



华南理工大学  
South China University of Technology

# 硕士学位论文

车载环境下媒体接入控制协议优化设计

---

作者姓名	陈钦波
学科专业	电路与系统
指导教师	吴宗泽副教授
所在学院	电子与信息学院
论文提交日期	2014年5月

# **Media Access Control Protocol Optimization in Vehicle Environment**

A Dissertation Submitted for the Degree of Master

**Candidate: Chen Qinbo**  
**Supervisor: Wu Zongze**

South China University of Technology  
Guangzhou, China

分类号：TP393.09  
学号：201120107536

学校代号：10561

华南理工大学硕士学位论文

# 车载环境下媒体接入控制协议优化设计

作者姓名：陈钦波

指导教师姓名、职称：吴宗泽副教授

申请学位级别：工学硕士

学科专业名称：电路与系统

研究方向：嵌入式与 SOC

论文提交日期：2014 年 6 月 5 日

论文答辩日期：2014 年 6 月 1 日

学位授予单位：华南理工大学

学位授予日期：年月日

答辩委员会成员：

主席：周智恒副教授

委员：吴宗泽副教授    傅予力教授    向友君副教授    李波副教授

---

# 华南理工大学

## 学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名：陈钦波

日期：2014年6月7日

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属华南理工大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许学位论文被查阅（除在保密期内的保密论文外）；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。本人电子文档的内容和纸质论文的内容相一致。

本学位论文属于：

☐ 保密，在年解密后适用本授权书。

☒ 不保密，同意在校园网上发布，供校内师生和与学校有共享协议的单位浏览；同意将本人学位论文提交中国学术期刊(光盘版)电子杂志社全文出版和编入 CNKI《中国知识资源总库》，传播学位论文的全部或部分内容。

(请在以上相应方框内打“√”)

作者签名：陈钦波

日期：2014.6.7

指导教师签名：吴伟

日期：2014.6.7

作者联系电话：020-87114702

电子邮箱：584781841@qq.com

联系地址(含邮编)：广州市天河区华南理工大学逸夫科学馆501(510641)

## 摘要

随着互联网技术和交通系统的不断发展，车联网（IOV，Internet of Vehicles）受到越来越多研究机构和厂商的关注和研究，是物联网应用于智能交通系统（ITS，Intelligent Transport System）领域的具体形式。媒体接入控制（MAC，Media Access Control）负责无线终端媒体接入系统和使用无线媒介，现有成熟的媒体接入控制主要是载波侦听多点接入/避免冲撞（CSMA/CA，Carrier Sense Multiple Access with Collision Avoidance）的媒体访问控制机制和时分多址访问技术（TDMA，Time Division Multiple Access）机制。目前存在的接入控制还有很多进一步实现和优化的空间。

首先学习了无线网络相关知识，在调查研究无线网络媒体接入控制技术和车联网相关技术的基础上，阅读和理解专用于车联网 WAVE（Wireless Access in Vehicular Environment）通信协议栈技术和 IEEE1609 系列协议族，分析这些协议的特性。

接着针对车联网通信时间短数量多等特点，IEEE1609 协议族提出的控制信道和服务信道概念，和为了解决单网卡存在的瓶颈，本论文提出了双无线网卡的工作方式，以及使用了 Libpcap 裸套接字来提高性能，实现一个初步的双网卡媒体控制接入系统平台，实现了控制信道和服务信道双信道，并且测试了 Libpcap 和 TCP/UDP 的性能差异。

然后针对任一网卡在抢占和使用信道方面进一步改进，本论文还提出了 TDMA 的理论优化方法，研究了 TDMA 中的时间片映射分配，提出位置辅助随机访问方案，并定义了冲突概率和吞吐量两个性能指标，利用遗传算法在 Matlab 上仿真求解性能，验证了该方案相比普通的随机访问方案有着较好的性能。

双网卡媒体接入控制系统为车联网应用系统提供了一个强有力的支撑系统，取得性能上的优化。而位置辅助随机访问方案性能理论上的进展，为以后车联网系统媒体接入控制进一步优化提供了更多的实现指导和方向。

**关键词：**车联网；媒体接入控制；WAVE；802.11p；CSMA/TDMA；Linux 网络编程

# Abstract

With the progressive development of the Internet technology and the transport system , research institutes and manufacturers have aroused more and more concern and study about the Internet of Vehicles (IOV), the Internet of Vehicles is a subset of today's leading edge of scientific research, also is the concrete format used in Intelligent Transport System (ITS) of the Internet of Things (IOT). Media Access Control controls the access and the usage of the wireless terminal system, currently the mature Media Access Control is CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) and TDMA (Time Division Multiple Access). Nowadays the media access control still has more further optimization.

This thesis firstly studies the wireless knowledge, and also the WAVE (Wireless Access in Vehicular Environment) wireless communication protocol stack technology and IEEE 1609 protocol. The analysis of their characteristics is also presented.

Considering a huge quantity information , a short communication time, the concept of control channel and service channel raised by IEEE 1609 protocol, and the bottlenecks of single NIC, this thesis proposes a dual NIC design scheme, uses the Libpcap raw socket to improve the performance , achieve a dual NIC media access system platform and the concept of control channel and service channel, tests the performance between Libpcap and TCP/UDP.

Then for the further improvement of any NIC's usage of channel, this thesis also propose a TDMA theory optimization, studies the TDMA time slot allocation map of Location-Assisted Random Access Scheme, proposes a Location-Assisted Multiple Access Scheme, defines the collision probability and throughput as two performance indicators, uses genetic algorithms to solve the problem. By the Matlab simulation performance, the scheme is verified to be a better performance compared to normal random access scheme.

Dual NIC media access control system provides a powerful support system for application system, achieves the performance optimization. The theoretical performance of Location-Assisted Multiple Access Scheme provides more further guidance of the media access of the vehicle networking system.

**Keywords:** Internet of Vehicles; WAVE; 802.11p; TDMA; Linux network program



# 主要符号表

IOV	Internet of Vehicles	车联网
ITS	Intelligent Transport System	智能交通系统
MAC	Media Access Control	媒体接入控制
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance	载波侦听多点接入/避免冲撞
TDMA	Time Division Multiple Access	时分多址访问技术
ION	Internet of Net	物联网
GIS	Geographic Information System	地理信息系统
GPS	Global Positioning System	全球定位系统
EDI	Electronic Data Interchange	电子数据交换
ETC	Electronic Toll Collection	电子不停车收费系统
LLC	Logical Link Control	逻辑链路层
OSI	Open System Interconnection	开放式互联系统
WSMP	WAVE Short Message Protocol	WAVE 短消息协议层
WAVE	Wireless Access in Vehicular Environment	
IBSS	Independent Basic Service Set	独立基本服务集合模式
BSS	Basic Service Set	基本服务集合模式
ESS	Extended Service Set	扩展服务集合
NAV	Network Allocation Vector	网络分配矢量
DFC	Distributed Coordination Function	分布式协调功能
PCF	Point Coordination Function	点协调功能
SCH	Service Channel	服务信道
CCH	Control Channel	控制信道
RSU	Road-Side Unit	路侧单元
OBU	On-Board Unit	车载单元

# 目录

摘要 .....	I
ABSTRACT .....	II
主要符号表 .....	III
目录 .....	IV
第一章绪论 .....	1
1.1 研究背景及意义 .....	1
1.2 国内外研究现状 .....	2
1.3 选题的目的及意义 .....	6
1.4 论文主要研究内容及结构安排 .....	7
第二章车联网与 WAVE 技术 .....	8
2.1 车联网相关技术概述 .....	8
2.2 IEEE802.11p .....	9
2.2.1 IEEE802.11 协议 .....	9
2.2.2 IEEE802.11p .....	14
2.3 IEEE1609 协议栈 .....	16
2.4 Linux 相关编程 .....	18
2.4.1 Linux 的特点 .....	20
2.4.2 Linux 体系结构 .....	21
2.4.3 Linux 设备驱动程序 .....	22
2.5 本章小结 .....	23
第三章基于双网卡的媒体接入控制优化 .....	24
3.1 方案描述 .....	24
3.2 方案实现 .....	27
3.2.1 libpcap 移植 .....	27
3.2.2 libpcap 抓包收发示例 .....	28
3.2.3 双网卡平台系统 .....	30
3.2.4 性能测试 .....	33



3.2.5 改进空间.....	36
3.3 本章小结 .....	36
第四章基于 TDMA 的媒体接入控制设计 .....	37
4.1 TDMA 概述 .....	37
4.2 位置信息辅助的多路访问方案 .....	39
4.3 冲突概率和吞吐量 .....	41
4.3.1 冲突概率.....	41
4.3.2 吞吐量 .....	44
4.4 建模求解 .....	44
4.5 仿真结果 .....	47
4.6 本章小结 .....	50
总结与展望 .....	51
参考文献.....	52
攻读硕士学位期间取得的研究成果.....	55
致谢 .....	56

# 第一章 绪论

## 1.1 研究背景及意义

当今随着世界取得不断进步，经济取得可喜发展，科技创新取得惊人提高，信息技术已经融入到生活的方方面面。在生活衣食住行以及工厂制造生产中协调融入信息科学技术，可以提升人们各种生活活动质量，提高生产制造效益<sup>[1]</sup>。如果世界上的所有物体都纳入同一个信息网络中，那将形成一个比互联网更加庞大的网络，其对人类社会的发展将是革命性的进步，于是物联网<sup>[2]</sup>（Internet of Net）的概念诞生了。物联网指的是将一切和人类相关的生活科技生产教育交通等相关有形无形的物体，有效利用现有各种通讯网络，有线抑或无线，短距离抑或长距离，纳入同一个更广泛的信息网络互联，基于云计算云平台等，提供安全可靠实时的信息服务和信息管理，协助人们远程控制、调度指挥等管理和服务功能，实现对世界上一切物体的高效、安全的管理控制和经营。

物联网是世界发展的一种趋势，这一趋势将会带来一个超大规模的新兴科学信息技术产业。针对物联网趋势所能带来的巨大效益，世界上多个企业、地区和国家都规划了相关的物联网科学技术长期企业和国家发展战略战术，我国也紧跟时代潮流大力研究物联网技术，开拓物联网及市场<sup>[3][4]</sup>。IBM 首先提出了“智慧地球、物联网和云计算”<sup>[5]</sup>，前几年国家领导提出“感知中国”战略规划。这个长远战略规划中已经明确了物联网新兴科学技术研究及相关制造生产产业为我国的重点发展战略。物联网相关制造生产及新兴科学技术研究在我国先科技经济发展阶段迎来大力展的曙光<sup>[6]</sup>。交通运输是现代社会发展的重要基石，也是带动国民经济的重要动力，当今世界大规模的城市化，各国经济越来越融合在一起，国民经济正在高速增长，人们对高效国家交通的需求在不断地提高，现代交通系统越来越多地牵涉到不同的领域，现有传统的交通没有得到足够的增长，现有传统的交通管理方式弊端日益显现。如何有效融合科学信息技术，全方位地进行交通信息采集，实时地进行传输，快速地进行信息处理，准确地进行发布交通信息和高效安全地控制相关交通，进而有效保障各个地区的交通安全，研发这样高效实时智能的智能交通系统（ITS，Intelligent Transport System）<sup>[7]</sup>成为目前研究的重点内容，也是提高交通现代化智能化管理的重要路径。智能交通系统是一种面向现代面向未来交通运输的现代电子科学技术信息系统，它有机结合汽车路线、汽车驾驶员、汽车等各种信息，科

学有效地利用主要对上述三者信息实时实地收集上报,快速计算决策以及高效安全智能控制,从而进一步提高交通系统高效的配备和利用<sup>[8]</sup>。智能交通系统是由以下几个部分有机融合成一体:车辆智能传感系统,车辆智能嵌入式控制系统,路边智能接入控制网络设备,车辆智能跟踪系统,车辆智能识别系统,智能运营车辆管理系统,中央智能调度调节系统,车辆智能信息服务发布分发系统等。

车联网(IVN, Internet of Vehicles)是当今研究前沿 IOT 的一个集合,是智能交通系统的未来发展方向,是智能交通系统领域在物联网范畴内的一种扩大化形式。车联网是指车辆快速移动的场景下,车载终端节点通过汽车上的无线电信号收发装置,以及路边固定终端节点的无线收发装置,通过通信协议组合构建连接起来形成的一个车用互联网<sup>[9]</sup>。车联网涉及到许多科学信息技术,比如大量数据实时处理技术,高频无线收发技术,高效通讯通信技术,智能自动控制技术以及新一代智能信息发布技术等。车联网旨在提供一个车-车、车-路高速的数据通信方式,实时准确智能地整合车载相关数据信息,解决城市交通拥堵问题,确保行车安全,同时提供其他的商务娱乐应用服务,最终实现智能交通、智能城市管理和智能城市信息服务<sup>[10]</sup>。世界上现阶段车联网大范围使用的公共移动通信网络是 3G 网络。3G 网络有着较好的网络覆盖范围和数据通信速率,但其接入网络和通信方法的限制,仍无法满足网络拓扑快速变化的车载环境。因此车联网协议栈<sup>[11]</sup>(WAVE, Wireless Access in Vehicular Environment)应运而生,由 IEEE 制定,专门用于 ITS 中基于无线的通信应用<sup>[12]</sup>。WAVE 和 3G 技术的主要参数对比如表 1-1 所示。

基于 WAVE 技术的应用,具体可应用到交通拥堵治理、交通公共安全、信息服务相关领域等。其中交通拥堵治理和交通安全是车联网中最基本最重要的应用,因此 WAVE 协议栈中开辟了专用通信信道用于这些方面的应用。

## 1.2 国内外研究现状

车联网技术目前是世界各国各地区智能交通系统发展研究中的重要部分,目的在于确保交通运输系统的安全性,提高交通运输系统的效率,这对国家地区具有十分重要的意义。车联网技术,其核心就是通过把现代通信技术融入到传统的道路交通系统中,实现将公路、铁路、车辆、乘客以及路边的设施完全联动,并通过公路、铁路、车辆、乘客以及路边的设施之间实施准确,频繁的信息交互,达到最大限度的优化公共交通资源,

