

山东大学

毕业论文(设计)

题目: ZigBee 网络广播通信协议性能
评估研究

姓 名 马永森
学 号 20061701128
学 院 控制科学与工程学院
专 业 自动化
年 级 2006 级
指导教师 李晓磊

2010 年 06 月 03 日

摘要

随着社会的发展与科技的进步，无线网络在人们的生产与生活中发挥着越来越重要的作用。ZigBee 作为一种短距离无线通信技术，具有低功耗、低速率、高安全和高容量等特点，在物联网、无线传感器网络等新兴技术的带动下，被广泛的应用于各个领域之中。目前，ZigBee 技术仍处于不断的发展与完善之中，在科研机构、联盟组织以及公司企业的广泛参与下，新的标准协议和应用框架也处于制定与开发之中，ZigBee 网络的性能评估与协议优化成为研究中的重点。

本文以 NS2 (Network Simulator, Version 2) 网络仿真工具为基础，设计并实现了节点呈三维立体分布的 ZigBee 网络仿真平台。NS2 仿真工具一般只能进行节点为二维平面分布的网络场景，在本文的 ZigBee 网络仿真模型中，节点的位置呈现为三维立体分布，且仿真场景较大，各个节点之间以点对点方式进行通信，这样就实现了复杂的 ZigBee 网络仿真平台的搭建。ZigBee 网络仿真平台基于 IEEE 802.15.4 标准规范，首先对工作方式、参数配置和应用扩展等进行相应设置，然后利用 Gawk 脚本语言对仿真模型输出的跟踪文件进行分析和处理，进而得出在不同工作方式下 ZigBee 网络的各项性能指标。

衡量 ZigBee 网络的性能指标主要有封包遗失率、传输成功率、传输时延、抖动率和吞吐量等，这些指标从不同的角度反映了 ZigBee 网络的运行性能。其中，封包遗失率和传输成功率反映了网络的可靠性，传输时延反映了网络的快速性，抖动率则反映了网络的稳定性。对于 ZigBee 网络而言，设备类型、数据发送时间间隔、信标模式及信标大小等都会影响到网络的运行性能。

关键字：ZigBee, IEEE 802.15.4, NS2, 性能评估, 吞吐量, 封包遗失率, 时延

ABSTRACT

With the development of society and the advancement of science and technology, the wireless networks are playing an increasingly important role in people's daily life. ZigBee is a short-range wireless communications technology, with, low rate, low power consumption high security and high capacity characteristics. Driven by the emerging technologies of the internet of things and wireless sensor networks, ZigBee are being widely used in various fields. Currently, ZigBee technology is still in development and improvement among the research institutions, ZigBee Alliance and companies, and the protocols and application frameworks are also in development, so that the evaluation of network's performance and protocol optimization become the focus of the study.

In this article, a simulation platform is designed based on NS2 (Network Simulator, Vision 2), in which all of the nodes are implemented in a three-dimensional distribution. The simulation model is performed in a peer-to-peer network so that data transmission between any two devices is supported. The ZigBee simulation platform is based on IEEE 802.15.4, and parameter configuration and application of extended are implemented first of all. Then the influence on ZigBee network's performance is analyzed and processed with the help of Gawk script language and the output trace file.

As to ZigBee, The performance indicators are mainly the packet loss rate, delivery rate, jitter, delay, and throughput. Among them, packet loss rate and delivery rate reflects the reliability of the network. Transmission delay reflects the network's fast, and jitter reflects the stability of the network. ZigBee network's performance is affected by the type of devices, the time interval of data transmission, beacon mode and the size of beacon.

KEYWORDS: ZigBee, IEEE 802.15.4, NS2, performance, throughput, packet loss rate, time delay

目 录

摘 要	i
ABSTRACT	ii
第一章 引言	1
1.1 ZigBee 技术简介	1
1.1.1 ZigBee 的技术特点	2
1.1.2 ZigBee 的应用前景	3
1.2 ZigBee 技术研究现状	4
1.3 本文研究方法 & 章节安排	5
第二章 ZigBee 网络的体系结构	7
2.1 IEEE 802.15.4	7
2.1.1 IEEE 802.15.4 网络的基本构成	7
2.1.2 IEEE 802.15.4 物理层	9
2.1.3 IEEE 802.15.4 介质访问控制子层	10
2.2 ZigBee 网络层	13
2.2.1 ZigBee 网络层服务实体	13
2.2.2 ZigBee 网络的路由寻址	14
第三章 NS2 仿真平台	16
3.1 NS2 简介	16
3.1.1 NS2 基本构成	16
3.1.2 NS2 在 Ubuntu 环境下的安装	17
3.2 NS2 仿真流程	19
3.2.1 OTcl 脚本语言	20

3.2.2	Trace 跟踪文件	20
3.2.3	Gawk 语言	21
3.2.4	Gnuplot 绘图	22
第四章	ZigBee 的 NS2 仿真模型	24
4.1	建立仿真模型	24
4.1.1	ZigBee 仿真模型的组成.....	24
4.1.2	ZigBee 仿真模型的工作方式设定.....	25
4.1.3	ZigBee 仿真模型的参数设定.....	27
4.2	性能指标	28
4.2.1	传输时延	28
4.2.2	传输抖动率	29
4.2.3	封包遗失率	29
4.2.4	吞吐量	30
4.3	仿真结果分析	30
4.3.1	链接建立时间	30
4.3.2	传输成功率	31
4.3.3	传输时延与抖动率	32
4.3.4	封包遗失率	32
4.3.5	吞吐量	33
	结论与展望	36
	致谢	37
	参考文献	38
	注 释	40

第一章 引言

近几年来,无线通信技术发展迅速,并逐渐渗透到人类生产与生活的各个方面。在超短距离无线通信方面,红外数据通信(Infrared Data Association, IrDA)技术、无线射频识别(Radio Frequency Identification, RFID)技术已经得到了广泛的应用;蓝牙(Bluetooth)技术以及超宽带(Ultra Wideband)技术主要应用于短距离无线通信,现在绝大多数的手机、笔记本电脑等设备都会配置蓝牙模块;基于IEEE802.11的无线高保真(Wireless Fidelity, Wi-Fi)技术适合于较远距离高速率无线通信,目前应用于机场、咖啡店、图书馆的无线网络便是依靠Wi-Fi技术的支持。

1.1 ZigBee 技术简介

对于多数的无线网络来说,一般要求系统具有数据传输速率高和传输距离远的特点。而在诸如工业控制、环境监测、商业监控、汽车电子、家庭数字控制网络等应用中,所需要的数据传输速率一般较低,系统所使用的终端设备通常为采用电池供电的嵌入式设备,比如无线传感器网络,这就要求系统必须具有成本低、功耗小的特点。针对这些特点和需求,由英国Invensys公司、日本三菱电气公司、美国库托罗拉公司以及荷兰飞利浦等公司在2002年共同宣布组成ZigBee技术联盟

(ZigBee Alliance)。从2004年ZigBee v1.0的推出到目前正在开发当中的ZigBee Smart Energy 2.0, ZigBee技术逐渐得到了各行各业的广泛关注,同时还有越来越多的企业和科研工作者投入到ZigBee产品的开发之中。

ZigBee是一种新兴的低功耗、低传输速率、低成本的无线网络技术,它是一种介于无线标记技术和蓝牙之间的技术提案,主要应用于近距离无线连接。目前,短距离无线通信技术已成为无线通信技术的一个重要分支,这是因为在现实生活中,存在着许多这样的应用情况:系统所传输的数据通常为小量的突发信号,即数据特征为数据量小,要求进行实时传送。如果采用传统的无线技术,虽然能满足上述要求,但存在着设备的成本高、体积大和能源消耗较大等问题。在这样的应用场合下,人们希望利用具有成本低、体积小、能量消耗小和传输速率低的短距离无线通信技术。ZigBee与其它的无线通信技术相比,在能耗与成本方面有着自己鲜明的特点和优势,这也使得ZigBee技术得到了越来越广泛的应用。

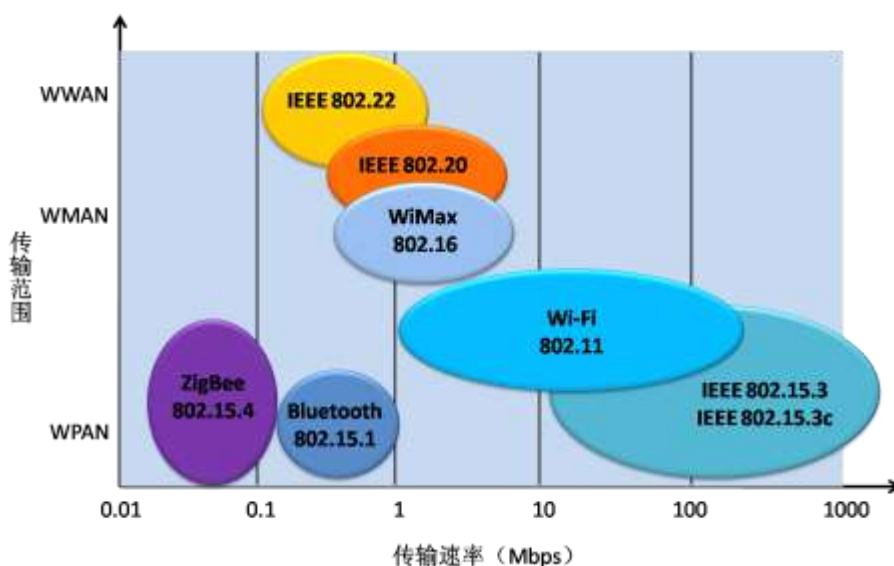


图 1-1 ZigBee 与其它短距离无线通信技术传输范围与速率的对比

1.1.1 ZigBee 的技术特点

每一种无线通信方式都有各自的特点，其应用场合也各有不同，ZigBee 技术的优势主要体现在以下几个方面：

- 低功耗

通常 ZigBee 网络中的设备所承载的应用数据速率都比较低，设备在不需通信时可以进入休眠状态，这时的能耗只有正常工作状态下的千分之一。ZigBee 协议中还定义了能量检测和链路质量指示功能，这样使得设备可以自动调整发射功率，从而能够在保证通信链路质量的前提下，最大限度的降低能量消耗。这样就使得每台设备的发射输出功率仅为 0~3.6dBm，在低功耗模式下，2 节普通的 5 号干电池可支持 1 个节点工作至少 6~24 个月。低功耗是 ZigBee 最大的技术优势。

- 低成本

ZigBee 的协议简单且所需的存储空间小，对通信控制器的要求较低，同时 ZigBee 协议是免专利费的，这样就有效降低了 ZigBee 的成本，每块芯片的价格仅 2 美元左右。

- 低速率

ZigBee 的工作速率较低，能够提供 250 kbps、40kbps 和 20kbps 的原始数据吞吐量，只能够满足低速率数据传输的应用需求。

- 短距离

在不使用功率放大器的前提下，ZigBee 节点的有效传输范围一般为 10-75m，能覆盖普通的家庭和办公场所。如果通过路由和节点间通信的接力，传输距离可以更远。

● 短时延

ZigBee 的响应速度较快，一般情况下，设备的搜索时延为 30ms，休眠激活时延为 15ms，活动设备信道接入时延为 15ms。这样一方面节省了能量消耗，另一方面使 ZigBee 技术更适用于对时延敏感的场所。

● 高容量

ZigBee 网络中的每个协调点能够同时管理 255 台设备，通过网络设备管理，每个网络中能够容纳多达 65000 台设备。

● 高安全

ZigBee 网络采用了碰撞避免机制，并为需要固定带宽的通信业务预留了专用时隙，从而能够避免发送数据时的竞争和冲突。同时，它还在物理层采用了扩频技术，能够在一定程度上抵抗干扰，在介质访问控制层采用完全确认的数据传输机制，保证了节点之间信息传输的高可靠性。另外，ZigBee 技术 提供了三级安全模式，包括无安全设定、使用接入控制清单防止非法获取数据以及采用高级加密标准的对称密码，以灵活确定其安全属性。

在无线通信领域内，各项技术没有绝对的优劣，每一项标准针对于不同的应用领域都有他自身的优势。表 1-1 中是主要的几种无线通信技术的对比，从中也能够看出其各自的优势与应用范围。

表 1-1 几种短距离无线通信技术的对比

规范	Bluetooth	UWB	Wi-Fi	ZigBee
标准	IEEE802.15.1	IEEE802.15.4a	IEEE802.11	IEEE802.15.4
工作频段	2.4GHz	3.1~10.6GHz	2.4GHz/5.8GHz	868/915MHz 2.4GHz
传输速率	723kbps	110Mbps~1.6Gbps	10~105Mbps	20kbps/40kbps 250kbps
传输距离	10m	4~20m	10~100m	10~300m
功耗	1~100mW	1mW	100mW	1~3mW
成本	高	较低	高	低
容量	8	128	32	65000

