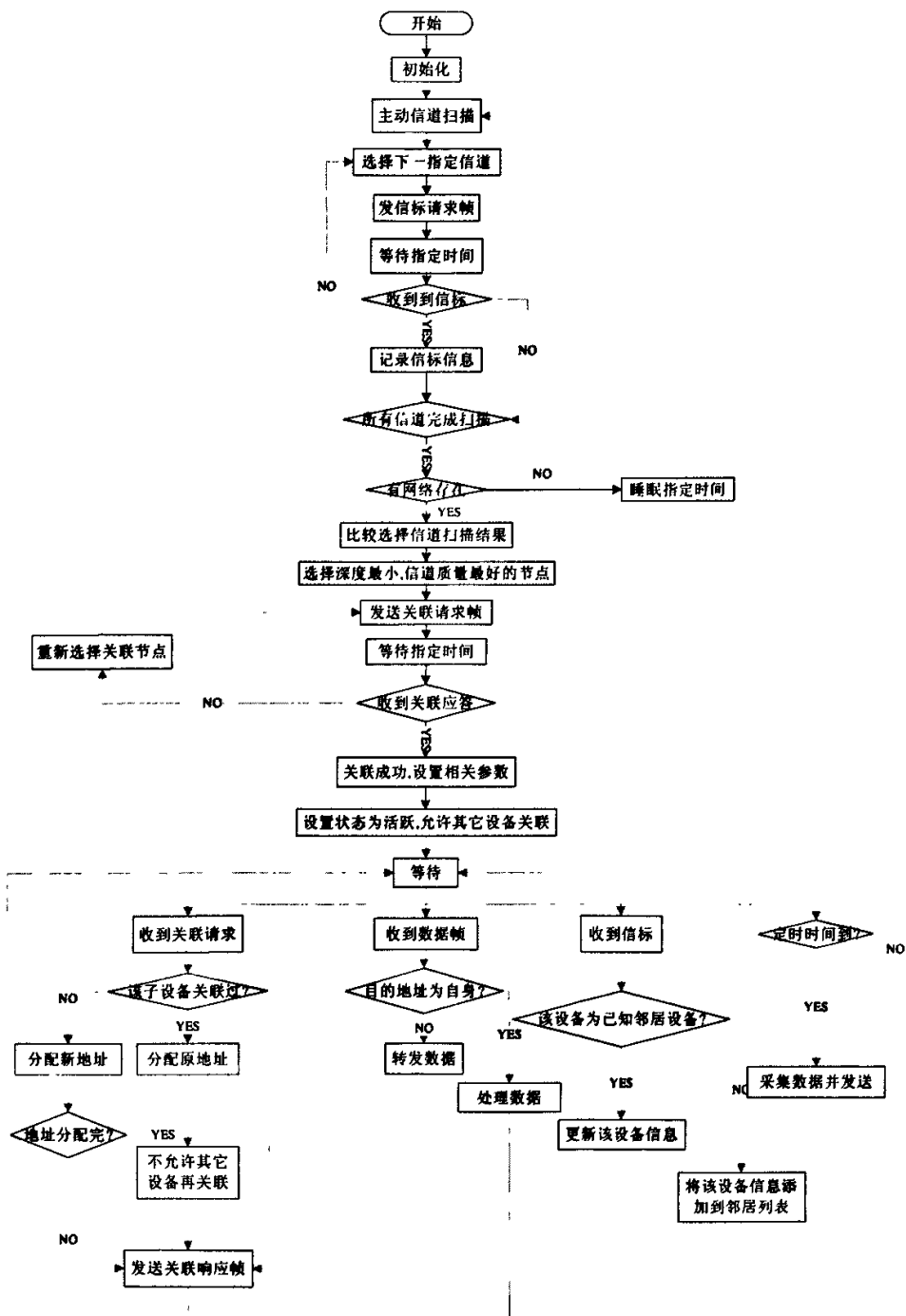


图 5-12 普通设备工作流程图

5.7.3 接收中断处理流程图

另外, CC2420 收到一帧数据后将产生一个中断, 所有和帧有关的处理都在接收中断里进行。首先将关闭接收中断以免处理过程中有另外的帧到达, 如果此时接收缓冲区满必须清空接收缓冲区, 然后对收到的帧进行过滤, 非法帧和我们不关心的帧将作丢弃处理, 对不同的帧有不同的处理方式。我们将数据帧中的内容重组成数据指示消息发送到网络层, 由网络层进行进一步处理(转发或直接处理), 而对 ACK 帧, 我们只记录帧序号以供网络层判断帧是否需要重发帧, 对于命令帧, 目前仅支持信标请求, 关联请求和关联响应, 其它命令帧将被丢弃。中断处理流程如图 5-13 所示:

