

学校代码: 10286
分类号: TN431
密 级: 公开
U D C: 621.3
学 号: 103293



Y2439385



WF2807235

东南大学

工程硕士学位论文

无线传感器网络低功耗异步 MAC 协议 设计

(学位论文形式: 应用研究)

研究生姓名: 刘野

导师姓名: 吴建辉 教授

虞建立 高工

申请学位类别 工程硕士 学位授予单位 东南大学

工程领域名称 集成电路工程 论文答辩日期 2013 年 5 月 14 日

研究方向 无线传感器网络 学位授予日期 20 年 月 日

答辩委员会主席 刁龙 评 阅 人 _____

東南大學
碩士學位論文

無線傳感器網絡低功耗異步 MAC 協議設計

專業名稱: 集成電路工程

研究生姓名: 劉野

導師姓名: 吳建輝

虞建立

本論文獲國家自然科學基金項目（No.61001104）和江蘇省自然科學基金項目（No.BK2010166）資助。

DESIGN OF AN ENERGY-EFFICIENT ASYNCHRONOUS MAC PROTOCOL FOR WIRELESS SENSOR NETWORK



Y2439385

A Thesis Submitted to
Southeast University
For the Academic Degree of Master of Engineering

BY

LIU Ye

Supervised by
Prof. WU Jian-hui
and
Senior Engineer YU Jian-li

School of Integrated Circuit
Southeast University

April 2013

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: 刘宇 日期: 2013.5.18

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（包括以电子信息形式刊登）论文的全部内容或中、英文摘要等内容。论文的公布（包括以电子信息形式刊登）授权东南大学研究生院办理。

研究生签名: 刘宇 导师签名: 吴忠耀 日期: 2013.5.18

摘要

无线传感器网络是由部署在监测区域内的大量无线传感器节点，通过自组织方式形成的网络。其目的是协作地采集、处理和传输感知信息。MAC 协议是无线传感器网络中最重要的技术之一，是连接物理层硬件和上层软件协议的枢纽，其设计优劣对网络性能影响较大。

无线传感器网络 MAC 协议设计的一个显著特征就是其应用相关性，针对不同应用场景需要设计出不同的 MAC 协议。典型的无线传感器网络应用包括环境监测和目标跟踪，它们都属于基于事件驱动应用场景，网络流量呈现相关突发特点：经过长时间的空闲状态后，检测到事件的多个节点同时向汇聚节点传输大批数据。数据同时传输会导致严重的碰撞，是传感器网络中能量浪费的主要因素之一。而现有异步 MAC 协议主要解决了同步和空闲能耗问题，针对相关突发流量问题还有不足之处。因此，本文提出了 SC-MAC 协议，一种以发送者为中心的 MAC 协议，利用单跳协作反馈机制、预约数据交互机制等，在不引入过多开销的基础上大幅度减少传输碰撞，节省了节点能耗。为了减少数据多跳延时，本文同时加入了一种延时优化机制。协议在 NS2 仿真平台进行实现，并对其性能同 PW-MAC 协议进行比较，结果表明 SC-MAC 协议无论在能耗还是数据传输延时方面都有较大的性能提升。

本文提出的 SC-MAC 协议适用基于事件驱动应用场景及对网络寿命要求较高的应用，对于无线传感器网络 MAC 协议进一步研究具有很大的实用价值。

关键词： 无线传感器网络；介质访问控制；异步协议；突发流量；能量有效

Abstract

Wireless sensor network is composed of massive sensor nodes, which are deployed in monitoring region forming a self organized network in order to cooperatively sense, collect and process information. MAC protocol is at the bottom of protocol stack, which has great influence on the performance of WSN.

A remarkable feature of MAC protocol design in wireless sensor networks is application correlation, needing to design different protocols for different application scenarios. Event-driven applications in wireless sensor networks feature correlated traffic bursts: after a period of idle time with light traffic loads, multiple sensors that have detected the same event have to transmit large amounts of data simultaneously to sink node or cluster head. The demand for simultaneous data transmission often causes severe collision, which is one of the most significant sources of energy consumption in wireless sensor networks. In this paper, we propose SC-MAC (sender-centric MAC), a new asynchronous duty cycle MAC protocol designed for burst traffic loads. SC-MAC achieves collision-free environment while do not introduce extra overhead. In order to minimize delivery latency in tree structure or other multi-hop networks, SC-MAC also introduces a latency optimization mechanism. We show the performance of SC-MAC through ns-2 simulation and compare it to PW-MAC, the state-of-the-art asynchronous MAC protocol. The simulation results show that SC-MAC significantly minimizes energy consumption and delivery latency.

SC-MAC is suitable for event-driven applications of wireless sensor network and the situation requiring long network life. It is also quite useful for MAC protocol research.

Keywords: wireless sensor network; media access control; asynchronous protocol; burst traffic; energy efficiency

目 录

| | |
|-------------------------------------|-----|
| 摘要 | I |
| Abstract..... | II |
| 目录 | III |
| 第一章 绪论 | 1 |
| 1.1 论文的背景与研究意义 | 1 |
| 1.2 国内外研究现状 | 2 |
| 1.3 研究内容与设计指标 | 4 |
| 1.3.1 研究内容 | 4 |
| 1.3.2 设计指标 | 5 |
| 1.4 论文的主要内容与结构 | 5 |
| 第二章 无线传感器网络 MAC 协议 | 7 |
| 2.1 无线传感器网络 MAC 协议概述 | 7 |
| 2.2 无线传感器网络 MAC 协议设计影响因素 | 7 |
| 2.3 无线传感器网络异步 MAC 协议发展现状 | 8 |
| 2.3.1 B-MAC | 8 |
| 2.3.2 X-MAC | 9 |
| 2.3.3 RI-MAC | 10 |
| 2.3.4 PW-MAC | 12 |
| 2.4 本章小结 | 13 |
| 第三章 SC-MAC 协议的设计 | 15 |
| 3.1 事件驱动应用场景特点分析 | 15 |
| 3.2 现有解决传输碰撞方法及比较 | 15 |
| 3.2.1 PW-MAC | 16 |
| 3.2.2 Flip-MAC | 16 |
| 3.2.3 RC-MAC | 17 |
| 3.2.4 TRAMA | 18 |
| 3.2.5 协议小结及对比 | 18 |
| 3.3 现有协议所面临问题深入分析 | 19 |
| 3.3.1 时间同步问题 | 19 |
| 3.3.2 竞争节点持续唤醒问题 | 19 |
| 3.3.3 节点公平性问题 | 19 |
| 3.3.4 异步广播支持问题 | 19 |
| 3.4 一种针对事件驱动应用低功耗异步 MAC 协议的提出 | 20 |
| 3.4.1 SC-MAC 协议概述 | 20 |
| 3.4.2 异步周期询问机制 | 21 |
| 3.4.3 预测唤醒传输机制 | 23 |
| 3.4.4 广播及路由支持 | 24 |
| 3.4.5 延时优化机制 | 25 |
| 3.5 本章小结 | 26 |
| 第四章 SC-MAC 协议在 NS2 平台上的实现 | 27 |
| 4.1 NS2 网络仿真平台 | 27 |
| 4.1.1 NS2 平台的简介 | 27 |

| | |
|-----------------------------------|-----------|
| 4.1.2 NS2 平台主要构件 | 28 |
| 4.1.3 NS2 仿真基本流程 | 29 |
| 4.2 SC-MAC 协议核心数据结构 | 30 |
| 4.2.1 控制帧结构 | 30 |
| 4.2.2 数据帧结构 | 31 |
| 4.2.3 邻居链表结构 | 32 |
| 4.3 SC-MAC 协议具体实施方案 | 33 |
| 4.4 SC-MAC 协议具体方案的代码实现 | 39 |
| 4.4.1 异步周期询问机制的实现 | 39 |
| 4.4.2 预测唤醒数据交互流程的实现 | 40 |
| 4.4.3 异常处理的实现 | 40 |
| 4.5 本章小结 | 41 |
| 第五章 SC-MAC 协议仿真与结果分析 | 43 |
| 5.1 仿真参数设置 | 43 |
| 5.2 突发事件源端协议性能分析 | 44 |
| 5.2.1 拓扑场景建立 | 44 |
| 5.2.2 协议性能分析 | 44 |
| 5.3 突发事件远端协议性能分析 | 46 |
| 5.3.1 拓扑场景建立 | 46 |
| 5.3.2 协议性能分析 | 47 |
| 5.4 随机突发场景下协议性能分析 | 48 |
| 5.4.1 随机突发场景建立 | 48 |
| 5.4.2 协议性能分析 | 48 |
| 5.5 本章小结 | 50 |
| 第六章 总结与展望 | 51 |
| 6.1 总结 | 51 |
| 6.2 展望 | 51 |
| 致 谢 | 53 |
| 参考文献 | 55 |
| 攻读硕士学位期间的研究成果 | 59 |
| 附录 | 61 |

第一章 绪论

1.1 论文的背景与研究意义

近年来，随着无线通信、网络技术和微处理技术的迅速发展，无线传感器网络技术逐渐成熟并成为人们关注和研究的重点^[1-5]。国内外涌现出了大量无线传感器网络应用系统和研究探索。这些应用逐渐步入了大规模部署的应用阶段，并贴近人类日常生活的方方面面。例如，新加坡南洋理工大学 NWSP 系统实现大规模实时天气监测^[6]、麻省理工学院的 PipeNet 系统实现自来水管监测^[7]等。

图 1.1 展示了香港科技大学的绿野千传^[8]项目的原型系统实验与观测。它是由部署在监测区域内大量低成本、低功耗、具备感知、数据处理、存储和无线通信能力的传感器节点通过自组织方式形成的网络，完成数据采集和处理，并传输网络监测区域内被感知对象的信息，发送给终端观测者。它改变了人与自然的交互方式，以传感网络形式将物质世界连接起来，为广泛的社会应用需求提供服务，成为一种全新的信息交互模式。



图 1.1 绿野千传项目实现大规模森林监控

无线传感器网络中，如图1.2所示，媒体介质访问控制（MAC）协议处于整个网络协议栈的下层，其主要功能为负责信道接入、控制节点的报文传输对无线共享媒体的占用以及通过一组规则和过程来更有效、有序和公平使用共享媒体以保证网络的整体性能。MAC 协议是无线传感器网络中最重要的技术之一，是连接物理层硬件和上层软件协议的枢纽，其设计优劣对网络性能影响较大。总体来说，研究MAC协议具有以下意义：

- ◆ MAC协议性能对组网性能影响非常大，决定网络性能的最关键技术之一；
- ◆ MAC协议是网络通信领域研究经久不衰的课题；
- ◆ 每次新的网络应用或者网络结构出现，MAC 协议通常需要重新研究和设计；
- ◆ MAC层是目前各种主流的通信标准制定的核心内容之一；

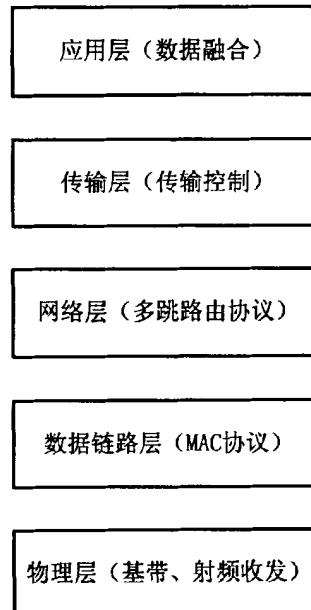


图1.2 无线传感器网络协议栈架构

传感节点通常采用电池供电。由于其体积很小，携带的能量十分有限，因此传感节点常常由于电池的耗尽而失效或废弃。而在大多数应用中，无法给电池充电或更换电池，一旦电池能量用完，这个传感节点也就失去了作用而死亡。因此在传感器网络设计过程中，任何技术和协议的使用都要以节能为前提。在MAC协议设计中，为使传感器网络有效且长时间工作，需要尽可能降低节点的功耗。现有的MAC层降低功耗的策略是使节点在没有通信需要的情况下及时进入睡眠以节约能量。在保证正常的通信功能和性能前提下，能量节约程度是无线传感器网络中非常重要的设计指标。通过降低功耗来延长传感器网络的生命周期，就意味着降低了传感器网络的总体成本。成本的降低就会推动无线传感器网络在民用和工业生产等众多领域的广泛应用。同时，在商业竞争中，性能相同的产品，降低价格就等于增强产品竞争力。因此，研究低功耗的MAC协议有着深远的意义和工程应用价值。本次课题研究的目的就是在现有的研究基础上进一步降低传感节点的功耗，增加网络寿命，进而推进无线传感器网络的发展。

1.2 国内外研究现状

无线传感器网络已成为学术界和工业界的研究热点，从目前国内外传感器网络已有研究成果统

计来看，对网络协议、能量、定位、可靠性、网络架构以及数据处理等问题的研究占所有研究成果的80%，网络协议的研究占其中35%。作为传感器网络协议栈重要基础架构的MAC协议，决定着无线信道的使用方式，负责为节点分配无线通信资源，直接影响网络整体性能，成为无线传感器网络协议研究的重中之重。已有大量针对传感器网络不同特点和具体应用的MAC协议相继提出。

无线传感器网络的MAC协议与应用高度相关，研究人员从不同的角度出发提出了多种MAC协议。但到目前为止，无线传感器网络MAC协议还没有一个统一的分类方式。可根据MAC协议的信道分配方式、数据通信类型、性能需求、硬件特点以及应用范围等策略，使用多种分类方法对其进行分类：

根据信道访问策略的不同可分为基于竞争的MAC协议（如S-MAC^[9]、T-MAC^[10]、P-MAC^[11]）、基于调度的MAC协议（如DEANA^[12]、TRAMA^[13]、TDMA-W^[14]）和混合MAC协议（如Z-MAC^[15]、AMAC^[16]）；根据信道使用数目可分为单信道MAC协议（如DW-MAC^[17]、R-MAC^[18]、S-MAC）和多信道MAC协议（如SSCH^[19]、MMSN^[20]、MMAC^[21]、EM-MAC^[22]）；根据协议栈层间交互可分为单层MAC协议（如SCP^[23]、R-MAC、S-MAC）和跨层MAC协议（如CAME^[24]、SARA-M^[25]、AIMRP^[26]）；根据节点是否同时唤醒分为同步MAC协议（如SCP、DW-MAC、S-MAC）和异步MAC协议(X-MAC^[27]、RI-MAC^[28]、PW-MAC^[29]）；

此外，近年来WSN技术迅速发展，与应用相结合成为主流趋势，学者们在原有MAC协议研究的基础上，提出了许多面向特定应用的MAC协议，具体如下：

无线多媒体传感器网络MAC协议研究：其理念是将传统无线传感器网络同图像、音频、视频等多媒体信息相结合的一种新式传感网络。无线地下传感器网络MAC协议研究：此种网络中节点一般工作在地下，如埋藏在土壤里观测土地质量信息的变化，或者是放置在地下空间，如放置在铁路隧道中，或矿井下检测空气成分等；无线水下传感器网络MAC协议研究：水下环境与陆地环境属性的不同导致水下传感器网络与传统陆地传感器网络的设计有很大不同。如水下无线传感器网络需要使用声波通信，而非射频信号。此外，水下节点移动性也是需要重点关注的问题；移动无线传感网MAC协议研究：无线传感器网络日渐与移动领域相结合，而节点的可移动性是学者们面临的新的挑战；无线体域网MAC协议研究无线体域网是指将无线传感器节点放置在人体的不同部位来监测人体的生理健康情况或者是提供短距离无线应用。由于传感器节点是在近人体表面通信，因此，发射功率继续十分低，以免影响人体承受能力；

本课题的研究方向为无线传感器网络异步MAC协议，其典型的发展历程为B-MAC、X-MAC、RI-MAC、PW-MAC，具体如下：

2004年，加州大学伯克利分校的J. Polastre 等人^[30]提出了B-MAC (Berkeley MAC) 协议，B-MAC采用基于自适应前导采样来减小空闲侦听。该协议采用较小的核心模块设计，可灵活搭配外围的功

能部件，形成适应性较强的MAC 协议，使得协议的吞吐率、延迟时间、能耗等方面的性能优于传统的无线传感器网络MAC 协议。在B-MAC 协议设计中，每个节点按照独立的睡眠调度运行，并在发送数据包前采用比较长的前导采样，使得周期性采样信道的目标节点能够有足够的时间检测到信道状态以接收数据包。如果节点检测到前导数据则继续接收下面的数据，否则进入睡眠以降低能耗。协议中采用清晰信道分配（clear channel assessment, CCA）来检测信道是否空闲以减小冲突可能性，用数据包退避来进行信道状态仲裁，采用链路层确认以提高协议的可靠性，并采用低功耗侦听（low power listening, LPL）进行数据通信。

2006年,科罗拉多大学波尔得分校的Michael Buettner提出了X-MAC协议，X-MAC协议再次缩短前导序列的长度，同时引入握手机制进一步减小发送前导序列的能量开销。前导序列由若干较小的频闪前导组成，其中包含目的地址，非接收节点尽早丢弃分组并睡眠。利用频闪前导之间的时间间隔，接收节点向源节点发送早期确认。发送节点收到早期确认后立刻发送数据分组，从而避免发送节点过度前导和接收节点过度侦听。X-MAC还设计了一种自适应算法，根据网络流量变化动态调整节点的占空比以减少单跳延时。X-MAC在能量效率、吞吐量和延时等性能上优于B-MAC。与传统的基于LPL的MAC协议相比，X-MAC更易于被支持分组无线发射器的无线传感器节点所实现。

2008年，莱斯大学孙严军等人提出了RI-MAC协议，RI-MAC协议是接收方主动发起通信的异步占空比MAC协议。在异步占空比的情况下，最大化的减少了发送和接收节点在建立通信时占用信道的时间，提高了信道的利用率。使用接收方主动发起通信数据传输模式实现不同流量负载下的有效节能传输。同X-MAC相比,RI-MAC在变化的流量负载下提高了吞吐量、提高了传递率、降低了能耗。

2011年，莱斯大学唐磊等人突出了PW-MAC协议，PW-MAC协议引入了预测唤醒机制，需要通信的两个节点同时醒来，同RI-MAC相比减少了发送的能耗。为了达到精准预测，PW-MAC提出了请求预测错误纠正机制，其可以有效的解决不可预测的硬件延时和操作系统延时以及时钟漂移与时钟抖动引起的预测错误。新颖的预测重传机制可以减少由于干扰和碰撞带来的重传开销。PW-MAC同RI-MAC相比，进一步减少了能耗。

1.3 研究内容与设计指标

1.3.1 研究内容

无线传感器网络中根据流量特征将应用分为两类^[31]。一类是时间触发应用，即传感节点周期向基站汇报自己采集到的数据；另一类是事件触发应用，即传感节点只有感知到相关事件时才向基站汇报。

在事件触发应用中，节点会长时间工作在低负载情况，当一个突发事件产生时（如敌人攻击、

火灾检测），节点被激活开始工作。在这种情况下，节点会感知到大批量的数据，因此需要高吞吐量通信信道及时发送突发的数据。多个节点同时发送数据会产生严重的碰撞，导致丢包率增加、信道吞吐量下降，从而增加了能耗。

本课题提出了一种新颖的事件触发应用解决方案，我称其为以发送方为中心的数据传输模式。

主要研究内容如下：

- ◆ 流量突发时的低功耗通信模式；
- ◆ 降低数据传输延时的通信策略；
- ◆ 减少丢包率的节点调度机制；
- ◆ 广播及路由支持；

1.3.2 设计指标

本次课题采用NS2仿真平台对协议性能进行验证和测试。NS2是一种针对网络技术的源代码公开的、免费的软件模拟平台，研究人员使用它可以很容易的进行网络技术的开发，而且发展到今天，它所包含的模块几乎涉及到了网络技术的所有方面。因此，学术界使用NS2研究和开发协议进行网络行为模拟和性能仿真的人员较多，每年在国内外发表的与网络技术相关的学术论文中，利用NS2给出模拟结果的文章非常多，通过这种方法得出的研究结果也被学术界广泛认可。

衡量MAC协议性能的指标包括能耗、传输延时、吞吐量、丢包率、公平性等。本课题实现的MAC协议对比对象为PW-MAC（目前学界提出了性能最优的单信道异步MAC协议），对比场景为典型的基于事件驱动应用场景，即随机拓扑随机相关事件突发场景。通过理论分析与以往经验，提高指标如下：

- ◆ 当感知半径<20米时，能耗降低10%，数据传输延时接近；
- ◆ 当感知半径<50米时，能耗降低30%，数据传输延时降低20%；
- ◆ 当感知半径<150米时，能耗减低50%以上，数据传输延时降低40%以上；