1.1.2 ZigBee 的应用前景

从 2002 年 ZigBee 联盟的成立到现在, ZigBee 技术得到了快速的发展, 目前已经有超过 68 家成员公司正在积极进行 ZigBee 规范的制定工作, 其中包括 Honeywell, Invensys, Motorola, Philips, Samsung 这样的国际知名企业, 因此 ZigBee 技术在未来的几年里有着良好的发展前景。同时, 近几年出现的物联网和无线传感器网络等新兴技术, 也为 ZigBee 技术提供了更好的发展契机。ZigBee 技术主要在以下几个方面有着广泛的应用:

● 工业

利用 ZIGBEE 无线网络特有的技术优势, 能够完成油田、电力系统的远程数据采集与监测; 将 ZigBee 技术与 GPRS/CDMA 结合起来, 可以实现远程抄表系统的构建; 利用 ZigBee 技术还可以建立工业现场的无线传感器网络^①。

● 农业

利用 ZigBee 节点收集各种土壤信息和气候信息, 或者实现湖泊、水库、河道的水文观测, 从而服务于农业生产。

● 商业

结合 ZigBee 与 RFID 各自的技术优势, 可以实现智能标签系统; 通过在餐厅、吧台、厨房、收银台、处理中心部署 ZigBee 节点设备, 构成完整的无线通讯网络, 可以实现餐厅的无线点餐系统; 通过 ZigBee 节点将各种设备互联, 可以实现物联网^②的搭建。

● 交通

由 ZigBee 技术与太阳能技术相结合, 可以用来搭建交通信号灯控制、道路信息采集与不停车收费系统; 利用 ZigBee 技术作为位置信息反馈的无线传输通道, 还可以实现智能公交系统的搭建。

● 医疗

ZigBee 设备可以作为老人与行动不便者的紧急呼叫器和医疗传感器; 由 ZigBee 无线网络可以实现病房的医疗监护和医疗仪器的数据采集。

● 家庭

通过 ZigBee 网络实现信息设备、通讯设备、娱乐设备、家用电器、监控装置等设备之间的互联, 使智能化、人性化的家居生活成为现实。

1.2 ZigBee 技术研究现状

在 ZigBee 技术的开发与研究过程中, 必然要对网络的性能指标进行评估, 进而根据评估结果网络做出相应改进。要对 ZigBee 网络的性能评估, 一种方法是以硬件为基础, 搭建实际的网络平台, 但实际网络系统的搭建往往代价较高; 另一种

方法是以软件为基础,对 ZigBee 网络进行实时仿真,这样不仅可以使得研究的周期缩短,还能够有效地降低成本。目前应用比较广泛的两款网络仿真软件是 Opnet 和 NS (Network Simulator, vision 2),其中 Opnet 一般用于大型网络的仿真,并且成本较高;而 NS 是一种完全开源免费的网络仿真平台,它所包含的模块几乎涉及到了网络技术的所有方面,这也使得 NS 在网络技术的研究过程中发挥着越来越重要的作用。

在最新版的 NS2 中包含了 IEEE 802.15.4 的仿真模型,这样使得对 ZigBee 的仿真相对简单。用户只需要编写 Tcl 脚本语言或者对 C++文件做简单的修改,可以建立起 ZigBee 网络的仿真模型,仿真模型会将网络运行的数据自动装入跟踪文件,最后编写 Gawk 程序对跟踪文件进行分析,进而得到网络的各项性能指标。

ZigBee 网络的性能指标主要有传输时延、抖动率、封包遗失率和吞吐量,其中传输时延与吞吐量反映了网络的快速性,抖动率反映了网络的稳定性,封包遗失率反映了网络的准确性。这些性能指标能够从各个方面体现网络的运行状态,并为网络协议的改进提供相对可靠地依据。

目前针对 ZigBee 的研究主要是网络性能的评估与协议的优化,其中文献[12]搭建了 ZigBee 的仿真模型,对 ZigBee 网络的性能进行了评估并与 IEEE 802.11 进行了对比;文献[10]对 ZigBee 的仿真模型进行了改进,并对星形和网状的 ZigBee 网络的性能进行了评估;文献[7]在原有的 ZigBee 模型基础上增加了 GTS 机制。

1.3 本文研究方法及章节安排

本文首先构造了节点位置呈三维分布的 ZigBee 网络仿真模型,使网络分别工作在信标使能模式和非信标使能模式,然后根据仿真结束后产生的跟踪文件,利用 Gawk 脚本处理语言得到网络的各项性能指标。当 ZigBee 网络工作在信标使能模式下时,通过改变信标的大小,分析网络性能的变化情况;在两种工作模式下,分别使数据封包大小取不同的数值,分析其对性能指标的影响。本文中研究的性能指标反映了 ZigBee 网络的稳定性、快速性和可靠性,主要包括封包遗失率、传输时延、抖动率、传输成功率和吞吐量。

本文在第一章中主要介绍了 ZigBee 技术的特点、应用和研究现状,下面几章中结构安排如下:

第二章对 ZigBee 网络的体系结构做了简要的介绍,主要包括 IEEE 802.15.4 对物理层和介质访问控制层的规范,以及 ZigBee 网络层的管理实体和路由寻址。

第三章主要介绍了 NS 2 仿真平台,包括 NS2 的结构组成和编程组成,NS 2 在 Ubuntu 环境下的安装, Otcl 脚本语言和 Gawk 脚本语言的编写,以及 Gnuplot

的使用。

第四章重点介绍了 ZigBee 仿真模型的建立、运行和分析。其中，模型的建立主要包括工作方式与参数的设定；仿真模型的运行包括工作模式与数据发送间隔等的设定；结果的分析包括工作模式和数据发送间隔对 ZigBee 网络的传输成功率、传输时延、抖动率、封包遗失率和吞吐量等性能指标的影响。

第二章 ZigBee 网络的体系结构

ZigBee 技术通过分层使得体系结构得到简化，每一层负责完成各自的任

并向上一层提供服务，各层之间的接口通过逻辑链路进行控制。ZigBee 技术的体系结构主要由物理（PHY）层、介质访问控制（MAC）层、网络层（NWK）以及应用层（API）组成，其各层之间的分布如图 2-1 所示。

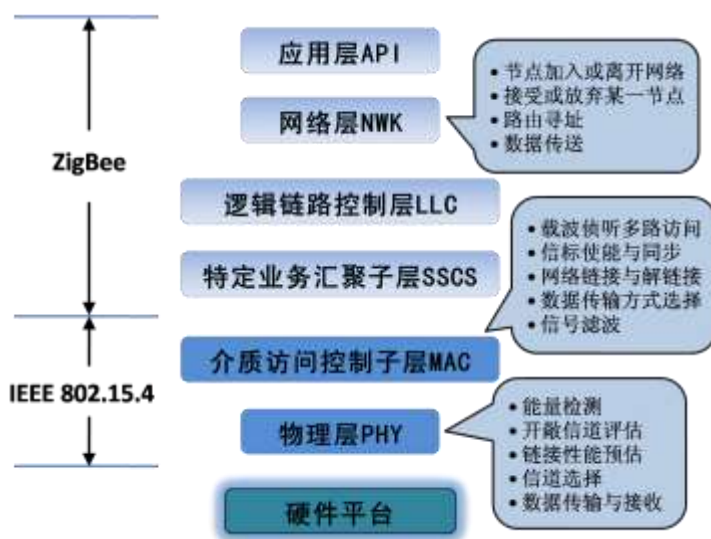


图 2-1 ZigBee 技术协议组成

从图 2-1 可以看出，ZigBee 技术的协议结构非常简单，只有三层，而并不像其他网络结构一样分为七层^③，其中物理层和介质访问控制层采用了 IEEE 802.15.4 协议标准。

2.1 IEEE 802.15.4

IEEE 802.15.4 标准对工作频段、拓扑结构等网络的基本构成进行了定义，并规定了物理层和介质访问控制层的基本功能。

2.1.1 IEEE 802.15.4 网络的基本构成

● 工作频段

IEEE 802.15.4 标准工作在免费的工业、科学、医疗(ISM)频段，并占有 2.4 GHz、

915 MHz 和 868 MHz 三个频段内的 27 个信道。对于不同的国家和地区，ZigBee 的工作频率是不同的，为了提高数据传输速率，IEEE 802.15.4 标准针对不同的频率范围规定了不同的信号调制方式。

表 2-1 IEEE 802.15.4 频段分配

频段 (MHz)	调制方式	码片速率 (kchip/s)	比特速率 (kb/s)	符号速率 (ksymbol/s)	符号
868~868.6	BPSK	300	20	20	二进制
902~928	BPSK	600	40	40	二进制
2400~2483.5	O-QPSK	2000	250	62.5	16 相正交

● 信道分配

在 IEEE802.15.4 标准使用的 3 个工作频段内，每一频段的宽度都不相同，因而分配的信道数也各不相同。IEEE 802.15.4 标准中定义了 27 个物理信道，其中 2.4GHz 频段定义了 16 个信道，915MHz 频段定义了 10 个信道，868MHz 频段定义了 1 个信道。每一信道的中心频率定义如下：

$$f_c = 868.3 \text{ MHz} \quad k=0 \quad (2-1)$$

$$f_c = 906 + 2 \times (k-1) \text{ MHz} \quad k=1, 2, \dots, 10 \quad (2-2)$$

$$f_c = 2405 + 5 \times (k-11) \text{ MHz} \quad k=11, 12, \dots, 26 \quad (2-3)$$

其中，k 为信道编号，频率和信道的分布如图 2-2 所示。

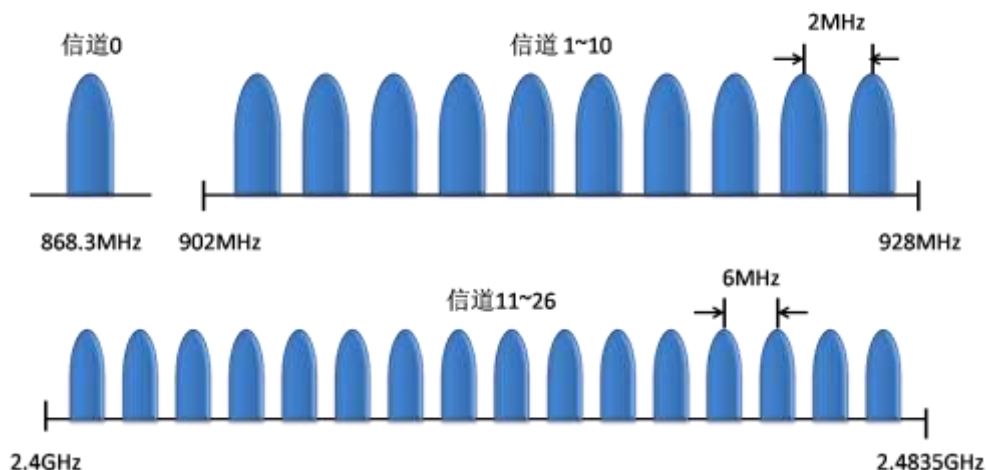


图 2-2 频率与信道分布

● 发射功率

发射功率在 IEEE802.15.4 标准中也有严格的限制，并会因地区的不同而有所差

异，一般情况下的发射功率范围为 $0 \sim +10$ dBm。发射功率也可根据需要，通过修改相应的服务原语^④进行控制。

● 接收灵敏度

接收灵敏度是在给定接收误码率的前提下，接收设备的最低接收门限值，通常用 dBm 表示。在这里是指在没有干扰且误码率小于 1% 的条件下，传送长度为 20 个字节的物理层数据包时，接收天线端所测量到的接收功率，通常的要求为 -85 dBm。

● 拓扑结构

IEEE 802.15.4 标准定义了两类物理设备，即完全功能设备（Full Function Device, FFD）和简化功能设备（Reduced Function Device, RFD）。完全功能设备可以作为网络协调点、协调点和普通节点，并能够和其它的完全功能设备或是简化功能设备通信。简化功能设备只能同完全功能设备进行通信，而且只能完成相对简单的应用。一个 IEEE 802.15.4 网络可以是简单的星型单跳网络或是具有自组织功能的多跳网络，但其中至少要存在一台完全功能设备。网络中的每台设备可以使用 64 位的 IEEE 地址或者是 16 位的短地址，因此每一个 IEEE 802.15.4 网络能够容纳 64k 台设备。

