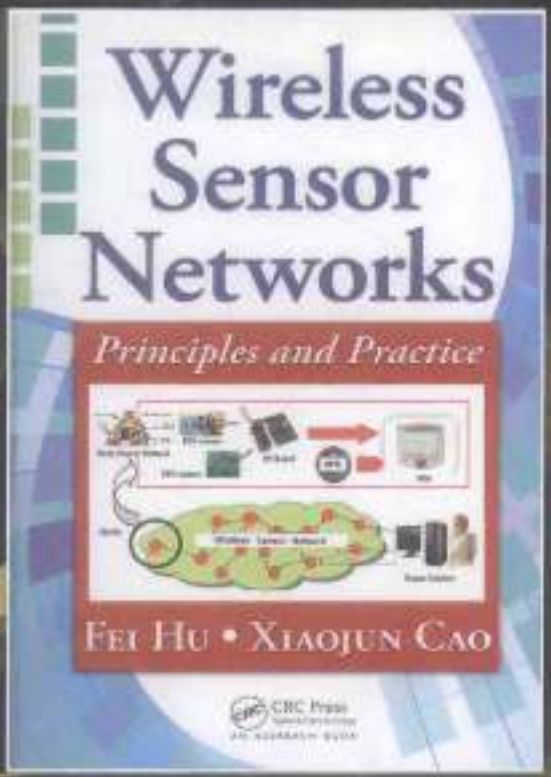
计 算 机 科 学 丛 书

CRC Press

无线传感器网络
原理与实践

[美] 胡飞 (Fei Hu) 曹小军 (XiaoJun Cao) 著
牛晓光 宫继兵 译

Wireless Sensor Networks
Principles and Practice



机械工业出版社
China Machine Press

无线传感器网络原理与实践

Wireless Sensor Networks Principles and Practice

随着信息技术和应用需求的发展，无线传感器网络已成为研究热点，并广泛应用于工业、医疗、交通、环境等领域。相比于传统的网络技术，无线传感器网络具有许多优势，但后者在网络结构的设计、模型与算法等方面存在很多极具挑战性的问题。

本书由两位优秀的科学家编写，他们的研究获得过美国国家科学基金会（NSF）、IBM和思科公司的资助。本书面向无线传感器网络领域的初学者，介绍无线传感器网络这一新兴领域的相关知识，内容涵盖无线传感器网络的关键技术、标准以及典型的应用等主题。

本书以简洁、通俗易懂的语言阐述复杂的概念和过程，将硬件设计、介质访问控制、路由方案、传输协议、操作系统支持、中间件、数据管理、本地化、同步、安全、能量控制、执行器/水下/视频传感器网络及相关前沿研究的知识展现在读者面前。书中还提供了大量的练习题、案例研究以及实际的无线传感器网络设计案例，方便读者学习。

本书特点

- 全面、系统地介绍了无线传感器网络的基础理论、标准和典型应用。
- 引入大量案例，生动阐述解决真实世界挑战和优化问题的设计和实现步骤。
- 每章包括丰富的练习题、实际的案例和延伸阅读，以及大量的参考文献，便于读者学以致用并深入探索。

作者简介



胡飞（Fei Hu）目前任阿拉巴马大学电子与计算工程学院副教授。1999年于同济大学获得博士（信号处理）学位，并在2002年获得美国克拉克森大学博士（电子与计算机工程）学位。研究领域包括传感器网络、无线网络、网络安全及其在生物医疗领域的应用，其研究先后获得了美国国家科学基金会、思科、Sprint等项目的资助。



曹小军（XiaoJun Cao）目前任佐治亚州立大学计算机科学学院副教授。他本科毕业于清华大学，在中国科学院获得硕士学位后，于2004年获得纽约州立大学布法罗分校计算机科学与工程博士学位。他的研究领域包括光纤通信网络、无线光通信网络、移动自组织网络、无线传感器网络等通信网络的模型建立、性能分析以及协议/算法设计等方面，获得了2006年度美国国家科学基金会杰出青年教授奖（NSF CAREER Award），其研究先后获得美国国家科学基金会、IBM公司和思科大学研究计划等项目的资助。

CRC Press

营销热线：(010) 88379604
客服热线：(010) 88378991 88361066
购书热线：(010) 68326294 88379649 68995259

华章网站：www.hzbook.com
网上书店：www.china-pub.com
数字阅读：www.hzmedia.com.cn

上架建议：计算机/通信/网络

ISBN 978-7-111-40699-0

9 787111 406990 >

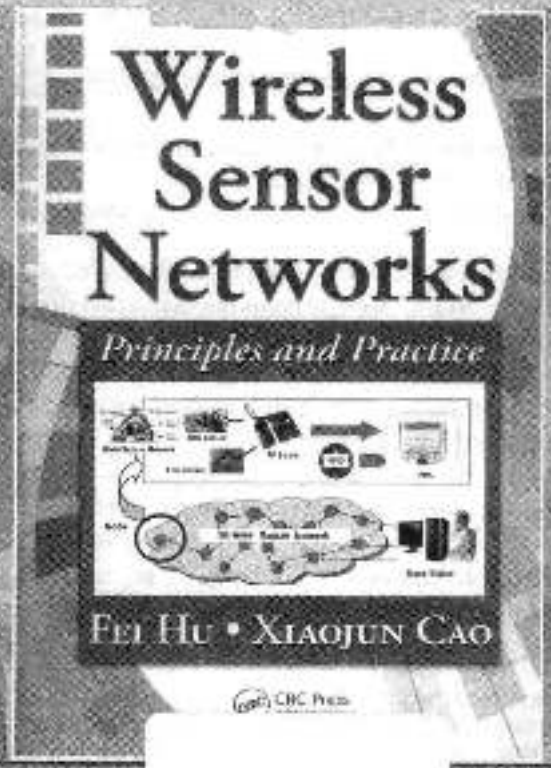
定价：79.00元

计 算 机 科 学 丛

无线传感器网络
原理与实践

[美] 胡飞 (Fei Hu) 曹小军 (XiaoJun Cao) 著
牛晓光 宫维兵 译

Wireless Sensor Networks
Principles and Practice



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

无线传感器网络：原理与实践 /（美）胡飞（Fei Hu），（美）曹小军（XiaoJun Cao）著；牛晓光，宫继兵译．—北京：机械工业出版社，2015.1
（计算机科学丛书）
书名原文：Wireless Sensor Networks: Principles and Practice

ISBN 978-7-111-49699-0

I. 无… II. ① 胡… ② 曹… ③ 牛… ④ 宫… III. 无线电通信－传感器 IV. TP212

中国版本图书馆 CIP 数据核字（2015）第 028513 号

本书版权登记号：图字：01-2011-5062

Wireless Sensor Networks: Principles and Practice by Fei Hu and XiaoJun Cao (978-1-4200-9215-8)

Copyright © 2010 by Taylor Francis Group, LLC

Authorized translation from the English language edition published by CRC Press, part of Taylor & Francis Group LLC. All rights reserved.

China Machine Press is authorized to publish and distribute exclusively the Chinese (Simplified Characters) language edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal.

本书原版由 Taylor & Francis 出版集团旗下 CRC 出版公司出版，并经授权翻译出版。版权所有，侵权必究。

本书中文简体字翻译版授权由机械工业出版社独家出版并限在中国大陆地区销售。未经出版者书面许可，不得以任何方式复制或抄袭本书的任何内容。

本书封面贴有 Taylor & Francis 公司防伪标签，无标签者不得销售。

本书介绍了无线传感器网络的基本概念、硬件构成、网络协议栈、操作系统、中间件、定位技术、安全策略、数据管理、同步等内容，并给出了典型无线传感器网络的工作机制，通过两个案例进一步说明了无线传感器网络相关技术的综合应用。本书内容丰富，语言简练，理论叙述深入浅出。书中提供了丰富的课后练习题、思考题和学习资源，可供学习者充分巩固所学内容。

本书适合作为高等院校物联网工程专业、计算机科学与技术及相关专业的“无线传感器网络”课程的教材，也可供从事相关领域工作的工程技术人员参考。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：朱　劼　　责任校对：殷　虹
印　　刷：北京诚信伟业印刷有限公司　　版　　次：2015 年 3 月第 1 版第 1 次印刷
开　　本：185mm×260mm　1/16　　印　　张：21.75
书　　号：ISBN 978-7-111-49699-0　　定　　价：79.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：(010) 88378991　88361066　　投稿热线：(010) 88379604
购书热线：(010) 68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

版权所有·侵权必究
封底无防伪标均为盗版
本书法律顾问：北京大成律师事务所　韩光／邹晓东

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了举世瞩目的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出，独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅反映了研究的前沿，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、翻译国外优秀教材上。经过多年的不懈努力，我们与Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com
电子邮件：hzjsj@hzbook.com
联系电话：（010）88379604
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

华章教育
华章科技图书出版中心

译者序

Wireless Sensor Networks: Principles and Practice

随着无线通信、电子与传感技术的发展，无线传感器网络引起了人们的广泛关注，它可以把虚拟世界与现实世界以前所未有的规模进行连接，在国家安全、环境监测、交通管理、空间探索、灾难预防等领域具有重大的应用价值。

本书全面、系统地介绍了无线传感器网络在理论和实践方面的基本原理和经典协议、算法以及应用实例。全书共分为18章，基本覆盖了无线传感器网络所有领域，包括硬件设计、介质访问控制、路由协议、传输协议、操作系统、中间件、数据管理定位、时间同步、安全、执行器/水下/视频传感器网络、能量控制和传感器网络仿真等，介绍了各领域所面临的主要技术挑战和最新研究成果。本书通过列举生动有趣的应用实例，使复杂的概念变得简单易懂，为解决无线传感器网络应用系统中面临的实际问题在架构、协议、建模、分析和解决方案等方面提供了有效的指导。在每章结尾，给出了大量的实际问题和练习，帮助读者们全面理解及掌握相关内容。

本书材料权威丰富，体系结构完整，内容新颖翔实，知识系统全面，行文通俗易懂，兼备知识性、系统性、可读性、实用性和指导性，是一本难得的高层次的教科书。本书既适合作为高等学校物联网工程、计算机、通信等信息技术类专业研究生教材或本科生课程参考书，也适合作为从事相关的研究或开发工作的专业技术人员的高级参考资料。

本书翻译工作由武汉大学计算机学院牛晓光和燕山大学信息科学与工程学院宫继兵合作完成，牛晓光负责前言、第2章至第5章、第11章、第15章至第18章及附录等其他部分，宫继兵负责第1章、第6章至第10章、第12章至第14章。牛晓光负责了初校统稿和最后的审校定稿。感谢机械工业出版社朱劼在书稿翻译过程中的悉心指导和全力支持，感谢武汉大学计算机学院的魏川博和陈曦同学参与协助翻译和清样审校。

限于译者水平，难免有错误和不当之处，敬请读者见谅并给我们提出宝贵意见。

译者

2014年12月于武汉大学

在当今信息爆炸的时代，无线传感器网络（WSN）已成为研究热点之一。科学与机械工程方面的最新进展为建立低能耗、低成本的无线传感器网络提供了有利条件。无线传感器网络为日益增加的应用需求提供了足够的发展空间与有效的解决方案，例如基础设施的保护与安全维护、管理与监控、健康护理、环境监测、食品安全与智慧能源等。相比于传统的网络技术，无线传感器网络具有许多优势，但后者在网络结构的设计、模型与算法等方面也存在很多极具挑战性的问题。无线传感器网络在设计中存在很多限制，例如有限的能耗、带宽、存储空间和计算能力，节点的高失效率与消息丢失率，不利的通信环境与独特的应用要求。目前为止，在学术与工业领域已有很多关于无线传感器网络的研究。

最近，出版了一些传感器网络领域方面的书籍，但大多数并不适合当作教材，因为其覆盖的领域有限或者编辑并不得当。本书则尝试综合讨论传感器网络中的主要技术、标准、主要问题和最新发展。它基本覆盖了此领域读者所需的所有主题，包括硬件设计、介质访问控制、路由协议、传输协议、操作系统、中间件、数据管理定位、时间同步、安全、执行器/水下/视频传感器网络、能量控制和传感器网络模拟等。

本书通过列举生动有趣的无线传感器网络应用实例，使复杂的概念变得简单易懂。另外，本书设计了丰富的课后练习、作业与详实的应用案例，帮助读者理解书中内容，并能够将自己的知识应用到无线传感器网络的设计中或者解决现实生活中的问题。本书还包括一些实际的传感器网络设计，比如医疗健康护理系统。

目标阅读人群

作为教材，本书适合计算机工程、电子工程或者计算机科学、物联网工程等相关专业的高年级本科生和低年级研究生使用。对于想全面了解无线传感器网络技术的传感器网络设计者、研究者和工程师来说，这是一本优秀的参考书。另外，本书也适用于政府部门工作人员，他们可以通过学习此书，利用无线传感器网络保障国土安全。

本书内容

计算机工程知识	第1章 绪论（无线传感器网络概述，基本的网络概念）
	第2章 硬件——传感器节点的体系结构与设计（具有微型控制器和无线通信模块的微型传感器节点）
网络协议栈	第3章 MAC层（相邻节点无线传输）
	第4章 路由层（建立源节点与目的节点间的无线通信路径）
	第5章 传输层（丢包恢复，拥塞控制）
计算机科学知识	第6章 操作系统（TinyOS等）
	第7章 中间件（向应用开发程序人员隐藏网络细节）
	第8章 传感器数据管理
高级无线传感器网络技术	第9章 定位（也称为定标，非常重要）
	第10章 时钟同步（修正传感器节点时钟漂移带来的误差）
	第11章 安全（无线传感器网络面临的攻击及相应安全机制）

（续）

特殊传感器网络	第12章　无线执行器与传感器网络（具有移动执行器）
	第13章　水下传感器网络（重点采用声波而非射频）
	第14章　视频传感器网络
其他技术	第15章　能量模型与低功耗设计
	第16章　无线传感器网络模拟器
应用案例分析	第17章　无线传感器网络应用案例：医疗健康护理系统
	第18章　无线传感器网络应用案例：灯光控制

教学建议

本书可以用于学时为一个学期（15周）的“无线传感器网络”课程教材，下表给出了本书中不同教学单元的时间分配建议。教师可以根据学生的反馈与学习的实际情况调整教学计划。

时间长度	教学内容	章节
2周	无线传感器网络概况、传感器节点硬件（对于计算机科学专业本科生，本内容可以缩减）	第1章与第2章
2周	MAC层（至少讲授两个MAC方案，重点为“能量节省”设计）	第3章
2.5周	路由层（讲授主动式/反应式路由协议，重点为可扩展性设计）	第4章
1.5周	传输层（讲授可靠的端到端传输和拥塞控制两部分内容）	第5章
1周	操作系统；中间件（对于计算机科学专业本科生，可用两周时间讲授本部分）	第6章与第7章
1周	传感器数据管理（对于计算机科学专业本科生，可用两周时间讲授本部分）	第8章
1周	传感器节点定位、时间同步（对于博士/硕士研究生，可用2～3周时间深入讲授本部分）	第9章与第10章
1周	无线传感器网络安全（讲授μTESLA、E-G、q-Composite等经典认证、密钥管理协议）	第11章
1.5周	特殊传感器网络（着重讲授水下无线传感器网络）	第12章到第14章
0.5周	能量模型；无线传感器网络模拟器	第15章与第16章
1周	典型应用案例学习	第17章与第18章
总计：15周	对于每一章，教师应讲授相关数学模型、协议原理与设计实例，可以将一些教学内容作为课后阅读作业	

提示：教师应留一些课堂实验的时间。

对于时长为10周的“无线传感器网络”课程，本书中各教学单元的时间分配建议如下表所示。

时间长度	教学内容	章节
1.5周	无线传感器网络概况、传感器节点硬件	第1章与第2章
1周	MAC层（重点讲授“能量节省”设计）	第3章
1.5周	路由层（讲授主动式/反应式路由协议，重点为可扩展性设计）	第4章
1周	传输层（讲授可靠的端到端传输和拥塞控制两部分内容）	第5章
0.5周	操作系统；中间件（对于计算机科学专业本科生，可用1.5周时间讲授本部分）	第6章与第7章

（续）

时间长度	教学内容	章节
0.5 天	传感器数据管理（对于计算机科学专业本科生，可用1.5周时间讲授本部分）	第8章
1 周	传感器节点定位、时间同步（对于博士/博士研究生，可用2～3周时间深入讲授本部分）	第9章与第10章
0.5 周	无线传感器网络安全（讲授μTESLA、E-G、q-Composite等经典认证、密钥管理协议）	第11章
1 周	特殊传感器网络（着重讲授水下无线传感器网络）	第12章至第14章
0.5 周	能量模型、无线传感器网络	第15章与第16章
1 周	典型应用案例学习	第17章与第18章
总计：10 周	对于每一章，教师应讲授相关数学模型、协议原理与设计实例，可以将一些教学内容作为课后习题作业	

对于计算机工程专业的学生，第2章的内容（传感器节点硬件设计）很重要，需要较多的时间进行系统学习。而计算机科学专业的学生需要细致地学习第6章至第8章内容（传感器网络操作系统和数据管理）。

还有一些章节，例如第8章～第10章（传感器网络定位、同步、安全），可以作为博士/博士研究生的学期考试课题（也就是说，要求学生更深入地研究此课题，然后基于他们的研究提交一份研究报告）。第17章与第18章的内容可以作为大学高年级学生的研究项目。

在课堂教学过程中，不建议对本书涉及的无线传感器网络知识点以综述的形式进行讲解。教师应选择适当的典型设计案例，对案例涉及的相关概念进行详细阐述。例如，当讲授介质访问（MAC）层时，教师应至少详细介绍一种典型的MAC协议（例如S-MAC协议）。

数学理论对于无线传感器网络设计尤其重要。因此，学生应仔细学习各章节中出现的一些经典数学模型。博士/博士研究生更应着重学习这些模型。

MATLAB®为美国MathWorks公司的注册商标，如需了解有关MATLAB的产品信息请联系：

MathWorks公司

3 Apple Hill Drive

Natick，MA 01760-2098 USA

电话：508-647-7000

传真：508-647-7001

电子邮箱：info@ mathworks. com

公司网址：www. mathworks. com

致谢

作为本书的主编，Xiaojun Cao博士编写了本书1/5的内容，Fei Hu博士编写了4/5的内容。我们对在此书的筹备与编写过程中给予帮助的人们表示真诚的感谢。阿拉巴马大学电子与计算机工程学院的学生帮助校对了本书的部分图表与数学公式，他们也协助编辑了本书的部分内容。感谢Rahul Mallampati帮助我们修订本书内容，编辑部分图表，并用微软Word规范了此书格式。还要感谢Barnali Chakraborty和Auerbach出版集团的工作人员，在写作过程中他们提供了长期的支持与帮助。

最后要声明一点，本书的大部分内容与其中的概念都基于目前已有的研究成果，因为篇幅有限，这里我们不能将其一一列出。我们尤其要感谢那些在无线传感器网络领域发表与出版了优秀论文与书籍的作者们。

声明

本书出版的目的是以教科书的形式为学生和工程师介绍无线传感器网络设计的概念与发展现状。需要说明的是，本书并不致力于介绍先进的创新和研究理念。尽管我们尽力利用本书中提到的参考文献，但书中难免存在一些缺点和错误。我们真诚地感谢本书中引用的无线传感器网络领域文献的作者。希望读者在发现本书中的错误时能与我们联系。Fei Hu 的电子邮箱为 fei. hu@ ieee. org，Matt Cao 的邮箱为 cao@ cs. gsu. edu。我们将会在再版中修改和完善书中相关内容。

推荐阅读



云计算与分布式系统：从并行处理到物联网

作者：[美] Kai Hwang 等 ISBN：978-7-111-41065-2 定价：85.00元



嵌入式系统导论：CPS方法

作者：[美] Edward Ashford Lee 等 ISBN：978-7-111-38021-2 定价：65.00元



云计算：概念、技术与架构

作者：[美] Thomas Erl 等 ISBN：978-7-111-46104-0 定价：85.00元



多处理器编程的艺术（修订版）

作者：[美] Maurice Herlihy 等 ISBN：978-7-111-41855-0 定价：59.00元

译者序
前言

第一部分 基础知识

第1章 绪论 …… 2
1.1 基础知识 …… 2
1.2 介质访问控制层 …… 6
1.3 路由 …… 7
1.4 其他通信问题 …… 7
1.5 传感器定位 …… 8
1.6 时钟同步 …… 9
1.7 电源管理 …… 9
1.8 特殊的无线传感器网络 …… 9
1.8.1 无线多媒体传感器网络 …… 9
1.8.2 水下声学无线传感器网络 …… 11
1.9 无线传感器网络的应用 …… 12
问题与练习 …… 15

第二部分 工程设计

第2章 硬件——传感器节点的体系结构与设计 …… 18
2.1 传感器节点的模块 …… 18
2.1.1 传感器 …… 18
2.1.2 微处理器 …… 19
2.1.3 存储器 …… 22
2.1.4 无线通信模块 …… 23
2.1.5 电源 …… 26
2.1.6 外围模块支持 …… 28
2.2 综合设计 …… 28
2.3 Mica节点设计 …… 31
2.4 定制节点——Spec …… 32
2.5 COTS微尘系统 …… 33
2.6 Telos节点 …… 35
2.7 CargoNet …… 36
问题与练习 …… 40

第三部分 网络协议栈

第3章 无线传感器网络中的介质访问控制技术 …… 44
3.1 引言 …… 44
3.1.1 无线传感器网络中的介质访问控制 …… 44
3.1.2 无线传感器网络中MAC设计的挑战性 …… 44
3.2 IEEE802.11标准概述 …… 47
3.2.1 点协调功能 …… 47
3.2.2 分布式协调功能 …… 47
3.3 MAC协议的分类 …… 49
3.3.1 基于竞争的MAC协议 …… 50
3.3.2 基于调度的MAC协议 …… 57
3.3.3 混合型与事件驱动的MAC协议 …… 60
3.4 总结 …… 68
问题与练习 …… 68
第4章 无线传感器网络的路由技术 …… 69
4.1 引言 …… 69
4.1.1 资源受限 …… 69
4.1.2 容错性 …… 70
4.1.3 数据报告与融合 …… 70
4.1.4 节点部署 …… 70
4.1.5 可扩展性和覆盖度 …… 70
4.1.6 网络动态性和异构性 …… 71

4.2 本章的组织结构 …… 71
4.3 无线传感器网络路由协议的分类 …… 71
4.3.1 主动式路由协议和反应式路由协议 …… 71
4.3.2 平面路由协议和分层路由协议 …… 72
4.4 以数据为中心的路由协议 …… 72
4.4.1 泛洪和闲聊 …… 73
4.4.2 SPIN：基于信息协商的传感器网络路由协议 …… 75
4.4.3 DD：定向扩散路由 …… 77
4.5 分层路由协议 …… 81
4.5.1 LEACH：低功耗自适应按簇分层路由协议 …… 82
4.5.2 TEEN：阈值敏感的能量高效传感器网络路由协议 …… 85
4.6 基于位置信息的路由协议 …… 88
4.7 多径 QoS 路由 …… 92
4.7.1 多径路由 …… 93
4.7.2 多径 QoS 路由协议 …… 94
4.8 小结 …… 95
问题与练习 …… 95
第 5 章 无线传感器网络传输层技术 …… 96
5.1 引言 …… 96
5.2 PSFQ …… 97
5.2.1 为什么 TCP 协议不适用于传感器网络 …… 97
5.2.2 基本工作原理 …… 98
5.2.3 协议描述 …… 101
5.3 ESRT：事件到汇聚节点的可靠传输协议 …… 104
5.3.1 可靠传输问题 …… 105
5.3.2 归一化事件可靠性与报告速率之间的关系 …… 106
5.3.3 拥塞检测 …… 110
5.4 E^2SRT：事件到汇聚节点的增强可靠传输协议 …… 110
5.5 CODA：传感器网络中的拥塞检测与避免 …… 115
5.5.1 开环逐跳反压 …… 117
5.5.2 拥塞检测 …… 118
5.5.3 基于点评的信道监听 …… 119
5.6 STCP：无线传感器网络的传输控制协议 …… 119
5.6.1 STCP 中的数据传输序列 …… 119
5.6.2 STCP 分组的格式 …… 119
5.6.3 连续数据流 …… 120
5.6.4 事件触发数据流 …… 121
5.6.5 可靠性 …… 121
5.6.6 拥塞检测与避免 …… 121
5.6.7 以数据为中心的应用 …… 122
5.7 GARUDA：实现有效可靠的下行通信 …… 122
5.7.1 无线传感器网络中下行数据可靠性面临的挑战 …… 122
5.7.2 GARUDA 基本设计 …… 123
5.7.3 GARUDA 架构 …… 125
问题与练习 …… 127

第四部分 计算机科学原理

第 6 章 传感器节点的操作系统 …… 130
6.1 TinyOS …… 130
6.1.1 概述 …… 130
6.1.2 组件模型 …… 131
6.1.3 执行模块与并发性 …… 133
6.1.4 主动消息 …… 134
6.1.5 实现状况 …… 134
6.1.6 主要特性 …… 134
6.1.7 低功率优化 …… 135
6.2 LA-TinyOS：无线传感器网络中的一种局部性感知的操作系统 …… 135

6.2.1 改变定时器以支持时间和空间局部性 …… 137
6.2.2 多级任务调度器 …… 137
6.2.3 LA-TinyOS 系统的代码结构 …… 138
6.3 SOS …… 139
6.3.1 模块 …… 139
6.3.2 动态内存 …… 141
6.4 RETOS：弹性可扩展多线程操作系统 …… 141
6.4.1 应用代码检查 …… 142
6.4.2 多线程系统 …… 142
6.4.3 可加载内核模块 …… 143
问题与练习 …… 144
第7章 无线传感器网络中的中间件设计 …… 145
7.1 引言 …… 145
7.2 无线传感器网络中间件参考模型 …… 146
7.3 中间件实例：Agilla …… 147
7.4 用于获取数据的中间件实例：Mires …… 148
7.5 数据存储实例：DSWare …… 149
7.6 无线传感器网络运行时支持实例：Mate …… 149
7.7 QoS 支持实例：MiLAN …… 150
问题与练习 …… 150
第8章 传感器数据管理 …… 152
8.1 传感器数据清理 …… 152
8.1.1 背景 …… 152
8.1.2 通用模型 …… 153
8.1.3 降低不确定性 …… 154
8.2 TinyDB：应用于传感器网络的可获取的查询处理系统 …… 156
8.2.1 数据模型 …… 156
8.2.2 基本语言特点 …… 156
8.2.3 基于事件查询 …… 157
8.2.4 TinyDB 定义的其他查询 …… 158
8.2.5 基于能量的查询优化 …… 158
8.2.6 TinyDB 策略一览 …… 159
8.3 数据聚合：独立于应用的数据聚合（AIDA） …… 160
8.4 传感器数据存储：层次化数据存储结构（TSAR） …… 162
8.5 多分辨率数据处理 …… 164
问题与练习 …… 164

第五部分 高级话题

第9章 传感器定位 …… 168
9.1 引言 …… 168
9.2 定位的基本要素 …… 168
9.2.1 接收信号强度指示 …… 169
9.2.2 到达时间 …… 170
9.2.3 到达时间差 …… 170
9.2.4 到达角度 …… 171
9.2.5 三角测量 …… 171
9.2.6 三边测量 …… 171
9.2.7 多边定位 …… 171
9.3 使用移动机器人进行传感器定位 …… 172
9.4 多维标度节点定位 …… 175
9.4.1 经典多维标度 …… 176
9.4.2 迭代多维标度 …… 176
9.5 无线传感器网络中的定位 …… 179
9.5.1 蒙特卡洛方法 …… 179
9.5.2 算法（1） …… 180
9.5.3 算法（2） …… 182
9.6 无 GPS 环境中的移动无线传感器网络的节点定位方法 …… 183
9.7 高精度低功耗的无线传感器网络定位系统 …… 185

9.8 LOCALE：稀疏移动传感器网络的协同定位估计 …… 189
9.8.1 协同位置估计 …… 189
9.8.2 LOCALE 中的定位 …… 189
9.8.3 局部定位阶段 …… 190
9.8.4 转换阶段 …… 190
9.8.5 更新阶段 …… 192
9.9 无线传感器网络定位的安全 … 192
9.9.1 SeRLoc …… 193
9.9.2 信标套件 …… 194
9.9.3 攻击容忍的节点定位 …… 194
9.9.4 稳健统计方法 …… 194
问题与练习 …… 195
第 10 章 无线传感器网络中的时间同步技术 …… 196
10.1 引言 …… 196
10.2 一般网络（非无线传感器网络）中的时间同步 …… 198
10.2.1 远程时钟读取 …… 198
10.2.2 偏移时延估计方法 …… 199
10.3 无线传感器网络中的时钟同步 …… 200
10.4 同步性能的评估 …… 202
10.4.1 精度 …… 202
10.4.2 协议开销 …… 203
10.4.3 收敛时间 …… 203
10.4.4 能效 …… 203
10.4.5 可扩展性 …… 203
10.4.6 鲁棒性 …… 203
10.5 无线传感器网络同步协议的例子 …… 203
10.5.1 参考广播同步 …… 203
10.5.2 时间扩散同步协议 …… 205
10.5.3 概率时钟同步 …… 207
问题与练习 …… 208
第 11 章 无线传感器网络安全与隐私 …… 209
11.1 引言 …… 209
11.1.1 一般攻击类型 …… 209
11.1.2 物理节点攻击 …… 209
11.1.3 针对无线传感器网络通信协议栈的攻击 …… 210
11.2 攻击与对策示例：虫洞攻击 …… 214
11.3 无线传感器网络安全示例：基于 Blom 模型的方法 …… 220
11.4 广播认证：基于时间的高效的容忍丢包的流认证协议 μTESLA …… 222
11.5 面向传感器节点的实用安全机制 …… 225
11.5.1 TinySec …… 225
11.5.2 MiniSec：一种面向无线传感器网络的安全通信架构 …… 226
11.6 案例：无线传感器网络中的安全时间同步 …… 226
问题与练习 …… 230

第六部分 特殊无线传感器网络

第 12 章 无线传感器和执行器网络 … 234
12.1 引言 …… 234
12.2 传感器－执行器协同问题 … 236
12.2.1 网络和能量模型 …… 236
12.2.2 ILP 算法 …… 237
12.2.3 传感器－执行器协同工作：分布式协议 …… 238
12.2.4 DEPR 概述 …… 239
12.3 层次化传感器－执行器协同工作机制 …… 240
12.3.1 层次化 WSAN 协同工作架构 …… 240
12.3.2 "传感器－传感器"协同工作层次——使用聚类 …… 241

12.3.3 “传感器－执行器”协同工作层次 …… 242
12.3.4 “执行器－执行器”协同工作层次 …… 242
问题与练习 …… 243
第13章 水下传感器网络 …… 244
13.1 引言 …… 244
13.1.1 水下无线传感器网络应用 …… 244
13.1.2 水下无线传感器网络与陆上无线传感器网络的区别 …… 244
13.1.3 网络拓扑 …… 245
13.1.4 声频信号传输 …… 246
13.1.5 水下传感器 …… 246
13.2 水下无线传感器网络协议栈 …… 247
13.2.1 物理层 …… 247
13.2.2 数据链路层 …… 247
13.2.3 网络层（路由层） …… 248
13.2.4 传输层 …… 248
13.3 介质访问控制设计实例 …… 249
13.4 路由设计实例：基于矢量的转发协议 …… 251
13.5 硬件原型设计 …… 253
13.5.1 硬件设计 …… 253
13.5.2 软件设计 …… 254
13.5.3 系统测试 …… 254
问题与练习 …… 256
第14章 视频传感器网络 …… 257
14.1 引言 …… 257
14.2 Panoptes …… 258
14.2.1 视频捕捉 …… 258
14.2.2 视频压缩 …… 259
14.2.3 数据过滤 …… 259
14.2.4 数据缓存 …… 259
14.3 Cyclops …… 259
14.4 视频传感器网络定标 …… 261
14.4.1 确定重叠的程度 …… 262
14.4.2 估计 k-overlap 值 …… 262
14.5 SenseEye …… 263
问题与练习 …… 265

第七部分 其他主题

第15章 无线传感器网络能量模型 …… 268
15.1 基本 WSN 能量模型 …… 268
15.2 基于仿真的能量模型 …… 270
15.3 能量感知路由 …… 273
问题与练习 …… 276
第16章 传感器网络仿真器 …… 277
16.1 GloMoSim …… 277
16.2 SensorSim …… 277
16.3 TOSSIM …… 278
16.4 PowerTOSSIM …… 281
16.4.1 PowerTOSSIM 的结构 …… 281
16.4.2 组件装配 …… 282
16.4.3 CPU 能耗分析 …… 282
16.4.4 PowerState 模块 …… 282
16.4.5 分析工具 …… 283
问题与练习 …… 283

第八部分 案例研究

第17章 案例研究 1：远程医疗服务 …… 286
17.1 引言 …… 286
17.2 远程心电图传感器网络的硬件设计 …… 287
17.3 可靠的 MASN 通信协议 …… 289
17.3.1 增强的基于聚类的 MASN 数据传输 …… 289
17.3.2 MASN 的路由性能 …… 291
17.4 MASN 的软件设计 …… 293

17.5 RFID 和可穿戴传感器的集成 …… 294
问题与练习 …… 298
第 18 章 案例研究 2：灯光控制 …… 299
18.1 引言 …… 299
18.2 Illumimote 系统的传感器 …… 300
18.3 系统结构 …… 301
18.4 校准 …… 301
18.5 系统评估 …… 302
问题与练习 …… 303
参考文献 …… 305
索引 …… 323

第一部分

Wireless Sensor Networks: Principles and Practice

基础知识

第1章

Wireless Sensor Networks: Principles and Practice

绪　论

1.1　基础知识

读者或许已经在其他书籍或者文章中见到过“传感器”（sensor）这个词。本书的目标是介绍具有无线射频（Radio Frequency，RF）通信能力的微型传感器，这些传感器可以组成一个无线网络，即**无线传感器网络**（Wireless Sensor Network，WSN）。那么接下来自然会提出一个问题：为什么无线传感器网络技术发展如此之快？



以下三项技术的结合使得无线传感器网络可以实现：1）微机电系统（MEMS），它使传感器机械部分可以放在一块非常微小的芯片中；2）数字电子技术，它可以让（带有微控制器的）微型芯片具有足够的能力来处理传入的传感器数据（如数据压缩、数据融合和网络操作）；3）无线射频（RF）通信技术，它可以实现多个传感器以多跳方式传递数据。

如图1-1所示，一个无线传感器网络通常包括一个**模拟感知芯片**来感知环境参数信息（如温度、光照等）。一个**微控制器**来执行本地数据处理（如数据压缩）和网络操作（如和相邻传感器进行通信），以及一个**无线射频收发器**用来发送和接收通过无线介质感知的数据。整个传感器可通过电池或其他能源（如太阳能）供电，生命周期一般为几个月到几年不等。

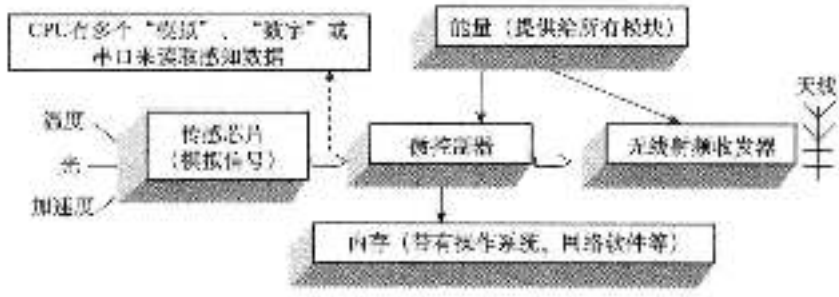


图1-1　无线传感器网络传感器硬件组成

我们将在第2章中详细介绍无线传感器网络中传感器节点的各个模块，这里有几点需要读者注意：

1）图1-1仅列出了无线传感器网络的传感器节点中最重要的模块。根据实际的应用需求，无线传感器网络的传感器节点中可能还包含其他电路组成部分。例如，可以将全球定位系统（GPS）接收器放入传感器节点中以便跟踪获得精确位置，也可以使用太阳能电池板吸收太阳能从而避免使用AA电池等。

2）基于以下事实，读者不要将任何能够感知环境参数的设备都叫做“无线传感器网络的传感器节点”。



模拟传感器、数字传感器和无线传感器网络传感器：1）**模拟传感器**检测环境参数并相应地交自己的电压水平或其他信号，它的输出是一个连续、微弱和带有噪声的模拟信号；2）**数字传感器**自带一个内部模数转换器（ADC）和一个低性能CPU（也称为微控制器），它能与计算机进行连接，展示感知到的数据；3）**无线传感器网络传感器**则是在数字传感器的基础上增加了无线射频通信能力，它的CPU可执行无线网络协议（如路由协议）。此外，无线传感器网络的传感器设计强调的是体积小、成本低和能耗少。

为了构建一个实际的无线传感器网络应用，无线传感器网络的传感器应该具有以下特征：体积小、成本低和能耗少。

1）体积小：无线传感器网络的传感器应便于携带，以满足大规模和便于部署的需求。例如，在疗养院中，每个病人可以携带多个医疗传感器以进行全天候的健康监控。如果这些传感器体积大（比如比一个手机大），病人携带它们就极其不方便。还有一个例子，如果要在一个大城市中进行环境监测，我们可以通过飞机布撒微型传感器，而如果节点体积大，就不容易部署。此外，对于传感器，最好增加其隐蔽性以获得安全且“未遭篡改”的环境感知信息。

2）成本低：即使网络中有大量传感器（数千以上），无线传感器网络也应能运行良好。因此，每个传感器必须保持低成本才能保证其应用普及。未来，单个传感器价格将不超过1美元[Akyildiz02]。

3）能耗少：由于在设计时就考虑到了每个传感器用完即可丢弃，因此无需替换传感器中的电池，在大规模网络中更是如此。如果希望无线传感器网络保持长时间运行，就要有低能耗作为保证。

在本书后面的介绍中，如果没有指定传感器类型（模拟的或数字的），“传感器节点”指的就是“无线传感器网络的传感器”。同时，“无线传感器网络的传感器”也常称为“智能尘埃”（mote）。

无线传感器网络应用范围广泛，包括健康、军事、国家安全及其他领域。例如，医生可以通过一个医疗传感器网络远程监控病人的生理参数。这种措施对病人而言极其便利，而医生亦可实现全天候（7天×24小时）地监控病人的健康状况。无线传感器网络还可以用来检测化学污染，如饮用水中的大肠杆菌含量（E. coli）。一个设计完善的无线传感器网络能够很快发现污染物的名字及位置[John06]。

移动自组织网络（MANET）[CPERKINS00]得到了更多的关注。它的典型实例就是由时刻活动的人们所携带的手提电脑（laptop）所构成的无线网络。考虑到节点的移动性，移动自组织网络的设计目标就是保证其路由协议能够适用于快速变化的网络拓扑结构。

那么，无线传感器网络和移动自组织网络的区别是什么呢？



无线传感器网络和移动自组织网络[Akyildiz02]：

- 无线传感器网络中传感器节点的数量可比移动自组织网络中的节点数量高几个数量级。因此，前者可以进行稠密部署。
- 由于低成本的设计目标，无线传感器网络的传感器节点易于失效。但是，移动自组织网络节点（如手提电脑）被设计成具有强大的计算能力。

- 大多数无线传感器网络应用不需要具有移动性，也就是说，传感器节点是固定的。但是，移动自组织网络节点则具有高度移动性。
- 无线传感器网络的传感器节点在能量（常常是由电池导致的）、计算能力（CPU 主频低）和内存（通常小于 100kB）方面严重受限。

5

与移动自组织网络相似，由于传感器节点间无线通信距离的限制，无线传感器网络必须采用逐跳通信机制。例如，目前大多数无线传感器节点的数据传输距离小于 300ft。因此，一个远程传感器节点不可能直接（通过单跳通信）与服务器进行通信。

除了无线信号广播距离的限制外，从能量消耗的角度看，多跳通信方法要好于单跳方法，因为信号的能级会随着距离的增加而快速衰减：

$$\mathrm{RSS} \propto \frac{1}{d^{\alpha}} \tag{1.1}$$

其中，RSS 表示接收器中的**信号接收强度**（received signal strength），d 表示发送器与接收器之间的无线信号传播距离，α 是**路径损耗比**（path loss ratio），其值一般为 2 - 5。

路径损耗比的值越大，RSS 就越小。路径损耗比 α 随着无线传播所在地形和天气条件的不同而变化。

如果 $\alpha = 2$ 并将距离 d 增大 10 倍，那么 RSS 就会比原值减小 100 倍。因此，可进行如下假设：



在无线传感器网络中，基于数据转发的多跳通信--般比直接发送者—接收者（单跳）的通信节省更多能量。同时也要记住：在无线传感器网络中，首先要关心的是能耗问题。这也就是有如此多的无线传感器网络通信机制以提高能效为主要目标的原因了。

无线传感器网络中有不同种类的传感器，它们能测量机械、热、生物、化学、光学和磁参数。这些传感器附着在传感器节点上用以测量环境参数。有些情况下，执行器（锚节点）基于传感器输入执行某种响应。但是，如果传感器通信是通过其他功能强大的硬件模块（如执行器）来进行的，那么在通常情况下设计无线传感器网络所要考虑的问题（如低能耗、低成本和短距离通信）就不存在了。本书关注的是资源受限的一般无线传感器网络［Jennifer08］。

需要注意的是，无线传感器网络的**设计和资源受限**（design and resource constraint）问题严重［Akyildiz02，CPERKINS00］。根据文献［Jennifer08］中的定义，**资源受限**（resource constraint）指每个传感器电源供应严重受限、无线通信距离短、网络带宽低、CPU 处理能力低以及存储容量小。**设计受限**（design constraint）是考虑到环境条件和应用需求而产生的。例如，室内环境常有很多障碍物，这就导致此种情况下的无线通信质量比户外环境中低。

6



每个摄像机利用光传感器来捕获像素，那么能够将采用无线通信的**多摄像机网络**（multi-camera network）称为传感器网络吗？如果摄像机不存在资源严重受限的情况（例如，其内存高于 1GB，且 CPU 运行在 16 位以上总线宽度上），我们一般会将这样的网络叫做自组织网络或一般无线网络（而不是无线传感器网络）。然而，如果每个摄像机存在严重的资源受限的情况（例如，

具有8位CPU、小于100KB的内存，以及无线通信距离受到限制），那么它就是一个无线传感器网络。最近，有人提出了视频传感器网络（Video Sensor Network，VSN）的概念，这种网络中包含很多低成本的视频传感器。VSN是一种特殊的无线传感器网络。需要记住的是：无线传感器网络的设计因为资源严重受限而面临诸多挑战。如果这些限制不存在了，就很容易借用传统无线网络的设计思想。


案例研究

每个机器人携带类似成百上千的传感器，那么能将多机器人系统（multi-robot system）称为无线传感器网络吗？正常情况下，由于以下原因，不会将多机器人系统称为无线传感器网络：虽然每个机器人就有体积较小、内存低和CPU速度慢的传感器，但它还有其他的电路元件来进行强大的CPU计算和长距离无线通信。因此，并不是所有的无线网络功能都由传感器实现。这种情况下，我们将多机器人系统称为移动自组织网络。


案例研究

每辆汽车都具有上百个微型传感器，那么能够将多车辆网络（Multi-vehicle network）称为无线传感器网络吗？如果我们仅研究由不同车辆上这些微小、具有无线通信功能传感器构成的无线网络，那么可以将多车辆网络称为无线传感器网络。车辆经常使用强大的射频无线来保持它们的通信，但主要的挑战是由于车辆移动性而产生的动态网络拓扑结构。因此，我们将这些车辆的集合称为车辆自组织网络（VANET）而不是无线传感器网络。

在理解了一般的无线传感器网络概念之后，问题自然就出现了：为这些资源严重受限的微型传感器设计网络协议所面临的挑战是什么？为了回答这个问题，本书将先介绍网络协议的概念，然后解释在设计每个协议层时面临的挑战。若想了解关于协议的更多细节，请读者参考其他相关教材，如《Computer Networks》、《Wireless Networks》、《Digital/Data Communications》等。

如图1-2所示，假设发送者（传感器）向远程服务器（接收者）报告事件数据（如火灾）。发送者需要使用多跳通信，通过多个中间传感器来传递数据。根据开放系统互联（Open Systems Interconnection，OSI）标准，共有七层网络协议，即应用层（application layer）、会话层（session layer）、表示层（presentation layer）、传输层（transport layer）、路由层（routing layer）、数据链路层（data link layer）和物理层（physical layer）。然而，一般的无线传感器网络并不需要会话层和表示层，如图1-2所示，在接收端仅需要以下五层协议就能实现感知数据的成功采集：

1）应用层：接收者需要将数据在屏幕上显示出来。应用层定义了传感器数据的显示格式，并管理传感器数据库。如果传感器数据需要显示在互联网页面上，应用层就需要兼容互联网应用层协议，如HTTP。

2）传输层：TCP是典型的传输层协议。传输层的主要功能包括：①实现“端到端”（End-to-End，E2E）的可靠数据传输；②减少网络拥塞。TCP通过报文重传（packet retransmission）和超时检验（time-out check）机制来保证“端到端”可靠性传输。TCP还通过控制数据速率以减少网络拥塞。但是，因其开销大，TCP并不适用于传感器网络。本书第5章将会详细讲解传感器网络中的传输层。

3）路由层：它在多个传感器之间实现逐跳数据转发。它搜索低能耗、低延迟或者具有其

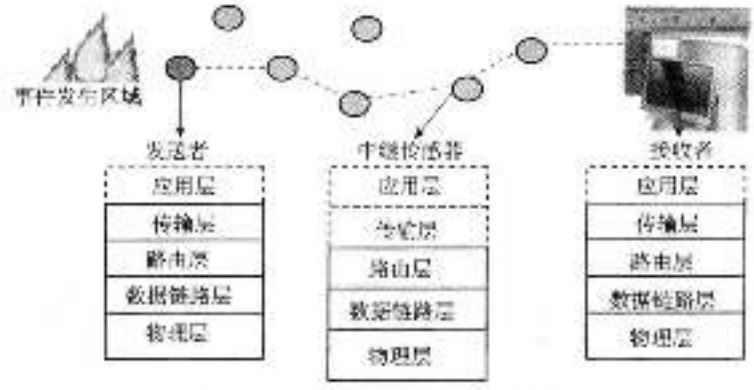


图 1-3 无线传感器网络网络协议栈

他优势的最优传输路径。最优路径一旦建立，感知数据就能一个接一个地通过传感器传递出去。路由层还维护路由以应对网络状况随时可能发生变化的情况（例如，路径上某个传感器可能电池耗尽）。

4）数据链路层：传输层负责“端到端”传输控制，而数据链路层仅处理相邻（1 跳距离）节点的通信问题。例如，一个传感器可以根据它的上行和下行传感器缓冲区设置来决定是否需要调整自己的发送速率。数据链路层有时称为介质访问控制（Medium Access Control Layer，MAC）层。实际上，MAC 仅处理距离为 1 跳的相邻节点的无线介质共享问题，就这点而言，它仅是数据链路层的一部分。当数据链路层执行差错检测、数据成帧和其他任务时，MAC 则保证所有相邻传感器不会发生信号传输冲突问题。

5）物理层：它通过编码/调制和其他无线通信模块将有用的数据转换为无线信号。因为物理层仅能看到“信号”（如代表电压水平的“0”或“1”），所以在此层中不能考虑任何更高层的问题（如路由、数据内容和可靠性等）。



同互联网一样，无线传感器网络也需要以上五层协议。需要注意的是：一般情况下传感器并不运行应用层协议，因为显示传感器数据是服务器的任务。处于发送者和接收者之间的用于转发数据的传感器也不应该运行传输层协议，因为传输层仅存在于两“端”（源节点和目的节点）之间。图 1-2 中使用虚线方框表示这些不存在的层。

在后面的介绍中，本书将概括介绍无线传感器网络各层的设计问题，Akyildiz 等人［Akyildiz02］则对此给出了更全面的概述。本书还将涉及其他重要的问题，如传感器定位等。

1.2 介质访问控制层

MAC 协议在一个共享的无线信道上协调信号传输［John06］。当一组传感器使用同一个无线信道通信时，因为在任何时候只能有一对用户使用该频率相互发送数据，所以必须由 MAC 协议进行调度并制定规则。MAC 协议决定无线信道占用的时间长短以及其他事务。

最常用的信道共享解决方案是**基于竞争的机制**（contention-based scheme）。通过该机制，要传输消息的传感器先监听信道是否空闲（也就是说，信道不忙），如果信道空闲，它就立即传输数据；如果信道忙，它就等待（有时使用指数退避）并稍后重试。

在许多无线网络 MAC 协议中，在给定时间帧内不发送或接收任何数据包的传感器将会进入**睡眠模式**以节省能量。一些信道共享的变种解决方案设计就是基于这种睡眠机制的。要点是无线传感器网络 MAC 机制应该是能量高效和无冲突的，并具有低复杂度调度控制和低内存需求，还能够适应变化的无线信道和网络状况。

1.3 路由

无线传感器网络使用多跳路由来转发数据。传统的路由机制，如互联网协议（Internet Protocol，IP），并不完全适用于无线传感器网络。举例来说，IP 是基于高可靠性有线（如光纤或电缆）连接的，这种情况下很少发生数据包错误。然而，无线传感器网络不是这样，因为无线链路具有高误码率（bit error rate）。一些 MANET 路由解决方案也不适用于无线传感器网络，因为它们是针对高度移动节点而优化的，并且是以两个相邻节点之间的对称链路（也就是说，如果节点 A 能够可靠地连接到 B，那么 B 就能够可靠地连接到 A）为前提的。这个前提并不适用于节点为固定的无线传感器网络。因此，无线传感器网络需要全新的路由解决方案［John06］。

对无线传感器网络而言，因其常以一种自组织（随机）的方式进行部署，所以相应的路由协议一般都是从发现邻居传感器开始的。传感器发出多轮 HELLO 消息（数据包）并建立本地邻居表（neighbor table）。邻居表通常包含以下信息中的一部分：每个邻居的 ID、位置、剩余能量和该传感器保持的延迟和链路质量估计［John06］。

1.4 其他通信问题

除了前面介绍的基本协议，还有一些其他通信问题［John06］：

1）可靠性（reliability）：因为无线通信的不可靠性，所以每个无线链路都具有高误包率，其值可能是 1/100（换句话说，由于无线干扰，100 个包中可能有 1 个被破坏）。应该如何衡量链路质量？我们使用丢包率（packet drop rate）和信号接收强度（received signal strength）等指标解决这个问题。另一个造成链路不稳定的原因是无线传感器网络链路通常是**非对称的**，也就是说，即使传感器 A 能成功地向 B 发送数据包，从传感器 B 到 A 的链路并不一定可靠［John06］。 10



通常在传输层实现数据传输的可靠性。发送者在发出一个数据包后就调用一个计时器（timer）。当计时器到期但没有接收到确认包时，发送者重传该数据包。

另一方面，在无线传感器网络中，端到端（E2E）重传机制并不适用，因为网络中转发传感器数量多且它们之间无线链路不可靠。因此，最好是在每跳之间进行重传而不是等待目的传感器反馈后再重传。这种情况下，可以说是在数据链路层而不是在传输层中实现了可靠性。

2）设计合适的唤醒/睡眠调度时间表：节省能量的最好办法是让传感器进入睡眠模式。然而，该任务的挑战在于：基于实际的数据传输时间状况，如何确定一组相邻传感器的唤醒/睡眠调度时间表。

3）单播（unicast）、组播（multicast）和选播（anycast）语义：在某些情况下，无线传感器网络服务器会将消息传递给一个包含多个传感器的目标地理区域内。那么，该服务器是应该与区域内特定的传感器通信，还是与这个区域内的所有传感器进行通信呢？针对这个问题有以

下几个选择：①在传感器消息中包含一个特定的目标地址（或传感器 ID），从而实现单播通信；②可以指定将消息发送给少数几个传感器（或传感器 ID），这是组播通信；③还可以指定一个事件区域，并向该区域内的任何节点都发送消息，这是选播通信。有时，我们不指定任何目标，仅将命令消息广播（洪泛）到整个网络。多数无线传感器网络路由机制支持上述的单播、组播、选播和广播通信 [John06]。

4）实时性：在一些无线传感器网络应用中，在一个指定的延时阈值内将数据传送到目的地是非常重要的。例如，如果病人心脏病发作，EKG（electrocardiogram）数据应该在 1 秒内传回到医生那里。

5）移动性：在大多数无线传感器网络应用中，传感器是固定的。如果它们是可移动的，
11 那么如何设计协议以满足大规模可移动网络拓扑的需要是一个极大的挑战。

6）断续：无线传感器节点的无线传输范围有限，因此在消息传递的路径上有可能缺失可用于转发数据的传感器。或者，这些转发传感器耗尽了电池能量而无法再工作。路由协议应能处理这种无线链路断开情况。

7）安全性：恶意用户可能对无线传感器网络协议实行多种多样的攻击。举个例子，攻击者可以将自己变成一个合法的转发节点，然后故意丢包。由于无线信号传输是不可靠和基于广播的，因此安全性是任何无线网络的重要研究领域。

8）拥塞：由于一些区域事件发生频繁，因此无线传感器网络在这些区域的流量密度高。好的路由协议应该尝试避开这个拥挤的区域传递数据。如何检测拥塞区域以及如何避免这些区域是两个具有挑战性的问题。

1.5　传感器定位

节点定位（node localization）：在无线传感器网络中往往需要确定传感器的精确位置。如果检测到事件发生，需要知道传感器的精确位置。节点定位有几个问题需要考虑。例如，如何有效地利用信标（位置已知的节点）来找到其他节点的位置？如果使用信标节点（beacon node），又如何确定它们的通信范围？基于不同的定位精度要求（例如，小于 5 米或者小于 1 米），需要有不同的定位算法。系统是在户内还是在户外？是二维（2D）还是三维（3D）的定位问题？定位算法的通信开销是多少（或者说，在单位时间内使用多少个命令消息）？定位一个传感器需要多长时间？此外，还有其他一些问题需要考虑 [John06]。

在户外应用中，可以给每个节点配备一个 GPS。该方案看似简单，但会提高传感器的成本，因此，大多数无线传感器网络应用并不采取这个方案。定位方案分为基于距离的和与距离无关的两种类型。在基于距离的方案中，先设定范围，即两个节点间的距离，然后利用几何原理计算精确的位置。下面是这个方案的一个例子：使用一些专门的硬件或电路来检测声音和无线电波到达的时间差，再将这个差值转换成距离相关的度量。在距离无关的方案中，不需要直接设定距离，而是使用跳数。一旦获得了跳数，就能够估算在每跳平均距离下两个节点之间的距离。
12 显然，后者或许不如前者定位精确，但后者不需要在传感器中增加额外的硬件 [John06]。



设计一个新的无线传感器网络协议时，要时刻考虑到传感器/系统的低成本需求。例如，虽然为每个传感器增配一个 GPS 能够轻松地解决很多问题，但 GPS 需要昂贵的卫星通信系统来接收时间/位置信息。目前，一些商用的传感器价格仍在每个 100 美元以上，而无线传感器网络发展的长期目标是将传感器价格降低至 1 美元以下，以便实现大规模部署。

1.6 时钟同步

无线传感器网络中每个传感器的时钟都应在相同的时钟控制方案下进行工作［John06］。在某些情况下，需要知道事件发生的具体时间，也需要精确的时间来完成某些网络任务。例如，在传感器准备启用睡眠/唤醒机制时，就需要知道什么时间进行睡眠和唤醒。在一些定位算法中，需要测量时间差。

微型传感器内部的时钟控制硬件/软件经常会发生时钟偏移，因此，有必要定期地对时钟读数进行同步。

传统的互联网使用网络时间协议（Network Time Protocol，NTP）［DLM91］来对不同网络主机上的时钟进行同步。但因为它需要频繁地交换消息，所以对于无线传感器网络，NTP过于复杂（需要更多的内存和计算开销），而GPS又成本过高。目前有些学者已提出一些好的时钟同步协议，如RBS［JElson02］、TPSN［SGaneriwal03］和FTSP［MMaroti04］。我们将在第10章中进行详细介绍。

1.7 电源管理

现在，大多数商用无线传感器网络节点（如Mica2和MicaZ［Crossbow08］）使用两节AA电池供电。如果持续执行感知任务而又没有一个好的电源控制机制，这些传感器在几天内就会耗尽电源能量。然而，大多数无线传感器网络应用要求电源寿命超过几个月甚至一年以上。因此，传感器中的电源管理就变得十分重要。

> 如今，可再生能源的研究已成为一个热点。这意味着可以在传感器中使用太阳能电池，还可利用传感器的动能点风能。例如，水下传感器能存储来自水流的能量。增强电池效率和降低电路功耗的技术每年都有新的进展。许多传感器产品支持对传感器中的每个模块（如模拟传感器芯片、无线收发器和微控制器）设置多种节能的状态（关闭、空闲和打开）。这些模块仅在执行任务时处于活动状态。

其他的节能方式涉及以下几个方面：因监听消息也会消耗能量，所以可以让传感器处于完全的睡眠状态；精确设计唤醒/睡眠调度机制，以便传感器在需要转发数据时才被唤醒。

因为每次无线传输都要消耗能量，所以若要达到节能的目标，无线传感器网络协议就要设计为最小化控制消息交换。因为CPU计算也消耗能量，所以有些协议要避免出现复杂的算法。现在，读者也就能理解为何那么多无线传感器网络协议都具有“高能效”（energy-efficient）这个特征了。

1.8 特殊的无线传感器网络

无线传感器网络的种类很多。例如，如果传感器具有视频捕捉功能，那么这种传感器网络称为视频传感器网络（Video Sensor Network，VSN）。在以下两节中，将重点介绍两类特殊的无线传感器网络：**无线多媒体传感器网络**（multimedia WSN）和**水下无线传感器网络**（underwater WSN）。

1.8.1 无线多媒体传感器网络

无线多媒体传感器网络（Wireless Multimedia Sensor Network，WMSN）是一类特殊的无线传感器网络技术［Akyildiz07，Purushottam07］。此类应用对传统的无线传感器网络设计提出了

很多挑战。顾名思义，该网络通过传感器采集多媒体（视频/音频）数据。相对于传统无线传感器网络所获得的数据（如浮点值）而言，多媒体数据需要大量的存储空间，因此带宽要求也更大。同时，该网络也需要具有更高的处理能力。尽管对资源的要求比传统的无线传感器网络多很多，但此类应用也带来很多好处，在军事和民用方面有大量应用。

无线多媒体传感器网络有很多新的应用，例如：

- 日后可用的相关活动的存储。例如，无线多媒体传感器网络可用来记录犯罪分子作案的过程（如抢劫犯罪）。
- 交通避让、强制执行和控制系统。例如，安装在交通灯上的摄像头能够监测逃逸车辆的车牌号码，并报告给就近的警察局。
- 先进的健康保健服务。医疗传感器网络［HU03］能用来提供医疗保健服务，这种服务能接收疾病的警报并定位病人所在位置。病人携带着传感器，从而能够让医生远程监控他（她）的各项身体指标的变化。这些指标包括体温、血压、血糖、ECG 和呼吸。另外，远程医疗人员也能通过视频和音频传感器、位置传感器、动作或活动传感器来监控他们的病人。这些传感器都能嵌入随带设备中［HU03］。
- 老年人自动救助。无线多媒体传感器网络能用来监测、记录和检测老人的行为，以使找出他们所患疾病的病因。具有可穿戴视频和音频传感器的网络能随时检测老年病人的健康状况。
- 环境监测。声音和视频传感器能用来监测动植物栖息地等自然环境。此类监测中，信息以时间方式组织后再进行传递。例如，通过图像处理技术，海洋学家能利用视频传感器捕捉与记录沙堤的演化过程［HOLMAN03］。
- 人员定位服务。随着多媒体信号处理技术的进一步发展，具有生物特征的人体视频流和图像能用来确定失踪人员的位置，或者确定嫌疑犯和罪犯。
- 工业过程控制。像图像、温度、压力和其他参数这样的多媒体信息能够用来管理实时序进行的工业过程。视频传感器网络（VSN）能用来监控诸如半导体芯片、汽车、食品或药品的生产制造过程。它的另一个用途是通过使用视频传感器来迅速发现制造过程中出现的错误。此外，机器视觉系统（Machine Vision System）可用来提供机器人操作产品的某些部件的位置和方向。将机器视觉系统和无线多媒体传感器网络相结合能够简化已有的视频监测系统，并增加这些具有连续性、高速率和高分辨率操作要求的系统的灵活性。

设计无线多媒体传感器网络需要考虑以下几个重要因素：带宽需求、能耗、针对具体应用的服务质量需求、支持异质应用的能力、多媒体覆盖、网内多媒体处理和与其他网络技术整合的能力。对传感器网络所采集数据的高分辨率和高质量需求在不断增长，这些需求都要求带宽不断增长。

无线多媒体传感器网络有多种设计方法［Purushottam07］。其中一种方法就是平面的、同构单层的、具有中心存储的设计。采用这种设计方法，网络易于扩展，能够轻易添加一个新的传感器。但它的缺点也很多，包括单点（中心存储）失效、可伸缩性差（因为是单层的和集中式架构）、处理能力有限以及单个传感器的通信范围有限，以上缺点导致其不能满足请求式的网络应用。例如，采用这种设计的监测网络无法根据实际情况唤醒多个摄像头进行目标识别。另外一种设计方法是使用多层网络（multi-tier network），在这种设计下，网络中处于较高层的节点具有对区域内数据进行集中处理的能力。这种设计使网络具有更好的可伸缩性，并且可以满足不同的成本/性能需求。例如，摄像机的性能越高，其使用的频次就越少，但处理图

14

15

片的功能却更强大。有的研究者提出单层分簇的设计方法，此方法中每个簇首节点包含多个传感器。该方法虽然使得处理能力和可见度略有提高，但一个簇首节点却不能使用其他簇首节点所采集的数据。

1.8.2 水下声学无线传感器网络

虽然传统的无线传感器网络部署在陆地上且已有了很多应用，但却不能用于海洋。原因之一就是部署于海洋的网络（又称作水下无线传感器网络）要具有水下生存能力，满足低维护需求，容忍高延迟的传输协议（这是由于在水下使用的是声音信号而不是无线信号）和高误码率，而传统的传感器网络不能满足以上要求。在设计**水下声学无线传感器网络**（underwater acoustic sensor network）时面临诸多挑战。由于网络部署在水中，传感器会被水腐蚀，缺少光照，声波信号（速度约1500米/秒）传输延迟是无线传输（光速）的10^5多倍，连接中断频繁且丢包率高。尽管存在这些挑战，但水下声学无线传感器网络仍然在很多应用中表现出色，包括导航辅助、灾难预防（即海啸预警）、环境监测、井下搜救、战场战术监测和海洋深度勘探。

水下声学无线传感器网络主要有三种类型［Akyildiz04a］：

- 用于洋底监测的固定2D水下声学无线传感器网络：该网络由固定在海底的传感器节点构成。
- 用于洋流监控的固定3D水下声学无线传感器网络：该网络由能够控制下潜深度的传感器构成。这些传感器还可用来监测多种海洋现象（如污染、生物活性和化学过程等）。
- 二维自主水下航行器网络（Autonomous Underwater Vehicles，AUV）：该网络的固定部分由锚节点和那些附着在自主航行器上用于指导航行的附加节点构成。

如本书前面所讨论的，这三种类型的网络通常被称为MANET，原因在于这些航行器具有很强的通信和数据处理能力。

三维自主水下网络用于检测、观测和捕获那些固定在洋底的传感器节点无法有效监控的水下现象或状况。值得一提的是，该网络中的传感器节点悬浮在海洋的不同深度处以观测某一现象。要保证传感器深度可控，可行的方案之一就是将传感器节点通过缆线与浮标相连，调整浮标缆线的长度就可以实现深度控制。虽然此方案使传感器网络易于部署，但过往船只会对浮标产生干扰，又或者在军事应用中浮标能够轻易被敌人找到并摧毁。此外，浮标还容易因天气及其他意外情况而发生变化。 16

基于以上原因，还有一种方案值得考虑，那就是将传感器设备固定在洋底而不是漂浮在海面上。采用此方法，将包含一个能由气泵充气浮标的传感器设备放到洋底。由于压力的作用，浮标能够将传感器带到海洋表面。传感器所处的深度能通过收起和释放缆线的长度来进行调节，缆线两侧连接传感器和锚节点并由传感器设备中的电控引擎来控制伸缩。该方案面临的一个挑战就是洋流会使得设备左右摆动。为实现3D监控的目标，该方案还需要解决多种挑战，包括：

- **感知覆盖**（sensing coverage）：传感器应通过协同的方式调整其下潜深度，以便凭借已知感知范围完全覆盖所要监控的洋流。这样，网络才能够收集到海洋在不同深度中所发生现象的信息。
- **通信覆盖**（communication coverage）：在3D水下无线传感器网络中，有可能无法直接与汇聚节点（sink node）建立连接链路。因此，传感器应能通过多跳路径将数据转发到洋表基站（station）。网络设备就必须统一协调各自的下潜深度并保证网络拓扑结构是相连通的，以保证每个传感器和汇聚节点之间至少存在一条可用路径。

AUV不需要缆索、线缆或远程控制就能工作，因此它们在海洋学、环境监测和水下资源

勘探等领域有着广泛的应用。前人已通过实验证明 AUV 潜水设备价廉，且可以携带多个水下传感器到达海洋中的任何深度。这些优势可以多种方式加以利用来提高水下传感器网络的能力。从研究的角度看，如果将固定水下传感器网络和 AUV 相结合并进一步改进，则需要以下新的网络协调算法：

- 自适应采样（adaptive sampling）：这种算法中的控制技术能够将航行器引导移动到能收集用处最大数据的位置，这称作自适应采样技术。例如，在一个需要调高采样率来监控某一现象的区域内，可以调整传感器节点的密度以适应需要。
- 自配置（self-configuration）：这种算法中的控制过程能够监测到因节点失效而产生的连接断裂。此外，AUV 用于安装和维护传感器网络架构，或者用于向所在网络中添加新的传感器。AUV 还可以充当临时转发节点来恢复网络的连通性。

17



分析比较

虽然水下无线传感器网络也是用“无线”介质来传输数据，但它们与使用无线信号的陆地传感器网络（Terrestrial Sensor Network，这么称呼是为了与水下传感器网络区分开来，它就是通常所说的无线传感器网络）并不相同。通常，未经官方许可而能使用的无线频谱是 433MHz 或 2.4GHz。但是，水下无线传感器网络使用“声波信号”（acoustic signal）作为无线传输介质。而声波信号比无线信号的频率低得多。例如，声波信号可以是 11kHz，在水下环境中它比无线信号传播的距离远得多。

1.9 无线传感器网络的应用

本节将介绍一些无线传感器网络的典型应用［Hartung06，Chehri06，Manish06］。一些重要的应用是关于环境监测的，例如鸟类栖息地监测、污染监测、地震监测、行星探测、水灾监测、森林火灾监测和污染研究。这些应用都与保护人类生活环境息息相关。

我们可以通过在森林中有策略地部署传感器网络来监测引起火灾的源头。之所以能实现这种效果，是因为传感器网络可长时间无需人员照看，具有高效的节能机制，并能利用可再生的能源技术。传感器以分布式协作的方式进行工作，还能解决障碍物（如树木和石头）遮挡传感器“视线”的问题。美国加州大学伯克利分校的研究者已将无线传感器网络应用于火灾环境中（称作 FireBug）［DOOLIN05］。在此应用中，能准确测量重要的环境参数，如待监测火焰所经过地点的相对湿度和温度。这种传感器网络的效果要优于目前的火灾监测系统，后者通常利用高科技机载红外传感器在一个广阔的区域内跟踪火焰及其密度［Hartungob］。

用户可以使用互联网远程监控和观测环境的生物多样性。卫星和机载传感器可以用来大尺度观测生物多样性，这也导致粒度不够细，无法进行小尺度观测，而正是这些小尺度生物多样性构成了整个生态系统的生物多样性。因此，若要实现细粒度（fine-grained）生物多样性观测，就需要在地面上部署传感器网络节点。图 1-3 给出了一个互联网和无线传感器网络相结合的应用场景。

在崎岖地形和极端恶劣的条件下，传感器容易滑落和遭到破坏。哈佛大学的研究人员使用
18 振动传感器来监测地震活动，他们将传感器网络部署在南美的一个活火山上。虽然他们仅使用了单跳的部署策略，但使用了一个相当有效的时间同步协议对监测数据进行精确的关联。研究者希望利用这个系统有效地监测并协助预防火山喷发、地震和其他类似的火山活动［JOHNSON05］。

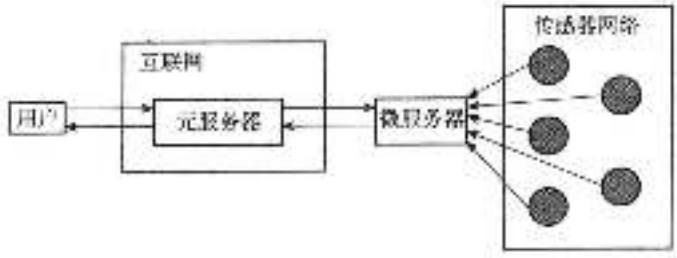


图 1-3 从无线传感器网络连接到互联网

就小尺度监测而言，传感器网络可以部署到一棵杉树上，使用覆盖范围大约 50 米的传感器节点。通过这种独特的部署，研究人员能够观测到一棵树上微气候的变化。

传感器网络还可以部署在自然公园和野生动物保护区中，实施近距离监控并整合那些从动植物身上采集到的数据。早期的野外监测方法容易出错，内容枯燥乏味，并对动植物有潜在危胁。通过分析来自传感器网络的数据，研究人员可以在不伤害动植物的情况下，得到如筑巢方式、开花季节和环境的不同作用等有用的信息。加州大学伯克利分校的研究人员在缅因州海岸的大鸭岛（Great Duck Island）上部署了一个无线传感器网络 [Anderson02]。将传感器放在洞中用来监测筑巢鸟儿的生活模式。这些传感器为生物学家提供统计数据。另外，此项研究工作在传感器网络的性能、路由和协议构建方面提供了有益启示。在这个传感器网络应用中，研究人员使用的是 Mica 传感器节点。传感器节点使用 Atmega103 微控制器，并提供速率为 40kbps 的双向通信。该微控制器运行在 4MHz 和 916MHz 频谱上，将 32 个传感器节点放在监控区域。这些节点将数据传送到网关，该网关负责将数据转发到远程基站。图 1-4 给出了传感器节点在网络忙、安静和非活动三种状态下的时间延迟模拟分析 [Hartung06]。

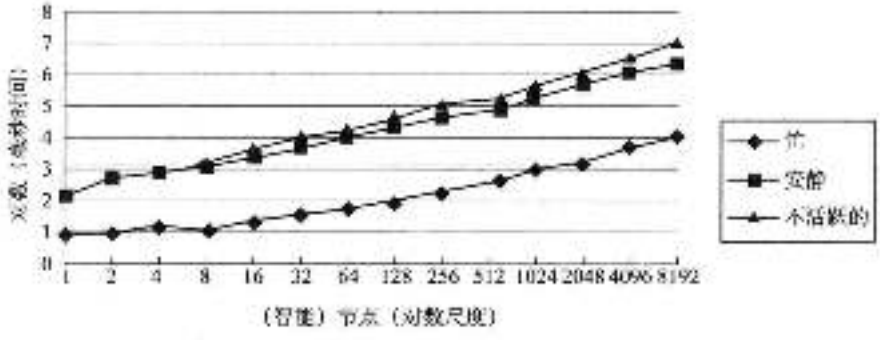


图 1-4 无线传感器网络延迟性能分析（源自 Anderson, J. 等人，Wireless sensor networks for habitat monitoring, Workshop on Wireless Sensor Networks and Applications (WSNA 2002), Atlanta, GA, September 2002.）

加州大学伯克利分校嵌入式网络感知中心（Center for Embedded Networked Sensing）的研究人员在加利福尼亚州的詹姆斯森林保护区（James Reserve Forest）内部署了一个传感器网络，该网络作用极大，可完成从监控土壤温度到跟踪野生动物等工作 [CERPA01]。他们采用多跳路由机制和多种类的异构传感器节点。

目前还有很多面向自然栖息地的监控应用，如用于监控蔗蟾蜍（Cane toad）种群数量的传感器系统 [Buluswo05] 和跟踪斑马活动的传感器网络 [JUANG02]。

无线传感器网络可应用于军事领域（军事进攻和防御）。它能用来采集军队状态的数据，包括手持设备和弹药的数量、军队实力以及军队的位置。将这些报告发送到上一级军队组织

处。由其依据当前的事件状态制定出恰当的决策。用于战场监控的传感器随机部署在可以近距离监测敌方营地的军事禁区和关键区域内。此外，无需人为干预，这些传感器网络可用来发现新的前进路线和路径。

另一个重要的军事应用是目标跟踪。传感器网络可以用于跟踪敌方部队的行进路线。跟踪获得的分析数据可以供智能弹药分配系统使用。当目标（假如一辆汽车）在传感器监测区域移动时，系统可以利用区域内的传感器测量数据对目标状态历史（例如移动轨迹）进行监测，其中每个传感器都可提供一个本地可用测量数据用于目标状态估计，然后可根据测量结果对目标状态进行估算。在进攻开始前或完成之后，传感器网络可部署在战场目标区域以评估战斗破坏程度。传感器网络还可以执行对潜在核武器、生物或化学攻击的预警任务，还可以将应对此类攻击的能力整合到这些网络中。

正如前面所提及的，无线传感器网络能用于病情诊断、药品监管和人体生理参数采集等领域。传感器采集的生理数据可用于药品研发，并且这些数据可以长时间存储。传感器网络还可监测老年人行为，这些小型传感器可协助医生确诊病人已有的病症。并且，每个传感器有其特定的功能，例如，某些传感器专门用于检测心率。用于医院药品监管的传感器网络有助于医院将进错药品和给病人开错药物的机率降至最低。

19
/
20

可以想象传感器网络无处不在，在家庭和办公室中都有它们的身影。这些基于传感器网络的设备可以与执行器相连接，执行器会在环境改变到某一状态时执行特定动作。终端用户可以与这些设备进行通信，远程制定控制决策。在智能家居中，根据周围环境状态的转换，传感器能够智能地进行决策，如需要改变什么以及具体执行什么动作等。晚上，当人走进房间时，电灯会自动打开。办公室的温度能够在几度之内变化。如果房间内的气流不均匀，那么可以利用分布式传感器网络来控制气流和温度。智能传感器和执行器能够嵌入冰箱、微波炉或空调等电器中，这样，终端用户就可通过互联网和卫星管理电器运行。若仓库中的每件物品都附着一个传感器节点，就可以通过网络查询该物品的描述信息，如物品类型、价格和序列号等。这些通过传感器采集到的信息都存储在后端数据库服务器中。新入库的商品将被分配一个传感器以满足仓库管理的需求。

无线传感器网络可以用来监控核反应堆的运转情况。它能控制核反应堆中的链式反应。传感器是通过观测诸如辐射量和温度这样的运转参数来监控核反应过程。监测人员利用获得的数据，并保证核反应堆以稳定状态运行。传感器节点用来感知反应过程的数据信息，并将其发送到检测异常状况（例如辐射量或温度的急剧变化）的汇聚节点上。一旦发生异常，汇聚节点会发出警报。

传感器网络的另一个应用是检测（具有危险性的）嫌疑人员。假设这样一个场景：一个人经常光顾销售化学品的商店，还多次去出售枪支的商店。检测出这样的人并将其列入嫌疑人员名单上，有助于确定这些人的动机和目标。通过恰当地挖掘由放置在各处传感器采集的数据，能够检测出有危险嫌疑的团体或个人。

无线传感器网络在矿井中也有许多应用。传感器能监测矿井环境中各种重要参数，如温度、光照度和氧气浓度等。它们还可以检测可能发生的异常，如火灾和有毒瓦斯气体超标。在每个目标（或者说是“人”）上附着一个小型传感器，传感器网络就能简单高效地对这些目标进行定位。此类功能对于其他应用也很重要，例如，地下矿井中的交通管理和跟踪或救援工作[Chehri06]。

总之，无线传感器网络在各类应用中都扮演着重要的角色，这些应用包括军事进攻和防御、环境监测、建筑自动化、交通管理、工业过程控制、民用基础设施保护以及目标跟踪。无线传感器网络在难以靠近或无法到达的地区特别有用。而有线网络则由于以下两点原因具有诸

21

多限制：1）成本因素：在有线网络部署中，用到的线缆成本会占传感器安装成本的80%；2）安全因素：在一些使用无线传感器网络能够自动采集数据的地方，有线网络很难（或不可能）进行部署和安装［Legg，Chchri06］。



无线传感器网络的优点是不仅具有自组织性质（例如，部署之后，无线的传感器会自动形成一个联通的网络），还具有在苛刻环境下的无线通信能力。本书并不将“有线”传感器网络作为介绍的内容，原因就是无线通信使得传感器网络的协议设计比有线通信更具挑战性。

问题与练习

1.1 多项选择题

1. 以下哪些选项不是无线传感器网络中“模拟传感器”和传感器之间的区别？（ ）
   A. 无线传感器网络中的传感器具有模拟/数字转换功能。
   B. 无线传感器网络中的传感器通过CPU（也称做“微处理器”）完成一些本地数据处理任务。
   C. 传统的模拟传感器通常不需要电源输入。
   D. 传统的模拟传感器无法通过自组织方式加入无线网络。
2. 以下哪些选项不是无线传感器网络（WSN）和移动自组织网络（MANET）之间的区别？（ ）
   A. 无线传感器网络通常比移动自组织网络的规模大（节点多）。
   B. 移动自组织网络更具移动性。
   C. 移动自组织网络的节点通常比无线传感器网络节点有更强大的存储能力。
   D. 无线传感器网络的设计/部署成本比移动自组织网络高。
3. 以下哪些特征可以用来区分水下传感器网络和陆地传感器网络？（ ）
   A. 水下传感器网络通常不使用无线通信，而使用声波通信。
   B. 水下传感器网络的传感器是移动的，而陆地传感器网络中的传感器是固定的。
   C. 水下传感器网络的传感器比陆地传感器网络中的传感器贵。
   D. 水下传感器网络的传感器通常使用太阳能供电。
4. 以下哪些选项是无线多媒体传感器网络（WMSN）所需要的？（ ）
   A. 由于涉及视频/音频数据，因此它们需要更大的存储容量。
   B. 它们需要考虑产学的服务质量。
   C. 它们需要高带宽。
   D. 以上都是。
5. 关于无线传感器网络（WSN）定位的描述，以下哪些选项是正确的？（ ）
   A. 无线传感器网络通常使用GPS进行定位。
   B. 无线传感器网络能轻易达到小于0.1米的定位精度。
   C. 无线传感器网络能够使用三角形原理来定位一个节点。
   D. 无线传感器网络定位并不需要时钟同步。

1.2 请解释无线传感器网络节点的硬件体系结构。传感器使用哪种类型的CPU？请在网上进行搜索后给出几个例子。

1.3 无线传感器网络有哪些设计和资源上的限制？

1.4 为什么水下传感器网络不能使用无线通信？

1.5 假设用无线传感器网络对葡萄园进行监控，请到网上搜索后绘制一幅可行的无线传感器网络系统图（包括传感器、汇聚节点和互联网服务器等）来构建这个应用。

第二部分

Wireless Sensor Networks: Principles and Practice

工程设计

第2章

Wireless Sensor Networks: Principles and Practice

硬件——传感器节点的体系结构与设计

本章将详细介绍传感器节点的硬件设计。无线传感器网络的节点（也可称作智能尘埃）包括模拟传感器、微控制器、存储器、无线射频（Radio Frequency，RF）通信模块、电池等其他组件。因为文献［Jason03］中应用的传感器节点的设计比较先进，所以本书选择其作为范例。

本章将介绍一些无线传感器网络中物理层（physical layer）的概念，如调制与无线信号传输。接下来的几章将详细介绍更高层的内容，如介质访问控制（MAC）层、路由层与传输层等。

在本章中，我们首先介绍传感器节点的每一个模块，然后再介绍由各模块组成的传感器节点。

2.1 传感器节点的模块

[27] 下面介绍传感器节点的硬件模块，任何一个模块的设计都要从性能和能耗两个角度考虑。



从计算机工程设计的角度看，一个传感器节点就是一个典型的嵌入式系统。众所周知，任何一个嵌入式系统都需要一个微处理器（也称为CPU或微控制器）控制所有芯片。另一方面，一个节点需要实现与其他节点无线组网。因此，CPU需要同RF收发器（无线电通信芯片）连接交互。如何使CPU同无线电通信芯片以高速并且低功耗的双向连接交互成为一个具有挑战性的问题。

2.1.1 传感器

目前已研制生成了上万种模拟/数字传感器，它们将配置到无线感知平台上形成无线传感器网络节点。近年来，微机电系统（Micro-Electro-Mechanical Systems，MEMS）与碳纳米管技术的发展促使许多新型传感器问世，比如医疗传感器和数字噪声传感器等。表2-1列出了一些常用的微型传感器及其主要参数［Jason03］。

表2-1 常用传感器的能耗与性能

传感器类型	电流	时间	额定电压（V）	制造商
图像传感器	1.9mA	330μs	2.7～5.5	Taos
温度传感器	1mA	400ms	2.5～5.5	Dallas Semiconductor
湿度传感器	550μA	300ms	2.4～5.5	Sensirion
压力传感器	1mA	35ms	2.2～3.6	Intersema
磁场传感器	4mA	30μs	任意	Honeywell
加速度传感器	2mA	10ms	2.5～3.3	Analog Devices
声学传感器	0.5mA	1ms	2～10	Panasonic
烟雾传感器	5μA		6～12	Motorola
被动红外（移动）传感器	0mA	1ms	任意	Melixis
光合光传感器	0mA	1ms	任意	Li-Cor
土壤湿度传感器	2mA	10ms	2～5	Ecn2o

来源：摘自Hill，J.L.，《System architecture for wireless sensor networks》，PhD dissertation，Department of Computer Science，University of California at Berkeley，Berkeley，CA，Spring 2003.

模拟传感器（analog sensor）和**数字传感器**（digital sensor）有以下不同：

1）模拟传感器根据监测到的物理现象产生原始模拟电压值，电压值形成连续的波形信号，通过特殊的芯片（如 ADC，即模数转换器）将波形信号数字化（即形成数字信号，如 0101001...），数字化后的信号才能被 CPU 和 DSP（数字信号处理）芯片处理。

在接收到原始模拟数据后，CPU 必须将这些数据处理成有意义的数值。例如，加速度计产生了一个 0.815V 的原始电压数值，则必须将其转换为一个有意义的加速度值。也就是说，要弄清楚 0.815V 是对应于 $0.5m/s^2$ 还是 $1.1m/s^2$。该模拟数据的转换过程可能比较复杂，因为不同类型传感器的时钟精度和电压量程都有所不同。

由于输出电压一般在时变信号中具有直流漂移的特点，因此通常使用放大器和滤波器使传感器输出的范围和精确度符合 ADC 的要求。

2）**数字传感器**实际上是把上述的电压处理硬件集成到传感器中，直接提供一个明确的数字接口。由于在其内部已经实现了所需要的补偿和线性处理，因此其输出已经是一个符合合适范围的数字数值。

我们通过利用商用微处理器（CPU）与上述传感器通过接口相连接的方式来设计传感器节点，它通常具备多种接口以便根据实际需求与模拟或数字传感器连接。

由于传感器节点内电池的电量十分有限，而且通常被设计成一次性使用（无法更换电池或进行充电），我们需要严格地控制传感器开启、采样与关闭的频率，因为这些操作对能耗有着巨大的影响。例如，即使大多数传感器有能力进行每秒钟数千次采样，但实际上有意义的只是每分钟内的几个采样结果，这样的**低占空比**（low duty cycle，工作时间所占比例）可以极大地节省能量。

尽可能缩短传感器的工作时间（即传感器的睡眠时间尽可能长）是很重要的，但是尽量缩短“切换”时间也同样重要。换言之，即能够快速地打开和关闭传感器以节省能量。例如，一个传感器用 100ms 打开并进行一次采样，假设采样过程中电流为 1mA，电压为 3V，那么整个采样过程消耗 300μJ 的能量。这和以 1000 mA 电流、3V 电压，但打开和采样的时间是 100μs（即采样速度快 1000 倍）所消耗的能量是一样的［Jason03］。

在某些应用中，传感器的额定电压不能很好地适应电池的输出电压，因此可能需要额外的电路。例如，有些传感器需要 ±6V 的电压，如果一个传感器仅使用 AA 电池或纽电池，那么就需要搭配特殊的**电压转换器**和**调节器**才能让传感器正常工作。同时，传感器自身的能耗和变压及稳压电路的能耗都应计入传感器的能量预算中。

28
29



目前，几乎所有的模拟传感器都是把环境参数转换为可读的低电压电平。但从事件检测的角度考虑，如何解析这些低电压电平是很困难的。而且，还需要捕捉很微弱的电流并使用 ADC 产生数字信号。因此，在模/数转换过程中，应消除来自硬件和环境的噪声。

2.1.2　微处理器

微控制器（microprocessor，MCU），也称作**微型 CPU**、**微处理器**（microprocessor）或**处理器**（processor），是节点中的一个重要模块。它可以通过集成在电路上的引脚（即接口）与闪存、RAM、ADC 以及数字 I/O 接口相互连接。如此密集的集成使得微处理器能够理想地应用于深度嵌入式系统，比如无线传感器网络。

在为无线传感器网络应用选择适用的微处理器时，需要考虑实际应用的需求，包括功耗、电压、成本、对外围模块的支持以及其他外围模块的数量等。

1）**功耗**：不同类型的微处理器在功耗水平方面有很大的差别。例如，8 位或 16 位微处理器的功耗水平为 0.25～2.5mA/MHz，低功耗与标准功耗微处理器之间如此大的功耗差别（10 倍）决定了无线传感器网络的系统性能。

多数人可能会认为“睡眠”可以使 CPU 处于完全“休息”的状态，此时耗电最少。事实并非如此。在睡眠模式下，CPU 停止执行，但它仍保持一些基本的存储器控制活动和时间同步以备定时唤醒。不同的 CPU 在睡眠模式下的电流消耗是 1～50μA。由于 CPU 有 99.9% 的时间都处于睡眠模式，因此睡眠模式下电流消耗在 1～50μA 范围内的差别要比工作模式下功耗峰值时 mA 级的差别对性能的影响更为显著。

30

如前文所述，能耗也受进入/退出睡眠模式的操作时间的影响，切换时间（进入睡眠/唤醒的时间）的范围为 6μs～10ms，唤醒的时间延迟用于启动和稳定系统时钟。CPU 进入/退出睡眠模式越快，节省能量越多。事实上，利用快速唤醒，可以实现让节点在不工作的短间隙内进入睡眠模式。例如，当正在发送一个数据包时，CPU 甚至可以在发送位间隙时进入睡眠模式，这样可以节省大量能量。

2）**电压**：电压的变化也决定着 CPU 的性能。传统的无线传感器网络微处理器能够在 2.7V～3.3V 内工作，新型的低功耗 CPU 甚至可以在低于 1.8V 的电压下工作。无线传感器网络需要承受较宽的电压变化。

3）**CPU 速率**：在无线传感器网络中，CPU 需要运行无线通信协议并且处理本地数据。这些操作不需要高速的 CPU 来完成，这就是目前无线传感器网络的 CPU 的速率都小于 4MHz 的原因。为了选择合适的 CPU，我们需要知道感知数据的总量，CPU 必须在延迟期限内完成规定的操作。

4）**CPU 动态速率**：有些无线传感器网络的 CPU 工作频率（即 CPU 的速率）可以动态变化。CMOS 芯片的耗电遵循公式 $P = CV^2F$，因此 CPU 的高频率会造成更多的电量消耗。但是 CPU 的执行时间和频率成反比，也就是说，高频的 CPU 使程序运行更快并节省电量。所以，不能认为增加或降低 CPU 的频率会使传感器的功耗产生明显的变化。

表 2-2 列出了选择 CPU 时需考虑的重要参数，比如电源、存储器大小、重编程能力、A/D 信道和工作电压等，并对适用于不同节点的 CPU 进行了比较。其中，Atmel AT90LS8535 在大多数无线传感器网络应用中表现出色。

表 2-2 一些微处理器的比较

	Atmel AVR AT90LS8535	Microchip PIC16F877 (Preliminary)	MC68H(R)C 908JL3	Amtel AT91M40400 16/32 Bit Strong Thumb
闪存	4K	8K×14	4K	外置存储器
耐久度	1K	1K	10K	N/A
MIPS/mA	1.25（粉）	1.66（Preliminary）	0.1（typical）	0.6 MIPS/mA（1.35mA 静态电流）
A/D 信道	8（10 位）	8（10 位）	12（8 位）	0
应用中编程（IAP）	否	是	是	是
工作电压	2.7～5.5V	2.0～5.5V	2.7～3.6V	2.7～3.6V
I/O 针脚	35	40	23	100

感谢 Hollar, S. E. – A., COTS dust, MS thesis, Mechanical Engineering, University of California at Berkeley, Berkeley, CA, Fall 2000.



应该注意的是，表中并没有列出所有在不同嵌入式系统中使用的 CPU，仅列出了一些常用的适用于小型、低功耗和低成本节点中的微处理器。在某些产品中，微处理器已经与各种存储器（如 flash 或者 ROM）集成在一起。

CPU 设计实例：传感器网络异步处理器/低功耗（SNAP/LE）［Virantha04］

在文献［Virantha04］中，作者介绍了一个称为 SNAP/LE（Sensor Network Asynchronous Processor/Low Energy）的低功耗微处理器设计方案，适用于无线传感器网络中的数据监测活动。

SNAP/LE 并没有为达到低功耗的目的而随意选择一个传统的微处理器，而是采用了一个自行设计的全新微处理器，能对无线传感器网络中的常见操作提供硬件支持，其目标是将网络寿命延至最长。SNAP/LE 是事件驱动的，在活动/空闲切换时开销极低。该处理器不仅满足了无线传感器网络节点的计算需求，而且比其他 CPU 的功耗更低。

SNAP/LE 的主要特点是使用了自动的、细粒度的电源管理机制——当电路没有执行某个操作时，不会有任何的电路切换活动。这种异步电路还会消除 CPU 内电路转换时短时脉冲波干扰的危险行为，这避免了其他可能的能量浪费。

SNAP/LE 的另一个值得关注的特点是硬件直接支持传感器事件的执行。这意味着不需要任何的操作系统（OS），比如 TinyOS。无操作系统能够减少静态或动态指令数，同时简化 CPU 的设计，这样就不必担心异常和虚拟存储的转换。

大部分传统的微尘 CPU 采用商用 COTS 微控制器，如 Berkeley 的 Atmel Mega128L［Atmel08］。SNAP/LE 不使用商用 CPU，它是专为低能耗 WSN 设计的处理器。它不仅符合 WSN 节点的计算要求，而且能耗小于其他 CPU。

31
/
32



定制的超大规模集成电路（Very Large-Scale Integrated Circuit，VLSI）与 COTS 设计方案相比，不能说哪种方案更胜一筹。通常，从时间和复杂度的角度考虑，多数研究者倾向于选择 COTS，因为许多公司能提供高性能、低成本的芯片用于组成节点。但是，从成本和性能的角度考虑，定制的 VLSI 设计是最好的选择，因为可以使芯片的尺寸最小并可以使速度/能量的性能达到最优。接下来介绍的 Spec［Jason03］同 SNAP/LE 一样，它也是一种定制的设计方案。

SNAP/LE 中 CPU 的设计目标如下：

1）**简单的编程模型。**一个优秀的 CPU 设计方案应该使编程变得更加容易。它的编程模型应支持以下运行模式：无线传感器网络节点在大多数时间内处于睡眠状态，需要定期地唤醒，然后进行无线通信以及感知数据处理。此外，CPU 应能高效地执行无线传感器网络的常见任务，如内部时钟调度和读取感知数据。

2）**超低功耗睡眠模式。**如前文所述，传感器在大多数时间内都保持睡眠状态，因此 SNAP/LE 在睡眠状态下的功耗是极低的。

3）**低开销唤醒机制。**由于快速地在睡眠和唤醒状态之间切换可以节省能量，因此 SNAP/LE 的设计目标是实现约 10ns 的切换时间，这比一般的传感器事件处理时间（约几毫秒）短很多。

4）**活动状态低功耗。**除了实现睡眠状态下的低功耗外，SNAP/LE 还实现了在活动状态下的低功耗。

SNAP/LE 使用 16 位的数据通路，其指令长度可以是一个或两个 16 位字长（双字长的指令

需要两个 CPU 时钟周期来执行）。

33 SNAP/LE 支持多条指令同时执行。图 2-1 为 SNAP/LE 的微体系结构，从中可以了解其潜在的并发性。事件队列存放未处理的事件，指令令牌通过流水线传输然后被计算模块（加法器、解码器等）转换。

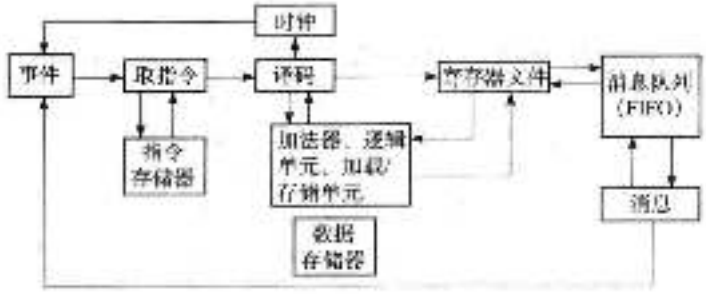


图 2-1 SNAP/LE 的微体系结构（主要部分）（摘自 Ekanayake, V. et al., An ultralow-power processor for sensor networks, ASPLOS'04, Boston, MA, October 7-13, 2004.）

SNAP/LE 采用**数据驱动**（data-driven）切换操作的方式来降低处理器整体的切换量，从而节省了能量。使用异步（即数据驱动）电路进一步节省了能量（为节省钟控处理器的能量，设计者需要通过时钟对处理器内的门电路和锁存器进行控制）。

SNAP/LE 的 CPU 内核包含一个重要的部件，即**事件队列**（event queue），它与**取指单元**形成了一个硬件，该硬件实现了先进先出的任务调度表。该调度表首先执行启动代码，当调度表取得最后一条启动指令中的“完成”指令后，停止取指令，然后等待**事件令牌**（event token）进入事件队列中。

每个事件令牌会说明什么事件已经发生。事件令牌通过两种方式插入事件队列中：1）当定时完成时，被定时协处理器（timer coprocessor）放入队列中；2）来自节点的通信模块或传感器上的数据被**消息协处理器**（message coprocessor）放入队列中。

SNAP/LE 只有一种称为“深度睡眠”的睡眠状态，仅需 10 多纳秒就能够从该睡眠模块唤醒。“深度睡眠”状态以及低唤醒延迟有助于节省能量。其他传统的无线传感器网络的 CPU 不具备这种特性，它们中大多数有多种“睡眠”状态。比如，它们会有一种消耗较少能量的“较深度”睡眠状态，但是其唤醒时间比“较轻度”睡眠所需的唤醒时间更多。Atmel 的微处
34 理器甚至有 6 种睡眠状态。

如图 2-1 所示，SNAP/LE 的 CPU 有如下硬件单元：加法器、逻辑单元、加载/存储单元，定时单元（与定时协处理器交互）、跳转单元、线性移位寄存器（产生伪随机数）和移位器。最常用的单元（如加法器、逻辑单元和加载/存储单元）放置在快速总线上，其余的部分放置在慢速总线上。所有的功能单元都设计成小型流水线以限制 SNAP/LE 在唤醒时的能耗。

2.1.3 存储器

讨论完 CPU 之后，我们讨论节点中的另一个重要组成部分——存储器。通常，无线传感器网络节点仅需少量的存储空间和程序内存。这是因为感知数据只在本地存储很短时间，然后通过网络发送到无线传感器网络基站。

目前，许多 CPU 有片内存储芯片（即闪存），大小通常小于 128K。片内存储芯片既可以用于程序的内存，也可用于临时数据存储。节点的 CPU 还有一个数据 RAM（通常大小为 32～128KB）可用于程序的执行。

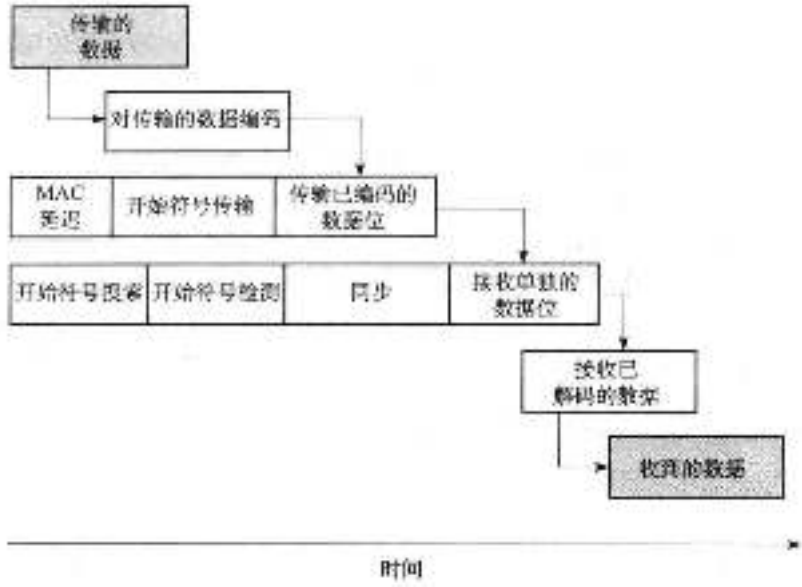


图 2-2　无线通信阶段传输到接收的过程（摘自 Hill, J. L., System architecture for wireless sensor networks, PhD dissertation, Department of Computer Science, University of California at Berkeley, Berkeley, CA, Spring 2003.）

35

闪存与静态随机存储器（Static Random Access Memory，SRAM）的区别如下［Jason03］：

1）从存储的角度考虑，闪存技术有着比 SRAM 更高的存储密度。比如，0.25μm 工艺闪存的存储密度可以达到 150kB/mm^2［AMD03］，而 Intel 公司最新的 90nm 工艺 SRAM 的存储密度为 60kB/mm^2。

2）从能耗的角度考虑，闪存是一种长期存储信息的技术，同时不需要供电。但是 SRAM 需要持续的电能以保存数据（但少于初始化存储器时所需的电能）。

3）从时间的角度考虑，闪存的写操作需要 4μs，而 SRAM 仅需要 0.07μs，两者都消耗 15mA 电流。

因此，如果需要长期保存数据，使用闪存比 SRAM 更有效。

2.1.4　无线通信模块

网络中的节点需要采用无线收发器（radio transceiver），然后通过无线射频（Radio Frequency，RF）通信组成无线传感器网络。节点上使用的低功耗、近距离的无线收发器具有以下特点：

1）发送和接收的功耗为 15 - 300mW。

2）接收与发送时能耗几乎相等。

3）只要无线收发器是开启的，无论是否正在接收数据，都会产生能量消耗。

4）数据包的接收能耗比发送能耗更高。对于传感器节点而言，天线的实际发射能耗（发送数据包）只占无线收发器能耗的很小比例，而无线接收器的能耗占了绝大部分。这个事实在无线通信研究中常常被忽视。

5）如果接收器一直开启，那么它的能耗将占节点整体能耗的绝大部分。**不能认为在没有数据接收时接收器是空闲的。**因此，要尽量使收发器在没有数据接收时处于睡眠状态（即完全

"关闭"状态)。

6) 如果使用较高的发射功率，则可以使信号传输的距离更远。功耗和传输距离的关系是
[36] 一个介于3~4次的多项式(该指数称为**路径损耗**，由于无线信号干扰而不同)。例如，如果在室内的传输距离为原来的两倍，那么消耗的能量是原来的8~16倍。

7) 数据传输距离主要由发射功率决定，但其他因素也会对无线通信距离产生影响。比如无线接收器的接收灵敏度、天线的增益与功效以及信道编码机制。

8) 在大多数无线传感器网络的应用中，出于低成本的需要，我们不能利用高增益的定向天线，因为这样需要特定的校准。所以，一般在无线传感器网络中采用全向天线。

在无线传感器网络中，一般用dBm(并非dB)衡量**发射信号强度**和**接收灵敏度**(注：dB单位是一个对数值，增加10dB表示功率扩大10倍。基准的0dBm表示1mW，所以1W为30dBm)。一般来说，接收器的接收灵敏度在-85dBm~-110dBm[Jason03]。

可以通过以下途径延长**无线信号传播距离**：

1) 增加接收天线的灵敏度。

2) 增加发送者的发送功率。

如果一个发送者的发送强度为0dBm，当接收者的接收灵敏度为-85dBm时，信号可以在户外空旷的空间内传播25~50米；当接收者的接收灵敏度为-110dBm时，距离可以达到100~200米。而且如果把灵敏度从-85dBm调为-100dBm，那么在传输同样远的距离时发射功率可以减少30倍[Jason03]。

目前，无线收发器广泛采用基于压控振荡器(Voltage Controlled Oscillator，VCO)的模式。这一类收发器拥有在各种载频(每一个载频称为一个**信道**)上通信的能力。多信道通信可以有效地抵御各种干扰信号，如果发现一个信道有高噪声，收发器可以立即切换到其他信道。

下面介绍一些关于无线通信的重要技术。

1. 调制方法

在讨论无线通信的时候，会涉及一个重要的术语——数字调制，即将感知数据加载到一个高频的无线载波信号中。如果没有进行调制，数据不能进行远距离传输，也不能很好地抵御噪声信号。

一个典型调制的例子是移动电话，声音信号(低频，小于4KHz)需要加载到高频(900MHz)的载波信号上，这样才能与数公里外的基站塔通信。900MHz的高频信号可以有效
[37] 地抵御障碍物和天气状况等带来的环境噪声(也称为无线信号干扰)。



大部分无线通信系统需使用调制解调器(用于调制和解调的设备，modem)将低频、窄频带的数字信号加载到高频、宽频带的载波信号上(如2.4GHz的信号)。这是因为低频信号不能有效抵御噪声，也不能远距离传输。这里将介绍几种常用的调制方法。实际上，有许多种调制方法，甚至可以用一本书来介绍这些调制方法，但本书只涉及一些基本方法。

调幅(Amplitude Modulation，AM)和**调制**(Frequency Modulation，FM)是两种常用的调制方法。AM不需要复杂的电路就可以简单快捷地对信号进行编码和解码。然而，它极易受噪声的干扰，因为数据只是在载波信号的幅度上(即强度)进行编码。任何外部的噪声都可以改变幅度。相比之下，FM不容易受噪声影响，因为所有的数据都是在同一幅度水平上传输的。

然而，FM也不是抵御噪声的最佳方式。**扩频**(Spread Spectrum，SS)技术能够更加有效

地抵御噪声干扰，它的原理是将信号扩展到更宽的频率范围内。有两种扩频技术：跳频（Frequency-Hopping，FH）和码分多路复用（Code-Division Multiple Access，CDMA）。

在FH中，将宽频带的载波分成许多信道，FHSS依靠伪随机算法不断地改变通信信道。由于攻击者不知道将会切换到哪一个信道，因此很难选择正确的信道加入噪声。每个信道的使用时间称为驻留时间，通常为100μs－10ms。

但是，FH应用于无线传感器网络中时也有不足。例如，它需要较大开销才能保持信道同步以及发现当前的跳频序列（如果一个节点规定了一个专用的信道使用序列，为了正常通信，它必须让其他节点知道该序列，因为所有的通信在特定的时间内必须在同一信道下进行）。如果一个节点试图找出其邻居节点使用的信道，它必须尝试搜索所有的信道。这样的操作开销过高，不适用于低占空比的网络。例如，蓝牙技术采用了FHSS，导致蓝牙设备的能耗较高。

CDMA采用了DSSS（Direct-Sequence Spread Spectrum，直接序列扩频技术），它没有将宽频带的信号分割为小的信道，而是通过高速的伪随机序列与信号相乘，然后直接在宽频带上进行传输。在接收时，收到的信号通过相关器恢复成原始的输入信号。 38

但对于无线传感器网络而言，CDMA同样存在高开销的问题。由于需要持续传输编码，而且信号的解相关也要消耗能量，这需要高比特率的通信，这在低比特率的无线传感器网络中是不适用的。

Lester［Jason03］使用两种商用无线通信模块RF Monolithics TR1000和Chipcon CC1000作为示例，介绍了低功耗无线收发器的能量消耗情况：

- TR1000：1）发射：发射强度为0.75mW时，功率为21mW。
  2）接收：接收灵敏度为－85dBm时，功率为15mW。
- CC1000：1）发射：发射强度为3mW时，功率为50mW。
  2）接收：接收灵敏度为－105dBm时，功率20mW。
  发射强度与TR1000同为0.75mW时，CC1000功率为31.6mW。
- 通信距离：在室外环境下，TR1000最高直线通信距离为300英尺，CC1000为900英尺。
- 寿命：CC1000在不转入睡眠模式的状态下，可以持续发送数据约4天或者保持接收状态9天。如果需要工作一年，CC1000必须在约2%的占空比下运行。

2. 比特率

互联网需要高速传输速率（主干网络的速率超过30Gbps），但无线传感器网络应用中不需要如此高速的通信速率，这是因为大多数情况下传感器仅发送部分数据的数值，所以目前许多节点仅提供10～100kbps的传输速率。

3. 开启时间

在前面的内容中已经强调了无线通信快速进入/退出睡眠状态的重要性。如果不需要发送数据，那么应该使配置/启动无线通信的时间和能量花费尽可能小。

如果无线传感器网络需要探测几秒内可能出现的紧急事件，那么无线通信至少需要每秒打开一次。如果无线通信的打开时间是50ms，那么很难达到占空比低于1%的要求。

另一个值得重视的现象是基于VCO频率合成器的多信道无线收发器必须在发送或接收数据前保持自身稳定，高频锁定的VCO晶振同样需要自身稳定，这显然需要尽可能减少稳定时间。CC1000的高频晶振需要2ms的稳定时间，而TR1000仅需要300μs就可以准备开始工作。这就是TR1000对事件的响应比CC1000快10倍的原因。

表2-3列出了一些常用于无线传感器网络的无线通信芯片。 39

表 2-3　适用于无线传感器网络的无线通信芯片及其参数

	TR1000	CC1000	CC2400	nRF2401	CC2420	MC13191/92	ZV4002
最大数据速率（kbps）	115.2	76.8	1000	1000	250	250	723.2
接收功率（mA）	3.8	9.6	24	18（25）	19.7	37（42）	65
发射功率（mA/dBm）	12/1.5	6.5/10	19/0	12/0	17.4/0	34（30）/0	65/0
睡眠功耗（μA）	1	1	1.5	0.4	1	1	140
开启时间（ms）	0.02	2	1.13	3	0.58	30	注
调制	OOK/ASK	FSK	FSK，GFSK	GFSK	DSSS-O-QPSK	DSSS-O-QPSK	FHSS-GFSK
数据包检测	否	否	可编程	是	是	是	是
地址解码	否	否	否	是	是	是	是
加密支持	否	否	否	否	128 bit AES	否	128 bit SC
错误检测	否	否	是	是	是	是	是
错误校正	否	否	否	否	是	是	是
应答	否	否	否	否	是	是	是
时间同步	bit	SFD/byte	SFD/packet	Packet	SFD	SFD	Bluetooth
定位	RSSI	RSSI	RSSI	否	RSSI/LQI	RSSI/LQI	RSSI

来源：摘自 Hill, J. L., System architecture for wireless sensor networks, PhD dissertation, Department of Computer Science, University of California at Berkeley, CA, Spring 2003; Hollar, S. E.-A., COTS dust, MS thesis, Mechanical Engineering, University of California at Berkeley, Berkeley, CA, Fall 2000.

注：制造商的文档中没有相关信息。

40

2.1.5　电源

电源为节点的正常工作提供必需的电能供应。如果使用电池，有三种常见的电池技术可以用于无线传感器网络——碱性电池、锂电池和镍氢电池［Jason03］：

1）碱性电池——在 AA 碱性电池上标识出的输出电压为 1.5V。实际上，当它工作时，输出电压范围在 1.65－0.8V（使用的时间越长，电压越低），电流可以达到 2850mA。

碱性电池是一种价廉、高容量的电源。但是有些传感器不能承受它较宽的电压变化幅度。它巨大的尺寸也是一个问题。即使没有用于任何设备，它也会自放电，在 5 年之后便不能再使用（由于电压很低）。

2）锂电池——锂电池的尺寸比碱性电池更小（最小的直径仅为几毫米）。另一个优点是其拥有恒定的电压输出，即使电量几乎被用完，它的电压也不会有大的衰减。锂电池还有一个出色的方面是，它甚至可以在－40℃低温下工作，这是碱性电池所无法达到的。CR2032 是最常用的一种锂电池，它的电压为 3V，容量为 255mAh，仅售 16 美分（本书编写时的售价）。

但是，锂电池有一个很大的缺点——放电电流很低。因此，这种电池不能应用于许多需要 1000mA 以上电流的节点。例如，锂电池可以很好地应用于 Crossbow Mica2Dot 节点（Crossbow 公司生产的最小节点），但不能用在 Mica2 节点上。

3）镍氢电池——镍氢电池可以很容易地重复充电，但它也有一些缺点：一支 AA 的镍氢电池的能量密度约为碱性电池的一半，但是成本却是后者的 4 倍；镍氢电池仅能提供 1.2V 的电压，但多数无线传感器网络的部件或节点需要 2.7V 的电压。

表 2-4 列出了上述三种电池的主要特点［Seth00］。

表 2-4　无线传感器网络的电池类型

类型	电压（V）	能量密度（mW-hr/g）	最大电流
碱性电池 AAP107－ND	1.5	90	130mA（24g）
镍氢电池 P034－ND（可充电）	1.2	55	大于 2600（26g）
锂电池	3.0	285	10mA（10.5g）

如果节点是在低电压下工作的，那么电池可以使用较长时间。假设一个节点的功耗是 250mW，额定电压是 2.7V，当设计其部件的工作电压低至 2V，那么在相同的电源情况下它工作的时间约为原来的 5 倍（假设使用 AA 电池）。因此，一个看似不重要的 CPU 参数（即硬件额定电压）都有可能造成多达 5 倍的系统寿命的区别。

随着使用时间的延长，几乎所有电池的输出电压都会下降，因此需要使用调压（voltage regulation）技术，将变化的输入电压转为稳定、持续的输出电压。标准的调压器只能产生比输入电压低的输出电压，但是如果使用升压转换器（boost converter），就可以得到高于输入电压的输出电压。但是调压器也有一些缺点。例如，调压器的静态电流消耗（即在没有电流输出时的能量消耗）也是相对较高的。

如果使用碱性电池，由于在无静态电流消耗的情况下设置一个调压器是很困难的，那么选择能够承受宽幅度电压的模块配置节点是很好的方案。如果节点所有模块都可以在 2.1～3.3V 的范围内工作，那么普通的碱性电池就足够使用了。

除了上述电池类的电源外，能量采集，特别是太阳能采集，作为一种延长无线传感器网络寿命和降低成本的方案正逐步受到重视。在对大型太阳能系统的研究已经取得长足进展的同时，基于微型太阳能采集技术的太阳能系统则一直受到能量收支等多方面的限制。表 2-5 列出了几种微型太阳能系统设计方案，不同的方案应用在不同的需求下，比如寿命、简易性、成本等。Heliomote［VRaghunathan05］和 Trio［PDutta06］是两种主要的微型太阳能系统设计方案。Heliomote 设计的重点是简易性，它使用单级（single-level）能量存储和硬件控制电池充电。Trio 更多地从寿命和灵活性的角度出发，使用两级能量存储和软件控制电池充电。



奇思妙想

能量，能量，能量！

你知道研究与发展最热门主题之一是可再生能源系统吗？人类现在正面临一个巨大的挑战：我们不能只依靠石油！看看无限的能源——太阳！为什么不将它开发出来进行实际应用呢？说时容易做时难。我们需要你——聪明的科学家和工程师，提出可行的和低成本的解决方案开采太阳能、风能、原子能和其他可再生能源。

表 2-5　微型太阳能系统示例

微型太阳能系统	设计目标	特点	来源
Prometheus Trio	寿命、灵活性	两级存储，软件控制充电	［XJian05，PDutta06］
Heliomote	简易性	硬件控制充电，NiMH 电池	［VRaghunathan05］
Everlast	寿命	最大功率点跟踪	［FSimjee06］
RF beacon	概念验证	不支持电力中断	［SBiswas03］
Fuss monitoring	紧凑、可靠、低成本	硬件控制充电，NiMH 电池	［PSikka06］
ZebraNet	移动	软件控制电池充电，Li＋电池	［PZhang04］

2.1.6　外围模块支持

在前文中已介绍了CPU（即微处理器）及其内部的设计规则，同时CPU还有一些针脚用于和其他外部设备连接。主要有以下两种针脚：

1）**数字I/O针脚**：作为基本接口机制，所有的CPU都包含标准数字I/O线，它与无线收发器、存储单元以及其他输出数字信号的模块连接交互。

需要注意的是，在数字I/O针脚中，数字通信协议用于读取数字传感器的数据。但其他一些外设芯片是通过无线信号或RS-232收发器使用串口通信协议连接到CPU的。

总的来说，有三种标准的数字通信协议：UART（Universal Asynchronous Receiver/Transmitter）、I^2C（Inter-Integrated Circuit）和SPI（Serial Peripheral Interface）。I^2C和SPI使用具有精确时钟信号的同步协议，而UART使用异步机制。

2）**模拟I/O针脚**：CPU也有和模拟传感器连接的模拟I/O针脚。对于这些针脚，CPU有内置的模数转换器，用于对采样时间进行精确控制，还能轻易地获得采样结果。如果CPU没有内置的转换器，那么在设计节点时就应该包含一个外置的转换器。

42
|
43

2.2　综合设计

典型的传感器节点体系结构

在前面内容中已经分别对节点各组件进行了详细介绍，现将它们综合起来。总的来说，一个节点主要实现本地传感器数据的计算以及同相邻节点的无线通信。

这一节不会重点介绍某种专门的通信或处理技术，而是介绍能够满足计算和通信需求的节点的体系结构以及设计原则，特别着重介绍能够实现低功耗的硬件设计方法。

1. 无线通信需求

任何一个节点都需要同其他节点进行无线通信。无线信号实际上是原始的电磁信号，无线发射器要用数字调制方法把数据调制到载波上，无线接收器对收到的信号进行解调和数据的解析。

在无线传感器网络中，节点主要发送两种数据：1）从环境中采集的传感器数据（sensor data）；2）控制数据，如无线网络协议。从网络协议的角度看来，这些数据都会压缩成数据位。图2-2描述了包交换（packet-based）无线网络通信协议的关键步骤。应当注意，许多操作必须同其他操作并行执行，就像汽车生产厂商并行地装配零件那样。在图2-2中，可以从时间上明显重叠着的层看出“平行”的性质。

如图2-2所示，编码是通信过程中的第一步。将模拟传感器的数据编码成为数字信号（即，位）以便传输。值得注意的是，编码中还应该包含错误检测/纠正码，当受到无线干扰导致一些位出错时，就可以用错误检测码发现这些错误。

为了缩短发送延迟，编码与实际的发送过程是“流水”进行的，也就是当第一个字节编码完成后，立即启动无线发送，然后就可以在发送前面的字节的同时对新的字节进行编码。

目前，有多种编码方式可供选择。比较简单的是直流平衡（DC-balance）方式，比如曼彻斯特（Manchester）编码。更先进也更复杂的方式是CDMA（已在本章介绍）。无论采用哪种编码方式，数据位（位，0或1）将被分组成不同的单元，称为符号（symbol）。然后将每个符号编码成一组无线传输位，称为码片（chip）。在曼彻斯特编码中，对于1位数据，每个符号用2个码片。CDMA中通常每个符号有15～50个码片，每个符号包含1～4个数据位。

当数据传递到无线通信协议然后准备发送到其他节点时，首先需要执行 MAC 协议，其主 44
要任务是确保相邻节点在发送数据时不发生冲突。CSMA（载波侦听多路访问）就是一个简单的例子。节点在发送数据前对信道进行侦听，如果信道正忙，那么就随机地等待很短的时间再重新开始以上的传输过程。

在 MAC 协议成功地将数据发送出去之后，路由协议负责数据在节点之间的传输，它找出一条最佳的（从节省能量的角度考虑）路径把数据传输到目的地（例如基站）。

当数据在发送者和接收者之间连续传输时，依靠一个精确的时间同步机制，发送者可以精确地控制每位的传输时间，这样接收者可以同发送者保持同步。

当接收者接收到数据后，对其进行解码和解调，将其还原成原始数据，再通过一些噪声清除算法将噪声清除。

2. 关键问题

Lester［Jason03］指出在节点设计时应当考虑一些关键问题：

（1）并发性

为了提高数据处理速度，设计一个提供细粒度并发性的体系结构是很重要的。无论是发送方还是接收方，无线通信的处理应该同应用层的数据处理甚至网络协议处理并行进行。当进行无线通信时，不应暂止必要的操作，如传感器事件检测和计算等。

（2）灵活性

值得注意的是，无线传感器网络在不同的实际应用中有着不同的服务质量（Quality of Service，QoS）需求。有些应用要求实时数据传输，而其他应用则允许有一定延迟；有些应用要求局部数据的压缩，而其他应用仅需将数据发送到汇聚节点；有些应用要求具备安全措施，而其他应用不需要考虑网络攻击。

因此，有必要把节点的体系结构设计为能够支持各种不同的应用环境。传统的嵌入式系统（如手机或蓝牙设备）必须遵循一个固定的通信协议，而无线传感器网络应当允许灵活设计通信协议，以使使带宽、延迟和网内处理达到平衡。

灵活的协议设计是需要灵活的硬件体系结构的，不同的硬件体系结构对于不同的应用产生的效果是不同的。比如，一个图像传感器网络需要大容量存储空间和高性能 CPU，而一个水下网络则需要声波（而非射频）通信系统。 45

（3）无线通信与处理速度解耦

设计节点时不能把无线通信的传输速率和 CPU 处理速率这两项指标耦合在一起。这是因为节点对 CPU 和无线收发器的最优化需求不同：1）无线通信倾向于用最大的传输速率将数据发送出去，因为传输时间越短能耗越少；2）针对低功耗 CPU 设计和动态电压调节的研究已经表明——通过将计算分散使得 CPU 持续在低负荷状态下工作，可以实现在低电压下运行。

因此，从节能观点出发，最优方案是使 CPU 尽可能慢地进行计算，然后当计算完成时，无线通信能够尽快地将数据发送出去。

CPU 和无线通信的解耦是很重要的。如果 CPU 的速度和数据传输的速率是耦合的，那么该系统的两个组成部分只能在非最优化的方式下工作。

3. 传统的无线设计

如今，很多嵌入式系统（如手机、802.11 无线网卡和蓝牙设备）选择使用专用的 CPU 来执行通信协议，以解决并发性和解耦的问题。这种专用 CPU 应在进行如下操作时能实时地执行通信协议：无线调制与解调、编码与解码以及其他操作。

以蓝牙设备为例，主机信道接口（HCI）在 UART（通用异步收发器）上实现一个高层分

组的传输接口，这个接口隐藏了通信同步、信号编码和 MAC 协议等。应设置专用 CPU 的速度满足无线通信协议的需求。

但是，上述的 CPU 运行模式不适用于无线传感器网络，因为它将无线通信和数据计算的资源分割。这会导致资源利用不能达到最优化，芯片-芯片的通信机制是低效率的。

用文献［Jason03］中的节点设计思想可以替换以上方案。不使用专用的 CPU，而是用一个单线程的引擎供应用层和协议层使用。这样通过 TinyOS 中细粒度交错事件处理可以虚拟（非实际）地满足系统的并发需求。

在接下来的部分中将主要介绍文献［Jason03］中涉及的部分节点（如 Reno、Mica 和 Spec）的设计思想。这些节点代表了近 20 年出现的较为先进的无线传感器节点，可以从中学
46 习到在实际应用中表现出色的硬件设计原则。

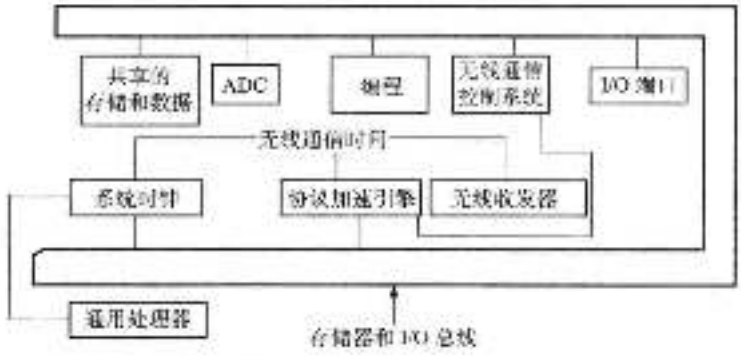


图 2-3 嵌入式无线设备的一般结构（摘自 Hill, J. L., System architecture for wireless sensor networks, PhD dissertation, Department of Computer Science, University of California at Berkeley, Berkeley, CA, Spring 2003.）

4. 节点设计示例：Reno

Reno 是文献［Jason03］中重点介绍的节点。它带有特殊的硬件——加速引擎，能够满足无线通信实时和高速的需求。

图 2-3 大致描述了 Reno 的体系结构。其 CPU 需要控制多并发操作（类似于 Windows 系统中的“多线程”），并且能够有效支持上下文切换（context switching），可以使用寄存器窗口降低上下文切换时的开销。Reno 的 CPU 包含多个寄存器单元，从而避免了将数据从寄存器放入存储器的操作，而是将 OS 简单地切换到空闲的寄存器单元。

如图 2-3 所示，通过一个共享总线将存储器、I/O 端口、ADC、系统时钟和硬件加速引擎等相互连接。正是由于其高速和低延迟的互联，数据可以轻松地在处理器、存储器和外设之间传递。该总线不仅支持直接的 CPU-外设交互，还支持外设直接与外设交互。应该注意的是，外设可以通过共享总线直接从存储器中取出数据，还能轻松地把数据发送到其他 UART 的外设中。

所以，Reno 能通过共享总线对无线通信进行以下改进：共享总线支持进行数据编码的外设直接从存储器中取出数据，再将数据放入数据传输加速引擎中，比如无线通信的调制电路。这与许多计算机运行模式是不同的。在通常的模式下，存储器读/写时需要 CPU 的参与。而在 Reno 中，CPU 不参与通信，CPU 仅需控制数据的传输。这样就将 CPU 从繁重的负载中释放
47 出来。

如果读者已学习过“计算机组成原理”或者“汇编语言”这两门课程，那么就应该知道

可以用相同的编址方式对每一个存储器单元和其他设备进行编址，即一个存储器地址可以是实际的存储器的位置，也可以是一个虚拟的设备数据缓冲区的位置，系统使用线缆把设备的数据缓冲区与物理存储器的位置相连。Reno 就使用这样的编址方式，使得原本不能共同工作的部件以一种新的方式联合在一起。假设数据编码器希望从无线收发器的缓冲区中取出数据，由于这个缓冲区已经映射到一个存储地址，因此编码器只需要从“存储器”中取出数据，然后转换数据，最后再写到存储器中即可。

最后，需要牢记的是，Reno 节点最主要的特点是它包含了有特殊用途的硬件——加速引擎，这样能够以快速和高能效的方式实现底层的操作。通过提高这些操作的效率，系统整体的能耗将大大降低。

2.3　Mica 节点设计

Mica 节点在 Reno 节点的基础上增加了关键的硬件加速引擎，Mica 用硬件加速引擎配合 CPU 提高了数据传输比特率和定时精确性。

Mica 的硬件部分包含一个 Atmega103 微处理器、一个 RFM TR1000 无线通信单元、外置存储器以及通信加速引擎。硬件加速引擎可以提升无线通信关键步骤的性能。

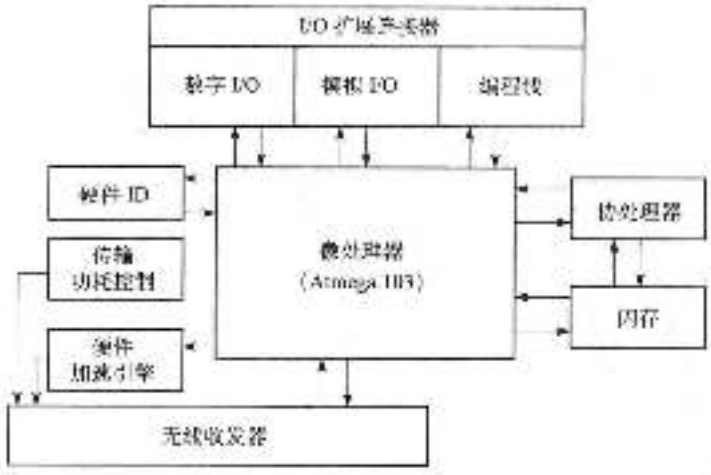


图 2-4　Mica 体系结构框图（摘自 Hill, J. L., System architecture for wireless sensor networks, PhD dissertation, Department of Computer Science, University of California at Berkeley, Berkeley, CA, Spring 2003.）

图 2-4 描述了 Mica 节点的体系结构。它有 5 个主要功能模块：CPU、无线通信单元、能量管理单元、I/O 扩展和二级存储器。读者可以在 http://www.tinyos.net 找到 Mica 节点的主要功能模块简介、系统整体概述、详细材料清单、设备示意图和所有硬件模块的技术手册。

Mica 节点使用 Atmel ATMEGA1031 或者 ATMEGA128（4MHz）。Atmel CPU 还包含一个 128KB 的 flash 程序存储器、一块 4kB 的静态 RAM、一个内置 8 通道 10 位模数转换器、3 个硬件时钟、48 条通用 I/O 线、一个外置 UART 和一个 SPI 端口。Mica 节点的无线通信模块包含一个 RF Monolithics TR1000 收发器。

- **节点 ID**：为了给每个节点分配一个唯一的标识，Mica 使用一个 Maxim DS2401 硅序列号作为标识。这是一个无需供电的带数据电子接口的低成本 ROM 设备［Dallas08］。
- **存储器**：Mica 使用一块 4Mbit 的 Atmel AT45DB041B 系列小型封装的 flash 芯片。该 flash 存储器负责存储传感器数据和应用程序两类信息。一般情况下，flash 存储器应比

128KB 的程序存储器大，这样才能容纳整个程序。这就是 Mica 不采用 Reno 使用的小于32KB 的电擦除可编程的 ROM 存储器的原因。

- **能量供给**：Mica 节点能够由 AA 碱性电池驱动，但需要提升输出电压。如果没有升压转换器，无线通信模块就不能运行。Mica 使用一个 Maxim1678 直流-直流转换器提供恒定的 3.3V 电能供应，该转换器能够接收最低 1.1V 的输入电压。值得注意的是，输入电压对无线收发器（TR1000）的发送信号强度和接收灵敏度有显著影响。

表 2.6 描述了 Mica 节点中不同硬件模块的功耗水平。当节点处于超低功耗的睡眠状态下时，电源系统是关闭的。此时整个系统在未调输入电压下运行，这样可以降低升压转换器和 CPU 的能耗。

表 2-6　Mica 节点硬件在活动及空闲时能耗状况（3V）

硬件设备	活动（mW）	空闲（μW）
CPU	16.5	30
Flash drive	45	30
LED	10	0
无线通信模块	21（TX），15（RX）	0
硅序列号	0.015	0

来源：摘自 Hill, J. L.; System architecture for wireless sensor networks, PhD dissertation, Department of Computer Science, University of California at Berkeley, Berkeley, CA, Spring 2003.