为公共交通系统的使用者和监控者提供智能化的服务的系统。世界上主要的研究国家和地区中主要以美国、欧洲和日本为主，它们经历了不同的发展阶段。从研究准备阶段，到可行性研究阶段，再到产品开发为主的阶段。这三部分国家和地区都在制定各自的 DSRC<sup>[13]</sup>标准。目前，国际上主要由欧洲的 ERITCO、美国的 ITS America 以及日本的 VERTIS<sup>[14]</sup>等机构在积极的促进智能交通系统向民用化商用化演变。

项次	参数名称	WAVE	WCDMA	TD-SCDMA	CDMA2000
1	移动性	≤120km/h	较低		
2	双工方式	频分双工、时分双工	频分双工、时分双工	频分双工	频分双工、时分双工
3	调制方式	二进制相移键控、四相相移键控信号、16 正交幅度调制、64Q 正交幅度调制	四相相移键控信号、二进制相移键控	四相相移键控信号、二进制相移键控	四相相移键控信号、二进制相移键控
4	频段	5.8GHz	900MHz, 1.8GHz		
5	复用技术	正交频分复用	码分多址	码分多址	码分多址
6	传输速率	3~27Mbps	≤3Mbps	≤1.28Mbps	≤3.6864Mbps

表 1-1 WAVE 与 3G 主要参数比较表格

美国是世界上最早进行 ITS 研究的国家。在上个世纪，美国的电子化路线导航引导系统计划已经在部署之中。在美国多个地方研究智能交通中，主要是国家自动公路系统会社推行的 AHS 项目。这项目主要研究车-车以及车-路之间的安全性，实时监控数据，致力于提高美国交通安全，确保行车机动性。美国不仅研究车-车和车-路这两方面，还将公共汽车、地铁等交通工具和其他的配套的设施能纳入到系统中，形成一个更庞大的

管理系统,达到智能管理交通和改善环境的目的<sup>[15]</sup>。日本从 20 世纪 80 年代末,就开始进行智能交通系统的研发。日本的主要研究项目是 Advanced Road Transportation Systems (ARTS)、Vehicle Information and Communication System (VICS) 和 Advanced Safety Vehicle (AVS)。日本早期主要研究集中于安全领域,近期则注重环保安全绿色。早期的 VICS 对车辆安全技术进行研究,并形成相应的产品线和产品生态系统。现阶段则研究车路协同,并开发相关实际应用,打造一个完善的一体化信息网络。欧洲一体化组织欧盟的研究主要分为两大类。第一类是政府引导,这一类是多个地区的政府合作共同研究的项目。而第二类是民间组织,这一类是民间各种厂家联合制订研发。早期欧洲智能交通技术发展不够完善不够成熟,后续出现了许多计划,如 DRIVE 计划、PROMETHEUS 计划和 ERTICO 等。现阶段主要是研究智能交通系统的互操作性和兼容性。除了上述两类,欧洲各国自己也开展相关研究项目,研究车路协同通信,以适合各国自己的交通运输国情。

对比上述国家和地区,我国智能交通管理系统发展起步较晚,发展水平较低,相关产业发展较小。直到 90 年代后相关的研究部门才开始研究 ITS 发展战略和地理信息系统(GIS, Geographic Information System)、全球定位系统(GPS, Global Positioning System)、电子数据交换(EDI, Electronic Data Interchange)等在交通管理中的应用<sup>[16]</sup>。同时也才开始交通信息网络的建设,进行铁道部铁路运输管理系统(TMIS, Transportation Management Information System)、调度指挥管理系统(DMIS, Dispatch Management Information System)等信息管理系统开发。与上述国家和地区相比,我国目前现在的车联网的发展空间还非常大。随着计算机科学、信息科学技术、传感通信科学技术和电子科学技术等技术的快速发展,我国从二十世纪 80 年代加大力度进行智能交通系统的技术研究。近几年物联网的快速发展,也推动了我国车联网的发展。经过长时间国家和企业的发展,我国车联网有了可观的完善,取得了明显的发展。不足的是,我国智能交通系统中现阶段所使用的技术仅能满足一些有限的应用,如 ETC (Electronic Toll Collection, 电子不停车收费系统)等的要求。

虽然我国车联网取得一定的进展和成就,但还不够完善,还有很多空间和工作要做。我们不能满足于现状,满足于目前应用。未来交通里边,大容量交通信息采集监控,实时传输,实时智能导航,实时智能控制,中央化智能化调控等复杂高新需求,需要各个

研究机构、高校、厂家继续努力，继续完善，需求带动研究，研究促进进步，进步带来需求等良性循环。在智能交通系统这一研究领域中，我国还有很多东西要做，还有很多难点要突破，还有很多技术要规模化平民化。我们绝不能安于不停车电子收费系统、门禁这些常规的应用，这些应用在国外早已普及和平民化了。对于一些针对未来长远的高级需求，比如更详细的车辆传感监控，超大规模的城市地区交通信息收集，高效不失真快速上传，大数据云服务化的智能交通信息模式识别和决策，高效多地区协调的路线导航，需要国家、地区的研究机构和厂家更多地努力。

因此我们智能交通系统的发展面临巨大的挑战和机遇：

- 我们经济发展较快，高速公路发展迅速，规模不断扩大，纳构不断复杂，运营管理难度不断增加，使得对高效稳定的智能交通管理手段的需求不断增大。
- 我国城镇化速度加快，城市面积不断增大，城市交通状况日益恶化，交通拥堵交通事故频发，交通污染严重，使得城市必须借助智能交通监管手段来改变这样的现状和趋势。
- 不断提高的人们生活水平，不断扩大的工作需要，使得人们出行更加频繁。人们已不再满足汽车仅仅是简单的代步工具，更希望它是安全的、舒适的、高效的和环保的，并带有娱乐、工作、辅助驾驶等功能，因而对定位导航、辅助驾驶、车载流媒体设备等车载电子设备提出了更高要求和迫切需求。
- 我国智能交通的发展起步较晚，尚处在初级阶段，但发展较快，已经建设了很多系统并投入使用，如城市电子警察系统、城市红绿灯信号控制系统和城市交通状态检测系统、城市交通信息实时发布系统、电子交通警察系统、电子不停车收费系统、高速公路紧急救援系统、高速公路智能运管系统、车载定位导航系统、车辆辅助驾驶系统等等。而以上系统相互独立，不能信息共享，需进行系统集成优化，做到信息资源共享。
- 我国交通状态复杂，行人、非机动车、机动车视行现象严重，导致交通违章行为和交通事故额繁发生，因此需要对行人和车辆给予足够的安全信息支持以保障其安全。

以上的挑战和机遇中，最先要解决的是车载环境下媒体的接入及控制。目前基于传统的无线网络媒体接入控制中，暂时没有专门针对车载环境下汽车高速移动的情况。基

于以上原因,本文以车载环境下媒体接入控制作为论文研究对象,非常吻合当前国家经济发展方向,吻合社会发展的方向,吻合科技发展的方向,具有现实应用意义。

车联网刚刚起步,还有许多挑战和问题,主要是 **RFID** 射频识别技术、传感技术、无线传输技术、云计算技术、车联网安全体系和定位技术等技术。以下是车联网在研究中必须重视的困难:

(1)信息要求的异构性:车联网系统的研发规划中,必须两种不同类型应用的差别。第一类道路安全紧急信息应特点是:低延时、高可靠性。第二类相服务娱乐数据信息则允许一定的丢包率,允许重传,其保障的是资源使用合理性,并没有要求很高的实时性。信息要求的异构性,要求这个智能交通系统中媒体接入控制能区分不同优先级,达到自适应,网络资源分配策略要公平,达到两者都兼顾。<sup>[3]</sup>

(2)高度动态时空变化:不同地区的汽车密度不同,农村地区的车辆密度较小,而城市地区的汽车密度较大。不同时间,同一地点的汽车密度也变化很大,上下班高峰期密度较大,而凌晨时汽车少之又少。车联网系统在规划部署时,必须考虑到这种高度变化的时空变化,高密度时期的通信是车联网系统的难点之一。<sup>[3]</sup>

(3)频繁的链路断开和重连:汽车在一些道路上行驶时,速度往往较高,导致停留在一个路边设施基站无线电有效覆盖范围的时间很短,汽车在不断地切换基站。汽车网络拓扑结构时时刻刻面临着变化,车-车以及车-路之间的通信连接也在时时刻刻重复着连接、断开、重连等。<sup>[3]</sup>

### 1.3 选题的目的及意义

目前国内智能交通和车联网发展还有很大空间,也未形成一个完善的车联网产业链。虽然国内有部分高校和厂商在做这方面的研究,但投入到市场的产品有限,尚未规模化。但我们不能因为落后就不发展,相反应该迎难而上,大力研究和发展车联网。一方面社会的发展,必然会向智能交通、智能城市发展,另一方面民族崛起国家科技水平提升,不允许我国在车联网方面落后于国家其他国家和地区。总之,研究和发展车联网,有助于改善人民生活水平,促进社会发展,提高国家综合实力。

目前尚未有成熟的 **WAVE** 解决方案,因此我们这里在调查研究无线网络媒体接入控制技术和车联网相关技术的基础上,研究 **WAVE** 协议和 **IEEE802.11p** 协议,并在 **Linux**



操作系统编程和 Linux 内核网络子系统的基础上，借助 Linux 开放的网络协议栈接口，实现一个初步的双网卡媒体控制接入系统平台，具有非常重要的意义。除了研究 CSMA/CA，这里还对 TDMA 的分配算法进行相关研究，具有前瞻性。

## 1.4 论文主要研究内容及结构安排

本课题的主要工作是研究车载环境下的媒体接入控制，包括 WAVE 协议栈、双网卡工作方式以及 TDMA 时间片分配的相关研究。本文主要包括以下几部分，具体安排如下：

第一章：论文绪论，主要是阐述论文的研究背景及意义、国内外车联网系统研究现状以及本次选题的研究目的和意义。

第二章：车联网相关技术和研究，首先是 WAVE 协议和 IEEE1609 协议族研究，包括 WAVE 协议和 IEEE1609 协议族架构、主要协议内容，然后讲述了 Linux 操作系统编程的相关内容。

第三章：主要按照 IEEE1609 协议族中的多信道协调机制的特点，提出了双无线网卡工作方式的媒体控制接入系统平台，并且移植 Libpcap 库来提供一个更高效的性能，最后测试了 Libpcap 和 TCP/UDP 的性能差异。

第四章：为了进一步优化任一网卡在抢占和使用信道效率，研究 TDMA 时间片分配算法，定义相关指标，通过 matlab 仿真，验证了该算法的改进型。

第四部分：对全文进行总结及展望

第五部分：参考文献。

## 第二章 车联网与 WAVE 技术

### 2.1 车联网相关技术概述

IEEE（美国电气和电子工程师协会）机构已经专门制定的协议标准 WAVE（Wireless Access in Vehicular Environments）是由 DSRC（Dedicated Short Range Communication）协议栈发展而来的，专门用于车联网中车-车、车-路的无线通信应用<sup>[17]</sup>。上世纪 90 年代不同国家地区根据各自情况制定不同的 DSRC 标准。最早可见的 DSRC 标准是由美国材料与试验协会发布的。日本的 DSRC 标准由日本 TC204 委员会制定。欧洲的 DSRC 标准则是由欧洲标准化委员会发布。中国则是参考了欧洲地区而发布了自己的标准。但是，这些标准均有待完善，许多问题尚未解决，比如数据传输速率有待提高、路边无线终端设备覆盖范围有待加宽，需要进一步与互联网融合，因而应用有限，比如不停车收费（ETC）等<sup>[18]</sup>。因此，美国 IEEE（美国电气和电子工程师协会）在 DSRC 标准基础上制定了 WAVE 协议（也即 IEEE802.11p 协议）。IEEE802.11p 这个通讯协定主要用在车用电子的无线通讯上。它设计上是从 IEEE802.11 扩充延伸而来，来符合智慧型运输系统 ITS 的相关应用。

在 IEEE802.11p 协议基础上，IEEE 又制定 IEEE1609 系列标准。IEEE1609 协议系列是一系列的接口和系统架构，专门针对车载通信环境中的无线电通信科学技术。这个系列标准旨在于将车辆-车辆之间和车辆-路侧之间的通信规范化标准化。并且这个系列还提供了一个通信标准，满足于智能交通系统里边的汽车路线导航、汽车行驶安全、电子自动收费等广泛需求。这个系列标准的制定，还考虑到其他许多方面的需求，比如汽车生产制造商、汽车信息服务提供商、交通主管部门等<sup>[19]</sup>。

IEEE1609 系列协议族的体系如图 2-1 所示：

从横向来看，整个框架分为两部分，管理平面 Management Plane 和数据平面 Data Plane。管理平面中包含管理信息库 MIB，以及各种管理实体。管理信息指车载网中的网管框架中被管对象的集合。被管对象维护了可供管理程序读写的若干控制和状态信息。管理信息库代表某个车载终端设备或服务的一套可管理对象。数据平面是参考开放式互联系统（Open System Interconnection, OSI）<sup>[20]</sup>七层网络模型来定义。最底层物理层 PHY 以及数据链路层中介质访问控制 MAC 子层下半部分由 IEEE802.11p 协议定义。数据链