1.4 论文的主要内容与结构

本文是针对无线传感器网络中事件驱动应用设计的低功耗异步 MAC 协议解决方案。旨在减少突发流量下节点间传输数据的干扰，降低节点能耗。以下是本论文的内容安排。

第一章：本章是绪论，主要介绍了无线传感器网络的发展趋势及节点低功耗要求。无线传感器网络 MAC 协议对网络性能的影响至关重要，因此本章对国内外 MAC 协议的现状及分类进行归纳和总结。最后提出了本论文的研究目的、研究内容及预期的性能指标。

第二章：主要介绍了什么是 MAC 协议，以及在无线传感器网络 MAC 协议研究中需要重点考

虑的因素。然后对无线传感器网络异步 MAC 协议进行详细分析。总结出优点及缺点，为本课题异步 MAC 协议设计奠定良好的理论基础。

第三章：首先分析本课题所针对的事件驱动应用场景特点，归纳总结研究重点。然后针对研究重点，分析现有 MAC 协议的不足。最后详细说明本课题提出的低功耗异步 MAC 协议设计方案。

第四章：介绍了 NS2 仿真平台，并详细说明了本论文方案在 NS2 平台的实现过程。

第五章：对本课题设计的面向事件驱动应用的低功耗异步 MAC 协议进行了仿真平台性能评估，给出仿真结果与分析。

第六章：总结与展望，对整篇论文进行总结，并对下一步的工作进行了展望。

第二章 无线传感器网络 MAC 协议

本章内容以无线传感器网络 MAC 协议为主线，首先对其进行概述，说明 MAC 协议的作用。然后介绍在无线传感器网络 MAC 协议设计中需要考虑的重点因素。最后异步 MAC 协议发展为顺序，一一介绍其原理并分析其优缺点。为本课题异步 MAC 协议设计奠定良好的理论基础。

2.1 无线传感器网络 MAC 协议概述

介质接入控制（MAC）协议是无线通信中一项重要的技术。其目的是在共享介质网络中，通过一组规范、准则使设备间有效、有序的使用通信介质，保证网络成功运行。MAC 协议的一项基本任务就是避免干扰节点间的碰撞。在传统无线声音数据网络中已经有很多 MAC 协议应运而生。典型协议有无线信道时分复用多址接入方式(TDMA)、码分多址接入方式(CDMA)以及与 IEEE802.11 类似的基于竞争协议。在无线传感器网络中，MAC 协议处于整个网络协议栈的底层，同传统 MAC 协议类似，它也是规则的制定者，负责协调多个传感器节点之间的信道资源分配与利用。但是同传统 MAC 协议不同的是，在无线传感器网络中，节点间大多采用电池供电，且节点的计算能力、处理能力和通信能力都存在很大的局限性。因此，需要针对无线传感器网络的特点，考虑影响协议性能的因素，进而研究出高效的 MAC 协议。

2.2 无线传感器网络 MAC 协议设计影响因素

1) 能量效率

无线传感器网络中，节点的能量通常由电池供应。人类可能很难到达监测区域，而且由于节点众多，更换电池工作量十分大。因此，对传感器节点进行能量补充十分困难。事实上，理想的无线传感器节点成本应该足够低，以便人们不需要对其进行能量补充。所以，延长网络中节点的生存周期至关重要，是无线传感器网路 MAC 协议研究的首要任务。

2) 数据可靠性

无线传感器网络广泛应用于工业、医疗等领域，数据的可靠传输是系统正常运行和病人生命安全的重要保证。然而由于外部干扰的存在，数据传输的可靠性很难得到保证。当前主流的无线传感器网络都工作在免授权的 ISM 频段，而目前 ISM 频段已被 WiFi、蓝牙等多个无线网络共享。采用 ISM 频段的无线传感器网络在家庭护理、独居老人监护等方面的应用将受到来自其它无线网络的严重干扰。

3) 可扩展性

由于节点能量耗尽、节点移动或者是新节点的加入等因素，会导致网络中节点规模的大小、区

域节点密度和拓扑等变化。因此，无线传感器网络 MAC 协议需要具有良好的可扩展性，当网络结构发生变化时，MAC 协议可以适应以上变化。

2.3 无线传感器网络异步 MAC 协议发展现状

2.3.1 B-MAC

无线传感器网络中节点的典型特征是使用电池供电，并且在无人管理的情况下长时间工作。空闲侦听是这些节点能耗损失的一个重要因素。空闲侦听是指传感器节点打开收发机，即使没有其它节点与其通信，也持续侦听可能到来的数据包。

解决空闲侦听的一个主要方法是占空比工作机制。在此种工作机制中，每个传感器节点周期性的打开收发机，在工作状态与睡眠状态之间切换。例如，在 15% 的占空比工作模式中，一个节点只有 15% 的时间打开收发机，这样就可以显著减少空闲侦听能耗。在工作状态下，节点进行发送和接收数据。而在睡眠状态下，节点完全关闭其收发机来节省能耗。

基于竞争的占空比 MAC 协议可以分为同步 MAC 协议和异步 MAC 协议两大类。在同步 MAC 协议中，邻居节点是相互同步的，即邻居节点同时在工作状态与睡眠状态之间切换。在工作状态下，邻居节点都处于唤醒中，进行数据交互。在睡眠状态下，邻居节点都关闭收发机节省能耗。此种工作方式可以大大的减少空闲侦听，但是同步邻居节点的需求需要额外控制开销和复杂的处理机制。此外，如果一个节点的邻居在不同的调度表中，此节点需要唤醒多次，浪费能耗。

异步协议的提出，解决了同步协议的同步问题。异步 MAC 协议中节点相对独立，每个节点拥有自己的占空比工作调度表。

2004 年，加州大学伯克利分校的 J. Polastre 等人提出了 B-MAC (Berkeley MAC) 协议。B-MAC 是一个基于 CSMA 技术的异步 MAC 协议，利用低功耗侦听技术 (LPL) 和长前导包模式完成低功耗异步通信。节点拥有唤醒和睡眠周期，且每个节点拥有独立的时间调度表。如图 2.1 所示，如果一个节点想要传输数据，首先发送一个超过接收节点睡眠周期的长度的前导包。在唤醒状态，节点对信道进行采样，如果检测到前导包的存在，它就保持唤醒并接收数据。在发送长前导包过程中，发送节点确信接收节点会在期间的某一时刻唤醒，检测到前导包，并保持唤醒等待接收数据。B-MAC 同时提供了一个时间调度调整窗口。应用者可以根据流量的变化调整节点的占空比。这种办法显著减少了能耗，并且没有同步开销。

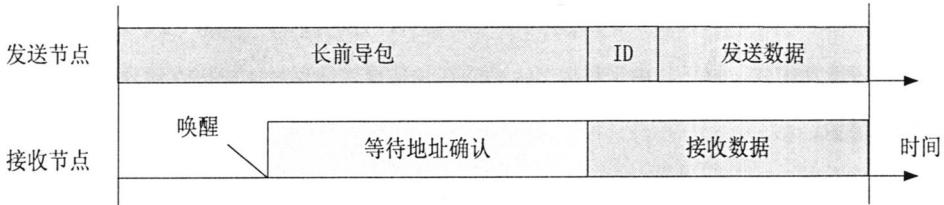


图 2.1 B-MAC 协议传输机制

B-MAC 通过 LPL 技术使节点的收发机周期性的进行信道采样。每当节点唤醒时，它就打开收发机检测信道状态。如果检测到活动状态，节点就完全进入工作状态，保持唤醒并等待接收数据。在接收数据完成后，节点转换到睡眠状态。如果在此期间没有接收到数据包，那么定时器超时并告知节点转换到睡眠状态。这种 LPL 是使用空闲信道评估方法（CCA）来完成的。图 2.2 展示了 CCA 的算法。最上方图显示了收发机的接收信号显示轨迹，一个数据包从第 22ms 到来，结束于第 54ms。中间图显示了 CCA 算法的输出，如果信号强度超过阈值，就判定为信道忙。如果信号强度小于阈值，就判定信道空闲。最下图显示了输出结果，其中 1 表示信道空闲，0 表示信道忙。

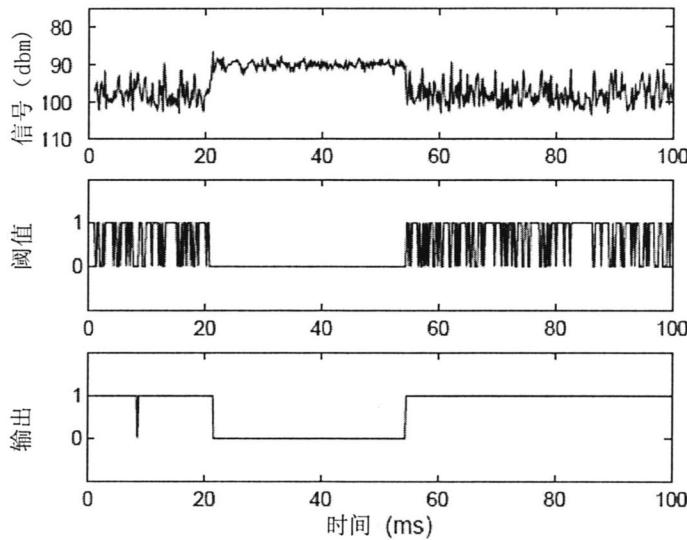


图 2.2 CCA 算法示意图

2.3.2 X-MAC

B-MAC 中提出的低功耗侦听技术使网络中节点相对独立，减少了同步开销及控制复杂度。但是低功耗侦听引入的长前导包有许多缺点：发送节点和接收节点的能耗损失比较大；非目标节点的串音能耗依然很高；每跳延时较高。首先，即使接收节点同发送节点一起唤醒，二者也必须在长前导包发送结束后，进行数据/确认的交互过程，增加了发送节点和接收节点的能耗。第二，低功耗侦听技术带来了串音问题，即非发送节点的目标节点在检测到信道中有前导包后，也必须一直等待到

前导包发送结束后，才发现自己并非接收节点，然后才转入睡眠状态。这会使发送节点的所有邻居节点产生串音能耗损失。第三，由于目标节点必须在长前导包结束后才能接收到数据包，因此每跳的延时显著增加。

2006年，科罗拉多大学波尔得分校的 Michael Buettner 提出了 X-MAC 协议，X-MAC 利用短前导技术进一步减少节点能耗和数据传输延时。其思想是在前导包中加入目标节点地址，使非目标节点快速转换到睡眠状态以减少串音能耗。使用短前导允许接收节点在唤醒之后就可以中断前导传输，并得知自己是目标节点。短前导减少了长前导包带来的能耗和延时的浪费。

如图 2.3 所示，X-MAC 将 B-MAC 中的长前导包，分割成多个短前导包，并且每个短前导包都包含目标节点的地址，以此来避免串音问题。当一个节点唤醒并接收到短前导包时，首先观察包含在前导包中的目标地址。如果自己不是目标节点，那么就转换到睡眠状态，并根据自己的占空比调度进行工作。减少了串音能耗浪费。

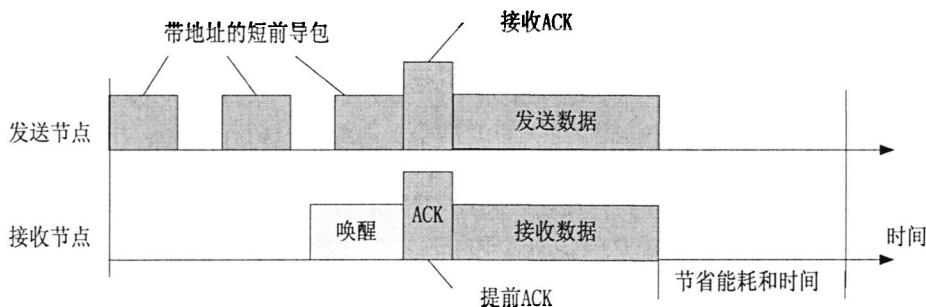


图 2.3 X-MAC 协议传输过程

在 B-MAC 协议中，尽管发送节点每次都发送一个完整的长前导包，平均来讲，接收节点在接收前导包时，唤醒的时间为半个前导时长。由于发送节点不知道接收节点什么时候唤醒，因此它必须发送整个前导包。这样会增加传输延时，另一种情况是多个发送节点都需要与同一个接收节点进行通信。当第一个发送节点竞争到信道后，其它节点只有等待信道空闲时，才发送前导包，以此类推。结果，接收节点会长时间唤醒。

为了解决以上问题，X-MAC 在短前导包中插入间隔，在此期间发送节点用来侦听信道。在这个间隔中，接收节点可以在接收到前导包后，发送提前 ACK 帧。当发送节点接收到目标节点的提前 ACK 帧后，发送节点停止发送前导包，开始发送数据。此机制可以减少多余前导，进而减少延时和能耗浪费。

2.3.3 RI-MAC

B-MAC 和 X-MAC 都属于基于 LPL 前导包传输的异步协议，在低流量负载情况下，能量效率较高。但是它们的前导包占用信道时间较长，远远超过了实际数据传输的时间。如图 2.4 所示，一

个发送节点的前导包，阻止了其它邻居节点的数据传输。这些节点不得不等待信道空闲，才可以进行数据传输。大大增加了节点能耗和数据传输延时。这一情况经常发生在突发传输或高流量负载环境下。在进行汇聚传输^[32-34]和事件驱动应用^[35-36]中，多个传感节点检测到相同事件会同时发送给终端节点或汇聚节点。由于在无线传感器网络中，流量可能是动态的，因此，一个完美的 MAC 协议应该适用于突发流量传输和高负载环境。此外，X-MAC 协议还存在隐藏终端前导碰撞问题。如图 2.5 所示，发送节点 2 首先发送前导包，由于发送节点 1 距离发送节点 2 较远，未能检测到信道正在被占用，因此，发送节点 1 随后也开始发送前导包。这样就导致接收端无法正确接收到前导包。因此接收节点 1 和接收节点 2 都无法正常接收到前导包进行数据接收。

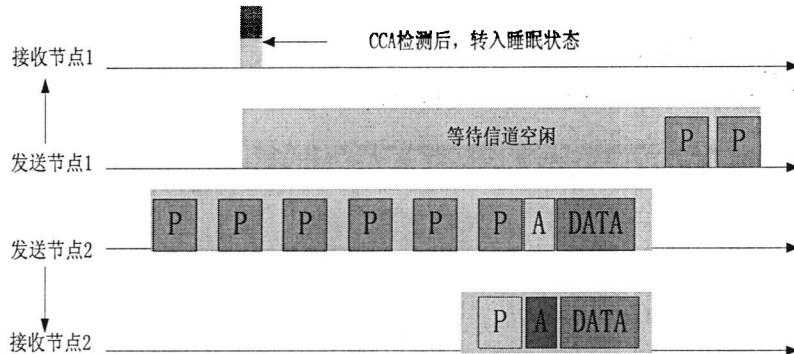


图 2.4 长前导影响邻居节点通信

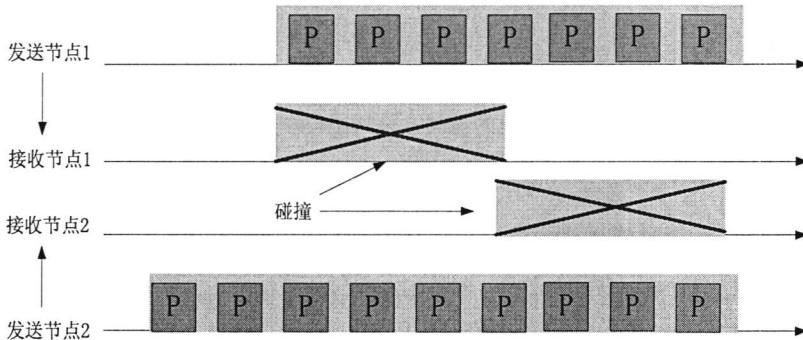


图 2.5 隐藏终端前导包碰撞问题

2008 年，莱斯大学孙严军等人提出了 RI-MAC 协议。RI-MAC 试图减少发送节点和接收节点进行数据交互预约时，占用信道的时间，同时是发送节点和接收节点的占空比工作调度表独立。RI-MAC 与 B-MAC 和 X-MAC 不同，改变了发送节点与接收节点进行数据交互预约的方式。在 RI-MAC 中，发送节点有数据需要发送时就唤醒并进行信道侦听，而不是发送前导包。当它接收到目标节点发送的信标帧之后，开始进行数据传输。在一次数据交互中，只有信标帧和数据时长占用信道，大大减少了前导包占用信道的时间，以便于其它节点进行数据交互。

图 2.6 给出了 RI-MAC 的传输过程。每一次数据传输都是由目标节点发起的。每个节点根据自

己的占空比调度周期的唤醒来检查是否有其它节点需要与自己通信。打开收发机以后，如果信道空闲，节点就广播一个信标帧，表示自己已经唤醒并等待接收数据。如果发送节点有数据需要传输，就保持唤醒状态，等待接收目标节点的信标帧。在接收到信标帧后，发送节点开始发送数据，然后目标节点会发送确认帧。值得注意的是，这个确认帧有两个作用，首先是确认数据已经接收完成。第二是可以进行新的数据传输。如果节点在广播信标帧以后没有接收到数据，那么节点就进入睡眠状态。同 X-MAC 相比，RI-MAC 显著减少了预约传输数据的信道占用时间。这个优点可以使其它节点尽快传输数据，减少传输延时。同时，由于接收节点在发送信标帧后只唤醒很少一段时间。因此，减少了接收节点的串音问题。

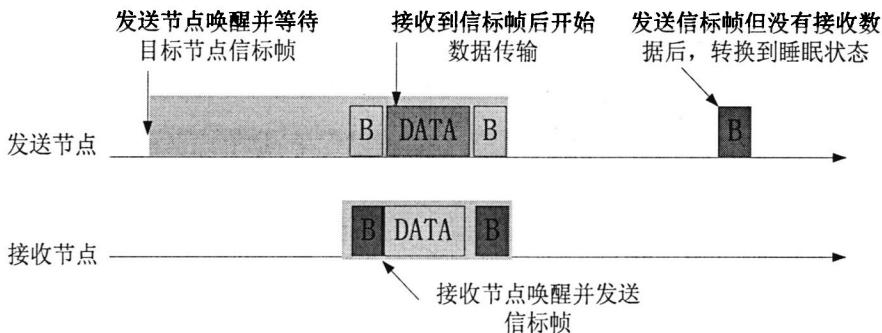


图 2.6 RI-MAC 协议传输过程

2.3.4 PW-MAC

RI-MAC 引入了接收端主动发起通信的机制，使用接收端主动唤醒信标帧避免了发送端主动通信前导包的使用。此种方法提高了信道利用率及更高效的碰撞检测。在 RI-MAC 中，每个节点通过周期发送信标帧表明自己唤醒。发送节点在接收到目的节点的信标帧后进行数据传输。然而，在 RI-MAC 中，当发送节点需要传输数据时，就立即唤醒等待目标节点发送信标帧，导致了发送节点空闲侦听能耗增大。

2011 年，莱斯大学唐磊等人突出了 PW-MAC 协议，PW-MAC 在 RI-MAC 的基础，使用了独立伪随机序列控制每个节点的唤醒时间，使发送节点可以准确的预测目标节点的唤醒时间。两个节点同时唤醒完成数据传输。同时降低了发送节点和接收节点的能耗。

PW-MAC 的目的是使发送节点在接收节点唤醒时，同时唤醒并打开收发机。为了使发送节点可以准确的预测接收节点唤醒时间。PW-MAC 使用伪随机唤醒机制而并非真的随机调度。相比使用固定调度，伪随机调度可以避免邻居节点的持续同时唤醒，减少信道碰撞。