- **外围模块支持**：Mica 节点的 I/O 子系统接口包含一个 51 针扩展连接器，这些针脚可以使节点与各种不同的传感或者编程电路板连接。该 51 针连接器有以下接口：8 条模拟连接线、8 条电能控制线、3 条脉冲宽度调制线、2 条模拟比较线、4 条外部中断线、1 个串口、一组微处理器编程专用线和一些总线接口。
- **无线通信模块**：Mica 使用一个 TR1000 无线芯片。该芯片允许 CPU 直接获得无线接收时的信号强度，还允许 CPU 在没有活动数据传输时对背景噪声的水平进行采样。在多跳网络应用中，可以通过这些信息（无线信号强度和噪声水平）计算信噪比，从而大大提高路径效率。

Mica 允许程序快速并有预测性地打开或关闭无线模块。因此，Mica 节点可以在没有全局控制的情况下轻易进入到低占空比下运行。

2.4　定制节点——Spec

使用商业成品（COTS）的部件配置节点虽然快速、简单，但是从生产成本、能耗和系统性能方面考虑，更好的解决方案是定制设计。

如果使用 COTS 芯片，由于接口的开销，芯片与芯片间的通信会增加系统延迟，同时降低效能。因此，Lester［Jason03］为节点主板开发了一个定制的专用集成电路（Application Specific integrated Circuits，ASIC），称为 Spec。通过定制设计，实现了对主要通信原语效率数量级的提升。

Spec 比大多数商业节点要小许多。它只有 2.5mm 的边长，采用 0.25μm 的 CMOS 工艺，还集成了微处理器、SRAM、通信加速引擎和一个 900MHz 的多信道收发器。

即便其 CPU、无线收发器和存储器被设计在一个芯片上，仍需要部分低成本的外置部件，包括晶振、电池、感应器和天线，这样才能构成完整的无线传感器节点。

图 2-5 给出了 Spec 的体系结构。CPU 核心是一个基本的 8 位 RISC 核心，16 位指令。含有 6 个存储块的（各 512 字节）存储体直接连接到 CPU 核心。把存储器分块是为了实现指令存储

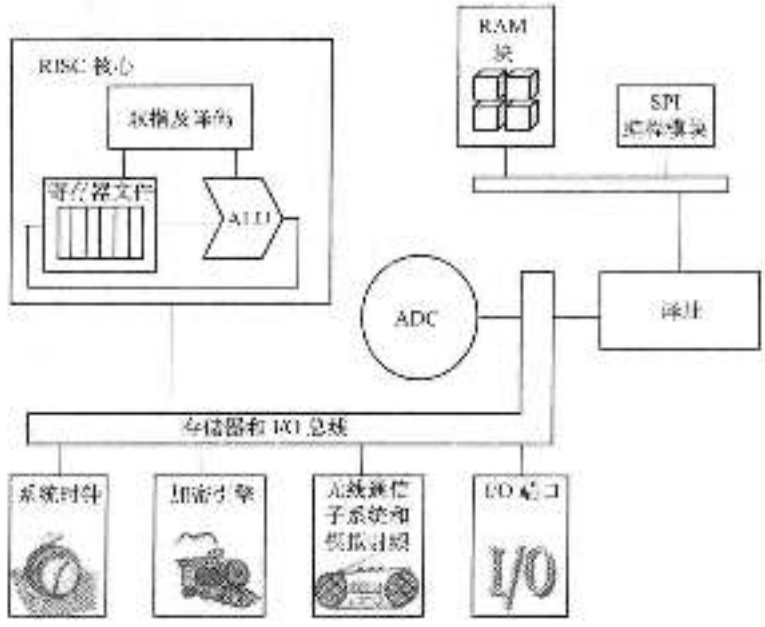


图 2-5 单片无线节点 Spec 体系结构框图

器和数据存储器的平滑集成。除了存储器控制器，CPU 核心还连接了一个超低功耗模数转换器、一个加密加速引擎、通用 I/O 端口、系统时钟、芯片编程模块和无线通信子系统。

无线通信子系统完成以下工作：1）通过精确的发送/接收时间控制，提取或生成数据位；2）通过位组合的四元找出开始标志（即接收器能够分辨出不同数据单元的界限）；3）形成发送数据流；4）与存储器交换数据；5）实现安全通信，能够自动加（解）密以及执行其他工作。

Lester [Jason03] 第一次使用了 VHDL（超高速集成电路硬件描述语言）的数字逻辑工具总结出 Spec 的特点。用 VHDL 模拟后，再用 Ambit Build Gates 将高级的 VHDL 代码映射为 National Semiconductor 公司提供的标准单元。定制设计节点的布局布线设计工作由 Cadence 设计公司开发的 SE（Silicon Ensemble）软件完成。除了 VHDL 仿真外，还将 Spec 下载到 Xilinx FPGA 上验证其功能。

在很多应用中，Spec 的数据处理速度均远远高于 Mica。由于集成设计的方法和硬件加速引擎的使用，Spec 在功耗方面显示出其极大的优势。由于 Spec 是一个完全集成的芯片，这就决定了其接口灵活性不如 Mica。

2.5 COTS 微尘系统

文献 [Seth00] 中介绍了几种值得关注的传感器系统。这些节点配置了 Atmel AT90LS8535（微处理器）和 RFM 916 MHz 无线收发器，还有 7 种模拟传感器（温度、亮度、气压、两向加速度仪、两向磁力计）。电源方面，它采用一个 3V 的锂离子钮扣电池，在连续使用的情况下能工作 5 天，在 1% 占空比下工作时间为 1.5 年。

它搭载了一个低速 CPU，149.475kHz 的 Atmel MCU。它具有 19 条指令，通过无线通信系统来发送与接收原始数据。每个时钟周期执行一条指令，那么它的原始数据速率为 149.475kHz/19 cycles/bit = 7.867kbps。

在无线环境中，噪声和干扰会导致数据包产生错误，因此 Hollar［Seth00］使用循环冗余校验（CRC）检查数据包中的位错误。

图 2-6 为 COTS 微尘系统的单跳通信协议框图。这是一个简单轮通过无线传输-接收配对，实现数据从一个设备到下一个设备传输的过程。它的通信协议应用于两类节点（如图 2-6 所示）：通过串口与计算机通信的基站节点，以及通过无线与基站节点通信的普通节点。普通节点持续发送的数据被基站节点接收，然后在计算机屏幕上显示信息。

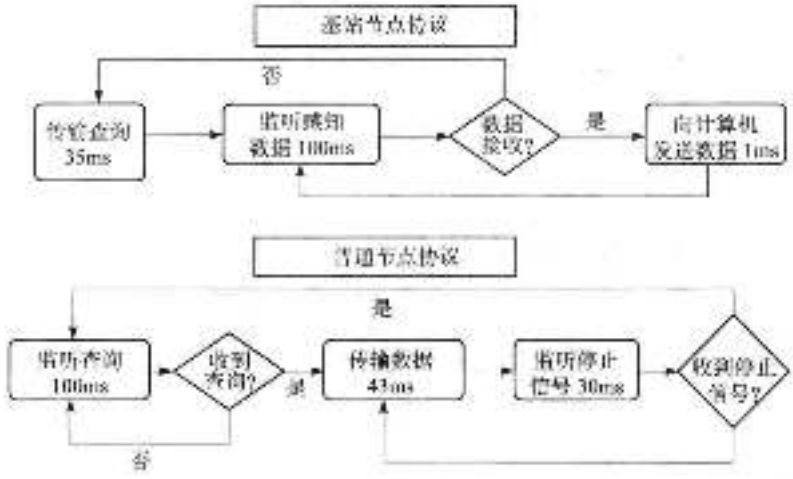


图 2-6　基站节点与普通节点使用的协议（摘自 Hollar, S. E. -A., COTS dust, MS thesis, Mechanical Engineering, University of California at Berkeley, Berkeley, CA, Fall 2000.）

52

Hollar［Seth00］还在两个设备之间运用了一个简单的时间同步协议。为了建立时间同步，基站节点必须首先对一个普通节点进行查询。如图 2-6 所示，在发送查询命令后，它开始监听响应，如果 100ms 后仍未监听到响应，它继续发送传输查询，这样的查询过程会一直重复，直到收到一条有效消息为止。收到消息后，基站节点通过串口把消息发送到计算机，然后基站节点继续监听数据包。

从图 2-6 中还可以看出上述两个协议在传输周期之后有一个监听周期，这使得节点能够迅速响应查询命令。握手协议使得两个节点间的相互通信尽可能的快。


案例研究

Hollar［Seth00］只展示了一个很基本的节点设计，没有考虑无线传感器网络的其他需求。比如，它不能很好地支持多跳通信，CPU/收发器的设计可以实现更高能效。之所以把它作为一个示例，原因是即便是一个简单的节点原型设计，我们也可以从中学到很多东西。

设计建议

文献［Seth00］总结了一些很好的设计经验：

1）CPU 和无线收发器的选择：最初，Hollar 使用 Scenix SX28AC 系列最大时钟周期为 50MHz 的微处理器。但是，第一个电路板制作好后，发现 RF Monolithics 收发器的芯片组与 Scenix 微处理器搭配会出现故障。

Hollar 找出了收发器芯片组出现故障的原因。第一个可能的原因是 CPU 产生的噪声渗透到

了无线收发器中。因为 CPU 的时钟频率锁定在低速的 1MHz 下，但是快速的上升和下降时间导致噪声处于接收频带。第二个可能的原因是电路板没有地线和电源层，这两个部分有助于隔离元器件之间的信号以及保持稳定的供电电压。

2）电源的选择：Scenix CPU 的工作电压为 5V，而 RFM 芯片的工作电压为 3V，因此就需要一种办法能够产生两种电压。一种方法是使用 3 节碱性电池，可以提供 3V 和 4.5V 的电压。但是，这种电池会随着使用而出现电压下降。由于增加了复杂性和元件数量，分别为微处理器和收发器芯片组设置两个调压器的方案不是最理想的。为了解决这个问题，Hollar 将目标设定为使用统一的工作电压，即使用 3V 锂离子电池，设计所有元件在 2.75～3.25V 范围内工作。 53

2.6 Telos 节点

Telos 系列节点（如 Telos-B）是一个被广泛使用的传感器平台。与 Spec 不同，Telos 没有把设计集成在单片上，而是采用了带硬件加速引擎的 COTS 部件组成了一个高功效的系统。表 2-7 总结了不同节点的主要特点。

表 2-7 Telos 之前伯克利系统节点及其属性

节点								
节点类型	WeC	Rene	Rene2	Dot	Mica	Mica2Dot	Mica2	Telos
时间	1998	1999	2000	2000	2001	2002	2002	2004
节点微处理器属性								
类型	ATMEGA8535	ATMEGA8535	ATmega163	ATmega163	ATmega128	ATmega128	ATmega128	TI MSP430
程序存储器（kB）	8	8	16	16	128	128	128	48
RAM（kB）	0.5	0.5	1	1	4	4	4	10
活动功耗（mW）	15	15	15	15	8	8	33	3
睡眠功耗（mW）	45	45	45	45	75	75	75	15
唤醒时间（μs）	1000	1000	36	36	180	180	180	6
节点非易失性存储器属性								
芯片	24LC256	24LC256	24LC256	24LC256	AT45DB041B	AT45DB041B	AT45DB041B	ST M25P80
总线类型	I^2C	I^2C	I^2C	I^2C	SPI	SPI	SPI	SPI
容量（kB）	32	32	32	32	512	512	512	1024
节点通信属性								
芯片	TR1000	TR1000	TR1000	TR1000	TR1000	CC1000	CC1000	CC2420
数据速率（kbps）	10	10	10	10	40	38.4	38.4	250
调制类型	OOK	OOK	OOK	OOK	ASK	FSK	FSK	O-QPSK
接收功耗（mW）	9	9	9	9	12	29	29	38
0 dBm 时发射功耗（mW）	36	36	36	36	36	42	42	35
节点能耗属性								
最低电压（V）	2.7	2.7	2.7	2.7	2.7	2.7	2.7	1.8
整体活动功耗（mW）	24	24	24	24	27	44	89	41
编程与传感器接口属性								
扩展	无	51 针	51 针	无	51 针	19 针	51 针	16 针
通信	IEEE1284 与 RS232	IEEE1284 与 RS232	IEEE1284 与 RS232	IEEE1284 与 RS232	IEEE1284 与 RS232	IEEE1284 与 RS232	IEEE1284 与 RS232	USB
集成传感器	否	否	否	无	否	否	否	是

在对比了 Atmel、Motorola 和 Microchip 的 CPU 性能后，Telos 的开发者选用了 MSP430 CPU。它具有以下优点：

1）无论在睡眠还是活动状态下，它的功耗都是最低的（参见表 2-7）。

2）能够承受最低 1.8V 的工作电压，能够在低电压下工作，就可以尽可能充分地获取电池电能。普通 AA 电池的截止电压为 0.9V。Telos 节点使用 2 支 AA 电池，则系统的截止电压为 1.8V，这与 MSP430 要求的最低电压基本相同。如果使用其他的 CPU，比如 ATmega128 MCU（Mica 系列），它最低能在 2.7V 下工作，这最多能够获取 AA 电池电能的 50%。

3）前面已经介绍过快速唤醒机制有助于节能。从表 2-7 中可以看出，MSP430 有着最快的唤醒时间，从预备状态（1μA）切换到活动状态最多仅需 6μs。

4）从存储器的角度考虑，如表 2-7 所示，MSP430 的片上 RAM 是最大的（10KB），这有助于片上信号的处理。大容量的 RAM 支持更高级的应用。

从无线通信的角度考虑，Telos 具有如下特点：

1）使用了 IEEE802.15.4 标准。这样一个标准化的无线通信使 Telos 可以与其他厂商生产的无线设备进行通信。

2）使用了 Chipcon CC2420 无线收发器。它使用 2.4GHz RF 波段，采用 O-QPSK（偏移四相相移键控）调制方式和直接序列扩频方式（DSSS），数据速率达 250kbps。这样的高数据速率（其他节点通常小于 150kbps）缩短了工作时间，有助于降低能耗。

54
:
56

Telos 节点可以通过板载的 USB 进行编程以及供电。考虑到便携式计算机通常只有 USB 端口，所以 USB 比 RS232 串口接口更适用。

Telos 节点有一个开关按钮，一个重置按钮和一个 16 针脚的 IDC 扩展头。重置按钮被按下后，会产生一个不可屏蔽的重置信号，使节点重新执行任务，也可以用开关按钮代替。开发者还可以在 16 针脚 IDC 扩展头上扩展 I^2C 和 UART，与 Mica 型传感器电路板相连［JPolastre04］。

在很多情况下，需要硬件写保护措施来保护存储器中完整的程序，这样还可以阻止一些先进的节点在使用远程编程时可能出现的写错误。Telos 是第一个为外置存储器提供硬件写保护功能的节点。在插入 USB 接口后，写保护措施自动失效，当使用电池供电时，存储器是被写保护的。

此外，Telos 节点还有一些带独立开关的子电路。如果检测到有故障发生，那么就可以关闭子电路而不是整个系统。这种电路保护的想法来源于大鸭岛［RSzewczyk04］实际的无线传感器网络应用。在大鸭岛的应用中，小部分的电路故障会造成整个节点的故障。如果检测到这些故障，切断部分电路板的电源有助于保护整个系统。

2.7 CargoNet

在文献［Mateusz07］中，研究者设计出了一种称为 CargoNet 的节点。目的是缩小无线传感器网络和射频识别（Radio Frequency Identification，RFID）之间的差距。CargoNet 最初的目标是面向供应链管理与资产安全提供货运箱级的环境监测服务，它采用定制设计电路以降低功耗和成本。

CargoNet 节点采用了全新的设计观念，称之为准被动唤醒（quasi-passive wake-up），实现了异步、多模式的唤醒机制。此机制可以将节点从睡眠模式中唤醒，然后执行超低功耗的操作。CargoNet 用于监测货运箱的内部状况，平均节点功耗低于 25μW。

CargoNet 依靠外部刺激信号唤醒传感器节点。这个想法其实并不新奇，因为其他的相关系统也对外部唤醒机制进行过探索，但是 CargoNet 的功耗更低。

例如，美国西北大学的研究者提出过采用类似唤醒策略进行振动监测与裂纹探测的方案。他们使用压电检波器作为输入传感器，在唤醒后监测非周期性的撞击以确保建筑物结构的安全。虽然它们的模拟前端平均功耗仅为 16.5μW，但是由于处理过程是在 Mica2 节点上执行的，因此平均能耗预算增加了 105μW。

再如，T-Mote [Tmote06] 也带有比较器，用于产生因声音或加速刺激引起的中断。但是由于使用了动态加速度仪和麦克风放大器，因此所需能耗较高（在 mW 范围内）。 57

下面先介绍一下 RFID 的一般概念。

RFID 用于取代在货物运输和分发时所使用的传统条形码技术。传统的条形码扫描要求扫描器与标记物品保持同一直线，因此操作人员必须将贴有标签的物品展平，才能确保成功读取。扫描器与条形码之间的距离要非常近（通常为几厘米）。此外，这些条形码上所承载的信息很少。

然而，RFID 使用读取器读取附着在产品上的标签。读取器与标签之间的距离可以从几英寸甚至到数十英尺（其范围取决于所使用的射频）。而且，读取器可以通过非直线光信号读取标签数据，该信号能够远距离传播或穿透非导材料，这样就可以在无人参与的情况下进行识别。传统的条形码是打印在表面上的，而且不可更改。但是 RFID 标签上的数据是可以更改的，因为它们是可以靠外部刺激改变状态的电子电路。

最近，有研究者提出了“主动式 RFID”这一概念。它实际上是一种带有电池和 CPU 的特殊传感器装置，可以为供应链提供更好的“可视性”。主动式 RFID 可以准确地收集货物在运输过程中所处的环境状况的数据，能够更好地进行风险管理，同时保持灵活性，可以在到达目的地之前检测出可能的货物损坏。

CargoNet 节点是一类“主动式 RFID”。它的准被动唤醒基于以下方面：外部刺激实际上可以唤醒节点，甚至可以为 CPU 提供能量。

CargoNet 可以使传感器对重复的刺激不敏感，这样可以大大降低冗余唤醒的发生，从而节省能量。准被动唤醒允许一个 CargoNet RFID 标签同时且持续对多个传感器的状态进行异常监测，但不需要消耗过多能量。

图 2-7 是一个 CargoNet 主动式 RFID 标签与 RFID 读取器的系统框图。它的核心硬件包括 MSP430 微控制器、一个实时时钟（Real-Time Clock，RTC）和 CC2500 2.4GHz 无线模块等。MSP430F135 基于 Flash 的微控制器是由德州仪器（TI）生产的。进入睡眠状态时，它有一个小于 0.1μA 的待机电流。

CargoNet 标签有一块容量为 16KB 的内置闪存。这是一个很小的存储器，但是它对于大多数的应用来说已经足够，原因如下：

1）它的设计允许存储器仅用于数据记录而不必关注程序存储。它的操作系统（OS）只占用极小的空间。

2）我们通常只需要记录非寻常事件（如极端温度和明显震动）。例程代码只占用小于 8KB 的空间。假设潜在威胁或显著的事件每天只发生一次并且每次需要 10 个字节记录，那么闪存要使用两年才会装满数据。 58

如果开发者需要测试很长的程序，或者在某些情况下需要存储节点中的更多细节信息，CargoNet 允许标签附带一个外置存储器（例如，Atmel 的 AT45DB081B 可以作为附加存储器，它有 8 Mbits 的容量和 2μA 的待机电流）。

MSP430 内部有一个快速启动的高频时钟振荡器。开发者也可以采用外部时钟，如低频表晶。CargoNet 建议采用单独的 Philips PCF8563 RTC 芯片，其计时电流很低（只有 0.35μA）。

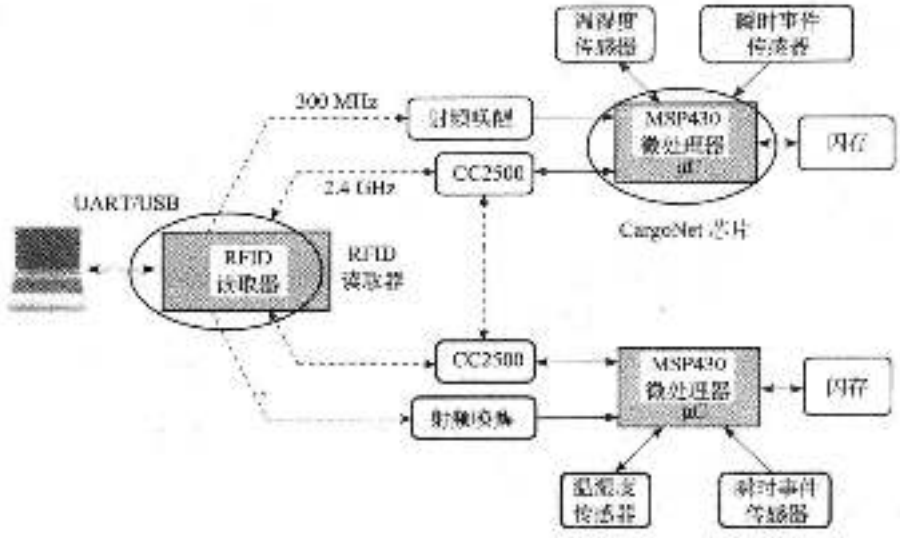


图 2-7 CargoNet 系统框图（摘自 Malinowski, M. et al., CargoNet: A low-cost MicroPower sensor node exploiting quasi-passive wakeup for adaptive asychronous monitoring of exceptional events, SenSys'07, Sydney, Australia, November 6 - 9, 2007.）

RTC 使得主动式 RFID 标签可以通过计算离上一个检查点的时间来定位发生损坏的地点。RTC 还可以发出频率一分钟一次的对温度和湿度传感器的轮询序列。

如图 2-7 所示，该主动式 RFID 标签使用 Chipcon 公司 CC2500 与 RFID 读取/查询器进行无线通信。不同于传统的 RFID 系统，CC2500 无线通信模块是完全双向的，除了将标签数据发送
[59] 给读取器，主动式 RFID 标签还可以接收来自 RFID 读取器的指令。该特点弥补了主动式 RFID 和无线传感器网络之间的空白，这是因为无线传感器网络中需要节点之间的双向通信。这种双向无线通信的功能在应用中是很需要的，比如可以进行时钟同步、记录邻居的身份以及验证传感器读数的有效性。

当基站（即图 2-7 中的 RFID 读取器/查询器）向主动式 RFID 标签发送数据查询请求，以进行重大事件（如高温或震动）的检查、数据转储或调整标签参数时，它会使用无线突发信号。CargoNet 节点在接收到一个 300 MHz 的无线突发信号后，由于该幅度超过了可以动态调整的阈值，节点会被准被动唤醒。

应当注意的是，CargoNet 主动式 RFID 标签一般不会快速地轮询或放大不断变化的环境刺激，这样做的目的是节约能量。相反，它只需要通过“准被动唤醒”技术将环境刺激与阈值进行比较。使用的比较器是基于 Linear Technology 的 LTC1540 芯片的，由于它是非线性的 D 类操作，通常只有 840nW 的静态功耗［Linear04］。

但对于一些变化速度不够快的刺激，比如温度和湿度，它们可能达不到“唤醒阈值”。对于这种情况，主动式 RFID 标签将对刺激进行轮询。

主动式 RFID 标签采用了 12 位精度的序列对 Sensirion SHT11 温度/湿度传感器进行轮询，轮询的时间大约为 55ms。如果一分钟轮询一次，相应的占空比仅为 0.092%，平均功耗为 1.5μW。如此低的占空比轮询根本不会影响标签的能量预算。

如果主动式 RFID 标签需要迅速唤醒以实现快速的刺激响应，例如温度事件，通过 PTC 热敏电阻或其他热能传感器就可以适应对温度的准被动唤醒，它会表现出高阻抗和强烈的特征

响应。

CargoNet 系统还采用了以下两个传感器：射频唤醒接收器和振动测量计。它们带有加强或整合微弱信号的线性放大器。

表 2-8 列出了设备和货物在运输时，CargoNet 传感器采集到的一系列测量值。



奇思妙想

通常人们将 RFID 和节点区分得很清楚。但是 CargoNet 设计了一个设备可以同时作为 RFID 标签和传感器节点。它可以将环境的数据收集到一个"标签"中，然后通过 RFID 读取器远程读取这些数据。在这里使用主动式 RFID 是因为 CargoNet 的 RFID 标签被电池驱动，可以表现出接近智能传感器节点的良好性能。

60

表 2-8 CargoNet 传感器类型

传感器类型	测量或应用
碰撞传感器	潜在的碰撞损坏
振动计量计	平均的低水平振动
倾斜开关	货物的方向与滚动
驻电麦克风	事件导致声音
光传感器	容器缺口或包装箱打开
磁开关	货物移动或包装箱打开
温度传感器	过热或潜在的腐坏
湿度传感器	潜在的霉菌损坏
射频唤醒	接收来自读取器或者其他标签的查询

来源：摘自 Malinowski, M. et al., CargoNet: A low-cost MicroPower sensor node exploiting quasi-passive wakeup for adaptive asychronous monitoring of exceptional events, SenSys'07, Sydney, Australia, November 6 - 9, 2007.

下面将更加详细地介绍 CargoNet 的"准被动唤醒"策略。图 2-8 显示了其基本的唤醒过程。在一个主动式 RFID 标签接收到一个刺激信号后，就把该信号和阈值进行比较，如果刺激强烈到足以受到重视，标签就会唤醒整个系统。

上述准被动唤醒机制的实现需要以下几个条件：

1）有一个始终开启的电路，即模拟前端，功耗应约为 mW 数量级或更少。在 mV 级信号情况下，需要 nW 级的比较器（如 LTC1540）将刺激提升到逻辑层并且唤醒主动式 RFID 标签。

2）在唤醒时间方面，主动式 RFID 标签必须有足够快的唤醒速度才能充分地处理收到的刺激。MSP430 仅需 6μs 的启动时间，这是很理想的。

3）标签必须保持很低的占空比。这样能够减少唤醒的次数以及缩短活动状态的时间。

除了与 RFID 读取器进行高频、快速、远距离的无线数据通信外，CargoNet 主动式 RFID 标签还有一

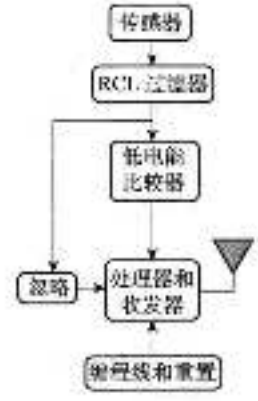


图 2-8 CargoNet 系统中的准被动唤醒机制（摘自 Malinowski, M. et al., CargoNet: A low-cost MicroPower sensor node exploiting quasi-passive wakeup for adaptive asychronous monitoring of exceptional events, SenSys'07, Sydney, Australia, November 6 - 9, 2007.）

个较低频、短距离的信号信道，用于查询和传递位置信息。这使其与其他进行位置信息检测的商用 RFID 标签相兼容。

由于低频无线连接的距离短，传递到标签的射频能量足以将其唤醒，然后高频的射频被打开。如果使用 CC2500，则消耗最多 20mA 电流。

问题与练习

2.1 多项选择题

(1) 一个传感器节点包括(　　)。

A. 模拟/数字传感器芯片

B. 无线收发器

C. CPU/存储器

D. 以上全部选项

61
|
62

(2) 模拟传感器与数字传感器的区别不包括以下哪些方面？(　　)

A. 模拟传感器需要标准的芯片对芯片通信协议。

B. 模拟传感器需要补偿与线性化，而数字传感器不需要。

C. 从 CPU 接口的角度考虑，数字传感器优于模拟传感器。

D. 数字传感器中不需要 ADC。

(3) 在一个传感器网络中，节点中的能量主要被消耗在(　　)。

A. 模拟传感部分

B. CPU 对信号处理的本地计算

C. 无线多跳通信

D. 唤醒/睡眠切换

(4) 关于传感器节点的 CPU，下面哪项叙述是不正确的？(　　)

A. 普通台式或者便携式计算机 CPU 的运算能力远高于传感器节点所采用的 CPU，传感器节点 CPU 通常被称为微处理器或数控制器。

B. 传感器节点 CPU 的工作频率通常小于 100MHz。

C. 当 CPU 处于空闲或睡眠状态时，不消耗能量。

D. CPU 的主要任务是执行通信协议以及在本地处理数据。

(5) 关于传感器节点的存储器，下面哪项叙述是不正确的？(　　)

A. 传感器节点只需要少量的数据存储和程序存储器。

B. 如果数据需要存储较长时间，那么用闪存代 SRAM 更加高效。

C. 程序的执行发生在闪存中，而不是在 SRAM 中。

D. 目前，SRAM 的容量通常小于 1MB。

(6) 传感器节点的无线通信有以下哪些特征？(　　)

A. 低功耗无线通信在接收状态下比发射状态下消耗更多能量。

B. 无线通信系统的发射距离取决于几个关键因素，最直接的因素是传输功耗。

C. 目前市面上大部分射频收发器采用基于 VCO 的无线通信体系结构，能够以各种不同的载波频率通信。

D. 在调幅（AM）方式下编码和解码最为方便，并且不易受噪声影响。

(7) 对于发送方，以下哪些操作是不需要的？(　　)

A. 收到接收方应答后再发送下一个分组。

B. 编码时增加错误检测位。

C. 在 MAC 协议的协助下，等待无冲突发生。

63

D. 将感知数据封装成不同分组。

(8) 将无线通信与 CPU 速率解耦的原因是：(　　)

A. 当微处理器的速率与数据传输速率耦合时，两个芯片都不会在最佳工作点运行。

B. 无线通信芯片在其传输速率达到最大值时效率最高，如果与 CPU 的处理速率耦合，那么就不能达到最高效率。

C. 无线通信与 CPU 是完全不同的芯片，在大部分情况下都需要解耦。

D. A 和 B。

(9) Spec 优于 Mica 的原因是：(　　)

A. Mica 节点被已有的片间接口限制了性能，开发定制的 ASIC 能够去除商用模块所带来的人工限制。

B. 通过使用定制的硅片，可以实现对关键通信原语的数量级效率的提升。

C. A 和 B。

D. 与 Mica 相比，Spec 的传输距离更长。

(10) 关于 Telos 节点的叙述，以下哪句是不正确的？(　　)

A. Telos 采用蓝牙通信标准（即 IEEE 802.15），更适合于短距离无线通信。

B. Telos 采用 MSP430 微处理器，在睡眠和活动状态下功耗最低。

C. Telos 没有采用硅片集成设计，而是采用带硬件加速引擎的 COTS 模块构建了一个无性能损失的高能效系统。

D. Telos 通过板载 USB 进行程序下载（使用引导加载器或者 JTAG）以及供电。

2.2　上网了解太阳能电池的特点和设计规则。

2.3　“传感器”（sensors）和“传感器节点”（sensor motes）之间有什么区别？

2.4　阅读文献「Mateusz07」，详细了解 RFID 是如何集成到 CargoNet 传感器节点的。

2.5　与其他节点（比如 Mica）相比，Telos 节点有什么优点？

第三部分

Wireless Sensor Networks: Principles and Practice

网络协议栈

第3章

Wireless Sensor Networks: Principles and Practice

无线传感器网络中的介质访问控制技术

3.1 引言

无线传感器网络是各种不同节点的集合，这些节点用于感知环境参数，如震动、温度、压力、声音以及污染物等。在无线传感器网络中，每个节点都是一个能自主工作的设备，它包含了通信模块、计算模块、感应模块和存储器。为了高效地在各传感器节点之间交换数据，无线传感器网络采用介质访问控制（Medium Access Control，MAC）协议协调信号在共享信道上的传输。否则，多个节点可能会同时访问传输介质（即无线信道），这会导致信号冲突、数据丢失、重传、能量浪费和传输延迟等。



MAC协议通过确定通信调度以及规则来决定多个节点共享访问物理介质的方式，这些规则如：1）哪个节点应该占用信道；2）什么时候节点能够占用信道，能占用多长时间；3）节点如何通过信道与相邻节点联系等。

3.1.1 无线传感器网络中的介质访问控制

MAC在多数网络范型中都扮演着重要角色，包括有线网络、移动自组网（MANET）以及无线传感器网络。有效MAC协议的设计必须考虑网络范型各自的特点。比如，与以太网（Ethernet）等有线网络不同，无线传感器网络中所使用的无线信道由于冲突、信号损失、噪声甚至链路断开而面临更多的数据丢失的问题，而无线连接中的信号冲突问题不能用有线网络中的方式检测。此外，无线传感器网络拥有十分有限的资源，如能量、带宽和计算能力，这些都限制了可用于其他无线网络（如Wi-Fi和MANET）的MAC协议在传感器网络中的应用性。



通常，其他网络范型下的MAC协议不能直接应用于无线传感器网络中，这是由于其独特的无线介质、微型节点和各种无线传感器网络应用造成的。

3.1.2 无线传感器网络中MAC设计的挑战性

作为一种特殊的无线网络，无线传感器网络与其他无线网络面临相似的挑战。下面将详细介绍这些传感器网络所面临的挑战以及资源限制为何对MAC的实现有巨大影响：

1）资源限制。

2）无线信道中的信号损耗。

3）接收端的信号冲突。

4）隐藏和暴露终端问题。

1. 资源限制

无线传感器节点拥有的资源是有限的，比如能量、带宽、计算能力和存储空间等。这些都应该是设计 MAC 时需要考虑的因素。在使用电池供电的传感器节点中，能量是最值得关注的问题。一旦电池耗尽，给电池充电或者更换电池是困难也是不切实际的。这就是为什么设计传感器网络中的 MAC 协议时，首要目标是使节点或网络的寿命最大化，其他的效能指标是次要的。比如，在不工作的时间段内关闭该模块，从而节省能量。


奇思妙想

由于传感器节点的通信比计算要耗能，所以在实现必需的网络操作时，尽量减少通信是一种设计高能效传感器网络 MAC 协议时常用的有效方法。

同其他有线网络（如光纤网络）相比，无线传感器网络的带宽是相当低的。带宽限制和传感器网络拓扑结构的动态性也是设计 MAC 协议时面临的挑战。具体地说，在无线传感器网络中，数据的发送以及存储均以分布式方式进行，并且每个传感器节点都是独立于其他节点的自行工作的设备，传感器节点需要相互通信才能自组织成为可以进行数据传输的网络系统，同时也要避免冗余。此外，由于微型节点很脆弱，可能会出现故障，而节点发生故障会使网络的拓扑结构发生变化。同样地，能量耗尽以及节点的移动也会导致拓扑结构的变化。

2. 无线信道上的信号损失

无线传感器网络使用无线信道作为传输介质，而衰减、反射、折射、散射等会使信号失真或者损失。信号衰减是指从源节点到目标节点的信号在空气中传播时能量的损失。当源节点与目标节点之间存在障碍物时传输信号会发生反射。障碍物的边会把原始的传输信号分成多个信号，而粗糙的障碍物表面引起的多个信号反射最终会造成信号的散射。文献［Rappaport96］中介绍了一种常用的使用全向天线的无线传播方式，其中还指出节点 j 接收到节点 i 发送的信号能量满足以下公式：

$$P_j = \beta \frac{P_i}{d_{ij}^{\alpha}} \tag{3.1}$$

在式（3.1）中，P_j、P_i、d_{ij}分别表示节点 j 接收到的能量、节点 i 发出的能量以及节点 i 和节点 j 间的距离，而 α 和 β 表示能量损失常数，一般由无线传输的环境决定。

该公式表明，无线信号在空气中传输的距离越远，能量损失越大。实际上，当且仅当节点 i 达到特定的发射功率级后，节点 i 才能与节点 j 建立无线连接；否则，由于信号损失，接收节点 j 不能正确地把节点 i 发送的信号解码成为传输的数据信息，或者根本无法收到信号。换句话说，每个节点都有一个有限的传输范围。对于电池供电的无线传感器网络而言，节点的传输范围是动态变化的，并且节点间的无线连接易受节点的故障和位置变化的影响，这就需要无线传感器网络拥有不同的连接访问控制模式。

3. 接收端的信号冲突

当两个或多个节点通过同一信道向其他节点同时发送数据时，多个信号可能会在同一接收端发生冲突，这会妨碍接收端正常获取有意义的数据信息。为了确保可靠数据传输，MAC 协议必须采取办法从冲突中恢复正常。冲突会导致能量的浪费、带宽利用率低以及大量数据的传输延迟。在有线网络（比如以太网）中，发送方通过比较发出的信号和收到的信号就能容易地检测出是否有其他节点也在发送数据，如果发生的话就能很快地从冲突中恢复。然而，在无

线传感器网络中，由于无线信号衰减或者传播途径中有障碍，可能会导致两个节点相互之间不能有效地发送和接收数据。比如，有两个发送者同时向一个接收者发送数据，如果两个发送者不在对方的传输范围之内或者它们之间有障碍物使它们不能听到对方存在，那么两个发送方的信号就会在接收方发生冲突，任何一个发送方自身是无法检测到这一情况的。因此有线网络中的方法不适用于无线传感器网络。

4. 隐藏和暴露终端问题［AWoo01］

如图 3-1 所示，每个节点周边的圆圈表示其在使用全向天线时的传输范围，同时还假设每个节点的传输范围都是同样大的。如果两个传感器节点均处于对方的传输范围内，那么称这两个节点在重叠范围内（或者相同的冲突域）。例如，节点 1 与节点 2 在重叠范围内，同样的，节点 2 与节点 3、节点 3 与节点 4 也都处于重叠范围内。显然，如果一个节点能够与相邻节点
70 进行通信，那么它们必然处于重叠范围内。为了尽可能减少冲突，MAC 协议的设计中广泛运用了**载波侦听**（carrier sense）技术。所谓载波侦听，就是指发送者在准备发送数据前先监听信道上的载波，检测是否有正在进行的数据传输。如果检测到有正在进行的数据传输，那么该节点必须等正在进行的传输完成后才开始自己的传输。

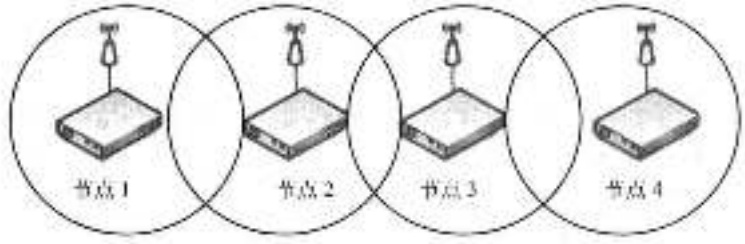


图 3-1　隐藏和暴露终端

载波侦听模式在以太网的 MAC 协议中发挥了很大作用。然而，由于无线信号的传播范围有限，位于发送节点处的无线信号与位于接收节点处的无线信号一般都会存在非常大的差异（我们称这种差异为节点间的不可见性），节点之间的不可见性使得载波侦听在无线环境下失去作用。假设图 3-1 中所有节点之间此时没有进行通信。某时刻节点 1 和节点 3 检测到有事件发生需要向节点 2 发送信息。在发送信息前，节点 1 和节点 3 都对信道进行监听后发现信道是空闲的，因此它们同时向节点 2 发送数据，这就导致在节点 2 处发生冲突。即使是节点 3 正向节点 2 传输数据，由于节点 1 和节点 3 不在重叠区域内，所以节点 1 不能检测到节点 3 发出的信号，因此节点 1 会认为该信道未被占用，也不知道节点 2 已经同其他节点建立了传输，那么节点 1 发出的信号就会干扰节点 3 到节点 2 的数据传输。这是因为节点 3 与节点 1 相互不可见，但它们都可以与节点 2 通信，这就是无线网络中的**隐藏终端问题**（hidden terminal problem）。



隐藏终端问题表明，在无线网络中，载波侦听也不能避免冲突的发生。此外，载波侦听还会使无线网络中的信道不能充分使用。

71 现在假设节点 2 正向节点 1 传输数据，节点 3 也打算向节点 4 发送数据。节点 3 先进行载波侦听，发现传输信道被占用，必须等待从节点 2 到节点 1 的传输完成。而只有在接收端的信号干扰或冲突才会导致数据丢失以及能量和带宽的浪费。所以事实上，节点 3 和节点 2

可以同时分别向节点4以及节点1发送数据，这是因为两组传输（即节点2到节点1以及节点3到节点4）没有发生在同一接收端。虽然节点4和节点1都应该能各自接收数据，但是节点3向节点4发送数据却被阻止。这就是无线网络中的**暴露终端问题**（exposed terminal problem）。

为了解决上述无线传感器网络中存在的问题，目前已有很多关于MAC协议设计的研究（如「Tobagi75，Karn90，Bharghavan93」）。几个较早的关于载波侦听和隐藏终端问题的MAC协议的研究成果都是从IEEE802.11标准「IEEE07」改进而来的，IEEE802.11标准还是很多针对无线传感器网络的MAC协议设计的基础。因此，在本章接下来的内容中，将首先简要介绍IEEE802.11标准，然后介绍MAC协议的分类，最后介绍各类中典型的无线传感器网络MAC协议，包括S-MAC「Wye02」、T-MAC「Tvdam03」、TRAMA「Vrajendran06」、Sift「Kjamieson03」、Z-MAC「Irhee08」以及B-MAC「Jpolastre04」。

3.2 IEEE802.11标准概述

图3-2为IEEE802.11标准的分层体系结构框图。IEEE802.11标准的物理层包括直接序列扩频（DSSS）、跳频扩频（FHSS）、红外、802.11a、802.11b以及802.11g。DSSS规定的物理介质是频率为2.4GHz或者5GHz的ISM波段，数据速率为1～54Mbps。FHSS的物理介质和数据速率与DSSS相同，只是信道数不同而已。信道数在每个国家由该国网络管理部门规定，比如DSSS的信道数在欧洲国家规定为70，而在日本为1；在美国规定FHSS为70，而在日本为23。红外的数据速率同DSSS、FHSS相同，只是其使用的是850－950nm波长的光波。



图3-2 IEEE 802.11协议体系结构（摘自 Stallings，W.，IT Prof.，6，32，September-October，2004.）

数据链路层（the data link layer）包括**逻辑链路控制**（logical link control）和MAC，它规定了两种访问控制方式：分布式协调功能（DCF）和点协调功能（PCF）。

3.2.1 点协调功能

IEEE802.11的MAC层规定了点协调访问机制（PCF）以提供非竞争的服务，这种机制只能用于基础结构模式，如图3-3所示。在基础结构模式中，节点使用集中式MAC算法通过接入点（AP）与网络连接。这种模式可以方便地支持高流量优先级。

3.2.2 分布式协调功能

分布式协调功能（DCF）用于在ad hoc模式下实现多个节点共享介质，如图3-4所示。DCF通过采用带冲突避免的载波侦听多路访问（Carrier Sense Multiple Access/Collision Avoidance，CSMA/CA）协议和IEEE802.11 RTS/CTS机制让多个节点共享介质「Karn90，Bharghavan93」。在CSMA/CA中，站（station）在发送数据前必须先监听信道一段时间，检测信道是

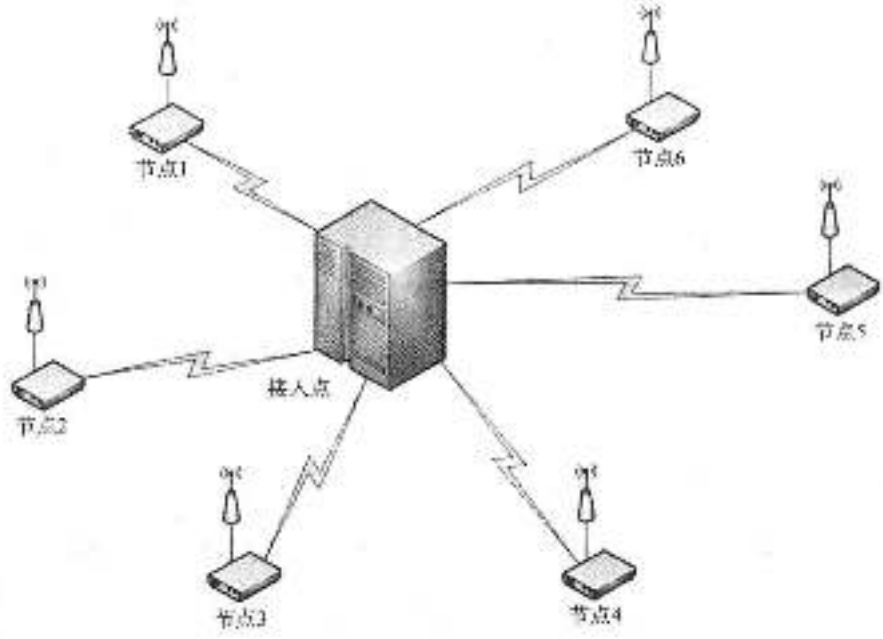


图 3-3　IEEE 802.11 基础结构模式（摘自 Stallings，W.，IT Prof.，6，32，September-October，2004.）

否空闲。如果检测到信道正忙，那么将进入退避过程，将发送推迟一段时间以避免冲突发生。在本章前面已经介绍了由于无线信道的特点以及隐藏终端问题，冲突检测是不可行的。于是在 IEEE802.11 中通过请求帧（Request-To-Send，RTS）和清除帧（Clear-To-Send，CTS）的交换，通知发送者和接收者传输范围内的节点在数据传输期间保持静默。

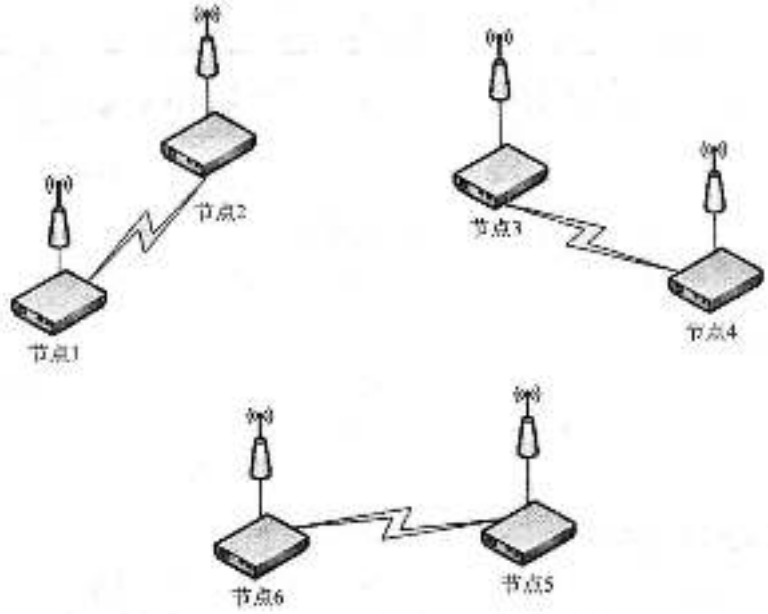


图 3-4　IEEE 801.11 ad hoc 模式（摘自 Stallings，W.，IT Prof.，6，32，September-October，2004.）

72
-
74

如图 3-5 所示，如果发送者要向接收者发送数据，那么先向接收者发送一个 RTS 帧，接收

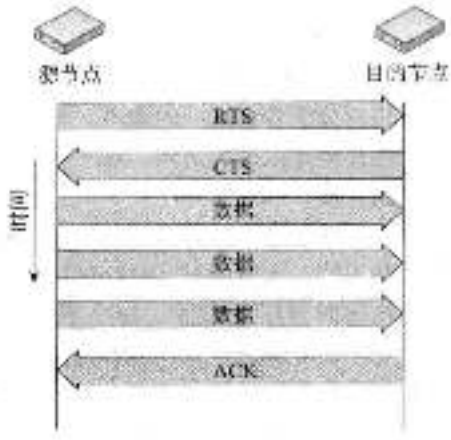


图 3-5 RTS/CTS 机制下的数据传输

者收到后会回复一个 CTS 帧。其他节点监听到 CTS 和 RTS 帧后就会在一段时间内暂停发送数据，这样就避免了冲突，也就解决了隐藏终端问题。发送数据前需要等待的时长包含在 RTS 帧和 CTS 帧中。图 3-6 和图 3-7 分别为 RTS 帧和 CTS 帧包含的主要字段。RTS 帧的结构由 5 个字段组成：

1）帧控制字段（2 字节）：该字段包含了关于使用的协议版本、电源管理、是否有数据碎片以及该帧是否受保护等信息。

2）持续时间字段（2 字节）：传输数据或管理信息、CTS 帧以及 ACK 帧所需的时间。

3）接收地址（RA）字段（6 字节）：数据传输到的目标节点的地址。

4）发送地址（TA）字段（6 字节）：发起数据传输的源节点的地址。

5）帧校验序列（FCS）字段（4 字节）：用于检验数据传输中的错误。它是 32 位长的循环冗余码（CRC），对所有字段（包括头部）进行校验，采用 32 次多项式。

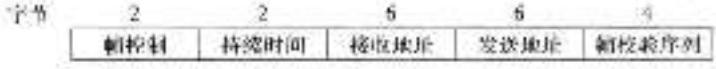


图 3-6 RTS 帧格式（摘自 IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and Metropolitan area networks—Specific requirements, Part 11: Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications, pp. 120-121, July 2007.）

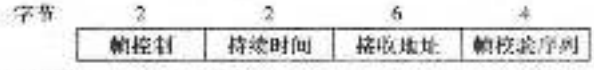


图 3-7 CTS 帧格式（摘自 IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and Metropolitan area networks—Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, pp. 120-121, July 2007.）

CTS 帧的结构由 4 个字段组成：

1）帧控制字段（2 字节）：与 RTS 中的帧控制字段相同。

2）持续时间字段（2 字节）：其值等于 RTS 帧中持续时间字段值减去 CTS 帧的传输时间。

3）接收地址（RA）字段（6 字节）：该 CTS 帧将要发送到的节点的地址。一般与收到的 RTS 帧中的 TA 字段相同。如果 CTS 帧为接收者发送的第一个数据包，那么该 CTS 帧的 RA 字段需要设置成即将发送 CTS 帧的接收者的 MAC 地址。

4）帧校验序列（FCS）字段（4 字节）：与 RTS 中的 FCS 相同。

在收到 CTS 帧后，发送者就可以立即启动向接收者进行数据传输。如果接收者成功接收到数据，那么它将向发送者发送一个 ACK 帧，如图 3-5 所示。

3.3 MAC 协议的分类

信道访问机制通常被分为 4 类：时分多址（TDMA）、频分多址（FDMA）、码分多址（CDMA）和空分多址（SDMA）[Kcoliver05]。在 TDMA 中，所有的节点使用相同频率的信道，同时为每个节点分配指定的时间片，以进行数据传输。节点逐一用自己的时间片发送数据。在

TDMA 中，为了成功进行通信，必须保持访问共享介质的节点之间的时间同步。FDMA 使用的技术与 TDMA 基本相似，与 TDMA 的为每个节点分配时间的区别在于 FDMA 对每个节点分配了不同的频率。CDMA 采用扩频技术和特殊的编码机制，让多个用户共享同一物理信道，但是为每个用户分配各自唯一的编码。而 SDMA 则是对节点进行空间上的分割，通过空间的复用或者多样性提供多信道的访问。通常来说，不同的网络技术通过不同的方式或者使用多种技术的组合访问介质。

在无线网络中，介质访问控制方式可以是分布式或者集中式的［Achandra00］。根据操作模式的不同，无线 MAC 协议可以大致分为随机访问协议、确定性访问协议及混合型访问协议。在随机访问 MAC 协议中，每个节点以随机的方式尝试访问传输介质；在确定性访问 MAC 协议中，每个节点通过主从式或者轮流使用令牌的方式访问传输介质；而混合型访问协议则是将随机性访问和确定性访问结合在一起，对传输介质进行访问。类似地，为了解决隐藏终端问题、资源受限和应用需求等挑战，研究者对适用于无线传感器网络的 MAC 协议做了大量研究，包括在现有 MAC 协议的基础上改进以及提出新的协议。根据信道访问模式的不同，可以将无线传感器网络中 MAC 协议分为 3 大类：

1）基于竞争的 MAC 协议。

77　2）基于固定分配的 MAC 协议。

3）混合与事件驱动的 MAC 协议。

3.3.1　基于竞争的 MAC 协议

基于竞争的 MAC 协议允许多个节点同时访问信道，这可能会产生冲突，但可以用多种办法解决，比如随机退避、RTS/CTS 协议以及冲突避免技术。随机退避中比较典型的例子是载波侦听多路访问（CSMA），即节点在发送数据前先对介质进行侦听，检测介质上是否有正在进行的通信。如果节点发现介质正忙，那它将退避一段时间后再重试发送；如果发现信道是空闲的，那么在发送数据前该节点将等待一段随机的时间（即竞争周期）。竞争周期的应用降低了两个节点同时开始发送数据的可能性，因此减少了冲突。面向无线传感器网络的基于竞争的 MAC 协议会采用 IEEE 802.11 标准 DCF 中的 RTS/CTS 握交换技术和超时技术，并结合无线传感器网络应用的特征，以实现无线传感器网络系统在能耗、寿命、延迟以及吞吐量等方面的性能优化。目前，典型的面向无线传感器网络的基于竞争的 MAC 协议有 S-MAC［Wye02］、T-MAC［Tvdam03］、WiseMAC［Aelhoiydi04］、DMAC［Glu04］、DSMAC［Plin04］以及 AC-MAC［Fli06，Jai04］等。下面将介绍 S-MAC 和 T-MAC 协议的基本设计。

1. S-MAC［Wye02］



显然，在无线传感器网络中，存在一些浪费能量的过程，包括空闲监听、冲突、串音以及开销控制。

空闲监听（idle listening）是指一个传感器节点等待其他节点可能向其传输数据的状态。在大多数传感器网络应用中，如果节点没有侦听到信号，那么大多数时间都处于空闲状态。而传统的 MAC 协议，比如 IEEE802.11 和 CDMA，都要求节点监听信道，等待可能的数据传输。研究表明，空闲监听消耗的能量占用于接收所消耗能量的 50% ~ 100%［Stemm97］。在很多无线传感器网络应用中，节点处于空闲监听状态的时间远比处于通信状态的时间要长，实际上这

会消耗节点的大部分能量。数据冲突会导致传输包中数据的损坏，那就必须丢弃，并且之后数据的重传也增加了能耗以及网络延迟。类似地，节点间的串音以及控制开销也会导致无线传感器网络节点的能量浪费。

为了减少上述的能量浪费，S-MAC 协议采用了周期性的睡眠和监听，然而，节点的周期性睡眠会导致数据包在逐跳转发过程中产生等待延迟。S-MAC 协议假设所有节点均用于同一应用 78
或者一组应用中，由于节点有相同的应用目的，那么可能就会出现一个节点拥有的信息比其他节点多的情况。S-MAC 允许拥有较多信息的节点更长时间地使用信道，这就能够维持更为重要的应用级的公平性而不是每一跳的公平性。

(1) 周期性侦听/睡眠

在很多无线传感器网络应用中，节点经常处于空闲状态，为了减少空闲监听时的能量浪费，S-MAC 采用了一组睡眠/唤醒状态。在睡眠状态下，节点将通信模块（消耗大部分能量）关闭而保持其他模块的开启状态。按照调度机制，在一段时间间隔后，节点从睡眠状态切换到唤醒状态，时间间隔的值取决于节点上的网络应用。在唤醒状态下，节点将通信模块打开，参与必要的网络通信。

为了适时地调度睡眠/唤醒状态以及与相邻节点的通信，节点应与相邻节点进行同步同步。为了避免时间同步可能产生的错误，S-MAC 采用了两种技术。第一，用于同步的时间既是相对的而非绝对的；第二，侦听时长大于时钟漂移时长。在 S-MAC 协议中，节点可以自由地选择侦听/睡眠的调度机制，但是最好与相邻节点保持同步以减少控制开销，这是因为当两个节点都处于唤醒状态时才能进行通信。也就是说，最理想的情况是相邻的节点都能同时转入睡眠或者唤醒状态。然而，在图 3-8 所示的多跳网络中，并非所有的相邻节点能够同步到相同的睡眠/唤醒调度机制下。例如，节点 A 和 B 采用相同的调度机制，同样节点 C 和 D 也采用相同的调度机制，但是可能和节点 A、B 采用的调度机制不同。一个节点通过向直接相邻的节点广播其调度机制达到交换调度机制的目的，以确保所有的相邻节点可以通信，即使它们采用不同的调度机制。

在 S-MAC 协议中，如果一个节点想与一个相邻节点通信，那么它必须等到该节点开始侦听（或处于唤醒状态）。如果超过一个节点要与同一个节点通信，那么在该节点处于唤醒状态时，这些节点必须竞争访问介质。对于这种竞争来说，需要采用 RTS/CTS 帧交换，第一个发出 RTS 帧并且收到接收者回复 CTS 帧的节点获得访问信道的权利。收到 CTS 帧后，节点就可以进行数据传输，之后继续按照调度机制进入睡眠或侦听模式。



图 3-8 4 节点网络的示例（摘自 Ye, W. et al., An energy-efficient MAC protocol for wireless sensor networks, Proceedings of IEEE INFOCOM, New York, June 2002, Vol. 3, 1567－1576.） 79

(2) 选择和维护调度机制

每个节点都应该在周期性地侦听/睡眠之前选择一个调度机制，并且与相邻节点交换调度机制。调度机制被保存在一个包含所有相邻节点调度信息的表中。调度机制的选择和插入按如下规则进行：

1）每个节点先侦听传输信道一段确定的时间后，如果节点没有收到其他节点的调度信息，那么该节点随机选择自己的侦听/睡眠调度机制并且将调度信息通过 SYNC 广播出去，声明自己将在 s 秒后进入睡眠状态。由于不需要遵循相邻节点的调度机制，该节点独立地选择自己的调度机制，它被称为**同步源**（synchronizer）。

2）在侦听期间，如果一个节点在选择自己的调度机制前接收到相邻节点的 SYNC 分组，

这个节点将采用收到的SYNC分组中的调度机制，这样的节点被称为**追随者**（follower）。假设追随者知道SYNC分组的发送者在t秒后将切换到睡眠状态，为了避免与其他追随者可能产生的冲突，在等待一段随机的时间t_r秒后，它将广播自己的调度机制并且声明它将在$t-t_r$秒后进入睡眠状态。

3）如果一个节点已经选择了一个调度机制，之后又从其邻居节点接收到了一个不同的调度机制时，该节点应同时满足自己已有的调度机制和接收的调度机制中的工作需求，在两种调度机制中的侦听期均保持唤醒状态。

（3）保持同步

无线传感器网络节点间的同步是通过发送SYNC分组来保持的。SYNC分组包括源节点的地址以及下一次睡眠的时间。下一次睡眠的时间并不是绝对的，而是与SYNC分组的传输时间相对的，约等于收到该分组的时间。这有利于消除时钟同步错误。目标节点在收到SYNC分组后立即启动计时，在计时停止以后，该节点切换到睡眠状态。为了发送数据包以及SYNC分组，唤醒状态的时间被分成两部分。第一部分用于接收SYNC分组，第二部分用于接收RTS帧。每一部分还被进一步划分成时间片用于SYNC或数据包传输前的载波侦听。

80 | 每个节点周期性广播包含其调度机制的SYNC分组，这样能让新加入的节点采用相同的调度机制。对新加入的节点来说，调度机制的选择过程与前文描述的是一样的。在将自己指定为同步源之前，新加入的节点会把初始的侦听时间设置得足够长，以增加接收到相邻节点调度机制的可能性。

（4）冲突与串音避免

为了避免多个节点同时向一个节点发送数据，S-MAC采用了RTS/CTS帧交换以及虚拟和物理载波侦听机制。研究表明，这是一个解决隐藏终端问题有效方法［Pkarn90，Bharghavan93，IEEE07］。所有节点在进行数据传输前都应该进行载波侦听，如果源节点侦听到信道正忙，它就转入睡眠状态。当目标节点进入唤醒状态时，源节点也进入唤醒状态。S-MAC发送广播数据包时，比如SYNC分组，不需要RTS/CTS帧交换而是直接进行。对于**单播**（unicast）数据包，在传输过程中源节点和目标节点都需要遵循RTS/CTS/数据/ACK的顺序。此外，每个数据包还包括一个用于标识需要的剩余传输时间的字段，这与IEEE802.11中的网络分配矢量（NAV）的概念相似。这样，一个节点在接收到发送到其他节点的包后就知道需要保持静默或回到睡眠状态的时长。

S-MAC协议减少了由于串音而造成的能量浪费。采用S-MAC协议的节点只要侦听到CTS或CTS帧，就会转入睡眠状态，这是由于随后的数据和ACK的传输通常将花费较长时间。例如，在图3-9中，节点C正向节点D发送数据，很明显节点C与D不可能处于睡眠状态。由于冲突发生在接收端，为了避免冲突，节点E不能发送数据并且还应该处于睡眠状态。节点B理论上可以在节点A处于唤醒状态时向其发送数据，这是因为节点D不在节点B的传输范围内。但是，节点B的数据发送可能会造成节点C在准备接收ACK时发生冲突。类似地，节点E在C与D传输数据时也不能进行数据传输。因此，发送者和接收者的所有直接相邻节点在侦听到RTS或者CTS帧后都应转入睡眠模式。也就是说，根据RTS/CTS帧中的NAV信息，节点需要转入睡眠状态以避免串音，直到当前传输结束为止。



图3-9　串音避免示例（摘自Ye，W. et al.，An energy-efficient MAC protocol for wireless sensor network，Proceedings of IEEE INFOCOM，New York，June 2002，Vol.3，1567－1576.）

81

(5) 消息传递

消息是一组有意义的数据，可以是一个比较大的数据包或一系列数据包。一方面，把一个长的消息装入一个数据包后，如果消息损坏就需要重传，这在能耗、延迟和带宽的利用率方面的代价是很昂贵的。另一方面，如果通过多个较短而且独立的数据包传递一条消息，就会大量增加控制开销，比如 RTS/CTS 帧的交换。因此，S-MAC 协议将一条长消息分割为多个碎片，并且以突发的形式发送。在整个突发中只用一次 RTS/CTS 帧交换，以便在所有碎片传输过程中保留介质。只有当发送者接收到来自接收者的 ACK 后才认为一个碎片已经发送成功；如果发送者没有收到 ACK 帧，那么它将保留介质的时间延长一个数据碎片发送的时间，并且立即重传当前碎片。接收者在接收到每个数据碎片后会发送 ACK 帧，这样可以解决隐藏终端问题。当前传输的 NAV 信息也包含在 ACK 以及数据包中。这样，传输路径周围的节点就能知道此时正进行的数据传输的持续时间，即使数据包损坏或者节点在传输时才切换到唤醒状态。

(6) 节能与增加延迟

在分析 S-MAC 中延迟带来的损失前，我们先了解一下多跳网络中基于竞争的 MAC 协议固有的一些延迟。这些延迟包括载波侦听延迟、退避延迟、传输延迟、传播延迟、处理延迟以及排队延迟。但是，在 S-MAC 中还存在另一种延迟，称为**睡眠延迟**（sleep delay），即源节点发现其目标节点处于睡眠状态而需要等待，一直到目标节点转为唤醒状态。假设规定 1 帧是一个完整的侦听和睡眠的周期，式 3.2 给出了在数据包以相同的概率在同一帧中到达的平均睡眠延迟：

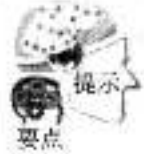
$$D_s = \frac{T_{frame}}{2} \tag{3.2}$$

其中，D_s 表示睡眠延迟，T_{frame} 表示帧长，并且等于 T_{sleep}（表示睡眠时间）与 T_{listen}（表示侦听时间）之和，如式 3.3 所示：

$$T_{frame} = T_{sleep} + T_{listen} \tag{3.3}$$

式 3.4 给出了 S-MAC 中相对能量节省，最后一项表示节点的占空比。由该式可看出，侦听的时间越短，平均睡眠延迟越大。

$$E_s = \frac{T_{sleep}}{T_{frame}} = \frac{T_{frame} - T_{listen}}{T_{frame}} = 1 - \frac{T_{listen}}{T_{frame}} \tag{3.4}$$



S-MAC 协议降低了节点的能耗，因此延长了网络寿命。但 S-MAC 是通过延迟来降低能耗的，那么把 S-MAC 应用在对时序要求严格的无线传感器网络中将不是最优的选择。

(7) S-MAC 协议性能

文献［Wye02］中的实验表明，与 IEEE802.11 中 DCF 相比，S-MAC 有很好的节能性质。如果源节点在以每 1～10 秒发送一条消息的传输负载下，那么一个类 IEEE 802.11 的 MAC 协议的能耗是 S-MAC 的 2～6 倍。为了减少 S-MAC 的延迟，文献［Wye04］介绍了一种名为自适应侦听（adaptive listen）的技术。其基本思想是把节点从低占空比的状态切换到更加活跃的状态。具体地说，自适应侦听使节点在完成当前的传输后继续保持唤醒状态一段时间，用于监听相邻节点的传输（理想状况下只有 RTS 或 CTS），而不是等待固定分配的侦听时间。如果它是下一跳的节点，那么就可以立即接收来自相邻节点的数据，否则该节点转入睡眠状态，直到下一次分配好的侦听时间。

2. T-MAC [Tvdam03]

为了更好地解决无线传感器网络中空闲监听的问题，提出了另一种基于竞争的 MAC 协议——T-MAC（timeout MAC）。它在不需要无线通信模块时关闭，从而降低能耗 [Tvdam03]。T-MAC 协议的基本思想是在相同的时刻将网络中所有节点的无线通信模块打开，并且在通信
83 结束一段时间后再将其关闭。T-MAC 协议是从 S-MAC 协议改进而来的，和 S-MAC 协议中根据预定的调度时间打开无线通信模块不一样，T-MAC 可以动态地调整侦听/睡眠工作周期，这样的结果就是 T-MAC 协议在不同的消息速率下比 S-MAC 更加节能。

（1）协议设计

与 S-MAC 协议类似，T-MAC 协议同样采用周期性的睡眠与唤醒机制来降低无线传感器网络的能耗。在睡眠状态下，节点的感应模块是打开的并且可以将采集的数据放入队列中。处于睡眠状态的节点也可以接收来自相邻节点的消息，并将其放入队列中。在活动或者唤醒状态，节点按需侦听或者传输数据。在数据传输过程中，为了保证避免冲突以及可靠传输，T-MAC 也采用和 S-MAC 一样的 RTS/CTS/数据/ACK 模式。如果在一个给定时间 TA（Time Active）内没有以下任何一个**激活事件**（activation event）发生，节点则转入睡眠状态。激活事件包括：

1）周期性定时器到时。

2）在无线信道上收到数据。

3）感知到无线通信的存在。

4）节点本身数据包或者 ACK 帧发送结束。

5）通过侦听 RTS/CTS 帧，确认邻居的数据交换已经结束。

每帧空闲监听最小的长度由 TA 值决定。由于在睡眠状态接收到的消息必须放入缓冲区内，因此最大的帧周期以缓冲容量为界。

1）簇与同步

T-MAC 协议中的同步是通过一种称为虚拟簇的技术完成的 [Wye02]。在虚拟簇中，有相同调度机制的节点形成簇，并非强制所有节点使用相同的调度机制。虚拟簇允许节点广播其调度机制，同时期望节点能够与相邻节点保持同样的调度机制。开始，每个节点先监听无线信道并且等待，如果一个节点在经过一段时间的监听和等待后没有收到数据，那么它将选择一个帧调度机制并且向邻居广播其 SYNC 分组。另一方面，如果一个节点收到了来自任意一个邻居的 SYNC 分组，那么它将按照收到的 SYNC 分组内的信息进行调度。此外，如果节点在广播其 SYNC 分组后收到了另一个 SYNC 分组，那么它将采用两种调度机制并且通知该 SYNC 的发送者存在多种调度机制。节点偶尔广播其调度信息，并且在无规律的间隔后，还要使听完整的时
84 间帧，才能监测出同一簇内存在的不同调度机制。由于同属于一个虚拟簇内（即与发送节点的调度机制相同）的相邻节点和采用发送节点的调度机制作为额外调度机制的相邻节点，在发送节点处于活动状态时均为能接收数据的唤醒状态，发送节点在其活动状态开始时就可以发送数据包。

2）竞争解决

在 T-MAC 协议中，每一帧包含一个活动状态和一个睡眠状态。在睡眠状态下，感应数据加入到发送队列。因此，在每帧活动状态的开始，每个节点已经缓存了大量的数据，需要以突发形式发送出去，这就导致了在活动状态开始时对介质的高度竞争。T-MAC 协议采用了 RTS/CTS 交换机制来应对信道竞争，节点在准备发送 RTS 帧前会在竞争周期内随机地对介质检测一段时间。在发送 RTS 帧后，如果接收者处于睡眠状态，发送者不会收到 CTS 帧。即使接收者处于活动状态，也可能由于冲突导致 RTS 帧丢失或者因为 RTS 或 CTS 帧串音，接收者不能回

复 CTS 帧。由于接收者在活动状态，那么发送者可以重试 RTS 帧的发送，但是如果发送者在两次重试后都没有收到 CTS 帧，它将转入睡眠状态。

在前面已经提到，在 TA 时间内如果没有激活事件发生，节点将转入睡眠状态。这意味着接收者如果没有及时收到 CTS 帧，它将自动切换到睡眠状态。因此，TA 值的大小必须能够保证发送者足以收到 CTS 帧 [Tvdam03]。作为第三方的相邻节点，在监听到 RTS 或者 CTS 帧后，和 S-MAC 协议中类节点转入睡眠状态不一样，T-MAC 选择继续监听。由于监听到发送节点发出的 RTS/CTS 帧的节点很有可能就是该发送节点接下来发出数据包的接收者，避免监听会导致节点无法监听并及时接收数据包，从而降低无线传感器网络的吞吐量。

3）T-MAC 的早睡问题

在文献 [Tvdam03] 中，研究发现 T-MAC 在所有节点向汇聚节点发送数据时存在一些缺陷。比如，假设有 4 个节点 A、B、C 和 D，数据传输方向是 A→B→C→D，即 A 只传到 B，B 传到 C，C 传给 D，如图 3-10 所示。以节点 C 为例，为了与节点 D 通信，节点 C 首先必须通过与其相邻节点 B 竞争获得传输信道的使用权。如果节点 C 收到节点 B 发出的 RTS 帧，那么它将向节点 B 回复 CTS 帧从而暂时失去传输信道的使用权。与此同时，节点 D 也会监听到节点 C 发出的 CTS 帧，节点 D 将会在节点 B 和 C 的通信过程中保持唤醒状态。同理，如果节点 C 监听到节点 B 向节点 A 发出的 CTS 帧，将会保持唤醒状态进行信道监听。而节点 D 由于不知道节点 A 和 B 通信的存在将会转入睡眠状态。这样，即使节点 C 在随后的竞争中获得了传输信道的使用权，但是由于节点 D 处于睡眠状态，节点 C 也只有等到下一个周期才能传输数据到节点 D。这种情况就称为早睡问题，有两种方法可以解决该问题：未来请求发送（Future Request To Send，FRTS）和满缓冲区优先（full buffer priority）。 [85]

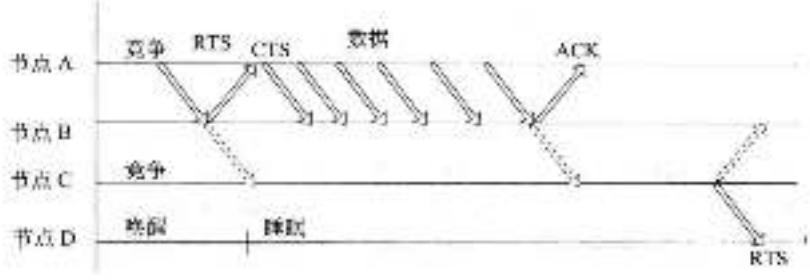


图 3-10　早睡问题（摘自 Van Dam，T. and Langendoen，K.，An adaptive energy-efficient MAC protocol for wireless sensor networks，Proceedings of the First International Conference on Embedded Networked Sensor Systems，Los Angeles，CA，November 2003，ACM，New York，171－180.）

4）未来请求发送（FRTS）

FRTS 的基本思想是让另一个节点知道还有信息要发送给它，即使传输介质在当前不可用。如图 3-11 所示，当节点 C 监听到 B 发送给 A 的 CTS 帧后，立即向下一跳的接收者 D 发出 FRTS 分组，通知节点 D 有数据传输。FRTS 分组包含目的地址以及当前传输需要的时间长度等信息，在这一段时间内节点 C 不能向节点 D 发送数据。如果一个节点被禁止发送数据，那么它也不能发送 FRTS 分组。

FRTS 分组的目的节点必须处于唤醒状态，这样才能接收 FRTS 分组以获取当前数据传输的信息。为了防止其他节点占用传输介质，之前信道的获得者（即节点 A）在发送实际数据前需要发送一个 DS（data-send）分组。该 DS 分组不包含任何有用信息，仅实现对信道的占用。因此，DS 分组和 FRTS 分组的冲突不会影响接下来的数据传输。FRTS 方法可以提高数据吞吐率。

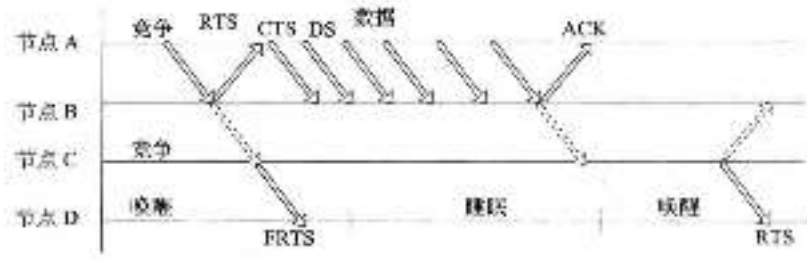


图3-11 未来请求发送（摘自 Van Dam, T. and Langendoen, K. An adaptive energy-efficient MAC protocol for wireless sensor networks, Proceedings of the First International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, November 2003, ACM, New York, 171－180.）

但 DS 分组和 FRTS 分组带来了额外的通信开销。

5）满缓冲区优先

当一个节点传输/路由缓冲区接近饱和时，节点将优先选择发送信息，而不是接收信息。如图 3-12 所示，节点 B 向节点 C 发送 RTS 帧，节点 C 因其缓冲区快满不发送 CTS 帧，而是向节点 D 发送 RTS 帧，然后将其数据发送给节点 D。

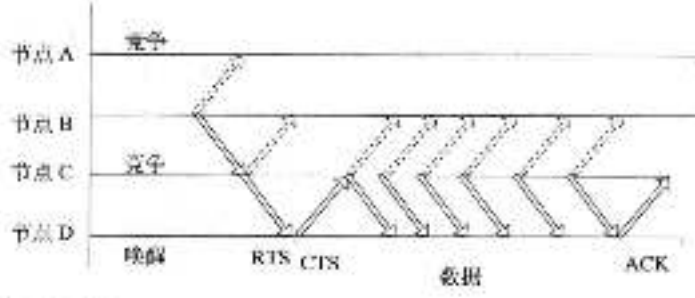


图3-12 满缓冲区优先（摘自 Van Dam, T. and Langendoen, K., An adaptive energy-efficient MAC protocol for wireless sensor networks, Proceedings of the First International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, November 2003, ACM, New York, 171－180.）

不过，在这种机制下，前一次竞争胜者的接收者有更高的概率来控制传输介质。在这个例子中，节点 C 在与节点 B 的竞争中失败，但幸运的是节点 C 是胜者（即节点 B）的接收者，那么实际上节点 C（并非节点 B）就得到了信道。显然，在这个例子中，节点 D 不存在早睡问题。此外，满缓冲区优先机制在网络中引入了一定程度的流量控制，这有利于无线传感器网络中多节点到汇聚节点的通信。

86
87

然而，当高负载流量不是以多节点到汇聚节点的形式流动时，必须谨慎地使用这种方法。当全向通信模式下的所有节点都采用全缓存优先法时，冲突的概率大大增加。冲突会降低整个 WSN 网络的性能，因此，T-MAC 设置了一个阈值来限制节点采用满缓冲区优先。

（2）T-MAC 协议性能

T-MAC 协议引入了在超过预定时间后关闭无线通信模块的思想，是解决空闲侦听和在消息速率有波动的不稳定环境（无论时间还是位置上）下降低能耗的有效途径［Tvdam03］。模拟显示，在低网络负载的情况下，T-MAC 协议的无线模块使用仅占 2.5%，与传统的基于 CSMA 的协议相比可以节省高达 96% 的能量。在高网络负载的情况下，T-MAC 协议通过不

进入睡眠模式被侦听不会增加延迟，同时保证较高的存叶率。在相同的网络流量下，T-MAC和S-MAC协议功耗相近，与基于CSMA的协议相比能耗下降多达98%。但是，在消息速率变化的网络中，T-MAC协议会比S-MAC协议节省更多能量（S-MAC协议会把节点无线电开启一段时间）。



T-MAC协议降低了节点的能耗，同比在不增加延迟的情况下可以延长网络的寿命。它减少了数据从源节点到目标节点的传输时间，并通过采用FRTS和满缓冲区优先的办法解决了早睡问题。

3.3.2 基于调度的MAC协议

在基于调度的介质访问中，各节点根据调度共享传输介质。与基于TDMA的协议类似，通常把时间分割为一个个有固定长度的时隙（time slot）。调度以某种方式决定了时隙的分配，以使每个节点都有机会访问介质，而且避免了冲突。通常，调度会周期性地重复，节点会形成一个簇。由于节点只能在指定的时隙内访问介质，基于调度的MAC协议一般可以避免竞争、冲突和空闲侦听。不需要额外的开销，调度能轻易地让节点转入睡眠状态，从而节省能量。此外，对QoS和优先级的支持能方便地通过基于调度的MAC协议实现。然而，在资源受限的无线传感器网络中，基于调度的MAC协议设计出现了一些挑战性问题：

1）节点间高精度的时钟同步不容易实现。

2）无线传感器网络的动态性，包括节点的增加、失效和移动，使得有效的时隙分配比较困难。

3）在多跳无线传感器网络中分配时隙较为困难。

4）调度中存在的复杂性和较差的扩展性使得网络性能明显降低。

针对以上问题，研究者对设计高效的、基于调度的MAC协议做了一些研究。典型的基于调度的MAC协议有TRAMA［Vrajendran06］、LEACH（Low Energy Adaptive Clustering Hierarchy）［Heinzelman02］、SMACS（Self-organizing Medium Access Control for Sensor Networks）［Ksohrabi00］、FLAMA（Flow-Aware Medium Access）［Vrajendran05］、SPARE MAC（Slot Periodic Assignment of Reception）［Leampelli07］、μ-MAC［Abarros05］、VTS MAC（Virtual Time Division Medium Access）［Eclopes06］、ER-MAC［Hkannan03］以及BMA MAC［Bitmap-Assisted MAC］［Ji04］等。LEACH协议在无线传感器网络的数据传输中引入分级机制。FLAMA协议采用分布式洗牌策略，根据流量信息和两跳内的邻居信息实现了高能效的流量自适应信道访问。在SPARE MAC协议中，节点在某段时间内是接收者，可以收到接收调度（Reception Schedule，RS）并且把接收调度的信息向所有邻居广播。μ-MAC协议根据上层提供的信息将传输信道分为竞争周期和非竞争周期。VTS-MAC协议将节点分为簇，而且将时间线分成时隙，使得网络中的节点个数与时隙个数相等。BMA MAC协议提出一个簇内MAC协议，将网络中的节点分为簇，只有在重大事件发生时簇内节点才能与簇首节点通信。

接下来将详细介绍TRAMA协议［Vrajendran06］的基本思想。

TRAMA协议［Vrajendran06］

TRAMA（流量自适应介质访问）协议（Traffic Adaptive Medium Access Protocol）是一个适用于无线传感器网络的基于调度的MAC协议。它确保在数据传输时不会发生冲突，而且基本

89 是指定接收者的节点转入低功耗状态，以达到节省能量的目的。该协议采用自适应的选举策略选出在某个时间段内传输数据的节点，并且允许节点自行决定何时切换到睡眠状态。通过流量信息，TRAMA 可以避免将时隙分配给没有流量的节点。

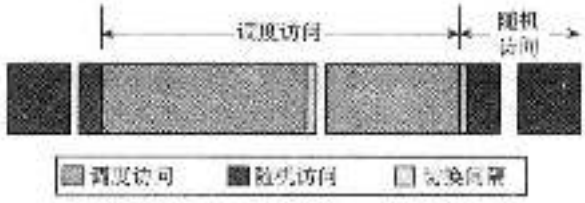


图 3-13　TRAMA 协议时间轴（摘自 Rajendran, V. et al., Energy-efficient, Collision-free medium access control for wireless sensor networks, proceedings of the First International Conference on Embedded sensor Systems (Sensys'03), Los Angeles, CA, February 2006, ACM, New York, vol. 12, No. 1, 63－78.）

TRAMA 协议的时间轴如图 3-13 所示，它包括随机访问和调度访问周期。随机访问周期被称为信号时隙，调度访问周期被称为传输时隙。在信号时隙内，节点以竞争的方式把单跳邻居的信息广播出去，这样它的邻居就可以获得以自己为中心两跳内的拓扑信息。在传输时隙内，节点以非竞争的方式传输数据和通告调度信息。调度信息包含了一组接收者，它们将接收源自该节点的信息。TRAMA 假设时钟同步之前已经完成。在调度访问周期内，时间被分成许多小的时隙，并且调度是固定的。当调度访问周期结束，节点将转回到随机访问周期。信号时隙和传输时隙的长度取决于应用的类型。在动态场景下，节点在网络中从一个位置移动到另一个位置，信号时隙较长。而在静态场景下，节点不需要移动，信号时隙较短。在无线传感器网络中，节点不经常移动，所以信号时隙比较短。

TRAMA 包含三个部分：

1）邻居协议（Neighbor Protocol, NP）

2）自适应时隙交换算法（Adaptive Election Algorithm, AEA）

3）调度交换协议（Schedule Exchange Protocol, SEP）

（1）邻居协议（NP）

TRAMA 协议从信号时隙开始执行。在信号时隙，每个节点随机选择时隙，向其相邻节点
90 广播其单跳邻居信息。在信号时隙结束时，期望所有的节点都能发现它们的邻居。因此，信号时隙的主要目的是允许增加和删除节点，这样能够发现网络拓扑的变化。网络中连通信息由这些信号分组建立。图 3-14 为信号分组中头的格式，信号分组携带邻居节点的增量更新信息，即使没有邻居节点更新，信号分组也被作为“存在”信号发送。否则，如果一个节点一段时间内没有被其他节点侦听到，那么该节点将被认为已经与网络断开连接。一个节点发出的邻居增加更新包含了该节点已增加和已删除邻居的相邻信息。

类型	源地址	目的地址	删除邻居节点数目	新增邻居节点数目	删除邻居节点 ID	新增邻居节点 ID

图 3-14　信号分组头部（摘自 Rajendran, V. et al., Energy-efficient, collision-free medium access control for wireless sensor networks, Proceedings of the First International Conference on Embedded sensor Systems (Sensys'03), Los Angeles, CA, February 2006, ACM, New York, vol. 12, No. 1, 63－78.）

如果一个节点 B 的所有邻居向节点 B 发送了其相应的单跳邻居信息，那么节点 B 就可以获得它邻居的相邻信息。也就是说，节点 B 最终会有其两跳范围内相邻节点的信息，就可以形成以其为中心的两跳内的本地拓扑结构。

应该注意的是，在随机访问周期内，可能发生的冲突会导致信号分组的丢失，从而导致网络中相邻信息的不一致。为了确保相邻信息的一致性，随机访问周期的长度和信号分组重传的次数应该根据实际网络或应用场景确定。

（2）自适应时隙选择算法（AEA）

在发现邻居后，TRAMA 协议采用自适应时隙选择算法（AEA）建立调度。节点根据式 3.5 在本地计算出两跳邻居内的优先度，再根据优先度确定哪一个节点是某一时隙的胜者：

$$\text{Prio}(u, t) = \text{hash}(u \oplus t) \tag{3.5}$$

其中，u 是节点编号，t 是时隙编号，$\text{hash}(u \oplus t)$ 为 hash 函数。

根据上式优先度的计算结果，时隙会分配给胜者（即优先度最高的节点）。为了提高能效，TRAMA 尽可能地把节点切换到睡眠状态，而且可以重用胜者不使用的时隙。比如，如果一个节点没有数据需要发送，它可能会放弃自己的传输时隙，那么该时隙就可以由其他节点使用。

在一个给定的时隙 t 上，在传输期间，节点 u 的状态是由两跳内相邻信息以及 u 的单跳邻居通告的调度信息决定的。每个节点有以下三种可能的状态：

1）睡眠状态

2）接收状态

3）传输状态

如果一个节点需要发送数据并且它是胜者（即根据式 3.5 计算出它有最高的优先度），那么它处于**传输状态**（transmit state）；如果一个节点现在是其他节点指定的接收者，那么它处于**接收状态**（receive state）；否则这个节点处于**睡眠状态**（sleep state），关闭通信系统，不参与任何数据交换。

（3）调度交换协议（SEP）

基于流量的调度信息由调度交换协议（SEP）建立和维护，它在传输时隙内周期性地向邻居广播调度信息。调度按以下步骤生成：

步骤 1：每个节点根据上层应用数据包产生的速率计算通过介质传输数据所需的时隙个数，SCHEDULE_ INTERVAL 表示调度间隔。

步骤 2：节点计算在时隙 $[t, t + \text{SCHEDULE_ INTERVAL}]$ 间它能够成为发送者的时隙个数，即它能在两跳邻居内赢得的时隙个数（根据式 3.5）。

步骤 3：节点将这些将发送数据的时隙通告给指定接收者。由于该节点的所有邻居都会收到有关于其传输调度的所有信息，这样就不会发生冲突。

但是如果该节点没有数据需要发送，它将把这些时隙标记为空（VACANT），并且将该信息发送给相邻节点，让其他节点能够充分利用空时隙。节点作为发送者的最后一个时隙用于广播该节点下一调度间隔的调度信息。

节点通过调度分组向邻居通告调度信息。如图 3-15 所示，调度分组由**源地址**（source address）、**时限**（time-out）、**宽度**（width）、**时隙数**（number of slots）和**位图**（bitmap）等字段构成。源地址表示发出调度通告的节点编号，时限表示该调度的有效期，宽度表示位图的长度，时隙数表示该节点能够成为胜者的时隙个数，位图标识指定的接收者。由于 MAC 层控制的数据传输目标是发送者单跳的邻居，并且从邻居协议（NP）已经获得了相邻信息，因此没有必要在调度分组中指定接收者的地址。TRAMA 采用位图模式标识出指定接收者。位图的长度等于单跳邻居的个数。位图中每一位代表一个单跳邻居，并且是按邻居的编号排序的。如果发送者想向某一相邻节点发送数据，那么位图中相应的位将被置 1。如果某节点不是指定接收

者，那么相应位置0。所以，当位图中所有位被置1时，该调度分组就是广播分组，这是因为所有的单跳邻居都会指定接收者。类似地，多播也很容易实现，只要在多播组节点对应位上置1即可。

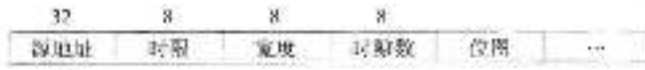


图3-15　调度分组格式（摘自 Rajendran, V. et al., Energy-efficient, Collision-free medium access control for wireless sensor networks, proceedings of the First International Conference on Embedded sensor Systems (Sensys-'03), Los Angeles, CA, February 2006, ACM, New York, vol. 12, No. 1, 63－78.）

节点采用捎带技术，在发送的数据分组内携带节点的调度摘要，尽可能减小了调度分组在广播过程中丢失所造成的影响。节点会维护单跳邻居的调度信息。当节点需要决定向哪里传输或者放弃时隙时，将查询这些信息。更新后的调度表将以摘要的形式被数据分组携带发送。

（4）TRAMA协议性能

TRAMA协议将时间分成连续时隙，根据两跳内节点流量信息，采用分布式选举机制确定在每个时隙上的发送者。同时根据流量信息，TRAMA避免将时隙分配给没有流量的节点，还允许节点自行决定进入睡眠状态的时间。TRAMA协议保证了相隔二跳以上距离的节点可以同时发送数据。与S-MAC协议的性能靠占空比决定不同，TRAMA协议的性能主要依靠流量模式决定。文献［Vrajendran06］中的模拟试验说明了在能耗和吞吐量方面，TRAMA协议的性能远
93 优于基于竞争的协议（如CSMA、802.11以及S-MAC）。但是，TRAMA协议由于调度开销，其延迟大于静态调度MAC协议（如文献［Bao01］）。与基于TDMA的协议类似，TRAMA协议适用于周期性的数据收集或监测传感器网络，这些网络对延迟不敏感，但是对传输可靠性以及能效要求较高。

3.3.3　混合型与事件驱动的MAC协议

在无线传感器网络中有很多既不是基于调度也不是基于竞争的MAC协议（如［Ksarvakar08］、［Ngajaweera08］、［Kjamieson03］、［Szhou07］、［Jpolastre04］、［Irhee08］）。部分MAC协议采用了将基于竞争和基于调度混合的思想，其他一些协议则是事件驱动的。混合型和事件驱动的MAC协议有Zebra MAC［Irhee08］、Sift MAC［Kjamieson03］、FAMA/TDMA Hybrid MAC［Ngajaweera08］、EZ-MAC［Ksarvakar08］以及A^3-MAC［Szhou07］等。

FAMA/TDMA混合型MAC协议将FAMA和TDMA两者结合，为传感器网络中的所有节点提供了介质访问。一开始，网络中的节点通过向基站发送RTS帧竞争获得对介质的访问，第一个成功发送RTS帧的节点获得传输信道的访问权，用于发送数据。在EZ-MAC协议中，数据以低服务访问延迟的形式发送。通过优化的结构序列保持较低的访问阻塞率。它还采用了调度机制。A^2－MAC是一个数据收集协议，同时是混合的分时取CSMA/TDMA协议。接下来将详细介绍几个混合型和事件驱动的MAC协议的例子。

1. Sift MAC［Kjamieson03］

在很多无线传感器网络应用中，传感器节点的目的是监测事件并且向特定节点——基站（base station）报告事件。当事件发生时，所有监测到事件发生的节点会将事件细节报告给基站。由于极有可能有多个邻近节点监测到了事件，它们将共享传输介质。当所有节点同时报告时，就会在传输信道上产生竞争，这样的情况称为空间相关竞争（correlated contention）。然而，由于多个节点同时监测到了相同的事件，它们会向基站报告相似的传感数据，那么就没有必要让所有监测到该事件的节点都向基站报告。只需要事件周边的部分节点向基站报告事件。

另一方面，网络中的节点可能会因电池或者其他原因而失效，那么某一地理区域内节点的密度就会变化。因此，就需要能够有效处理空间相关竞争和随时间变化的节点密度的传感器网络MAC协议，这就是Sift MAC协议的目标[kjamieson03]。 94

（1）协议设计

与传统的CSMA协议类似，Sift MAC协议使用固定长为32时隙的竞争窗口（contention window）。Sift MAC协议在一个给定的间隔内选择一个时隙的概率不是均匀的。在该协议中，节点在时隙$r \in [1, CW]$（其中CW表示竞争窗口的长度）中竞争发送数据。节点根据假想的节点数N竞争一个特定的时隙，假想的节点数N在每个没有发送发生的时槽后会发生变化。假设节点数开始时定为一个较大的值，表明每个节点相应地赢得信道访问的概率就比较小。如果在第一个时隙内没有节点传输，那么所有节点就减少假想节点数，同时使其在下一时隙能够发送的概率倍增。这个过程会不断重复，确保能够很快地在可能的节点范围中选出胜者，从而避免因冲突导致的长延迟。

例如，如果只有一个节点竞争传输信道，那么它能在竞争窗口中获得一个时隙用于传输数据。在数据传输完后，所有的节点再竞争新的时隙，以及估算假想节点数N的值。

（2）Sift MAC协议中退避概率分布

每个节点采用非均匀的概率函数P_r选择时隙$r \in [1, CW]$。当没有节点选择一个时隙用于数据传输时，$r \in [1, CW]$被称为是静默的。类似地，当有多个节点选择了相同的时隙r时，时隙$r \in [1, CW]$被称为是冲突的。如果有节点获得了时隙，那么称之为成功的。只有在仅一个节点选择了竞争窗口内的一个时隙r时它才能赢得一个时隙，该时隙是该竞争窗口内的第一个非静默时隙。Sift MAC采用了式3.6的非均匀概率函数P_r：

$$P_r = \frac{(1-\alpha)\alpha^{CW}\alpha^{-r}}{1-\alpha^{CW}} \quad \text{其中 } r \in [1, CW] \tag{3.6}$$

在式3.6中，α表示（0，1）范围内的分布参数，它会导致P_r的指数增长。这就意味着竞争窗口中靠后的时隙有更高的选中概率。

可以从有CW个阶段的决定过程看出每个节点如何选择时隙。节点从阶段1开始，先把当前的节点数N估计为N_1，然后以相同的概率选择时隙1。如果没有节点选择时隙1，那么节点 95
就认为估计是错误的，然后将估计值减为N_2。之后节点再以一定的概率选择时隙2，如果时隙2还是静默的，那么再将估计值减为N_3……持续上述过程，直到之前出现了$r-1$个静默时隙，那么N_r是N的估计值，如图3-16所示。

由于当前节点数$N \in [1, N_1]$，那么在决定过程中应当一直保持较高的成功概率。因此，需要遵守如下两条性质[Kjamieson03]：

1）当$N = N_r$时，成功的概率应当很高。

2）成功的概率应当是恒定的。

假设在竞争窗口中有r个静默时隙。设P_r^1为节点选择时隙r同时有$r-1$个静默时隙的概率，那么时隙$r+1$是一个获胜时隙的概率由式3.7给出：

$$N_r P_r^1 (1-P_r^1)^{N_r-1} \approx N_r P_r^1 e^{-NP_r^1} \text{（当 } N_r \text{ 很大，} P_r^1 \text{ 很小时）} \tag{3.7}$$

因此，只有在$N_r P_r^1$基本保持恒定时才能很好地保证性质2，那么成功的概率$N_r P_r^1 (1-P_r^1)^{N_r-1}$就不会随时间发生明显变化。

为了确定一个服从恒定的$N_r P_r^1$分布，协议选择使用指数方法，考虑到可能N相当大，而竞争窗口的时隙数很少，根据式3.8减少假想节点数：

$$\beta = \frac{N_{r+1}}{N_r} \tag{3.8}$$

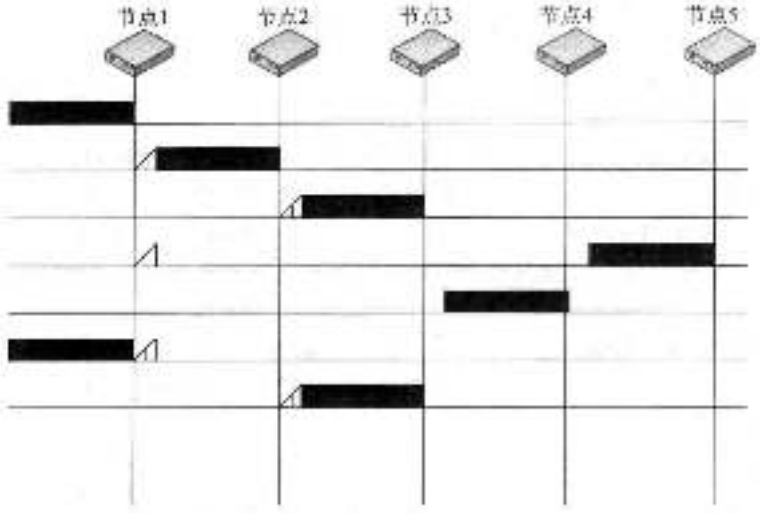


图 3-16　5 个节点场景下 Sift 协议运行时间图。灰色部分是包的传输时机。当信道空闲时，节点在传输之前会按照 Sift 分布规律进行随机退避

在式 3.8 中，β 是恒定的并且 $0 < \beta < 1$。假设没有冲突或者不会有两个节点选择竞争窗口内同一时隙，那么对于节点 S

$$\begin{aligned} P_r^1 &= P_r\{S \text{ chooses } r \mid \text{silence in earlier slots}\} \\ &= P_r\{S \text{ chooses } r \mid S \text{ did not choose earlier slots}\} \\ &= \frac{P_r\{S \text{ chooses } r\}}{P_r\{S \text{ did not choose earlier slot}\}} \\ &= \frac{P_r}{1-(P_1+P_2+\cdots+P_{r-1})} \qquad (3.9) \\ &= \frac{(1-\alpha)\alpha^{CW-r}}{1-\alpha^{CW-r+1}} \qquad (3.10) \end{aligned}$$

$$\frac{P_r^1}{P_{r+1}^1} = \frac{(1-\alpha)\alpha^{CW-r}}{1-\alpha^{CW-r+1}}\alpha \approx \alpha \text{（对于很小的 } \alpha^{CW-r}\text{）} \qquad (3.11)$$

如果 α 与 β 相等，那么式 3.8 与式 3.11 可以相等，因此：

$$N_r P_r^1 = N_{r+1} P_{r+1}^1$$

这证明了即使 N 的值从 N_1 变化到 1，成功的概率都应该是恒定的。对于性质 1，当 $N = N_1$ 时成功的概率应当很高，式 3.10 也指出了 $P_{CW}^1 = 1$，所以如果竞争窗口中所有时隙都是静默的，那么最后一个时隙必须被一个节点选择。因此，α 的取值应该使得处于阶段 CW 的一个节点相信只有一个活动节点。相应地，如果当前活动节点数为 1，意味着 $N = 1$。

96
|
97

从式 3.9 可以看出，如果 $\alpha = \beta$ 并且 $1 = N_{CW} = \alpha^{CW-1} N_1$，那么 $\alpha = N_1^{\frac{1}{CW-1}}$。

（3）协议规范

在 Sift MAC 协议中，每个节点有以下四种状态：

1）空闲状态：节点等待来自其他节点的数据。

2）竞争状态：节点竞争传输信道，希望得到对介质的访问权。

3）接收状态：节点接收来自其他节点的数据。

4）Ack 等待：节点在向某一节点传输完数据后等待该节点的 ACK。

节点在不同状态间切换的伪代码如图 3-17 所示。在图中，函数 pickslot() 用于根据式 3.6 的 Sift 分布选取时隙。指令 moveto（state）用于将某节点的状态改变成给出的状态 state。指令 wait（time）用于等待参数 time 给定的时间。

t_{ss} 是最小时间分隔，如果两个节点各自传输数据超过 t_{ss} 秒，那么相互之间就能听到对方传输的发生。t_{dp} 是始于 ACK 帧的一段时间延迟，用于将节点的状态从发送分组状态转为准备接收 ACK 状态。t_{dp} 是在新的数据传输开始时增加的时间延迟。因此，$t_{cts} + slot * t_{ss}$ 是进行完修数据传输和随后的 ACK 传输所需的时间。$t_{ackwait}$ 是节点等待接收 ACK 的时间。

（4）RTS 与 CTS 机制

为了避免冲突，传感器网络中采用 Sift MAC 协议的所有节点应用 RTS/CTS 交换模式。与 Sift 协议的退避分布用于竞争数据包的发送类似，退避分布也能用于竞争 RTS 帧的发送。因此，仅用“RTS”、“CTS”以及“ACK”替换伪代码中的“frame”就可以实现 RTS 的竞争。

（5）Sift MAC 协议的性能

Sift MAC 协议的基本思想是在一个固定大小的竞争窗口中使用一个增量的、非均匀的概率分布，节点随机地选择传输时隙，这与传统的基于竞争的 MAC 协议类似。Sift MAC 协议适用于并非每个节点都需要报告每个监测事件的传感器网络。模拟试验表明，Sift MAC 协议在空间相关竞争发生时表现出色，能够很好地适应活动节点数的变化。实验结果表明，当网络中参与同一事件报道的节点数达到 512 时，Sift MAC 协议的报道延迟仅相当于 802.11 协议的 1/7。

```
Idle State
wait (channel idle)
if (recv frame for self)
moveto Receive
end if
if (xmit queue not empty)
moveto Contend
end if
Contend state
slot _ pickslot ()
wait t_cts + slot*t_ss
if (channel busy)
moveto Idle
end if
Transmit frame
moveto AckWait
Receive state
Check frame CRC
wait t_dp
Send ACK
moveto Idle
AckWait state
Wait t_ack timeout
if (recv an ACK for self)
discard frame
moveto Idle
end if
Retransmit frame
moveto AckWait
```

图 3-17 Sift MAC 协议中状态切换伪代码（摘自 Jamieson, K. et al., Sift: A MAC protocol for event-driven wireless sensor networks, Proceedings of the Third European Workshop on Wireless Sensor Networks, Zurich, Switzerland, Leture Notes in Computer Science, Vol. 3868, 260 - 275, Springer, New York, May 2003.）

2. B-MAC [Jpolastre04]

为了满足无线传感器网络部署和监测的需要，设计了 B-MAC 协议以实现下述目标：

1）低功耗监听（LPL）。

2）有效避免冲突。

3）实现简单，代码量和 RAM 占用较小。

4）信道充分利用。

5）可以被网络协议重构。

6）可以容忍无线通信频率及网络拓扑结构的变化。

7）可以扩展到大规模节点中。

B-MAC 协议为实现这些目标提供了一些接口，如图 3-18 所示。为了侦听传输信道，B-MAC 协议采用了空闲信道评估（Clear Channel Assessment，CCA）以及数据分组退避机制。

```
interface MacControl {
command result_t EnableCCA();
command result_t EnableCCA();
command result_t DisableCCA();
command result_t EnableAck();
command result_t DisableAck();
command void* HaltTx();
}
interface MacBackoff {
event uint16_t initialBackoff(void* msg);
event uint16_t congestionBackoff(void* msg);
}
interface LowPowerListening {
command result_t SetListeningMode(uint8_t mode);
command uint8_t GetListeningMode();
command result_t SetTransmitMode(uint8_t mode);
command uint8_t GetTransmitMode();
command result_t SetPreambleLength(uint16_t bytes);
command uint16_t GetPreambleLength();
command result_t SetCheckInterval(uint16_t ms);
command uint16_t GetCheckInterval();
}
```

图 3-18　B-MAC 协议接口（摘自 Polastre, J., Interfacing Telos to 51-pin sensorboards, http://www.tinyos.net/hardware/telos/telos-legacy-adapter.pdf, October 2004.）

（1）协议设计

在 B-MAC 协议中，在认为传输信道是空闲时会对信号强度进行采样。例如，在当前传输完成或通信模块没有收到任何数据时就认为信道是空闲的。采样数据放入一个队列中，用衰减因子 α 对队列的中位数进行指数加权移动平均值计算。中位数用于提高对本底噪声（noise floor）估计值的鲁棒性。对本底噪声进行估计后，开始监听传输信道上接收的信号强度。B-MAC协议检测接收的信号强度中的孤立点（异常值）。例如，如果节点检测到孤立点，那么就可以认定信道是空闲的，这是由于一个有效的数据包不可能存在低于本底噪声的孤立点。相反，如果没有检测到孤立点，那么认定信道正忙。

98
:
100

通过图 3-18 中的 MacControl 接口，采用 B-MAC 协议的节点可以打开或关闭 CCA。如果 CCA 不可用，那么 B-MAC 协议采用调度协议。当 CCA 可用时，B-MAC 协议采用分组退避机制。在分组退避中，是没有设定初始退避时间的，而是采用事件驱动方式，它会返回退避时间或者忽略该事件。如果事件被忽略，则会设定一个较短的退避时间。在初始退避时间后，运行 CCA 孤立点算法。如果信道不是空闲的，那么给服务发送一个事件，以便进行阻塞退避定时。

B-MAC 协议提供链路层 ACK 支持。如果需要链路层 ACK，那么接收节点发送 ACK 到源节点。在发送节点收到 ACK 后，则将发送节点发送消息缓冲区内的应答位置位。B-MAC 在周期性的传输信道采样中采用低功耗监听（LPL）机制。每个节点监听信道的传输情况，如果检测到信道上有正在进行的传输，那么它等待传输完成，传输完成后节点转入睡眠状态。如果没有需要接收的数据包，那么定时器将节点转入睡眠状态。两个 LPL 的间隔尽可能大，从而减少信道的采样时间。

（2）B-MAC性能

B-MAC协议在吞吐量和能耗方面优于S-MAC和T-MAC协议。S-MAC和T-MAC协议的性能取决于占空比。B-MAC提供了灵活的接口，从而实现了超低功耗的运行、有效避免冲突和信道的高利用率，还实现了空闲信道估计。在支持在线重构以及向系统服务提供双向接口的同时，B-MAC还采用可适应前导采样机制来降低占空比，减少空闲监听，实现了低功耗运行。B-MAC可以在超低占空比下运行，不会产生因同步和状态保持而带来的额外开销。实验研究表明，B-MAC协议的传递速率、吞吐量、延迟以及能耗都优于S-MAC［Jpolastre04］。 101

3. Z-MAC［Irhee08］

Z-MAC协议是一种混合型协议，它综合了TDMA和CSMA的优点，弥补了两者的不足。Z-MAC采用CSMA作为基本协议，但是会根据竞争程度采用TDMA。Z-MAC协议的开销主要是由开始时的建立过程带来的。在建立过程中，为节点分配用于数据传输的时隙。分配时隙之后，节点在预定的周期（称为“帧”）内使用分配到的时隙传输数据。分得时隙的节点称为该时隙的占有节点，其他节点称为该时隙的非占有节点。对于任何时隙，非占有节点传输数据的优先权低于占有节点。优先权通过竞争窗口的大小确定。如果在某一时间点，该时隙的占有节点没有发送数据，那么非占有节点可以使用该时隙发送数据。当竞争程度较低（流量较低）时，Z-MAC协议的执行与CSMA相似；而当竞争程度较高（流量较高）时，Z-MAC协议的执行与TDMA相似。

（1）Z-MAC的建立过程

在协议开始时，Z-MAC执行建立过程。它包含以下几个步骤：

1）相邻节点寻找。

2）时隙分配。

3）本地帧交换。

4）全局时间同步。

- **相邻节点寻找**

网络中每个节点通过发送ping消息找到一跳相邻节点，ping消息包含一张当前其一跳相邻节点的列表。通过收到的相邻节点的一跳信息表就可以得到两跳内的相邻节点信息。

- **时隙分配**

Z-MAC协议采用分布式RAND（DRAND）算法［Irhee06］为数据传输分配时隙。RAND算法［Rramanathan97］是一种集中式的时隙分配算法，而DRAND算法是RAND算法的分布式实现，DRAND算法循环运行。如图3-19所示，DRAND有四种状态：IDLE状态、REQUEST状态、RELEASE状态和GRANT状态。开始时，每个节点都处于IDLE状态。在IDLE状态下，节点先“抛硬币”（toss the coin），这种方法的结果为正面或者反面的概率各为1/2。如果结果为正面，那么进行抽彩（run a lottery），如果未中彩（lose lottery），则继续保持IDLE状态；如果中彩（win lottery），则转入REQUEST状态，同时该节点向其所有一跳相邻节点广播request消息。 102

假设节点B是节点A的一跳相邻节点。如果节点B处于IDLE状态或RELEASE状态时收到来自节点A的REQUEST消息，那么节点B回复一条允许消息并且转入GRANT状态。如果节点B处于REQUEST状态或GRANT状态，那么它回复一条拒绝消息给节点A。如果节点A在规定的时间内没有收到允许消息或拒绝消息，那么它将再次发送请求消息。

- **本地帧交换**

在分配时隙后，每个节点需要确定可以使用时隙发送数据的发送周期，该发送周期被称为

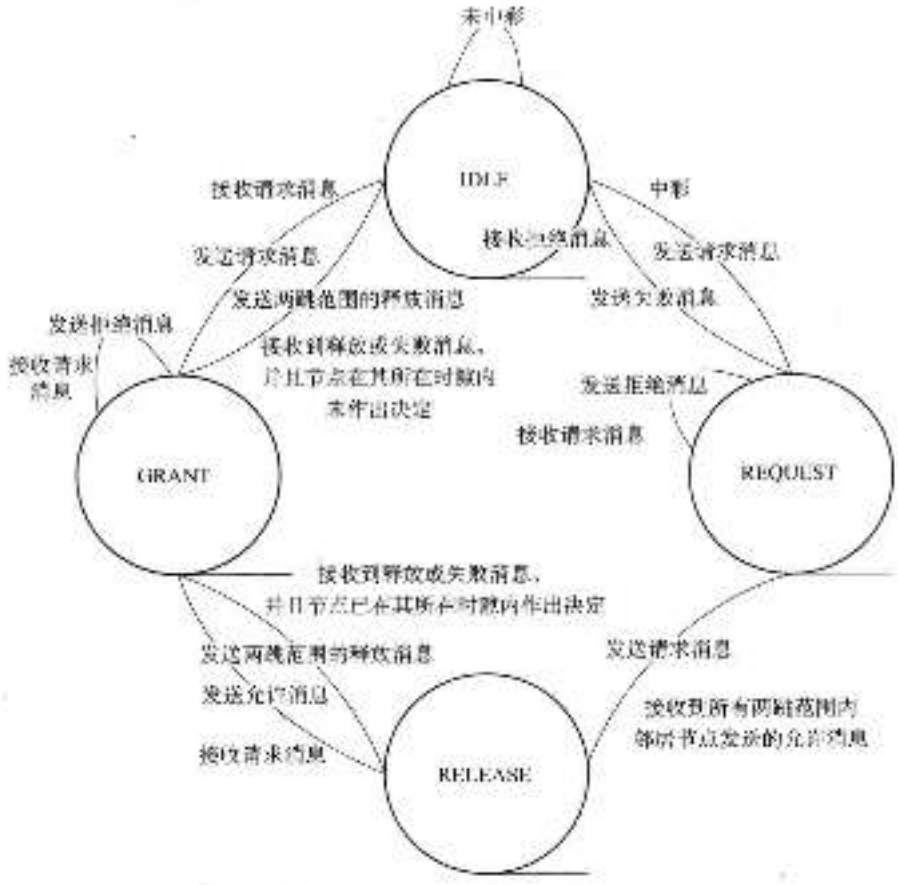


图 3-19 DRAND 状态转移图（摘自 Rhee, I. et al., DRAND: Distributed randomized TDMA Scheduling for Wireless ad-hoc networks, Proceedings of the IEEE MobiHoc, Florence, Italy, May 2006, 190－201.）

时间帧（Time Frame, TF）。在节点确定了发送数据的周期后，节点需要将最大时隙数（Maximum Slot Number, MSN）广播到整个网络中，并且要适应本地时隙的变化。如果网络中有新增的节点，那么 DRAND 算法会为新增的节点分配新的时隙。MSN 的变化同样也需要广播到整个网络中。

在高竞争的情况下，Z-MAC 协议需要时钟同步，因此采用了实时传输协议（RTP/RCTP）[Hschulzrinne96]。在 RTP/RCTP 中，网络中每个节点以一定的速率发送控制信息，限制该速率仅占会话带宽的小部分。在 Z-MAC 协议中，每个节点将数据发送速率限制在一个预先确定的范围内，该范围由能量和带宽决定。

- **全局时间同步**

本地成帧机制要求节点能在时隙 0 内完成同步，这需要所有节点在一个预先确定的时间内进行时隙 0 同步。通过时间同步协议 TPSN [SGaneriwal03]，所有节点都会同步到时隙 0 上。在 TPSN 中，每个节点维护一个 16 位的寄存器作为晶振触发时钟。TPSN 分两步执行：第一步，网络中所有节点构建一个分层的结构，每个节点 k 属于第 i 层，第 i 层的节点可以与 $i-1$ 层的节点通信。只有一个节点处于第 0 层，被称为“树根节点”。第二步是同步阶段，属于第 i 层的节点与第 $i-1$ 层的节点同步。这样，网络中的每个节点都能够和“树根节点”同步，也就可以同步到时隙 0 上。在全局时间同步之后，网络中所有节点就实现了本地时间的同步。

（2）Z-MAC协议的传输控制

Z-MAC协议中的每个节点有以下两种工作方式：

1）轻度竞争方式（Low-contention Level，LCL）

2）激烈竞争方式（High-contention Level，HCL）

节点通常一直在LCL方式下工作，直到其收到了两跳相邻节点发送的直接竞争通知（Explicit Contention Notification，ECN）消息后才按照HCL方式工作。节点在激烈竞争时就会发送一条ECN消息。一旦某节点收到ECN消息，那么它将切换到HCL方式下工作。

103
|
104

直接竞争通知（ECN）消息

ECN消息用于通知当前时隙占有节点的两跳相邻节点在竞争程度激烈时不要成为隐藏终端。在Z-MAC协议中，每个节点需要估计竞争程度，可以采用以下两种方法：

- **计算ACK包的丢包率**

由于两跳的竞争可能会导致冲突，带来数据丢失，因此源节点可以通过计算传输中ACK包的丢包率来衡量竞争的程度。但是，这种方法需要接收者向发送者回复ACK，因而会造成额外的开销并且降低信道利用率。

- **测量信道的噪声等级**

当竞争程度激烈时，传输信道中的噪声等级就会提高。测量传输信道中的噪声水平不需要额外的开销。为了测量传输信道中的噪声，节点需要计算噪声退避数。噪声退避是源节点在发送数据包前采用使用空闲信道评估（CCA）对信道进行检测所花费的退避时间。采用CCA，节点只有在检测到信道是空闲时才能传输数据。当发送节点检测到信道竞争时，它会向竞争节点发送退避消息。如果存在多个竞争节点（即竞争激烈），那么发送节点向其中状态为HCL的竞争节点发送ECN消息。节点f收到其一跳相邻节点i发送的一条ECN消息时，首先检查自己是否为该条消息的目的节点。如果是目的节点，那么节点f向其下一跳相邻节点广播一条ECN消息（这些消息称为两跳ECN消息）；如果不是目的节点，那么丢弃该消息。节点在收到两跳相邻节点发出的ECN消息后，将其HCL标志置位。

（3）发送规则

当一个节点需要发送数据时，它首先检查其是否是该时隙的占有节点。如果该节点是时隙的占有节点，那么它检查信道是否空闲。如果节点发现传输信道是空闲的，那么它可以向目的节点发送数据。否则，它将设计一个计时器，等待时间为T_3，在时间到后，执行CCA，如果信道是空闲的，就可以发送数据；如果信道不空闲，它将等待一段随机的时间，再重复上述过程。如果节点处于HCL方式下，并且它不是时隙的占有节点，那么它将传输时间延迟T_5，然后在竞争窗口$[T_2, T_4]$内进行随机退避。完成随机退避后，节点检测信道，如果信道未被占用，那么发送数据；如果信道被占用，那么节点等到信道空闲再重复上述过程。

105

（4）Z-MAC协议调度的接收

Z-MAC协议是在B-MAC协议[Jpolastre04]上实现的。所以Z-MAC协议也使用低功耗侦听（Low Power Listening，LPL）方式。每个节点维护一个侦听占空比，侦听占空比之间的间隔为检查周期，每次发送数据包之前先发送前导，前导长度等于检查周期。因此，在低占空比下，Z-MAC协议空闲侦听的能耗与B-MAC相近。检查周期是接收调度中重要的因素，这是因为检查周期必须允许一个数据分组的完全传输。因此，时隙的大小必须大于检查周期、T_3、T_4、CCA周期以及传播一个数据分组需要时间的总和。

（5）Z-MAC协议的性能

Z-MAC协议能够根据竞争的程度动态地在CSMA和TDMA之间调整信道访问的方式。协议

利用两跳相邻节点的拓扑信息和松散的同步时钟提升了激烈竞争下的 MAC 协议性能。与TDMA 一样，Z-MAC 实现了激烈竞争环境下信道的高利用率和低延迟。Z-MAC 还有一个重要的特点是能够抵抗同步失败、时隙分配错误以及时变信道状态变化。在最坏情况下，Z-MAC 协议的性能退回到 CSMA。与 B-MAC [Polastre04] 相比，在信道激烈竞争情况下，Z-MAC 协议更有优势，在低竞争状态下与之相当（特别是在能耗方面）。



Sift MAC 协议 [Kjamieson03] 在一跳竞争中展示出了较高的性能，但是在两跳竞争中，它需要依赖 RTS/CTS 而且开销更高。Z-MAC 适用于高数据速率以及两跳竞争激烈的应用中。

106

3.4　总结

本章首先介绍了无线传感器网络中 MAC 协议设计面临的挑战。为了应对这些挑战，研究人员对适用于不同无线传感器网络应用的 MAC 协议设计做了大量研究。本章选取了几个典型的 MAC 协议进行介绍，包括基于竞争的 S-MAC 和 T-MAC、基于调度的 TRAMA 以及混合与事件驱动的 MAC 协议：Sift Mac、Z-MAC 和 B-MAC 协议。

问题与练习

3.1　多项选择题

（1）以下哪项不是 TRAMA 协议中的状态？（　　）

A. 睡眠状态　　B. 接收状态

C. 传输状态　　D. 唤醒状态

（2）Sift MAC 协议中竞争窗口的大小为（　　）。

A. 16　　B. 32

C. 512　　D. 上述选项都不对

（3）Z-MAC 协议综合了哪两种传统的 MAC 协议？（　　）

A. CDMA 和 TDMA　　B. FDMA 和 CSMA

C. CDMA 和 SDMA　　D. CSMA 和 TDMA

3.2　为什么在无线传感器网络 MAC 协议的设计中能量是应该考虑的重要因素？

3.3　Sift MAC 协议的性能是否取决于无线传感器网络中的节点数？为什么当网络中节点数增加时其性能会变化？

3.4　S-MAC 和 T-MAC 的主要区别是什么？

3.5　在网络采用 TRAMA 协议时，节点有哪些不同的状态？描述使用 TRAMA 时，不同状态下的操作。

107

3.6　描述 Z-MAC 协议中每个状态的操作。

3.7　解释在 B-MAC 协议中采用 LPL 和空闲信道估计（CCA）的重要性。

3.8　什么是 T-MAC 协议中的早睡问题？如何解决该问题？

3.9　采用 S-MAC 协议的节点如何选择和交换它们的调度？

3.10　什么是无线传感器网络中的隐藏和暴露终端问题？举例说明在无线传感器网络是如何处理这些问题的？

108

无线传感器网络的路由技术

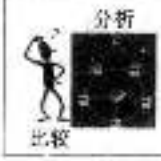
4.1 引言

无线传感器网络（Wireless Sensor Network，WSN）由部署在监测区域内的大量传感器节点组成，通过无线通信方式形成一个分布式自组织网络系统，其目的是通过感知网络分布区域内各种环境信息来实现指定范围内的复杂目标检测与追踪，包括气候变化、地震活动、战场敌军布防、工业监控等诸多领域。为了实现目标检测和追踪，传感器节点需要把感知信息发送到基站（Base Station，BS）或者汇聚节点（sink）进行集中处理。考虑到传感器节点的通信能力有限制且网络覆盖区域大，感知信息无法直接发送给基站或汇聚节点，只能采取多跳转发的数据传输方式。同样，汇聚节点为了获取特定位置感知信息而发送的查询命令也只能采用多跳转发的方式发送到相应位置的传感器节点。因此，无线传感器网络路由的主要功能就是寻找源节点和目的节点间的优化多跳路径并将感知数据沿着优化路径正确转发。

路由在有线网络、无线网络和移动自组织网络（Mobile Ad Hoc Network，MANET）中起着重要的作用，已经引起了广泛的研究。然而，由于无线传感器网络的特殊的资源限制条件和应用需求，现有 Internet 和 MANET 网络的路由协议并不适用于无线传感器网络。例如，绝大部分的互联网路由协议都假设网络在误码率极低的高可靠有线链路上工作，MANET 路由协议大多为拥有对称链路的高移动性节点间的通信选择优化路径。这些假设条件对于无线传感器网络而言都是不可能达到的。除了与 MANET、无线局域网等传统无线网络路由协议一样要面临无线通信环境下链路不稳定性等挑战外，无线传感器网络路由协议还有其不同于传统无线网络的独有问题：资源受限（包括能量、通信带宽、计算能力）、高损耗无线链路、网络容错性、数据报告与融合、节点部署、网络可扩展性与覆盖度、网络动态性，以及节点/链路的异构性 [Nyamaa04]。

4.1.1 资源受限

传感器节点通常是靠能量十分有限的电池供电。由于传感器节点分布区域广，部署区域环境复杂，有些区域甚至人员不能到达，因此传感器节点通过更换电池或者充电的方式来补充能量是不现实的。通过能量均衡优化设计延长网络的生存周期是无线传感器网络路由协议设计的重要目标。如果网络中节点能量消耗不均衡导致个别节点过早能量耗尽，那么不仅能量耗尽的节点自身失去感知数据采集能力，还会导致该节点不能转发其他节点的感知数据，引起大量的报文重传、路径重新选择，从而大大增加了网络传输延迟并缩短网络生存周期。



无线传感器网络路由协议的首要设计目标是能量的高效使用，国内外研究者已经针对无线传感器网络中能量感知的路由技术展开了广泛的研究；而一般说来，传统 Internet 路由协议设计并不会特别考虑能量消耗因素。

同样，传感器节点有限的通信带宽、存储能力、计算能力也是设计无线传感器网络路由协议时要考虑的主要因素。例如，由于存储能力有限，传感器节点不能通过存储大规模网络的全

网拓扑信息或者大规模路由表来进行路由选择。

4.1.2 容错性

与传统有线网络不同，传感器网络中的节点和链路更容易出现故障或失效的情况。传感器
[110] 节点有可能因为能量耗尽或者物理损坏而不能正常工作。无线链路会因为传感器节点故障、环境干扰和障碍物等因素影响而失效。这些不可靠性要求路由机制能够具有一定的容错能力。无线传感器网络路由协议可以通过动态选择替代路径或者利用网络冗余性来解决网络中不可预测的失效问题。

4.1.3 数据报告与融合

在无线传感器网络中，传感器节点需要把感知数据以多跳转发的方式报告给汇聚节点。数据报告根据应用和时间响应特征，可以分为时间驱动、事件驱动、查询驱动和混合驱动四种方式。在时间驱动方式中，节点以固定时间间隔来感知环境并对感知数据进行报告，适用于需要周期性数据监控的应用。在事件驱动方式中，只要感知区域内有事件发生，节点就会把该事件信息报告给汇聚节点。在查询驱动方式中，汇聚节点向网络中特定区域发送查询命令以使得该区域内节点收集感知数据并进行报告。上述三种方式的组合就构成了混合方式的数据报告。这些不同类型的数据报告方法在响应及时性、能耗、通信开销等方面有着不同的表现，相应地，需要根据其各自特点选择适当路由协议以取得最佳的路由稳定性和能量开销。

无线传感器网络中存在着大量的冗余数据。例如，物理位置相邻的多个传感器节点可能会把相同的感知信息或者同一事件的不同方面报告给汇聚节点。为了减少不必要的通信量和相应的各种资源消耗，无线传感器网络需要依据某种标准对不同节点产生的数据包进行融合。融合技术包括重复抑制、信号处理、数据合并等方法。

4.1.4 节点部署

无线传感器网络的节点部署方案是根据具体应用的需求决定的，并且对路由协议的性能有着重要影响。常用的节点部署方案有两种：随机部署和人工部署。在随机部署的网络中，节点通过无线自组织方式建立网络。工作在随机部署网络下的路由协议应能够自学习网络拓扑信息并以能量高效的方式动态地转发数据。如果节点以人工方式进行部署，这种部署方案节点的拓扑已知，数据可以通过预先定义的固定最优路径进行传播。然而，人工部署的传感器网络仍然需要动态路由机制以适应由于节点/链路失效带来的拓扑结构的变化。

4.1.5 可扩展性和覆盖度

为了获取精确信息，在监测区域内通常部署大量传感器节点，传感器节点数量可能达到成
[111] 千上万，甚至更多。由于监测区域范围或节点密度不同，不同传感器网络应用的网络规模也不同，路由协议必须能适应在大量节点参与的环境下以能量高效方式完成的数据转发工作。节点加入或撤出都会使网络规模发生变化，监测区域内事件的集中发生会导致网络在特定时间段内出现大量的数据包，这就要求路由机制具有高度扩展性，能够适应网络规模和通信量的变化。

由于自身严格的资源限制，传感器节点的通信距离和感知距离都十分有限，它们只覆盖了无线传感器网络中较小的物理监测区域。因此，针对具体的应用需求，保证必需的网络连通度和覆盖度也是路由协议应该考虑的一个至关重要的因素。

4.1.6　网络动态性和异构性

有些无线传感器网络应用为了满足系统需要，会考虑通过集成到装置使传感器节点或者汇聚节点能够在监测区域中移动。节点移动会导致网络的拓扑结构和连接性不时发生改变。来自于移动节点的路由消息更具有挑战性，因为路由的稳定性直接影响路由消息的可靠传输。节点/链路失效同样会影响网络的拓扑结构和连接性。另外，监测事件也可能是动态的或者静态的。取决于具体应用，如动态目标跟踪和静态森林火灾预警监测等。这就要求设计无线传感器网络路由协议时必须兼顾上述网络动态性和监测事件动态性因素。

很多无线传感器网络应用中都假设传感器节点和节点间无线链路为同构的。然而在实际网络中，不同节点（甚至同一节点的不同阶段）在可用能量、通信范围（节点能够直接收发数据的最大距离）、存储能力/处理能力方面都存在不同。例如，对称链路是有线网络（包括以太网和光网）的基本特征。而在传感器网络中并非所有无线链路都是对称链路。这就要求传感器网络路由协议在进行路径选择时必须充分考虑网络的异构性。



传感器节点在能量、通信能力、存储能力、处理能力等方面都十分有限。这些节点本身的限制和上述提到的传感器网络面临的种种挑战性问题使得研究适用于传感器网络的全新路由协议成为一项必要工作。

4.2　本章的组织结构

针对传感器网络的特征，本章将首先介绍传感器网络路由协议设计的几个基本概念，包括洪泛（flooding）、闲聊（gossiping）和理想分发（ideal dissemination）。接下来对现有的传感器网络路由协议分类方法进行介绍。然后本章着重介绍几种典型的路由协议，包括基于信息协商的传感器网络路由协议（SPIN）、定向扩散路由协议（DD）、低功耗自适应按簇分层路由协议（LEACH）、阈值敏感的能量高效传感器网络路由协议（TEEN）、地理位置和能量感知的路由协议（GEAR）以及多径路由协议。

4.3　无线传感器网络路由协议的分类

与当前的有线/无线网络（如以太网和移动自组织网络等）的路由技术相比，传感器网络中的路由技术存在着很大的差别和挑战［Rwheinzelman99，Jkulik02］。针对传感器网络中大量的资源受限传感器节点的全网统一编址方案由于部署和维护工作量过大而不可实行，因此目前较普遍采用的基于 IP 的路由协议在传感器网络中并不适用。针对不同的传感器网络应用，研究人员提出了许多路由协议。根据不同的分类标准，路由协议可以划分成不同的类别。［Njamal04］

4.3.1　主动式路由协议和反应式路由协议

根据路径的发现方法，路由协议可以分成主动式路由（proactive routing）协议、反应式路由（reactive routing）协议和混合式路由（hybrid routing）协议。在主动式路由协议中，网络中的每一个节点都要周期性地向其他节点发送最新的路由信息，并连续不断地维护到各个节点的路由，每个节点都要保存一个或更多的路由表来存储路由信息。一旦节点有数据要发送，它立刻就能通过查找路由表获取有效路由信息并进行转发。而反应式路由协议恰恰相反，节点的拓

扑结构和路由信息是按需建立的，仅当需要找到到达目的节点的路由信息以便发送数据时，源节点才开始寻找路由。混合式路由协议则是在不同层次分别采用不同的路由策略，混合了主动式路由和反应式路由的优点。

主动式路由协议的优点是路径建立过程几乎没有延迟；而反应式路由协议则需要通过发起一个路由发现过程来建立适当的路径，这会给数据传输带来比较大的延迟，因此反应式路由协议不适用于实时性要求高的传感器网络应用。另一方面，在无线移动自组织网络中，节点的移动会导致网络拓扑结构一直处于快速变化的状态。对于主动式路由协议而言，为了在网络动态变化的情况下保证路由信息的有效性，节点必须不断地进行路由评估和维护，由此造成的路由开销会占据大部分网络带宽资源。随着网络拓扑结构变化速度的加快，到达远距离目的节点的路由精确度将有所降低。特别是当网络拓扑结构的变化速度超过路由请求的频度时，路由信息将完全失效。

4.3.2　平面路由协议和分层路由协议

根据节点在路由过程中是否有层次结构、作用是否有差异，无线传感器网络路由协议可以
113 分为**平面路由**（flat routing）协议和**分层路由**（hierarchical routing）协议。在平面路由协议中，所有节点具有相同的地位和功能，节点间协同完成感知任务。节点会根据需要与网络中任意可达节点进行通信，发布或接收路由信息。平面路由协议简单、健壮性好，但建立、维护路由的开销大，适合小规模网络。在分层路由协议中，传感器网络通常被划分成多个簇或层次，每个簇由一个簇首节点（Cluster-Head，CH）和多个簇成员节点（non-Cluster-Head，non-CH）构成。分层路由协议通常会根据网络中异构节点在能力上的差异，为不同类型的节点分配不同的角色，进行局部范围内的数据融合以降低报告数据的冗余性，从而最大限度地延长网络生存周期。簇首节点不仅负责其所在簇内信息的收集和融合处理，还负责簇间数据的转发，它的可靠和稳定对全网性能影响较大，其失效将导致所在簇内所有节点路由失败。对于中小规模的传感器网络而言，簇的维护开销过大，并不适合采用分层路由协议。分层路由协议扩展性好，适应大规模网络。

事实上，我们还可以根据协议操作、网络流、能量、QoS 感知等不同标准对现有的路由协议进行分类。接下来，本章主要对以下四类典型的传感器网络路由协议进行详细介绍：以数据为中心的路由协议、分层路由协议、基于位置信息的路由协议和多径路由协议。

4.4　以数据为中心的路由协议

为获取尽可能精确、完整的信息，无线传感器网络通常密集部署在很多地理区域内，传感器节点的数量可能达到成千上万，甚至更多。一般情况下，传感器节点在检测到特定事件发生或者接收到来自系统使用者（如汇聚节点或者基站）的查询命令后会产生感知信息并把该信息向汇聚节点或基站报告。由于密集部署区域内节点监测范围互相交叠，邻近节点报告的信息存在一定程度的冗余，各个节点单独传送数据会造成网络能量和通信带宽资源的浪费。如果传感器节点像 IP 路由器一样可靠并且被个网统一编址，那么这种冗余性问题很容易解决。但是，由于传感器节点随机部署，构成的传感器网络与节点编号之间的关系是完全动态的，相应地，节点编号与节点位置没有必然联系。因此，对节点进行全网统一编址并像 IP 路由协议一样通过地址来访问每个节点的路由机制对于传感器网络是不可行的。针对这种情况，研究人员提出了**以数据为中心的路由协议**（data-centric routing protocol），即汇聚节点进行事件查询时，直接将对关心事件的查询命令发送到某个区域，而不是发送到该区域内的某个确定编号的节点。这

种以数据本身作为查询或传输线索的思想更接近于自然语言交流的习惯。查询命令可以通过高层的具有说明性的查询语言来表达，相应地，作为查询命令重要参数的关心事件需要用基于属性的命名规律进行详细描述。

最早提出的以数据为中心的路由协议包括 SPIN 协议和定向扩散协议［Jkulik02，Rwheinzelman99，Clatanagonwiwat00］，它们通过节点间的数据协商来消除数据冗余并降低能耗。受这两种协议影响，研究人员提出了许多类似的以数据为中心的路由协议，比如谣传路由协议（Rumor Routing）［Bdavid02］、最小代价数据转发算法（Minimum Cost Forwarding Algorithm，MCFA）［Fyc01］、基于梯度的路由协议（Gradient Based Routing，GBR）［Cschurgers01］、COUGAR 协 114
议［Yyao02］、能量感知路由协议（Energy-Aware Routing）［Rshah02］等。谣传路由协议适用于数据传输量较小并且节点地理位置信息不可知的网络应用，它的基本思想是：事件区域中传感器节点产生生存周期长的代理（long-lived agent）消息，代理消息沿随机路径向外扩散传播，同时汇聚节点发送的查询消息也沿随机路径在网络中传播，当代理消息和查询消息的传输路径交叉在一起时，就会形成一条汇聚节点到事件区域的完整路径。为了降低能量消耗，MCFA 没有使用节点唯一标识以及相应的路由信息表，每个节点只需以最小的代价维护从自身到汇聚节点的方向估计，而仅仅使用该方向性信息进行数据转发。COUGAR 协议将整个网络视为一个分布式数据库系统，它采用对传感器网络这个数据库系统进行查询（declarative query）的方式获取相应的感知信息，同时它还使用网内数据融合来节省能量。能量感知路由协议在源节点和目的节点之间建立多条路径，根据路径上节点的通信能量消耗以及节点的剩余能量情况，给每条路径赋予一定的概率，使得数据传输均衡消耗整个网络的能量，从而延长整个网络的生存周期。

接下来，本节会详细介绍和分析三种典型的数据分发协议：洪泛/闲聊、SPIN 和定向扩散协议。

4.4.1　洪泛和闲聊

洪泛（flooding）协议充分利用传感器网络无线通信介质的广播特性进行数据分发，是一种最为经典和简单的路由协议。在洪泛协议中，传感器节点不需要对网络的拓扑结构进行维护，也不需要进行路由计算。节点在进行监测数据报告或接收到其他节点的数据包时，用广播方式向所有邻居节点转发数据，邻居节点重复执行上述过程，直到数据包到达目的节点或者该数据包的生存周期结束而被丢弃。洪泛协议的优点是实现极其简单，但是它在数据广播转发过程中产生的冗余数据包大大加重了网络负荷。在基本洪泛协议中，节点会对接收到的数据包进行直接转发，无论相邻节点是否已经从其他源节点接收到副本。这会导致数据内爆（implosion）问题［Rwheinzelman99，Jkulik02］。如图 4-1 所示，节点 D 希望将一条监测数据（a）发送给目的节点 C，使用洪泛协议，节点 D 首先将数据副本（a）广播给它的每一个邻居节点（即节点 A 和 B），节点 A 和 B 又将相同的数据副本（a）转发给节点 C，这样，目的节点 C 就收到了两个相同的数据包，这种数据传输方式就会导致数据内爆。内爆会导致同一数据包的多个副本同时在网络中转发，相应地，每个节点会收到同一数据的多个副本。 115

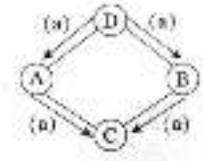


图 4-1　洪泛的内爆问题［Rwheinzelman99］

由于传感器网络节点部署密集，地理位置相邻的传感器节点覆盖的监测区域会出现重叠，可能对区域内同一个事件做出同样的反应，所感知信息也可能会有部分相同。节点会先后收到同一个区域内多个相邻节点发送的相同监测数据，这种现象称为重叠（overlapping）。如图 4-2

所示，节点 A 负责对区域 q 和 r 进行监测，节点 B 的监测区域为 r 和 s。假设区域 q 和 r 内的监测数据为（q，r），区域 r 和 s 内的监测数据为（r，s）。在监测到数据以后，节点 A 和 B 会分别将各自的监测数据（q，r）和（r，s）传送给目的节点 C。显然，节点 C 会接收到两份关于区域 r 的监测数据（r）。

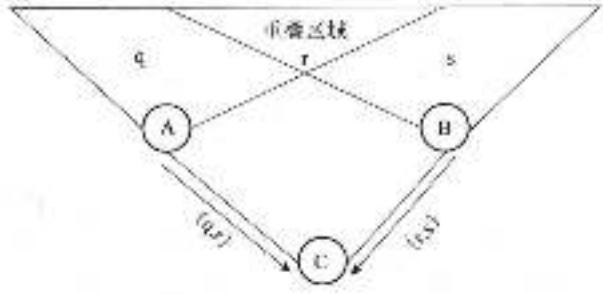


图 4-2　重叠区域示例［Rwheinzelman99］

由于洪泛协议存在的内爆和重叠问题会带来不必要的网络通信量，能量消耗巨大，会导致网络的生存周期大幅缩短，因此不适合于大规模的传感器网络。针对这种情况，研究者提出了概率转发、数据包 ID 标定等策略来减少洪泛过程中的冗余数据包。例如，节点可以为每个数据包分配唯一的 ID，并且缓存所有转发过的数据包 ID。在收到新的广播请求后，节点查询请求数据包 ID 是否在转发缓存表中，如果查询到，则忽略请求，否则广播请求数据包。同样，节点也可以按照一定概率随机地决定是否响应邻居节点的数据请求。但是，上述策略并不能完全解决数据广播带来的冗余问题，而且会对网络性能带来显著的负面影响。闲聊（gossiping）协议是对洪泛协议的改进，当节点收到数据之后，并不像洪泛协议那样采用广播形式将数据包发送给所有邻居节点，而是将数据包发送给某个随机选择的邻居节点。闲聊协议考虑节点的能量消耗和数据冗余性，在选择下一跳时只是随机选择一个节点进行数据转发，但是所选择的路径往往不是最优路径，这将导致数据包的端到端传输延迟增加，甚至在数据没有到达目的节点
116 之前就结束了生命周期。



闲聊协议能够避免信息内爆的现象，但是洪泛和闲聊协议都解决不了监测区域重叠而带来的大量冗余数据包问题。在重叠区域内，多个传感器节点会把对同一事件的基本相同的监测数据分别发送给同一目的节点，从而引起严重的冗余性问题。

理想分发

在理想情况下，传感器节点能够综合考虑数据传输距离、传输时间和能量消耗等因素而选择出一条最佳路径，将数据沿该路径向目的节点转发，而且目的节点对于源节点发出的每个数据包只接收一份，没有任何冗余。这种情况被称为**理想分发**（ideal dissemination）［Rwheinzelman99，Jkulik02］。例如，假设在初始时刻传感器网络中节点 D 持有数据（a，c），节点 B 持有数据（c），而节点 A 和 C 没有任何数据，如图 4-3 所示。为了高效地对数据进行全网分发，节点 D 根据理想分发策略对邻居节点进行有序地选择性数据发送：首先，节点 D 分别向节点 A 和 B 发送数据（a，

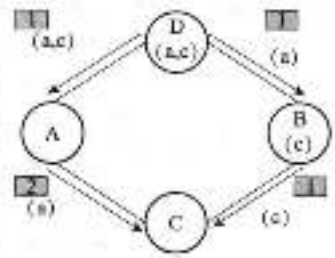


图 4-3　理想分发示例［Rwheinzelman99］

c）和（a），与此同时节点B向节点C发送数据（c）；接下来，节点B或C都可以向节点D发送数据（a），这样就完成了全网数据分发工作。理想分发过程不会产生无谓的传输能量消耗，节点也不会接收到任何冗余数据。当然，在实际的分布式自组传感器网络中，理想分发是不可能实现的。

4.4.2 SPIN：基于信息协商的传感器网络路由协议

为了克服前面提到的洪泛和闲聊协议在数据分发过程中存在的内爆、重叠、资源利用不合理问题，研究人员提出了一组基于信息协商的传感器网络路由协议（Sensor Protocols for Information via Negotiation，SPIN）[Rwheinzelman99，Jkulik02]。

1. SPIN协议设计

SPIN协议的基本思想是使用元数据（metadata）对原始的感知数据命名，元数据描述传感器节点感知数据属性，每个节点在发送完整的数据之前首先使用元数据与邻居节点协商来确定其他节点是否需要该数据，感兴趣的节点向数据发布节点发出数据请求，最后数据发布节点会向请求节点发送感知数据。与洪泛和闲聊协议盲目进行数据发布造成网络资源浪费不同，SPIN协议引入了基于阈值的能量自适应调整机制，通过对数据分发过程中传感器节点的可用能量进行感知，并以此为依据自适应调整参与数据转发工作的积极程度，从而有效延长个网的生存周期。因此，在SPIN协议中，数据转发路径是由网络拓扑结构和传感器节点的可用能量资源共同决定的。

SPIN协议采用了应用层分帧原则（Application-Level Framing，ALF）[Ddclark90]对报告的感知数据进行分帧组包的。由于使用ALF规则，SPIN协议需要根据具体网络应用系统的应用层数据单元对感知数据进行分帧组包，这样每个数据包的数据内容对于应用程序而言都有特定意义。相应地，SPIN协议在设计元数据时要选取传输协议和应用程序共同关注的属性进行命名。通过协商确保传输有用数据，而且是仅仅通过元数据来进行协商，而不是通过实际感知数据进行协商。元数据的数据量较小，所以传输元数据消耗的能量相对较少。如果实际感知数据是全网唯一的，那么其相应元数据也是全网唯一的。同样，如果网络中两个感知数据相同，那么它们的元数据也是完全相同的。一般说来，元数据的格式与具体的应用相关[Rwheinzelman99]。

SPIN协议的另一个重要方面是使用资源管理器监测节点中的可用资源，并作出相应是否参加特定的数据分发的决定。节点内应用程序在发送或处理数据之前会探测资源管理器，采用SPIN协议的节点通过能询资源系统的方式计算当前可用的能量和资源。因此，SPIN协议在路由选择时会综合考虑网络拓扑结构、应用程序数据分布和节点可用资源等因素。

SPIN协议中使用三种类型的消息：ADV消息、REQ消息和DATA消息。SPIN的协商过程采用了三次握手方式。节点在发送DATA数据消息之前，首先用包含DATA相对应元数据的ADV消息向邻居节点通告。当邻居节点接收到该ADV消息后，若需要接收，则向传输发起节点发送REQ请求消息。传输发起节点在收到REQ请求消息后，才将DATA数据消息发送给发送REQ请求的邻居节点。

2. SPIN类别

SPIN协议可适用于不同的无线传感器网络应用和网络场景，它有4种不同的实现形式[Rwheinzelman99]：SPIN-PP、SPIN-BC、SPIN-EC和SPIN-RL。

- SPIN-PP：适合于点对点信道，假设节点能量不受限以及信道不丢包。
- SPIN-EC：在SPIN-PP的基础上增加了能量限制。
- SPIN-BC：适合于广播信道，假设节点能量不受限以及信道不丢包。

- SPIN-RL：在 SPIN-BC 的基础上增加了可靠性传输限制。

（1）SPIN-PP

SPIN-PP 采用点到点的通信模式，并假定两节点间的通信不受其他节点的干扰，数据不会丢失，能量使用不受限制。数据分发节点通过 ADV 向其邻居节点广播消息，有接收数据需求的节点通过 REQ 发送请求，数据分发节点向发送 REQ 请求的节点发送数据 DATA。接收 DATA 消息的节点会对消息内的感知数据和节点本身缓存的感知数据进行数据融合和冗余消除操作，然后再向它的邻居节点广播包含最新融合结果相对应元数据的 ADV 消息，重复如上的过程，最终将数据发送到目的节点。