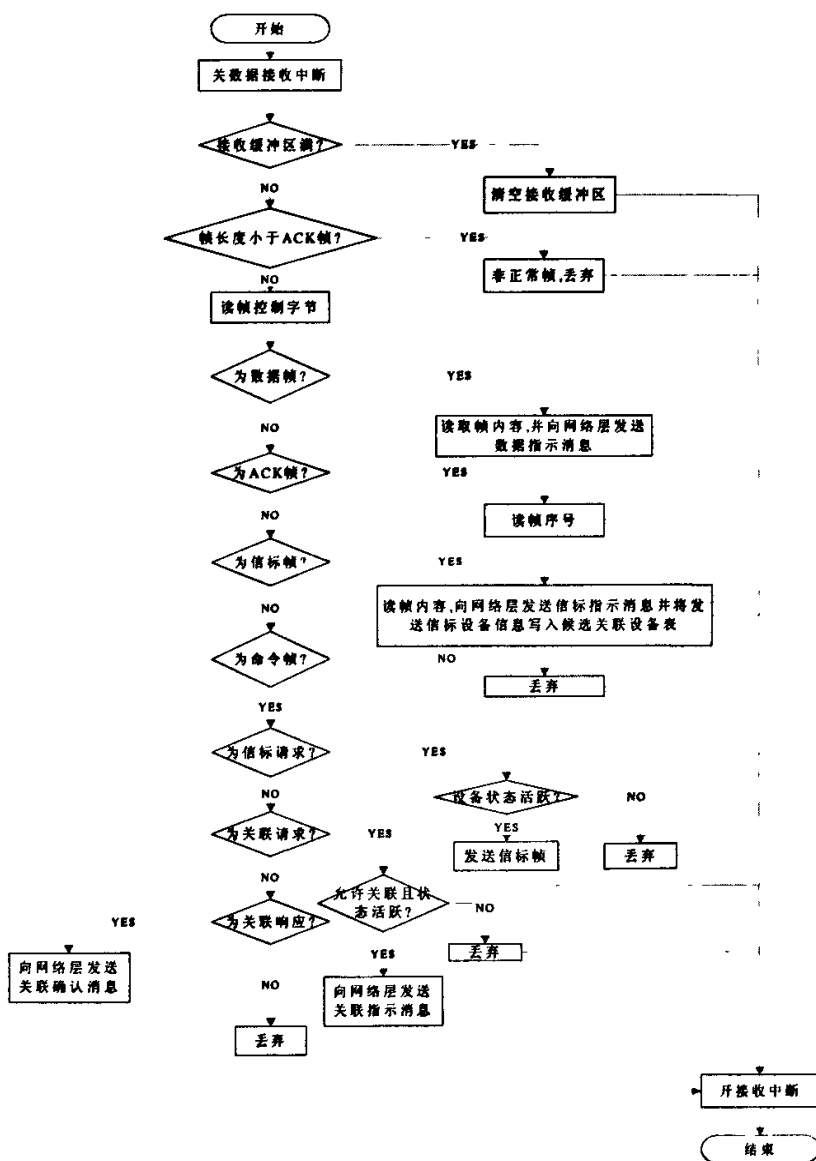


图 5-13 接收中断处理流程图

5.8 本章小节

本章主要讨论了低速无线 mesh 网络的软件实现问题, 包括软件总体结构, 分层模式, 内存管理等关键问题, 并在此基础上进一步阐述了具体的实现细节, 介绍了协议的事件驱动模型, 在此基础上引出了设备的有限状态机, 在 IEEE802.15.4 标准的基础上提供了 MAC 层和网络层的算法实现细节, 最后根据协议栈介绍了网关设备和普通设备的工作流程以及中断处理流程。在这个软件基础上可以比较方便的进行改进, 通过采用不同的算法实现网络性能的提高。总体实验平台也存在很多可以改进的地方, 硬件平台可以采用能力更强大处理器, 比如 ARM, 并在该硬件平台基础上开发更高效的软件系统以提高整体性能。目前对 Mesh 网络的研究方兴为艾, 不断有高效的算法被提出, 也有很多商用系统和测试系统出现, mesh 网络在不远的将来必将得到广泛的应用。

第6章 总结和展望

本文主要对基于 IEEE 802.15.4 标准和 IEEE 802.15.5 草案的低速无线 mesh 的关键技术进行了研究。本文的研究重点建立了在接收优先的情况下有效估计网络中节点吞吐率的模型,取得了一定的效果,另外对于网络公平性这个重要指标改进了标准 CSMA/CA 算法,大大提高了节点的流量公平性,同时改善了非饱和状态下节点的吞吐率和系统的丢包率。本文的主要研究内容包括接收优先时低速无线 mesh 网络中节点的吞吐率分析、节点公平性的优化研究以及符合标准的实验平台的软件实现。

6.1 论文已完成的内容