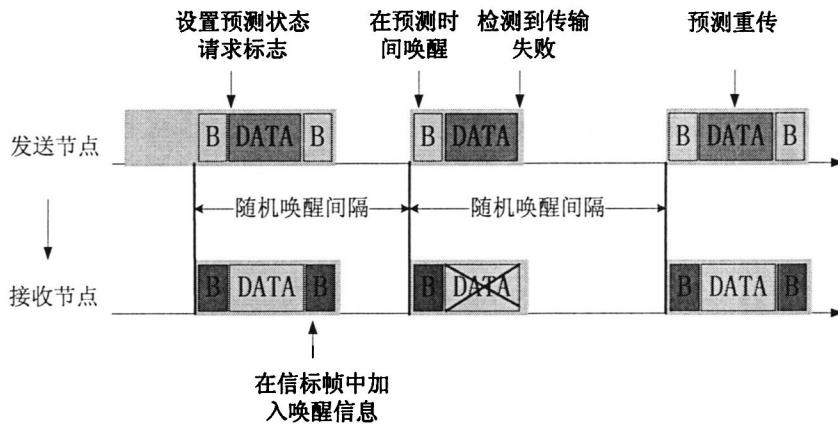


图 2.7 PW-MAC 协议传输过程

如图 2.7 所示，描述了 PW-MAC 的预测唤醒机制。同 RI-MAC 一样，每个节点周期唤醒并广播一个信标帧（图 2.7 中“B”），表示自己已经处于唤醒状态，并准备接受数据帧。在 PW-MAC 中，如图 2.7 中接收节点，周期唤醒间隔由伪随机序列产生。PW-MAC 中的发送节点获得或更新邻居节点的预测状态时按需进行的。如果发送节点需要向接收节点发送数据，但是其并不知道接收节点的预测状态信息。那么发送节点就打开收发机并等待接收节点的信标帧。在接收到接收节点的信标帧后，当发送节点发送数据帧时，在数据帧的包头部分设置一个特殊标志位，表示需要获取接收节点的唤醒信息。当接收节点接收到这个数据帧时，接收节点发送另一个信标帧，起到确认接收作用，并允许下一个数据接收。为了回应发送节点的预测状态请求，接收节点将自己的“当前时间”及“预测状态”装载在这个信标帧中。接收节点的“当前时间”用来计算二者的时钟偏差。为了精准的确定时钟偏差，在接收节点发送信标帧时，立即装载在其中。通过获取确认信标帧中的预测信息，发送节点就可以预测接收节点将来的唤醒时间。将来，如果发送节点还有数据要发送给该节点，就可以在接收节点唤醒前一刻醒来。PW-MAC 显著减少了空闲侦听时间。

PW-MAC 同时提出了基于预测的重传机制。如果发送节点接收到接收节点的信标帧，但是没有接收确认信标帧，那么发送节点认为数据包或者确认信标帧传输失败。发送节点因此转入睡眠状态，并在接收节点下次唤醒时间进行数据重传，因此，减少了等待接收节点的能耗。PW-MAC 虽然减少了节点传输預占用信道的时间，以及空闲侦听的等待能耗。但是，面对无线传感器网络的突发负载网络状况，在 PW-MAC 基础之上，异步 MAC 协议仍需进一步改善。这也是本课题开展的工作。下面将介绍面对突发负载情况下，现有 MAC 协议解决传输碰撞的方法。

2.4 本章小结

本章首先介绍了 MAC 协议的功能，它负责信道接入、控制节点的报文传输对无线共享媒体的

占用以及通过一组规则和过程来更有效、有序和公平使用共享媒体以保证网络的整体性能。然后分析了无线传感器网络 MAC 协议研究需要重点考虑的因素如能量效率、数据可靠性及可扩展性。最后介绍了无线传感器网络异步 MAC 协议。以时间为顺序，典型的异步 MAC 协议发展为 B-MAC、X-MAC、RI-MAC、PW-MAC。介绍了每个协议提出的机制，解决的问题，以及存在的局限性，为本课题的研究必要性奠定基础。

第三章 SC-MAC 协议的设计

本章内容是本论文的核心。主要研究了一种以发送方为中心(Sender-Centric)的异步低功耗 MAC 协议—SC-MAC。本章内容组织如下，首先对事件驱动应用场景的特点进行分析，总结出设计重点。并针对研究内容分析现有协议的不足。最后对 SC-MAC 协议进行概述，并介绍各个机制具体设计过程及通信流程。

3.1 事件驱动应用场景特点分析

在事件驱动应用网络中，传感器节点的首要任务就是向汇聚节点汇报监测信息。通常情况下，这些传感器节点长时间工作在低流量负载环境，当“感兴趣”的事件发生时，网络会突然活跃并进入突发负载或高流量负载状态。在这种情况下，感知到相关事件的多个节点会产生大量数据，并在很短的时间内将数据汇报给汇聚节点。由于这个“感兴趣”的事件会导致节点能耗和传输延时的上升。因此，设计适用于事件驱动应用的无线传感器网络 MAC 协议应满足以下两个条件：(1) 在空闲状态下，节点保持低功耗工作；(2) 在突发负载或高流量负载环境下，节点应在低功耗的前提下，快速传输数据。

此外，如图 3.1 所示，当“感兴趣”事件发生时，在网络中靠近事件源端区域内的节点活动频繁，数据的传输碰撞情况严重。在远离事件源端处，数据的传输碰撞情况减少。因此，在设计基于事件驱动应用 MAC 时，应考虑事件近端和事件远端的区分对待，综合考虑问题并分层优化。

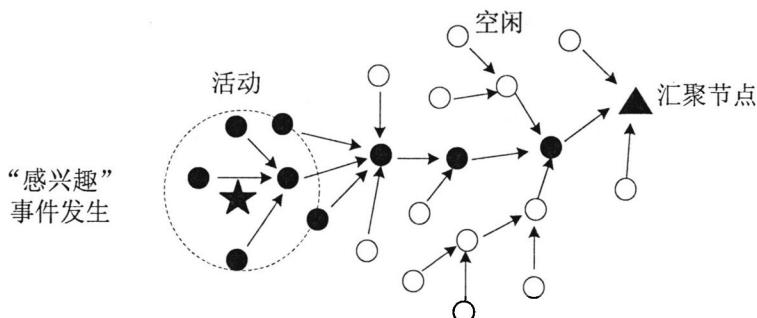


图 3.1 事件驱动应用示例

3.2 现有解决传输碰撞方法及比较

传统的无线传感器网络 MAC 协议通常使用指数退避方法和 TDMA 方法解决竞争节点问题。另两种典型方法分别是 RC-MAC 和 Flip-MAC 中提出的前端解决方案，下面具体分析每种机制并总结出其

优缺点。

3.2.1 PW-MAC

PW-MAC中指出处理多竞争发送节点传输的挑战是减少接收节点的活跃时间和减少碰撞检测和丢包重传的开销。为了实现以上目标，PW-MAC中使用beacon帧协同竞争节点间的传输。发送节点接收到未包含退避窗口（BW）的beacon帧（我们称为基本beacon帧）时就立即发送数据；如果接收到包含BW的beacon帧，就根据BW域的参数选择自己的真实退避时间，然后再发送数据。具体传输过程如图3.2所示，发送节点1和发送节点2同时收到接收节点发送的基本beacon帧后同时发送数据，此时检测到碰撞的接收节点会发送一个含有BW的beacon帧，各发送节点会根据BW选择再次发送数据的时间，从而避免碰撞。

此种方法可以简单且有效的解决少数节点竞争信道的问题，但其弊端是当竞争节点增多时，发生持续碰撞的概率呈指数上升。最大化的退避窗口增大了所有活动节点的能耗和数据传输延时，MAC协议的性能迅速降低。

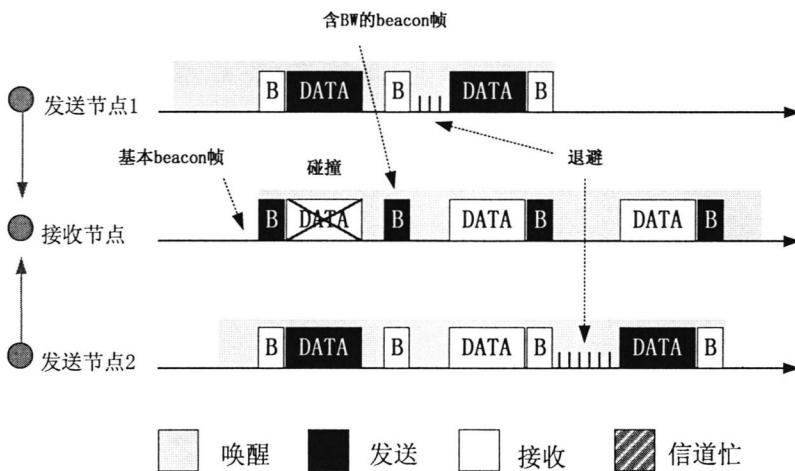


图3.2 PW-MAC传输碰撞过程

3.2.2 Flip-MAC

Flip-MAC^[37]的数学依据是二项分布。具体传输过程如图 3.3 所示，首先，接收节点发送 probe 询问是否有节点要向自己发送数据包。如果有，所有发送节点进行“抛硬币”过程：每个发送节点都会在两个备选地址中选择一个作为自己的伪装地址。如果同接收节点选取的随机地址匹配，则进入下一轮筛选；不匹配，则淘汰，不参与随后的信道竞争。这样进行多次筛选，直到没有发送节点与接收节点选取的随机地址匹配；筛选过程结束后，就进入确认阶段，倒数第二次（最后一次无匹配）地址匹配过程中，没有被淘汰的发送节点作为最后的信道竞争者。然后进行正常的 CSMA 过程。

从拓扑上来分析，一个单跳汇聚节点在实际应用中会有多少邻居，只有在邻居数目多的情况下，才会体现出其优势。在多跳大范围场景中，节点既作发送者又作为接受者，就会出现同时协商（concurrent negotiation），同时就应考虑路由情况。从发送数据的类型来看，若是多个节点感知到相同数据上传（例如监测到异常事件），那就完全可以通过简单的方法进行协商，选择一个 Leader 发送。若是不同数据均需要上传，那么恰巧有一些节点每次都没“竞选”上，这就很棘手。在周期汇报应用中，多个节点同时上传的概率相对较小。

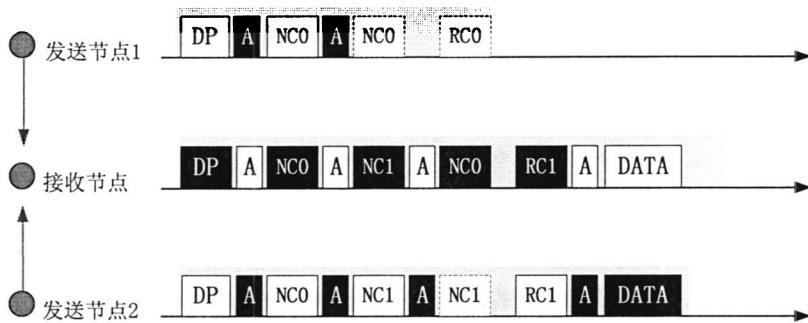


图3.3 Flip-MAC传输碰撞处理过程

表3.1 数据包缩写及含义

| 缩写 | 含义 |
|-----|---------|
| DP | 需要发送数据？ |
| NC0 | 选择随机地址0 |
| NC1 | 选择随机地址1 |
| RC0 | 确认随机地址0 |
| RC1 | 确认随机地址1 |

3.2.3 RC-MAC

如图3.4所示，RC-MAC^[38]中，所有竞争发送节点在接收到接收节点的beacon后进行随机退避（CW），在开始发送数据前通过清晰信道检测（CCA）判断信道是否空闲。接收节点接收到数据包后，在回复的ACK中包含下次调度发送节点的地址（ID）。该节点被允许可以直接发送数据，其它竞争节点需要随机退避后竞争信道。

此种方法在一定程度上加快了数据传输，但在能耗方面仍没有改善。

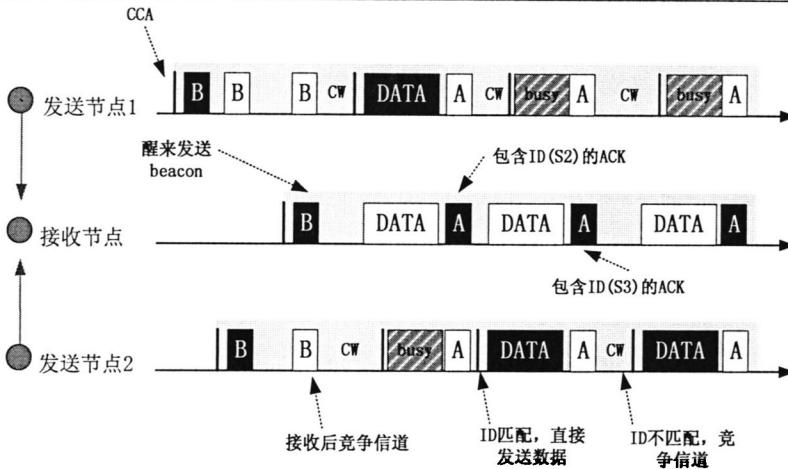


图3.4 RC-MAC传输碰撞处理过程

3.2.4 TRAMA

TRAMA 是 Rajendran 等人提出的典型的基于 TDMA 的 MAC 协议，能够实现单播、组播、广播模式下的高效无冲突信道访问，从而降低能耗。TRAMA 协议中的能量效率主要通过两种方式得到：调度发送节点的传输过程，避免在接收节点处发生冲突；使节点在没有数据包需要发送时切换到睡眠模式，避免把时槽分配给没有数据需要发送的节点。虽然 TRAMA 可以实现无冲突的信道访问，但是协议设计过程复杂。而且，TDMA 类的协议都需要严格的时间同步，时槽分配和机制维护，这些控制开销非常大，从而浪费能量。当节点密度比较大时，控制消息碰撞也会造成协议性能降低。

3.2.5 协议小结及对比

上述内容介绍了每种 MAC 协议在应对传输碰撞时所采用的方法。下面将对每种协议进行核心思想的概括，并指出每个协议的优缺点。

表 3.2 协议小结及优缺点分析

| 协议名称 | 核心思想 | 优点 | 缺点 |
|----------|---------------------|----------------|----------------------------|
| PW-MAC | 通过指数退避后重新竞争信道 | 算法简单，适用于少数节点竞争 | 竞争节点增多时，节点持续唤醒，浪费能耗，且公平性降低 |
| Flip-MAC | 通过二项分布逐渐减少同时竞争节点的数目 | 适用于大规模节点竞争情况 | 算法复杂，不易实现。决策时间长，数据延时增加 |
| RC-MAC | 接收节点发放令牌，指定发送节点 | 信道利用率高，数据传输快 | 低优先级发送节点持续唤醒，浪费能耗 |
| TRAMA | 时分复用 | 创造无竞争传输环境 | 需要严格的时间同步、时槽分配算法等。控制开销多。 |

3.3 现有协议所面临问题深入分析

通过上一小节对每个 MAC 协议进行了协议机制的介绍。可以看出，现有 MAC 协议在处理传输碰撞时，主要有以下不足之处：时间同步问题、竞争节点唤醒等待问题、节点公平性问题和异步广播支持问题，下面进行详细分析。

3.3.1 时间同步问题

基于调度的 TDMA 机制是实现竞争节点无碰撞传输的最直接办法。但是这类协议通常需要严格的时间同步，同时需要额外控制开销来维持节点同步。时槽分配算法和调度维护也是不可缺少。选择一个合适有效的调度软硬件开销和复杂度都比较大，另外节点失效等许多因素原因导致的拓扑变化比较频繁，在基于调度的 TDMA 类协议中要维持动态拓扑变化开销也比较大。此外，空闲时槽的存在导致 TDMA 协议信道利用率较低。

3.3.2 竞争节点持续唤醒问题

同 TDMA 类协议相比，基于竞争的 CSMA 机制相对灵活且简单。不需要同步及拓扑信息的支持。节点加入和离开网络较为自由。然而，当多个邻居节点同时产生大量数据时，未成功竞争到信道的节点持续唤醒，直到竞争成功并完成数据传输才进入睡眠状态。当竞争节点数量较少时，其它节点的持续唤醒时间较短，无需频繁切换工作/睡眠状态。但是当竞争节点数量增多时，发生持续碰撞的概率呈指数上升。最大化的退避窗口增大了所有活动节点的能耗。

3.3.3 节点公平性问题

传统网络中节点公平性一般是为了确保所有节点的数据可以公平的享用网络带宽，同时解决网络拥塞。在基于事件驱动应用场景中，这里的节点公平性是指竞争节点的平均持续唤醒时间问题。在竞争信道时，如果某一些节点持续的竞争不到信道资源。就会导致唤醒时间较长，增加能耗，导致节点死亡，拓扑改变的问题。同时，由于长时间竞争信道失败，数据实时性难以保证，这在目标跟踪和实时监测应用中问题较为突出。

3.3.4 异步广播支持问题

多跳广播在无线传感器网络中是一项很重要的服务，会应用到路由发现、网络询问和信息分发等。在异步占空比工作 MAC 协议中，由于每个节点独立随机唤醒，因此实现单跳广播传输非常困难。一般需要源节点发送多次数据，导致冗余传输。现有面向全唤醒工作 ad hoc 网络的广播协议对

于无线传感器网络并不适用。如应用到多跳广播，源节点和转发节点都会持续发送数据包，增加无线碰撞，问题将会更加放大。因此，需要在单播传输机制的基础上，提出切实可行的多跳广播机制。

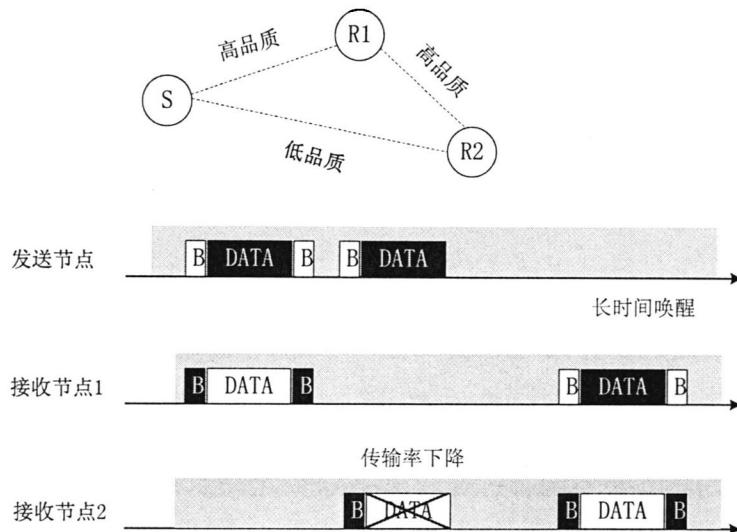


图 3.5 异步占空比工作广播挑战

在异步占空比工作条件下，实现高能效、低延时、高可靠的多跳广播非常有挑战性。如图 3.5 所示，发送节点的多个邻居都在不同时间唤醒，此时需要发送节点长时间保持唤醒以保证每个邻居都有机会接收到广播包，导致能耗增加。其次，低链路品质传输会导致接收率下降，以及传输延时和能耗的上升。当一次传输失败后，接收节点会进入睡眠状态。如果发送节点需要重传数据就必须等到接收节点的下一个唤醒周期。最后，广播进程信息对于一个节点避免冗余传输十分重要。但是刚刚唤醒的节点并没有更新的传播进程信息。由于节点关闭收发机后，传播进程可能会改变，因此简单的利用串音获得进程信息并不可靠。

3.4 一种针对事件驱动应用低功耗异步 MAC 协议的提出

本节提出了一种以发送方为中心(Sender-Centric)的异步低功耗 MAC 协议—SC-MAC。旨在解决无线传感器网络中事件驱动应用中，由于突发流量多节点碰撞导致的能耗浪费问题，同时解决关于时间同步、节点公平性和广播支持等问题。本节首先对 SC-MAC 协议进行概述，然后针对具体问题详细说明解决方案。

3.4.1 SC-MAC 协议概述

图 3.6 给出了 SC-MAC 的基本传输过程，其中，有数据需要传输的发送节点根据自己的时间调度表与接收节点进行通信，而不是在接收节点唤醒时，同时向接收节点发送数据。在 SC-MAC 中，

接收节点周期唤醒，并广播询问帧检查是否存在潜在通信。接收节点广播询问帧后，如果接收到硬件 ACK (HACK)，那么就认为有数据要向自己发送，这时接收节点立即睡眠，并根据邻居链表存储的各发送节点调度信息依次醒来接收数据。相反，如果接收节点在广播询问帧后未接收到 HACK，那么它就立即睡眠，直到下一个询问周期（backcast 间隔）的开始才唤醒并广播询问帧。相应的，在 SC-MAC 中，发送节点感知到“感兴趣”事件后，在接收节点广播询问帧的前一时刻醒来并打开收发机。在接收到询问帧后，发送节点回复 HACK 来提醒接收节点彼此需要进行数据通信。然后，各发送节点进入睡眠状态，并根据自己的发送调度表同接收节点进行通信。SC-MAC 使同时感知到事件的多个节点分时与接收节点进行通信，避免了数据碰撞，减少了节点能耗和数据传输延时。在 RI-MAC 和 PW-MAC 中所采用的二进制指数退避算法，不仅导致能耗增加，而且会增加数据包端到端的延时。在极端情况下，当竞争节点众多，最大退避限制会导致接收节点崩溃。另一方面，发送节点和接收节点只需要维护各自的调度表，开销十分小，而不需要过渡关注时间同步和其它网络维护开销。为了进一步减少数据包传输延时，我们在本章节后面同时提出了一种延时优化机制。