（2）SPIN-EC

SPIN-EC 在 SPIN-PP 的基础上考虑了节点的功耗因素，增加了能量阈值感知机制。当发现网络中有数据要转发时，节点根据自身的能量存储变化情况并结合能量阈值来动态决定是否参与转发：如果能量不低于设定阈值，则进行信息协商和数据转发；否则便减少信息交换过程的参与，使得传感器节点能够在有效的能量管理下进行数据分发。

（3）SPIN-BC

119 SPIN-BC 针对无线传输介质的广播特性进行设计，采用一对多的通信方法，使得节点能够通过一次广播把相同的数据包发送到所有在通信覆盖范围内的节点。与 SPIN-PP 相同，SPIN-BC的信息协商也是采用了三次握手方式，只是存在以下几点不同：

①在 SPIN-PP 中，每次数据传输只能发送到单个目的节点。因此节点需要对每个邻居节点分别进行元数据 ADV 消息通报。然而，SPIN-BC 通过充分利用广播信道的特点，节点通过单次广播就能把数据发送到其通信范围内的所有节点。

②与 SPIN-PP 不同，SPIN-BC 不允许节点对接收到的 ADV 消息立即进行响应。在接收到 ADV 消息后，节点先判断自身是否需要 ADV 通告的数据。感兴趣的节点设定随机定时器来控制 REQ 请求消息的发送，防止产生重复的 REQ 请求。若节点在自身定时器到时之前接收到来自其他节点的对于同一数据的 REQ 请求，则取消自身定时器以放弃自身 REQ 消息的发送，从而避免了网络中出现重复请求。

③在 SPIN-BC 中，无论接收到多少个 REQ 请求消息，节点对同一 DATA 消息只广播一次。

（4）SPIN-RL

为了解决无线传感器网络中有损无线链路带来的数据差错与丢失问题，SPIN-RL 在 SPIN-BC 的基础上做了两点改进。第一，节点会记录 ADV 请求消息的相关状态，如果在给定时间间隔内没有接收到请求的 DATA 消息，则重新发送请求。第二，限制节点重新发送 DATA 消息的最小时间间隔，SPIN-RL 规定，如果节点在发送数据（a）后又接收到对数据（a）的请求消息，它必须等待给定时间间隔后才能再次发送。

3. SPIN 协议性能的评估［RWheinzelman99，Jkulik02］

在采用 SPIN 协议的无线传感器网络中，节点在发送感知数据之前先要通过描述感知数据属性的元数据进行协商，这样能够保证节点只有在其他节点需要时才进行发送，从而避免不必要的传输能量消耗。文献［Jkulik02］通过 NS2 模拟器对 SPIN 协议的性能进行了模拟实验评估。模拟实验为 SPIN 协议设计了资源管理器（resource manager）。资源管理器通过统计节点的所有行为来计算其能量消耗情况，并根据每个节点的可用能量决定是否参与数据分发活动。

在模拟实验中，网络规模为 25 个节点，随机分布在 40 米 ×40 米的区域内，假设网络在通信过程中没有数据丢失和排队延迟的情况。在初始化阶段，模拟程序为每个节点从 25 条数据中随机选择 3 条数据，这意味着不同节点会有数据重叠的情况。具体的模拟实验参数如表 4-1

所示。

表 4-1　SPIN 协议模拟实验的参数［Jkulik02］

参　数	数　值
节点数	25
链路数	59
平均节点度	4.7 个邻居节点
网络直径	8 跳
平均最短路径长度	3.2 跳
通信距离	10 米
无线传输速度	3×10^8 米/秒
数据处理延迟时间	5～10 毫秒
无线通信速率	1Mbps
发射能耗	600 毫瓦
接收能耗	200 毫瓦
数据长度	500 字节
元数据长度	16 字节

表 4-2 展示了以**理想分发机制**为参考基准协议的模拟实验结果。从实验结果可以看出，SPIN-PP 协议的能量开销远小于洪泛和闲聊协议，它的功耗仅相当于洪泛协议的 27%。这主要是因为洪泛和闲聊协议产生的冗余数据包会消耗大量的能量。从表 4-2 可以发现，洪泛协议发送的 77% 的 DATA 消息是重复的，在闲聊协议中数据冗余度更是高达 96%。而 SPIN-PP 协议只是由于使用了 ADV 和 REQ 两类信息协商消息，才产生了少量的控制流量开销。

收敛时间为从源节点发出数据到网络中最后一个目的节点接收到该数据的时间。洪泛协议采取洪泛的方式进行数据分发，它的收敛时间最短，仅比理想分发机制多 10 毫秒。SPIN-PP 协议的收敛时间比洪泛协议多 80 毫秒，这是因为 SPIN-PP 协议在发送数据前需要进行信息协商，这会增加网络传输延迟。虽然看上去在收敛时间方面 SPIN-PP 协议比洪泛协议表现差很多，但是实际上两个协议之间的收敛时间差是固定不变的，并不随着工作时间的增长而变大。这样，当传感器网络工作时间增长到很大时，SPIN-PP 与洪泛协议之间的收敛时间差距就可以忽略不计［Jkulik02］。

表 4-2　SPIN-PP 协议模拟实验结果［Jkulik02］

性能（相对于理想分发机制）	SPIN-PP	洪泛	闲聊
相对功耗增加量	0.45J	6.3J	44.1J
收敛时间增加量	90ms	10ms	3025ms
相对功耗 vs 节点度相关线比率	1.25 x	5 x	25 x
传输数据冗余度	0	77%	96%

文献［Jkulik02］中的其他模拟试验及分析结果表明，在消耗能量相同的条件下，SPIN-EC 协议可以比洪泛协议多发送 60% 的数据。SPIN-PP 和 SPIN-EC 协议的性能表现都要好于闲聊协议，甚至在某些条件下功耗和传输延迟表现接近理想分发机制。此外，由于能够利用广播信道进行一对多通信，SPIN-BC 和 SPIN-RL 的数据传输速率更快，功耗更低。SPIN-RL 协议能够高效地解决无线信道数据差错与丢失问题，其数据传输能效是洪泛协议的 2 倍。

4.4.3　DD：定向扩散路由

定向扩散路由（Directed Diffusion，DD）是一种典型的以数据为中心、基于查询的路由协议［Clntanagonwiwat00］。应用定向扩散路由的传感器节点使用基于属性的命名机制来描述数据，

并根据需要通过向所有节点发送对某个指定数据的兴趣消息来完成数据收集。兴趣消息用来表示查询的任务，表达网络用户对监测区域内感兴趣的信息，例如监测区域内的目标名称、地理位置、数据发送速率、持续时间长度、时间间隔、温度、湿度等，以属性数值对的形式进行描述。汇聚节点通过其邻居节点传播兴趣消息。在兴趣消息传播过程中，节点利用兴趣缓存机制动态维护报接收数据的属性并建立反向的从数据源到汇聚节点的数据传输路径，同时汇聚节点重新发送兴趣消息以激活传感器来采集与兴趣信息内属性数值对描述相匹配的信息，最后将感知数据沿之前建立好的传输路径进行正向传输，直到汇聚节点。

定向扩散路由由以下几个元素组成：

- 数据（data）：以属性数值对命名。
- 兴趣（interest）：对已命名数据的感知任务。
- 梯度（gradient）：节点到兴趣消息传播路径中上游邻居节点的链路。
- 事件（event）：事件发生后，事件信息会沿多条路径向兴趣的发出节点转发。
- 加强（reinforcement）：一种从多条向汇聚节点发送感知数据的传输路径中选择一条优化路径的机制。

121 / 122

1. 命名模式

定向扩散路由协议使用代表任务特征的多个属性数据对来对任务进行命名描述。例如，动物追踪任务的描述如下：

```
Type = animal                    //监测类型为动物
Interval = 0.5s                  //每隔0.5秒回送事件
Timestamp = 02:02:19             //兴趣产生时间
ExpiresAt = 02:12:19             //任务失效时间
RECT = [-100, 100, 200, 400]     //执行任务节点所在区域
```

如果一个任务描述通过特征表示指定了网络使用者希望获取的感知数据，这样的任务描述被称为兴趣（interest）。网络使用者希望获取的感知数据，即兴趣数据，可以使用属性数值对来命名。例如，在特定区域内监测到动物的传感器会产生一个如下的返回数据消息（应答）：

```
Type = animal                    //监测类行为动物
Instance = cow                   //实例类型
Location = [122, 210]            //节点位置
Confidence = 0.90                //匹配的置信度
Timestamp = 02:02:20             //事件的产生时间
```

传感器网络的应用类型决定了命名方法中的属性类型和相应数值范围的选取。命名方法的选择对于定向扩散路由协议而言非常重要，它决定了任务描述的准确度，甚至会影响到网络的监测性能。

2. 兴趣扩散和梯度建立

汇聚节点周期性地向邻居节点广播兴趣消息。兴趣消息中含有任务类型、任务周期、时间戳、失效时间和目标区域等参数。在初始阶段，兴趣消息作为一种探索手段以确定特定区域内是否存在动物活动，是为了建立源节点到汇聚节点的数据传输路径。为此，最初的兴趣消息仅指定传感器节点以较低的数据发送速率报告事件信息，在上面的例子中，汇聚节点规定传感器节点每隔0.5秒进行一次事件报告。汇聚节点在收到从源节点发来的数据后，启动建立到源节点的加强路径（见后文描述），后续事件报告信息将以较高的数据发送速率进行传输。

123

兴趣扩散过程如图4-4所示。首先，在接收到来自终端用户（如汇聚节点内部的应用程

序）的感知任务后，汇聚节点向邻居节点广播兴趣消息。每个节点都在本地保存一个兴趣列表，对于每一个兴趣，列表中都有一个表项记录发来该兴趣消息的邻居节点、数据发送速率和时间戳等任务相关信息，以建立该节点向汇聚节点传递数据的梯度关系，如图 4-5 所示。每个兴趣可能对应多个邻居节点，每个邻居节点对应一个梯度信息。通过定义不同的梯度相关参数，可以适应不同的应用需求。每个表项还有一个字段用来表示该表项的失效时间值，超过这个时间后，节点将删除这个表项。

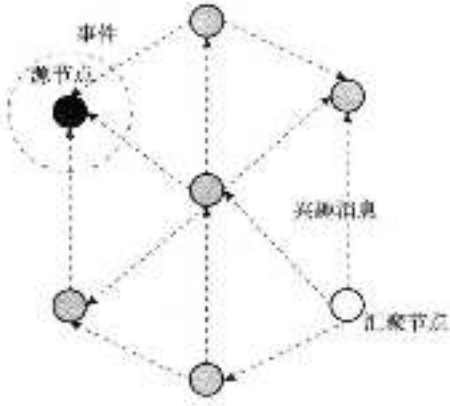


图 4-4　兴趣扩散［Cintanagonwiwat00］

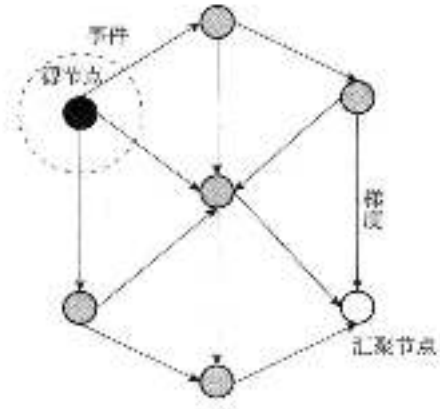


图 4-5　梯度建立［Cintanagonwiwat00］

当节点收到邻居节点的兴趣消息时，首先检查兴趣列表中是否存有参数类型与收到兴趣相同的表项，而且对应的发送节点是该邻居节点。如果有对应的表项，就更新表项的失效时间值；如果只是参数类型相同，但不包含发送该兴趣消息的邻居节点，就在相应表项中添加这个邻居节点；对于任何其他情况，都需要建立一个新表项来记录这个新的兴趣。如果收到的兴趣消息和节点刚刚转发的兴趣一样，为避免消息循环则丢弃该信息；否则，转发收到的兴趣消息。节点在转发兴趣消息时可以根据具体应用要求选用不同的兴趣扩散策略，如表 4-3 所示。

表 4-3　扩散的设计选择［Cintanagonwiwat00］

扩散元素	设计选择
兴趣扩散	洪泛 基于位置的定向/受限洪泛 基于先前缓存数据的定向扩散
数据传播	单一加强路径传输 基于传输质量的多路径传输 基于概率转发的多路径传输
数据缓存和聚合	支持节点失效的容错性数据传输 支持协同感知和数据量压缩 支持兴趣的定向扩散
路径加强	路径加强邻居节点数量选定规则 路径加强时机选定规则 路径负加强机制和规则

3. 数据传播

当传感器节点感知到与兴趣匹配的数据时，会把数据发送到梯度上游邻居节点，并按照兴趣表项上的数据传输速率设定传感器模块采集数据的速率。由于可能从多个邻居节点收到兴趣

消息，节点会向多个邻居节点发送数据，汇聚节点可能收到经过多条路径的相同数据。中间节点收到其他节点转发的数据后，首先查询兴趣列表的表项。如果没有匹配的兴趣表项就丢弃数据。如果存在相应的兴趣表项，则检查与这个兴趣对应的数据缓冲区（data cache），数据缓冲区用来保存最近转发的数据。如果在数据缓冲区中有与接收到的数据匹配的副本，说明已经转发过这个数据，为避免出现传输环路而丢弃这个数据。如果设置的邻居节点数据发送速率大于等于接收的数据速率，则全部转发接收的数据；如果记录的邻居节点数据发送速率小于接收速率，则降速按比例转发。对于转发的数据，数据缓冲区保留一个副本，并记录转发时间。在数据传播过程中，节点为了避免传输环路、提高传输效率和可靠性，可以采用数据缓冲区、梯度调节、路径选择等可选策略，如表4-3所示。

4. 路径加强

定向扩散路由协议通过正向路径加强机制来建立优化路径，并根据网络拓扑的变化修改数据转发的梯度关系。兴趣扩散过程是为了建立源节点到汇聚节点的数据传输路径，这个过程中建立的梯度为探测梯度。汇聚节点在接收到源节点发来的较低速率的数据后，通过发送路径加强消息建立到达源节点的加强路径，以获得更高的数据传输速率。加强后的梯度称为数据梯度。路径加强消息与兴趣消息基本相同，只是任务周期更小（即数据传输速率更高），消息描述如下：

```
Type = animal                    //监测类型为动物
Interval = 10ms                  //每隔10毫秒回送事件
Timestamp = 03:02:19             //兴趣产生时间
ExpireeAt = 03:12:19             //任务失效时间
RECT = [-200, 100, 200, 400]     //执行任务节点所在区域
```

假设以传输延迟作为路径加强的标准，汇聚节点选择首先发来最新数据的邻居节点作为加强路径的下一跳节点，向该邻居节点发送路径加强消息。路径加强消息中包含新设定的较高数据传输速率，则断定这是一条路径加强消息，从而更新相应兴趣表项的到邻居节点的数据传输速率，将其作为加强路径上的下一跳节点。这个邻居节点继续转发路径加强消息，直至到达源节点，从而完成加强路径的建立。路径加强机制可以采用多种路径选择规则来建立加强路径，如表4-3所示。在加强路径上的节点如果发现下一跳节点的数据传输速率明显减小，或者收到来自其他节点的新位置估计，推断加强路径的下一跳节点失效，就需要使用上述的路径加强机制重新确定下一跳节点。

124
-
126

然而，以上描述的过程可能会导致多条路径被加强。假设数据传输路径 P_1 和 P_2 都被选为加强路径。如果汇聚节点认为路径 P_2 更好，那么它会继续发送路径加强消息对 P_2 进行加强，同时它还需要对路径 P_1 实施一个负加强机制以降低通过该路径的数据传输速率。负加强机制包括两种方法。一种方法是持续加强方法，如果那条路径没有被持续加强，那么该路径梯度就会超时终止，其数据传输速率也就相应地降低。因此，汇聚节点将定期地加强路径 P_2，路径 P_1 最终会因超时而对其数据传输速率进行降级。另一种方法是通过发送带有低数据传输速率的兴趣消息而显式地对路径 P_1 进行负加强。

实际上，上述路径加强机制对于多数据源和多汇聚节点的网络也是同样有效的。另外，在定向扩散路由中，除了汇聚节点，加强路径上的中间节点同样能够发起路径加强过程，这对于加强路径上发生链路失效或路径降级时的局部修复是十分必要的。导致加强路径失效或降级的因素包括节点能量耗尽、环境干扰、雨塞等。



定向扩散路由与传统IP方式路由的通信方式存在以下区别：

- 定向扩散路由中所有的通信都是点到点通信，而非传统IP方式路由的端到端通信。
- 定向扩散路由更适用于面向任务的无线传感器网络应用。
- 定向扩散路由中的节点不需要分配全网唯一的地址标识。

5. 定向扩散路由协议总结

定向扩散路由协议是受自然界生物系统（如蚂蚁种群）的启发而提出来的。与一群蚂蚁利用信息素通过相互协作方式找到从蚁巢到食物源的最优路径类似，定向扩散路由定义了兴趣、梯度等概念，通过利用局部兴趣、梯度信息交换的方式，实现能量高效地建立链状的优化数据传输路径。

定向扩散路由协议是一种以数据为中心、查询驱动的路由协议，采用了许多能量高效的机制，包括基于路径加强策略的路由优化、网内数据聚合与缓存等技术。由于采用了相邻节点逐跳数据扩散的通信机制节点，定向扩散路由协议既不要求节点维护全网拓扑信息，也不需要对 [127]
节点进行全网统一编址。数据融合和数据缓存技术可以使得网络能量开销和传输延迟更小。文献［CIntanagonwiwat00］对定向扩散路由协议进行了性能评估。实验结果表明，定向扩散路由协议在能量开销等方面表现突出，其性能优于传统的理想组播数据分发协议。但是，查询驱动的定向扩散路由并不适用于有持续数据传输需求的传感器网络应用［Njamal04］。

4.5 分层路由协议

可扩展性是设计传感器网络，特别是那些节点数目和分布区域都非常大的传感器网络时考虑的主要问题之一。平面路由协议应用在大规模传感器网络中会出现以下问题：

- 收敛时间过长。
- 随着节点密度的增大造成网络负载过重。
- 节点用于存储路由所需网络信息的开销过大。
- 传输延迟、协议复杂性、路径不稳定性明显增强。
- 事件跟踪能力不足，反应较慢。

由于资源受限，传感器节点很难在大规模传感器网络中进行长距离通信。因此，研究人员提出了基于对网络进行分簇或分层方式的路由协议，以确保大规模网络能够在不降低通信质量的前提下正常工作。簇是由多个具有相同任务、位置相近或者具有相同功能/资源的节点组成的节点集合。分层路由协议可以看成是一组工作在不同粒度层次上的平面路由协议。例如，对于一个两层的分层路由协议而言，簇间模块实际上就是一个计算簇与簇之间的簇级路径的平面路由协议；而簇内模块则是一个计算簇内节点之间的节点级路径的平面路由协议。分层路由协议能够为全网范围内的节点通信提供一个簇级路径，相对于平面路由协议的节点级路径，它的路径长度更短、路径稳定性更强。分层路由协议的簇级路径能够大大缓解上述平面路由协议在大规模网络中面临的问题，相应地，也就更适用于大规模传感器网络。另外，数据融合可以在簇内完成，这将大幅度地减少发送到汇聚节点的通信量，从而进一步节省能量消耗。

典型的分层路由协议包括 LEACH［Heinzelman02］、TEEN［AManjeshwar01］、APTEEN［Marati02］、PEGASIS［Slindsay02］、分层 PEGASIS［ASavvides01］、MECN［Vrndoplu99］、SMECN［Lli01］、SOP［Lsubramanian00］、传感器融合路由协议［Qfang03］、VGA［JNal-

[128] karak04]，能量感知的分层路由协议［Qi01］和TTDD［Ye02］等。

LEACH协议以循环的方式随机选择簇首节点，成员节点根据接收到的来自簇首节点的信号强度决定从属的簇，簇首节点作为本簇内普通节点与汇聚节点通信的网关进行数据转发。TEEN协议属于分层路由协议，利用过滤方式来减少网络通信量，适用于感知突发性事件和监测对象的急剧变化，例如温度、气压、降雨量等。APTEEN协议是对TEEN协议的改进，兼有主动型和响应型两种类型的数据传输模式，既可以周期性采集数据又可以对事件做出快速反应。PEGASIS协议将网络中的所有节点形成一个簇，称为链，每个节点只与链上的邻居节点通信，网络中只有簇首节点能够与汇聚节点直接通信。分层PEGASIS协议是对PEGASIS协议的改进，它能够降低端到端的传输延迟。MECN协议利用低功耗的GPS定位技术计算和构建能源有效的子网，要求子网内部节点数目较少并且各节点间传输数据都消耗更低的能量，从而实现不必考虑网络内所有的节点就可以发现全局能耗最低的路径。SMECN协议是对MECN协议的改进，它考虑到任意两节点间可能存在障碍物而不能直接通信的情况，并且取消对于网络充分连通的假设，所构造的满足最小能量转发的子网要小于MECN所构造的子网。在传感器融合路由协议中，节点通过协同处理任务对局部区域内的数据进行融合操作，其中融合操作对象取决于具体任务和资源的要求。VGA协议利用GPS定位技术把网络分成若干区域（zone），每个区域有一个本地融合节点（local aggregator）和一个主融合节点（master aggregator），通过使用数据处理和网内处理技术来延长网络生存周期。数据融合在两个层面上执行：本地融合和全局融合。能量感知的分层路由协议根据位置接近程度对节点进行分组，每个组都有权利根据本组节点的可用能量、传输延迟、传输可靠性等条件选择路径，从而最大程度地降低路由能量开销。TTDD协议主要是解决网络中存在多个汇聚节点和汇聚节点移动问题，当多个节点探测到事件发生时，选择一个节点作为发送数据的源节点，源节点以自身作为网格（grid）的一个交叉点构造出一个网络。利用构造出的网络，源节点可以把数据发送到网络中的任一汇聚节点。

4.5.1　LEACH：低功耗自适应按簇分层路由协议

低功耗自适应按簇分层路由协议（Low-Energy Adaptive Clustering Hierarchy Protocol，LEACH）是一种为传感器网络设计的能量高效的分层路由协议，其主要思想是以循环的方式随机选择簇首节点，将整个网络的能量负载平均分配到每个节点上，从而达到降低能耗和延长网络生存周期的目的［Heinzelman02］。LEACH协议实际上是一个包括了分簇、路由、介质访问控制等多种技术的协议框架，其设计使用的技术如下：

1）本地化的数据传输控制技术。

2）低功耗的介质访问控制技术。

3）自适应动态随机分簇技术。

[129] 4）应用相关的数据压缩和融合技术。

1. 协议设计

由前面章节的介绍可知，由大量传感器节点组成的传感器网络多用于环境监测和特定对象跟踪等应用。由于节点密度较大以及实际应用监测精度需求，位置相近的传感器节点采集到的感知数据存在着很大的相关性和冗余性。LEACH协议通过对传感器节点进行分簇并在每个簇内对相关数据进行处理，从而减少了不必要的数据传输。

LEACH协议按照地理位置将传感器网络中的节点组织成簇（cluster）的结构形式，每个簇都有一个簇首节点（Cluster Head，CH），其他节点作为簇成员节点（non-Cluster Head，non-

CH)。所有的簇成员节点负责采集感知数据，它们只能与本簇的簇首节点通信。簇首节点负责对接收到的本簇内成员节点的感知数据进行融合处理，并把融合数据直接发送到汇聚节点。因此，簇首节点会比簇成员节点消耗更多能量。为了避免节点长期担当簇首而过早耗尽能量，LEACH 协议使用轮转的方式选举节点成为簇首节点，从而让所有的节点都有机会成为簇首节点进而达到均匀消耗网络中节点能量的目的。



LEACH 协议的执行过程是周期性的，每次执行称为一轮（round），每轮循环分为**初始化阶段**（setup phase）和**稳定状态阶段**（steady state phase）。在初始化阶段，邻居节点动态地形成簇，随机产生簇首节点；在稳定状态阶段，簇首节点收集簇内成员节点的感知数据并进行数据融合，然后把融合后的结果发送给汇聚节点。

2. 初始化阶段：形成簇和选举簇首节点

在 LEACH 协议中，每个节点都具有相同的初始能量，能够通过自身功率控制模块调节发射功率。LEACH 协议假设节点初始情况下发射功率相同并且能够调节发射功率，当发射功率 130
足够大时，节点可以直接与汇聚节点进行通信。由于簇首节点的能量消耗比簇成员节点大得多，为了实现网络中节点能量消耗均衡，LEACH 会通过定期从高能量节点中随机选举簇首节点的方式进行簇首节点轮换［Hwendi00］。

在初始化阶段，每个节点都会根据预置的比例以及节点本身曾经担任过簇首节点的次数来决定是否担任本轮的簇首节点。对于节点数目为 n、簇首节点在所有节点所占的预置比例为 p 的传感器网络而言，每轮会选举出 $p \times n$ 个簇首节点。每个节点由阈值 $T(n)$ 决定其是否成为簇首节点，网络中的每个节点产生一个［0，1］之间的随机数，当这个数据大于阈值时，该节点成为簇首节点。其中阈值 $T(n)$ 可由式（4.1）计算得出：

$$T(n) = \begin{cases} \dfrac{p}{1 - p * (r \bmod (1/p))} & n \in G \\ 0 & \text{其他} \end{cases} \tag{4.1}$$

式中，r 为当前轮数，G 为在最近（$r \bmod (1/p)$）轮中没有成为簇首节点的集合，p 是预置的簇首节点的比例（如 5%）。

从式（4.1）可知，在第 0 轮（即 $r=0$），每个节点当选簇首节点的概率均为 p。在接下来的 $1/p$ 轮中，第 0 轮中被选举为簇首的节点就不能再次成为簇首节点。随着有资格竞选簇首的节点个数的减少，集合 G 内节点当选簇首的节点的概率就会相应增加。在经过（$1/p-1$）轮后，$T(n)$ 的值为 1，这时就轮到还没有成为簇首的节点被选举为簇首节点。这样就保证了每个节点都有机会成为簇首节点，达到消耗均衡网络能量的目的。

网络中的部分节点在选择自己成为簇首节点后，会发布公告消息通知网络中其他节点自己是簇首节点且簇已建立。同时，网络中的簇成员节点在接收到此公告消息后根据自己与簇首节点之间的距离来选择加入哪个簇，并且通过向簇首节点发送加入请求消息表示加入，簇首节点在接收到加入请求消息后将该节点加入到簇成员表中。在接收到所有簇成员节点的加入请求消息后，簇首节点作为簇内的控制中心，需要协调簇成员节点的数据传输。为了防止数据传输过程中出现冲突，簇首节点设计一个 TDMA 调度方案，给每个簇成员节点分配一个工作时隙用于发送感知数据。簇内所有成员节点在接收到簇首节点发出的 TDMA 调度方案后就进入稳定状态阶段。上述分布式成簇算法的流程图如图 4-6 所示。

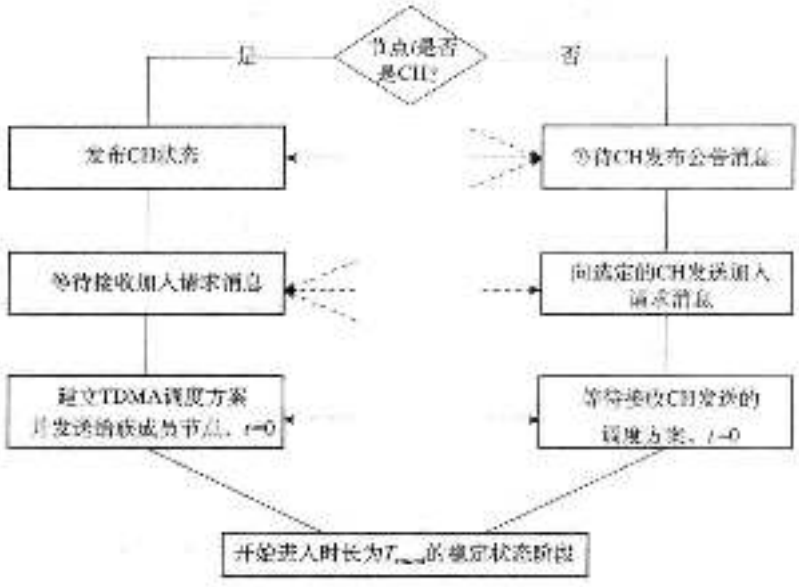


图 4-6　LEACH 协议的成簇流程图［Heinzelman 02］

3. 稳定状态阶段

在接收到簇首节点的 TDMA 调度方案后，簇成员节点就开始按计划在分配给自己的时隙内发送感知数据。为了保证所有节点能够同时进入稳定状态阶段，汇聚节点在全网范围内广播一个相应的时间同步消息。稳定状态阶段的操作会分成多个帧（frame），如图 4-7所示。每个节点在获知自己的工作时隙后，必须在工作时隙内把感知数据发送给簇首节点，而且在一个时隙内只能发送一次数据。每个帧操作的时间长度与簇内节点个数有关，在簇内节点固定的情况下，每个帧操作的时间长度是固定不变的。



图 4-7　LEACH 协议工作原理示意图

为了降低功耗，每个簇成员节点根据接收到簇首节点公告消息的信号的强弱，通过功率控制模块实现以最小的发射功率向簇首节点发送感知数据。另外，簇成员节点在进入分配给自己的工作时隙以外，都将进入休眠状态以节省能量，而簇首节点则需要一直保持工作状态以接收簇内成员节点的感知数据。簇首节点一旦收到所有簇内成员节点的感知数据，就执行数据融合工作，然后以较高的发射功率将处理后的数据传输到汇聚节点。在经过稳定状态阶段的数据传输之后，这一轮结束，网络进入下一轮的初始化和稳定状态阶段。为了最大限度地减少网络控制开销，LEACH 协议中稳定状态阶段的持续时间要明显长于初始化阶段的持续时间。

131
|
132

4. 集中式 LEACH 协议（LEACH-C）

LEACH 协议是一个由每个节点根据随机数自主决定是否当选簇首节点的分布式路由协议。因此，在特定区域内，LEACH 协议每轮产生的簇首节点数量和位置并不固定。相应地，簇的位置也不固定。LEACH-C 协议采用了集中式的成簇算法，由汇聚节点根据全网信息挑选簇首节点，可以有效地解决 LEACH 协议的这一不足。在 LEACH-C 协议中，每个节点通过 GPS 定位技术获取自身的地理位置信息，并把自身地理位置和当前能量报告给汇聚节点。汇聚节点根据

所有节点的报告计算平均能量，当前能量低于平均能量的节点不能成为候选簇首节点。簇首节点从剩余候选节点中选出合适数量和最优地理位置的簇首节点，最后把簇首节点集合和簇的结构信息进行全网广播。如果节点没有当选簇首节点，它将在本轮按照 TDMA 传输计划进入休眠状态。当选簇首节点的节点将在本轮周期内感知数据的接收、融合和向汇聚点转发工作。

5. LEACH 协议总结

LEACH 协议打破了原有分层路由协议中簇首节点固定不变的思想，采用簇首节点随机轮询机制将能量负载均匀分布到网络中的所有节点，有效地避免了部分担任簇首的节点成为通信量/能量的“热点”（hotspot）或者出现单点故障问题（single point failure）。另外，LEACH 协议还采用了动态成簇、网内数据处理、功率可调的数据传输、冲突避免等机制来延长网络生存周期。文献［Heinzelman02］中对 LEACH 协议进行了性能评估。实验结果表明，LEACH 协议的功耗仅相当于直接通信方式功耗的 1/7，相当于最小传输能量路由协议功耗的 1/8～1/4。LEACH-C 协议通过利用全网所有节点的位置和能量信息进行集中式的优化分簇，能够进一步提高网络性能。

然而，在 LEACH 协议中，由于节点只能通过簇首节点发送数据，分属不同簇的两个相邻节点之间不能直接通信。因此，LEACH 协议得到的通信路径并不是最优的。动态成簇机制虽然会增加网络节点能耗的均衡性，但是周期性的簇形成和维护会引入额外的通信开销，一定程度上抵消了协议本身节省的能量。LEACH 协议假设所有传感器节都能与簇首节点和汇聚节点直接通信，这使得 LEACH 协议不适用于监测区域较大的传感器网络应用。因此，研究人员从支持节点异构性、可扩展性、能量高效等方面对 LEACH 协议进行了改进。 133

4.5.2 TEEN：阈值敏感的能量高效传感器网络路由协议

传感器网络可以分为主动型和响应型网络。主动型网络不断采集被监测对象的相关信息，并以特定时间间隔向汇聚节点发送这些信息。SPIN、LEACH 和定向扩散路由协议适用于周期性环境监测或者基于查询环境监测的主动型传感器网络。响应型网络主要用来监测某个特定事件的发生，传感器节点只有在节点监测到相关事件时，才会向汇聚节点发送信息，一般会有实时性要求，如对灾害的监测。阈值敏感的能量高效传感器网络路由协议（Threshold-sensitive Energy-Efficient sensor Network protocol，TEEN）和周期/阈值自适应的能量高效路由协议（Adaptive Periodic Threshold-Sensitive Energy Efficient Network protocol，APTEEN）是专门为具有实时性要求的响应型应用设计的传感器网络分层路由协议［AManjeshwar01］。TEEN 协议是一种以数据为中心的分层路由协议，其基本思想是通过过滤方式来减少数据传输量，能够根据实际应用需要动态调整网络在能量效率、监测精度、响应时间方面上的性能表现。

1. 网络模型

在传感器网络中，汇聚节点可以在任何时间对全网进行广播，但是受自身能量和通信距离的限制，网络中的很多传感器节点不能直接与汇聚节点进行通信。因此，与单层分簇网络结构的 LEACH 协议不同，TEEN 协议定义了多层网络结构进行网络通信，在每一层网络结构中节点根据距离的不同形成不同的网络簇。簇首节点负责收集簇内成员节点的感知数据，对所收集数据进行融合处理，并把融合数据发送到更高层的簇首节点或者汇聚节点。TEEN 协议的多层分簇网络结构如图 4-8 所示。节点 1.1、1.1.1、1.1.2、1.1.3、1.1.4 和 1.1.5 组成了最底层的第一层网络簇，其中节点 1.1 被选举为本簇的簇首节点。同样，节点 1.2 和 1 也被选举为各自所在簇的簇首节点。一方面，节点 1.1、1.2 和 1 作为所在最底层簇的簇首节点；另一方面，

它们又组成了以节点1为簇首的第二层网络簇。这种分层模式不断重复，直至形成簇成员节点可以直接与汇聚节点通信的最上层网络簇，最终形成一个以汇聚节点为根节点的树状多层分簇网络结构。

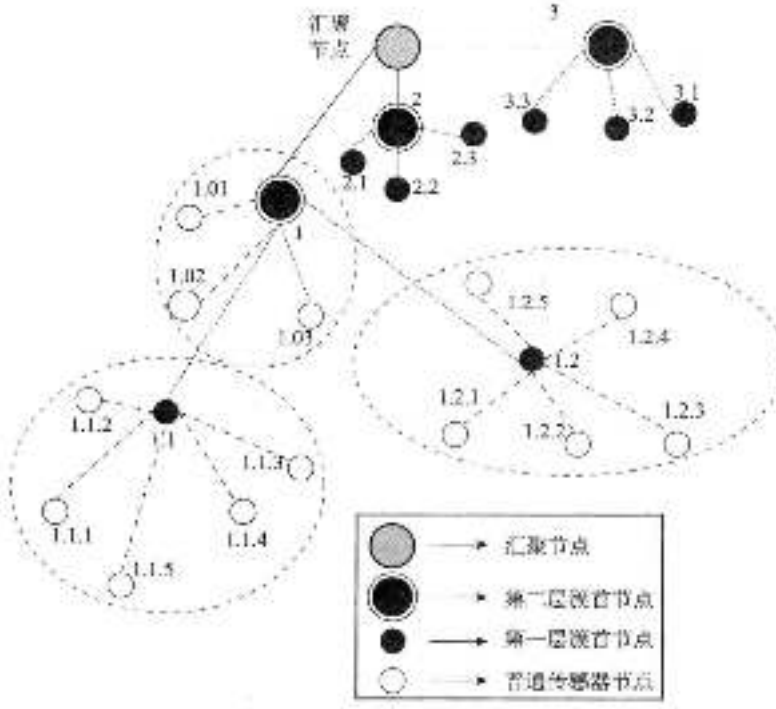


图4-8　TEEN协议的多层分簇网络结构示意图［AManjeshwar01］

在这种多层分簇网络结构中，TEEN协议规定节点只能与其直接簇首节点通信。因此，节点不能像LEACH协议中的节点那样通过调节发射功率直接与汇聚节点通信。在TEEN协议中，第一层网络簇内节点在采集到感知数据后，沿从低到高的层级分簇结构，经由多个簇首节点转发至汇聚节点。每一层级上的簇首节点都会对接收到的感知数据进行必要的数据处理操作（如数据压缩和融合等）以节省通信能耗。另外，为了保证网络节点能量消耗的均衡性，TEEN协议采用了与LEACH协议相同的簇首节点轮换机制。

2. TEEN协议的操作

在应用TEEN协议建簇的过程中，随着簇首节点的选定，簇首节点除了通过TDMA方式调度数据外，同时还向簇内成员节点广播有关数据的**硬阈值**（hard threshold）和**软阈值**（soft threshold）。

硬阈值

硬阈值是被监测对象的属性感知数据所不能逾越的阈值。传感器节点只有检测到感知数据超过硬阈值，才唤醒通信模块把感知数据发送给簇首节点。硬阈值能够保证传感器节点仅发送系统应用感兴趣的取值范围内的感知数据，而不是所有感知数据，从而显著地减少不必要的数据传输。

软阈值

软阈值是被监测对象属性的感知数据的变化范围。传感器节点只有检测到感知数据变化幅度超过软阈值，才唤醒通信模块把感知数据发送给簇首节点。结合上述硬阈值定义可知，传感

器节点只有在监测到的数据值比硬阈值大，并且该数据值与上一次传输的监测数据值之差的绝对值不小于软阈值时，才向簇首节点报告感知数据。软阈值能够进一步减少那些数据值高于硬阈值但基本没有发生变化的感知数据的传输。

TEEN 协议通过调节两个阈值的大小，可以在精度要求和系统能耗之间取得折中。如果要求系统能够反映被监测对象细微的变化，则可以减小软阈值，但这种情况下能耗也会相应增加；如果要求系统监测精度不高，则软阈值可以设定得比较大，这样情况下能耗就会相应降低。

TEEN 协议的工作过程如图 4.9 所示。在每轮的簇建立阶段，簇首节点已经确定，则该簇首节点将重新设定和发布硬阈值和软阈值参数。在簇的稳定工作阶段，节点不断地感知周围环境。当首次监测到数据超过硬阈值时，节点会将感知数据传送至簇首节点，同时将该感知数据保存为感知值（Sensed Value，SV）。此后，只有满足以下两个条件后，节点才会继续向汇聚节点报告感知数据，并将当前监测数据保存为 SV［AManjeshwar01］：

1）当前感知数据的数值比硬阈值大。

2）当前感知数据与 SV 之差的绝对值不小于软阈值。



图 4.9 TEEN 协议工作时序图［AManjeshwar01］

3. TEEN 协议总结

TEEN 协议通过分别利用硬、软阈值对感知数据进行过滤，将数据融合技术与路由技术相结合，既能够把监测对象的突发性变化及时地传送到汇聚节点，又能有效地降低数据传输量，并且基于多层树状结构的分簇网络不要求节点只有大功率的通信能力。TEEN 协议能够根据应用监测需求及当前监测结果周期性地调整阈值，从而实现以最小的通信开销完成实时性监测任务。因此，TEEN 协议适用于实时应用系统，可以对突发事件做出快速反应。但是 TEEN 协议的阈值设置机制会导致某些数据不能够上报，对周期性应用系统则显得支持不足。

文献［AManjeshwar01］对 TEEN 协议的性能进行了模拟实验评估。在模拟实验中，网络规模为 100 个节点，每个节点的初始能量为 2 焦耳，它们随机分布在监测区域内。TEEN 协议采用与 LEACH 协议相同的成簇算法，节点功耗模型包括闲时能量消耗（相当于节点无线通信模块的功耗）和感知能耗（相当于节点无线通信模块功耗的 10%）。模拟实验主要考察了平均能量消耗（average energy dissipated）和当前存活节点个数（total number of nodes alive）两个性能指标。其中，平均能量消耗表示在整个网络生存周期内每个节点由于数据发送/接收、感知、数据融合等行为而消耗的平均能量。当前节点存活个数则表示网络生存周期的大小。模拟实验结果表明，TEEN 协议的性能要优于 LEACH 及 LEACH-C 协议。

4. 周期/阈值自适应的能量高效路由协议（APTEEN）

作为对 TEEN 协议的扩展，周期/阈值自适应的能量高效路由协议（Adaptive Periodic Threshold-Sensitive Energy Efficient Network protocol，APTEEN）是一种混合协议，有主动和响应两种类型的数据传输模式，可以根据用户需要和应用类型来改变 TEEN 协议的周期性和相关软、硬阈值的设定，既能周期性地采集数据又可以对突发事件做出快速反应。它最大的特点是

134
136

随着簇首节点的确定，簇首节点要向簇内成员节点广播发布以下四类参数：

- 属性（attribute）：用来表示用户期望通过网络感知的一组物理参数。
- 阈值（threshold）：该参数包括硬阈值和软阈值，其定义和用途与 TEEN 协议相同。
- 计数时间（Count Time，CT）：表示节点向汇聚节点发送感知数据的最大时间周期。
- 调度（schedule）：采用 TDMA 调度方式，为簇内每个节点分配相应的时隙以共用传输信道。

运行 APTEEN 协议的节点在发送数据时会采用与 TEEN 协议相同的数据发送方式，并且规定如果节点在计数时间内没有发送任何数据，便强迫节点采集并向汇聚节点传送感知数据。
137 APTEEN 协议可以支持三种不同的查询类型：

- 历史查询（historical query）：用来分析过去的数据。
- 一次查询（one time query）：用来快速获得网络的瞬间状态。
- 持续查询（persistent query）：用来在一段时间内持续监测某一事件。

APTEEN 采用了 TDMA 调度方式，簇首节点为簇内每个节点分配一个数据传输时隙。此外，APTEEN 协议还能够根据应用监测需求及当前监测结果周期性地调整阈值、计数时间长度，从而实现以最小的通信开销完成实时性监测任务。模拟实验表明，在能量消耗和网络生存周期的性能指标上，TEEN 和 APTEEN 的性能都要优于 LEACH 协议，APTEEN 协议的性能位于 TEEN 和 LEACH 协议之间。由于减少了数据传输的次数，TEEN 的性能表现最好。TEEN 和 APTEEN协议的主要缺点是：多层分簇网络结构构建、设置阈值功能和计数时间、管理数据传输调度、基于属性的名字查询处理等机制在具体实现上较为复杂，同时也会带来许多额外的开销。

4.6　基于位置信息的路由协议

随着传感器技术、定位技术、嵌入式计算技术的不断发展，许多传感器网络中的传感器节点都通过使用 GPS（Global Positioning System，全球定位系统）设备或者测距设备（ranging device）对自身进行定位，以满足环境监测、目标追踪应用对于节点位置信息的要求。除了利用 GPS 定位，传感器网络还可以利用粗粒度连通性、三边测量、四边测距、声源多模感知等技术对网络中的节点进行定位［Bulusu00，Ward97，Moore04，Girod01］。路由协议可以根据源节点和目的节点的位置信息计算出两节点间的距离，从而根据通信距离调节适当的发射功率并估算出数据传输的能量消耗。另外，回顾本章前面对定向扩散路由协议的介绍可知，汇聚节点根据节点位置信息向特定区域内的节点发布兴趣消息。相应地，研究人员提出了许多基于位置信息的路由协议。利用节点的地理位置信息，这些路由协议能够向特定的区域或者方向传送数据，而不像以往的路由协议需要进行全网广播。这样能够大幅度减少网络中不必要的通信量，显著提高网络性能。典型的基于位置信息的路由协议有：GAF 协议（Geographic Adaptive Fidelity，基于地理位置的拓扑协议）［Yxu01］、GEAR 协议（Geographical and Energy Aware Routing，地理位置和能量感知的路由协议）［Yyan01］、GOAFR 协议（Greedy Other Adaptive Face Routing，贪婪和其他自适应表面路由协议）［Fkuhn03］、SPAN 协议［Bchen02］等。

GAF 协议是能量感知的基于地理位置的路由协议，其最初是应用在无线自组织网络中，但对于很多传感器网络同样适用。GAF 协议以节点地理位置信息把监测区域划分成很多虚拟网
138 格，每个虚拟网格内的节点相互协作，以节点所在的虚拟网格信息作为路径建立的依据。由于数据查询命令中通常包含了地理位置信息，GEAR 协议在查询命令发布时考虑了目的节点的地理位置信息，通过能量和地理位置信息感知的启发式邻节点选择机制，建立通向目的区域的数

据传播路径。GOAFR 协议把贪婪路由策略和表面路由策略相结合，其主要思想是能够自适应地形成一个区域，在这个区域中选择最近的节点作为下一跳节点，能够保证协议性能是最坏情况下的最优值。SPAN 协议根据各个节点的地理位置，从中选择出一些协调节点。协调节点将组成一个骨干网络，传感器节点采集的感知数据将由骨干网络传送至汇聚节点。

GEAR：地理位置和能量感知的路由协议

与单播通信路由协议不同，GEAR 协议能够把数据发送到目的区域内的所有节点，而这种通信方式正是以数据为中心的传感器网络应用的基本工作方式。GEAR 协议根据传感器节点的地理位置信息和剩余能量信息，以启发式下一跳节点选择策略建立汇聚节点到达目的区域的优化路径。每个节点都维护自身到达目的区域的估计代价（estimated cost）和实际代价（learned cost）。其中估计代价由节点的剩余能量和到达目的区域的距离两部分组成，而实际代价则是结合实际通信情况针对网络中存在的路由空洞问题对估计代价的改进。GEAR 协议会根据代价信息选择合适的下一跳节点，以实现能量高效的数据传输。一旦数据到达目的区域，GEAR 协议便采用迭代地理转发策略把数据分发给区域内的所有节点。

实际上，GEAR 协议是对定向扩散路由协议的改进，它限制兴趣消息只能向特定区域发送，而非进行全网范围的广播，从而节省大量的能量开销。

1. GEAR 协议的数据传播过程

在 GEAR 协议中，向目的区域内的所有节点发送数据分组的过程分为以下两个阶段：

1）向目的区域转发数据。

2）在目的区域内分发数据。

在第一阶段，GEAR 协议根据邻居节点到目的区域中心的距离以及邻居节点的剩余能量计算其估计代价，并选择一个估计代价最小的节点作为下一跳节点。之后，在感知数据沿反向路径回传的过程中，通过“捎带”机制计算每个途经节点到目的区域的实际能耗，并以此计算每个节点到目的区域的实际代价。以后则可以根据调整后的实际代价进一步优化到达目的区域的路径。下一小节将介绍邻居节点路径代价的详细计算方法。

在第二阶段，GEAR 协议在目的区域内可以采取四种方式分发数据，即迭代地理转发和区域内洪泛。迭代地理转发的基本思想是：把目的区域分成若干个子区域，分别向每个区域发送数据副本，该过程逐级迭代进行，直至收到数据副本的节点发现自己可以覆盖该子区域，而且距离自己最近的邻居节点并不在该区域。区域内洪泛则是在区域内对数据进行广播。迭代地理转发策略适用于网络节点密度较高的情况，而区域内洪泛策略则更适合在网络节点密度较低的情况下使用。后面的小节将详细介绍上述两种目的区域内的数据分发方式。 139

2. 能量感知的邻居节点计算

GEAR 协议根据邻居节点通往目的区域的实际代价来确定下一跳节点，使得数据分组朝向目的区域转发，同时可以均衡邻居节点的能量消耗。

GEAR 协议的关键是建立和维护节点的实际代价。我们假设节点 N 准备转发数据分组 P 到目的区域 R 中的所有节点，其中目的区域 R 的中心点是 D。节点 N 维护的自身到达目的区域 R 的实际代价用 $h(N, R)$ 表示，估计代价用 $c(N, R)$ 表示。如果节点 N 没有关于节点 N_i 的实际代价 $h(N_i, R)$，则使用估计代价 $c(N_i, R)$ 作为 $h(N_i, R)$ 的缺省值。$c(N_i, R)$ 的定义如式 4.2 所示：

$$c(N_i, R) = \alpha d(N_i, R) + (1-\alpha)\, c(N_i) \tag{4.2}$$

其中 α 是可调系数，$d(N_i, R)$ 是把节点 N_i 到目标中心点 D 的距离（与节点 N 的邻居节点中

到目标中心点 D 最长距离进行归一化处理后的数值），$e(N_i)$ 是节点 N_i 消耗的能量（与节点 N 的邻居节点中已消耗的最大能量进行归一化处理后的数值）。

在节点 N 从其邻居节点中选择到达目的区域 R 的实际代价最小的节点 N_{nn} 作为下一跳节点后，节点 N 重新计算自己到达目的区域 R 的实际代价，如式 4.3 所示：

$$h(N, R) = h(N_{nn}, R) + C(N, N_{nn}) \tag{4.3}$$

其中 $C(N, N_{nn})$ 是从节点 N 传送一个数据分组到节点 N_{nn} 所需的能量消耗，也可以由节点 N 和 N_{nn} 的节点间距离和各自剩余能量计算得出。

在获得其所有邻居节点到达目的区域 R 的实际代价或估计代价后，节点 N 开始从邻居节点中选择代价最小的作为下一跳节点，一般会面临以下两种情况：

1）至少存在一个到目的区域中心点 D 代价更小的邻居节点。

如果存在到目的区域代价更小的邻居节点，GEAR 协议采用贪婪算法建立一条到达目的区域的数据传输路径。对于节点 N 而言，它会从邻居节点中选择到目的区域 R 代价最小的节点作为下一跳节点，并按照式 4.3 将自己的实际代价更新为该下一跳节点的实际代价加上自己到该节点一跳通信的代价。在选择过程中需要考虑以下三种情况：

①如果节点 N 的所有邻居节点的传输能量开销相同，那么选取距离目的区域中心点 D 最近的邻居节点作为下一跳节点。

②如果节点 N 的所有邻居节点的传输能量开销和到达目的区域中心点 D 的距离均相同，那么把待传输的数据分组平均分配给每个邻居节点进行转发。

③对于其他情况，则综合考虑传输距离和能量消耗因素，以实际代价最小为依据选择下一跳节点。

2）不存在到目的区域中心点 D 的代价更小的邻居节点。

如果不存在到目的区域中心点代价更小的邻居节点，则陷入了路由空洞。在这种情况下，节点可以选择一个实际代价最小的邻居节点作为下一跳节点，按照式 4.3 更新自己的实际代价并通知父节点和所有邻居节点。

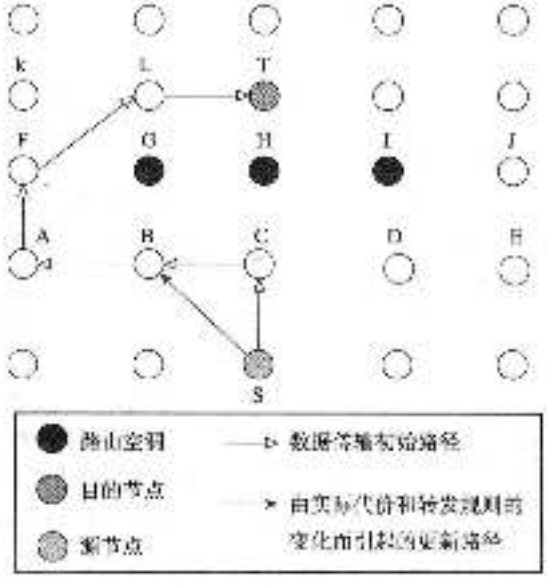


图 4.10　发生路由空洞时的路由发现示例［Yan01］

路由空洞问题的解决方法如图4-10所示。在网格型传感器网络中，节点S为源节点，节点T为目的区域的中心点，每个节点都可以与其8个邻居节点直接通信，每个网格的边长都为1。比如，节点B和C的距离就为1，相邻节点间的一跳通信代价为1，假设节点G、H、I均为能量耗尽的失效节点。为简单起见，式4.2中的可调系数α设置为1，$d(N_i, R)$设置为节点N_i到目标中心点D的距离。在初始时刻，节点S向目的节点T发送数据分组，邻居节点B、C、D到目的节点T代价均小于节点S，其中：

140 ~ 141

$$\begin{aligned} h(B, T) &= c(B, T) = \sqrt{5} \\ h(C, T) &= c(C, T) = 2 \\ h(D, T) &= c(D, T) = \sqrt{5} \end{aligned} \tag{4.4}$$

节点S会选择实际代价最低的节点C作为下一跳节点并把数据发送给节点C。然而，由于节点C的所有邻居节点到节点T的代价都比节点C大，则会陷入路由空洞。这时，节点C会采取以下操作：

- 选取邻居节点中到节点T代价最小的节点B作为下一跳节点。
- 将自己的代价值设置为节点B的代价加上节点C到节点B一跳通信的代价，即$h(C, T) = h(B, T) + C(C, B) = \sqrt{5} + 1$，同时将这个新代价值通知给节点S、B、D。

在下一周期时，节点S的邻居节点B、C、D到节点T代价分别为：

$$\begin{aligned} h(B, T) &= c(B, T) = \sqrt{5} \\ h(C, T) &= c(C, T) = \sqrt{5} + 1 \\ h(D, T) &= c(D, T) = \sqrt{5} \end{aligned} \tag{4.5}$$

这时节点S再发送数据到节点T时就会选择节点B而不是节点C作为下一跳节点，从而绕过路由空洞。

GEAR协议就是通过这种方式来解决路由空洞问题，从而使数据传输得以顺利进行。当第一个数据分组到达目的节点时，正确的实际代价值就会被往后传播一跳距离。每当一个数据分组被转发时，正确的实际代价值都会被传播给一跳范围内的所有节点。另外，由于代价是由归一化的节点能量和距离共同计算得到，可通过改变系数α来调整路径长度和能量消耗因素对路径选择的影响程度。

142

3. 迭代地理转发

当查询命令传送到目的区域后，可以通过受限洪泛方式[Finn87]分发到目的区域内的所有节点。洪泛方式适用于节点密度比较小的情况。受限洪泛方式利用了无线信道的广播特性，节点只需发送一次数据分组，但是发送节点通信范围内的所有节点都能接收到该广播数据分组。

然而，洪泛方式产生的大量冗余数据分组传输消耗了大量的能量。特别是对于节点密度比较大的传感器网络应用，洪泛方式带来的过多的能量开销会严重缩短网络的生存周期，这时可以采用迭代地理转发的方式进行数据分发。如图4-11所示，假设大矩形就是目的区域R，当数据分组P转发到了位于目的区域内的节点N_i时，节点N_i发现自己就在目的区域内，于是把目的区域分成4个小矩形子区域并向各子区域发送数据分组P的副本。该数据分组的传播是一个迭代过程，当节点发现自己是某一个子区域内唯一的节点，或者某个子区域内没有节点存在时，停止向这个子区域发送数据分组。当前所有子区域转发过程全部结束时，整个迭代过程终止。

对于节点密度较低的目的区域，迭代地理转发方式有时会无法停止下来。在迭代地理转发

过程中，数据分组转发和子区域划分操作只有在发现子区域为空或节点唯一时才会终止。然而由于传感器节点的有效通信距离要小于子区域的直径，子区域内靠近边缘的节点无法覆盖整个子区域，相应地，也就无法判断子区域是否为空。在这种情况下，数据分组会围绕一个空的子区域不断地进行无用的转发，直至由于超过最大转发次数（即 TTL 字段减至 0）而被丢弃。这种无谓的分组转发主要发生在节点密度较低的网络中，会消耗大量的能量。另外，迭代地理转发采用单播通信方式，不能充分利用无线信道的广播特性，增加了数据分组的转发次数和能量开销。

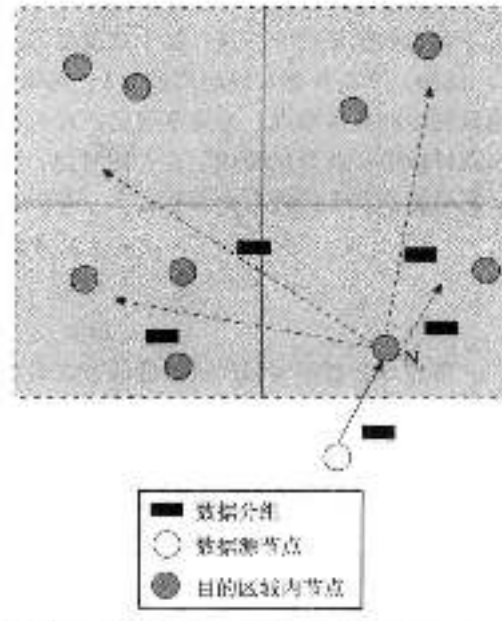


图 4-11　目的区域内的迭代地理转发［Yan01］

受限洪泛机制和迭代地理转发机制各有利弊。当目的区域内节点较多时，迭代地理转发的数据分组转发次数少，而节点较少时使用受限洪泛策略的路由效率高。GEAR 协议可以使用如下方法在两种机制中做出选择：当数据分组到达区域内的第一个节点时，如果该节点的邻居节点数量大于一个预设的阈值，则使用迭代地理转发机制，否则使用受限洪泛机制。

4. GEAR 协议总结

GEAR 协议通过定义路由代价为节点到目的区域的距离和节点剩余能量，利用指南机制获取并不断更新实际路由代价，进行数据传输的路径优化，从而形成能量均衡高效的数据传输路径。在目的区域内，GEAR 协议根据节点的密集程度选择受限洪泛机制或迭代地理转发机制进行数据分发。文献［Yan01］将 GEAR 协议与同类的非能量感知路由协议 GPSR（Greedy Perimeter Stateless Routing，无状态的贪婪路由协议）进行了性能分析比较。GEAR 协议在路径建立的能量消耗和数据接收率方面均优于 GPSR 协议。模拟实验结果表明，在流量分布不均匀的情况下，GEAR 协议的数据接收率比 GPSR 协议高 70% ~ 80%；对于流量分布均匀的情况，GEAR 协议的数据接收率比 GPSR 协议高 25% ~ 35%。而且在上述两种情况下，GEAR 协议取得的网络连通性均优于 GPSR 协议。

143
~
144

4.7　多径 QoS 路由

在传感器网络中，引入多径路由是为了提高数据传输的可靠性、网络吞吐量和实现网络负载均衡。多径路由可以通过经由多条路径发送冗余数据或者利用备用路径对失效通信路径快速恢复等方式来保证数据传输的抗毁性。由于传感器网络中节点通信能力有限，单一路径所能够提供的有限带宽不能够满足多媒体传感器网络应用的通信要求。而多径路由通过利用多条路径同时参与数据传输，可以获得更大的联合带宽以及更小的端到端延时，从而满足视频监控等有高带宽、低延时要求的传感器网络应用的需求。同样，通过把流量分散到多条路径，可以减少网络拥塞现象的发生，在一定程度上实现网络负载均衡。

因此，许多 QoS 路由协议都根据实际系统应用需要，采用不同的多径路由策略来提高不同的网络性能指标。接下来，将分别介绍传感器网络中多径路由的基本原理和采用多径路由策略的 QoS 路由协议。

4.7.1　多径路由

在多径路由中，源节点和目的节点之间存在多条路径，分为链路不相交多路径（link-disjoint multipath）、节点不相交多路径（node-disjoint multipath）和编织多路径（braided multipath）三类。如果源节点和目的节点之间的任意两条路径都没有共同的链路，则称之为链路不相交多路径。如果源节点和目的节点之间的任意两条路径都没有共同的中间节点，则称之为节点不相交多路径。如果源节点和目的节点之间有两条以上路径存在共同的中间节点或链路，则称之为编织多路径。目前，研究人员提出的多径路由对于网络性能的改善主要体现在二个方面：

第一，提高数据传输可靠性。由于多径路由可以同时沿多条路径发送多个数据副本，即使部分路径失效，仍然会有数据副本从其他路径传送至目的节点，从而获得比单路径路由更高的数据传输成功率。但是，多数据副本的重复性传输会增加网络通信量和能量开销。

第二，增加网络吞吐量。在多径路由中，源节点可以把数据分割沿多个路径发送，每个路径形成一个数据流。这样虽然单个路径带宽有限，但是多个数据流同时传输大大增加了源节点至目的节点的网络吞吐量。如果多个数据流在传输过程中能够考虑到相邻节点间的无线干扰因素，那么通过 MAC 协议进行各数据流间的协同工作，多径路由能够取得最大程度的网络吞吐量。

第三，实现网络负载平衡。在多径路由中，源节点可以从建立到达目的节点的多条路径中选出一条路径作为主路径，其他路径则作为备用路径。数据通过主路径进行传输，同时利用备用路径低速传送数据来维护路径的有效性。为了保证网络中节点网络负载/能量消耗均衡性，多条路径轮流担任主路径进行数据传输，这样能有效避免部分路径上的节点由于一直参与数据传输而导致能量过早耗尽的情况。另外，一旦主路径上节点失效，多径路由会立即从备用路径中选出新的主路径，从而保证连续的数据传输。 145



一般来说，在传统单路径路由协议中，如果通信路径上有节点失效，那么该路径上的所有节点会通过广播路由发现消息的方式重新建立通信路径。而多径路由协议可以通过从备用路径中选出新的主路径以替代失效主路径，从而保证不间断的数据传输，整个路径恢复过程不需要任何额外的通信开销。

在文献［Chang04］提出的多径路由协议中，节点根据到下一跳节点的距离调节自身发射功率，从而实现以最小的能量消耗进行数据转发。因此，单位长度数据分组传输的能量消耗是由主路径选择策略所决定。路由问题可以转化为关于最大化网络生存周期的线性规划问题，通过对线性规划问题的求解可以得到理论上最优的通信主路径。源节点根据通信路径上所有链路代价之和最小的原则建立最小代价路径，其中链路代价综合考虑了发送/接收能耗和链路两端节点的剩余能量等因素。一旦发现备用路由的代价小于主路径，多径路由会立即选择该备用路径作为新的主路径，从而保证能耗最低的数据传输。模拟实验结果表明，无论对于固定数据速率场景还是随机数据速率场景，文献［Chang04］提出的多径路由协议取得的网络生存周期都非常接近线性规划求解得到的理论最优值。

由前面介绍可知，多径路由机制通过向多条路径发送多个数据副本来提高网络可靠性，但是这样会产生大量的冗余流量。针对这种情况，文献［Dulman03］提出了一个兼顾网络冗余性和可靠性的多径路由协议，它根据路径的数量 N 和每条路径的失效概率计算出传输冗余度，然后按照传输冗余度将要发送的数据分组分成 N 个部分重叠的数据片段，并分别将这些数据片段沿不同路径进行传输。这样，由于数据片段内的数据信息存在适当冗余，即使有部分数据片

段在传输过程中丢失，目的节点仍然能够利用接收到的数据片段恢复出全部数据。

定向扩散路由协议就是通过多径路由方式来实现数据传输的健壮性［Intanagonwiwat00］。
146 在定向扩散路由的基础上，文献［Ganesan01］进一步分析了如何构造少量路径以使得失效主路径的修复能够不需要全网范围的广播而在局部范围内完成，从而提高网络的能量效率。文献［Ganesan01］对不相交多路径和缠绕多路径这两种典型多径路由机制在能量消耗、延迟和抗毁性等方面进行了分析比较。研究发现，在发生相关性路径失效的情况下，不相交多路径和缠绕多路径的抗毁性基本相同；而在发生独立性路径失效的情况下，缠绕多路径的抗毁性比不相交多路径高50%，备用路径维护开销只有不相交多路径的1/3。因此，对于相关性路径失效和独立性路径失效的路径修复而言，缠绕多路径路由机制是一种兼具较高能量效率和抗毁性的可行方案。

4.7.2　多径 QoS 路由协议

传感器网络的应用范围非常广泛，不同类型的应用往往有着不同的 QoS 需求（如传输延迟、能量消耗、通信带宽、优先级、公平性、连通性、可靠性等）。因此无线传感器网络中的多径路由的 QoS 保障的一大挑战是节点如何在资源有限的条件下协作，以满足系统应用的特定 QoS 需求。传感器网络 QoS 保障机制是在传统网络 QoS 保障机制基础上结合传感器网络的自身特征发展而来的。有些传统网络的 QoS 性能参数（如传输延迟、能量消耗、可靠性等）保障对于资源受限传感器网络仍然具有重要的意义和挑战。但是对于许多节点协作共同完成监测任务的传感器网络应用而言，像传输公平性这类传统网络 QoS 性能参数变得不再重要。SPEED 协议［The03］和 SAR（Sequential Assignment Routing，有序分配路由）协议［Ksohrabi00］是两个典型的多径 QoS 路由协议。

SPEED 协议是一个实时的路由协议，它在一定程度上实现了端到端的数据速率保证、网络拥塞控制和负载均衡机制。为实现上述目标，SPEED 协议首先交换节点的传输延迟，以得到网络负载情况；然后，节点利用局部地理信息和传输速率信息作出路由决定，同时通过邻居反馈保证网络传输速率在一个全局定义的传输速率阈值之上。节点还通过反向压力路由变更机制避开延迟太大的路径和路由空洞。

路由协议分为按需路由（on-demand routing protocol）协议和表驱动路由协议（table-driven routing protocol）两类。表驱动路由协议使节点维护的路由表可以较准确地反映网络的拓扑结构。节点一旦发送报文，可以立即获取目的节点路由，因此，该路由协议的延时较小，但是协议周期性路由维护需要付出较多的路由控制报文，开销较大。按需路由协议并不需要周期性地维护路由，仅在需要路由时才由源节点建立源节点到目的节点的路径，因此，产生的路由控制信息比表驱动路由协议要少得多。但因为数据传输之前必须先获取路由，所以存在一定的延时。SAR 协议以基于路由表驱动的多路径方式满足网络低能耗和健壮性的要求。为了能够建立起每个节点到达汇聚节点的多径路由，从汇聚节点的每个邻居节点开始，以它们为树根，依次扩展建立树状结构。从汇聚节点开始，每一棵树都会尽可能地向具有满足 QoS 或者剩余能量较多的邻居节点延伸和扩展。一般情况下，相邻两棵树之间存在重叠覆盖，使得每个传感器节点位于多棵树上，也就有了多条通往汇聚节点的路径。当构建树完成后，大多数节点都将成为以汇聚节点为根的构建树的一部分，并且由于汇聚节点周围的邻居节点都是这些树的树根节点，因此形成的多条路径针对距离汇聚节点一跳范围内的节点是不相交的。重叠的多少对于建立路由的控制开销和路径可靠性有很大影响。如果重叠过多，则建立一棵树的开销等于建立一颗全局树；如果重叠过小，则一个节点所属的树很少，无法起到通过多路径提高可靠性的作用。每

个节点由于有多条路径到达汇聚节点，因此在选择路径时，能够综合考虑路径上的能量资源、QoS（如传输延迟、通信带宽、丢包率等）与所发送数据分组的优先级等因素。对于路径失效的情况，采用局部维护的方式进行路径维护，在节点密度较大的情况下，这种方式成功可能性很大，但是同时也增加了路径长度。为了避免这一问题，汇聚节点会周期性地重新发起路由计算。文献［Ksohrabi00］对SAR协议进行了模拟实验验证，结果表明，SAR协议根据网络当前的资源情况，为具有不同优先级的业务提供了有质量区别的可靠性服务，但同时也增加了路由维护的开销，特别是当节点数量很大时，开销将十分巨大。

4.8 小结

本章首先分析了传感器网络的特点给路由协议设计带来的影响，归纳了传感器网络路由协议设计的关键技术，即能量高效性、可扩展性和数据传输高效性及常用的解决思路；然后从协议设计的角度对已有路由协议进行分类，并详细介绍了典型路由协议（包括SPIN、定向扩散路由、LEACH、TEEN、GEAR、SAR和多径路由）的工作机理。

问题与练习

4.1 多项选择题

1. 下列路由协议哪些是以数据为中心的路由协议？（　　）

A. 谣传路由（Rumor Routing）　B. MCFA　C. SPEED　D. CBR

2. 下列行为中哪些不是用于数据扩散的？（　　）

A. 洪泛（flooding）　B. 闲聊（gossiping）

C. 定向传播（directional propagation）　D. 以上都不是

3. SPIN协议没有使用哪些数据分组？（　　）

A. ADV　B. REQ　C. ACK　D. DATA

4.2 试说明传感器网络中数据路由选择所面临的三个挑战。

4.3 试分析SPIN-PP和SPIN-BC协议的区别，并简单介绍模拟软件NS2中SPIN协议的资源管理器的使用方法。

4.4 简述定向扩散路由协议中梯度是如何形成的。

4.5 简述LEACH协议中簇的形成过程，以及LEACH和LEACH-C协议中簇首节点的选举过程。

4.6 简述TEEN协议中硬阈值和软阈值的含义和作用。

4.7 简述GEAR协议各个阶段的工作过程，并解释GEAR协议使用受限洪泛机制和迭代地理转发机制的原因及其各自适用条件。

4.8 试对不相关多径路由协议和缓冲多径路由协议的特点进行分析和比较。

147 | 148

149

第5章

Wireless Sensor Networks: Principles and Practice

无线传感器网络传输层技术

传输层是OSI网络参考模型的第四层，它的主要目的是为系统应用提供可靠透明的数据传输服务。为了实现这一目的，传输层需要完成以下两个任务：①通过端到端重传等策略保证数据分组在网络中的可靠传输；②避免或减轻由于网络局部区域的高负荷数据流而可能导致的网络拥塞。目前有线网络和无线局域网传输层的主要技术不能适用于传感器网络，因为这些网络采用的传输层技术都以标准的TCP技术为基础，这会消耗大量的能量、计算和存储资源，并且对于传感器网络而言不具备良好的容错性和可扩展性，在网络拓扑频繁变化或网络规模和节点密度显著增大时不能可靠地工作。本章将介绍传感器网络传输层面临的主要挑战以及几个典型的传感器网络传输层协议。

5.1 引言

为了在资源受限的传感器网络中实现可靠、透明的数据传输，传输层协议的设计需要考虑以下7个方面的要求［Ylyer05］：

1）**通用性**：无线传感器网络传输层协议应该与具体应用、网络层协议和介质访问层协议无关。例如，一个针对特定分层网络拓扑结构设计的传输层协议，就很有可能不适用于采用平面拓扑结构的传感器网络应用。

2）**支持异构数据流**：传输层协议应该能够同时支持连续数据流和事件触发数据流两种数据传输模式。其中连续数据流中各个传感器节点源源不断地将数据向汇聚节点发送，数据流量大，很容易发生拥塞。因此对于连续数据流，传输层协议采用自适应数据速率调整机制对数据量过大的区域及时进行数据速率限制以减少网络拥塞的发生。事件触发数据流则是只有在感知到事件发生时才进行数据传输，其数据量不大但发送时间集中，需要很高的传输可靠性以保证系统及时地检测到事件。

3）**可靠性自适应调整**：根据传输数据的冗余程度，无线传感器网络中数据传输的可靠性要求分为完全可靠性（不允许丢失数据分组）和部分可靠性（容忍丢失少量数据分组）。在一些无线传感器网络应用中，传感器节点可以通过降低数据传输可靠性要求来减少数据重传次数，从而达到节省能量的目的。

4）**拥塞检测和避免**：拥塞检测和避免是传输层协议的主要工作。无线传感器网络中存在着一些称为“热点”的区域，这些热点区域中的通信量远远高于其他区域。网络拥塞一般发生在热点区域中，如何快速、高效地检测出可能的拥塞并采取措施进行拥塞避免是传输层协议设计中所要考虑的关键问题。

5）**分布式/集中式拥塞控制**：一方面传输层协议可以把计算密集的拥塞控制任务集中到汇聚节点执行；另一方面，由于拥塞控制最终是由节点通过调整自身数据速率来实现的，为了避免集中式控制过程中传感器节点发送给汇聚节点的状态信息以及汇聚节点向传感器节点发布的控制信息带来的额外带宽开销，传输层协议还可以把部分拥塞控制任务分布到各个节点，以分布式的方式实现拥塞检测和避免。

6）**规模可扩展**：一方面，传感器节点可能会因为出现故障或耗尽能量而停止工作；另一方面，系统用户可能会投入更多的传感器节点，网络规模甚至达到成千上万。这些情况都会使

网络规模和节点密度发生很大的变化，因此传输层协议必须具有较强的规模可扩展性。

7）功能可扩展：传输层协议应该支持系统用户对网络进行进一步的性能优化并支持新的应用。



对于任何网络而言，传输层协议都应该完成两个主要任务：1）实现端到端的可靠数据传输（即不丢包），但不是像介质访问控制层协议那样的逐跳可靠数据传输。不过，可以通过逐跳确认的方式来保证端到端传输的可靠性。本章中介绍的许多传感器网络传输层协议都是通过逐跳数据分组丢失重传的方式来实现端到端可靠数据传输的目的。2）解决网络拥塞问题，包括检测可能发生拥塞的位置以及如何避免拥塞事件的发生。一般情况下，只有同时实现了上述两个任务的传输层协议才能称为完全传输层协议。但是由于具体应用要求以及由于效率的考虑，有些传输层协议仅仅侧重于实现上述两个任务中的一个（可靠性或拥塞问题），这也是允许的。

5.2 PSFQ

5.2.1 为什么TCP协议不适用于传感器网络

与互联网一样，无线传感器网络同样需要采用适当的传输层协议以满足相应的数据传输要求：

1）**可靠的端到端数据传输**：数据应该以无损或几乎无损的方式进行端到端（传感器节点与汇聚节点之间）数据传输。

151
|
152

传感器网络中的网络流量包括大量的**传感器节点发送至汇聚节点的感知数据和少量的汇聚节点发送至全网或部分传感器节点的感知数据查询或控制命令。**

传感器节点发送至汇聚节点的感知数据包括突发事件信息和周期性一般监测信息两类。由于新检测到的突发事件是所监测对象的重大变化，对于监测系统而言非常重要，因此该信息需要绝对可靠的发送至汇聚节点。也就是说，这类数据的传输不允许出现分组丢失的情况。而对于周期性的一般性监测数据而言，由于在时间和空间上存在一定的相关冗余性，因此可以允许出现一定程度的数据分组出错/丢失情况。例如，对于环境温度监测和动物跟踪的传感器网络应用而言，少量感知数据的丢失对于监测任务基本没有影响。因此，传感器网络传输层协议不应为一味追求绝对传输可靠性而设计得过于复杂，这样会造成协议健壮性和能耗效率的下降，只需要满足实际应用的可靠性要求即可。

然而，由汇聚节点发送至传感器节点的查询或控制命令则需要进行绝对可靠的传输。为了使传感器网络能够在运行过程中根据实际监测需要而不间断地动态更新自身监测任务，文献[Chieh-Yih05]提出了由汇聚节点把新任务的程序映像分发给网络传感器节点的无线空中下载技术。在分发程序映像的整个过程中，包含程序映像的所有数据分组都必须可靠地传送到每个节点，否则就会导致监测任务升级更新的失败。

2）**拥塞检测和避免**：当无线传感器网络中多个传感器节点同时发送感知数据时，局部区域会出现网络拥塞的现象。如何及时发现可能出现拥塞的传感器节点并采取有效措施避免拥塞的发生是传输层协议的主要任务之一。

作为最具代表性的可靠传输层协议，TCP协议已经在Internet上成功应用了几十年。在进

行数据传输之前，TCP 协议栈首先使用三次握手机制建立一条端到端的全双工数据通道，然后通过基于滑动窗口的数据流控制机制调节发送速率。当检测到传输超时或者三重确认报文（ACK）时，就认为数据包丢失并进行重传。

TCP 协议使用 20 字节长的报文头来表示拥塞控制和其他相关信息。报文头会增加数据报文的长度，特别是对于本身较短的数据报文而言，由于报文头相对于数据本身所占比例过大，因此会消耗过多的网络带宽资源。在无线传感器网络中，感知数据是由少量监测指标的实际数值组成，一般数据分组的长度只有几个字节。这种情况下 TCP 协议 20 字节的报文头开销就会显得过大。

TCP 协议的设计目的是简化接收方（对于无线传感器网络而言，大部分情况下接收方是汇聚节点）的操作。汇聚节点仅需要向发送节点确认接收情况（如果接收到数据就发送 ACK 消息，否则不发送任何消息）。发送节点则需要执行一系列的复杂流控操作以保证可靠数据传输。然而，由于无线传感器网络中传感器节点（发送方）资源非常受限，而汇聚节点（接收方）的能量、计算和存储资源则可以认为完全不受限，TCP 协议对发送方的过高操作负荷要求显然不适用于无线传感器网络。所以，对于无线传感器网络而言，理想的传输层协议应该是由汇聚
153 节点完成保证可靠数据传输的各种复杂操作。

另外，TCP 协议力求在数据传输过程中不丢失一个数据分组。而本章前面已经提到，很多无线传感器网络应用并不需要完全可靠的、无差错的数据传输。



在 Internet 中，TCP 协议总是力求实现完全可靠传输，即没有数据分组丢失（其中，数据分组出错可以认为数据分组丢失，因为接收方会丢弃出错的数据分组）。在无线传感器网络中，对于上行数据传输而言，由于上行方向（传感器节点→汇聚节点）的感知数据存在一定的相关冗余性，数据传输只需部分可靠性即可；而对于下行数据传输（汇聚节点→传感器节点）而言，由于汇聚节点总是分发重要数据（比如查询或者控制命令），必须实现完全可靠传输。

本节主要介绍传输层协议的可靠性，而关于网络拥塞的相关问题则会在后面进行讨论。考虑一个问题：如何设计一个适用于无线传感器网络的传输层协议，以保证可靠数据传输？为了能够运行在低端传感器节点（如加州大学伯克利分校研制的 Mote 系列节点）上，传输层协议应该具备较低的复杂度和较高的能量效率；为了使得各种监测应用不受传感器网络自身的节点和通信不可靠性的影响，传输层协议应该能够充分利用网络中感知信息的相关性以及节点的高连通性，实现高效率、健壮的数据传输。

针对这种情况，文献［Chieh-Yih05］提出了 PSFQ（Pump Slowly, Fetch Quickly，慢存快取）协议，它能够把用户数据可靠地、低能耗地传输到目的传感器节点，具有复杂度低、健壮性强、扩展性好的特点。

PSFQ 协议代表一种简单方法，具有最低路由基础设施要求（不同于完成类似任务的 IP 组播路由要求）、最低程度信令（信令指传感器节点间交换的协议消息），因此降低了数据可靠性传输的通信开销。PSFQ 协议对恶劣通信环境下高误码率反应迅速，在高误码率的条件下仍然能够成功进行可靠数据传输。

5.2.2　基本工作原理

如何使数据分组出错/丢失降至最少？PSFQ 协议采用了非常有趣、简单的设计思想：一方面，以相对较慢的数据速率分发源节点的数据（称为“慢存”，即 pump slowly）。这是因为分

发数据过快会导致无线信道数据丢失率急剧增加；另一方面，正在经历数据丢失的节点则迅速地主动向其上游邻居节点索取（恢复）已丢失的数据片段，进行本地恢复（称为“快取”，即 fetch quickly）。值得注意的是，通过直接相邻节点间的快速本地数据恢复对于最大程度地降低 [154]
丢失恢复开销十分重要。否则就要采用传统端到端错误恢复机制请求源节点重新发送数据，对于多跳、不可靠链路而言，这会带来大量的恢复开销甚至会加剧数据丢失。因此在传感器网络中，采用源节点重发丢失数据的方式进行数据恢复是不可取的。

1）逐跳（本地）错误恢复：在传统端到端错误恢复机制中，只有最终的目的节点负责检测丢失和请求重传。为什么端到端的错误恢复机制不适用于无线传感器网络？因为无线传感器网络通常在恶劣的无线通信环境下工作，而一般情况下事件监测区域距离汇聚节点过远，需要通过多跳转发技术交换消息。

在传感器网络数据传输过程中，每一跳中都会有一定数量的数据分组被丢弃（由于部分数据分组在经由无线信道传输时产生误码而被转发节点丢弃）。例如，为了简单说明这个问题，假定无线信道一跳的数据分组丢包率为 p（$0 < p < 1$），经过 n 个转发跳成功交换一条消息的概率为 $(1-p)^n$。误码随着转发跳数的增加而呈指数递增，因此很可能发生数据分组丢失和重传的情况。经过多跳转发后，最终目的节点只能接收到很少量的完整数据分组。

日常生活中也有很多类似的例子。如果某大学生有一门课考试不及格，他（她）可以通过及时重修来保证 4 年后按时毕业。但是如果有十门课考试不及格，那么他（她）可能需要 5 年以上的时间来完成本科学习（其中包括不及格课程的重修），将不能按时参加毕业典礼。

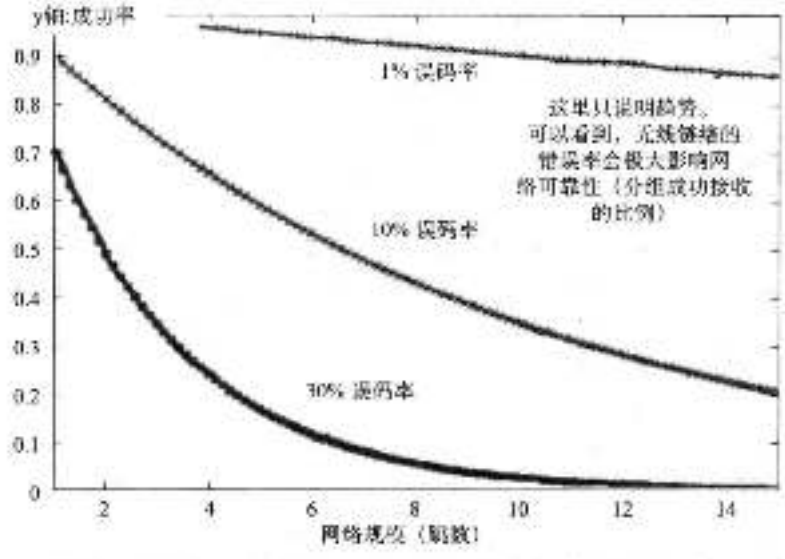


图 5-1　使用端到端恢复在多跳网络中成功交付一条数据分组的概率 [Chieh-Yih05]

文献 [Chieh-Yih05] 用数据说明了这个问题，如图 5-1 所示。由该图可以看出成功率和网 [155]
络规模的变化曲线。对于较大规模无线传感器网络（转发跳数 > 14），当链路误码率高于 10% 时，在有损链路环境中使用端到端错误恢复机制几乎不能交付一条数据分组，这是因为大部分数据分组在经过多次转发过程中由于发生误码而被丢弃了。对于 14 跳的通信路径而言，即使采用端到端恢复机制，仍然有 80% 的数据分组不能被有效恢复。另外，由图 5-1 中三种误码率情况下的传输成功率曲线可以看出，无线链路误码率因素会极大地影响网络可靠性（数据传输成功率）。



“雪球”效应：无线传感器网络由低成本、资源受限的传感器节点组成，无线链路的误码率一般较高。如果数据分组的丢弃问题不能在发生误码的链路上进行恢复解决，那么在下一跳链路上因数据分组误码而丢弃的问题会更加严重。而对于传统 Internet 而言，其骨干网采用可靠性非常高的光纤作为通信介质，因此不会出现无线传感器网络数据传输过程中的分组误码丢弃的累积问题。

文献［JZhao03］指出，在密集无线传感器网络中经常遇到 10% 或者更高的链路误码率。在很多情况下，比如军事应用、工业过程监测、灾后重建，误码率更高。这些事实说明端到端误码恢复机制并不适合无线传感器网络用于实现可靠数据传输。

PSFQ 协议采用逐跳错误恢复机制，中间节点也负责丢失检测和恢复丢失的数据片段，在逐跳基础上进行可靠数据交换。逐跳错误恢复实质上是将多跳转发操作分割成一系列单跳发送过程，消除误码累积。因此，与端到端误码恢复机制相比，逐跳错误恢复机制可扩展性更好，抗误码能力更强，同时降低了数据分片重新排序的情况。

2）**多次重传**：无线传感器网络通过重传的方式对发生误码的数据分组进行恢复，并且很多情况下上游节点会对同一误码数据分组进行多次重传以提高传输成功率。相应地，数据分组的链路传输时延依赖于为成功接收而进行重传的次数。

156 接收节点（包括中间节点和目的节点）使用队列（即存储缓冲区）保存所有接收失败的数据分组信息。它只有成功接收到重传的某个数据分组后，才会清除队列里该数据分组的相关信息。为了降低传输时延，必须使单个“可控时间帧”内成功传输一个数据分组的概率达到最大。

一种直观方法是：在下一个分组 $i+1$ 传递到达之前尽可能多次重传数据分组 i（这样可以提高成功传输的概率），称为“快取”。也就是说，在新数据分组传递到达之前清除接收节点（比如中间传感器节点）的队列，以便使队列尽可能短，从而降低传输时延。

文献［Chieh-Yih05］研究了如何确定最佳重传次数，以保证在获得较高传输成功率（即在一个时间帧内成功传输数据分组的概率）的同时不能在重传上浪费过多的能量。通过严格的数学模型进行分析后，发现了采用不同重传策略时数据分组传输成功率与无线链路分组丢失率的关系。单跳成功传输一条消息的概率与链路分组丢失率的变化曲线如图 5-2 所示：信道误码小于 60% 时，数据分组传输成功率有明显提高。但是，进一步增加重传次数带来的成功率改善程度会迅速下降，当重传次数大于 5 时，带来的成功率改善可忽略不计。这就意味着 PSFQ 协议分发操作定时器与提取操作定时器的最佳比率约等于 5。

3）**快速恢复**：如果不在发生误码的链路上对丢失的数据分组进行及时恢复，那么本地丢失事件会传播给下游节点，在下一跳链路上的数据分组误码丢失问题会更加严重。在 PSFQ 协议中，传感器节点根据所接收数据分组序列号（数据分组头部有一个序列号字段 SeqID）的连续情况判断是否有数据分组丢失情况发生。例如，如果某下游节点先后接收到 SeqID 分别为 3 和 5 的数据分组，它就会判断出 SeqID 为 4 的数据分组已丢失。

作为一个具体例子，考虑以下情况：节点 1 在向节点 2 发送数据过程中数据分组（SeqID = 99）由于误码而丢失，但节点 1 并没有及时重传该数据分组以恢复丢失数据，而仅仅期望其他下游节点能够参与完成节点 2 的数据恢复工作。在接到节点 2 的数据重传请求后，节点 1 的所有下游节点会启动数据恢复重传过程。但是，这些下游节点却没有该数据分组。在这种情况下，经过下游节点协商，某个下游节点（假设为节点 12）会向节点 1 发送数据重传请求，请

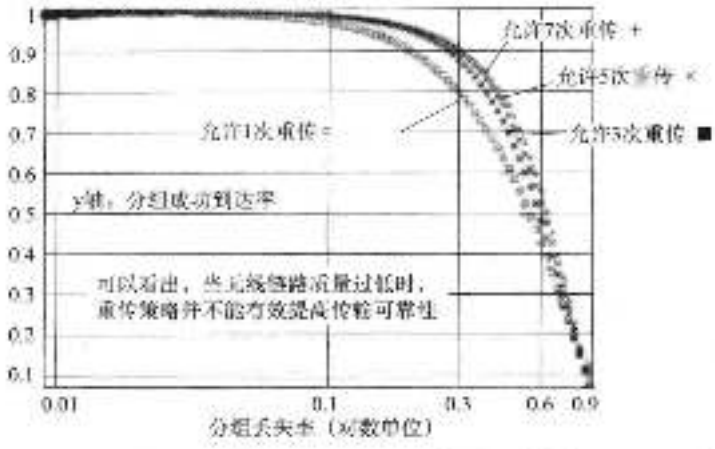


图 5-2 采用多次重传策略时的单跳数据传输成功率 [Chieh-Yih05]

求重新发送 SeqID 为 99 的数据分组。最终，节点 1 还是要重传数据分组。这样不但增加了额外提取操作的控制消息开销，还增加了数据传输延迟。因此，最好的本地数据重传恢复策略是节点 1 在接收到节点 2 的 NACK（negative acknowledgement，否定确认）消息后立即对 SeqID 为 99 数据分组进行重传操作。

可见，通过及时快速的本地重传恢复，能够避免丢失事件的传播扩散以及下游节点的不必要的提取操作。因此，PSFQ 协议规定：中间节点只转发具有连续序列号的数据分组。为了保证数据分组的有序转发以及对下游节点提取请求的及时响应，传感器节点使用数据缓存区来保存接收到的数据分组。由于 PSFQ 协议的主要应用目的为重新分配任务，数据缓存区的大小由更新任务的代码段长度来决定。



奇思妙想

保证数据分组的有序传输是大部分传输层协议的主要工作之一。例如，Internet 的 TCP 协议就是通过基于滑动窗口的数据分组发送机制实现数据分组的可靠有序传输。在滑动窗口机制中，每个数据分组都被分配一个唯一的序号。只有窗口内最小序号的数据分组成功被目的主机接收后，才允许更高序号的数据分组进行发送。有序传输能够简化传输层协议的操作。相反，如果采用无序传输的方式，为了保证可靠数据传输，传输层协议需对大量的不连续丢失数据分组序号信息进行动态地保存及更新，这将大大增加协议的复杂性。

5.2.3 协议描述

PSFQ 协议由三个协议功能组成：

- **消息转发（转发操作，pump operation）**：源节点（感知事件发生区域内的传感器节点或者汇聚节点）将消息发送到网络中，中间节点缓存消息，并按照适当的策略转发消息，以满足宽松的传输延时要求。
- **转发过程的错误恢复（提取操作，fetch operation）**：中间转发节点维护一个数据缓存区，利用所缓存的信息检测数据丢失情况（根据接收数据分组的序号连续性加以判断），在必要时通过发送 ACK 或者 NACK 消息触发相关节点的错误恢复操作。
- **选择性状态报告（报告操作，report operation）**：源节点（即数据发送节点）获取有关

分发状态统计数据，并将其作为随后决策（比如在任务重新分配空中下载时调整数据分发的频率）。因此，反馈和报告机制对于 PSFQ 协议而言非常重要，而且报告机制必须具备自适应性（即报告时机要根据实际通信环境进行动态调整）和可扩展性（即开销要尽可能低）。

接下来，针对重新分配任务应用介绍 PSFQ 协议的主要操作（如分发、提取、报告）。在重新分配任务应用中，用户必须将控制命令或二进制代码段分发到目标传感器节点，重新对传感器节点子集进行任务分配。


奇思妙想

PSFQ（慢存快取）协议的基本思想不难理解。由于无线传感器网络的传输码率非常高，在数据传输过程中中间节点需要花费较多的时间对因发生误码而丢失的数据分组进行恢复操作，因此源节点不能过快地向无线传感器网络发送大量数据分组。与此类似，为了保证公路畅通，必须限制在行驶缓慢的单车道上通行的汽车数量。另一方面，如果发生数据分组丢失现象，必须及时对其进行恢复，否则随后的数据分组会因“雪球效应”导致误码不断累积恶化。仍用上面的例子，对于在单行道上通行的汽车而言，一旦前面有汽车发生事故，必须立即对事故汽车进行清理，以保证后面拥塞的汽车能够顺利通行。

1. 分发操作

虽然 PSFQ 协议在每跳转发过程中采用了错误恢复机制，但它不是路由协议而是传输协议。PSFQ 协议本身不需要进行路径发现，而是工作在现有路由协议上面，在中间节点上创建和维护一个数据缓存区，以进行逐跳本地丢失恢复和有序数据交付，从而实现可靠数据传输。

本节主要介绍分发操作。源节点通过分发操作向网络中的目标节点缓慢地分发数据，这能够从一定程度上避免网络拥塞的发生，而网络拥塞是传输层协议主要解决的问题。

分发操作对于控制代码段及时分发给所有目的节点、提供基本流量控制、使重新分配任务操作不影响无线传感器网络现有任务的正常运转非常重要。这就要求合理安排数据转发时机，采用一个简单的数据分组转发调度方案，该方案采用两个分发定时器 T_{min} 和 T_{max}。基本的分发操作如下：

源节点按周期时间 T_{min} 定期向其下游节点发送数据分组。下游节点在接收到数据分组后，根据其序号检查本地数据缓存区，如果是重复分组，则将其丢弃。如果是一个新分组，则将其存入数据缓存区，同时将其 TTL 减 1；假如 TTL 不等于零并且序号也不连续，那么说明发生误码分组丢失，节点转而执行提取操作进行错误恢复（将在下一小节介绍）；假如 TTL 不等于零并且序号连续，那么设置转发该分组的时间，继续对其向目的节点执行分发操作。

157
~
159

中间节点延迟一段随机时间（位于 T_{min} 和 T_{max} 之间）后，将接收到的分组转发给其下游邻居节点。在重新分配任务应用中，PSFQ 协议只是重新广播。当相互干扰节点都需要重新转发并且设置转发时间高度相关时，不适合使用 RTS/CTS 机制，因此转发前的随机时延对于避免传输冲突是必需的。

分发定时器 T_{min} 和 T_{max} 主要具有以下作用：

T_{max} 是一个重要的参数。第一，对于本地丢失恢复，需要提供时间缓冲区。PSFQ 协议的主要动机之一是在可控时间框内从直接邻居节点迅速恢复丢失分组。在这种目的下，在上游节点传送下一个数据分组之前，按照 PSFQ 协议分发操作，节点必须至少等待 T_{min} 后才会转发分组，

因此有机会恢复丢失分组。第二，必须减少冗余广播。在密集传感器网络中，常常在无线覆盖范围内存在多个直接邻居节点。利用 T_{wait} 后，节点在实际开始转发消息之前有机会监听其他转发节点传输的同一条消息。如果在计划转发时间之前接收到一定数量的相同消息，则取消转发，从而避免无意义的重复转发。

T_{wait} 用来提供最后一个中间转发节点成功接收到整个文件（即控制命令或者可执行二进制代码）的最后一个数据片段的宽松统计时延限度。假设主动提取操作（下节介绍）在一个 T_{wait} 间隔内恢复任意丢失数据分组，那么时延范围（用 $D(n)$ 表示）和 T_{wait} 之间的关系如下：

$$D(n) = T_{wait} \times n \times \text{转发跳数}$$

其中 n 表示一个文件的分片数量。

2. 提取操作

节点一旦检测到文件片段中出现序列号不连续的情况，则立即进入提取模式。提取操作是主动操作，一旦检测到数据丢失就请求上游节点重传数据。

PSFQ 采用“丢失累积”的思想，即 PSFQ 对所有丢失的数据采取批处理，尽量在一次提取操作中恢复所有丢失消息。

（1）丢失累积

研究者发现，由于无线信号衰减有很强的相关性，因此无线环境中的数据丢失是“突发”的。也就是说，如果无线链路不稳定，那么这样的恶劣通信条件会持续一段时间并且损坏一批数据。所以数据的丢失通常是成批的（也被称为突发丢失）。PSFQ 会累积丢失，用提取操作处理各个丢失分组的“窗口”，而不是处理单个分组丢失。

由于突发丢失，节点收到的分组中可能存在多处序列号不连续的情况，提取操作中累积多个丢失窗口可提高成功恢复的概率。

（2）提取定时器

在分发操作中使用了分发定时器，在提取方式中，同样也需要一个定时器。通常当一个节点发现分组丢失后（通过检查序列号的连续情况），会向其上游节点发送 NACK 消息，请求重传丢失的数据片段。 160

假如节点在提取定时器 T_r（$T_r < T_{wait}$）内没有收到上游节点重传的数据或者只恢复了部分丢失的数据片段，那么该节点以 T_r 为周期（稍微随机化，避免与相邻节点同步）重发 NACK，直到所有丢失的数据片段恢复为止或者重传次数超过预定阈值而结束提取操作。

PSFQ 安排发送的第一个 NACK 的随机延时很短，NACK 介于 0 到 Δ 之间（$\Delta \ll T_r$）。如果在发送 NACK 前监听到相邻节点发送 NACK 请求恢复相同的丢失数据片段，则取消第一个 NACK 的发送。因为 Δ 很小，所以发生这种情况的概率相对较小。通常，响应其他节点发送的 NACK 的重传不能保证被取消其第一个 NACK 的节点监听到。

为了避免网络拥塞，不能转发 NACK 消息。也就是说，上游节点收到 NACK（来自下游节点）后不会将其继续向上游转发。当然这也有例外，如果节点收到同一个 NACK 的次数超过了预定的阈值，同时这些相同 NACK 请求的分片已经不在本地的缓存区内，那么 NACK 将只能被转发一次，将 NACK 的范围拓宽一些，以提高错误恢复的概率。

（3）主动提取

我们注意到，在前述的提取操作中有一个“白区”。提取操作是一种反应式的丢失恢复机制，只有当接收到序列号更大的分组时才检测到丢失。在这之前不能要求接收者能够检测出丢失。那么，如果丢失了一个文件的最后一个分片，接收节点就无法检测出最后一个分片已经丢失，因为不会再发送序号更大的分片。又如，如果文件比较小，由于突发丢失，可能导致后续

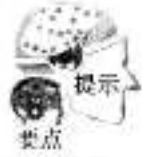
直到最后分片全部丢失。在这种情况下，采用前述的反应式丢失检测是无法检测分片丢失的。

为了解决上面提到的问题，PSFQ 采用基于定时器的主动式提取操作，即假设如果没有收到最后一个分片并且经过时间 T_{pr} 后不再有新的分组交付，那么节点进行主动式提取操作，发送 NACK 请求下一个分片或者剩余分片。

那么该如何选择 T_{pr}？显然，如果提取方式触发过早，就会浪费额外的控制消息，因为上游节点可能在转发上一条消息或者还没有收到新的分片；如果触发太迟，那么节点要浪费太多时间等待文件的最后一个分片，导致文件总的交付时延增大。因此设置合适的 T_{pr} 对应该权衡这个因素。

161

PSFQ 选择合适的 T_{pr} 的方法是：T_{pr} 应该正比于文件最大序列号 S_{max} 与最近所收到分组的文件最大序列号 S_{last} 之差（差值即为文件剩余的分片数量），即 $T_{pr}=\alpha(S_{max}-S_{last})T_{max}$（$\alpha\geq 1$），$\alpha$ 是一个扩展因子，用于调整触发主动提取机制的时延，在大多数情况下将 α 设为 1。这样，T_{pr} 保证了在接近文件末尾时，节点可以提前启动主动式提取操作，同时保证在分片未传输完时等待较长时间。



设计一个网络协议不像写 C 语言代码那样容易，需要考虑很多很多的细节。比如上面介绍的定时器的概念就是一个难以控制的问题。因为我们不能使定时器过早或过晚超时。

(4) 基于信号强度的提取

在传感器网络中，无线链路的不稳定性会导致转发过程中数据分组的突发性丢失。断断续续地接收来自多跳远距离节点的数据分组可能会引起节点发送不必要的 NACK 和进行多余的重传，因此 PSFQ 在提取操作时还考虑到了分组接收的信号强度。一个节点检测到所收到分组序列号不连续时，如果该分组是由一个平均信号质量最好的节点发送的，那么只需要响应一个 NACK，由平均质量最好的节点重传。这就有效抑制了因接收到多跳远节点的分组而触发不必要的 NACK。

3. 报告操作

报告机制用于以简单且可扩展的方式将数据交付状态反馈给用户。当节点收到一个数据分组，并且该数据消息头的“报告位”被置 1 时，进入报告方式。

到达源节点的路径上的每个节点将自己的状态信息添加到报告消息中，然后将累积的报告发送到用户节点；若报告消息中已包含自己的 ID，那么忽略这条报告消息，以免出现回环。

162

在网络的规模非常大的情况下，有可能节点收到的报告消息中没有多余的空间添加自身的状态信息。为了解决这个问题，该节点产生一条新的报告消息，先发送新产生的报告消息，再转发收到的报告消息。这样做能保证路径上的节点首先使用有空间的报告消息而不是产生新的报告消息，这是因为这些节点也会填满信息的报告消息。

5.3　ESRT：事件到汇聚节点的可靠传输协议

在某些无线传感器网络系统中，系统对于事件的监测需要多个节点对感知对象状态的联合可靠检测，而不是它们中各节点的单独报告。ESRT（Event-to-Sink Reliable Transport Protocol，事件到汇聚节点的可靠传输协议）以事件到汇聚节点（即基站）传输的可靠性为设计目标，并不需要保证所有感知数据包的端到端可靠传输服务，而是寻求以最低能耗和无拥塞方法实现

可靠事件检测［Akan05］。

ESRT协议适用于典型的无线传感器网络应用，包括事件检测、信号估计/跟踪，但它并不是用于保证端到端数据传输服务的。事件到汇聚节点可靠性的概念将ESRT与现有的以端到端可靠性为重点的传输层模型区分开来。比如，上一节介绍的PSFQ更加适用于汇聚节点到事件（Sink-to-Event）的可靠性控制。



前面介绍了传感器网络中不同的方向（上行指从传感器节点到汇聚节点；下行指从汇聚节点到传感器节点）。这两种方向有不同的可靠性需求和通信特征。例如，ESRT只关注一个方向——上行。后面将介绍下行传输的可靠机制（即GARUDA，见5.7节）。

5.3.1 可靠传输问题

文献［Akan05］对无线传感器网络中的可靠传输问题做了正式的定义。许多无线传感器网络应用需要根据事件区域内的若干个传感器节点对事件的聚合报告来进行事件的可靠检测或者对事件特征进行估计。假设为了进行可靠的时间跟踪，汇聚节点必须每隔τ个时间单位进行事件特征的确认。τ表示一个决定间隔的持续时间，根据不同的实际应用来确定τ值。在每个决定间隔结束时，汇聚节点根据在此期间接收到的传感器节点的报告作出决定。

假设汇聚节点在决定间隔τ结束时得到事件可靠性指示器（event reliability indicator）r_i。需要注意，对r_i的计算只能采用汇聚节点本地的有效参数。

ESRT按照所接收数据分组数量衡量事件特征从源节点到汇聚节点的可靠传输。无论应用层的特定参数如何，所接收数据分组数量与汇聚节点检测和提取事件特征所需要的信息量密切相关。因此，可以将接收数据分组数量作为传输层一个简单而又精确的事件可靠性指标。对所观测到的和所需要的可靠性定义如下：

定义5.1：观测事件可靠性（Observed Event Reliability）r_i等于汇聚节点在决定间隔i期间所收到的数据分组数目。

定义5.2：所需事件可靠性（Desired Event Reliability）R等于事件可靠检测所需要的数据分组数量，其值通常由应用决定。

要求观测事件可靠性r_i高于所需事件可靠性R。这样，才能认为该事件能够被可靠检测。否则，应该采取适当措施使r_i达到所需事件可靠性R。

在无线传感器网络中，ESRT为传感器节点观测到的不同类型的事件分配不同的ID。节点会将观测到事件的ID发送到汇聚节点，汇聚节点接收到一个事件ID分组时就将其相应事件ID的接收分组计数器加1，进而通过计数器数值的大小计算出观测事件的可靠性r_i。汇聚节点并不关心是由哪个节点发出的数据。

从统计学的角度考虑，节点可以增加事件信息报告速率，使得汇聚节点对可靠性的计算更加准确。因此ESRT定义了节点的报告速率f。

定义5.3：一个节点的报告速率f等于该节点在单位时间发送的分组数量。

定义5.4：无线传感器网络中的传输层问题（从可靠性考虑，而不是从拥塞控制考虑）是：配置源节点报告速率f，以最低资源利用率使得汇聚节点实现所需事件检测可靠性R的问题。

上述事件到汇聚节点可靠性概念的基本原理是：相邻区域内的传感器节点对于同一事件产

163

生的报告数据存在冗余性，在一定程度上容忍单个分组的丢失，即汇聚节点估计事件特征的失真度 D_i 不会大于一定的阈值 D_{max}。报告速率跟采样速率、量化等级参数、感知特征数量等有关。因此，报告速率 f 控制注入到传感器场中的流量大小，同时调整物理现象相关采样值的数量。这反过来又会影响所观测事件的失真，即事件检测可靠性。

5.3.2　归一化事件可靠性与报告速率之间的关系

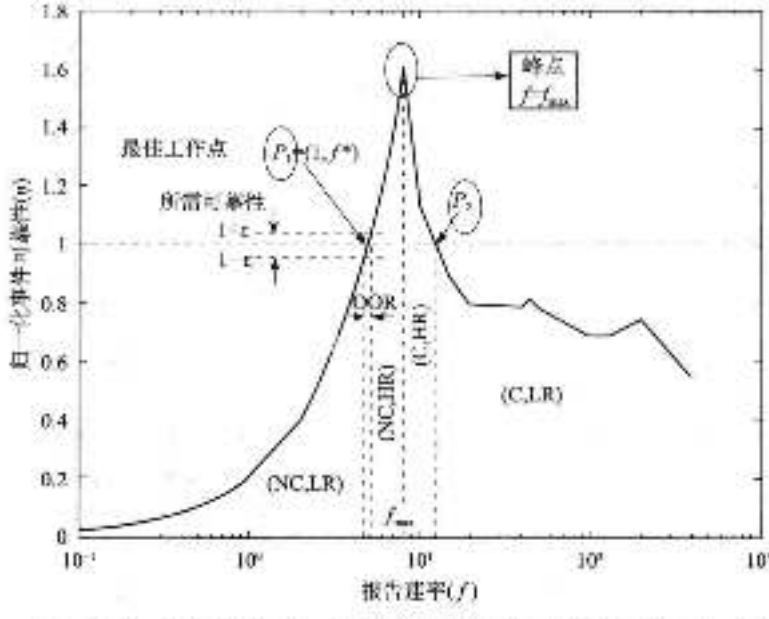
为了研究汇聚节点观测事件可靠性 r 和传感器节点报告速率 f 之间的关系，文献[Akan05]采用 NS-2 模拟工具构造了一个无线传感器网络，200 个节点被随机安放在 100 × 100 的区域内。假设已经随机安放好的节点拓扑不再变化。

不同应用的所需事件可靠性 R 是不同的，文献[Akan05]用一个合适的参数来测量事件
164 可靠性，即 $\eta = r/R$。其中，η_i 表示在每个决定间隔 i 结束时的归一化事件可靠性。

使用归一化可靠性 η 要优于观测事件可靠性 r，因为 η 反映了 r 在所需事件可靠性 R 中的权重（重要性）。我们的目标是达到 $\eta = 1$ 的系统状态。应当注意，η 可以大于 1，即实际的可靠性大于所需可靠性。这一点也许看起来是有“吸引力”的，但是这并不是需要的，因为高可靠性会消耗更多能量并且在网络中累积了过多的数据（可能造成拥塞）。

令人关注的是，文献[Akan05]中模拟的结果显示，从某些特性区域中能够了解到 η 与 f 的关系，f 处于不同的范围，则 η 的趋势也不同。

系统的目标是尽可能地在 $\eta = 1$ 的状态下工作。假设当 $f = f^*$ 时，有 $\eta = 1$，那么称 f^* 为最佳工作点（Optimal Operating Point，OOP），如图 5-3 中的 P_1 点。



5-3　在归一化事件可靠性 η 与报告速率 f 下的 5 个特性区域[Akan05]

如图 5-3 所示，可以看到，直线 $\eta = 1$ 与事件可靠性曲线相交于点 P_1 和 P_2。似乎 P_1 和 P_2 都是最佳工作点。但是，在点 P_2 虽然能够可靠检测事件，但由于报告速率 f 已经超过了峰值点
165 f_{max}，网络发生拥塞，一些源节点发送的数据分组丢失。因此，不能称 P_2 为最佳工作点。


奇思妙想

这是一个很好的研究方法！通常，研究者这样做研究：首先，他们明确一些有挑战性但未解决的问题。然后，他们设法用理论模型得到一些定量的结果。这些数学分析的结果是非常重要的，因为实际工程中的技术设计都是基于某些理论的。接下来，他们用软件模拟或者硬件试验验证数学分析的正确性。但是，在这里，ESRT 采用了不同的研究策略：它通过模拟找出有意义的 5 个区域的可靠性与速率的关系！然后研究者进行理论建模和算法设计。

图 5-3 中定义了一个宽度为 2ε 的公差带（tolerance zone），这里的 ε 是一个协议参数。采用以下判别条件来确定图 5-3 中的 5 个由虚线分割的特性区域（η 为归一化事件可靠性）：

区域 1：（NC，LR），即无拥塞，低可靠性

$$f < f_{max}, \eta < 1 - \varepsilon$$

由于可靠性低，该区域不够理想。

区域 2：（NC，HR），即无拥塞，高可靠性

$$f \leqslant f_{max}, \eta > 1 + \varepsilon$$

由于可靠性较高并且没有发生拥塞（因为其报告速率并不高，即 $f < f_{max}$），该区域较为理想。

区域 3：（OOR），即最佳工作区（optimal operating region）

$$f < f_{max}, 1 - \varepsilon \leqslant \eta \leqslant 1 + \varepsilon$$

这是最佳工作区，其他所有区域都应该通过改变 f 来接近该区域。

区域 4：（C，HR），即拥塞，高可靠性

$$f > f_{max}, \eta > 1$$

由于网络拥塞（$f > f_{max}$），该区域较差，唯一的优点是仍然保持满意的可靠性。

区域 5：（C，LR），拥塞，低可靠性

$$f > f_{max}, \eta \leqslant 1$$

166

由于该区域内网络拥塞并且可靠性低，因此该区域为最差区域。

正如上文分析的，在判断系统正处于哪个特性区域的时候，需要知道两个时变的参数（报告速率 f 和归一化事件可靠性 η）以及两个固定参数（峰值速率 f_{max} 和容差区域 ε）。

设 S_i 表示在决定间隔 i 结束时的网络状态，有

$$S_i \in \{(NC, LR), (NC, HR), (C, HR), (C, LR), OOR\}$$

可以看出，上述 5 个状态取决于当前事件可靠性和网络是否拥塞。因此，在实际网络实现中，ESRT 从两个方面定义当前状态 S_i：①汇聚节点在每个决定间隔 i 中计算出的归一化可靠性 η_i；②拥塞检测机制。

在 ESRT 中，汇聚节点可以根据当前状态 S_i 以及 f_i 和 η_i 的值，ESRT 可以计算出报告速率更新 f_{i+1}，然后将 f_{i+1} 广播给源节点。在下一个决定间隔结束时，汇聚节点就可以得到与 f_{i+1} 对应的归一化事件可靠性 η_{i+1}。再与网络拥塞报告综合，ESRT 就可以决定新的网络状态 S_{i+1}。不断重复上述过程，直到网络转移到 OOR 状态。下面将介绍该算法，图 5-4 为基本的状态转移规则。

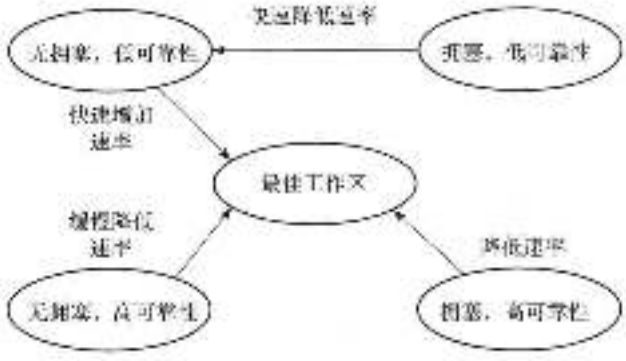


图 5-4　ESRT 协议的状态及转换［Akan05］


奇思妙想

有限状态机（FSM）——这是一个解决一些系统控制问题的基本研究方法。即使我们可以用更加先进、复杂的控制模型或者数学算法控制一个系统，但最终我们都需要用 FSM 定义所有的系统状态以及状态间相应的切换动作。事实上，所有的网络协议都是基于 FSM 模型设计出来的。考虑一个有趣的问题：你怎么用 FSM 模型定义人类？也许你会说一个人可以有“睡觉”、“吃”、“学习”、“爱”、“生病”等状态。你还可以定义状态间切换的条件和动作。例如，为了进入“吃”的状态，我们至少需要一个条件，称为“饿”，相应的动作是“张开嘴并咀嚼食物”。

状态转移算法包括以下 5 个方面：

1.（NC，LR）状态

在（NC，LR）状态，网络不会发生拥塞。但是可靠性低于所需可靠性。从图 5-3 中可以看出 $f < f_{ss}$，$\eta < 1 - \varepsilon$。进入该状态的原因可能是①中间路由节点失效或关电；②链路错误造成分组丢失；③源节点没有发送足够多的信息。

对于原因①，需要通过这些中间节点的分组被丢弃。即使源节点发送了再多的信息，也会导致可靠性下降。不过，一些算法提供了无线传感器网络中容错路由或者重路由［Cintanagonwiwat00］，ESRT 可以和这些机制协同工作。

在网络中，由于采用差错码纠错能力技术和重传技术的能效很低，所以链路错误造成的分组丢失可能十分严重。然而，无论分组丢失率如何，链路错误造成的分组丢失总数与报告速率 f 成正比。因此，假设在连续决定间隔内，信道状况对分组丢失的影响效果不会出现明显偏差。在网络应用中，这个假设对于固定传感器节点、慢时变、空间隔离的事件到汇聚节点的通信信道是合理的。因此，即便存在链路错误造成的分组丢失，开始时可靠性仍然是线性递增的。

如果系统处于（NC，LR）状态下，汇聚节点需要通知源节点主动增加报告速率 f，以便尽可能达到所要求的可靠性。可以通过如下方式增大报告速率 f：在无拥塞情况下，对于 $f < f_{ss}$，r 与 f 的关系是线性关系（如图 5-3 所示）。因此可以使用如下乘性增加策略计算报告速率更新 f_{i+1}，即

167
168

$$f_{i+1} = \frac{f_i}{\eta_i}$$

其中，η_i 表示在决定间隔 i 结束时汇聚节点观测到的归一化事件可靠性。

2.（NC，HR）状态

在（NC，HR）状态下，网络未发生拥塞，但所达到的可靠性已经超过了所需可靠性，即 $\eta > 1 - \varepsilon$ 并且 $f \approx f_{max}$。这不是一个很坏的情况，因为没有发生拥塞，可靠性也达到了要求。但是由于源节点报告的速率高于所需的报告速率，浪费了传感器节点能量。因此，应该降低报告速率，以便节省能量。

但是，由于非常靠近 OOP 状态，为了维持事件到汇聚节点的可靠性，因此在降低报告速率时应该非常谨慎。汇聚节点采用一种可控方式降低报告速率 f：将斜率减半。那么，更新的报告速率计算公式如下：

$$f_{t+1} = \frac{f_t}{2}\left(1 + \frac{1}{\eta_t}\right)$$

这种更新策略既能降低网络能耗，又不会给事件可靠性带来负面影响。

3.（C，HR）状态

在（C，HR）状态下，网络发生拥塞，所达到的可靠性高于所需可靠性，即 $\eta > 1$ 且 $f > f_{max}$。这不是一个理想的状态。首先，我们不希望网络中出现拥塞状况；其次，可靠性也超过了所需的可靠性。在这种情况下，ESRT 通过降低报告速率 f 来避免拥塞发生，同时节省能量。与（NC，HR）状态一样，由于要注意维持事件到汇聚节点的可靠性，所以在降低报告速率时仍然应该非常谨慎。但是，网络在（C，HR）状态下工作时比在（NC，HR）状态下工作时会更加远离最佳工作点。因此，需要采取更加主动的方法减轻拥塞，同时尽可能地进入（NC，HR）状态。ESRT 用如下公式更新报告速率：

$$f_{t+1} = \frac{f_t}{\eta_t}$$

169

4.（C，LR）状态

在（C，LR）状态下，网络存在拥塞，观测可靠性不合格，即 $\eta \leq 1$ 且 $f > f_{max}$。这是所有可能中最坏的情况，因为同时存在可靠性低、网络拥塞和浪费能量的问题。因此，ESRT 协议主动降低报告速率，使网络尽可能快地进入到 OOR 状态。

为了确保充分降低报告速率，采用指数递减法，报告速率更新如下：

$$f_{t+1} = f_t^{(\frac{k}{k+1})}$$

式中 k 表示网络处在（C，LR）状态下的连续决定间隔个数，包括当前决定间隔，即 $k \geq 1$。如果没有检测到状态转移，系统能比较主动地降低报告速率 f。这个策略也能确保在（C，LR）状态下收敛到 $\eta = 1$。

5. OOR 状态

在 OOR 状态，网络在最佳工作点的 ε 容差区域内工作，在这个范围内以最低能耗得到所需可靠性。因此，在下一个决定间隔，报告速率不变：

$$f_{t+1} = f_t$$



奇思妙想

如果以较慢的速率接近一个点，那么可以用对数或线性速率。但是，如果要追求更快速率，乘性递增是最好的选择。当然，指数速率也足够快。

5.3.3 拥塞检测

虽然 ESRT 的主要目标是保证最佳可靠性，但它也会对网络拥塞产生影响。这可以从上述 5 个状态看出。另一方面，为了决定当前的状态，汇聚节点也需要检测网络的拥塞状况。那么汇聚节点如何检测拥塞？

170 由于不能采用 TCP 协议，那么不能用传统的方法确定网络拥塞程度。因此，ESRT 采用基于传感器节点本地缓冲区监测算法的拥塞检测机制。任何传感器节点由于大量输入分组而导致其路由缓冲区溢出，则可判断出现拥塞，并将这个信息通知汇聚节点。下面详细介绍这个拥塞检测机制。

在事件到汇聚节点的模型中，在每个报告周期 $1/f$ 内产生的流量取决于报告速率 f 和源节点数量 n。由于报告速率 f 在每个决定间隔 $\tau > 1/f$ 结束时受汇聚节点周期性的控制，所以 f 在一个报告周期内不会发生变化。假设 n 在一个报告周期内不会发生明显变化，在下一个报告周期内产生的流量变化可以忽略。因此，任一传感器节点在连续多个报告周期内的输入流量是恒定的。这样就可以说明，在每个报告周期结束时缓冲区填满程度的增加是恒定的。

设 b_i 和 b_{i-1} 分别表示第 k 个报告周期和第 $k-1$ 个报告周期结束时的缓冲区填满程度，b 表示缓冲区的容量。如图 5-5 所示。对于一个给定的传感器节点，设 Δb 表示在最近一个报告周期结束时观测到的缓冲区的长度增量，即

$$\Delta b = b_i - b_{i-1}$$

因此，如果在第 k 个报告周期结束时，当前缓冲区长度与最近的缓冲区长度增量之和大于缓冲区的容量，即 $\Delta b + b_{i-1} > B$，那么该传感器节点就可以检测下一个报告周期自己将进入拥塞状态。然后将其要发送的分组中的拥塞通知（Congestion Notification，CN）位置 1，通知汇聚节点在下一个报告周期将处于上行拥塞状况。

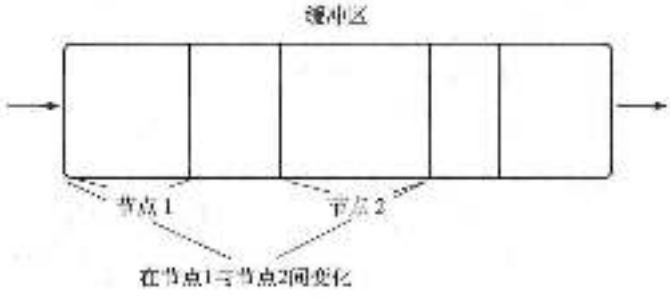


图 5-5 传感器节点中缓冲区监测图示［Akan05］


奇思妙想

检测节点本地缓冲区占用情况是检测网络拥塞程度的常用方法，TCP 协议也采用了该方法，但是该方法仅适用于源节点。

5.4 E²SRT：事件到汇聚节点的增强可靠传输协议

前一节介绍了 ESRT，虽然其算法能够使不同的状态趋向 OOR 状态，但在图 5-6 中，
171 文献［Sun08］的仿真实验结果显示，当协议的期望可靠性 R 被设置为超出当前网络支

持范围（比如网络中节点的部署策略、传感器的资源和网络规模）时，网络将不会进入OOR状态。

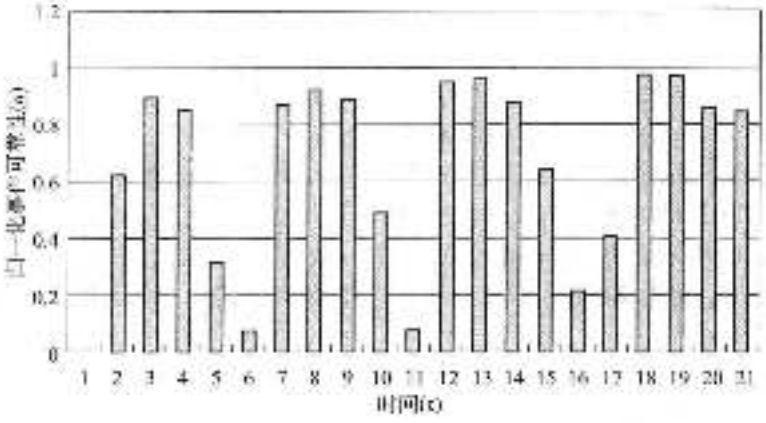


图5-6 可靠性需求过度情况下ESRT机制的归一化事件可靠性变化情况

仿真实验结果还表明，原始的ESRT机制（比如缓冲区等级监测机制）自身并不能检测出这种情况的发生。那么在用原始ESRT算法根据该期望可靠性R产生一个新的下一周期报告速率时，这些值就会导致网络发生严重的阻塞或者使网络在一个很低的速率下运行，这会浪费大量的带宽。因此，网络会在（C，LR）和（NC，LR）状态之间跳转。

在这种跳转中达到的实际可靠性r远低于期望可靠性。显然，这也不是在当前网络设置下能达到的最高可靠性。这通常意味着系统是以昂贵的代价和低效率的方式工作的，网络一直试图想达到超过本身能力的可靠性，这会导致更严重的拥塞、更多的冲突和更大的时延。相应地，网络的吞吐量和整体的可靠性都会受到很大影响。

文献［Sunil08］中大量的仿真实验结果显示，传感器网络对可靠性的要求有一个阈值，这个阈值是由网络设置决定的，比如网络规模、无线信号类型、底层基本架构和协议的选择等。当期望可靠性低于该阈值时，ESRT算法可以在几个控制循环内收敛到OOR状态。但是，当期望可靠性高于阈值时，网络很快进入跳转状态。

当网络不能提供期望可靠性时，只有（NC，LR）和（C，LR）两种状态存在，如图5-7所示。 172

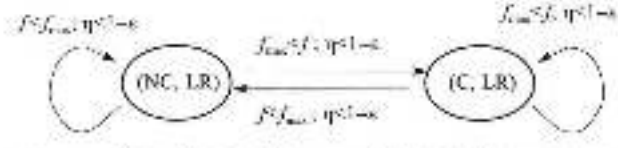


图5-7 期望可靠性过高时ESRT协议状态模型及转移图

例如，假设期望可靠性的标准是汇聚节点每10秒能成功接收到4000个分组，但是在仿真实验设置中，网络每10秒最多只能处理约3500个分组。显然，可靠性需求超过了网络的能力，不存在OOR状态。ESRT没有考虑这种情况，那么网络只能在（NC，LR）和（C，LR）状态之间跳转。

针对以上情况，文献［Sunil08］提出了一种事件到汇聚节点的增强可靠传输协议E^{2}SRT（Enhanced Event-to-Sink Reliability Transport）。

提出的机制——E²SRT

在介绍E²SRT协议前，先对ESRT中过度需求可靠性（over-demanding desired reliability）问题进行正式定义。

E²SRT中的过度需求可靠性问题是期望可靠性R要比R_{max}足够大的情况。此时，$(R_{max}/R) < 1-\varepsilon$。当期望可靠性过度需求时，称网络处于过需可靠性（Over-demanding Reliability，OR）状态。过度需求可靠性表示为R_{OR}。

下面通过数学分析证明当期望可靠性过度需求时，ESRT不会收敛到OOR状态，会在两个低可靠性状态（NC，LR）和（C，LR）之间跳转。

引理5.1：在OR状态下，归一化可靠性参数$\eta_r = r/R$不属于$[1-\varepsilon,\infty)$。

证明：由于R_{max}是网络在当前设置条件下所能达到的最高可靠性，那么显然观测事件可靠性$r_i \leqslant R_{max}$，所以

$$\eta_r = r_i/R \leqslant R_{max}/R < 1-\varepsilon$$

因此，得到$\eta_r \in (0,1-\varepsilon)$。

173 **引理5.2**：在OR状态下，网络只有两种工作状态，即（NC，LR）和（C，LR）。

引理5.2是从引理5.1简单扩展而来的，它揭示了OR状态的特质，这是E²SRT工作的基础。

需要注意，这些结果都是在所需可靠性超过了传感器网络能力时得到的，那么隐含以下假设：

$$\eta_{max} < 1-\varepsilon, R_{max} < R$$

那么只有（NC，LR）和（C，LR）两个状态是有效的。

引理5.3：仅在OR状态，初始状态为S_i =（NC，LR），而且网络没有发生拥塞时可靠性是线性变化的，那么网络状态会转移到S_{i+1} =（C，LR）。

证明：从S_i =（NC，LR）开始，ESRT主动增加f_i如下：

$$f_{i+1} = \frac{f_i}{\eta_r}$$

那么，

$$f_{i+1} = \frac{f_i}{\eta_r} = \frac{f_i}{\frac{\eta_r}{\eta_{max}} \cdot \eta_{max}}$$

由于

$$f_{max} = f_i \cdot \frac{R_{max}}{r_i} \text{ 以及 } R_{max}/R < 1-\varepsilon$$

因此

$$f_{i+1} = \frac{f_i}{\eta_r} = \frac{f_i}{\frac{r_i}{R} \cdot \frac{R_{max}}{R_{max}}} = f_{max} \cdot \frac{R}{R_{max}} > f_{max} \cdot \frac{1}{1-\varepsilon}$$

为了解决这个问题，文献［Sun08］中将该问题分解为如下的子问题：

1）如何检测出是否存在过度需求可靠性的情况？

2）如果上述情况存在，如何在不需要完全了解网络状况的情况下快速地收敛到最高可
174 靠性？

设计中考虑的主要问题是如何在一个给定的网络设置中使网络接近最高可靠性点（Maxi-

mum Reliability Point, MRP)（f_{max},η_{max}）。与 ESRT 机制相似，同样允许 MRP 附近有一个宽度为 ε 的公差带。如果在决定间隔 i 结束时，归一化事件可靠性 η_i 在区间 $[\eta_{max}-\varepsilon,\eta_{max}]$ 中，同时没有检测到网络拥塞，那么网络处于最大工作区域（Maximum Operating Region, MOR）。

在这里，沿用了 ESRT 中公差带的定义。这是一个协议参数，是用户根据需要决定的。ε 越小，系统越接近 MRP，但是收敛时间较长。

如果得到了 MRP，汇聚节点就可以将所需可靠性降低，这样网络就可以收敛到 ESRT 中的 OOR 状态。但是由于以下原因，计算 MRP（f_{max},η_{max}）的精确值是很困难的：

- 初始化的部署
- 节点移动或失效，或者其他导致网络拓扑变化的原因
- 事件的位移
- 无线信号干扰

因此，算法假定存在预设恒定的 MOR 是不可行的，需要采用更加高级的能够适应网络环境变化的算法。算法应该能够接收传感器网络的反馈并且通过递归的方式估算出 MRP。

E^2SRT 中提出的新算法继承了 ESRT 的主要特性，比如通信模型和网络模型的定义。它是基于汇聚节点的，高能效，有更快的收敛时间。作为一个增强性的协议，E^2SRT 在 OR 状态工作，它更能适应网络突然变化和资源限制等状况。

接下来将详细介绍 E^2SRT 是如何接近 MOR 以及如何在 3 个 OR 状态下工作的。

在每个决定间隔结束时，汇聚节点计算出归一化事件可靠性 η_i，同时根据拥塞报告决定当前网络状态 S_i。根据当前网络状态 S_i、η_i 和 f_i 的值以及 ESRT 中的区域判别条件，E^2SRT 机制会计算出下一决定间隔的报告速率 f_{i+1} 报告给汇聚节点，再由汇聚节点广播给其他节点。相关的节点会在下一决定间隔根据这个已更新的报告速率报告事件分组。该过程会不断重复，直到达到 MOR 状态。图 5-8 为状态转换图。

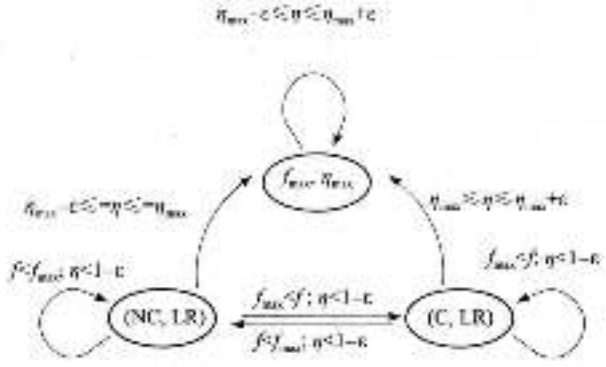


图 5-8 期望可靠性过大条件下 E^2SRT 协议状态模型及转换图

E^2SRT 引入了一个递归算法，能够对 MRP 进行数轮估计后收敛到 MOR。如图 5-9 所示，在归一化事件可靠性是报告速率（对数形式）的函数的曲线中，MOR 附近显示出线性和对称性质。前面已经讨论过，网络只会在（NC，LR）和（C，LR）两个状态之间摆动。显然，（NC，LR）总是处于 MRP 左边，而（C，LR）总是处于 MRP 右边，那么 MHP 处于两个状态之间某点。将上一个（C，LR）状态的报告速率记为 $f_{(C,LR)}$，上一个（NC，LR）状态的报告速率记为 $f_{(NC,LR)}$。

对报告速率更新的估计如下：

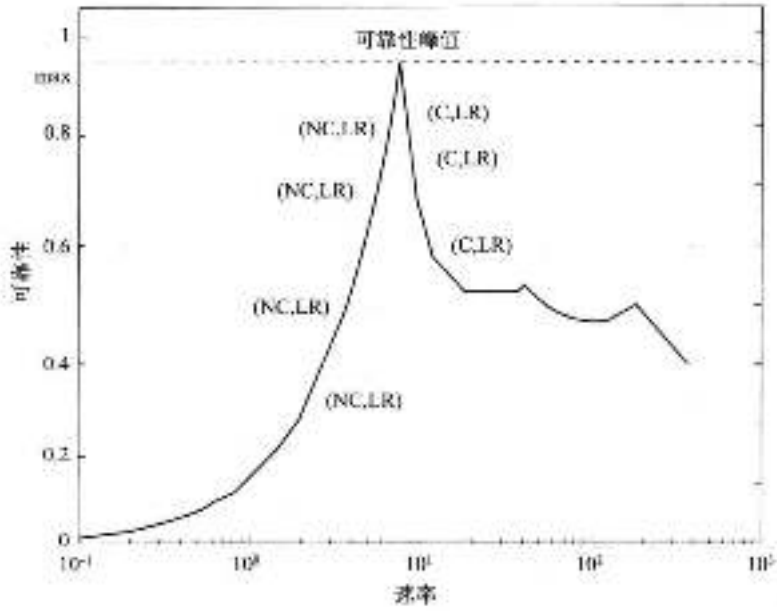


图 5.9　F²SRT 的递归收敛情况

$$f_{t+1} = 10^{\frac{\log f_{(NC,LR)} + \log f_{(C,LR)}}{2}}$$

根据上式，无论从两个状态中哪一个状态开始，网络会在（NC，LR）或（C，LR）状态中持续多个决定间隔。如果 f 离状态切换点很远，为了提高收敛速率，在计算更新报告速率时给最近记录的相反状态的速率加上更大的权重，具体增加方法如下：

（1）（NC，LR），即无拥塞，低可靠性

由于不可能进入 OOR 状态，那么更新策略的目标就是促使网络进入 MOR 状态，而非 OOR 状态。如引理 5.3 指出的，利用 ESRT 算法，网络最终不可避免地会进入最不理想的（C，LR）状态。现在已经知道了网络处于 OR 状态，这是由于它会跳入（C，LR）状态至少一次，然后再跳回（NC，LR）状态。报告速率的更新如下：

$$f_{t+1} = 10^{\frac{1}{k+1}\log f_{(NC,LR)} + \frac{k}{k+1}\log f_{(C,LR)}}$$

（2）（C，LR），即拥塞，低可靠性

在该状态中，若检测出该状态是从（NC，LR）状态转换过来的，那么就知道网络处于 OR 状态；或者网络自身仍旧处于（C，LR）状态，这就意味着应该进一步降低报告频率。这里引入一个参数 k，用于对网络连续处于（C，LR）状态的时间间隔进行计数。k 增大，意味着 $f_{(NC,LR)}$ 比 $f_{(C,LR)}$ 更加接近 MOR，就应该给 $f_{(NC,LR)}$ 比 $f_{(C,LR)}$ 更多的权重。综合这些考虑，更新报告频率如下：

175
|
177

$$f_{t+1} = 10^{\frac{k}{k+1}\log f_{(NC,LR)} + \frac{1}{k+1}\log f_{(C,LR)}}$$

（3）MOR（最大工作区域）

在该状态下，网络在最大工作点附近公差带为 ε 的区域内工作，此时网络已经在最低功耗下尽力满足了可靠性需求。下一决定间隔的报告频率保持不变：

$$f_{t+1} = f_t$$

图 5-10 给出了整个 E²SRT 协议算法概括的伪代码。

```
k = 1;
ESRT=1;
/* ESRT=1 indicates that the network is in normal ESRT operation*/
E^2SRT()
/* Probe the network state*/
If S_{i-1}=(NC, LR) and S_i=(C, LR)
ESRT=0 /* OR state is detected*/
End;
If (ESRT)
/* ESRT operations takes action*/
…
end;
else if (ESRT = 0)
        if Si=(NC, LR) and |n_{i-1}-n_i|<= ε/2
        /*network is in NOR states*/
        /*keep f toward frequency used in last state */
        f_{k+1} = f_i
        end;
        If (C, LR) /*state=(C, LR)*/
        /* decrease f toward frequency used in last (NC, LR) state */
        f_{k+1}=10^{\frac{k}{k+1}\log f_{NC,LR}+\frac{1}{k+1}\log f_{C,LR}}
        k = k+1;
        end;
        else if(NC, LR) and |n_{i-1}-n_i| > ε/2
        /* state=(C, LR)*/
        /* increase f toward frequency used in last (C, LR) state */
f_{i+1}=10^{\frac{\log f_{NC,LR}+\log f_{C,LR}}{2}}  k=1
            end;
end;
```

图 5-10　E²SRT 协议运行算法


奇思妙想

很多学生一直在问一个问题："我怎么进行研究？"看一下 E²SRT 的例子。它从一个已有的机制（ESRT）开始，试图找出其隐藏的缺陷或者未解决的问题，最后找出一个好方法解决这些问题。"改进"是开始做研究的好方法。但到最后，你应该进行高水平的研究——自己定义一个有意义的重要的研究问题，然后用一个全新的方法（其他人没有发现的）去解决它！看一看这些研究者，他们都在做相同的事情——"发现一个新问题，再想出一个好的解决方案"。