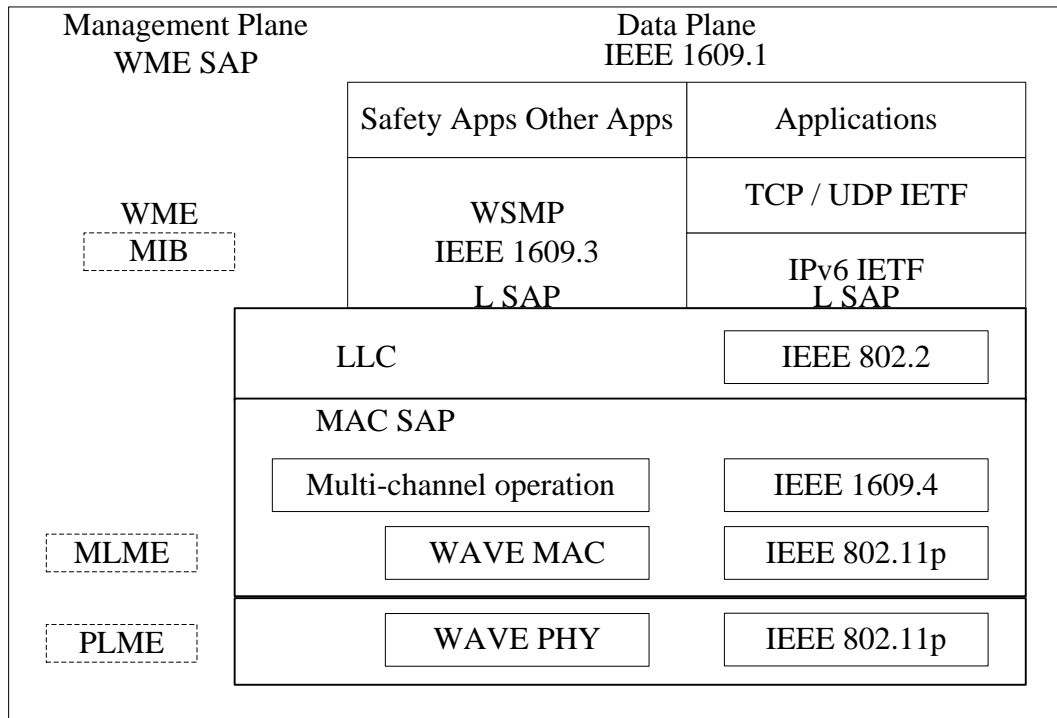


图 2-1 IEEE1609 协议族框架

路层中介质访问控制 MAC 子层上半部分由 IEEE1609.4 协议定义，主要是定义了多信道协作。数据链路层中逻辑链路层（Logical Link Control, LLC）仍由 IEEE802.2 定义。再往上分为两种方式，一种是使用 TCPv6/IPv6。另一种是由 IEEE1609.3 标准所规定的 WAVE 短消息协议层（WAVE Short Message Protocol, WSMP），与传统的 TCP/IP 不同。安全应用和其他服务应用建立在它们的服务之上。最上面的 IEEE1609.1 则定义了 WAVE 的资源管理（Resource Manager）。除了上面这个图之外，还有一个没有出现在图中的 IEEE1609.2 协议，主要是定义了应用和管理消息的安全服务。

## 2.2 IEEE802.11p

IEEE802.11p 协议是在 IEEE802.11<sup>[21]</sup>协议基础上针对车联网提出的修订。

### 2.2.1 IEEE802.11 协议

IEEE 工作组制定了很多常见的协议，其中 IEEE802 规范定义了数据链路层，即网卡 NIC 如何访问传输介质如双绞线、同轴电缆、红外线、射频无线电等，同时还定义了在这些传输介质上是如何传输数据信息的方法。传输数据信息的网络设备之间是如何连接建立的，它们之间是怎样维护的，最后又是怎样拆除的，这些也是 IEEE802 规范定义的。遵循 IEEE802 标准的产品包括网卡 NIC、集线器、交换机、路由器等用来组建局域

网的设备<sup>[22]</sup>。IEEE802 下有很多标准和组，从 IEEE802.1 到 IEEE802.23，每个组专门负责一块领域，如 IEEE802.1 负责高层局域网协议，IEEE802.3 负责常见的 ether LAN，IEEE802.11 负责 wirelessLAN 等。

其中 IEEE802.11 专门定义了无线局域网访问控制方法与物理层规范，这也是一个系列，包含了不同的协议，如表 2-1 所示：

协议	发布时间	频率频带	有效传输距离	最大传输速度
802.11	1997 年	2400-2500MHz	0.1 km	2Mbit/s
802.11a	1999 年	5150-5350/5470-5725/5725-5875MHz	0.075 km	54Mbit/s
802.11b	1999 年	2400-2500MHz	0.1 km	11Mbit/s
802.11g	2003 年	2400-2500MHz	0.15 km	54Mbit/s
802.11n	2009 年	2400-2500MHz	0.15 km	600Mbit/s
802.11p	2010 年	5850-5925MHz	1 <sup>[23]</sup> km	27Mbit/s

表 2-1 IEEE802.11 协议族主要通信协议列表

IEEE802.11 无线协议同其他的标准如 IEEE802.3 以太网、IEEE802.4 令牌总线一样，重点规定了 ISO 七层协议框架中的最底两层，物理层 PHY 和数据链路层 LNK。由于 IEEE802.2 逻辑链路控制层 LLC 的存在，使得所有的上层协议上层应用完全感知不到下层的区别，所以，正是 LLC 使得局域网的任何上层应用程序都能运行在 IEEE802.11 上，感觉不到是以太网还是无线局域网。

IEEE802.11 协议定义了两类通信设备：工作基站（Access Point，AP）和无线工作站（Station，STA）。<sup>[21]</sup>AP 一般由一个无线口和有线口构成，一方面作为网络接入中心控制点，负责控制无线网络中无线设备终端节点接入和通讯；另一方面负责将无线网络与有线网络连接起来，将无线网络中的 STA 联通到有线网络，从而接入到 Internet。STA

一般指的是无线终端设备，如手机、笔记本电脑等，一般是接入到 AP 网络接入中心控制点。IEEE802.11 协议定义了两种工作模式：独立基本服务集合模式（Independent Basic Service Set, IBSS）以及基本服务集合模式（Basic Service Set, BSS）。BSS 中，至少存在一个中心控制节点 AP，负责中心控制、调度和认证等，其他的无线终端节点 STA 均连到这个 AP 中，即使两台 STA 之间要通信，也是通过 AP 来控制转发。两个或者两个以上的 BSS 可构成扩展服务集合（Extended Service Set, ESS），比如候机场、图书馆等，用户看到是单独的一个 BSSID，而实际背后是由多个 AP 互联组成的。IBSS 中，不存在中心控制节点 AP，仅由无线终端节点组成。IBSS 也即是 ad hoc。IBSS 中，一般根据某种算法来选择 STA 作为中心控制节点。IBSS 主要目的是用户自发地形成一个无线局域网，用于分享文件等，这种网络一般都是临时性的。

在有线网络中比如以太网 token 网中，由于需要通过实体物理线路接入到中心控制设备的接口，才能加入相应的网络，而且中心控制设备一般集中存放在机房中，所以要控制终端设备是否要接入网络中，操作上相对简单，铺设线路到允许接入的地方。对于终端设备来说，它的一个网口在同一时刻只能接入到一个网络中，要切换网络，必须切换实体物理线路。而无线网络中，它的媒介是无线电磁波，并不像实体物理线路那样具有定向性和可控制性。如果不加任何控制，那么处于无线覆盖范围内的任意一个无线终端节点均能接入到无线网络中。同时，对于无线终端设备中，它往往能监听到多个无线网络存在。所以需要在软件层次上来控制无线网络的接入。在 BSS 和 IBSS 中，STA 加入无线网络均得经过关联 Association 和认证 Authentication 过程。

无线设备在联网过程中，一般中心节点 AP 会周期性地广播它的 BSSID，所以 STA 最开始先扫描可以使用的网络。如果中心节点不广播它的 BSSID，那这种方式下使用者必须通过其他方式来获得 BSSID，比如人与人口头之间交流，这种方式更加限制陌生用户的使用。然后 STA 针对某个无线网络进行关联，关联即为 STA 绑定到这个无线网络上来，忽视其他的无线网络，未关联的 STA，相当于有线网络中线路上没有连接到中心控制设备上。关联之后，STA 就开始在针对这个无线网络收发数据。对于开放使用的网络，对用户没有限制，这时 STA 无需认证过程就可以在该无线网络中自由使用了。对于有限制的网络，还需要认证过程，限制符合身份的 STA 使用无线网络，由于无线媒介对所有 STA 可见，所有 STA 都能收到无线网络中的数据报，所以认证过程不传输明文，通常是传输经过加解密的认证数据，比如 WEP Open System、WEP Share Key、

WPA-PSK、WPA2-PSK、WPA-EAP 和 WPA2-EAP 等。还可以更高层次的认证过程，比如限定 MAC，网页认证等，这些不属于强制内容，不在这里讨论。

无线网络中，信号可能有意或无意被干扰。一个终端节点发送数据之后，有可能这个数据会丢失，对方可能接收不到。因此大部分传输需要采用肯定确认机制（Positive acknowledge of data transmissions）。这种机制要求发送一个 Frame 给对方时，对方必须回应一个 Ack 确认帧给源节点。源节点只有在收到了 Ack 帧才能算该 Frame 发送成功。否则，需要重发该帧或者重发次数过多而丢弃该帧。如图 2-2 所示：

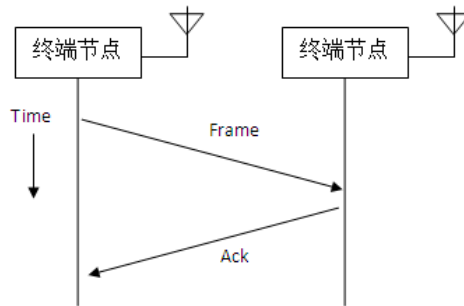


图 2-2 肯定确认机制

STA 在加入网络后，一般不止一个 STA。在网络通信过程中，这些 STA 使用同一个媒介同一信道。无论有线还是无线，同一时刻不能有两个或两个以上同时利用媒介发送数据，否则信号会互相干扰。STA 需要先监听信道是否正在占用，是否忙，忙就退避一段时间。IEEE802.11 中使用了物理监听和虚拟监听。物理监听就是监听当前无线电媒介是否忙。忙则自身不能发送数据。虚拟监听是一种定时器，网络分配矢量（Network Allocation Vector, NAV）。如图 2-3 所示：

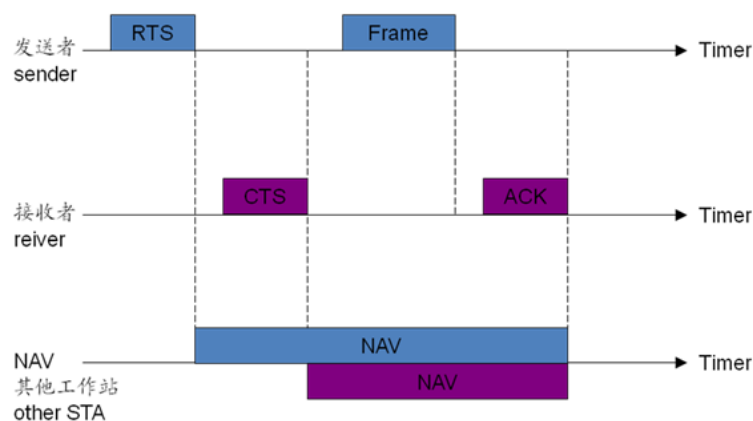


图 2-3 网络分配矢量 NAV

如图所示，发送者在发送 RTS、Frame 帧时，均有相应帧回应，所以其他的 STA 中标记信道忙的时间应该直到 ACK 帧发送结束。这时从逻辑上设定的虚拟监听。

无线网络中有两个问题值得注意，一个是隐藏节点（hidden nodes），一个是暴露节点（exposed nodes）。

隐藏节点是指在一个这样子的无线终端节点，处于发送者的有效无线范围外，又处于接收者的有效无线范围内。该节点因为监测不到发送节点的无线电信号，导致和发送节点同时向目标接收节点发送报文，这样子在接受节点出会造成冲突。如图 2-4 所示，节点 A 能感知节点 B 的存在，节点 B 能感知到节点 A、C 的存在，节点 C 能感知到节点 B 的存在。但 A、C 两个节点无法感知到对方存在。如果节点 A、C 同时向节点 B 发送数据，那么节点 B 收到的是两个信号的混扰，这个过程是错误的。解决这个问题方法之一是使用请求发送/允许发送（Request To Send/Clear To Send，RTS/CTS）。节点 A 或者节点 C 必须先向 B 节点发送一个 RTS 帧，等待 B 发送 CTS 帧之后才能节点 A 或者节点 C 才能发送数据。因为节点 B 发送的 CTS 都能被节点 A 和 C 收到，所以节点 A 或者 C 是在告诉对方自己有数据要发送。如图 2-5 所示：

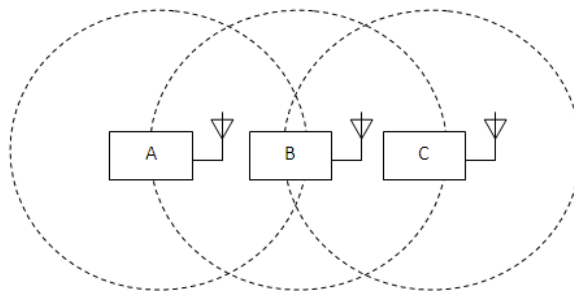


图 2-4 隐藏节点

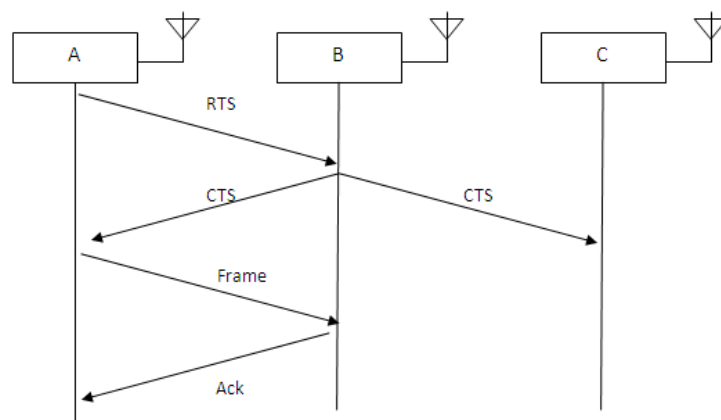


图 2-5 使用 RTS/CTS



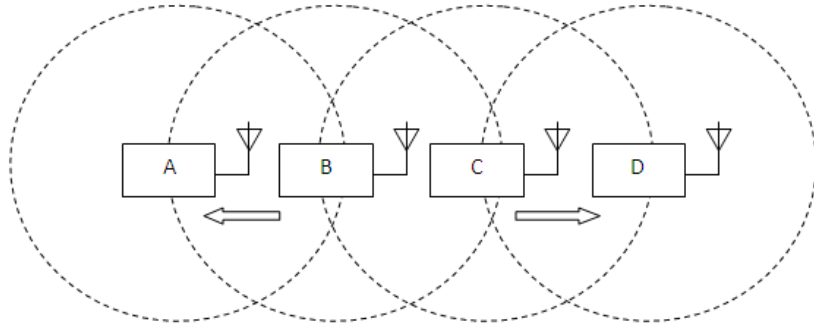


图 2-6 暴露节点

暴露节点是指一个这样子的无线终端节点，处于接收者的有效无线范围之外，又处于发送者的有效无线范围内。节点因为检测到发送节点的发送信号而延迟发送，但实际上因为它处于接收节点的通信范围之外，所以它的发送实际上不会造成冲突。由于推迟处理，这样子会导致网路容量降低。如图 2-6 所示，节点 B 向 A 发送数据，C 也在向节点 D 发送数据。由于节点 B、C 能感知到对方的存在，所以一方在发送数据时，另一方必须等待，延迟一段时间才能发送。这种问题即为暴露节点问题，会增加不必要的延迟，特别是车载环境中实时性非常重要的应用。而实际逻辑上，节点 B 发给节点 A 的时候，C 节点也可以发给节点 D，虽然节点 B、C 出信号会混淆，但节点 A、D 最终都能收到正确的信号。