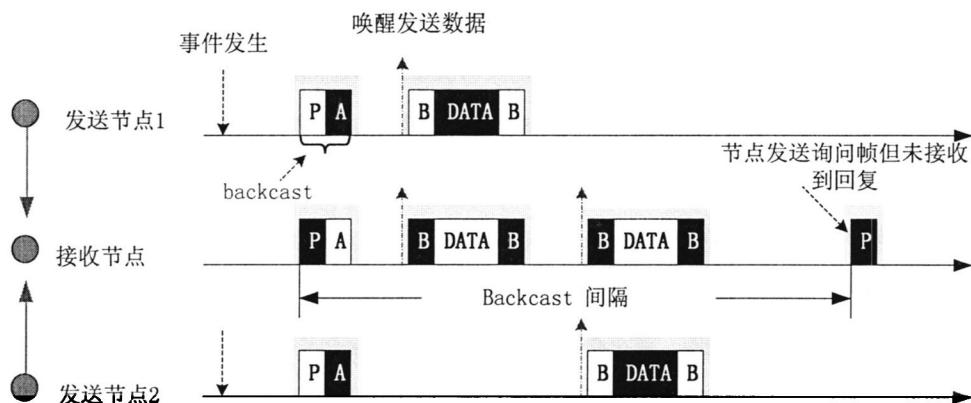


图 3.6 SC-MAC 协议概述

3.4.2 异步周期询问机制

Backcast 通信原语^[39]是一种数据链路层的帧交换机制。一个发起者的帧传输激励，会促使多个确认响应，并且在发送端可以无破坏接收。图 3.7 显示了 3 个节点的 backcast 交互过程。两个响应者具有自动确认功能。在 Backcast 交互中，首先由发起者发送一个带有硬件广播地址的询问帧。两个响应者都会接收到此询问帧并发送确认帧。尽管这两个确认帧会在发起者处发送碰撞，但是这种碰撞是无损的，可以使发送者正确的进行解码，并判断出至少有一个邻居进行了相应。

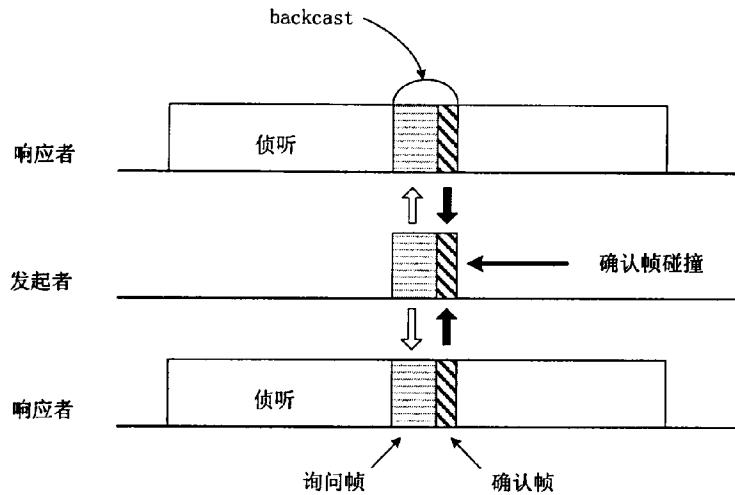


图 3.7 backcast 通信原语示意图

除了广播地址外，一个 backcast 询问帧可以带有多播地址或单播地址，使得只有一部分邻居节点进行相应。而对于目的地址的选择取决于接收机的能力及不同的通信服务。例如，在一个周期性睡眠的网络中，使用广播地址比较适用，而单播地址适用于同一个节点进行通信的场景。

一个 backcast 原语成功通信的关键在于无损的确认帧碰撞。此种情形由以下几种情况促成。能量捕获：一个确认帧的能量大于其它确认帧能量总和；延时捕获：一个确认帧达到时间比其它确认帧到达时间早；信息重捕获：此种情况是“包中包”模式，接收机发现能量突变而采取的重新同步或者接收机使用 OOK 调制模式。

为了验证 backcast 通信原语的可靠性，Dutta 等人使用基于 IEEE802.15.4 标准的 CC2420 进行了一系列的实验测试。测试数据表明，现在商用的收发机可以正确解码重叠信号的概率至少是 97%。图 3.8 展示了实测结果。在办公楼当中摆放测试床，并包含 47 个 Telos 节点。在实验中使用 12 个节点作为响应者。实验测试了三种回复机制对确认帧接收率的影响，分别为硬件确认（Hardware-generated ACKnowledgments）、软件确认（Software-generated ACKnowledgments）和带有随机前导的硬件确认（Vary-Preamble HACKs）。其中，HACK 是由接收机自动产生；SACK 需要通过处理器处理后产生，会产生延时抖动；而 VP-HACK 是比较，在发起端同时开始接收，但不同时间结束对接收率的影响。在实验室，每隔 125 毫秒发送一个数据包，共 500 个。结果显示了随响应节点数增加，链路质量显示（LQI）和确认帧接受率（ARR）的变化。数据表明，使用 HACK 的效果最佳，且无丢包。

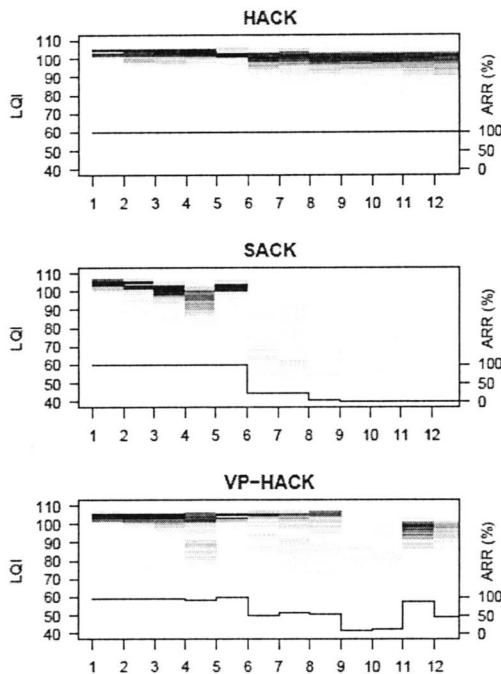


图 3.8 实测结果

Backcast 通信原语可以应用到许多服务，如低功耗异步唤醒、低功耗单播、预约服务等。在 SC-MAC 中，使用 backcast 通信原语完成单跳协作反馈。接收节点周期发送询问帧表示“是否有发送节点想和我进行通信？”相应地，感知到事件的多个节点在接收到这个询问帧后，同时回复 HACK 来表示“自己需要与接收节点进行通信”。通过上述单跳协作反馈，接收节点和每个发送节点在本次 backcast 周期建立了潜在的通信连接，并会根据发送节点的调度表进行数据交换。

大多数的 MAC 协议都采用固定或随机间隔实现周期唤醒/睡眠工作。在 SC-MAC 中采用固定间隔实现周期唤醒，接收节点会固定周期发送询问帧 P，在这个询问帧中包含了接收节点的两个特殊参数：下一次 backcast 周期的时间和周期间隔。当接收到询问帧后，发送节点会将这两个信息存储到一个邻居链表里，根据这两个信息，发送节点可以计算出接收节点将来每次 backcast 周期的时间。其中 backcast 周期的时间是网络的实时时间，而周期间隔可以根据用户需要进行调整。

3.4.3 预测唤醒传输机制

在突发流量负载下，为了营造出无碰撞传输环境，同时不引入复杂的操作和过多的控制开销，SC-MAC 中发送节点采用伪随机序列唤醒机制来确认与接收节点的开始通信的时间。尽管 PW-MAC 中节点也是使用伪随机唤醒机制，但二者在本质上是不同的。换句话说，PW-MAC 中的伪随机机制是以接收节点为中心，接收节点根据自己的调度唤醒接收数据，多个发送节点在感知到事件后，会同时向接收节点发送数据，产生激烈的碰撞。而在 SC-MAC 中，接收节点会根据每个发送节点的调

度信息，依次唤醒进行数据交互，这是一个以发送节点为中心的伪随机唤醒机制。

我们采用线性同余法（LCG）产生伪随机序列^[40]，其公式如下：

$$X_{n+1} = (aX_n + c) \bmod m. \quad (1)$$

其中， $m > 0$ 是模， a 是乘因子， c 是线性因子， X_n 是当前种子，产生的 X_{n+1} 会成为下一个种子。

为了保证所有发送节点在接收节点的一次 backcast 周期内至少唤醒一次，种子的最大值应小于 backcast 周期间隔。

在 SC-MAC 中，如果一个新节点加入网络而且有数据需要发送，那么它会一直保持唤醒，并等待接收节点的询问帧。在接收到询问帧后，这个新节点将询问帧中包含的接收节点下一次 backcast 周期的时间和周期间隔两个参数存储到自己的邻居链表里，然后直接发送数据，在这个数据中包含了一个 footer，footer 中含有发送节点的伪随机唤醒信息。接收节点接收到数据后，将 footer 中的发送节点伪随机唤醒信息存入一个邻居链表里，用来计算发送节点唤醒时刻。客观来讲，SC-MAC 仍会有一定的概率产生碰撞，碰撞概率与发送节点数目呈正比和 backcast 通信周期间隔呈反比。在这种情况下，我们使用二进制指数退避方法，进行退避重传。

3.4.4 广播及路由支持

为了应对异步广播问题挑战，本文提出了设计 SC-MAC 中广播机制的一些决定。首先，由于每个节点都是异步占空比工作，我们选择使用单播传输数据到每个唤醒的节点。确认帧帮助发送节点准确得知接收节点是否接收到广播包，以及使用重传增加广播包可靠性。其次，为了使节点减少唤醒时间，每个节点仍采用预测唤醒机制，完成对每个邻居的广播包传输。在 SC-MAC 中，接收节点和转发节点都会根据自己的调度表周期唤醒。然而，发送节点只有当需要传输数据时才唤醒。这就使得发送节点很难实现广播和完成路由选择。因此，在 SC-MAC 中，发送节点必须增加一个广播询问过程。而其它节点可以使用自己的 backcast 间隔作为广播询问过程或新建立一个广播询问过程。所有的节点都应将自己的广播询问周期参数告知邻居节点，以便完成广播。最后，为了减少冗余传输，SC-MAC 广播机制协调各节点获得广播进程信息，同时，包括各节点间的链路品质信息来避免低品质链路数据传输。

图 3.9 给出了 SC-MAC 中的广播机制具体流程。在这个简单的网络中，包括 3 个节点，发送节点、接收节点 1、接收节点 2，彼此都在通信范围内。此时，发送节点 S 需要广播一个数据包到其它节点。当接收节点 R1 唤醒后，S 像发送单播包一样将数据传输给 R1。但是，在数据包和确认帧中都包含了一个“广播进程及链路品质”信息。节点通过获知此信息来避免不必要的传输，以及决定是否转发此数据到没收到广播包的节点。如在此网络中，发送节点 S 的数据包中，包含了 R1 和

R2 都未接收到广播包的信息，以及与这两个节点间的链路品质信息。当 R1 接收到此数据包后，在确认帧中加入信息，表明自己已接收到广播包，且自己与 R2 之间的链路品质比 S 与 R2 之间的链路品质高。所以由 R1 转发广播包给 R2。S 在得知此信息后，发现自己的邻居都将接收到广播包，表明自己任务已经完成。这时立即转入睡眠状态。R1 通过预测唤醒信息，与 R2 同时醒来，并将广播包发送给 R2，同时在数据包中加入进程信息，表明 S 和 R1 都已经接收到广播包。这样当 R2 接收到广播包后，发现所有邻居都已经接收到此广播包，自己也转入睡眠状态。至此，一个广播包的传输过程完毕。从这个简单的例子当中，可以看出 SC-MAC 广播机制的一些特点：保证节点在所有邻居都接收到广播包后，或者已将任务分派给其它节点后转入睡眠状态节省能量；避免数据的低链路品质传输，提高传输率；节点间预测唤醒，提升网络寿命。

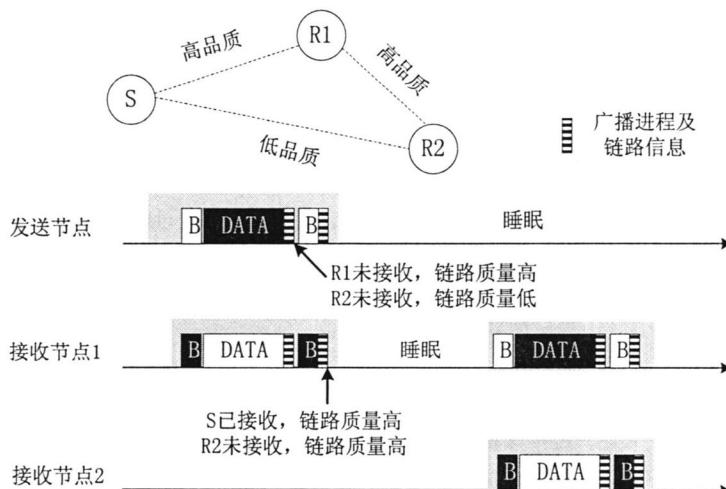


图 3.9 SC-MAC 协议广播机制

3.4.5 延时优化机制

如图 3.1 所示，碰撞经常发生在“感兴趣”事件附近，而远离事件区域的大部分转发节点在传输被监测的数据时不发生碰撞。如果在中转数据时，依然按照原始 SC-MAC 机制传输，那么转发节点会在发送 HACK 进行确认后转入睡眠状态，并按照自己的发送调度转发数据，这样会增加数据多跳传输延时。因此需要提出一种机制进行延时优化。

图 3.10 展示了 SC-MAC 延时优化通信机制。发送节点在接收到目标节点的询问帧后，产生自动硬件确认帧，并立即发送数据帧。如果成功发送，那么发送节点和接收节点都将进入睡眠状态。接收节点同样不需要预测本周期内邻居节点的唤醒时间。如果由于数据碰撞或其它干扰问题，接收节点未成功接收数据，那么接收节点就立即进入睡眠状态，并根据邻居节点的发送调度进行预测唤醒。由于没有接收到确认帧，因此，发送节点同样进入睡眠状态，并根据自己的发送调度进行重传数据。

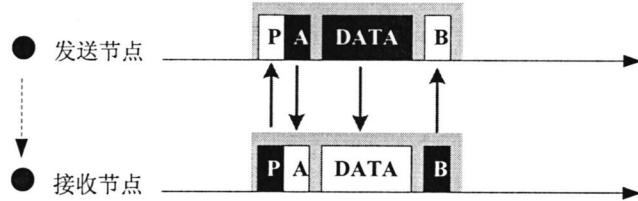


图 3.10 SC-MAC 协议延时优化机制

睡眠间隔决定了节点占空比大小，在 PW-MAC 中，占空比被设定为 0.5L 到 1.5L 之间的随机数（假设 L 为节点的睡眠间隔）。在 SC-MAC 中，backcast 间隔设定为 1L，以便使 SC-MAC 同 PW-MAC 在空闲时能量消耗一致。但是，从中可以看出，PW-MAC 的平均每跳延时为 $0.75L$ ，而 SC-MAC 的平均每跳延时为 $0.5L$ 。显然，同 PW-MAC 相比，SC-MAC 可以在每跳节省 $0.25L$ 的数据传输延时。

3.5 本章小结

在分析了事件驱动应用场景特点及现有协议不足基础上，本章对 SC-MAC 协议进行概述。它是一种针对无线传感器网络事件驱动应用的 MAC 协议。通过使用基于 backcast 通信原语的单跳协作反馈和伪随机唤醒传输机制，实现突发流量负载下，多个节点无竞争的数据传输。然后，介绍了在 SC-MAC 协议中的几个核心机制，如异步周期询问机制、预测唤醒传输机制和广播及路由支持机制。同时，本课题在此基础上提出了一种延时优化机制，可以加快数据完成多跳传输。本章完成了课题的设计要求，下一章将主要讲述本课题设计的 MAC 协议在网络仿真平台 NS2 上的具体实现过程。一方面使协议细节更加具体。另一方面，为协议在具体平台的实现奠定了基础。

第四章 SC-MAC 协议在 NS2 平台上的实现

本文所设计的无线传感器网络低功耗异步 MAC 协议 SC-MAC，选用了目前学术界广泛使用的 NS2 网络仿真平台软件进行协议具体算法和机制的实现。接下来本文将首先对 NS2 平台进行概述，然后详细说明 SC-MAC 协议的具体实现过程。

4.1 NS2 网络仿真平台

4.1.1 NS2 平台的简介

NS2 是指 Network Simulator version 2，是一种针对网络技术的源代码公开的、免费的软件模拟平台^[41]，研究人员使用它可以很容易的进行网络技术的开发，而且发展到今天，它所包含的模块已经非常丰富，几乎涉及到了网络技术的所有方面。所以，NS 成了目前学术界广泛使用的一种网络模拟软件。在每年国内外发表的有关网络技术的学术论文中，利用 NS 给出模拟结果的文章最多，通过这种方法得出的研究结果也是被学术界所普遍认可的，此外，NS 也可作为一种辅助教学的工具，已被广泛应用在了网络技术的教学方面。因此，目前在学术界和教育界，有大量的人正在使用或试图使用 NS。

NS2 是一种面向对象的网络仿真器，本质上是一个离散事件模拟器。由 UC Berkeley 开发而成。它本身有一个虚拟时钟，所有的仿真都由离散事件驱动的。目前 NS2 可以用于仿真各种不同的 IP 网，已经实现的一些仿真有：网络传输协议，比如 TCP 和 UDP；业务源流量产生器，比如 FTP, Telnet, Web CBR 和 VBR；路由队列管理机制，比如 Droptail, RED 和 CBQ；路由算法比如 Dijkstra 等。NS2 也为进行局域网的仿真而实现了多播以及一些 MAC 子层协议。

NS2 使用了 OTcl 和 C++ 两种编程语言分别处理模拟器需要做的事情。一方面，具体协议的模拟和实现，需要一种程序设计语言，能够高效的处理字节、报头等信息，能后应用合适的算法在大量的数据集合上进行操作。为了实现该任务，程序内部模块的运行速度非常重要，而运行模拟环境的时间、寻找和修复 bug 的时间，以及重新编译和运行的时间相对不重要。在这种情况下，C++ 语言非常合适。另一方面，许多网络中的研究工作都围绕着网络组件和环境的具体参数的设置和改变而进行，需要在短时间内快速地开发和模拟出所需要的网络环境，并且方便修改和发现、修复程序中的 bug。在这种情况下，网络环境布置时间尤为重要。因此，脚本语言就具有很大优势，具有面向对象特性的 Tcl 脚本语言就可以满足此要求。图

4.1 为 NS2 平台的架构图。对于一般的 NS2 用户来说，OTcl 脚本语言用来实现网络拓扑的生成。而 OTcl 事实上是构架在 Tcl 之上的面向对象后的延伸脚本。事件调度（Event Scheduler）和网络组件（Network Components）这两部分主要由 C++ 程序语言进行编写。

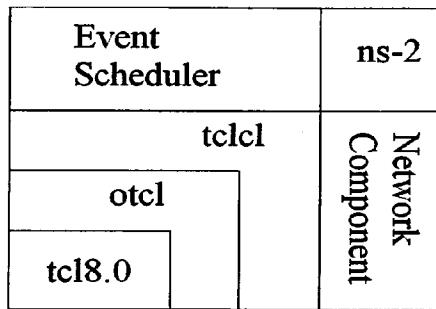


图 4.1 NS2 的构架图

4.1.2 NS2 平台主要构件

上小节已经提到，NS2 是一种面向对象的离散事件驱动的网络仿真器。因此，它的所有构件已编写成相应的 C++ 类和 OTcl 对象。NS2 有一个丰富的构件库，图 4.2 展示了构件库的层次关系。有了这个构件库，用户可以完成自己所要研究的系统的建模工作。此外，还提供了跟踪和检测的对象，可以把网络系统中的状态和事件记录下来以便分析。同时，NS2 的构件库还提供了大量的数学方面的支持，包括随机数产生、积分、随机变量等。从图 4.2 可以看出，NsObject 是基本网络构件的父类，它本身的父类是 TclObject 类。主要的基本网络组建可以分为两类，连接器（Connector）和分类器（Classifier）。它们都是 NsObject 的直接子类，同样也是很多基本网络组件的父类。分类器的派生类组件对象包括地址分类器（AddrClassifier）和多播分类器（McastClassifier）。连接器的派生类组件对象包括队列（Queue）、链路延迟（Delay）、各种代理（Agent）和跟踪对象（Trace）类。