5.5　CODA：传感器网络中的拥塞检测与避免

前面介绍的传输机制已经实现了无线传感器网络传输层的第一个目标——可靠性。在这一节，将介绍一个实现第二个目标的解决方法，即拥塞控制的解决方法。

为了说明拥塞问题，文献［Wan03］用模拟试验的结果（参见图 5-11）显示了无线传感器网络中适量的活动源节点以变化的报告速率进行数据发布时受拥塞的影响。

从图 5-11 可以得到一个有意义的结论：存在一个"拐点"，即当源事件的速率增长超过某个网络能力阈值（本例中为 10 个事件/秒）时，拥塞发生更加频繁，而且在汇聚节点处丢弃的

分组总数急剧增长。从中还可以看出，甚至在低到中等源事件速率下也会发生拥塞。丢弃的分组包括 MAC 信令、数据事件分组本身和扩散消息分组。

图 5-11 所示的丢弃速率不仅说明网络中存在大量的分组丢失，还说明网络存在拥塞。更重要的是，由于分组传输的失败，浪费了大量的能量。而在无线传感器网络中，能量资源是最为重要的。

不同的无线传感器网络应用会带来偶尔的或者频繁的数据速率“突发”（即突然产生大量的事件数据）。一些应用（比如亮度监测）可能只会从网络中很小的区域内产生少量的流量，而在其他应用（比如图像传感器网络）中，可能会在整个感知区域内产生大量的突发数据，这会导致大量的分组丢失（如图 5-11 所示）。

178
|
179

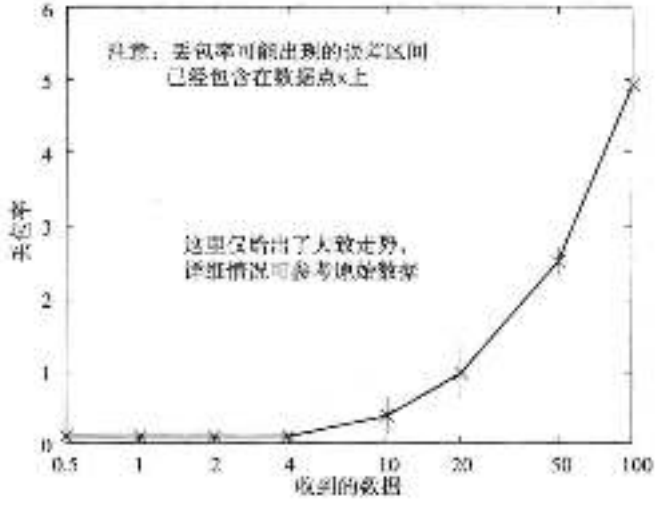


图 5-11　源节点速率与汇聚节点丢包率之间的函数关系

无线传感器网络拥塞控制机制必须能够在短暂或者持续拥塞期间，保持汇聚节点交付信令的精确度。这里重点关注以下 3 种不同的拥塞场景：

- 密集传感器网络：突发数据事件会产生持续的热点（hot spot），热点与源节点附近（一跳或两跳内）某处开始的突发速率成比例。在这个场景下，拥塞控制应该是本地（在源节点附近）的、快速的，并且能够提供从拥塞点到源节点的反压，这是有效的。
- 低数据速率的稀疏传感器网络：短暂的热点可能发生在传感器网络中任何区域，但更可能在到汇聚节点的方向上远离事件源。在这种情况下，采用综合了局部反压（在同一个热点区域的节点之间）和分组丢弃技术的快速机制会更有效。由于拥塞短暂，源节点不会受到反压的影响。
- 产生高数据速率事件的稀疏传感器网络：在该场景下，短暂的和持续的热点都分布在整个感知区域内。为了控制拥塞，就需要一个快速机制解决局部的短暂拥塞，还需要对所有导致持续拥塞的源节点进行闭环的速率调节。

180

文献［Wan03］为无线传感器网络设计了一个高能效的拥塞控制机制 CODA（Congestion Detection and Avoidance），它由以下三部分组成。

1. 拥塞检测

拥塞控制的第一步是进行准确有效的拥塞检测。在 CODA 中，每个传感器节点综合利用其观测到的当前与以往的信道负载状况、当前节点内缓冲区使用情况等信息来检测邻居区域是否

发生拥塞。由于介质是共享的，相邻节点可能通过信道同时发送数据，所以必须知道信道的状态。如果通过持续监听信道获取本地负载状况，耗费的能量较高。因此，CODA 采用以合适的时间对信道监听的采样机制。这样既能得到精确估计同时降低了能耗。一旦检测到拥塞，节点通过反压机制向（源节点方向）上游相邻节点发送信令。

2. 开环逐跳反压

在 CODA 中，一个节点一旦检测到拥塞就立即广播一个反压信号。反压信号向源节点方向上游节点传递。对于密集网络中的突发数据事件，反压信号会直接传递给源节点。收到反压信号的节点根据本地拥塞策略（如丢弃分组），降低其发送速率或丢弃分组。当上游节点（向源节点方向）收到一条反压信号，它会根据其本地网络条件，决定是否继续向上传递该反压信号。如果检测到拥塞，便继续向上传递反压信号。

3. 多源闭环调整

在 CODA 中，闭环速率调整运行较慢，它能在发生持续拥塞时进行单汇聚节点下的多源拥塞控制。当源节点事件速率低于信道最大理论吞吐量的一定比例时，源节点自行调整。但是，当源节点事件速率大于该阈值时，节点可能导致拥塞的发生，那么就会触发闭环调整，源节点依靠汇聚节点调整。在这种情况下，源节点需要来自汇聚节点的恒定、慢速的反馈（即 ACK），以维护其速率。源节点收到的 ACK 作为自同步机制，使源节点保持当前的事件速率。如果源节点没有收到 ACK，那么会降低速率。

开环和闭环控制的关系如下：在上述不同的场景下，热点（即拥塞区域）可以发生在感知范围内不同的区域中，因此 CODA 需要开环逐跳反压和多源闭环调整机制。这两种控制机制可以单独使用，但是共同使用会更加有效，因为它们能够互补。

从上述的描述中可以看出，速率控制机制在源节点、汇聚节点或者中间节点上的操作是不同的。源节点知道发送流量的特性，中间节点不知道发送流量的特性。汇聚节点能够根据接收到的网络中所有节点事件报告非常准确地估算出网络中源节点的流量特性。一般来说，汇聚节点能力较强，可以实现复杂的估算。CODA 的目标是在无拥塞条件下不进行任何操作，但是在检测到拥塞后能够迅速响应，缓解热点周围的拥塞状况。


奇思妙想

开环与闭环控制方案都被广泛应用于系统控制应用中。开环控制的实现较为简单，而闭环控制运用输出反馈来调节输入，对系统的控制更加精确、稳定。

5.5.1 开环逐跳反压

在前文中已经简要描述了快速/慢速拥塞控制机制。反压机制是一种快速控制机制。如果一个节点检测到拥塞，它会立即广播一条抑制消息给它的一跳上游（源节点方向）节点。它是通过查询路由协议得到上游节点信息的，路由协议处于无线传感器网络协议栈中传输层协议的下一层。

当上游节点（源节点方向）收到反压信号后，根据本地网络状态决定是否向上游节点继续传递该反压信号。例如，节点收到反压信号后，它可能将到达的分组丢掉，防止队列填满而堵塞。

以上讨论的是开环控制。对于闭环拥塞控制，需要在本地处理持续的拥塞，而不是传递反

压信号。

CODA 定义拥塞程度（depth of congestion）来表示载有反压信号的消息在到达非拥塞节点前转发的跳数。路由协议和本地分组丢失策略可以采用拥塞程度以缓解网络拥塞，具体方法如下：

1）路径优化选择：传输路径过于拥塞时，用瞬时拥塞程度作为指示，路由协议据此另外选择较佳路径。这样能够减轻发生严重阻塞的路径上的流量；

2）信令消息主动丢弃：不把拥塞控制和路由联系在一起，节点可以进行抑制或者将有关路由协议或数据分发协议的重要信令消息（比如定向扩散协议中的兴趣、SPIN 数据广播等）
182 丢弃。该方法能够以透明的方式使事件流离开阻塞区域和远离热点。

5.5.2　拥塞检测

为了检测拥塞状况，可以采用一些简单的方法。比如检查节点的队列是否填满，或者对当前通信信道流量负载进行测量——如果负载接近上界，那么就表示发生拥塞。

对于第一个方法，对队列大小进行监测，其执行开销低，但是该方法不能提供准确的拥塞检测，因为队列会由于一些本地状况而溢出。对于第二个方法，通过监听共享信道，可以得到信道负载状况或者关于冲突检测效果方面的协议信令的信息。因此，CODA 倾向于选择第二种方法。但是，如果持续对信道进行监听会导致高能耗，那么只能在适当时采用该方法，以降低系统能耗。

对于信道监听最佳时间的考虑，可采用 MAC 协议中的技巧。通常，传感器节点会在发送数据分组前对信道进行监听，这个信道监听过程称为载波检测（carrier sense）。如果信道在这段时间内持续空闲，那么无线通信模块切换到传输模式发送数据分组。

因此，对信道进行监听的最佳时间就是在进行载波检测时。由于在分组传输前都要进行载波检测，那么当节点需要发送分组前对信道进行监听并且测量负载是不会有额外消耗的。

图 5-12 所示是一个典型的有热点和拥塞区域的场景。在该例中，节点 1 发送数据给节点 3，节点 4 发送数据给节点 5，两组数据流都需要经过节点 2。节点 2 将分组存储在缓冲区内，最终会丢弃分组。这个例子说明：在拥塞相邻区域内，接收节点（即图 5-12 节点 2）的缓冲区占用率较高。此时，节点 2 开始进行信道负载测量。在缓冲区清空后，信道负载测量就会自然停止，这表示很可能拥塞已经减轻，数据在周围流畅地传递。

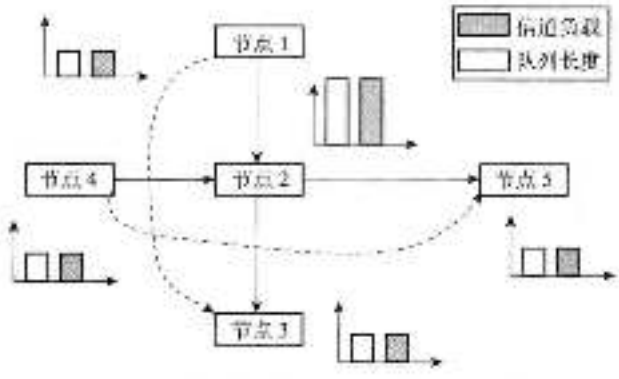


图 5-12　基于接收节点拥塞检测的无线网络

5.5.3 基于采样的信道监听

监听周期定义为分组传输时间的倍数。当节点开始监听信道时，要求对信道至少监听一个监听周期，以便测量信道负载状况。在一个监听周期内，节点如果连续地在退避期间进行监听，那么能耗较高。因此，CODA 进行周期性采样，不进行采样时便可以关闭无线通信模块，以节省能量。

采样机制如下：设监听周期长度为 E，在 N 个连续监听周期内进行信道负载测量，每个监听周期中，用一个预先定义的采样速率获取信道状态信息，即在单个监听周期内信道忙与空闲状态的时间。然后根据 ϕ_n（在监听周期 n 期间的信道负载测量结果）的指数平均以及参数 $\alpha(0<\alpha<1)$ 计算前 N 个连续监听周期的信道负载测量结果 $\bar{\phi}$，如下式：

$$\bar{\phi}_{n+1}=\alpha\bar{\phi}_n+(1-\alpha)\phi_n,\quad (n\in\{1,2,\cdots,N\},\bar{\phi}_1=\phi_1)$$

如果发送缓冲区在 n 达到 N 之前清空，那么忽略平均值，将 n 设为 1。数组 (N,E,α) 可以用于调整采样机制以实现对特定无线通信系统体系结构的信道负载的精确测量。

根据上式，可以获得对变的信道负载状况。当负载超过一个阈值时，意味着网络发生拥塞。在这种情况下，节点广播抑制消息作为反压信号，同时运用本地拥塞策略。只要拥塞持续发生，节点会以最小间隔继续广播这条消息，直到广播次数达到特定的最大值。抑制消息为开环反压机制提供了基础。

183
:
184

5.6 STCP：无线传感器网络的传输控制协议

文献[Iyer05]针对无线传感器网络提出了一个通用、可扩展的可靠传输协议——STCP。与网络中资源非常有限的传感器节点相比，可以认为基站拥有不受限制的资源，因此基站实现了 STCP 协议的主要功能。

5.6.1 STCP 中的数据传输序列

TCP 协议进行三次握手的目的是建立一个端到端（end-to-end）的 TCP 连接。类似地，运行 STCP 协议的传感器节点在传输数据前需要与基站建立一个关联（association）（与 TCP 中的"连接"概念相似），这是由会话初始分组（session initiation packet）实现的。

会话初始分组向基站传递如下信息：来自该节点的数据流的大小、数据流的类型、传输速率和要求的可靠性。当会话初始分组达到基站后，基站会存储其所有信息，然后为每个数据流设置定时器及其他参数，向源节点发送应答分组（ACK）。传感器节点收到 ACK 后，确认和基站建立了关联，此时可以开始向基站发送分组。在反向路径上，基站根据数据流类型决定发送 ACK 或 NACK 信号。

5.6.2 STCP 分组的格式

图 5-13 给出了会话初始分组的报文格式。STCP 中会话初始分组是受多流（multiple-streams）概念启发而设计出来的，如果一个传感器节点上有多个感知设备，它们中的部分或全部需要传输各自的感知数据，节点只需要发送一个会话初始分组。应该注意的是，源节点发送的数据是与各个数据流独立关联的，这是因为不同流的传输性质可能是不同的。在图 5-13 中，第一个字段是序列号（Sequence Number），16 位。对于会话初始分组，它被置为 0。第二个字段是流（Flows），8 位，表示从该节点发出的流的数量；时钟（Clock）字段表示发送时的本地当前时间；流序号（Flow ID）字段用于区分不同流的数据分组；数据流位（Flow Bit）字段用

下指定流为连续的（数据流不会停止）还是事件触发的（当检测到事件时才发送分组）。对于连续数据流，传输速率（Transmission Rate）字段表示源节点发送分组的速率。可靠性（Reliability）字段给出了流所要求的可靠性。

序列号（16）		数据流（8）	选项（8）
Clock（32）			
流 ID#1（8）	数据流位（8）	传输速率（8）	可靠性（8）
流 ID#2（8）	数据流位（8）	传输速率（8）	可靠性（8）
流 ID#N（8）	数据流位（8）	传输速率（8）	可靠性（8）

图 5-13　会话初始分组格式

图 5-14 为 STCP 数据分组头的格式，它与会话初始分组的头相似。数据分组的序列号是一个非零整数（对于会话初始分组是 0）。流 ID（Flow ID）指出流的类型，基站通过该字段确定分组的传输性质。

序列号（16）	流 ID（8）	CN（1）	选项（7）	时钟（32）

图 5-14　STCP 数据分组头格式

分组头中包含了一个与拥塞控制相关的重要字段——拥塞通知（Congestion Notification，CN）。由于它是一个 1 位长的字段，当其为 1 时，意味着发生了拥塞。Clock 字段给出了发送该分组时的本地时间。基站用 Clock 值计算该节点中此 Flow ID 流的估计传输时间（Estimated Trip Time，ETT）。

185
~
186

图 5-15 为应答分组的格式。ACK/NACK 字段表示它为肯定或否定应答，其他字段参见前文中的解释。STCP 用 32 位长 Clock 字段与序列号字段，从而避免了回环问题。Options 字段用于将来的扩展。

序列号（16）	流 ID（8）	CN（1）	ACK/NACK（1）	选项（6）

图 5-15　STCP 应答分组格式



关于分组格式：当设计一个网络协议时，首先应该知道分组格式。这是因为分组头中字段内容不同时协议的操作也不同。有时候没有一个标准化的分组格式，在这种情况下，需要自己定义一个分组格式。而且要尽量减少字段长度——如果能用 3 位表示 5 种情况就不要 4 位表示。

5.6.3　连续数据流

基站可以通过会话初始分组知道源节点的发送速率。因此，它可以估算下一个分组预期到达的时间。基站会设置一个定时器，如果在预期时间内没有收到下一个分组，它将发送否定应答（NACK）。

当基站收到来自一个节点的分组后，它将通过以下方法之一计算下一个分组到达的预期时间：

1）超时（time-out）的值由（$T+\alpha\times ETT$）决定，其中 T 是两个连续传输之间的间隔，α 是一个随 ETT 变化的正整数。基站在（$T+\alpha\times ETT$）单位时间内不断地检查是否接收到了每个传感器节点的分组。如果在该时间内收到了数据分组，那么使 α 减少 0.5。如果分组丢失（发生

超时）或在发送 NACK 后收到分组，那么使 α 增加 0.5。

2）第二个方法是基于 Jacobson/Karels 算法［VJacobson88］的改进方法。Jacobson/karels 算法考虑的是往返时间（Round Trip Time，RTT）的变化。但本方法使用 ETT 替换 RTT，在考虑 ETT 的情况下修改 Jacobson/Karels 算法。基站根据下面的描述动态地改变 δ、μ 和 ϕ 的值：

$$\text{采样 ETT} = \text{基站时钟} - \text{分组时钟值}$$

$$\text{差值} = \text{采样 ETT} - \text{估计 ETT}$$

$$\text{估计 ETT} = \text{估计 ETT} + (\delta \times \text{差值})$$

$$\text{偏差} = \text{偏差} + \delta(|\text{差值}| - \text{偏差})$$

$$\text{超时} = \mu \times \text{ETT} + \phi \times \text{偏差}$$

若源节点在发送分组后没有接收到 NACK，那么分组肯定已经到达基站，除非 NACK 丢失。因 187
此，基站为所有发送过 NACK 的分组维护一个记录。如果一个发送过 NACK 的分组达到，基站从记录中将相应项清除。基站周期性地检查该记录，如果发现有存在的项，则重传 NACK。

5.6.4 事件触发数据流

由上一小节可知，连续数据流采用否定应答（NACK）消息机制以保证可靠性。由于假设在连续数据流中数据包很少丢失，相应地，节点很少回送 NACK 消息。相反，如果采用肯定应答（ACK）消息，网络中会存在大量的 ACK 消息，这将加重连续数据流的网络拥塞程度。

在事件触发数据流中，由于数据发送不频繁，连续的两个分组到达基站的间隔较长，因此基站无法预计数据分组的到达时间，就不需要时钟同步。由于可靠性的需求，用肯定应答（ACK）告知源节点分组是否到达基站。

与可靠传输的 TCP 协议相同，源节点将每个已经发送的分组放入缓冲区。在收到基站回发的 ACK 后，将相应与可靠传输的 TCP 协议相同，的分组从缓冲区内删除。节点维护一个周期性触发的缓冲定时器。当定时器触发后，重传缓冲区内存在的分组。

5.6.5 可靠性

传感器节点在会话初始分组中指定每个流所需要的可靠性。对于连续数据流，基站计算出一个运行中的平均可靠性，以成功收到分组的比例来衡量可靠性。如果当前可靠性已经达到所需要的可靠性，那么即使基站在预期时间内未收到分组，它也不会发送 NACK。只有在当前可靠性低于所需水平时基站才发送 NACK。

对于事件触发数据流，基站计算接收到的分组个数与接收到的分组中最大序列号的比值，以此作为该数据流的实际可靠性。在传感器节点上，在发送一个分组前先假设该分组不能到达基站，计算这种情况下的有效的可靠性，若可靠性仍然大于所需可靠性，节点在发送该分组后不将其放入缓冲区，以便节省存储空间。如果节点接收到 ACK，则增加可靠性。

5.6.6 拥塞检测与避免

拥塞检测与避免是传输层协议中一个重要的方面。Floyd 和 Jacobson 设计的随机早期检测机制（Random Early Detection，RED）仅要求中间节点在发现拥塞时丢弃分组［SFloyd 93］，但是丢弃操作会导致该分组传输超时或者基站向源节点发送 NACK 报文。对于传感器网络而言，RED 的拥塞丢弃机制会增加不必要的网络负载并且可能影响系统的监测结果，因此 STCP 并未采用该机制。

在文献［KRamakrishnan90］提出的拥塞控制机制中，中间节点监测流量负载，然后通过将分组中的 CN 位（拥塞通知位）置 1 以便显式地通知端节点。STCP 采用该显式拥塞通知方

法，并做了一些改进。

188 每个 STCP 数据分组的头中有一个 CN 位。传感器节点在其缓冲区内维护两个阈值：th_{low}（下界限）和 th_{upper}（上界限）。当缓冲区达到 th_{low} 时，以一定的概率将 CN 位设置为 1。对此概率取值时可以采用类似于 RED 中所采用的方法。当缓冲区达到 th_{high} 时，意味着拥塞严重，节点将所转发的每个分组中 CN 位设置为 1。

在收到该分组后，基站将应答分组中的 CN 位设置为 1，以便将发生拥塞的路径信息通告给源节点。源节点在收到此拥塞通知后，选择其他路径发送后续分组或降低发送速率。注意，节点是依靠网络层的路由算法查找新路径的。

5.6.7　以数据为中心的应用

对于以数据为中心的应用，我们通常只关注全网整体的感知信息，而不是单个传感器节点的数据。比如，对地震活动的监测和查询网络中温度最高值。在这些应用中，中间节点能通过**数据聚合**（data aggregation）过程将关联的数据聚合在一起。由于聚合的数据来自大量的传感器节点，所以不应该要求基站通过 ACK 或 NACK 向所有传感器节点发送应答，因为这样会耗尽网络资源和能量。

因此，在以数据为中心的应用中，STCP 不提供任何应答机制，这与 Internet 中的 UDP 相似。STCP 假设来自不同节点的数据互相关联而且能够容忍部分丢失，只要事件总体能够可靠地发送到基站。ESRT 协议也采用了类似机制。

5.7　GARUDA：实现有效可靠的下行通信

ESRT 关注的是事件由传感器节点到汇聚节点（上行）可靠性问题。本节讨论关于点对多点的下行可靠数据交付（即从汇聚节点到多个传感器节点）的问题。文献［Seung-Jong08］提出了 GARUDA（原义为神话中神灵乘坐的一种鸟），它可以有效保证下行数据传输的可靠性。

汇聚节点向传感器节点发送的通常都是重要数据（比如数据查询指令），因此来自汇聚节点的每条消息应通过可靠传输到达目的节点。比如，对于一个图像传感器节点网络应用，汇聚可能会发送以下三类消息，它们都需要可靠交付给传感器节点：1）软件无线空中升级（over-the-air）代码：如果网络中采用了能够改编程序的可配置传感器，那么汇聚节点需要给传感器发送升级后的图像检测/处理软件；2）数据查询指令：汇聚节点可能向节点发送数据查询指
189 令；3）数据采集指令：最终，汇聚节点会向传感器请求数据结果。

5.7.1　无线传感器网络中下行数据可靠性面临的挑战

1. 环境限制

首先需要考虑的是无线传感器网络中通信带宽和节点能量等资源限制给下行数据可靠传输带来的影响。由于无线传感器网络资源受限，这就要求传输可靠性保证机制必须能够在满足系统可靠性的前提下尽可能降低自身的开销。另外，节点失效（可能由于能量耗尽）会导致网络拓扑的动态变化，下行数据可靠性应该能够适应拓扑结构的动态变化，因此不能采用未考虑网络动态性的固定结构化机制（如广播树）。

另一个挑战源自网络的扩展性。无线传感器网络由大量节点组成，网络的直径可能很大。因此，在下行数据传输过程中，汇聚节点可以利用网络中存在大量的冗余节点/路径实现空间复用（spatial reuse），这样可以降低时延。但是，实际上传输层协议采用的丢失恢复机制可能严重限制空间复用。

2. ACK/NACK 考虑

节点在收到数据后应该向发送方报告 ACK 还是 NACK 取决于不同的情况。比如，若分组丢失率很低，报告 NACK 的方式能够节省更多带宽，因为向发送方发送的 NACK 很少。但是对于分组丢失率高的情况，报告 ACK 的方式可以节省更多的通信开销。

此外，采用 NACK 方式需要处理最后分组丢失问题，这个问题已经在前面讨论过。基于 NACK 的丢失恢复方法要求节点按顺序转发数据，以防出现 NACK 内爆问题［CYWan02］。这显然限制了网络中的空间复用。

3. 可靠性语义

在无线传感器网络中，需要考虑感知数据的位置相关性和冗余性。

- 位置相关性：在很多情况下，需要找出事件的具体位置。基站发出的数据查询指令可能是和位置相关的，比如“发送房间 X，Y 和 Z 的温度”。
- 位置冗余性：对于节点密度较大的无线传感器网络而言，同一事件区域内的节点的感知数据存在冗余性，它们不需要各自的感知数据全部可靠地传输到汇聚节点。GARUDA是一个下行（汇聚节点到事件的可靠数据传输机制，它采用了“局部可靠性”，即汇聚节点只需要保证与相邻区域内部分节点可靠通信。 190

ARUDA 根据上述特征定义了“可靠性语义”，将可靠性语义分成四类：

- 对整个无线传感器网络可靠交付，这是默认语义。
- 对无线传感器网络中某个区域的节点可靠交付，这是基于位置交付的典型代表。
- 对感知区域内所有节点的可靠交付，这是冗余感知数据交付的典型代表。
- 对随机选取的网络节点子集的可靠交付，这与传感器网络应用实际的监测解析度有关。

5.7.2 GARUDA 基本设计

GARUDA 设计的核心部分是**可快速建构的丢失恢复结构**（instantaneously constructible loss recovery infrastructure），称为 GARUDA 核（core）。GARUDA 核可以近似看作网络拓扑图的**最小支配集**（Minimum Dominating Set，MDS）。支配集是一些节点的集合，通过它们可以很容易地到达其他所有节点（比如从集合中的某个节点出发最多一跳远信即可到达其他节点）。

虽然 MDS 不是一个解决网络问题的新想法［RSivakumar99］，但还是对实现 GARUDA 的丢失恢复过程有很大的作用。它在第一个分组交付时构造核，采用两阶段丢失恢复策略来恢复丢失的数据。丢失恢复采用无序转发并且实现了使重传开销和时延最小化的目标。它还采用基于候选的方式构造核以支持多种可靠性语义，如图 5-16 所示。

GARUDA 是基于突发方式的，意味着它可以把一个分组可靠地传输给网络中所有节点。因为 GARUDA 能够确保可靠地交付任意长度消息的第一个分组，所以 GARUDA 没有直接转发 NACK 方式中无法解决的所有分组丢失问题。这使得 GARUDA 既能利用 NACK 方式的优点，又能避免采用 NACK 方式引起的缺陷。

在下面对 GARUDA 的介绍中，首先在假设第一个分组已经被可靠交付的情况下，讨论核的基础设施。然后，再了解如何实现第一个分组的可靠交付。

1. 丢失恢复服务器：GARUDA 核

GARUDA 的核为一组在网络本地选举产生的丢失恢复服务器，这里所说的服务器并不是指真正的机器，而是指提供丢失恢复服务的节点。使用核的构造算法时需要考虑两个问题：1）如何选择核心节点（core node）才能达到使重传开销最低的目的？2）如何使得核的构造算法适应因节点失效或其他原因造成的网络拓扑变化？ 191

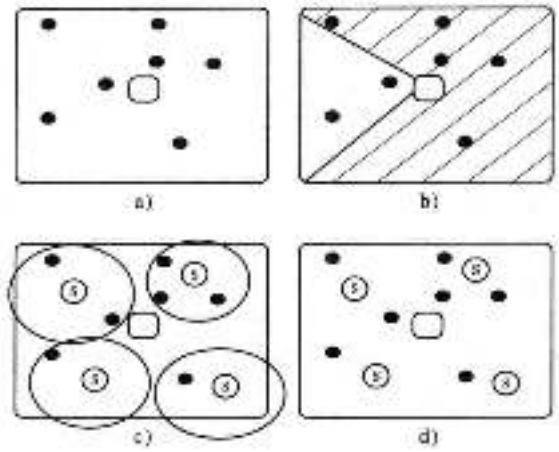


图 5-16　可靠性语义类型。a）对所有节点的可靠交付；b）对一个子区域的可靠交付；c）邻居感知区域的最少节点的可靠交付；d）对 80% 节点的概率可靠交付

GARUDA 在第一个分组交付期间完成核的构造。一旦第一个分组被可靠交付，就可以确定传感器节点与汇聚节点之间的跳数（hop count）。任何一个跳数是 3 的整数倍的节点，而且它监听不到其他任意一个核心节点，那么就将它自身选为 GARUDA 核心节点。之所以将跳数为 $3i$ 的节点作为核心节点是因为它可以覆盖跳数为 $3i-1$ 和 $3i+1$ 的节点，因此它就可以在从汇聚节点到传感器节点的方向上如同 MDS 一样发生作用。

总之，核的即时构造（在第一个分组中每条新消息交付期间）有效解决了网络中节点失效的问题。

2. 丢失恢复过程

（1）无序报文转发

在传统的传输层协议中，比如 Internet 的 TCP 协议，所有分组的交付都是按序进行的，即发送方在没有收到接收方对当前分组的应答之前是不会开始发送较大序列号分组的。有时网络可能会丢弃分组，那么在发送序列号较大分组前就需要重传这些丢失的分组。按序转发的主要缺点是：在发生分组丢失后，较大序列号分组被禁止转发，保留的下行（汇聚节点到事件节点）网络资源未被充分利用。 192

因此，GARUDA 采用了能够克服上述缺点的无序转发策略，丢失了分组的节点可以继续转发接收到的序列号更大（或更小）的分组。

为了抑制不必要的重传请求，GARUDA 运用可扩展的可用性位图（Availability Map，A-map）在核心节点之间交换的机制，A-map 包含元数据信息，比特位代表分组的有效性。若下游核心节点丢失分组，在接收到上游节点发送的 A-map 后，丢失的分组对应于 A-map 中的分组位置 1 时（表示上游节点有丢失的分组）才能请求丢失分组。核心节点只有在确认上游节点有自己丢失的分组时才请求丢失的分组，因此 GARUDA 核恢复阶段非常高效。

（2）两阶段丢失恢复

GARUDA 一旦完成核的建立，就立即进入两阶段丢失恢复过程：第一阶段由核心节点恢复所有丢失分组；第二阶段由非核心节点恢复丢失分组。

由于只选择了跳数为 $3i$ 的节点作为核心节点，那么在网络中非核心节点占节点总数的大部分。因此，首先进行核心节点的丢失恢复可以防止其与大量非核心节点的竞争。

第二阶段的丢失恢复在非核心节点监听到来自核心节点的消息（表明核心节点已经收到了

所有分组）后开始。因此，第二阶段恢复在每个区域都不会与第一阶段恢复相重叠，这样防止了与第一阶段恢复的竞争。

5.7.3　GARUDA 架构

在详细描述 GARUDA 机制前，首先假设有一个拓扑如图 5-17 所示的网络。之前提到过，在第一个分组的交付过程中能够找出跳数为 $3i$ 的核心节点。将所有相对汇聚节点具有相同跳数的节点称为一个“组带”（band），组带 ID（bID）与跳数相同。

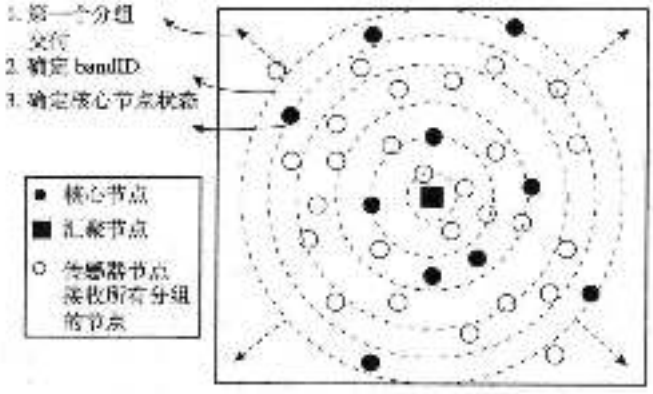


图 5-17　GARUDA 中核心的建立过程

193

考虑具有相同 bID 的节点（即处于同一组带中的节点），显然，组带可以看成以汇聚节点为圆心的同心圆。此外，每个核心节点的 bID 为 3，6，9 ……

1. 核心节点构造过程

- 汇聚节点（即基站）

当汇聚节点发送第一个数据分组时，在该分组上标记 bID 为 0。任何节点收到第一个分组后，将分组上的 bID 加 1，同时将该值设为自己的 bID。

- 组带 $3i$ 上的节点

所有处于组带 $3i$ 上的节点都有可能成为核心节点。当一个 bID 为 $3i$ 的节点转发第一个分组时（在收到该分组时等待一段随机时延后），它会首先检测是否已经监听到同一组带内其他核心节点。如果它没有监听到同一组带有其他核心节点，那么它选择自己为核心节点。这样做的原因是减少任意两个核心节点之间的竞争（也能减少核心节点数）。

如果一个在组带 $3i$ 上的节点还没有将自身选为核心节点，当它收到一条显式的核心请求消息后，本阶段它将自己选为核心节点。

为了保持组带与组带之间的通信，每个处于组带 3（$i+1$）上的节点至少都应该知道一个处于组带 $3i$ 上的节点。如果它通过一个组带 $3i$ 上的节点收到第一个分组，它就可以隐式地知道该信息，这是由于每个分组都携带了之前访问的核心节点的 ID。

- 组带 $3i+1$ 上的节点

当一个处于组带 $3i+1$ 上的节点 S_1 接收到第一个分组后，它首先检查该分组是否来自核心节点。如果源节点 S_0 为核心节点，那么节点 S_1 将其核心节点设为 S_0。否则，它将 S_0 设为候选核心节点，同时启动一个核选举定时器，该定时器的时长大于第一个分组交付的重传定时器时长。如果节点 S_1 在核选举定时器超时前监听到了核心节点 S'_0，那么它将 S'_0 设为核心节点。

但是，如果在核选举定时器超时前没有监听到任何核心节点，那么它将选择 S_0 为核心节

点，并向 S_2 单播该决策通知。

- 组带 $3i+2$ 上的节点

处于组带 $3i+2$ 上的一个节点 S_3 接收到第一个数据分组时，它并不知道任何一个处于组带 $3(i+1)$ 上的节点，在不选择核心节点的情况下将第一个分组转发，同时启动核选举定时器。如果在该定时器超时前 S_3 监听到了一个组带 $3(i+1)$ 上的节点 S_1，那么 S_3 选举 S_1 为核心节点，并向 S_1 单播该决策通知。如果 S_3 没有监听到任何一个处于组带 $3(i+1)$ 上的节点，那么
194 它发送一条目标仅为组带 $3(i+1)$ 上节点的任意播（anycast）核心请求消息。处于组带 $3(i+1)$ 上的任意节点在收到该任意播消息后于一个随机等待时延后作出回复。其中，核心节点的时延设置较小，这样能够重复利用已经选出的核心节点。

此外，组带 $3i+2$ 若恰好处于网络边界时会出现边界条件，那么就将其候选核心节点的组带设为 $3i$。该边界条件可以在没有收到任何对上述任意播核心请求消息回应时检测到。

2. 两阶段丢失恢复

（1）核心节点丢失的恢复

1）丢失检测：当核心节点收到一个无序分组时，可以推断出发生了分组丢失，如果它在查看 A-map 后发现上游核心节点有本地丢失的分组，那么向其发送恢复请求。

2）丢失恢复：当上游核心节点收到来自下游核心节点的单播重传请求后，进行丢失分组的单播重传。图 5-18 说明了处于组带 $3i$ 和组带 $3(i+1)$ 上的节点之间的丢失检测和丢失恢复。如果任意一个在单播请求路径上的非核心节点有被请求的分组，那么它中断发送并且重传所请求的数据包。

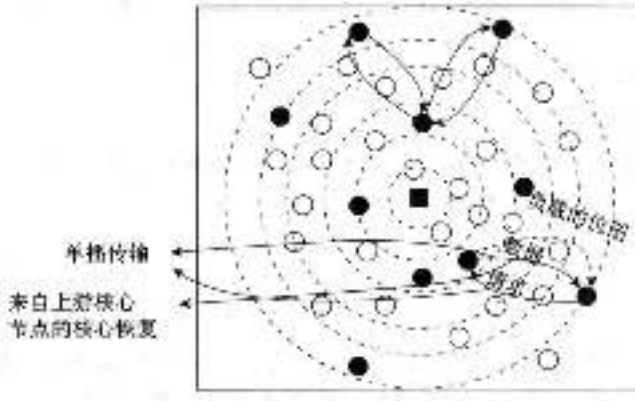


图 5-18　GARUDA 中核心节点的丢失恢复


奇思妙想

GARUDA 协议构造组带（band）结构来选举产生核心节点和非核心节点，这是一个非常巧妙的想法。研究者受“投石入静水后会产生涟漪荡漾成圈”现象的启发，提出了一种有意思的网络逻辑拓扑结构——“同心环状网”。也就是说，当有节点在全网范围内广播一个消息时，网络就可以形成以该节点为圆心的同心环状网。同心环状网的概念与 GARUDA 协议中的组带非常类似。但是在实际网络中构造出同心环/组并不是一件容易的任务，需要考虑诸如广播
195 时间、跳数、邻居节点间通信冲突等因素。

（2）非核心节点丢失的恢复

非核心节点监听其核心节点的所有发送和重传，一旦发现其核心节点发送的 A-map 中全部

位被置 1，则立即进入非核心节点恢复阶段，启动对该核心节点的重传请求。如果非核心节点在核心节点存在定时周期内没有监听到其核心节点的发送或重传，则显式地向核心节点发送重传请求，核心节点利用其当前 A-map 做出响应。因为核心节点的所有重传均被非核心节点监听到，所以排除了相同丢失分组的冗余重传。

问题与练习

5.1　多项选择题

（1）下列哪一项不属于传输层的任务？（　　）

A. 端节点到目的节点的可靠传输　B. 网络拥塞检测　C. 网络拥塞避免　D. 缓冲区管理

（2）为什么无线传感器网络中不能采用 TCP？（　　）

A. TCP 在传感器中使用时开销过大

B. 在无线传输导一跳中误码错误

C. TCP 能耗较大

D. A 和 B

（3）下列哪些项属于 PSFQ 的特征？（　　）

A. 分发数据慢　B. 恢复数据快　C. 逐跳错误恢复　D. 以上所有选项

（4）如果单跳会造成 10% 的无线丢失率，那么 5 跳连接的丢失率为（　　）。

A. 40%　B. 5%　C. 10^{-5}　D. 0.2

（5）PSFQ 协议没有以下哪些功能？（　　）

A. 数据分发　B. 错误恢复（提取）

C. 端到端重传和定时器设置　D. 状态报告

（6）ESRT 没有以下哪些特点？（　　）

A. 可以实现汇聚节点到传感器节点的可靠性。

B. 根据可靠性需要调整节点的报告速率。

C. 其目标是达到 OOR 状态。

D. 如果在（C，LR）状态，那么需要快速地降低报告速率。

（7）E^2SRT 在以下哪些方面提升了 ESRT 的性能？（　　）

A. 当所需可靠性超过了当前网络设置的能力时，网络不会收敛到规格化可靠性为 1 的 OOR 状态。

B. 网络在（C，LR）和（NC，LR）之间跳转。

C. 在趋近 OOR 期间，节省了大量的发送能耗。

D. A 和 B

（8）CODA 有以下哪些特点？（　　）

A. CODA 是为了实现可靠性。

B. CODA 实现了拥塞的降低。

C. CODA 实现了可靠性和拥塞的避免。

D. 以上都不正确

（9）STCP 没有以下哪些特点？（　　）

A. STCP 是在传感器节点中实现的。

B. STCP 是一个无线传感器网络中通用的、可扩展的和可靠的传输层协议范式。

C. STCP 提供了可靠性和拥塞控制。

D. STCP 主要在基站运行。

（10）GARUDA 有以下哪些特点？（　　）

A. 实现了汇聚节点到传感器节点的可靠传输。

B. 采用了支配集的概念来构造核心。

196

C. 对核心节点和非核心节点采用不同的数据恢复方式。

D. 以上所有选项

5.2　试解释为什么 TCP 不适用于无线传感器网络?

5.3　试解释在分组丢失的情况下，PSFQ 协议中每个节点如何设置重传定时器?

5.4　为什么 ESRT 提出了“状态”的概念，而且把 OOR 状态作为目标?

5.5　除了 ESRT 中采用的公式，是否有其他的函数能够实现相似的速率?

5.6　ESRT 如何检测拥塞?

197　5.7　E^2SRT 对 ESRT 做了哪些改进?

5.8　描述 GARUDA 如何形成核心节点。

第四部分

Wireless Sensor Networks: Principles and Practice

计算机科学原理

传感器节点的操作系统

虽然操作系统是一个典型的计算机科学研究领域，但设计传感器节点硬件的工程师们也应了解无线传感器网络操作系统的特征，因为一个成功的无线传感器网络系统需要硬件和软件的紧密结合。例如，如果一个无线传感器网络操作系统有一套中断指令，那么如何设计这些用于协调微处理器和模拟传感器的指令？如果一个操作系统具有唤醒命令，那么当需要发送数据时，又如何设计一个用来触发无线收发器的唤醒电路？本章将介绍一些最常用的无线传感器网络操作系统（如 TinyOS）。

6.1 TinyOS

TinyOS 是美国加州大学伯克利分校的研究者专为无线传感器网络设计开发的开放源代码操作系统［Levis06］。由于传感器节点资源严重受限，因此 TinyOS 被设计为微型（小于 400 字节）、灵活的具有可重用组件的操作系统，这些可编程组件能够集成到特定应用的系统中［Levis06］。TinyOS 是事件驱动型操作系统，也就是说，它定义了一套能够被异步的传感器网络中事件（如火警）触发的功能。TinyOS 是通过语法与普通 C 语言类似的 NesC 语言实现的。



既然 TinyOS 是一个操作系统，它就要具有操作系统的一些基本功能，如管理文件、为应用程序分配内存和回收闲置的 CPU 资源等。TinyOS 与其他操作系统的不同之处在于：它能满足无线传感器网络的内存容量小和 CPU 速度慢的特点；同时，它也要设法使能耗最小化以延长电池寿命。

6.1.1 概述

任何 TinyOS 程序都能够用一个软件组件图来表示。每个**组件**（component）都是一个独立的计算实体。不同组件之间具有接口，以保证它们能够彼此访问。

从计算的角度看，组件分为三个抽象层次：**命令**（command）、**事件**（event）和**任务**（task）。命令和事件是组件间通信（如两个组件进行通信）的机制，而任务则是用于实现组件内部（在一个组件内）的并发性。

当一个组件请求另一个组件执行某项操作（即服务）时触发一个“命令”。例如，一个软件实体可以请求传感器报告当前的监测数据。

“事件”是一个特殊的软件实体，在以下三种情况下产生：1）命令执行完毕后，产生事件消息，标识该服务完成；2）传感器硬件遇到特别事件（如无线收发器的唤醒）时，产生一个硬件中断，标识这个新事件；3）当网络消息到达一个传感器或基站时，产生一个异步信号，通知相关事件，标识这个新事件的产生。

从传统操作系统角度来看，命令类似于“下行呼叫”，而事件可视为“上行呼叫”。事件和命令不能相互阻塞。它们发生在不同的时间段。例如，TinyOS 用一个时间段发出服务请求（即发出命令），用另一个时间段发出完成信号（即产生相应的事件）。这两个时间段是分离的。命令可立即返回，而事件信号的完成则要滞后。

为什么要用到“任务”这个组件层次？在很多情况下，不可能在一个命令或事件处理程

序中立即完成所有操作。当这些操作需要用到多个传感器硬件资源（如无线收发器、模拟传感器、闪存等）时，更是如此。因此，命令和事件处理程序可以产生一个任务——一个由 TinyOS 调度程序稍后执行的功能函数。既然任务是稍后执行的，那么相比而言，命令和事件的响应显得非常迅速，它们可立即返回结果，从而将内部需要延时的扩展计算交给任务。 202

虽然任务用于执行重要的计算，但它不能无限制运行。"运行—完成"是它的基本执行模式。因此，相对于线程而言，任务的层级更轻，它描述了一个组件内的并发性，并且仅能访问这个组件内的资源（也就是说，一个任务在执行期间不能访问两个组件）。标准的 TinyOS 任务调度程序使用的是非抢占式的先入先出的调度策略。

关于组件要说明的是，在 TinyOS 中，所有硬件资源都用组件来表示。例如，一个组件收到 getData( ) 命令后，过一段时间，只要有硬件中断触发它，就会发出一个 dataReady( ) 事件。

TinyOS 已经为无线传感器网络程序员定义了许多组件。应用程序员可以编写组件来构造一个应用程序。然后，将这些组件与 TinyOS 组件连接起来以实现所需服务。下面将进一步介绍组件模型。

6.1.2 组件模型

组件封装了一套特定的由接口指定的服务集合。不仅无线传感器网络应用程序由一系列组件组成，TinyOS 本身也包含了一套与任务调度程序相适应的可重用的系统组件。

一个应用程序可以通过连接配置文件和一系列组件连接起来。连接配置文件定义了应用程序使用的一套完整的组件。具体的组件实现是独立于连接配置文件的。

通过对整个程序的分析和代码内联，TinyOS 编译器能够去除一些不必要的组件。在不同组件边界进行代码内联的操作可以改进程序的大小和效率。

下面说明关于接口（interface）的概念。如图 6-1 所示，任何组件都可以有两种类型的接口：1）它提供的接口；2）它使用的接口。通过这些接口，组件能够直接和其他组件相互作用。只要一个组件给每个实例单独的名字，那么这个组件可以多次使用或者提供同样的接口类型。

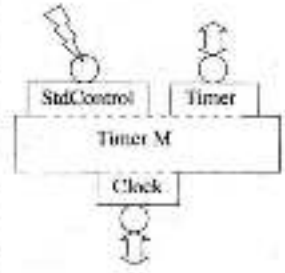


图 6-1 TimerM 组件的说明和图形描述

一个组件使用一个接口来表示专门的服务（如发送消息）。在图 6-1 中，组件 TimerM 一共有三个接口：1）它提供了 StdControl 和 Timer 两个接口；2）它使用了一个 Clock 接口。在图 6-1 中，提供的接口放在组件 TimerM 的上面，使用的接口放在组件 TimerM 的下面。双向箭头描述了命令和事件，闪电状的箭头描述了命令。

所有接口的细节如图 6-2 所示，接口是双向的，包含**命令**（command）和**事件**（event）。接口的提供者实现命令的功能，接口的使用者实现事件的功能。例如，Timer 接口（见图 6-2）定义了"start"与"stop"两个命令和一个"fire"事件。

注意：在这个例子中，没有使用两个单独的接口（一个用来表示命令，另一个用来表示事件）来表示 Timer 接口和它的使用者之间的相互作用。这是因为把命令和事件放在同一个接口里会使说明更加简单，并且有助于减少把组件连接起来所带来的错误。

TinyOS 用 NesC 语言实现。NesC 语言里面的组件有两种类型：**模块**（module）和**配件**（configuration）。

模块被用于调用或者执行命令。一个模块可以声明私有状态变量和数据缓冲区。

```
interface StdControl {
command result_t init();
command result_t start();
command result_t stop();
}
interface Timer {
command result_t start(char type, uint32_t interval);
command result_t stop();
event result_t fired();
}
interface Clock {
command result_t setRate(char interval, char scale);
event result_t fire();
}
interface SendMsg {
command result_t send(uint16_t address, uint8_t length, TOS_MsgPtr
msg);
event result_t sendDone(TOS_MsgPtr msg, result_t success);
}
```

图 6-2 TinyOS 接口类型实例

配件使用接口把其他组件连接起来。图 6-3 对 TinyOS 定时器服务做了定义。这个定时器服务是基于配件 TimerC 实现的。这个配件把定时器模块 TimerM 和硬件时钟组件 HWClock 连接起来。配件可以把多个组件聚集成一个**宏组件**（macro component），这个宏组件表示了一个接口集。

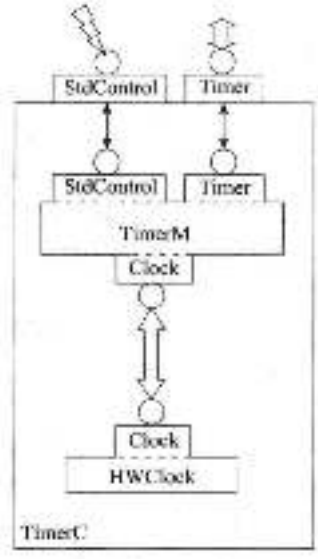


图 6-3 TinyOS 的定时器服务：TimerC 配件

组件使用它的接口（称为接口命名空间，interface namespace）来指明它所使用的命令和事件。配件通过把不同接口的本地名称连接在一起来将不同接口连接起来。换言之，组件可以调用一个接口而无需显式地声明该接口的实现。这将使在组件图中引入一个使用相同接口的新组件变得非常容易。

一个接口可以多次连接到其他接口。图 6-4 描述了一个实例，Main 的 StdControl 接口连接到 Photo、TimerC 和 Multihop。

带参数的接口：在一个组件中，带参数的接口用于导出相同接口的多个实例，这些实例用不同的标识符（一般是一个小整数）来表示。例如，在图 6-1 中，接口 Timer 是一个使用 8 位 id 的参数化接口，8 位 id 是一个额外参数。这样一个参数化的接口使得一个 Timer 组件能够实现多个独立的定时器接口，分别用于每个客户组件。由于 ID 的选择是唯一的，当每次需要一个标识符时，将会使用一个特殊的唯一关键字。

2003
1
2005

现在我们来看如何用 NesC 语言构建一个 TinyOS 应用程序。首先，使用 NesC 构建一个顶层配件；然后，在配件里定义不同的接口，将所有需要的组件连接起来。图 6-4 展示了一个名为 SurgeC 的应用程序。这个应用程序由以下几个组件组成：Main、Photo、TimerC、Multihop 和 SurgeM。这样一个应用程序周期性的（TimerC）读取光传感器的读数（Photo），通过多跳路由（Multihop）把它们发送回基站。

NesC 是基于 C 语言的语法和执行机制。不过，它和 C 语言有以下两方面的不同。1）NesC 不使用函数指针。NesC 编译器了解一个程序的精确调用图。这样一个调用图能够实现组件之

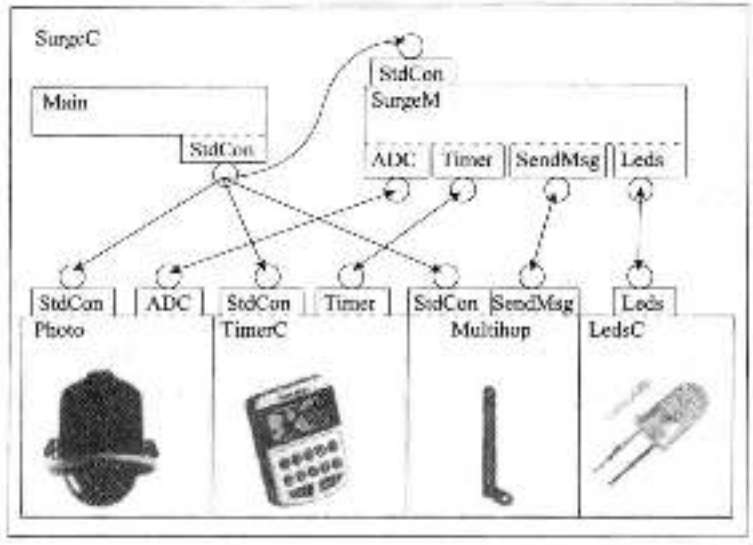


图 6-4　Surge 应用程序的顶层配件

间的优化，这种优化能够消除模块之间调用的开销。2）NesC 不支持动态内存分配。NesC 静态地声明程序的所有状态。这种方案有利于预防运行时间分配故障时产生的内存碎片。

6.1.3　执行模块与并发性

一个无线传感器网络能够产生许多事件，例如，异常传感器数据监测、低电量报警和传感器的休眠/唤醒等。无线传感器网络的以事件为中心的特性要求事件处理需要具备精细粒度的并发性。事件会在任何时间发生，我们应该怎样处理这些事件？有两种方法可以实现：使用像 Windows 那样传统的操作系统，对到来的事件进行原子操作后放入队列中，在合适的时间再去执行；另一种方法就是在主动消息（Active Message，AM）的方式下，立即执行事件处理程序。 206

对无线传感器网络而言，许多事件都是重要的（例如一个需要立即引起注意的检测事件），所以第二种方法对于无线传感器网络而言更加合适。然而 TinyOS 执行模块的核心由从运行到完成不间断计算的任务组成，事件处理程序是通过硬件异步控制的。

NesC 把任务定义为显式的实体。当执行程序的时候，任务被发送到任务调度程序等待执行。调度程序按照某一时间顺序（例如先进先出）执行任务。任务的执行必须遵循从运行到完成的原则，换言之，一旦任务执行，就必须完成这次执行。可以使用“原子性”表示任务从运行到完成的本质。然而，如果遇到中断处理程序或者当程序响应中断调用的事件或命令时，任务就不是原子性的。

TinyOS 定义了下述同步代码（SC）和异步代码（AC）：1）SC——仅在任务中实现；2）AC——至少在中断处理程序中可以实现。组件中通常既有 AC 代码也有 SC 代码。TinyOS 可以让程序员构建响应速度快、并发的数据结构，这些数据结构可以安全地在 SC 与 AC 之间共享数据。在不同任务之间，TinyOS 使用非抢占式的策略来减少消除竞争（即 CPU 资源之间的竞争）。然而，和 AC 和 SC 之间的竞争一样，在 SC 和 AC 之间也存在潜在竞争。

通常，在 AC 里有对共享状态的更新时，可能会发生数据竞争。在这种情况下，应该如何保证“原子性”呢？有两种方法可避免竞争：1）把所有冲突的代码转换成任务（代码仅指 SC）；2）采用原子片段（atomic section）更新共享状态。原子片段是能够保证执行原子性的一

段小的代码序列。在任何一个原子片段中都不能使用任何循环，也不能运行任何中断。

概括地说，可以使用如下方法确保无竞争的程序执行：

无竞争（Race-free invariant）：对共享状态的任何更新仅在 SC 里或者在原子片段里实现。NesC 确保在编译期间满足上述无竞争。换言之，NesC 编译器能够预防几乎所有数据的竞争。

出于以下原因，在任何程序中都应避免数据竞争。

数据竞争会导致一系列不确定的程序错误。如果无竞争，组件组装基本可以忽略并发性。换言之，由于在编译期间编译器能够捕获任何共享异常，因此组件组装无需关心哪个组件产生
207 并发性或者组件之间怎样连接在一起。

强大的编译期间分析能力使得大量不同的并发数据结构和同步原语都能被采用。即使更新共享状态，NesC 也有一些并发队列变量和状态机能够直接在事件处理程序中方便地处理与时间相关的活动。在整个数据包到达之前，NesC 一直在中断处理程序中处理无线通信事件。在接收到数据包后，中断处理程序会发布一个任务。

6.1.4　主动消息

我们要考虑一个重要的问题是，即 TinyOS 怎样处理传感器之间的无线通信？于是，主动消息（Active Message，AM）的概念便成为 TinyOS 通信抽象的核心［TVon92］。一个主动消息是带有一字节处理程序 ID 的小数据包，它是一个长度仅为 36 字节的短数据包。当传感器收到一个主动消息，它会立即将这个消息（使用一个事件）分发给一个或多个处理程序。这些处理程序会注册以接收这些消息。处理程序注册（handler registration）是通过静态连接和一个参数化接口来实现的。

TinyOS 使用主动消息接口实现不可靠的、单跳的数据报协议。主动消息接口也给无线射频装置和内嵌的串行端口（用于连接的节点，例如基站）提供一个非统一的通信接口。

多跳的、可靠通信可以通过在主动消息接口之上的更高层协议来实现。主动消息交换是事件驱动的，可以把本地 CPU 计算和无线通信紧密结合起来。

6.1.5　实现状况

到目前为止，TinyOS 已经应用在许多硬件平台上，适用于许多公司的传感器产品。和细粒度的仿真环境类似，可通过添加可视化、调试和支持工具，对 TinyOS 环境进行扩展。

将 TinyOS 安装在传感器节点和台式机、笔记本电脑、掌上电脑之类的机器上，便可以在传感器网络和互联网之间构建代理。这个代理使得传感器网络能与使用 Java、C 或 MATLAB 实现的服务器端工具进行整合。TinyOS 也可以构建与数据库引擎（如 PostgreSQL）的软件接口。

6.1.6　主要特性

绝对尺寸：事实上，TinyOS 是一个微型操作系统，一个基本的 TinyOS 环境仅需大约 400
208 字节。如果包含一个相关联的 C 进行时元讲（例如浮点型函数库），TinyOS 仅使用 1KB 的空间。如果增加一些基于 NesC 的应用程序，大部分情况下，仅需不超过 16KB 的空间。但对于一些大型的 TinyOS 应用程序，例如 TinyDB，64KB 的空间也已经足够了。

资源占用优化：除了采用标准的技术（例如去除符号表等）减小代码规模，TinyOS 也使用全程编译技术（whole-program compilation）来去除无用代码。跨组件的优化能够去除冗余的操作和模块间的开销。

事实上，NesC 通过全程代码分析移除边界通道，通过跨组件优化（这些优化包括常量传

用、共同的子表达式的消除）来优化整个调用路径。这种全程代码的优化比未经优化的代码和在 NesC 语言之前手写的原始 C 代码规模更小，运行速度更快。

软硬件透明性：TinyOS 使用灵活的组件模型，从而方便地移动硬件和软件边界。例如，组件会产生两类事件：软件回调和硬件中断。

6.1.7　低功率优化

TinyOS 有一系列减少能量损耗的特性。例如，**分相操作**（split-phase operation）和**事件驱动执行模型**（event-driven execution model）能够避免轮转复和重量级并发（例如线程）。TinyOS 使用这些操作来减少能量损耗。当任务队列为空，TinyOS 调度程序控制微处理器进入低功耗的睡眠模式，这种睡眠模式能够进一步减少能量消耗。

6.2　LA-TinyOS：无线传感器网络中的一种局部性感知的操作系统

LA-TinyOS（Locality-Aware TinyOS，局部性感知操作系统）是一种新型无线传感器网络操作系统［Huang07］。该系统利用事件产生的局部性（locality）特点来提高事件感知能力，同时减少能量损耗。无线传感器网络的局部性包括：时间（temporal）局部性和空间（spatial）局部性。其定义如下所示。

时间局部性：当无线传感器网络系统发生待感知事件时，该事件很可能会在它第一次出现后的有限时间内再次被感知到，这种现象称为时间局部性。

空间局部性：如果感知事件是由一个穿过传感器网络的移动物体引起的，该感知事件有可能被邻居区域内的节点再次感知。这种现象称为空间局部性。

209



奇思妙想

如果你对计算机系统结构课程还有印象，回忆一下，高速缓存的设计也使用了同样的原则：1）基于时间局部性，如果一个指令在某个时间被使用，在不久的将来有可能会被再次使用。2）基于空间局部性，如果一个指令被选定执行，它邻近的指令也有可能被执行。因此，高速缓存通常存储局部感知指令来加快 CPU 的执行速度。

有时，时间局部性和空间局部性会在同一个感知事件中发生。例如，在一个环境监测应用中，传感器监测到一个入侵者。这种入侵事件有可能被同一个节点在一段时间内持续感知，也就出现了时间局部性。如果入侵者在附近活动，入侵事件很快就会被邻居节点感知，也就是出现了空间局部性。

传感器通常使用任务管理器来监测事件。任务管理器周期性地主动监测事件。由于任务管理器通常监测异常事件，所以监测周期会非常长。

监测周期越长，传感器的能量损耗越少。然而，当一个异常事件发生时，由于时间和空间局部性，需要提高任务管理器的触发频率来更加紧密地进行感知事件监测，这就需要采用一个更短的感知周期。如果是一个局部感知的任务管理器，也就是说，基于时间和空间局部性，任务管理器能自动地调整感知周期，那将会非常适合。

遗憾的是，大多数无线传感器网络操作系统没有提供内核级的支持来促进局部感知任务的开发。因此，大多数无线传感器网络应用没有实现局部感知任务，或是在用户状态构造这些任务（也就是说，没有在操作系统内实现）。这种用户状态通常是易于出错、低效率和冗余的。

局部性感知操作系统通过考虑实现局部感知任务来改善 TinyOS，即把 LocalityM 组件添加

到 TinyOS 来实现的。LocalityM 为编程者提供了一个叫做 LocalityControl 的接口，以配置它们的局部元素。LocalityM 维护了一个数据结构（如表 6-1 所示）。这种数据结构用于记录所有的局部配置。这个表被称为局部配置表。

```
registerEvent (string EventName);
configureLocality (event_table_entry_T e,
uint8 TimerID,
uint32 GracefulLength,
uint8 HopCount,
(void _ ) FuncEnter,
(void _ ) FuncLeave);
triggerEvent (string EventName);
```

表 6-1 LA-TinyOS 操作系统局部配置表

事件	计时器 ID	理解长度	任务	跳数	自适应函数
"A"	1	2000	dataTask	2	Enter() /leave()
"B"	2	1000	getMax	1	Null/Null

在上面的 3 个命令中，registerEvent 在局部配置表中注册了一个新条目，configureLocality 指定了局部配置，当系统监测到感知事件进入它的局部范围时调用 triggerEvent。

一个使用局部配置数据结构和命令的示例代码如图 6-5 所示。它是 LA-TinyOS 系统中支持局部感知功能的 Oscilloscope 组件。

```
1: implementation
2: {
3:     command result_t StdControl.start(){
4:     event_table_entry_T*e;
5:     call SensorControl.start();
6:     call Timer.start(TIMER_REPEAT, 1000);
7:     ..
8:     e = call LocalityControl.registerEvent("A");
9:     call LocalityControl.configureLocality(e, 1, 2000, 10, 2, enter1,
       leave1);
11:    reg dataTask() "A";
12:    }
13:    async event result_t ADC.dataReady(uint16_t data){
14:      if (data>0x03B0){ // an anomaly
15:        call LocalityControl.triggerEvent("A");
16:    }
17:    pack->data[packetReadingNumber] = data;
18:    post dataTask();
19:    }
20: }
```

图 6-5 LA-TinyOS 中支持局部感知功能的 Oscilloscope 组件的部分代码

在第 8 行中，可以看到一个被命名为 A 的事件在局部配置表中注册。在第 9 行中，调用 configureLocality 来指定这个事件的局部配置。在第 11 行中，reg 运算符把 dataTask 和这个事件关联起来。如果我们回头看表 6-1，可以看到第一行就给出了这个事件的局部配置。

在第 14 行中，可以看到当传感数据大于一个指定的阈值（0x03B0）时会检测到一个感知事件，并且事件 A 被触发进入它的局部（第 15 行）。

在表6-1的最后一列（即“自适应函数”列），enter1和leave1是指向这个程序提供的自适应函数的指针。它的意思是当事件A被监测到，LA-TinyOS系统会执行enter1进入它的局部，然后执行leave1离开它的局部。

6.2.1 改变定时器以支持时间和空间局部性

现在让我们来看看LA-TinyOS系统是怎样基于局部配置更新定时器的。如表6-2所示，组件TimerM维护了一系列软件定时器。定时器ID告诉我们它是一次性定时器（也就是说，定时器到时之后会终止）还是周期性定时器，它缺省的定时器时长（计数器的值），以及在定时器到时之前的剩余时间。

表6-2 组件TimerM中的软件定时器列表

计时器ID	类型	状态	周期	过期时间
0	ONE SHOT	On	300	240
1	REPEAT	On	1500	360
2	REPEAT	Off	500	450

如果一个定时器中断被触发，HWClock的中断处理程序将减少每个软件定时器的Time-to-Expired字段的值。当Time-to-Expired值为0，意味着定时器到时，然后执行一个相应的处理程序。

当一个事件进入局部时，LA-TinyOS系统使用下面的数据结构改变软件定时器的周期：

```
setLocalityTimer (uint8_t TimerID,
uint32_t ReducedPeriod);
```

在上面的数据结构中，LA-TinyOS系统计算减少的周期，以表示其对时间局部性的适应。通过搜索局部配置表，能够很容易地找到TimerID。缺省的周期保存在内核数据结构中。当一个事件离开它的局部时，setLocalityTimer再次被用来重新设置周期。

211
212

在表6-1中，第二列显示了描述每个事件的理想长度。每当检测到感知事件时，计数器就重新设置为它的全值。当一个事件进入局部区域时，LA-TinyOS系统在每次定时器中断时减少局部性程度计数器。

最终局部性程度计数器值为0。然后与它相关联的软件定时器（如表6-2第4列所示）重新设置为缺省的值，表明这个事件正要脱离局部性。

上面所描述的是时间局部性的例子。LA-TinyOS系统怎样实现空间局部性？它通过广播报警消息来实现。广播的跳数定义了报警区域。在局部配置表中（见表6-1），跳数也是可用的。当一个传感器节点接收到一个警告消息时，它激活一个相应的感知事件来进入它的局部区域。

6.2.2 多级任务调度器

在6.1节提到过，TinyOS使用非抢占式的先进先出的**调度器**（non-preemptive FIFO scheduler）。现在问题是：这样一个简单的调度器无法从定期的、非急切的常规任务中区分出与感知事件相关联的紧急任务。

为了解决这个问题，在不改变TinyOS非抢占式调度的前提下，LA-TinyOS系统提出了一种三级调度器，如下所示：

第一级：当传感器节点检测到一个感知事件时，它会在局部配置表中注册相关联的任务。这些任务放在第一级的队列中，并且最先被调度执行。

第二级：对于空间局部性，任务和通过告警消息触发的进入局部区域的事件相关联。这些

任务放在第二级的先进先出队列中。

第三级：当第一级和第二级的任务没有出现时，非急切的正常任务在第三级的先进先出的调度器中被服务。

上面介绍的三级先进先出调度器能够确保 LA-TinyOS 系统根据任务的重要性来执行。


奇思妙想

多级分层管理念已被用于解决许多问题。它的基本思想是避免平面（也就是说只有一种级别的）拓扑结构。在平面拓扑结构中，所有的节点都被认为是处于相同的情况。通过区分不同级别，获得了处理不同的优先权的灵活性。

213

6.2.3　LA-TinyOS 系统的代码结构

LA-TinyOS 系统的代码结构如图 6.6 所示。可以看出，通过增加一个 LocalityM 模块和一个分层的调度器，LA-TinyOS 系统增强了 TinyOS 的性能。

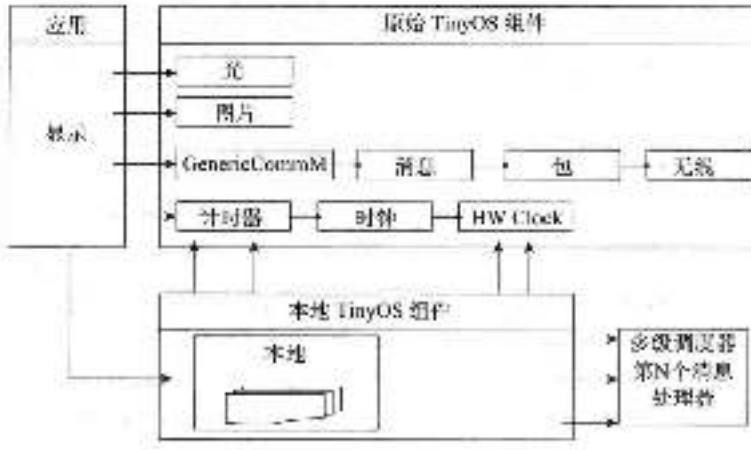


图 6.6　LA-TinyOS 系统的代码结构

之前讨论过，局部性配置表用于注册和配置由时间局部性或空间局部性引起的事件，并且当任务进入和离开局部区域时，LocalityM 组件轮流地使用原始的 TinyOS 内核组件自动地调整任务的检测周期。

如果没有 LocalityM，程序员仍然可以使用原始的 TinyOS 组件（如图 6-7 所示）对一个局部感知的应用程序进行编程。

现在总结一下使用 LA-TinyOS 系统处理局部感知应用的优点。

首先，LA-TinyOS 系统内核组件覆盖了所有的局部感知代码。相比原始的 TinyOS 实现，内核执行更加可靠。

其次，由于程序开发人员仅仅需要在初始化阶段注册一个局部事件，因此 LA-TinyOS 系统允许程序员很容易地编程局部感知事件，然后在检测到感知事件时利用一个方法调用进入它的局部区域。

214　最后，当发生的局部事件多于一个的时候，程序员可以使用 LocalityM 处理所有事件的局部感知代码。因此对于多重局部性事件，LA-TinyOS 系统提供了更加高效的实现。相比之下，TinyOS 实现需要对每个事件提供冗余的局部感知代码。

6.3　SOS

文献［Han05］中提出了 TinyOS 的另一个改进版本——SOS。它表明在不损失大量的能量或性能的前提下，无线传感器网络操作系统能够实现动态的和通用的操作系统语义。下面介绍它的主要特性。

SOS 有一个共同的内核和动态应用模块。这些模块在程序运行的时候被加载或卸载。模块使用系统跳转表发送消息并与内核通信。模块也能为其他要调用的模块注册函数接入点。

SOS 不涉及内存保护，这和 TinyOS 类似。不过，它能避免常见的错误。这是优于 TinyOS 的一个方面。在 SOS 中，动态存储器用于应用模块和内核中。由于减少了复杂性并提高了临时内存的重用性，从而使得编程更加容易。

与使用二级任务调度器的 LA-TinyOS 系统类似，SOS 系统也提出使用优先调度器把当前运行的进程从中断上下文中移出来以保证对时序要求严格的任务的实时性要求。

SOS 内核拥有动态链接模块、灵活的优先调度和一个简单的动态内存子系统。SOS 内核服务提供了一个更高层的应用程序接口，将程序员从管理底层服务或者重新实现常用的抽象中解放出来。

6.3.1　模块

SOS 程序使用模块（位置无关的二进制文件）实现特定的任务或函数。SOS 由多个相互作用的模块组成。从功能的角度来看，模块和 TinyOS 中组件的概念类似。SOS 程序员主要开发包括驱动、协议和在模块层使用的应用组件。

在 SOS 中维护模块性和安全性使之不因模块间的松耦合带来很高的代码开销，是具有挑战性的。所有的 SOS 模块是自包含的并与位置无关。这些模块使用消息和功能接口来维护模块性。除非底层硬件或者资源管理功能需要改变，否则大多数的应用程序不需要修改 SOS 内核。

1. 模块结构

图 6-7 展示了 SOS 模块之间的交互关系。我们已经看到，SOS 实现了一个定义完整并且优化的带有入口和出口的模块，这样一类模块组成一个模块化结构，从而维护 SOS 系统的模块性。执行流要么从调度器传送来的消息进入模块，要么从注册函数（外部使用）进入模块。

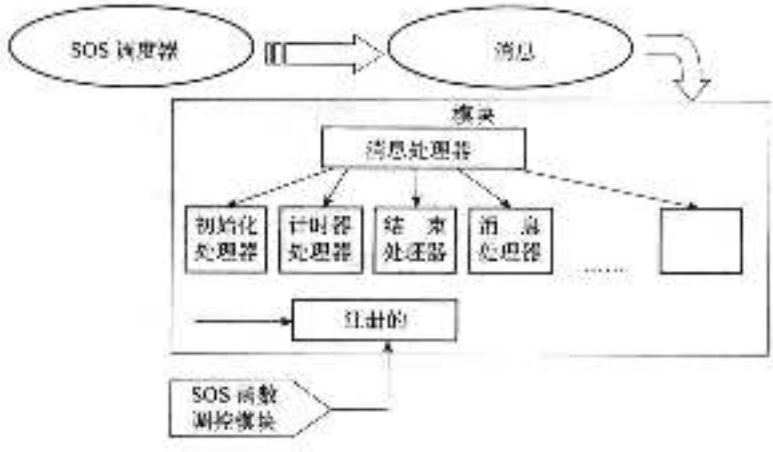


图 6-7　SOS 系统模块交互关系

一个特定模块处理器用于处理模块之间的消息。一个处理器接受两个参数：被传送的消息和模块的状态。

当插入一个模块时，SOS 内核产生初始化消息。初始化消息的处理器设置模块的初始化状态，其中包括初始的周期定时器、函数注册和函数订阅。

当移除一个模块时，SOS 内核产生结束消息。结束消息处理器释放所有的传感器资源，包括定时器、内存和注册的函数。

除了上面所说的初始化消息和结束消息，还有其他专门的模块消息，包括定时器触发程序处理结果、传感器读数和来自其他模块或节点的输入数据消息。

SOS 异步地处理消息（即用队列存储消息）。和 TinyOS 相似，SOS 主调度环从优先队列取出消息，然后把消息传送给目的模块的消息处理程序。

215
-
216

特定模块的操作需要同步执行。在这些模块之间，SOS 使用直接的函数调用。函数的注册和订阅过程执行这些直接的函数调用。

RAM 用于存储模块的状态。在内存中，模块重新定位。模块间函数的位置可以通过注册过程获得。

2. 模块交互

消息用于实现模块间的交互。模块之间通过消息进行异步通信。消息也能中断执行而转向预定的子部分，这些子部分存储在一个执行顺序预先计划好的队列中。

尽管上面的消息机制很灵活，但是执行速度缓慢。因此，SOS 提供对模块注册函数的直接调用。这种直接的函数调用绕过调度程序，从而提供模块之间更低延迟的通信。

SOS 使用函数注册和订阅实现直接的模块间通信和从内核到模块的函数调用。函数控制块（Function Control Block，FCB）用于存储已注册函数的关键信息。FCB 由 SOS 内核创建，由元组{模块 ID，函数 ID}进行索引。FCB 包括一个有效的标志、一个订阅者引用计数和原型信息。

模块 ID 和函数 ID 用于定位需要的 FCB，类型信息用于提供附加的安全级别。如果查找成功，内核将一个指针返回指向已订阅函数的指针。

需要访问内核函数的模块使用的跳转表如图 6-8 所示。这样一个跳转表也使得模块间保持和内核的松耦合性，而不依赖于特定的 SOS 内核版本。这也使得在升级内核时不必重新编译 SOS 模块。因此，同样的模块能够在异构的 SOS 内核调度中运行。

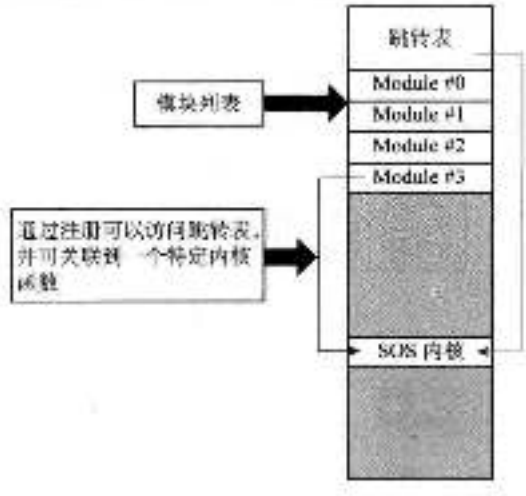


图 6-8　SOS 中跳转表的构成和函数关联

3. 模块插入和删除

模块插入（module insertion）：模块插入是一个用来保持侦听网络中新模块加入请求的分布式协议。这个通过网络发布请求和传播模块映像的分布式协议不依赖于SOS内核。当前SOS使用类似于面向移动设备应用平台（MOAP）的发布/订阅协议。当给一个模块的请求被协议捕获，协议需要检查这个模块是否是已经安装在节点上模块的更新版本。同时也需要检查节点是否对这个模块感兴趣，以及是否有这个模块需要的空闲程序内存。

为了验证上面的两个条件正确与否，分布式协议检查包头部的元数据（metadata）。这个元数据包括如下信息：1）模块的唯一标识，2）用于存储模块本地状态的内存大小，3）用于区分每个模块的版本信息。如果SOS内核发现没有足够的内存来存储模块的本地状态，协议将会终止模块的插入。

模块插入过程会产生一个由元数据唯一模块标识ID为索引的内核数据结构，这个数据结构用于存储待插入模块可执行程序的绝对地址。这个数据结构也存储指向包括模块状态的动态内存的指针以及模块的唯一标识。最后，通过调度一个模块的初始化消息，SOS内核调用模块的处理程序。

内核通过分派一个结束消息来启动模块删除（module removal）。这个消息控制模块适时地释放它占用的资源。这个消息也会通知依赖于已删除模块的其他模块。在分派结束消息后，内核释放如下所有资源来实现垃圾回收：动态分配的内存、定时器，传感器驱动和模块拥有的其他资源。在模块删除后，用FCB来确保平台的完整性。

217
~
218

6.3.2　动态内存

动态内存分配可以使用灵活的队列长度以适应最坏情况和一般任务的复杂程序讲义，例如沿着协议栈向下传送一个数据缓冲区。在SOS中，动态内存基于一个简单的、最适合的固定块内存分配。这种内存分配使用以下三个固定块尺寸。

最小的内存块用于包括消息头部在内的大多数SOS内存分配。稍大的内存块用于需要移动大的连续的内存的应用，例如模块插入。实际上，最大内存块是所有空闲块的链表，其大小为链表上所有空闲内存块的存储空间总和，可用于任何复杂的应用。

在SOS中，所有的数据结构（例如队列、列表等）在执行时间内动态地增加或减少。动态地使用和释放内存（即动态内存）创造了一个能有效重用临时内存的系统。动态内存也能动态调整特定环境和条件下的内存使用。

模块能够通过转移内存所有权来拒绝数据移动。SOS使用少量探测基本的连续内存超支情况数据来注解动态内存块。内存注解用于崩溃后的内存分析以识别可疑的内存占有者，例如占用大量系统内存的坏的模块或者溢出的内存块。SOS的内存注解也能保证内存卸载时的垃圾回收。

6.4　RETOS：弹性可扩展多线程操作系统

弹性可扩展多线程操作系统（Resilient, Expandable, and Threaded Operating System, RETOS）旨在为无线传感器网络节点提供一个具有容错性的、可重构的、资源利用率高的多线程操作系统[Hojung07]。图6-9给出了这个操作系统的完整架构。

事件驱动方法由于能够在资源受限的传感器节点内高效实现而被广泛应用于传感器操作系统。然而在RETOS中，应用程序开发者通过程序分离处理技术来明确地管理任务和事件的状态；RETOS明确地把应用程序从内核中分离出来。应用程序独立和动态地加载到系统中（内

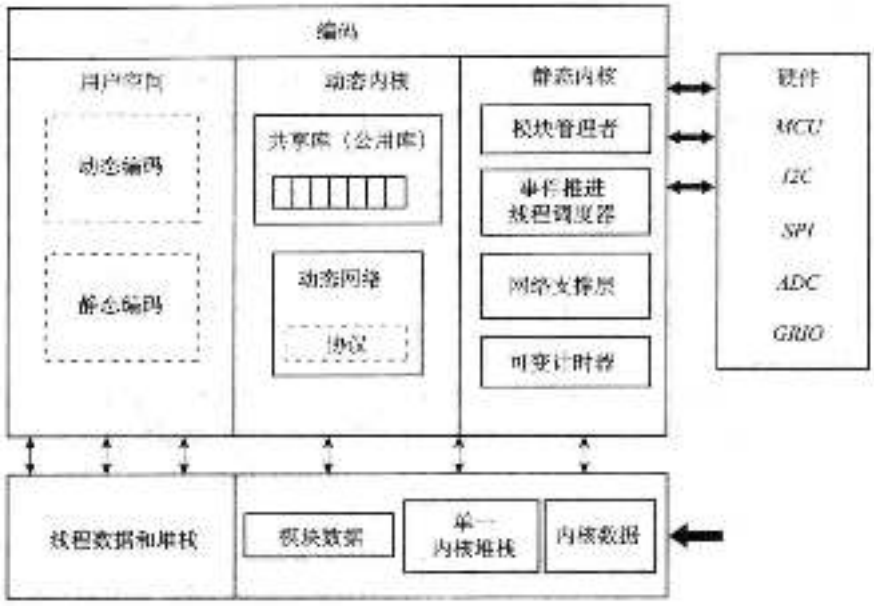


图 6-9　RETOS 系统架构

核模块也一样）。RETOS 使用可加载模块框架实现内核的重新配置。

6.4.1　应用代码检查

RETOS 使用一种称为应用代码检查（Application Code Checking，ACC）的软件技术实现静态和动态的代码检查。ACC 的目标是避免用户程序在合法的边界和直接的硬件操作之外访问内存。因此，ACC 经常检查机器指令的目的字段。为避免应用程序读内核或者其他应用程序的数据，ACC 也要检查指令的源字段。

ACC 使用静态代码检查来验证编译时的直接或者立即寻址指令和与计算机相关的跳转。（指令寻址方式的细节，请参考汇编语言课程。）

ACC 使用动态代码检查来验证执行时的间接寻址指令是否正确使用。返回指令也需要进行动态检查。

图 6-10 展示了构造可信代码的过程。应用源代码被编译成汇编码。然后编译器检查代码插入到动态代码需要检查的位置。

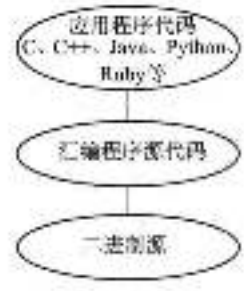


图 6-10　可信代码的产生

动态代码插入后，在二进制代码上执行静态代码检查。当编译器不能察觉某些应用程序错误时，这些错过的错误将会报告给内核。在接收到报告的错误后，内核通知非法指令地址的用户，并且安全地终止程序。

219
~
220

6.4.2　多线程系统

我们已经知道，TinyOS 是事件驱动型操作系统，TinyOS 程序员关心通过明确的并发控制能否达到程序的最佳执行。

然而，RETOS 采用了不同的方法（例如多线程技术），通过利用底层系统的抢占式和阻塞

输入输出特性实现高并发性。尽管多线程的方法很吸引人，但是在资源受限的传感器节点环境中，实现多线程是有挑战性的。在多线程的环境里，每个线程需要一个堆栈以维护状态变量。调度时间表用于实现两个堆栈之间的上下文切换。RETOS 认真地考虑了内存使用、能量消耗和调度效率。RETOS 已经分别实现了单一内核堆栈与堆栈大小分析、可变计时器和事件推迟线程调度器。

1. 最小化内存使用

RETOS 为内核提供了两种减少内存使用的技术：

1）**单一内核堆栈**（single kernel stack）用于减少所需要的线程堆栈的大小。这种机制把线程堆栈分成两种类型：**内核堆栈**（kernel stack）和**用户堆栈**（user stack）。RETOS 对内核堆栈执行严格的、受控的访问限制。这样能够保系统在内核模式（如线程抢占）下不会任意地交错执行流程。在线程抢占的情况下，硬件上下文被保存在每个基于内核堆栈共享的线程控制块中。

2）**堆栈大小分析**（stack-size analysis）用于给每个线程分配一个适当大小的堆栈。为了减少内存的使用，需要估计一个准确的线程堆栈大小。堆栈大小分析已经在 RETOS 中实现。RETOS能自动地为每个线程生成一个最小的、系统安全的堆栈。

2. 用可变计时器减少能量消耗

在多线程计算中需要消耗能量，这些消耗能量的操作包括：**定时器管理**（timer management）、**上下文切换**（context switching）和**调度操作**（scheduling operation）。

定时器管理：在多线程的系统中，从能量消耗的角度来看，可变定时器技术（而不是固定周期定时器）更加节能。系统定时器处理来自线程的定时器请求，然后系统定时器不依赖于当前运行的线程更新剩余时间。通过可变定时器，定时器中断间隔会再次被设定。这个中断间隔设置为当前运行线程时间片的最早超时时间。调度操作不会像事件驱动系统处理程序之间传递消息那样频繁。在大多数的无线传感器网络应用中，上下文切换的开销不是一个重要问题。 221

3. 事件感知线程调度

线程调度基于优先级感知的实时调度接口实现内核的动态优先级管理。调度 RETOS 线程有三种策略：SCHED_ RR、SCHED_ FIFO 和 SCHED_ OTHER。

事件感知的线程调度用于增加线程的事件响应时间。为了处理重要的事件，调度器直接提高这个处理特殊事件线程的优先级。当这个事件发生的时候，提高优先级的线程能够很快地抢占其他线程。

6.4.3 可加载内核模块

RETOS 支持动态应用加载。一种内存重定位机制用于支持动态应用加载。PIC（位置无关代码）方法不支持内存重定位。

一种内存重定位机制如图 6-11 所示。一个 RETOS 文件由一个通用部分和一个硬件相关部分组成。RETOS 文件格式有已编译代码。如果传感器使用 RETOS，那么传感器的微控制器要支持不同的寻址特性，例如重定位类型和相对内存访问指令。因此，这样的文件格式有专门的硬件信息帮助相应硬件进行重定位。 222

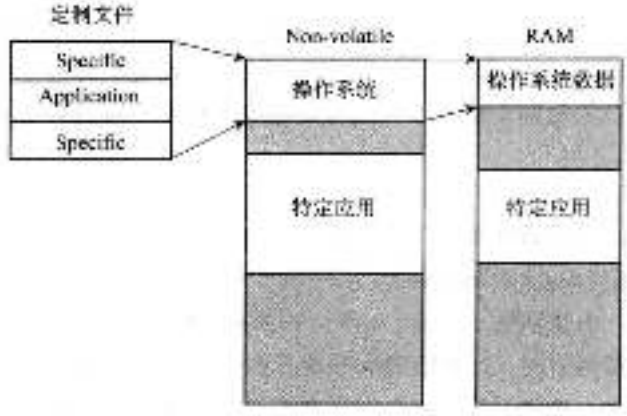


图 6-11　简单的 RETOS 重定位机制

问题与练习

6.1　与传统的操作系统（例如微软的 Windows）相比，TinyOS 有哪些特殊的性质？

6.2　解释 TinyOS 的架构。

6.3　在 TinyOS 的基础上，LA-TinyOS 系统在哪（些）方面做了增强？

223

6.4　解释 SOS 模块插入原理。

6.5　使用模块重定位对 RETOS 有什么好处？

无线传感器网络中的中间件设计

本章将介绍无线传感器网络中的中间件体系结构，探讨的内容来源于文献［MiamiaOS］，更多的细节请读者参阅该文献。

7.1 引言

通常，网络协议栈可被分为五层，自上而下依次为：应用层、传输层、路由层、MAC（Media Access Control，介质访问控制）层和物理层。例如，美国 Crossbow 公司的 motes（即传感器节点）支持用户使用 NesC（类似于 C 语言）来构建传感器网络控制程序。如图 7-1 所示，用户在应用层构建了这些程序来控制无线传感器网络的操作，例如相邻传感器节点间的数据聚合操作。此处请注意：应用层不处理无线传感器网络的路由问题，因为这是路由层的任务；同样，应用层也不处理应由传输层解决的网络拥塞问题。

虽然以上在应用层直接编写的程序可以完成很多无线传感器网络的数据处理和其他高层无线应用，但是对于程序员来说，构建这些应用层程序还是不够方便，原因如下：

1）大多数无线传感器网络系统都没有提供便捷的编程/编译工具。例如，在 TinyOS 环境下使用 NesC 进行编程需要经过长时间的学习，程序员需要学习很多不同的程序对象接口。此外，TinyOS 的安装和配置至今仍是一个难题。

2）更为重要的是，程序员需要熟悉很多无线传感器网络的内部操作细节，以便构建高效、易于使用的应用层程序。例如，为开发一个传感器数据查询软件（无线传感器网络应用层基本功能），程序员需要了解路由层的细节，因为需要借助于路由协议来向各个传感器传送查询命令。此外，程序员还需理解网络拓扑结构，因为数据查询命令可能需要从某个监测区域获取数据。

3）显然程序员可以构建应用层程序控制传感器的行为，但很多无线传感器网络操作都不是仅依靠单个传感器节点而是需要多个节点的协作才能完成。例如，为节省无线通信能量消耗，数据查询命令最终是通过数据聚合技术来实现数据的收集。也就是说，驻留在某个节点上的程序需要控制很多其他节点。显然，这对于任何程序员都是一个具有挑战性的任务。



图 7-1 中间件的位置

因此，为了减轻无线传感器网络程序员的繁重工作压力，需要在传统的网络体系中加入一个新功能层，称为中间件层（middleware layer）。如图 7-1 中右侧部分所示，中间件层位于应用层和传输层之间。通过无线传感器网络中间件，低层复杂的操作细节可被隐藏和屏蔽。程序员可以摆脱对麻烦的无线传感器网络动态网络拓扑和底层嵌入式操作系统应用程序接口的考虑。

一个好的无线传感器网络中间件能够为程序员提供一些可重用的代码服务，这可以帮助程序员访问和使用各网络资源的功能，这将降低在代码分发、数据聚合和电量管理上的工作量。

虽然传统的中间件模式（在分布式计算系统中使用）也能通过隐藏上下文信息来提供透
226 明性抽象，但是它的主要目的是满足单个节点的需要。而无线传感器网络应用则是以数据为中心的，要求中间件能够在所有可用的节点而不是单个节点上实现操作。此外，无线传感器网络中间件还要沿着转发路径在中间节点上实现数据聚合。由于传统的分布式系统中间件采用端到端模式，因而不需要支持数据聚合功能［Miaomiao08］。

数据管理是中间件的一个重要任务，中间件需要对数据结构和操作等细节提供适当的抽象。否则，开发应用的程序员必须对异构数据和底层操作进行管理［Miaomiao08］。

在设计传感器网络中间件时，应保证其在具有有限处理能力和能源的传感器节点上尽量地被轻量级执行。



奇思妙想

如何减少程序员的工作量是很多程序开发平台的目标。如果无线传感器网络程序员在开发应用层程序之前，需要知道所有网络操作的细节，那将耗费大量的时间。很多无线传感器网络公司都试图将复杂的传感器或网络操作封装成一系列 API 接口，这也是中间件的任务之一。基于这些友好的 API 接口，程序员可以快速地开发出有用的应用程序。

7.2　无线传感器网络中间件参考模型

模型概述

如图 7-2 所示，无线传感器网络中间件包含 4 个主要组成部分：

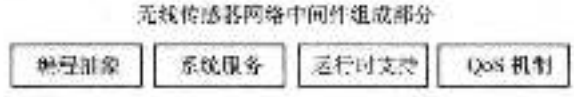


图 7-2　无线传感器网络中间件组成部分

1）编程抽象：中间件的设计应该定义一组友好的 API 接口，将复杂的无线传感器网络操作隐藏起来。

227 2）系统服务：在定义编程抽象后，中间件应该在内部提供这些抽象的具体实现。这些实现是系统程序而非用户级程序的一部分，因而被称为**系统服务**（system service）。

3）运行时支持：在获得以上系统服务代码后，传感器操作系统应该能够以最优化方式执行这些代码，该功能被称为**运行时支持**（runtime support）。

4）服务质量（Quality of Service，QoS）：在应用层，人们一般使用 QoS 定义一些明显的应用性能指标，例如，数据分辨率、处理速度和网络时延等性能。中间件应能够适应不同的服务质量需求。



无线传感器网络中间件的这三个组成部分间具有紧密的联系。定义编程抽象的目的在于隐藏无线传感器网络的复杂操作，用户仅需要中间件提供一系列的系统服务，而中间件设计者需要为这些系统服务提供运行时支持（runtime support）。

图 7-3 显示了以上组成部分的细节。需要注意的是，这仅是一个典型的中间件参考模型，并不是所有无线传感器网络中间件都要包含这 4 个组成部分。

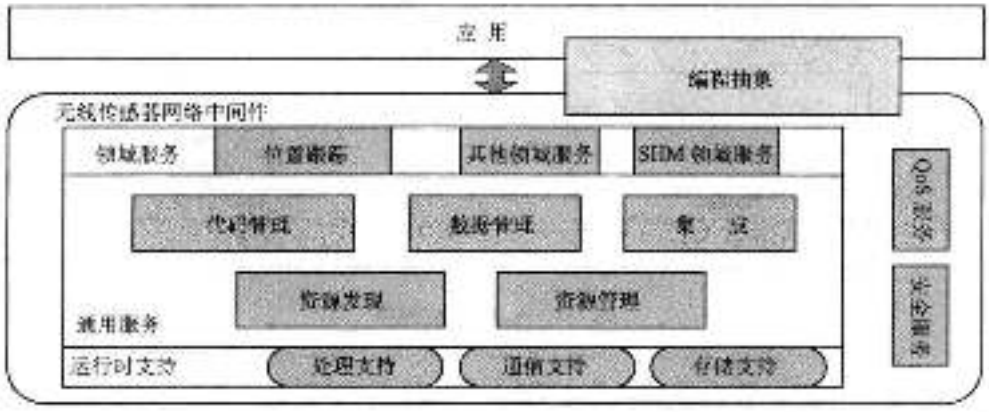


图 7-3　无线传感器网络中间件参考模型

中间件仅在传感器节点上实现的认识是错误的。事实上，由于用户可以在系统中不同的位置上编写系统程序，因此中间件可以驻留在传感器节点、汇聚节点（即基站）以及与汇聚节点通信的用户终端上。分布在不同位置的中间件组件间需要相互通信以实现共同的目标，例如执行数据查询。图 7-4 说明了这一点。

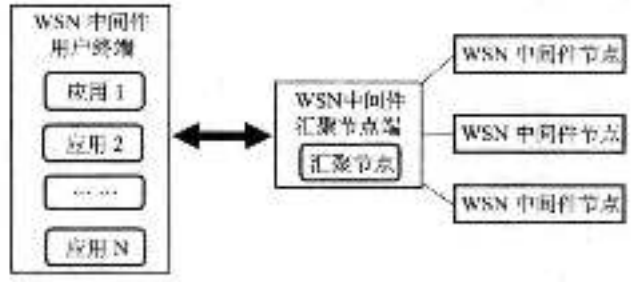


图 7-4　无线传感器网络中间件的系统结构

7.3　中间件实例：Agilla

有一类中间件是基于“移动代理”的概念实现的，它是一个可执行线程，可在节点间迁移，该“移动代理”封装了执行代码、系统状态和应用数据。

Agilla［CFok05］是基于代理的中间件实现。我们可以在无线传感器网络中引入新的代理并重新编写网络应用程序。

图 7-5 展示了 Agilla 系统模型。应注意到，每个传感器节点可以支持多个代理，每个节点维护着元组空间和邻居列表：

1）**元组空间**（tuple space）可由多个驻留在同一节点的代理们共享。Agilla 提供了专门的指令来实现各节点间元组空间的远程访问。

2）**邻居列表**（neighbor list）包含了无线传感器网络中所有直接邻接节点的地址，这便于代理的迁移。

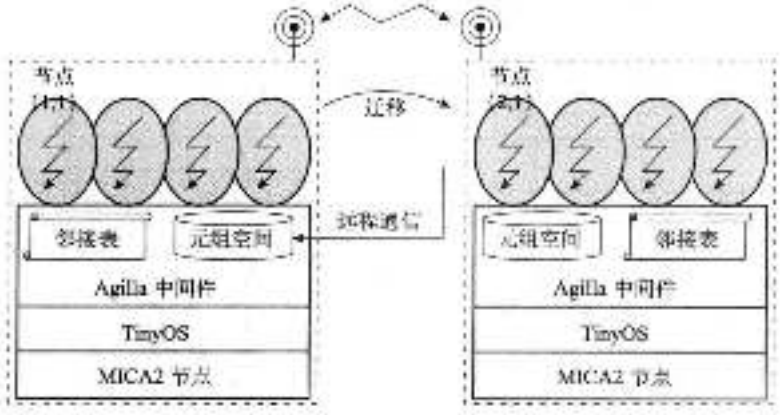


图 7-5　Agilla 系统模型


奇思妙想

移动代理是一个热点研究问题，它的基本特征是可以将未完成任务从一个物理实体转移到另一个实体，这样的“链接”就可以完成系统级任务。请注意，移动代理与多代理不同，后者假设各代理是不会在实体间迁移和转移的。

如图 7-6 所示，Agilla 中的移动代理由一个栈、一个堆和一些寄存器组成，其中堆是用于存储系统变量的内存空间。和通常的 CPU 结构相似，寄存器用于存储代理 ID（对每个代理是唯一的）、程序计数器（Program Counter，PC，记录下一条指令的地址）和条件代码。

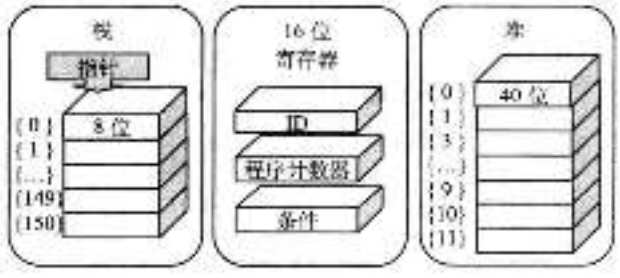


图 7-6　Agilla 代理结构

代码迁移（code migration）可通过在节点间移动或复制一个代理来实现。元组空间可在迁移过程中将所有寄存器变量打包，当代理迁移时也将其状态变量和运行时代码带走。当代理到达一个新的节点后，再恢复代码的执行。多跳迁移（迁移多次）是通过中间件操作系统实现的。

7.4　用于获取数据的中间件实例：Mires

无线传感器网络中间件的典型任务是进行数据管理，包括数据获取、数据处理和数据

存储。

本节使用 Mires［ESmoto04］作为数据获取的实例，它包含一系列的功能，例如，事件定义，事件登记或撤销、事件检测和事件投递。图 7-7 展示了 Mires 中间件的体系结构。

Mires 使用了一种发布/订阅模式（如图 7-8 所示）实现基于事件的数据获取。这种模式支持同步通信，并且便于传感器节点和汇聚节点间的消息交换。采用发布/订阅模式的系统包含两个基本组成部分：事件订阅者（在汇聚节点上）和事件发布者（即事件代理，在传感器节点上）。

在 Mires 中，汇点的应用层订阅感兴趣的事件数据，其订阅消息向网络节点广播，各网络节点将它们搜集的数据发布到网络中。

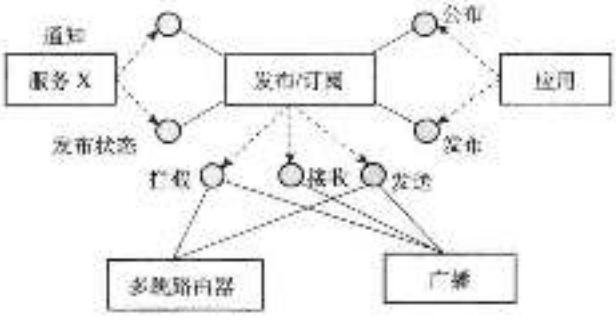


图 7-7 Mires 体系结构

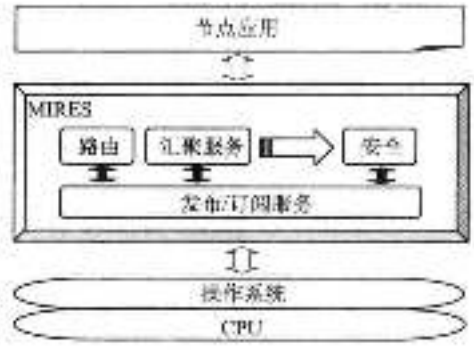


图 7-8 Mires 发布/订阅模式的组成部分

基于查询的数据模型中间件可使用 TinyDB［SRM05］的洪泛法实现查询请求在网络中的散布和传播。

7.5 数据存储实例：DSWare

无线传感器网络中间件需要支持的重要任务之一是以数据为中心的存储，数据服务中间件（DSWare）［SLi03］就是这样的中间件。如图 7-9 所示，DSWare 实现类似于数据库的包含各类数据服务的抽象：

1）事件检测模块对应于上面讨论的数据获取服务。

2）群组管理模块实现无线传感器网络的重要功能——数据汇聚。

3）调度模块可依据能效或延时性能对所有中间件服务进行调度。

4）数据存储模块按照数据间语义关系来存储数据，它存储地理邻接区域相关数据来实现网络内数据处理。

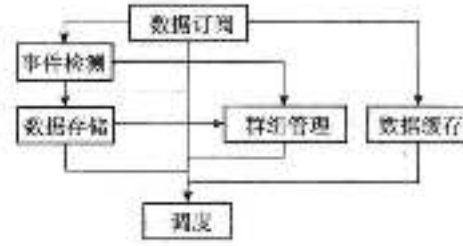


图 7-9 DSWare 的组成部分

5）缓存模块提供了经常使用数据的多个副本，DSWare 在网络中散布缓存数据，以实现高可用性和快速查询的执行。

7.6 无线传感器网络运行时支持实例：Mate

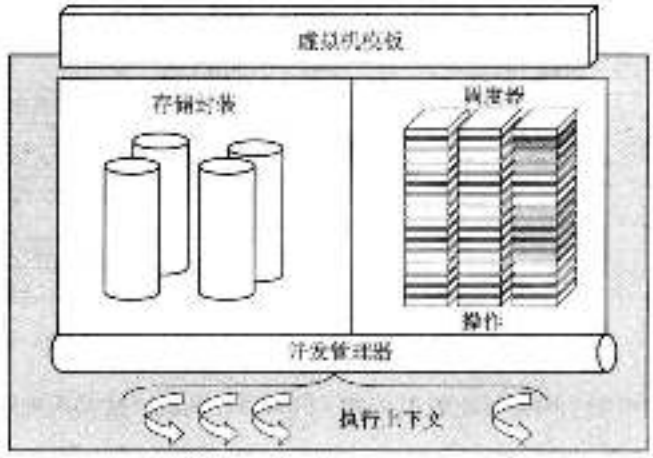
正如我们前面所提到的，所有定义的中间件服务都要具有某种形式的运行时支持，以保证

提供一个良好的执行环境。

运行时支持应具有以下基本功能：进程间通信（Inter-Process Communication，IPC）、存取控制、能量管理（电压调节和组件管用）。这些功能是非常重要的，它们可用于实现高级别的中间件服务，例如，多线程处理、任务调度、同步内存访问和扩频信号的频谱管理）。

通常情况下，可用各类虚拟机（virtual machine）实现运行时支持，可在嵌入式操作系统的顶部，将虚拟机作为专用平台内核使用。Mate［PLevis02］正是这样的实例，它构建于TinyOS之上，图7-10说明了Mate的结构。

Mate架构的核心是调度器，它维护着一个用来缓存进程/线程上下文信息的缓冲器并对多进程/线程的上下文进行切换。Mate的并发模型是基于静态的命名资源（如共享变量），这些资源能被任何操作显式指定。



234　图7-10　Mate的架构

7.7　QoS支持实例：MiLAN

QoS支持对于应用而言是非常重要的，所需要具有的性能有：容错性、可靠性、安全性和实时数据处理。无线传感器网络中的QoS需求可用以下参数来表达：包延迟、抖动和丢失、吞吐量和等待时间。不过，通常需要更多的指标定量化度量其性能，例如，数据精度、聚合延迟、聚合度、覆盖性和准确度。提供QoS支持的无线传感器网络可保证高效的数据获取。

MiLAN［WBHeinzelman04］定义了一系列QoS支持，如图7-11所示。无线传感器网络应用程序需要在开始时将QoS参数集合从应用层传送到MiLAN（即中间件层），这种QoS传送是通过基于状态的可变QoS需求图和节点QoS图来实现的。

1）基于状态的可变QoS需求图：该图根据当前系统的执行状态来为每个性能参数设定最小满足QoS。

2）节点QoS图：该图用于从QoS需求视角来决定无线传感器网络中哪些节点需要提供QoS支持。

问题与练习

7.1　说明无线传感器网络系统设计中中间件的作用，指出它的组成部分并简述各部分的作用。

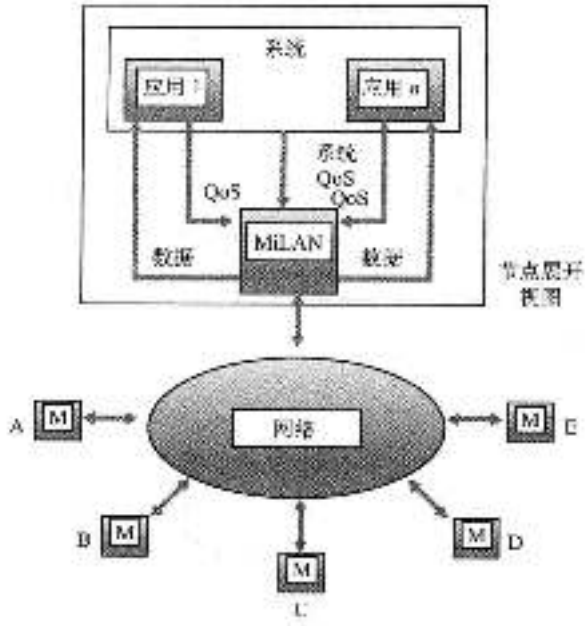


图 7-11 MiLAN 中的 QoS 支持

7.2 在设计无线传感器网络系统中间件过程中遇到的挑战有哪些?

7.3 为什么要使用编程抽象?

7.4 Agilla 如何实现代码管理?

7.5 为什么 Mires 使用一种发布/订阅体系结构?

7.6 DSWare 是如何实现数据存储的?

7.7 说明 MiLAN 中 QoS 支持的组件。

235
-
236

第8章

Wireless Sensor Networks: Principles and Practice

传感器数据管理

为了能检测到若干事件，在收集到传感器数据后，需要进行进一步处理（如噪声移除）。本章将介绍一些典型的传感器数据管理问题，包括传感器数据处理（如数据清理）、传感器数据库结构、数据查询策略和数据聚合（data aggregation）等。

8.1 传感器数据清理

8.1.1 背景

在精确性、准确性、硬件耐受性和外部噪声方面，各种传感器有着根本的不同。例如，光敏传感器（photovoltaic sensor）有很大的噪声分布［Bychkovskiy03］。传感器的运行环境会影响数据获取的性能。其他外部或不可控制的因素也会影响传感器读数。多数情况下，这会导致不精确的测量结果。例如，贴附在桥面的应变式传感器可以测量到货车的重量值，但是该值可能会因其他物体的震动而受到影响。

很多国家正在开发价廉的、任何地方可用的，电池耗尽时方便遗弃的传感器。但这些便宜的传感器往往是对内外噪声高度敏感的，既不精确也不准确。

当对实体或模型测量时，存在若干误差源。误差主要分为两种：系统的（误差）和随机的（噪声）。系统误差源于操作环境的改变，如温度、湿度或传感器老化，系统误差可通过校准改正［Bychkovskiy03］。

237

随机噪声可能有以下来源：1）随机硬件噪声；2）测量不准确；3）环境的影响和噪声；4）测量导出值的计算不精确（即在相同条件下测量同一现象的不一致性）。Elnahrawy 和 Nath（2003）已经讨论了减少随机噪声的方法以及对传感器数据的重要影响［Elnahrawy2003］。

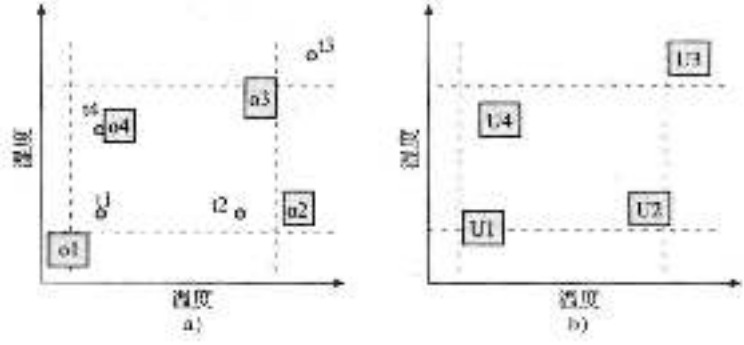


图 8-1 a）基于观察数值，样品 1 和样品 4 将被舍弃；b）基于不确定范围，只有样品 3 将被舍弃

低成本传感器产生的随机误差将严重影响传感器数据的准确性。这将导致不精确的甚至错误的答案，从而影响即时关键决策或执行器激活（activation of actuator），进而蒙受巨大损失。因此，传感器数据误差不容忽视。

为了支持以上观点，文献［Elnahrawy2003］给出了一个例子——利用价廉的温湿度无线传感器监控细菌增长（bacteria growth）。当温度和湿度达到某个给定阈值时，此样品将被丢弃。如图 8-1 所示，基于传感器数据，样品 1 和样品 2 应该丢弃，而样品 3 和样品 4 应该保留。但

真实的情况是，样品1和样品2应该保留，样品3应该丢弃。



为什么传统数据库系统不需要进行数据清理？传统数据源获取数据时有明确的数据输入软件，或者说事务活动是一系列可信任的步骤，因此不需要进行数据清理。银行、公司或者个人通常使用这种数据。传统数据库系统的数据清理的模型是假定的，任何噪声数据假定已经被单独的数据库功能模块以离线的方式处理掉了。

238

然而，对于无线传感器网络数据来说，数据清理是与传统数据库系统不同的。这些数据是连续产生的并形成数据流，而且不能使用离线方法去处理数据，而是以实时的方式处理。

对于无线传感器网络，需要考虑一些重要的性能度量指标，如带宽、能量损耗和数据误差。数据误差非常重要，因为在确定传感器真实读数时，它会造成不确定性。

Elnahrawy 和 Nath 介绍了一种清理和查找传感器噪声数据的有效方法［Elnahrawy2003］。该方法降低了因随机噪声使传感器数值上升而造成的不确定性。更确切地说，是以联机方式使用贝叶斯方法减少不确定性。该方法称为贝叶斯清理（Bayesian-based cleaning，BayC），贝叶斯清理可以在独立传感器或基站使用。其假定任意独立的传感器的读数都是同等重要的。贝叶斯清理所做的假定如下：在空间中部署 n 个传感器，n 个传感器构成集合，即 $S = \{s_i, i = 1, \ldots, n\}$，并形成一个无线传感器网络。路由选择、拓扑结构维护、通信等网络技术已经在该网络中应用。同时在某特定的时间 t 下，每一个 s_i 都作为一个元组保存到传感器数据库中。从传感器读取的数值也保存到数据库相应的属性中，每一个传感器提供一个或多个数值与每一个量度对应，同一个传感器可以感知不同的现象。也就是说，在同一个位置可以放置一些专用传感器，然后将来自这些传感器的数据合并处理，如同数据来自一个虚拟的多属性传感器。

所有的属性值假定都是实数值，通过扩展贝叶斯清理框架，使之能处理离散属性的情况。注意，通常将收集的各个传感器数据加上时间戳，而稍后在讨论传感器读数时会丢弃该时间戳 t。

8.1.2 通用模型

如图8-2所示，贝叶斯清理框架包含以下主要两个模块：

1）**清理模块**：主要通过**计算无法预测数据的不确定模型**（computing uncertainty model）以在线清除噪声传感数据。清理模块有三个输入：①噪声传感数据，②每一个传感器或误差模型的噪声特征元数据，③利用先验知识所了解到的各个传感器的真实读数的分布。后文将简要讨论后两个输入。清理模块的输出是不确定（噪声）传感器的读数的概率模型，即一个概率密度函数，下一节将提供这个模型的计算细节。

239

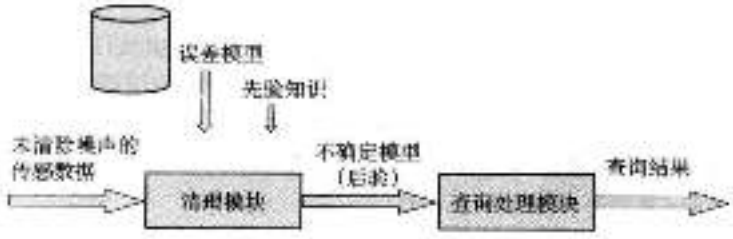


图8-2 传感数据清理框架

2）**查询处理模块**：基于当前读数的不确定模型，将查询发送到系统，并返回查询结果。

不确定模型是基于概率的（即统计分布），而传统的查询假定每个读数是一个单一的值，因此传统查询在这里并不适用。该模块使用的算法是使用统计方法去计算一个随机变量的函数。

传感器误差模型应该反映噪声的分布，假定噪声服从平均值为零的高斯分布。为了确定误差模型，在基于每一个传感器的性质（包括正确性、精确性等）和通常部署条件下校正测试的基础上，计算该分布的方差。校正测试可以由生产商或用户在安装后、使用前进行。同时需要考虑环境因素或区域特征。误差模型随着时间的推移也会发生改变并可能被新的模型取代。在清理模型中，误差模型应以元数据的形式存在。当传感器噪声特征不同源（homegeneous）时，不同类型的传感器甚至是个体传感器都应该进行标定以获得其自身的误差模型。

可以通过多种方法获得先验知识，即真实传感器数值的分布。已有的感知环境的数据，如历史数据、低噪声数值，其中专家知识和主观推测，都可以用来计算分布情况。如果待感知的环境遵循某个参数模型，那么分布可以实时动态地计算出来。例如，如果在 t-1 时刻到 t 时刻
240 内，易坏物品的温度值下降了百分比 x，那么 t-1 时刻的（已清理）数值通常作为 t 时刻数值的先验分布。清理模块将时刻 t 的误差模型结果和所观察到的噪声数值作为两个输入，进而获得传感器在时刻 t 的不确定模型。该方法在这种情况下的使用与 Kalman［LEWIS86］的过滤方法类似。

对于噪声传感器读数的不确定的建模，最直接的方式是假定每个传感器数值分布满足高斯分布模型，即该分布模型的中心与观察到的数值相近，其方差等于噪声分布的方差。通常认为，使用先验知识可以获得更准确的估计［KAY93］，因此，使用先验知识降低噪声传感器的不确定性是可行的，具有相对小的方差的先验知识会更有用，它降低了不确定性，并使整体准确性有所提高。

如果先验知识达不到相对窄的分布（与噪声分布相比），那么使用贝叶斯清理方案会更好。在大多数情况下，即使噪声分布的范围很宽，通过计算仍可以获得有效的先验知识（例如，在部署了噪声传感器的大范围区域内收集诸如温度等拥有成熟分布模型的环境数据）。

在节点级别和数据库级别（基站）皆可以进行数据清理和查询处理。当然，会有一定的通信和处理损耗（如能量损耗、存储开销），选择哪种处理方法取决于传感器的性能和应用。

节点级别：在节点级别清理数据时，**资源约束传感器**（resource-constrainod ensor）需要一定的存储空间去保存先验知识和误差模型。此外，将先验知识从基站发送到传感器将产生很大的通信开销。

数据库级别：因为基站拥有足够的计算能力处理数据库操作，因此可以假定数据库级，任何处理和存储操作都是没有开销的，这也是在基站执行数据清理和查询处理的主要优点。因为不需要发送动态先验知识给传感器，所以节省了通信费用。

8.1.3　降低不确定性

本节将说明与传感器噪声数值相关的降低不确定性的方法，也就是如何计算传感器的更精确的不确定性模型。文献［Elnahrawy2003］提出了一种实时数据清理的方法，它将真实数值的先验知识、传感器的误差模型和观察到的噪声数值集中在一个以贝叶斯理论为基础的计算步骤中。在给定参数（θ）的条件下，数据 x 的可能性，即概率，表示为 $p(x \mid \theta)$，$p(\theta \mid x)$
241 表示 x 发生的情况下，θ 的后验概率密度函数

$$p(x \mid \theta) = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}} = \frac{p(x \mid \theta)p(\theta)}{\int p(x \mid y)p(y)dy} \tag{8.1}$$

假定一个传感器只有一个属性，属性 o 是含有噪声的，也就是说，该属性值比真实值 t 或

高或低。真实值 t 服从均值为 $\mu = t$、方差为 δ^2 的高斯分布，即 $p(o \mid t) \sim N(t, \delta^2)$。对 t 应用贝叶斯原理，以获得更准确的不确定性模型（后验概率密度函数）$p(t \mid o)$。而观察值 o 与误差模型合并满足分布 $\sim N(0, \delta^2)$，故真实数值分布的先验知识 $p(t)$ 表示如下：

$$p(t \mid o) = \frac{p(o \mid t)p(t)}{p(o)} \tag{8.2}$$

某些专用的传感器 s 的读数服从均值为 μ_s、标准方差 σ_s 的高斯分布，即 $t \sim N(\mu_s, \sigma_s)$（先验知识）。通常来讲，真实数值 t 的先验分布并无严格的规定，可根据实际情况选择合适的分布，只是，对于感兴趣的属性，使用高斯分布更方便对先验知识建模。

$$\mu_t = \frac{\delta^2}{\sigma_s^2 + \delta^2}\mu_s + \frac{\sigma_s^2}{\sigma_s^2 + \delta^2}O \tag{8.3}$$

$$\sigma_t^2 = \frac{\sigma_s^2 \times \delta^2}{\sigma_s^2 + \delta^2} \tag{8.4}$$



案例研究

该案例［Elnahrawy2003］说明在某时刻如何获得一个温度传感器的不确定模型。假定预先知道温度 t 服从均值为 9 百分度、标准方差为 4 的高斯分布，即 $t \sim N(\mu_s = 9, \sigma_s^2 = 4^2)$。同时也假定该传感器的噪声的标准方差为 10，即 noise $\sim N(0, \delta^2 = 10^2)$。如果当前观察到的噪声温度 O 为 15，那么根据公式 8.3 及公式 8.4，可知真实的温度应服从均值约为 9.8、标准方差约为 3.7 的后验分布，即 $p(t \mid o) \sim N(9.8, 3.7^2)$。如图 8-3 所示。

242

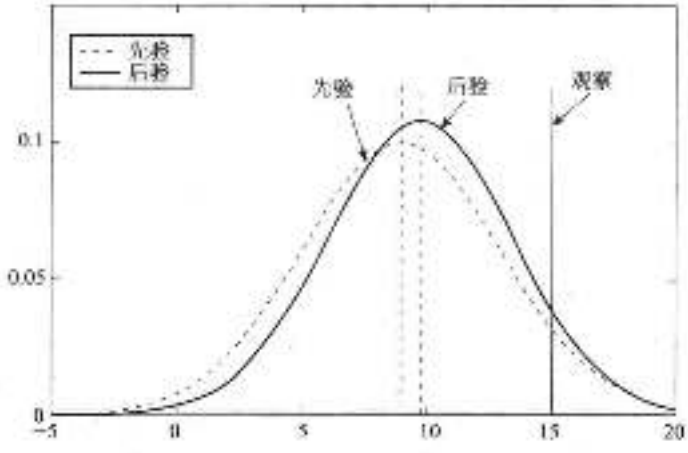


图 8-3　真实温度的不确定模型与观察到的错误读数的对比图



奇思妙想

很显然，通过使用先验知识，贝叶斯清理可以极大地降低不确定性。此外，先验分布的方差与噪声的方差相比非常小时，也就是说，当先验数据居主导地位时，后验知识产生错误的可能变得很小，不确定性也大大降低。所以，贝叶斯清理的不确定性模型与非先验方法比，会更加准确。公式 8.3 同样说明了一个有趣的事实：贝叶斯清理的方法在先验知识和观察到的噪声数据之间做了很好的权衡，传感器噪声越少，观察到的数值就越重要，模型也更依赖于观察值；相反，传感器噪声级别很高时，观察值则应忽略。

8.2 TinyDB：应用于传感器网络的可获取的查询处理系统

TinyDB 旨在处理传感器网络查询处理过程中的若干问题：

1）何时是发出某种传感器数据查询的合适时间？

243 2）哪一个传感器节点的数据与发出的查询有关？

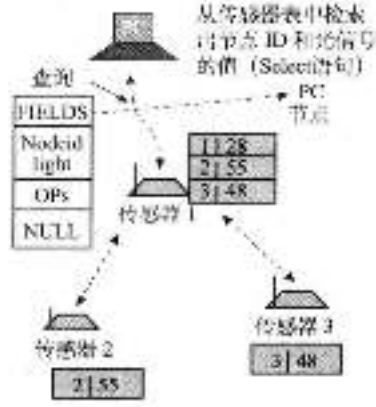


图 8-4 无线传感器网络查询和结果传播示意图

3）获取传感器数据样本的正确顺序是什么？如何将其他 WSN 操作与获取样本操作交替进行？

4）为了处理或传送某个数据样本，是否值得消耗计算能力或带宽？

在以上问题中，问题 1 是唯一需要获取的，其他三个问题可以通过修改解决传统查询处理的方法达到，问题 2 和 3 可以通过索引和优化的方法解决，问题 4 与流处理和近似查询应答所面临的问题比较相似。

数据查询的基本原则如图 8-4 所示。当解析并优化查询后，查询将发送给传感器网络，查询开始传播时将初始化一个路由树，然后经由路由树传播并处理，查询的结果通过路由树返回基站。



关于无线传感网络数据查询，应记住以下几点事实：1）它建立在明确的路由拓扑结构上。例如，一个以基站为根节点的分层树结构。这种结构有利于查询命令和快速数据查找高效地进行传播。2）虽然 TinyDB 数据查询命令与 Microsoft SQL 的语法类似，但它在无线传感网络的内部实现（即如何在传感器间发送查询命令及传感器如何反馈需要的查询结果）更具有灵活性。

244 3）数据查询属于应用层的问题，但是它需要路由层的支持。

8.2.1 数据模型

在 TinyDB 中，传感器数据保存在具有以下结构的表中：行是某个时刻某个传感器的数据，列表示由传感器产生的一个属性（如光和温度）。

8.2.2 基本语言特点

TinyDB 中的数据查询要满足实时响应的需求。一个查询命令由 SELECT-FROM-WHERE-

GROUPBY 子句组成，支持选择、连接、投影、聚合。其语法与 Microsoft SQL 相似。FROM 子句指向传感器表或存储的表，称之为物化点（materialization point）。日志查询可以创建物化点，并提供对子查询和窗体流操作的基本支持。

为了定义一个查询，必须先定义查询中一个重要的参数，即采样间隔。两次采样的开始时间的差称为一个查询周期（epoch），其提供了结构计算和能耗最小化的方法。考虑下面的查询：

```
SELECT nodeID, light, temp
FROM sensors
SAMPLE PERIOD 5s FOR 15s
```

该查询可获得 15 秒内，每隔 5 秒报告一次传感器的 ID、光和温度值，采样时间共 15 秒。查询的结果通过多跳拓扑传送给网络树的根节点（基站），基站将结果记入日志并输出给用户。查询结果包含一个多元组流，元组间时间周期为 5s，每个元组拥有一个时间戳以区分其产生的时间。

查询开始后，传感器按照 SAMPLE PERIOD 子句的时间要求初始化数据集。当然在 TinyDB 中有一个简单的同步协议确保所有传感器具有正确的时间戳。

注意，每一个查询都有一个 ID，使用 STOP QUERY ID 命令可以明确地停止一个查询。停止一个查询的方法还有：1）如上所示，使用 FOR 子句设置查询周期，到达时限后查询停止。2）使用停止条件。

在 TinyDB 中，基于传感器数据流上的物化点，引入了窗口（windows）的概念。可使用小的缓冲区集聚物化点中的数据，并在其他查询中使用该数据。考虑下面的例子：

```
CREATE
STORAGE POINT recentlight SIZE 8
AS (SELECT nodeid,  light,  FROM  sensors
SAMPLE PERIOD 20s)
```

245

该语句使用一个本地（单独节点）传感器存储最近数据的一个流视图。

8.2.3 基于事件查询

在数据获取时，TinyDB 提供基于事件的数据收集方式，这是除连续和基于轮询机制之外的又一种方法。将产生事件的代码提前编译植入传感器节点后，另一个查询或操作系统就可以触发 TinyDB 事件了。

考虑下面的查询：

```
ON EVENT animal detect (loc);
SELECT AVG (light), AVG (temp), event.loc
FROM sensors AS s
WHERE dist (s.loc, event.loc) <10m
SAMPLE PERIOD 2s FOR 30s
```

当位于鸟巢附近的传感器监测到一只鸟时，该查询可用于获得该传感器的光和温度的平均值。每当监测鸟的事件发生时，监测传感器将产生查询，每隔 2 秒收集一次（共持续 30 秒）附近传感器的光和温度平均值。

基于事件的机制使得系统在一些外界条件改变时被唤醒，而不必持续轮询或等待数据到来。许多中央处理器（即微处理器）硬件内置了中断线用于唤醒处于休眠状态的传感器，使

之开始数据处理，事件触发查询有效地降低了能耗。

8.2.4 TinyDB 定义的其他查询

TinyDB 里定义了更多满足不同特殊情况的查询，下面给出几个例子。

网络健康度查询：应用于网络的元查询，例如，查询网络树上父节点和邻节点，或查找某个节点的电池寿命低于某阈值。下面的网络健康度查询用于报告所有电池容量小于 k 的传感器节点：

```
SELECT nodeid, voltage
WHERE voltage < k
FROM sensors
SAMPLE PERIOD 10 minutes
```

激励式查询：在这类查询中，通过数据查询后，用户需采取一些行为。OUTPUT ACTION
246 子句便是用于此目的。例如，在一栋大楼中，当温度值高于某个阈值时，风扇将被开启：

```
SELECT nodeid, temp
FROM sensors
WHERE temp > threrehold
OUTPUT ACTION power-on (nodeid)
SAMPLE PERIOD 10s
```

注意，OUTPUT ACTION 子句表明为了满足查询的结果应调用的传感器控制命令。在上例中，电源开启命令被激发，使微处理器的引脚输出“高”的信息，并关闭转换单元，为传感器控制的风扇供电。当温度低于某个阈值时，也可以使用 OUTPUT ACTION 命令关闭风扇。

离线发送：有时一些事件发生得太快，以致于无法以实时的方式获得记录数据。因此，TinyDB 提供将结果记录在 EEPROM 中的功能，以备以离线的、非实时方式发送。

8.2.5 基于能量的查询优化

上节介绍了数据查询语法，本节将讨论这些查询的传感器的内部实现，主要是关注数据获取、选择、聚合的优化。

在基站端，TinyDB 将查询解析为简单的二进制格式，然后传播至传感器网络进行初始化和执行。查询在传播前经过简单的查询优化，对采样、选择、连接的顺序做优化处理。

优化器将选择能量消耗最小的查询方案，这种优化考虑了若干问题，如 CPU 处理和无线通信的消耗，它们都会消耗传感器的电能。

首先，考虑将元数据类型保存在优化器中，TinyDB 的每个传感器都维护一个描述本身位置属性、事件及用户定义函数的元数据目录。通过路由协议，元数据通过查询优化器定期发送到基站供其使用。



奇思妙想

什么是元数据？考虑一个数据库（如 Microsoft Access），MS Access 使用一个表去存储原始数据，而数据库操作（如索引和排序）要由一系列命令来创。在哪里存储这些命令呢？使用元数据。因此，元数据是数据的数据，即从大量的原始数据中提取或建立一个少量的控制信息。

247

使用 TinyOS 类 C 编程语言，可以将元数据在编译时通过静态链接在节点中进行维护。如在查询元数据时需要某一事件或属性，可以在一个接口文件中定义它，然后使用处理函数去引用该接口文件。例如，为了使查询处理器获得传感器网络拓扑结构，TinyOS 程序组件采用一个叫做父节点的属性，该属性可以通过使用一个处理函数访问，它返回查询树上当前节点的父节点 ID。

基于事件的元数据有以下结构：名字、唯一签名、用于查询优化的频率估计（frequency estimate）。任何用户定义的谓词需包括一个名字和一个签名，以及一个自定义函数创建者提供的选择度估计（selectivity estimate）。

TinyDB 中的基于属性的元数据如表 8-1 所示，主要包括能耗、取数据时间、属性取值范围等。

表 8-2 显示了若干能耗和采样时间的例子，表明在不同数量级上，不同的传感器能耗和取值时间是不同的。

接下来通过一个简单例子来说明元数据的使用。假定监测一个由植物及其生物过程建立起来的小环境 "DELINOO"。表 8-2 表明加速表和磁强计每次采样的能耗在数量级上有很大的区别，这也意味着不同的数据查询方案（使用不同采样和选择的顺序）在能耗方面有很大的不同。例如，以下三个查询策略将产生不同的能耗：1）磁强计和加速表的采样在选择（selection）前进行，2）磁强计先采样，在加速表采样前对数据进行选择，3）加速表先采样，然后在磁强计采样前对数据进行选择。

表 8-1　属性的元数据字段

元数据	描述
能耗	采样属性值消耗的能量（焦耳）
采样时间	采样属性值消耗的时间值（秒）
容量	属性的恒定值（如 id）
变化率	属性改变的频率（单位/秒）
范围	属性值动态区间（单位区间）

248

表 8-2　不同传感器的能耗汇总

传感器例子	每次采样所需时间（ms）	启动时间（ms）	电流（mA）	每次采样所需能量（mJ）
气候传感器				
日照强度	500	800	0.350	0.525
大气压力	35	35	0.025	0.003
湿度	333	11	0.500	0.500
表面温度	0.333	2	5.6	0.0056
环境温度	0.333	2	5.6	0.0056
标准 Mica 节点传感器				
加速表	0.9	17	0.6	0.0048
（振动性）热敏电阻	0.9	0	0.033	0.00009
磁强计	0.9	17	5	0.2595
其他传感器				
有机附产品	0.9	>1000	5	>5

8.2.6　TinyDB 策略一览

表 8-3 列出了 TinyDB 中使用的主要技术。

表 8-3　查询处理技术一览

技术	概述
基于事件的查询	避免轮询开销
生命期查询	满足用户指定的寿命约束
实情获取/谓词	避免选择查询中不必要的采样损耗
共享聚合计划	避免聚合查询时不必要的采样损耗
事件批处理	减少多个事件查询触发时的执行损耗
语义路由树（SRT）	避免查询传播损耗或对常量属性使用空间查询不必要的节点
通信调度	在非活动时间将无线节点的处理器和无线设备
数据优先化	依据一个用户指定的优先化方法选择最重要的数据进行传输
抑制	避免聚合查询时不必要的传输
速率自适应	有意地丢弃若干元组以避免无线信道饱和，允许发送最重要的元组

8.3　数据聚合：独立于应用的数据聚合（AIDA）

如果不使用数据聚合的方案，那么传感器是互相独立的，任意传感器都可以将其收集的数据发送到终端节点（汇聚节点）。这样的策略不能有效利用无线传感网络的特点。也就是说，邻近的传感器因为在相同区域监测到相同的事件，所以会产生大量冗余数据，因此不必每个传感器都单独地发送数据，这样的方案由于冗余、长距离数据传送浪费了很多能量。

网内数据聚合可以克服上述缺点。如图 8-5 所示，每一个传感器仅仅将数据传送给它的相邻传感器，有一个算法可以用于选择合适的传感器作为数据聚合节点，聚合节点将所有的冗余数据去除，或者基于从邻近多个传感器接收的输入生成一个新的值（如平均值）。与未采用数据聚合的方案相比，采用数据聚合将使网络通信量大幅减少。

249
|
250

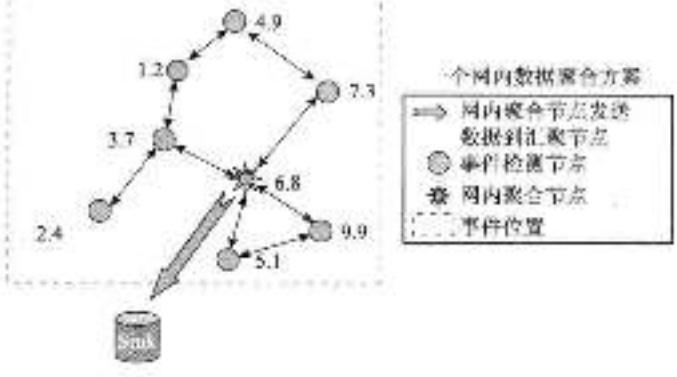


图 8-5　网内数据聚合方案



在无线传感网络中，由于传感器间数据冗余性和相关性强，数据聚合是一个重要的概念。当然，数据聚合（data aggregation）的概念与数据融合（data fusion）是不同的，数据聚合通常在路由层协议（如网络拓扑发现）上使用，用于执行高层次的数据分析。数据融合从信号处理的视角去看待，其应用在物理层（physical layer）。例如，如何在两个空间相关的时间序列中推导出一个新的信号（signal）？因为数据融合属于信号处理问题，所以本书不做更多阐述。

基于网格的数据聚合［Karthikeyan］适用于很多传感器网络，如军事侦察和气象预报。该方案将传感器网络环境分成若干预先确定的网格，每个网格都有一个网格中心（即数据聚合节点）用于观测数据并发送数据到汇聚节点。

在无线传感网络中，AIDA［Tian04］是一种自适应的独立于应用的数据聚合，AIDA包括两个功能组件：1）功能单元，用来聚合和分离（deaggregate）网络包（单元）；2）AIDA聚合控制单元，用来自适应地控制定时器的设定和对聚合的期望程度进行裁剪。 251

AIDA协议［Tian04］用于：

1）输出通信（即数据发出）：将网络层的分组（包）放入一个聚合池中。AIDA聚合功能单元在结束一个聚合后，基于同一聚合会话所连接起来的分组的数目和下一跳的地址，将结果下发到介质访问控制（MAC）层用于下一跳传输。

AIDA聚合控制单元决定要聚合的分组数目和聚合的时间。控制单元是基于反馈的自适应组件，它可以根据局部当前网络状况做出实时在线决策。

2）输入通信（即接收数据）：数据到达MAC层后，将会上发至AIDA功能单元。在AIDA功能单元中，接收到的聚合分组重新分离成原始网络单元，然后每个单元被上发至网络层，用于下一跳路由选择。

有人认为聚合不是一个好的想法，尤其在许多聚合数据到达同一节点，在每一个中间节点进行分解和重聚合，这会浪费网络资源。但是，为了确保层间的可组合性，并使网络组件能独立选择路由，AIDA仍然使用数据聚合。这是因为将多网络数据单元聚合成一个AIDA聚合单元，可以降低MAC层信道竞争（通过在MAC层使用等待/退避的操作）的总开销并减少控制分组（如802.11中的RTS/CTS/ACK，可最MAC的确认报文）的通信总开销。通过使用数据聚合，仅仅在每次聚合时产生以上开销。

虽然数据聚合有极大的好处，但设计一个选择合适聚合时间和参数的实时的自适应的AIDA控制单元仍然是个挑战。例如，何时进行数据到达聚合，是周期进行还是等待足够数据到达后进行？使用什么样的数据压缩方案？如何在聚合前去除“奇怪数据”（偏离其他数据）？

下面的问题很有趣：在网络层的何处应用聚合？为了避免传统网络层的改变，AIDA使用一个委托（delegation）方法来拦截所有的函数调用，在MAC层和路由层（网络层）建立直接通信。通过委托方法，AIDA数据聚合层成为MAC层和路由层之间的接口之一。

He［Tian04］设计了不同风格的AIDA，包括确定的、按需、动态反馈的方案。聚合方案的选择可基于静态阈值，或与一个动态的即时反馈控制系统结合共同完成最终方案。在接下来的讨论中，出于比较的目的，也会提到无聚合的基准。 252

1）无聚合：当不使用聚合时（基准方案），在路由层和MAC协议间只使用传统网络栈（不使用直接通信）。

2）确定性方案：“确定性”即AIDA聚合一个确定数目的网络数据单元到一个AIDA有效负载中。当路由层对确定数目的数据单元做数据聚合后，将通过AIDA负载发送到MAC层用于传输。为了避免AIDA在收集确定数目的数据单元前等待时间不确定，设定了一个超时时间T_{out}。如果时间超过该值，那么将不等待更多的数据单元而对当前已收集的数据单元进行聚合。

3）按需方案：在该方案中，数据聚合仅是一个可选择的操作，因为传感器将试图一直工作，也就是说，一个传感器不会花很长时间为了聚合去收集足够的数据单元。相反，当MAC层当前可用于传输时，不管已经收集了多少数据单元，马上结束聚合，即刻发送数据。仅仅在传感器没有数据发送或卡在那里（例如，发送信息队列已经准备好或无线介质繁忙使MAC层