理论分析方面:

- (1) 建立了一种接收优先情况下估计基于 IEEE 802.15.4 CSMA/CA 协议的 mesh 网络中节点吞吐率的模型。该模型根据节点所处的位置和环境从估计节点的发送概率和退避时接收的包个数入手,分析了一个完整的退避周期内节点的平均发送成功概率,平均发送冲突概率,平均退避时隙以及平均接收的包个数,从而得到节点的吞吐率,为评估网络的性能提供了很好的工具。
- (2) 基于节点流量公平改进了标准 CSMA/CA 算法。本方法主要是考虑 mesh 网络中节点的负荷不公平,形成业务的“热区”和“冷区”,如果对他们采取相同的初始退避窗口会导致节点的流量公平性很差且非饱和情况下系统的吞吐率也不理想。为了解决这个问题,本文采用了根据节点的负荷设置初始退避窗口并根据节点实际获得的吞吐率动态进行调整的方案,获得了很好的效果。

仿真方面:

- (3) 使用 OPNET 仿真软件建立接收优先情况下基于 IEEE 802.15.4 标准的节点饱和吞吐率仿真模型,为第 3 章的理论分析提供了参照。
- (4) 使用 OPNET 仿真软件建立基于 IEEE 802.15.4 标准的 mesh 网络的节点流量公平性仿真模型,验证了第 4 章提出的改进算法的有效性。

实现方面:

- (5) 基于 MSP430F149 和 CC2420 构造的硬件实验平台,实现了符合 IEEE 802.15.4 标准和 IEEE 802.15.5 草案的软件系统,该系统采用分层结构,包括内存管理,层间采用原语进行通信,基本完成了 15.4 标准规定的功能,部分完成了 15.5 规定的功能,具有较好的扩展性,为实际研究算法的性能