NS2 使用时间调度器对所有组件的工作和该工作发生的时间进行控制和调度。其核心部分是一个离散事件模拟引擎。调度器类负责记录当前的时间，调度网络时间队列中的事件，并提供函数产生新事件，指定时间发生的时间。在一个网络模拟器中，典型的时间包括分组到达、时钟超时等，模拟时钟的推进由时间发生的时间量决定。模拟处理过程的速率不直接对应着实际时间。一个时间的处理可能又会产生后继的时间。模拟器所做的就是不停的处理一个个事件，直到所有的事件都被处理完或者某一特定事件发生为止。

在 NS2 仿真中，最常使用的网络构件是节点（Node）、链路（Link）、代理（Agent）、应用（Application）、队列（Queue）和跟踪（Trace）对象等等。其中节点主要用于仿真网络上的节点，包括主机和各种网络设备等，节点中可以配置很多属性，如节点类型、节点地址类型、所用路由算法等；链路则是将节点连接起来的一条通路，在链路上有队列和延迟，可以仿真实际网络上的包缓

冲和传输延时等情况；代理是 NS2 最有特色的对象之一，它可以仿真传输层的协议（如 TCP、UDP），也可以用于仿真网络业务流量（如 CBR）。跟踪对象是 NS2 中用于存储仿真结果的一个对象，用于配置需要跟踪的一些参数，并将其写入跟踪日志文件（Trace file）中。应用对象可用来仿真各种应用层对象（如 FTP、HTTP）。

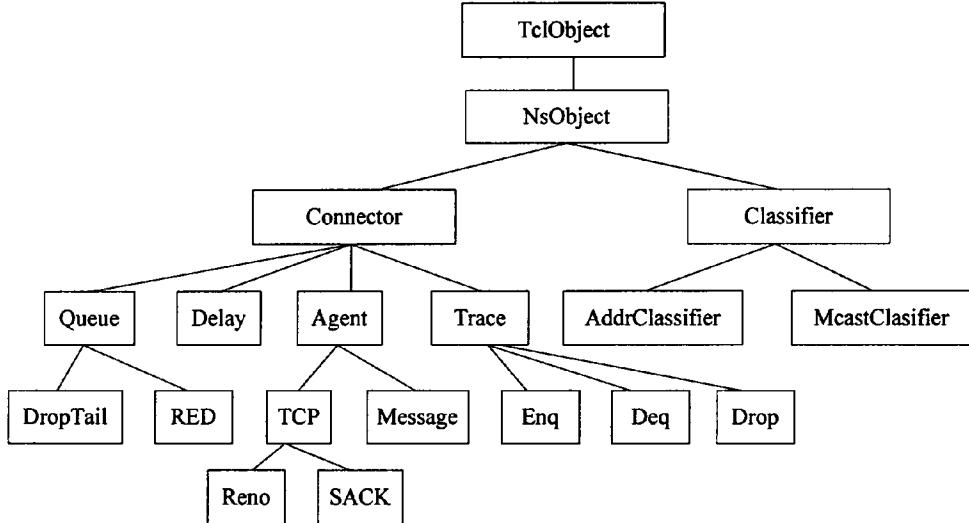


图 4.2 NS2 构件库的层次结构

4.1.3 NS2 仿真基本流程

使用 NS2 进行网络仿真的基本流程如图 4.3 所示，首先，用户要进行问题定义，考虑要仿真何种网络、拓扑结构，以及是否需要对源代码进行修改或添加。如果需要修改或添加代码，则需要对 NS2 源代码进行修改、重新编译和调试；如果不需要修改源代码，即利用 NS2 已有的构件即可完成仿真工作，那么只需用户编写 Tcl 仿真脚本。并执行该脚本进行仿真，生成相应的跟踪文件。最后，根据结果判断是否达到预期，并进一步分析问题，进行问题定义、源代码修改、Tcl 脚本生成。

整个仿真过程主要有三个部分的工作量：修改添加源代码、编写 Tcl 场景仿真脚本和结果分析，下面对这三个步骤中需要注意的问题做进一步描述：

- (1) 源代码修改添加：这一步只有在仿真需要修改源代码时才进行考虑，修改源代码是一项具有挑战性的工作，也是协议实现的核心；
- (2) Tcl 脚本编写：这是 NS2 仿真中必不可少的环节。在修改源代码实现相应协议后，编写 Tcl 脚本来描述网络环境，如节点规模、拓扑结构和流量等；
- (3) 仿真结果分析：结果分析是体现仿真工作成效的重要一环，仿真结果分析要求用户熟悉 NS2 的 trace 文件结构，并且使用 Gawk 等语言对数据进行处理。最后得出性能示意图；

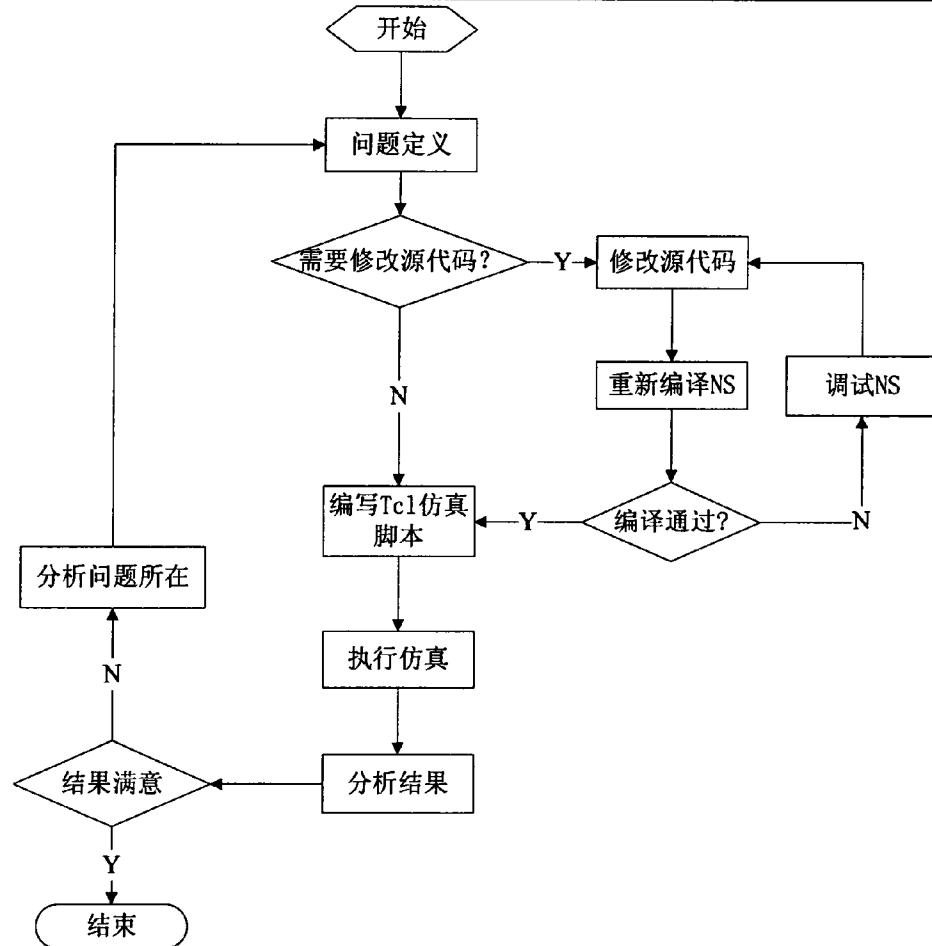


图 4.3 NS2 仿真基本流程

4.2 SC-MAC 协议核心数据结构

在 SC-MAC 协议实现的过程中，使用了控制帧、数据帧。由于 SC-MAC 协议为异步 MAC 协议，因此，并未引入同步帧。下面对各帧结构进行详细说明。

4.2.1 控制帧结构

控制帧包括 POLL 帧、HACK 帧、BEACON 帧、ACK 帧。

```

struct rts_frame {
    struct frame_control    rf_fc;
    u_int16_t      rf_duration;
    u_char         rf_ra[ETHER_ADDR_LEN];
    u_char         rf_ta[ETHER_ADDR_LEN];
    u_char         rf_fcs[ETHER_FCS_LEN];
    double        next_pollcast_time;
    // this RTS also serves as a BEACON
}

```

```

    u_int16_t      cw; // backoff window
    u_int16_t      intended_receiver;
    u_int16_t      on_demand;
};

struct cts_frame {
    struct frame_control    cf_fc;
    u_int16_t      cf_duration;
    u_char         cf_ra[ETHER_ADDR_LEN];
    u_char         cf_fcs[ETHER_FCS_LEN];
};

```

本课题利用 NS2 源代码已有的 RTS\CTS 帧进行修改实现 POLL 帧和 HACK 帧。其中，在 RTS 帧中加入全局变量 *next_pollcast_time*, 用来记录本节点下次进行 backcast 询问的时间。邻居节点在接收到帧后就可以记录下时间，并在以后的通信中预测唤醒。由于邻居节点第一次接收到此帧时，除记录时间以外，需要是当做 BEACON 帧接收，并发送数据。因此，POLL 帧需要扩展成 BEACON 帧。添加 *cw*(退避窗口)、参数 *intended_receiver* 和 *on_demand* 是实现数据的按需传输，根据用户需要自己设定开启或关闭利用 CTS 帧实现 HACK 帧功能。由于 CTS 帧和 HACK 帧完成确认功能，因此不需要进行格式修改。

```

struct rrtts_frame {
    struct frame_control    rf_fc;
    u_int16_t      rf_duration;
    u_char         rf_ra[ETHER_ADDR_LEN];
    u_char         rf_ta[ETHER_ADDR_LEN];
    u_int16_t      cw; // backoff window
    u_int16_t      intended_receiver;
    u_int16_t      on_demand;
    u_char         rf_fcs[ETHER_FCS_LEN];
};

struct ack_frame {
    struct frame_control    af_fc;
    u_int16_t      af_duration;
    u_char         af_ra[ETHER_ADDR_LEN];
    // as this ACK also serves as a BEACON
    u_char         af_ta[ETHER_ADDR_LEN];
    u_int16_t      cw; // backoff window
    u_int16_t      intended_receiver;
    u_char         af_fcs[ETHER_FCS_LEN];
    u_int16_t      wakeup_time_flag;// if i save sender's wakeup time ,set 1
};

```

上述帧结构为 BEACON 帧和 ACK 帧结构。需要注意，在 ACK 帧中需要加入存储调度信息标志，表明自己已经存储了该邻居的调度信息。

4.2.2 数据帧结构

```

struct hdr_mac802_11 {
    struct frame_control dh_fc;
    u_int16_t dh_duration;
    u_char dh_ra[ETHER_ADDR_LEN];
    u_char dh_ta[ETHER_ADDR_LEN];
    u_char dh_3a[ETHER_ADDR_LEN];
    u_int16_t dh_scontrol;
    u_char dh_body[0]; // XXX Non-ANSI
    int my_next_random_seed;
    double my_last_wakeup_time; // my next wakeup time from now
};


```

数据帧结构中包含了本节点下一次随机唤醒所使用的种子 *my_next_random_seed* 和本节点下一次唤醒时间 *my_last_wakeup_time*。邻居节点利用这两个参数即可预测此节点以后的唤醒时间，进行预测唤醒。

4.2.3 邻居链表结构

```

class myneighbor_list_item {
public:
//enum link_stat {OK, SUSPICIOUS, BROKEN, UNKNOWN};
// extended contents
myneighbor_list_item(int macindex, u_int32_t initCW = 0);
~myneighbor_list_item();
int macindex_; // my mac id
u_int32_t id; // neighbor id
myneighbor_list_item *next; // pointer to next item
double transmission_bound_time_;
bool if_collision_;
u_int32_t cw_;
};

class neighbor_policast_list_item{
public:
neighbor_policast_list_item();
~neighbor_policast_list_item();
u_int32_t nodeid; // upstream nodes id
double next_policast_time;
u_int16_t wakeup_time_flag;
neighbor_policast_list_item *next;
};


```

本课题定义了邻居链表、邻居 backcast 通信时间链表和邻居随机唤醒链表。三种链表结构如上，其中 *myneighbor_list_item* 类存储了该节点的邻居地址、退避窗口、传输上线时间等。后两项用来判断节点在某一时间时候还在唤醒状态。*neighbor_policast_list_item* 实现了对邻居节点进行 backcast

通信原语时间参数的存储。最后在 `mysender_list_item` 中存储了邻居节点的伪随机唤醒参数，用来实现发送节点和接收节点同时唤醒进行数据传输，节省能耗。

4.3 SC-MAC 协议具体实施方案

本小节将具体介绍 SC-MAC 从节点初始化到周期工作的流程。首先对加入网络的节点进行初始化，接收节点建立通信询问调度表，发送节点建立数据发送调度表，然后根据所述接收节点建立的通信询问调度表确定通信周期，节点在所述通信周期里进行传输工作流程，当一个通信周期结束时，如发送节点和接收节点正在通信，则判断负载传输没有完成，将发送节点和接收节点同时中断通信，然后进入下一通信周期继续进行传输工作流程，如发送节点和接收节点没有通信，则判断负载传输完成，然后直接进入下一通信周期继续进行传输工作流程。

如图 4.4 所示，节点加入网络进行初始化时，首先需要根据节点类型建立相应的调度表。接收节点建立通信询问调度表，发送节点建立数据发送调度表；然后根据接收节点的通信询问调度表，将时间分成重复的通信周期；最后节点根据通信周期的工作流程进行工作；本课题中用户通过对节点的网络协议栈编程和使用节点用户接口的方法建立通信询问调度表和数据发送调度表。

如果是接收节点，通信询问调度表中会生成两个参数：询问周期间隔参数和下一次通信询问时间参数。每进入一个新的通信询问周期，这两个参数都会更新。本课题中使用固定时间参数作为询问周期间隔，下一次通信询问时间=本次通信询问时间+询问周期间隔参数。在本课题实验中，将询问周期间隔设定为 1s，用户可以根据需要自定义周期询问间隔。

节点在每个通信周期里的传输工作流程如图 4.5 所示，具体包括如下步骤：

- 1) 接收节点根据初始化时建立的通信询问调度表，广播发送询问帧，当发送节点在所设定的超时时间内接收到该询问帧时，进入步骤 2)，否则进入步骤 3)；这里可以具体分为三种情况：(1)发送节点未存储接收节点的通信询问调度表时，超时时间设置为 5 个通信周期时长，本课题实验中的一个通信周期时长为 1s，故此时的超时时间为 5s；(2)发送节点已存储接收节点的通信询问调度表时，超时时间根据 IEEE802.15.4 标准中规定一个信标帧的超时时间进行设置；(3)发送节点没有检测到数据，那么一直处于睡眠状态，此时没有超时时间，而且发送节点一定不会接收到该询问帧，故直接进入步骤 3)；
- 2) 判断发送节点是否已经存储了接收节点的通信询问调度表，如果为是，则发送节点发送硬件确认帧 HACK 后，进入步骤 4)，否则进入步骤 5)；
- 3) 如图 4.6 所示，接收节点在所设定的超时时间到时，进入睡眠状态，同时，判断发送节点是否正处于睡眠状态，如果是，则保持睡眠状态，否则，在所设定的超时时间到时，进入睡眠状态；完成上述操作后进入步骤 6)；

4) 如图 4.7 所示, 判断接收节点是否接收到发送节点的硬件确认帧 HACK, 如是, 则继续保持唤醒状态, 否则, 进入睡眠状态;

41) 发送节点开始发送检测到的数据, 当接收节点接收到发送节点检测到的数据时, 进入 411), 当接收节点使用空闲信道评估 (CCA) 方法检测到碰撞时, 进入 412), 当接收节点既没有接收到数据也没有检测到碰撞, 进入 413); 其中本课题实验中使用空闲信道评估 (CCA) 提供的模式 1 方法检测碰撞, 即接收节点在一定时间内 (本课题方案中设置的时间为 1ms, 用户可以根据实际情况进行自定义设置) 未接收到正确的帧首定界符, 但通过 CCA 方法检测到信道正在活动, 就认为, 此时发生了碰撞;

411) 进入正常通信过程, 然后判断接收节点是否存储了这些检测到的数据中包含的数据发送调度表, 如是, 则直接回复不带确认标识的数据确认帧后进入步骤 6), 否则, 就将这些检测到的数据中包含的调度信息存储到接收节点的邻居链表里, 然后在数据确认帧里添加确认标识并发送该数据确认帧后进入步骤 6);

412) 接收节点立即睡眠, 并在本次通信周期内, 根据每个发送节点的数据发送调度表依次醒来, 并接收发生碰撞的数据, 由于发生碰撞, 发送节点在所设定的超时时间内不能接收到接收节点发送的数据确认帧, 因此, 发送节点进入睡眠状态, 然后根据数据发送调度表醒来发送检测到的数据; 完成上述操作后进入步骤 6);

413) 接收节点和发送节点都进入睡眠状态, 完成上述操作后进入步骤 6);

5) 如图 4.8 所示, 发送节点将接收节点发送的询问帧里包含的接收节点的通信询问调度表存储到一个邻居链表里, 然后直接发送检测到的数据, 当接收节点接收到发送节点检测到的数据时, 进入 51), 当接收节点使用空闲信道评估 (CCA) 方法检测到碰撞时, 进入 52), 当接收节点没有接收到发送节点检测到的数据也没有检测到碰撞时, 进入 53);

51) 进入正常通信过程, 然后判断接收节点是否存储了这些检测到的数据中包含的数据发送调度表, 如是, 则直接回复不带确认标识的数据确认帧后进入步骤 6), 否则, 就将这些检测到的数据中包含的调度信息存储到接收节点的邻居链表里, 然后在数据确认帧里添加确认标识并发送该数据确认帧后进入步骤 6);

52) 使用二进制指数退避方法进行数据重传, 即接收节点发送带有退避时间的信标帧, 发送节点接收到所述信标帧后, 在所述退避时间内随机选择一个时间重新发送检测到的数据, 完成上述操作后进入步骤 6); 本课题实验中选取的基础竞争窗口为 31 个时隙 (slot), 当重试计数器累增, 竞争窗口即移至下一个 2 个指数倍数。当竞争窗口到达最大极限 255 时就会维持该数字, 直到它被重新设定;

53) 接收节点和发送节点都进入睡眠状态, 完成上述操作后进入步骤 6);