无法访问信道），AIDA 层才进行数据聚合。按需方案避免了信息延迟。

4）动态反馈方案：该方案是按需和确定性方案的结合。该方案通过检测以下两个参数进行工作：①AIDA 输出队列大小：如果队列有空间，则将聚合更多数据单元；②当前队列延迟：如果延迟时间过长，将减少聚合大小（如聚合的数据单元更少）。AIDA 使用控制理论动态地调整聚合的程度以使 MAC 延迟保持在某个设定点。

8.4 传感器数据存储：层次化数据存储结构（TSAR）

传感器应用程序需要访问当前和历史传感器数据，故需要处理和存储传感器网络产生的数据。传感器数据挖掘可以从历史数据发现不寻常的模式、分析历史数据的趋势及离线分析特殊事件。存储传感器历史数据的存储系统在设计时必须考虑几个问题，如数据存储的位置、是否索引、应用程序以低能耗、低等待时间访问数据的方法。

学者们已经提出了传感器数据存储的各种方法，最简单的方案是允许传感器将数据或事件
[253] 以流的方式传送到基站，以进行长期归档存储［PBonnet01］。数据应进行索引以备稍后的高效访问。该方案的优点是：存储是集中的，对存储的访问是高效而价廉的；但其缺点是写入存储比较低效和昂贵。

一种替代的方案是允许传感器自身存储其数据（如使用内置闪存），故数据写入是本地的和高效的。然后某个传感器将处理一个简单的读请求，更复杂的读请求扩散到网络处理。该方案优点是采用分布式存储且写入代价小，但是读操作效率低且代价高。

其他一些传感器存储解决方案介于上面两个方案之间。一种是地理位置哈希表（Geographic Hash Table，GHT）方案［RATNASAMY01，RATNASAMY02］。该方案中，每个数据项都有一个键与之对应，分布式地理位置哈希表将键映射到不同的传感器，读取存储时，查找网内哈希表，找到存储数据项的节点。因此，该方案不需要使用扩散。

与适用于能量受限传感器节点的平面和同源结构不同，文献［Peter05a］提出了一种称为层次化数据存储结构（Tiered Storage ARchitecture，TSAR）的新型存储结构。TSAR 将无线传感网络组织成一个多层结构，这是一个预测型的存储架构，该结构将归档存储与缓存和预测结合。TSAR 将多层结构中资源丰富的传感器节点层用于缓存和预测。

TSAR 将数据保存在每个传感器的闪存中，传感器将元数据（简明标识信息）发送至附近的代理节点（一个特殊的拥有网络控制功能的传感器）。元数据与数据本身相比可能在数量级上小很多，从而降低了通信花销。资源丰富的代理节点之间相互作用，在多个节点间为网格中的感知数据建立起一个分布式索引。应用程序可以使用索引对历史数据进行有效地查询和读取。例如，一个读请求通过索引可以快速定位满足要求的数据，然后从相应的网络中检索。传感器存储的数据和代理存储的元数据分离使 TSAR 拥有平衡可控代理资源和降低传感器能耗的能力。

TSAR 将无线传感网络组织成三层，最底层是自由运程传感器节点，这些节点是低功率传感器节点；中层是可控的、能量丰富的传感器代理；最高层是各种应用程序和用户终端。

中层具有重要的作用，其传感器代理节点拥有有效计算、存储、存储资源的能力。在一个典型的无线传感网络应用中，代理层可能包括可控基站级别的节点（如 Crossbow 或 Stargate），每一个这样的节点会配备多路无线设备。例如，节点可能有一个 802.11 无线设备与一个无线网状网络通信，还有一个无线设备（如 802.15.4）连接着底层传感器节点。

中层代理节点可以使用太阳能电池以延长使用寿命，每个代理可以管理其附近的数百个低层传感器。一个典型的无线传感网络的部署中将包含多个分布在不同地理位置的代理。

最高层是无线传感网络应用程序，这些应用可通过查询接口查询网络［MADDEN02a］。 254
TSAR 旨在设计一个充分利用中层代理节点相对丰富的资源，以弥补底层传感器资源的稀缺性的存储系统。

TSAR 使用如下原则设计用于多层网络的传感器存储系统：

原则 1：本地存储，全局访问：与网络存储相比，本地存储在花销上更小、更有效率，并成为未来几年的趋势。为了使网络使用寿命最大化，可以在传感器自身闪存中进行数据的本地存储，与通过昂贵的无线传输交换存储消息相比，这将节约更多能量。TSAR 使用一个基于本地存储的有效信息检索机制。

原则 2：从元数据中区分数据：元数据通过预定义语法使用专门的数据字段。元数据使用标识符（如地点、时间或求和的数据值）。使用包含元数据的数据记录可减少查找和检索的时间，代理将其索引化以提供高效的数据库查找。TSAR 系统有一个关于所有数据的统一的逻辑视图，可以充分利用多层网络的特质，从而提升其性能和功能。

原则 3：支持以数据为中心的查询：在传感器应用中，创建允许 TSAR 通过值或属性（即地点或时间）定位数据的接口是很重要的，因此，索引元数据可以降低查找开销。

TSAR 系统设计的关键特性是基于以上原则的。代理节点使用分布式索引，传感器节点将原始数据和应用相关的元数据两者组成的数据写入本地闪存。元数据可以由 TSAR 查找和比较。例如，在基于视频的感知应用中，元数据包括描述视野的坐标、平均亮度、位移值和其他基本信息（如时间和传感器位置）。元数据的大小随应用的不同而不同，与从图像或声音数据抽取出的原始数据相比，该例中的元数据小很多。

传感器不仅在本地存储数据，而且将元数据的汇总定期地发布到附近代理。汇总包含传感器 ID、汇总产生的时间间隔（t_1，t_2）、一个识别相应数据记录（如闪存中的位置）的句柄（handle）以及一个与记录相关的元数据的粗粒度表示。例如，温度传感器的元数据汇总包括在一个时间间隔中观察到的最大和最小温度值。

代理使用发布的汇总来构造一个索引。因为是从整个系统收集的信息，所以该索引是全局的。索引提供了分布式数据的一个统一视图，应用程序通过查询该视图可以访问存储在任一传感器上的数据。执行查询时，在分布式索引中查找匹配，匹配的结果用来从传感器中检索数据。 255

TSAR 确保不丢失汇总信息（包括正在搜索的值）或无假阴性（false negative）。当然，在远处节点中可能会发生以下情况：没有在匹配汇总中找到与查询匹配的值，从而产生假阴性，这是网络资源的浪费。

TSAR 提出了一种新的索引结构——区间跳图（interval skip graph），区间跳图将跳图［ASPNES03］和区间树（一种基于区间的二叉搜索树）［CORMEN01］结合起来。跳图是一个适用于对等系统的有序的、分布式数据结构［HARVEY03］。TSAR 通过使用区间跳图来发现包含待搜索数据集合内任意值的所有区间。区间跳图有两个优点，这对于传感器网络来说是非常理想的。第一个优点是访问第一个匹配区间的搜索复杂度只有 $O(\log n)$，访问后续的匹配区间（successive interval）的复杂度是固定不变的。第二个优点是用区间索引代替数值索引，索引汇总更加方便，区间索引也适用于能量受限节点（energy constrained nodes），因为与传送所有传感器数据相比，传送汇总更加节省能量。

区间跳图能有效地查找出包含与某个查询有关的数据的传感器节点。当查询发送到网络时，传感器在本地存档（local archive）中快速地定位相关数据记录，然后将回应发给一个中层代理。为了使用这样的查找，每个传感器保存一个传感器数据档案。

尽管在资源丰富的设备（如便携式电脑）中实现归档存储很方便，但传感器的资源是有限的。因此，TSAR 存档子系统充分考虑了传感器数据的特点。例如，传感器数据的一个显著特点是时间序列的数据流，因此可以按照时间序列归档数据。

事实上，很多信号处理方案利用时间序列存储来执行操作，例如处理数字信息时有很多时间序列操作，如快速傅里叶变换（Fast Fourier Transform，FFT）、小波变换、聚类、相似度匹配、目标检测等。

如上所述，每个原始数据记录含有一个相关的元数据字段（包含一个时间戳、传感器设置、校准参数等）。原始传感器数据存储在记录的数据字段。注意，因为存储系统不知道或不关心这样的一个字段，所以该字段是不透明的和基于特定的应用的。例如，一个视频传感器可能在这个数据字段里存储二进制图像。

8.5　多分辨率数据处理

在［GANESAN03a］中提出了一个有趣的概念，即使用多分辨率方法从一个传感器网络抽取传感器数据。多分辨率是指从不同的层次观察数据，例如粗粒度层次和细粒度层次。用户可以从大的范围快速地、粗略地查看低分辨率数据，然后决定是否获取更详细的、包含更昂贵的数据集的高分辨率数据。在一些情况下，压缩的低分辨率传感器数据可用于时空查询，以获得
256］大量数据的统计估计［DAI04］。

为了反映不同分辨率层次，［GANESAN03a］提出了数据维度（data dimension）的概念。传感器数据在多维坐标轴（如时间的、空间的及多传感器模态间）具有相关性，这些相关性可以用来降低数据维数（data dimensionality）。节点可利用时间相关性，通过对自身存储的历史数据的本地处理来压缩数据。路由协议还可以利用相邻节点间数据存在的空间相关性，这样可以实现最大程度地压缩数据。

许多应用中的数据都存在时间相关性，例如在一个视频传感器网络中，新捕捉的图像往往与上一张图片有很强的关系，两张图片中的大部分背景像素点没有太大的改变。

为了获得不同分辨率（即数据维度），Ganesen 等人使用小波子带编码（wavelet subband coding，一种流行的信号处理技术），用于多分辨率分析和压缩［CORMEN01，Shanmugasundaram04］。小波与其他信号处理技术相比具有很多优点，尤其在表示时空数据集方面，例如，可将数据在多个空间和时间维度内分解。同时，为了获得更好的压缩，可以抽取数据一些重要的性质，如不同范围上的陡变（abrupt change）。当小波阈值应用在典型时间序列信号的压缩时，仅仅需要几个系数就可以合理准确地完成信号重建。

问题与练习

8.1　多项选择题

1. 以下哪个原因会导致传感器数据产生噪声？（　　）

A. 硬件/电路噪声　　B. 运行环境　　C. 测量误差　　D. 以上皆是

2. 传统数据库通常不需要数据清理，是因为（　　）。

A. 假定已经由独立数据库功能模块离线清理掉噪声数据

B. 数据源有明确的数据输入操作或事务活动

C. 传统数据库占据太多空间，不容易进行数据清理

D. A 和 B

3. 关于贝叶斯清理，下列哪项不正确？（　　）

A. 贝叶斯清理包含清理模块和查询处理模块。

B. 传统查询方法假定每个读数均是一个单一值，这种方法可以用于查询处理模块。

C. 先验知识是真实传感器读数的分布。

D. 实时数据清理将真实数值的先验知识、传感器的误差模型和观察到的噪声数值集中在一个步骤中。 257

4. TinyDB 有以下哪些特点？（　　）

A. 数据查询命令格式与 SQL 相似。

B. 基于事件的查询能避免轮询开销。

C. 不能运行数据聚合查询。

D. A 和 B

5. 以下特点中哪项不是数据聚合方案所具有的？（　　）

A. 通常用于网内的数据减少。

B. AIDA 的功能包括两个部件，一个是功能单元，用来聚合和分离网络包（单元）；另一个是 AIDA 聚合控制单元，用来自适应地调整时帧的设定和期望的聚合程度的数值。

C. AIDA 能动态地调整聚合的程度。

D. AIDA 对其他网络协议层是不透明的。

8.2　解释贝叶斯滤波通用模型，指出如何实时清理数据噪声。

8.3　贝叶斯清理如何使用先验知识去除不确定性？

8.4　TinyDB 的主要特点是什么？ 258

8.5　说明数据聚合与网内路由协议的关系。

第五部分

Wireless Sensor Networks: Principles and Practice

高级话题

第9章

Wireless Sensor Networks: Principles and Practice

传感器定位

本章将讨论无线传感器网络的定位方法，即传感器如何利用信息交换确定目标传感器节点的大致位置。我们首先基于文献［Xiang04］讨论传感器节点定位的基本知识，然后举例说明一些经典的定位算法。

在许多无线网络中，节点定位都是一个重要且有趣的研究方向。本章将介绍七个比较好的节点定位方法，它们都具有优美的数学模型。为了保持这些算法的原始含义，本书在引用时保留了原始的数学符号和算法程序。

9.1 引言

虽然研究者已在一些领域（诸如具有自主控制和车载导航的移动机器人、虚拟现实系统以及蜂窝网络中的用户定位与跟踪）对定位问题进行了研究，但在无线传感器网络中确定传感器节点位置的这个关键问题仍有待解决。

传感器网络的典型形式是一个分层的网络协议栈。在应用层，位置感知程序需要节点定位。传感器节点的位置信息通常是传感器采集的数据中不可或缺的一部分。例如，使用传感器网络探测和跟踪目标，在确定探测目标的位置时，需要每个传感器的物理位置。在网络层，传感器网络的众多通信协议是建立在已知传感器地理位置的基础上的。例如，知道位置信息和传输范围就可以使用地理路由算法，该算法通过多跳传感器网络传播信息。

在大多数情况下，传感器部署后的位置信息是未知的，也没有设施对其定位。因此，需要找到一些方法定位传感器网络中部署的每一个传感器。

全球定位系统（Global Positioning System，GPS）是最常用的定位技术之一。许多应用程序是基于GPS开发的。如果每个传感器节点都安装GPS，那么就可以被定位，但这种方法是不可行的，原因有三。第一，由于视线（Line-Of-Sight，LOS）条件的限制，GPS并不总是可用的。举例来说，它无法在室内、水下或地铁中工作；第二，目前一个普通的GPS接收器的价格大约为100美元，给每个传感器节点安装GPS接收器会使成本过高，而这些传感器节点通常被设计为低成本和用后可丢弃的。第三，GPS接收器的功耗很大（相对于一个微小的传感器节点来说）。

基于前面的讨论，我们需要其他有效的节点定位系统。考虑到传感器网络的应用场景，为其设计定位系统的挑战远大于设计应用于其他领域的定位系统。传感器节点需要被设计成小型的并且是低功耗的，它们通常随机且高密度地部署在一个大的区域中。被部署后，这些传感器节点自组织成一个分布式传感器网络。理想的节点定位系统也相应地要具有低计算量和低功耗。定位系统应当能够适应没有基础设施支持的自组织部署并进行定位，也应该能够进行自身定位。定位系统能够扩展以包含大量的传感器节点，并且能够适应动态变化的环境。

9.2 定位的基本要素

大部分的定位方法都是先取得未知传感器节点和锚传感器节点（已定位）之间的近似距离或角度，然后利用几何算法计算未知传感器节点的位置。因此，对节点定位来说，最重要的部分是距离测量、角度测量和几何约束。在接下来的章节中，将讨论可以确定这些先决条件的

技术。

9.2.1 接收信号强度指示

无线传播的一个重要特性是无线信号随着发射机和接收机之间距离的增加而衰减。接收到的无线信号强度随着距离的增加呈指数级衰减。接收机可以根据接收信号强度指示（Received Signal Strength Indication，RSSI）测量衰减。RSSI通过测量接收信号的能量估计与发送器之间 262
的距离。基于发射能量可以计算传播损耗，并将这个损耗转换成近似距离。这种方法主要用于射频（Radio Frequency，RF）信号。在［Rappaport96］中，研究了无线传播模型，并使用这些模型预测与发射机给定距离的RSSI平均值。理想无线传播模型如下：

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{4\pi^2 d^2 L} \tag{9.1}$$

公式9.1使用发射机和接收机间的距离函数表示接收信号能量。在这个理想模型中，P_t是发射能量，G_t是发射机的天线增益，G_r是接收机，L是系统损耗，λ是系统波长。通常采用把G_t、G_r和L设为1的方式将它们从公式中去掉。文献［ASavvides01］使用无线集成网络传感器（Wireless Integrated Network Sensors，WINS）的传感器节点接收到的RF信号强度进行距离估计。在这个实验中，使用了不同的配置策略评估接收信号强度与发射机和接收机的距离之间的关联性，这些策略包括不同的发射机功率水平和不同的传感器部署策略。接收到的无线信号强度的能量随着距离的增加呈指数衰减，如图9-1所示。

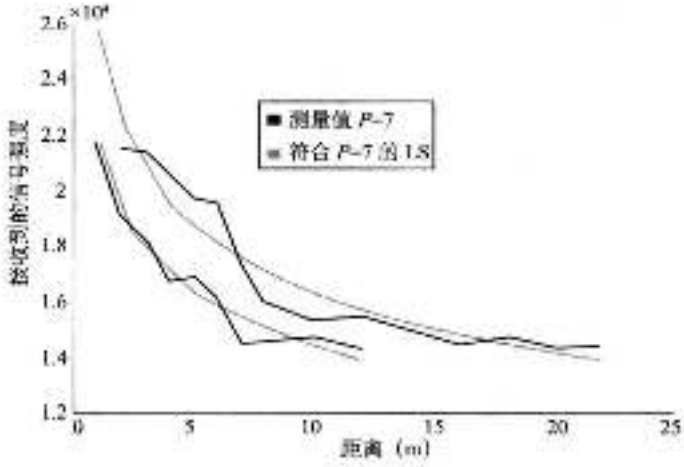


图9-1 接收到的无线信号强度的能量随发射机与接收机的距离增加呈指数衰减 263

从理论上说，无线信号的能量随到信号源距离的平方递减。其结果是，接收无线传输的节点能够使用接收信号的强度计算其到发射机的距离。RSSI提供了对这种硬件测距问题的一个可行的解决方案：使用存在于绝大多数节点间的无线信号计算距离进行定位［Jonathan08］。在实际中，RSSI距离测量存在噪声。产生噪声的原因是在真实的环境中，无线信号的传播是不均匀的。举例来说，无线信号经由柏油路面传播是不同于其经由草地的传播。物理障碍（如墙或家具）会反射或吸收无线电波。结果导致使用信号强度的距离估计的精度不如其他方法，如到达时间差法（Time Difference of Arrival，TDoA）。

对无线信号传播进行更精细的物理分析和提高传感器节点无线信号校准的精度能更好地使用RSSI数据。考虑到性价比，这种更复杂的RSSI使用方法也许能够证明它是一种优越的测距

技术。遗憾的是，其必需的技术目前还不存在。

9.2.2　到达时间

基于光波的传播速度和测量得到的无线信号在节点间的经过时间，可估计发射机和接收机之间的距离。这种方法可以应用于不同的信号，例如 RF、声波、红外线和超声波。该技术的实现依赖于对到达时间（Time of Arrival，ToA）的测量。到达时间可以使用先进的测时技术进行测量。GPS 使用的就是这样一种先进的测时技术进行距离估计［HHW97］。在 GPS 系统中，每颗卫星（发射机）发射一种唯一的编码，接收机复制这个编码，然后逐步调整它的内部时钟，使其与接收到的编码一致，这个过程称为锁定。一旦接收机锁定了一颗卫星，它就可以确定来自该卫星的无线信号的准确时间。基于这个时间，可以通过从接收时间减去已知的传输时间来确定到达时间。

虽然到达时间提供了很高的准确度，但是这种精度的测量要求传感器节点有相应的快速处理能力以解决小的计时误差。

9.2.3　到达时间差

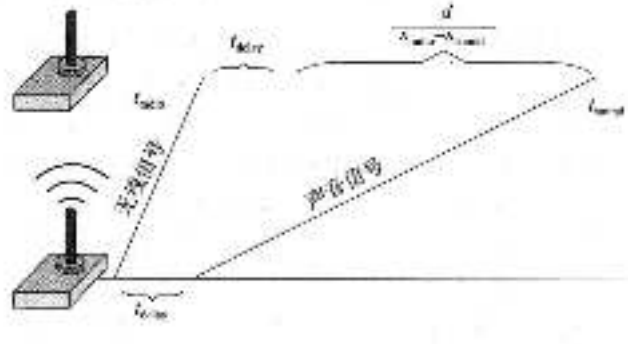
264

发射机和接收机间的距离可以使用各种不同速度的通信介质的到达时间差进行测量。例如，在传感器节点间使用超声波和无线信号两种不同的通信方式测量到达时间。超声波和无线信号的传播速度明显不同。由于有这样的差异，无线信号用来同步发射机和接收机，超声波信号则用来估计两者之间的距离。到达时间差技术已经在 Active Bat 项目［BWarneke01］和 AHLoS 项目中［ASavvides01］使用。

在到达时间差方法中，每个节点都配备了扬声器和麦克风。有些系统使用了超声波，而其他系统则使用了可听频率。但是，这个一般性的数学方法是与实际硬件无关的。在到达时间差方法中，发射机先发出无线信号（如图 9-2 所示）。它等待一个固定的时间间隔 t_{fix}（可能为零），然后扬声器产生一个固定模式的线性调频信号。当监听节点捕捉到无线信号时，会记录当前时刻 t_{radio}，并开启它们的麦克风。当麦克风检测到线性调频模式的信号时，监听节点会再次记录当前时刻 t_{sound}。有了 t_{radio}、t_{sound} 和 t_{fix}，监听节点就能够计算出发射机与它们之间的距离 d，这源于空气中无线电波的传播速度大于声波的传播速度。

$$d = (t_{radio} - t_{sound}) \approx (t_{sound} - t_{radio} - t_{fix}) \tag{9.2}$$

到达时间差方法在视线（LOS）条件下是非常精确的；这种方法在没有回声的地方并且扬声器和麦克风进行相互调校之后的效果最好。目前有几个研究组正在研究相关问题，这些问题的解决有助于改善该领域的准确度。



265

图 9-2　到达时间差方法说明

9.2.4 到达角度

到达角度（Angle of Arrival，AoA）是指接收机从发射机接收到的信号的角度。到达角度系统能够估计接收到的信号角度并且能够使用简单的几何关系估计发射机和接收机的相对位置。到达角度也可以结合距离估计推导出相对位置。

到达角度系统的实现依赖于具有天线阵列的智能天线，这种天线能够测量信号到达的角度。智能天线是一个连接着数字信号处理器的天线阵列。这样的配置不但能够估计到达角度，而且能够通过组合分集增益、阵列增益和干扰抑制显著地增强无线链路的容量。到达角度技术有两个主要的不足，使得它不适于传感器网络：第一，复杂的天线阵列的开销很大。第二，对具有大量节点的系统，到达时间技术的扩展性不足。

9.2.5 三角测量

三角测量是一种几何方法，它利用到达角度确定传感器的位置。根据每个锚节点的角度和一些参考框架中未知的传感器节点，未知传感器节点的位置可以使用三角法则中的正弦和余弦计算出来。三角测量的计算方法如图9-3所示［CSavarese02］。

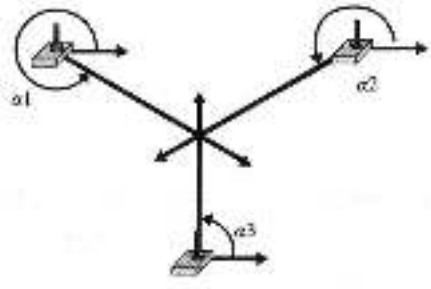


图9-3 三角测量

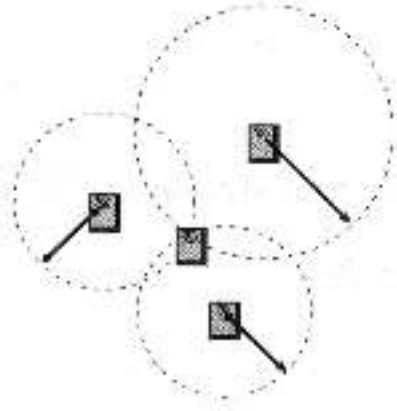


图9-4 三边测量

9.2.6 三边测量

三边测量是一种几何方法，它使用三个锚节点和一个未知节点间的距离确定该未知节点的位置。

在二维平面上，使用三个参考节点可以确定未知传感器节点的位置。未知传感器节点的位置可以通过计算三个圆的交叉点进行估计。图9-4说明了该计算的几何约束［ASavvides01］。

9.2.7 多边定位

未知节点的位置也可以利用多于三个锚节点的多边定位技术进行估计。在文献［JBeutel99］中，Beutel使用最小二乘法研究了多边定位。

在三维空间中，给定n个锚节点和它们到未知节点的距离，有

$$\begin{bmatrix} d_1^2 \\ d_2^2 \\ \vdots \\ d_n^2 \end{bmatrix} = \begin{bmatrix} (x_1-u_x)^2+(y_1-u_y)^2+(z_1-u_z)^2 \\ (x_2-u_x)^2+(y_2-u_y)^2+(z_2-u_z)^2 \\ \vdots \\ (x_n-u_x)^2+(y_n-u_y)^2+(z_n-u_z)^2 \end{bmatrix} \tag{9.3}$$

这里，d_i 是第 i 个锚节点到未知节点的距离。(x_i, y_i, z_i) 是第 i 个锚节点在三维空间中的位置，(u_x, u_y, u_z) 是未知节点在三维空间中的位置。

266
|
267

上述公式通过线性计算可以转换为如下形式：

$$Au = b \tag{9.4}$$

$$\begin{bmatrix} d_1^2 \\ d_2^2 \\ \vdots \\ d_n^2 \end{bmatrix} = \begin{bmatrix} (x_1-u_x)^2 + (y_1-u_y)^2 \\ (x_2-u_x)^2 + (y_2-u_y)^2 \\ \vdots \\ (x_n-u_x)^2 + (y_n-u_y)^2 \end{bmatrix} \tag{9.5}$$

$$u = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{9.6}$$

$$b = \begin{bmatrix} d_1^2 - d_n^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 - z_1^2 + z_n^2 \\ d_2^2 - d_n^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 - z_2^2 + z_n^2 \\ \vdots \\ d_{n-1}^2 - d_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 - z_{n-1}^2 + z_n^2 \end{bmatrix} \tag{9.7}$$

u 可以通过下式计算得到［GGolub96］：

$$u = (A'A)^{-1} * A'b \tag{9.8}$$

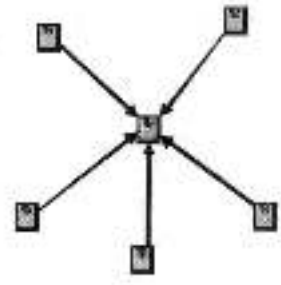


图 9-5　多边定位

图 9-5 说明该计算的几何场景［ASavvides01］。

9.3　使用移动机器人进行传感器定位

容迟传感器网络

针对处于缺乏一直在线（always-on）基础设施的移动环境中的传感器节点，文献［Kavek04］提出了容迟传感器网络（Delay-Tolerant Network，DTN）结构。在延迟的时间段内，这些传感器节点被认为具有监测环境的能力。通信是基于消息交换的抽象，而不是使用分组

268

交换。

可以定义“束”的概念来描述非交互通信的中等长度的消息。使用这一概念有助于网络管理，因为它允许网络路径选择和调度机制预先知道请求传输的数据的大小和相关性能要求等信息。

Pathirana 等［Pubudu05］提出了一种新颖的容迟传感器网络的定位方法。该方法利用了数据采集移动机器人上每个传感器设备的 RSSI 测量结果。在容迟传感器网络中，使用一个或多个移动机器人进行节点定位，消除了小型设备处理能力的限制。利用机器人的移动性减少定位的错误和静态参考定位信标的数量。

［IPeterson09］提出了一种基于扩展卡尔曼滤波（Robust Extended Kalman Filter，REKF）的状态估计方法，用于容迟传感器网络中的节点定位。在该方法中，定位被定义成非线性动态系统的在线估计。它的模型消除了明显的错误数值和测量误差。

下面讨论该系统的动态模型和非线性测量模型。

1. 系统动态模型

假设传感器随机分布在待监测环境中。网络中 n 个传感器和移动机器人的动态模型可以用二维笛卡儿坐标表示［ASavkin03］：

$$\dot{x}(t) = Ax(t) + B_1 u(t) + B_2 \omega(t); \tag{9.9}$$

这里，$A = \begin{bmatrix} \Theta & 0 \\ & \ddots \\ 0 & \Theta \end{bmatrix}$，$B_1 = \begin{bmatrix} \Phi \\ \vdots \\ \Phi \end{bmatrix}$，$B_2 = \begin{bmatrix} \Phi & 0 \\ & \ddots \\ 0 & \Phi \end{bmatrix}$，$\Theta = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$，

$$\Phi = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \tag{9.10}$$

动态状态向量 $x(t) = [x_1(t) \ldots x_i(t) \ldots x_n(t)]'$ 和 $x_i(t) = [x_i(t)\ Y_i(t)\ \dot{X}_i(t)\ \dot{Y}_i(t)]'$，这里 $i \in [1 \ldots n]$，$x_i(t)$ 和 $Y_i(t)$ 表示移动机器人的第 i 个传感器（Sensor$_i$）在时刻 t 的位置，而它们的一阶导数 $\dot{X}_i(t)$ 和 $\dot{Y}_i(t)$ 表示沿 X 和 Y 方向的速度。 269

如果 $x_r(t) = [x_r(t)\ \ y_r(t)\ \ \dot{x}_r(t)\ \ \dot{y}_r(t)]'$ 表示移动机器人的绝对状态（分别是 X 和 Y 方向上的位置和速度），并且 $x_i'(t) = [x_i'(t)\ \ y_i'(t)\ \ \dot{x}_i'(t)\ \dot{y}_i'(t)]'$ 表示相同顺序的 Sensor$_i$ 的绝对状态，那么 $x_i(t) = x_r(t) - x_i'(t)$。

假设 $u(t)$ 是参照各自加速度计读数的移动机器人的 2D 行驶/加速命令，$\omega(t)$ 表示传感器移动时的其未知的 2D 行驶/加速命令。假设传感器是静止的并设 $\omega(t) = 0$。

这个系统可以表示成输入为（$u(t)$）、测量结果为（y）的形式，如图 9-6 所示。由于传感器是静止的，所以省略了 B_2。现在的问题是从测量结果 y 估计出状态 x。

由于传感器位置是未知的，因此在算法的开始，假设需要定位的传感器位于位置（0，0）。该算法能够保证这个假设的状态能够收敛到实际的状态，并能进一步在规定时间内计算出未知节点的位置（移动机器人的位置/状态是已知的）。

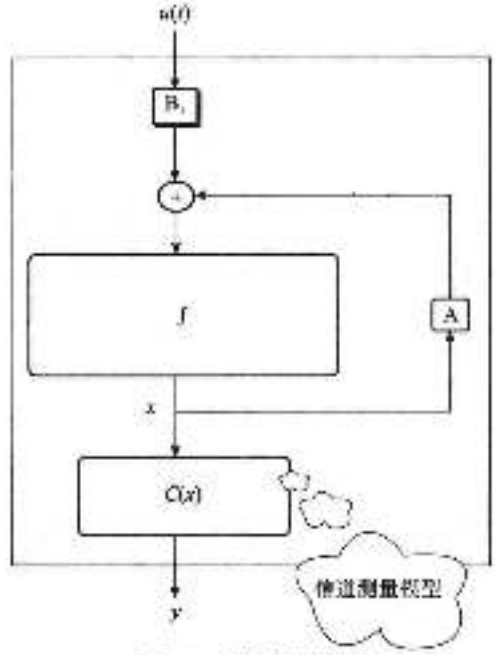


图 9-6 定位估计系统 270

2. RSSI 测量模型

根据前面的讨论，两个通信实体的距离可以使用接收机的前向链路 RSSI 进行计算。当存在多个发射机时，数据关联会明确的。通过检查数据包中的源（发射机）标识符可以精确地获知测量数据是来自于哪个发射机。

针对本例，在移动机器人中 RSSI 是以分贝为单位进行测量的。如果 Sensor_i 表示第 i 个传感器（参见图 9-7），那么可以采用 [HXia96] 中的方法计算 Sensor_i 的 RSSI 值 $p_i(t)$：

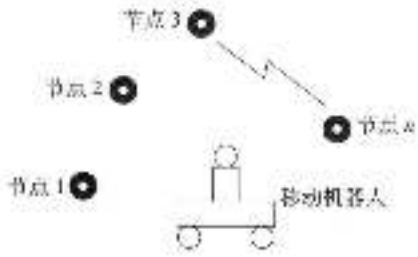


图 9-7　网络几何形状

$$p_i(t) = p_n - 10\varepsilon\log d_i(t) + v_i(t) \tag{9.11}$$

这里，p_n 是由传输功率、波长和移动机器人的天线增益确定的常数，ε 被称为路径损耗比（通常情况下，值为 2～4），$v_i(t)$ 表示测量中不确定成分的对数，$d_i(t)$ 是移动机器人与 Sensor_i 之间的距离。它可以用第 i 个传感器相对于该移动机器人的位置表示，即 $(X_i(t), Y_i(t))$。

[271]

$$d_i(t) = (X_i(t)^2 + Y_i(t)^2)^{1/2} \tag{9.12}$$

当移动机器人在其覆盖范围内移动时，观测向量

$$y(t) = \begin{bmatrix} p_1(t) \\ \vdots \\ p_n(t) \end{bmatrix} \tag{9.13}$$

会进行采样。这个移动机器人利用 n 个传感器组成的测量系统的测量公式为：

$$y(t) = G(x(t)) + v(t) \tag{9.14}$$

这里，$v(t) = [v_1(t) \cdots v_n(t)]'$，$G(x(t)) = \begin{bmatrix} p_n - 10\varepsilon\log(X_1(t)^2 + Y_1(t)^2) \\ \vdots \\ p_n - 10\varepsilon\log(X_n(t)^2 + Y_n(t)^2) \end{bmatrix}$

随机扩展卡尔曼滤波（REKF）的简要介绍如下：首先，使用运用简单运动学方程推导出的差分方程集的状态空间模型。该模型有两种噪声输入：1）测量噪声（所有测量系统都会考虑此因素），v 在 $y = G(x) + v$；2）ω 加速度，由于其是未知的，也被认为是一种噪声。在这个应用中，初始条件明显错误的原因在于传感器节点的位置是未知的。如果在容迟传感器网络中使用 REKF，那么在一个相应的时间间隔内，第 i 个系统（移动机器人和 Sensor_i）可以表示成非线性的不确定系统，并有下面的积分二次约束（Integral Quadratic Constraint，IQC）：

$$\begin{aligned} &(x(0) - x_0)'N_i(x(0) - x_i) \\ &+ \frac{1}{2}\int_0^t (\omega(s)'Q_i(t)\omega(s)) + v(t)'R_i(t)v(t)dt \\ &\leq d + \frac{1}{2}\int_0^t z(t)'z(t)dt \end{aligned} \tag{9.15}$$

这里，$Q_i > 0$，$R_i > 0$ 且 $N_i > 0$（$i \in \{1, 2, 3\}$）是系统 i 的加权矩阵。初始状态（x_i）是各系统的初始估计状态。注意，这个初始状态可以根据前一系统的最终状态和网络的其他有效数据（即机器人的位置和速度）推导得出。考虑到该方程形式的不确定关系，应将系统的固有测量噪声、移动机器人未知的加速度和初始条件中的不确定性作为有界的不确定输入。特别地，有标准范数有界不确定量的测量公式可以写成如下形式：

[272]

$$y = C(x) + \delta C(x) + v_0 \tag{9.16}$$

这里，$|\delta| \leqslant \xi$，ξ 是一个常数，表示噪声的乘数有界部分的上限值。选择 $z = \xi C(x)$ 和 $v = \delta C(x)$：

$$\int_0^T |v| dt < \int_0^T z' z dt \tag{9.17}$$

认为 v_i 和 ω 中相应的不确定量 ω_i 满足边界

$$\Phi(x(0)) + \int_0^T [\omega_0(t)' Q \omega_0(t) + v_0(t)' R v_0(t)] dt \leqslant d \tag{9.18}$$

这个更加逼真的方式去除了算法开发中所有的噪声模型假设。这个方式也保证了协议的健壮性。Pathirana 等 [Pubudu05] 基于这些算法进行实验，结果显示可以收敛到实际的传感器节点位置（见图 9-8）。

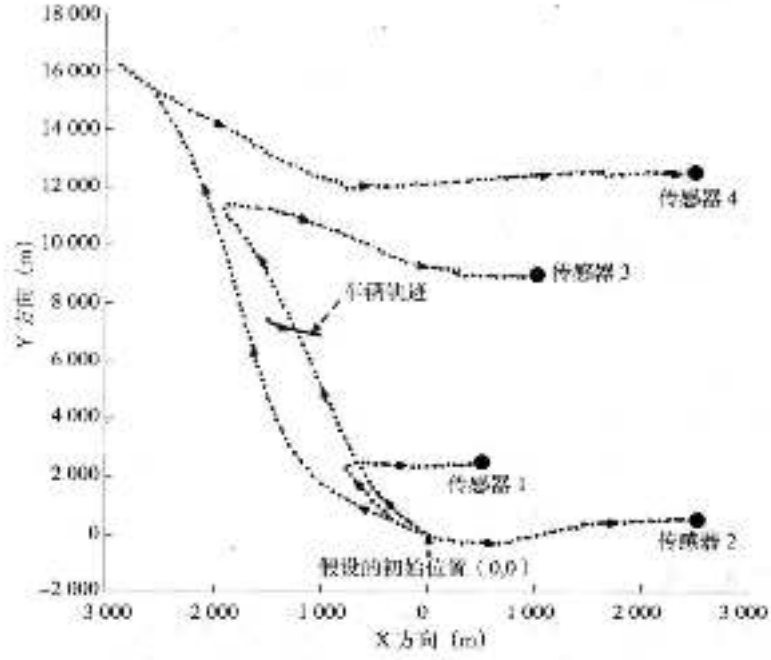


图 9-8 收敛到实际的传感器节点位置的位置估计轨迹 [273]

9.4 多维标度节点定位

大多数的现有定位算法都是使用基于 TOA、TDOA 和 RSSI 的范围测量结果的三边测量或多边测量方法。文献 [Xiang04] 使用了降维技术估计传感器节点在二维或三维空间中的坐标。该文献提出了一种基于降维技术——多维标度（MultiDimensional Scaling，MDS）的集中式节点定位技术。它利用逐对节点距离获得节点在二维或三维空间中的位置。如果所有逐对节点间距离都是已知的，通过一个简单的特征分解就可以获得这些节点的位置。

为了估计分布式无线自组织传感器网络中所有节点的位置，必须知道一小部分节点的位置，不论是通过手工配置获取还是装备 GPS 来获取。已知位置信息的节点称为锚节点，没有位置信息的节点称为未知节点。在锚节点的协助下可估计所有节点的位置。通常，锚节点向它的相邻节点广播自身的位置，相邻的未知位置节点测量与它们相邻节点的空间关系，并利用锚节

点广播的位置信息估计自身位置。对于一个未知节点来说，一旦估计出自身位置，它就变成了锚节点并且能够协助其他未知节点估计自身位置。

多维标度（MDS）已经被广泛应用于分析待定位目标节点的数据的相异性。它可以用来发现数据中的网络拓扑结构［IBorg97］。多维标度可作为一种数据分析方法来发现几何空间中隐藏的距离和模型数据背后的维度。

多维标度通常是从标注对象在二维空间中的任一坐标开始。接着，它计算所有点构成的坐标对的欧式距离组成距离矩阵。然后，多维标度把这个矩阵与已知距离进行比较。最后，每个对象的坐标都被调整到最优（协强系数 stress 最小）的结果。

使用多维标度进行位置估计的优势在于，即使使用的是错误的距离信息，它仍然能够得到精确的位置估计。虽然有很多种多维标度方法，但这里只讨论经典多维标度和它的迭代优化。

9.4.1　经典多维标度

使用 $T=\lfloor T_x\rfloor_{,,}$ 表示 n 个传感器节点在二维空间中的真实位置。使用 $d_{ij}(T)$ 表示传感
274 器 i 和 j 间的距离。基于它们在 T 中的位置，可以得到：

$$d_{ij}(T)=\left(\sum_{x=1}^{2}(t_{ix}-t_{jx})^2\right)^{1/2} \tag{9.19}$$

节点 i 和 j 间已记录的距离记为 δ_{ij}。忽略距离测量中的误差，δ_{ij} 等于 $d_{ij}(T)$。$X=\lfloor x_i\rfloor_{,,}$ 表示二维空间中 n 个传感器节点的估计位置。如果已取得 T 中所有节点的逐对距离，则可以使用经典多维标度算法估计传感器节点的位置：

1）计算距离平方的矩阵 D^2。

2）计算矩阵 J，$J=I-e\times e^T/n$，其中 $e=(1,1,\cdots,1)$。

3）对矩阵应用双定心法 $H=-(1/2)JD^2J$。

4）计算特征分解 $H=UVU^T$。

5）如果需要 i 维的解决方法（在二维环境中 $i=2$），则 V_i 是矩阵最大的矩阵 i 特征值，U_i 是 U 的前 i 列。经典标度的坐标矩阵是 $X=U_iV_i^{1/2}$。

9.4.2　迭代多维标度

如果某传感器节点对间的距离未知，则可使用迭代多维标度法计算邻近传感器节点的相对坐标。迭代多维标度是一种迭代算法，它基于二维空间中传感器节点位置估计的多元优化。由于只有部分节点对的距离是有效的，对一些 i，j 来说，δ_{ij} 是未知的。为了能够进行计算，如果 δ_{ij} 已知，则定义权重 ω_{ij} 为 1，否则为 ω_{ij} 为 0，并假设

$$\delta_{ij}=d_{ij(T)} \tag{9.20}$$

X 被随机初始化为 $X^{[0]}$，并被迭代算法逐步更新为 $X^{[1]}$，$X^{[2]}$，$X^{[3]}\cdots$ 直到近似为 T。

通过最小化下面的方程，可以使矩阵 X 近似为 T：

$$\sigma(X)=\sum_{i<j}\omega_{ij}(d_{ij}(X)-\delta_{ij})^2 \tag{9.21}$$

当梯度等于 0 时，方程得到最小值。迭代算法更新后的公式如下：

275

$$X=V^{-1}\left(\frac{\omega_{ij}\delta_{ij}}{d_{ij}(T)}A_{ij}\right)T \tag{9.22}$$

其中 A_{ij} 是一个矩阵，它的 $a_{ii}=a_{jj}=1$，$a_{ij}=a_{ji}=-1$，其他所有元素都为零，并有

$$V=\sum_{i<j}\omega_{ij}A_{ij} \tag{9.23}$$

如果 V^{-1} 不存在，则使用方程 9.24［IBorg97］所示的 V 的广义逆矩阵（Moore-Penrose）代替它

$$V^{-1}=(V+11')^{-1}-n^{-2}11' \tag{9.24}$$

迭代的步骤总结如下：

1）随机初始化 $X^{[1]}$，设 $T=X^{[1]}$ 和 $k=0$，并计算 $\sigma(X^{[1]})$。

2）$k=k+1$。

3）使用更新公式计算 $X^{[k]}$ 和 $\sigma(X^{[k]})$。

4）如果 $\sigma(X^{[k-1]})-\sigma(X^{[k]})<\varepsilon$，$\varepsilon$ 是一个小的正常数，则停止；否则设 $T=X^{[k]}$，并转到步骤 2。（ε 是基于精度要求的经验阈值。通常设 ε 为平均跳距的 5%。该算法得到的传感器节点的相对位置保存在 $X^{[k]}$ 中。）

这些多维标度技术以分布式的方式对每组邻近的传感器节点估算出局部相对位置图，这些局部图再拼接到一起就形成了全网节点物理位置图。接下来将讲述分布式传感器定位方法的细节。

1. 跳距和测距估计

在文献［Xiang04］中，距离测量模型是基于 RSSI 的。跳距被定义为传感器的 RF 通信半径。接收机可以通过测量发射机到接收机的 RF 信号强度衰减估计其到发射机的距离。例如，图 9-9 中有四个传感器节点 A、B、C 和 D。跳距为 r_0。A 和 D 间的距离 r_{ad} 可以根据 D 处 A 的信号强度和 r_0 推导出来。

其他测距方法，如 TOA、TDOA、AOA 和超声波，也可以应用到上述情况下。虽然它们可能会生成比 RSSI 更精确的距离测量结果，但是每个传感器都需要更复杂的硬件。

2. 相对位置对齐到物理位置

在估计出相邻节点组的逐对距离后，使用多维标度技术能够计算出它们相对位置的局部图。要利用分布式定位方法计算所有传感器的物理位置，就需要把相对位置对齐到物理位置。 276
这需要已知位置的传感器的协助。在二维情况下，定位传感器的邻近组中剩余的节点至少需要三个传感器节点的物理位置。这样，每组邻近的传感器中都必须含有至少三个已知物理位置的节点。这些传感器节点可以是锚节点，也可以是已通过计算获知自身物理位置的节点。

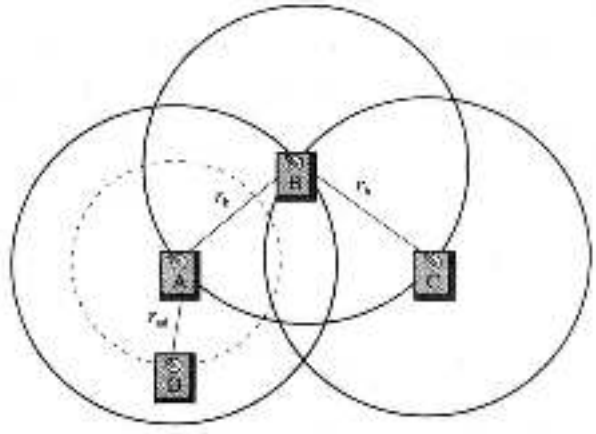


图 9-9 跳距和信号强度

对齐的过程包括移动、旋转和坐标反射。使用 $R=[r_i]_{2\times n}=(R_1, R_2\cdots, R_n)$ 表示二维空间中 n 个传感器节点的相对位置集合。$T=[t_i]_{2\times n}=(T_1, T_2, \cdots, T_n)$ 表示二维空间中 n 个传感器节点的真实位置集合。在下面的讲解中，假设节点 1、2 和 3 是锚节点。向量 R_i 可通过 $R_i^{[1]}=R_i+X$ 移动到 $R_i^{[2]}$。它可以顺时针旋转角度 α 之后，成为 $R_i^{[3]}=Q_\alpha R_i$，其中：

$$Q_1 = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{9.25}$$

它也可以通过点线反射为：

277

$$S = \begin{bmatrix} \cos(\beta/2) \\ \sin(\beta/2) \end{bmatrix} \tag{9.26}$$

对 $R_2^{(1)} = Q_2 R_2$，其中

$$Q_2 = \begin{bmatrix} \cos(\beta) & \sin(\beta) \\ \sin(\beta) & -\cos(\beta) \end{bmatrix} \tag{9.27}$$

在对齐之前，只知道 R 和三个传感器节点 T_1、T_2、T_3或更多传感器节点的物理位置。给定这些传感器节点的位置，就可以计算出 T_4、$T_5 \cdots T_n$的位置。基于这些规则，有

$$(T_1 - T_1,\ T_2 - T_1,\ T_3 - T_1) = Q_1 Q_2\ (R_1 - R_1,\ R_2 - R_1,\ R_3 - R_1) \tag{9.28}$$

已知 R_1，R_2，R_3，T_1，T_2和 T_3，可以计算

$$Q = Q_1 Q_2 = \left(\frac{R_1 - R_1,\ R_2 - R_1,\ R_3 - R_1}{T_1 - T_1,\ T_2 - T_1,\ T_2 - T_1} \right) \tag{9.29}$$

则，$(T_4,\ T_5,\ \cdots,\ T_n)$ 可以由下面的公式计算出来：

$$\begin{aligned} &(T_4 - T_1,\ T_5 - T_1,\ \cdots,\ T_n - T_1) = Q\ (R_4 - R_1,\ R_5 - R_1,\ \cdots,\ R_n - R_1) \\ &(T_4,\ T_5,\ \cdots,\ T_n) = Q\ (R_4 - R_1,\ R_5 - R_1,\ \cdots,\ R_n - R_1) + (T_1,\ T_1,\ \cdots,\ T_1) \end{aligned} \tag{9.30}$$

3. 分布式物理位置估计

标记为“起始锚”的锚节点向全网络发起洪泛广播。当被称为“终止锚”的锚节点收到洪泛信息时，它们向起始锚节点反馈自身的位置信息和从起始锚节点到它们的反向路径。这样，起始锚节点就获知了终止锚节点的位置和到达它们的相应路径。起始锚节点利用这些路径信息估计距离其一跳范围内节点的位置。图 9-10 说明了这个过程：A 是起始锚节点，D 和 H 是终止锚节点。A 知道 D 和 H 的位置以及到达它们的路线，分别是（A，B，C，D）和（A，E，F，G，H）。A 估计出 B 的位置 B’ 在 AD 虚线上，E 的位置 E’ 在 AH 虚线上。A 也估计出 AD 和 AH 方向上各自的平均跳距。

有了使用 RSSI 方法获得的相邻节点间的逐对距离，多维标度就可计算出相邻传感器节点的局部图或相对位置。在图 9-10 中，A 计算出了相邻节点 A、B、E、J 和 K 的相对位置。通过把 A、B 和 E 的相对位置对齐到它们的物理位置，J 和 K 的物理位置也可以通过计算得到。以同样的方式，从起始锚节点到终止锚节点路径上的传感器节点也可进行本地映射和对齐。图 9-11 说明了从起始锚节点到终止锚节点的位置传播估计的过程。

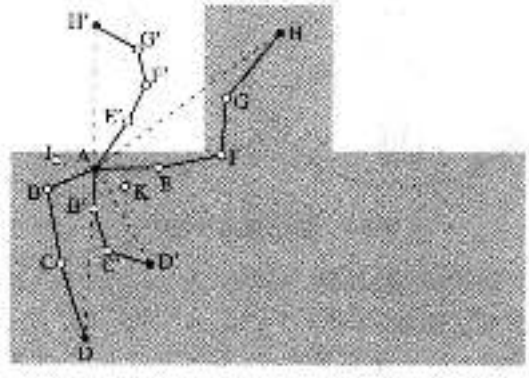


图 9-10 相邻节点位置估计

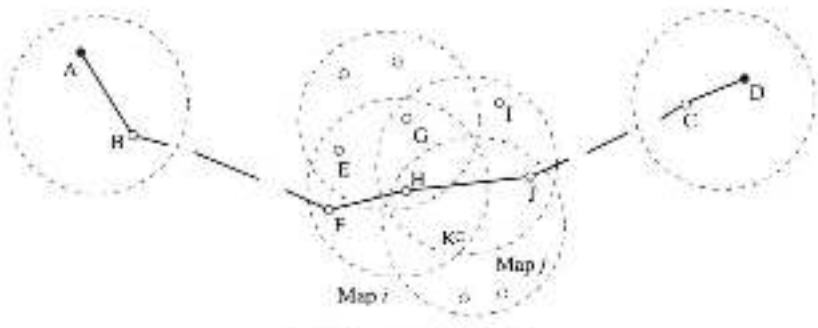


图9-11 位置估计传播

在图9-11中，A是起始锚节点，D是终止锚节点。其余的传感器节点的传播贯穿从A到D的路线，每个本地映射用虚线的椭圆表示。映射i包含相邻的传感器节点E、F、G、H和K。由于已计算出E、F和G的物理位置，使用前面提到的多维标度和对齐技术可计算出H和K的物理位置。接着，相邻的传感器节点H、K、I、J和C构成了映射j，并可进一步计算出I和J的位置。

从起始锚节点到终止锚节点的路线周围所有节点的位置都可以估算出来，包括终止锚节点。例如，在图9-11中，节点E、F和G各自的估计位置是E′、F′和G′。给定了G的物理位置，则可以比较G′和G。如果它们不相等，则以A为中心旋转$\angle G'AG$进行对齐，然后缩放AG′到AG。路径上所有传感器节点的坐标都可以进行同样的对齐操作，例如E′和F′。通常，E′和F′的位置都是正确的并且近似于各自的真实位置。这个位置估计过程是在起始锚节点到终止锚节点的路径上迭代执行的，直到估计的位置收敛。

文献[Xiang04]的实验结果说明，这个过程通常能够得到路径上传感器节点的高精度位置估计。这些有着高精度位置估计的节点可以被视为锚节点，它们可以初始化更多路线上传感器节点的位置估计。这种估计方法可以在自组织传感器网络的不同部分并行执行，直到所有的传感器都被精确定位。

9.5 无线传感器网络中的定位

在文献[Masoomeh07]中，作者研究了一个有一小部分传感器节点装备了硬件（诸如GPS）的网络，这部分传感器节点总是能够感知到自身位置。除此之外，所有的传感器节点都是一样的。

无线距离被建模成不规则的，前提是假设传感器节点无线通信距离的正常分布是均值为r，标准差为σ。模拟器使用σ随机确定每个包的发送者和接收者是否在无线距离内。最初，节点随机传播到整个网络。

传感器p的一跳邻居节点是那些能够直接与其通信的节点。该算法并不需要非常严格的同步时钟。在一个时间段内，每个节点和种子能够向任一方向移动距离v，其中$0 \leq v \leq v_{max}$。这些节点知道v_{max}，但在任一时间段内它们不知道v的取值或移动的方向。

9.5.1 蒙特卡洛方法

系统状态需要通过一些观测进行估计时，系统可以使用贝叶斯模型表示，该系统状态的后验分布只依赖于当前的观测和状态[ADoucet01]。在动态系统中顺序得到观测值，并使用新到达的观测更新后验分布。蒙特卡洛方法估计一组采样的分布状态，并更新这些采样作为到达的新观测。

278 ↓ 279

280

虽然提出了不同的蒙特卡洛方法，但此处只关注粒子滤波方法［ADoucet01］。这一技术用于估计机器人的位置，它是完全分布式的并且易于实现［DFox99］。该方法的目标是把系统的数据分布表示成一组权重为 N 的采样：

$$p(S_t \mid Q_0 \cdots t) \approx \{s_t^{(i)}, w_t^{(i)}\}_{i=1,\cdots,N} \tag{9.31}$$

其中，$p(S_t \mid Q_0 \cdots t)$ 是系统状态在时刻 t 的分布，$S_t^{(i)}$ 是系统状态在时刻 t 的一个采样，$w_t^{(i)}$ 是非负的数值权重，它们的和为 1。需要一组最小数量的采样使这组采样收敛于系统的后验分布（详情见 Doucet 等的［ADoucet01］）。

蒙特卡洛方法的步骤如下：

初始化：从初始系统分布 $p(S_0)$ 中选择 N 个采样。

采样：N 个采样值，$\tilde{s}_t^i$，$i=1$，…，N，来自于分布 $p(S_t \mid S_{t-1})$，其中 $p(S_t \mid S_{t-1})$ 是转移方程或移动模型。计算每个采样的权重并归一化这个值，$w^{(i)} = \eta \tilde{w}^{(i)}$，其中 η 是归一化因子。

重复采样：依据权重从当前采样集中选择（进行替换）N 个采样。

每个节点都使用一组加权的采样表示自身可能的位置。使用蒙特卡洛方法，每个节点利用每个观测更新自身的采样。在接下来的讨论中，$d(a, b)$ 表示位置 a 和 b 间的距离，r 表示理想的无线距离。首先介绍算法（1）。

9.5.2 算法（1）

在算法（1）中，每个节点都有一组可能的位置（采样）。这些采样都有不同的权重，这是对它们质量的估计。从概念上讲，此值代表了在给定邻居节点的估计位置的条件下当前节点位置的可能性。该算法的步骤如下：

步骤 1：**初始化**。节点不知道它们自身的位置，因此从所有传感器节点中随机选择第一组采样值，此时仅使用附近的种子给采样分配权重。

步骤 2：**采样**。基于下面的转移公式，节点生成新的采样：

[281]

$$p(S_t \mid S_{t-1}) = \begin{cases} \dfrac{1}{\pi (v_{max} t)^2} & d(S_t, S_{t-1}) \le v_{max} \\ 0 & d(s_t, S_{t-1}) > v_{max} \end{cases} \tag{9.32}$$

其中，v_{max} 是节点的最大速度，$d(S_t, S_{t-1})$ 表示采样在时刻 t 和 t-1 的位置距离。

在每个时间步骤内，新的采样从当前采样生成，方法是在以当前采样 d 位置为圆心的圆中随机选择一个点，该圆的半径是 $(v_{max} + \alpha)$。如果 α 非常小，当传感器移动很慢时，新的采样就没有足够的变化率。α 的经验取值是 $\alpha = 0.1r$。

在选择一个采样后，按照如下方法，相邻节点信息可以用于生成它的权重。节点 p 的采样 s 的权重 $w_s(p)$ 按如下方法计算：对于节点 p 的每个相邻节点 q，找到一个针对采样 s 的部分权重 $w_s'(q)$，采样 s 的权重是节点 p 的每个相邻节点提供的部分权重之和：

$$w_s(p) = \prod_{q=1}^{k} w_s'(q) \tag{9.33}$$

其中，k 是节点 p 的一跳和两跳相邻节点的数量，q 是节点 p 的相邻节点。

采样 s 对应于一跳种子相邻节点 q 的部分权重是：

$$w_s'(q) = \begin{cases} 1 & d(s,q) \le r \\ 0 & \text{其他} \end{cases} \tag{9.34}$$

采样 s 对应于两跳种子相邻节点 q 的部分权重是：

$$w_1'(q) = \begin{cases} 1 & r \leqslant d(s,q) \leqslant 2r \\ 0 & \text{其他} \end{cases} \tag{9.35}$$

使用节点 q 的采样 q_i 的权重 $w(q_i)$ 计算采样 s 对应于一跳种子相邻节点 q 的部分权重为：

$$w_1'(q) = \sum_{q_i} w(q_i), \text{其中 } d(s,q_i) \leqslant r + v_{max} \tag{9.36}$$

类似地，对两跳相邻节点 q，$w_2'(q)$ 按下式计算：

$$w_2'(q) = \sum_{q_i} w(q_i), \text{其中 } r - v_{max} \leqslant d(s,q_i) \leqslant 2r + v_{max} \tag{9.37}$$ 282

如果 $w_s(p)$ 比阈值 β 大，则保留采样 s。参数 β 是 [0, 1] 区间内的一个实数，它的取值依赖于节点相邻节点的数量。因此，不同的节点有不同的 β 值。β 值随着相邻节点数量的增加而减小。这是因为 $w_s(p)$ 是由节点数量决定的，其值最大为 1。节点的部分权重通常是小于 1 的，并且对应种子节点的部分权重是 0 或 1。此处使用 $\beta = (0.1)^t$，其中 t 是节点的一跳和两跳相邻节点的数量。

在计算 $w_s(p)$ 之后，对权重进行归一化，以保证它们的和为 1。因此，如果为节点 p 选择了 N 个采样，则第 i 个采样可以按照如下公式归一化为

$$\frac{w_i(p)}{\sum_{j=1}^{N} w_j(p)} \tag{9.38}$$

步骤 3：重复采样。在这一步骤中，逐步移除低权重的采样，逐步减小集合，只保留那些最高权重的采样。每个节点从它当前的集合中计算一个新的采样集合，新的采样集合包含所有旧的采样，但更新了它们的权重的概率比例。由于数量是固定的，因此低权重的采样被选中的机会小，高权重的采样在新采样集中就很可能重复。

图 9-12 显示的是 MSL 的伪代码。该算法与用来传播路由信息的距离矢量算法的基本结构相同。

```
If (node not localized or number of samples are zero)
If (node has first-hop or second-hop neighbors)
find N samples with weights greater than β
Normalize the weights of the samples
Else
closeness = ∞
keep the last set of samples
Else
Sample (α) (Sampling step with parameter α)
If no sample found
closeness = ∞
keep the last set of samples
Normalize weights
Resample the sample set (Re-sampling step)
Send locations and closeness to first- and second-hop neighbors.
```

图 9-12 每个节点中的算法（1） 283

每个节点使用它相邻节点的位置估计加权其自身的采样。只使用有高精度位置估计的相邻节点可以降低通信开销。使用被称为接近度的参数可测量这些估计的质量。有 N 个采样的节点 p 的接近度的计算公式为：

$$\text{closeness}_p = \frac{\sum_{i=1}^{N} w_i \sqrt{(x_i - x)^2 + (y_i - y)^2}}{N} \tag{9.39}$$

其中，N 是节点 p 的采样个数，(x_i, y_i) 表示第 i 个采样的坐标（$i = 1, \cdots, N$），w_i 表示

第 i 个采样的权重，(x, y) 是节点 p 当前位置的估计。

种子的接近度总是为0，并且节点的接近度总是大于0。接近度低表示位置估计精度高，这是评价节点位置估计精度的好方法。

在算法（1）的开始，种子的接近度是0，节点的接近度是∞。因此，在第一个时间段内，只有种子能够为一跳和两跳的相邻节点提供信息。随着处理的进行，节点更新它们的估计和接近度，并把这些信息发送给它们的相邻节点。

一个节点移动到没有相邻节点的新位置后将不再接收新的位置信息。如果出现这种情况，节点使用其前一采样集合估计位置。这个节点必须重新本地化，当前采样集合也必须重新初始化。

该算法有很高的通信成本，因为每个节点都使用它所有第一跳和第二跳相邻节点的信息。接下来讨论算法（2）。

9.5.3　算法（2）

由于需要在节点间进行采样的传输，因此算法（2）的通信是非常频繁的。在算法（2）中，为每个节点都分配了一个权重，它只使用那些给它的采样提供权重的相邻节点（相对于采样和相邻节点）的权重。在计算出这些权重之后，MSL 计算一个位置估计和一个接近度值。每个节点向它的相邻节点广播其位置估计和接近度。这种方式可以显著地减少通信开销，因为它不需要传输采样值。

在该算法中，分配给节点的权重依赖于其位置估计的质量。为了达到这一目的，节点的权重可以定义成其接近度的函数：

284

$$w_i = b^{-\text{proximity}_i} \tag{9.40}$$

算法（2）的性能对 b 的取值不敏感，此处令 $b=7$。与算法（1）相似，算法（2）中的节点只使用那些接近度值低的相邻节点的位置。

在算法（2）中，采用如算法（1）的方式计算相邻种子节点的权重，但第一跳的相邻非种子节点的权重按如下公式计算：

$$w'_i(q) = \begin{cases} w_i & d(s, q) \leqslant r + v_{\max} + v_{\text{corr}} \\ 0 & \text{其他} \end{cases} \tag{9.41}$$

因为节点使用的信息比算法（1）少（使用一个相邻节点的位置估计而不是一组加权采样），所以需要考虑额外的不确定性。使用参数 v_{corr} 可以达到这个目的。算法（2）对于 $v_{\text{corr}} \in [0.2r, 0.5r]$ 中的取值不敏感，此处 $v_{\text{corr}} = 0.3r$。

第二跳的相邻非种子节点的权重按如下公式计算：

$$w'_i(q) = \begin{cases} w_i & r - v_{\max} - v_{\text{corr}} \leqslant d(s, q) \leqslant d(s, q) \\ & \leqslant 2r + v_{\max} + v_{\text{corr}} \\ 0 & \text{其他} \end{cases} \tag{9.42}$$

该算法使用蒙特卡洛方法，与蒙特卡洛定位（Monte Carlo Localization，MCL）相似[LHu04a]，但这里通过几种方式改进了蒙特卡洛定位并使其一般化。通过修改采样过程以使该方法可以在静态网络中工作并比蒙特卡洛法优越，即使只使用相邻种子节点的信息。

第一，在算法（1）和算法（2）中，节点使用的信息来自那些比它们的估计更精确的相邻节点，从而能够改进低速节点或低种子节点密度的网络的性能。

第二，通过修改采样过程和允许采样的权重大于阈值 β，能够加快定位算法的收敛速度。在移动网络中，这会使执行时间更快、位置估计更准确。

根据上述介绍可知，在算法（2）中，传感器节点在移动时必须预设最小速度，它不能在低于最小速度的状态下工作或在静态网络中工作。算法（1）可以进行高精度的估计，不论传感器节点是处于静态网络还是处于高速或低速移动中。

9.6 无 GPS 环境中的移动无线传感器网络的节点定位方法

设想在发生火灾的建筑物内执行搜索任务，有一组移动节点在一个楼层内进行探测。目标 285
是定位火源。在一个设备群中，这些节点协同移动。这个设备群沿一条能够覆盖这个区域的路径进行移动，移动过程中会进行温度测量。无 GPS 环境的移动无线传感器网络的节点定位管理中，有很多问题需要解决。最重要的是，位置估计的叠加误差可能会累积到很高，这是由移动过程中方向和距离评估中的机械误差造成的，它存在于所有测量系统中。这类误差的来源是环境的改变或制造中的缺陷。这意味着随着移动的进行，节点位置和方向的不确定性会不断增加。

Akcan 等［Akcan06］提出了一种针对无 GPS 的具有移动节点的传感器网络中方向定位问题的解决方案。针对每个节点的本地坐标系统，它提出了一种基于运动的计算节点位置和方向的算法。该算法非常快并且无需额外内存。另外，它不受位置累积误差的影响。更具体地说，该算法不受节点速度的影响。

无 GPS 环境中的节点定位算法假设传感器能够使用一种知名的距离测量方法（如 TOA）测量它们到相邻节点的距离。它也需要运动执行器，这能使每个节点沿着特定的方向（比如向北）移动指定距离。

首先，核心定位算法使用两个相邻节点 n_1 和 n_2 描述，这两个节点产生两个可能相关的位置。随后将讨论一个验证算法，该算法使用一个共同的第三个相邻节点选择正确的解决方案。

核心定位算法：核心定位算法按轮的方式工作，并且每轮定位操作都由三个步骤构成：

1）测量相邻节点间的距离。

2）继续单个节点的运动。

3）交换（相邻节点）方向和距离值。

只要节点需要定位，就启动一轮定位算法的工作。不要求每一轮之间有任何的连续性和模式。同时，也没有假设这些轮持续的时间。但是，需要假设节点在一轮工作期内不改变它们的方向。

图 9-13 显示了两个节点 n_1 和 n_2 在一轮内的典型运动。在时刻 t_1，n_1 的位置是 (x_0, y_0)，n_2 的位置是 (x_2, y_2)，并且节点测量两者之间的初始距离 d_1。在时刻 t_1 和 t_2 之间，每个节点 $\{n_i \mid i=1, 2\}$ 移动方向是 α_i 且路过的距离是 v_i。在时刻 t_2，两个节点的位置分别是 (x_1, y_1) 和 (x_3, y_3)，计算它们之间的距离 d_2 并交换 v_i 和 α_i。只有在接收到所有信息后，每个节点再以其自身为原点并在它的本地坐标系中计算其他节点的位置和方向。选择 n_1 的位置 (x_0, y_0) 为原点，并在 n_1 的本地坐标系中求解方程： 286

$$
\begin{aligned}
&x_1 = v_1\cos\alpha_1 && y_1 = v_1\sin\alpha_1 && \text{(i)}\\
&x_3 = x_2 + v_2\cos\alpha_2 && y_3 = y_2 + v_2\sin\alpha_2 && \text{(ii)}\\
&(x_3 - x_1)^2 + (y_3 - y_1)^2 = d_2^2 && x_2^2 + y_2^2 = d_1^2 && \text{(iii)}
\end{aligned}
\qquad (9.43)
$$

将方程 i 和 ii 代入 iii，得到

$$x_2 A + y_2 B = C \qquad (9.44)$$

并有定义：

$$A = v_2\cos\alpha_2 - v_1\cos\alpha_1,\quad B = v_2\sin\alpha_2 - v_1\sin\alpha_1$$

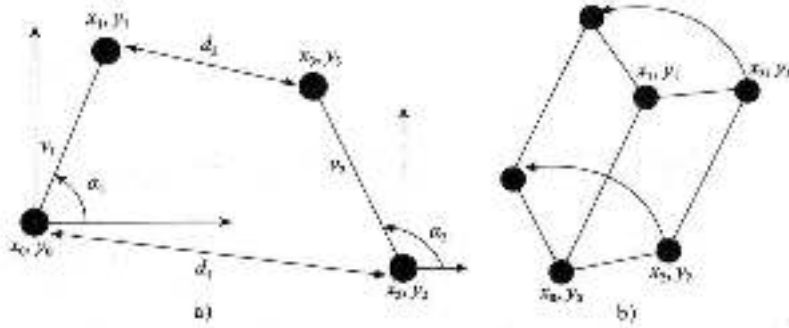


图 9-13 两个带角度和距离的节点的共基运动

$$C=\frac{1}{2}\left(d_2^2-d_1^2-v_1^2-v_2^2+2v_1v_2\cos\left(\alpha_1-\alpha_2\right)\right) \tag{9.45}$$

将

$$x_2=\frac{C-y_2B}{A}\text{和}\ y_2=\frac{C-x_2A}{B}\quad\text{代入}\ x_2^2+y_2^2=d_1^2 \tag{9.46}$$

得到

287

$$x_2^2D-2x_2E+F=0,\ y_2^2D-2y_2G+H=0 \tag{9.47}$$

再有公式：

$$D=A^2+B^2,\ E=AC,\ F=C^2-d_1^2B^2$$
$$G=BC,\ H=C^2-d_1^2A^2 \tag{9.48}$$

注意在两个方程中 x_2^2 和 y_2^2 的系数是相同的，即 D。

使用公式（9.48），每个变量都可以独立求解

$$x_2=\frac{E\pm\sqrt{E^2-DF}}{D},\ y_2=\frac{G\pm\sqrt{G^2-DH}}{D} \tag{9.49}$$

只要 $D\neq0$，都可以使用方程 9.49 成对求解。在实际中，可能会使用方程 9.49 计算 x_2 或 y_2 中的一个，并使用式 9.48 推导出其他变量。当 $A=0$ 但 $B\neq0$ 时，可以使用式（9.46）计算 x_2；当 $A\neq0$ 但 $B=0$ 时，可以使用式 9.44 计算 y_2。

图 9-14 是计算从 n_2 到 n_1 的位置的核心定位算法。每个节点通过解方程寻找其所有相邻节点可能的位置。计算出可能的位置后，节点必须使用一个额外的共同邻居节点（n_3）来完成验证步骤，这是因为对于计算得出的同一个相邻节点的所有可能位置中只有一个是真实的。

```
CoreLocalization (n1, n2, v2, α2)
2: d2 ← inter-distance(n1, n2)
2: Move node n2 by v2 and α2
3: d1 ← inter-distance(n1, n2)
4: Retrieve v2 and α2 from n2
5: Calculate positions of n2 using equations (4), (5) and (6)
Verification (NeighborList NL)
1: for each neighbor pair (m, n) in NL do
2:       if m and n are neighbors then
3:         dm,n ← measured inter-distance(m, n)
4:         for each position pair (m^i, n^j)(i, j = 1, 2) do
5:         Compute Euclidean distance D between m^i and n^j
6:         if D ≈ dm,n then
7:           mark m^i and n^j as exact positions
```

图 9-14 核心定位算法

如果整个无线传感器网络为了完成一个目标而向着特定方向移动，那么方向定位算法就非常有用了。为了反映实际的移动特性，文献［XHong99］使用了**参考点群组移动**（Reference Point Group Mobility，RPGM）模型。文献［TCamp02］研究了许多其他的移动模型。但是使用RPGM模型是因为它具有通用性。在RPGM模型中，单个传感器节点的运动的建模与整个组的随机选择的移动方向相关。每个传感器绕此固定参考点随机移动，并且整个组沿着组的逻辑中心移动。定位算法计算每个节点的位置/方向。在RPGM模型中，传感器不需要知道群组的中心。

传感器节点的一跳相邻节点决定了组中这个传感器节点随机运动的环境。因此，可以去掉参考点而使用相邻节点表示运动的参考点。

图9-15显示了改进过的移动算法。网络根据一个方向向量进行移动。该算法要求每个节点的相邻节点数不小于k，每个节点都尽量维护一个半刚性的群组编队并与网络保持联通。这是一种效果最佳的k连接算法，算法中一个传感器试图保持与相邻节点的距离小于它的RF通信距离。它根据相邻节点的数量动态调整这个距离，这样既可使这些相邻节点随网络移动，也能保持它们在k跳的范围内。这种方法可以避免网络分区。距离函数返回给定节点的无线距离。

显然，只要知道整个组的初始运动方向，就可以保持整个网络的连通性。这很像仿生计算中群集的概念。例如，一个寻找石油的无线传感器网络可能按Z字形运动，目标是寻找溢油点并在发现之后覆盖污染区域。在本例中，可以指定一个虚拟边界，并且当覆盖区域时，网络中的传感器能保持足够近的距离以便于通信。

移动算法只需要本地位置信息。这非常适合规模很大的网络（>1000个传感器节点），因为它要求传感器节点与其直接相邻节点通信而无需进行信息洪泛。

```
MoveNode(Node N, NeighborList NL, DirectionVector Ū, INT k, RangeFactor
 RF)
1: V̄ ← Ū
2: count ← 0
3: for each localized neighbor n in NL do
4: ūN,n is the vector from N to n
5: V̄ ← V̄ + ūN,n
6: count ← count + 1
7: if count < k then
8: RF ← RF /2
9: V̄ ← (RF * range (N) * V̄ + Ū)/(count + 1)
10: Move node N by V̄
```

图9-15 k个相邻节点的移动算法

9.7 高精度低功耗的无线传感器网络定位系统

传统的定位算法有两个局限。第一，受自身物理特性和能量方面的限制，传感器节点的有效距离是有限的。例如，Cricket系统中的超声波传感器的有效距离只有几米［NPriyantha05］。第二，为这些传感器装备进行一次性定位的特殊电路的成本非常昂贵，因为大多数节点是静止不动的。

为了克服这些限制，研究者提出了许多无距离限制的定位方案。大部分算法通过相邻节点间的无线连通信息估计节点的位置。虽然这些方法不需要定制高成本的硬件，但是它们的精度无法令人满意。

Stoleru等实现了传感器定位的高精度方法，且不需要高成本（从通信和计算的复杂度来

说）[Rahul05]。它使用称为 spotlight 的概念。传感器节点不需要为定位增加新的硬件。所有复杂、昂贵的硬件和计算都在一个 Spotlight 设备中进行，该设备可以发射一束可操纵的激光束照亮放置在已知地形的传感器节点。

该方法的定位精度（即小于 1m）高于基于距离的定位方案。它比基于超声/声波的方案有更长的有效范围（大于 1000m）。由于所有复杂的硬件/软件都在一个复杂设备中，因此成本比给每个传感器添加额外硬件要低。

Spotlight 是一种典型的与距离无关的定位方案，它可以在户外环境中工作。单个设备和传感器节点间需要可视。

Spotlight 系统的主要思路是在传感器节点部署的区域生成控制事件。利用传感器节点感知到事件生成的时间和该生成事件的时空属性，可以推导出传感器节点的空间属性（即位置）。



图 9-16 使用 Spotlight 系统的传感器网络定位

如图 9-16 所示，一个传感器网络部署和定位的场景如下：使用无人机随机部署无线传感器节点。完成部署后，在节点间执行时间同步协议。飞行器（如直升机）装备 Spotlight 设备，飞过网络区域并生成光照事件。每个传感器检测事件，当检测到事件，它们的 Spotlight 设备返回带时间戳的报告。Spotlight 设备计算传感器节点的位置。

这样一个 Spotlight 系统在使用前要满足下面的条件：

1）传感器节点能够与 Spotlight 设备进行通信。

2）飞行器知道自身的位置和方向，也拥有传感器区域的地图。

3）Spotlight 设备能够产生人的空间事件，该事件可被传感器检测到（即使存在背景噪声（日光））。

4）在 Spotlight 设备和传感器节点间彼此可视。

Spotlight 定位系统使用如下定义：

假设空间 $A \subset R^3$ 包含所有的传感器节点 N，并且每个节点 N_i 位于 $p_i = (x_i, y_i, z_i)$。Spotlight 定位系统需要支持三个主要函数来获取 $p_i = (x_i, y_i, z_i)$，分别为事件分布函数（Event Distribution Function，EDF）$E(t)$、事件检测函数 $D(e)$ 和定位函数 $L(T_i)$。下面是它们的正式定义：

定义 9.1：事件 $e(t, p)$ 是可检测的现象，它发生在时刻 t 和点 $p \in A$。事件的示例如光、热、烟和声音等。令 $T_i = \{t_{i1}, t_{i2}, \cdots, t_{in}\}$ 是节点 i 检测到的事件的 n 个时间戳的集合。令 $T' = \{t'_1, t'_2, \cdots, t'_m\}$ 是传感器区域生成的事件的 m 个时间戳的集合。

定义 9.2：事件检测函数 $D(e)$ 定义了一个二态检测算法。给定事件 e

$$D(e) = \begin{cases} \text{true}, & \text{检测到事件} \\ \text{false}, & \text{未检测到事件} \end{cases} \tag{9.50}$$

定义 9.3：事件分布函数 $E(t)$ 定义了 A 中的事件在时刻 t 的分布点：

$$E(t) = \{p \mid p \in A \wedge D(e(t, p)) = \text{true}\} \tag{9.51}$$

定义 9.4：定位函数 $L(T_i)$ 定义了一个输入为 T_i 的定位算法，T_i 是节点 i 检测到的事件的

290 - 291

一系列时间戳：

$$L(T_i) = \bigcap_{\forall t} E(t) \tag{9.52}$$

如图9-17所示，传感器节点支持事件检测函数 $D(e)$，这个函数检测是否有外部事件发生。该检测算法的实现方式是使用简单的基于阈值的检测算法，或者是高级的数字信号处理（Digital Signal Processing，DSP）技术。

Spotlight设备实现了事件分布函数 $E(t)$ 和定位函数 $L(T_i)$。定位函数是一个聚合算法，它可以计算多个点集的交集。

事件分布函数 $E(t)$ 也描述了事件随时间的分布。这是Spotlight系统的核心，比其他两个函数复杂。$E(t)$ 是由Spotlight设备实现的（而不是在传感器节点中实现的）。

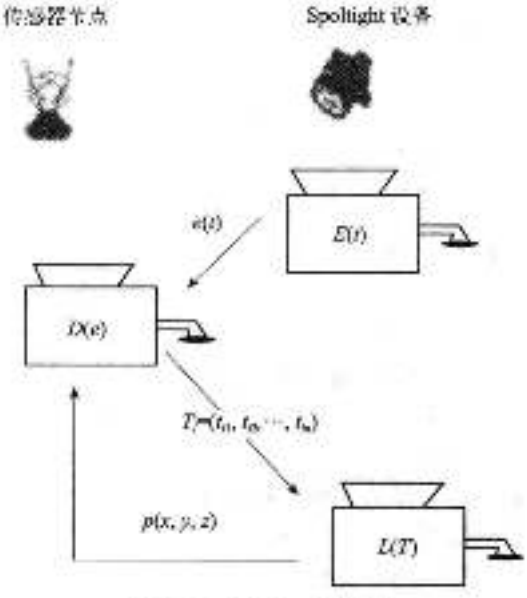


图9-17 Spotlight系统结构

基于这三个函数，定位过程如下：

1）**事件分布**：在一段时间内，Spotlight设备在空间 A 中分布事件。

2）**事件检测**：在事件分布过程中，传感器节点记录检测到事件的时间序列 $T_i = \{t_{i1}, t_{i2}, \cdots, t_{in}\}$。

3）**事件报告**：在事件分布之后，每个传感器节点将检测时间序列发送到Spotlight设备。

4）**位置估计**：Spotlight设备使用事件序列 T_i 和已知的 $E(t)$ 函数估计传感器节点 i 的位置。

在Spotlight系统中，核心的技术是事件分布函数 $E(t)$。为简单起见，假设节点集沿着一条直线（$A = [0, 1] \subset R$）放置。Spotlight设备沿着这条线以恒定速度 s 生成点事件（如亮点）。

节点 i 检测到的事件的时间戳集是 $T_i = \{t_{i1}\}$。事件分布函数 $E(t)$ 是

$$E(t) = \{p \mid p \in A \wedge p = t * s\} \tag{9.53}$$

其中，$t \in [0, 1/s]$。定位函数的结果是：

$$L(T_i) = E(t_{i1}) = \{t_{i1} * s\} \tag{9.54}$$

对于位置在 p_i 的节点 i，$D(e(t_{i1}, p_i)) = \text{true}$。事件分布函数 $E(t)$ 的实现是前向的。如图9-18a所示，当光源以一定角速度 $Sa = d\alpha/dt = (\sin'(\alpha))/d$ 发出一束光后，会有一个恒定速度为 s 的亮点事件沿着直线在距离 d 处产生。

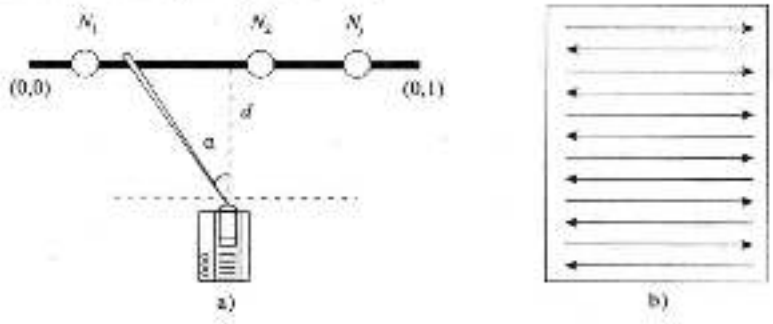


图9-18 点扫描FDF的实现

除了这种简单的单线情况，点扫描的事件分布函数也可扩展到节点放置在二维平面 R^2 的情况。在这种情况下，Spotlight 系统逐步扫描平面，激活传感器节点。该场景如图 9-18b 所示。

- **线扫描事件分布函数**

有些设备（如二极管激光器）能够同时产生一系列事件。它们可以很容易地支持线扫描事件分布函数。假设传感器节点放置在一个二维平面上（$A = [l, l] \subset R^2$），并且扫描速度是 s。节点 i 检测事件的时间集是 $T_i = [t_{i1}, t_{i2}]$。线扫描事件分布函数的定义如下：

$$E_s(t) = \{p_k \mid k \in [0, 1] \wedge p_k = (t \times s, k)\} \tag{9.55}$$

对 $t \in [0, l/s]$，有 $E_s(t) = \{p_k \mid k \in [0, 1] \wedge p_k = (k, t \times s - l)\}$。

对 $t \in \left[\frac{l}{s}, \frac{2l}{s}\right]$，$E_s(t) = E_s(t) \cup E_s(t)$。

显然，可以使用两个事件线的交集定位一个传感器节点，如果如图 9-19 所示。更正式的方程如下：

$$L(T_i) = E(t_{i1}) \cap E(t_{i2}) \tag{9.56}$$

其中，对位置在 p_i 的节点 i，$D(e(t_{i1}, p_i)) = \text{true}$，$D(e(t_{i2}, p_i)) = \text{true}$。

- **区域覆盖事件分布函数**

除了线覆盖以外，也可以进行区域覆盖。其他设备（例如投射灯）可以产生覆盖一块区域的事件。它们可以实现**区域覆盖事件分布函数**。区域覆盖事件分布函数将空间 A 划分为多个部分，并给每个部分分配一个唯一的二态标识符，称为代码。假设在平面（$A \subset R^2$）上进行定位，A 的每个部分 S_k 有唯一的代码 k。区域覆盖事件分布函数的定义如下：

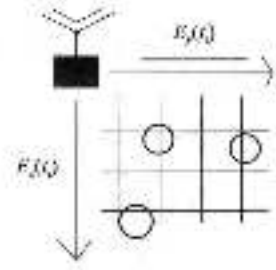


图 9-19　线扫描事件分布函数的实现

$$\text{BIT}(k, j) = \begin{cases} \text{true}, \text{如果 } k \text{ 的第 } j \text{ 位为 } 1 \\ \text{false}, \text{如果 } k \text{ 的第 } j \text{ 位为 } 0 \end{cases}$$

$$E(t) = \{p \mid p \in S_k \wedge \text{BIT}(k, t) = \text{true}\} \tag{9.57}$$

294

对应的定位算法是：

$$L(T_i) = \{p \mid p = \text{COG}(S_k) \wedge (\text{BIT}(k, t) = \text{true}, t \in T_i) \wedge (\text{BIT}(k, t) = \text{false}, t \in T - T_i)\} \tag{9.58}$$

其中，$\text{COG}(S_k)$ 表示 S_k 的重力中心。

如图 9-20 所示，平面 A 分为 16 个部分。每个部分 S_k 有唯一的代码 k。Spotlight 设备根据如下这些代码分布事件：如果 k 的第 j 个位是 1，则区域 S_k 在时刻 j 被一个事件覆盖（用光线照亮）。节点放置在区域 S_k 中的任何位置都会被定位到这个区域的重力中心。例如，在区域 1010 中的节点在时刻 $T = \{1, 3\}$ 中检测到事件。在时刻 $t = 4$，区域中的每个节点都被定位。

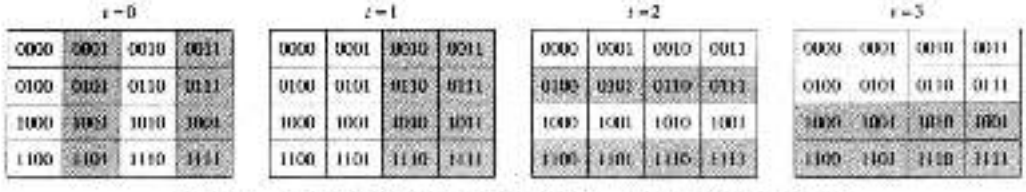


图 9-20　区域覆盖事件分布函数的步骤。事件覆盖了阴影区域

9.8 LOCALE：稀疏移动传感器网络的协同定位估计

Zhang 和 Martonosi 提出了稀疏网络中的节点自组协同定位估计算法（Low-Density Collaborative Ad hoc Localization Estimation，LOCALE）[Zhang08]。该算法具有如下特性：

第一，它是分布式定位算法，不需要主控节点。

第二，它是建立在协同定位方法上的，即几个传感器节点共同寻找一个特定的位置。

第三，它在稀疏移动传感器网络中使用效果最好。它在高密度场景中可能工作表现不佳，但可在移动的场景中使用。

LOCALE 能够主动预测和保持位置估计，即使网络连接中断。它使用航位推算（Dead Reckoning，DR）系统实现这一目标。当一个传感器节点遇到一个相邻节点，它们相互交换位置估计值，然后使用两个位置估计值的加权线性组合来修正节点的位置。最终的效果是它能够均 295
匀平滑传感器节点的运动并给出每个传感器节点的位置分布。传感器节点使用这个分布能够得到较好的位置预测。此外，也可以得到预测精度的置信估计。

LOCALE 具有如下主要特性：

- 传感器节点与它遇到的相邻节点交换信息之后，会重新调整其自身的位置估计。
- 它的定位精度远高于常用的信标跟踪方法。
- 当传感器节点没有精确的估计时能够快速地进行误差纠正。
- 适用于稀疏异构的无线传感器网络。
- 低功耗设计。

9.8.1 协同位置估计

LOCALE 是一种容迟的协同定位策略，它适用于稀疏移动的传感器网络。如图 9-21 所示，它包括三个主要阶段保持和修正定位估计。第一个阶段称为局部定位阶段，它利用传感器节点的运动跟踪信息维护粗粒度的位置估计。这个阶段允许传感器节点在长时间断网的情况下维护位置信息，即使位置信息不够精确。在转换阶段，传感器节点使用其相邻节点的位置估计进行自身位置估计。在更新阶段，传感器节点合并来自相邻节点的估计和已有的估计。该阶段修正传感器节点的位置估计。

下面看一下 LOCALE 如何表示节点的位置。

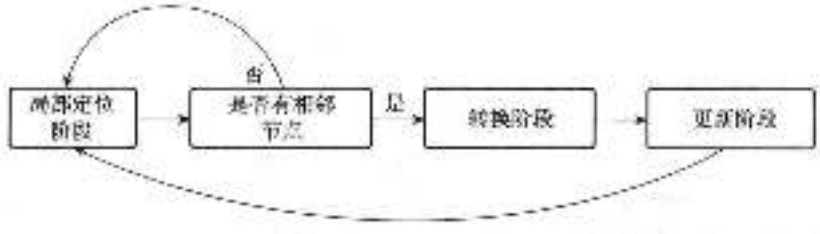


图 9-21 LOCALE 概述：定位误差在局部定位阶段增加，在更新阶段随协同而减小 296

9.8.2 LOCALE 中的定位

如果 LOCALE 需要从多个估计中预测和融合位置信息，它需要绝对位置估计和估计确定性（即置信度），并且节点的位置实际是相邻节点的平滑均匀位置估计。

可以用正态分布描述位置估计（均值）和确定性（方差）。虽然单个节点的位置估计可能

不是正态分布，基于中心极限定理，平均的估计应该是正态分布的。使用如下概率密度函数：

$$p(X) = \frac{1}{2\pi\sqrt{|C|}} * e^{-\frac{1}{2}(X-\bar{x})C^{-1}(X-\bar{x})} \tag{9.59}$$

这个方程表示节点（X）的真实位置相对于估计位置（$\bar{x}$）的概率。为了定义方程，只需要估计位置（$\bar{x}$）和协方差矩阵 C。这里只关注二维情况；然后，使用高度信息可以很容易地推广到三维情况。

$$C = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix} \quad X = \begin{pmatrix} x \\ y \end{pmatrix} \tag{9.60}$$

在这个矩阵中，对角线上的两个值分别是两个坐标系轴上坐标变化的方差。而其他值是两个轴之间的协方差。当节点运动并遇到相邻节点时，这些值将被更新。

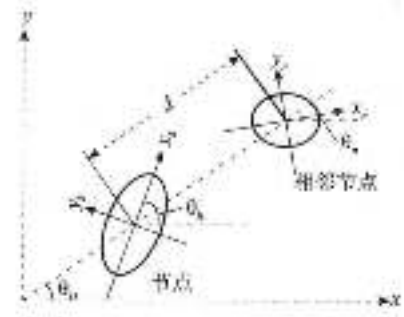


图 9-22　不同方向的两个相邻节点的表示

LOCALE 保存三个变量：位置估计 X、协方差矩阵 C 和局部坐标与全局坐标之间的角度 θ。图 9-22 说明了相邻节点对应的角 θ_n，其中每个节点都有自身的局部坐标（x_n，y_n）（x_m，y_m），并且相对于全局坐标（x，y）的角度是（θ_n，θ_m）。

9.8.3　局部定位阶段

在本阶段中，每个节点基于运动跟踪方法维护局部位置估计。LOCALE 采用低成本、低精度的航位推算传感器来跟踪每个节点相对于各自前次测量位置的运动情况。

由此可以通过合并相对测量分布和已有的估计分布得到一个新的分布：

$$N = N_{old}(X_1, C_1) + N_{odom}(X_2, C_2) \tag{9.61}$$

整合以后得到带均值和方差的新分布：

$$N_{combined} = N(X_2 + X_2, C_1 + C_2) \tag{9.62}$$

运动协方差矩阵是针对运动方向的。局部坐标 C_1 使用协方差矩阵表示为：

$$C_1 = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \tag{9.63}$$

局部协方差矩阵通过下式旋转到全局坐标：

$$C = R(-\theta)^T C_1 R(-\theta) \tag{9.64}$$

其中 θ 是节点移动的方向，并且旋转矩阵为

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin\theta \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \tag{9.65}$$

297
-
298

值得注意的是，新的估计位置分布的均值和协方差矩阵就是对两者进行简单求和。通过合并相应的运动跟踪信息，传感器节点能够把运动信息和协方差信息进行合并。

9.8.4　转换阶段

虽然可以使用相邻节点的位置信息来修正节点位置，但不能只使用相邻节点的位置估计值，也不能用它与自己的位置进行融合，这是因为任何两个节点间都存在距离。因此，LOCALE 将相邻节点的估计转换成一个适合合并的格式，并且这个格式应该是从主节点位置上观测得到的。转换信息需要两个节点间“相对”位置的信息。图 9-23 说明了这种转换的原理。

对于“相对”位置信息，LOCALE 允许使用多种格式，例如距离的相对测量和相邻节点的方向，或可以说明相邻节点在通信距离某处的简单信息，从中并不难得到距离和方向信息[BKusy07]。

在图 9-23 中可以看到，将相邻节点观测信息融合到主节点的局部帧中的过程。它主要包含以下几个步骤：

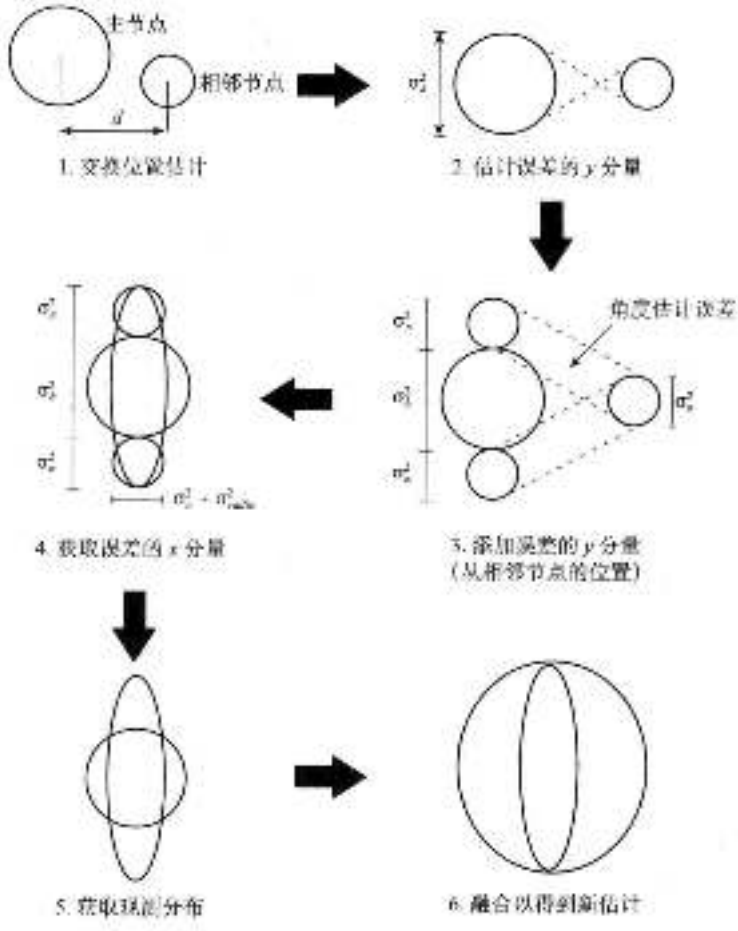


图 9-23　无方向测量的相对位置估计

在步骤 1 中，遵照相对坐标旋转观测结果，使两个观测结果的 x 轴向一致：

$$C_{i_x} = R(\theta_{ij} - \theta_{i_x})^T C_{ij} R(\theta_{ij} - \theta_{i_x})$$
$$C_{i_y} = R(\theta_{ij} - \theta_{i_y})^T C_{ij} R(\theta_{ij} - \theta_{i_y}) \tag{9.66}$$

在步骤 2 中，由于主节点位置的不确定性导致的角度不确定性，可以计算转换的协方差矩阵的 y 分量。

在步骤 3 中，使用协方差矩阵的 x 分量确定变换的观测分布。注意，该矩阵应该考虑距离的可变性，它是主节点方差的 x 分量的总和，并且 $Range^2(1-2\sqrt{2}/3)$。

观测误差的协方差应当是 0，因为所有节点都是基于相对坐标的。将要被合并的观测的平均值都减去距离 d，并且两个相邻节点间的期望距离向量是 $Range/\sqrt{2}$，方向是 $\theta_{i_y} - \theta_{i_x}$。

$$C_{1/\text{Observed}} = \begin{pmatrix} \sigma_{\text{radio}}^2 + \sigma_x^2 & 0 \\ 0 & \sigma_\lambda^2 + 2\sigma_x^2 \end{pmatrix}$$

$$X_{\text{Observed}} = \begin{pmatrix} x_0 + d \times \cos(\theta_0 - \theta_x) \\ y_0 + d \times \sin(\theta_0 - \theta_x) \end{pmatrix} \tag{9.67}$$

为了完成最后的融合，变换的观测和主节点的分布需要旋转到全局坐标系：

$$C_{1/\text{Observed}} = R(-\theta_0)^T C_{1/\text{Observed}} R(-\theta_0) \tag{9.68}$$

299 ~ 300

正如看到的，变换阶段的一个比较有利的事实是系统可以把相邻节点的观测投影到自身观测上，从而在下一阶段获得更为精确的融合结果。这里只需要概率性测量，并不需要特殊的无线配置或硬件。

9.8.5 更新阶段

在前面两个阶段中，可以看到 LOCALE 通过融合相邻节点的观测改进定位精度。使用这种方式能够使测量误差平均化。另一方面，如果传感器节点有不同的运动模式，定位估计会有不同的确定性，可以使用方差表示这些确定性。最后，融合这些由相应的方差加权的估计，位置分布被融合成一个加权线性组合。

在更新阶段，系统进行自我估计的准备和最后的融合过程，如图 9-24 所示。因为观测者的数量会增加，所以应由这些分布合并成结果计算为调和平均数。

位置估计结果上的融合因子表示每个分布的权重。融合因子定义如下：

$$K = C_0 * [C_0 + C_{1/\text{Observed}}]^{-1} \tag{9.69}$$

按如下方式使用融合因子计算新的协方差矩阵和新的位置估计：

$$C_{\text{merged}} = C_0 - KC_0$$

$$\hat{X}_{\text{merged}} = \hat{X}_0 + K(\hat{X}_{\text{observed}} - \hat{X}_0) \tag{9.70}$$

协方差矩阵的新角度是：

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2b}{a-d}\right), \quad C = \begin{pmatrix} a & b \\ b & d \end{pmatrix} \tag{9.71}$$

最后，将融合的分布旋转回局部坐标。

$$C_{\text{new}} = R(-\theta_{\text{merged}})^T C_{\text{merged}} R(-\theta_0) \tag{9.72}$$

融合的位置和协方差矩阵在每轮计算中作为新的自身位置估计保存在内存中。这个融合算法就是线性组合。只要有相邻节点到来，这个过程就可以不断重复下去。 301

在更新阶段，连同 LOCALE 的其他部分，便可在在稀疏传感器网络中实现容迟协同定位。

9.9 无线传感器网络定位的安全

对于定位有四个重要的评价指标：能量、效率、精度和安全。无线传感器节点应用在严酷的环境中，如地雷探测、战场监测和目标跟踪等军事装备所遭遇的恶劣环境，甚至是更加恶劣的环境。在这些独特的环境中，无线传感器网络节点应自主工作并处理所面临的挑战。敌人可以在物理上捕捉或妥协攻击（使传感器节点同时为我方和敌方工作）一个或更多个传感器节点，这依赖于攻击者的攻击方式。通过写入恶意代码，攻击者能够操纵传感器节点的工作、抽取加密信息或完全摧毁它。如果从节点中抽取到敏感信息，安全屏障（如身份验证）就能够被绕开，攻击者就能够在系统内部发动攻击，这会导致大多数系统失效。

为了理解这一点，设想有一个基于信标的定位模型，并且传感器节点不能确定自身位置，

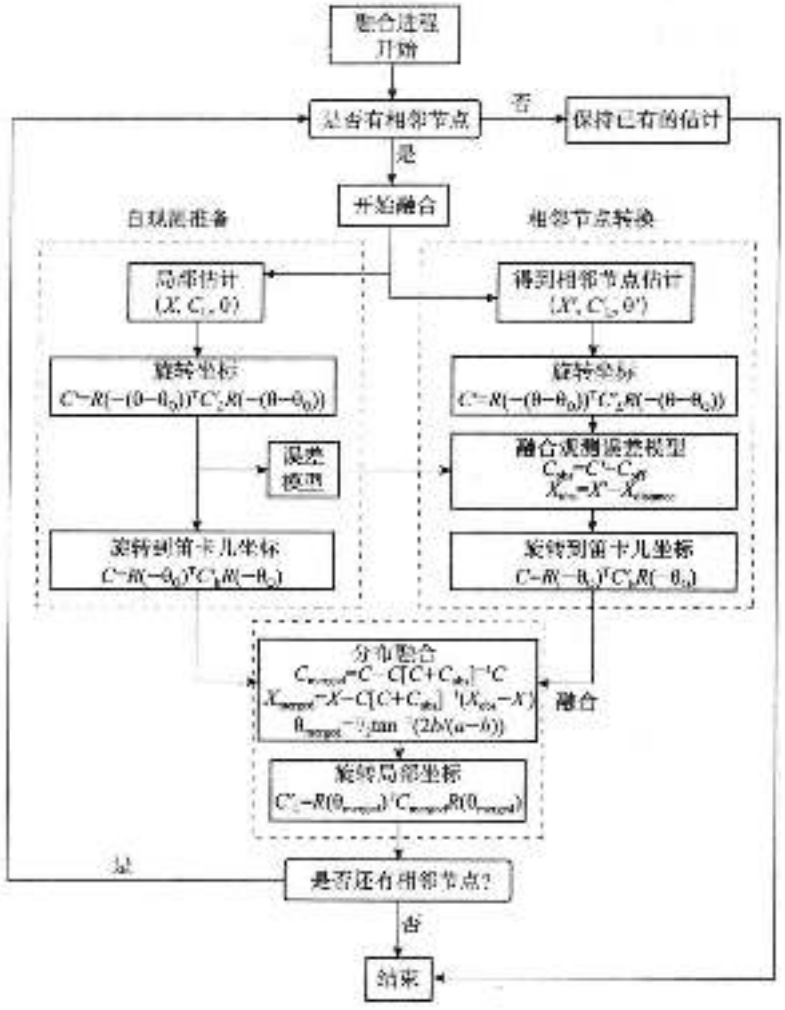


图 9-24 当遇到相邻节点时的融合过程框图

它们也无法知道信标节点传输的位置信息是否可信和精确。恶意的节点可能传输伪造的数据，导致接收信息的节点估算出错误的位置数据。这称为信息不对称。在这种情况下，一个实体比其他实体拥有更多的信息。文献［ASrinivasan06］中讨论了基于信标定位中的这种模型，并且提出了一种解决内部攻击的有效方式。攻击者也可以发动 sybil、虫洞攻击或者使用重复攻击扰乱定位进程。

在本节中，我们将回顾已有的定位安全技术，并讨论它们的优点和弱点。

9.9.1 SeRLoc

在文献［LLazos04］中，Lazos 和 Poovendran 提出了一种在不可靠无线传感器网络环境中进行节点定位的算法，称为 SeRLoc。它是一种与距离无关的分布式、节约资源的技术，它不需要针对位置发现而进行节点间通信。该方法对虫洞攻击、sybil 攻击和节点妥协攻击有很强的抵抗能力。在该方法中，有两个节点集合：传感器节点集合 N 中的节点装备了全向天线，而定位节点集合 L 中的节点拥有定向天线。传感器节点能够使用定位节点发送的位置信息确定它们自

身的位置。每个定位节点在天线扇区发送不同的信标。

在 SeRLoc 中，攻击者必须冒充众多的节点对定位进程进行妥协攻击。同时敌方没有动机冒充传感器节点，因为这些节点不需要其他节点的协助就可以计算出其自身的位置。在该模型中，存在两种对抗虫洞攻击的技术：区域独特属性（sector uniqueness property）和通信范围违背属性（communication range violation property）。

302 ~ 303

为了提高定位精度，可以部署更多的定位节点，也可以使用更多的定向天线。整个过程顺利进行的前提是没有无线媒介干扰。这是对真实世界设定的非常强的假设。

9.9.2 信标套件

在文献［DLiu05］中，Liu、Ning 和 Du 提出了一套检测恶意信标节点的技术，这些恶意节点向在重要应用中负责位置服务的传感器节点提供不正确的信息。该技术包括恶意信号检测、重放信号检测、恶意节点识别、规避错误检测以及撤销恶意信标节点。信标节点主要有两个作用：为传感器节点提供位置信息和检测其他信标节点的信标信号。信标节点不必牢记信标信号。它可以请求位置信息。执行检测的节点称为检测节点，被监听的节点称为目标节点。该文献建议检测节点在向目标节点请求位置信息时应使用非信标 ID，因为这样可以观测到目标节点的真实行为。撤销方案的工作方式以每个信标节点维护的两个计数器为基础，这两个计数器分别是警报和报告计数器。警报计数器负责记录对应信标节点的可疑性，报告计数器记录该节点报告的警报数。

当检测节点确定一个目标节点行为可疑时，报告将被发送到基站。被接受的警报计数器的报告来自于那些报告计数器低于某个阈值且其监控的节点是还未被撤销的检测节点。满足这些条件，各个节点的警报和报告计数器将加 1。这两个计数器是以离散方式工作的，而撤销机制是集中式的。文献［ASrinivasan06］使用了连续的和一个基于信任的机制来增强它们的健壮性。

9.9.3 攻击容忍的节点定位

在文献［DLiu05a］中，Liu、Ning 和 Du 提出了两种基于矩阵的鲁棒方法，它们能够容忍传感器网络中对于基于信标位置发现的恶意攻击。第一种方法是攻击容忍的最小均方估值，这可以过滤掉恶意的信标信号。通过检查在不同信标信号的位置参考之间的不相容性，可以去除有害数据以阻止攻击，不相容性可以使用估计的均方误差表示。第二种方法是，基于投票的位置估计量化部署区域为网格，并使每个位置参考对节点可能驻留的单元格进行“投票”。该方法使用一种迭代求精的选举方法容忍恶意信标信号。这两种方法都能够抵御恶意攻击已经，即使攻击已经绕开了认证过程。

304

9.9.4 稳健统计方法

［ZLi05］介绍了利用无线网络不同层次上的信息冗余特性采取容忍攻击而非消除攻击的思想。文中使用了两类定位方法：三角测量和基于射频的指纹识别。他们使用两种统计模型保障传感器网络的安全定位，两者都是基于过滤数据中的孤立点思想，这些数据来自用于定位估计的范围估计。

在三角测量模型中，使用了自适应最小二乘和最小中位数平方估计器。当发生攻击时，自适应估计器切换到最小二乘估计并在受攻击的情况下展示最小二乘的计算优势。在指纹模型中，欧式距离度量是不够安全的，因此他们提出了基于中位数的最近相邻节点方案抵御位置攻

击。作者也讨论了传感器网络中对定位的攻击。文献［ZL05］提出的统计方法建立在传感器的良性观测数量总是大于恶意观测数量的假设之上。

问题与练习

9.1 使用软件（工具）学习并验证讨论过的一种无线传感器网络定位算法的效率。

9.2 从算法复杂度、精度和实际的实现（分布式传感器）角度比较不同的定位方法。

9.3 为什么在无线传感器网络定位中需要考虑安全性？ 305

第10章

Wireless Sensor Networks: Principles and Practice

无线传感器网络中的时间同步技术

本章中，我们将讨论无线传感器网络中时钟同步机制的基本概念。我们的讨论基于对一篇综述文章［Sundararaman05］的概括。读者可以参考［Sundararaman05］来了解更多细节（例如不同无线传感器网络中同步机制的比较）。



不要认为在无线传感器节点中定义时间很容易。如何让一个传感器节点确认它所声称的时间是正确的（即与世界标准时间一致）？或许有人会说，可以让服务器给所有传感器节点广播一个标准时间。但是，无线传输延迟是不可忽略的。而且，当服务器生成一个消息（包含正确时间）准备传输时，它需要执行一系列的本地CPU指令实现一条消息。这种本地延迟也不能被忽略。最终，当一个传感器节点接收到来自服务器的消息声称"现在是上午9：00"时，这个传感器节点能够将自己的本地时间设置为"上午9：00"吗？

10.1 引言

时间同步是无线传感器网络的关键问题之一，因为所有的感知事件都要有准确的时间戳。
[307] 尤其在对象跟踪应用中，如果时间信息不准确，就不能确定对象的轨迹。因为我们通过链接在不同时间的对象位置形成轨迹。

在有线网络（如因特网）中，已经有成功的时钟同步协议，例如NTP（网络时间协议）。但是，由于某些原因，这些协议并不适用于无线传感器网络。

首先，这些有线网络同步协议由于无线干扰所带来的高差错率而无法在无线环境中很好地工作。

其次，一个无线传感器网络可以有成千上万的资源受限的传感器节点。在巨大的无线传感器网络中，这种同步协议要有高度的可扩展性。同时，它应该实现无中心控制的、自组织的、合作的同步。

再次，这些同步协议需要重点考虑能量保护问题。能量不能源源不断地供给每个传感器节点，同时由于它们体积较小限制了所能存储和收集的能量。

因此，由于本身所具备的不可靠的无线连接、高密度、非常有限的能量和存储空间等特性，无线传感器网络需要一种全新的时钟同步协议。

在设计一种时钟同步协议之前，我们首先需要理解计算机时钟的概念。它具有以下基本特点：

1）一个计算机时钟由一个电子设备生成，这种设备计算在某个频率下一个精确加工的石英晶体中的振荡数。

2）一个时钟可以由一个硬件和一个软件共同决定。这个硬件（石英晶体）和软件（时间控制程序）共同为操作系统和用户提供一个准确的、稳定的、可靠的时间函数。

3）一个计算机时钟本质上是一个计时器。这个计时器计算石英晶体的振荡。两个寄存器，即计数寄存器和保持寄存器，按如下步骤协同工作：计数寄存器在石英晶体的每次振荡时减1。当计数器达到零，传感器生成一个时间中断用于完成一个特定的时间任务，同时计数器依

据保持寄存器被重置（恢复为初始的计数值）。

我们如何给一个感知事件提供一个时间戳呢？这个时间戳依据系统时钟值获得，而系统时钟值来自计时器（如上所述）的读数。例如，每当计数器达到零时，计时器给系统时钟增加 1。

尽管我们期望所有的传感器节点内部时间计数器有完全一致的步调，但事实上，每个传感器节点中石英晶体的频率工作都略有不同，导致时钟值不断地偏离。我们称这种偏差为**时钟频差**（clock skew，后面将定义此概念），它导致不同传感器节点中时间概念的不一致。

一个内部计时器生成的时钟值也称为**软件时钟**（software clock）。它由晶体振荡决定。遗憾的是，大多数的晶体振荡都不够准确，因为使时间增长的频率不能完全一致。即便一个 0.001% 的频率偏差也能够造成大约每天 1 秒的时钟错误。 308

现在我们看到了时钟同步的目的——在分布式的传感器网络系统中纠正时钟频差。有两种常用的纠正时钟频差的方法：

1）绝对同步（absolute synchronization）：所有传感器节点的时钟应该同步到一个精确的实时标准时间，例如 UTC（Universal Coordinated Time，通用协调时间）。换言之，所有的本地时钟不能只是与彼此进行同步，也要与物理时间关联。

2）相对同步（relative synchronization）：在一些应用中，我们不能要求所有的时间同步到一个全局时间。相反，时钟彼此间进行相对同步，因为这一需求只是提供一个事件次序，而不是提供每个事件发生时准确的真实时间。

接下来让我们进一步定义在时钟同步中要用到的重要概念。



时间（time）：在一个传感器节点 p 中，时钟读数（即它所声称的时间）由函数 $C_p(t)$ 定义。如果 t 是全局标准时间，$C_p(t)=t$ 表示一个标准时钟（即没有时钟倾斜）。

时钟偏移（clock offset）：我们定义一个时钟（$C_p(t)$）报告的时间与真实时间（t）的时间差为偏移（offset）。时钟 C_p 的偏移被表示为 $C_p(t)-t$。在时刻 t，时钟 C_p 相对时钟 C_q 的时间偏移为 $C_p(t)-C_q(t)$。

时钟频率（clock frequency）：频率是指时钟计数的速率。在时刻 t 时钟 C_p 的频率是它的时间函数 $C_p'(t)$ 的导数（定义见上文）。一个标准时钟的频率是 1。

时钟频差（clock skew）：时钟频差是指时钟（频率为 $C_p'(t)$）和标准时钟（频率为 1）的频率差值，即一个传感器节点的时钟频差为 $C_p'(t)-1$。在时刻 t，时钟 C_p 相对于时钟 C_q 的频差是 $(C_p'(t)-C_q'(t))$。

时钟漂移（clock drift）：时钟 C_p 的漂移是指时钟值相对于实际时间偏差的二阶导数，记作 $C_p''(t)$。在时刻 t，时钟 C_p 相对于时钟 C_q 的漂移记为 $(C_p''(t)-C_q''(t))$。

下面我们考虑在无线传感器网络中，相对于标准时间 UTC 的物理时钟同步。假设 UTC 时间是 t，我们当然希望对于所有的 p 和所有的 t 都有 $C_p(t)=t$，即时钟频率 $dC/dt=1$。

但由于时钟频差的存在，传感器节点 p 时钟中的时间 $C_p(t)$ 并不等于 t。在这种情况下，如果一个计时器（时钟）的时间频率在一定范围内，它被认为按一定规范工作。 309

$$1-\rho \leqslant \frac{dC}{dt} \leqslant 1+\rho$$

其中，常数 ρ 表示制造商规定的节点内时钟频率的最大偏差率。

或者，我们可以认为时钟频差（相对于标准时钟）在一定的范围内：

$$-\rho \leq \left(\frac{dC}{dt}-1\right) \leq \rho$$



根据前面的定义，$C(t)$ 是本地时间。它可以是任意函数，即它可能与标准时间不一致（函数为 $C(t)=t$）。我们也知道时钟频率（率）是它的导数，即 $C'(t)=dC/dt$。对于标准时间，$C'(t)=1$。时钟频差是本地时间的频率和标准时间的频率差值，即 $(dC/dt)-1$。

图 10-1 表示相对于 UTC，快速、慢速和标准时钟的行为。

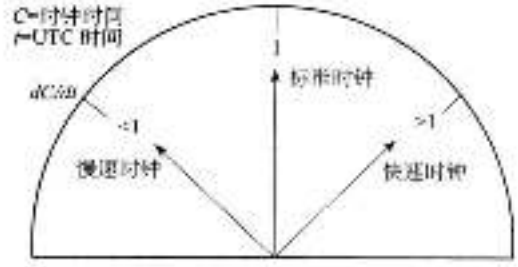


图 10-1　表示相对于 UTC，快速、慢速和标准时钟的行为

对于利用节点间的网络消息交换实现同步的时钟同步协议，有一些基本需求：

1）同步协议应该对于没有上限的消息传输延迟和不可靠的无线通信具有鲁棒性。

2）如果一个节点想要与另一个节点同步，它必须能够预测对方时钟的本地时间，而由于网络延迟，这是很困难的。 310

3）我们不能执行时间回退，即我们不能设置回退的时钟。所有时钟只能一步步地增长，直到达到正确值。

4）从网络通信的角度看，我们应该最小化同步开销。例如，我们不能使用太多的节点间消息交换。

10.2　一般网络（非无线传感器网络）中的时间同步

在讨论无线传感器网络中的同步问题之前，我们先看一下在一般网络（即非无线传感器网络）中已经得到解决的几个问题。

10.2.1　远程时钟读取

如前所述，消息交换过程可以用来完成任何两个节点之间的时钟同步。因为一个节点不知道其他节点的本地时钟值，它只能估计其他节点的时钟时间。这种估计应考虑网络延迟的影响。得到估计的时钟值后，它可以计算节点时钟之间的时间差，调修本地时钟。

然而，不确定性和无上限的消息延迟的存在使同步变得很困难。因此，一个同步协议的有效性在于它能够避免由于不确定的消息延迟而影响同步的质量。

远程时钟的读取方法［FCristian89］能够处理进程间（进程是节点上的一个时钟估计程序）存在的无上限的消息延迟问题。通过使用远程时钟读取方法，可将多个客户程序同步到一个精确的时间服务——UTC。

图 10-2 显示了远程读取其他节点的时间的过程：

1）在本地时间点 T_0，客户端发送一个消息向服务器请求时间戳。

2）服务器返回带有时间戳 S_{srv} 的消息。注意 S_{srv} 是服务器上的本地时间。

3）客户端在当地的时间 T_1 收到此消息。

4）客户端设置本地时间为 S_{srv}（来自服务器的准确时间）+（$T_1 - T_0$）/2（发送消息所需的时间）。

5）为了提高准确性，将重复步骤 1～4 和并使用平均值。

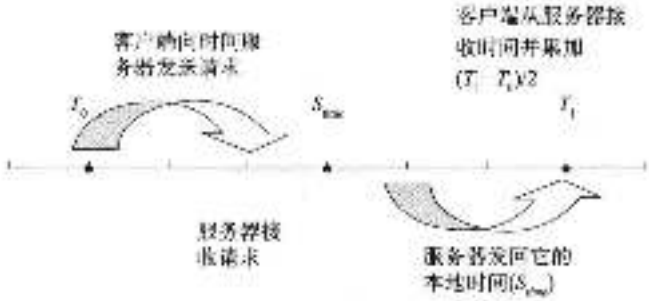


图 10-2　Cristian 的时钟同步协议

10.2.2　偏移时延估计方法

目前，互联网上采用最广泛的时钟同步方法是 NTP［DLM92］，它通过偏移量的时延估计方法对时钟偏移（见前面的定义）进行估计。 311

Cristian 时钟同步协议采用了基于树状分层结构的时间服务器设计。树的根节点是主服务器，与 UTC 同步。辅助服务器，作为主服务器的备份，都包含在树的下一个级中。客户端节点位于树的最低级。这些客户节点需要与树的根服务器同步（根服务器与 OTC 同步）。

由于消息传输在网络中的延迟不同，客户端节点无法准确估计目标节点的本地时间，因此 NTP 执行许多次往返试验并选择延迟最小的试验。这类似于前面所述的 Cristian 的远程时钟的读取方法［Cristian89］，这也依靠与估计的消息延迟相同的策略。

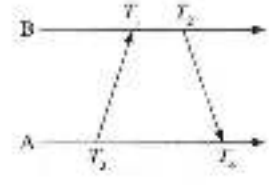


图 10-3　偏移和延迟估计

如图 10-3 所示，假设节点 A 和 B 交换 NTP 时间戳。在 T_1 时刻，节点 A 发送一条消息；在 T_2 时刻，节点 B 获取了这条消息，并在 T_3 时刻发出反馈消息，反馈消息在 T_4 时刻由节点 A 接收。假设 A 和 B 的时钟是稳定的，并且以相同的速度运行。 312

$a = T_2 - T_1$ 是从 A 到 B 的消息传输延迟。

$b = T_4 - T_3$ 是从 B 到 A 的消息传输延迟。

虽然由于通信链路的不对称性会导致 a 和 b 不相同，但在大多数情况下其差值很小。我们定义时钟偏移 θ 和往返延迟 δ 如下：

$$\theta = \frac{a+b}{2}, \delta = a - b$$

值得注意的是，当消息在 A 和 B 间传送时，我们能够在消息头中记录三个时间戳 T_1、T_2 和 T_3 的值。然而，时间戳 T_4 的值不能存放在消息头中，因为它只有在消息抵达时才能确定。这样，A 端和 B 端能独立地使用单个双向消息流计算时钟偏移 θ 和往返延迟 δ，如图 10-4

所示。

基于图 10-4，我们就可以描述 NTP 协议了，具体如下。

假设服务器 A 和 B 为了实现时间同步而交换时间消息。服务器 A 在第 i 轮计算中计算参数对 (O_i, D_i)，这里，O_i 是此轮计算中的偏移量（即 θ），D_i 是传输延迟（即 δ）。对于每轮计算 (O_i, D_i)，我们选择与最小延迟相对应的那个偏移。

具体而言，延迟和偏移参数对 (O_i, D_i) 由以下方法计算：假设消息 m（从 A 到 B）的传输时间为 t，而消息 m'（从 B 到 A）的传输时间为 t'。已知 O_i 是 A 的时钟和 B 的时钟的偏移，如果 A 的本地时钟是 $A(t)$，B 的本地时钟是 $B(t)$，则有：

$$A(t) = B(t) + O_i$$

继而，

$$T_{i-2} = T_{i-3} + t + O_i$$
$$T_i = T_{i-1} - O_i + t'$$

一般情况下，我们假设互联网（但不在一个无线传感网中）中 $t = t'$。然后，我们让这两个等式互减，偏移 O_i 可被估算：

$$O_i = \frac{T_{i-2} - T_{i-3} + T_{i-1} - T_i}{2}$$

我们还可估计往返延迟 D_i 为：

$$D_i = (T_i - T_{i-3}) - (T_{i-1} - T_{i-2})$$

NTP 计算 8 个最近的参数对 (O_i, D_i)，选择对应于最小 D_i 的 O_i 值来估计整个网络的平均偏移 O。

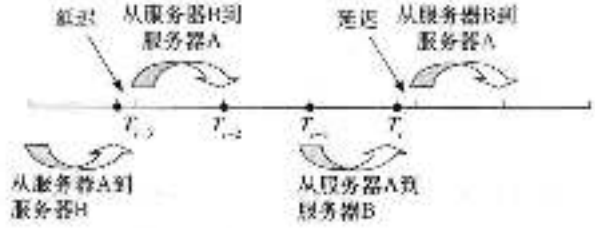


图 10-4　两个服务器（A 和 B）间的时间流图

由于采用平均方法（经过 8 轮计算），偏移/延迟估计协议与前面介绍的 Cristian 时间同步方法相似［FCristian89］。但是，考虑到消息的复杂性，两个方法的同步开销都很大。由于前向和后向消息的使用在一定程度上抵消了部分延迟，因此 NTP 的精度要比 Cristian 时间同步方法高。

10.3　无线传感器网络中的时钟同步

传统的时钟同步协议在有线网络中能够很好地工作。但是，它们并不适用于无线传感器网络环境，原因如下：

1）**能效**：由于所有的传感器节点都是由电池供电的，因此在无线传感器网络中节省能量就变得非常重要。传统的协议（如 NTP［DLM91］）使用外部标准（如 GPS 或 UTC）将整个网络同步到一个精确的时间源上。然而，GPS 的使用对能量要求很高，而无线传感器网络往往不具备这个条件。这样，即使维护普通的时间一致性也变得很困难。

2）**动态网络拓扑**：对于一个没有任何移动节点的静态拓扑结构的传感器网络而言，需要

一个初始化的设置使该网络开始工作。然而，随着新传感器节点的不断加入，或者是有些节点由于能量耗尽而停止工作时，每个节点的邻居以及该网络的配置会发生变化。更糟糕的情况是在有些应用中传感器节点是可以移动的，整个网络的拓扑结构变化得更加剧烈。因此，无线传感器网络的时钟同步协议应该同时考虑到具有静态和动态拓扑结构的网络，而且必须能够保证自配置（self-configuration）（这是通过使用合适的邻居节点发现或簇首节点选择协议实现的）。

3）**端到端延迟**：互联网使用 NTP 协议很好地实现同步，这是由于它是基于充分互联的有线网络。在这样的网络中，消息传输延迟相对稳定（即我们能在整个网络中获得一个不变的端到端延迟）。然而，无线传感器网络具有高误码率，并在共享介质上进行无线传输。在网络中两个端点之间进行延迟范围的假设是不可能的。因此，以充分互联且延迟恒定为前提条件的同步协议无法适用于多跳的无线传感器网络。

313
:
314

4）**无线损耗**：传统的有线网络很少发生数据丢失事件。但是，在一个无线传感器网络中，由于频繁的无线损耗，我们必须使用多轮消息交换确定时间参数（如时钟偏移）。

- **无线传感器网络同步协议分类**

研究者已经提出几十种不同的无线传感器网络时钟同步协议，这些协议可分为以下几类：

（1）发送者－接收者 vs. 接收者－接收者同步协议

发送者－接收者同步协议（Sender-to-receiver synchronization）：我们已经在前面介绍了 NTP 协议，它属于“发送者－接收者”消息交换方法。此种方法一般包含三个步骤：1）发送者向接收者发送一个消息（以这个消息的本地时间作为时间戳）；2）接收者发回一个消息（以这个消息的本地时间作为时间戳）；3）通过衡量所有轮的往返时间计算发送者和接收者之间的消息延迟。

这些步骤需要运行多轮才能够得到一个平均值。

不足之处：每轮消息交换的延迟变化很大，具体情况依赖于发送者和接收者之间的距离而定。当一个消息经过多跳到达接收者时，延迟可能会很大且在不同轮中差异很大。虽然我们在多轮消息往返后能计算平均消息延迟，但却无法做到精确估计时间参数。同时，过多轮的消息交换会显著地加网络开销。而且，当计算时间偏移时，我们必须考虑接收者处理消息所用时间以及发送者准备发送消息所用时间的优化问题。

接收者－接收者同步协议（receiver-to-receiver synchronization）：我们可以使用基于接收者－接收者的同步协议克服以上问题。该协议基于以下假设：一个发送者向两个相邻的接收者发布消息时，我们认为这两个接收者几乎同时收到该消息。该方法应用了无线传输介质的广播属性。不用通过在发送者与接收者之间发布多轮消息，接收者便能够交换它们接收到该消息的时间，然后基于两个接收时间的差计算偏移。

显然，该方法能够降低消息延迟的变化程度。我们只需关心不同接收者的传播延迟和接收时间的差即可。

315

（2）时钟校正 vs. 自由时钟

时钟校正（clock correction）：今天人们处理同步问题常用的方法是通过在每个节点中实现类似于原子钟或全球时间标准（例如 UTC）的功能校正本地时钟。这样的时钟校正方案提供了方便的参考时间。一个节点可断间或持续的方式校正其本地时钟以保持整个网络同步。

自由时钟（untethered clocks）：时钟校正需要持续的同步操作，这将消耗很多能量。自由时钟并不是根据全球时钟校正本地时钟。相反，它只维护本地时钟与其他节点时间的一个对照表。这样，就能监控相对时间差。例如，参考广播同步（Reference Broadcast Synchronization，

RBS）[Elson02]（稍后介绍）建立一张参数表用以将每个节点的本地时钟和网络中其他节点的本地时钟关联起来。使用此表可对本地时间戳进行比较。这样，时钟就可以自由运行，同时还可保证全球时间标准。

（3）内同步 vs. 外同步

内同步（internal synchronization）：在内同步方法中，无线传感器网络不具备全球时间基础。因此，目标是使得由传感器节点的读数得到的本地时间差最小。

外同步（external synchronization）：在外同步方法中，系统依赖于一个标准时间源（如UTC）。NTP [DLM91] 通过此种方式同步互联网中的节点。但是，除非应用有需求，否则无线传感器网络一般不会去实现外同步。原因在于，能耗是一个首要的考虑因素，使用一个外部时间源一般会包含高能耗的需求。

内同步方法通常需要更多的校正操作，外同步方法则能为系统提供便于使用的参考时间。内同步可在端到端（peer-to-peer）（即在没有中心服务器的情况下）或主－从（master-slave）模式中实现，而外同步则只能在主－从模式中实现，因为该方法需要一个主节点通过时间服务（如GPS）进行通信将从节点和它自己同步到一个参考时间上。

（4）概率同步 vs. 确定性同步

确定性同步（deterministic synchronization）：这是一个常用的实现时钟同步的方法。该方法使用确定性同步算法/协议给出一个时钟偏移的上限，也就是说，它能给出一个确定的时钟精度。

概率同步（probabilistic synchronization）：该方法并不能提供一个绝对的时钟精度，它仅能给出一个显示其时钟偏移控制的概率值，即该方法在绝对概率中会有失效概率。虽然概率同步方法的时钟精度比确定性概率方法差，但它不需要在同步协议中执行过多的消息传送操作，并
316 由此避免了额外的处理操作，这使得该方法有助于节能。

（5）静态网络 vs. 动态网络

动态网络（mobile network）：传感器节点可以移动，并且只有进入到其他传感器节点的通信范围内时，才可与这些传感器节点进行通信。我们需要一个鲁棒同步协议应对由于节点的移动性而导致的网络拓扑结构的变化。

静态网络（stationary network）：大多数无线传感器网络的网络拓扑结构是静态的，也就是说，传感器节点不会移动。设计静态无线传感器网络的同步协议相对容易些。

10.4　同步性能的评估

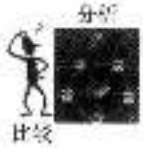
时间同步协议依赖于每个应用的特征和需求，例如，大多数无线传感器网络采用低成本低精度的同步协议。然而，在一些对安全性要求高的应用中，例如飞机导航或军事系统的入侵检测应用，则需要高精度的同步协议精确地识别发生在特定时刻的事件。

在文献 [Sundararaman05] 中，使用如下性能指标来衡量同步协议的质量。

10.4.1　精度

硬件时钟（hardware clock）：传感器中的内部硬件（震荡电路）产生初始的时钟信号。如前所述，硬件时钟具有时钟频差，因此，不能直接使用由这样的硬件时钟产生的时间。

逻辑时钟（logic clock）：因为硬件时钟并不精确，传感器通常采用逻辑时钟和时间。在同步协议执行过程中，利用软件调整逻辑时钟。这里，所有讨论都是指逻辑时钟。



基于逻辑时钟概念，我们能够定义两类同步精度：

绝对精度：使用外部标准（如 UTC）衡量传感器节点逻辑时钟的最大时钟频差误差或时钟偏移误差。

相对精度：在不与一个标准时钟进行比较的情况下，只测量同一网络中节点逻辑时钟读数间最大时钟频差误差或时钟偏移误差。

317

显然，任何无线传感器网络同步协议的目标是获得绝对精度或相对精度。然而，高同步精度是以增加计算复杂度和通信开销为代价的（即节点间交换消息的次数）。

准确性是一个与精度相似的指标，它衡量无线传感器网络所保持的时间与标准时间相比的准确程度。具有高准确性的同步协议才能保证高精度。

10.4.2　协议开销

为了减少协议开销（即节点间交换消息的数量），可以采用携带式机制。它是一个将应用程序数据确认消息和携带时间同步信息的消息结合起来的处理过程，即不使用单独的消息传输时间信息。这样的控制信息能够被嵌入到普通的传感数据包中。这样不仅能够减少通信开销，还可节省存储空间。

10.4.3　收敛时间

收敛时间是指协议同步整个网络所需的总时间。一些同步协议仅能进行偶尔的时钟控制，也不常使用消息交换。因此，这些协议的收敛时间并不长。然而，另一些要求高时钟精度的同步协议在每次同步操作中需要大量的消息交换，导致了更长的收敛时间。

10.4.4　能效

能效一直是无线传感器网络协议设计中最重要的问题之一。复杂度低的协议可以节省更多的能量。

10.4.5　可扩展性

无线传感器网络的同步协议可以将大规模网络同步到相同的时间信号上，换句话说，它应该具有将几百个节点进行同步的能力。

10.4.6　鲁棒性

无线传感器网络在恶劣环境下采用低带宽无线通信，因此，其同步协议应能够容忍高丢包率和强无线干扰。

318

10.5　无线传感器网络同步协议的例子

10.5.1　参考广播同步

RBS（Reference Broadcast Synchronization）协议是一个基于“接收者－接收者（receiver-to-receiver）”的设计方案［JElson02］。两个接收者几乎可以同时接收到来自同一发送者的同一个消息。RBS 是因消息（即来自同一个发送者的广播）几乎以很小的延迟到达各接收者而得名。

如果每个接收者在消息一经抵达时就记录本地时间，那么所有接收者就可以通过比较它们的本地时钟值（必须是同一个消息抵达时产生的）获得高精度时钟同步。

RBS 通过时间关键的路径广播消息，该路径传递一个用于在协议中计算非确定性时钟误差的消息。在图 10-5 中，能够看到“发送者 - 接收者”和“接收者 - 接收者”之间的时间关键路径的差别。

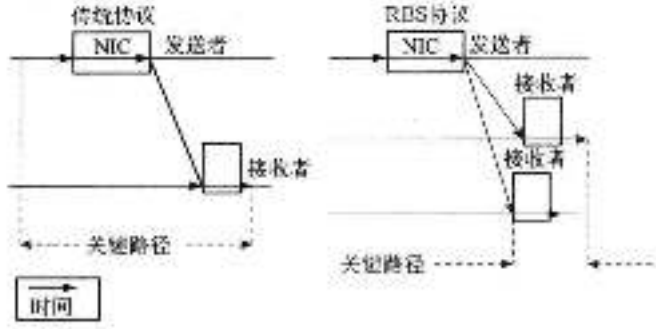


图 10-5　传统协议（左）和 RBS 协议（右）的时间关键路径

[319]

移除或减少非确定性传输延迟所产生的效果非常重要，因为这会降低同步协议的精度。它们还使得接收者在估计消息发送时间上产生困难，反之亦然。

在无线传感器网络中，当一个发送者向一个接收者发送消息时，以下四个时间因素是不确定的：

- 发送时间：这个时间由消息离开发送端之前的所有操作的时间组成，它包括：1）发送者（在本地机器上）构造消息所需时间；2）消息从发送者主机传输到网络接口所需时间（然后，准备离开发送者）。
- 访问时间：在无线传感器网络中，发送者或接收者在无线信道准备就绪（即没有其他节点使用该信道）前都需要等待一段时间。
- 传播时间：这是真正的“空中传输”时间。它是消息离开发送者和到达接收者这个过程的所需时间。
- 接收时间：接收者一旦收到消息后在本地进行处理所需时间。

与基于“发送者 - 接收者”方法相比，RBS（基于“接收者 - 接收者”方案）只考虑消息到达不同接收者的那部分时间。因此，该方案直接去除了消息传输过程中两个最大的不确定源，即发送时间和访问时间。这样，该协议能够为传感器网络提供高度的同步精确性。

按照以下简单的步骤评估两个接收者的时钟之间的相位偏移（phase offset）：

1）一个发送者向两个接收者广播一个参考包（即消息）。

2）每个接收者记录该包被接收时的本地时间。

3）（重要步骤）两个接收者交换各自记录的本地时间，该本地时间是它们接收到同一个包的时间。

4）通过计算两个接收者接到相同消息时的本地时间差，就能够计算出这两个接收者的时钟偏移。

为了应用 RBS 协议来获得高时钟精度，对每个接收者而言，在消息抵达后尽快记录下本地时钟读数就变得非常重要。然而，接收节点或许无法快速地记录消息抵达的时间。例如，当消息抵达时，节点正忙于其他计算。

显然，RBS 不能只采用一种消息传输机制缓解这些非确定性的时间因素所带来的负面效

果。在实际应用中，RBS 协议采用来自同一发送者的参考消息序列。令参数 i 和 j 表示两个接收者。假设总共有 m 个消息被发送。接收者 j 会计算相对于任何其他接收者 i 的偏移作为每个被节点 i 和 j 接收到包的平均时钟差：

$$\text{Offset}[i,j] = \frac{1}{m}\sum_{k=1}^{m}(T_{i,k} - T_{j,k})$$

这里，$T_{i,k}$ 是节点 i 在接收到广播 k 时的时钟。

借助于“接收者－接收者”的时间对比，RBS 通过将接收者从发送者中分离出来而去除了关键路径上最大的误差源（发送时间和访问时间）。时钟偏移和频差能够分别进行估算。此外，由于本地时钟从未被改变，因此时钟校正也不会对估计产生影响。

然而，对于一个有 n 个节点的单跳网络而言，该协议需要 $O(n^2)$ 次消息交换，因此它可能导致通信开销很大。因为有大量的消息交换，收敛时间（即同步整个网络所需要的时间）会很高。

10.5.2　时间扩散同步协议

时间扩散同步协议（Time-Diffusion synchronization Protocol，TDP）是一个可扩展协议。它可保证无线 WSN 中所有传感器节点具有一个同步时间，这一同步时间有一个较小的时间偏差范围这个时间偏差来自于全网时间“均衡”。该协议包含多个算法。