提供了很好的平台。

6.2 需进一步完成的工作

- (1) 第3章建立的节点吞吐率估计模型不够准确,这主要是由于对两个重要参数即节点的发送概率和节点在退避期间平均接收的包数目采用了估计的方法,估计方法的准确度直接影响吞吐率估计的精确性,下一步应该考虑采用更准确的估计模型。
- (2) 第4章中改进的算法虽然取得了不错的效果,但是网络的实际情况异常复杂,本文中的改进算法仅适用于节点有相同的包生成速率的情况,对节点有不同包生成速率的情况不能取得很好的效果,接下来应该研究这种情况,对每个节点的负荷进行动态估计。另外,改进算法中对节点实际获得的吞吐率估计算法比较简单,将来可对其作进一步改进以取得更好的性能。
- (3) 无线 mesh 网络的关键技术还包括很多其它方面,例如路由技术,天线技术,功率控制技术以及安全技术等,限于本人的知识范围,本文仅对节点吞吐率和公平性做了部分研究,接下来可对其它技术进行研究。
- (4) 第5章的软件实现部分仅是实现了 mesh 网络的基本功能,包括网络建立和数据收发等,如果要实现一个完整的系统软件还有很多工作要做,比如网络层的复杂路由发现算法,路由维护过程,MAC 层的节点脱离网络过程等等。另外,受实验平台硬件系统的限制,将来可考虑将实验系统迁移到功能更强大的硬件上,可选方案为 ARM+FPGA,ARM 实现控制功能,FPGA 可实现部分算法。

致 谢

两年半的硕士生活即将结束，这期间经历的许多人和事，对我来说都是难于忘怀的。两年多前当我第一次踏入移动通信国家重点实验室的大门时，深深为这里的研究氛围和学术气息吸引，也对自己以后的学习生活充满了期待。而今我即将走出校门，踏入社会，依然对我朝夕相伴的实验室和我的师友们依依不舍，充满感激。

首先，衷心感谢我的导师徐平平教授。徐老师在学习上生活上给予我极大的关心和帮助。在学识上，徐老师知识渊博，对技术发展方向有独到的见解和敏锐的洞察力；在培养学生上，徐老师严格要求。在徐老师的悉心指导、言传身教之下，我不仅在专业知识上有了很大的提高，科研方法和独立科研的能力也有了很大的进步。

感谢毕光国教授在学习上的关心和帮助。

感谢师兄师姐的帮助；感谢同窗好友郑彦光，干克江，陈华敏，在你们的陪伴和帮助下，我在实验室度过了2年多充实而快乐的时光；感谢黄齐波，许恒锦，左婷，徐汉青，林旺德给予我的帮助；感谢研一的新同学们，你们努力学习，专心科研，使实验室形成了良好的氛围。

感谢我的室友们，感谢你们在我遇到困难的时候给予无私而热情的帮助，怀念我们一起“腐败”的日子，怀念我们在宿舍的高谈阔论，你们广博的知识和深刻的见解令我艳羡不已。

最后要感谢我的家人，无论何时你们总是我最坚强的后盾，在我失意的时候你们安慰我，让我倍增勇气；在我取得小小成绩的时候你们鼓励我，让我保持淡定平和的心境；在我不能独立的时候，你们毫不犹豫在经济上给我大力支持，而你们却饱尝艰辛，毫无怨言。我的成长离不开你们，你们也让我更添承担责任的力量！