分布式协调功能（Distributed Coordination Function, DCF）和点协调功能（Point Coordination Function, PCF）是 IEEE802.11 所定义两种 MAC 控制方法<sup>[24]</sup>。DCF 是节点共享无线信道进行数据传输的基本接入方式，它把载波侦听多路访问/碰撞避免（Carrier Sense Multiple Access with Collision Avoidance, CSMA/CA）<sup>[25]</sup>技术和确认（ACK）技术结合起来。它采用二进制指数回退策略来避免冲撞。PCF 是一种可选的无线网络媒介访问控制方法，主要用于支持近乎实时的应用。

这些方法并没有百分百地确保冲突不发生，而是尽可能减少冲突发生。当无线网络中终端节点数目越来越多时，冲突几率也会越来越大。

### 2.2.2 IEEE802.11p

802.11p 是作为 IEEE802.11 的修正议而出现，遵循相同的 IEEE802.11 协议架构，目的是为了适应车联网环境。在 PHY 上，802.11p 采用了 OFDM<sup>[26]</sup>，与 802.11a 一致，这种方式抗干扰性强，更能适合快速变化的车辆环境，信道带宽缩小一半，从 IEEE802.11a

的 20MHz 变成 10MHz，从而减小多普勒效应，同时也减少了码间串扰。IEEE802.11p 还将信道进行划分，将 5.850-5.925GHz 这 75MHz 的带宽划分为 7 个 channel。

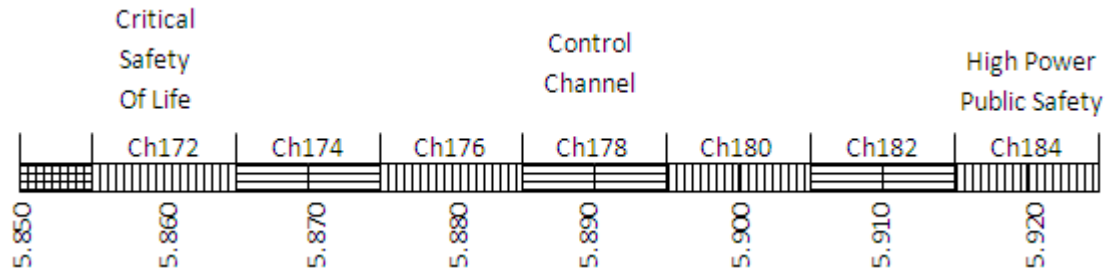


图 2-7 WAVE 频谱分配图

如图 2-7 所示, IEEE802.11p 定义了两类型信道, 一种是服务信道 (Service Channel, SCH), 另一种是控制信道 (Control Channel, CCH)。Ch178 即为 CCH, 频率范围为 5.885GHz-5.895GHz, 主要负责传输实时性要求比较高的数据, 以及 WSA 报文, Ch174、Ch176、Ch180 和 Ch182 都是 SCH, 主要负责传输实时性要求不高的应用或者娱乐商务应用, 如视频通话、邮件等应用。最左 Ch172 和最右 Ch184 的信道保留着专门用途。5.850-5.855GHz 这频带为空闲频带, 不使用。

IEEE802.11p 中, 其主要控制中心节点 AP 称为路侧单元 (Road-Side Unit, RSU), 移动工作站 STA 称为车载单元 (On-Board Unit, OBU)<sup>[27]</sup>。RSU 指的是固定在各种道路两侧的中心控制节点设备, 负责一块区域的无线网络, 并且一般通过有线连接到互联网。WAVE 网络组网可以有两种方式。其中第一种是 WIBSS (WAVE Independent Basic Service Set)。WIBSS 指的是在没有路侧单元 RSU 时, 多个车载单元 OBU 之间自发组成的自组网络, 这种网络概念上属于 Ad-hoc 网络范畴。第二种是 WBSS (WAVE Basic Service Set)。WBSS 指的是存在路侧单元 RSU 时, 一个路侧单元 RSU 和多个车载单元 OBU 组成的网络, RSU 和 OBU 之间按照 WAVE 协议规定的方式通信。CCH 上会周期性地广播 WSA (WAVE Service Announcement), 其作用相当于 IEEE802.11 网络中 Beacon 报文作用。车载单元 OBU 驶入路侧单元 RSU 所覆盖的有效范围 WBSS 中, 通过收到的 WSA 选择加入满足需要的 WBSS。在驶离 WBSS 有效覆盖范围之后, 将退出该 WBSS 网络, 选择加入下一个 WBSS 网络。

除了上述描述的 WBSS 和 WIBSS 这两种方式之外, IEEE802.11p 协议还新增了一种 “MAVE Mode” 模式, 在这种模式下, OBU 加入无线网络时, 没有信道扫描、关联和认证等加入网络的过程, 直接在无线网络中发起通信, 减少冗长的认证过程, 适应了车载

环境下车辆快速移动从而停留在无网络覆盖范围内时间大大缩短的情况。而非 **WAVE Mode** 模式下，其加入网络过程为：

OBU 发送 probe request→

←RSU 发送 probe response

OBU 发送身份验证 Authentication request→

←RSU 发送 Authentication Confirm

OBU 发送关联 Association request→

←RSU 发送 Association Confirm，注册一个用户

## 2.3 IEEE1609 协议栈

IEEE1609 标准是以 IEEE802.11p 通讯协定为基础的高层标准，是车联网协议的组成部分，是 IEEE 针对车载环境所定义的通讯系统架构和标准化服务接口。IEEE1609 协议族在 11p 协议基础上，定义上层相关的协议，包括数据链路层、IP 网络层、数据传输层、通信安全等方面，定义了多信道协调工作来增强信道可用性，区分紧急重要数据和普通数据。IEEE1609 系统列标准协议如表 2-2 所示[28]。

协议	用途
IEEE1609.0	整体性概述，介绍了 <b>WAVE</b> 架构和服务，以及协议族其它系列协议标准。 <sup>[29]</sup> 。
IEEE1609.1	详细定义了 <b>WAVE</b> 架构的资源管理( <b>Resource Manager</b> )功能 <sup>[30]</sup> 。
IEEE1609.2	详细定义了 <b>WAVE</b> 架构的安全机制。
IEEE1609.3	详细定义了 <b>WAVE</b> 系统中的网络服务 ( <b>Networking Services</b> ) <sup>[31]</sup> 。
IEEE1609.4	详细定义了 <b>WAVE</b> 系统中 <b>MAC</b> 层的多信道协调机制 ( <b>Multi-channel Operation</b> ) <sup>[32]</sup> 。

表 2-2 IEEE1609 系列

其中，IEEE1609.4 的多信道协调，是针对 IEEE802.11p 协议的扩展。多信道协调方式指信道划分为控制信道 **CCH** 和服务信道 **SCH**。对于实时性高地报文，还有 **WAVE** 传输控制协议的报文，都在控制信道 **CCH** 上传输。这类报文有路况报文，车辆故障报文，

车祸报文等。而实时性要求不高的报文，允许一定丢包率，这类则在传输信道 SCH 上传输，这类报文有天气预报，娱乐新闻，视频图像等引用<sup>[4]</sup>。由于单个天线在某个时刻只能以一个频率收发无线信号，所以要同时支持两种信道，要么需要两根以上天线，要么单根天线不断地切换频道频率。IEEE1609.4 中明确了单天线的信道协作机制。不同的无线设备之间的信道协调，在时间上必须满足通信设备之间的同步，使得无线设备工作在约定的信道上。如图 2-8 所示。

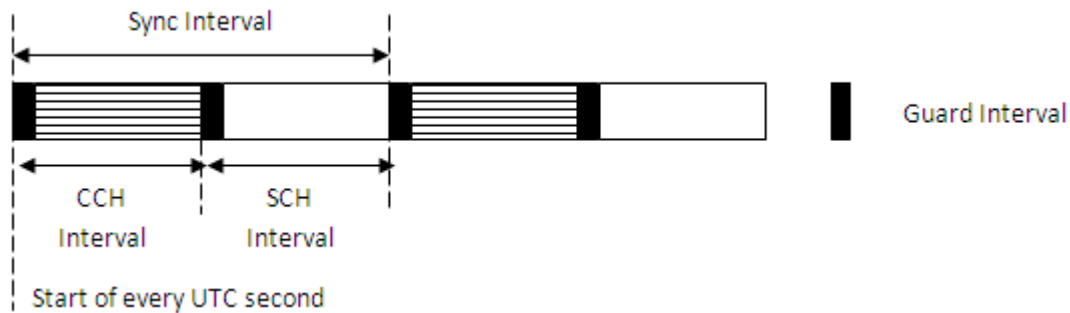


图 2-8 CCH 和 SCH 的间隔

IEEE1609.4 协议中允许使用 GPS 卫星信号中的时间信息来同步所有设备，也允许 RSU 广播 WSA 广播信号进行同步。多天线多无线模块，可选择其中一个工作在控制信道上，剩下的在服务信道上工作。

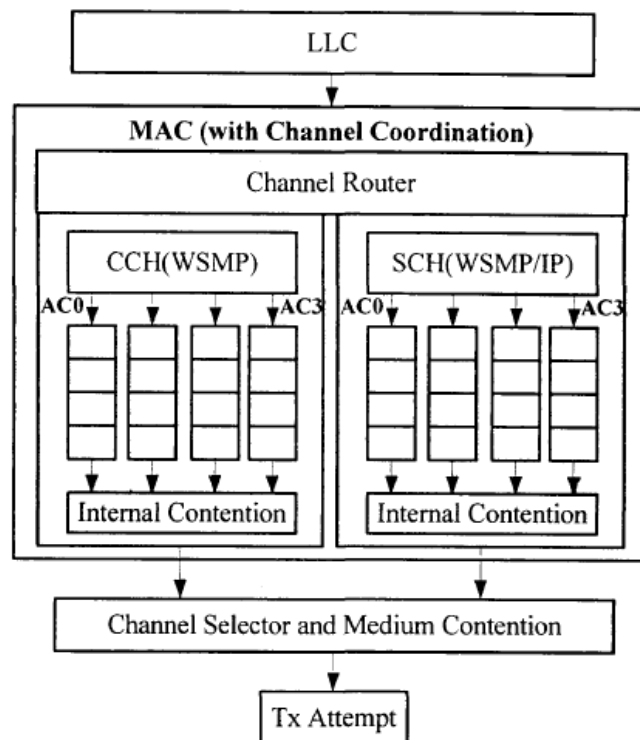


图 2-9 信道协调机制中的两个 MAC 实体

IEEE1609.4 协议所定义的多信道协调机制框架建立在 802.11p 所定义的 MAC 实体之上, SCH 和 CCH 均看成独立存在的 MAC。如图 2-9 所示, MAC 中新增了一个 channel Router, 用来路由上层报文, 选择控制信道 CCH 还是服务信道 SCH。对于实时性要求高的报文只能路由到控制信道 CCH 上, 其他的普通数据则路由到服务信道 SCH。上。控制信道和服务信道进一步采用了 EDCA 机制, 有 4 个发送队列, 不同发送队列有不同的优先级。一个上层数据报发送出去的流程如下: MAC 收到上层的数据包之后, 首先经过 Channel Router 路由选择合适的信道, 然后根据上层数据包的发送优先级选择合适的发送队列, 接着 EDCA 机制根据设置选择一个上层数据包传递给下层的 MAC 实体, 最后改 MAC 实体通过 DCF 竞争取得物理无线媒介发送权时将数据发送出去。可以看到, 数据在发送过程中会经过多个层次的竞争, 形成不同等级的优先权, 紧急的数据优先级比较高, 会被尽快发送出去, 延迟小, 而一般的普通数据则优先级较低, 会有稍高的延迟, 从而保证 WAVE 的 QoS。由于 WAVE 多了 Channel Router 等, 所以上层数据包的头部需要新增一些参数配置, 比如功率、信道编号、数据速率等。