为了保证高精度时钟同步，TDP 采用周期运行机制，因而具有交替出现的“活跃的（active）”和“不活跃的（inactive）”两个阶段。在每个活跃阶段，存在多个“循环（cycles）”，每个循环持续时间为 τ。在每个循环中，通过“选举/改选过程（Election/Reelection Procedure，ERP）”选择一些节点作为主节点。

通过 ERP 选择出来的主节点可以并行进行时间消息的扩散。通过这些时间扩散消息，能够动态创建一个树状传播结构。

ERP 还会在这个树中选择一些非叶子节点作为“扩散领导节点”，这些节点也传播时间消息。有可能发生这样的情况：有的节点不具备作为扩散领导节点的资格，因此也就不会传播这个扩散（消息）。

ERP 有两个重要目标：

1）ERP 能使用艾伦方差（Allen variance，一种方差计算方法）清除那些时钟方差高于阈值函数所确定值之上的孤立点。艾伦方差算法通过进行消息交换以及通过同级评价方法（Peer Evaluation Procedure，PEP）计算各对相邻节点间的偏差来确定方差。

2）ERP 能获得这些节点中网络流量（也称为负载）分布，原因在于主节点和扩散领导节点角色的存在对能源提出了更高的要求。基于诸如可得到的高于一个可变阈值的能量等级（水平）这样的因素，它通过轮流指定主节点的方式获得负载分布。

TDP 是一个典型外部同步（见 10.3 节）协议，换句话说，它采用时间消息的扩散机制收敛到本地时间上，并最终达到一个共同的“系统范围时间”。

如前所述，TDP 具有交替的活跃和不活跃状态阶段。每个活跃阶段包括多个循环，每个循环的持续时间为 τ，每个循环（τ）包含两个顺序执行的任务：1）使用 PEP 确定主节点和扩散领导节点，2）运行主时间扩散过程（Time diffusion Procedure，TP），且每个 TP 包含多轮（每轮持续时间为 δ）。图 10-6 给出了循环、PEP 和多轮 TP 之间的关系。

ERP 选择主节点后，每个主节点并行地开始沿着树状结构进行时间消息广播，如图 10-7 所示。主节点时间能调制到外部的精确时间，该精确时间由周期广播参考时间的外部时间服务

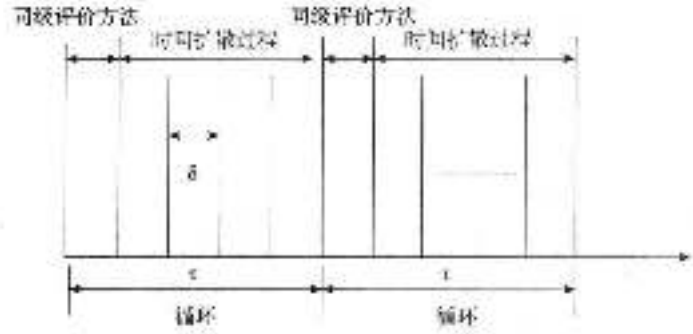


图 10-6 一个循环中，多轮 TP（每轮持续时间为 δ）和 PEP 之间的时间关系说明，每循环持续时间为 τ

器产生。如果该服务器不存在，该协议则使用 UTC。

参照图 10-7，读者就可以了解到主节点是如何进行时间消息扩散的（从树中级别 1 开始）：首先，主节点向它的邻居发送多个时间消息。邻居节点发回确认，该确认消息包括来自主节点时钟的本地时钟两样本艾伦方差（two-sample Allen variance）。基于接收到的样本，主节点计算（a）自身节点及每个邻居节点的离群孤立比率 γ_0，（b）艾伦方差的均值以及（c）艾伦偏差的均值。现在，值（a）、（b）和（c）被放在 RESULT 消息中，然后再被发送给每个邻居。

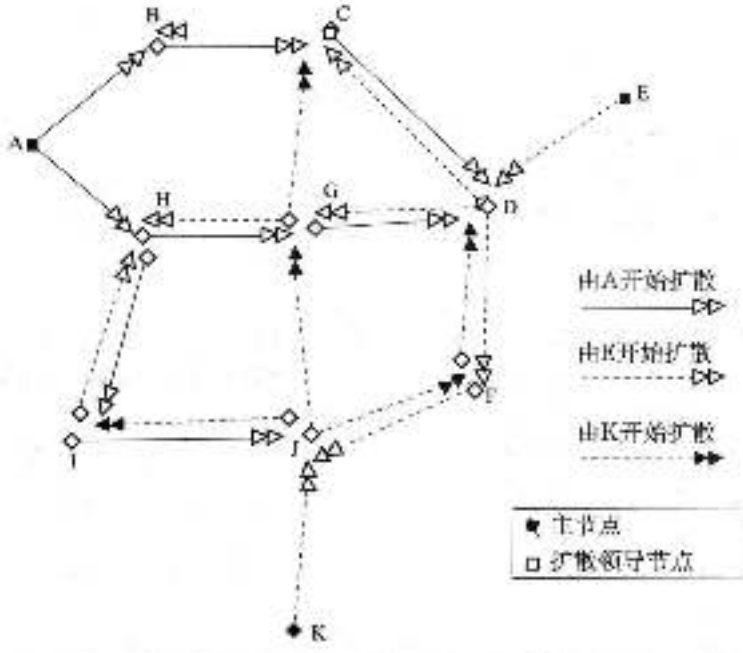


图 10-7 具有 3 个主节点的且 n 为 3 跳的时间扩散，在每一轮中，节点取不同接收时间的跳段偏平均值。这些时间来自主节点的扩散广播

在每个后续步骤（$j=2, 3, \cdots, n$）中，在每个处于扩散领导节点和及其邻居之间的级别大于 1（$j>1$）的过程中都会重复级别为 1 的过程。

在这个时间消息扩散过程中，所有传感器将会监听到这个离群孤立比率和（关于它们邻居的）平均艾伦偏差。它们将利用这些值估算它们邻居的时钟的质量。如果节点的离群孤立比率大于 1，它的本地时钟就可由经过至少两次的艾伦方差其邻居们的时钟产生。在这种情况下，这个节点在当前循环（的时间扩散过程）中不会成为一个扩散领导节点，或下一个循环中的主节点。

值得注意的是，当 TDP 考虑到负载分布（前文已有提及）时，有资格在下一个循环中成为主节点的节点并不能保证可以实现此目标。只有当节点中可用能量处于一定（动态可调整的）阈值之上时，该节点才有可能成为主节点。负载平衡由主节点角色更迭来实现。

TP 过程主要完成从每个主节点以树状 n 跳方式进行时间扩散的功能。这里，n 是预先指定的小于网络直径的参数。

总之，TDP 协议能获得一个系统范围的所有节点的“权衡”时间，使用迭代加权平均技术进行计算，并在同步过程中涉及所有节点。

322
/
323

10.5.3　概率时钟同步

大多数无线传感器网络同步方案都属于确定性算法，换句话说，它们能在时钟偏移估计中保证误差上限。另一方面，像这样的确定性算法在同步过程中需要交换大量的信息。这并不适用于资源受限的无线传感器网络。

因此，有人提出利用概率算法在一定概率上给出合理的同步准确性即可。此方法的优点是计算复杂性低、网络开销小。PalChaudhuri 等人 [SPalChaudhuri03] 提出了一个基于对 RBS 扩展的概率同步方案。该方法给出了时钟同步准确性的概率约束，还允许在低成本 WSN 中于计算资源和能量资源的同步精度之间取得权衡。

如前所述，RBS 利用从发送者到一组接收者的多个消息。通过交换消息，所有接收者都知道实际接收时间的差值。发送者以独立分布方式发布一组消息，因此接收时间的差值能够用带零均值的高斯分布（或正态分布）来描述。

同步误差也可用高斯概率分布进行描述。然后，能容易地以误差小于最大误差的方式计算出给定的最大同步误差和实际同步概率之间的关系。

让我们假设最大同步误差（限于两个进行同步的节点之间）为 ε_{max}，然后，用高斯分布特性推导出误差为 $\varepsilon \leqslant \varepsilon_{max}$ 的同步概率：

$$P(|\varepsilon| \leqslant \varepsilon_{max}) = \frac{\int_{-\varepsilon_{max}}^{\varepsilon_{max}} e^{(-x^2/2)} dx}{\sqrt{2\pi}}$$

由此，能看到当提高 ε_{max} 时，失败概率（$1 - P(|\varepsilon| \leqslant \varepsilon_{max})$）以指数速减。

在文献 [SPalChaudhuri03] 中，最大时钟同步误差被转换为实际的协议参数（消息数量和同步开销），所获得的误差（小于最大指定误差）的概率为：

$$P(|\varepsilon| \leqslant \varepsilon_{max}) = 2erf\frac{\sqrt{n}\varepsilon_{max}}{\sigma}$$

这里，n 是保证误差所需的同步消息的最大数量，σ 是分布的偏差。

324

总之，概率同步方法能够在同步精度和传感器资源成本之间取得很好的权衡。在对安全要求苛刻的传感器网络应用中（如核电站监控），概率同步方案会因其无法产生确定的同步精度而无法胜任。

问题与练习

10.1 为什么传统的有线网络中的时钟同步方案并不适用于无线传感器网络？以 NTP 为例说明它们的缺点。

10.2 解释偏移、频差和漂移的概念。

325 10.3 使用某种软件实现讨论过的一种无线传感器网络时钟同步算法。

无线传感器网络安全与隐私



国土安全是每一个国家最为重视的问题。特别是网络安全，在当今社会中具有越来越重要的地位，因为很多社会活动都是基于计算机交流的。如果用于有特殊要求的应用，无线传感器网络也需要具有安全方案，例如建筑物监控。

11.1 引言

11.1.1 一般攻击类型

Tanya 等学者总结了无线传感器网络的一般攻击类型以及相应对策［Tanya06］。根据文献［CKarlof03］提供的分类标准，传感器网络的攻击可以分为三类：

（1）节点级/便携计算机级攻击者

一般情况下，节点级攻击者没有足够的资源发动强大的攻击。但是它可以攻击低能量的传感器节点。便携计算机级攻击者可以访问功能更强的设备，例如笔记本电脑。这种功能强大的设备可以使攻击者有能力发动更强大的攻击。

（2）内部/外部攻击者

外部攻击者没有进入传感器网络的特殊路径，因为它不知道无线传感器网络密钥。但是，它可以通过使用被动窃听获取数据。内部攻击者是比较难以预防的，因为它可以访问网络中使用的加密密钥或者其他代码。一个被攻击节点（之前它是网络中的合法一部分）可以被认为是一个内部攻击者。

（3）被动/主动攻击者

被动攻击者通过被动监听网络数据获取隐私与保密要求。但是，主动攻击者可能通过主动攻击无线传感器网络损坏其功能。例如，攻击者可能会假装成合法节点，将错误数据注入到网络中。



了解各种无线传感器网络攻击的区别是设计抵御机制的前提。需要注意的是，目前存在很多种 WSN 攻击分类的标准。例如，当我们从“攻击强度”的角度考察某种攻击时，此种攻击的发起者可能会从“攻击代价”的角度采用低成本设备来降低开销。在现今社会，鉴别外部入侵者相对容易。但是发现内部的间谍就没那么容易了。同样的，在无线传感器网络中，内部攻击比外部攻击更具有威胁性。接下来，本章从其他角度对无线传感器网络攻击进行分类，例如五层协议模型。

11.1.2 物理节点攻击

在无线传感器网络中，传感器节点易于受到恶意物理篡改（例如，破坏节点外部封装直接

获取内存数据)。这样的物理篡改使得节点软件易受到外部攻击。遗憾的是，现今的商业传感器硬件无法抵抗物理篡改［YangXiao07］。如果攻击者获取到一个节点，他/她可以轻易地利用节点软件的缺陷。

针对无线传感器网络的物理攻击主要包括以下两种类型［Tanya06］：

1）侵入式攻击：侵入式攻击者使用逆向工程的探针技术研究设备的芯片级组件。攻击者使可无限制地访问存储在这些组件中的任何或者全部信息。逆向工程分析可以轻易地导致传感器系统遭受重大损失。

2）非侵入式攻击：非侵入式攻击者不打开也不会物理篡改嵌入式装备。例如，侧信道攻击可以使用密码系统的物理实现中收集到的信息，获取一些硬件信息，例如能量消耗、软件操
328 作的执行时序或者电磁（EM）波频率。



从前面的讨论中我们可以看到，要研究无线传感器网络的安全，我们不仅需要掌握传感器网络协议和密码学知识（这是一个典型的计算机科学领域），我们还需要学习一些电气工程知识，例如电磁波、逆向工程等。因此，无线传感器网络安全是一个交叉学科。

对于上述两种类型的攻击，入侵式攻击更为普遍。遗憾的是，目前并没有有效防止传感器节点受到物理篡改攻击的解决方法。传感器节点的微控制器和存储器缺乏基于硬件的内存保护。虽然有些嵌入式系统的加密处理器在物理上是安全的，但它们并不具有抵御物理篡改的一整套防护计划。因此，重要的是开发适用于低成本、低能耗要求传感器网络的可优化加密处理器。

另一方面，非侵入式攻击（例如侧信道攻击）也可以导致严重后果。例如，使用简单能耗分析和差分能量分析的侧信道攻击可以破坏信息认证码（MAC）［KOkeya05］。文献指出，可以通过能耗分析攻击提取安全密钥位。能耗分析可以启动无线传感器网络中的块密码攻击。在密码学中，块加密算法使用一个对称密钥对固定长度的位组进行加密。线性或差分密码分析通常用来发动这些攻击。如果块密码被用来作为哈希函数，攻击可以破坏哈希函数。

还有一种侧信道攻击叫做计时攻击，它利用非恒定指令执行时间泄露秘密信息。非恒定的指令执行时间可由有条件分支和各种优化技术引起。传感器节点操作系统是事件驱动，并且为了降低存储开销而对指令进行了优化，不同指令的执行时间之间存在较大差异，这使得计时侧信道攻击有可能发生。解决此类攻击的方法之一是使用指令执行时间恒定的软件。然而，这并不容易在无线传感器网络中应用。因此，寻找传感器网络中抵御计时攻击的方法是未来研究的重要课题之一。

基于频率的攻击也属于侧信道攻击。它的目的是提取对称加密算法的密钥。

目前有一些抵御侧信道攻击的方法，例如能耗随机化、CPU 时钟随机化、使用虚假指令、
329 使用位分裂等。

11.1.3 针对无线传感器网络通信协议栈的攻击

本节将从通信层的角度对无线传感器网络攻击进行分类，分为以下几类：物理层攻击、链路层攻击、网络与路由层攻击和传输层攻击［Tanya06］。

1. 物理层攻击

干扰是对无线传感器网络物理层最有威胁的攻击，它可以在传感器节点通信的无线信道内

部启动射频信号干扰。通过干扰几个关键节点的通信甚至可以扰乱整个网络，因为这些节点可能是所有的路由路径的交点。

扩频（SS）通信是一种常见的防御干扰攻击的方法。SS 包括跳频和码扩频。[AWood03] 也提出了一种抵御干扰攻击的解决方法。他们提出了一种通过隔离临近节点达到隔离受到干扰的网络区域的方案。通过这种隔离，我们可以利用剩下的网络实现预期的功能。



在无线物理层，由于干扰攻击容易开展，因此抵御无线信号干扰攻击是最具挑战性的问题之一。只要利用无线信号频率检测器和功能强大的信号发生器，由于产生点多干扰信号，干扰器就可以阻碍在一定频率下的正常数据通信。CDMA（码分多址接入）可以在一定程度上实现抗干扰通信。但在资源有限的无线传感器网络中，CDMA 可能会导致较高的通信开销。

2. 链路层攻击

数据链路层协议定义了相邻节点访问共享无线信道的调度方案。以下是一些链路层攻击的例子：攻击者通过损坏调度协议导致传输冲突；它可能通过重复传输耗尽正常节点的能量；它还可能导致相邻节点之间无法公平使用无线信道的现象。研究人员已经针对这些攻击提出了一些解决方案，例如冲突检测技术，通过改进 MAC（媒体接入控制）协议限制请求的速率，并为每个数据包使用更小的帧［AWood02］。

330



请注意，MAC 在本章中有着不同的含义。例如，它可能意味着介质访问控制协议，该协议管理无线接入中的无线共享方案。它也可能意味着消息认证码，这是一个特殊的通过原始信息数据计算出来的二进制序列。这样的代码用于认证目的，即验证接收到的消息是否来自一个正常的源（而不是来自攻击者的机器）。

3. 路由层攻击

我们知道，路由协议（也称为网络协议）试图找到一个从发送者到目的地的优化传输路径。这样的路径可能具有更高的能效，或者更低的延迟，或者更少的拥塞，或其他优势。在此路径上的节点被称为中继点，中继点具有类似互联网中的路由器功能。攻击者可能误导或损坏这样的路径。在本节，我们将根据［CKarlof03］［Tanya06］的研究，介绍几种路由协议攻击。

（1）欺骗、篡改或重放路由信息攻击

所有的数据传输都由路由协议控制。一个传输路径是通过相关传感器节点之间的协议消息建立的。因此，对于路由协议的直接攻击主要面向节点之间的路由信息交换。攻击者可以伪造、篡改或者重放路由信息，从而产生路由环（即从未到达目标端）、吸引或者排斥网络流量（即误导性路由）、延长或者缩短源路由、产生虚假错误信息（即报告错误状态）、分区网络（即使路由难以隔离子网络）、增加端至端延迟等情况。

（2）选择转发攻击

无线传感器网络使用逐跳路由协议转发节点数据。正常的多跳路由协议假定所有的中继节点会盲目地、诚实地转发接收到的信息。然而，当攻击者使用选择性、不诚实的转发策略时，它可以拒绝某些消息甚至完全删除它们。选择性转发会导致重要数据丢失，甚至可能破坏网络。

选择性转发攻击的一种特殊形式被称为“黑洞”（black hole）。类似于宇宙中的黑洞，攻

击节点可以拒绝转发它接收到每一个数据包。这种攻击的后果是，相邻节点会认为恶意节点失败，而选择一个替代路线。

在其他形式的选择性转发中，攻击者可以改变某些节点的通信，按其意图使用另外的节点进行通信。这种攻击可以有效地抑制这些节点的数据发送，而不会被怀疑。

331 在大多数情况下，当攻击者处于数据传输路径上时（即成为一个中继点），会发生选择性转发攻击。然而，攻击者可以通过其相邻节点偷听到数据流，然后它通过在每一个它感兴趣的转发数据包上干扰或引起冲突，启动选择性转发。

一般情况下，引发选择性转发攻击的攻击者会选择抵抗力最小的传输路径作为目标，然后尝试着将自己加入到这条数据流的实际传输路径中。

（3）sinkhole 攻击

与“黑洞”具有某些相似之处，sinkhole 攻击通过一个恶意节点吸引附近的数据传输，这个恶意节点可能是一个外部攻击者或者是本地的被攻击节点。sinkhole 攻击最终在攻击者的周围构造了一个“洞”。通过将数据吸引到自己这一边，它还有很多机会篡改应用程序数据。事实上，sinkhole 攻击可以发动很多其他攻击（例如选择性转发）。

sinkhole 攻击是如何将流量吸引到自己这边的呢？一个简单的方法就是让自己相比于周围的节点来看更具吸引力，比如通过向基站欺骗或重复播放广告它可以提供更高质量的传输路径。我们知道，无线传感器网络路由协议会回应这些广告。一旦周围的节点看到这个“有吸引力”的路径，它们会更倾向于向这条路径转发数据。



黑洞与 sinkhole：虽然它们都是使用一些方法通过恶意节点吸引数据流，但黑洞导致数据“消失”，而 sinkhole 并不只是丢弃这些数据。相反，它们保存这些数据进行进一步的处理，例如进行内容分析。因此，sinkhole 可能比黑洞更难监测到。

一些协议实际上可能会尝试使用包含可靠性或延迟信息的端到端认证验证路由的质量。例如，一个传感器节点可以随时要求告知数据去向。然后，一个“强有力”的 sinkhole 攻击者，例如一个具有强大射频模块的笔记本电脑攻击者，可以直接（即使用单跳而不是多跳）将延迟信息发送到基站或者使用虫洞攻击（将随后讨论）将延迟信息发送。由于被攻击的节点会广告出“看似”高品质的路径，攻击者的每一个相邻节点便被有可能通过恶意节点转发数据包（这些数据包本应到达基站）。更糟糕的情况是：一个正常节点可能会将这个“好”路径广告给其他临近节点。结果，sinkhole 攻击者创建了一个大的“势力范围”，吸引来自临近节点的流量。

那么，为什么传感器网络容易遭受 sinkhole 攻击呢？这归咎于无线传感器网络的协议模
332 式。在无线传感器网络中，一般是由基站作为所有节点传输数据的最终目的地。利用这一特性，一个被攻击节点就能很容易地向基站提供一条高质量的数据传输路径，于是所有的节点都会喜欢这条路径，然后向其发送数据。

（4）Sybil 攻击

在无线传感器路由方案中，Sybil 攻击可以严重破坏地理位置路由协议［Newsome04］。这是因为地理位置路由协议使用位置感知方案，要求节点与其临近节点交换坐标信息。通过使用 Sybil 攻击，攻击者可以使自己同时出现在多个地点。这使得网络无法有效定位数据包的地理位置，因为我们原本期望每一个传感器节点都有不同的坐标。

如果每对相邻节点使用一个唯一密钥初始化洞项或者扩频通信，那么攻击者都会很难启动上述攻击了。

(5) 虫洞攻击

虫洞攻击是无线传感器网络中最棘手的威胁。在这里我们重点介绍一些它的主要特点。在 11.2 节中，我们将分析对抗虫洞攻击的一些有效机制。

在虫洞攻击中，攻击者通过网络中的一个低延迟链路传输信息，然后将这些信息在网络其他部分重放。虫洞攻击通常包括两个远距离的恶意节点，然后将它们误认为是相邻节点。



奇思妙想

假设一个邮差需要将一些重要包裹从纽约市运送到旧金山市。在一般情况下，他会路过很多邮局。尽管过程很慢，但是这种多跳路径是安全的。但是，如果有人说，“嗨，我给你建立了一个路径。这个路径连接了纽约市附近的一个邮局（简称 A）和另一个旧金山市附近的邮局（简称 B）。从 A 到 B，仅仅需要 1 个小时的路程，因为在它们之间有高速列车”。基于正常的邮寄服务规则，邮差应找到最快捷的方式运输最重要的邮件。因此，他将采取这条路径传送邮件。但实际上，那条路径被攻击者完全控制了。然后，攻击者可以做任何它想做的事情了（例如打开每一封邮件，进行阅读）。

（上述比喻有助于理解虫洞攻击。）

333

在虫洞攻击中，攻击者应该设置两部机器：一部在消息源附近，另一部放置在基站旁（最终目的地）。在这两个机器间存在一个高质量链路（例如高速光纤）。

通过使用具有高质量链路的虫洞攻击，攻击者可以说服一般使用多跳的正常节点相信它们距离基站仅有一站了。

我们可以看到，虫洞实际上可以创建一个 sinkhole：攻击者提供了一条到基站的高品质路径，很有可能将周围区域的数据传输都吸引到这个“有吸引力”的路径上。

当然，如果消息源非常接近于基站，那么发动虫洞攻击就不那么容易了。

(6) HELLO 洪泛攻击

很多无线传感器网络路由协议要求节点向它们的临近节点广播 HELLO 数据包，这就是所谓的邻居发现。在收到这样的数据包后，节点可以假设发送者是在一个合适的接收距离内。但是，具有强大无线通信能力的攻击者可以使网络中的所有节点认为它是它们的邻居。

使用 HELLO 洪泛攻击的攻击者可以欺骗网络中的所有节点相信它是它们的邻居。如果攻击者实际处于很远的距离，那么这样的攻击会有效地导致大部分传输信息丢失。

受到 HELLO 洪泛攻击，无线传感器网络可能会陷入混乱状态。即使一个节点监测到一个路由问题，数据仍然不能被正确转发，因为所有的临近节点都会向攻击者发送信息。

特别是如果一个无线传感器网络路由协议依赖于邻近节点之间的本地信息交换进行拓扑维护或流量控制，它会很容易受到此类攻击。

为了发动 HELLO 洪泛攻击，攻击者并不需要有能力创建合法流量。它可以简单地使用一个强大的无线重新广播路由搜索包。这种高功率无线可以让网络中的每一个节点收到 HELLO 数据包。因此，在某种意义上，HELLO 洪泛攻击是一种单向的广播虫洞。

注意：当我们使用“洪泛”概念时，通常指的是在网络中通过一个多跳拓扑结构向每一个节点达成的信息传输。尽管我们如此命名这个概念，在这里我们仍然使用 HELLO 洪泛攻击表示攻击者使用单跳广播向大量节点传送信号。

（7）确认欺骗攻击

为了实现建立路径的可靠性，一些传感器网络路由算法依赖于显性或者隐性的数据链路层确认（ACK）。然而，由于无线链路的广播特性，攻击者可以欺骗寻址到邻近节点的数据链路层确认。

[334] ACK 攻击者的目的是说服一个邻近节点相信一个已不能工作的节点仍在工作，或者宣称一个微弱信号是强信号。这样的 ACK 攻击可以导致使用数据链路可靠性决定路径的网络中出现重大数据损失。

ACK 攻击加强了微弱或者不能工作的无线链路。这是一个细微但非常有效地操纵数据链路层确认机制的方法。由于数据包在经过微弱或不能工作链路时很容易丢包，攻击者可以利用 ACK 欺骗有效地发动选择性转发攻击。其结果是，目标节点将在这些链路上传输数据包。

4. 传输层攻击

传输层（例如 TCP）使用定时器、重传和端到端重传实现从信息源到目的地的可靠数据包传输［Internet07］。然而，由于资源限制，有线网络中的传输层协议并不能直接应用于传感器网络。在前面的章节中已经讨论过无线传感器网络传输层协议。

无线传感器网络传输层攻击的例子有洪泛和去同步攻击。洪泛攻击通过发送多个端到端链接建立请求，有效消耗节点的内存。去同步攻击通过使用数据包的不同序列号，试图伪造数据包发送到链接的一端或者两端。它触发链接的终点，请求重传“被认为”丢失的数据包。

源身份验证（source authentication）和客户端难题（client puzzle）是两种可能抵御此类攻击的解决方法［AWood03］。然而，我们依然不能确定这些方法是否可以用于传感器网络，应采取什么样的改进方式改善这些计划。

5. 流量分析攻击

我们知道，无线传感器网络的主要目的是从大量远程节点中收集数据到基站。因此，网络中的传输模式是多对一。这样就给了攻击者对网络发动攻击的机会。例如，攻击者可以分析传输模式，收集传感器网络的拓扑结构，以及通过观察流量和模式确认基站位置。

另一种流量分析攻击是通过观察流量，推断出多个路径的交叉点上的“重要”节点。然后攻击者可以攻击和破坏这些节点，最终将网络划分成几个相互分离的子网络。攻击者也可能对节点的顶点割集上发动拒绝服务（DoS）攻击。这些 DoS 攻击可能会耗尽传感器节点的能量，从而缩短网络的生命周期。

流量分析攻击可以以其他形式进行。例如，攻击者可以观察其临近节点的数据包发送速
[335] 率，然后关注具有更高数据包发送速率的节点。或者它可以观察一段时间内节点间数据包的发送情况，并尝试跟踪被转发数据包的发送路线，最终到达基站。

我们如何对抗流量分析攻击？一个可能的解决方案是“迷惑”攻击者。例如，在一个源和目的地之间，我们建立随机和多跳路径，或者使用概率路由，或者在网络中引入假消息。

在一个基于地理位置的概率路由（PGR）中，它根据临近节点一个子集中节点的链路质量和剩余能量选择随机下一跳。实验结果显示 PGR 高效节能，并具有较高的网络吞吐量。

使用“迷惑”信息可能会增加网络的能源消耗和网络内流量。这些信息看起来像是真的，所以，假消息不能被优化。

11.2 攻击与对策示例：虫洞攻击

首先，我们根据发动攻击的技术对虫洞攻击进行分类。

1. 基于报文封装方式的虫洞攻击

文献［Issa06］分析了一种通用的虫洞攻击。它使用动态源路由协议（DSR）作为例子。

在 DSR 中，如果节点 S 需要找到到达目的地 D 的路径，节点 S 将向整个网络发送一个路由请求（RREQ）数据包。听到该请求的每一个节点处理该数据包，增添自己的身份，然后重播此请求。为了限制网络中的洪泛数量，每个节点仅广播它收到的第一个 RREQ，删除其他与此要求一样的副本。D 收到这个 RREQ 后，它会生成一个路由答复（RREP），将其发送回节点 S。基于 RREP 消息，源节点 S 选择出最优路径，此路径是具有最少跳数的路径或者与收到的第一个回复消息相关的路径。

遗憾的是，DSR 协议很容易受到攻击。例如，听到 RREQ 数据包的攻击者可能将此包单跳转发给靠近目的地的第二个攻击者。第二个攻击者使用重放攻击，即重播 RREQ。根据 DSR 规则，第二个攻击者的临近节点在收到 RREQ 后，会丢弃随后收到的通过多跳路径转发的合法请求。这样的攻击事实上就是虫洞攻击，使数据包（要传递给基站的）在两个恶意节点之间传送。攻击者可以在此"快捷"路径上对数据包为所欲为。这样的虫洞攻击消除了发现超过两跳合法路径的可能性，因为攻击者通常使用单跳的高质量链路。 336

两个恶意节点构建一个虫洞路由的另一种方法不是由它们自己建立一个单跳路径，而是，他们可能仅是发现了两者之间的最短路径然后利用此路径，这也可能是多跳路径。路径建立过程如图 11-1 所示。节点 A 和 Z 尝试发现两者之间的最短路径，在它们周围存在两个恶意节点 X 和 Y。节点 A 广播一个 RREQ 后，节点 X 得到此请求并将其封装在一个数据包中通过与节点 Y 之间的路径（6－7－8－9）发送给 Y。节点 Y 打开此数据包，并进行重播，重播后的数据包到达节点 Z。这样节点 X 和 Y 成功将自己加入到节点 A 和 Z 的路径中。任何确定最短路径为"好"路径的路由协议都容易受到此类攻击。

另外，在上述例子中，两个恶意节点（X 和 Y）并不需要任何加密方案，也不需要具备特殊功能，例如高速有线链路或者高功率信息源。因此，此类虫洞攻击是很容易发动的。

2. 基于带外信道方式的虫洞攻击

在这种类型的虫洞攻击中，攻击者在恶意节点之间建立一个带外高带宽信道。这种高带宽信道可能是一个远距离定向无线链路或者是有线链路。由于这样的攻击需要特定和专门的硬件，相比于基于报文封装方式的攻击，此类攻击比较难以启动。

图 11-2 展示了这种攻击。节点 A 向节点 Z 发送了一条 RREQ，节点 X 和 Y 是处于它们两者之间的具有带外信道的两个恶意节点。节点 X 将 RREQ 发送给 Y，节点 Y 是 Z 的相邻节点。节



图 11-1　基于报文封装方式的虫洞攻击

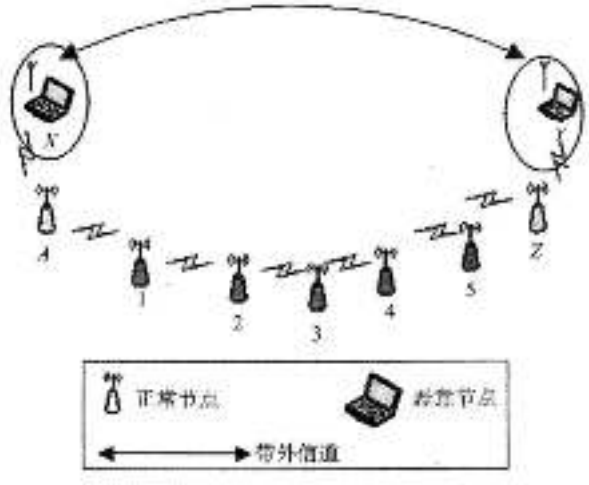


图 11-2　基于带外信道方式的虫洞攻击

点 Y 将数据包广播到其相邻节点，包括节点 Z。节点 Z 收到两个 RREQ—A－X－Y－B 和 A-1－2－3－4－5－Z－Y。第一条路径比第二条更短、更快。节点 Z 会选择第一条路径，这会导致节点 X 和 Y 在节点 A 和 Z 路径上发动虫洞攻击。

3. 基于大功率传输方式的虫洞攻击

在这种情况下，当一个恶意节点收到一条 RREQ 后，它通过一个更高的功率水平重播该请求，有些节点的天线并不能接收到此请求。所有收到此高功率广播的节点将此请求重播给目的地。因此恶意节点可以很轻易地将自己设置在源与目的地之间的路径上，甚至不需要第二个恶意节点的加入。

一种减轻此类攻击的方式是要求每一个节点准确地确定接受信号的强度，并使用无线传播模型推导出距离。我们知道，距离越长，接收信号强度（RSS）越弱。因此每个节点都可以确定所接收到的信号是否在适当的功率阈值内。利用此方案，可以很容易地监测到使用高功率的恶意节点，因为正常节点不会有那么高的功率。

4. 基于报文转发方式的虫洞攻击

在这种攻击类型中，一个恶意节点协助在两个相隔距离很远的节点之间发送数据包（比如节点 A 和 B，它们之间间隔多跳），从而使节点 A 和 B 相信它们是单跳邻居。这种攻击可以由一个恶意节点单独发动，也可以通过与其他大量恶意节点的合作，扩大被攻击节点的临近节点的数量，达到几跳的范围，从而实现攻击。

337
|
338

5. 基于协议偏离方式的虫洞攻击

这种类型的虫洞攻击尝试违反一些路由协议的规则。例如，ARAN［KSanzgiri02］路由协议选择具有最短延迟时间的路径为最优路径，而不会选择跳数最少的路径作为最优路径。因此，攻击者通过缩短它的路由搜索延迟，使它的节点相比于其他节点更具吸引力。

攻击者是如何缩短其路由延迟时间的呢？在 ARAN 路由规则中，正常节点在转发 RREQ 前会随机等待一段时间。这是因为无线链路为共享广播介质——多节点同时收到上游节点转发 RREQ，之后随机等待一段时间再重新转发 RREQ，以便有效避免传输冲突。然而，恶意节点不会遵循这些规则。它可以在没有任何延迟的情况下，通过广播 RREQ 创建一个虫洞。通过这样方式，攻击者的 RREQ 数据包会首先到达目的地。这种路径延迟时间看起来比同团的正常节点的延迟时间要少。因此，恶意节点就很可能成为源与目的地之间路径上的中间节点。

上述情况实际上是［YCHu03］描述的急送攻击（rushing attack）的一种特殊形式。

表 11-1 总结了虫洞攻击的不同模式以及附加需要［Issa06］。

表 11-1　虫洞攻击方式总结

名称	最少需要的恶意节点数量	特殊要求
报文封装	2	无
带外信道	2	带外链路
大功率传输	1	大功率通信模块
报文转发	1	无
协议偏离	1	无

虫洞防御机制——LITEWORP

文献［Issa06］提出了一个虫洞检测和抵御方案，名为 LITEWORP，其基本思路是隔离恶意节点。任何安全方案都有一些假设，LITEWORP 也做了一些假设以保证其有效运行：

1）通信链路是双向的，也就是说，如果节点 A 可以发送数据包到 B，则 B 也可发送至 A。

2）破坏一个节点只能在节点部署完成之后，并且破坏过程需要占用一段时间。在邻居发现过程完成之前，是不存在任何内部或外部恶意节点的。然而，如果采用安全的邻居发现协议（例如 Hu 和 Evans 使用的定向天线［LHu04］或者文献［YTuts06］中设计的更强大的可信节点），这个假设可以忽略。 339

3）无线传感器网络中节点是固定的（这对于大多数无线传感器网络来说是一个合理假设）。然而，由于节点能量耗尽、节点故障、恶意节点骗离、路由缓存中的路由驱逐或者节点作用的变化（例如，簇首节点、数据聚合等），网络的拓扑结构可能会发生变化。

4）每一个数据包的转发者需要明确在数据包的来源，即节点是从哪里收到这个数据包的。

5）能够采用一个密钥管理协议（例如 SECOS［IKhalil05］）以对网络节点进行对称密钥的预分配。

建立邻居列表

LITEWORP 首先提出了一个邻居节点发现协议，目的在于建立单跳邻居节点和相邻节点的数据结构。一个邻居节点是传输范围内的任一节点。这样的数据结构对于检测恶意节点、发送本地响应报文以对检测到的恶意节点进行隔离是很重要的。

HELLO 消息是发现邻居的一种常用方法。在一个节点（例如 A）部署到网络中后，它立即广播一个一跳距离的 HELLO 消息。任何节点（例如 B）在听到这个 HELLO 消息后，向 A 发送一个回复。节点 A 接收在预先设定的时间间隔内收到的所有回复。

通过收集这些回复，节点 A 在它的邻居列表中添加响应者。邻居发现还没有结束。节点 A 会向所有一跳距离节点广播此列表。当任何邻居（例如节点 B）监听到此列表时，会存储此列表。

在完成上述邻居发现过程后，每个节点都有其直接邻居的列表和其每一个直接邻居的邻居列表。然而，每个节点仅执行一次上述过程，而且假设此过程是安全的（可以利用安全邻居发现协议完成）。

注意：在每一个节点上建立这样一个列表之后，节点只会向其直接邻居节点发送数据包。此外，两跳邻居节点信息被用来决定一个转发数据包是否来自转发者的邻居。例如，节点 C 收到节点 B 转发的数据包，而且发现此数据包来自上一跳节点 A。节点 C 会查找其邻居列表，如果发现节点 A 不是其两跳邻居节点则丢弃此数据包。

在建立其单跳和两跳邻居节点列表后，节点 A 可以激活本地监控程序，找出虫洞攻击者。

在这里，我们将展示如何使用本地监控建立监测算法监测虫洞攻击的前四个节点，同时也展示现有监测方式如何用来监测第五个节点。 340

（1）带外信道和报文封装方式的虫洞攻击检测

LITEWORP 引入了哨兵（守护）节点的概念。假设 α 是另一个节点 A 的守护节点。α 通过以下步骤，可以监视从节点 X 到节点 A 的无线链路，它的作用是监控传感器网络通信：

1）我们要求守护节点 α 存储从节点 X 到节点 A 通信链路上的每个控制数据包的报头信息，然后标记上截止时间 τ。

2）节点 α 窃听到从节点 A 发送的每个数据包。节点 A 声称每个数据包都来自节点 X，节点 α 在其拥有邻居列表的表缓存中查询相应条目。

3）如果发现存在相应条目，节点 α 将其丢弃，因为假设正常转发已经完成。

4）如果没有发现相应的条目，节点 A 被认为伪造数据包。因此，α 将恶意节点数量 $MalC(\alpha, A)$ 增加 V_f（V_f 为节点 A 伪造数据包的数量）。

5）如果从节点 X 发送到节点 A 的数据包的一个条目停留在表缓存区 τ 之外，那么节点 A 被指控丢失相应数据包。节点 α 将恶意节点数目 $MalC(\alpha, A)$ 增加 V_f（V_f 为节点 A 丢弃数据包的数量）。

6）如果传入节点 A 的数据包与相应的从节点 A 输出的值不同，那么节点 A 被指控篡改数据包。因此，α 将恶意节点数目 $MalC(\alpha, A)$ 增加 V_m（V_m 为节点 A 篡改数据包的数量）。

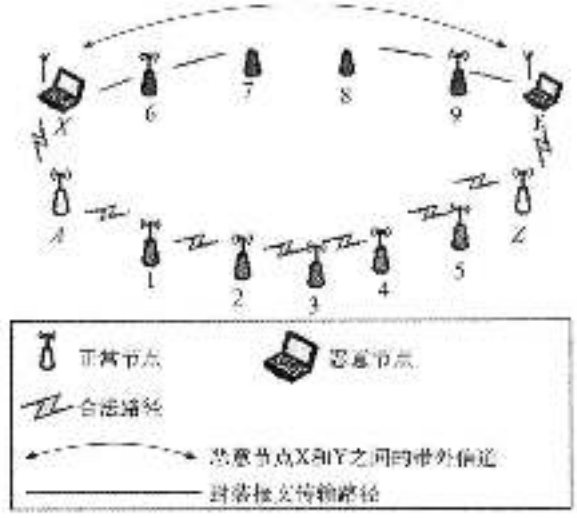


图 11-3　带外信道和报文封装方式的电洞攻击检测

让我们考虑图 11-3 中的场景。节点 X 和 Y 是两个恶意节点，希望在两个正常节点之间建立一个虫洞（源节点为 A，目的节点为 Z）。当节点 X 监听到发送自节点 A 的 RREQ 数据包时，它将此数据包引导至节点 Y。节点 Y 添加上其获得 RREQ 的上一跳节点身份后，重播此 RREQ 数据包。节点 Y 对于上一跳节点有两个选择，要么追加节点 X 的身份，要么追加 Y 的某个邻居的身份，比如 9。

若选择第一种方式，节点 Y 的所有邻居会拒绝 RREQ，因为它们从其存储的两跳邻居数据结构可知，节点 X 并不是节点 Y 的邻居。

在第二种情况下，单跳和二跳邻居列表的信息并不足以让所有守护节点检测到攻击。然而，通过使用本地监控，节点 X 到 Y 的链路上的守护节点可以检测到节点 Y 伪造路径请求，因为它们的表缓存中不具有来自节点 X 的相应的数据包的信息。

在这两种情况下，恶意节点 Y 都会被检测到，并且守护节点会增加节点 Y 的 $MalC$ 值。

LITEWORP 也可以利用路由应答（RREP）报文检测节点 X 和 Y 的行为。当目的节点 Z 收到 REEQ 后，它生成一个 RREP 包，将其发送回节点 X。节点 Z 到 Y 的通信链路上的守护节点可以监听到 RREP，在它们的表缓存中保存一个条目。节点 Y 使用带外信道或者报文封装将路径回复发送回节点 X。经过时间 τ 后，守护节点中的表缓存定时器停止，因此守护节点监测到节点 Y 丢弃 RREP 数据包，增加节点 Y 的 $MalC$。但是，如果节点 Y 更聪明一些，它可以通过一个较慢的正常路径转发 RREP 的一个副本。在这种情况下，节点 Y 的 $MalC$ 不会被增加。当节点 X 从节点 Y 获得 RREP 后，节点 X 添加上一跳身份，将其转发回 A。

像前面一样，节点 X 有两个选择——或者添加节点 Y 的身份信息，或者添加 Y 的某个邻居的身份信息，比如 6。选择第一个方案的话，节点 A 拒绝 RREP，因为它知道节点 Y 不是节点 X 的邻居。同样，节点 X 的所有邻居知道节点 Y 不是 X 的邻居。在第二种情况下，从节点 6 到 X

的通信链路上所有守护节点监测到节点 X 伪造 RREP，因为在它们的表缓存中没有来自节点 6 的相应条目。

（2）大功率传输方式的虫洞攻击检测

传感器网络可以通过使用对称双向信道检测到此类攻击。如果一个恶意节点 X 试图使用高功率传输转发数据包 P，到其目的节点，或者将其加入到最短多跳路径中，所有的没有将节点 X 列为邻居节点的节点会意识到这是欺骗数据包，然后丢弃此包。

341
-
342

（3）报文转发方式的虫洞攻击检测

传感器网络可以很容易地通过每个节点上存储的邻居列表检测到此类攻击。假设一个恶意节点 X 是两个非邻居节点 A 和 B 的邻居。如果节点 X 试图通过转发两个正常节点之间的数据包欺骗它们，节点 A 和 B 可以监测出恶意节点 X 的恶意行为并拒绝转发的数据包，因为节点 A 和 B 知道它们并不是彼此的邻居节点。

（4）协议偏离方式的虫洞攻击检测

LITEWORP 无法监测到此类攻击。然而，我们可以在某个协议中使用其他研究者的方案抵御自私行为。在这里，“自私”（也称为贪婪）指的是节点企图否认要求与其他节点的合作服务，达到节省自己资源（如电池电量）的目的。

MAC 层的贪婪问题已被 Kyasanur 等人解决［Kyasanur03］。路径数据包转发的自私问题已被［SCapkun03］解决。针对此攻击的一种解决方法叫做急送攻击（rushing attack）［YCHu03］。在这种方法中，节点可以快速转发信息并不需要等待协议规定的退避时间。

（5）响应与孤立算法

以上方案仅仅解决了虫洞攻击的检测问题。下一步是使用本地响应和隔离模块诊断攻击者，做出合适的回应将攻击者在网络中孤立，从而消除其损害网络其他部分的能力。LITEWORP 提出了一个攻击者隔离方案，此方案受本地监控模块控制，只有在检测到恶意节点时该算法被激活。

LITEWORP 使用本地响应方案在本地传播监测信息，这里的本地是指可疑节点的两跳之内。通过从所有单跳和两跳邻居的列表中删除可疑节点完成本地响应。

以下是 LITEWORP 的本地响应算法。当一个引导节点 α 在本地监控时，检测到一个节点 A 的恶意行为时，该算法被激活：

1）当声誉值 $MalC$（α，A）超出阈值 Ω 时，守护节点 α 将节点 A 在邻居列表中撤出，然后广播给节点 A 的每一个邻居节点，比如 D，并发送一个认证警报信息说明节点 A 是可疑的恶意节点。

注意：要永久隔离恶意节点，我们可以在节点间使用共享安全密钥，以便在将来出现不实指控时验证节点。在下面的内容中，将会介绍更多基于密钥管理无线传感器网络安全的知识。另外，如果网络中所有节点的时钟松散同步，α 可以像 TESLA［APerrig02］那样认证本地两跳多播，或者像 μTESLA［APerrig022］那样通知节点 A 的邻居节点。注意，α 隔离节点 A，并没有等待来自其他节点的警报 γ，因为假设节点是信任自身的。

343

2）当节点 D 得到警报信息，它验证该信息的真实性，并将 α 的身份存储在与节点 A 相关的警报缓存区中。

3）当节点 D 获取了足够的关于节点 A 的警报信息后（我们可以定义一个警报信息的阈值），它会通过在所有邻居列表上停用该节点来孤立该节点。

4）隔离后，D 不再向节点 A 发送数据包，也不接收节点 A 发送的数据包。

上面的方法可以从网络中删除恶意节点。此外，它缩短了检测与响应之间的时间，因为信息可以在本地处理。它不会导致大量的网络流量，因为它仅向节点 A 的邻居节点发送消息（仅

在监测阶段）。每条信息的转跳数最多为两跳。

LITEWORP 还定义了一个有用概念，叫做**检测置信度**（detection confidence），用 γ 表示。它可以用来有效降低攻击者利用高优先级数值来诬陷合法节点的可能性。诬陷（framing）是一种攻击，恶意节点通过扮演守护节点，开始发送一个关于正常节点的虚假指控。如果 γ 设置为无穷大，那么一个节点仅信任自己，则不会遭受此类诬陷攻击。



奇思妙想

从 LITEWORP 中我们可以学到很多关于无线传感器网络安全的优秀理念。通过保留一个可靠的邻居列表，我们可以检测到任何试图加入路由过程中的“坏”节点。在我们找到这些“破坏分子”后，我们要把它们送进“监狱”，即我们必须将它们隔离出正常通信。

充分发挥你的想象力吧！

11.3 无线传感器网络安全示例：基于 Blom 模型的方法

正如前面提到的，安全密钥可被用来实现身份认证（即验证源）和保密性（即加密信息）。然而，密钥管理对于无线传感器网络来说是一个挑战，因为我们需要处理密钥预分配问题，即为了实现安全目的，我们应如何在不同的传感器节点中预分配密钥？Du 等人提出了一个基于 Blom 模型［Blom85，Blundo93］的无线传感器网络密钥预分配方案［DuW05］。

假设 N 是无线传感器网络中的节点总数。如果在任何两个节点之间存在一个安全通信，这两个节点为了加密和解密消息，需要共享一个密钥。如果不使用任何巧妙的密钥预分配方案，
344 为了保证任何两个节点可以共享至少一个密钥，则每个节点需要存储（N-1）个密钥。

在 Blom 模型的密钥预分配方案中，节点仅需要存储（$\lambda + 1$）个密钥，其中 $\lambda \ll N$。显然，Blom 模型在节点捕获方面并不具有足够的灵活性，因为我们不能确保任意两个节点共享一个密钥。然而，在现实中，如果两个节点并不相邻，我们并不需要保证这两个节点共享一个密钥，两者并不需要通信。

事实上，Blom 方案可以确保 λ-secure 属性，即只要攻击者破坏掉不多于 λ 个节点，正常节点之间的通信链接仍然安全。当然，如果攻击者破坏了多于 λ 个数量的节点，整个网络密钥都将被破坏。

阈值 λ 是一个重要安全参数。通过选择一个更大的阈值 λ，密钥共享的概率增加，从而获得更好的安全性能。因此，通过设置一个大的 λ 阈值，可以迫使攻击者去捕获相当比例的网络节点以达到破坏整个无线传感器网络通信的目的。另一方面，增加 λ 值会需要更大的存储空间以存储大量的密钥信息。

Du 所提出的密钥预分配方案［DuW05］是对 Blom 模型的改进。前者使用概率方法能提升网络抵御节点捕获的适应性。不同于 Blom 模型，它并不需要太多的额外内存。

Blom 模型使用单密钥空间确保任意一对节点可以计算出一个共享密钥，而 Du 提出了一个使用多密钥空间的新方案。该方案首先使用 Blom 模型创建总的 ω 个空间（其中 $\omega > 2$），然后它要求每个传感器节点加载从密钥空间中随机选出的 τ 个空间中的密钥信息（其中 $2 \leq \tau < \omega$）。Blom 模型告诉我们，只要两个节点从一个共同空间加载密钥信息，这两个节点就可以计算出一个共享密钥。

虽然只有一个关于两个节点可以生成一个共享密钥的概率性保证，但 Du 通过分析表明，当使用相同数量的内存时，这种新方案相比于传统的概率密钥预分配方案具有更大的灵活性。



奇思妙想

许多学生和研究人员都在问同一个问题：我该如何提出一个好的方案，解决一个具有挑战性的问题呢？我们可以从文献［DuW05］学到一些方法：Blom 模型的算法是在二十多年前提出的。它“湮没”在成千上万的在 IEEE、ACM、Elsevier 上发表的文献中。可能并没有提出解决一个新问题的直接方法。然而，通过大量阅读传统的有关密码学的文献，不断地问自己：“尽管这篇文章并不是关于无线传感器网络安全的，我可以从它这里借来一些想法吗？我可以做一些延伸或者改进，将其应用到资源有限的无线传感器网络中吗？”总是问自己上述问题，总有一天，你会说：“嗯，我可以用这个想法！”

345

为了理解 Du 的方案，我们先简要回顾一下 Blom 方案（Du 方案为了更好地应用于资源非常有限的无线传感器网络中，它在 Blom 方案的基础上做了一些细微调整，但 Blom 方案的主要框架不变）。

假设存在一个商定的矩阵 G，大小为 $(\lambda + 1) \times N$，其位于一个有限域 $GF(q)$ 中（其中 $q > N$）。注意，矩阵 G 不是保密的。攻击者也可能知道这个矩阵。

在密钥生成阶段，无线传感器网络基站在 $GF(q)$ 上随机生成了一个 $(\lambda + 1) \times (\lambda + 1)$ 的对称矩阵 D，并计算出一个 $N \times (\lambda + 1)$ 的矩阵 $A = (D \cdot G)^T$，其中 $(D \cdot G)^T$ 是 $D \cdot G$ 的转置。

注意：矩阵 D 必须保密，不应对攻击者或任何传感器节点公开。另一方面，我们将在随后讨论，$(D \cdot G)^T$ 的一行应向每一个传感器节点公开。因为 D 是对称的，可以很容易知道：

$$A \cdot G = (D \cdot G)^T \cdot G = G^T \cdot D^T \cdot G = G^T \cdot D \cdot G = (A \cdot G)^T \qquad (11.1)$$

因此，$A \cdot G$ 是一个对称矩阵。如果我们让 $K = A \cdot G$，我们知道 $K_{ij} = K_{ji}$，其中 K_{ij} 是 K 中第 i 行第 j 列的元素。我们的想法是使用 K_{ij}（or K_{ji}）作为节点 i 与节点 j 之间的共享密钥。共享密钥 $K_{ij} = K_{ji}$ 的生成过程如图 11-4 所示。为了进行上述计算，节点 i 和 j 应能分别计算出 K_{ij} 和 K_{ji}。这个过程可以通过下述密钥预分配步骤达成，对于 $k = 1, \ldots, N$，有 1）节点 K 存储 A 矩阵中的第 k 行信息；2）节点 K 存储 G 矩阵中第 k 列信息。随后我们将说明一个节点并不需要存储整列信息，因为每一列都可从一个单一域元素中生成。

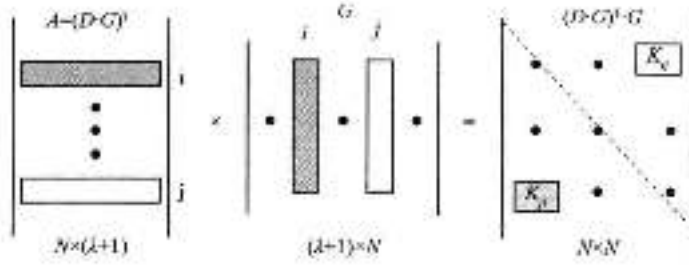


图 11-4　Blom 方案中的密钥生成

346

然后节点 i 和 j 可以根据以下步骤生成一个共享密钥（也称作对偶密钥）：它们首先交换其在矩阵 G 中的列，然后使用矩阵 A 中其私有的行分别计算出 K_{ij} 和 K_{ji}。在前面已经提到，矩阵 G 是对外公开的，它的列信息可以以明文传输。已有研究证明上述方案是 λ - 安全的［Blom85］，前提是 G 矩阵中任何 $\lambda + 1$ 列是线性独立的。这种 λ - 安全特性确保了只要参与共谋的恶意节点数不超过 λ（不包括节点 i 和 j），攻击者就无法获得关于 K_{ij} 或 K_{ji} 的任何信息。

Du［DuW05］展示了一个矩阵 G 的例子。矩阵 G 中任何 $\lambda + 1$ 列都必须是线性独立的。

由于每对共享密钥是由有限域 $GF(q)$ 中的一个元素表示的，我们必须把 q 设置为大于我们所需要的密钥大小。因此，如果想生成一个 64 位密钥，可以选择 q 为大于 2^{64} 的最小素数（或者，可以简单地设置 $q = 2^{64}$）。

假设 s 是 $GF(q)$ 的一个本原元素，即在 $GF(q)$ 中的每一个非零元素可以表示为 s^i。可以生成一个如下所示的可行 G 格式［MacWilliams77］：

$$G = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ s & s^2 & s^3 & \cdots & s^{\lambda} \\ s^2 & (s^2)^2 & (s^3)^2 & \cdots & (s^{\lambda})^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s^{\lambda} & (s^2)^{\lambda} & (s^3)^{\lambda} & \cdots & (s^{\lambda})^{\lambda} \end{bmatrix} \tag{11.2}$$

由于 s 是本原元素，只要 $i = (j \bmod q)$，就能得到 $s^i = s^j$。可以看到，矩阵 G 中任何 $\lambda + 1$ 列都是线性独立的［MacWilliams77］。

因为矩阵 G 具有一个很好的特性，即它的列都可以由本原元素 s 的指数幂生成，为了在节点 k 上存储矩阵 G 的第 k 列信息，我们仅需要在此节点上存储种子 s^k。矩阵 G 的列可以在需要时重新生成。

文献［DuW05］还提供了有趣的理论分析和详细的实验结果。这些结果清楚地表明基于 Blom 方案的扩展方案具有低内存开销和良好的安全性能。

11.4　广播认证：基于时间的高效的容忍丢包的流认证协议 μTESLA

347

本节将讨论安全领域的另一个重要问题：来源认证。我们将着重介绍广播认证，因为在无线传感器网络中，基站会经常广播一个命令消息（例如，请报告某区域的传感器采集值）。任何一个收到此命令的传感器节点需要证实此消息来源，因为发送此指令的可能是正常基站，也可能是攻击者。

用来认证广播消息的传统方法并不适用于传感器网络，因为它们大部分都是依赖于非对称数字签名进行认证。非对称数字签名要求通信双方分别具有公钥和私钥。源节点可以利用其私钥对信息进行加密。任何一个拥有源节点公钥的节点都可以将信息解密。但是，如果信息来自于一个攻击者，攻击者并不拥有正确的公钥，那么它发出的信息并不能被接收节点解密，那么数字签名就失效了。

虽然非对称数字签名可以认证消息，但它们需要公钥/私钥，所以相对于对称密钥（在两个节点间仅需要少量密钥），它们的内存存储开销更大。而在传感器网络中，节点的内存有限，所以非对称认证是不适用的。

TESLA 协议［Aperrig00］是一种非对称机制，它提供了一种有效的广播认证方法。然而，TESLA 协议为了生成一个数字签名密钥，需要大约 24 个字节的数据包，这超过了普通无线传感器网络的可用资源开销。事实上，大多数无线传感器网络中的每条消息大概需要 30 个字节。因此，在每个数据包中公开一个 64 位（相当于 8 个字节）的密钥和 MAC（消息认证代码）会占用数据包的 50% 以上的开销。基于这些情况，纯 TESLA 对于节点的广播系统是不适用的。

因此文献［Aperrig01］提出了一个解决方案 μTESLA，该方案用来克服传感器网络中TESLA的不足。μTESLA 面临困难的问题是：为了实现强有力的消息认证，不对称机制的性能要优于对称机制。这是因为以下事实：如果我们仅仅使用对称机制的性能（即发送者和接受者两者使用同一密钥），一个被攻击的接收者可以获得这个密钥，然后轻易地伪造来自发送者的消息。

μTESLA 通过延迟发布对称密钥消息的方式解决了 TESLA 存在的极高计算量、通信量和存

信息的问题。其基本思想是：当一个无线传感器网络的基站发送一个数据包时，它计算该数据包中的 MAC，但并没有公开 MAC 密钥。节点收到的数据包被缓存在节点内存中，直到相应的 MAC 密钥被基站公开。因为只有基站知道密钥，所有的传感器节点可以检查数据包在传输过程中是否被攻击者修改。随后，节点收到公开的 MAC 密钥，认证已被缓存了一段时间的数据包。 [348]


奇思妙想

"使用对称安全机制实现非对称认证"，这就是 μTESLA 的主要思想。非对称机制意味着每一个 MAC 仅使用一个密钥。然而，对称机制需要两个密钥（公钥/私钥）。μTESLA 仅使用一个 MAC 密钥。但是发送者（基站）在发送 MAC 消息时，并不将 MAC 密钥公开给发送者。相反，发送者等待一段时间（这个延迟时间大于消息在基站和传感器之间的最大往返延迟）后，才公开先前的 MAC 密钥。因此，这样的延迟实现了"不对称"的效果。

μTESLA 采用了一个众所周知的单向函数来生成 MAC 密钥，而每一个 MAC 密钥都是密钥链上的一部分。发送者在密钥链上随机选择最后一个密钥（K_n），能够反复应用 F 计算其他所有密钥：$K_i = F(K_{i-1})$。假设最后一个密钥是 K_{100}。它能够根据以下公式计算出其他所有密钥：

$$K_{99} = F(K_{100}),\quad K_{98} = F(K_{99}),\ \cdots,\ K_0 = F(K_1)$$

由于 $F(\cdot)$ 是一个单向函数，给定 K_{100}，我们可以很容易地计算出 K_{99}，K_{98}，…，K_0。但是，如果给定 K_0，我们不能得出 K_1，K_2，…K_{100}。

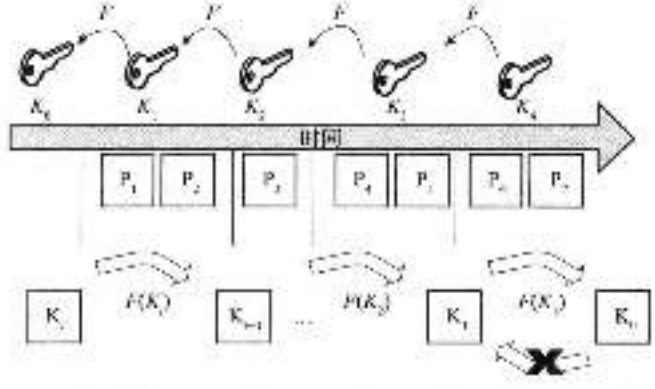


图 11-5　μTESLA 单向密钥链。发送节点通过应用单向函数 F 依次迭代生成单向密钥链的每个密钥（自右向左）。发送节点为单向密钥链上的每个密钥分配一个工作时间间隔（按时间先后自左向右），在每个工作时间间隔内，使用相应的密钥。这样，发送节点反向依次使用单向密钥链的密钥（K_1，K_2，…，K_n）计算数据包的 MAC

图 11-5 展示了 μTESLA 中的单向密钥链概念。它具备以下特点： [349]

1）μTESLA 假定整个无线传感器网络的所有节点中都有某种类型的松散时间同步协议。因此，所有的节点可以分辨出不同发送时间间隔。

2）当基站发送消息（数据包）时，它使用同一个密钥在一个时间间隔内认证所有发送的数据包。

3）接收者知道 K_0（密钥链的验证凭证）。

在图 11-5 中，数据包 P_1 和 P_2 发送时间间隔为 1，在密钥 K_1 中包含一个 MAC（注意：如果没有密钥 K_1，接收者就没有途径验证 MAC 是否源自一个正确的基站）。数据包 P_1 有一个使用密钥 K_2 的 MAC，到目前为止，接收者无法认证任何数据包，因为基站在那个时间间隔中不能公开每个 MAC 相应的密钥（直到某些时间间隔后才能公开）。

请注意 μTESLA 单向密钥链的一个优点：它可以容忍之前 MAC 密钥的丢失。假设在一些时间间隔之后，由于无线信道的不可靠，密钥 K_1（用于验证数据包 P_1 和 P_2）没有被节点接收到。然而，只要节点随后得到密钥 K_2，它就能验证 $K_0 = F(F(K_2))$，然后得知 $K_1 = F(K_2)$。所以它可仍然验证之前收到的所有数据包。


案例研究

消息认证码（Message Authenticated Code，MAC）：一个消息认证码的例子是用密钥加密的哈希函数。哈希函数可以将任意的消息映射到一个固定长度的消息上。哈希函数实际上是一个单向函数，因为根据一个散列结果，你不能推断出原始消息。如果我们使用密钥对此散列结果进行加密，就可以得到 MAC。通常情况下，发送者发送消息（消息 M，MAC）到接收者。接收者可以使用相同密钥解密 MAC，然后同 M 比较结果。如果它们是一样的，我们可以确定 M 确实是从正确的源发送出来的。

μTESLA 协议的实现

μTESLA 由一系列的运行步骤组成，包括建立发送者、发送认证包、安全引导一个新接收者，认证发包信息。

建立发送者：在这个阶段，发送者（基站）建立了一个密钥链。此密钥链长度为 n，发送者通过随机选择最后一个密钥 K_n 生成密钥链，然后利用单向函数 F 生成其他值。单向函数的一个例子是一个加密的散列函数，例如 MD5；$K_i = F(K_{i+1})$。之前提到过，函数 F 的单向特性意味着密钥可以向前计算但不可倒推。

广播认证数据包：如图 11-5 所示，时间被分割成时间间隔。单向密钥链上的每一个密钥都与一个时间间隔相关联。对于每一个时间间隔，发送者使用那个间隔上的密钥计算此间隔上数据包的 MAC（消息认证码）。发送者在时间间隔后的一个预先设置的延迟后，公开此时间间
350 隔的密钥。


实例研究

现在的问题是：在每一个时间间隔内，基站需要等待多长时间才能公开密钥呢？假设基站在时间间隔 76 使用密钥 K_{76}。当然，基站不能在时间间隔 76 期间公开密钥 K_{76}，因为如果那样做，攻击者可以立刻获取密钥 K_{76}。只要攻击者知道了 K_{76}，它就可以利用此密钥制造 MAC。可以用这样的一个 MAC 来广播命令消息。于是，攻击者可以向节点伪造任何命令消息。

因此基站要等待一段时间后再公开密钥。那它应该等待时间间隔 77、78 后还是其他时间段后公开密钥 K_{76} 呢？μTESLA 的解决方法是：设定的延迟是几个时间间隔，这个时间必须要大于任何合理的发送者（即基站）与接收者（即节点）之间的往返时间（RTT）。


奇思妙想

为什么基站需要至少等待一个 RTT 的时间以后，才能公开 MAC 密钥呢？答案很简单：我们不能给攻击者任何机会去接收相应 MAC 密钥并伪造一个命令消息。如果我们等待一个 RTT 的最大值（可以从经验数据中得知），对于攻击者来说，要伪造一个命令消息就为时过晚了，因为所有的节点都已经获得正确的时间间隔密钥了。

安全引导新接收者：正如之前提到的，每一个节点仅需知道密钥 K_0，它是密钥链上的最后一个密钥。我们称 K_0 为验证凭证。基于密钥链的单向特性，很明显，一个节点通过应用单向哈希函数可以验证其收到的 MAC 密钥 K_i 是否是正确的，该单向哈希函数如下所示：

$$F(\cdots(F(F(K_i)))) = K_0$$

如果它不等于 K_0，我们就能知道这个密钥并不属于正确的密钥链。

向每个节点分配验证凭证 K_0 的步骤叫做安全引导（bootstrap）。我们看到，在 μTESLA 中，通过确认接收者具有单向密钥链上的一个真实密钥作为验证凭证，可以很容易对一个新的节点进行安全引导。

松散的时间同步对于 μTESLA 的正确运行也是很重要的，因为接收者可以知道每一个时间间隔的开始。

上面提到的两个要求，即节点中的松散时间同步和认证密钥链验证凭证，可以与某个机制配合使用，该机制确保新鲜度（即验证消息是新的，而不是一个攻击者重播的消息）和点到点认证（即验证消息源是一个正常基站，而不是一个攻击者）。

为了保证 μTESLA 的正确运行，基站需要在保证安全的情况下让节点知道以下参数：当前时间 T_S（用于时间同步）、过去的时间间隔 i 中使用的单向密钥链上的密钥 K_i（在 RTT 时间后公开）、时间间隔 i 的起始时间 T_i、时间间隔的长度 T_{int} 和公开延迟 δ。我们可以使用以下通信来进行安全的参数传输：

节点→基站：Nonce

基站→节点：$T_S \mid K_i \mid T_i \mid T_{int} \mid \delta$, MAC $(K_{00}, N_0 \mid T_S \mid K_i \mid T_i \mid T_{int} \mid \delta)$

注意：我们在上述通信中使用“nonce”（即在整个会话中仅使用一次的随机数字），是为了保证每一条传输消息都是“新鲜的”而不会被重播的。也要注意，基站不需要加密消息，因为此系统不需要保密性。MAC 使用基站和节点的共享密钥认证数据。

认证广播数据包：如果在之前的时间间隔中，一个节点收到用于 MAC 的密钥 K_j，它可以通过检查此密钥是否与上一个它所知道的真实密钥验证（K_i）相匹配，利用 F 的单向函数 $K_i = F_{j-i}(K_j)$，从而验证密钥 K_j 的正确性。如果验证成功，则新的密钥 K_j 是真实的，节点可以验证所有在时间间隔 i 到 j 发送的数据包。接收者为了下一次检查，还可以用密钥 K_j 取代存储的密钥 K_i。

11.5　面向传感器节点的实用安全机制

在这一节，我们将介绍一些实际应用于传感器硬件的安全机制。我们将特别讨论数据链接层安全，因为此类安全对于实现传感器节点之间的安全有着重要的作用。

11.5.1　TinySec

在传统网络中（如互联网），信息安全（包括真实性、完整性和保密性）通常是通过一个端到端的安全机制实现的，如 SSH［TYlonen96］、SSL［SSL］或者 IPSec［IPSec］。这是因为互

联网大多采用端到端的通信。发送者和接收者之间的路由器只需要查看消息头，而不需要访问
352 消息正文。

然而，无线传感器网络大多采用一对多（一个基站对多个传感器节点）或者多对一（多个传感器节点对一个基站）的通信模式。此外，在环境监测类应用中的无线传感器网络一般拥有大量的节点。因此，WSN 中的相邻节点经常会接收到相同或者相关的环境事件。如果每个节点分别将数据包发送到基站，将会浪费和占用大量的能量和带宽。为了避免发送冗余消息，无线传感器网络使用网络内处理（例如数据聚合）来消除重复数据［Samuel02］。

因为网络内部处理要求中间的传感器节点能看消息内容（或者执行其他程序），所以端到端的安全机制可能并没有逐跳的安全机制（即数据链路层）那么重要。事实上，如果我们仅仅使用端到端的安全机制，所有消息的完整性只能被最终目的节点检查，那么我们就不能检测每个传感器遭受的网络攻击情况。例如，攻击者可能在处于中间位置的节点插入数据包。因此，数据链路层的安全需要在未经授权的数据包第一次进入网络时检测它们。一些研究者提出了在有线网络中抵御拒绝服务（DoS）攻击的数据链路层安全机制［Mohamed02］。

TinySec 是无线传感器网络数据链路层的安全机制［Karlof04］，用以实现相邻节点之间的消息真实性、完整性和保密性，同时允许进行网络内部处理。当然，端至端的安全机制仍可应用于传感器网络中，作为 TinySec 的补充。

TinySec 协议自身通信开销不大，可以很容易地集成到其他无线传感器网络应用程序中，也能适用于各种传感器节点硬件和无线平台。如果希望了解关于 TinySec 的更多细节，请参考文献［Karlof04］。

11.5.2 MiniSec：一种面向无线传感器网络的安全通信架构

Minisec 也是一种数据链路层的安全机制［Mark07］。它比 TinySec 消耗的能量要低，但它可以实现更高水平的安全性。这是通过利用以下三种技术达成的：

1）它采用分组密码实现保密性和真实性。

2）它仅发送 IV（初始化矢量）的几个位，然而，它可以保留每个数据包中一个完整长度 IV 的安全性。相比之下，以往的方法（例如 TinySec）需要对于明文进行两轮处理（一轮用于加密，一轮用于认证）和全长的 IV 传输。

3）在广播模式中（即从基站到传感器），MiniSec 采用基于一个 Bloom 过滤器的重放保护机制以避免所有发送者的状态报告。但是，这种在能耗上的改善相应会导致内存消耗的增加。由于存储器技术发展迅速，这是传感器节点一个可以接受的折中。

353 如果想更多地了解 TinySec 和 MiniSec，请参考文献［Karlof04，Mark07］。

11.6 案例：无线传感器网络中的安全时间同步

无线传感器网络安全涉及路由层、数据链路层、硬件芯片、时间同步等很多方面。本节将介绍无线传感器网络时间同步机制所面临的安全问题及解决方案。

现有的时间同步方案（无线传感器网络或其他网络）在设计时没有考虑到安全性因素，因此容易受到恶意攻击。本节首先将集中讨论一种无线传感器网络时间同步机制所面临的不能由传统加密技术来解决的攻击——延迟攻击（delay attack），然后介绍 Hui 等人提出的**时间转换**（time transformation）和**泛化极端学生化偏差**（Generalized Extreme Studentized Deviate，GESD）两种方法［Hui07］，它们能够有效剔除网络中由延迟攻击所造成的异常数据。

前面的章节中已经讨论了无线传感器网络的时间同步。许多无线传感器网络应用要求将其

所有节点进行时间同步。这类应用包括：数据链路接入调度、μTESLA 及网络内部聚合等。所有的无线传感器网络时间同步方法都依赖于节点之间的信息交换。

当传感器网络部署在一个敌对的环境中（例如战场）时，时间同步协议对于攻击者来说是一个非常有吸引力的目标。例如，时间同步是目标追踪的前提条件，因为时间需要被准确地记录以便估计目标运行轨迹。因此，如果一个攻击者可以攻击时间同步协议，那么一个移动物体的预估方位可能会严重偏离其实际方位。

文献［Hu07］对延迟攻击定义如下：攻击者故意拖延一些时间消息，例如 RBS 方案中的信标消息，达到破坏时间同步过程的目的。图 11-6a 显示正常的没有遭受延迟攻击的 RBS 方案。图 11-6b 和 11-6c 显示了两种在 RBS 方案中实施延迟攻击的方法。在图 11-6b 中，两个相邻节点分别作为节点 A 和节点 B 的基准节点。它们在不同的时间向节点 A 和节点 B 发送基准信标 b。因此，节点 A 和节点 B 误认为它们在相同的时间收到信标，而事实上它们是在不同的时

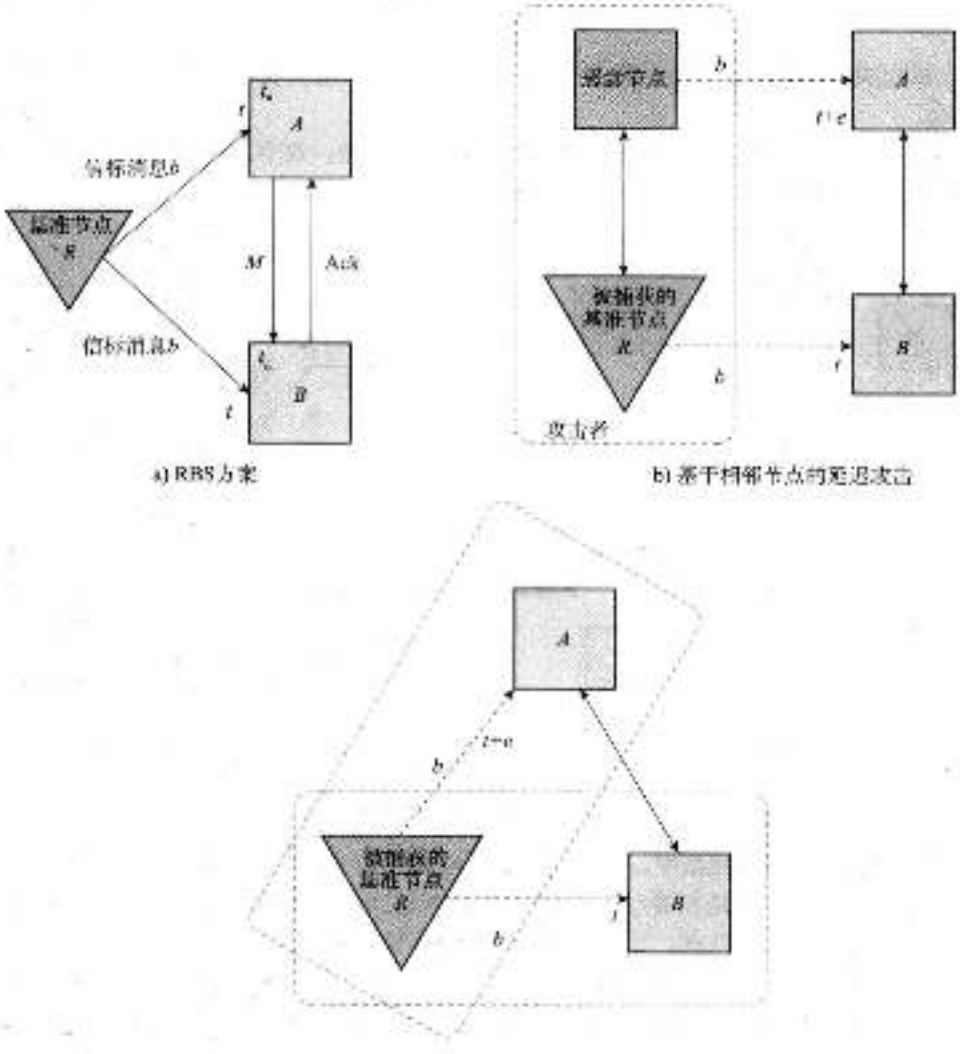


图 11-6　RBS 方案和延迟攻击

间收到该信标。图 11-6c 显示如果一个恶意节点有一个定向天线（而不是全向天线），它可以单独发动上述攻击。因此，节点 A 和节点 B 只能接收到一个信标消息。

注意：如果一个正常节点和一个已被捕获的节点同步，也可以发动延迟攻击。这个恶意节点可以故意使信标接收时间延迟，达到误导正常节点同步到错误时间的目的。

上面例子说明了针对基于“接收者－接收者”方式的时间同步模型的延迟攻击。延迟攻击同样也会发生在基于“发送者－接收者”方式的时间同步模型［Ganeriwal03］中，其中发送者和接收者通过交换时间同步消息来估算它们之间的往返传输时间（RTT）。这样可以发现时钟偏移后进行较为精准的补偿估计。如果传感器节点与一个恶意节点进行同步，它可能会获得错误的时间偏移值，从而同步到错误的时间。因此，这些方法也会遭受延迟攻击。

抵御延迟攻击的总体思路是找出恶意时间消息并排除它们，其基本步骤如下：

354
-
355

1）从参与节点中收集时间偏移量数据集合。

2）使用某种方法（如基于孤立点检测的统计算法）来识别延迟攻击发送的恶意时间偏移量。

3）被识别的恶意时间偏移量将被排除，而剩下的正常时间偏移量将被用于估计实际的时间偏移量。

文献［Hu07］提出了两个用于收集时间偏移量的模型：双节点模型（two-node mode）和相邻节点模型（neighboring-node mode），如图 11-7 所示。

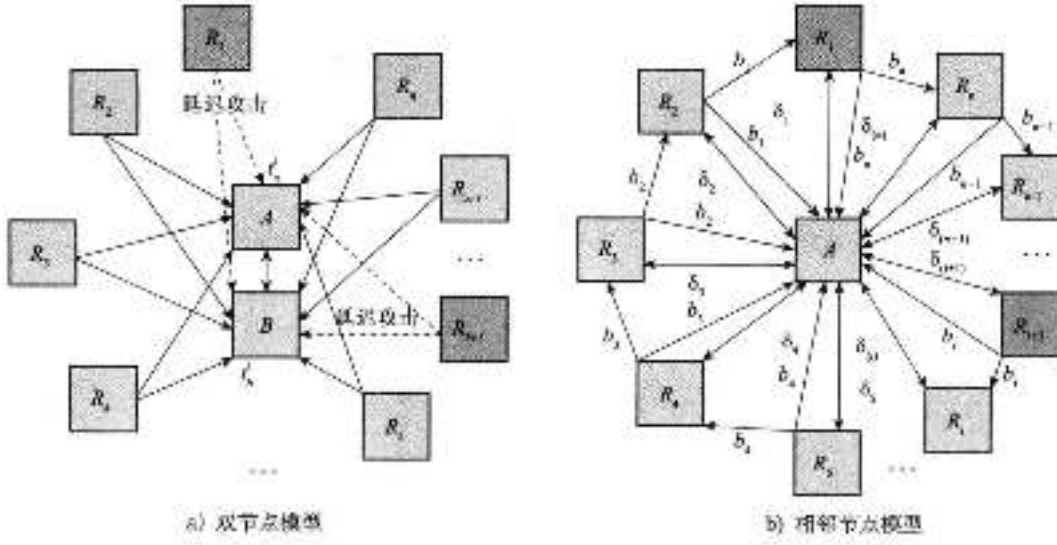


图 11-7　两种安全时间同步模型

双节点模型：在这种模型中，一个节点仅需与其簇首节点同步。如图 11-7a 所示，假设节点 B 是簇首节点，节点 A 是在节点 B 簇中的普通节点。出于安全考虑，节点 A 只信任簇首，而不信任其簇内的任何其他节点。节点 A 也只需与簇首节点 B 同步。

为抵御延迟攻击，节点 A 使用多个参考节点（R_1，R_2，…，R_n）获得一组时间偏移量。如果（t'_i，t_i）代表参考节点 R_i 分别接收到来自节点 A 和节点 B 的信标消息的时间（即当信标消息分别从节点 A 发送到 i 和从节点 B 发送到 i 时节点 R_i 的信标接收时间）。将 $\delta_i = (t'_i - t_i)$ 定义为时间偏移量。这样可以得到一组 n 个时间偏移量 $\{\delta_1, \delta_2, \cdots, \delta_n\}$。基于这些收集到的时间偏移量，可以使用一些统计算法来检测和排除恶意的时间偏移量，并获得更准确的节点 A 与 B 之

间的实际时间偏移量的估计值。

相邻节点模型：在这个模型中，一个节点需要与其多个邻居节点（大于 2）同步以监测延迟攻击。使用相邻节点模型的原因是双节点模型是不完善的，当一个或多个相邻节点可能受到攻击时，双节点模型就不适用了。正常节点可能会与恶意节点同步，然后发动延迟攻击，如图 11-7b 所示。假设节点 A 有 n 个相邻节点：R_1，R_2，…，R_n，在 A 与其每个相邻节点之间运行 RBS 机制，每次都使用一个不同的节点作为获取时间偏移量的参考节点。在收集了 n 个时间偏移量后，首先检测出异常值，然后进行排除，这样就能对实际时间偏移量做出正确估计。 356



奇思妙想

抵御延迟攻击的策略是基于孤立点检测，换句话说，就是从大量数据中找出那个“奇特”的值。正如你看到的，我们可以从不同的角度实现安全性；尽管传统的加密/解密方案可以用于大部分的应用，如果一个内部节点被捕获并成为“间谍”，我们需要其他的非加密方式找出这个“间谍”。本节介绍了利用数学统计方法来检测异常行为。请记住：所有学科都有所关联，可以衍生出一些“神奇”的方法应对挑战。

除了上述两种模型外，还有其他一些收集时间偏移量的时间同步模型。这些模型都有一个共同点：它们都收集一个时间偏移量数据集合，该集合中可能包含恶意的时间偏移量。

接下来的一个问题是：如何从时间偏移量数据集合中检测和排除恶意时间偏移量并获得一个较为精确的补偿估计？

可以想象，如果没有延迟攻击，节点间的时间偏差会遵循类似于统计分布的模式。延迟攻击的存在使得恶意的时间偏移量明显不同于正常的时间偏移量。从统计学的角度来看，这些恶意时间偏移量被称为孤立点（outlier），其定义为：极大偏离绝大部分观测值的值，会使人们怀疑该值是由其他机制生成的［Hawkins80］。

目前已有很多方案用来监测孤立点（文献［Iglewicz93］对已有的相关工作做过全面介绍与分析比较）。［Hnil07］介绍了一种检测孤立点的算法——GESD。GESD 是以极端学生化偏离（ESD）测试为基础的（也被称为格布拉测试，Grubb's test）。ESD 测试可以在随机取样的正常样本中检测到异常值。

ESD 测试的定义如下：数据样本为 $\Gamma = \{x_1, x_2, \ldots, x_n\}$，$\Gamma$ 的平均值为 $\bar{x}$，Γ 的标准差表示为 s。

$$T_i = |x_i - \bar{x}| / s, \quad i = 1, \cdots, n$$

T_i 也被称为 x_i 的对应 T 值。设 x_j 为使 $|x - \bar{x}| / s$ 达到最大的观测值，其中 $i = 1, \ldots, n$。当 T_j 超过一个预置的临界值 λ 时，x_j 就成为一个孤立点。原则上，如果 T_j 不超过临界值 λ，我们不需要将 x_j 定义为孤立点。假设测试中发现一个孤立点，我们从样本集合中排除这个值 x_j 后，在剩下的 $n-1$ 个数值中继续寻找孤立点。不过，ESD 测试一次只能检测到一个孤立点。 357

GESD 过程［Hnil06］是对 ESD 测试过程的改进，它可以一次找出多个孤立点。GESD 有两个重要参数：1）r 是对数据集中孤立点数量的估计值；2）λ_i 值是（100 * α）% 的双侧临界值，可由下面公式得出：

$$\lambda_i = \frac{t_{n-i-1,p}(n-i)}{\sqrt{(n-i-1+t^2_{n-i-1,p})(n-i+1)}}$$

其中 $i = 1, \cdots r$；$t_{v,p}$ 是自由度为 v 的 t 分布曲线上的（100 * p）百分点；$p = 1 - [\alpha/2\ (n - i$

+ 1)]。对于给定 α，n 和 r，我们可以计算出临界值 λ_r。

定义（基于 GESD 的延迟攻击检测）：对于给定的时间偏移量集合 $\Gamma = \{\delta_1, \delta_2, \cdots, \delta_n\}$，任何被 GESD 检测为孤立点的时间偏移量所在的节点都可以看成在遭受延迟攻击。

在 GESD 中，r 是对恶意时间偏移量（即数据集中的孤立点）数量的估计值。r 值的选择对于 GESD 的检测效果影响非常大。一方面，如果 r 值过小并且在 n 个时间偏移量集合中恶意时间偏移量（孤立点）超过 r 个，那么部分恶意时间偏移量就不会被检测出来；另一方面，如果 r 值过大，GESD 会浪费大量时间对正常节点进行不必要的检测。

一般情况下，时间偏移量的数量较小（例如 20），我们假设恶意时间偏移量的数目不超过时间偏移量总数量的一半。相应地，GESD 检测过程设置 r 为时间偏移量的总数量的一半。如果不对恶意时间偏移量的数目进行假设，GESD 可能无法正常工作，因为它可能会把恶意时间偏移量误认为成正常值，而把正常时间偏移量误认为恶意值。

定义（r 值估计）：假设时间偏移量集合 Γ 的平均值为 $\bar{x}$，标准差为 s，那么 r 值为满足以下条件的时间偏移量 x_i 的数量：

$$|x_i - \bar{x}| / s > 2, \text{ 其中 } i = 1, \cdots, n$$

由前面定义可知，r 值是集合 Γ 中所有与平均值 $\bar{x}$ 相差超过标准差 2 倍的元素（时间偏移量）的数量。在大多数情况下，数据和时间偏移量是符合正态分布的，这意味着 95% 节点的时间偏移量与平均时间偏移量 $\bar{x}$ 相差不超过标准差 s 的 2 倍。相应地，在网络中恶意节点数量较小（即小于网络节点总数的 5%）的情况下，我们将 r 值设置为时间偏移量的平均值。

[358]

问题与练习

11.1　多项选择题

（1）下列关于传感器网络安全的描述中，哪些项是不正确的？（　　）

A. 密钥管理包括密钥的生成和分发，它是保证传感器网络安全的重要环节。

B. 传感器网络安全最重要的目标是要保证传输数据的机密性，其他安全目标是次要的。

C. 传统网络的安全机制由于其计算开销过大而不适用于资源严格受限的传感器网络。

D. 传感器网络的安全机制在无线传感器节点内占用较少的存储空间（小于 100K 字节）。

（2）下列关于传感器网络攻击的描述中，哪些项是不正确的？（　　）

A. 攻击者不属于传感器网络中的成员节点，它无法获得存储于网络中成员节点内部的加密密钥。

B. 侧信道攻击（Side-channel attack）是指任何利用密码芯片运算过程中泄露出来的各种物理信息（如功耗、执行时间、电磁辐射等）来破解密码系统的攻击。

C. 干扰攻击（Jamming attack）是典型的物理层攻击。

D. 链路层攻击总是试图破坏正常的介质访问控制操作。

（3）下列关于传感器网络中路由层攻击的描述中，哪些项是正确的？（　　）

A. 攻击者能够对网络中的路由控制命令进行误导。

B. 攻击者通过发动 Sinkhole 攻击能够把网络中的数据包吸引至自身处。

C. Sybil 节点能够伪造新的身份。

D. 以上描述均正确。

（4）虫洞攻击具备下列哪些特征？（　　）

A. 恶意节点会把接收到的数据包通过低时延链路秘密发送给处于不同位置的另一恶意节点，由其对该数据包进行重放。

B. 如果恶意节点的有效通信距离较大，那么它能够通过发送广播包欺骗网络中所有节点，使其认为该恶意节点为其邻居节点，这样会导致网络中大量的报文丢失。

[359]

C. 由于传感器网络无线通信所固有广播特性，攻击者能够窃听到发送给其邻居节点的数据包并广播伪造的路由应答报文。

D. 恶意节点在发动虫洞攻击时总是试图隐藏自己身份。

(5) 下列关于虫洞攻击的描述中，哪些项是正确的？(　　)

A. 攻击者能够通过对路由报文进行封装的方式来建立虫洞。

B. 两个攻击者能够利用低劣的通信信道（相对于正常的传感器节点间的链路）来发动虫洞攻击。

C. 两个恶意节点能够通过增大发射功率的方式建立虫洞。

D. 为了发动虫洞攻击，恶意节点一般不遵守正常路由协议要求，而是在接收到数据包后立即转发，这样可以保证数据包最快到达目标节点并且造成该恶意节点与目标节点间链路数据传输延时最小的假象。

(6) 下列关于时间同步安全的描述中，哪些项是正确的？(　　)

A. 延迟攻击是一种试图以通过延迟时间戳数据包发送的方式来导致时间同步过程失败的攻击。

B. 传统的加/解密方法能够解决时间同步过程中的安全问题。

C. 孤立点检测方法旨在通过利用统计均值剔除异常值的方式来抵御延迟攻击，保证时间同步安全。

D. 以上描述均不正确。

(7) 与 TinySec 相比，MiniSec 具有以下哪些优势？(　　)

A. 由于利用了单播和广播通信的本质性区别，MiniSec 能够提供两种能量优化的通信方式。

B. MiniSec 的能效比 TinySec 更高。

C. MiniSec 可以在路由层工作。

D. A 和 B。

(8) 下列关于 μTESLA 的描述中，哪些项是正确的？(　　)

A. μTESLA 首次利用了 TESLA 来剔除恶意节点。

B. μTESLA 把整个广播数据源认证过程划分成多个时间间隔。

C. μTESLA 会延迟发布认证密钥。

D. μTESLA 通过单向散列函数来生成密钥。

11.2　阐述 LITEWORP 协议检测虫洞攻击的基本原理，并给出相关工作示意图。

11.3　为什么 μTESLA 会延迟发布认证密钥？应该延迟多长时间？

11.4　请以一个采用广播通信方式的无线传感器网络为例介绍 μTESLA 的基本工作原理。

11.5　阐述如何应对时间同步过程中的存在的延迟攻击。

11.6　阅读关于 TinySec 和 MiniSec 的相关文献，列出两种协议存在的主要区别。

11.7　阐述基于 Blom 思想的密钥管理方案的密钥生成原理。

360

第六部分

Wireless Sensor Networks: Principles and Practice

特殊无线传感器网络

第12章

Wireless Sensor Networks: Principles and Practice

无线传感器和执行器网络

12.1 引言

无线传感器和执行器网络（WSAN）[Akyildiz04]是分别针对异构的传感器和执行器的分布式无线通信和控制系统。传感器具有通常WSN所具有的特点，例如，低成本、低能耗和近距离无线通信的多功能设备[Akyildiz02]。执行器负责搜集和处理传感器数据，并在网络环境下履行职责。不同于传感器，执行器具有充足的资源，例如，高效处理能力、高效传输能力和长期的电池续航能力。

执行器不同于传统意义的驱动器（actuator）。驱动器通常指能够将电控信号转换为物理动作的设备，可作为流动控制阀、泵、马达等。执行器除了具有驱动器的功能外，还具有一个更重要的功能：它可作为一个独立的实体在网络中执行网络相关任务，即对数据进行接收、传输、处理和转播。例如，机器人可以通过若干电机马达（即驱动器）和物理环境进行交互。然而，从网络化视角来看，机器人构成了一个独立的网络实体，此时可被称为执行器[Melodia07]。

如图12-1所示，传感器和执行器被部署在广阔的区域内，汇聚节点负责监控整个网络，并与任务管理器节点以及传感器/执行器节点通信。类似于WSN，WSAN（Wired Sensor and Actor Network，无线传感器/执行器网络）可具有成百上千的传感器节点。这样的部署密度对于执行器是不必要的，因为执行器通常比传感器昂贵，并且能在大范围内提供高效的执行能力。

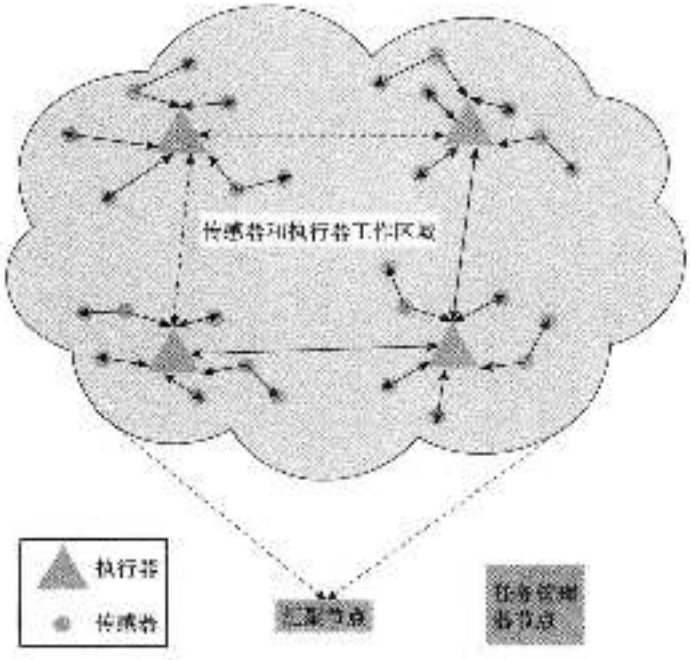
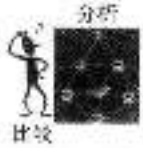


图12-1 WSAN的物理结构



正如我们所介绍的，WSAN 是一类特殊的 WSN，二者的最大区别在于 WSAN 包含有少量的执行器，这些执行器是可移动的，比传感器具有更好的 CPU 性能和更长的无线通信时间。执行器需要和传感器协作判断如何对特定探测事件做出响应。

如果一个传感器探测到一个事件，那么所有传感器会将它们接收到的信息发送给执行器，执行器负责处理这些输入数据并启动适合的动作，或者将数据通过一级级路由传输到汇聚节 364
点，由汇聚节点选择一个执行器处理该事件。第一种情况被称为自动化架构（如图 12-2a 所示），该架构没有中心控制节点（例如，汇聚节点）；而第二种情况被称为半自动化架构（如图 12-2b 所示），在该架构中，由汇聚节点（中心控制器）收集数据并协调行动过程。这两类模式各有利弊。自动化模式可缩短动作反应时间，而半自动化模式具有更好的全局管理能力，因为汇聚节点可以通过检查所有执行器的资源状态做出决策。

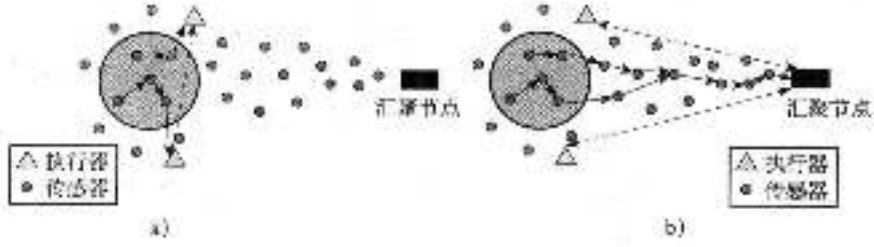


图 12-2　自动化和半自动化架构

在传感器检测到某个事件后，它们会直接（不经过汇聚节点）要求执行器对事件做出响应。这些传感器和执行器相互配合，构建它们间有效的路由路径。这种协作过程被称为传感器 - 执行器协同工作［Akyildiz04］。

另一方面，当执行器接收到传感器发送来的事件处理请求时，它可能会因为能力有限而不能高效地处理该事件，此时它会和其他执行器合作共同决策如何处理该事件。这种协作过程被称为执行器 - 执行器协同工作。

WSAN 的主要功能需求有：

1）WSAN 要求执行器间（也包含传感器和执行器间）能够实时地协作和通信，以保证能够及时地做出正确响应。

2）WSAN 中能量的使用效率要得到保证，尤其是传感器，它具有有限的资源和电池续航能力。

3）和 WSN 相同，WSAN 中的协议和算法仍然需要具有可扩展性和伸缩性，因为传感器的数目可能是无穷多的。

WSAN 在战场监视，核、生物、化学攻击检测，家居自动化和环境监测中都有广泛应用，Akyildiz 和 Kasimoglu［Akyildiz04］提供了一些好的案例，如下所示。

- **火灾监控**：当建筑物发生火灾时，温度/烟雾传感器能够监测到火源地点和火势，并把这些参数信息发送给自动喷水灭火装置（即执行器），该喷洒器会在火灾失控前将其扑灭。 365
- **污染监控**：传感器可对水或空气中可见或可检测的污染物排放进行监测，执行器（例如污染清除器）会对污染采取应对措施。

- **建筑物监视**：在建筑物内部，可通过移动、声音和光线传感器来监测到人侵者的存在，并通过摄影机来跟踪该人侵者，安保人员随后（也可称为执行器）可据此尽快到达该监测地点。

12.2　传感器－执行器协同问题

正如前面所讨论的，传感器和执行器间需要实时通信。Melodia 等［Melodia07］介绍了一系列方案来解决传感器－执行器协同问题。除了实时性要求以外（即有限定的通信延迟），还需要考虑通信可靠性问题。该文章介绍了一种“延迟限定可靠性”。它考察事件发生区域中传感器发出的所有包和预定延迟时限内接收到的包（指的是可靠的包）之间的比例关系。请注意，此处所指可靠性是与源节点到执行器间数据包的实时传送相关的，并且在网络层中进行此比例的计算。

延迟时限（latency bound）B 的定义如下：从传感器采集到事件特征信号的时刻到执行器最终接收到数据包的时刻间的最大时间间隔。

显然，若一个数据包没能在延迟时限 B 内到达执行器，该包就失去了使用价值。r_A被定义为应用所要求的最小事件可靠性，可靠性的不满足度表示为（$r_A - r$），就是在给定时间内所需的事件可靠性阈值 r_A和实际观测到的事件可靠性 r 间的差值。

现在，传感器－执行器协同问题可以确切地表达为：

在以下两个条件下，如何创建事件发生区域中各个传感器到执行器间的路由路径？

- 实际可靠性 r 要大于阈值 r_A（即 $r \geq r_A$）
- 路由路径满足最低能量消耗

基于以上目标，Melodia 等［Melodia07］通过基于事件驱动的多执行器区域划分解决了传感器－执行器协同问题，并使用了一种称为**整数线性程序**（Integer Linear Program，ILP）的数
366 学模型。为解释和说明 ILP，首先需要定义 WSAN 的网络模型和能量模型。



当我们在一系列约束/条件下试图最小或最大化一个目标函数时，并且这些条件可表示为数学公式（即 A > B、A < B、A = B 等形式，其中 A、B 为函数），此时可以考虑使用 ILP。ILP 实际上是一个函数最优化问题。在MATLAB（一个数学工具）中可以创建这些 ILP 约束。

12.2.1　网络和能量模型

Melodia 等［Melodia07］使用一种图模型描述网络拓扑，则 WSAN 可表示为一个图 $G(S_v, S_E)$，其中 $S_v = \{v_1, v_2, \cdots, v_n\}$ 是有限维空间中顶点的有限集合（$N = |S_v|$），S_E是这些顶点间边的集合，即 $e_{ij} \in S_E$，当且仅当顶点 v_i 和 v_j（下面也简称为 i 和 j）在彼此的传输范围内。

设 S_A表示执行器的集合（$N_A = |S_A|$），并指定用来从一个或多个源收集数据的执行器为数据收集节点。

设 S_S表示数据源的集合（$N_S = |S_S|$），该集合代表了检测事件的所有传感器节点，即在事件发生区域中驻留的所有传感器。

定义 $P = \{(s, a): s \in S, a \in A\}$ 为源到目标连接的集合。

能量模型：按照文献［Heinzelman02］中的模型，假设每个位（物理层中的 bit）的能量消耗为 $E = 2E_{elec} + \beta d^{\alpha}$，其中 α 是路径损耗指数（$2 \leq \alpha \leq 7$），β 是常量，E_{elec}是收发器发送或接收

1 位电路所消耗的能量。

12.2.2　ILP 算法

基于 ILP 的传感器－执行器路由搜索问题就是寻找数据汇聚树（data aggregation trees，简写为 da-tree）。该树包含了所有事件区域中驻留的传感器（指数据源）到合适的执行器的通信路径。da-trees 的叶子是传感器源节点（并不是所有的传感器源都必须是叶子节点），并且每个执行器或者是 da-tree 的根节点，或者是通信的参与者。

基于 ILP 的算法旨在构建一系列的 da-tree，并且每个传感器源节点只属于其中一棵树，每个 da-tree 具有一个执行器作为根节点。因此，每个传感器源节点和某一个执行器关联，实现最优化的基于事件驱动的区域划分。

ILP 算法是通过以下两个主要步骤实现基于事件驱动的区域划分：1）选择传感器信息要 367
发送到的执行器的最优化子集；2）选择好执行器后，针对它们构建最小能量 da-tree，以满足所需要的事件可靠性约束（即上面所提到的两个条件）。

因此，可以根据以执行器为根的 da-trees 对事件发生区域中源节点进行划分。图 12-3 给出了事件区域划分的一个例子。

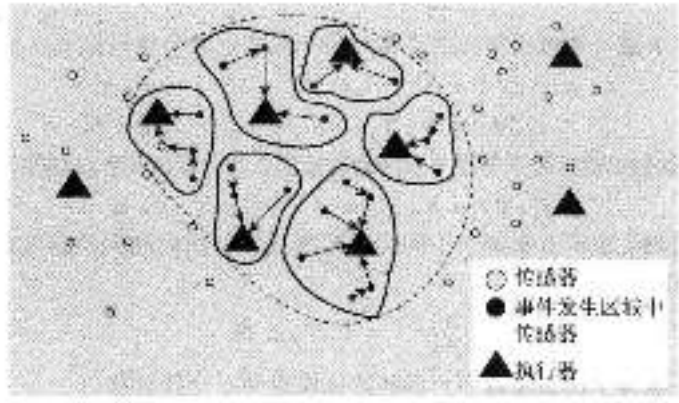


图 12-3　具有多执行器的基于事件驱动的区域划分

在使用 ILP 算法形式化表达区域划分问题之前，先介绍在 ILP 模型中使用的符号：

e_{ij} 是一个布尔变量（0 或 1），当节点 i 和 j 在彼此的传输范围内时该值为 1。

c_{ij} 是节点 i 和 j 间链路的能量消耗，即 $2E_{elec}+\beta d_{ij}^{\alpha}$，其中 d_{ij} 是节点 i 和 j 间的距离。

x_{ij}^{k} 是一个布尔变量，当链路 (i, j) 是与执行器 k 关联的 da-tree 的一部分时，该值为 1。

f_{ij}^{sk} 是一个布尔变量，当源节点传感器 s 发送数据到执行器 k，并且链路 (i, j) 是 s 到 k 的路径时，该值为 1。

l_{sk} 是一个布尔变量，当传感器 s 向执行器 k 发送数据时，该值为 1。

p_{ij} 是与链路 (i, j) 关联的传播延迟，定义为 d_{ij}/v，其中 v 是信号传播速度。

d 是每个传感器节点上处理、排队、介质访问的时延参量。

B 是每个源节点－执行器流的传输延迟的界限。

r 和 r_{th} 分别表示事件实际可靠性和所需要的事件可靠性阈值。

b_{sk} 是布尔变量，当源节点 s 和执行器 k 间连接超过延迟界限（即端到端延迟大于延迟界限 B）时，其值为 1。 368

Q 是未服从约束的源节点数目。

基于 ILP 的区域划分问题（即 da-trees 构建）可形式化表达如下：

$p_{\mathrm{MS}}^{\mathrm{Cap}}$ 表示具有多执行器的事件区域传感器划分。

给定 e_{ij}、c_e、p_e、ε、$\bar{d}$、B、$r_{\min}$；要找到 x_{ij}^k、f_{ij}^{sk}、I_{sk}、b_{sk}、r；

最小化

$$C^{\mathrm{DD}} = \sum_{k\in S^A}\sum_{(i,j)\in S^V} x_{ij}^k \cdot c_e + \gamma \cdot Q \tag{12.1}$$

以上的公式称为目标函数。一旦建立了所有从传感器到执行器的路由，整个系统应具有最小能量消耗。在该公式中，对不满足条件的传感器数目 Q 添加了一个补偿系数 γ。

还要受到以下约束条件的限制：

$$\sum_{i\in S^V}(f_{ij}^{sk} - f_{ji}^{sk}) = I^{sk}, \forall s \in S^s, \forall k \in S^A \tag{12.2}$$

（该约束条件可确保传感器源节点仅可从所选执行器的 da-tree 上生成一个数据流，而非源节点不产生任何数据流。）

$$\sum_{i\in S^V}(f_{ij}^{sk} - f_{ji}^{sk}) = -I^{sk}, \forall s \in S^s, \forall k \in S^A \tag{12.3}$$

（该约束条件要求由每个传感器源节点生成的数据流都仅被一个执行器收集。）

$$\sum_{j\in S^V}(f_{ij}^{sk} - f_{ji}^{sk}) = 0 \tag{12.4}$$

（该约束条件要求输入和输出流间的平衡对于非源节点和非执行器节点是无效的。）

$$\forall s \in S^s, \forall k \in S^A, \forall i \in S^V \text{ s.t. } i \neq s, i \neq k$$

$$f_{ij}^{sk} \leqslant e_{ij}, \forall s \in S^s, \forall_s \in S^s, \forall_i \in S^V, \forall j \in S^V \tag{12.5}$$

369 （该约束条件确保数据流是基于邻接节点间的连接建立的，即这些节点在彼此的传输范围内。）

$$f_{ij}^{sk} \leqslant x_{ij}^k, \forall s \in S^s, \forall k \in S^A, \forall i \in S^V, \forall j \in S^V \tag{12.6}$$

（该约束条件强制要求所有来源于不同传感器源节点但指向相同执行器的数据流都汇聚到与该执行器关联的 da-tree 上。）

$$\sum_{i\in S^V} I^{sk} = 1, \forall_s \in S^s \tag{12.7}$$

（该约束条件要求每个传感器源节点都将数据发送到一个执行器。）

$$f_{ij}^{sk} \leqslant I^{sk}, \forall s \in S^s, \forall k \in S^A, \forall i \in S^V, \forall j \in S^V \tag{12.8}$$

（该约束条件确保在某执行器没有被源节点选择时，所有从源节点到某执行器的流变量都是零。）

$$\varepsilon \cdot \left[B - \sum_{(i,j)\in S^E} f_{ij}^{sk}(p_e + \bar{d})\right] \leqslant b^{sk}, \forall s \in S^s, \forall k \in S^A \tag{12.9}$$

（该约束条件要求当且仅当从传感器源节点 s 到执行器 k 间流不满足延迟约束 B 时布尔变量 b_{sk} 为 1。负系数 ε 用于调节方括号中的值使其小于 1，因此，当不满足延迟约束时，公式 12.9 中左边的部分是一个较小的正数，将使得布尔变量 b_{sk} 为 1。另一方面，当满足延迟约束时，公式 12.9 左边的部分是一个负值，b_{sk} 将取 0 值使公式 12.1 所示的目标函数最小化。）

$$Q = \sum_{s\in S^s}\sum_{k\in S^A} b^{sk}; \quad r = \frac{|S^s| - Q}{|S^s|} \geqslant r_{\min} \tag{12.10}$$

（Q 代表不满足约束条件的传感器源节点数目，可靠性 γ 通过满足约束条件的传感器源节点数目和所有传感器节点数目的比例来度量，r 应大于所需的阈值。）

12.2.3　传感器－执行器协同工作：分布式协议

下一步需要做的是将上面的数学模型转换为实际的传感器－执行器协同协议。分布式协同

的目标是在传感器源节点（驻留在事件发生区域中）和执行器间构建 da-trees，并保证公式 12.1 所示的目标函数最小化。即在提供所需的可靠性 r_d 同时满足能量开销最小化。 370

正如前面所述，传感器－执行器协同工作协议的结果是构造一系列包含从源节点到执行器的所有路由通路的 da-trees，这是对具有多执行器的事件区域划分问题的一个近似解法。Melodia 等［Melodia07］称这种协议为基于事件驱动的分布式区域划分和路由协议（Distributed Event-driven Partitioning And Routing Protocol，简称 DEPR）。

我们已经知道，具有局部路由决策能力（即基于局部拓扑结构）的路由算法可以生成接近全局能量效率最优化的路由通路［Melodia05］。因此，DEPR 协议的目标是通过局部拓扑信息和贪婪路由选择决策方法实现最小化能量消耗。

为保证每个传感器－执行器路由通路都满足预定的延迟约束，需要某些形式的端到端信息反馈。DEPR 协议依靠收集来自接收执行器的反馈信息实现，每一个执行器将公布可靠性的观测值。

DEPR 协议通过每个传感器节点进行本地行为控制的方式实现具备以下主要特征的全网协同工作的效果：1）满足事件可靠性 r 大于所需可靠性阈值 r_d；2）能量消耗最小化。

然而，DEPR 通过修改平均路由长度调整路由延迟从而控制网络可靠性。通常在无线网络中，可在发射节点上通过改变发送能量控制每个通信链路的能量消耗。也就是说，可通过降低发射节点天线的功率级别来节省更多的能量。另一方面，改变发射器的功率级别可以控制信号传播距离，功率越大，信号传播距离越远，并具有更低的传输延迟。

DEPR 协议在它的底层路由算法中做出了一些假设：可以使用传感器定位方案保证每个传感器知道自己的位置，并可通过每个节点广播的位置信息获知相邻节点的位置；执行器需要定期地在传感区域中公布其位置信息，以保证各传感器能够获知执行器的位置；可以通过现有的某个时间同步协议实现整个网络的同步［Sundararaman05］。

12.2.4　DEPR 概述

再次回忆一下，DEPR 协议的目标是在所有源节点和若干执行器之间建立 da-trees，这些执行器称为汇聚节点。在汇聚节点和传感器源节点间建立的 da-tree 可将传感器数据传送到汇聚节点。该协议可将事件发生区域划分成若干个分区，每个子区中包含的传感器节点都与某个汇聚节点关联。

DEPR 协议要求每个传感器具有四个状态，分别为：空闲、启动、加速、汇聚状态。当可靠性需求没有得到满足时，状态转换的目的是降低通信链路的跳数，从而降低传输延迟。而当可靠性需求被满足时，则状态转换的目的是节省能量。 371

除了状态的转换，DEPR 协议可通过调整发射功率控制能量和可靠性。发射功率等级影响着无线信号的接收质量，从而影响分组差错率。提高发射功率会增加可靠性。

发射功率也决定着无线通信的范围，从而影响路由层中下一跳的可选路径。提高发射功率可减少到达预订目标的链路跳数，也可通过增加直通链路的数目使网络更具连通性。

相反的，降低发射功率会减少能量的消耗，从而延长网络通信的时限。然而，较低的发射功率会导致传感器通信范围的缩短，因此需要更多的转发节点（即更多的跳数），从而导致端到端通信延迟的增加。

总而言之，有效的能源控制方案可以在能量消耗和路由延迟间达到平衡，这也是 DEPR 协议的目标。在以下的讨论中，会介绍 DEPR 协议的细节。

每个包的延迟时间可由执行器通过接收到的包头中时间戳计算得出。在决策期间，执行器会通过未按期到达的包数占所有发送包数的比例计算数据到达可靠性 r，并且定期将计算结果广播

给其邻接节点。传感器节点基于汇聚节点观察到的可靠性（和该汇聚节点关联）控制着它们的状态转换（在空闲、启动、加速、汇聚状态间切换），并且在每次状态变换后广播其状态。

传感器的状态变化规则如下：

初始是空闲状态。此时传感器探测环境数据并监控无线信道传入的数据包。当监测到特殊事件或从相邻传感器接收到第一个数据包时，传感器进入启动状态。

传感器节点期待从关联的汇聚节点（即执行器）得到反馈信息。若公布的事件可靠性 r 值低于事件可靠性阈值的下限 $r_{\perp}$，那么需要通过缩短转到端链路的长度来降低传感器－执行器路由延迟。因此，当 $r<r_{\perp}$ 时，处于启动状态的传感器会以 $P_{\uparrow\uparrow}$ 的概率进入加速状态，该概率值是随可靠性的不满足度（$r_{\perp}-r$）而单调增长的。注意，此处使用了概率策略（$P_{\uparrow\uparrow}$）避免出现状态转换的死锁（即系统震荡），该问题会在所有传感器同时改变状态时出现。

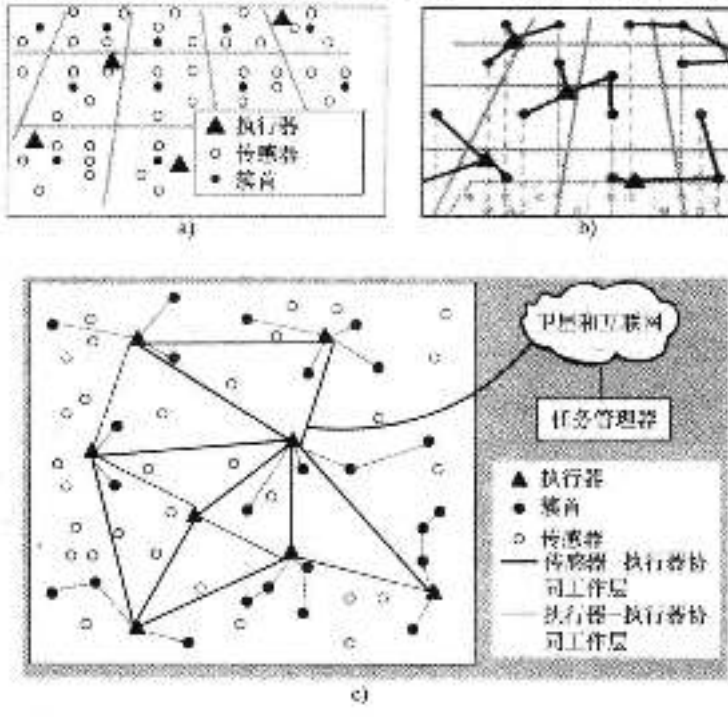
若事件可靠性 r 高于事件可靠性阈值的上限 $r_{\top}$（即 $r>r_{\top}$），则需要考虑节省能量。此时，处于启动状态的节点会以 $P_{\uparrow\downarrow}$ 的概率进入汇聚状态，该概率值随着可靠性的超出值（$r-r_{\top}$）而单调增加。
372 在这种情况下，节点通过向 de-tree 上最近的邻居节点传输数据以实现能量消耗最小化。

接下来，传感器可根据汇聚节点的反馈信息在启动和汇聚状态间切换。DEPR 协议的目标是通过调整各传感器的状态使得整个网络通信在能量消耗最低的情况下趋于可靠性阈值。若在超时时限内不再产生和接收数据包，则传感器回到初始的空闲状态。

12.3　层次化传感器－执行器协同工作机制

12.3.1　层次化 WSAN 协同工作架构

在文献［Yuan06］中提出了一种三层结构的传感器－执行器协同模型，如图 12-4 所示。



373 图 12-4　三层协同工作模型

- **第 1 层　传感器 - 传感器协同工作**：如图 12-4a 所示，传感器 - 传感器协同工作是通过领导从其他相邻传感器收集数据的聚类技术实现的，传感器 - 传感器协同工作的目标在于实现能量消耗最小化和网络生命周期最大化。
- **第 2 层　传感器 - 执行器协同工作**：如图 12-4b 所示，传感器 - 执行器协同工作的目标是当簇首向适合的执行器发送数据时使延迟最小化。另一个目的在于使得执行器能够执行大多数耗能较多的任务，如路由计算和数据汇聚。
- **第 3 层　执行器 - 执行器协同工作**：如图 12-4c 所示，执行器 - 执行器协同工作的目标是控制执行器高效、可靠地执行任务，主要目的在于通过对各节点任务的最优化分配和合作实现总体任务处理性能的最大化。



多级协同工作方案已用于多种问题的解决，其基本特点是将节点间协同工作关系分成两个或更多层次，并在相邻的两层间定义严密的映射关系。高层比低层的节点数目更少（而各节点能力较强）。因此，在高层中通信量较少，对于低层更关心可扩展性和能效，以应付大量节点间的通信需求。

12.3.2　"传感器 - 传感器" 协同工作层次——使用聚类

传感器 - 传感器协同工作基于以聚集为基础的路由协议，LEACH［WBHeinzelman02］和 TEEN［AManjeshwar01］都可采用，因为它们都使用聚类。然而它们没有使用地理位置信息，这些信息对于文献［Yuan06］所提出的基于网格的路由架构是非常必要的。每个传感器/执行器在获知它们所属网格之前需要知道它们所处的地理位置。

GAF（Geographical Adaptive Fidelity）［YXu01］是一个基于位置的路由算法，但它不是基于聚类的算法。GAF 把网络区域划分成虚拟单元格。在每个单元格中，节点间相互合作依次成为活动节点或休眠节点。GAF 通过关闭不必要的节点来节省能量。

图 12-5 展示了基于 GAF 的聚类算法，静态节点的无线通信范围 R 如图 12-5 中虚线所示。假设虚拟单元格是一个边长为 r 的方格，两个相邻方格中的传感器节点间的最远距离不大于 R［Yuan06］，由此可得

$$r \leqslant \frac{R}{\sqrt{5}} \tag{12.11}$$

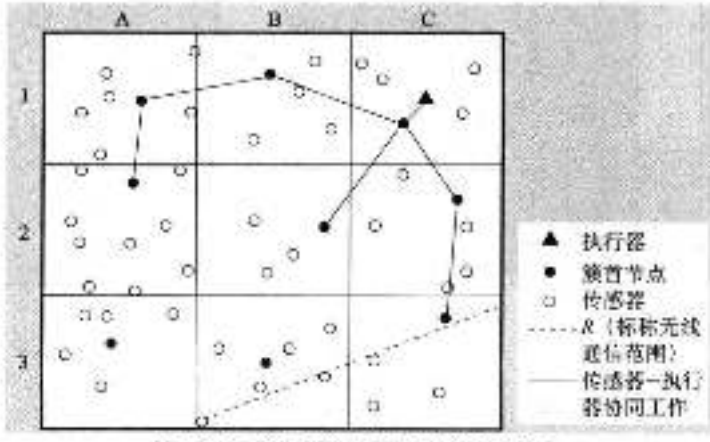


图 12-5　基于通信范围 R 的聚类和路由

在邻接网格中的传感器间可直接相互通信。在每个网格中，会推选一个传感器作为簇首节点以汇聚与它关联的传感器的探测数据，并负责为合适的执行器监控和报告数据。

12.3.3　“传感器－执行器”协同工作层次

为实现传感器－执行器协同工作，需要所有执行器定期或在执行器移动期间持续地公布它们的信息（例如，当前位置和准确时间）。在簇首节点（静态传感器）接收到这些信息后，它们可以获得时间的同步，并可维持一个邻接执行器路由表，以应付移动的执行器节点。

由于整个探测区域被分为基于位置的网格，每个网格的序列号可认为是相应节点簇首节点的ID。更近一步，每个簇首节点不需要交换和记录其他簇首节点的ID信息，因为事件信息是通过簇首节点向执行器逐网格传递的，而且只有包含在路由通路中的网格才参与通信过程。

当检测到一个重要的事件时，哪个执行器会对此作出响应？在12.2节中，使用了基于ILP的事件区域划分算法实现所有传感器相互合作并根据不同的条件选择合适的执行器完成此任务。这些条件可以是：执行器与事件区域间的距离，传感器的能量消耗或者执行器的行为有效范围。当某个执行器与事件区域接近时，它最先得到消息，也能最快做出响应。因此，选择执行器的最好条件是簇首节点和执行器间的距离，即执行器和事件区域间的距离。

在Yuan等［Yuan06］提出的传感器－执行器协同工作模式中，簇首节点负责为最近的执行器监控和传送数据。当簇首节点和所有的执行器间距离都较远时，簇首节点将和最近的并与执行器接近的簇首节点合作（基于地理位置信息）实现数据的监控和传送。

在簇首节点找到最近的执行器后，它会记录该执行器的一些信息（例如，当前位置、准确时间）并维护路由表。当某个事件发生时，每个簇首节点可将事件信息立刻转发给最近的执行器（在一跳或几跳范围内），而不用花费较多的能量和时间去建立路由。

当事件区域比较大并与多个执行器存在关联时，该区域中的每个簇首节点仍然和最近的执行器进行通信。所有和同一执行器关联的簇首节点可构建一个针对该执行器的数据聚合树，所有由同一事件触发的执行器可在执行器－执行器协同工作层构建一个第2级聚合树，如图12-6所示（仅针对位于事件区域中心位置的执行器）。这种策略可以获得最优化的能源效率，并满足事件的可靠性和时限要求。

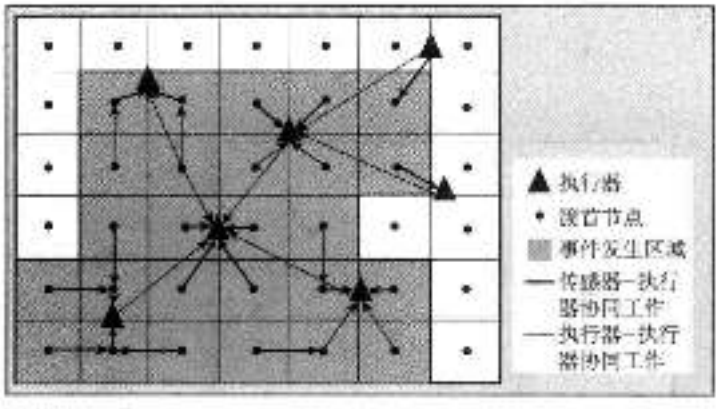


图12-6　两级聚合树

12.3.4　“执行器－执行器”协同工作层次

根据监测到的事件的特点，会触发一个或多个执行器用于执行一个或多个任务。为解决任

务分配问题，执行器－执行器协同机制可以采取两种途径：执行优先（Action-First，简写 AF）方案和决策优先（Decision-First，简写 DF）方案。

1. 执行优先方案

当邻近事件区域的执行器通过传感器－执行器协同工作途径接收到事件信息时，这些执行器会立刻采取行动，而不必和远端的其他执行器协商。每个参与的执行器会向其他执行器（距离 1 跳或 m 跳距离）广播行为信息。远端的执行器在获知到此事件信息后，会立刻决定加入或退出该事件的处理。

在事件处理过程中，不需要太多的执行器参与其中，因此，Yuan 等［Yuan06］提出使用预置行为阈值来控制参与事件处理的执行器数目的方式，该行为期望值可表示为：

$$ex(N, A) = \alpha d(N, A) + \beta e(N) - \gamma n(A) + \delta p(A)$$

其中：

$ex(N, A)$ 是执行器 N 参与行为 A 的期望值。

$d(N, A)$ 是执行器 N 和行为区域 A 间的距离。

$e(N)$ 是执行器 N 剩余能量。

$n(A)$ 是执行行为 A 的执行器数目。

$p(A)$ 是行为 A 的优先级。

α、β、γ 和 δ 是比例参数。



通常是定义一系列因素，并根据它们的重要程度为其分配不同的权重，最终的公式可用于表达所有因素的综合效应。如何为不同的因素分配合适的权重是一个难题。

374 - 377

若行为阈值用 TH 表示，则当 $ex(N, A) > TH$ 时，执行器 N 会参与到行为 A 中。通过这种方式，从探测到行为响应的延迟将非常小。如果某个执行器由于行为范围或能量限制不能执行该行为，该执行器会发送“help”信息给其他执行器接替它完成此任务。

2. 决策优先方案

在该种方案中，所有接收事件信息的执行器间紧密合作以使它们的任务处理能力最大化。12.2 节中已经讨论了基于 ILP 的方法，可实现执行器间任务分配的最优化，该方法仍然适用于本方案。

在基于 ILP 的方法中，执行器－执行器协同模型可用于解决重叠区域问题（即某区域位于多个执行器的作用范围内）。同样地，在 DF 方案中，根据事件特点和基于位置的网格划分，事件区域可按照不同执行器来实现最优化分割，每个执行器的作用范围可在未重叠区域中有效地定位。

问题与练习

12.1　说明 WSN 和 WSAN 的区别。

12.2　在文献［Melodia07］提出的算法中，本章简略地介绍了传感器－执行器协同工作算法，请阅读原始文献，并详细解释该算法。

12.3　为什么文献［Yuan06］使用一个三层结构的协同工作机制，而不是单层结构？ 378

第13章

Wireless Sensor Networks: Principles and Practice

水下传感器网络

13.1 引言

13.1.1 水下无线传感器网络应用

地球70%的表面由水构成，因此，使用水下设备实现水下通信是非常重要的。水下无线传感器网络（Underwater Sensor Networks，USN）使用大量互连的水下传感器和移动航行器来完成协作监控任务。

水下无线传感器网络能够对沿海海洋的三维环境进行适应性采样，完成重要的水下任务，如污染监测、海洋/风力监测以及生物监测。污染监测有助于发现金属的含量，如水中铅的含量。海洋/风力监测对分析气候变化、气象预报或理解人类活动对海洋生态系的影响尤其重要。生物监测可以用于持续跟踪鱼类或微生物。

如果将水下传感器与其他传感器结合，可能有更多有用的应用。例如，地震传感网络能将海啸预报发送到沿海地区，或者研究海底震（海啸）的效应。地震监测为石油开采提供若干存储管理的方法。一些水下导航传感器能探测出海底危险，定位危险的岩石或浅水域的浅滩，
379 探寻沉船或进行地形分析。

各种装载传感器的水下移动航行器能侦察水雷，搭载声光传感器的航行器可以执行快速环境评估和探测类似水雷的物体。



奇思妙想

近来，低成本、大范围水下无线传感器网络备受关注。特别是美国海军，已经投入很多精力在实用水下无线传感器网络的设计上。应注意，虽然水下无线传感器网络是一个特殊的无线传感器网络，但是其与陆上无线传感器网络相比有很大的不同。读可能将在后续章节中做进一步解释，这也是为何用单独一章来详细说明水下无线传感器网络的原因。

许多传统水下网络不使用无线（声波）传感器，而是使用电缆连接少量的高成本水下传感器，该方案具有以下缺点：

1）因为接线费用高昂，很难实现实时监控。许多时候，直到仪器被回收，才能检索到记录的数据。因为电缆长度的限制和岸上控制系统与监控仪器缺少交互，很难完成实时系统重新配置。

2）高昂的部署成本对于大面积水域监控应用而言并不适合。

在设计水下无线传感器网络时仍然存在很多挑战。例如，对无线通信来说，声频链路（acoustic link）有带宽的限制；水下的传播时延比射频陆上信道的传输时延高五个数量级；因为不能利用太阳能，传感器电池能量是有限的，通常电池不能再充电，以及一些其他问题。

13.1.2 水下无线传感器网络与陆上无线传感器网络的区别

接下来将列举若干陆上和水下传感网络的主要区别：

最重要的区别是无线通信频率：在水下，因为水的特殊性质（射频信号在短距离上衰减很

严重），常用的陆上无线频率（如2.4GHz和833MHz）在水中不能使用。但是，与水下无线信号相比，声频信号（通常小于1MHz）可以传输更长距离。

水下无线传感器网络需要特殊的声频调制解调器和高级的水下收发器，在极端水下环境中 380
需要保护传感器。例如，水会侵蚀传感器。通信距离越远，接收器所需要的信号处理技术越复杂。以上方面使得水下无线传感器网络的设计成本较高。

水下传感器与密集部署的陆上传感器网络相比，通常是稀疏部署。

从陆上传感器读出的数据通常是关联的，而水下网络因其传感器间的长距离使得数据关联的情况不可能发生。

因为水下无线传感器网络使用声频通信需要更多能量，所以声频水下通信与陆上无线通信相比能耗更高。

陆上传感器节点的存储容量是有限的，因为水下信道可能是间歇的，故水下传感器需要具备将一些数据缓存起来的能力。

13.1.3　网络拓扑

通常用一个三维（而不是二维）网络拓扑检测或观察物体。注意，一个部署在海底的传感器网络不能执行海洋三维环境的协同采样。

如图 13-1 所示，在水下二维网络中，传感器节点在不同深度浮动以观测水下参数。结在绳子端的金属物体可以将传感器固定到海底，当然，传感器也可以装置在一个可以将其拉回水面的漂浮的救生圈上。通过调节传感器与锚之间线的长度，可以控制传感器所在的深度级别。

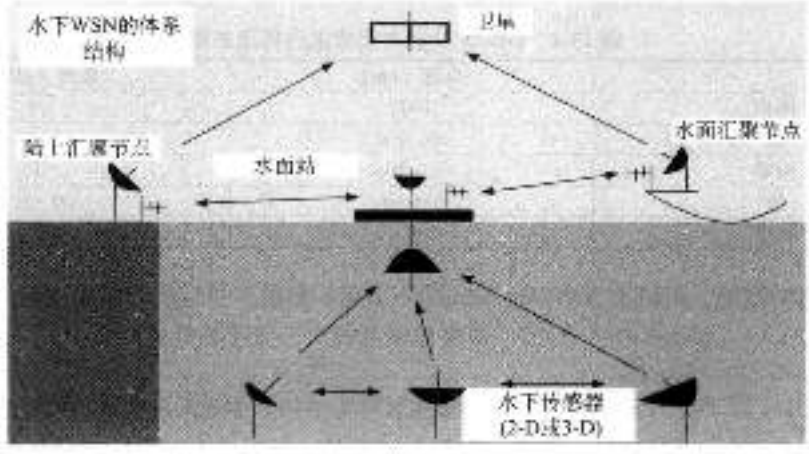


图 13-1　水下二维传感器网络架构 381

设计水下无线传感器网络协议需要考虑海洋流动（current）对传感器移动性的影响。


奇思妙想

请注意，大多数无线传感器网络（WSN）路由协议仅假设应用在一个二维平面结构上，而不是三维立体结构上。对于水下的情况，所有的传感器位于不同深度级别（垂直方向）。在每个深度级别上，大量的传感器形成平面拓扑结构。一个传感器需要与水平的和垂直的多个传感器保持通信连接。

13.1.4　声频信号传输

如上所述，在水中，无线信号的传输效果并不理想，但可使用声频信号。不过，路径损耗、噪声、多路径、多普勒扩展（Doppler spread）及高传输时延都会影响声频通信，这些因素说明声频信道具有时空易变性。

路径损耗，也称为信号衰减，是由声频能量转换成热能时能量被吸收造成的。长距离或高频率将带来更多信号衰减。波的散射和反射、折射、分散等多信号传输现象也会造成信号衰减。水的深度对衰减也有影响。

通信噪声通常是由船舶交通繁忙区域内的机器（如泵噪声）和船体运动（如船体由于污损不平滑产生的水动力噪声或螺旋桨空化噪声）导致的。环境噪声来自波水、水流、风暴、强风、雷雨等，这些也与地震和生物现象有关。

水平方向（即从海底到表面）的多路径信号传输（即一个信号从一个源发出后有多个传输路径）是显而易见的，灵敏的收发装置能利用多路径信号增强信号。

多普勒扩展是声能的分散，是波阵面的延展的结果。它随着传输距离的增加而扩展。有两种常见的几何扩展：深水通信的球面扩展和浅水通信的柱面扩展。

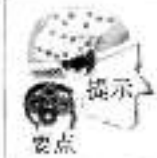
声频传输时延在水下与无线信道相比（音频传输速率大约1500米/秒）要高五个数量级。如此长的传输时延会减小系统的存吐量，这对于设计一个高效的网络协议是非常不利的。

这些因素限制了主要依赖信号范围和频率的水下声频信道的可用带宽。大范围声频通信（约万米）仅有一个几千赫（kHz）的带宽，小范围的声频通信（约100米）可能有一个大于
[382] 100千赫的带宽。不管是大范围或小范围声频通信，以上的因素都会导致低比特率。

表13-1　UW-A信道不同范围的可用带宽

	范围（km）	带宽（kHz）
超长	1000	<1
长	10～100	2～5
中等	1～10	～10
短	0.1～1	20～50
超短	<0.1	>100

基于链路的范围，可将水下声频通信链路分为超长距离、长距离、中距离、短距离、超短距离五类，表13-1［Melodia07］给出了不同传输范围水下信道的带宽。



与基于射频的陆上无线传感器网络不同，水下无线传感器网络因为长距离、可变的声频时延，需要一系列新的协议。水下的声波与人类听到的声音相似，在水下进行长距离的传输速度很慢，所以水下传感器需要特殊的无线接收装置——声频调制解调器来与其他节点通信。商用声频调制解调器非常昂贵。

13.1.5　水下传感器

水下传感器节点结构与通用无线传感网络节点相似。微处理器/CPU通过一个传感器接口电路板或海洋仪器与模拟水下传感器一起工作（如图13-2所示）。该处理器从模拟传感器接收数据（如水污染程度或金属水平），将数据保存在电路板内存中，处理后通过控制水下调制解调器发送给其他网络设备。通常电子仪器安装在一个侧面呈锥形的外壳保护下的构架上，从而
[383] 避免水的侵蚀。

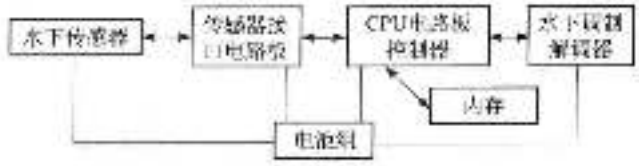


图 13-2　水下传感器内部结构

通过测量温度、密度、盐度、化学物、传导率、酸碱度、氧气含量、溶解的甲烷气体及浊度等参数，水下传感器能够评估水质。使用一次性的传感器可以检测神经毒蛋白质（如，蓖麻子含有的剧毒），从而确定是否是潜在的恐怖试剂。DNA 微矩阵（microarray）传感器可以检测自然微生物在数量和活动程度上变化。力感/扭矩传感器能同时测量几个力（forces）和力矩（moments）。

开发更便宜、更平面的水下纳米传感器（nano-sensor）是一种趋势。所有的水下传感器都需要周期性的清理机制以免被侵蚀或污染，（这会影响水下设备的寿命）。为了更好地理解海洋系统，研究者正在研究用于物理、化学及生物参数数据采集的综合传感器。

13.2　水下无线传感器网络协议栈

13.2.1　物理层

根据调制解调方案，使用频移键控（Frequency Shift Keying，FSK）调制方案是一种简单的方式。在 FSK 方案里，在连续脉冲间插入时间保护可以有效减少多路效应（multipath effect），也可以在频率间使用动态频率保护，使通信适应音频信道的多普勒扩展。

但是，FSK 带宽很窄，并且对于高数据率通信应用并不稳定。大范围、高吞吐量应用可使用相关调制技术，例如，差分移相键控（Differential Phase Shift Keying，DPSK）利用前后码元之间的关系对信息编码。

近来，正交频分复用（Orthogonal Frequency Division Multiplexing，OFDM）扩频技术已经成为水下通信的一个有前景的解决方案。OFDM 也称为多载波调制（multi-carrier modulation），因 384
其调制信号同时在多个载波信道中传输，因此每个独立载波的信号宽度比很多其他调制方案要宽。OFDM 系统在多路环境中表现稳定，有效地实现了长光谱。

除了调制设计，还需要解决其生物退层问题。例如，用于水下通信的廉价的发射器/接收器的调制解调设备需要开发。

13.2.2　数据链路层

该层解决多个相邻的传感器声频信道访问问题，因为信道访问是数据链路层设计的主要问题，所以该层也叫介质访问控制（Medium Access Control，MAC）层。当附近节点尝试同时访问信道时，如何确保没有冲突呢？这就需要使用一个有效的访问调度方案。

在水下无线传感器网络中，信道访问控制应该适应有限的带宽和高/可变的时延。频分多址（Frequency Division Multiple Access，FDMA）因为其窄带宽可能不会被使用。