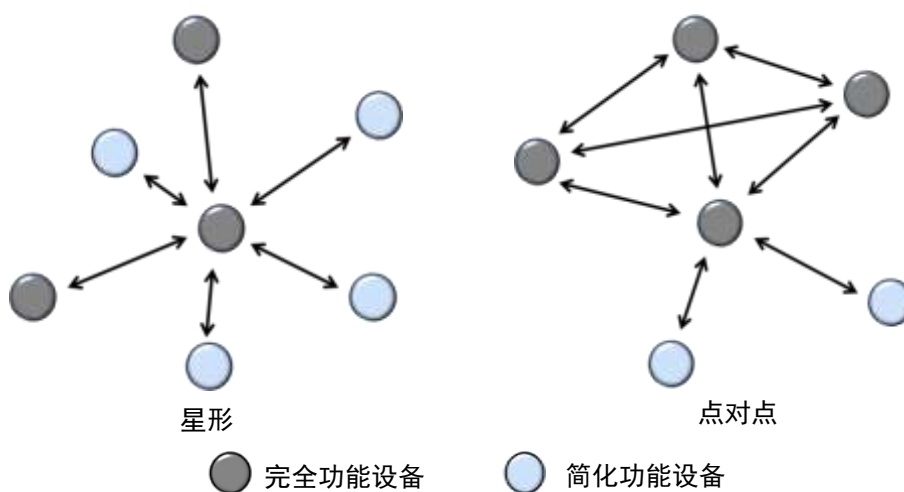


图 2-3 IEEE 802.15.4 网络拓扑结构

2.1.2 IEEE 802.15.4 物理层

IEEE 802.15.4 的物理层 (PHY) 通过硬件平台提供了介质访问控制子层与物理层之间的信道接口。在 IEEE 802.15.4 的物理层中包含一个物理层管理实体 (Physical Layer Management Entity, PLME)，该实体通过调用物理层的管理功能函数，为物理层管理服务提供其接口。同时，物理层管理实体还负责维护由物理层所管理的目标数据库。物理层通过服务接入节点 (Service Access Point, SAP) 提供物理层的

数据传输与管理,以及在对等网络中的介质访问控制层实体之间传输协议数据单元 (Media Protocol Data Unit, MPDU)。IEEE 802.15.4 的物理层主要完成下面的工作:

- 启动和停止无线收发装置

无线收发装置有三种工作状态,分别是发送、接收和休眠,其中发送和接收之间的转换时间必须在 12 个典型周期之内。为了减少对其他设备和系统的干扰和影响,在保证设备能够正常地工作条件下,每个设备的发射功率应尽可能的小。发射功率过大将造成接收输入端堵塞,无法正常工作。在 IEEE 802.15.4 标准下,要求接收机的最大输入电平应大于等于-20 dBm

- 能量检测 (Energy Detection, ED)

能量检测是信道选择算法中的一个重要组成部分,它对在 ZigBee 信道带宽之内,对所接收到的信号功率进行估计,而不需要在信道上对信号进行鉴别和译码。通常能量检测的时间为 8 个符号周期,其结果可以协助网络层作信道选择选法,或是用来作开敞信道评估 (Clear Channel Assessment, CCA)。

- 链接性能指示 (Link Quality Indication, LQI)

链路品质信息表示了所接收的数据包强度和品质方面的特性,在 IEEE802.15.4 标准中,对于每一个接收封包都要进行链接性能指数评估,主要依靠物理层的接收能量检测、信噪比估计或者是两者相结合的方法来实现。

- 开敞信道评估 (Clear Channel Assessment, CCA)

物理层通过能量检测或是载波侦听对开敞信道进行评估——当检测到的能量值高于能量阈值或者检测到存在载波,则认为传输介质处于繁忙状态。

- 数据传输与接收

当无线网络工作在 2.4GHz 频段下时,物理层把每四位信息映射为 32 位的伪随机噪声 (Pseudo-random Noise, PN) 序列,这些序列再通过偏移量相移键控 (Offset Quadrature Phase Shift Key, O-QPSK) 与载波进行联合与调制。

2.1.3 IEEE 802.15.4 介质访问控制子层

IEEE 802.15.4 的介质访问控制子层 (MAC) 提供了物理层和特定业务汇聚子层 (Service Specific Convergence Sublayer, SSCS) 之间的接口,并通过介质访问控制层管理实体 (MAC Layer Management Entity, MLME) 提供数据传输服务与管理服务。IEEE 802.15.4 的介质访问控制子层主要完成以下的几项任务:

- 信标的生成与同步

当系统工作在信标使能模式下^⑤,协调点会周期性的发出信标以使设备同步。在信标使能模式下,与协调点链接的设备通过跟踪信标来与之同步,信标同步对于

数据轮询、节能和孤立点检测具有重要意义。信标在每一帧的开始时进行发送，且信标的发送具有比其他的发送及接收更高的优先级。

- 定义超帧结构

IEEE 802.15.4 标准下允许使用超帧结构，其格式由协调点决定。如图 2-4 所示，超帧之间以信标为界，一般由动态部分和静态部分组成。其中，动态部分又可以分为竞争接入期（Contention Access Period, CAP）和竞争避免期（Contention Free Period, CFP），每个竞争避免期可以容纳 7 个预留时隙，而每个竞争接入期必须为其它的设备申请入网预留出充足的时间。

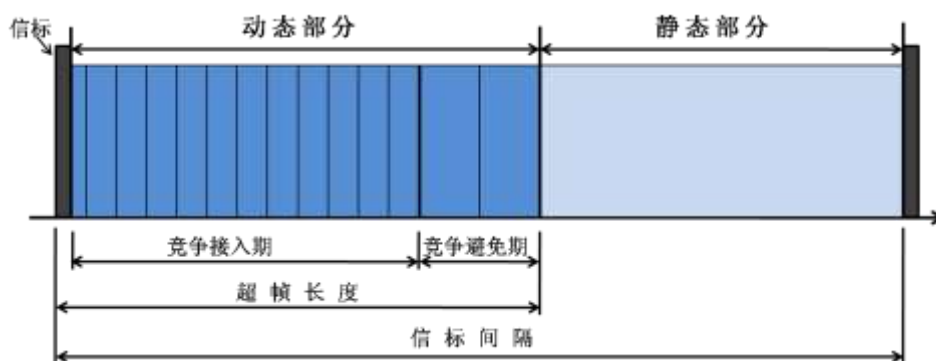


图 2-4 超帧结构

图 2-4 中的信标间隔（Beacon Interval, BI）和超帧长度（Superframe Duration, SD）定义如下：

$$BI = aBaseSuper frame Duration \times 2^{BO} \quad (2-4)$$

$$SD = aBaseSuper frame Duration \times 2^{SO} \quad (2-5)$$

其中，aBaseSuperframeDuration = 960 symbols, BO 为信标序号（Beacon Order），SO 为超帧序号（Superframe Order），信标序号和超帧序号都由协调点所决定。所有的基于竞争的任务必须在竞争避免期开始前完成，同时，所有的基于预留时隙的任务必须在竞争避免期结束前或是下一个预留时隙开始前完成。

- 链接的建立与断开

802.15.4 标准在介质访问控制子层加入了链接与解链接功能，这样不仅可以使星型网络自动建立，而且能够建立起自组织的点对点网络。为了提高节点之间链接的可靠性，介质访问控制子层采用了多种机制，包括帧的确认与重发以及带有 16 位循环冗余码校验（Cyclic Redundancy Check, CRC）的数据校验。

- 信道接入

IEEE 802.15.4 标准也是采用带冲突避免的载波侦听多路访问（CSMA/CA）机制，但考虑到传输速率比较低，这里并没有发送应答机制。在信标使能模式下，协调点可以将动态超帧的一部分作为预留时隙（Guaranteed Time Slot, GTS），当介质访问控制层在竞争其中传送信息时，将使用带时隙的 CSMA/CA 算法；在非信标使能模式下，介质访问控制层将使用非时隙的 CSMA/CA 算法来发送信息。

● 数据传输

IEEE 802.15.4 标准中定义了三种数据传输模式：

- 由普通节点到协调点的数据传输
- 由协调点到普通节点的数据传输
- 由某个普通节点到另一普通节点的数据传输

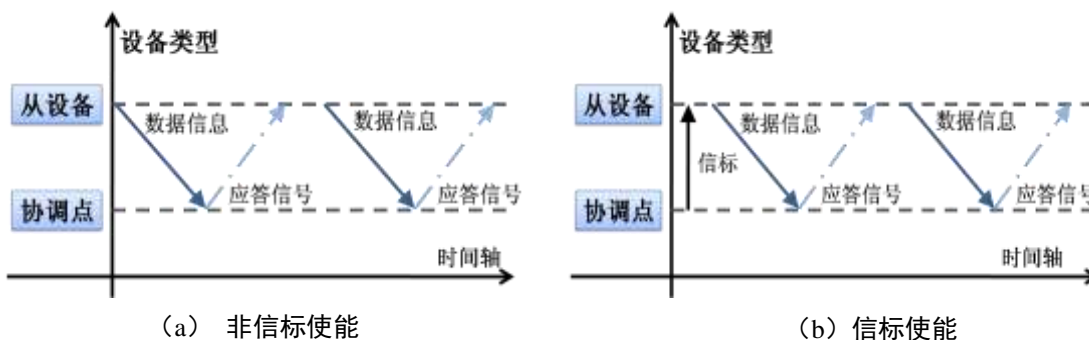


图 2-5 普通节点到协调点的数据传输

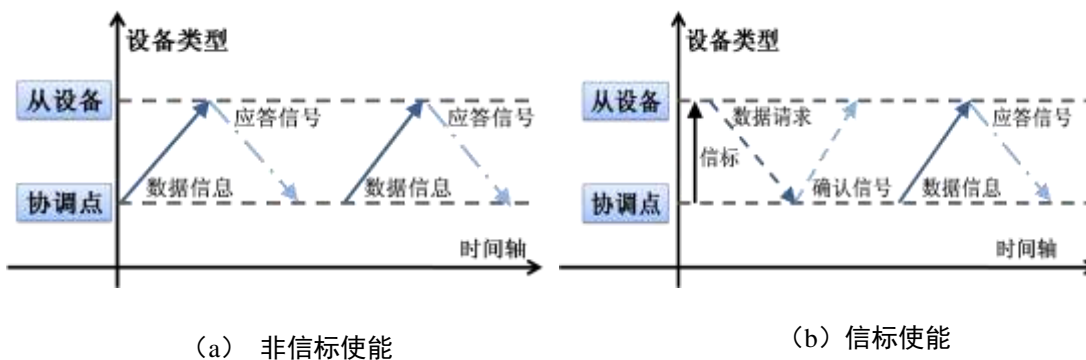


图 2-6 协调点到从设备的数据传输

在前两种数据传输方式中，根据信标的有无，数据传输的过程也是不同的，如图 2-5 及图 2-6 所示。

为了便于研究网络的性能，又可以将数据传输分为以下三种：

- 直接数据传输

无论是两台设备之间还是设备与协调点之间，都可以采用直接数据传输方式进

行通信。

■ 间接数据传输

间接数据传输仅适用于协调点到普通设备的通信,在这种传输模式下,协调点会把数据帧放置在传输队列中,外围设备通过检测信标帧的有无,便可判断在传输队列中有没有待处理的封包。在一些特殊情况下,间接数据也会发生在非信标使能状态下,比如说协调器在链接过程中把链接应答帧放入传输队列,外围设备通过轮询机制进行提取。

■ 预留时隙数据传输

这种方式只适用于普通设备到协调点的数据传输,无线网络中的设备一般是由电池提供能量,IEEE 802.15.4 标准充分考虑到了能量供给问题,超帧结构和间接数据传输等都能够起到降低能耗的作用。

2.2 ZigBee 网络层

ZigBee 网络层的主要功能是为应用层提供合适的服务接口,并确保介质访问控制层的正常工作。其中,前者主要是靠网络层的服务实体来实现,后者主要靠信息传输过程中的安全机制以及路由发现、维护和转交实现。

2.2.1 ZigBee 网络层服务实体

为了向应用层提供接口,网络层提供了数据服务实体和管理服务实体。其中,数据实体通过数据实体服务接入点提供数据传输服务;管理实体则通过管理实体服务接入点完成网络的管理工作,并负责网络信息库的管理与维护。

网络层数据实体为数据传输提供服务,它规定设备之间的数据传输要符合应用协议数据单元(Application Protocol Data Unit, APDU)的格式,并且这些设备必须处于同一个网络之中。网络层数据实体主要提供如下服务:

- 生成网络层协议数据单元(Network Protocol Data Unit, NPDU)

网络层数据实体通过增加一个适当的协议标头,从应用支持层协议数据单元中生成网络层的协议数据单元。

- 指定拓扑传输路由

网络层数据实体能够发送网络层的协议数据单元到一台合适的设备,该设备可能是最终目的设备,也可能是通信链路中的中间通信设备。

网络层管理实体提供网络管理服务,并允许应用与堆栈之间的相互作用。网络层管理实体主要提供如下服务:

- 配置一台新设备

为保证设备正常工作的需要，设备应该具有足够堆栈。配置选项包括对协调器和网络设备的初始化操作。

- 网络初始化

- 连接和断开网络

能够连接或者断开一个网络，并可以为建立一个 ZigBee 协调器或者 ZigBee 路由器，使某台设备同网络断开。

- 寻址

ZigBee 协调器和 ZigBee 路由器具有为新加入的网络设备分配地址的能力。

- 邻居设备发现

具有发现、记录和汇报有关单跳邻居设备信息的能力。

- 路由发现

具有发现和记录有效地传送信息的网络路由的能力。

- 接收控制

具有控制设备接收机接收状态的能力，即控制接收机什么时间接收、接收时间的长短，以保证介质访问控制层的同步和正常接收。

2.2.2 ZigBee 网络的路由寻址

ZigBee 网络层支持 Tree、Z-AODV 以及 Tree 与 Z-AODV 相结合的路由算法。

- Z-AODV 路由算法

对于传统的无线网络，主要有 DSDV，DSR，AODV 等路由算法。其中，DSDV（Destination-Sequenced Distance-Vector）协议采用时间驱动和事件驱动技术控制路由表的传送，即每个移动节点都保留一张路由表，其中包括所有有效目的节点、路由跳数、目的节点路由序列号等信息，目的节点路由序列号用于区别有效和过期的路由信息以避免环路产生；DSR（Dynamic Source Routing）协议是最早采用按需路由思想的路由协议，包括路由发现和维护两个过程，它的主要特点是使用了源路由机制进行数据包转发；AODV（Ad-hoc On-demand Distance Vector）协议以 DSDV 协议的逐跳路由、序列号、定期广播机制为基础，加入了 DSR 的按需路由发现和维护机制。