刘俊平

2008年1月于东南大学移动通信国家重点实验室

参考文献

- [1] P802.11, "IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications, " Nov. 1997.
- [2] [http: //www.ydshs.com/keyan/2002-4/4-9.htm](http://www.ydshs.com/keyan/2002-4/4-9.htm)
- [3] IEEE 802.15.4, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)", 2003
- [4] IEEE Standard for Part 15.3: Wireless Medium Access Control(MAC) and Physical Layer(PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs) [S]. September 2003.
- [5] Heecheol Song, Bong Chan Kim, Jae Young Lee, and Hwang SooLee, "IEEE 802.11-based Wireless Mesh Network Testbed, "Mobile and Wireless Communications Summit, 2007. 16th IST 1-5 July 2007 Page(s): 1 - 5
- [6] Nagesh S. Nandiraju, Deepti S. Nandiraju, and Dharma P. Agrawal, "Multipath Routing in Wireless Mesh Networks, " Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on Oct. 2006 Page(s): 741 - 746
- [7] Paul A.S. Ward, "Wireless Mesh Networks, " Proceedings of the 4th Annual Communication Networks and Services Research Conference (CNSR'06)
- [8] Song, Heecheol; Kim, Bong Chan; Lee, Jae Young; Lee, Hwang Soo; "IEEE 802.11-based Wireless Mesh Network Testbed, " Mobile and Wireless Communications Summit, 2007. 16th IST 1-5 July 2007 Page(s): 1 - 5
- [9] IAN F. AKYILDIZ, "A Survey on Wireless Mesh Networks, " IEEE Radio Communications • September 2005
- [10] ELIZABETHM. ROYER, CHAI-KEONG TOH, "A review of current routing protocols for ad hoc mobile wireless networks, " IEEE Personal Communications April 1999
- [11] Xun Chen, Peng Han, Qiu-Sheng He, Shi-liang Tu, Zhang-Long Chen, "A Multi-Channel MAC Protocol for Wireless Sensor Networks, " Computer and Information Technology, 2006. CIT '06. The Sixth IEEE International Conference on Sept. 2006 Page(s): 224 - 224
- [12] Aoun, B.; Boutaba, R.; Kenward, G, "Analysis of Capacity Improvements in Multi-Radio Wireless Mesh Networks, " Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd Volume 2, 7-10 May 2006 Page(s): 543 - 547
- [13] NAOUEL BEN SALEM AND JEAN-PIERRE HUBAUX, EPFL, "SECURING WIRELESS MESH NETWORKS, " IEEE Wireless Communications • April 2006
- [14] 张勇, 蔡杰, 宋梅, 宋俊德, "无线 mesh 网络公平性研究, " 中国科学技术大学学报, 第 37 卷第 2 期, 2007 年 2 月
- [15] G. Bianchi, "Performance Analysis of the IEEE802.11 Distributed Coordination Function, " IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL.

18, NO. 3, MARCH 2000

- [16] Wang, Ying; Yan, Nuo; Li, Tong, "Throughput Analysis of IEEE 802.11 in Multi-Hop Ad Hoc Networks, "Wireless Communications, Networking and Mobile Computing, 2006. WICOM 2006. International Conference on Sept. 2006 Page(s): 1 - 4
- [17] Mukesh M. Hira, Fouad A. Tobagi, "Throughput Analysis of a Path in an IEEE802.11 Multihop Wireless Network, " IEEE Communications Society subject matter experts for publication in the WCNC 2007 proceedings
- [18] 那成亮, 周廷显, 李立宏, 于路, "用于多跳无线传感器网络的自适应退让算法," 东南大学学报, 第 36 卷第 2 期, 2006 年 3 月
- [19] Xin Wang and Koushik Kar, "Throughput Modelling and Fairness Issues In CSMA/CA Based Ad-Hoc Networks, " INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE Volume 1, 13-17 March 2005 Page(s): 23 - 34 vol. 1
- [20] Jangeun Jun and Mihail L. Sichitiu, "Fairness and Qos in multihop wireless networks, " Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th Volume 5, 6-9 Oct. 2003 Page(s): 2936 - 2940 Vol.5
- [21] Xin Wang and Koushik Kar, "Throughput Modelling and Fairness Issues In CSMA/CA Based Ad-Hoc Networks, " INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE Volume 1, 13-17 March 2005 Page(s): 23 - 34 vol. 1
- [22] Rima Khalaf and Izhak Rubin, "Throughput and Delay Analysis in Single Hop and Multihop IEEE 802.11 Networks, " Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on 1-5 Oct. 2006 Page(s): 1 - 9
- [23] A.V. Babu and Lillykutty Jacob, "Performance Analysis of IEEE 802.11 Multirate WLANs, " 2005 International Conference on Wireless Networks, Communications and Mobile Computing
- [24] 刘永强, 严伟, 赵通, 赵慧周, "面向路径的无线多跳网络端到端吞吐量分析," 电子学报, 第 5 期, 2007 年 5 月
- [25] Babu, A.V.; Jacob, L.; "Fairness Analysis of IEEE 802.11 Multirate Wireless LANs, " Vehicular Technology, IEEE Transactions on Volume 56, Issue 5, Part 2, Sept. 2007 Page(s): 3073 - 3088
- [26] 段中兴, 张德运, "多速率无线局域网的自适应退避算法," 西安交通大学学报, 第 40 卷第 6 期, 2006 年 6 月
- [27] Chuanxia WU, Junhuan FENG and Pingzhi FAN, "On a New Queue Backoff Fair Algorithm for Ad Hoc Networks, " Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on 27-29 Aug. 2003 Page(s): 335 - 339
- [28] Brahim Bensaou, Yu Wang, Chi Chung KO, "Fair Medium Access in 802.11 based Wireless Ad-Hoc Networks, " Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on 11 Aug. 2000 Page(s): 99 - 106