如果使用时分多址（Time Division Multiple Access，TDMA），需要设计一个好的信道调度方案，以克服可变声频时延。因为 TDMA 需要一个通用参考时钟，所以基于 TDMA 的方案应基于精确的时钟同步。

基于竞争的技术，如使用准备发送（Ready-To-Send，RTS）和清除发送（Clear-To-Send，

CTS）来避免冲突的载波侦听多路访问（Carrier Sense Multiple Access，CSMA）并不实用，因为在水中 RTS/CTS 控制包传输的时延很长。声频时延的高可变性也使得预测传输的开始和结束时间非常困难，所以还会存在碰撞。

因为在水下信道中需要克服若干挑战，如声频的可变和长传输时延，非常有限的水下通信带宽，所以像在陆上传感器网络中那样在水下无线传感器网络中只使用 MAC 方案是不可行的。

码分多路访问（Code Division Multiple Access，CDMA）虽然需要更复杂的硬件设备，但却是一个有效的解决方案。对于由水下多路径引起的选择性频率衰减（frequency-selective fading），CDMA 具有鲁棒性，因为正交编码可区分由多路设备传输的同步信号。CDMA 也可减少因降低电量消耗和增加网络吞吐而导致的重传的包的数量。如果采用 CDMA，为了使传感器间干扰最小，需使用高自相关和低互相关的属性来设计通信访问代码。为了使网络效率最大化，
385 需要找出最佳的数据包长度。

13.2.3　网络层（路由层）

网络层旨在源和目的节点间查找一个有效路由。已经有很多面向陆上传感器网络的路由协议，但是对于水下传感器网络，因其时延过长，不能直接使用这些协议。

现有的无线传感网络路由协议主要包括三类：先发性、反应性及地理信息路由协议。

先发性协议一直维护一个包含每一个节点到其他节点的最新路由表，因此避免了由路由发现而造成消息延迟。但是当该协议用于水下网络时，会产生高昂的路由更新开销。

反应性协议仅在需要找到一个到目标节点的路由时，才会开始一个路由发现过程，其不需要维护一个“总是正确”的路由表。

地理信息路由协议基于传感器位置选择每个中继节点。全球定位系统（Global Positioning System，GPS）接收器可用于陆上系统以精确地估计传感器位置。但是，在水下，因为水下传感器网络不使用无线信号，故 GPS 接收器的使用效果并不好。

设计与声频信道的间歇性连通相关的、具有鲁棒性的路由算法是重要的。

13.2.4　传输层

水下传感器网络同样需要一个传输层来完成端到端（end-to-end）可靠传输，执行流控制和拥塞控制。最常用的传输层协议 TCP 使用基于窗口的依赖于对往返时间（Round Trip Time，RTT）的准确估计的流控制机制。遗憾的是，因水下往返时间的变化性很大，很难有效设置基于窗口机制的超时时间，故 TCP 不适用于水下环境。

为了实现可靠的数据传输，水下传感器网络传输层协议应能处理以下问题：高传输延迟、窄带宽、能效、高误码率及高易变的网络拓扑。运用如下原则可设计出用于水下环境的有效传输层方案，如盲区、最小能耗、基于速率的包传输、对局部拥塞的及时响应及可靠性。下面将讨论更多细节：

- 路由层若干参数（如每跳的时延）对处理通信盲区（该区域信号非常微弱）非常有帮助。

386
- 因水下传感器是电池驱动的，需要考虑最小能耗。
- 基于速率的包传输因能进行更准确的拥塞控制，故优于基于窗口的方案（如 TCP）。
- 对局部拥塞的及时响应能立即适应局部情况，在拥塞的情况下应减少响应时间。因此，与其依赖基站，不如由中继节点及时对局部拥塞进行处理。逐跳的（hop-by-hop）可靠性控制优于端到端控制。

13.3　介质访问控制设计实例

如前所述，因存在能量限制、长传输时延、低数据传输率及水下环境同步问题，设计水下传感器网络的介质访问控制协议是个挑战。陆上传感器网络的 MAC 协议并不适合水下声频通信介质，因为每 1.5 千米距离会产生 1 秒的传输时延。本节将使用［Min07］的例子说明如何设计适合水下环境的有效 MAC 协议。

首先回顾一下该领域的相关工作。SeaWeb［JRice00］水下网络项目中使用了频分多址技术，但是仅拥有有限的可利用带宽，且效率不高。SeaWeb2000［JGProakis01］使用具有 RTS/CTS 握手/交换机制的载波监听多路访问/冲突避免（Carrier Sense Multiple Access/Collision Avoidance），但使用 RTS/CTS 包非常耗费能量。



我们已经了解了一些用于陆上无线传感器网络的 MAC 方案，如 S-MAC。这些协议将能效放在首位考虑。它们试图让传感器长时间处于休眠状态，并减少信道访问调度复杂度以进一步降低能耗。但是在水下无线传感器网络中，MAC 设计首要考虑的是对长时间可变声波延迟的适应性。

Min 和 Volkan［Min07］设计了 UWAN-MAC 协议，其基本思想如图 13-3 所示。该图解释了如何完成一个局部同步调度，该方案即使在长时间、不确定传输延迟存在的情况下同样适用。

1）侦听周期确定：如图 13-3 所示，假设传感器 A 在其周期开始时向邻居节点广播一个 SYNC 包（阴影的矩形），然后进入休眠状态（为节能而关闭收发器电路），这个 SYNC 包表明 A 的通信循环周期 T_A 的开始。

387

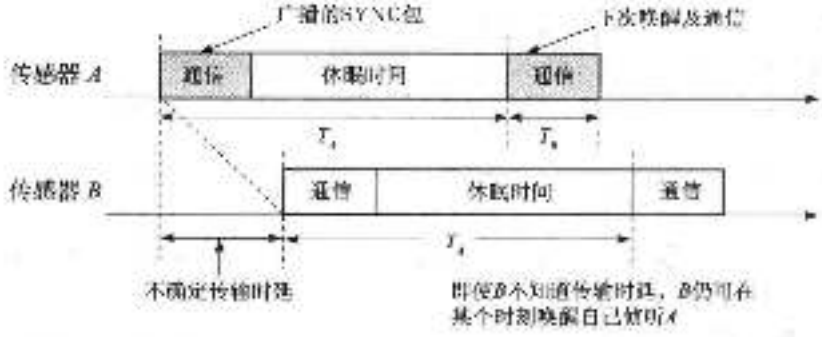


图 13-3　UWAN-MAC 协议的基本思想

假定传感器 B 是 A 的邻居，当 B 加入该网络后，将首先尝试获取 A 广播的 SYNC 包，从而完成与 A 的通信同步（白色矩形表示 B 接收到来自 A 的 SYNC 包）。

B 没有采用固定的睡眠唤醒工作模式，而是遵循了与 A 相同的周期性传输休眠工作模式，因此 B 能够在 A 的每一次周期性传输前适时地唤醒以对 A 进行侦听，而无需知道 A 与 B 之间的水下声波传输时延。（当然，以上机制需要假设相邻循环周期间的传输时延相对固定，并且每个循环周期内时钟漂移不明显。在循环周期不太长的情况下，这种假设是非常合理的。）

应将上述的 SYNC 传输协议用于任何相邻传感器（如 A 和 B），该调度算法因为不需要绝对时钟信息，故不需对传感器的时钟做任何调整。

2）发送起始时间确定：网络新扑控制协议帮助 MAC 协议使一个节点保持与邻居节点的联

系。即使使用上述方案确定了侦听时间，初始传输时间也是由每个传感器随机独立地选择。但是，一旦某个节点选择了一个确定的传输开始时间，节点将坚持在下一循环周期的同一时间传输数据。只要循环周期远大于数据传输持续时间（如图 13-3 所示，$T_s \gg T_a$），信道访问冲突的可能性将变得很小。

在 UWAN-MAC 协议中，每个节点将其存储的邻居节点列表（通过邻居发现协议）与节点
388 列表（包含已接收到信号的节点）比对，比对后产生“缺失节点列表”。然后在下一个传输周期时，在数据包的头部发送“缺失”邻居节点列表。

在正常情况下，每个节点都在其发送的 SYNC 包首部发送其循环周期（即图 13-3 的 TA）。接收到 SYNC 信息的节点改变其当前的循环周期。其邻居节点通过解码出修改了的 SYNC 信息进而改变它们的唤醒时间。如果一个节点，（比如 B），在上述修改过程中与 A 节点失去联系，B 将使用缺失节点列表，将 A 加到其邻居列表中。

T_s（循环周期）实际上包含三个部分：1）数据传输（包括发送和接收）；2）空闲侦听；3）休眠状态。数据传输完成后，一个节点不是立即进入休眠状态，而是进入空闲侦听模式。在侦听模式，节点仍然低耗运行。如果它监听到什么，将进入接收模式。

持续侦听也可用于获知新的节点加入。侦听的持续时间长度需要小心选择：因空闲过程消耗能量，就过长持续时间会降低协议能效，但是过短的持续时间可能获取不到足够的新节点信息。

处理节点加入：当一个新的节点加入网络时，只要它收到邻居节点的信息，新的节点可发送一个 HELLO 报文给那个邻居节点告知其传输时间表。这种 HELLO 报文应在数据传输阶段发送。通常，一个传感器将其传输时间间隙平均分成 M 个时隙（time slot），一个 HELLO 报文的传输时间对应一个时隙。当一个已存在的节点周围同时出现多个新的邻居节点进入网络时，新节点会随机从 M 个时隙中选择一个以传输 HELLO 报文，这样的随机 HELLO 传输方式有效避免了来自不同新节点的 HELLO 报文可能产生的冲突。

处理节点失效：当信道条件不佳或发送方出现故障时，节点可能不会在预定唤醒时间收到信息。如果接收节点没有在预定时间收到发送节点的信息，那么接收节点把发送节点放到其缺失节点列表中（如上所述）。

处理可变声频传输延迟：如图 13-3 所示，各个循环的声频传输延迟是固定的，这是一种理想的情况。真实情况是，声频传输延迟随信道波动（传感器运转或水流的影响）而改变。可以假定节点知道最大传输延迟。例如，对于密集部署的水下无线传感器网络，110 米距离的最大传输延迟约为 70 秒。

可用一个新的 MAC 协议来处理可变延迟：每个节点 i 在其传输持续时间的两端放置一个“保护时间”（guard time）。为了降低包传输冲突率，可用一种完全局部化的方式选择每个节点的保护时间。

图 13-4a 给出了一个没有使用保护时间的例子。因可变传输延迟，发生了接收 - 接收（re-
389 ceive-receive）冲突。

为了避免该冲突，节点 A 或 B（这里为 A 添加了保护时间）重新选择传输开始时间以避免冲突。图 13-4b 提供了一个可行的解决方案，设 $\tau_1 < \tau_2 < \tau_3$，节点 A 由于选择了新的传输开始时间而使冲突不再发生。

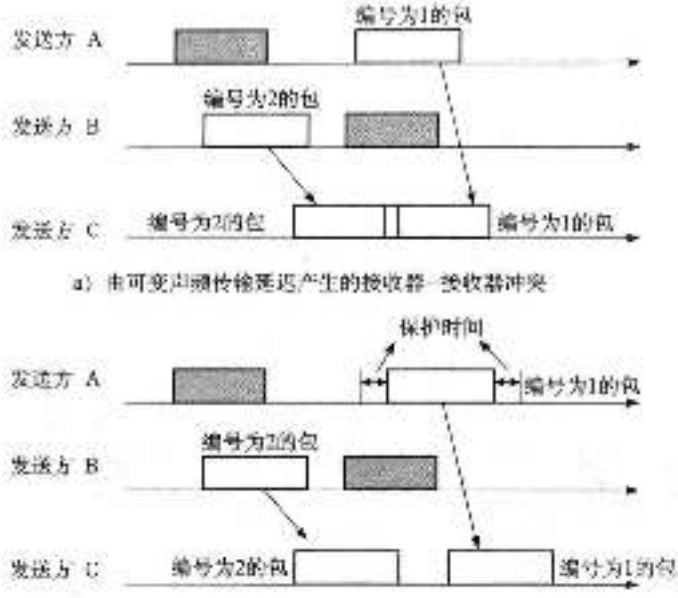


图 13-4　传输延迟冲突和具有保护时间的冲突避免策略



"保护时间"是个好的理念。在无线多信道通信中，通常在通信带（一个宽带宽）中添加一个"保护信道"（guard channel）避免相邻带的信号干扰。保护时间可使传输时间更具"柔性"（flexible），从而避免接收器－接收器冲突。

13.4　路由设计实例：基于矢量的转发协议

水下传感器网络路由协议要满足两个需求：1）能效的需求（因为传感器是电池驱动的）；2）水下传感器的移动（因为水流的冲击）。因此路由协议需要处理节点的移动问题，矢量转发协议可满足以上需求。 390

水下传感器网络路由的方向是从海底到海面，因此，使用一个路由矢量代表这个路径。VBF 的设计思想如图 13-5 所示，节点 S_1 是源点，S_0 是接收节点，路由矢量是 $S_1S_0^2$，路由管道（routing pipe）是预先控制的半径为 W 的管道，VBF 的每个节点不需要状态信息，其网络的大小是可伸缩的。为使网络更有效率，仅转发路径上的节点参与转发过程，非转发路径上的节点不参与转发。

VBF 的任一数据包均含有发送节点、目的节点及转发（中继）节点的位置信息。当一个节点收到一个数据包时，通过两种方式计算它的位置与传送装置（节点）的关系：1）传送装置的距离；2）信号的到达角（Angle Of Arrival，AOA）。接收包的所有节点递归地计算它们的位置。

如果一个节点确定其离路由矢量足够近（即能被纳入路由管道），它将其位置加入到包中，更新数据包，然后转发到下一个节点。否则，它丢弃数据包。传感器网络的那些满足包转发条件的节点形成一个路由管道（如图 13-5 所示）。

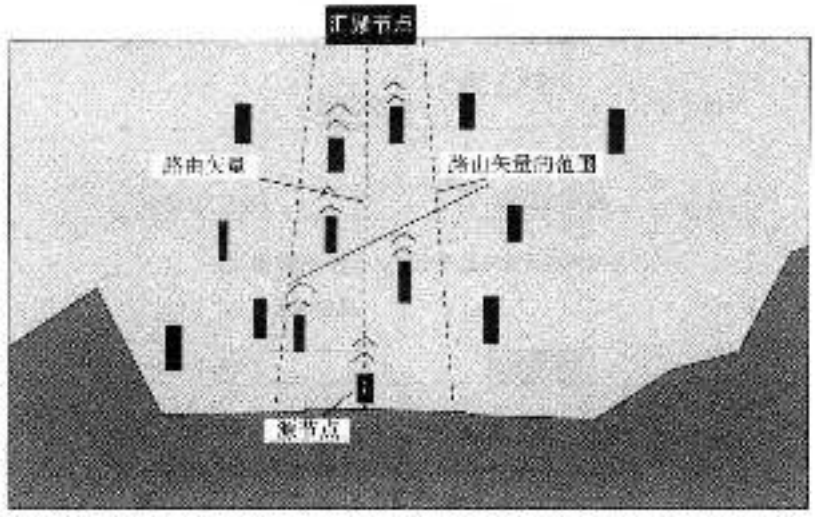


图 13-5　矢量转发协议的场景

从发送节点 A 到接收节点 B 的路由矢量通常表示为 $\overrightarrow{AB}$。在一个 3D 空间，如果 A 的坐标为 (A_x, A_y, A_z)，B 的坐标为 (B_x, B_y, B_z)，那么矢量 $\overrightarrow{AB}$ 可以表示为 $(B_x - A_x, B_y - A_y, B_z - A_z)$。

391

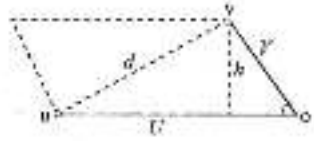


图 13-6　矢量计算图

假设点 $V = (v_x, v_y, v_z)$，且另一点 $U = (u_x, u_y, u_z)$，点 V 与 U 之间的距离可表示为：$d = \sqrt{(v_x - u_x)^2 + (v_y - u_y)^2 + (v_z - u_z)^2}$（如图 13-6 所示）。

VBF 路由协议是以协议包（或数据包）交换为基础的，每个协议包由三个位置字段组成：DP、TP、FP，分别是发送、目的、转发节点的坐标值。当一个数据包到达 TP 的指定区域时，包被入一个由范围字段控制的区域中。

任一协议包中有一个字段叫半径，是预先定义的阈值。传感器节点使用它确定是否足够靠近路由矢量，如果被考虑近路由管道，该节点将用于包转发。

VBF 能执行两种传感器数据查询：1）位置相关（location dependent）。汇聚节点对某一区域感兴趣时，该查询可提供区域位置信息；2）位置无关（location independent），通过使用位置无关查询，汇聚节点可以获取网络中某一事件的发生情况。例如，汇聚节点可以获取网络中是否存在金属污染的信息。



奇思妙想

路由管道是一个有趣且有用的理念。在某些情况下，依赖一系列单个节点完成多跳无线通信的普遍性不够，因此，研究者提出了一个路径上由密集节点组成的管道概念。因为任何管道内的节点都有助于转发数据，故管道节点越密集，路由方案越具有鲁棒性。另一方面，由于要维护路由管道，这样的路由健壮性会使路由更加复杂。

392

13.5　硬件原型设计

本节将讨论由［Hu2009e］（本书作者）设计的一个水下传感器网络原型。多数商用的水下通信系统［DSPComm08］是为长距离通信（几千米的链路距离）设计的，其使用的调制解调器传输速率接近 1～40kbps，但因成本太高而在应用方面存在限制。

在［Hu2009e］的设计中，传感器节点的大多数功能是由软件定义的，以使在多任务情况下以较低成本对平台进行重配置。水下传感器硬件相对简单且仅提供如下功能：放大输入输出信号、为各种环境下的传感器和探针（probe）提供信号调节。其余的节点功能由软件定义，包括调制和解调。

该系统使用一个嵌入式处理器满足网络连接和路由选择的需要，并使用一个数字信号处理器（Digital Signal Processor，DSP）在软件层面上完成调制和解调。图 13-7［Hu2009e］显示了 DSP 芯片与其他元件（如发送器（Tx）和接收器（Rx））的连接。

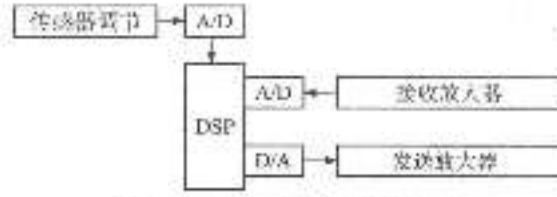


图 13-7　硬件与 DSP 的相互作用


案例研究

使用扩音器充当水听器（hydrophone）是一种简单方案，而且满足原型的设计目的。但是，却不适合商业应用。仍需设计定制的声频调制解调器以完成水下通信。

[393]

13.5.1　硬件设计

水听器（用于电信号与声频波的互相转换）的价格近 1000 美元，即使为小范围应用设计的小型变换器［Transducer08］的价格也同样不菲。因此，需要一个相对便宜的产品替代商业水听器。为了实现原型，采用小型外壳防水的扩音器。在要求的频率范围，这些扩音器可以产生足够清楚的音调，作为发送/接收水听器使用（参见图 13-8）。



图 13-8　充当水听器的扩音器

［Hu2009e］使用两种类型的传感器，即 pH 传感器和温度传感器。信号经过调整后，为模数转换器（Analog-to-Digital Converter，ADC）提供一个有效的信号。

pH 放大器的增益恒定（constant gain）为 2.4，由一个非反相放大器（non-inverting amplifier）组成。因 pH 传感器通常的阻抗（impedance）为 50MΩ，所以 pH 放大器应能接收高阻抗源。水下节点模数转换器仅接收不超过 5kΩ 的信号源。

温度放大器的增益恒定为 2，覆盖了模数转换器的模拟范围，同时提供很好的精确性。温度放大器使用惠斯登电桥（Wheatstone bridge）和一个反相差动放大器。温度传感器使用一个 10kΩ 的热敏电阻。

为了使相邻水下节点执行声频通信协议，使用一个微控制器控制所有传感器。该微控制器是一个定点数字

信号处理器（Digital Signal Processor，DSP），该处理器内置256kB的支持直接存储器存取（Direct Memory Access，DMA）的内存。采用定点处理器是因为浮点处理更加昂贵，操作起来需要更多电能。主板上有一个足够用于信号采样和信号处理的100kB的RAM。DMA系统允许采样信号直接存入内存，因此不需要使用昂贵的CPU周期将样本从ADC移动至内存。DMA允许在采样的同时对信号解调，所以在长时间的解调计算中不会有样本丢失。

微处理器插在一个接口板上，板上承载若干可直接访问处理器的辅助设备，包括若干数字I/O端口、一个基于数字式电位计的数模转换器及一个10位/12位的模数转换器。这些辅助设备与CPU结合在一起，真正地使系统节省了时间和损耗。

装配的水下接口板（包括DSP和声频通信模块）如图13-9所示，该图中没有标示出模拟传感器。

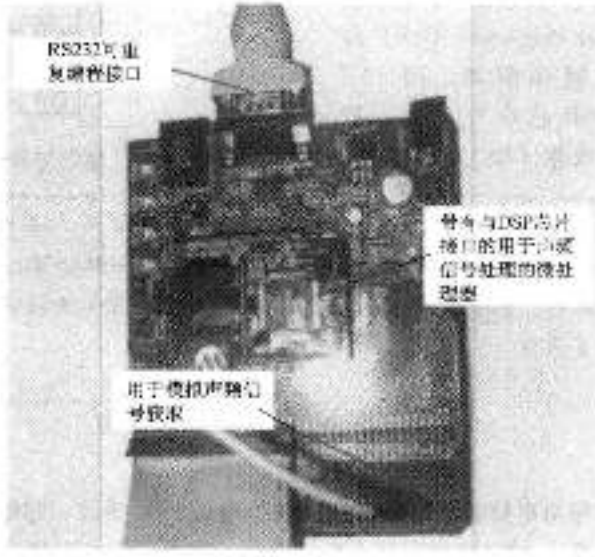


图13-9　已装配微控制器和声频收发器的水下节点

394
/
395

13.5.2　软件设计

系统的软件分为两类：与发送相关和与接收相关。接收端的软件结构如图13-10所示，包括模拟传感器数据过滤组件和声频解调组件。发送端的软件结构如图13-11所示，包括调制组件和循环冗余码校验（Cyclic Redundancy Check，CRC）错误控制组件。

13.5.3　系统测试

水面传感器（即汇聚节点）通过一个2400波特（baud）的RS-232连接线与一个电脑连接，因此水面节点不使用传感器。水下节点由pH传感器和温度传感器构成。水下节点每30秒钟轮询（poll）一次其传感器查看是否有信息。一旦检索到传感器数据，就会构建一个包并计算该包的CRC码。

当水下传感器将构建的数据包发送到汇聚节点后，汇聚节点必须向水下节点发回一个确认信号（ACK）。此时水下节点知道该包已经接收成功，无需重新发送。

如果CRC显示包损坏或包根本没有到达汇聚节点，接收器将发送一个否定应答信号。最后，发送器因等待确认信号而超时，将重发包。错误恢复顺序如图13-12所示。这称为停止等

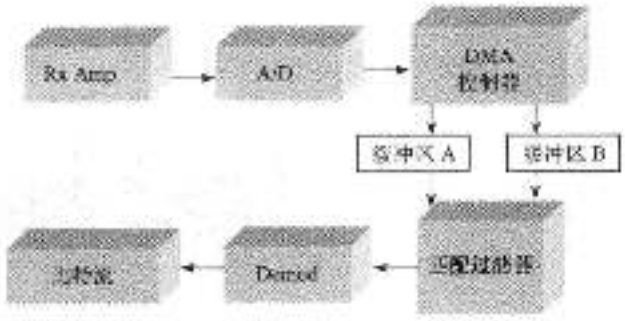


图 13-10　接收端软件框图



图 13-11　发送端软件框图

待自动重发请求（Automatic Repeat request，ARQ）机制。

实验室的测试设备如图 13-13 所示。系统达到了 15.625bit/s 的理论比特率，平均比特误码率（Bit Error Rate，BER）为 0.091，因为无线网络没有最完备的错误检查机制，所以 CRC 不能避免所有错误。这些问题可在图 13-14 中看到，即 pH 和温度结果的峰值点。这些点是孤立点（pH 值已经远超出 14，应是位错误造成的）。

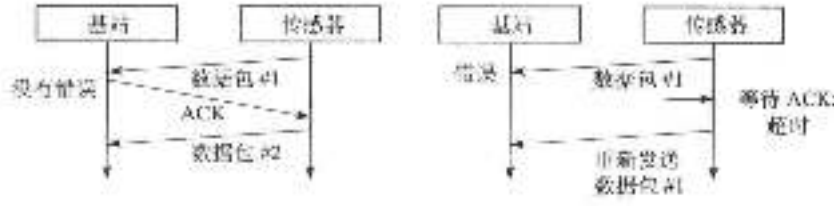


图 13-12　ARQ 交互

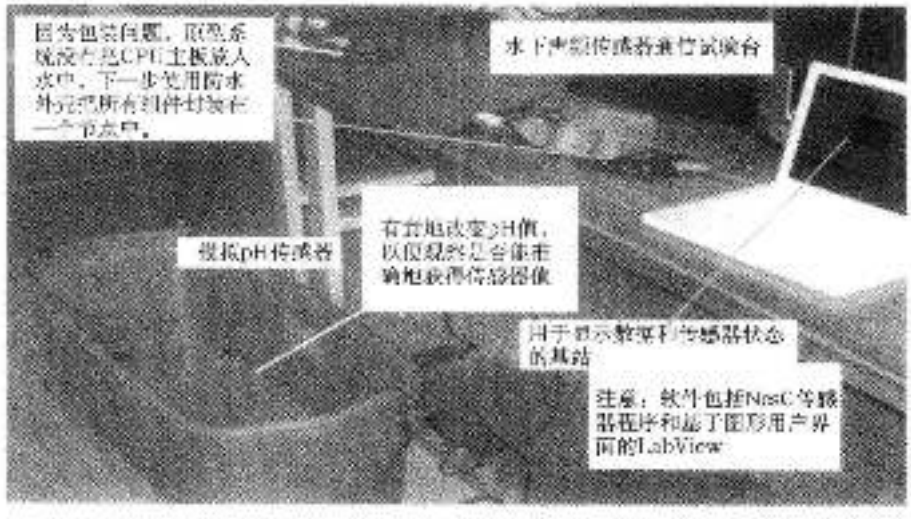


图 13-13　实验室的测试设备

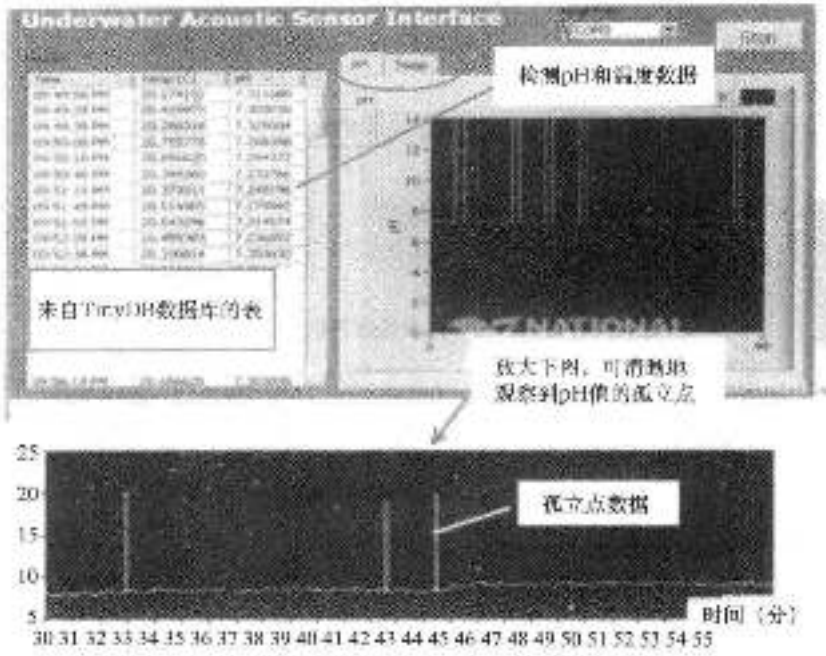


图 13-14　水下传感器数据收集图形用户界面

问题与练习

13.1　解释水下和陆上传感器网络的不同。

13.2　修改适用无线传感器网络的 MAC 层协议（参见第 3 章）用于水下无线传感器网络，需要做哪些改进？

13.3　可以直接将第 4 章的路由协议用于水下传感器网络吗？给出理由。

398　13.4　参照 13.5 节，画出水下传感器网络系统架构图（硬件/软件）。

视频传感器网络

14.1 引言

当前无线传感器网络可使用上百个不同的模拟传感器收集环境数据。视频捕捉获取了其他传感器无法收集的宝贵信息，可将视频传感器与其他类型的传感器融合在一起可提供多媒体数据，也可在视频监控应用中单独使用视频传感器。

使用视频传感器是让人兴奋的，但是视频数据需要极大的存储空间和无线带宽。为了克服这些问题，要使用各种资源节约（resource-conserving）技术保证视频传感器捕捉和传输视频数据的能力。

接下来，将给出视频传感器网络（Video Sensor Network，VSN）的若干应用，以及与VSN相关的资源需求［Feng05］：

水下勘探：海洋专家可使用VSN观察和研究水下沙丘的发展，也可使用数字图像处理技术分析沙洲的演变过程。无线应用的特点要求使用能自给自足的视频传感器。例如，可能使用太阳板或其他动态方法（水的流动）产生传感器电能。VSN协议是非常节能的，为了减少能耗，可间歇性地建立网络连接，也就是说，只有数据收集是持续进行的。虽然视频传感器在一个低耗状态下运行，但网络协议应传输最重要的视频数据。

环境监测：对于一些建筑/露天的监测应用，可能事先不知道哪些数据是重要的。为了提高网络的可伸缩性、减少网络流量并节省存储空间，视频传感器应尝试尽可能地过滤无用数据。例如，在一个区域内，使用图像比对技术可发现任何新进入的目标。如果视频传感器不能监测到图像改变，那么将不发送视频数据。

应急响应系统：可部署基于视频的传感器网络用于应急响应应用，视频传感器在某个时间段（即紧急情况的持续时间）能捕捉和传输高质量视频。VSN系统应具备响应快、低耗的自适应特征，为应急响应人员提供整个事件过程的视频数据。

在以上三个应用中，包括如下通用任务［Purushotham07］：

1）**目标检测**：VSN的一个目标是在观察区域检测熟悉或新的目标或场景的改变。例如，一个动物习性监测应用应能检测一个动物何时进入或离开某个区域；一个建筑物安全系统应能检测入侵事件，如一个人正在进入某区域。为了执行目标检测，视频传感器可使用许多已提出的目标检测算法。这些算法应花费最少的时间检测每个新进入安全区域的目标。

2）**目标识别**：目标检测是发现是否有一个新目标已经进入区域，而目标识别是为了认出目标到底是什么，例如，当检测到一个新目标后，需要确定它的类型（如正常人员或敌人，斑马或鹿）。识别的过程能发现该目标是否是感兴趣的，识别通常通过将获取到的视频/图像与数据库中目标图像比对来完成。好的匹配算法能快速找出目标对象。

3）**目标跟踪**：当识别出感兴趣目标后，可能在目标移动时需要跟踪该目标。目标跟踪通常首先确定当前的位置和目标的轨迹，然后当目标移出一个摄像传感器的可视范围进入另一个传感器范围时，将跟踪任务从一个传感器移交给另外一个传感器。

为了执行以上三个任务，需要设计硬件和软件以捕捉需要的信息。



当前已提出许多目标检测/识别/跟踪的方案，通常需要机器学习的知识才能有效地对目标分类。神经网络是一个传统的执行目标识别的方案。目标跟踪通常需要准确的目标定位技术。

400

14.2　Panoptes

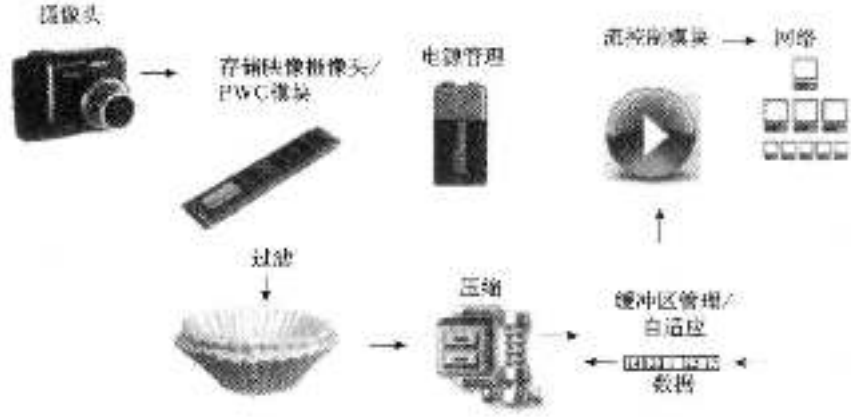
[Feng05] 已经详细介绍了视频传感器硬件的设计。在设计和选择一个视频传感器节点时，需考虑电源，内存空间及 CPU 速度方面的需求。虽然基于 Intel 的 StrongARM 的个人数字助理（Personal Digital Assistant，PDA）在 MIT 和 ISI 的一些重要研究项目中被广泛使用，但它仍不能满足视频传感器设计的低耗要求。此外，传统视频传感器设计存在诸多缺点，如下所示：

1）**I/O 带宽有限**：许多嵌入式传感器使用基于 PCMCIA 的设备，通常能耗很高。一些设备使用 USB 接口，遗憾的是，低耗、小的视频传感器对 USB2.0（455Mb/s）的支持不是很好，需要大量的处理器存储进入的数据。

2）**缺少对浮点数据的处理**：许多嵌入式设备使用 StrongARM 处理器和 Xscale 处理器，但是，这两种处理器不支持浮点运算。而视频压缩算法是基于浮点运算的。

3）**存储带宽**：传统设备对存储带宽并没有做优化，而对视频传感器来说，存储带宽对大量的图像视频数据的处理是很重要的。

Panoptes 视频传感器 [Feng05] 可克服上述缺点。它使用 Linux 操作系统（因其设备控制机制简单和系统修改的灵活性）。在 Panoptes 中，视频感知任务通过一些组件完成，包括捕捉、压缩、过滤、缓存、自适应及流控制。Panoptes 系统若干重要的组件如图 14-1 所示。接下来，将简要介绍这些组件。



401

图 14-1　Panoptes 传感器软件的组件

14.2.1　视频捕捉

[Feng05] 使用 Philips 网络摄像头的视频接口与 Linux 相连。Linux Kernel 解压视频数据，然后发送到用户空间，通过存储映像访问，解压后的视频数据（大于 10 帧/秒）便可使用了。当读入一幅视频帧后，过滤算法和压缩器将进一步处理它。

14.2.2　视频压缩

为了减少存储量和网络流量，在空间上和时间上压缩视频帧是必要的。Panoptes 可以使用多种压缩格式，如 JPEG、差分 JPEG 及条件补充压缩格式（conditional replenishment compression format）。虽然 JPEG 本身不能完成对数据的时间压缩（temporal compression），但是能减少数学计算代价（与将数据转换成 MPEG 相比），因此节省了传感器的能耗。因为压缩是由 CPU 来执行的，因此视频的质量和压缩的水平依赖 CPU 的处理能力。

14.2.3　数据过滤

视频传感器必须有能力在传感器层面过滤数据，从而减少网络信息流量。注意，在传感器层面过滤数据（而不是在网络层面）可减少总体网络设计的开销。对于许多应用（如视频安全监控），传感器应能过滤不感兴趣的数据，仅压缩和传输想要的数据（如新的面孔）。对于环境观察来说，过滤能产生一个时间流逝图像（time-elapsed image），从而只压缩图像数据而忽略图像背景，使要传输的数据大幅减少［Stockdonoo］。

用户可以设置 Panoptes 的过滤器，决定哪种视频数据应被过滤掉。因为基于离散余弦变换（Discrete Cosine Transform，DCT）的视频压缩成本相对较高，所以应当运行低复杂度的能减少压缩帧数目的过滤算法。

14.2.4　数据缓存

数据缓存是视频传感器的关键组件。使用某种基于优先权控制的方案来缓存视频数据，可以确保在出现网络拥塞事件或运转中断之前将所有重要的数据传输出去。在一个缓存方案中，如果视频传感器的缓存已满，有效的优先权控制机制应能确定首先丢弃哪些数据。Panoptes 使用一个基于优先权的流机制来支持视频传感器。

402



视频传感器主要需要两方面的知识：1）数字图像/视频处理（如数据压缩、目标检测及跟踪）；2）网络多媒体处理（如缓存和 QoS 方案）。分布式数据压缩是一个将网络协议和图像处理结合的好例子。

14.3　Cyclops

Cyclops［Rahimi05］把上述的两个方面拆分为：1）局部图像捕捉（使用传感器）；2）无线网络通信。Cyclops 拥有支持高速数据传输的可编程逻辑存储单元。它也有专门的微处理器（MCU）来充当传感器到网络的接口。通过使用这些硬件组件，Cyclops 把高速数据传输与嵌入式 MCU 的低速性能区分开来。区分的好处是使费时的图像捕捉和分析在本地完成并且不受传感器通信影响。这种区分处理的机制特别适用于经常需要执行异步事件（如 MAC 层无线接入）的网络化传感器节点，当然它需要遵守严格的延迟约束。

但是应注意到，Cyclops 的一个设计目标是使能耗最小化，以满足大范围部署和延长的需求。这个目标使平台的计算能力和成像尺寸面临严格的约束，因此，仅有某些特定类型的应用使用 Cyclops。当应用需要高速处理或高分辨率图像时，Cyclops 的效率不高。


奇思妙想

总之，Cyclops有两个显著特点：1）高效节能的架构；2）将视频处理与网络传感器通信分离。为了实现这两个优点，Cyclops利用并行计算避免持续的传感计算，使用计时时钟资源的按需控制方案以减少能耗，利用自动检验策略使子系统运行在低耗状态。

Cyclops节点硬件包括成像器、MCU、复杂可编程逻辑器件（Complex Programmable Logic
403 Device，CPLD）、外部SRAM及一个外部闪存（如图14-2所示）。MCU通过设定传感器参数、发送捕捉视频帧指令、告知何时对图像计算来控制视频传感器。

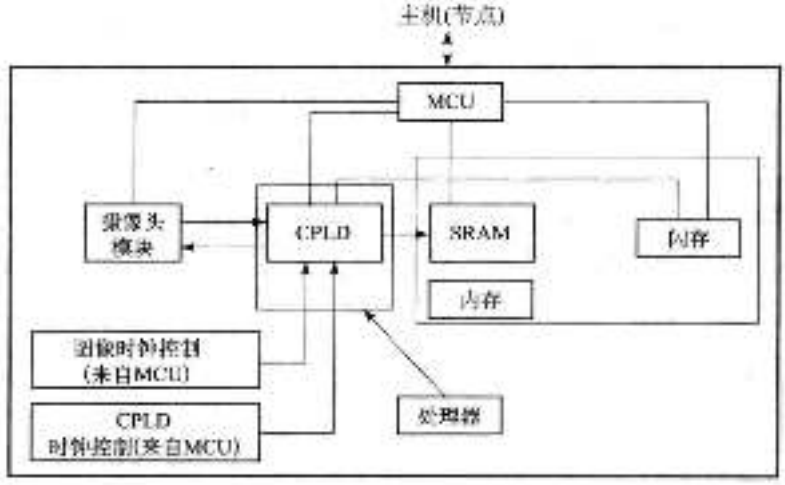


图14-2　第一代Cyclops硬件架构

CPLD产生用于图像捕捉的高速时钟同步信号和内存控制指令，MCU同CPLD协同工作，既为图像捕捉提供了低耗的处理方式，又实现了高速时钟同步访问。当然CPLD也能执行一些图像处理任务，如视频捕捉时的背景消减及触差计算。该设计保证了以最经济的方式使用硬件资源，因为在视频捕捉时CPLD已经同步了。如果MCU在某个时刻不需要CPLD同步或处理，MCU可发送一个中断命令去停止CPLD（这样可以节能）。

Cyclops节点的另一个重要特点是使用外部SRAM，当MCU的内置内存不够时，外部SRAM是有用的。外部存储可为图像存储、计算和处理提供足够的空间。捕捉和计算的时候，外部存储允许直接访问存储资源。当不需要额外存储资源（内存足够使用）时，SRAM保持睡
404 眠状态。Cyclops也将数据永久存储在外部闪存中，以用于模板匹配。

总线结构：MCU、CPLD和存储模块均共享同一地址总线和数据总线。该特点使硬件组件间实现了快速方便的数据传输。这样的通用总线结构需要特殊的组件间同步数据访问机制。

Cyclops的每个模块工作在几种用电状态。当前的状态越低耗，唤醒至活动状态所消耗的电量越多。因此，应用不应仅简单工作在某个状态，而应考虑节电（进入低耗状态）和用电（模块回到正常耗电状态）的平衡点。

Cyclops还有一个异步的触发命令电路，可作为一个寻呼信道来执行事件触发，将应用从睡眠状态快速唤醒。例如，触发电路能与一个红外线传感器、一个麦克风或一个磁感应器相连接，当检测到物体运动或声音时触发图像捕捉。

控制Cyclops平台的固件应支持自动释放到最低用电状态，允许长时间图像计算，支持

MCU 和 CPLD 同步访问共享的资源（如 SRAM）。这些需求表明，一个以网络为中心的方法不适用于异步事件。实际上，Cyclops 需要一个连续的方法执行连续的图像捕捉和处理。该方法中，触摸伴随着一系列的并发性不强的长时间同步操作。

Cyclops 固件使用 NesC［Gay03］语言编写，运行在 TinyOS 操作系统环境下。TinyOS 允许以易于连接的模块形式使用抽象功能。同时，操作系统提供用于事件定时控制的调度程序和服务。

14.4　视频传感器网络定标

如 14.1 节所述，VSN 执行若干常见任务，如目标检测、目标识别及目标跟踪。目标检测能检测在视频传感器的范围内出现一个新的目标，目标识别确定目标是什么，目标跟踪使用多个视频传感器持续地跟踪目标。

VSN 在执行上述三个任务前，要在初始设置时定标（calibration）。VSN 定标需确定每个摄像头的位置及方向（如角度），摄像头的位置是在一个参考坐标系中的 3D 坐标，而方向是指摄像头镜头的朝向。仅当获得了这两个定标的统计参数时，才能知道每个视频传感器可观测到的范围。

通过使用定标信息（位置和方向），整个观测区域可被划分成若干子区域。系统能够计算出至少覆盖了 2 个传感器的子区域。另外，相邻传感器间的关系也可通过定标信息确定，如确定相邻摄像头的重叠观测区域。

确定重叠的观测区域后，系统可以对每个传感器所负责的感知区域进行划分。系统也可通过三角测量的方法计算出目标或事件发生地点的位置。当目标移出一个传感器视野后，可将跟踪职责交给其他传感器。

摄像头的定标在计算机视觉领域是一个热门的研究方向，许多技术能准确地估计摄像头的位置和方向（如［Horn86］和［Tsai87］）。通常，这些技术假定事先知道一些地标的坐标，通过使用这些地标的投影结合光学的原理，从而确定一个摄像头的坐标和方向。

不过，不能简单地使用这些基于摄像头的定标方案，因为视频传感器本身有严格的计算限制和能耗约束。视频传感器有限的定标能力使目标的位置确定变得不精确。

当然，可将地标的概念借鉴过来用于 VSN 的定标，但是使用地标的代价高昂（与传感器相比），所以不能在 VSN 中应用地标。如果不使用地标，那么可以在每个视频传感器中安装一个定位装置，如 GPS 和方向数字罗盘［Sparton08］。通过这两个装置能直接确定节点的位置和方向。虽然这个想法相当有用，但目前 GPS 系统是非常昂贵的（与微传感器相比）且定位不是那么准确（误差在 5～15 米）。另外一个方案是使用相对准确的基于超声波的定位和测距技术［Priyantha00］。但是在低耗的视频传感器里使用额外的硬件，无疑会增加成本和能耗。

因此，准确的定标对资源受限的没有基础设施支持的无线传感器网络来说是一个挑战。不使用已知地标或任何定位技术，可以对视频传感器定标吗？

如果通过绝对位置的方法去实现精准定标的成本很高，那么在相邻节点间确定相对关系可能是唯一的可行的选择。这也带来了以下问题：1）如何在不使用已知地标或定位基础设施的情况下，确定视频传感器间相对位置和方向？2）这些技术的准确性如何？3）这类基于邻近节点相互关系定标的应用的性能如何？

考虑一个随机部署视频传感器的无线网络，任一传感器节点由一个低耗的成像传感器（如 Cyclops）和一个射频（RF）节点（如 Crossbow 节点或 TelosB 节点）组成，不使用 GPS 硬件。我们的目标是确定一个称为 k-overlap 的参数，表示一个有 k 个视频传感器重叠的可视区域的分

数值。假定有一个参考目标出现在环境中的任何位置，假定事先知道参考目标的维度和视频传感器的焦距。

Kulkarni［Purushottam07］介绍了一种近似估计技术，以确定重叠的程度和摄像头传感器的重叠区域。

14.4.1　确定重叠的程度

为了确定 k-overlap 的值，先分析一个一般情况（即 k 是任意的值）。1-overlap 表示一个传感器的可视区域的分数值，该区域不与其他传感器可视区域重叠。2-overlap 表示两个传感器重叠区域的分数值，以此类推。

如图 14-3 所示，A1 是一个传感器的可视区域，A2 是一个两个摄像头都可观察到的区域，A3 是图上所有三个摄像头可观察的区域。很显然，一个传感器的所有 k-overlap 区域的并集构成该传感器的总的可见范围（即一个传感器的 k-overlap 值的和为 1）。

下一步将确定每个传感器的 k-overlap 值，此处 $k = 1 \ldots n$，n 是系统的传感器总数。

14.4.2　估计 k-overlap 值

假定已随机部署若干参考目标，每个参考目标的位置标为做一个参考点。假定环境中的参考点统一分布，视频传感器开始对环境采集图像，处理图像后，就可看到每个摄像头能观察到的参考点。

假定一个摄像头 i 能看到的总的参考点数为 r_i，r_i^k 表示 r_i 中那些可同时为 k 个摄像头所看到的参考点数。摄像头 i 的 k-overlap 值定义为：

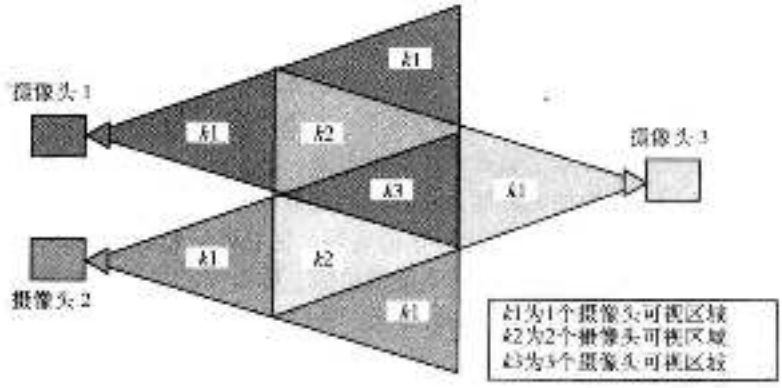


图 14-3　摄像头的不同重叠（k-overlap）程度

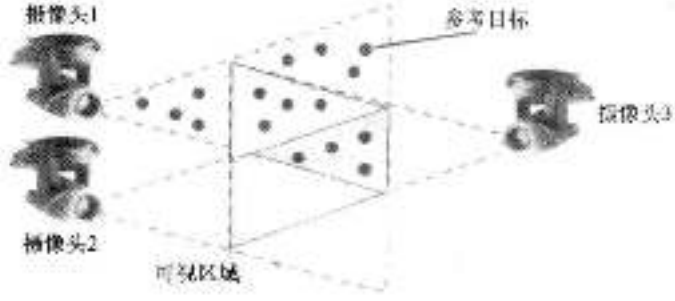


图 14-4　使用参考点分布估计 k-overlap

$$O_i^k = \frac{r_i^k}{r_i} \tag{14.1}$$

如图 14-4 所示，摄像头 1 可看到 16 个参考点，其中包括只有自己能看到的 8 个点；摄像头 1 和 3 同时看到 4 个点；摄像头 1、2 和 3 同时看到另外 4 个点。故摄像头 1 的可视区域的 1-overlap为 0.5，2-overlap 和 3-overlap 为 0.25。类似地，可以计算出其他摄像头的 k-overlap 值。

14.5　SensEye

SensEye［Purushottam07］是一个由多个层组成的视频传感器网络（如图 14-5 所示）。如前所述，一个传感器节点包括一个模拟视频传感器、一个 MCU，一个无线收发器及 RAM 和闪存。

所有在同层的传感器都是同质的（即同样类型），而不同的层传感器是异质的（即视频传感器有不同的性能）。高层传感器比底层传感器的性能（包括处理能力、组网能力、成像能力）更好，换句话说，高层传感器能耗更高。为了降低系统能耗，应当仅在底层传感器无法胜任高效率的图像捕捉时，才使用高层传感器。因为不同任务在多层执行，需要能效好的协议去协调不同层的视频传感器。

SensEye 在给不同层分配任务时做了很好的权衡。它使用一个三层结构（如图 14-5 所示）：

1）SensEye 的最底层由 Crossbow 节点（RF = 900MHz）［Crossbow08］、低保真（low-fidelity）Cyclop 或 CMUcam 视频传感器组成。

2）第二层由装有网络摄像头的 Stargate［Stargate08］节点组成。每个 Stargate 拥有一个运行 Linux 的嵌入式 XScale 处理器（400MHz）。显然，与第一层视频传感器相比，第二层的网络摄像头能捕捉高分辨率的图像。为了维持上游和下游的通信，每一个二层节点有两套无线设备：①一个用于在 Stargate 节点间进行点对点通信的 802.11 无线设备；②一个用于与第一层节点通信的 900MHz 无线设备。

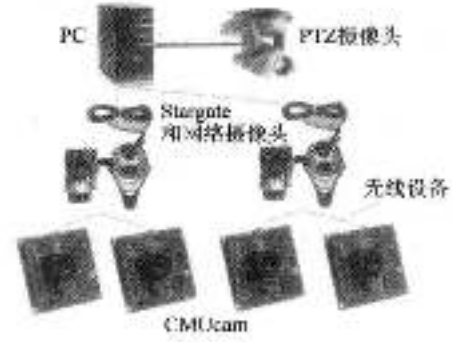


图 14-5　多层 SensEye 硬件结构

3）第三层由稀疏部署的连接到嵌入式系统（如手提电脑）的高分辨率变焦云台（PTZ）摄像头组成。这些摄像头支持编程，可用来填补第二层产生的覆盖范围的空隙，也可执行定标。


奇思妙想

多层视频传感器的部署是个有趣的想法。在一层中实现目标检测/识别/跟踪的算法是低效的，高层节点的 CPU 性能更好，但是也更耗电。使用这种层级结构会已经在很多场景中应用，例如，Internet 主干就是这样的多层结构。主干路由器是超高速的（>40Gbs），但是也相当昂贵。校园网使用价廉的路由器控制网络传输。这种基于树的多层结构与人类社会的结构类似。

407
|
409

SensEye 多层摄像头传感器网络的设计基于下面三个原则：

原则 1：把任务安排给较低层：尝试把任务安排到较低层可减少能耗，但是，如果较低层传感器不能胜任某些任务（如正确地、可靠地、快速地执行一些任务），那么就要需较高层传感器帮助。

原则 2：仅当需要时唤醒节点：为了节约能源，每个节点上的处理器、无线装置及传感器

是轮流工作的。SensEye 仅在需要时通过使用触发器将节点从睡眠状态唤醒，例如，当较低层传感器检测到一个新的目标后，需要获得一张高分辨率图像时，才唤醒一个高保真摄像头。尽可能地使这些装置工作在睡眠状态，将极大地延长网络使用寿命。

原则 3：利用覆盖范围的冗余：尝试利用摄像头覆盖范围的重叠来定标。例如，使用两个摄像头（有一个重叠覆盖范围）可定位一个目标并计算其（x，y，z）坐标，然后使用坐标值来唤醒其他节点或确定目标的轨迹。而且，通过利用覆盖范围的冗余，可提升能耗性能并使系统使用寿命最大化。

SensEye 在很短的等待时间内可检测到目标，即保证了高可靠性，又具备良好的能效。当无法在同质的单层的网络中实现上述目标时，可在跨越不同层的所有可能分配的排列中寻找某个点上解决该问题。

SensEye 使用四种类型的摄像头：1）Agilent Cyclops（14.3 节介绍）；2）CMUcam 视觉传感器［CMUcam08］；3）Logitech Quickcam Pro 网络摄像头；4）Sony PTZ 摄像头。

通过三种不同的平台可实现射频通信：Crossbow 节点［Crossbow08］、Intel Stargates［Stargate08］及 mini-ITX 嵌入式电脑。这些节点可与摄像头的不同层接口：

1 层：由一个低耗摄像头传感器组成，如 Cyclops，并与一个低耗的节点传感器平台连接。选择的是，因为没有成熟的产品，SensEye 仅使用一个 Cyclops 摄像头的原型。Cyclops 的软件提供帧捕捉、帧差分及目标检测功能。

2 层：由性能更好的平台和摄像头组成。每个 2 层节点有一个唤醒线路，一旦它收到从 1 层节点传来的触发指令，即可将节点从睡眠状态或挂起（suspended）状态唤醒。在实现 SensEye 时，会使用一个 Intel Stargate 传感器平台（有一个附属节点充当唤醒触发器）。

410

因为 Stargate 的硬件不支持自动唤醒功能，所以 Turducken［Sorber05］设计了一个延迟线路用来作为触发器。Logitech 网络摄像头通过 USB 接口连接 Stargate。

3 层：由连接到运行 Linux 的嵌入式电脑的 Sony SNC-RZ30N PTZ 摄像头组成。

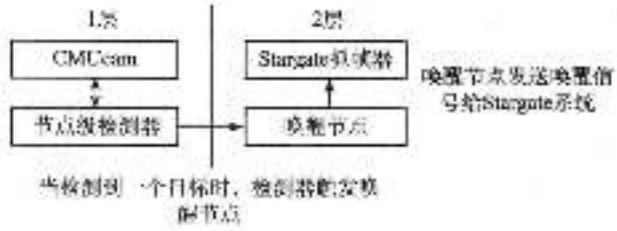


图 14-6　SensEye 的软件架构

SensEye 的软件框架如图 14-6 所示。在该图中，假定 1 层由连接到 CMUcam（或 Cyclops）摄像头的节点组成。我们可以用 Cyclops 代替 CMUcam。SensEye 的前两层结构有四个软件组件：1）CMUcam 帧差分器；2）节点级检测器；3）唤醒节点；4）Stargate 目标识别。

1 层的帧差分器：1 层节点捕捉用于差分（differencing）的图像。CMUcam 能捕捉一个图像并将其量化（quantize）成一个低分辨率帧，然后使用参考的背景帧执行帧差分。该帧差分的过程通过使用非零差分值使目标凸显出来。CMUcam 在帧差分过程中有两个工作模式：1）低分辨率模式，将当前图像（88×143 或 176×255）转换为用于差分的 8×8 图像；2）高分辨率模式，转换为用于差分的 16×16 图像。

节点级检测器：1 层节点将目标检测的结果报告给较高层节点。开始时，1 层节点发送初始化命令发送给它的模拟视频传感器以建立背景和帧差分参数。视频传感器周期地捕捉一张图

像并执行帧差分。节点使用一个用户指定的阈值，通过分析返回的帧差分结果判断一个目标是否出现或移动，如果检测到某个事件，节点向较高层节点广播一个触发指令。 411

唤醒节点：2 层的节点（连接到 Stargate）从较低层节点接收到触发指令，确定是否唤醒 Stargate 以进行进一步视频处理。该过程需要定位坐标。注意：通常在 1 层节点不计算目标的坐标，因为将导致在 1 层节点间过多地协调操作。SensEye 因此使用 2 层节点计算这些坐标，而 1 层节点仅计算如 θ 和 ϕ 参数，及目标图像的质心（centroid）。然后 2 层节点使用定标算法计算出坐标值，如果目标的位置在 Stargate 节点的可视范围，则唤醒 Stargate 节点，否则忽略触发指令。

高分辨率目标检测与识别：Stargate 节点能立即捕捉被唤醒的网络摄像头当前的视觉图像。然后在捕捉的图像和参考背景间执行帧差分，帧差分找出图像里出现的可能的目标的像素和轮廓，SensEye 通过使用基于颜色阈值过滤和相邻区域均值的平滑技术移除噪声像素。下一步，使用目标识别算法找出每个可能的目标。SensEye 使用基于目标像素颜色均值的方案，即计算出目标的红色、绿色及蓝色部分的均值，然后三色的均值与包含许多已定义目标的库进行比对，从而对目标分类。通过添加成熟的分类技术、面部识别及其他视觉算法，对 SensEye 进行了扩展。

PTZ 控制器：3 层有若干可重定目标的摄像头，能填充覆盖范围空白，增加覆盖范围冗余。利用 PTZ 摄像头的底座和倾斜值，使用定位技术即可完成定标。摄像头提供了一个供外部驱动程序控制其运动轨迹的 HTTP API。SensEye 使用基于 HTTP 的摄像头驱动［Sony08］去重定位 PTZ 摄像头。

问题与练习

14.1　详细说明视频传感器网络的几种应用。

14.2　视频传感器网络与一般传感器网络相比，有哪些特殊需求？

14.3　说明视频传感器节点定标的需要性。

14.4　为什么 SensEye 使用一个三层结构？ 412

14.5　说明确定 k-overlap 值的基本原则。

第七部分

Wireless Sensor Networks: Principles and Practice

其他主题

第15章

Wireless Sensor Networks: Principles and Practice

无线传感器网络能量模型

传感器节点由电池驱动，因此传感器寿命的量化需要用到能量消耗模型。所有无线传感器网络（WSN）协议的设计均以保证能量效率为核心目的。本章将介绍一些常用的无线传感器网络能量模型。



在许多情况下，对性能指标需要用数学或者仿真模型来描述。目的是在真实硬件环境中对传感器网络协议进行测试之前，获得其协议性能的估计值。在实际中，除了对系统的生命周期进行近似估计之外（如一个传感器的电池寿命），很难测量由传感器节点的能量消耗。因此，基于模型的能量计算可以解决这个难题，为无线传感器网络性能值的计算提供一个合适指标。

15.1 基本WSN能量模型

前面讨论过，传感器节点的通信消耗了传感器节点的大部分能量。其他部分（如CPU计算）会消耗一些能量，但仍然少于通信所消耗的能量。图15-1为一个简单的无线链路及其能量模型，此模型已经被广泛应用于无线传感器网络能量计算中。

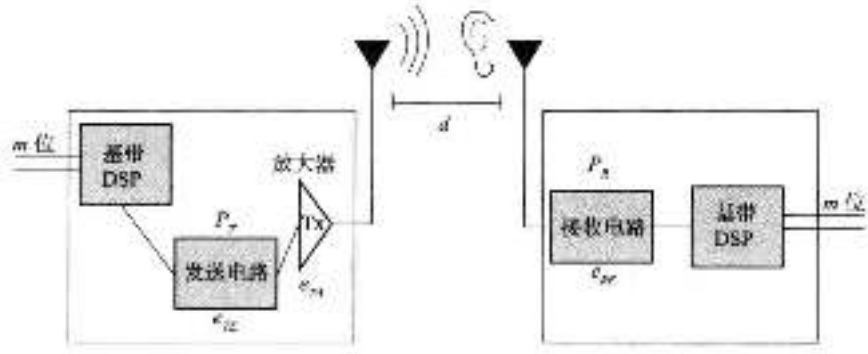


图15-1 无线传感器网络能量模型

在一跳无线链路中传输一个 m 位的数据包所消耗的能量在文献［Carlos04］中表述为：

$$E_t(m,d) = [E_T(m,d) + P_T T_m + E_{encode}] + [E_R(m) + P_R T_m + E_{decode}] \quad (15.1)$$

其中：

E_T 表示发送电路和功率放大器的能耗。

E_R 表示接收电路的能耗。

P_T 表示发送电路的功率。

P_R 表示接收电路的功率。

E_{encode} 表示编码的能耗。

E_{decode} 表示解码的能耗。

假设在发送和接收电路中每一位的能量消耗存在一个线性关系，E_T 和 E_R 可以写成

$$E_T(m, d) = m(e_{TC} + e_{TA}d^{\alpha}) \quad E_R(m) = me_{RC} \tag{15.2}$$

其中：

e_{TC}、e_{TA}和e_{RC}是硬件相关的参数。

α 是路径损耗指数，值的范围为 2（适用于自由空间）~4（适用于多径信道模型）。

适用的 MAC 协议类型会极大地影响收发器启动时间 T_{st}。为了尽可能地降低功耗，收发器应当尽可能长时间地处于休眠模式。虽然休眠模式可以节省大量电能，但需要注意的是，频繁地启动和关闭收发器同样会消耗能量。

415 ~ 416

关于 e_{TA}的明确表达式可以推导如下［Carlosl4］：

$$e_{TA} = \frac{\left(\frac{S}{N}\right)_r (NF_R)(N_0)(BW)\left(\frac{4\pi}{\lambda}\right)^{\alpha}}{(G_{ant})(\eta_{amp})(R_b)} \tag{15.3}$$

其中：

$(S/N)_r$ 是接收方解调器针对一个可接受的 E_b/N_0 所需的最小信噪比。

NF_R是接收电路的噪声。

N_0 是 1Hz 带宽中的热噪声层（W/Hz）。

BW 是信道中噪声带宽。

λ 是波长（米）。

α 是路径损失指数。

G_{ant}是天线增益。

η_{amp}是发送电路功率。

R_b是原始比特率（bps）。

该 e_{TA}表达式可用于有某种硬件配置的情况。如果将上面等式变形如下，$(S/N)_r$ 对于 e_{TA}的决定性作用将更加明显：

$$e_{TA} = \xi \times (S/N)_r \tag{15.4}$$

其中

$$\xi = \frac{(NF_R)(N_0)(BW)\left(\frac{4\pi}{\lambda}\right)^{\alpha}}{(G_{ant})(\eta_{amp})(R_b)}$$

由于该式强调了 e_{TA}和误码率 P 之间的关系，因此将这种依赖关系明确表示出来十分很重要。P 由 E_b/N_0 决定，E_b/N_0 由 $(S/N)_r$ 决定。已知 E_b/N_0 与数据速率无关。为了将 E_b/N_0 和 $(S/N)_r$ 联系起来，必须考虑数据速率和系统带宽，即

$$(S/N)_r = (E_b/N_0)(R/B_T) = \gamma_b(R/B_T) \tag{15.5}$$

其中：

E_b 表示每比特信息需要的能量。

R 表示系统数据速率。

B_T 表示系统带宽。

γ_b 表示每比特的信噪比，即 E_b/N_0。

417

尽管上述模型可以准确地计算出发送或者接收的能量，但许多无线传感器网络开发者却青睐如图 15-2 所示的简化模型［Akyildiz02］。

在发送方，能量消耗包括两个方面：1）本地电子设备的能量消耗；2）经过距离 d 发送 k 比特信息消耗的能量。在接收方，则只包括接收 k 比特信息时本地电子设备的能量消耗。

例 15.1：假设每一跳的距离为 5m，发送方与接收方之间的距离为 100m。数据量为 1M 比

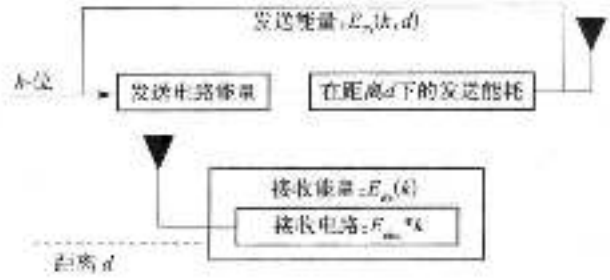


图 15-2　简化的能量模型

特。系数 E_{elec} 50nJ/bit，E_{amp} 100 pJ/bit。问无线传感器网络消耗的能量为多少？

解：总共有 100m/5m = 20 对发送和接收传感器节点（每个传感器节点在上一跳中为接收方，在下一跳中为发送方）。

对于每对传感器节点，能量消耗情况如下：

发送方消耗的能量：$E_{Tx}(k, Td) = E_{elec} * k + E_{amp} * k * d^2$

即 $E_{Tx}(k, Td) = (50 * 10^{-9}) * 10^6 + (100 * 10^{-12}) * 10^6 * 5^2 = 0.0525J$

接收方消耗的能量：$E_{Rx}(k) = E_{elec} * k = 0.05J$

每对传感器节点消耗的能量 $= E_{Tx}(k, d) + E_{Rx}(k) = 0.1025J$

整个无线传感器网络消耗的能量 $= 0.1025 * 20 = 2.05J$

15.2　基于仿真的能量模型

之前介绍的数学模型可以提供无线传感器网络能量消耗的量化结果，另外一个测量能量的方法则通过仿真模型来实现。在文献 [DSchmidt07] 中，基于有限状态机（FSM）的仿真模型
418 可以准确地测量出 Crosebow 传感器节点消耗的能量。


奇思妙想

无论使用数学模型还是仿真模型，两种方法都没有使用真实的硬件平台来测量能量。在实际中，很难使用仪器测量出 CPU、无线通信芯片以及其他电路的能量消耗。所以一般在无线传感器网络的研究中，可以创建一些基于经验数据（如实验测量）或者系统状态分析（如 FSM 模型）的能量模型。

该模型主要是在系统层次上描述能量。一些软件工具（如 SPICE）则可以在晶体管或寄存器层次上建立传感器节点芯片的能耗模型。虽然这些仿真工具涵盖了所有的影响因素，包括泄漏量和交换的能量，但由于这些工具都需要深入了解硬件结构（如寄存器接口等），因此，建立能量模型仍然面临很大困难。同时，由于在电路层次上仿真十分耗时，一般要花很长时间去仿真一个有着大量节点的网络系统。

电路层次上的能量仿真十分复杂，而在指令层次（即运行代码）上的硬件仿真能量模型已经成功创建并应用于一些 CPU 能耗的计算。这种方法通常通过测量综合软件基准程序来实现。这些基准程序使用一系列只执行一种指令的程序，以便每一个 CPU 指令的能耗可以通过测量而得出。然而，要给内部指令的依赖以及算法操作数对 CPU 能耗的影响这两个参数建立模型，需要用到一些特殊的能量测量方法。

基于指令的能量模型的优点是，可以在不需要深入了解硬件电路知识的条件下，实现相对精确的能量仿真。此外，相对于电路仿真，指令层次的仿真能够缩短仿真的运行时间。然而，

CPU 模型的建立代价相对较高，在运行时所建模型开销很大，这是因为传感器节点用于计算的能量非常有限。



如果你学习过“计算机组成原理”课程，应该能够理解在 CPU 性能测量中基准程序的重要性。这些基准都是 CPU 设计者在比较不同的 CPU 的速率和能耗后精心确定的。然而，在这里不仅测量了 CPU 的能量消耗，同时也会涉及其他部分的能量消耗，如无线通信能量等。因而原本的基准将不再那么有用。

419

为了解决上述的问题，Schmidt 等人［DSchmidt07］提出了一个基于组件的高层次建模方法。我们知道，一个无线传感器网络节点是由多种硬件组件组成的，如微控制器（即 CPU）、无线收发器芯片、传感器电子元件以及各种其他设备，如 LED 管、闪存等。每一个组件都可以在不同的状态下运行。例如，一个收发器芯片能够在以下四种状态下运行：关电状态、空闲状态、发送状态、接收状态。同样，CPU 也可以在空闲状态、中断状态以及计算状态下运行。常用方法是使用有限状态机（FSM）为组件运行过程建立模型，并且按以下规则为每个状态添加一个能量模型：

1）称传感器节点组件中每个操作状态为 FSM 中的一个状态。

2）将每个操作状态之间可能的变化建立成 FSM 中状态切换的模型。

3）FSM 中的每个状态和每单位时间消耗的能量相关。

4）FSM 中的每个切换和两个操作状态间的切换所需的持续时间相关。

5）FSM 有良好定义的初始状态，与通电后组件达到的稳定状态相对应。

那么，如何获知状态保持以及状态间切换所消耗的时间和能量？通常，这些数据能够从实验或者简单的测量中获得。可以从传感器节点的每一个组件开始建立 FSM 模型，然后建立整个传感器节点的 FSM 模型。

在大多数情况下，只有微控制器（CPU）可以触发状态的切换。虽然 FSM 通常表示系统可能存在的所有状态，但它并不能反映在正常运行时状态切换会出现的限制因素。无线通信就是一个很好的例子，一旦其收发器开始帧传输，那就不能被中断，这样的情况在 FSM 模型中是无法表示的。因此，第二步是对典型情景下的传感器节点的动态行为进行分析，为这些限制因素建立模型。

SDL（Specification and Description Language，规范和描述语言）模型能够定义一个节点的动态行为。SDL 模型包含实现 SDL 语义的运行时环境，以及代码转换模式。在 SDL 模型中能够表示省电策略，既可以采用应用模型中显式的省电策略，还可以采用运行时环境中隐式的策略。无线传感器网络的动态行为能够被规范化成为一系列的通信状态机。SDL 模型则以状态机为基础，该种状态机与描述传感器硬件的状态机相似。

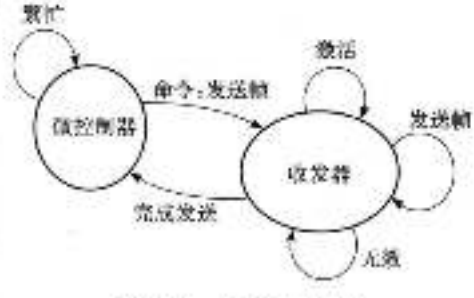


图 15-3　帧传输流程图

图 15-3 为节点动态行为的一个示例。它表示的任务是传感器通过无线接口将一帧的数据传输到一个远距离节点。首先，微控制器是唯一运行的组件，这是起始状态。接下来，收发器被触发，开始传输数据。需要传输的数据量决定了在传输模式下的时长。在传输之后，收发器再次进入休眠状态以节省

能量，同时，又被切换到初始状态。

那么根据上述的流程图，计算出特定任务所消耗的能量十分容易，计算过程参见式 15.6。此处 P_{state} 是某状态消耗的功率，t_{state} 为处于某状态的时间，P_{trans} 和 t_{trans} 分别是在两个状态转换时消耗的功率和时间。

$$E = \sum_{state} P_{state} * t_{state} + \sum_{trans} P_{trans} * t_{trans} \tag{15.6}$$

这样一个基于有限状态机的分析性模型可以集成到软件仿真器中，这会使得预测系统能耗的仿真更加精确。

另一方面，平台资源、网络资源以及能量资源等环境因素会对传感器节点的行为造成影响。由于一个因素的状态能显著影响其他的仿真因素，并最终改变仿真的结果，因此需要对这些因素建立模型，以精确地仿真出一系列传感器节点的能耗。

我们再看一个例子。无线信号收发器芯片主要用于同网络中其他传感器节点通信，它会显著地影响大规模无线传感器网络中传感器节点的能耗。一些网络层次上的因素，如网络拥塞以及无线带宽的限制，都将导致大量传输错误。这将改变网络中每个传感器的通信模式，并影响系统能耗。此外，由于在时间同步网络中引入了时钟振动，例如时钟/定时器不准确这样的平台限制也会使情况更加糟糕。因此，需要设计针对性较强的仿真器捕获这些影响因素。

420
~
421

已经构建出一些能模拟网络行为的仿真器［SAM06］，这些专门的仿真器可以进一步被编程为不同的仿真组件。为了形成一个系统层次的仿真器，需要采用基于消息的接口将组件相互连接起来。

那么在上述组件－系统仿真器中，如何建立能耗模型？以下两个步骤提供了很好的思路：

步骤 1：使用 FSM 描述单个传感器的能耗行为。

步骤 2：使用基于消息的接口将能耗仿真与其他网络层次的操作（如拥塞控制）耦合起来。

有两套方法能够实现以上构想：1）如果硬件组件（如 CPU）的能耗仿真已经在软件仿真模块中实现，那么把能耗集成到已有的仿真器中将很容易。这样，只需要实现能量仿真组件和系统仿真间的接口即可。2）如果给已有的仿真器添加另一个接口比较困难，那么可以将能耗仿真的实现作为一个新的仿真组件。在第二个方法中，软件仿真器是独立的；用一个仿真组件仿真传感器硬件组件的行为，另一个仿真组件通过实现能量模型来仿真它们的能量消耗。

这两套方法可以容易地集成到能量仿真器框架中。如图 15-4 所示，“网络节点”是仿真器框架中的核心部分，它需要把能耗集成到仿真软件中。实际上，该核心组件控制着用于仿真节点不同操作状态的所有仿真器。

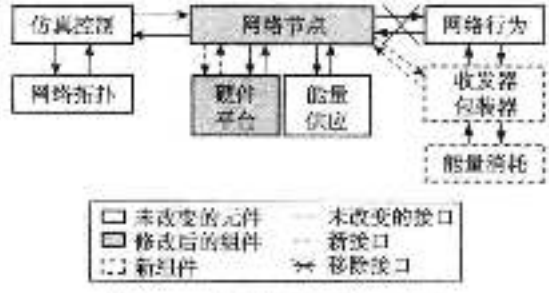


图 15-4　仿真器集成框架的结构

对于仿真的硬件，可以把能耗部分集成到仿真软件中。这就是之前第一套方法要到的。

针对网络层次上的系统行为仿真，需要开发能够实现能量模型的新组件，而描述网络行为的仿真组件替换为包装器（wrapper）（如图 15-4 所示）。包装器会在原始的仿真组件（仿真网络行为）以及增加的追踪每个传感器节点能耗的组件之间传送消息。

Schmidt 等人在文献［DSchmidt07］中提出了一个在 Crossbow MicaZ 上进行能量仿真的示例。MicaZ 节点含有一个 8 比特 RISC 架构的，时钟为 7.3728MHz 的 Atmel 微控制器，控制器包

含 4Kb 的内部 SRAM、4Kb 的数据 EEPROM 以及 128Kb 的闪存。其收发器芯片在高达 250kbit/s 的数据速率下工作。一个 512Kb 的闪存可以通过两个各 264 比特的 SRAM 页面缓冲区实现。三个 LED 用来显示设备的运行状态，且每个节点装有给每个节点唯一 ID 的序列号芯片。MicaZ 节点有 51 个扩展引脚，作为与任意传感器的接口。节点的整体结构如图 15-5 所示。

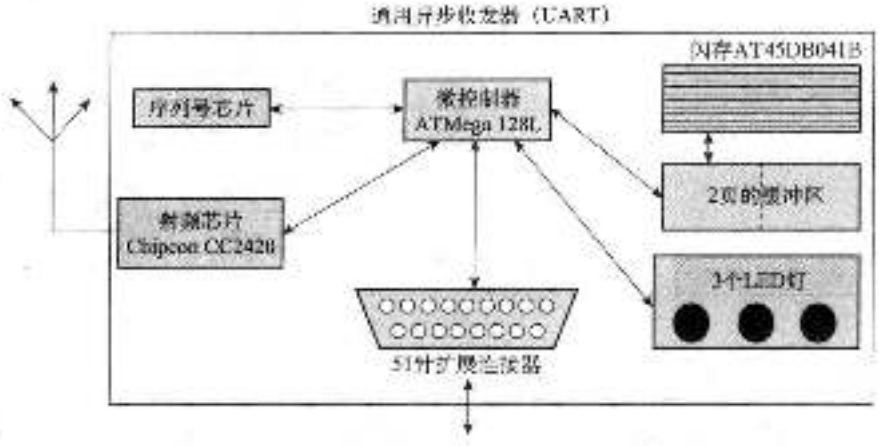


图 15-5　MicaZ 的结构

为了对 MicaZ 的能量消耗进行仿真，Schmidt 等人［DSchmidt07］考虑了微控制器、收发器芯片和闪存的能耗。LED 可以关闭，这样能够减少能耗，同时序列号芯片的能耗可以忽略不计。

MicaZ 节点中收发器和微控制器的组件模型如图 15-6 所示。该 FSM 模型还能显示状态切换所花时间。例如，收发器需要数微秒的时间而微控制器则只需要几个时钟周期。

微控制器会处于不同的能耗状态。MicaZ 节点的持续输入电压为 3V（2 支 AA 电池）状态，每个状态消耗的能量以毫安计。值得注意的是，收发器发送状态的能量消耗不是一个固定的值，而是一个范围。某状态的实际能量消耗取决于收发器芯片选择的输出功率。

15.3　能量感知路由

镍－镉电池和锂离子电池已经在无线设备和传感器中广泛应用。在电池中，大量的电芯以串行或并行（或者两种方式的）组合排列。每个电芯的活性物质由两个被电解质隔离的电极（正极和负极）组成。电池连接了负载后，其中连续发生的氧化还原反应可以把电子从正极传递到负极。

图 15-7 描述了上述现象，它采用了一个简化的对称电化学电池。图 15-7a 是一个充满电的电池，其电极表面包含最大浓度的活性物质。当电池连接到外接负载时，会有电流流过外部电路。

放电过程如图 15-7b 所示。在这种情况下，化学物质在电极表面被消耗，同时通过电解质的扩散获得补充。但是，扩散的过程无法及时补偿消耗，这也导致电解质中出现浓度梯度（concentration gradient）的原因。

较高的电流负载使得电解质浓度梯度升高，也就是说，在电极表面的活性物质浓度会更低［Doyle93］。而低浓度会使电池电压降低。最后，电压会降到事先预定的一个断电临界值，这意味着在电极表面电化学反应将不能够维持下去。在这种情况下，电池便停止工作（参见图 15-7e）。

422
·
423

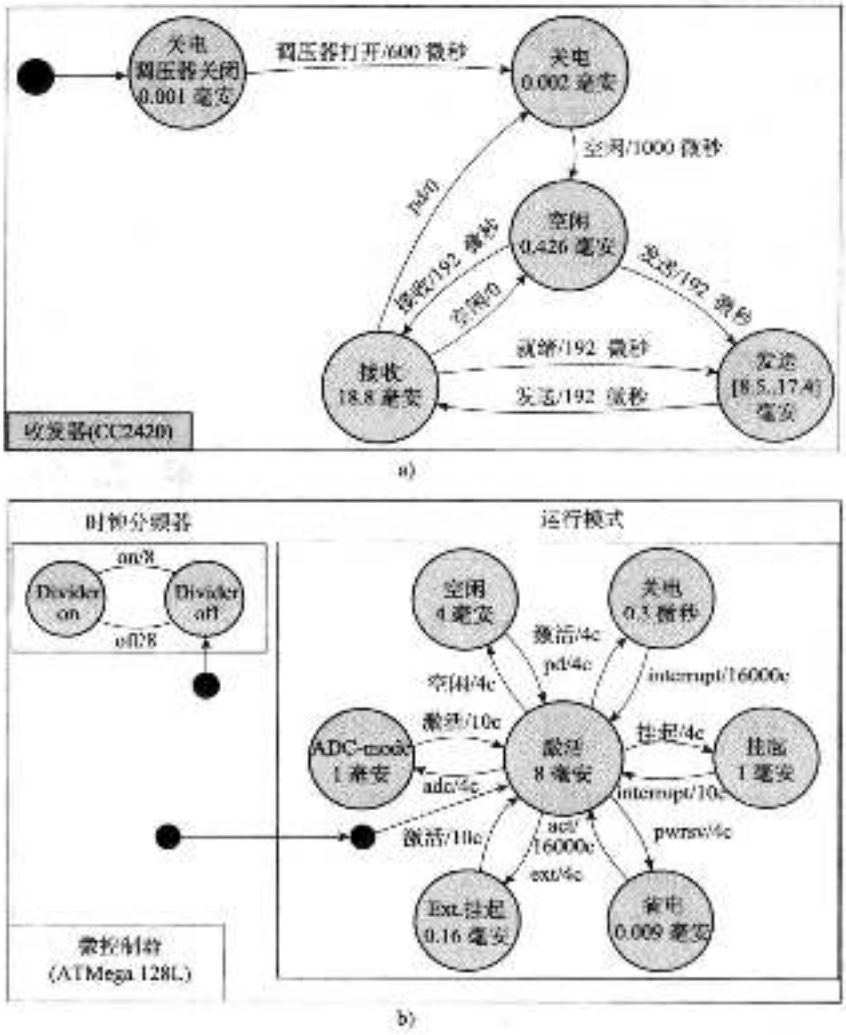


图 15-6　传感器节点 MICAz 的能量模型（只显示了微控制器和收发器）

值得注意的是，我们无法使用还没有到达电极的电活性物质。这种未用到的电荷称为放电损失。这不是物理意义上的“损失”，它仅仅代表着该物质由于反应速率和扩散速率上的不同而无法用于放电。

在电池停止工作之前，如果电池电量低或者为零，即电池处于恢复状态（参见图 15-7c），可以发现浓度梯度在一段长时间后变平，然后又一次到达平衡。

根据以上的恢复过程，电极表面附近活性物质的浓度使得之前未使用的电荷能够被放出（参见图 15-7d）。因此，电池恢复可以降低浓度梯度，弥补放电损失，从而延长电池的寿命（参见图 15-7f）。

424
~
425

一些在镍－镉电池和锂离子电池上的实验已经证明，放电损失占整个电池容量的 30% [Rakhmatov03]。因此，需要准确地模拟电池的行为，使传感器网络的系统性能最优。

数据流（如视频/音频）传输中的能量感知路由（BAR）可以简单地建模为从一个数据源传输到其相对应的目的地的数据包流。但是，如何将源和目的地这一对传感器节点的通信寿命最大化呢？文献 [Chi06] 提出了 BAR 的概念，其基本思想是选择恢复情况较好的传感器节点

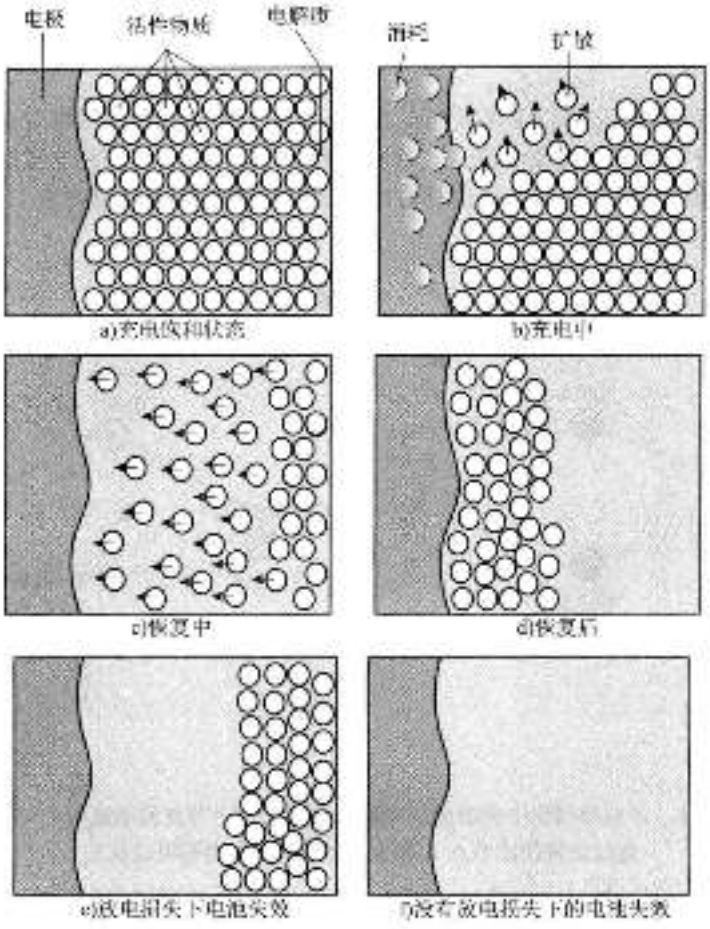


图 15-7　不同状态下的电池反应

作为中继节点，而让“疲惫”的传感器节点休息以待恢复。如果能够动态地调整路由路径，使节点电池的能力得到有效恢复，就可以使传感器节点的放电损失最小化，从而延长系统寿命，使源节点到目的节点间的数据吞吐量最大化。

在文献［Chi06］中，提出了能量感知路由协议。他们使用 BAR 算法在源节点到目的节点间建立路由路径。在描述能量感知路由协议之前，需要做一些假设。

假设节点在无线传感器网络中是随机部署的，每个节点知道其地理位置（这可以通过一些准确的节点定位算法实现，详见第 9 章）。节点由 AA 电池供电。现在我们的目标是数据流应用，如将传送视为一种流的视频监视应用。如果一个节点在源节点到目的节点的路由路径上，则称其为路由节点（routing node）。在每个时间片里，路由节点可以被分配任务（“活动”状态），也可以处于“空闲”状态。此任务可以是路由活动、视频播放、软件运行或者其他需要消耗能量的功能。可以在同一时间片内分配多个任务。

下面定义一些参数。设 C 是电池剩余的容量，β（常量）是实验的化学参数，不同的电池 β 值也不同。β 越大，放电损失越小。

如图 15-8［Chi06］所示，在这个传感器网络中，源节点 S 把数据包传送给目的节点 D。电池的剩余容量 C 和参数 β 如图中所示，比较下面的两种途径。

途径一：S 通过多跳路径 $S\rightarrow A\rightarrow C\rightarrow F\rightarrow E\rightarrow D$ 向 D 传送数据包。一段时间（如 45 分钟）后，节点 A 用光了能量，路由路径则变为 $S\rightarrow B\rightarrow C\rightarrow F\rightarrow E\rightarrow D$。整个连接可以持续约 90 分钟［Chi06］。

可以通过不停地交替选择以上两条路径这样一个简单的办法延长网络寿命。因此，方法二是将节点 A 和 B 互相交替作为路由节点。当 B 作为路由节点时，A 恢复电池电量，不将某一个节点的能量全部用尽，循环往复。用这种方法，整体的路径持续时间约为 113 分钟［Chi06］，提高了 24.8%。

426
-
427

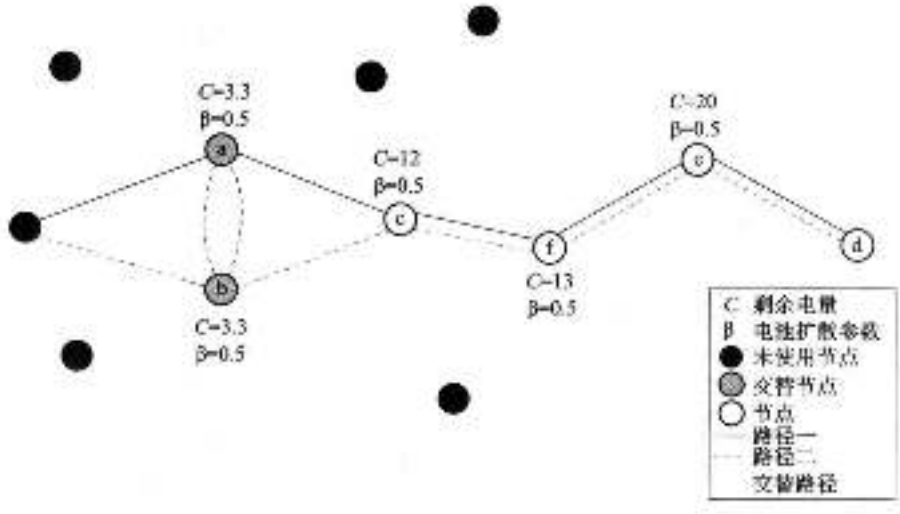


图 15-8　传感器网络中的能量感知路由 BAR。每个节点的电流为 I = 3.5A；通过交替使用节点 A 和 B，网络的生命周期得以延长。

总而言之，在能量感知的节能路由协议中，可以交替地恢复电池以延长节点的寿命。通常选择电量恢复最充足的节点作为路由节点，这是一个重要的思想。


奇思妙想

BAR 是一个相当出色的想法。虽然已经提出了很多关于能效的无线传感器网络路由协议，它们都是通过选择节能的路径来延长系统寿命，极少有深入到电池本身，探究电池充电/放电的原理的。可以看出，解决相同的问题，可以通过不同的硬件层次（系统层次、电路板层次、组件层次、芯片层次和晶体管层次等）来实现。然而，越接近底层硬件，对模型精确度的要求越高。

428

问题与练习

15.1　图 15-1 为一个最典型的无线传感器网络能量模型。请使用这个模型来说明距离为 100 米且只有 1 跳的通信，比 10 跳（每跳 10 米）的中继通信所消耗的能量高。

15.2　访问网址 http://www.xbow.com，阅读 MICAz 节点的数据表，说明其能耗特点，并寻找一些能够仿真 MicaZ 能耗行为的能量模型。

429

15.3　除了 BAR 例子之外，你能够找到其他基于能量感知的无线传感器网络协议的设计例子吗？

传感器网络仿真器

开发合适的仿真工具对于研究无线传感器网络几为重要。在大多数情况下，无法提供大规模（多于1000个节点）无线传感器网络测试实验床，而基于软件仿真的性能测试代价很小。如今，许多仿真工具都可以模拟出无线传感器网络中的噪声、干扰以及其他不确定的因素，甚至能够分析出不同硬件模块的能耗。本章将主要介绍一些典型的无线传感器网络仿真器。



一些工程师可能会低估软件仿真在无线传感器网络设计中的作用。实际上，为了节省性能测试的时间和费用，通常先使用基于离散事件的仿真工具检验大规模无线传感器网络中网络协议的效率。这些工具包括准确的无线通信模型和能量分析方法。在获得仿真结果后，能够避免一些潜在的工程设计错误。

16.1 GloMoSim

GloMoSim可以为有线和无线网络系统（包括无线传感器网络）创建可变的仿真环境。它是基于Parsec提供的并行离散事件仿真功能而设计的。大多数网络系统采用类似于OSI七层网络体系结构的分层方法建立的。GloMoSim使用类似的分层方法。标准的API（应用程序接口）在不同的仿真层次使用，这允许不同人设计在不同层开发的模型能够快速地整合。GloMoSim库现在采用的协议包含以下方面：

层	协议
移动层	随机路点（random waypoint）、随机行走（random drunken）、基于轨迹（trace-based）
无线传播层	均匀衰减以及自由空间
无线模型层	噪声累积
数据包接收模型	SNR有界，基于带BPSK/QPSK调制的BER
数据链路（MAC）	CSMA、IEEE802.11和MACA
网络（路由）	带AODV的IP、Bellman-Ford、DSR、Fisheye、LAR机制1、ODMRP和WRP
传输	TCP和UDP
应用	CBR、FTP、HTTP和Telnet

出处：Zeng, X等人著，GloMoSim: A library for parallel simulation of large-scale wireless networks, Proceedings of the 12th Workshop on Parallel and Distributed Simulations, May 26－29, 1988, Banff, Alberta, Canada.

为了运行GloMoSim，需要用到最新的Parsec编译器（目前已经包含在GloMoSim发布的版本中）。如果协议开发者编写纯粹的C源代码，他们需要用到Parsec编译器。Parsec代码被广泛运用于GloMoSim核，而大多数用户都无需知道内核是如何工作的。

16.2 SensorSim

SensorSim建立在NS-2模拟器之上，并提供了额外的特性用于模拟传感器网络。这个平台的主要特征包括：1）感知信道和传感器模型；2）电池模型；3）无线传感器网络的轻量级协议栈；4）场景生成；5）混合仿真。

图16-1为内部仿真模块。它提供一个用户图形接口（GUI），用于感知数据的产生和可视

化。图 16-2 为单个传感器节点的仿真体系结构，它包括精确的无线传感器网络发送/接收能耗
[432] 模型。

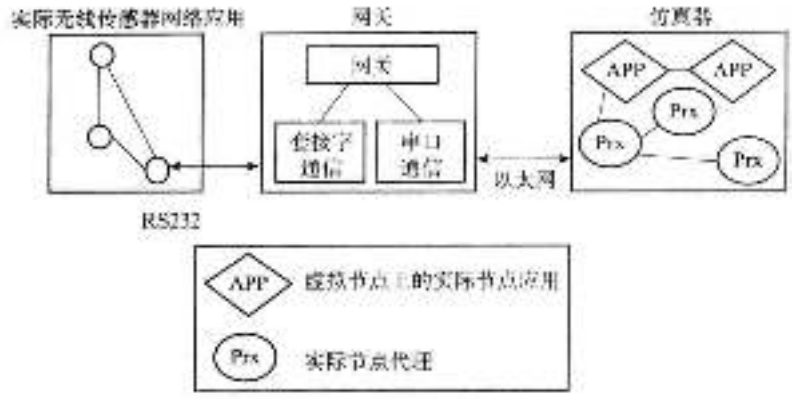
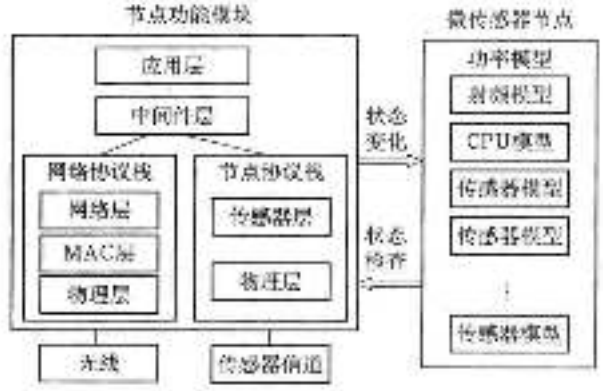


图 16-1　SensorSim 系统模型



[433] 图 16-2　SensorSim 传感器节点模型

16.3　TOSSIM

TOSSIM［Philip03］用于描述成千上万的 TinyOS 节点在网络位粒度下的行为和反应。TOSSIM 的整体结构图如图 16-3 所示。它由五个部分组成：1）TinyOS 组件图的接口；2）离散事件队列；3）重新实现的 TinyOS 硬件抽象组件；4）可扩展的无线通信和模拟数字转换器（ADC）模型；5）用于与仿真器交互的外部程序的通信服务。

在 TOSSIM 中，离散事件仿真直接从 TinyOS 组件图中产生。它运行的代码与实际无线传感器网络硬件上运行的代码相同。通过替换一些低层次组件（如图 16-3 的阴影部分所示），TOSSIM 将硬件中断转换为离散仿真事件。

TOSSIM 采用简单但有效的方法对传感器网络场景进行抽象。传感器的状态包含它在无线信道中监听到的内容。这套抽象概念能测试出理想的无线链路（即比特错误率为零），同时，能够轻易地获取隐藏的终端问题，并捕获在数据包传输过程中可能出现的各种问题（如标志检测失败和数据损坏）。

如图 16-3 所示，TOSSIM 引擎提供了一系列通信服务与外部应用交互。这些服务允许用户程序通过 TCP 套接字与 TOSSIM 接口连接，从而帮助程序员监控或者激活正在运行的仿真程

序。用户同时能够通过这些服务了解 ADC 和无线模块的状况，例如传感器读数和丢包率。

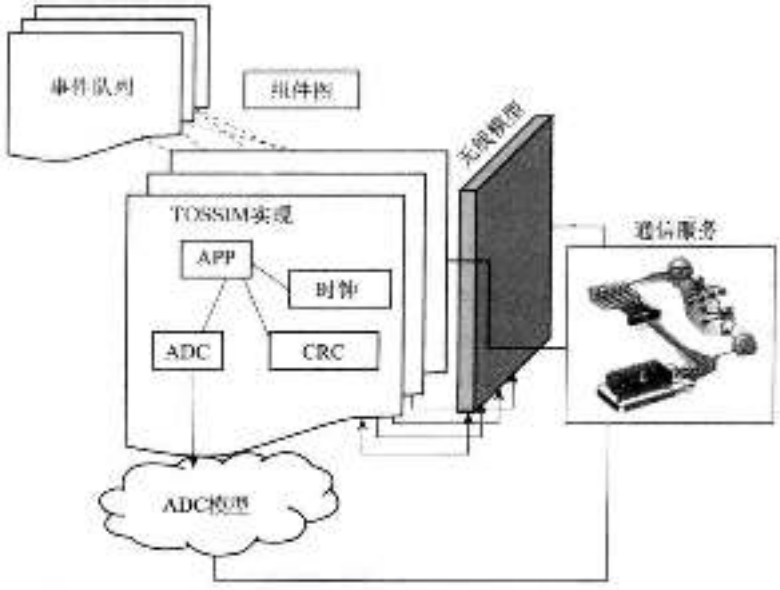


图 16-3　TOSSIM 的结构：帧、事件、模型、组件等

TOSSIM 拥有对 TinyOS 工具链的支持，这个特征简化了模拟网络和真实网络间的转变过程。编译本地代码时允许开发者使用 TOSSIM 中传统的调试工具。在调试工具中，用户可以设置调试断点，单步调试实时代码（如数据包接收）而不干扰程序运行。

TOSSIM 将每个硬件资源描述为一个组件，模仿下层基本硬件（包括 ADC、时钟，传送电位计、EEPROM、启动序列组件以及其他无线组件）的行为。

TOSSIM 网络模型可以轻松地捕获传感器内部的反应。每个比特的传输都能激活模型，它通过观察其他节点接收到的事件改变信道的状态。

值得注意的是，除应用层的特点之外，TOSSIM 还允许用户开发、测试以及评估物理层/MAC 层的网络协议。

图 16-4 为一个 TOSSIM 执行过程的例子。

TinyOS 开发者可以在 TOSSIM 中按需要自由选择无线信号模型的精度和复杂度。由于无线信号模型并不依赖于仿真器，因此用户能够很容易地对这些模型进行修改。

434
|
435

在 TOSSIM 中，网络被建模为有向图。通常在一个图模型中，用顶点代替节点。当节点 u 发送信息给 v 时，图中的每条边 (u, v) 代表其错误率。而且，边 (u, v) 和边 (v, u) 是不同的。图模型支持非对称连接的精确仿真，因为在来回的方向上有不同的错误率，使每条边来回值不对称。比特错误是独立的。连接概率可以由用户指定，并在运行过程中调整。传输事件不断传送到每个相连节点的仿真输入信道。每个节点都有其网络信道的本地视图。

例如，假设节点 T 通过零错误信道发送数据给节点 R，在每个比特事件上，T 发送 0 或者 1。节点 R 会根据从信道中收到的信息，调整其内部状态。在每个比特事件上，节点 R 读出状态后，会将比特信息发送给 TinyOS 组件。

用户使用 TCP/IP 协议使 PC 应用能够与 TOSSIM 通信，完成驱动、监视以及启动仿真等一系列动作。该仿真 - 应用协议是基于 TinyOS 抽象的命令/事件接口。

用户通过向 TOSSIM 发送命令启动仿真并调整传感器内部状态。这些命令可以改变无线连

```
Time (1 MHz ticks)  Action
-------------------------------------------------------------------
100   Dequeue simulator event at time 100.
The clock interrupt handler is called, signaling the application Timer event.
The application's Timer handler requests a reading from the ADC.
The ADC component puts a simulator ADC event on the queue with time stamp.
The interrupt handler completes; the clock event re-enqueues itself for the next tick.
400 Dequeue and handle simulator ADC event at time 400.
The ADC interrupt handler is called, signaling an ADC ready event with a sensor value.
The application event handler takes the top three bits and calls LEDs commands.
The ADC interrupt handler completes.
1000 Simulator event is dequeued and handled at time 1000.
The clock interrupt handler is called, signaling the application Timer event.
... execution continues as above.
```

图 16-4 执行范例

接错误率或者感知数值、开启或者关闭传感器以及注入网络数据包。

用户还能够编写他们自己的系统，用新的方法与 TOSSIM 连接。在编译节点的时候，监听/启动的代码和注释则会被移除。

TOSSIM 有一个称为 TinyViz 的可视化工具，它是专为 TOSSIM 设计的基于 Java 平台的 GUI。TinyViz 使仿真可视化、可控制且容易分析。它可以对仿真状态进行可视化反馈。同时，它利用专门的机制控制仿真过程，如调整感知数值和无线连接错误率。

TinyViz 还有一个插件接口，这个接口允许开发者实现他们自己的与应用相关的可视化界面以及基于 TinyViz 引擎的控制代码。TinyViz 引擎管理 TOSSIM 的事件/命令接口，并将 TOSSIM 事件发送给加载的插件。在某些情况下，这十分有用。例如，插件可以用于实现节点在收到数据时网络流量的可视化。TinyViz 插件还可以向 TOSSIM 发送命令以调用仿真器。比如，当用户在可视化窗口关闭传感器时，控制插件会把相应的断电指令发送给 TOSSIM。

除了上面提到的网络和控制插件外，TinyViz 还有一系列默认插件，用于基本的调试和分析。某些插件可以显示（以列表形式）所有调试的信息，其他一些插件则可以通过图形化方式显示出无线信道和 UART 数据包中的数据。传感器插件在 GUI 中显示传感器的数值，同时让用户在仿真过程中设置各个传感器数值。无线通信模型插件能够在 GUI 上根据节点间的距离更新无线连通性。它还能够以图形化方式显示连接概率，为变化情况下的网络表现的试验提供基
436 础机制。

基于 TOSSIM 的内置模型，用户能够自己编写有用的 TinyViz 插件。例如，用户可以通过改变误比特率来模拟无线信号障碍物（如金属阻挡），还可以通过在事先约定好的时刻关闭节点以模拟节点失效，或者利用插件和仿真数据去检测并分析应用的行为。TinyViz 使用 TOSSIM 的通信服务让用户对大型网络有全面的了解，还能在仿真过程中检测节点的内部情况。

16.4　PowerTOSSIM

在实际的无线传感器网络实验中，要准确地测量出每个芯片组件（如 CPU 和内存）的能耗是十分困难的。但 PowerTOSSIM［Victor04］可以跟踪硬件功率状态的切换，它由 TinyOS 代码对应的硬件抽象组件组成。PowerTOSSIM 还有一个精确的基于基本块级分析的 CPU 周期计数机制以及关于节点级能量消耗情况的可视化工具。

16.4.1　PowerTOSSIM 的结构

PowerTOSSIM 的结构如图 16-5 所示。PowerState 模块接受仿真的硬件组件（如无线模块、传感器、LED 等）发出的能量请求，然后给每个组件发出功率状态切换消息。根据能量模型的计算，这些信息最终会产生详细的能耗数据或者使以可视化方式显示能耗。

PowerTOSSIM 能够记录下任何被仿真的传感器中每个硬件组件功率的状态，其跟踪过程是通过在仿真运行时记录的特定功率状态切换消息实现的。PowerTOSSIM 能够对组件 PowerState 发出调用请求，以跟踪每个节点的硬件功耗状态，并且在运行过程中将其保存在文件中。

一个有挑战性的问题是如何估计 CPU 的使用。因为 PowerTOSSIM 把程序转换为本地主机上的二进制代码运行，它不知道传感器占用 CPU 的时间长度。但是，它能够通过仿真代码执行的基础块与对应节点二进制代码的周期计数之间的映射描绘出 CPU 的状况。PowerTOSSIM 将其产生的功率状态数据与能量模型结合起来，以判断每个节点和每个组件的能量使用情况。以上追踪过程可以在脱机的情况下进行，以获得每个传感器的硬件组件具体能耗，也可以输出到 TinyViz 可视化工具中，实时显示能耗数据。为了获得更高的效率和灵活性，通常将功率状态切换数据的产生和处理过程分开进行。

437

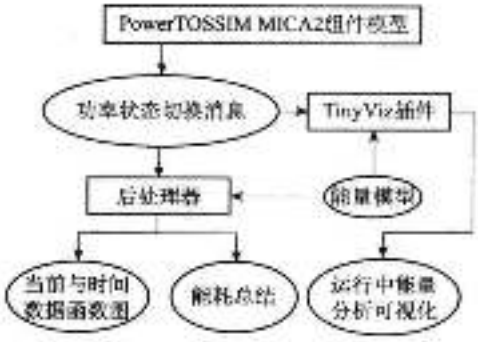


图 16-5　PowerTOSSIM 的结构

效率：和 TOSSIM 一样，PowerTOSSIM 也能够模拟具有成千上万节点的大型网络。为了保持其扩展性，应当避免仿真中过高的开销。如果只是在运行时记录硬件状态切换的消息，那么开销很低。类似地，如果让仿真像本地二进制代码一样运行，就可以避免指令层次仿真的开销。

灵活性：PowerTOSSIM 在捕获和模拟节点的功率状态时有高度灵活性，但是因为设计方案一直在发展，所以它没有设定特定的硬件平台。通过解耦设计，将新的能量模型引入到 PowerTOSSIM 分析工具中，便能评价出新的硬件设计的能效。仿真软件本身并不需要重新执行。



奇思妙想

记住这种理念：复杂系统设计中的模块性。记得因特网网络层次结构，如应用层和传输层吗？因特网不只使用一个层，因为在不接触整个系统的情况下去修正每个子模块会简单得多。只要模块中的接口是一样的，就能够很容易地按照新设计要求更新每个模块。

438

16.4.2 组件装配

对于传感器的每个硬件组件来说，TinyOS 有一个负责控制硬件组件操作的专用软件模块。例如，ChipCon CC1000 无线设备中无线通信的大多数方面都能通过 CC1000RadioIntM 模块实现。TOSSIM（详见上一节）通过其自身的软件模块来模拟这些 TinyOS 硬件驱动，从而在代码变化很小的情况下把 TinyOS 应用链接到仿真硬件。

和 TOSSIM 一样，PowerTOSSIM 能将每个仿真硬件驱动与在仿真时记录下的功率状态切换消息装配在一起。PowerTOSSIM 发出请求（从每个硬件驱动）给一个称为 PowerState 的新模块，这个模块能够在每个硬件组件的功率状态改变时产生日志消息。通过在一个独立的模块中实现功率状态切换，就可以轻松地扩展接口，以支持新的硬件组件，如新的传感器平台（非 Crossbow 产品）。

16.4.3 CPU 能耗分析

PowerTOSSIM 能够把 TinyOS 应用代码编译为能够在仿真机器上直接运行的二进制文件。这种设计虽然十分有效，但不容易判断出相比于“空闲”状态，或者任何其他低能耗状态，CPU 花了多长时间运行在“活动”状态（持续执行指令时）。在很多情况下，需要记录 CPU 处于“活动”状态的时间，以便准确计算消耗的能量，尤其对于 CPU 密集的操作（如安全算法）或者一些特殊的场合（如传感器可能花费很多时间在低能耗睡眠模式，偶尔被唤醒执行计算）更是如此。

如今，大多数传感器节点的微控制器在执行指令时的能耗是基本恒定的。这是因为它们没有像高级处理器那样使用复杂的芯片级能量管理策略。在传感器节点中，大部分元件（如指令核心、SRAM、ADC、振荡器、时钟以及其他外围设备）在“控制器处于活动”模式时通常是开启的。如在 Crossbow MicaX 传感器节点中，ATMEL Atmega128L CPU 在执行指令时消耗 8mA 能量，而在空闲时只要 3.2mA。同时，每条指令的周期时间都被完整地记录下来而且通常是确定的，至少是预知性的。因此，PowerTOSSIM 通过记录每个功率状态下 CPU 运行的时间就能够计算出 CPU 的能量使用。节点处于空闲状态的时间取决于外部因素，如时钟中断的频率，而这些已经被 TOSSIM 模拟出来。

尽管 TOSSIM 无法获知执行 CPU 指令所用的时间，但 PowerTOSSIM 能够通过对每条指令的执行进行仿真来判断出 CPU 的执行时间。具体的策略包括以下四步：

439

1）检测二进制代码，获得每个没有分支指令的基本程序块的执行数。

2）将每个程序块映射为相应的汇编指令。

3）用简单的指令分析判断每个程序块的 CPU 周期数。

4）将仿真基本程序块的执行数与和它们相应的周期数相结合，以便获得整个 CPU 周期数。

当仿真结束时，PowerTOSSIM 会使用到基本代码段的执行计数器。离线处理这些计数器后，可获得 CPU 循环计数总数。这样一个过程相当准确而且在仿真期间开销很少。

16.4.4 PowerState 模块

如果在仿真器中分散功率状态跟踪代码，会导致很高的开销。因此 PowerTOSSIM 使用一个

称为 PowerState 的 TinyOS 模块。其他 TinyOS 组件向 PowerState 发送请求以注册硬件功率切换。PowerStat 有一个单独的接口，此接口与每一个可能的状态切换相关的命令相连。每个函数测试是否能进行功率分析，如果可以，发出一条记录了传感器 ID、某个功率状态切换以及当前仿真时间的消息。

以下所示为记录日志中的一些摘录［Victor04］：

```
0: POWER: Mote 0 LED_ STATE RED_ OFF at 18677335
0: POWER: Mote 0 LED_ STATE YELLOW_ OFF at 18677335
0: POWER: Mote 0 ADC SAMPLE RSSI_ POST at 18990479
0: POWER: Mote 0 ADC DATA_ READY at 18990479
0: POWER: Mote 0 RADIO_ STATE TX at 18992551
0: POWER: Mote 0 RADIO_ STATE RX at 19199375
```

16.4.5　分析工具

PowerTOSSIM 有几种工具可以用来分析并使能耗数据可视化。这些工具接受由 PowerState 产生的日志文件的输入、CPU 分析信息以及硬件能量模型。

其中一个工具叫做后处理器，它能够计算出每个传感器不同硬件组件消耗的总能量，并按照时间顺序输出每个传感器节点的功耗情况。

PowerTOSSIM 还有一个针对 TinyViz（TOSSIM 软件的一部分）的插件。这个插件可以在仿真运行时报告每个节点能耗状况。为了利于实现可视化，插件根据仿真过程中能量消耗的多少给传感器指定不同的颜色，从而实现网络中的能量热点可视化。 [440]

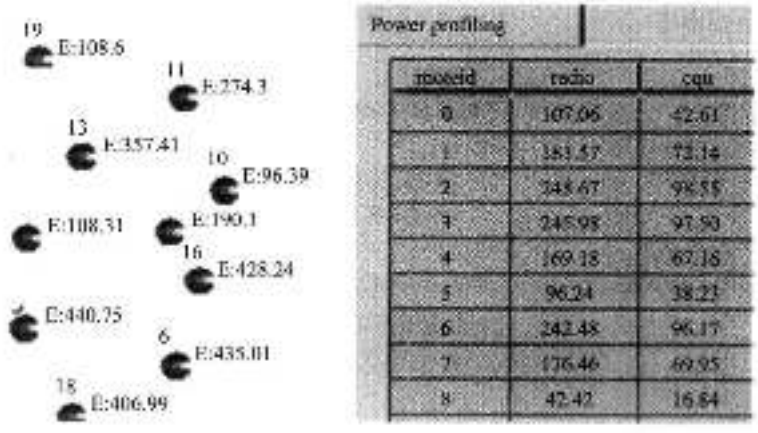


图 16-6　TinyViz 的 PowerProfiling 插件截图

图 16-6 为一个典型的可视化屏幕截图。右边的表格显示仿真网络中每个组件在运行时的能量消耗情况。根据仿真开始后传感器节点能耗总量大小，为每个传感器节点分配一种颜色。

问题与练习

16.1　与真实的无线传感器网络测试实验床相比，说明仿真的优点和缺点。

16.2　查阅文献［CloMoSim］，并下载软件，做一些简单的无线网络演示。

16.3　使用 PowerTOSSIM 观察一个传感器节点的 CPU 和射频收发器中消耗的能量。 [441]

第八部分

Wireless Sensor Networks: Principles and Practice

案例研究

案例研究1：远程医疗服务

本章将介绍一个基于无线传感器网络的重要应用——远程医疗服务（Tele-Healthcare）的案例。本章素材来源于作者（Hu）的研究工作。本章内容主要参考了作者（Hu）及其同事早先发表和出版的文献，包括［Hu08，Hu2009a，Hu2009b，Hu2009c，Hu2009d，Hu2009f，Sunil08a］等。

17.1 引言

当前，特别是在发达国家，心血管疾病已经成为发病和死亡的最大诱因［MCHunink97］。2000年世界健康报告显示，冠状动脉疾病（CAD）每年都会导致大约700万人死亡，男性死亡人数占所有男性死亡人数的13%，女性死亡人数则占据12%的比例。因此，开发低成本高品质的心脏医疗保健系统成为一项严峻的挑战。

许多新型的心血管疾病医疗服务系统应运而生，例如初级预防、次级预防和患者自我保健计划等。这些新的系统进而促进了以院外监护和患者随访为基础［CardioNet08］的新型护理方法的发展［LAShort98］。因此，提供了与心脏患者联系新模式的远程心电图治疗系统的发展和应用引起了广泛的关注［Istepanian04］。绝大多数远程系统都采用了可佩带的装置（例如便携式心电图仪、血压计和脉搏仪）对心脏患者生理状态数据进行远程收集（包括心电图、血压和脉搏等）。

445

在医院或疗养院，心电图传感器连接成的无线自组网是一种很有前景的自动对心跳异常进行检测的办法［Martin00］。如今，有很多心电图机被冠以“便携式”的称呼——但这往往并不意味着它们是微小的，事实上，这些装置中的大部分需要电源接口来供电，而且十分笨重，以至于需要用货车才能进行移动。



正如第1章中所提到的，无线传感器网络最大的优势在于传感器体积小，价格低廉且功耗较低。只要其中任何一项特征缺失，就可以把这样一个网络系统归类为设计上更简单的普通无线网络或是自组织网络。

可以将心电图传感器互联起来，形成一个低功耗的医疗自组传感网（Medical Ad hoc Sensor Network，MASN），这可以在很大程度上提高心电图监测的便携性和时效性。同时，MASN也可以被看作是一种特殊形式的无线传感器网络。一个简单的MASN如图17.1所示。每一个患者的心电图信号将被自动收集和处理（如进行模拟－数字转换），然后以无线方式传输至一个远程心电图服务器进行数据分析（如通过数据分类发现心律不齐的症状）。如果其中一个心电图传感器报告了异常的心跳信号，在医生诊室与患者无线设备（例如传呼机或移动电话）之间的应急通信渠道会发出警报并为患者提供一些必要的医疗建议（例如服用药物和进行其他进一步的治疗）。另外，在一个典型的MASN中，患者的心电图传感器节点可以通过相邻传感器节点传输数据，从而实现多跳通信。



图 17-1　远程心电图传感器网络（MASN）

446

17.2　远程心电图传感器网络的硬件设计

心电图传感器和射频通信硬件

一个远程心电图传感器网络（MASN）由多个无线心电图通信单元组成。每一个单元称作一个移动平台（mobile platform）。如图 17-2 所示，每个平台由一个能与三导联的心电图监测系统进行连接的定制传感器板组成。这个心电图监测系统装置在一个无线通信板上（也称作射频节点）。在心电图传感器板收集到患者有效的心电图数据后，射频节点提供有限的本地信号处理功能（例如过滤心电图噪声），然后无线通信系统将心电图信号传回服务器以进行特征模式信息的提取。

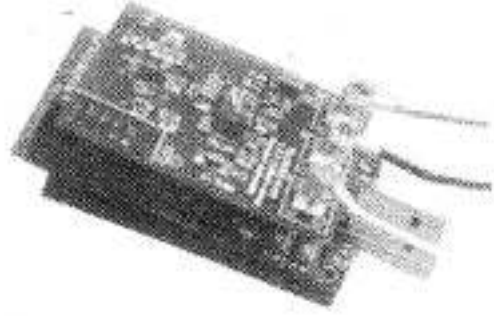


图 17-2　移动平台的外观（包括心电图传感器和射频模块）

MASN 移动平台的逻辑结构模块如图 17-3 所示。注意，感知芯片检测来自患者身体的模拟信号输入，例如心电图（心跳信号）、SpO_2（氧浓度水平）和体温等。



现在有许多有关心电图传感器的商业产品。然而，如果这些传感器只能生成模拟信号，就无法将所有的心电图传感器互联成一个网络，因为所有这些传感器需要 CPU 和射频芯片。还要记住，即使一个心电图传感器可以与一个网络进行连接，无线传感器网络的定义也已经指出每个传感器节点都有严重的资源限制。而且无线传感器网络协议都应该适用于大规模传感器互联（多于 1000 个传感器）。所有这些无线传感器网络特征使得它们的设计富有挑战性，在本章中，平均每个传感器网络都用来保护数以千计的患者。

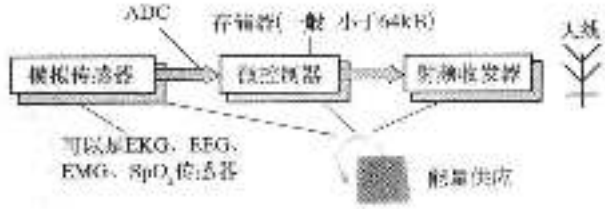


图 17-3 MASN 移动平台：逻辑结构

原始射频节点（如图 17-2 所示）是 Crossbow 公司生产的 TelosB 节点 [Crossbow08]。它提供了一个 10kB 的片上 RAM 和通信范围达 125 米的集成板载天线的 IEEE TelosB Chipcon 无线通信芯片 [Chipcon08]。通过使用 TI MSP430 微控制器 [Ti08]，由于 TelosB 板载的 ADC 带扩展槽，能与定制的传感器电路板连接，因此它能够良好地运行。但是 TelosB 节点也有着一些缺点。首先，当在大规模网络中部署时花费过高（2009 年每一个 TelosB 节点的价格大概是 200 美元）。其次，根据心电图信号传回服务器的频率的不同，TelosB 节点连续运行时间大约是 3～6个月，这无法满足大多数医疗应用要求寿命至少为一年的需要。第二，它的无线通信组件无法加强也无法被替换（如无法使用更好的无线信号收发器/天线以覆盖更大的范围）。

综合上述原因，一般使用 Ember CPU-RF 芯片 [Ember08] 构建射频节点。如图 17-4 所示，射频电路板的内核是微控制器（MCU）/ZigBee [ZigBee08] 收发器单元。

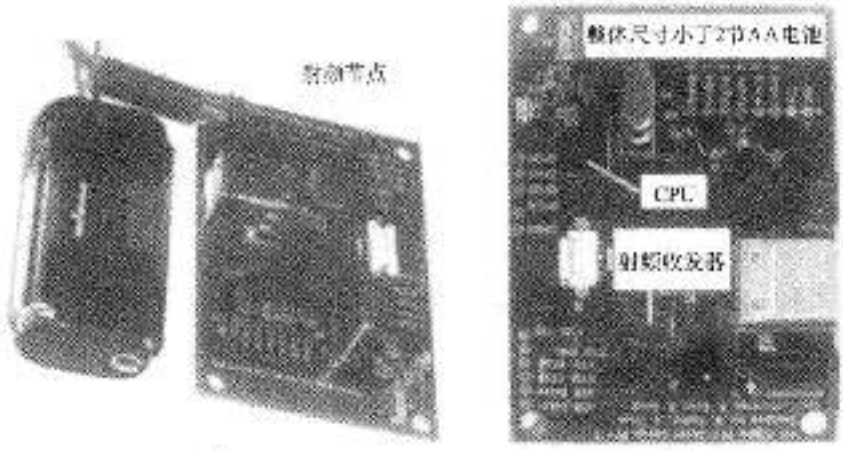


图 17-4 带 ECG 射频通信功能的定制射频板

447
~
448

在做出最后决定之前还要考虑到所有可能的选择。例如，既可以使用独立的 MCU 和收发器，也可以使用将两者一体化的 SoC（System-on-Chip，片上系统）。选用 SoC 是因为它的实现成本低、编程的复杂程度低，并且印刷电路板（Printed Circuit Board，PCB）布局更简单（由于电路布局中的部分更少）。


奇思妙想

和分离芯片相比，SoC 芯片通常可节约制造成本，而且 SoC 芯片内部的芯片间接口已经经过优化。如果使用分离芯片，工程师在进行芯片与芯片连接时容易出错。但另一方面，分离芯片的优点在于可以通过对任意组件进行替换来优化改善系统的性能。

图 17-5 展示了一个心电图传感器板和射频板之间的连接。射频板首先接收模拟的心电图数据（来自患者身体），将数据转换成数字格式，然后使用网络协议将数据打包，最后将数据包通过射频天线发送出去。它的射频接收器也可以从相邻的节点接收心电图数据，以实现患者与患者之间的多跳通信。



图 17-5　ECG 传感器与射频节点间的通信

心电图模拟传感器板是基于哈佛大学 CodeBlue 团队的研究成果［CodeBlue06］设计而成的。传感器板上的心电图导联线的扩展引脚与标准 3 导联心电图监测系统兼容，颜色编排也参照标准 3 导联心电图监测系统。尽管存在各种类型的导联，但这个系统的设计适合任意一体式 3 导联心电图连线。

如果不能用真实患者进行试验，为了测试方便，可以使用心电图生成器硬件模拟不同的心跳信号。如图 17-6 所示，所使用生成器的原型是 430B 模型，12 导联 ECG 生成器。该生成器 449
可以在六个不同预设频率（每分钟心跳数为 60、75、100、120、150 以及 200）和六个不同预设振幅（0.1、0.2、0.5、1.0、2.0 和 5.0mV）下提供完整的 PQRST 波形。它还可以生成方波。如果未来这个系统可以适应 12 导联监测系统，那么它可以提供一个非常好的测试接口。图 17-6 还显示了 430B 心电图模拟器和无线通信板之间的连接。

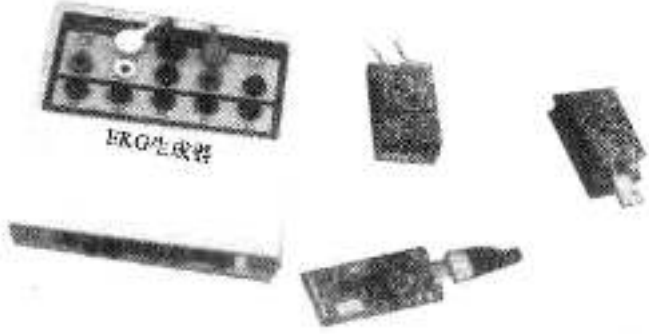


图 17-6　430B 心电图患者模拟器


奇思妙想

在许多国家，对于使用人体或动物进行医学测试都有着严格的政策规定，可能需要经过很长的申请过程才能得到许可。值得庆幸的是，现在已经有许多非常精确的商业信号生成器可以模拟人体不同类型的参数。例如上面提到的 430B 模型，12 导联心电图信号生成器就可以模拟许多种类的心脏病信号。

17.3　可靠的 MASN 通信协议

17.3.1　增强的基于聚类的 MASN 数据传输

对患者的心电图信号进行快速、可靠的检测是非常重要的。我们使用一种基于聚类的能量感知的心电图信号收集机制。在这种机制下，心电图数据以融合数据包的形式可靠地传输到汇

450 聚节点（即服务器）[Sunil08，Sunil08a]。


奇思妙想

在分布式计算中，聚类是一个很好的想法。它的基本思想是根据一些共同的属性（例如相近的物理位置和相似的CPU能力）将节点分成不同的“簇”。通常来说，每个簇都有一个成员作为簇首节点（Cluster Head，CH）。簇与簇之间的通信是通过簇首节点与簇首节点之间的联系实现的。现在有一些有关聚类机制的研究问题，例如在移动节点下簇的形成、簇首节点的选择规则、簇内和簇间的路由机制，簇的大小以及可靠聚类。

由于考虑到传感器节点能量水平的确定、事件触发簇的形成以及基于簇成员密度和事件相似度的可量性动态适应等因素，我们提出的MASN路由机制不同于LEACH［WBHeinzelman02］以及其他的聚类机制。具体介绍如下。

首先假设传感器节点能获知它们的最大能量（E_{max}）、残余能量（E_r）和临界能量（E_{th}）。这里的临界能量指传感器节点能够识别出它们处于n个能量级别中任一级所需的最小能量。残余能量小于临界能量的传感器节点属于第0级能量水平。开始，传感器节点的能量按照如下方式划分为n个能量级别：

$$n = \left\lfloor \log_x \frac{E_{max}}{E_{th}} \right\rfloor \tag{17.1}$$

其中，第L级能量水平的能量范围定义为该级最高能量值和最低能量值的差，x是该能级最大与最小能量值的比值。x的值取决于应用的要求。传感器节点的能量水平（L）由以下式决定：

当$E_r < E_{th}$时，$L=0$

当$E_r \geqslant E_{th}$时，$L = n - \left\lfloor \frac{E_{max}}{E_r} \right\rfloor$ (17.2)

如果事件参数的振幅超越了预定阈值Δ，传感器节点就会参与簇的形成过程。这里Δ的值取决于已测定的事件参数。

在形成簇的过程中，拥有最高能量水平的传感器节点将有机会成为簇首节点，以确保该簇
451 有更长的寿命。在缺少高能量水平传感器节点的区域内，低能量水平的传感器节点则主动成为簇首节点。这样做主要是为了确保汇聚节点能够进行可靠事件检测。然后传感器节点根据能量水平和AMRP值选择它们的簇首节点。这里，AMRP值定义为r个相邻节点能够与要求成为簇首节点的节点进行通信所需要的平均最小功率级别，如下式所示［OYounis04］：

$$\mathrm{AMRP} = \frac{\sum_{i=1}^{r} \mathrm{MinPWR}_i}{r} \tag{17.3}$$

其中：

MinPWR_i表示一个节点x_i（$1 \leqslant i \leqslant r$）与CH节点通信所需要的最小功率级别。

r是相邻节点的数量。

传感器节点根据它们自身的能量水平，通告自己成为簇首节点。声称自己成为簇首节点的传感器节点用最大功率（MaxPWR）向它的相邻节点广播通告消息。归一化AMRP定义为AMRP与MaxPWR的比值。

其他接收到这些通告消息的传感器节点会根据一个簇头能量水平和通信能量的函数决定是否加入一个簇首节点。每一个传感器节点在向其他节点通告自己成为簇首节点之前都需要等待一段随机的时间。发出通告消息的延迟时间由传感器节点能量水平（L）与归一化的平均最小

可达能量（nAMRP）的函数决定。

汇聚节点根据某个事件在时间 T 内所需报告的数据包总数给该事件赋一个可靠性值（REL）。这里的可靠性函数根据簇内传感器节点的数目和簇事件接近度分布在事件区域形成的簇内。事件区域里的每一个簇内通过多跳通信，将融合数据包头中簇成员的数量发送给汇聚节点。通过分析融合数据包中已经测定的事件参数的值，汇聚节点使能够知道哪些簇首节点与事件最为接近。汇聚节点按下式为每个簇赋可靠性值：

$$CR_i = \frac{\mathrm{REL} * (J_i)(m_i)}{\sum_{i=1}^{z} J_i m_i} \tag{17.4}$$

其中：

CR_i 是赋予第 i 个簇的可靠性值。

z 是簇的总数。

J_i 是该簇的事件接近度。

m_i 是簇内传感器节点的数目。

如果 $J_i = 1$，那么根据簇的成员密度，可靠性分布在所有的簇中。通过赋予 J_i 一个较高的值，汇聚节点可以从更接近事件的簇内获取更多的数据包。由于簇与簇之间的差异，事件接近度参数 J_i 在 0～1 之间变化。

如果事件向其他区域传播，那么汇聚节点将更改簇的可靠性数值。不同区域的传感器同样会根据已测定的事件参数值形成。这种汇聚节点（即服务器）的动态可靠性调整在最大程度获取事件信息方面将非常有效。

452

17.3.2　MASN 的路由性能

能量消耗：在 MASN 网络设计中，一个重要的问题是能量的消耗。实验表明，传感器电池中大部分能量都是在无线通信中消耗的，而不是在本地数据处理（例如心电图数据压缩）或是本地数据感知（如图 17-7 所示）的过程中消耗的。因此任何一个 MASN 网络协议（例如寻找最佳路径）都应该有较低的复杂度以减少能耗。

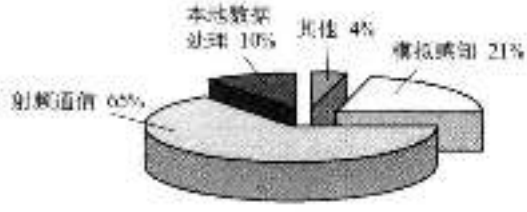


图 17-7　MASN 的能量消耗

吞吐量：为了更好地对患者的健康状况进行观察，传感器需要以很高的报告频率发送数据，同时也要以高数据速率通过无线方式发送大量的感知数据。图 17-8 为数据包接收率（接收的数据包与发送的数据包的比值）在不同发送率（每秒发送的网络数据包）下的关系。可以看出，当发送率大于 25 个包/秒时，随着发送率的提高 MASN 的性能急剧下降。因此在每一个医疗传感器上使用一个合适的报告频率是十分重要的。

可扩展性：我们在传感器节点数量增加（如果每个患者携带一个传感器那么意味着更多的患者）的情况下对 MASN 的性能进行了研究。如图 17-9 所示，尽管此时有大量的患者，但是 MASN 系统依然保持良好的性能（接收率大于 80%）。这表明 MASN 适用于大型疗养院。

移动性：在使用者有移动行为的情况下，我们对 MASN 延迟性能进行了测试。当前，如果使用者的移动过快（例如在 30MPH 的速度下），那么系统无法实现实时数据收集（延迟大于 10 秒），如图 17-10 所示。这是未来研究的一个课题。

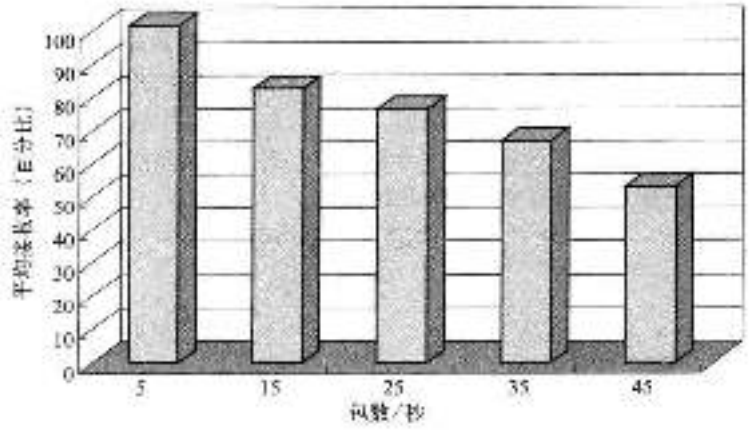


图 17-8　接收率与发送率的关系

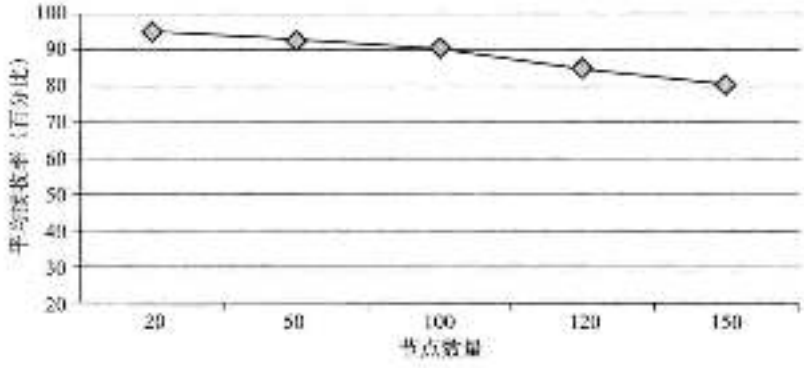


图 17-9　接收率与节点数目的关系

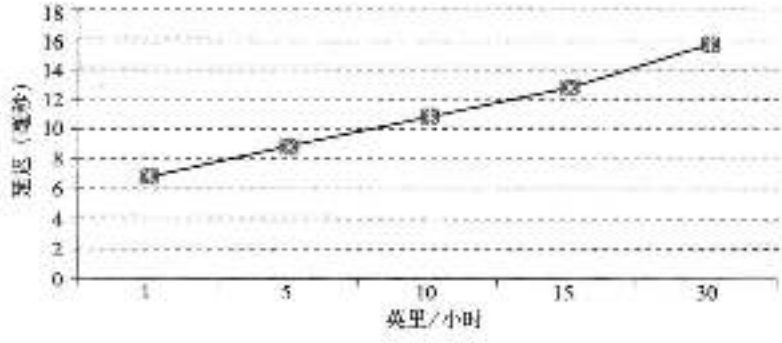


图 17-10　端到端延迟与移动性的关系

延迟：融合数据包延迟指从一个事件被传感器节点检测到的时间开始，到第一个聚合事件数据包被传送到汇聚节点所花费的时间。这个参数表示网络对于事件的反应速度。在该机制和 HEED 机制中，认为事件发生时簇立即形成。图 17-11 所示的实验结果显示，由于存在启动阶段，HEED 机制下第一个融合数据包传送到汇聚节点所需时间比其他融合数据包更长。在这个阶段，系统中没有数据包报告给汇聚节点，簇在开销消息的帮助下得以形成。而在该机制下，在簇形成的同时事件数据包就会发送到汇聚节点。

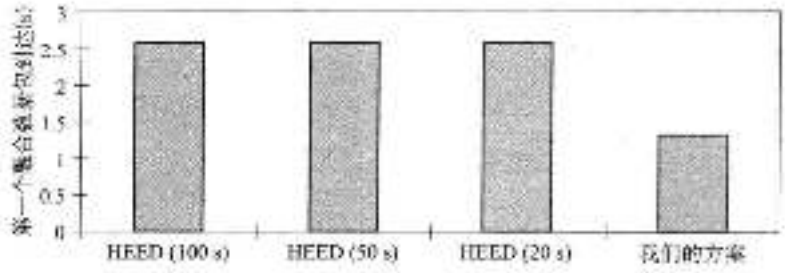


图 17-11　从簇首节点到汇聚节点的第一个融合数据包延迟

17.4　MASN 的软件设计

心电图传感器节点无线通信软件

所有的 MASN 射频节点控制软件在一个称作 TinyOS［TinyOS07］的专门的操作系统中运行。在接收所有患者心电图数据的医疗服务器中，可以监控整个 MASN 网络的拓扑结构。如图 17-12 所示，如果两个患者距离足够近，那么就会在他们之间显示一个无线信号连接，表示在他们之间传输心电图数据的可能性。

在该软件中，通过从服务器到任意心电图传感器节点的无线命令传输，可以远程控制心电图传感器的性能参数（例如心电图监测阈值）。如图 17-13 所示，心电图服务器（即 MASN 工作站）控制参数能够发送给传感器，从而改变传感器的检测频率（也就是每秒需要收集的心电图数据量）。

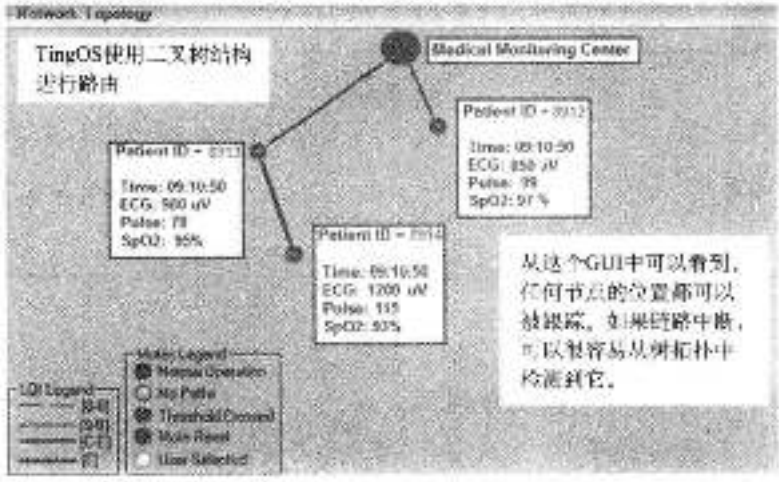


图 17-12　有三个心脏病患者的简单疗养院的控制软件截图

VitalDust Plus［CodeBlue06］的作用是显示数据。它有两个模块，一个是用于移动平台进行采样和通过无线通信传播重要信号数据而设计的 TinyOS 软件，另一个则是以图形化方式显示重要信号的 Java 图形用户界面应用程序。