6) 等待通信周期到时, 结束本通信周期的传输工作流程。

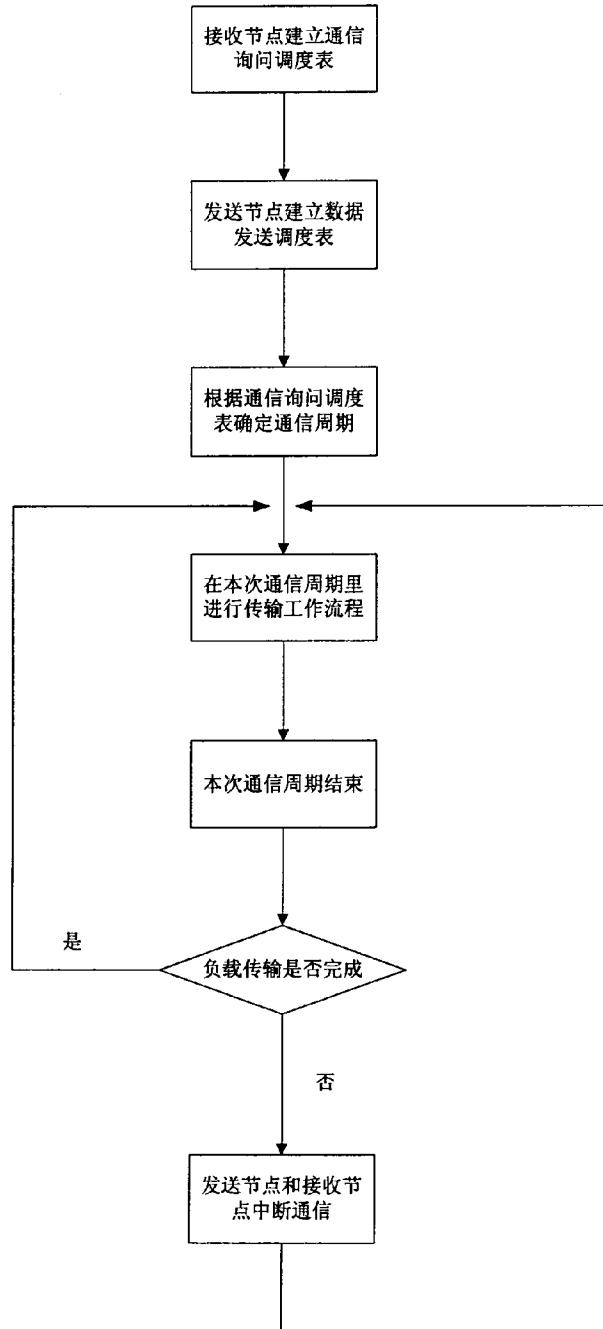


图 4.4 节点周期循环工作流程图

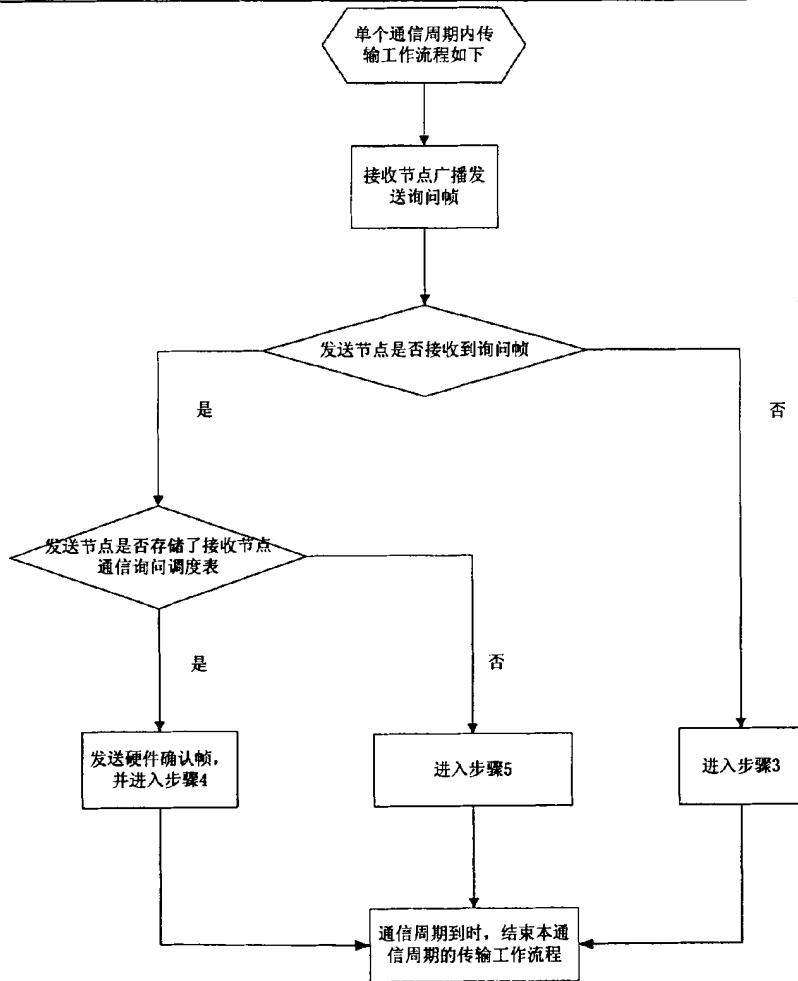


图 4.5 节点单周期内工作流程图

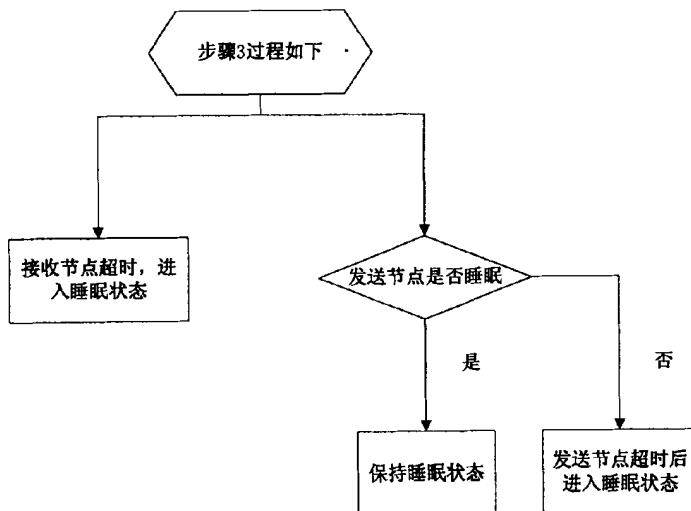


图 4.6 步骤3节点工作流程图

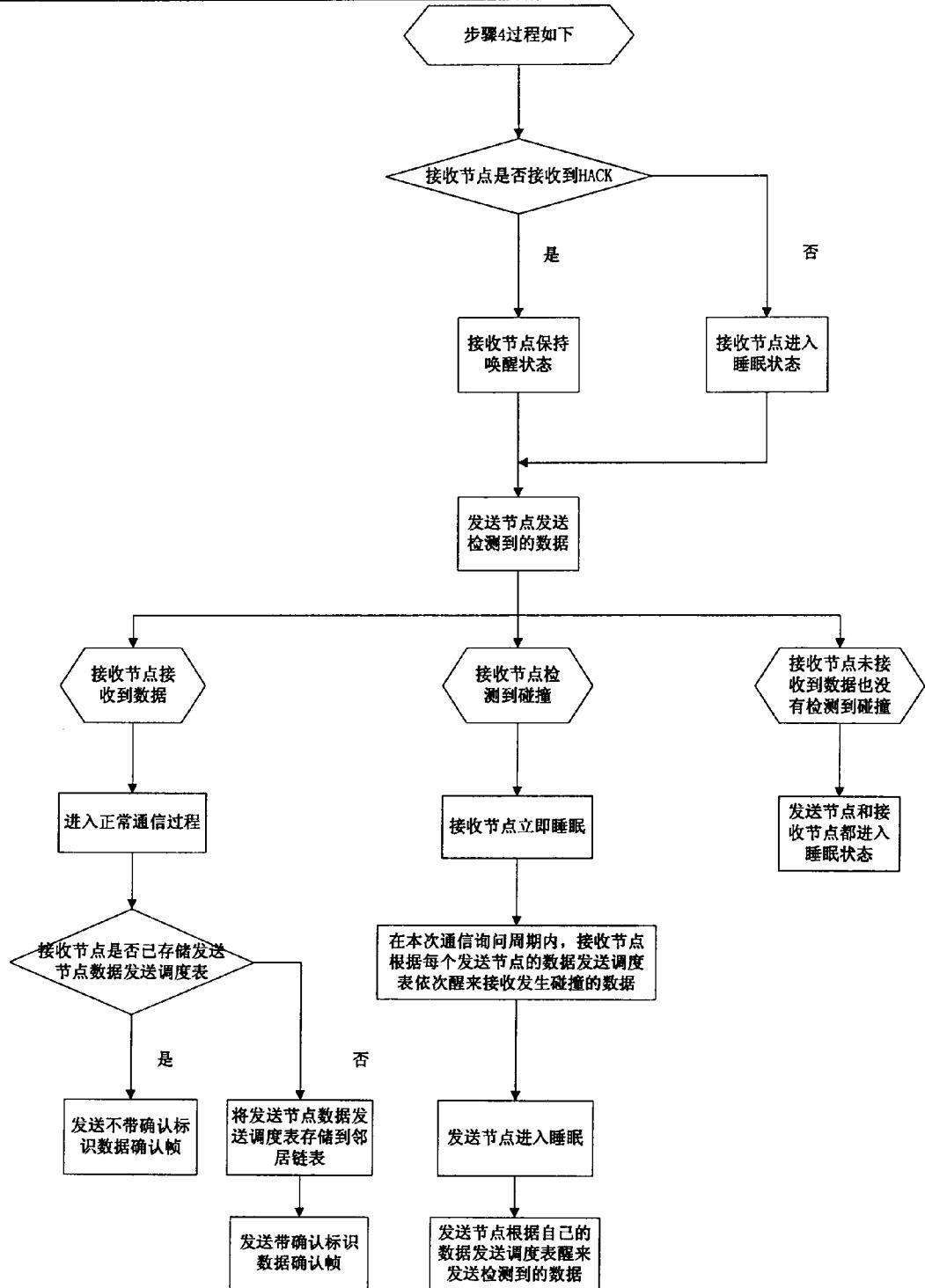


图 4.7 步骤 4 节点工作流程图

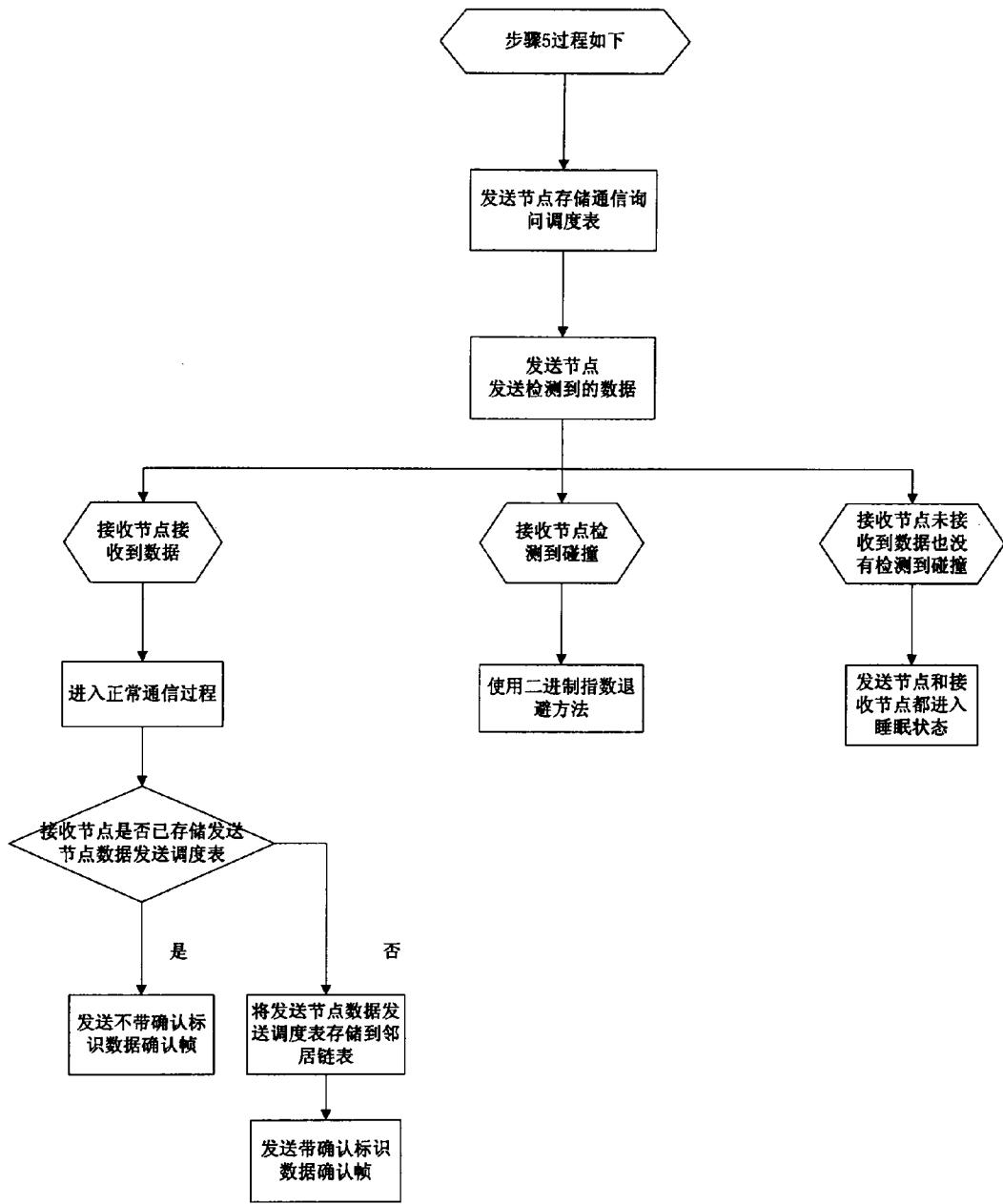


图 4.8 步骤 5 节点工作流程

4.4 SC-MAC 协议具体方案的代码实现

在 SC-MAC 协议中，主要的通信流程包括 backcast 通信原语、数据交互流程、载波侦听、退避策略、碰撞重传。本节将对异步周期询问和预测唤醒数据交互流程的实现进行代码解析。其它代码实现可结合上一小节的 SC-MAC 具体实施方案和 NS2 源码中已有的 802.11 协议进行代码实现。

4.4.1 异步周期询问机制的实现

Backcast 通信原语的实现流程分为目标节点发送 POLL 帧、发送节点接收 POLL 帧并回复 HACK 帧、目标节点接收 HACK 帧三个步骤。如图 4.9 所示，首先节点进入 start_hello() 函数进行初始化，启动周期询问定时器，当定时器超时后，利用 sendRTS() 装载 POLL 帧。然后启动退避定时器，进入 transmit() 函数发送 POLL 帧。同时在发送定时器超时后，选择是否转入睡眠状态。如图 4.11 所示，发送节点接收 POLL 帧时首先进入 recv() 函数，当接收定时器超时后，表明接收成功，并发送 HACK 帧。如图 4.10 所示，目标节点接收 HACK 帧的过程与接收 POLL 帧类似。

◆发送 POLL 帧：

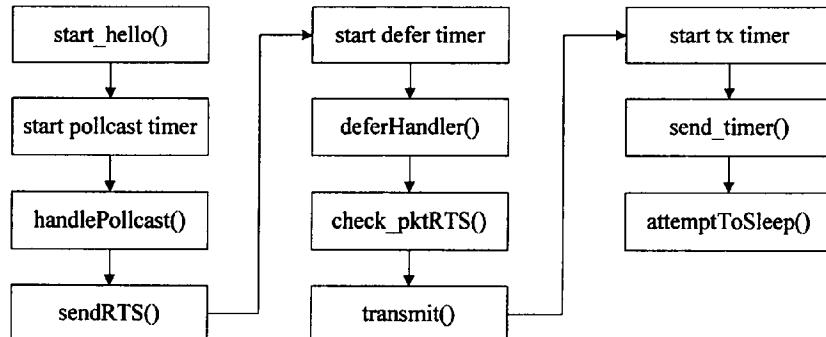


图 4.9 发送 POLL 帧函数流程图

◆接收 HACK 帧：

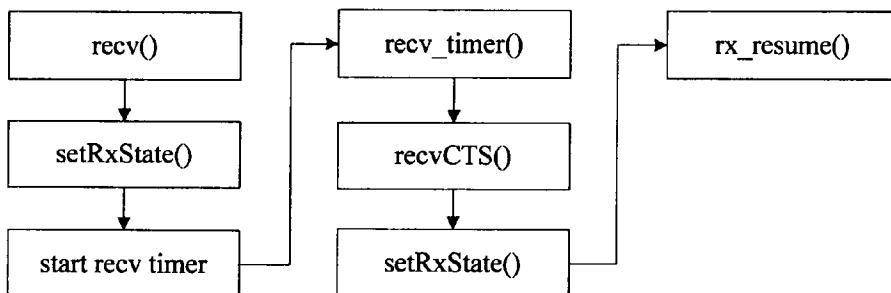


图 4.10 接收 HACK 帧函数流程图

◆接收 POLL 帧并回复 HACK 帧:

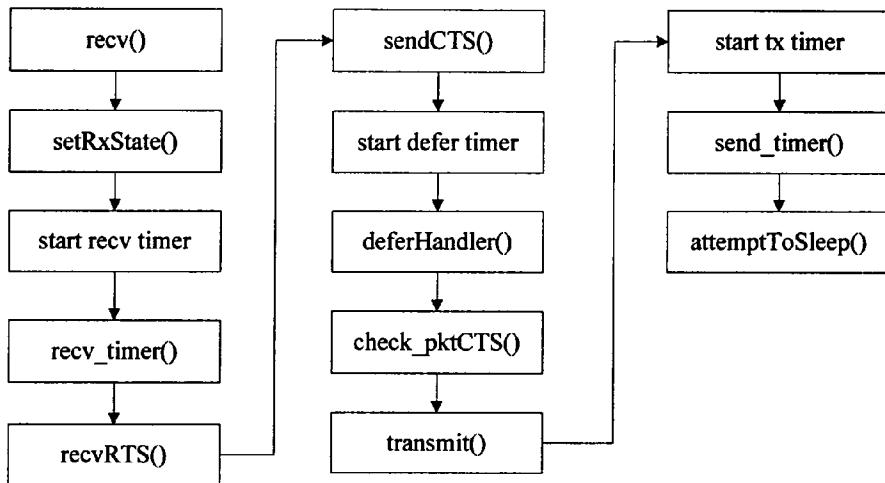


图 4.11 接收 POLL 帧并回复 HACK 帧函数流程图

4.4.2 预测唤醒数据交互流程的实现

SC-MAC 协议的数据交互流程分为上层触发 DATA 需要发送、接收 BEACON 帧发送 DATA 帧、接收 DATA 帧发送 ACK 帧、接收 ACK 帧四个步骤。节点有 DATA 需要发送时，唤醒侦听接收节点的 BEACON 帧。根据存储在邻居列表中的历史信息可知接收节点状态。如果这时接收节点处于侦听状态就根据是否 COL 启动相应定时器准备发送 DATA; 在 wakeup() 后有个 sinrMonitor 过程。信道忙会启动信道能量定时器，直到信道空闲。(当发送完成后 IFTimer 超时，txactive=0, 不影响接收。之后会根据 dwell time 进入睡眠); 当接收到目标节点的 BEACON 帧后，如果信道空闲就启动 defer 定时器并发送 DATA 帧。否则就启动 backoff 定时器进行退避定时器，待信道空闲后发送 DATA 帧；接收 DATA 帧发送 ACK 帧实现过程与上一过程类似。

4.4.3 异常处理的实现

在 SC-MAC 协议中异常处理包括数据重传和碰撞检测两部分。具体流程如下：

(1) 数据重传

如图 4.12 所示，当没有收到 ACK 时，发送节点会侦听等待接收节点的 BEACON 进行重传，discardTmer 多次超时都没有发送成功，重传计数器到达上限会丢弃此数据包。

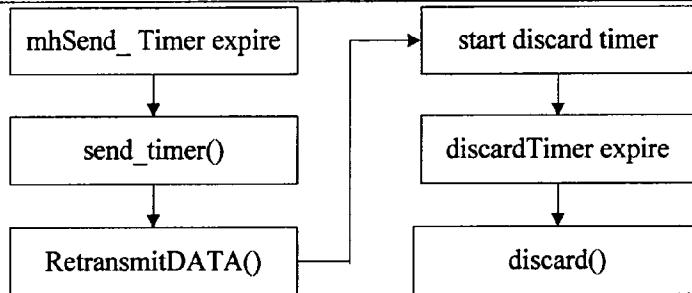


图 4.12 数据重传函数流程图

(2) 碰撞检测

如图 4.13 所示，在发送 BEACON/ACK 后的 dwell time 时间内，刚接收 pkt 时启动定时器检测是否有碰撞产生。在这段时间内未接收到正确的 DATA，则认为发生了碰撞，发送带 BW 的 BEACON。

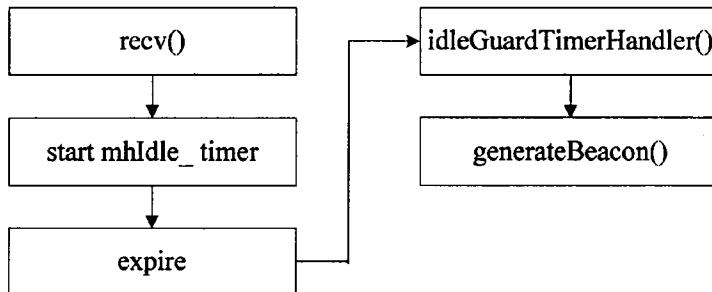


图 4.13 碰撞检测函数流程图

4.5 本章小结

本章对所设计的 SC-MAC 协议在 NS2 仿真平台进行实现的过程进行了详细说明。首先，介绍了 NS2 仿真的基本架构及使用 NS2 进行协议仿真的基本流程。然后，介绍了 SC-MAC 中的核心数据结构，如控制帧、数据帧结构，及各种链表。最后，详细说明了主要通信步骤的宏观源代码流程。为研究人员在 NS2 平台上实现其它 MAC 协议提供了一定程度的指导。

第五章 SC-MAC 协议仿真与结果分析

上述章节已经详细说明了 SC-MAC 协议设计和实现的过程，本章将对比 SC-MAC 协议和 PW-MAC 协议的性能。对比的指标为节点占空比和传输延时，其中节点占空比反应了节点的能量有效性。