IEEE1609.3 协议定义了 WAVE 的无线接入网络层服务 WAVE 短消息协议 (WAVE Short Message Protocol, WSMP), 对应 OSI 模型的网络层和传输层, 主要提供报文传输和寻址。IEEE1609.3 还定义了用于网路管理的 WAVE 管理实体 (WAVE Management Entity, WME)。当 WAVE 的发起方需要发送时, 填充目的广播地的 MAC, 设置 WAVE 短信息协议参数, 经过一定步骤之后传在信道上发送出去。目的设备接收到 WAVE 短信息协议帧之后, 依据 ID 值来将 WAVE 短信息交给对应的上层服务程序。WSMP 可在 CCH 和 SCH 上传输。IEEE1609.3 定义了设备的两种角色, 服务使用者 (User) 和服务提供者 (Provider)。建立服务过程中, 服务提供者在 CCH 上广播 WSA, 通知其他设备有哪些服务内容。这种广播帧中, 包含了许多重要信息, 如 Service Priority, Provider Channel, Service ID 等, 这些信息由 WME 产生。User 在控制信道上监听 WSA, 搜集形成一个有效的服务列表。当 User 的上层应用程序需要使用某个服务时, 就从服务列表中找到所需服务的信息, 然后根据该信息所包含的工作信道、传输速率、服务 ID、优先级等参数, 在适当的时间跳转到适当的频道上以适当的速率接收服务提供者所提供的服务。对于实时性较高的数据应用, 可以选择直接在 CCH 上传输, 这种方式不需要 WSA 广播, 直接对等通信, 进一步减小延迟。

## 2.4 Linux 相关编程

常见的嵌入式系统大体上分为两类：商用型，需要费用，一般费用不低，有售后服务提供支持，系统较为可靠，一般不开放源码；免费型，大部分开源，由全球爱好者自行维护和开发，有开发社区支持，但不保证商用型那种售后服务。

以下是常见的商用型嵌入式操作系统有<sup>[33]</sup>：

- **VxWorks** 是美国公司开发的嵌入式实时操作系统（RTOS），其最大特点是实时性高。公司提供了完整的开发工具，用户开发环境友好，持续发展能力好，用户能较好地专注于产品开发。
- **Windows Embedded** 是美国公司开发的模块化实时嵌入式操作系统，主要是针对智能化、移动式。由于公司的背景，界面上和 windows 界面较为统一，相比 VxWorks 不够实时性。

而最常见的免费型嵌入式操作系统是 **Linux**。**Linux** 是一套免费使用和自由传播的类 Unix 多用户网络操作系统，它是从一群开源爱好者自发开发组织的，兼容 POSIX 标准，具有 BSD 和 SYSV 的扩展性。它性能稳定，源码开放，支持大量的 UNIX 工具软件以及现有大部分的网络协议<sup>[34]</sup>。

**Linux** 操作系统诞生于 1991 年的 10 月 5 日。目前 **Linux** 存在着许多不同的 **Linux** 发行版本，它们最主要的共同点是 **Linux** 内核。**Linux** 可安装在各种常见的电子设备上，比如智能手表、智能手机、PC、超级计算机等。目前在智能手机和平板上广泛使用的 **Android** 系统，就是以 **Linux** 内核为基础开发出来的系统。自由软件中，从大的方向可划分为操作系统和应用软件。有了 **Linux** 这一开源的操作系统为基础，其他的自由软件也得到极大的发展，整个自由软件才具备真正的活力。只要遵循 **GNU** 通用公共许可证，任何个人和机构都可以自由地使用 **Linux** 的所有底层源代码，也可以自由地修改和再发布。严格从 **Linux** 源代码来讲，**Linux** 是指内核。而由于目前大多 **Linux** 发行版中均使用了 **Linux** 内核源代码以及其他配套的相关开源自由软件，所以 **Linux** 也常用来指基于 **Linux** 内核的、包括其他开源自由实用工具的完整操作系统。。由于这些支持用户空间的系统工具和库主要由理查德·斯托曼于 1983 年发起的 **GNU** 计划提供，自由软件基金会提议将该组合系统命名为 **GNU/Linux**，但 **Linux** 不属于 **GNU** 计划。

**Linux** 最初是作为支持英特尔 x86 架构的个人电脑的一个自由操作系统。目前 **Linux** 已经被移植到更多的计算机硬件平台，远远超出其他任何操作系统。**Linux** 是一个领先的操作系统，其主要运行在两大类硬件上，一类是嵌入式设备上，如路由器、机器人等，另一类是服务器，如常见的 **Web** 服务器等。

Linux 的哲学思想基本为：一切都是文件。也就是 Linux 下任何设备任何资源，都可以当做一个文件，有着类似一般文件的操作方法和接口，例如 open、close、read、write、ioctl 等。

### 2.4.1 Linux 的特点

- 完全免费和自由

Linux 是一款免费的操作系统，世界上任何一个人都可以免费获得和修改其源代码，甚至参与开发。正是因为这种社区式的合作开发方式，使得更多人可以参与 Linux 的发展，使得 Linux 发展越来越强大。

- 开放性

现有厂商千差万别，各家厂商需要设备在互联时，若遵循一致的标准规范则可彼此兼容方便实现互联，这就是开放性。Linux 遵循开放系统互连国际标准和 POSIX 标准。这样子遵循 POSIX 的 Windows CE 等也能方便和 Linux 方便互联。<sup>[35]</sup>

- 完全兼容 POSIX 标准

POSIX 表示可移植操作系统接口，定义 Unix 操作系统上一套标准的 API，目的在于提供更高的移植性。Linux 属于类 Unix 系统，自然 Linux 也就遵守 POSIX 标准。

- 多用户多任务

多任务使得计算机可以同时执行多个进程程序，各个程序运行相互独立。在单核 CPU 系统上，依靠任务切换实现微观单任务串行宏观多任务并行。在多核 CPU 系统上，可以实现真正的多任务同时并行。多用户允许多个不同的用户在同一时刻同时登陆到系统同时使用系统资源，每个用户有自己的资源和权限，互不影响。<sup>[36]</sup>

- 良好的界面

Linux 传统交互界面是 shell，字面意思即出于内核之外的外壳程序。Shell 是基于文本的命令行界面，功能强大，效率高效。同时它也是可编程的，从而可以扩充提供高级的功能。Linux 也提供图形用户界面，一般是 XWindow。它实现了一般桌面系统的各个元素，如窗口、滚动条等，提供一个非常直观的界面给用户，增加交互性。Linux 还给程序开发人员提供了直观方便的编程调用接口界面，这使得开发程序开发效率大大提高。

- Linux 的设备独立性



Linux 将一切当做文件来看待，设备也不例外。任何用户都可以像使用系统中其他文件一样，操纵、使用这些硬件设备，而不必关心和了解底层硬件的差异性，使得 Linux 的设备独立性特别强。Linux 设备驱动开发有很好的框架，驱动开发人员只需要按照框架开发即可方便地添加新设备到操作系统中。<sup>[37]</sup>

- Linux 供了丰富的网络功能

Linux 的最强大的功能之一是它的网络功能。自由和免费性使得很多大公司都选择了 Linux 作为服务器，在其上面部署各种强大丰富的网络功能。可以说，Linux 和网络是互为促进的。网络使得开源爱好者更加方便地协同开发，而 Linux 又给网络应用开发商和提供商带来了免费稳定可靠的操作系统基础。<sup>[38]</sup>

- Linux 提供了可靠的系统安全

由于 linux 开源，所以大基数的爱好者发现 bug 更快更多，从而减少了潜在 bug 只被黑客发现，减短了由于 bug 受伤害的威胁。同时 Linux 采取了很多技术和措施来保障系统安全。

- Linux 支持多种平台

Linux 可以运行在多种硬件平台上，在 Linux 源代码目录下划分了针对不同底层硬件的底层源代码，添加新硬件支持也很方便，主要添加相应底层代码即可。

## 2.4.2 Linux 体系结构

Linux 操作系统由 4 个主要的子系统所组成<sup>[39]</sup>：

- 用户应用程序—在某个特定的 Linux 系统上运行的应用程序集合，它将随着该计算机系统的用途不同而有所变化，但一般会包括文字处理应用程序和 Web 浏览器。
- O/S 服务—这些服务一般认为是操作系统的一部分（XWindow 系统，命令外壳程序 shell 等等）；此外，内核的编程接口（编译工具和库）也属于这个子系统。
- Linux 内核—包括内核抽象和对硬件资源（如 CPU）的间接访问。
- 硬件控制器—这个子系统包含在 Linux 实现中所有可能的物理设备，例如，CPU、内存硬件、硬盘以及网络硬件等都是这个系统的成员。

Linux 内核由 5 个主要的子系统构成

- 进程调度程序（SCHED）：负责控制系统中所有进程以合理顺序访问和使用 CPU 资源。进程调度程序有不同的策略，这些策略要保证合理性，不同性质的进程需要不同的优先级，例如交互性的进程，一般需要快速响应但处理很快，这种优先级高。

- 内存管理程序 (MM): 使多个进程可以安全地共享机器的主存系统。此外, 内核管理程序支持虚拟内存。现代计算机虚拟内存管理使得程序可以使用超过系统中的物理内存数量的内存。<sup>[40]</sup>Linux 内存管理是尽可能地缓存在内存中, 而不像 windows 那样子放在磁盘上。当需要使用更多内存时, 系统按照一定策略选择部分内存交换到磁盘上, 空出内存空间, 等到需要时再交换回来。
- 虚拟文件系统 (VFS): 向系统使用者提供一个完整通用的文件系统视图, 抽象了公共文件接口, 由硬件驱动和底层去实现不同文件之间的细节差异。除了支持自身原生的文件系统如 ext2 ext3 ext4 等, 还支持其他文件系统, 如 FAT32、NTFS 等。
- 网络接口 (NET): 提供了对许多建网标准和网络硬件的访问。
- 进程间通信 (IPC) 子系统。

### 2.4.3 Linux 设备驱动程序

操作系统的主要作用有两个, 一个是屏蔽底层千差万别的硬件差别, 给上层提供一个相同的接口, 另外一个管理和调度系统资源。驱动程序主要是透明化硬件的差别, 使得用户无需关心硬件的差别。在 Linux 中, 所有设备均抽象化文件, 所以所有用户在使用硬件时跟使用普通文件并无多大差别。设备驱动处于物理硬件和内核上层之间, 具有重要的桥梁作用。Linux 中设备分为三大类, 字符设备、块设备和网络设备<sup>[41]</sup>。

#### ● 字符设备

字符设备指的是按照字节流顺序访问的设备, 单个字符依序读写, 例如串口、键盘等设备。字符设备由字符设备驱动程序驱动, 字符设备驱动程序要实现 VFS 中的 open、close、read 和 write 等几个接口。从这几个接口可以看出, 用户应用程序在使用字符设备时, 跟使用普通文件的方式没有差别。字符设备在文件系统的路径一般存放在/dev 下, 可以通过 FS 节点来访问。字符设备相比普通文件, 差别在于字符设备并不能随机定位访问位置 lseek, 而普通文件可以。

#### ● 块设备

块设备指的是每次按照一块来传输的设备, 例如磁盘、软盘等。一般这些设备上能容纳文件系统, 一般每次访问块的大小为 512B 字节, 或者 2 的更高次幂字节, 但不支持字符的寻址, 只能读取完整一块, 再从中获取所需的字符。块设备允许随机访问。块设备也可以当做字符设备提供给应用程序使用, 但更多的是用来存放一个文件系统来使用, 所以块设备驱动和字符设备有不同的内核接口。

## ● 网络设备

网络设备指的是完成内核中网络子系统用于交换数据的设备,最常见的是 NIC 网卡,也可以是软件虚拟的,比如 `loopback` 回环接口。网络设备负责网络数据包的收发,但它并不负责网络连接相关信息的记录。网络设备驱动程序在收到网络数据报时,首先验证包的完整性有效性,然后按照包的协议传递给上一层网络协议;或者收到上一层网络协议封装好的数据包,通过网络设备发送出去。网络设备在文件系统并不存在对应的节点,而是分配一个唯一的名字表示,如 `eth0`、`wlan0`,并提供相应 `ifconfig`、`iwconfig` 等专用于网络设备的工具。

Linux 内核编程中,和用户态编程有很大的区别。最主要是内核模块运行于内核空间,具有很大权限,可以访问所有的硬件资源,一旦出错,系统可能直接崩溃宕机。而应用程序运行用户态中,所有资源都是内核调度提供的,程序一旦出错,一般只有该程序会结束,而系统仍照常运行。由于内核模块运行在内核态中,所以并发控制、中断、内存映射等都需额外小心。

## 2.5 本章小结

本章主要介绍了论文的相关技术背景,首先研究了 IEEE1609 框架,这是整个车联网协议框架的核心,分为数据平面和管理平面两大块。接着讨论了 IEEE802.11 无线协议,定义了物理层和数据链路层的标准,同时也提出了 AP/STA、BSS/IBSS 等概念,介绍了隐藏节点、暴露节点等问题。之后介绍了 IEEE802.11p,这个协议主要是 IEEE802.11 协议针对车联网所做出的修订,提出信道划分、WAVE 模式。IEEE1609.4 提出了多信道协调机制,IEEE1609.3 提出了无线接入网络层服务 WAVE 短消息协议。本章最后主要是介绍了 Linux 操作系统的特点, Linux 操作系统的体系架构, Linux 内核的结构, Linux 设备驱动。Linux 是一个开放、自由的操作系统,适用于各种平台和应用。

为了解决 IEEE1609 协议族的多信道协调,第三章提出了双网卡系统方案,并使用 `libpcap` 来提高性能。

为了进一步优化双网卡中任一网卡使用信道的效率,第四章提出位置辅助随机访问理论方案,解决数量较多时的信道使用效率。

## 第三章 基于双网卡的媒体接入控制优化

由于车联网中特殊性，车联网中有的信息是属于紧急重要，，例如事故告警信息，需要快速和高优先级，而有些信息是属于服务，如邮件服务内容服务等允许一定的延迟。在一般系统上，单网卡很难做到这两者的区分和兼顾，IEEE1609 中也有了控制信道和服务信道的多信道协调概念。因此在这里提出了使用双网卡系统，不同网卡负责不同的信息传输，实现了 IEEE1609 标准协议中的控制信道和服务信道的多信道协调特点，同时为了进一步提高性能，使用了 Libpcap 裸套接字方法。