- [29] Zhongbang Ydo, Pingyi Pan, and Zhigang Cao, "An Enhanced CSMA/CA Mechanism for Multihop Ad Hoc Networks, " 10th Asia-Pacific Conference on Communications and 5th International Symposium on Multi-Dimensional Mobile Communications
- [30] 吴传霞, 范平志, 冯军焕, "一种 Ad Hoc 网络信道接入排队退避公平算法," 系统仿真学报, Vol.16 No.5, May 2004
- [31] X. L. Huang, B. Bensaou, "On Max-min Fairness and Scheduling in Wireless Ad-Hoc Networks: Analytical Framework and Implementation, " ACM MobiHoc'01, pp. 221-231, Oct. 2001
- [32] Bozidar Radunovi, Jean-Yves Le Boudec, "Rate Performance Objectives of Multihop Wireless Networks, " IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 3, NO. 4, OCTOBER-DECEMBER 2004
- [33] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single node case, " IEEE/ACM Transactions on Networking, vol. 1, pp. 344-357, 1993.
- [34] Rabbi, Md. Forhad; Rahman, Md. Taufiqur; Uddin, Md. Afser; Salehin, G.M. Abdullah; "An Efficient Wireless Mesh Network: A New Architecture, " Communication Technology, 2006. ICCCT '06. International Conference on Nov. 2006 Page(s): 1 - 5
- [35] Ken Duffy, David Malone, and Douglas J. Leith, "Modeling the 802.11 Distributed Coordination Function in Non-Saturated Conditions, " IEEE COMMUNICATIONS LETTERS, VOL. 9, NO. 8, AUGUST 2005
- [36] Li Bin Jiang and Soung Chang Liew, "Proportional Fairness in Wireless LANs and Ad Hoc Networks, " IEEE Communications Society / WCNC 2005
- [37] Texas Instruments, msp430x1xx Family User's Guide Extract, 2005
- [38] 秦龙编著, 《MSP430 单片机应用系统开发典型实例》, 中国电力出版社, 2005
- [39] 魏小龙编著, 《MSP430 系列单片机接口技术及系统设计实例》, 北京航空航天大学大学出版社, 2002
- [40] Chipcon, CC2420_Data_Sheet_1_4, Chipcon, 2006
- [41] MSP430 IAR Embedded Workbench™ IDE User Guide for Texas Instruments' MSP430 Microcontroller Family
- [42] 李志飞, "移动计算网络研究及实现分析," 学位论文 南京航空航天大学 2002

攻读硕士学位期间的研究成果

1. 学术论文

- [1] 刘俊平, 徐平平, “基于 Zigbee 的无线 mesh 网络实验平台设计”, 中国科技论文在线
- [2] 刘俊平, 徐平平, “基于 IEEE 802.15.4 标准的 Mesh 网络软件实现”, 第二十二届南京地区研究生通信年会

2. 发明专利

- [1] 刘俊平, 徐平平, 郑彦光, 多跳无线传感器网络动态比例公平接入优化方法, 已提交
- [2] 郑彦光, 徐平平, 刘俊平, 分布式无线传感器网络竞争接入周期多级配置方法, 已提交