与 DSDV 保存完整的路由表不同的是，AODV 通过建立按需路由来减少路由广播的次数，这是 AODV 对 DSDV 的重要改进。与 DSR 相比，AODV 的好处在于源路由并不需要包括在每一个数据包中，这样会降低路由协议的开销。AODV 是一个纯粹的按需路由协议，那些不在路径内的节点不保存路由信息，也不参与路由

表的交换。

Z-AODV 算法是针对 AODV 算法的改进, AODV 是基于序列号的路由, 它总是选择最新的路由。Z-AODV 是基于路径的能量消耗的路由, 考虑到节能、应用方便性等因素, 简化了 AODV 的一些特点, 但仍保持 AODV 的原始功能。Z-AODV 需要首先建立路由发现, 然后选择跳数较少的路由, 成为最佳路径, 因此 Z-AODV 更适用于连续的数据传输。

● Tree 路由算法

Tree 路由是一种由网络协调器展开生成树状网络的拓扑结构, 适合于节点静止或者移动较少的场合, 属于静态路由, 不需要存储路由表。树型路由对传输数据包的响应较快, 因为树型路由不需要建立路由表。其缺点是所选择的路由并非最佳的路由, 不能获得最小路由, 树型路由适用于突发型的数据传输。

● Z-AODV 与 Tree 相结合的路由算法

在 ZigBee 网络中可以将 Z-AODV 和 Tree 两种路由算法相结合, 使用 Z-AODV 和分等级的 Tree 路由并存的混合路由方法, 构成网格型拓扑结构的网络。对于连续型的数据传输, 采用 Z-AODV 路由首先建立路由发现, 然后选择跳数相对较少的路由, 成为最佳路径; 对于突发型的不连续数据传输, 则使用抑制路由发现的方法, 即在路由表中没有响应的目的节点的地址时, 采用 Tree 路由方法, 因为这种路由不需要建立路由表, 因此对传输的数据响应较快。这种路由选择机制在网络性能和低功耗方面有明显的优势, 在能量控制机制的共同作用下, 可以有效地平衡节点能量, 避免节点耗尽能量而过早地失去作用。

第三章 NS2 仿真平台

3.1 NS2 简介

NS (Network Simulator) 是由美国加州的 Lawrence Berkeley 国家实验室开发的网络仿真软件，它的前身是 S.keshavs REAL 仿真器，现在由南加州大学、施乐公司 (Xerox)、加州大学与 Lawrence Berkeley 国家实验室协作研发。NS 在包括多数的有线网络和无线网络在内的研究中得到了非常广泛的应用，由于 NS 的源代码是完全对外开放的，在众多的科研人员的共同努力之下，目前 NS 中几乎包括了所有的通信协议的仿真实现。同时，能够实现大型网络仿真的 NS3 也正处在开发和完善的过程中，这也使得 NS 逐渐成为网络协议的研究与开发中必不可少的工具。

3.1.1 NS2 基本构成

NS2 主要包含 Tcl/Tk、OTcl、NS2 和 Tclcl，其中 Tcl 是一个开放脚本语言，用来对 NS2 进行编程；Tk 是 Tcl 的图形界面开发工具，可帮助用户在图形环境下开发图形界面；OTcl 是基于 Tcl/Tk 的面向对象扩展，有自己的类层次结构；NS2 为本软件包的核心，是由 C++编写的面向对象的仿真器，它以 OTcl 解释器作为前端；Tclcl 则提供 NS2 和 OTcl 的接口，使对象和变量出现在两种语言中。从图 3-1 中可以看出 ns-allinone-2.34 的结构组成。

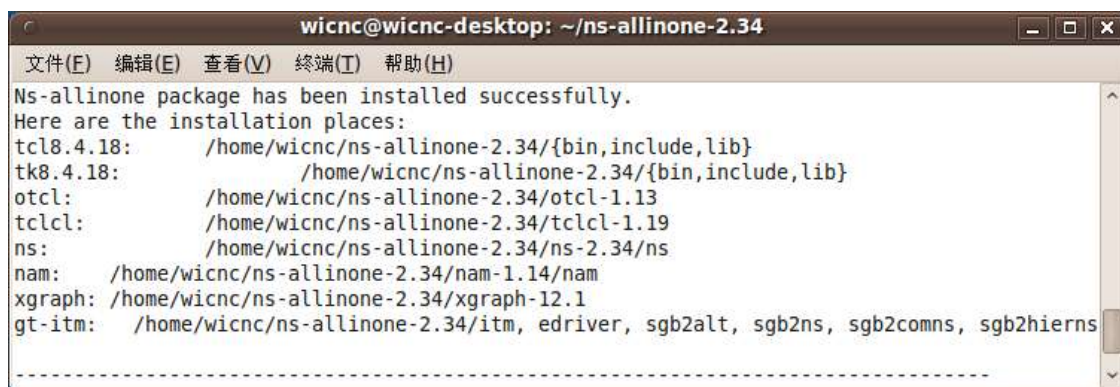


图 3-1 ns-allinone-2.34 的结构组成

另外 NS2 中还包含一些可选的软件包：通过 nam 可以根据网络模拟软件的 trace 输出文件来运行动画；利用 gawk 和 trace 跟踪文件可以对整个仿真过程进行分析；

通过 gdb 可以进行程序的调试；以及通过 Gnuplot 和 Xgraph 可以对数据文件进行图形界面显示等。

在编程组成方面，NS2 采用两级体系结构。为了提高代码的执行效率，NS2 将数据操作与控制部分的实现相分离，事件调度器和大部分基本的网络组件对象后台

使用 C++实现和编译，称为编译层，主要功能是实现数据包的处理；NS2 的前端

是一个 OTcl 解释器，称为解释层，主要功能是对模拟环境的配置、建立。从用户的角度来看，NS2 是一个具有仿真事件驱动、网络构件对象库和网络配置模块库的 OTcl 脚本解释器。NS2 中编译类对象通过 OTcl 连接建立了与之对应的解释类对象，这样用户间能够方便地对 C++对象的函数进行修改与配置，充分体现了仿真器的一致性和灵活性。一般情况下的网络仿真，无需修改 C++文件，只需编写 OTcl 脚本即可。当 NS 中没有所需的网络元素时，则需要修改相关的 C++文件，从而实现 NS 的功能扩展。

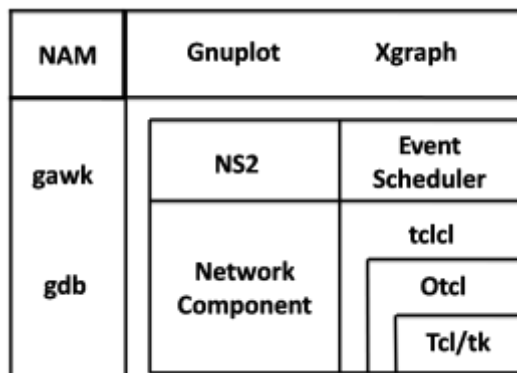


图 3-2 NS2 的基本模块构成

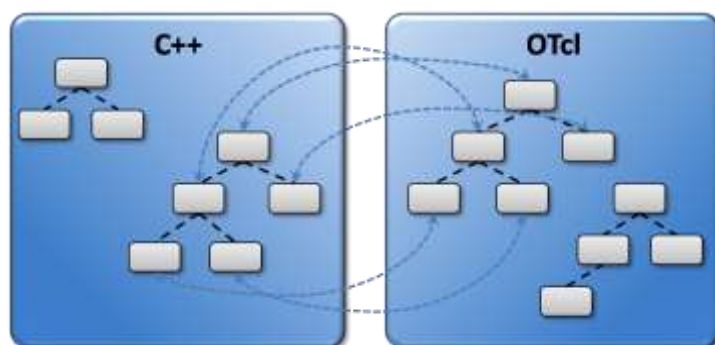


图 3-3 NS2 的编程语言构成

3.1.2 NS2 在 Ubuntu 环境下的安装

NS2 主要基于 UNIX 平台，需要 tcl-8.0.5、tk-8.0.5、otcl-1.0a4、tclcl-1.0b8 和 ns-2.1b6 的支持，同时可以稳定运行于 Linux 平台。若以 Windows 为平台，则需预先安装 Cygwin，由于 Cygwin 的安装过程较为繁琐，并且在 Windows 环境下 NS2 的运行不是很稳定，这里以 Linux 作为平台，并以 Ubuntu 为例简要介绍 NS2 的安装过程。

- 首先需要安装必要的软件包，在 Ubuntu 平台下打开一个新的终端，分别输入以下内容：

```
$ sudo apt-get install build-essential
```

```
$ sudo apt-get install tcl8.4 tcl8.4-dev tk8.4 tk8.4-dev
```

```
$ sudo apt-get install libxmu-dev libxmu-headers
```

- 下载 NS2 软件包，这里以 ns-allinone-2.34 为例，下载完成后，将 NS2 软件包解压并移动到安装位置^⑥：

```
$ tar xvfz ns-allinone-2.34.tar.gz
```

```
$ sudo mv ns-allinone-2.34 /home/wicnc
```

- 由于 gcc 4.0 版本是用 ld -share 来生成共享库的，但是在 gcc 4.0 以上的版本，这个命令改为了 gcc -shared 即可。如果使用的是 Ubuntu9.10 及以上系统，则需改变配置：

```
cd /ns-allinone-2.34/otcl-1.13
```

```
sudo gedit configure.in
```

这时会打开一个 gedit 编辑器，将文件 configure.in 中的 SHLIB_LD="ld -shared" 改为 SHLIB_LD="gcc -shared"，保存并退出；

```
sudo gedit configure
```

再将文件 configure 中的 SHLIB_LD="ld -shared" 改为 SHLIB_LD="gcc -shared"。

- 重新进入 ns-allinone-2.34 文件夹，安装 ns-allinone-2.34：

```
cd
```

```
cd ns-allinone-2.34
```

```
./install
```

- 安装结束后，会出现修改环境变量的提示，如图 3-4 所示。



图 3-4 修改变量提示

将提示窗口中的内容添加到文件 .bashrc 中。

```
sudo gedit .bashrc
```

将终端中提示的内容添加到打开的文件末尾，然后保存并关闭文件。

- 环境变量修改后 ns-allinone-2.34 的安装基本就完成了，打开一个新终端，输入 ns 并回车，若出现%则表明 ns-allinone-2.34 安装成功。若要运行仿真实例，需要打开一个新的终端，以 ns *.tcl 的方式执行脚本文件，并可以通过 nam 来观察仿真的运行情况，如图 3-5 所示。

cd zigbee/demo

ns beaconeabled.tcl

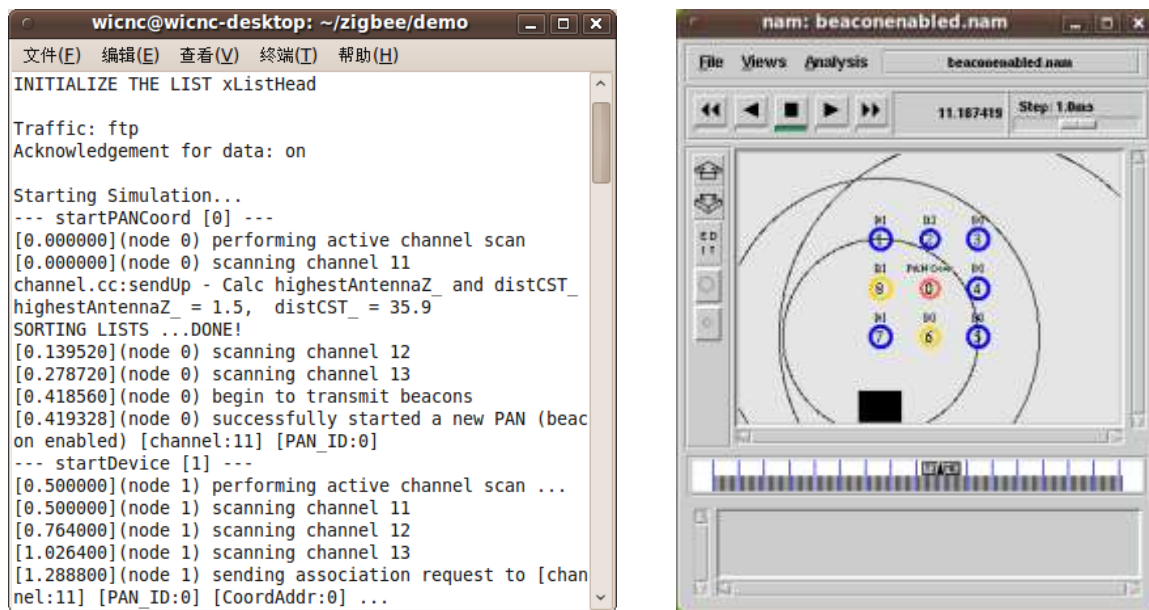


图 3-5 NS2 仿真运行

3.2 NS2 仿真流程

NS 安装成功后可以运行简单的 Tcl 脚本以便对仿真流程有一个大致的认识，通常情况下一完整的网络仿真主要由以下几个部分组成：

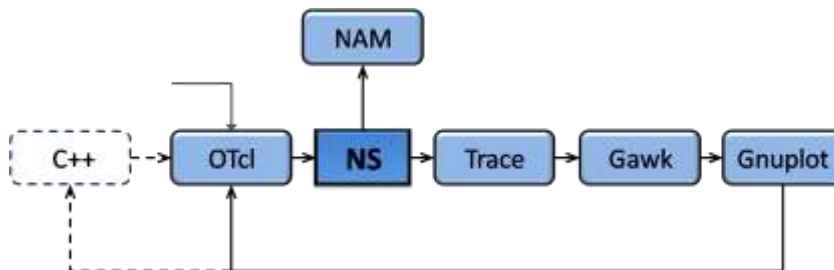


图 3-6 NS2 网络仿真流程

- 编写 OTcl 脚本，定义网络的基本属性，主要包括节点配置、拓扑结构、协议选择、仿真时间等；
- 利用 NS 执行编写好的 OTcl 脚本，生成*.tr 和*.nam 文件，必要时可以使用 gdb