实验仿真了三种典型的场景：星型拓扑场景、链式拓扑场景和随机拓扑场景。其中星型拓扑场景模拟了突发事件源端网络情况，而链式拓扑场景模拟了突发事件远端网络情况。最后通过随机拓扑随机突发流量模拟真实基于事件驱动应用场景网络情况。下面将首先介绍节点仿真参数的设置，接着介绍在各场景的仿真结果并进行分析。

5.1 仿真参数设置

无线信道模型：WirelessChannel，仿真环境中信道为理想环境，无噪声影响及信道误码。丢包现象只有数据包碰撞和缓存溢出产生；无线传播模型：TwoRayGround，此模型基于物理光学，不仅考虑了直接的路径，还考虑了地表反射路径；天线类型：OmniAntenna，即在水平方向图上表现为 360° 都均匀辐射，也就是平常所说的无方向性；队列类型：DropTail/PriQueue，即丢弃队尾数据，当队列的长度达到规定的最大长度时，之后所有到来的报文都被丢弃。

其它仿真参数如表 5.1 所示：

表 5.1 仿真参数

| 参数名称 | 参数大小 |
|-----------------------|-------------|
| Bandwidth | 250 Kbps |
| SIFS | 192 μ s |
| Slot | 320 μ s |
| CCA check delay | 128 μ s |
| Tx range | 250 m |
| Carrier sensing range | 550 m |
| Backoff window | 0-255 |
| Retry limit | 5 |
| Queue size | 50 pkt |

5.2 突发事件源端协议性能分析

5.2.1 拓扑场景建立

在“感兴趣”事件发生时，网络中靠近事件源端区域内的节点活动频繁，数据的传输碰撞情况严重。为了模拟突发事件源端情况，仿真实验场景 1 为星型拓扑，如图 5.1 所示，汇聚节点在中间，所有发送节点以 200m 为半径分布在汇聚节点周围。网络流量负载方面，模拟为当有事件产生时，所有发送节点都向汇聚节点发送数据，并通过改变竞争节点数目和数据产生间隔实现流量的变化。仿真时间持续 500s，每个发送节点从 100s 开始产生数据，并在 400s 结束。

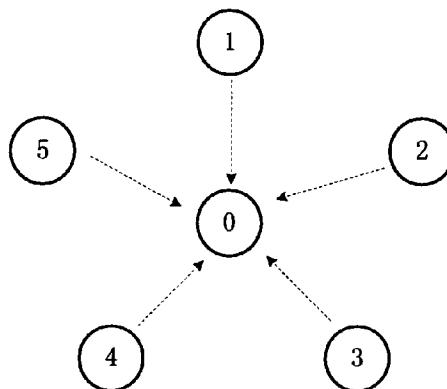


图 5.1 星型拓扑场景

5.2.2 协议性能分析

如图 5.2 横坐标所示，一般认为数据产生间隔 $<0.5s$ 时，网络呈大流量负载； $0.5<\text{数据产生间隔}<2s$ 时，网络呈中等流量负载；数据产生间隔 $>2s$ 时，网络呈低流量负载；图 5.2 纵坐标为节点的占空比（Duty-Cycle），表示节点的唤醒时间，这个参数间接体现节点的能耗。另一方面，发送节点的数据决定了碰撞剧烈程度，其中 1 个节点发送数据，表示无碰撞；3 个节点同时发送数据，表示轻微碰撞；5 个节点同时发送数据，表示严重碰撞。

从图 5.2 曲线趋势可以得出，在低流量负载情况下，SC-MAC 与 PW-MAC 的能耗相近；但是，随着流量负载的增加，PW-MAC 的能耗呈指数上升，原因是随着数据产生间隔的减少，每个发送节点传输的数据增多，占用信道的时间也就增加。当其它节点退避时间超时后，通过清晰信道检测（CCA）发现信道忙，这些节点就会加大退避窗口再次进行退避，如此循环，导致能耗迅速上升。而 SC-MAC 的能耗缓慢上升是由于数据量逐渐增大的缘故；从中可以得出，SC-MAC 不仅有效的降低了能耗，而且网络负载适应力强。另一方面，随着竞争节点的增多，PW-MAC 的退避机制性能恶化，导致能耗上升。而 SC-MAC 中，能耗基本不受竞争节点数目影响，原因是 SC-MAC 的机制足够使每个发送节点在 backcast 周期时分化传输，实现了顺利通信。

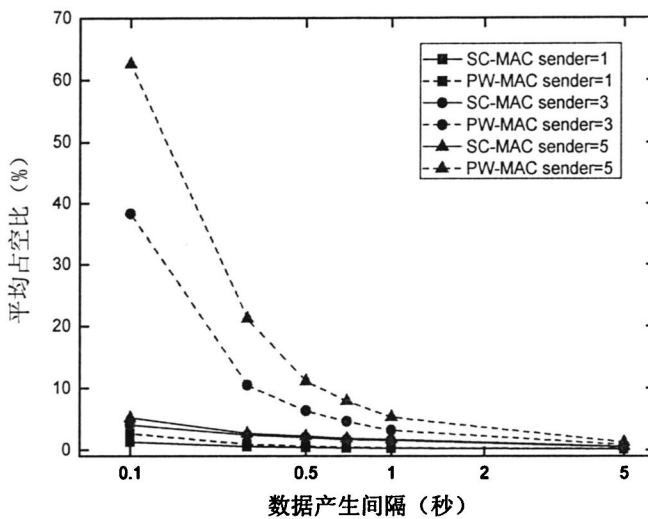


图 5.2 星型拓扑能耗仿真结果

从图 5.3 中可以看出，传输延时和能耗的趋势相似。在低流量负载情况下，SC-MAC 与 PW-MAC 的延时相近；但是，随着流量负载的增加，PW-MAC 的延时呈指数上升，而 SC-MAC 的延时曲线比较平稳。值得注意的是，SC-MAC 与 PW-MAC 的延时曲线交点出现在数据产生间隔=0.5s 处，即当数据率为 2pkt/s 时，两协议的延时几乎相等。以此为分界点，当数据率大于 2pkt/s 时，SC-MAC 的数据延时比 PW-MAC 明显减少。原因在于 PW-MAC 在大流量负载情况下，退避延时会显著增加，性能恶化速度快；当数据率小于 2pkt/s 时，SC-MAC 的传输延时同 PW-MAC 比，略有上升。原因是分时传输在一定程度上影响了传输速度。

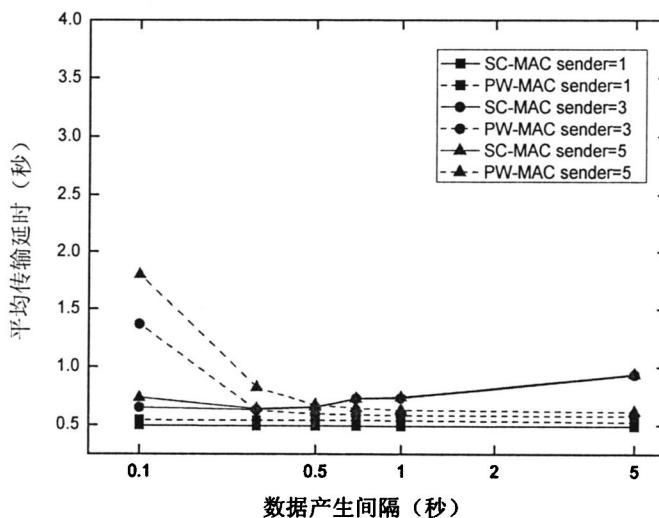


图 5.3 星型拓扑延时仿真结果

本仿真中，我们设置的 backcast 周期间隔为 1s 中，从图 5.4 中可以看出，无论网络负载如何变化，SC-MAC 的每跳的传输延时都可以保证在 1s 内。这对于 WSN 的实时应用可靠性服务具有重要意义。

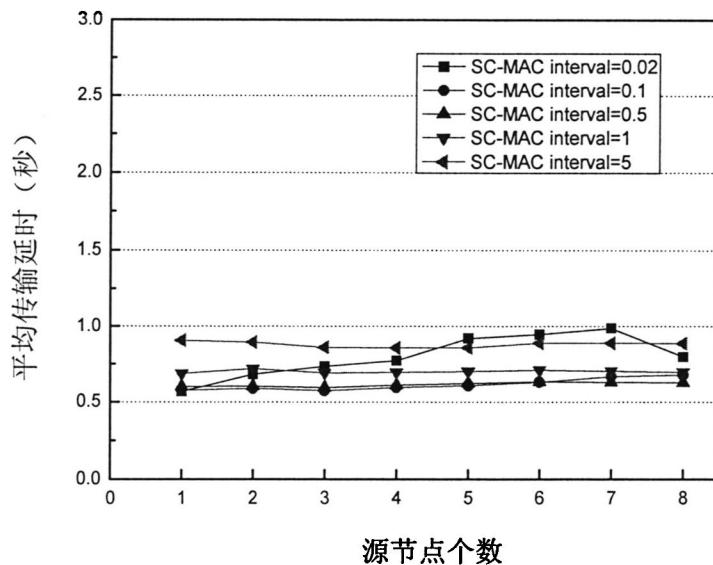


图 5.4 星型拓扑延时上限仿真结果

5.3 突发事件远端协议性能分析

5.3.1 拓扑场景建立

与突发事件网络源端不同，在远离事件源端处，数据的传输碰撞情况减少，通常呈无碰撞传输环境。因此，本次仿真实验的拓扑结构为链式模型，该场景由 5 个传感器节点组成，如图 5.5 所示，其中汇聚节点为 0 号节点，而 1 号到 4 号节点分别为源节点来测试 1 跳到 3 跳的性能。每个传感器节点的间距为 150m，这样设置可以保证网络中节点逐跳传输至汇聚节点。在仿真过程中，逐渐调整数据发送间隔。当源节点产生数据周期值越小，表明网络中流量负载越大，而源节点产生数据周期值越大，则网络流量负载越小。

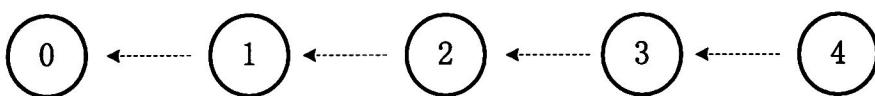


图 5.5 链式拓扑场景

5.3.2 协议性能分析

节点的平均占空比和平均传输延时分别如图 5.6 和图 5.7 所示。由于是链式拓扑场景，且无碰撞传输，SC-MAC 和 PW-MAC 的平均占空比十分相近。当网络负载增大时，转发节点需要向下游节点，在 PW-MAC 协议中上游节点就会一直保持唤醒，知道接收到转发节点的按需 BEACON 帧，而在 SC-MAC 中，上游节点如果没有接收到转发节点的周期询问帧，那么它就进入睡眠状态，并在下一个周期询问期唤醒。因此，PW-MAC 在大流量负载下能耗明显升高。从图 5.7 中，可以看出 SC-MAC 的传输延时较 PW-MAC 小，且随跳数增多，延时优势明显。原因在于在 SC-MAC 中应用的延时优化机制。

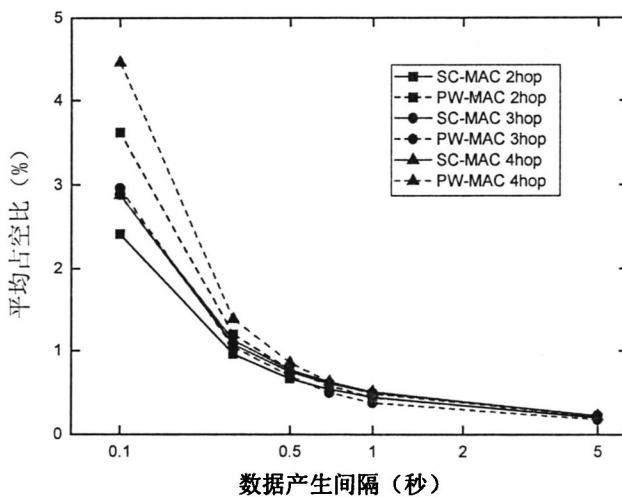


图 5.6 链式拓扑能耗仿真结果

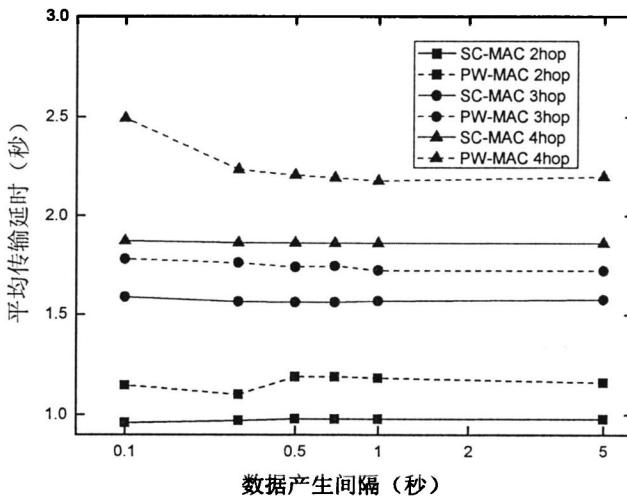


图 5.7 链式拓扑延时仿真结果

5.4 随机突发场景下协议性能分析

5.4.1 随机突发场景建立

通常情况下，基于事件驱动应用场景的网络，节点都是随机散布，而突发事件也是呈随机现象。因此，在仿真 3 中，模拟随机拓扑场景下 PW-MAC 和 SC-MAC 协议的能耗及延时性能。如图 5.8 所示，随机场景的面积为 1000 米×1000 米，随机散布 50 个节点，其中网关节点放置在右上角。网络中流量负载引入 RCE(Random Correlated-Event)模型，即随机相关事件模型，在感知范围内的节点都会产生数据向网关节点进行传输。通过调整感知范围可以改变网络中流量负载。当感知半径逐渐增大时，感知范围内所包含节点越多，产生的突发数据也就越多。每次仿真中随机位置触发 100 个事件、每个事件间隔 20 秒。

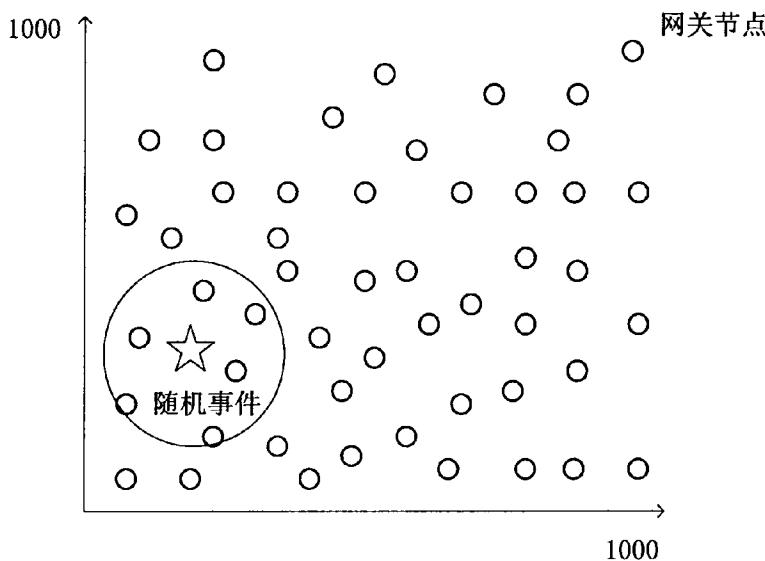


图 5.8 随机拓扑场景

5.4.2 协议性能分析

仿真结果如图 5.9 和图 5.10 所示。图 5.9 展示了对感知半径变化节点的平均占空比。图 5.10 展示了随感知半径变化，数据包传输延时的结果。可以明显看出，无论是能耗和传输延时，SC-MAC 都较 PW-MAC 有显著提高。从能耗的角度来看，随着感知半径的增大，SC-MAC 的能耗都在 0.5% 以下，而 PW-MAC 呈指数增长，最高达到 2.8%；从延时方面来看，SC-MAC 都在 4 秒以下，而 PW-MAC 最高延时已经达到 9 秒。如表 5.2 所示，本文以表格的形式给出了仿真结果同设计指标的能耗对比。当感知半径为 20 米时，由于被感知的节点数目很少，因此数据传输碰撞情况较少，能耗和延时比较接近。当感知半径小于 100 米时，被感知节点数目逐渐增加。数据传输碰撞情况升高，PW-MAC 协

议性能较 SC-MAC 有明显下降。因此能耗和延时性能降低量明显。当感知半径大于 150 米时，SC-MAC 协议的性能更加显著。此外，从表中可以看出本文完成了设计预期指标。其中计算公式为：

$$\text{优化量} = \frac{P_{pw-mac} - P_{sc-mac}}{P_{pw-mac}} \quad \text{其中 } P_{mac} \text{ 为参数结果;} \quad (2)$$

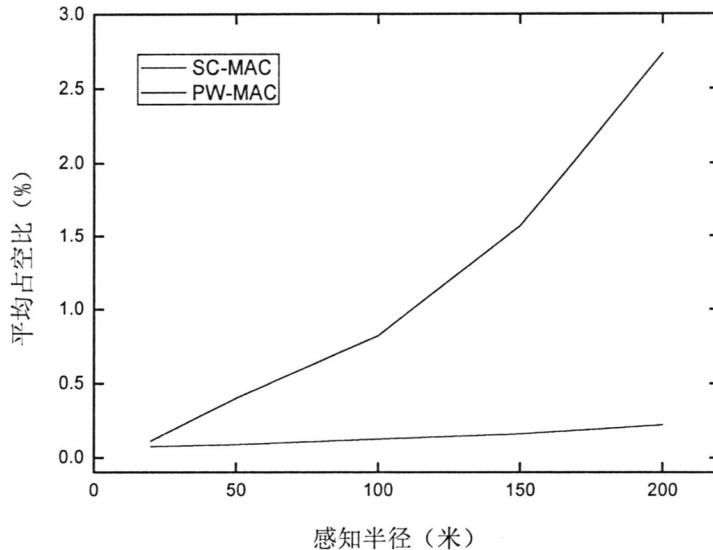


图 5.9 随机拓扑能耗仿真结果

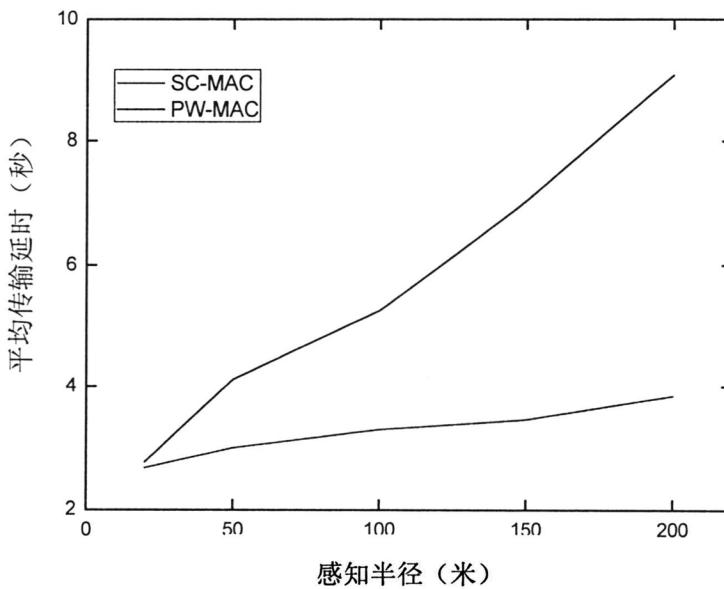


图 5.10 随机拓扑延时仿真结果

表 5.2 协议实际性能同指标能耗对比结果

| 感知半径 (m) | PW-MAC 能耗 (%) | SC-MAC 能耗 (%) | 能耗优化量 |
|----------|---------------|---------------|-------|
| 20 | 0.11304 | 0.07614 | 32.6% |
| 50 | 0.40118 | 0.08918 | 77.7% |
| 100 | 0.81878 | 0.12692 | 84.5% |
| 150 | 1.56943 | 0.16247 | 89.6% |
| 200 | 2.73393 | 0.22356 | 91.8% |

表 5.3 协议实际性能同指标延时对比结果

| 感知半径 (m) | PW-MAC 延时 (秒) | SC-MAC 延时 (秒) | 延时优化量 |
|----------|---------------|---------------|-------|
| 20 | 2.77123 | 2.68295 | 3.1% |
| 50 | 4.10949 | 3.0114 | 26.7% |
| 100 | 5.24886 | 3.30412 | 37.1% |
| 150 | 7.04547 | 3.4635 | 50.8% |
| 200 | 9.07828 | 3.84488 | 57.6% |