### 3.1 方案描述

802.11p 和 IEEE1609 协议中规定了控制信道和服务信道两种信道，控制信道主要是用来传输重要紧急的数据，而服务信道用来传输普通的应用数据。这里首先提出利用目前成熟的 IEEE802.11 协议网卡双无线网卡通信平台系统方案来模拟 WAVE 中的两种信道方式。本设计中，在应用层使用 libpcap 函数库直接指定虚拟网卡进行收发数据，然后在这基础上实现控制信道和服务信道工作方式。

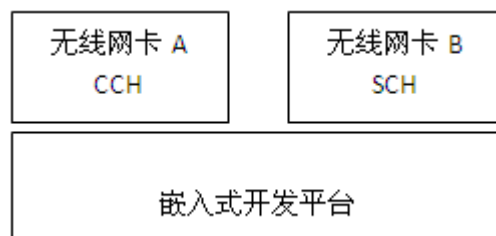


图 3-1 车联网通信终端系统硬件架构

硬件方案开发平台选用友善之臂的 Tiny6410 开发板，无线网卡选 RT3070 无线网卡。



图 3-2 友善之臂 Tiny6410 实物图

友善之臂 Tiny6410 是一款很常见的性能非常好的开发板，其主处理器 CPU 是三星 S3C6410（ARM11 架构）。该处理器 CPU 集成了众多强大功能的部件。在多媒体处理方面，它在硬件上支持 Mpeg4, H.264/H.263 等编解码；它的视频输出方式支持 TV 和 LCD 两种显示方式；它集成了 3D 图形硬件加速器以及 2D 图形图像缩放翻转等。友善之臂 Tiny6410 的存储方面为：256M 的 DDR RAM，256M 或者 1G 的 Nand Flash。友善之臂 Tiny6410 引出了 4 排引针，方便开发者定制自己需要的功能。Tiny6410SDK 是友善之臂公司开发的一个底板参考，在上面集成了常见的外设，如网络接口、串口、USB 主从接口、音频接口、中断式按钮开关、LED 灯、蜂鸣器等。<sup>[42]</sup>

本硬件平台上的操作系统为 Linux 操作系统，源码开放，移植性好，可裁剪和定制性高。开发板同时提供了相应配套的开发套件，比如编译器 arm-linux-gcc 等。

每个终端系统均由 tiny6410 和两块无线网卡组成。这两个网卡在逻辑上是绑定成一个虚拟网卡提供给上层。系统应用程序在这个虚拟网卡进行传输。在这次实现中，虚拟网卡主要是虚拟成一个进程方式提供给其他进程。

按照 IEEE802.11p 和 IEEE1609 协议族的定义，信道分为控制信道和服务信道，因此设置无线网卡 A 一直工作在控制信道 CCH 上，无线网卡 B 一直工作在服务信道 SCH 上，如图 2-7。无线网卡 A 主要用于传输紧急重要信息。另一块无线网卡 B 会根据控制信道上所广播的信道信息而设置工作频段。普通服务数据只能在服务信道即无线网 B 上传输。

由于控制信道网卡 A 一直工作在固定的专用频率上，且这个频率会被限制保护，因此保证紧急重要数据的传输。对于普通服务数据，紧急重要性较低，使用上比较自由，可通过控制信道协调另一块网卡 B 选择空闲的服务信道进行传输。

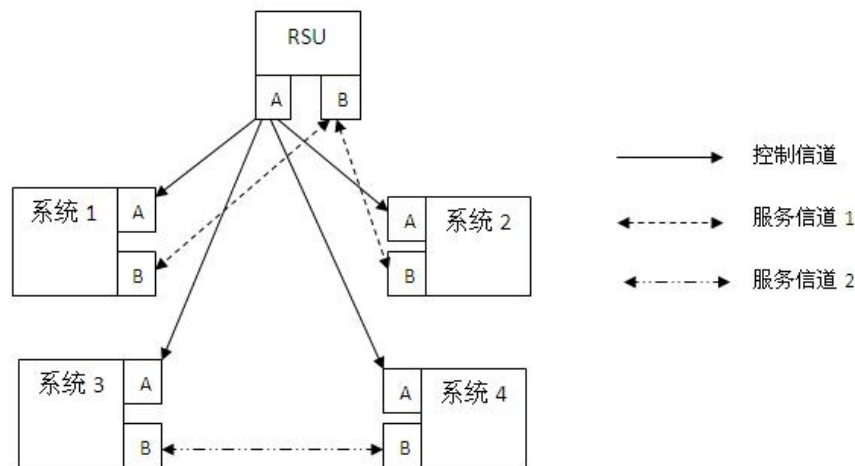


图 3-3 系统图

上述无线双无线网卡系统还有一个好处：当控制信道网卡由于一些原因而不能工作时，服务信道网卡将取代控制信道网卡工作在控制信道上传输紧急重要数据传输。

如图 3-3 所示，会有一个系统充当中心节点 **RSU**，其余系统均为客户端。系统启动时，中心节点 **AP** 的控制信道网卡会定期的发送信道协调信息，以及根据需要发送紧急重要数据。客户端的控制信道网卡监听控制信道上的信息，加入网络。

传输紧急重要数据时，中心节点 **RSU** 和客户端直接在控制信道上传输紧急重要数据。传输大量普通服务数据时：

#### （1）中心节点发送

如果已经存在服务信道通讯，则直接在服务信道上请求发送。如果只建立有控制信道通讯，中心节点则直接在控制信道上发布信道协调信息，声明将要使用的服务信道频率。然后系统之间在服务信道上建立通讯进行大量数据传输。

#### （2）客户端发送

如果已经存在服务信道通讯，则直接在服务信道上请求发送。如果只建立控制信道通讯，则在控制信道上向中心节点发送请求。中心节点 **AP** 收到请求后，将根据需要广播服务信道频道值，最终客户端的服务信道网卡将在服务信道上建立通讯进行传输。当传输结束时，会通过控制信道网卡控制信道或服务信道通知 **AP** 普通服务数据传输已结束。

控制信道上，大部分数据是中心节点 **AP** 下行发送给各个客户端，各个客户端较少在控制信道上发送数据给 **AP**，除了少量必须回应数据或请求数据。目的是尽量减少 CSMA 冲突几率。

普通服务数据不局限于 **AP** 和终端之间传输，它们也可以在终端之间进行传输。

如图 3-3 所示，系统 1、系统 2 在服务信道 1 上和 **AP** 进行普通服务数据传输，而系统 3 和系统 4 在另一个服务信道 2 之间进行数据传输。系统 3 和系统 4 服务信道使用完毕时，由于没有跟中心节点建立服务信道通讯，因此只能在控制信道上向中心节点通知。系统 1 和系统 2 除了可以在控制信道上向中心节点上通知和请求服务信道服务，也可以在当前服务信道上进行请求和传输。

对图 3-3 来说，每个系统均使用了双无线网卡，均使用了两个信道。对整个图来说，使用了一个控制信道和两个服务信道。

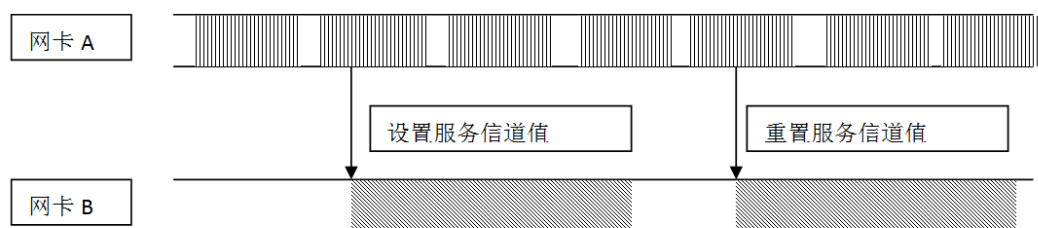


图 3-4 控制信道广播服务信道参数值

### 3.2 方案实现

传统内容广告系统均使用 **TCP/UDP** 方式为基础，在这基础上进一步设计开发内容广告系统。**TCP** 建立连接时，一般需要 3 次握手，释放一般连接需要 4 次握手<sup>[43]</sup>，连接分长连接和短连接，对于长连接来说，可以减少不必要的握手，但会一直占用系统资源，对于短连接，则会造成频繁的握手。**UDP** 虽然没有握手过程，但本身也包含各种复杂的机制。在本文章中提出使用 **libpcap** 裸套接字的方式，直接对网卡进行收发操作，同时使用双无线网卡协调分工完成紧急告警和数据服务。

### 3.2.1 libpcap 移植

libpcap<sup>[44]</sup>是一个开源的开发包，用于捕获和发送网络数据报，很多网络监控软件都是在 libpcap 基础上开发起来的，如 Tcpdump、Wireshark 和 ethereal 等。Libpcap 跨平台性很好，支持很多平台，在常见的类 unix 上可以工作。

**Libpcap** 主要由网络分接头 **Network Tap** 和数据过滤器 **Packet Filter** 两部分组成。<sup>[45]</sup> 网络分接头 **Network Tap** 主要是获取网络数据拷贝，过滤器 **Packet Filter** 则按一定规则用来过滤数据报是否满足需要。**Libpcap** 底层上是使用 **raw socket** 技术。对于发包，比较简单，设置好数据包格式，然后传递给底层网络接口的设备驱动程序，由其处理好 **MAC** 竞争后发送出去。在接收数据包方面，底层网络接口驱动程序在接收到数据包时，确定是否有进行在此接口上监听，有的话则拷贝一份数据包，则原数据包仍按照原来的处理传递给上层协议处理。新的数据包拷贝会传递到 **BPF** (**BSD Packet Filter**) 过滤器，过滤器上有许多用户定义的规则，用户可以使用通配符、正则表达式等设置过滤条件，然后再转化为 **BPF** 过滤器语法。满足 **BPF** 过滤器规则数据包最终放入内核缓冲区再给用户进程，不满足的数据包则直接丢弃。如果用户不指定任何规则，就认为所有数据包均满足，均传递给应用程序。



一般应用开发中 libpcap 的所主要使用的函数接口<sup>[46]</sup>是:

pcap\_open\_live(): 类似于打开文件, libpcap 也有一个开打网络设备的步骤, 即这里这个函数, 传递进设备名字等参数, 而返回一个类似于文件描述符的描述字, 用于后续其他操作。

pcap\_compile(): 将用户指定的匹配条件编译到系统中, 生成用于 BPF 过滤引擎的代码。

pcap\_setfilter(): 绑定过滤器, 一旦绑定, 就开始抓包和过滤。

pcap\_loop()函数 pcap\_dispatch(): 这些用于获得过滤后的数据包。

pcap\_next()和 pcap\_next\_ex(): 这两个函数也是用于获取数据包。

pcap\_close(): 有打开就有关闭, 这个函数就是用于释放网络设备和资源。

上述是常见的基本的步骤, 还有其他几个接口用来辅助开发的, 如用于查找网络设备的 pcap\_lookupdev()函数, 获得网路号掩码的 pcap\_lookupnet()函数。

本次实现 libpcap 移植到 tiny6410 开发板的过程:

(1) 从 <http://www.tcpdump.org> 网站上下载 libpcap 源码包, 本此移植过程 libpcap 版本号为 1.0.0。

(2) 解压, 然后进入到目录中

(3) 配置和编译

```
$ CC=arm-linux-gcc ac_cv_linux_vers=2 ./configure --host=arm-linux --with-pcap=linux
```

```
$ make
```

修改 Makefile 中内容, /home/wangxing/armlinux/4.5.1/是电脑上交叉编译工具链目录。

```
prefix = /home/wangxing/armlinux/4.5.1/
```

(4) 安装相关头文件和函数库 libpcap.a 放到 arm-none-linux-gnueabi 目录下相关文件夹里边去。之后就可以使用 arm-linux-gcc 来编译相关的 libpcap 应用的软件。

```
$ make install
```

(5) 编译过程还产生了 libpcap.so.1.2.1 文件, 这个文件通过串口下载到开发板中。

```
tiny6410 $:rz
```

```
SecureCRT: send libpcap.so.1.2.1 libpcap.a tcpdump
```

```
tiny6410:
```

```
$ mv libpcap.so.1.2.1 /lib/libpcap.so
```

### 3.2.2 libpcap 抓包收发示例

这里展示一个简单的无线网卡收发例子。一个无线网卡会发送一个以下格式的内容：

“Packetspammer %02droadcast packet#%05d -- :-D --%s ----",Rate/2, Ordinal, Hostname

最终效果如图所示，其中图 3-5 是指 x86 架构的 PC 平台上无线网卡收发的，图 3-6 是 tiny6410 开发板接收到的。

```

lab@lab-virtual-machine: ~/packetspammer
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
0010: 62 72 6F 61 64 63 61 73 74 20 70 61 63 68 65 74 broadcast packet
0020: 23 30 30 30 31 33 20 2D 20 20 3A 2D 44 20 2D 2D #0013 -- :-D --
0030: 6C 61 62 2D 76 69 72 74 75 61 6C 2D 6D 61 63 68 lab-virtual-mach
0040: 69 6E 65 20 62 6F 62 6F 00 00 00 00 ine bobo....
transmit
receive
rtap: 0000: 00 00 0D 00 04 80 02 00 02 00 00 00 00 .....
RX: Rate: 1.0Mbps, Freq: 0.0GHz, Ant: 0, Flags: 0x0
0000: 50 61 63 68 65 74 73 70 61 6D 6D 65 72 20 32 34 Packetspammer 24
0010: 62 72 6F 61 64 63 61 73 74 20 70 61 63 68 65 74 broadcast packet
0020: 23 30 30 30 31 34 20 2D 20 20 3A 2D 44 20 2D 2D #0014 -- :-D --
0030: 6C 61 62 2D 76 69 72 74 75 61 6C 2D 6D 61 63 68 lab-virtual-mach
0040: 69 6E 65 20 62 6F 62 6F 00 00 00 00 ine bobo....
transmit
receive
rtap: 0000: 00 00 0D 00 04 80 02 00 02 00 00 00 00 .....
RX: Rate: 1.0Mbps, Freq: 0.0GHz, Ant: 0, Flags: 0x0
0000: 50 61 63 68 65 74 73 70 61 6D 6D 65 72 20 31 38 Packetspammer 18
0010: 62 72 6F 61 64 63 61 73 74 20 70 61 63 68 65 74 broadcast packet
0020: 23 30 30 30 31 35 20 2D 20 20 3A 2D 44 20 2D 2D #0015 -- :-D --
0030: 6C 61 62 2D 76 69 72 74 75 61 6C 2D 6D 61 63 68 lab-virtual-mach
0040: 69 6E 65 20 62 6F 62 6F 00 00 00 00 ine bobo....
transmit
lab@lab-virtual-machine:~/packetspammer$