对程序进行调试；

- 由 Nam 可以对仿真过程有一个直观的展现，再利用 Gawk 编程对 Trace 文件进行分析，可以得到网络各方面的性能指标；
- 由 Gnuplot 将各项性能指标以坐标图的形式呈现，从而做出更直观的对比；
- 修改 OTcl 脚本以便重新配置网络，重复以上的仿真过程。必要时可以修改 C++ 文件。

3.2.1 OTcl 脚本语言

OTcl 实际上是在 Tcl (Tool Command Language, 工具命令语言) 的基础上做了一个面向对象的封装，成为一种面向对象的脚本语言。Tcl 是一种非常通用的脚本语言，在包括 Windows, UNIX, Mac 在内的常用平台上都能够解释运行。在引入了面向对象机制后，Tcl 的功能更加强大，使用也更加方便。Otcl 支持继承、封装和多态性等面向对象的概念，这点与 C++, Smalltalk 等面向对象的语言类似，但 OTcl 的编程风格跟 C++ 又有着很大的不同。

表 3-1 OTcl 与 C++对比

OTcl	C++
语法简单，编程灵活，代码紧凑，易于维护；可以由 C, C++, Java 进行扩展，移植性强。	语法要求严格，标准体系庞大，开发维护复杂；支持 C 的几乎全部功能，但与 Java 并不兼容。
在 Windows, UNIX, Mac 等多种平台上都能够解释运行；经过扩展后能够支持面向对象程序设计。	广泛应用于 Windows, Unix 等多种平台；支持面向对象程序设计、泛型程序设计等多种程序设计风格。
过于简单的语法规则使得其通用性较差，执行效率较低。	经过多年的发展已形成规范统一的标准，并得到广泛应用，执行效率高。

由 OTcl 脚本编写无线网络场景首先是设置仿真参数，主要包括信道类型、广播模型、网络接口类型、接口队列类型、拓扑结构、路由协议等方面的选择；然后是仿真器的初始化，包括有仿真器与拓扑对象的建立，Trace 文件和 nam 文件的设置；接着是节点的有关属性与位置分布的设置，以及数据流向的定义；最后是结束程序的定义和仿真的启动与停止。

OTcl 脚本主要是对节点属性和数据传输进行定义，其他的配置大多是在 C++ 文件中进行定义，并通过 OTcl 进行调用。有关无线网络的 OTcl 脚本将在下一章中详细介绍。

3.2.2 Trace 跟踪文件

OTcl 脚本运行结束后一般会产生一个或是多个*.nam 和*.tr 文件，其中*.nam 文件是用来动态展示仿真过程的，操作也比较简单。*.tr 文件以固定的格式记录下了仿真的整个过程，包括事件类型、发生时间、封包类型、封包大小及地址信息等，对此进行分析便可以得出仿真网络的各项性能指标。

ns-allinone-2.34 中定义了 Trace 文件的基本格式，而不同的网络环境下，Trace 文件的格式也是不同的，其中 IEEE 802.15.4 协议下的 Trace 文件格式是由 p802_15_4trace.cc 规定的。具体的格式如表 3-2 所示。其中事件类型分为 s：传送（Send），r：接收（Receive），D：丢弃（Drop），f：转送（Forward）。事件分层分为 AGT：应用层（Agent Trace），RTR：路由层（Router Trace），MAC：介质访问层（MAC Trace）。

表 3-2 无线网络跟踪文件格式

事件类型	发生时间	节点	事件分层	丢包原因	封包地址	封包类型	封包大小	地址信息
.....
s	2.267200033	_4_	MAC	---	0	CM7	8	[0 ffffffff 4 0]
D	2.267648067	_5_	MAC	APS	0	CM7	8	[0 ffffffff 4 0]
r	2.267712047	_1_	MAC	---	0	BCN	20	[0 ffffffff 4 0]
.....

3.2.3 Gawk 语言

生成 Trace 跟踪文件后需要进一步的分析才能够得到网络的性能指标，这时便要用到 Gawk 语言。Gawk 是 awk 的 GNU 版本，它能够实现 awk 的所有功能，它在包括 Linux 在内的任何平台上都能够实现强大的数据处理功能。

总的来讲，Gawk 是一种用于处理文本的编程语言工具，它在很多方面与 Shell 语言类似，能够实现样式装入、数学运算、流程控制、进程控制等基本功能，只要编写一段简短的程序，还可以实现文件输入、数据处理、数据排序、生成报表等许多复杂功能。Gawk 还具有自己的一些特点，比如内部使用直译器而无需预先编译；变量无类型之分并且可以使用文字作为数组标注^⑦等，因此在编写程序时 Gawk 语言相比于其他语言更加方便简洁且高效。另外一方面，Gawk 的一些内建功能使它更善于处理数据列、字段类型的数据。尤其要指出的是，Gawk 内建有 pipe 功能，能够将正在处理中的数据传送给外部的 Shell 命令加以处理，然后再将 Shell 命令处理后的数据返回到 Gawk 程序，这也使得 Gawk 能够很容易的使用系统资源。

Gawk 在读取数据文件时以数据行为基本单位，每一数据行又分为不同的字段，以无线网络的 Trace 文件为例，一般是用空格符来分隔相邻字段的，如表 3-3 所示。

表 3-3 跟踪文件中的数据行与字段

\$1	\$2	\$3	\$4	\$5: \$6	\$7	\$8	\$9	\$10: \$13
D	2.267648067	_5_	MAC	APS	0	CM7	8	[0 fffffff 4 0]

在表 3-3 中，\$1: \$13 是字段变量，程序执行过程中会不停的装入数据行的值，结合 Trace 的格式并加以简单的运算，便可以得到仿真的各项性能指标。

Gawk 的指令与 C 语言非常类似，一个典型的 Gawk 程序的基本结构如下所示：

Pattern1 { Actions1 }

Pattern2 { Actions2 }

.....

PatternN { ActionsN }

其中 Pattern 为“关系判断式”，程序运行过程中先判断 Pattern 的结果，若为真则执行相应的 Actions，若为假则不执行。下面是分析无线网络性能指标的 Gawk 框架：

```

BEGIN {
    # 变量初始化
}
{
    # 读取跟踪文件中有用数据
    # 根据条件将数据存入指定变量
}
END {
    # 文件读取结束后对数据进行处理
    # 将所得性能指标输出到指定文件
}
    
```

图 3-5 给出了 Gawk 程序的工作流程。流程中的判断循环是自动重复进行的，在程序中不需要去编写这个循环。

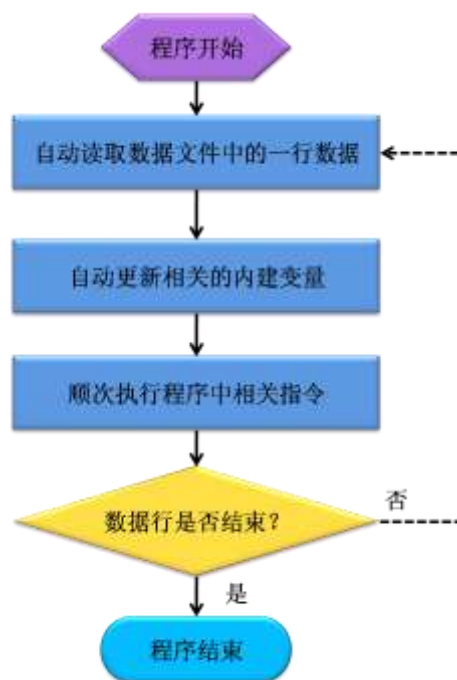


图 3-7 Gawk 程序工作流程

3.2.4 Gnuplot 绘图

Gawk 一般会将计算所得的各项性能指标装入不同的文件中，我们可以利用 Gnuplot 将这些数据以图表的形式呈现出来，进而做出更进一步的分析。

Gnuplot 是一种具有命令导向功能的交互式绘图程序（Command-driven

Interactive Function Plotting Program), 在终端中输入命令行可以设定或修改绘图环境, 它可以在下面两种模式下运行: 在图表正常显示的情况下, 通过终端在交互模式下绘制图表; 或者是从文件中读取命令, 以批处理的方式生成图表。这里可以通过 Gawk 脚本语言计算出网络的各项性能指标并写入不同的文件中, 然后利用 Gnuplot 在交互模式下将文件中的数据显示为坐标图。

Gnuplot 的使用非常简单, 在终端中输入 `gnuplot` 并按回车, 便可以进行坐标图的绘制。用户可以随时改变程序的输出模式以及坐标的名称和范围, 相应的改变在下次坐标图的绘制中便得到应用。同时, 利用 Gnuplot 可以实现多个数据文件绘制于同一张坐标图中, 这样可以更直观的看出不同因素对结果的影响。

以上便是利用 NS2 进行网络仿真的流程, 关于 NS2 的使用在参考文献[3]、[4]中有详细的介绍, 参考文献[5]以例子的形式对 Otcl 脚本和网络的仿真模型做了详细的介绍, Gawk 脚本处理程序在参考文献[6]、[16]中有详细的讲解, 利用 Gnuplot 实现数据文件的可视化在参考文献[11]中有详细的介绍。

第四章 ZigBee 的 NS2 仿真模型

基于 NS2 的 ZigBee 仿真模型主要由无线场景定义以及众多的原语组成，其中无线场景定义主要是选择路由协议、拓扑结构和网络配置，事件的调度以及程序的开始和结束，这一部分由 OTcl 脚本编写而成，但具体的过程是由 C++来实现的；而原语本质上是 ZigBee 中的各种基本操作，用以实现不同层之间的信息交换，其中介质访问控制子层定义了 35 条原语，是 ZigBee 仿真模型的主体。

4.1 建立仿真模型

ZigBee 的仿真模型主要由队列管理、能量检测、载波侦听、信标使能、应答机制、路由寻址协议等组成，另外还要通过参数设定对网络进行具体配置。

4.1.1 ZigBee 仿真模型的组成

NS2 仿真器内部封装了许多功能模块，最基本的是节点、链路、代理、数据包格式。

- 事件调度器 (Scheduler)：目前 NS2 提供了四种具有不同数据结构的调度器，分别是链表、堆栈、日历表和实时调度器。
- 节点 (Node)：由 TclObject 对象组成的复合组件，在 NS2 中可以表示端节点和路由器。
- 链路 (Link)：由多个组件复合而成，用来连接网络节点。所有的链路都是以队列的形式来管理分组的到达、离开和丢弃。
- 代理 (Agent)：负责网络层分组的产生和接收，也可以用在各个层次的协议实现中。每个代理连接到一个网络节点上，由该节点给它分配一个端口号。
- 数据封包 (Packet)：由头部和数据两部分组成，一般情况下的数据封包只有数据部分。

由以上的功能模块便可以搭建起无线网络的仿真模型，在 ZigBee 仿真模型中，C++函数的调用关系如图 4-1 所示，OTcl 脚本则主要负责下面几部分的定义与初始化：