5.5 本章小结

本章对 SC-MAC 协议和 PW-MAC 协议进行了仿真分析比较。在星型拓扑、链式拓扑和随机拓扑场景下，分别模拟分析网络占空比和数据端到端平均延时。通过星型拓扑场景的仿真，分析基于 backcast 通信协作机制带来的能耗上的减少程度。通过链式拓扑场景的仿真，分析了延时优化机制对于数据包延时性能的提升程度。最后，从随机拓扑场景中，可以看出，在基于事件驱动的应用中 SC-MAC 协议较 PW-MAC 协议无论在能耗和延时方面都具有明显优势，完成了设计指标要求。

第六章 总结与展望

6.1 总结

无线传感器网络是本世纪 IT 热点技术之一，提供了一种泛在的、智能信息获取新模式，在军事侦察、环境监测、医疗监护、工业测控等领域具有广泛的应用前景。同时，无线传感器网络还可以作为底层的基础平台支撑未来的普适嵌入式计算应用。可以预见，随着技术的成熟，传感器网络技术将实现产业化，并获得极高的使用率。MAC 协议是无线传感器网络协议标准的重要组成部分之一，与网络中节点能耗，数据传输延时和网络吞吐量的性能指标息息相关。本文针对无线传感器网络 MAC 协议进行研究，工作主要包括以下几个方面：

1) 将国内外现有的典型协议根据不同的方式进行分类、归纳和总结。然后重点分析本文所研究的异步 MAC 协议的发展状况。介绍每种异步 MAC 协议的核心机制，优点和缺点。为本文的协议设计奠定了理论基础。

2) 针对无线传感器网络中的基于事件驱动应用，设计一种以发送节点为中心的低功耗异步 MAC 协议。文本所设计的 SC-MAC 协议的主要机制包括基于 backcast 通信原语的单跳协作、伪随机节点预约、异步广播支持和延时优化机制等。

3) 对本文所设计的 SC-MAC 协议在 NS2 仿真平台进行实现。介绍了 SC-MAC 中的核心数据结构，如控制帧、数据帧结构，及各种链表。并且详细说明了主要通信步骤的宏观源代码流程。为研究人员在 NS2 平台上实现其它 MAC 协议提供了一定程度的指导。

4) 在 NS2 平台实现 PW-MAC 协议，并将其与本文所设计的 SC-MAC 协议进行仿真对比分析。在星型拓扑、链式拓扑和随机拓扑三种场景中，分别对比了占空比和平均延时性能。结果表明，SC-MAC 协议较 PW-MAC 协议有所改善，这对于无线传感器网络异步 MAC 协议研究具有较大的参考价值。

6.2 展望

当前流行的无线传感器网络收发机普遍提供多信道支持。因此，在物理层技术可行的背景下，将单信道 MAC 协议机制，同多信道传输相结合，设计多信道异步低功耗 MAC 协议将是以后的研究方向。同时，无线传感器网络广泛应用于工业、医疗等领域，数据的可靠传输是系统正常运行和病人生命安全的重要保证。然而由于外部干扰的存在，数据传输的可靠性很难得到保证。当前主流的无线传感器网络都工作在免授权的 ISM 频段，而目前 ISM 频段已被 WiFi、蓝牙等多个无线网络共享，采用 ISM 频段的无线传感器网络在家庭护理、独居老人监护等方面的应用将受到来自其它无线网络的严重干扰。正如理论和实验研究结果表明，由于网间干扰的存在，无线传感器网络的通信可靠性受到严重影响。这也将是我们以后所要开展的研究工作。

此页不缺內容

致 谢

本论文所有工作是在我的导师吴建辉老师和责任导师刘昊老师及校外导师虞建立工程师悉心指导下完成的，他们严谨的工作作风和实事求是的态度使我受益匪浅，无论是在生活上还是学习上吴老师和刘老师都给予了我无私的帮助和指导，在此对吴老师、刘老师对我两年多来的谆谆教诲和宽容表示衷心的感谢。

其次，我要感谢陈昊博士带领我走进 MAC 协议研究的道路，他的言传身教让我懂得了如何做研究、怎么去研究。同时感谢蒋富龙博士在我完成论文工作过程中，对我细心的指导，分析课题的问题，并进行完善，帮助我顺利完成课题论文。更重要的是，他在生活中对我的帮助，指导我如何更好的去处理人际关系，与人沟通交流。在我面临重大压力的时候，帮我排忧解难，共度难关。感谢 WSN 小组的所有同学，大家团结一致、积极进取，营造了良好的科研氛围。在生活中大家同样团结一致，我们一起渡过了充实而又丰富多彩的研究生生活。

最后我还要感谢我的家人和朋友，有了他们的理解和支持我才能顺利的完成学业。再次衷心感谢所有帮助过我的老师和朋友们。

此页不缺內容

参考文献

- [1] Petrova, M'ahonen, and J.Riihijarvi. Interference measurements on performance degradation between colocated IEEE 802.11 g/n and IEEE 802.15.4 networks, in Proceedings of the 6th International Conference on Networking (ICN '07), Sainte-Luce, Martinique, 2007.
- [2] Ahmed, SKanhere, J. Sanjay. Poster abstract: Multi-channel interference in wireless sensor networks. Proceedings of the 2009 International Conference on Information Processing in Sensor Networks .2009. 367-368
- [3] Wood, Stankovic, gang Zhou. DEEJAM: Defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks. 2007 4th Annual Ieee Communications Society Conference on Sensor, Mesh and Ad-Hoc Communications and Networks. 2007(1): 60-69
- [4] L.Mo. Canopy closure estimates with GreenOrbs: sustainable sensing in the forest, presented at the Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, Berkeley, California, 2009.
- [5] Woo, Tong, and D. Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in Proceedings of the 1st international conference on Embedded networked sensor systems, Los Angeles, California, USA, 2003.
- [6] Hock Beng Lim, Keck Voon Ling, Wenqiang Wang, Yuxia Yao, Mudasser Iqbal, Boyang Li, Xiaonan Yin, and Tarun Sharma. The national weather sensor grid. In *Proceedings of the 5th international conference on Embedded networked sensor systems* (SenSys '07). 2007. 369-370.
- [7] Ivan Stoianov, Lama Nachman, Sam Madden, and Timur Tokmouline. PIPENET: a wireless sensor network for pipeline monitoring. In *Proceedings of the 6th international conference on Information processing in sensor networks* (IPSN '07). 2007. 264-273.
- [8] Lufeng Mo, Yuan He, Yunhao Liu, Jizhong Zhao, Shao-Jie Tang, Xiang-Yang Li, and Guojun Dai. Canopy closure estimates with GreenOrbs: sustainable sensing in the forest. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems* (SenSys '09). 2009. 99-112.
- [9] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.* 2004. 493-506.
- [10] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems* (SenSys '03). 2003. 171-180.
- [11] Tao Zheng, Radhakrishnan, S., Sarangan, V. PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks, *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, 2005.4-8
- [12] Rajendran, Garcia-Luna-Aceves, Obraczka. Energy-efficient channel access scheduling for power constrained networks, *Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on*, 2002(2):509-513
- [13] Venkatesh Rajendran, Katia Obraczka, and Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wirel. Netw.* 2006. 63-78.
- [14] Zhihui Chen, Khokhar. Self organization and energy efficient TDMA MAC protocol by wake up for wireless sensor networks, *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, 2004.335-341
- [15] Injong Rhee, Warrier, Jeongki Min, Sichitiu. Z-MAC: A Hybrid MAC for Wireless Sensor Networks, *Networking, IEEE/ACM Transactions on*, 2008, 16(3): 511-524
- [16] Sang Hoon Lee, Joon Ho Park, Choi. AMAC: Traffic-Adaptive Sensor Network MAC Protocol

-
- through Variable Duty-Cycle Operations, *Communications, ICC '07. IEEE International Conference on*, 2007. 3259-3264
- [17] Yanjun Sun, Shu Du, Omer Gurewitz, and David B. Johnson. DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing* (MobiHoc '08). 2008. 53-62.
- [18] Du shu,Saha, Johnson,. RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks, *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*,2007. 1478-1486
- [19] Paramvir Bahl, Ranveer Chandra, and John Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking* (MobiCom '04). 2004.216-230.
- [20] Gang Zhou, Yafeng Wu, Ting Yan, Tian He, Chengdu Huang, John A. Stankovic, and Tarek F. Abdelzaher. A multifrequency MAC specially designed for wireless sensor network applications. *ACM Trans. Embed. Comput.*, 2010.
- [21] Thabtah,Cowling, Yonghong Peng. MMAC: a new multi-class, multi-label associative classification approach, *Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on*, 2004, 217-224
- [22] Lei Tang, Yanjun Sun, Omer Gurewitz, and David B. Johnson. EM-MAC: a dynamic multichannel energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (MobiHoc '11). ACM, New York, NY, USA, , Article 23 , 11 pages.
- [23] Wei Ye, Fabio Silva, and John Heidemann. Ultra-low duty cycle MAC with scheduled channel polling. In *Proceedings of the 4th international conference on Embedded networked sensor systems* (SenSys '06). 2006. 321-334.
- [24] Xiao-Hui Lin, Yu-Kwong Kwok. On channel adaptive energy management in wireless sensor networks, *Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on*, 2005. 397-404
- [25] Rossi, M. Zorzi. Probabilistic Algorithms for Cost-based Integrated MAC and Routing in Wireless Sensor Networks. Proc. of the 3rd Int'l Modeling and Performance Analysis of Wireless Sensor Networks (Senmetrics 2005). San Diego, 2005: 86-96.
- [26] Sunil Kulkarni, Aravind Iyer, and Catherine Rosenberg. An address-light, integrated MAC and routing protocol for wireless sensor networks. *IEEE/ACM Trans.* 2006, 14(4): 793-806.
- [27] Michael Buetner, Gary Yee, Eric Anderson, and Richard Han. X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks, in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, 2006. 307-320
- [28] Yanjun Sun, Omer Gurewitz, and David B. Johnson. RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems* (SenSys '08). 2008. 1-14.
- [29] Lei Tang, Yanjun Sun, Gurewitz, Johnson. PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks, *INFOCOM, 2011 Proceedings IEEE*, 2011. 1305-1313
- [30] Joseph Polastre, Jason Hill, and David Culler. Versatile Low Power Media Access for Wireless Sensor Networks, in *Proceedings of the Second International Conference On Embedded Networked Sensor Systems* (SenSys 2004), 2004. 95-107
- [31] Al-Karaki and Kamal. Routing Techniques in Wireless Sensor Networks: A Survey, *IEEE Wireless Comm.*, 2004, 11(6): 6-28
- [32] Hongwei Zhang, Anish Arora, Young-ri Choi, and Mohamed Gouda. Reliable Bursty Convergecast in Wireless Sensor Networks. In *Proceedings of the Sixth ACM International Symposium on Mobile Ad*

- Hoc Networking and Computing (MobiHoc 2005), May 2005. 266–276
- [33] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In Proceedings of the Fifth Annual International Conference on Mobile Computing and Networking (MobiCom 1999), August 1999. 263–270
- [34] Bret Hull, Kyle Jamieson, and Hari Balakrishnan. Mitigating Congestion in Wireless Sensor Networks. In Proceedings of the Second International Conference On Embedded Networked Sensor Systems (SenSys 2004), November 2004. 134–147
- [35] Ye Liu, Fulong Jiang, Hao Liu, Jianhui Wu. SC-MAC: A sender-centric asynchronous MAC protocol for burst traffic in wireless sensor networks, Communications (APCC), 2012 18th Asia-Pacific Conference on , 2012. 848-853
- [36] Incel, Jansen, and S. Mullender, MC-LMAC: A Multi-Channel MAC Protocol for Wireless Sensor Networks, vol. Technical Report, University of Twente, 2006.
- [37] Carlson, Terzis. Flip-MAC: A density-adaptive contention-reduction protocol for efficient any-to-one communication, *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, June 2011. 1-8
- [38] Pei Huang, Chen Wang, Li Xiao, Hongyang Chen. RC-MAC: A receiver-centric medium access control protocol for wireless sensor networks, *Quality of Service (IWQoS), 2010 18th International Workshop on*, 2010. 1-9
- [39] P. Dutta, R. Mus`aloiu-E., I. Stoica, and A. Terzis. Wireless ACK collisions not considered harmful,” in HotNets-VII: Proceedings of the 7th Workshop on Hot Topics in Networks, Oct. 2008.
- [40] Donald E. Knuth. The art of computer programming, third edition, volume 2: Seminumerical algorithms, section 3.2.1: The linear congruential method. Addison-Wesley, 1997. 10–26
- [41] 徐雷鸣, 庞博, 赵耀. NS 与网络模拟. 北京: 人民邮电出版社, 2003

此页不缺內容

攻读硕士学位期间的研究成果

论文成果

- [1] Ye Liu; Fulong Jiang; Hao Liu; Jianhui Wu, "SC-MAC: A sender-centric asynchronous MAC protocol for burst traffic in wireless sensor networks," *Communications (APCC), 2012 18th Asia-Pacific Conference on*, vol., no., pp.848,853, 15-17 Oct. 2012

专利成果

- [1] 刘昊, 刘野, 蒋富龙, 吴建辉, 一种无线传感器网络的突发流量负载传输方法, 申请号: 201210113650.8

此页不缺內容

附录

随机场景 TCL 脚本

```
*****
Main.tcl
*****
#无线节点参数设置
set val(chan)      Channel/WirelessChannel    ;# channel type
set val(prop)      Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)     Phy/WirelessPhy            ;# network interface type
set val(mac)       Mac/SCMAC                  ;# MAC type
set val(ifq)       Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)        LL                         ;# link layer type
set val(ant)       Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)    100                        ;# max packet in ifq
set val(nn)        51                         ;# number of mobilenodes
set val(rp)        DSR                        ;# routing protocol
set val(x)         1000                       ;# X dimension of the topography
set val(y)         1000                       ;# Y dimension of the topography
set val(energymodel) EnergyModel      ;
set val(initialenergy) 100000          ;# Initial energy in Joules

#数据流参数
set cbr_interval 20.0           ;#设定的 CBR 流量器的发送间隔
set cbr_maxpkts 10              ;#设定的 CBR 流量器发送的包数
set cbr_packetsize 1

#MAC 参数
source NS2para.tcl

#建立仿真实例
set ns [new Simulator]

#开启 trace
set tracefd [open random_51.tr w]
$ns trace-all $tracefd

#建立 topo
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

# 创建 god 对象
set god_ [create-god [expr 1+$val(nn)]]; 
$god_ on
```

```

#创建物理信道对象
set chan_1_ [new $val(chan)]

#配置无线参数
$ns node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -channel $chan_1_ \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    #-energyModel $val(energymodel) \
    -idlePower 13.5 \
    -rxPower 13.5 \
    -txPower 24.75 \
        -sleepPower 0.015 \
        -transitionPower 0 \
        -transitionTime 0.005 \
    # -initialEnergy $val(initialenergy)

puts "MAC type is $val(mac)"

#建立节点和拓扑
source topology.tcl

#load routing file
set opt(rtf) /random_1000_1000_51.rtf
set dsr [new Agent/DSRAgent]
$dsr node $node_(1)
$dsr rt_rq_max_period 100
$dsr rt_rq_period 30
$dsr send_timeout 300
$dsr load-rttable-file $opt(rtf)

#代理和数据流建立
source traffic_10.tcl

#定义结束过程
source otherproc.tcl
proc finish {} {

```

```

global tracefd ns
$ns flush-trace
close $tracefd

exit 0
}

#模拟结束

$ns at 1000.0 "finish"
$ns at 1000.0 "puts \"NS EXISTING..\"; $ns halt"
puts "Start Simulation..."

$ns run
*****
NS2para.tcl
*****
# default system parameters
set opt(radioBW) 2.5e5 ; # 250 kbps
Agent/UDP set packetSize_ 2000;           # prevent fragmentation in CBR

set opt(RTSThreshold)      2000
set opt(CSThresh)          [Phy/WirelessPhy set CSThresh_]
set opt(RXThresh)          [Phy/WirelessPhy set RXThresh_]
set opt(CWMin)              31 ;# 7; 2^3-1 802.15.4 spec
set opt(SlotTime)          0.000320 ;# 20 symbols @ 62.5 ksymbol rate
set opt(SIFS)                0.000192 ;# turnaround time; 192us 12 symbols @ 62.5 ks; we also use this
as T_ack (interval between DATA and ACK) though in the standard it's a random number between
aTurnaroundTime and (aTurnaroundTime + aUnitBackoffPeriod)
set opt(PreambleLength)    40 ;# 32 bit; SHR header
set opt(PLCPHeaderLength)  8 ;# 8 bits; length field
set opt(DutyCycleLength) 1 ;

Mac/802_11 set RTSThreshold_      $opt(RTSThreshold)
Mac/802_11 set CSThresh_          $opt(CSThresh)
Mac/802_11 set DutyCycleLength_   $opt(DutyCycleLength)
Mac/802_11 set CWMin_             $opt(CWMin)
Mac/802_11 set SlotTime_          $opt(SlotTime)
Mac/802_11 set SIFS_              $opt(SIFS)
Mac/802_11 set PreambleLength_   $opt(PreambleLength)
Mac/802_11 set PLCPHeaderLength_ $opt(PLCPHeaderLength)
Mac/802_11 set PLCPDataRate_     $opt(radioBW)
Mac/802_11 set LongRetryLimit_   0; # no retransmission in orig XMAC
Mac/802_11 set ShortRetryLimit_ 0; # no retransmission in orig XMAC
Mac/802_11 set dataRate_         $opt(radioBW);
Mac/802_11 set basicRate_        $opt(radioBW);

```

| | |
|--|--|
| Mac/SCMAC set RTSThreshold_ | \$opt(RTSThreshold) |
| Mac/SCMAC set CSThresh_ | \$opt(CSThresh) |
| Mac/SCMAC set RXThresh_ | \$opt(RXThresh) |
| Mac/SCMAC set DutyCycleLength_ | \$opt(DutyCycleLength) |
| Mac/SCMAC set CWMin_ | \$opt(CWMin) |
| Mac/SCMAC set SlotTime_ | \$opt(SlotTime) |
| Mac/SCMAC set SIFS_ | \$opt(SIFS) |
| Mac/SCMAC set PreambleLength_ | \$opt(PreambleLength) |
| Mac/SCMAC set PLCPHeaderLength_ | \$opt(PLCPHeaderLength) |
| Mac/SCMAC set PLCPDataRate_ | \$opt(radioBW) |
| Mac/SCMAC set LongRetryLimit_ | 5; # 5 retrans by default |
| Mac/SCMAC set ShortRetryLimit_ | 5; # 5 retrans by default |
| Mac/SCMAC set dataRate_ | \$opt(radioBW); |
| Mac/SCMAC set basicRate_ | \$opt(radioBW); |
| Mac/SCMAC set base_beacon_frame_length | 12 ; # preamble(6) + FC(2) + src (2) [+ seq(1) + dst(2) + bw(1)] + crc(2) |

ns-random 32

```

Mac/802_11 set CWMin_          31 ; # not used
Mac/SCMAC set  CWMax_          255;
*****
traffic.tcl
*****
#建立发送和接收代理
#发送代理 0
set udp0 [new Agent/UDP]
$ns attach-agent $node_(1) $udp0
$god_ addUdpAgent 1  $udp0

#发送代理 1
set udp1 [new Agent/UDP]
$ns attach-agent $node_(6) $udp1
$god_ addUdpAgent 6  $udp1

#接收代理 0
set null0 [new Agent/Null]
$ns attach-agent $node_(2) $null0

#接收代理 1
set null1 [new Agent/Null]
$ns attach-agent $node_(8) $null1

```

```

#连接代理
$ns connect $udp0 $null0
$ns connect $udp1 $null1

#建立数据流 0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ $cbr_packetsize
$cbr0 set maxpkts_ $cbr_maxpkts
$cbr0 set interval_ $cbr_interval

#建立数据流 1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packetSize_ $cbr_packetsize
$cbr1 set maxpkts_ $cbr_maxpkts
$cbr1 set interval_ $cbr_interval

#开启和结束时间
$ns at 100.0 "$cbr0 start"
$ns at 500.0 "$cbr0 stop"

$ns at 200.0 "$cbr1 start"
$ns at 500.0 "$cbr1 stop"
***** topology.tcl *****
***** #建立节点 *****
# a node talks to no one

for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns node]
    set mac_($i) [$node_($i) getMac 0]
    $god_ new_node $node_($i);
    $node_($i) random-motion 0; # disable random motion
    puts "i=$i, node_($i) created...."
}

#坐标设置
$node_(0) set X_ 1.0
$node_(0) set Y_ 1.0
$node_(0) set Z_ 0

source random_1000_1000_51.tcl
*****
```