```

图 3-5 x86 发送和接收反馈图

```

File Edit View Terminal Help
RX: Rate: 1.0Mbps, Freq: 2.442GHz, Ant: 2, Flags: 0x0
0000: 50 61 63 68 65 74 73 70 61 6D 6D 65 72 20 34 38 Packetspammer 48
0010: 62 72 6F 61 64 63 61 73 74 20 70 61 63 68 65 74 broadcast packet
0020: 23 30 30 30 31 32 20 2D 20 20 3A 2D 44 20 2D 2D #0012 -- :-D --
0030: 6C 61 62 2D 76 69 72 74 75 61 6C 2D 6D 61 63 68 lab-virtual-mach
0040: 69 6E 65 20 62 6F 62 6F ine bobo
receive
rtap: 0000: 00 00 12 00 2E 48 00 00 00 02 8A 09 A0 00 80 01 .....H.....
0010: 00 00 ..
RX: Rate: 1.0Mbps, Freq: 2.442GHz, Ant: 2, Flags: 0x0
0000: 50 61 63 68 65 74 73 70 61 6D 6D 65 72 20 33 36 Packetspammer 36
0010: 62 72 6F 61 64 63 61 73 74 20 70 61 63 68 65 74 broadcast packet
0020: 23 30 30 30 31 33 20 2D 20 20 3A 2D 44 20 2D 2D #0013 -- :-D --
0030: 6C 61 62 2D 76 69 72 74 75 61 6C 2D 6D 61 63 68 lab-virtual-mach
0040: 69 6E 65 20 62 6F 62 6F ine bobo
~Cdevice wlan0 left promiscuous mode

[root@FriendlyARM /]# ./packetspammer wlan0
Packetspammer (c)2007 Andy Green <andy@warmcat.com> GPL2
device wlan0 entered promiscuous mode
DLT_IEEE802_11_RADIO Encap
RX Filter applied
(delay between packets 1000000us)

```

图 3-6 Tiny6410 开发板上接收到的数据

具体代码使用 libpcap 的抓包框架，即可实现。以下是其中所用到的数据结构体，从这些可看出，这种方法可定制任何的数据头部，然后让无线网卡直接发送出去。

```
static const u8 u8aRadiotapHeader[] = {
```

```
    0x00, 0x00, // <-- radiotap version
```

```
    0x19, 0x00, // <-- radiotap header length
```

```
    0x6f, 0x08, 0x00, 0x00, // <-- bitmap
```

```
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, // <-- timestamp
```

```

0x00, // <-- flags (Offset +0x10)
0x6c, // <-- rate (Offset +0x11)
0x71, 0x09, 0xc0, 0x00, // <-- channel
0xde, // <-- antsignal
0x00, // <-- antnoise
0x01, // <-- antenna
};

static const u8 u8aIeeeHeader[] = {
    0x08, 0x01, 0x00, 0x00,
    0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
    0x13, 0x22, 0x33, 0x44, 0x55, 0x66,
    0x13, 0x22, 0x33, 0x44, 0x55, 0x66,
    0x10, 0x86,
};

typedef struct {
    int m_nChannel;
    int m_nChannelFlags;
    int m_nRate;
    int m_nAntenna;
    int m_nRadiotapFlags;
} __attribute__((packed)) PENUMBRA_RADIOTAP_DATA;

```

### 3.2.3 双网卡平台系统

在这个系统中，无论是控制信道还是服务信道上的数据，在底层都是使用了如图 3-7 所示的数据帧格式。长度是一个 16bit 的字段，用于指示后续的类型 ID 的长度。不同类型 ID 其后续的有效数据内容不同。对于控制信道上的重要紧急数据，类型 ID 后续直接跟其内容。而对于服务信道上的应用数据，有效数据最开始的多少字节按照类型 ID 约定的字段是有效内容的长度。总之，这种数据帧方式简单，可扩展性强。这种底层的格式暂时只区分控制信道广播、控制信道内容单播等，对于上层的不同内容，比如安全信

息中的交通事故、道路状况、车辆故障等，在有效数据中再次封装，由应用层封装和解封装。

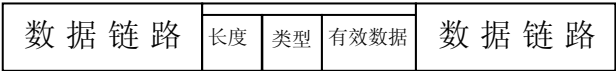


图 3-7 数据帧格式

首先，网卡抽象层中用 `pthread_create()`<sup>[47]</sup> 启动两个线程，分别对两个无线网卡进行监控，使用 `libpcap` 专门负责底层的收发包，使用 `iwconfig` 来设置无线网卡的工作频道。这两个线程只接受上层封装好的数据，然后选择合适频道，封装好长度和类型 ID，从底层无线网卡发送出去。接收时，根据长度和类型 ID 区分出控制信道上的 CCH 数据和服务信道的 SCH 数据。在这里，CCH 数据的 ID 设为 0x0F，SCH 数据的 ID 设为 0xF0。



图 3-8 平台抽象图

RSU 上有一个线程，专门用来广播紧急事件，同时接收其他 OBU 上传的紧急事件。考虑到 OBU 会离开一个区域，所以 OBU 上传紧急事件之后，可选择由 RSU 代为广播紧急事件。在广播过程中，所有的 RSU 和 OBU 均会记住有哪些终端节点存在。



图 3-9 发送安全广播数据

图中数据封装很直接，安全信息类中，用一个字节来表示，其中用 0x01 表示交通事故，0x02 表示道路状况，0x04 表示车辆故障，剩下的其他值作为扩展用。安全等级也是用一个字节表示，0x01 表示特大，0x02 表示严重，0x04 表示重要，0x08 表示普通，

剩下其他值作为扩展用。是否重复、重复间隔时间和重复次数只是作为应用层序使用，不进行封装。接着是最后的消息摘要，首先是表示消息摘要的长度，如“Car Collision!”不含双引号的长度为 14，即 0x0e，接着是消息摘要的内容。对于上图的情况，发送到 CCH 上时的有效数据内容是

0x01 0x04 0x0e 0x43 0x61 0x72 0x20 0x43 0x6f 0x6c 0x6c 0x69 0x73 0x69 0x6f 0x6e 0x21



图 3-10 接收到紧急广播数据

如图 3-10 所示，OBU 接收到紧急广播数据时，会在屏幕上弹出一个窗口，同时用 QSound bells("sound/alert.wav")播放提醒铃声。



图 3-11 发送服务数据

发送服务数据时，要指定发送内容，在这里暂时以文本文件和图片文件为例，指定



图 3-12 显示收到的服务数据

发送对象，指定发送的服务信道。发送时，终端节点会先在控制信道上广播要发送的对象和发送的服务信道，待对方做好准备时，再在服务信道上发送服务内容。对方接收到后，将接收到的文本或者图片显示出来。

### 3.2.4 性能测试

在这种系统中，数据发送和接收受底层硬件、内核等元素影响，无法直接分析其速率性能。这里主要针对 `libpcap` 和 `TCP/UDP` 这两种做对比，从而得到性能上的感知。

在这次测试中，无论 `libpcap` 还是 `TCP/UDP`，每次发包都选择相同的大小，比如 `1514B`。单次测量发包的时间，受其他因素影响太大，所以改成测试多次累积时间。

首先测试读 `10M` 文件的时间，中间不加发送接收等其他调用。

```
gettimeofday(&start, NULL);
while( (ReadSize = read (fd, Buf, BUFSIZE)) >0)
{
    //ret = pcap_sendpacket(device, (const u_char *) Buf, ReadSize);
    //if( ret != 0)
    //{
    //    printf("sendpacket error %d %s\n", ret, pcap_geterr(device));
    //    break;
    //}
    gettimeofday( &end, NULL);
    printf("%ld\n", 1000000 * ( end.tv_sec - start.tv_sec) + end.tv_usec - start.tv_usec);
}
```

上述代码中已经把 `libpcap` 发送的代码注释掉了。

接着测试读 `10M` 文件并通过 `libpcap` 发送出去的时间。

```
gettimeofday(&start, NULL);
while( (ReadSize = read (fd, Buf, BUFSIZE)) >0)
{
    ret = pcap_sendpacket(device, (const u_char *) Buf, ReadSize);
    if( ret != 0)
    {
        printf("sendpacket error %d %s\n", ret, pcap_geterr(device));
        break;
    }
}
gettimeofday( &end, NULL);
printf("%ld\n", 1000000 * ( end.tv_sec - start.tv_sec) + end.tv_usec - start.tv_usec);
```

同样测试在 TCP 和 UDP 下的情况下的数据。对于这些测试，单次的数据均具有较大的波动性，所以实际操作中用 shell 脚本控制其自动运行多次，比如 65536 次，然后做平均值处理。接着测试 20M 等文件的时间。

```
#!/bin/bash
exec 1> result30M
for((i= 1;i< 65536;i++))
do
sudo ./speedtest wlan0 file30M
done
```

上述代码中设置 shell 的标准标准输出到 result30M 文件，其中每一行的结果就是读取或者加上发送的全部时间，多次运行得到大量的结果，对这些结果通过 `awk '{total+=$1} END{print total/NR}' result30M` 得到平均值。

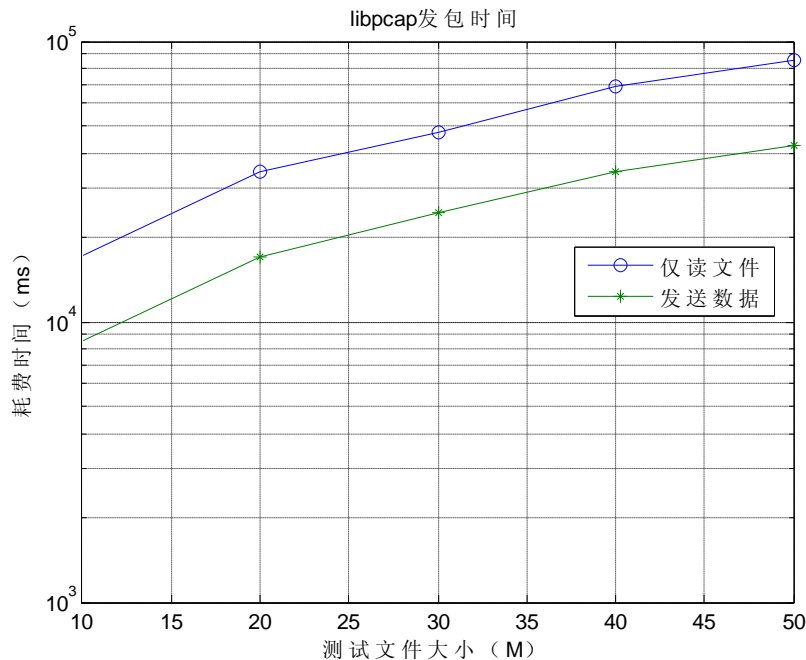


图 3-13 libpcap 发包时间

从这三个图中可观察到，libpcap 的发包时间远小于 TCP/UDP 的发包时间。首先是 TCP 发送中有 ack 确认机制，发送一个包需要对方回应，这在广播安全紧急事故的是否非常不必要。在广播安全紧急事故的时候，RSU 只需要把信息广播出去。而且是车载环境下，汽车终端节点随时变化，维持 TCP 的连接性也是不必要的。UDP 的发包时间远小于 TCP，但由于 UDP 协议的特性，仍比 libpcap 发包时间长。libpcap 的发包时间和读文件在一个数量级，而 TCP/UDP 的发包时间远长于读文件的时间。libpcap 方式中，时间更多消耗在对信息的处理。