- 仿真参数配置
- 模型初始化

- 建立仿真器
- 配置拓扑对象
- 设置跟踪文件
- 建立无线信道
- 移动场景配置
- 节点位置与状态定义
- 代理定义
- 应用定义
- 事件调度

4.1.2 ZigBee 仿真模型的工作方式设定

- 信标模式

ZigBee 网络的仿真模型可以工作在信标使能模式与非信标使能模式下，在仿真模型运行前，需要事先设定 ZigBee 网络的工作模式。在 NS2 的 OtcI 脚本语言中，可以使用下面的两句来设定 ZigBee 网络的工作模式：

```
$ns_ at 0.0 "$node_ (0) sscs startPANCoord 1 1 1"
$ns_ at 0.5 "$ node_ (0) sscs startDevice 1 1 1"
```

其中第一句用来设置节点 0 为协调点，工作在信标使能模式，且 txBeacon=1，BO=SO=1。第二句用来设定节点 1 为完全功能设备，且同样有 txBeacon=1，BO=SO=1。如果没有对变量进行设定，程序会自动将其设为默认值。

```
$node sscs startPANCoord <txBeacon = 1> <BO = 3><SO = 3>
```

```
$node sscs startDevice <isFFD = 1> <assoPermit = 1> <txBeacon = 0> <BO = 3> <SO = 3>
```

同时也可以对节点的工作模式进行自定义，例如：

```
$node_ (0) sscs startDevice 0 # 简化功能设备
$node_ (0) sscs startDevice # 完全功能设备，无信标模式
$node_ (0) sscs startDevice 1 1 1 # 完全功能设备，信标使能模式
```

在程序运行过程中也可以用以下语句改变节点的工作模式：

```
$node sscs startBeacon <BO = 3><SO = 3> # 启动信标使能模式
$node sscs stopBeacon # 终止信标使能模式
```

- 能量模型

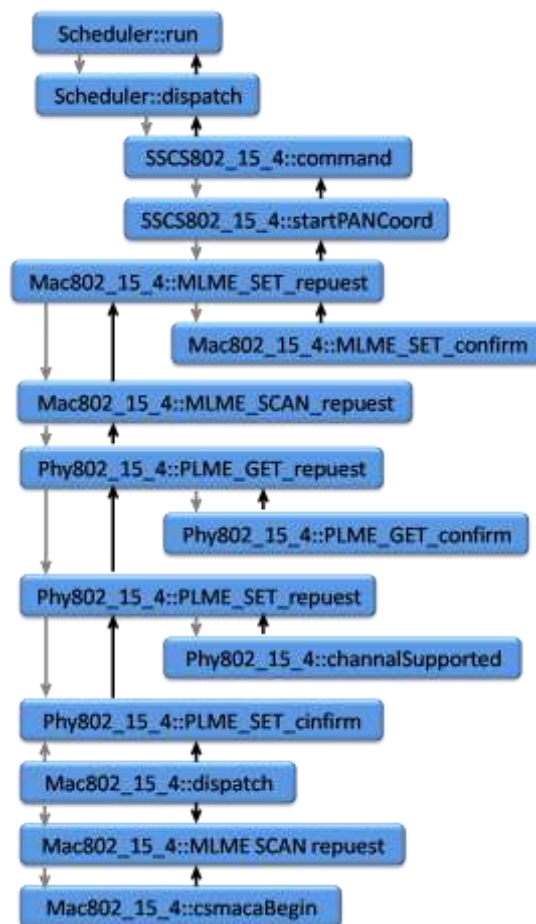


图 4-1 ZigBee 仿真模型的 C++ 构成

能量检测作为信道选择算法中的一部分，主要是用来估算信道有效带宽内的信号强度。NS2 中能量模型的实现很简单，它将能量的大小设定为节点的固有属性，用来显示节点当前的能量水平。节点在发送数据和接受数据时会有能量消耗和时间消耗，相关的函数便会据此降低节点的能量水平。在模拟开始时，设置 `initialEnergy_` 可以实现对节点的能量初始化，同时通过 `txPower_` 和 `rxPower_` 这两个属性分别设定节点发送数据和接受数据时的能量消耗。下面语句用来开启能量模型并设置能量参数：

```
-energyModel "EnergyModel" \           # 开启能量模型
-initialEnergy 1000 \                   # 初始能量值
-rxPower 35.28e-3 \                     # 发送消耗功率
-txPower 31.32e-3 \                     # 接收消耗功率
-idlePower 712e-6 \                     # 空载功率
-sleepPower 144e-9 \                   # 休眠功率
```

● 广播模型

NS2 支持 3 种无线传播模型：FreeSpace、TowRayGround 和 Shadowing 模型，这些传播模型可以帮助移动节点预测接收到的每个分组信号的能量。无线传播模型也是节点的一个属性，可以在节点配置时进行设定。

FreeSpace 模型为每一节点规定了广播半径，其它的节点只要是在这一半径之内都能够正常接收到数据封包，这一模型只适合于短距离的无线传输。接收信号的功率可以由下面的公式得到：

$$Pr(d) = \frac{Pt \times Gt \times Gr \times \lambda^2}{(4 \times \pi)^2 \times d^2 \times L} \quad (4-1)$$

其中， d 为数据传输距离， Pr 为接收端信号强度， Pt 为发送端信号强度， Gt 为发送增益， Gr 为接收增益， λ 为信号波长， L 为路径损耗。

TowRayGround 模型在 FreeSpace 模型的基础之上将地面的反射也可考虑在内，这样在做较远距离无线仿真时能够得到更为精确的结果，接收信号功率如下所示：

$$Pr(d) = \frac{Pt \times Gr \times Gt \times hr^2 \times ht^2}{d^4 \times L} \quad (4-2)$$

其中， hr 为接收天线高度， ht 为发送天线高度。

以上的两种模型在做更远距离数据传输时具有一定的局限性，它们都没有将信号衰减计算在内，而 Shadowing 模型内部建立了一个信号衰减模型，因此对于远距离的无线网络仿真一般采用 Shadowing 模型。

● 数据传输方式

IEEE 802.15.4 定义了三种数据传输方式，TxOptions 选项便是用来设定数据传输方式。若 TxOptions=0x00，则为直接数据传输，介质访问控制子层会直接以载波侦听多路访问的方式传输数据；若 TxOptions=0x02，则为预留时隙数据传输，此时介质访问控制子层会根据时隙而不是载波侦听来传输数据；若 TxOptions=0x04，则为间接数据传输，只有协调点能够使用这种方式传输数据。

4.1.3 ZigBee 仿真模型参数设定

● 数据流参数

数据类型：这里选择固定码率（Constants Bit Rate, CBR）数据流，CBR 具有单程传输、封包大小和传输间隔固定的特点，这样更有利于对网络的传输性能进行分析。

封包大小：封包大小定义为 100bytes，其中不包括信标和其它标志位。

传输方向：所有的数据信息都是单向传输的。

信道选择：仿真中只选用 2.4GHz 频段的前三个信道。

● 节点参数

节点数：仿真环境由 56 个节点组成，其中包括一个协调点。

节点状态：所有的节点都是静止的。

节点位置：所有的节点构成长方体分布，如图 4-2 所示，协调点位于右下角，层距为 3 米。对于同层内的节点距离，左右之间为 10 米，前后之间为 15 米。

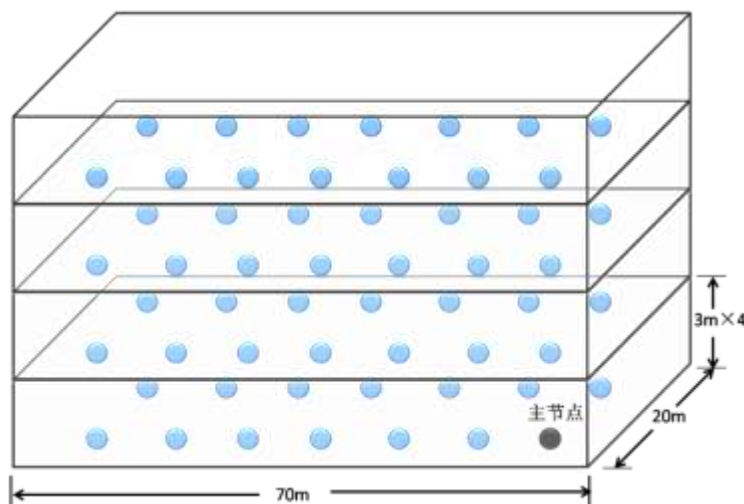


图 4-2 节点的空间分布

● 物理参数

广播模型：由于仿真中节点的分布范围较大，从设备与协调点的位置关系较为

复杂，因此这里采用 Shadowing 模型。

天线类型：这里选用全方位天线（OmniAntenna），它能以相同的功率向四周传送数据。

路径损耗：受距离、障碍物和天线高度的影响，信号的能量会产生衰减，但是在实验中认为衰减为零，即路径损耗为 1.0。

节点距离：各个节点之间的距离分别为 3m、10m 和 15m，如图 4-2 所示。

● 路由层参数

队列类型：仿真中使用的是队尾丢弃算法（DropTail），这是遵循“先进先出”原则的队列管理技术——最先进入队列的封包会被首先处理。当封包数超出队列上限值时，最后进入队列的封包将会被丢弃。

队列长度：仿真中使用的标准队列长度为 150 个封包，一旦超过这个极限，超出部分都将被丢弃。由于网络的性能只取决于资源的有效利用率，单纯的增加队列长度对于网络性能并没有很大影响。

路由协议：仿真中使用的是 AODV 路由协议。

表 4-1 给出了 NS2 仿真模型的整体参数设置。

表 4-1 NS2 仿真模型参数设置

基本参数		扩展参数	
数据流	CBR	广播模型	Shadowing
主节点数/节点数	1/56	天线类型	OmniAntenna
节点运动状态	无	队列类型	DropTail
节点距离	3m/10m/15m	队列长度	150
封包大小	100bytes	发送增益	1.0
频段选择	2.4GHz	接收增益	1.0
信道数	3	路径损耗	1.0
数据传输方式	直接传输	路由协议	AODV

4.2 性能指标

为了衡量无线网络的运行性能，仿真中定义了多个性能指标，通过这些指标可以得到 ZigBee 网络的运行状态及可靠性。需要指出的是，这里提到的性能指标关注的是数据的传输，其他方面并没有做过多的考虑。

4.2.1 传输时延

传输时延 (Propagation Delay) 指的是一个封包从发送到接收所耗费的时间, 传输时延针对的是每一个单独的数据封包, 只要数据包传输成功, 都会存在不同程度的延迟, 这当中包括被重复发送的数据封包。对所有数据封包的传输时延取平均, 即可得到平均传输时延, 可以用来衡量网络的整体传输性能。另外, 传输时延还可以反映介质访问控制子层的容量——延迟时间越长则表明其容量越小。传输时延及平均传输时延可以由下面的公式得到:

$$\text{Packet_delay} = \text{Time_receive} - \text{Time_transmit} \quad (4-3)$$

$$\text{Average_delay} = \frac{\text{Sum_Packet_delay}}{\text{Sum_Packets_receive}} \quad (4-4)$$

其中, Packet_delay 为封包传输时延, Time_receive 为封包接收时间, Time_transmit 为封包发送时间, Average_delay 为平均传输时延, Sum_Packet_delay 为所有封包的传输时延之和, Sum_Packets_receive 为接收到的封包数的总和。

4.2.2 传输抖动率

传输抖动率 (Jitter) 就是传输时延的变化量 (Delay Variance), 由于网络的流量随时都在变化, 当流量大的时候, 许多封包就必须在节点的队列中等待, 因此每个封包从传送端到目的端所耗费的时间也会有所不同, 由此造成的差异就是所谓的抖动率。抖动率反映了网络的稳定性, 抖动率越大则表明网络的稳定性越差。计算抖动率的公式如下:

$$\text{Jitter} = \frac{[\text{receive}(m) - \text{sent}(m)] - [\text{receive}(n) - \text{sent}(n)]}{m - n} \quad (4-5)$$

其中 m 和 n 为封包序号, receive(m), receive(n) 为封包接收时间, sent(m), sent(n) 为封包发送时间。

4.2.3 封包遗失率

封包遗失率 (Packet loss Rate) 是指仿真过程中丢弃数据包数量占发送数据包的比率, 通常情况下封包遗失率与数据包长度以及包发送频率有关。封包遗失是由于网络传输错误或网络拥塞引起的, 如当网络内转发设备接收到错误封包或者是 CRC 校验、帧定位等发生错误时会丢弃该数据封包。与传输时延类似, 封包遗失在计算时也会把重复发送考虑在内, 当一个数据封包发送失败后会等待下一次发送, 如果在仿真过程中发送成功, 则这个封包之前的丢弃并不计算在内。封包遗失

也会像传输时延一样影响网络的实时性能,对于一些面向连接的应用,封包遗失会导致数据重传,从而增加传输时延。封包遗失率过高表明网络设备出现故障或是网络负载过大,但是封包遗失率并不一定表示网络存在故障,很多业务在少量丢包的情况下也能继续进行。封包遗失率的计算公式如下:

$$\text{Packet_loss_rate} = \frac{\text{Packet_drop}}{\text{Packet_sent}} \times 100\% \quad (4-6)$$