图 17-14 是一个提高版软件的屏幕截图。它显示一个服务器正在从两个移动平台 mote30 和 mote40 处接收患者相关数据。患者数据图形区域显示的是与所选移动平台相关的心电图波形。只有那些从当前选定的移动平台上获取的数据才会被送至 MATLAB 进行信号处理。连接质量区域显示的则是与所选移动平台相关的无线信号的质量。

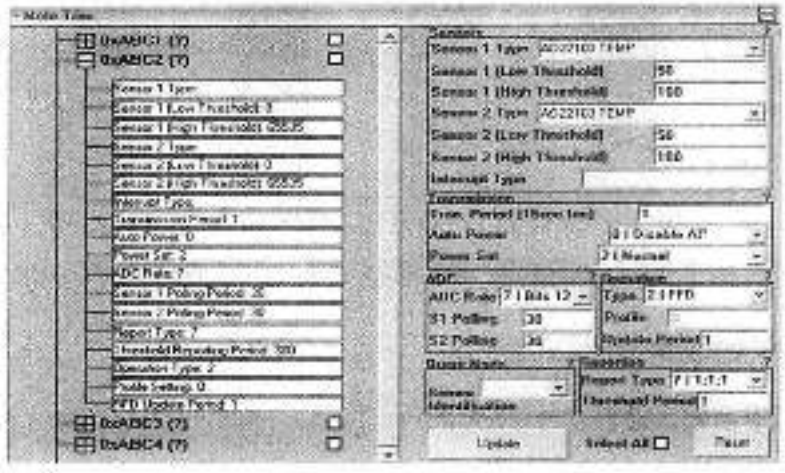


图 17-13　调整 ECG 传感器参数的远程控制软件

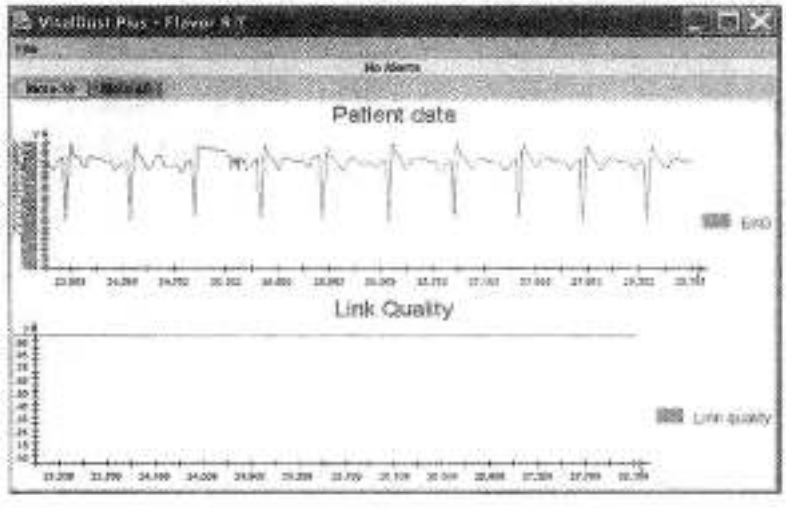


图 17-14　提高版 VitalDust Plus

17.5　RFID 和可穿戴传感器的集成

每一个 RFID（无线射频识别）系统都由一个读取器和若干标签（tag）组成。读取器包含一根天线和一个收发器，它可以读取标签上的信息，并且可以将它们传送给处理设备。标签，或应答器（transponder），则是一个包含了射频电路以及需要发送的信息的集成电路。

在供应链/物流监测应用中，RFID 已经被许多组织（如沃尔马公司和美国国防部[Wang06]）用于替代通用产品代码（UPC）。相关产业和远程医疗服务公司已经意识到 RFID 的用处和成效，并开始将其应用到医疗服务之中以减少失误并节省开支。比如说，中国台北医科大学附属医院使用 RFID 标签来定位患者和医院的设备，推出了基于位置的医疗服务（LBMS），并获得了成功[Wang06]。Exavera's eSheperd 也在一个 Wi-Fi 网络中使用 RFID 来追踪患者、医护人员和相关物资的供应，包括医护人员给患者分发的药品[Exavera07]。En-Vision America 则通过使用带 ScriptTalk 的 RFID，创造了一个为视障患者提供处方信息的新途径

457

EnVision07]。当患者使用 ScriptTalk 读取器来提交一个处方时，药店系统的软件则使用专用的小型打印机生成并打印一个贴纸智能标签。这个存储了处方信息的智能标签由药剂师放置到处方容器内。在家中，患者则可以使用一个便携式的 ScriptTalk 读取器，通过语音合成技术来读取标签中的信息。

为什么需要将 RFID 和传感器集成起来？正如上文所讨论的，传感器和 RFID 有着不同的应用场景。另一方面，它们代表了两种互补性的技术，如果两种技术能够有效地结合则会有很大的优势。以下列举 RFID 和无线传感器结合后的一些优点：

1）RFID 有追踪患者的功能，因此在残疾患者的追踪应用中，它与无线传感器能够相互补充。无论患者去哪，定位在不同位置的 RFID 标签都可以告诉他们是否处于危险状态（例如在一个有陡坡的道路上）。RFID 同样可以帮助他们辨别需要摄入的药物。

2）MSN（Mobile Sensor Network，移动传感器网络）传感器可以提供 RFID 所不能提供的各类医疗状况感知功能。更重要的是，这些无线传感器具有 CPU，可以运行数据处理和数据通信软件。RFID 读取器没有这种智能处理功能，它们可以利用 MSN 传感器向控制中心发送残疾患者的追踪信息。因此 MSN 使得 RFID 可以进行远程传输。

3）RFID 是一个单跳无线系统，也就是说，一个 RFID 读取器只能够与单跳的标签（通常小于 3 米）进行通信。通过与传感器网络运行集成，RFID 能够利用无线传感器网络中的多跳以及高级网状网络协议来处理任意数量的 RFID 读取器以及它们之间复杂的通信问题。

4）RFID 通常是一个封闭系统，也就是说，当前的商用 RFID 读取器除了一些简单参数配置之外，不允许改变其内部控制软件。通过与无线传感器集成，程序员可以更新传感器存储器中的代码来间接地处理 RFID 标签的数据。例如，一个传感器的程序可以在数据库内存储残疾患者的追踪信息以分析患者的动作。

在实际的工作中，我们已经将无线医疗传感器（EKG/EMG）成功地与 RFID 读取器集成到一块电路板上，而且还开发了一个集成软件对 RFID/传感器的行为进行控制。此外，还可以确保不同的 RFID/传感器板在通信时不发生冲突。 458

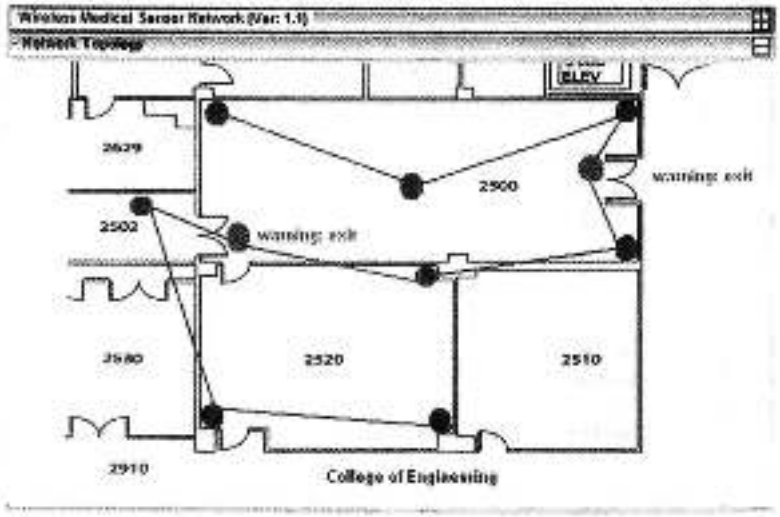


图 17-15　患者追踪 RFID

RFID 在道路指引上的应用：RFID 可以用来与残疾患者保持联系。如果患者接近了一个道路不通的区域，这个患者的 RFID 读取器就会发出一个警告信号（例如患者的传感器会发出声音）。系统软件会自动绘制患者移动的轨迹（只要在途经的每个地方都有一个标签）。图 17-15 为记录的结果。

RFID 在药物服用指导上的应用：成熟的 RFID 软件可以让程序员在一个 RFID 标签上填写所有的处方信息，再将标签贴到药瓶上。这个标签的内容包含患者的姓名、处方的名称、药瓶中药物的量、每次服用的剂量、每天服用的剂量以及编程节点（读取器）ID。在处理标签的时候，ID 会被打印到标签上。RFID 应用的屏幕截图如图 17-16 所示。程序员将 RFID 标签放置在读取器上，在所有字段中填上之前介绍的信息并且点击“写标签”按钮。如果程序员想检查是否所有的信息都被完全正确地写入，那么他可以点击“读标签”按钮，屏幕就会显示之前输入的信息。如果发现从标签上读出的信息有错误，程序员可以对相应的字段进行更正并重新将信息写入标签。按钮上的状态栏能提示程序员读取或写入操作是否成功。

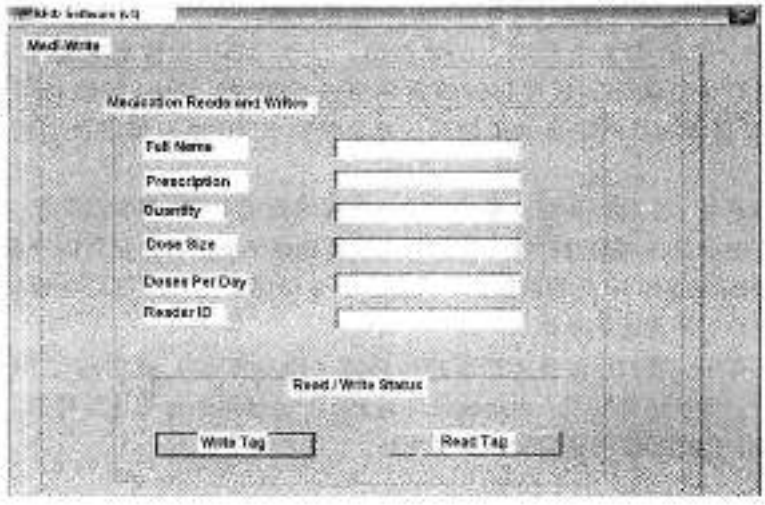


图 17-16　患者药物服用管理的 RFID

RFID 数据库：在上述药物服用指导应用的后台程序中，当“写标签”按钮被按下时，将
459 会在数据库中插入新项，新项包含了程序员提交的所有信息以及一个新的 RFID 标签 ID，这个标签的 ID 将存储在数据库的 tagID 字段中。在数据库内还有一个 dosesToday 字段，该字段表示当天已经服用的药物的剂量。当插入新项时，这个数值被设定为 0。当前数据库内容的屏幕截图如图 17-17 所示。

tagID	name	rx	QTY	doseSize	dosesDay	dosesToday	readerID
[illegible]	Mary Wee	Prozac	36	1	2	0	3
[illegible]	Linda Ineffirle	Zoloft	48	2	2	0	4
[illegible]	Al Barr	Tylenol	48	2	2	0	2
[illegible]	Lou Smith	Aspirin	36	2	2	0	1

图 17-17　RFID 数据库截图

在如上所述的数据库中，数据库字段名称的意义如图 17-18 所示。

数据库栏名称	含义
Name	姓名
Rx	处方
QTY	剂量
doseSize	每次服用剂量
doseDay	每天服用剂量
readerID	读取器 ID

图 17-18　数据库字段名称

在几种情况下患者和护理人员将收到警告。这些情况包括：数据库中不存在给定的药物；患者试图服用不是自己的药物；患者没有药物；患者服用的药物没有达到当天所需要的剂量。对于以上任一错误，屏幕上都会弹出一个提示框，提示框包含了时间标签、没有正确服用药物的患者信息和弹出原因，以告知管理人员。

通过检查数据库中 tagID 字段的 RFID 标签 ID，就能够确定一种药物是否在数据库中。如果它不在药物数据库中，甚至从未进入过系统之中，那么将发出一个警告。同样，为了检查一个患者是否在服用不属于自己的药物，软件系统可以查询数据库中的 tagID。

如果数据库中某一项的 readID 字段的值与患者节点 ID 不匹配，就说明这个患者在试图服用不属于他自己的药物，此时必须发出警告。通过检查数据库中标签 ID 对应的 QTY 栏的值，就可以很容易地判断出患者的药物是否用完。如果 QTY 栏中的数值为 0，则应该发出警告，这样能够再开具处方。

最后，可能也是最重要地，必须要通过检查确保患者没有过量服用药物。在数据库中查询标签 ID 对应的 doseToday 字段的值，如果该字段的值与 doseDay 字段的值相等，那么患者就不能再服用药物。如果患者试图过量服用，那么必须立刻发出警告以保证患者不会过量服用。图 17-19 ~ 图 17-22 为上述情况下弹出的警告信息。

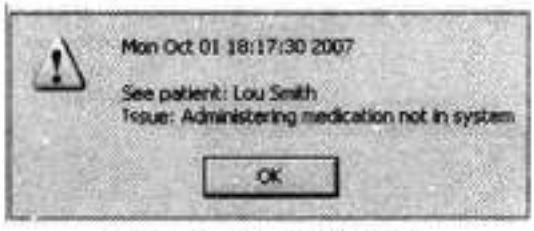


图 17-19　数据库中无此药物



图 17-20　患者试图服用不属于他的药物

460
~
461

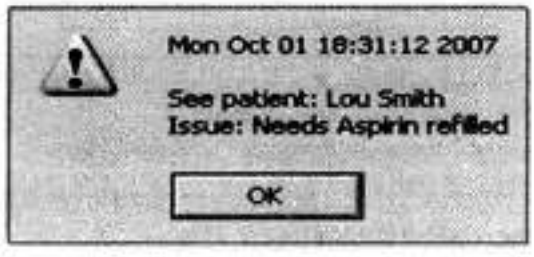


图 17-21　患者没有药物

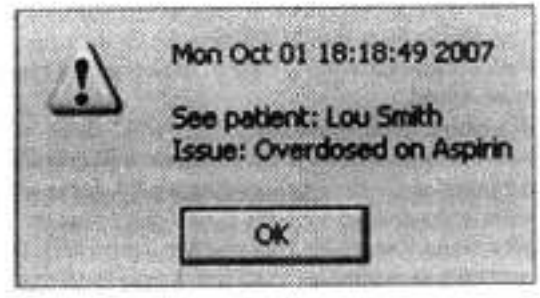


图 17-22　患者过量服用药物

问题与练习

17.1 多项选择题

1. 如果心电图传感器没有射频电路，那么以下哪些功能不能实现？（ ）

A. 对心电图信号中的热噪声和电源声进行过滤。

B. 将模拟信号转换为数字格式。

C. 通过除去标准心电图模式来压缩心电图数据流。

D. 能够接近的簇首节点来传送数据。

462

2. 心电图数据在无线传送的过程中会因为一些因素被损坏。以下哪些不是主要考虑因素？（ ）

A. 在每一跳中无线传输错误的累积。

B. 无线信号被障碍物干扰。

C. 转发节点的 CPU 在处理数据过程中无意地更改数据。

D. 网络攻击者可以对心电图数据进行篡改。

3. 设计一个定制的通信板代替 Crossbow 节点的优势包含如下哪些选项？（ ）

A. 更低的单元成本。

B. 更大的射频通信范围。

C. 软件更容易地变更和修正。

D. 所有选项。

4. RFID 技术的优势（与一般的产品条形码比较）包括如下哪些？（ ）

A. 产品代码读取距离更长。

B. 能读取更多的产品信息。

C. A 和 B。

D. 运行无线网络协议的可能性。

5. 在所讨论的工作中，患者的心电图信号和药品信息能够在同一个网络数据包中同时被读取，是因为以下哪些因素？（ ）

A. RFID 读取器和射频节点集成在同一块电路板上。

B. 能够从心电图传感器和药物传感器中获取感知数据。

C. 患者关于药物名称的声音信号。

D. 以上均不是。

17.2 一些远程医疗系统通过移动电话传输医疗数据。另外一些则通过使用部署在建筑物中的无线 LAN 来传输数据。与这两种方法相比，基于传感器网络的远程医疗系统有什么优势？（提示：思考一些传统方法不太适用的情境。）

17.3 通过网络查找资料，总结 RFID 系统的运行原则、设计中面临的挑战以及应用范例。

17.4 一些公司设计了专门的 RFID 产品以监控药物服用的过程。例如，可以把 RFID 标签放置在每一片药片附近以监测服用的剂量。请通过网络查找资料，列举一些这类应用的范例。

17.5 本章提出了一种增强的基于簇的传感器网络路由机制。与 LEACH 机制相比，它有哪些优势？

17.6 除了心电图传感器，一些其他的医用传感器也在研究和开发之中。你能否通过对葡萄糖传感器和胰岛素泵的监测，对糖尿患者进行一些独立的研究，并且总结出它们的运行原理？

463

案例研究2：灯光控制

本章将介绍一个有趣的无线传感器网络应用——灯光控制。本章内容主要参考了文献[Hamin06，Heemin07]。

18.1 引言

通过光传感器收集实时灯光信息是大有用处的。实时数据能够解释由于灯丝老化、供电电压变化、固定装置位置的变化、颜色过滤等原因，光的特征（如光强度）是如何变化的。如果需要在特定区域中长时间保持所需灯光强度，会花费大量时间和精力，因此测量实时灯光显得尤为重要。尽管目前能够通过手持型人工曝光表测量光强度[Sek，Kan]，但这些设备并不支持自动的灯光控制。必须由手动在空间中不同的点间移动它们。照相机只提供反射光的强度，因此对与表面和材料无关的入射光的测量进行研究是十分重要的。

在文献[Hamin06]中，提出了一个称为Illuminator的智能光控制系统。它能够通过使用光传感器，根据用户的要求测量入射光，探测并调整最佳光控配置文件。Illuminator能够使用无线传感器网络技术帮助媒体制作人员定型、控制、布置表演灯光以及制作电影。Illuminator有三个任务（根据灯光布置和用户的限制）：1）推荐最佳的传感器部署；2）收集灯光特征；3）调整最佳光控配置满足用户的限制。这些限制代表着灯光的美学效果要求，并包含场地要求的灯光强度或者灯光条件的高级描述。

465



尽管本书提供了不同的无线传感器网络应用，但所有的系统都有类似的网络难题，如传感器部署、拓扑控制、路由协议和减少拥塞。然而，这些系统又有不同的“模拟传感器”设计以及相应的感知数据分析软件。因此，你可以将更多的学习注意力放到具体的模拟传感器硬件部分以及与射频电路板的接口上。

Heemin[Hamin06]系统（在本章中称其为Heemin）假设Illuminator在控制时，灯都有固定的位置，而Heemin并不了解灯的位置与特征等信息。使用全景云台追踪和聚光是一个常用技术，并且很容易实现[Spo]。Heemin使用移动的标签，通过固定的照明设备，使得移动的舞台元素、设备以及演员被照亮。标签是一个能够感受到灯光强度及其位置的单个实体。

为了在特定位置产生所需的灯光水平，需要根据调光水平了解灯光投射模式以及灯光强度。这种信息称为灯光特征，获取该信息的过程称为光特征提取。光特征提取过程为：在每个调光水平一个接一个地开灯，然后使用无线光传感器测量入射光强度。

Illuminator系统可以根据用户的要求和已有的灯光数据，检测出最佳使用下的灯光水平。Illuminator系统还可以用不同的物理光源重建类似甚至相同的灯光效果。为了得到预期的重建效果，它要求在当前设置下进行再次对每盏灯特征提取。比如要在不同的地点或者不同的时间实现相同的灯光效果。如果电影拍摄时布景有任何改变，即使电影摄制组试图建立一个与之前相同的灯光系统，相应灯光设置也会发生改变。Heemin使用Illuminator系统记录灯光设置的结果（不仅仅是设备的物理设置和分配）。

Heemin Illuminator系统的典型使用场景如图18-1所示。基于用户的限制和可用的光传感

器，Illuminator 系统会推荐传感器的布置方案。然后，用户根据 Illuminator 的推荐方案部署传
466 感器。Illuminator 系统使用已部署的传感器，自动提取灯光特征。一旦光特征提取过程结束，除了用于连续照明或追踪的光传感器，其余的传感器都会停止使用。在排练过程中，Illuminator 系统通过在线动态调整光控配置来控制灯的状态。用户或许希望在反复排练时改善灯光的设计，那么可以通过以下几种方法来完成，例如改变用户的限制、增加更多传感器进行光特征提取、增加或移动光传感器加强照明效果等。例如，如果用户发现对于一些区域而言，由于有障碍物，光特征提取并不充分，那么就需要在有障碍物的地方部署更多的传感器。

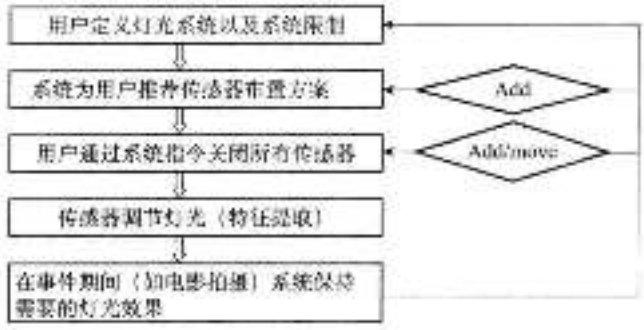


图 18-1　Illuminator 系统的使用场景

无线传感器网络能帮助进行灯光的持续性管理。电影拍摄事件的顺序和观众看到电影情节顺序有很大不同。电影情节片段的拍摄顺序要遵循费用最小化以及演员、工作人员和地点的利用最优化的原则。值得注意的是，在不同时间拍摄的片段需要在连贯播出的时候让人看不出拍摄场景差异，或者，系统为了播出的效果必须要能够控制差异的程度。因此，需要在每个片段
467 中监视并重现出一样的灯光质量（照明度和颜色）。这样在不同时间或地点拍摄出的片段才不会出现巨大的差别，这些差别通常不易被人们感知但却影响电影胶片。Hoomin［Hamin06］已经将注意力集中到灯光仪表化，并将其作为先进电影技术（ATC）［WMH02］的首要部分。



Hoomin［Hamin06］在介绍自动灯光控制的重要性时举了一个有趣的例子：在《指环王》三部曲中，三部不同的电影在拍摄时间和还度安排有极大不同。尽管这部电影的工作人员超过了2400人，但依靠人工记录数据而且环境一直在变化，维持连续性是十分困难的。因此，连续性管理对道具、场景、演员以及镜头信息、灯光来说都是必需的，尽管连续性管理一般用于非技术性元素的管理上。

Illumimote 支持三种不同的灯光感应模式：入射光强度、颜色浓度和入射光角度（从最强光源出来的射线角度）。它同时还支持以下两种感应模式：姿态和温度。Illumimote 系统与商用曝光表性能相当，能够满足无线传感器网络应用中规模和能量的要求。

Illumimote 的设计标准包括以下方面：1）灯光强度和色温感知；2）抵抗红外能量的健壮性；3）动态范围宽；4）快速反应时间；5）高准确性。Illuminote 系统和 Crossbow 的 Mica 节点是兼容的，Crossbow 是一个无线传感器网络系统研究和开发的常用平台。

18.2　Illumimote 系统的传感器

Illuminator 系统的数据获取基于以下三个基本的照明特征：信号强度（浓度）、频率（颜

色）以及传送向量（入射光角度和传感器姿态）。Illuminote 系统中包括以下传感器：1）入射光强传感器：检测入射光强度，具有商用曝光表的精度（如 ekonic 1.558Cine［Sek］曝光表）；2）色度传感器：检测红色、绿色和蓝色的值用于计算色温［WS82］；3）入射光角度传感器：判断最强光源的入射光角度；4）环境传感器：在电路板上加入的一些额外的传感器，以提供更多有关本体感受的信息［BFM06］。系统还包含一个基于重力的姿态传感器（加速计），当传感器与地面不平行时进行地平面相对转换。温度传感器用于检测可能在高强度光环境下出现的过热状况。

18.3　系统结构

Heemin Illuminote 的系统结构如图 18-2 所示。图中只显示了八个信道中的一个——光传感器信道。分配的光传感器信道个数取决于获取照明特征的检测电路的个数。例如，色温单元需要红、绿、蓝三个信道的光度。来自八个光获取单元以及四个环境单元的信号经过信道选择单元复用，然后将信号送到 ADC 中转换为 10 位数字信号。输出数据通过 I2C 数据总线或者采用线缆输出的 16550A 可兼容 UART 连接，传输到传感器节点（Heemin 使用 MicaZ 节点）。Illuminote 单元的操作可以通过 I2C 总线或者本地板载的 Atmel Atmega48 微控制器由节点直接控制。

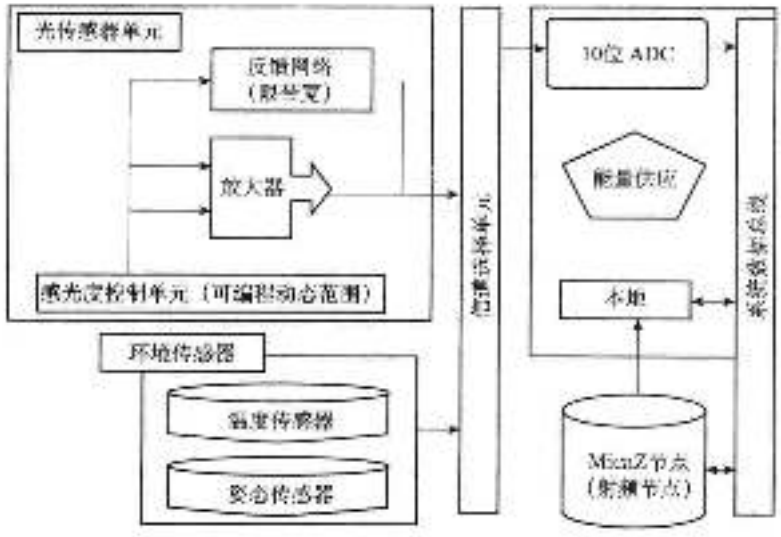


图 18-2　Illuminote 系统结构图

18.4　校准

为了把数字化的传感器数据转换为灯光强度（lux），Heemin 使用两个系数进行线性转换（即 $y = ax + b$，其中 y 为转换之后的 lux 值，x 是 ADC 的读数，a 和 b 是系数）。具体地说，Heemin 通过以下三个步骤求出最佳的系数值：

步骤 1：在 2D 平面上绘出 Illuminote 系统的 ADC 读数关于商业曝光表所测的 lux 值的函数图。

步骤 2：使用 MATLAB 的 polyfit 指令拟合出一条最适合描述 ADC 值的线性直线（即 $y = ax + b'$）。

步骤 3：通过将线性直线（$y = ax + b'$）投影到 $y = 18$ 上，计算 $a = 1/a'$，$b = -b'$ 得到校准

系数 a 和 b。然后将收集到的 ADC 输出值和校准后的 a 和 b 用于 Illuminote 的六个感应设置。


奇思妙想

MATLAB 的 polyfit 功能已经在许多线性回归和函数插值计算中使用。它的基本思想是使用多项式函数拟合一系列实验数据点。

光源的色温是黑体（black-body）辐射源的开氏温度，此温度与光源的色度是相匹配的[WS82]。然而，由于许多光源（除了炽热光源）像辐射源一样不释放辐射，因此 Heemin 使用关联色温（Correlated Color Temperature，CCT）表示光源色温。色温校准可以通过设置将红、绿、蓝（RGB）原始读数转换为 RGB 相对光强度的因数实现。

18.5　系统评估

为了评估 Illuminote 系统的性能，Heemin 将无线感知系统与 Illuminote 集成到一起。实验设置情况如图 18-3 所示。为了建立光源，使用钨平衡白炽灯，它在 6 英尺距离处产生接近 3200K 的色温以及近 3000lux 的亮度。这在电影设置中是一种定义良好的特定色温的常用光源。为了产生不同的亮度，Illuminote 被放置在 11 个不同的点上，在离光源 6 - 36 英尺的距离上每隔 3 英尺放一个点。

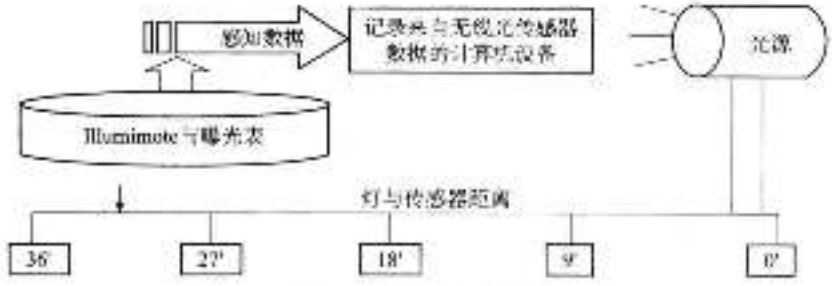


图 18-3　系统实验设置

Heemin 为实验的无线感知系统开发了三个嵌入式软件组件。首先，将感光度控制软件下载到 Illuminote 主板。第二，Heemin 在 MicaZ 节点中为 Illuminote 驱动和光感知应用编写程序。它使用 SOS，这是一个由 UCLA 的 NESL 开发出的针对节点级别的无线传感器网络的操作系统[HKS05]。最后，在计算机上（基站），使用 Java 程序监控和记录光的测量值，用可视化接口进行实时调试与分析。图形用户界面（GUI）如图 18-4 所示，它显示了 Illuminote 的实时状态。使用 GUI，能够方便地在图形界面下测试以及评估 Illuminote，GUI 的设计也朝着能够被电影摄影师使用的方向前进。

整个 Illuminator 系统可以被分为三个子系统：传感器网络、Illuminator 核心和 DMX 控制器和调节器。图 18-5 为整个 Illuminator 灯光控制系统的连接示意图。

Heemin 使用传感器网络测量光强度与传感器的位置。传感器网络又由两个子网络构成：一个是 Cricket 定位系统，另一个是拥有 Illuminator 光感应板的单跳 MicaZ 网络[PFG06]。三个 Cricket [Priyantha05] 节点被用作预校准其位置的信标节点，同时一个 Cricket 节点与每个 Illuminote 配合以定位光传感器模块。为了管理两个传感器网络平台，同时运行两个 Java 模块：SerialServer 模块用于 Illuminator 核心与传感器网络间的接口，Localizer 模块用于根据超声波范

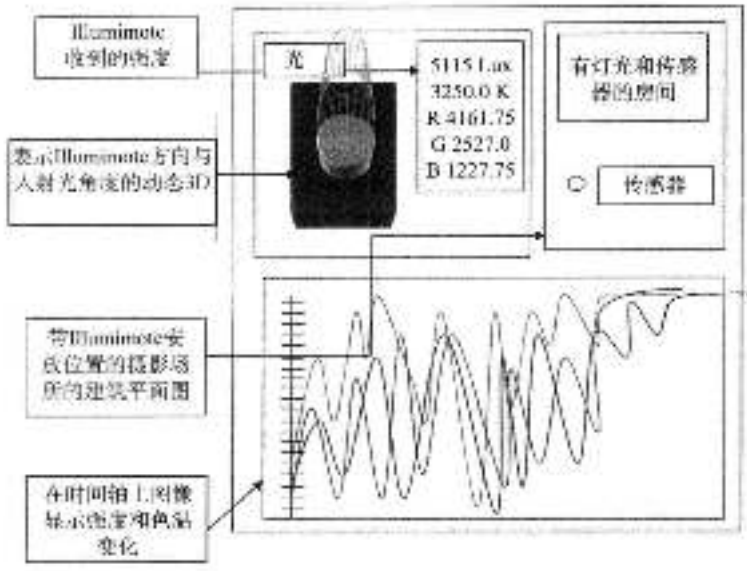


图 18-4　实时可视化图形界面截图

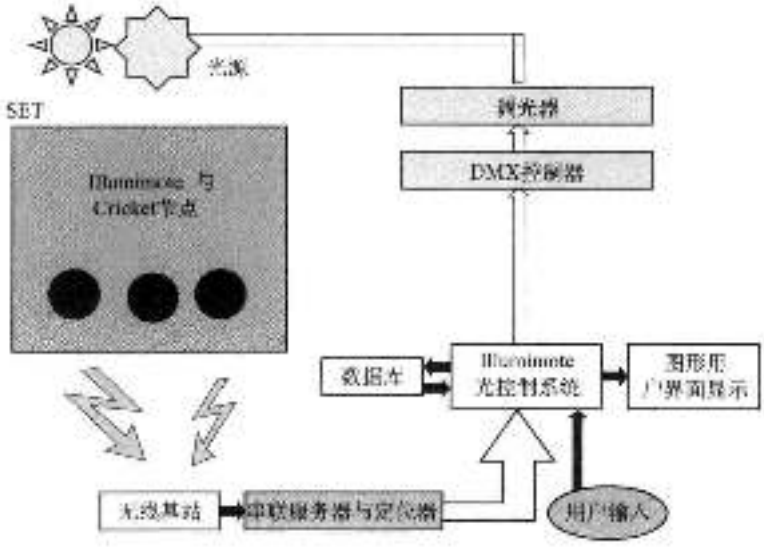


图 18-5　Illuminator 灯光控制系统

监测域对 Cricket 节点的位置进行计算。

问题与练习

18.1　多项选择题

1. 本章提出的光控传感器网络的重要性包括：（　　）

A. 实时数据解释了光强度和色温等特征在灯丝老化、供电电压改变、固定装置位置改变和颜色过滤等原因下，是如何随着时间以及放置位置变化的。

B. 通过对灯光的实时测量，不需要维持在多个现场的特定区域及长时间下所需要的光强度。

470
-
472

C. 目前的手持人工眼光表还没有整合到支持自动光控制的系统中，必须以人工方式在空间中不同点间移动。

D. 以上所有选项。

2. Illuminator 的作用不包括：（ ）

A. 光源与光源之间的通信。

B. 提取光的特征。

C. 调整出最佳灯光控制配置，满足用户要求。

D. 推荐传感器的部署方案。

3. 无线传感器网络有利于灯光的管理，是因为：（ ）

A. 需要在每次拍摄中监控和重复光的质量（亮度和颜色），以使在不同时间或位置拍摄出的片段不会出现巨大差别。

B. 由于光环境一直在变化，而仅靠人工记录数据来维持一致性是十分困难的。

C. 许多高预算的电影在发行之前需要进行大量的后期数字图像处理，其花费的成本很高。

D. 以上所有选项。

4. Illuminator 系统在以下哪些情况下是有用的？（ ）

A. 娱乐与媒体制作。

B. 实现满足室外公共场合要求的照明控制策略。

C. 水下成像的声学感知。

D. A 和 B 都是。

5. Illuminator 系统获取的数据包括：（ ）。

A. 信号方度（强度）

B. 频率（颜色）

C. 发射向量（入射光的角度和传感器姿态）

D. 以上所有选项

18.2 为什么我们需要使用传感器网络控制灯光？

18.3 光传感器有哪些功能？

18.4 解释 Illuminator 结构的各个模块。

473 18.5 解释色温校准的原理。

参考文献

[Aburrose05] A. Burrose, U. Roedig, and C. Sreenan, μ-MAC: An energy-efficient medium access control for wireless sensor networks, in *Proceedings of the Second IEEE European Workshop on Wireless Sensor Networks*, Istanbul, Turkey, January 2005, pp. 70–80.

[Abolhasan04] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, A review of routing protocols for mobile ad hoc networks, *Ad Hoc Networks (Elsevier)*, 2, 1–22, January 2004.

[Achandra00] A. Chandra, V. Gummalla, and J.O. Limb, Wireless medium access control protocols, *IEEE Surveys and Tutorials*, 3(2), 2–15, Second Quarter, 2000.

[ADoucet01] A. Doucet, N. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, 2001.

[Aelhoiydi04] A. El Hoiydi and J.-D. Decotignie, WiseMAC: An ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks, *IEEE Computers and Communications*, 1, 244–251, July 2004.

[Akan05] Ö.B. Akan and I.F. Akyildiz, Event-to-sink reliable transport in wireless sensor networks, *IEEE/ACM Transactions on Networking*, 13(5), 1003–1016, October 2005.

[Akcan06] H. Akcan, V. Kriakov, H. Brönnimann, and A. Delis, GPS-free node localization in mobile wireless sensor networks, in *Proceedings of the Fifth ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE '06)*, Chicago, IL, June 25, 2006, ACM, New York, pp. 35–42.

[Akyildiz02] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless sensor networks: A survey, *Computer Networks (Elsevier)*, 38, 393–422, March 2002.

[Akyildiz04] I.F. Akyildiz and I.H. Kasimoglu, Wireless sensor and actor networks: Research challenges, *Ad Hoc Networks (Elsevier)*, 2, 351–367, October 2004.

[Akyildiz04a] I.F. Akyildiz, D. Pompili, and T. Melodia, Challenges for efficient communication in underwater acoustic sensor networks, *SIGBED Review*, 1(2), 3–8, July 2004.

[Akyildiz07] I.F. Akyildiz, T. Melodia, and K.R. Chowdhury, A survey on wireless multimedia sensor networks, *Computer Networks (Elsevier)*, 51(4), 921–960, March 2007.

[Ahuja93] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ, February 1993.

[AManjeshwar01] A. Manjeshwar and D.P. Agrawal, TEEN: A routing protocol for enhanced efficiency in wireless sensor networks, in *Proceedings of the 15th IEEE International Parallel and Distributed Processing Symposium*, San Francisco, CA, April 2001, pp. 2009–2015.

[Anderson02] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, Wireless sensor networks for habitat monitoring, in *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, Atlanta, GA, September 2002.

[AMD03] AMD, AM49DLG400G Stacked Multi-Chip Package (MCP) Flash Memory and SRAM, 2003, http://www.amd.com/usen/assets/content_type/white_papers_and_tech_docs/26090a.pdf

[APerrig00] A. Perrig, R. Canetti, J. Tygar, and D. Song, Efficient authentication and signing of multicast streams over lossy channels, in *IEEE Symposium on Security and Privacy*, Oakland, CA, 2000.

[APerrig01] A. Perrig et al., SPINS: Security protocols for sensor networks, in *Proceedings of ACM MOBICOM*, Rome, Italy, 2001.

[APerrig02] A. Perrig, R. Canetti, J.D. Tygar, and D. Song, The TESLA broadcast authentication protocol, *CryptoBytes*, 5(2), 2–13, Summer/Fall 2002.

[ASavkin03] A. Savkin, P. Pathirana, and F. Faruqi, The problem of precision missile guidance: LQR and H1 control frameworks, *IEEE Transactions on Aerospace and Electronic Systems*, 39(3), 901–910, July 2003.

[ASavvides01] A. Savvides, C.-C. Han, and M. Srivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, in *Proceedings of the Seventh ACM International Conference on Mobile Computing and Networking (Mobicom)*, Rome, Italy, July 2001, ACM, New York, pp. 166–179.

[ASPNES03] J. Aspnes and G. Shah, Skip graphs, in *Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Baltimore, MD, January 12–14, 2003, pp. 384–393.

[ASrinivasan06] A. Srinivasan, J. Teitelbaum, and J. Wu, DRBTS: Distributed reputation-based beacon trust system, in *Second IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC '06)*, Indianapolis, IN, 2006, pp. 277–283.

[ASrinivasan08] A. Srinivasan and J. Wu, A survey on secure localization in wireless sensor networks, in *Encyclopedia of Wireless and Mobile Communications*, B. Furht, Ed., CRC Press/Taylor & Francis Group, Boca Raton, FL, 2008 (accepted for publication).

[Atmel01] Atmel Corporation, Atmega103(L) Datasheet, 2001, Atmel Corporation: http://www.atmel.com/atmel/acrobat/doc0945.pdf

[Atmel08] Atmel Corporation, http://www.atmel.com, 2008.

[AWoo01] A. Woo and D. Culler, A transmission control scheme for media access in sensor networks, in *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking (MOBICOM '01)*, Rome, Italy, July 2001, pp. 221–235.

[AWood02] A. Wood and J. Stankovic, Denial of service in sensor networks, *IEEE Computer*, 35(10), 54–62, October 2002.

[AWood03] A. Wood, J. Stankovic, and S.H. Son, Jam: A jammed-area mapping service for sensor networks, in *Real-Time Systems Symposium*, Cancun, Mexico, 2003.

[Bao01] L. Bao and J.J. Garcia-Luna Aceves, A new approach to channel access scheduling for ad hoc networks, in *Seventh Annual International Conference on Mobile Computing and Networking*, Rome, Italy, 2001, pp. 210–221.

[Bchen02] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, *ACM Wireless Networks*, 8(5), 481–494, September 2002.

[Bdavid02] D. Braginsky, Estrin rumor routing algorithm for sensor networks, in *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, September 2002, ACM, New York, pp. 22–31.

[BFM06] J. Burke, J. Friedman, E. Mendelowitz, H. Park, and M.B. Srivastava, Embedding expression: Pervasive computing architecture for art and entertainment, *Journal of Pervasive and Mobile Computing*, 2(1), 1–36, February 2006.

[Bharghavan93] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, MACAW: A media access protocol for wireless LAN's, in *Proceedings of ACM SIGCOMM Conference (SIGCOMM '94)*, London, U.K., August 1994, pp. 212–225.

[BHW97] B.H. Wellenhoff, H. Lichtenegger, and J. Collins, *Global Positions System: Theory and Practice*, Springer-Verlag, New York, 1997.

[Bkrap00] B. Krap and H.T. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in *Proceedings of MobiCom 2000*, Boston, MA, August 2000, pp. 243–254.

[BKusy07] B. Kusy, G. Balogh, A. Ledeczi, and M.M.J. Sallai, in-Track: High precision tracking of mobile sensor nodes, in *Fourth European Workshop on Wireless Sensor Networks (EWSN '07)*, Delft, the Netherlands, January 2007.

[Blom85] R. Blom, An optimal class of symmetric key generation systems, in *Advances in Cryptology: Proceedings of EUROCRYPT '84*, T. Beth, N. Cot, and I. Ingemarsson, Eds. Lecture Notes in Computer Science, Vol. 209, pp. 335–338. Springer-Verlag, Berlin, Germany, 1985.

[Blundo93] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, Perfectly secure key distribution for dynamic conferences, in *Advances in Cryptology—CRYPTO '92*, E. Brickell, Ed., Lecture Notes in Computer Science, Vol. 740, pp. 471–486. Springer-Verlag, Berlin, Germany, 1993.

[Bulusu00] N. Bulusu, J. Heidemann, and D. Estrin, GPS-less low cost outdoor localization for very small devices, *IEEE Personal Communications Magazine*, 7(5), 28–34, October 2000.

[Bulusu05] N. Bulusu, C. Chou, W. Hu, S. Jha, A. Taylor, and V. Tran, The design and evaluation of a hybrid sensor network for cane-toad monitoring, in *Proceedings of Information Processing in Sensor Networks*, Los Angeles, CA, April 2005.

[BWarneke01] B. Warneke, M. Last, B. Liebowitz, and K.S.J. Pister, Smart dust: Communicating with a cubic millimeter computer, *IEEE Computer*, 34(1), 44–51, 2001.

[BYCHKOVSKIY03] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, A collaborative approach to in-place sensor calibration, in *Proceedings of IPSN '03*, Palo Alto, CA, 2003.

[CardioNet08] CardioNet Inc. has developed an integrated technology and service—mobile cardiac outpatient telemetry (MCOT)—Which enables heartbeat-by-heartbeat, ECG monitoring, analysis and response, at home or away, 24/7/365. On CardioNet project details, please see: http://www.cardionet.com/

[Carlos04] C. Pomalaza-Ráez, Wireless sensor networks energy efficiency issues, (Lecture notes), Fall 2004, University of Oulu, Oulu, Finland.

[CcEnz04] C.C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Peiris, WiseNET: An ultralowpower wireless sensor network solution, *IEEE Journal*, 37(8), 62–70, August 2004.

[CERPA01] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, Habitat monitoring: Application driver for wireless communications technology, in *Proceedings of ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, San Jose, Costa Rica, 2001.

[Chi06] C. Ma and Y. Yang, Battery-aware routing for streaming data transmissions in wireless sensor networks, *Mobile Networks and Applications*, 11, 757–767, 2006.

[Chieh-Yih05] C.-Y. Wan, A.T. Campbell, Member, IEEE, and L. Krishnamurthy, Pump-slowly, fetch-quickly (PSFQ): A reliable transport protocol for sensor networks, *IEEE Journal on Selected Areas in Communications*, 23(4), 862–872, April 2005.

[Chipcon08] On the Chipcon Inc. RF transceiver products, please see http://www.chipcon.com, Visited in June 2008.

[CIntanagonwiwat00] C. Intanagonwiwat, R. Govindan, and D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, in *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00)*, Boston, MA, August 2000. ACM Press, New York, pp. 56–67.

[CKarlof03] C. Karlof and D. Wagner, Secure routing in sensor networks: Attacks and countermeasures, *Ad Hoc Networks*, Special issue on *Sensor Network Applications and Protocols (Elsevier)*, 1(2–3), 293–315, September 2003.

[Chang04] J.-H. Chang and L. Tassiulas, Maximum lifetime routing in wireless sensor networks, *IEEE/ACM Transactions on Networking*, 12(4), 609–619, August, 2004.

[Chehri06] A. Chehri, P. Fortier, and P.-M. Tardif, Application of ad-hoc sensor networks for localization in underground mines, in *Proceedings of the IEEE Annual Wireless and*

Microwave Technology Conference (WAMICON '06), Melbourne, FL, December 4–5, 2006, pp. 1–4.

[CMUcam08] The CMUcam2. http://www-2.cs.cmu.edu/~cmucam/cmucam2/index.html

[CodeBlue06] M. Welsh and B. Chen, CodeBlue: Wireless sensor networks for medical care, Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA, 2006.

[CORMEN01] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd edn. The MIT Press, Cambridge, MA, 2001.

[CPERKINS00] C. Perkins, *Ad Hoc Networks*. Addison-Wesley, Reading, MA, 2000.

[CSavarese02] C. Savarese, Robust positioning algorithms for distributed ad hoc wireless sensor networks, Master's thesis, University of California at Berkeley, Berkeley, CA, 2002.

[Cschurgers01] C. Schurgers and M.B. Srivastava, Energy efficient routing in wireless sensor networks, in *Proceedings of IEEE MILCOM '01*, Vienna, VA, October 2001, Vol. 1, pp. 357–361.

[Crossbow08] On all wireless sensor network products (including motes, sensor boards, gateway, etc.) from Crossbow Inc., please see: http://www.xbow.com, Visited in June 2008.

[CYWan02] C.Y. Wan, A.T. Campbell, and L. Krishnamurthy, PSFQ: A reliable transport protocol for wireless sensor networks, in *Proceedings of the ACM WSNA*, Atlanta, GA, September 2002, pp. 1–11.

[DAI 04] H. Dai, M. Neufeld, and R. Han, ELF: An efficient log-structured flash file system for micro sensor nodes, in *SenSys '04: Proceedings of the Second International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, 2004, ACM Press, New York, pp. 176–187.

[Dallas08] Dallas Semiconductor, DS2401 Silicon Serial Number: http://pdfserv.maxim-ic.com/arpdf/DS2401.pdf

[Ddclark90] D.D. Clark and D.L. Tennenhouse, *Architectural Considerations for a New Generation of Protocols*, 20(4), 200–208, September 1990, ACM.

[DELIN00] K.A. Delin and S.P. Jackson, Sensor web for in situ exploration of gaseous biosignatures, in *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, 2000.

[DFox99] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, Monte Carlo localization: Efficient position estimation for mobile robots, in *AAAI 1999*, Orlando, FL, 1999, pp. 343–349.

[DLiu05] D. Liu, P. Ning, and W. Du, Detecting malicious Beacon nodes for secure location discovery in wireless sensor networks, in *25th IEEE International Conference on Distributed Computing Systems (ICDCS '05)*, Columbus, OH, 2005, pp. 609–619.

[DLiu05a] D. Liu, P. Ning, and W. Du, Attack-resistant location estimation in sensor networks, in *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, CA, April 2005, pp. 99–106.

[DLM91] D.L. Mills, Internet time synchronization: The network time protocol, *IEEE Transactions on Communications*, 39(10), 1482–1493, October 1991.

[DLM92] D.L. Mills, Network time protocol (version 3): Specification, implementation, and analysis, Technical Report, Network Information Center, SRI International, Menlo Park, CA, March 1992.

[DOOLIN05] D. Doolin and N. Sitar, Wireless sensors for wildfire monitoring, in *SPIE Symposium on Smart Structures and Materials*, San Diego, CA, March 2005.

[Doyle93] M. Doyle, T.F. Fuller, and J. Newman, Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell, *Journal of the Electrochemical Society*, 140(6), 1526–1533, 1993.

[DSchmidt07] D. Schmidt, M. Krämer, T. Kuhn, and N. Wehn, Energy modelling in sensor networks, *Advances in Radio Science*, 5, 347–351, 2007. See http://www.adv-radio-sci.net/5/347/2007/

[DSR] D. Johnson, D. Maltz, and J. Broch, The dynamic source routing protocol for multihop wireless ad hoc networks, in *Ad Hoc Networking*, C. Perkins, Ed., Addison-Wesley, Boston, MA, 2001.

[DSPComm08] DSPComm, available: www.dspcomm.com, Visited in 2008.

[Dulman03] S. Dulman, T. Nieberg, J. Wu, and P. Havinga, Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks, *IEEE WCNC*, New Orleans, LA, March 2003.

[DuW03] W. Du, J. Deng, Y.S. Han, and P.K. Varshney, A pairwise key pre-distribution scheme for wireless sensor networks, in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, Washington DC, October 27–30, 2003, ACM, New York, pp. 42–51.

[DuW05] W. Du, J. Deng, Y.S. Han, P.K. Varshney, J. Katz, and A. Khalili, A pairwise key predistribution scheme for wireless sensor networks, *ACM Transactions on Information and System Security*, 8(2), 228–258, May 2005.

[Eslopez06] E.E. Lopez, J. Vales-Alonso, A.S. Martinez-Sala, J. García-Haro, P. Pavón-Mariño, and M.V.B. Delgado, A wireless sensor networks MAC protocol for real time applications, *Personal and Ubiquitous Computing*, 12(2), 111–122, January, 2008, ACM.

[Elnahrawy2003] E. Elnahrawy and B.R. Badrinath, Cleaning and querying noisy sensors, in *Proceedings of the Second ACM International Conference on Wireless Sensor Networks and Applications*, San Diego, CA, September 19, 2003.

[Ember08] On the RF and CPU chips from Ember Inc., see http://www.ember.com, Visited in June 2008.

[EnVision07] En-Vision America, ScripTalk, http://www.envisionamerica.com/scriptalk/scriptalk.php, downloaded 22 August 2007.

[EShih01] E. Shih et al., Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, in *Proceedings of the ACM MOBICOM*, Rome, Italy, July 2001, pp. 272–286.

[ESouto04] E. Souto et al., A message-oriented middleware for sensor networks, in *Proceedings of the Second International Workshop Middleware for Pervasive and Ad-Hoc Computing (MPAC '04)*, Toronto, Ontario, Canada, October 2004, ACM Press, New York, pp. 127–134.

[Exavera07] Exavera Technologies, eShepherd overview, http://www.exavera.com/healthcare/eshephend.php, downloaded 22 August 2007.

[FCristian89] F. Cristian, Probabilistic clock synchronization, *Distributed Computing*, 3, 146–158, 1989.

[Fenp05] W. Feng, E. Kaiser, W.C. Feng, and M.L. Baillif, Panopres: Scalable low-power video sensor networking technologies, *ACM Transactions on Multimedia Computing, Communications, and Applications*, 1(2), 151–167, May 2005.

[Finn87] G. Finn, Routing and addressing problems in large metropolitan-scale internetworks, Technical Report, ISI/RR-87-180, USC/ISI, March 1987.

[Fkuhn03] F. Kuhn, W. Roger, and Z. Aaron, Worst-case optimal and average-case efficient geometric ad-hoc routing, in *Proceedings of International Symposium on Mobile Ad Hoc Networking & Computing*, Annapolis, MD, June 2003.

[FLi06] F. Li, Y. Li, W. Zhao, Q. Chen, and W. Tang, An adaptive coordinated MAC protocol based on dynamic power management for wireless sensor networks, in *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, Vancouver, British Columbia, Canada, July 2006, ACM, New York, pp. 1073–1078.

[FSimjee06] F. Simjee and P.H. Chou, Everlast: Long-life, supercapacitor-operated wireless sensor node, *ISLPED*, Tegernsee, Germany, 2006.

[Ftobagi75] F. Tobagi and L. Kleinrock, Packet switching in radio channels, Part II: Hidden-terminal problem in carrier sense multiple access and the busy-tone solution, *IEEE Transactions on Communications*, 23(12), 973–977, December 1975.

[FYe02] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, A two-tier data dissemination model for large-scale wireless sensor networks, in *Eighth Annual International Conference on Mobile Computing and Networking (ACM Mobicom '02)*, Atlanta, GA, September 2002. ACM Press, New York, pp. 148–159.

[Fyc01] F. Ye, S. Lu, and L. Zhang, A scalable solution to minimum cost forwarding in large sensor networks, in *Proceedings of IEEE INFOCOM '01*, Anchorage, AK, April 2001, pp. 304–309.

[GANESAN03] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, An evaluation of multi resolution storage in sensor networks, in *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, CA, 2003.

[GANESAN03a] D. Ganesan, D. Estrin, and J. Heidemann, Dimensions: Why do we need a new data handling architecture for sensor networks? *SIGCOMM Computer Communication Review*, 33(1), 143–148, January 2003.

[Ganesan01] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, Highly resilient, energy-efficient multipath routing in wireless sensor networks, *ACM SIGMOBILE Mobile Computing and Communication Review*, 5(4), 11–25, 2001.

[Gay03] D. Gay, P. Levis, R.V. Behren, M. Welsh, E. Brewer, and D. Culler, The nesC language: A holistic approach to networked embedded systems, in *Proceedings of SIGPLAN '03*, 2003.

[GGolub96] G. Golub, *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, 1996.

[Girod01] L. Girod and D. Estrin, Robust range estimation using acoustic and multimodal sensing, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, Maui, HI, October 2001.

[GloMoSim] X. Zeng, R. Bagrodia, and M. Gerla, GloMoSim: A library for parallel simulation of large-scale wireless networks, in *Proceedings of the 12th Workshop on Parallel and Distributed Simulations, PADS '98*, May 26–29, 1998, Banff, Alberta, Canada.

[Glu04] G. Lu, B. Krishnamachari, and C.S. Raghavendra, An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks, in *Proceedings of the IEEE 18th International Parallel and Distributed Processing Symposium*, Santa Fe, NM, April 2004, pp. 224–231.

[Hamin06] H. Park, Design and implementation of a wireless sensor network for intelligent light control, PhD dissertation, Department of Electrical Engineering, UCLA, Los Angeles, CA, 2006. Also see: http://nesl.ee.ucla.edu/fw/documents/journal/2006/Sensors_Illumimote_HeeminPark.pdf

[HanC05] C. Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava, A dynamic operating system for sensor nodes, in *Proceedings of the Third international Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, Seattle, WA, June 6–8, 2005, ACM, New York, pp. 163–176.

[Hartung06] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, FireWxNet: A multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments, in *Proceedings of the Fourth International Conference on Mobile Systems, Applications and Services (MobiSys '06)*, Uppsala, Sweden, June 19–22, 2006, ACM, New York, pp. 28–41.

[HARVEY03] N. Harvey, M.B. Jones, S. Saroiu, M. Theimer, and A. Wolman, Skipnet: A scalable overlay network with practical locality properties, in *Proceedings of the Fourth USENIX Symposium on Internet Technologies and Systems (USITS '03)*, Seattle, WA, March 2003.

[Hawkins80] D.M. Hawkins, *Identification of Outliers*. Chapman and Hall, New York, 1980.

[Heemin07] H. Park, J. Burke, and M.B. Srivastava, Design and implementation of a wireless sensor network for intelligent light control, in *Proceedings of the Sixth International Conference on Information Processing in Sensor Networks (IPSN '07)*, Cambridge, MA,

April 25–27, 2007, ACM, New York, pp. 370–379.

[Heinzelman02] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Communications*, 1, 660–670, October 2002.

[HKS05] C.-C. Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava, A dynamic operating system for sensor nodes, in *Proceedings of the Third International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, Seattle, WA, 2005, ACM Press, New York, pp. 163–176.

[Hojung07] H. Cha et al., Resilient, expandable, and threaded operating system for wireless sensor networks, in *IPSN '07*, Cambridge, MA, April 25–27, 2007.

[HOLMAN03] R. Holman, J. Stanley, and T. Ozkan-Haller, Applying video sensor networks to nearshore environment monitoring, *IEEE Pervasive Computing*, 2(4), 14–21, 2003.

[Honeywell08] Honeywell, 101 Columbia Road, Morristown, NJ 07962 USA. See: http://www.honeywell.com.

[Horn86] B.K.P. Horn, *Robot Vision*, 1st edn, The MIT Press, Cambridge, MA, 1986.

[Hschulzrinne96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, RTP: A transport protocol for real-time applications, RFC1889, January 1996.

[HU04] F. Hu and S. Kumar, Multimedia query with QoS considerations for wireless sensor networks in telemedicine, in *Proceedings of Society of Photo Optical Instrumentation Engineers International Conference on Internet Multimedia Management Systems*, Orlando, FL, September 2003.

[Hu08] F. Hu, M. Jiang, L. Celentano, and Y. Xiao, Robust medical ad hoc sensor networks (MASN) with wavelet-based ECG data mining, *Ad Hoc Networks Journal (Elsevier)*, 6(7), 986–1012, September 2008.

[Hu2009a] F. Hu, Y. Xiao, and Q. Hao, Congestion-aware, loss-resilient bio-monitoring sensor networking, *IEEE Journal on Selected Areas in Communications (JSAC)*, 27(4), 450–465, anuary 2009.

[Hu2009b] F. Hu, S. Lakdawala, Q. Hao, and M. Qiu, Low-power, intelligent sensor hardware interface for medical data pre-processing, *IEEE Transactions on Information Technology in Biomedicine*, 13(4), 656–663, May 2009.

[Hu2009c] F. Hu, M. Jiang, M. Wagner and D. Dong, Privacy-preserving tele-cardiology sensor networks: Towards a low-cost, portable wireless hardware/software co-design, *IEEE Transactions on Information Technology in Biomedicine*, 11(6), 617–627, November 2007.

[Hu2009d] F. Hu, L. Celentano, and Y. Xiao, Error-resistant RFID-assisted wireless sensor networks for cardiac tele-healthcare, *Wireless Communications and Mobile Computing (Wiley)*, 9, 85–101, February 2009.

[Hu2009e] F. Hu, P. Tilgman, S. Mokey, J. Byron, and A. Sackett, Secure, low-cost prototype design of underwater acoustic sensor networks, *Journal of Circuits, Systems, and Computers (World Scientific)*, 17(6), 1203–1208, 2008.

[Hu2009f] F. Hu, Q. Hao, M. Qiu, and Y. Wu, Low-power electroencephalography sensing data RF transmission: Hardware architecture and test, in *ACM MobiHoc 2009—The First ACM International Workshop on Medical-grade Wireless Networks (WiMD '09)*, New Orleans, LA, 2009.

[Huang07] T. Huang, K. Hou, H. Yu, E.T. Chu, and C. King, LA TinyOS: A locality-aware operating system for wireless sensor networks, in *Proceedings of the 2007 ACM Symposium on Applied Computing (SAC '07)*, Seoul, Korea, March 11–15, 2007, ACM, New York, pp. 1151–1158.

[Hui07] H. Song, Secure wireless sensor networks: Building blocks and applications, PhD dissertation, Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, 2007.

[Hwendi00] W. Heinzelman, A. Chandrashekaran, and H. Balakrishnan, Energy efficient communication protocol for wireless microsensor networks, in *Proceedings of 33rd*

Hawaii International Conference on Systems Sciences, Cambridge, MA, January 2000.

[HXia96] H. Xia, An analytical model for predicting path loss in urban and suburban environments, in *Proceedings of the Personal Indoor Radio Communication (PIRMC '96)*, Taipei, Taiwan, 1996.

[IBorg97] I. Borg and P. Groenen, *Modern Multidimensional Scaling Theory and Applications*, Springer, New York, 1997.

[IEEE07] IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and Metropolitan area networks—Specific requirements, *Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, pp. 120–121, July 2007.

[Iglewicz93] B. Iglewicz and D.C. Hoaglin, *How to Detect and Handle Outliers*, ASQC Basic References in Quality Control, ASQC Quality Press, Milwaukee, WI, 1993.

[IKhalil05] I. Khalil, S. Bagchi, and N.B. Shroff, Analysis and evaluation of SECOS, a protocol for energy efficient and secure communication in sensor networks, *Ad Hoc Networks Journal (ADHOC)*, 5(3), 360–391, 2007.

[Intel02] Intel Corp, Intel Press Release: Intel Builds World's First One Square Micron SRAM Cell, 2002: http://www.intel.com/pressroom/archive/releases/20020312tech.htm.

[Internet07] (post date: 7-18-07). http://robotics.eecs.berkeley.edu/~roosta/SIR12006.pdf

[INTERSEMA 2002] INTERSEMA, 2002. MS5534A barometer module, Technical Report (October). Go online to http://www.intersema.com/pro/module/file/da5534.pdf

[IPetersen99] I. Petersen and A. Savkin, *Robust Kalman Filtering for Signals and Systems with Large Uncertainties*, Birkhäuser, Boston, MA, 1999.

[IPSec] Security architecture for the Internet Protocol, RFC 2401, November 1998.

[Irhee06] I. Rhee, A. Warrier, J. Min, and L. Ki, DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks, in *Proceeding of IEEE MobiHoc*, Florence, Italy, May 2006, pp. 190–201.

[Irhee08] I. Rhee, A. Warrier, M. Aia, J. Min, and M.L. Sichitiu, Z-MAC: A hybrid MAC for wireless sensor networks, *IEEE/ACM Transactions on Networking*, 16(3), 511–524, June 2008.

[Issa06] I. Khalil, Mitigation of control and data traffic attacks in wireless ad-hoc and sensor networks, PhD thesis, Purdue University, West Lafayette, IN, 2006.

[Istepanian04] R.S.H. Istepanian, E. Jovanov, and Y.T. Zhang, Guest editorial introduction to the special section on M-health: Beyond seamless mobility and global wireless health-care connectivity, *IEEE Transactions on Information Technology in Biomedicine*, 8(4), 405–414, 2004.

[Jason03] J.L. Hill, System architecture for wireless sensor networks, PhD dissertation, Department of Computer Science, University of California at Berkeley, Berkeley, CA, Spring 2003.

[Jaein07] J. Jeong, X. Jiang, and D. Culler, Design and analysis of MicroSolar power systems for wireless sensor networks, Technical Report No. UCB/EECS-2007-24, http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-24.html

[JBeutel99] J. Beutel, Geolocation in a picoradio environment, Master's thesis, ETH Zurich, Zurich, Canton of Zurich, Switzerland, 1999.

[JElson02] J. Elson, L. Girod, and D. Estrin, Fine-grained network time synchronization using reference broadcasts, in *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA, December 2002, pp. 147–163.

[Jennifer08] J. Yick, B. Mukherjee, and D. Ghosal, Wireless sensor network survey, *Computer Networks*, 52(12), 2292–2330, August 22, 2008.

[JGProakis01] J.G. Proakis, E.M. Sozer, J.A. Rice, and M. Stojanovic, Shallow water acoustic networks, *IEEE Communications Magazine*, 39(11), 114–119, November 2001.

[John06] J.A. Stankovic, Wireless sensor networks, Department of Computer Science,

University of Virginia, Charlottesville, VA, 2006. See: http://www.cs.virginia.edu/~stankovic/psfiles/wsn.pdf

[Johnson05] J. Johnson, J. Lees, M. Ruiz, M. Welsh, and G. Werner-Allen, Monitoring volcanic eruptions with a wireless sensor network, in *Proceedings of the Second European Workshop Wireless Sensor Networks (EWSN '05)*, Istanbul, Turkey, January 2005.

[Jonathan08] J. Bachrach and C. Taylor, Localization in sensor networks, computer science and artificial intelligence laboratory, Massachusetts Institute of Technology, Cambridge, MA; http://people.csail.mit.edu/jrb/Projects/psschap.pdf; Visited in 2008.

[JUANG02] P. Juang, O. Hidenkazu, M. Martonosi, L. Peh, D. Rubenstein, and Y. Wang, Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet, *ASPLOS X*, San Jose, CA, October 2002.

[JZhao03] J. Zhao, R. Govindan, and D. Estrin, Computing aggregates for monitoring wireless sensor networks, in *Proceedings of the IEEE ICC Workshop Sensor Network Protocols Applications*, Anchorage, AK, May 2003, pp. 139–148.

[Ji04] J. Ai, J. Kong, and D. Turgut, An adaptive coordinated medium access control for wireless sensor networks, in *Proceedings of the Ninth IEEE International Symposium on Computer and Communications 2004*, Alexandria, Egypt, July 2004, Vol. 1, pp. 214–219.

[Jkulik02] K. Joanna, W. Heidemann, and H. Balakrishnan, Negotiation-based protocols for disseminating information in wireless sensor networks, *ACM Wireless Networks*, 8(2/3), 169–185, March–May 2002.

[Jli04] J. Li and G.Y. Lazarou, A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks, in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, April 2004, ACM, New York, pp. 55–60.

[JNal-karaki04] J.N. Al-Karaki, R. Ul-Mustafa, and A.E. Kamal, Data aggregation in wireless sensor networks—Exact and approximate algorithms, in *Proceedings of IEEE Workshop on High Performance and Routing 2004*, Ames, IA, April 2004, pp. 241–245.

[Joseph05] J. Polastre, R. Szewczyk, and D. Culler, Telos: Enabling ultra-low power wireless research, in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks 2005 (IPSN 2005)*, Los Angeles, CA, April 15, 2005, pp. 364–369.

[JPolastre04] J. Polastre, Interfacing Telos to 51-pin sensorboards, October 2004, http://www.tinyos.net/hardware/telos/telos-legacy-adapter.pdf

[Jpolastre04] J. Polastre, J. Hill, and D. Culler, Versatile low power media access for wireless sensor networks, in *Proceeding of Second International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, October 2004, ACM, New York, pp. 95–107.

[JRice00] J. Rice et al., Evolution of seaweb underwater acoustic networking, in *Proceedings of the MTS/IEEE OCEANS*, Providence, RI, September 2000, Vol. 3, pp. 2007–2017.

[Karlof04] C. Karlof, N. Sastry, and D. Wagner, TinySec: A link layer security architecture for wireless sensor networks, in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, Baltimore, MD, November 3–5, 2004, ACM, New York, pp. 162–175.

[Karthikeyan] K. Vaidyanathan, S. Sur, S. Narravula, and P. Sinha, Data aggregation techniques in sensor networks, Technical Report OSU-CISRC-11/04-TR60, Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, downloadable from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.937, Visited in 2009.

[Kaveh04] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach*, 1st edn. Prentice Hall, Englewood Cliffs, NJ, 2004, ISBN: 8178086468.

[KAY93] S. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, Prentice Hall, Upper Saddle River, NJ, 1993.

[Keoliver05] K.E. Oliver. Introduction to automatic design of wireless networks. *CrossRoads ACM Student Magazine*, 11(4), 1–4, 2005.

[Kjamieson03] K. Jamieson, H. Balakrishnan, and Y.C. Tay. Sift: A MAC protocol for event-driven wireless sensor networks, in *Proceedings of the Third European Workshop on Wireless Sensor Networks*, Zurich, Switzerland, Lecture Notes in Computer Science, Vol. 3868, pp. 260–275, Springer Link, New York, May 2003.

[KOkeya05] K. Okeya and T. Iwata. Side channel attacks on message authentication codes, in *Second European Workshop on Security and Privacy in Ad Hoc and Sensor Networks*, Visegrad, Hungary, July 2005.

[Kon] K. Minolta, Minolta Color Meter IIIf. http://konicaminolta.com. 2008.

[KRamakrishnan90] K. Ramakrishnan and R. Jain. A binary feedback scheme for congestion avoidance in computer networks. *ACM Transactions on Computer Systems*, 8(2), 158–181, May 1990.

[KSavgri02] K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, and E. Belding-Royer, A secure routing protocol for ad hoc networks, in *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP)*, Paris, France, 2002, pp. 78–87.

[Ksarvakar08] K. Sarvakar and P.S. Patel, An efficient hybrid MAC layer protocol utilized for wireless sensor networks, in *Proceedings of Fourth IEEE Conference on Wireless Communication and Sensor Networks '08*, Allahabad, India, December 2008, pp. 22–26.

[Ksohrabi00] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie, Protocols for self-organization of a wireless sensor network, *IEEE Personal Communications*, 7(5), 16–27, October 2000.

[Kyasanur03] P. Kyasanur and N.H. Vaidya, Detection and handling of MAC layer misbehavior in wireless networks, in *Proceedings of the International Conference on Dependable Systems and Networks (DSN '03)*, San Francisco, CA, 2003, pp. 173–182.

[LAShort98] L.A. Short and E.H. Saindon, Telehomecare rewards and risks, *Caring*, 17(42), 36–40, 1998.

[Laura07] L.J. Celentano, RFID-assisted wireless sensor networks for cardiac tele-healthcare, MS thesis, Advisor: Dr. F. Hu, Department of Computer Engineering, Rochester Institute of Technology, New York, October 2007.

[Lcampelli07] L. Campelli, A. Capone, M. Cesana, and E. Ekici, A receiver oriented MAC protocol for wireless sensor networks, in *Proceedings of Mobile Ad Hoc and Sensor Systems 07*, Pisa, Italy, October 2007, pp. 1–10.

[LEWIS86] F.L. Lewis, *Optimal Estimation: With an Introduction to Stochastic Control Theory*, John Wiley & Sons, Inc., New York, 1986.

[Legg] G. Legg, ZigBee: Wireless technology for low-power sensor networks, TechOnline, May 2004.

[Levis06] P. Levis et al., TinyOS: An operating system for sensor networks, in *Ambient Intelligence*, W. Weber, J. Rabaey, and E. Aarts, Eds., Springer-Verlag, Berlin, Germany, 2004.

[LHu04] L. Hu and D. Evans, Using directional antennas to prevent wormhole attacks, in *Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2004.

[LHu04a] L. Hu and D. Evans, Localization for mobile sensor networks, in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom)*, Philadelphia, PA, 2004, pp. 45–57.

[Linear04] Linear Technology. LTC1540: Nanopower comparator with reference. Datasheet, 7 December 2004. http://www.linear.com/pc/downloadDocument.do?navId=H0,C1,C1154,C1004,C1139,P1593,D1777

[LLazos04] L. Lazos and R. Poovendran, SeRLoc: Secure range-independent localization for wireless sensor networks, in *ACM WiSe*, Philadelphia, PA, 2004, pp. 21–30.

[LLi01] L. Li and J.Y. Halpern, Minimum-energy mobile wireless networks revisited, in *Proceedings of IEEE International Conference on Communications*, Helsinki, Finland, June 2001, Vol. 1, pp. 278–283.

[Lsubramanian00] L. Subramanian and R.H. Katz, An architecture for building self-configurable systems, in *Proceedings of MobiHoc 2000*, Boston, MA, November 2000, pp. 63–73.

[Macwilliams77] F. Macwilliams and N. Sloane, *The Theory of Error-Correcting Codes*, Elsevier Science, New York, 1977.

[MADDEN02a] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, TAG: A Tiny Aggregation service for ad-hoc sensor networks, in *Proceedings of OSDI*, Boston, MA, 2002a.

[Manish06] M. Raghuvanshi, Implementation of wireless sensor mote, MTech thesis, Department of Nuclear Engineering and Technology, Indian Institute of Technology, Kanpur, India, 2006, see http://home.iitk.ac.in/~ynsingh/mtech/manish2006.pdf

[Manjeshwar02] A. Manjeshwar and D.P. Agarwal, APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, in *Proceedings of 15th IEEE Parallel and Distributed Processing Symposium*, Fort Lauderdale, FL, April 2002, pp. 195–202.

[Mark07] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, MiniSec: A secure sensor network communication architecture, in *Proceedings of the Sixth International Conference on Information Processing in Sensor Networks (IPSN 2007)*, Cambridge, MA, April 2007.

[Martin00] T. Martin, E. Jovanov, and D. Raskovic, Issues in wearable computing for medical monitoring applications: A case study of a wearable ECG monitoring device, in *Proceedings of the International Symposium on Wearable Computers (ISWC)*, Atlanta, GA, 2000, pp. 43–50.

[Masoomeh07] M. Rudafshani and S. Datta, Localization in wireless sensor networks, in *IPSN'07*, Cambridge, MA, April 25–27, 2007.

[Mateusz07] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J.A. Paradiso, CargoNet: A low-cost MicroPower sensor node exploiting quasi-passive wakeup for adaptive asychronous monitoring of exceptional events, in *SenSys '07*, Sydney, Australia, November 6–9, 2007.

[MELEXIS02] MELEXIS, INC. 2002. MLX90601 infrared thermopile module, Technical Report (August). Go online to http://www.melexis.com/prodfiles/mlx90601.pdf

[Melodia05] T. Melodia, D. Pompili, and I.F. Akyildiz, On the interdependence of distributed topology control and geographical routing in ad hoc and sensor networks, *Journal of Selected Areas in Communications*, 23, 520–532, March 2005.

[Melodia07] T. Melodia, D. Pompili, V.C. Gungor, and I.F. Akyildiz, Communication and coordination in wireless sensor and actor networks, *IEEE Transactions on Mobile Computing*, 6(10), 1116–1129, October 2007. On Melodia's underwater sensor network papers: D. Pompili, T. Melodia, and I. Akyildiz, Three-dimensional and two-dimensional deployment analysis of underwater acoustic sensor networks, *Ad Hoc Networks (Elsevier)*, 7(4), 778–790, June 2009; I.F. Akyildiz, D. Pompili, and T. Melodia, State of the art in protocol research for underwater acoustic sensor networks, *ACM Mobile Computing and Communication Review* (Invited Paper), October 2007.

[MGHunink97] M.G. Hunink et al., The recent decline in mortality from coronary heart disease, 1980–1990. The effect of secular trends in risk factors and treatment, *Journal of the American Medical Association*, 277, 535–542, 1997.

[Miaoniao08] M. Wang, J. Cao, J. Li, and S.K. Das, Middleware for wireless sensor networks: A survey, *Journal of Computer Science and Technology*, 23(3), 305–326, 2008.

[Min07] M.K. Park and V. Rodoplu, UWAN-MAC: An energy-efficient MAC protocol for underwater acoustic wireless sensor networks, *IEEE Journal of Oceanic Engineering*, 32(3), 710–720, July 2007.

[MMaroti04] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, The flooding time synchronization protocol, in *Proceedings of the Second International ACM Conference on*

Embedded Networked Sensor Systems (SenSys), Baltimore, MD, 2004, ACM Press, New York, pp. 39–49.

[Mohamed02] M.G. Gouda, E.N. Elnozahy, C.-T. Huang, and T.M. McGuire, Hop integrity in computer networks, *IEEE/ACM Transactions on Networking*, 10(3), 308–319, June 2002.

[Moore04] D. Moore, J. Leonard, D. Rus, and S. Teller, Robust distributed network localization with noisy range measurements, in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, November 03–05, 2004.

[Newsome04] J. Newsome, E. Shi, D. Song, and A. Perrig, The sybil attack in sensor networks: Analysis & defenses, in *Proceedings of the Third international Symposium on Information Processing in Sensor Networks (IPSN '04)*, Berkeley, CA, April 26–27, 2004, ACM, New York, pp. 259–268.

[Ngajaweera08] N. Gajaweera and D. Dias, FAMA/TDMA hybrid MAC for wireless sensor networks, in *Proceedings of Fourth IEEE International Conference on Information and Automation for Sustainability '08*, Colombo, Sri Lanka, December 2008, pp. 67–72.

[Njamal04] N.A. Jamal and A.E. Kamal, Routing techniques in wireless sensor networks: A survey, *IEEE Wireless Communications*, 11(6), 6–28, December 2004.

[NPriyantha05] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, Mobile-assisted topology generation for auto-localization in sensor networks, in *Proceedings of Infocom*, Miami, FL, 2005.

[OYounis04] O. Younis and S. Fahmy, Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach, in *Proceedings of the IEEE INFOCOM*, Hong Kong, China, March 2004.

[PBonnet01] P. Bonnet, J.E. Gehrke, and P. Seshadri, Towards sensor database systems, in *Proceedings of the Second International Conference on Mobile Data Management (MDM '01)*, Hong Kong, China, January 2001, pp. 314–810.

[PDutta06] P. Dutta et al., Trio: Enabling sustainable and scalable outdoor wireless sensor network deployments, *IEEE SPOTS*, Nashville, TN, 2006.

[Peter05a] P. Desnoyers, D. Ganesan, and P. Shenoy, TSAR: A two tier sensor storage architecture using interval skip graphs, in *SenSys '05*, San Diego, CA, November 2–4, 2005.

[PFG06] H. Park, J. Friedman, P. Gutierrez, V. Samanta, J. Burke, and M.B. Srivastava, Illuminore: Multi-modal and high fidelity light sensor module for wireless sensor networks, *IEEE Sensors Journal*, 7(7), 996–1003, 2007.

[Philip03] P. Levis, N. Lee, M. Welsh, and D. Culler, TOSSIM: Accurate and scalable simulation of entire TinyOS applications, in *SenSys '03*, Los Angeles, CA, November 5–7, 2003.

[Pkarn90] P. Karn, MACA—A new channel access method for packet radio, in *ARRL/CRRL Amateur Radio Ninth Computer Networking Conference*, Montreal, Quebec, Canada, September 1990, pp. 1–5.

[PLevis02] P. Levis and D. Culler, Maté: A tiny virtual machine for sensor networks, in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, San Jose, CA, 2002, ACM Press, New York, pp. 85–95.

[Plin04] P. Lin, C. Qiao, and X. Wang, Medium access control with a dynamic duty cycle for sensor networks, in *Proceedings of Wireless Communications and Networking Conference*, Piscataway, NJ, March 2004, Vol. 3, pp. 1534–1539.

[Pompili06] D. Pompili, T. Melodia, and I.F. Akyildiz, Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks, in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom '06)*, Los Angeles, CA, September 23–29, 2006, ACM, New York, pp. 298–309.

[Pompili09] D. Pompili, T. Melodia, and I.F. Akyildiz, Three-dimensional and two-dimensional

deployment analysis for underwater acoustic sensor networks, *Ad Hoc Networks*, 7(4), 778–790, June 2009.

[Priyantha00] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, The cricket location-support system, in *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking (MobiCom '00)*, Boston, MA, August 2000, pp. 32–43.

[Priyantha05] N. B. Priyantha, The cricket indoor location system, PhD thesis, Computer Science and Engineering, Massachusetts Institute of Technology, MA, June 2005. Available at: http://nms.lcs.mit.edu/papers/bodhi-thesis.pdf

[PSikka06] P. Sikka, P. Corke, P. Valencia, C. Crossman, D. Swain, and G. Bishop-Hurley, Wireless ad hoc sensor and actuator networks on the farm, *IEEE SPOTS*, Nashville, Tennessee, 2006.

[Pubudu05] P.N. Pathirana, N. Bulusu, A.V. Savkin, and S. Jha, Node localization using mobile robots in delay-tolerant sensor networks, *IEEE Transactions on Mobile Computing*, 4(3), 285–296, May/June 2005.

[Purushottam07] P. Kulkarni, SensEye: A multi-tier heterogeneous camera sensor network, PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, MA, February 2007.

[PXie05] P. Xie, J.-H. Cui, and L. Li, VBF: Vector-based forwarding protocol for underwater sensor networks, UCONN CSE Technical Report, UbiNet-TR05-03 (BECAT/CSETR-05-6), February 2005.

[PZhang04] P. Zhang, C.M. Sadler, S.A. Lyon, and M. Martonosi, Hardware design experiences in zebranet, in *ACM Sensys*, Baltimore, MD, 2004.

[Qfang03] Q. Fang, F. Zhao, and L. Guibas, Lightweight sensing and communication protocols for target enumeration and aggregation, in *Proceedings of MobiHoc 2003*, Annapolis, MD, June 2003, pp. 165–176.

[Qli01] Q.Li, J. Aslam, and D. Rus, Hierarchical power-aware routing in sensor networks, in *Proceedings of the DIMACS Workshop on Pervasive Networking*, Piscataway, NJ, April 2001, pp. 1–5.

[Radu05] R. Stoleru, T. He, J.A. Stankovic, and D. Luebke, A high-accuracy, low-cost localization system for wireless sensor networks, in *SenSys '05*, San Diego, CA, November 2–4, 2005.

[Rahimi05] M. Rahimi et al., Cyclops: In situ image sensing and interpretation in wireless sensor networks, in *Proceedings of the Third International Conference on Embedded Networked Sensor Systems (SenSys '05)*, San Diego, CA, November 2–4, 2005, ACM, New York, pp. 192–204.

[Rahimi03] M. Rahimi, H. Shah, G. Sukhatme, J. Heidemann, and D. Estrin, Studying the feasibility of energy harvesting in a mobile sensor network, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipai, Taiwan, May 2003, pp. 19–24.

[Rakhmatov03] D. Rakhmatov and S. Vrudhula, Energy management for battery-powered embedded systems, *ACM Transactions Embedded Computing Systems*, 2(3), 277–324, August 2003.

[Rappaport96] T.S. Rappaport, *Wireless Communication: Principles and Practices*, Prentice-Hall PTR, Upper Saddle River, NJ, 1996.

[RATNASAMY01] S. Ratnasamy et al., Data-centric storage in sensornets, in *ACM First Workshop on Hot Topics in Networks*, Princeton, NJ, 2001.

[RATNASAMY02] S. Ratnasamy et al., GHT—A geographic hash-table for data-centric storage, in *First ACM International Workshop on Wireless Sensor Networks and Their Applications*, Atlanta, GA, September 2002.

[Reshah02] R.C. Shah and J.M. Rabaey, Energy aware routing for low energy ad hoc sensor networks, in *Proceedings of the IEEE WCNC '02*, Orlando, FL, March 2002, Vol. 1, pp. 350–355.

[Rkannan03] R. Kannan, K. Ram, S.S. Iyengar, and V. Kumar, Energy and rate based MAC protocol for wireless sensor network, in *Proceedings of the ACM SIGMOD 2003*, 32(4), 60–65, December 2003.

[Rramanathan97] S. Ramanathan, A unified framework and algorithms for (T/F/C) DMA channel assignment in wireless networks, in *Proceedings of IEEE INFOCOM*, San Francisco, CA, April 1997, Vol. 2, pp. 900–907.

[RShah03] R. Shah, S. Roy, S. Jain, and W. Burnette, Datamules: Modeling a three-tier architecture for sparse sensor networks, *Journal of Ad Hoc Networks (Elsevier)*, 1(2–3), 215–233, 2003.

[RSivakumar99] R. Sivakumar, P. Sinha, and V. Bharghavan, CEDAR: A core-extraction distributed ad hoc routing algorithm, *IEEE Journal on Selected Areas in Communications*, Special issue on *Ad Hoc Networks*, 17(8), 1454–1465, August 1999.

[RSzewczyk04] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler, An analysis of a large scale habitat monitoring application, in *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Baltimore, MD, November 2004.

[Rwhcinzelman99] R.W. Heidemann, K. Joanna, and H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, *ACM Mobicom '99*, Seattle, WA, August 1999, pp. 174–185.

[SAM06] T. Kuhn and P. Becker, A simulator interconnection framework for the accurate performance simulation of SDL models, in *System Analysis and Modeling: Language Profiles*, Lecture Notes in Computer Science, Vol. 4320, Springer, Berlin, Germany, 2006, ISBN 3-540-68371-2.

[Samuel02] R.S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, TAG: A Tiny AGgregation service for ad-hoc sensor networks, in *Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA, 2002.

[SCapkun03] S. Capkun, L. Buttyán, and J.-P. Hubaux, SECTOR: Secure tracking of node encounters in multi-hop wireless networks, in *Proceedings of the First ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, Fairfax, VA, 2003, pp. 21–32.

[Sek] Sekonic, Sekonic L-558Cine DualMaster. http://www.sekonic.com/Products/L-558Cine.html. 2008.

[SENSIRION02] SENSIRION, 2002. SHT11/15 relative humidity sensor. Tech. rep. (June). Go online to http://www.sensirion.com/en/pdf/Datasheet_SHT1x_SHT7x_0206.pdf

[SensorSim] S. Park, A. Savvides, and M.B. Srivastava, SensorSim: A simulation framework for sensor networks, in *Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Boston, MA, August 20, 2000. MSWIM '00, ACM, New York, pp. 104–111.

[Serh00] S. Edward-Austin Hollar, COTS Dust, MS thesis, Mechanical Engineering, University of California at Berkeley, Berkeley, CA, Fall 2000.

[Seung-Jong08] S.-J. Park, R. Vedantham, R. Sivakumar, and I.F. Akyildiz, GARUDA: Achieving effective reliability for downstream communication in wireless sensor networks, *IEEE Transactions on Mobile Computing*, 7(2), 214–230, February 2008.

[SFloyd93] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Transactions on Networking*, 1(4), 397–413, August 1993.

[SGaneriwal03] S. Ganeriwal, R. Kumar, and M.B. Srivastava, Timing-sync protocol for sensor networks, in *Proceedings of the First International ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, CA, 2003, ACM Press, New York, pp. 138–149.

[Shanmugasundaram04] J. Shanmugasundaram, Querying peer-to-peer networks using P-trees, Technical Report TR2004-1926, Cornell University, Ithaca, NY, 2004.

[SLi03] S. Li, S. Son, and J. Stankovic, Event detection services using data service middle-

ware in distributed sensor networks, in *Proceedings of the Second International Workshop Information Processing in Sensor Networks (IPSN '03)*, Palo Alto, CA, April 22–23, 2003, pp. 502–517.

[SLindsay02] S. Lindsay and C.S. Raghavendra, PEGASIS: Power-efficient gathering in sensor information systems, in *Proceedings of Aerospace Conference*, Big Sky, Mont, June 2002, Vol. 3, pp. 1125–1130.

[SLindsey02] S. Lindsey and C.S. Raghavendra, PEGASIS: Power efficient gathering in sensor information systems, in *Proceedings of the 2002 IEEE Aerospace Conference*, Big Sky, Mont, March 2002, pp. 1–6.

[Sony08]Sony SNC-RZ30N Camera driver. http://cvs.nesl.ucla.edu/cvs/viewcvs.cgi/CoordinatedActuation/Actuate/

[Sorber05] J. Sorber, N. Banerjee, M.D. Corner, and S. Rollins, Turducken: Hierarchical power management for mobile devices, in *Proceedings of MOBISYS*, Seattle, WA, 2005, pp. 261–274.

[SPalChaudhuri04] S. PalChaudhuri, A. Saha, and D.B. Johnson, Probabilistic clock synchronization service in sensor networks, Technical Report TR 03-418, Department of Computer Science, Rice University, Houston, TX, 2003.

[Sparton08] Sparton SP3003D Digital Compass. 2008. http://www.sparton.com/

[SRM05] S.R. Madden, M.J. Franklin, J.M. Hellerstein and W. Hong, TinyDB: An acquisitioned query processing system for sensor networks, *ACM Transactions Database Systems*, 30(1), 122–173, 2005.

[Spo] Spotlight. Website: http://www.spotlight.it

[SRoundy03] S. Roundy, B.P. Otis, Y.-H. Chee, J.M. Rabaey, and P. Wright, A 1.9ghz rf transmit beacon using environmentally scavenged energy, in *IEEE International Symposium on Low Power Electronics and Devices*, Seoul, Korea, 2003.

[SSL] OpenSSL. http://www.openssl.org

[Stargate08] Stargate platform. http://www.xbow.com/Products/XScale.htm. 2008.

[Stemm97] M. Stemm and R.H. Katz, Measuring and reducing energy consumption of network interfaces in hand-held devices, *IEICE Transactions on Communications*, E80-B(8), 1125–1131, August 1997.

[Stockdon00] H. Stockdon and R. Aredholman, Estimation of wave phase speed and nearshore bathymetry from video imagery, *Journal of Geophysical Research*, 105(9), 22015–22033, September 2000.

[Sundararaman05] B. Sundararaman, U. Buy, and A. Kshemkalyani, Clock synchronization for wireless sensor networks: A survey, *Ad Hoc Networks (Elsevier)*, 3, 281–323, May 2005.

[Sunil08] Z. Feng, S. Kumar, F. Hu, and Y. Xiao, E^2SRT: Enhanced event-to-sink reliable transport for wireless sensor networks, *Wireless Communications and Mobile Computing (Wiley)*, November 2008 (accessible online). DOI: 10.1002/wcm.705.

[Sunil08a] S. Kumar, K.K.R. Kambhatla, B. Zan, F. Hu, and Y. Xiao, An energy aware and intelligent cluster-based event detection scheme in wireless sensor networks, *International Journal of Sensor Networks (InderScience)*, 3(2) 123–135, February 2008.

[Schoo07] S. Zhou, R. Liu, D. Everitt, and J. Zic, A^2-MAC: An application adaptive medium access control protocol for data collections in wireless sensor networks, in *Proceedings of IEEE ISCIT07*, Sydney, Australia, October 2007, pp. 1131–1136.

[Tanya06] T. Roosta, S.P. Shieh, and S. Sastry, Taxonomy of security attacks in sensor networks and countermeasures, in *First IEEE International Conference on System Integration and Reliability Improvements*, Hanoi, Vietnam, December 2006.

[TAOS] TAOS, INC. 2002. TSL2550 ambient light sensor, Technical Report (September). Go online to http://www.taosinc.com/images/product/document/tsl2550.pdf

[TCamp02] T. Camp, J. Boleng, and V. Davies, A survey of mobility models for ad hoc network research, *Wireless Communications and Mobile Computing*, 2(5), 483–502, 2002.

[TI08] One of the largest chip production company—Texas Instruments, see http://www.ti.com. Visited in June 2008.

[Tian04] T. He, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, AIDA: Adaptive application-independent data aggregation in wireless sensor networks, *Transactions on Embedded Computing System*, 3(2), 426–457, May 2004.

[TinyOS07] On TinyOS operating system, see http://www.tinyos.net, Visited in June 2007.

[Tmote06] Tmote invent user's manual, Technical Report, Moteiv, Inc., San Francisco, CA, February 2006.

[Transducer08] International Transducer Corporation. Available: www.itc-transducer.com. Visited in 2008.

[Tsai87] R.Y. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, *IEEE Journal of Robotics and Automation*, RA 3(4), 323–344, August 1987.

[The03] H. Tian, J.A. Stankovic, C. Lu, and T. Abdelzaher, SPEED: A stateless protocol for real-time communication in sensor networks, in *Proceedings of Distributed Computing Systems 2003*, Providence, RI, May 2003, pp. 46–55.

[Von92] T. von Eicken, D.E. Culler, S.C. Goldstein, and K.E. Schauser, Active messages: A mechanism for integrating communication and computation, in *Proceedings of the 19th Annual International Symposium on Computer Architecture*, Gold Coast, Australia, May 1992, pp. 256–266.

[Tvdam03] T. Van Dam and K. Langendoen, An adaptive energy-efficient MAC protocol for wireless sensor networks, in *Proceedings of First International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, November 2003, ACM, New York, pp. 171–180.

[TYlonen96] T. Ylonen, SSH – Secure login connections over the internet, in *Proceedings of the Sixth USENIX Security Symposium*, San Jose, CA, 1996.

[VJacobson88] V. Jacobson, Congestion avoidance and control, in *Proceedings of the ACM SIGCOMM Symposium*, Stanford, CA, August 1988.

[Victor04] V. Shnayder, M. Hempstead, B. Chen, G.W. Allen, and M. Welsh, Simulating the power consumption of large scale sensor network applications, in *SenSys '04*, Baltimore, MD, November 3–5, 2004.

[Vinantha04] V. Ekanayake, C. Kelly IV, and R. Manohar, An ultra low-power processor for sensor networks, *ASPLOS '04*, Boston, MA, October 7–13, 2004.

[Vrajendran05] V. Rajendran, J.J. Garcia-Luna-Aceves, and K. Obraczka, Energy-efficient, application-aware medium access for sensor networks, in *Proceedings of IEEE Mobile Adhoc and Sensor Systems '05*, Washington, DC, November 2005, pp. 630–637.

[Vrajendran06] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves, Energy-efficient, collision-free medium access control for wireless sensor networks, in *Proceedings of the First International Conference on Embedded Sensor Systems (SenSys '03)*, Los Angeles, CA, February 2006, ACM, New York, Vol. 12, No. 1, pp. 63–78.

[VRaghunathan05] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, Design considerations for solar energy harvesting wireless embedded systems, *IEEE SPOTS*, Los Angeles, CA, 2005.

[Vrodoplu99] V. Rodoplu and T.H. Meng, Minimum energy mobile wireless networks, *IEEE Journal on Selected Areas in Communications*, 17(8), 1333–1344, August 1999.

[Wan03] C. Wan, S.B. Eisenman, and A.T. Campbell, CODA: Congestion detection and avoidance in sensor networks, in *Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys '03)*, Los Angeles, CA, November 5–7, 2003, ACM, New York, pp. 266–279.

[Wang06] S. Wang, W. Chen, C. Ong, L. Liu, and Y. Chuang, RFID application in hospitals: A case study on a demonstration RFID project in a Taiwan hospital, in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06)*, Kauai, HI, January 4–7, 2006, Vol. 8, p. 184a.

[Wang08] M.M. Wang, J.N. Cao, J. Li, and S. Das, Middleware for wireless sensor networks:

A survey, *Journal of Computer Science and Technology*, 23(3), 305–326, May 2008.

[Ward97] A. Ward, A. Jones, and A. Hopper, A new location technique for the active office, *IEEE Personal Communications*, 4(5), 42–47, October 1997.

[WBHeinzelman02] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, in *Proceedings of the IEEE Transactions on Wireless Communications*, 1(4), 660–670, October 2002.

[WBHeinzelman04] W.B. Heinzelman et al., Middleware to support sensor network applications, *IEEE Network*, 18(1), 6–14, 2004.

[WINS] Wireless integrated network systems(wins). http://wins.rsc.rockwell.com/, 2008.

[WMB02] E. Wagmister, B. McDonald, J. Brush, J. Burke, and T. Denove, Advanced Technology for Cinematography, 2002. Website: http://hypermedia.ucla.edu/projects/atc.php

[WS82] W. Gunter and W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, 2nd edn. John Wiley & Sons, New York, 1982.

[Wstallings04] W. Stallings, IEEE 802.11 Wireless LANs: From a to n, *IT Proceedings*, 6, 32–37, September–October 2004.

[Wye02] W. Ye, J. Heidemann, and D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in *Proceedings of IEEE INFOCOM*, New York, June 2002, Vol. 3, pp. 1567–1576.

[Wye04] W. Ye, J. Heidemann, and D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, *IEEE/ACM Transactions on Networking*, 12(3), 453–506, July 2004.

[WSu05] W. Su and I.F. Akyildiz, Time-diffusion synchronization protocol for sensor networks, *IEEE/ACM Transactions on Networking*, 13(2), 384–398, 2005.

[XHong99] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, A group mobility model for ad hoc wireless networks, in *MSWiM '99*, Seattle, WA, 1999, ACM Press, New York, pp. 53–60.

[Xiang04] X. Ji, Localization algorithms for wireless sensor network systems, PhD thesis, Department of Computer Science and Engineering, The Pennsylvania State University, Philadelphia, PA, 2004.

[XJiang05] X. Jiang, J. Polastre, and D. Culler, Perpetual environmentally powered sensor networks, *IEEE SPOTS*, Los Angeles, California, April 2005.

[Yyan01] Y. Yan, R. Govindan, and D. Estrin, Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks, Technical Report UCLA-CSD TR-010023, August, 2001.

[YangXiao07] Y. Xiao, V.K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, A survey of key management schemes in wireless sensor networks, *Computer Communications Journal (Elsevier)*, Special issue on *Security on Wireless Ad Hoc and Sensor Networks*, 30(11–12), 2314–2341, September 2007.

[YCHu03] Y.-C. Hu, A. Perrig, and D.B. Johnson, Packet leashes: A defense against wormhole attacks in wireless networks, in *IEEE INFOCOM*, San Francisco, CA, 2003.

[Yiyer05] Y. Iyer, S. Gandham, and S. Venkatesan, STCP: A generic transport layer protocol for sensor networks, in *Proceedings of 14th IEEE International Conference on Computer Communications and Networks*, San Diego, CA, October 2005.

[YTirta06] Y. Tirta, B. Lau, N. Malhotra, S. Bagchi, Z. Li, and Y.-H. Lu, Controlled mobility for efficient data gathering in sensor networks with passively mobile nodes, in *Sensor Network Operations*, Wiley-IEEE Press, Hoboken, NJ, 2006.

[Yuan06] H. Yuan, H. Ma, and H. Liao, Coordination mechanism in wireless sensor and actor networks, in *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS '06)*, April 20–24, 2006, Vol. 2, Zhejiang, China, pp. 627–634.

[Yxu01] Y. Xu, J. Heidemann, and D. Estrin, Geography informed energy conservation for ad hoc routing, in *Proceeding of MobiCom 2001*, Rome, Italy, July 2001, pp. 70–84.

[YXu01] Y. Xu, J. Heidemann, and D. Estrin, Geography informed energy conservation for ad hoc routing, in *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.

[Zhang08] P. Zhang and M. Martonosi, LOCALE: Collaborative localization estimation for sparse mobile sensor networks, in *Proceedings of the 2008 International Conference on Information Processing in Sensor Networks (IPSN '08)*, St. Louis, MO, April 22–24, 2008, pp. 195–206.

[Zigbee08] On Zigbee wireless communication standard please see http://www.zigbee.org. Visited in June 2007.

[ZLi05] Z. Li, W. Trappe, Y. Zhang, and B. Nath, Robust statistical methods for securing wireless localization in sensor networks, in *Proceedings of IPSN '05*, Los Angeles, CA, 2005.

索　引

索引中标注的页码为英文原书页码，与书中边栏的页码一致。

A

Activeattackers（主动攻击者），328
Actor-actor coordination（执行器-执行器协同），365，374，377-378
  action first（AF）scheme（执行优先（AF）方案），377
  decision-first（DF）scheme（决策优先（AD）方案），378
Advanced technology forcinematographs（ATC）（先进摄影技术（ATC）），468
AIDA（独立于应用的数据融合（AIDA）），249-253
  aggregation control unit（融合控制单元），251-252
  aggregator（融合节点），250-251
  dynamic feedback scheme（动态反馈方案），253
  fixed scheme（确定性方案），253
  function unit（功能单元），251
  grid based data aggregation（基于网格的数据融合），251
  on-demand scheme（按需方案），253
Analog-to-digital converter（ADC）（模拟数字转换器（ADC）），394-395，434-435，439，448
Authentication（认证），329，335
μTESLA（基于时间的高效的容忍丢包的流认证协议（μTESLA）），347
  asymmetric digital signature（非对称数字签名），348
  bootstrap（安全引导），351
  commitment（验证凭证），351
  freshness（新鲜性），352
  key chain（密钥链），349
  message authentication code（MAC）（消息认证码（MAC）），348
  point-to-point authentication（逐跳认证），352
  round trip time（RTT）（往返时间（RTT）），351
  symmetric key（对称密钥），348
Automatic Repeatrequest（ARQ）（自动重传请求（ARQ）），397

B

Battery aware routing（BAR）（能量感知路由（BAR）），425-428
  lithium-ion（锂离子电池），425，427
  nickel-cadmium（镍-镉电池），425，427
B-MAC（Berkeley MAC）（B-MAC（Berkeley MAC）协议），100-101
  clear channel assessment（CCA）（空闲信道评估（CCA）），100，105
  low power listening（低功耗侦听），100
Blom based scheme（基于 Blom 模型的密钥预分配方案），344
  key generation（密钥生成），346
  multiple key space（多密钥空间），345
  probabilistic key predistribution scheme（随机密钥预分配方案），345
  single key space（单密钥空间），345

C

CargoNet（CargoNet 节点），57-62
  acceleration stimuli（加速激励），58
  high frequency clock oscillator（高频时钟振荡器），59
  quasi passive wakeup（准被动唤醒），57
  RFID（射频识别（RFID）），58
    active RFID（主动 RFID），58
  stimuli signa（激励信号），57
  Texas Instruments（TI）（德州仪器（TI）），58
  vibration detection and autonomous crack monitoring（振动监测与裂纹探测），57
Client puzzle（客户端谜题），335
Clock offset（时钟偏移），313
Clock synchronization（时钟同步），13，308-311
  network time protocol（NTP）（网络时间协议（NTP）），13，308，311，314
Code division multiple access（CDMA）（码分多路复用（CDMA）），39，77
Color temperature（色温），470
  correlated color temperature（CCT）（关联色温（CCT）），470
Congestion avoidance and detection（CODA）（拥塞检测与

避免（CODA），178-184
closed-loop, multisource regulation（多源闭环调整），181
sink regulation（汇聚节点调节），181
congestion detection（拥塞检测），167，170-171，181，183
carrier sense（载波侦听），183
channel load（信道负载），183-184
congestion notification（CN）bit（拥塞通知（CN）位），186，188-189
continuous flow（连续数据流），187-188
event based flow（事件触发数据流），187
round trip time（RTT）（环回时间（RTT）），187
localized congestion control（本地化拥塞控制），180
depth of congestion（拥塞深度），182
epoch time（周期时间），183
estimated trip time（ETT）（估计传输时间（ETT）），185
open-loop backpressure mechanism（开环反向压力机制），184
open-loop, hop-to-hop backpressure（开环逐跳段反向压力），181
backpressure signal（反向压力信号），181
random early detection（RED）（随机早期检测（RED）），188
reliability field（可靠性字段），186
sensor transmission control protocol（STCP）（传感器网络传输控制协议（STCP）），185
session initiation packet（会话初始数据包），185
sparsely deployed sensors（稀疏布署传感器网络），180
closed-loop rate regulation（闭环数据速率调节），180
high data-rate events（高数据速率事件），180
localized back pressure（本地化反向压力），180
packet-dropping technique（丢包技术），180
TCP three way handshake（TCP 三次握手），185
association（关联），185
threshold higher（上界限），188
threshold lower（下界限），188
Congestion, high reliability（C, HR）（拥塞，高可靠性（C, HR）），168-169
Congestion, low reliability（C, LR）（拥塞，低可靠性（C, LR）），168，170
Coronary artery disease（CAD）（冠状动脉疾病（CAD）），445
COTS dust system（COTS 微尘系统），52-54
base mote（基站节点），52
floating mote（普通节点），52
Cyclic redundancy check（CRC）（循环冗余校验（CRC）），52，306

D

Delay attack（延误攻击），354-357
collusion based（合谋方式），355
directional antenna delay attack（定向天线延误攻击），355
RBS scheme（参考广播同步（RBS）机制），355
receiver-receiver based synchronization（基于接收者-接收者机制的时间同步），354
sender-receiver based synchronization（基于发送者-接收者机制的时间同步），354
Delay tolerant sensor network（DTN）（容迟传感器网络（DTN）），368-374
Denial of service（DoS）（拒绝服务攻击（DoS）），335，352
probabilistic geographic routing（PGR）（基于地理位置信息的随机路由（PGR）），336
Desynchronization attack（失同步攻击），335
Digital signal processor（DSP）（数字信号处理器（DSP）），393，395
Direct memory access（DMA）（直接存储器存取（DMA）），395
Distributed event-driven partitioning and routing protocol（DEPR）（基于事件驱动的分布式区域划分和路由协议（DEPR）），371-373

E

Electrocardiogram（EKG）（心电图（EKG）），11，446-448
Enhanced event-to-sink reliable transport（E^{2}SRT）（增强型事件汇聚可靠传输协议（E^{2}SRT）），171-179
actual reliability（实际可靠性），172
desired reliability（预期可靠性），171
maximum operating region（MOR）（最大工作区域（MOR）），175
maximum reliable point（MRP）（最高可靠性点（MRP）），175
over-demanding reliability（OR）（超规格可靠性（OR）），170
Event-to-sink reliable transport（ESRT）（事件到汇聚节点的可靠传输协议（ESRT）），163-170
congestion detection（拥塞检测），170-171
local buffer level monitoring scheme（本地缓冲区级别监测算法），170
event reliability indicator（事件可靠性指标），163
desired event reliability（预期事件可靠性），163
observed event reliability（观测事件可靠性），163
multiplicative decrease（乘性减少），169

normalized reliability（归一化可靠性），164
reporting frequency（报告频率），164
required event detection reliability（所需事件检测可靠性），164
Extreme studentized deviate（ESD）（极值学生化偏差（ESD）），357-358

F

Finite state machine（FSM）（有限状态机（FSM）），418-422
FSM based simulation model（基于有限状态机的仿真模型），418
Frequency division multiple access（FDMA）（频分多路复用（FDMA）），77-78

G

GARUDA（GARUDA），163，189-195
data collection command（数据采集命令），189
downstream point to multipoint data delivery（下行点到多点数据传输），189
dominating set（支配集），191
minimum dominating set（MDS）（最小支配集（MDS）），191
GARUDA framework（GARUDA 架构），193-195
core node（核心节点），194
core solicitation message（核心请求消息），194
loss detection（丢包检测），195
loss recovery（丢包恢复），195
loss recovery mechanism（丢包恢复机制），190
loss recovery process（丢包恢复过程），192-193
out-of-sequence packet forwarding（无序报文转发），192-193
two stage loss recovery（两阶段丢包恢复），193
over-the-airprogramming code（无线编程代码），189
redundancy aware delivery（冗余性感知的数据发送），191
reliability semantics（可靠性语义），190-191
location dependency（位置相关性），190
location redundancy（位置冗余性），190
Generalized Extreme Studentized Deviate（GESD）（泛化极端学生化偏差（GESD）），357-258
GESD based delay attack detection（基于 GESD 延迟攻击检测），358
GloMoSim（GloMoSim），431-432
OSI layered network architecture（OSI 分层网络体系结构），432
parallel-discrete event simulation（并行离散事件仿真），431
Parsec（Parsec），431
WSN hardware testbed（无线传感器网络测试台），431
GPS free node localization（无需 GPS 的节点定位方法），285-289
core localization algorithm（核心定位算法），286
directional localization（方向性定位），286
reference point group mobility（参考点群组移动），289
range free localization（距离无关的定位方法），290
spotlight（Spotlight 装置），290

H

Hierarchical routing（分层路由），113-114，128-138
adaptivePeriodic Threshold- sensitive Energy Efficient Network protocol（APTEEN）（周期/阈值自适应的能量高效路由协议（APTEEN）），137-138
count time（CT）（计数时间（CT），137 节点发送监测数据的最大时间周期）
historical query（历史查询），138
one time query（一次查询），138
persistent query（持续查询），138
threshold（阈值），137
time division medium access（TDMA）（时分介质访问（TDMA）调度），137
Low Energy Adaptive Clustering Hierarchy protocol（LEACH）低功耗自适应类簇分层路由协议（LEACH）），129-134
cluster head node（CH）（簇首节点（CH）），130-131
LEACH-Centralized protocol（LEACH-C）（集中式 LEACH 协议（LEACH-C）），133
non-cluster head node（non-CH）（簇成员节点（non-CH）），130
setup phase（建立阶段），130
steady state phase（稳定阶段），130
threshold-sensitive Energy Efficient sensor Network protocol（TEEN）（阈值敏感的能量高效传感器网络路由协议（TEEN）），134-138
hard threshold（硬阈值），136
sensed value（SV）（感知值（SV）），136
soft threshold（软阈值），136
Hierarchical WSAN Coordination architecture（WSAN 分层协同工作体系结构），373-374
sensor-sensor coordination（传感器-传感器协同），374

I

IEEE 802.11（IEEE 802.11 标准），72-76

distributed coordination function (DCF) (分布式协同功能(DCF)), 74-76
point coordination function (PCF) (点协同功能(PCF)), 73
Illuminate's sensor (Illuminate 传感器), 468
color intensity sensor (色度传感器), 468
incident light angle sensor (入射角传感器), 468
incident light intensity sensor (入射光强传感器), 468
situational sensor (环境传感器), 468
Illuminator (Illuminator), 465-467, 472
DMX controller and dimmer (DMX 控制器和调节器), 471
illuminator core (Illuminator 控制核心子系统), 471
Insider attack (内部攻击), 327
Invasive attack (侵入式攻击), 328-329
Iterative multidimensional scaling (迭代多维标度), 275-280
distributed physical location estimation (分布式物理位置估计), 278-280
hop distance (跳距), 276
ranging estimation (测距估计), 276
ultrasound (超声波), 276

J

Jamming (干扰攻击), 330, 332

L

Laptop class attack (便携计算机级攻击), 327
Light acquisition unit (光获取单元), 469
Light characterization (光特征提取), 466-467
LITEWORP (LITEWORP), 339-344
alert buffer (告警缓冲区), 344
detection confidence (置信度), 344
scalable and energy efficient crypto on sensors (SECOS), (适用于传感器节点的能量高效可扩展密钥) (管理协议 (SECOS)), 340
sentry (guard) node (守护节点), 341
shared secret keys (共享密钥), 344
Localization in wireless sensor networks (无线传感器网络定位技术), 280-285
Monte Carlo localization (MCL) (蒙特卡洛定位算法 (MCL)), 285
Monte Carlo method (蒙特卡洛方法), 280-281
closeness (接近度), 284
distance vector algorithm (距离矢量算法), 283
initialization (初始化), 281
re-sampling (重新采样), 281
sampling (采样), 281
Location based routing protocol (基于位置信息的路由协议), 138-144
geographical and energy aware routing (GEAR) (地理位置和能量感知的路由协议 (GEAR)), 139-144
estimated cost (估计代价), 139
learned cost (实际代价), 139
recursive geographical forwarding (迭代地理转发), 139
restricted flooding (限制性洪泛), 139
Low density collaborative ad hoc localization estimate (LOCALE) (稀疏网络中的节点自组协同定位算法 (LOCALE)), 295-302
collaborative location estimate (协同位置估计), 296
estimated location (估计位置), 297
global coordinate system (全球坐标系统), 300
local phase (局部定位阶段), 296
transform phase (转换阶段), 296
true location (实际位置), 297
update phase (更新阶段), 296
dead reckoning (DR) (方位推算 (DR)), 295

M

Medical ad hoc sensor network (MASN) (医疗自组传感网 (MASN)), 446-450
dynamic reliability adaptation (动态可靠性调整), 453
electrocardiogram (ECG) simulator (心电图 (ECG) 模拟器), 450
hybrid energy efficient distributed clustering (HEED) (能量高效的分布式混合路由协议 (HEED)), 454
location-based medicare service (LBMS) (基于位置的医疗服务 (LBMS)), 457
logic architecture (逻辑结构), 447
mobile platform (移动平台), 447
normalized average minimum reachability power (nAMRP) (标准化平均最小可达能量 (nAMRP)), 452
printed circuit board (PCB) layout (印刷电路板 (PCB) 布局), 449
RF communication board (无线通信板), 450
RFID for patient tracking (病人跟踪 RFID), 459
database (数据库), 459
for medicine-taking guide (服药指导), 459
for road guide (路径指引), 459
system on chip (SoC) (片上系统 (SoC)), 449
MediumAccess Control (MAC) (介质访问控制 (MAC)), 67-109
collision at the receiver's end (接收端冲突), 70

contention based MAC protocol（基于竞争的 MAC 协议），77-88
determination of listen cycle（侦听周期确定），387-388
determination of transmit start time（发送起始时间确定），388
newcomer（节点加入），389
node failure（节点失效），389
variable acoustic delay（可变声波传输延迟），389
hidden and exposed terminal problem（隐藏/暴露终端问题），70-72
hybrid and event based MAC protocol（混合 MAC/事件驱动 MAC 协议），94-106
schedule based MAC protocol（基于调度的 MAC 协议），88-94
signal loss in the wireless channel（无线信道信号损耗），69-72
Mica mote design（Mica 节点设计），48-50
memory（存储器），48
mote ID（节点 ID），48
peripheral（外围模块），49
power supply（能量供给模块），49
radio（无线通信模块），49
Middleware layer（中间件层），226，235，433
Agilla（Agilla），228-231
agent based middleware（基于代理的中间件），228
code migration（代码迁移），231
neighbor list（邻居列表），230-231
tuple space（元组空间），230
data storage ware（DSWare）（数据存储中间件（DSWare）），233
mate（Mate），234
MiLAN（MiLAN），235-236
sensor QoS graph，（节点 QoS 图），235
state based variablerequirements graph（基于状态的可变需求图），235
Mires（Mires），231-233
data acquisition（数据获取），231
publish/subscribe service（发布/订阅服务），232
query based data model（基于查询的数据模型），233
TinyDB（TinyDB），233
programming abstraction（编程抽象），227
quality of service（QoS）（服务质量（QoS）），228
reusable code service（可复用代码服务），226
runtime support（运行时支持），228
system service（系统服务），227-228
MiniSec（MiniSec），253
bloom-filter-based replay protection（基于 Bloom 过滤器的重放保护），353
Mobile ad hoc networks（MANET）（移动自组织网络（MANET）），5-6，68，110
Mobile stage element（移动舞台元素），466
Mobile tag（移动标签），466
Mote（智能尘埃（节点）），5，27-62
Mote class attacker（节点级攻击者），327
Multi resolution data processing（多分辨率数据处理），256-257
high resolution data（高分辨率数据），256
low resolution data（低分辨率数据），256
spatial correlation（空间相关性），257
spatiotemporal correlation（时空相关性），257
temporal correlation（时间相关性），257
Multipath and QoS-based routing（多径 QoS 路由），145-148
multipath routing（多径路由），145-147
alternate path（备用路径），146
link disjoint（链路不相交），145
load balancing（负载均衡），145
node disjoint（节点不相交），145
primary path（主路径），145
QoS based routing（QoS 路由），147-148
Sequential assignment routing（SAR）（有序分配路由（SAR）），147

N

Neighboring node model（邻节点模型），356
No congestion，high reliability（NC，HR）（无拥塞，高可靠（NC，HR）），169
No congestion，low reliability（NC，LR）（无拥塞，低可靠（NC，LR）），168-169
Node localization（节点定位），12，261
range based localization（基于距离的定位方法），12
range free localization（距离无关的定位方法），12
Nonconstant execution time（非恒定执行时间），329
Noninvasive attack（非侵入式攻击），328

O

Offset delay estimation（偏差延迟估计），311-314
Optimal operating region（OOR）（最佳工作区域（OOR）），168
Outside attacker（外部攻击者），327-328

P

Passive attacker（被动攻击者），328
Path loss ratio（路径损耗比），6
PowerTOSSIM（PowerTOSSIM），437-441

power state transition message（功率状态转换消息），439
power state module（功率状态模块），440
power state tracking code（功率状态跟踪代码），440
TinyVia（TinyVia），437
postprocessor（后处理器），440
Probabilistic clock synchronization（概率时钟同步），324-325

R

Radio signal propagation（无线信号传播），6
Received signal strength（RSS）（接收信号强度（RSS）），6
Reference broadcast synchronization（RBS）（参考广播同步（RBS）），319-321
access time（访问时间），320
nondeterministic clock error（非确定性时钟误差），319
nondeterministic transmission delay（非确定性传输延迟），319
propagation time（传播时间），320
Remote clock reading（远程时钟读取），311
Resilient expandable and threaded operating system（RETOS）（弹性可扩展多线程操作系统（RETOS）），219-222
application code checking（应用代码检查），219-221
destination field（目的字段），219
dynamic code（动态代码），220
source field（源字段），219
static code（静态代码），220
loadable kernel module（可加载内核模块），222
multithreading system（多线程系统），221-222
single kernel stack（单一内核堆栈），221
stack-size analysis（堆栈大小分析），221
Round trip delay（往返延迟），313
Routing layer（路由层），109-125
data centric routing protocol（以数据为中心的路由协议），114-128
flooding（洪泛），115-117
gossiping（闲聊），115-117
ideal dissemination 理想分发，117
implosion（内爆），115
data reporting（数据报告），111
event driven（事件驱动），111
query driven（查询驱动），111
time driven（时间驱动），111
directed diffusion（定向扩散路由），122-128
data aggregation（数据聚合），125
data caching（数据缓存），125
data propagation（数据传播），125
gradient establishment（梯度建立），123-125
interest（兴趣），122
interest propagation（兴趣扩散），123-125
naming（命名），123
reinforcement（路径加强），126-127
hierarchical routing（分层路由），128-138
hybrid routing（混合路由），113
network dynamics and heterogeneity（网络动态性和异构性），112
node deployment（节点部署），111
manual deployment（人工部署），111
random deployment（随机部署），111
proactive routing（主动式路由），113
reactive routing（反应式路由），113
sensor protocols for information via negotiation（SPIN）（基于协议协商的传感器网络路由协议（SPIN）），117-122
data advertisement（ADV）（广播消息（ADV）），118
data message（数据消息），118
data request（REQ）（请求消息（REQ）），118
SPIN-BC（broadcast）（SPIN-BC（广播）），119-120
SPIN-EC（energy conservation）（SPIN-EC（节能）），119
SPIN-PP（point to point）（SPIN-PP（源端）），119
SPIN-RL（reliable）（SPIN-RL（可靠）），120
RTS/CTS（RTS/CTS 机制），75
request-to-send（RTS）（请求发送（RTS）），74
clear-to-send（CTS）（允许发送（CTS）），74

S

Security of WSN localization（无线传感器网络安全定位），303-305
adversary（敌方），303
attack-resistant location estimation（攻击容忍的节点定位），304-305
attack resilient minimum mean square estimation（攻击容忍的最小均方估计），304
voting based location estimation（基于投票的节点定位），304-305
beacon suite（信标套件），304
robust statistical model（稳健统计模型），305
RF based fingerprinting（基于射频的指纹识别），305
triangulation（三角测量），305
SeRLoc（SeRLoc），303-304
Selective forwarding（选择转发），331-332
black hole（黑洞），331
SenseEye（SenseEye），408-412
Agilent Cyclops（Agilent Cyclops），410

CMUcam Vision sensor（CMUcam 视觉传感器），410
flash memory（闪存），408
high resolution object detection and recognition（高分辨率目标检测与识别），412
high resolution pan-tilt-zoom camera（高分辨率变焦云台（PTZ）摄像头），409
high tier sensor（高层传感器），408
Logitech Quickcam Pro Webcam（Logitech Quickcam Pro 网络摄像头），410
low tier sensor（低层传感器），408
mote level detector（节点级检测器），411
PTZ controller（PTZ 控制器），412
RAM（RAM），408
Sony PTZ camera（Sony PTZ 摄像头），410
Tier-1 frame differentiator（第一层帧差器），411
wakeup mote（唤醒节点），412
Sensor data cleaning（传感数据清理），237-243
Bayesian based cleaning（BayC）（贝叶斯清理（BayC）），239
cleaning module（清理模块），239
database level（数据库级别），241
query processing module（查询处理模块），240
random noise（随机噪声），237
sensor level（节点级别），241
systematic error（系统误差），237
Sensor localization（节点定位），261-305
angle of arrival（AoA）（到达角（AoA）），266
global positioning system（GPS）（全球定位系统（GPS）），262
integral quadratic constraint（IQC）（积分二次约束（IQC）），272
line of sight（LOS）（视线（LOS）），262
multidimensional scaling（MDS）（多维标度（MDS）），274
classical multidimensional scaling（经典 MDS），274-275
eigen decomposition（特征分解），274
iterative multidimensional scaling（迭代 MDS），275-280
multilateration（多边定位），267-268
received signal strength indication（接收信号强度指示），262-264
robust extended Kalman filter（REKF）（抗差扩展卡尔曼滤波（REKF）），269
self localization（自身定位），262
time difference of arrival（TDoA），（到达时间差（TDoA）），264
time of arrival（ToA）（到达时间（ToA）），264
trilateration（三边测量），266
triangulation（三角测量），266
trilateration（三边测量），266-267
wireless integrated network sensor（WINS）（无线集成网络传感器）（WINS），263
S-MAC（Sensor MAC）（S-MAC（Sensor MAC）协议），78-83
adaptive listen（自适应侦听），83
idle listening（空闲侦听），78
network allocation vector（NAV）（网络分配向量）（NAV），81
overhearing（串音），78
periodic listen and sleep（周期性侦听/睡眠），79-80
sleep delay（睡眠延迟），82
Sensor node's dynamic behavior（传感器节点的动态行为），420
specification and description language（SDL）（规范和描述语言（SDL）），420
Sensor network asynchronous processor/low Energy（SNAP/LE）（传感器网络异步处理器/低功耗（SNAP/LE）），31-35
lower power sleep mode（超低功耗睡眠模式），33
boot code（启动代码），34
data driven switching activity（数据驱动的切换活动），34
deep sleep state（深度睡眠状态），34
event queue（事件队列），34
event Token（事件令牌），34
timercoprocessor（定时协处理器），34
messagecoprocessor（消息协处理器），34
instruction fetch（取指令），34
low power consumption（低功耗），33
low-overhead wake-up mechanism（低开销唤醒机制），33
Sensor-actor coordination（传感器-执行器协同），366-373
data aggregation tree（数据聚合树），367
event-driven partitioning（事件驱动划分），367
integer linear programming（ILP）（整数线性规划（ILP）），367
analog sensor（模拟传感器），28
carrier sense media access（CSMA）（载波侦听介质访问（CSMA）），45
chip（芯片），44
commercial off the shelf（COTS）（商业成品（COTS）），32
digital sensor（数字传感器），28
dynamic CPU speed（动态 CPU 处理速度），31
low duty cycle（低占空比），29
memory（存储器），35
on-chip storage（flash memory）（片内存储（闪存）），

35
static random access memory (SRAM) (静态随机存取存储器 (SRAM)), 36
microcontroller (微控制器), 30
peripheral support (外围支持), 43
analog I/O pin (模拟 I/O 引脚), 43
digital I/O pin (数字 I/O 引脚), 43
power source (电源), 41-43
radio (无线通信模块), 36
amplitude modulation (AM) (调幅 (AM)), 38
frequency hoping (FH) (跳频 (FH)), 38
frequency modulation (FM) (调频 (FM)), 38
modulation scheme (调制方法), 37
radio propagation distance (无线信号传播距离), 37
radio transceiver (无线收发器), 36
receiver sensitivity (接收灵敏度), 37
transmission strength (发射信号强度), 37
voltage controlled oscillator (VCO) (压控振荡器 (VCO)), 37
sensor mote architecture (传感器节点体系结构), 44
host channel interface (HCI) (主机信道接口 (HCI)), 46
symbol (符号), 44
voltage converter and regulator (调压器/稳压器), 30
voltage regulation (稳压), 42
SensorSim (SensorSim), 432-433
NS-2 simulator (NS-2 模拟器), 433
WSN sending/receiving energy consumption Model (传感器网络发送/接收能量消耗模型), 432
SiftMAC (SiftMAC 协议), 94-99
backoff probability distribution (退避概率分布), 95-98
correlated contention (关联性竞争), 94
Sinkhole attack (Sinkhole 攻击), 332-333
Smart antenna (智能天线), 266
Source authentication (源认证), 335
Space Division Multiple Access (SDMA) (空分多路复用 (SDMA)), 77
Spec (Spec 节点), 50-51
Spoofed routing information (欺骗路由信息), 331
altered routing information (篡改路由信息), 331
replayed routing information (重放路由信息), 331
Spotlight localization (Spotlight 定位系统), 291
area cover event distribution function (区域覆盖事件分布函数), 294
digital Signal Processing (DSP) (数字信号处理 (DSP)), 292
event Detection Function (EDF) (事件检测函数 (EDF)), 291
事件分布函数 (EDF) 291
event report (事件报告), 293
line scan event distribution function (线扫描事件分布函数), 294
localization function (定位函数), 291
location estimate (位置估算), 293
Spread Spectrum (SS) (扩频 (SS)), 330
code spreading (码扩频), 330
frequency hopping (跳频), 330
Sybil attack (Sybil 攻击), 333
Synthetic software benchmark (综合软件基准程序), 419

T

Telos mote (Telos 节点), 54-57
nRet Quadrature Phase-Shift Keying (O-QPSK) (偏移正交相移键控 (O-QPSK)), 54
Tiered Storage ARchitecture (TSAR) (层次存储数据存储结构 (TSAR)), 253-256
data centric query (以数据为中心的查询), 255
Geographical Hash Table (GHT) (地理位置哈希表 (GHT)), 254
interval skip graph (区间跳图), 256
three tier architecture (3 层体系结构), 254
TimeDiffusion synchronization Protocol (TDP) (时间扩散同步协议 (TDP)), 321-323
active phase (活跃状态), 321
Allan deviation (艾伦方差), 322
Allan variance (艾伦偏差), 321
diffused leader (扩散领导节点), 323
election/reduction procedure (ERP) (选举/改选过程 (ERP)), 321
external synchronization (外同步), 321
inactive phase (不活跃状态), 321
peer evaluation procedure (PEP) (同级评价过程) (PEP), 321
time diffusion procedure (TP) (时间扩散过程 (TP)), 322
Time Division Multiple Access (TDMA) (时分多路复用 (TDMA)), 77
T-MAC (Timeout MAC) (T-MAC (Timeout MAC) 协议), 83-88 (T 间休息)
activation event (激活事件), 84
contention resolution (竞争解决), 85
early sleeping (早睡问题), 85-86
future request to send (FRTS) (未来请求发送 (FRTS)), 86-87
TinyDB (TinyDB), 243-249

attribute based metadata（基于属性的元数据）（基于事件的元数据），248
event based query（基于事件的查询），246
actuation query（驱动式查询），246-247
event based data collection（基于事件的数据收集），246
network health query（网络健康度查询），246
offline delivery（离线发送），247
materialization point（物化点），245
power based query optimization（基于能量的查询优化），247-249
TinyOS（TinyOS），201-209
active message（主动消息），208
asynchronous code（AC）（异步代码（AC）），207
command（命令），202
configuration（配置），204
event driven operating system（OS）（事件驱动操作系统（OS）），201
event（事件），202
footprint optimization（内存优化），209
hardware/software transparency（软硬件透明性），209
interface（接口），203
locality-Aware TinyOS（LA-TinyOS）（局部性感知操作系统（LA-TinyOS）），209-215
adaption function（响应函数），211
anomalous event（异常事件），213
graceful length counter（局部性长度计数器），213
kernel component，214（内核组件），214
locality aware event（局部性感知事件），214
locality aware task（局部性感知任务），210
locality configuration data structure（局部性配置数据结构），211
locality configuration command（局部性配置命令），211
multiple-level scheduler（多级任务调度器），213
spatial locality（空间局部性），209
temporal locality case（时间局部性事例），213
temporal locality（时间局部性），209
Time-to-Expired（到时），211
module（模块），204
NesC（NesC），201
parameterized interface（带参数的接口），205
race free invariant（无竞争），207
SOS（SOS），215-219
final message（结束消息），216
function control block（FCB）（函数控制块（FCB）），217
init message（初始化消息），216
metadata（元数据），218
Mobile Oriented Application Platform（MOAP）（移动设备应用平台（MOAP）），217
module insertion（模块插入），217-218
module removal（模块删除），218
three-level task scheduler（三级任务调度器），215
synchronous code（SC）（同步代码（SC）），207
task scheduler（任务调度器），203
task（任务），202
TinySec（TinySec），352-353
in network processing（网内处理），353
IPSec（IPSec），352
SSH（SSH（secure shell）），352
SSL（SSL（secure socket layer）），352
TOSSIM（TOSSIM），434-437
analog-to-digital converter（ADC）（模拟数字转换器（ADC）），434
discrete event queue（离散事件队列），434
discrete event simulation（离散事件仿真），434
hidden terminal problem（隐藏终端问题），434
TCP socket（TCP 套接字），435
TinyOS component graph（TinyOS 组件图），434
TinyOS tool chain（TinyOS 工具链），435
TinyViz（TinyViz），436
TOSSIM event（TOSSIM 事件），436
UART packet（UART 数据包），436
TinyViz engine（TinyViz 引擎），436
TinyViz plugin（TinyViz 插件），436
Traffic Adaptive Medium Access（TRAMA）（流量自适应介质访问（TRAMA）），89-94
adaptive election algorithm（自适应选举算法）91-92
neighbor protocol（邻居协议），90-91
schedule exchange protocol（SEP）（调度交换协议（SEP）），92-93
Transport layer protocol（传输层协议），151
congestion（拥塞），152
centralized congestion control（集中式拥塞控制），152
localized congestion control（局部化拥塞控制），152
congestion avoidance（拥塞避免），153-154
three duplicate acknowledgement packet（三重确认包），153
timer out（超时），153
end to end reliable transmission（端到端可靠传输），151
hop to hop reliable transmission（逐跳可靠传输），151
Tungsten-balanced incandescent lamp（钨平衡白炽灯），470
two node model（双节点模型），356

U

Underwater Sensor Networks (USN) (水下传感器网络(USN)), 379
  Autonomous Underwater Vehicles (AUV) (自治水下航行器(AUV)), 16
  acoustic signal propagation (声波信号传播), 382-283
  attenuation (衰减), 382
  Doppler spread (多普勒扩展), 382
  multipath (多路径), 382
  onshore sink (陆上汇聚节点), 381
  surface sink (水面汇聚节点), 381
  surface station (水面站), 381
  underwater sensor (水下传感器节点), 381, 383-384
    oceanographic instrument (海洋仪器), 383
  USN protocol stack (水下传感器网络协议栈), 384
    data link layer (数据链路层), 385
    physical layer (物理层), 384-385
    routing layer (路由层), 386
    transport layer (传输层), 386-387

V

Vehicle Ad Hoc NETworks (VANET) (车载自组织网络(VANET)), 7
Vector based forwarding (VBF) (矢量转发(VBF)), 390-392
  Angle Of Arrival (AOA) (到达角度(AOA)), 391
  location dependent sensor data query (位置相关的传感器数据查询), 392
  location independent sensor data query (位置无关的传感器数据查询), 392
Video Sensor Networks (VSN) (视频传感器网络(VSN)), 399-412
  Panoptes (Panoptes), 401-403
    data buffering (数据缓存), 402
    data filtering (数据过滤), 402
    video capture (视频捕获), 402
    video compression (视频压缩), 402
  Cyclops (Cyclops), 403-405
    bus architecture (总线结构), 405
    complex programmable logical device (CPLD) (复杂可编程逻辑器件(CPLD)), 403
    external flash (外部闪存), 403
    external SRAM (外部SRAM), 403
VSN calibration (VSN定标), 405-407

W

Wireless Multimedia Sensor Network (WMSN) (无线多媒体传感器网络(WMSN)), 14-16
  homogeneous single tier design (同构单层设计), 15
  multi-tier network (多层网络), 15
Wireless Sensor and Actor Network (WSAN) (无线传感器/执行器网络(WSAN)), 363-378
  actor-actor coordination (执行器-执行器协同), 365
  actor (执行器), 363
  actuator (执行装置), 363
  automated architecture (自治架构), 365
  semi-automated architecture (半自治架构), 365
  sensor-actor coordination (传感器-执行器协同), 365
Wireless Sensor Network (WSN) (无线传感器网络(WSN)), 3
  analog sensing chip (模拟感应芯片), 3
  base station (基站), 109
  data reporting and aggregation (数据报告与聚合), 111
  microcontroller (微控制器), 3
  radio transceiver (无线收发机), 3
  sink (汇聚节点), 109
Wormhole attack (虫洞攻击), 333, 336-344
  dynamic Source Routing (DSR) attack (动态源路由(DSR)攻击), 336
  encapsulationEncryption (封装), 336
  high power transmission (大功率传输), 338
  out of band channel (带外信道), 337
  packet relay (报文转发), 338
  protocol deviation (协议偏离), 339
WSN protocol stack (无线传感器网络协议栈), 8
  application layer (应用层), 8
  data link layer (数据链路层), 8
  Mica2 (Mica2), 13
  MicaZ (MicaZ), 1
  Open System Interconnection (OSI) (开放系统互联(OSI)), 8
  physical layer (物理层), 8
  presentation layer (表示层), 8
  session layer (会话层), 8
  routing layer (路由层), 8
  transport layer (传输层), 8
    end to end (E2E) reliable transmission (端到端(E2E)可靠传输), 8
    network congestion (网络拥塞), 8
    packet retransmission (报文重传), 8
WSN time synchronization (无线传感器网络时间同步), 307-325
  absolute synchronization (绝对同步), 309
  clock correction (时钟校正), 316
  clock drift (时钟漂移), 309

clock frequency（时钟频率），309
clock skew（时钟频差），308
clock synchronization protocol（时钟同步协议），310
counter register（计数寄存器），308
deterministic synchronization（确定性同步），316
external synchronization（外同步），316
holding register（保持寄存器），308
internal synchronization（内同步），316
internal timer counter（内部定时计数器），308
mobile network synchronization（移动网络同步），317
network dynamics（网络动态性），314
network time protocol（NTP）（网络时间协议（NTP）），308
precision（精度），317-318
absolute precision（绝对精度），317
hardware clock（硬件时钟），317
logic clock（逻辑时钟），317
relative precision（相对精度），317
probabilistic synchronization（概率性同步），316
receiver to receiver synchronization（接收者-接收者同步），315
relative synchronization（相对同步），309
sender to receiver synchronization（发送者-接收者同步），315
software clock（软件时钟），308
stationary network synchronization（静态网络同步），317
time-stamp record（时间戳记录），307
unattended clock（自由时钟），316
WSN transport protocol（无线传感器网络传输协议），154，163
event-to-Sink Reliable Transport（ESRT）（事件到汇聚节点的可靠传输协议（ESRT）），163-171
hop to hop error recovery（逐跳错误恢复），156
message relaying（pump slowly）（消息转发（慢存）），158
pump slowly fetch quickly（PSFQ）（慢存快取（PSFQ）），154
relay initiated error recovery（fetch quickly）（转发节点发起的错误恢复（快取）），158
fetch timer（获取定时器），160
loss aggregation（丢包情息收集），160
proactive fetch（被动获取），161
signal strength based fetch（基于信号强度的获取），162
selective status reporting（选择性状态报告），159
message header（消息头），162
report bit（报告位），162

Z

Z-MAC（Zebra MAC）（Z-MAC（Zebra MAC）协议），102
distributed RAND（DRAND）（分布式时隙调度算法（DRAND）），102
explicit contentionnotification（ECN）（明确竞争通告（ECN）），105
high contention level（HCL）（高竞争级（HCL）），104
low contention level（LCL）（低竞争级（LCL）），104
maximum slot number（MSN）（最大时隙数（MSN）），104
Real-Time Transport Protocol（RTP/RCTP）（实时传输协议（RTP/RCTP）），104
timing synchronization protocol for wireless sensor networks（TPSN）（无线传感器网络时间同步协议（TPSN）），104
two hop neighborhood（两跳邻居节点），103