4.2.4 吞吐量

网络的吞吐量(Throughput)是指单位时间内传输的数据总量,由于 ZigBee 协议支持的数据传输速率比较低,这里采用单位时间内传输的位数作为吞吐量。同时,这一指标只考虑总的的数据吞吐量,而不去考虑其它的网络开销。仿真中的吞吐量又分为节点吞吐量和平均吞吐量,其中节点吞吐量是这一节点单位时间内接收到的数据量,平均吞吐量是网络中所有节点吞吐量的平均值,计算公式如下所示:

$$\text{Throughput_node} = \frac{\text{Packet_receive_bits}}{\text{Time_end} - \text{Time_start}} \quad (4-7)$$

$$\text{Throughput_ave} = \frac{\sum_{n=0}^N \text{Throughput_node}(n)}{N} \quad (4-8)$$

其中 N 为节点总数。

上述的性能指标的计算是在同一个 Gawk 程序中完成的,各个指标被写入不同的文件中,然后可以由 Gnuplot 将其以坐标图的形式呈现出来。

4.3 仿真结果分析

在仿真过程中,影响网络性能指标的因素主要有数据发送间隔、设备类型和网络的工作方式,这里主要利用 Gawk 和 Gnuplot 对 ZigBee 网络的仿真过程进行分析。

4.3.1 链接建立时间

在 ZigBee 网络的仿真过程中,首先要启动主从节点并建立彼此之间的链接。当网络工作在在无信标使能模式下,ZigBee 网络中的 55 个从节点可以在 15 秒之内与主节点之间建立起有效的链接,如图 4-3 所示。

```
wicnc@wicnc-desktop: ~/ns-allinone-2.34/ns-2.34/tcl/ex/wpan
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
[13.858208](node 45) scanning channel 12
[13.894272](node 54) sending data request command ...
[13.895520](node 54) ack for data request command received
[13.897888](node 54) association response command received
[13.897888](node 54) association successful (non-beacon enabled) [channel:11] [PAN_ID:0] [CoordAddr:8]
[14.091840](node 55) sending data request command ...
[14.093088](node 55) ack for data request command received
[14.095456](node 55) association response command received
[14.095456](node 55) association successful (non-beacon enabled) [channel:11] [PAN_ID:0] [CoordAddr:6]
[14.120288](node 45) scanning channel 13
[14.383968](node 45) sending association request to [channel:11] [PAN_ID:0] [CoordAddr:31] ...
[14.385056](node 45) sending association request command ...
[14.386528](node 45) ack for association request command received
[14.878048](node 45) sending data request command ...
[14.879297](node 45) ack for data request command received
[14.881665](node 45) association response command received
[14.881665](node 45) association successful (non-beacon enabled) [channel:11] [PAN_ID:0] [CoordAddr:31]

Transmitting data ...

NS EXITING...
```

图 4-3 非信标使能模式下设备启动时间

当网络工作在信标使能模式下，由于要进行信标的传输，链接建立的时间会更长。在 $BO=SO=3$ 时，55 个从节点至少要消耗 35 秒才能与主节点建立起可靠链接，如图 4-4 所示。

```
wicnc@wicnc-desktop: ~/zigbee
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
[33.526080](node 55) scanning channel 13
[33.680864](node 54) sending data request command ...
[33.682144](node 54) ack for data request command received
[33.684608](node 54) association response command received
[33.684608](node 54) association successful (beacon enabled) [channel:11] [PAN_ID:0] [CoordAddr:0]
[33.684608](node 54) begin to synchronize with the coordinator
[33.788160](node 55) sending association request to [channel:11] [PAN_ID:0] [CoordAddr:0] ...
[33.789888](node 55) sending association request command ...
[33.791584](node 55) ack for association request command received
[34.283104](node 55) sending data request command ...
[34.284384](node 55) ack for data request command received
[34.286848](node 55) association response command received
[34.286848](node 55) association successful (beacon enabled) [channel:11] [PAN_ID:0] [CoordAddr:0]
[34.286848](node 55) begin to synchronize with the coordinator

Transmitting data ...

NS EXITING...
```

图 4-4 信标使能模式下设备启动时间

4.3.2 传输成功率

这里的传输成功率是指能够成功接收到数据封包的节点数占节点总数的比例。

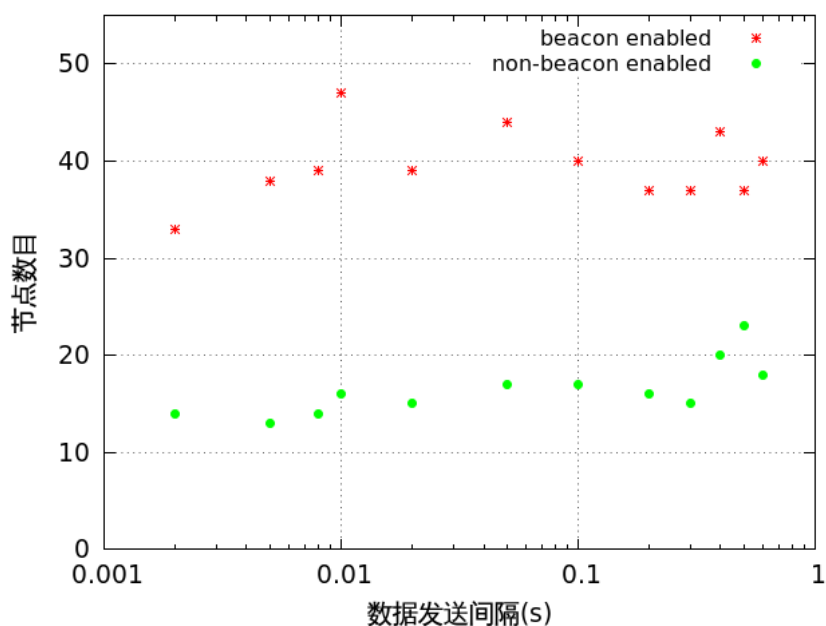


图 4-5 传输成功率

在 ZigBee 网络的仿真场景中，由于节点数目较多且许多节点与协调点之间的距离较远，这样会造成许多节点不能成功接收到数据封包。当 ZigBee 网络工作在信标使能模式下，数据传输的成功率会有明显的提高，如图 4-5 所示。

4.3.3 传输时延与抖动率

在仿真过程中，数据封包由协调点顺次发送到每一个从节点，在数据传送期间，数据封包并不需要在节点队列中进行等待。在这种情况下，传输时延只会受到数据封包大小的影响，数据封包越大则传输时延随之增大，如表 4-2 所示^⑧。

表 4-2 数据封包大小对传输时延的影响

封包大小 (Byte)	50	55	60	65	70	75	80	85	90	95	100
传输时延 (ms)	3.296	3.456	3.616	3.776	3.916	4.096	4.256	4.416	4.576	4.736	4.896

4.3.4 封包遗失率

首先，封包遗失率还会受到数据发送间隔的影响，如表 4-3 所示，当数据发送间隔超过 0.2s 时，封包遗失率明显升高。

表 4-3 数据发送间隔对网络性能的影响

发送间隔 (s)	吞吐量 (kbps)	发送封包数	丢弃封包数	封包遗失率 (%)
0.002	151.014	24813	15	0.0605
0.005	153.602	21768	17	0.0781
0.01	149.078	23408	22	0.0940
0.02	149.710	23214	23	0.0991
0.05	111.979	11964	12	0.1003
0.1	59.576	4455	8	0.1795
0.2	12.974	2055	5	0.2433
0.3	8.917	1367	5	0.3658
0.4	7.402	1142	7	0.6130
0.5	5.966	963	8	0.8308

另一方面，封包遗失率会受到网络的工作模式的影响，如图 4-6 所示，当网络工作在非信标使能模式下，封包遗失率普遍较低，而在信标使能模式下，由于网络要进行信标等额外开销的传输，使得封包遗失率随之上升。当 ZigBee 网络工作在非信标使能模式下，封包遗失率在 0.1% 之下，当 ZigBee 网络工作在信标使能模式下，封包遗失率大为升高，在 0.4% 到 1.4% 之间。这也说明了信标使能模式对 ZigBee 网络的封包遗失率的影响很大，如图 4-6 所示。

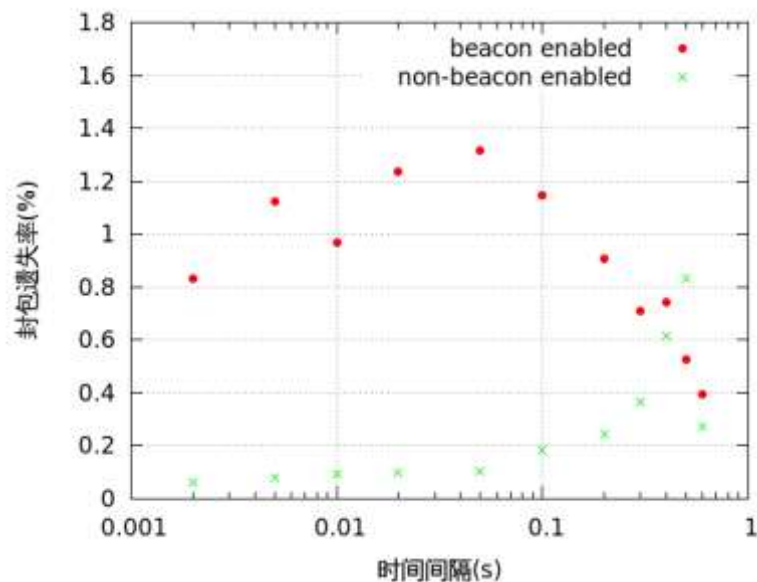


图 4-6 信标模式对封包遗失率的影响

4.3.5 吞吐量

首先，网络的吞吐量会受到数据发送间隔的影响。

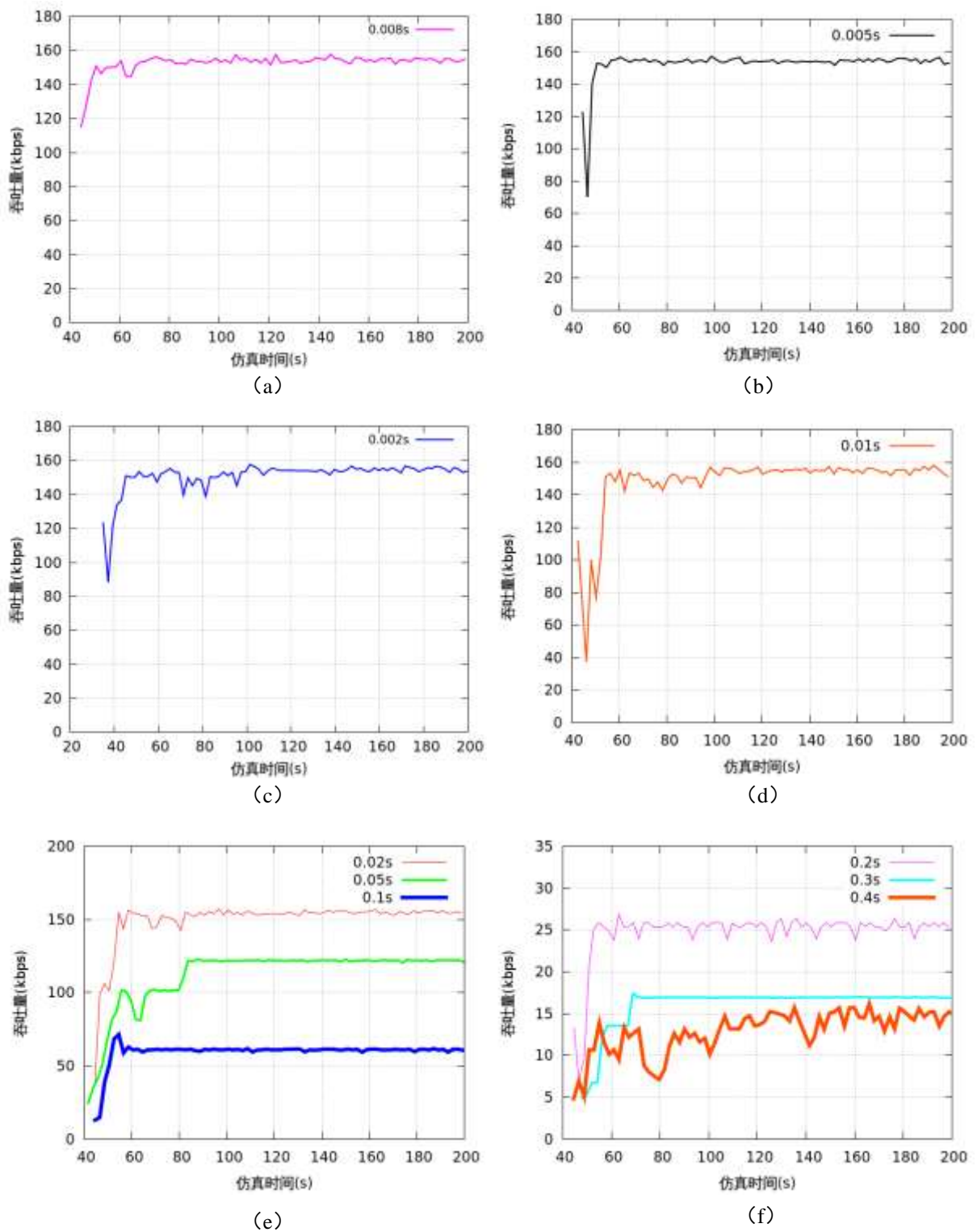


图 4-7 数据发送时间间隔对吞吐量的影响

当 ZigBee 网络工作在非信标使能模式，在数据封包大小一定的情况下，数据发送间隔越小网络的吞吐量就会越大。图 4-7 中是在非信标使能模式下数据发送间

隔对网络吞吐量的影响，此时的数据封包大小为 100Bytes。

从图 4-7 可以看出，随着数据发送间隔的增大，网络的实时吞吐量在不断降低。其中在 0.2s 与 0.4s 时，吞吐量存在不同程度的波动，说明此时网络的状态并不稳定；在 0.002 秒和 0.008 秒时同样存在吞吐量波动的情况，这主要是由数据发送间隔过小导致的。当数据发送间隔为 0.005s 时，网络的吞吐量能够达到较高的水平且波动较小，通过分析可以得出，在数据发送间隔为 0.005s 时，ZigBee 网络的运行状态是稳定且高效的。

另一方面，ZigBee 网络的吞吐量还会受到网络工作模式的影响。当网络工作在信标使能模式下，会增加许多额外的开销，这样就会导致吞吐量的下降，从图 4-8 能够看出这种影响是很明显的。

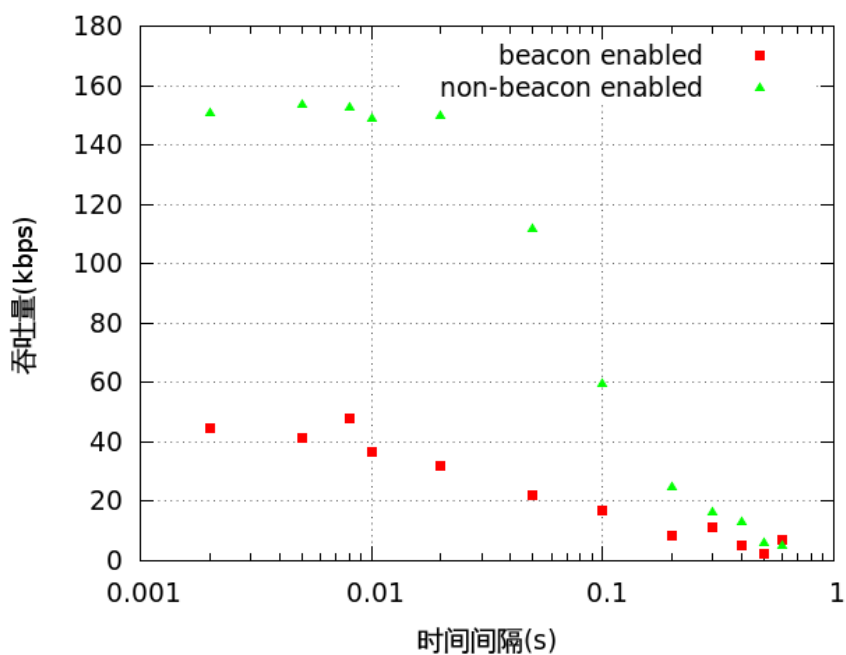


图 4-8 信标模式对网络平均吞吐量的影响

结论与展望

本文以 NS2 网络仿真工具为基础,对 ZigBee 网络的性能指标进行了分析,主要进行了如下的工作:

1) 搭建 ZigBee 网络仿真模型。仿真模型中节点位置呈立体分布,所有节点为完全功能设备,以点对点方式进行数据传输;广播模型为 Shadowing,天线类型为 OmniAntenna,信道选择 2.4GHz 频段内的前三个信道;数据类型为 CBR 数据流,数据发送方式为直接数据传输;网络可以工作在信标使能模式和无信标使能模式。

2) 对原有 ZigBee 仿真模型进行改进。为仿真模型增加静态扫描功能;对 AODV 路由协议进行改进;对模型中节点的睡眠模式进行了改进;修改了模型中的竞争介入机制。

3) 分别分析了数据发送时间间隔和信标模式对 ZigBee 网络性能的影响,其中性能指标主要包括设备启动时间、传输成功率、封包遗失率、传输时延、抖动率以及吞吐量。

通过以上所做的工作,结合仿真模型的参数设定以及得到的各项性能指标,可以得出如下的结论:

1) 采用信标使能模式会提高设备的启动时间和数据传输的封包遗失率,同时能够提高数据传输的成功率,即保证尽可能多的节点成功接收到数据封包;在信标使能模式下,由于增加了额外的开销,又会导致吞吐量的下降。

2) 数据发送间隔对吞吐量有着很大的影响,但当数据发送时间间隔过小时,网络吞吐量的提升并不明显;同时,封包遗失率并不会随着数据发送时间间隔的增大而降低。

由于 NS2 中的 ZigBee 仿真模型并不完善,同时 ZigBee 协议本身也有待改进,因此针对 ZigBee 的研究还需要进一步的研究。在今后的工作中,对于 ZigBee 的研究可以从以下几个方面进行:

1) 对于复杂的 ZigBee 网络,其路由协议并不完善,且在 NS2 仿真模型中并不支持 ZBR 和 Z-AODV 路由协议;

2) NS2 中的广播模型较为简单,对于分布于建筑物内的 ZigBee 网络,FreeSpace、TowRayGround 和 Shadowing 模型并不会将墙面的阻挡考虑在内。

3) 在 ZigBee 网络的仿真模型中,休眠模式与唤醒机制并不完善,使得仿真中的能量消耗较大。

4) ZigBee 仿真模型中的数据发送机制过于简单,可以考虑改进现有的队列管理方式。

参考文献

- [1] IEEE Std 802.15.4-2006. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) [S]. 2006
- [2] ZigBee Alliance. ZigBee Overview [R]. 2009
- [3] The VINT Project. The *ns* Manual [M]. 2010.5
- [4] 柯志亨. NS2教学手册. <http://hpds.ee.ncku.edu.tw/~smallko/ns2/ns2.htm>, 2010.2
- [5] Jae Chung, Mark Claypool. NS by Example. <http://nile.wpi.edu/NS/>, 2010
- [6] Free Software Foundation, Inc. The GNU Awk User's Guide. <http://www.gnu.org/manual/gawk/gawk.html>, 2009
- [7] Dimitri Marandin. Simulation of IEEE802.15.4/ZigBee with Network Simulator-2 (ns-2). <http://www.ifn.et.tu-dresden.de/~marandin/ZigBee/ZigBeeSimulation.html>, 2008.4
- [8] Information Science Institute ISI. Network simulator ns-2. <http://www.isi.edu/nsnam/ns>, 2010.5
- [9] Samsung/CUNY. Ns2 simulator for 802.15.4. <http://www-ee.ccny.cuny.edu/zheng/pub>, 2004.1
- [10] Iyappan Ramachandran. IEEE 802.15.4 MAC implementation in NS-2. http://www.ee.washington.edu/research/funlab/802_15_4/, 2006.2
- [11] Nishanth R. Sastry. Visualize your data with gnuplot. <http://www.ibm.com/developerworks/library/l-gnuplot/>, 2004.8
- [12] J. Zheng, Myung J. Lee. A comprehensive performance study of IEEE 802.15.4 [J]. Wiley Interscience, 4: 218-237, 2006
- [13] J. Zheng, Myung J. Lee. Will IEEE 802.15.4 make ubiquitous networking a reality?-- a discussion on a potential low power, low bit rate standard [J]. *IEEE Communications Magazine*, 42(6): 140-146, 2004.6
- [14] Chandramani Kishore Singh, Anurag Kumar, P. M. Ameer. Performance evaluation of an IEEE 802.15.4 sensor network with a star topology [J]. *Wireless Netw*, 14:543-568, 2008
- [15] Marina Petrova, Janne Riihijärvi, Petri Mähönen, Saverio Labella. Performance Study of IEEE 802.15.4 Using Measurements and Simulations [J]. *WCNC*, 2006
- [16] Iyappan Ramachandran. Analysis of CAP of IEEE 802.15.4 Superframe [R]. 2005.11
- [17] 蒋挺, 赵成林. 紫蜂技术及其应用[M]. 北京: 北京邮电大学出版社, 2006.6
- [18] 孙戈. 短距离无线通信及组网技术[M]. 西安: 西安电子科技大学出版社, 2008.3
- [19] 柯志亨, 程荣祥, 邓德隽. NS2仿真实验——多媒体和无线网络通信[M]. 北京: 电子工业出版社, 2009.3

- [20] 方路平, 刘世华, 陈盼等. NS-2网络模拟基础与应用[M]. 北京: 国防工业出版社, 2008
- [21] 瞿雷, 刘盛德, 胡咸斌. ZigBee技术及应用[M]. 北京: 北京航空航天大学出版社, 2007
- [22] 金纯. ZigBee技术基础及案例分析[M]. 北京: 国防工业出版社, 2008
- [23] 王春艳. IEEE 802.15.4协议仿真与研究[D]. 大连: 大连海事大学硕士学位论文, 2008
- [24] 方梅. 时隙CSMA/CA机制下IEEE 802.15.4网络性能研究[D]. 合肥: 合肥工业大学出版社, 2009
- [25] Chuda Liu. NS Tutorial: mobile and wireless network simulation[R]. 2004.4.10
- [26] Petr Jurcik, Anis Koubaa, Eduardo Tovar等. A Simulation Model for the IEEE 802.15.4 Protocol: Delay/Throughput Evaluation of the GTS Mechanism [J]. 2007.10
- [27] Gang Lu, Bhaskar Krishnamachari, Cauligi S. Raghavendra. Performance Evaluation of the IEEE 802.15.4 MAC for Low-Rate Low-Power Wireless Networks [J]. IEEE Xplore, 2004
- [28] J. Zheng. 802.15.4 and ZigBee Routing Simulation at Samsung/CUNY [R]. 2003.11.14
- [29] 李方敏, 李仁发, 叶澄清. 网络仿真软件ns的结果输出和分析[J]. 计算机工程, 26(9): 14-16, 96, 2000.9
- [30] 姜宁康, 李毓麟. NS网络仿真技术及其应用分析[J]. 小型微型计算机系统, 22(4): 415-417, 2001.4
- [31] 蒋云霞. 基于NS2的IEEE 802.15.4标准的研究[J]. 中国科技信息, 11:112-113, 2008.6
- [32] 耿萌, 于宏毅, 张效义. ZigBee 路由协议分析与性能评估[J]. 计算机工程与应用, 43(26): 116-120, 2007
- [33] 赵铮, 于宏毅. IEEE 802.15.4 MAC 协议分析与仿真评估[J]. 计算机工程与应用, 43(4): 119-121, 2007.1
- [34] 冉鹏. ZigBee网络路由协议性能研究与算法优化[D]. 上海: 同济大学硕士学位论文, 2007
- [35] 胡科路. IEEE 802.15.4 仿真研究与性能分析[D]. 武汉: 华中科技大学硕士学位论文, 2006

注 释

^①无线传感器网络（Wireless Sensor Network, WSN）综合了微电子技术、嵌入式计算技术、现代网络及无线通信技术、分布式信息处理技术等先进技术，能够协同地实时监测、感知和采集网络覆盖区域中各种环境或监测对象的信息，并对其进行处理，处理后的信息通过无线方式发送，并以自组多跳的网络方式传送给观察者。

^②物联网（The Internet of Things, IOT）是一个动态的全球网络基础设施，它具有基于标准和互操作通信协议的自组织能力，其中物理和虚拟的“物”具有身份标识、物理属性、虚拟的特性和智能的接口，并与信息网络无缝整合。物联网将与媒体互联网、服务互联网和企业互联网一道，构成未来互联网。

^③根据国际标准组织发布的开放互联模型，通信网络一般分为七层，即物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。

^④不同层次之间对话的语言称为原语，即不同层之间通过原语来实现信息交换。原语可以分为四类：请求型原语，用于高层向低层请求某种业务；确认型原语，用于证实某个动作已经完成；指示型原语，用于提供业务的层向高层报告一个与特定业务相关的动作；响应型原语，用于应答，表示来自高层的指示原语已收到。

^⑤ZigBee 网络的工作模式由协调点决定，协调点可以决定是否开启信标使能模式，普通的设备不具备这一功能。

^⑥本文中的 ns-allinone-2.34 安装在主文件夹下的 wicnc 文件夹中，由于文件夹的设置不同，在具体的安装过程中需要作相应的修改，将 wicnc 改为实际中的安装目录。

^⑦关联数组和数组类似，由以名称作为键的字段和方法组成，它包含标量数据，可用索引值来单独选择这些数据。和数组不同的是，关联数组的索引值不是非负的整数而是任意的标量，同时关联数组的元素没有特定的顺序。

^⑧表中的数据封包大小并没有将信标等额外开销计算在内，若考虑到额外开销，数据封包大小要增加 15 到 20Byte。