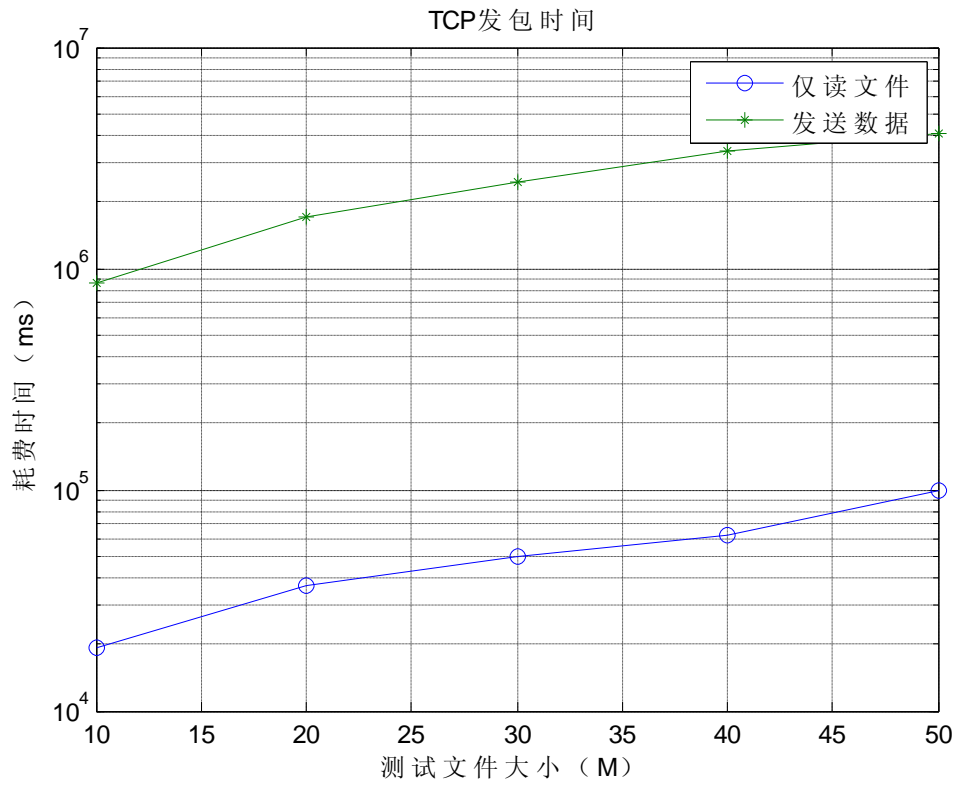


图 3-14 TCP 发包时间

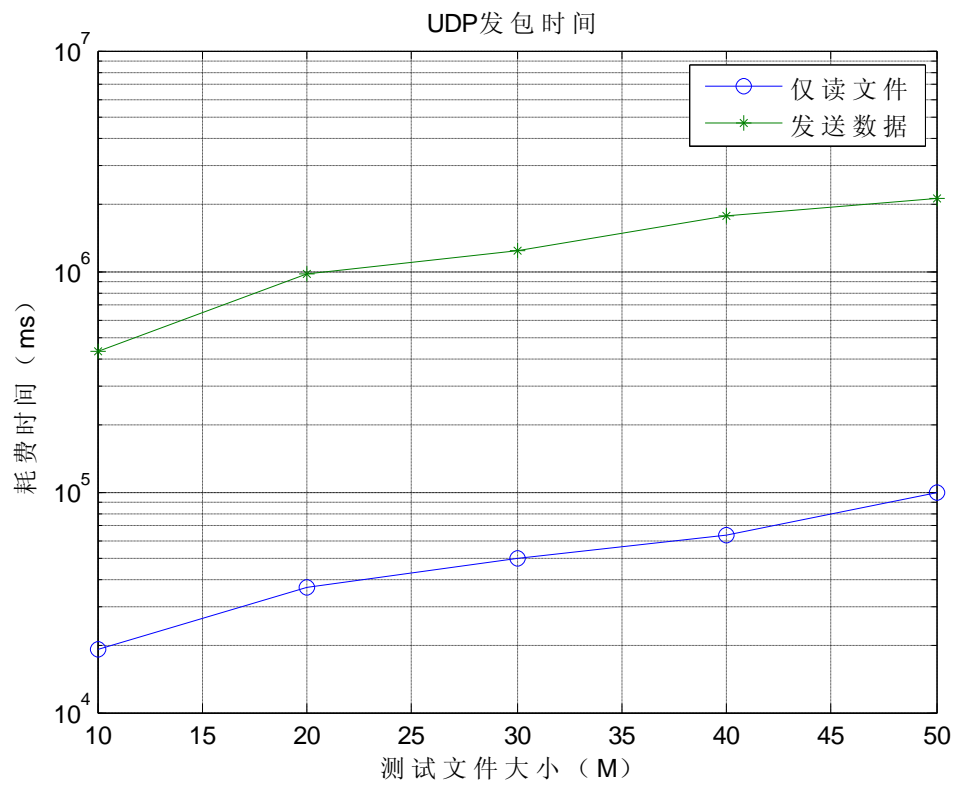


图 3-15UDP 发包时间



### 3.2.5 改进空间

上述的系统实际上还有很多可以改进的空间。

首先是 libpcap 开发包的改进。除了 libpcap 开发包之外，还有 libpcap-mmap<sup>[48]</sup>。它们两者之间的捕获数据包的结构相同。但 libpcap-mmap 采用了 MMAP 方法，建立用户进程内存到内核内存的映射，减少了从内核内存拷贝到用户进程内存的开销。

其次，基于 libpcap 的上述程序是属于应用层，应用层在发生系统调用时，会有很多消耗，导致时间和速度有所损耗，实际上可以把这些应用层模块写成内核模块，在内核中直接做好各种类型判断、有效性判断，只提供少数接口给应用，这样子运行起来将更直接更加有效。

## 3.3 本章小结

本章主要介绍双无线网卡系统的设计方案，包括硬件及软件。本设计主要思路是硬件上利用现有的 IEEE802.11 网卡，软件上使用 libpcap 开发包提供收发链路层次的数据包的能力。在这个基础上，提出了网卡抽象层和事务抽象层的模型，定义了所使用的数据帧的格式。在这个平台上可以对车联网业务进行开发和验证。同时测试对比了 libpcap 和 TCP/UDP 方式的速度差异，验证了 libpcap 方式比 TCP/UDP 更适合紧急时间广播的优越性。其中，针对这种双网卡工作方式发表了专利《一种无线网卡的通讯方法》。

## 第四章 基于 TDMA 的媒体接入控制设计

无线网络中包含许多的无线终端节点，空间中传输的无线电波则是无线网络的媒介。由于无线电波的特殊性，在某个频率上同一时刻只能有一个无线终端发射无线电信号，其他的终端接收信号。各个无线终端在竞争无线媒介发射无线电信号时，需要遵循一定的媒体访问控制方法 MAC。无线网络 MAC 协议的设计主要有两大类：基于竞争的随机媒体访问和基于调度的媒体访问。

第一类就是常见的载波侦听多点接入/避免冲撞 (CSMA/CA)。这种协议采用抢占信道的方式，IEEE802.11 系列都采用了这种方式。当节点需要发送数据使用无线媒介发射无线电波时，通过竞争方式使用无线信道，如果在发射无线电波时检测到有其他的无线终端在使用无线媒介，发生了碰撞，退避等待时机发送，直到成功或者放弃，放弃时报告给上层协议，由上层再决定重传。这种方式的主要好处是当网络中的节点数目较少时，无线信道利用率较高，同时也无需节点时钟同步，算法设计简单。缺点是当网络中节点较多且网络中的数据流量较大时，节点竞争信道更加频繁，发生碰撞的概率也较大，容易导致节点发包失败的概率越大。

上一章的双网卡系统解决了多信道协调问题，但任一网卡在抢占和使用信道时是使用了基于竞争的 CSMA/CA 方法，仍然面临着上述的缺点，还有相当大的改进空间，因此提出了 TDMA 的媒体访问信道接入理论研究方法。

### 4.1 TDMA 概述

基于调度的媒体访问控制协议主要是采用时分复用 TDMA (Time Division Multiple Access) 的方法，按照事先约定的策略依序给节点分配一定的时槽 (slot)，用于发送数据，特定的时槽只允许特定的节点发送数据，而其他节点不能发送数据，只能接听。这种方式少了 CSMA/CA 的信道冲突问题。非自己时槽时，无线终端节点除了接听之外，还可以转入睡眠状态以节省电源消耗，等到约定时隙再苏醒过来。TDMA 机制很适合于那种节点数量庞大，网络中数据流量大的场景，这种场景下使用 CSMA/CA 竞争会较为严重。缺点是 TDMA 机制需要无线终端节点之间的严格时隙同步，时隙分配算法有一定的复杂度。时隙同步一般可通过 GPS 来解决，或者使用广播帧的方式。TDMA 一般主要是研究时隙分配算法。

信道接入协议有很多种分类。同步信道协议是指所有终端节点在时间上是同步的；异步信道接入协议是基于竞争模式的。收方驱动协议是收方通知发方自己已经准备好接收数据；发方驱动是发方通知收方自己又有数据需要发送。单信道信道接入协议是所有的信号在同一个信道上传输；双信道信道接入协议有两个共享信道，分别为控制信道和数据信道；多信道信道接入协议是有多个信道，相邻可使用不同信道同时进行通信。固定分配类协议按照网络中最大节点数量做出传输时间安排，为每个用户分配确定的信道；动态分配类协议使用本地参数来分配时隙，根据时隙分配表的产生过程与拓扑结构的依赖关系可分为拓扑非透明算法、拓扑透明算法和拓扑半透明算法。

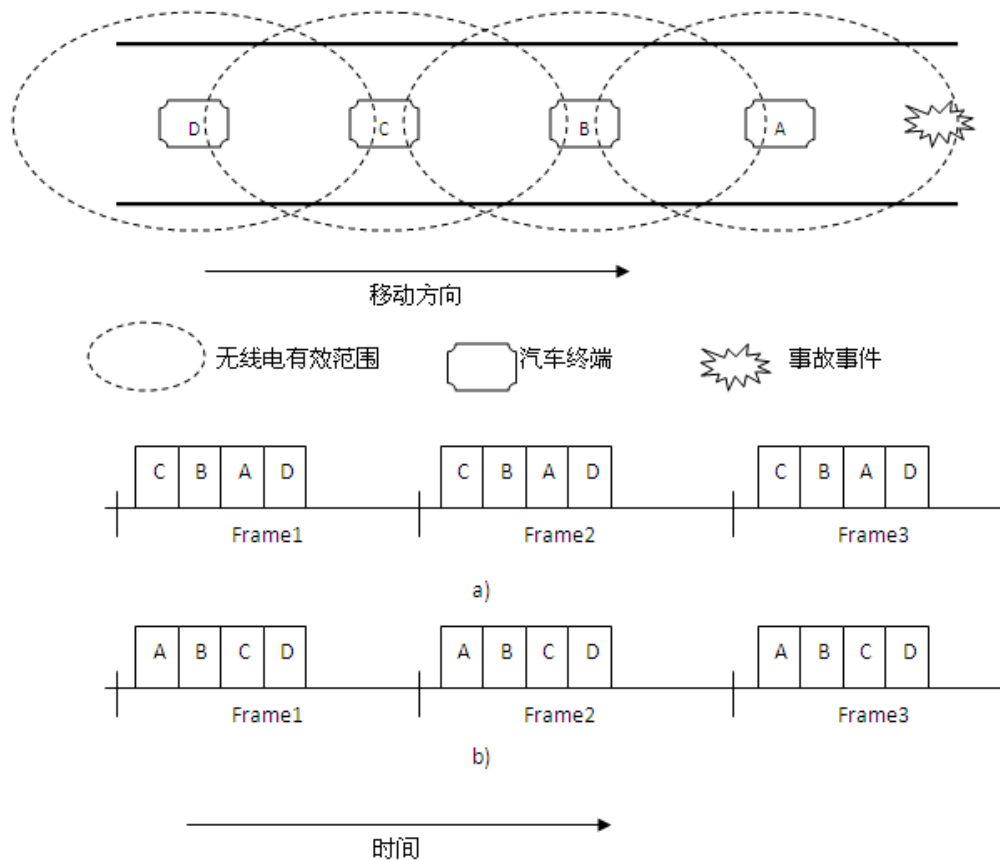


图 4-1 不同时隙安排图

如图 4-1 所示，车 A 的前面发生了紧急事件，这个事件需要通过信息传播给后续车辆 B、C、D。按照 a) 的时隙安排，Frame1 中，C 和 B 发送的数据不包含紧急事件，A 发送后 B 才收到紧急事件，而 C、D 不在 A 的有效传输范围内而无法感知到紧急事件。接着在 Frame2 中，C 发送的数据中仍没有紧急事件，而 B 发送的数据中有，相当于在 Frame2 中讲紧急事件传输给 C 了。最后再 Frame3 中，C 发送了数据才将紧急事件通知给了 D。

而在 b) 中情况很不一样。A 发送了数据，告知 B；轮到 B 发送的时隙时，B 已经知道了事件，所以可以发送给 C；C 发送时也知道了事件，所以可以直接告知 D 发生了紧急事件。整个过程仅在 Frame1 即可完成。

从这个例子可看出，不同的时隙分配算法会有不同的效果<sup>[49]</sup>。由于车载网中，车载终端做为无线终端，数目较多，因此研究 TDMA 方式是很有必要的。在车载环境中，常见的情境有自组织网络，是由多个车载无线终端组成的无中心、多跳的临时自治系统。这种网络无需基础设施，可快速组网。一般情况下每个节点只能感知其一跳范围内的邻居节点的信息，存在明显的节点隐藏和暴露节点问题。

如果有两个终端节点同时使用同一时隙，那么会导致冲突发生。时隙访问如何尽量避免冲突是一个重要的问题。于是本次论文提出了位置信息辅助的多路访问方案。

## 4.2 位置信息辅助的多路访问方案

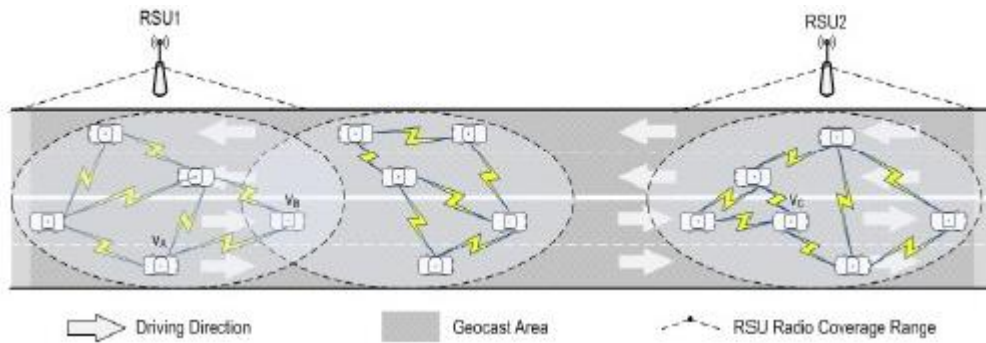


图 4-2 位置信息辅助的多路访问方案情景图

在这里假设时间带宽被分为  $M$  个独立正交的时隙，每个时隙只能分配给一个车载终端节点使用。多个车载终端节点占用同一时隙则视为冲突，将没有一个终端节点能成功收发数据。所有车载终端节点通过 GPS 或其他途径来获得地理位置。一般来说，每个车载终端节点的位置是独立的，和其他的车载终端节点位置互不干扰互不重叠的。

由于车辆地理信息的唯一性，因此可以根据汽车终端节点的地理位置来分配时隙。一个位置-信道的映射算法用来帮助汽车终端节点选择时隙。

如图 4-3 所示，将汽车终端节点地理位置抽象成坐标，分为两维，一维是汽车的行走方向，即  $x$  轴方向，另一维是垂直车行走方向，即  $y$  轴方向。最大值  $K$  则由道路匝数决定。普遍来说，上述坐标的形成过程分为三部分：

1) 一个主汽车终端节点要发送数据，在它的无线通信范围内广播。

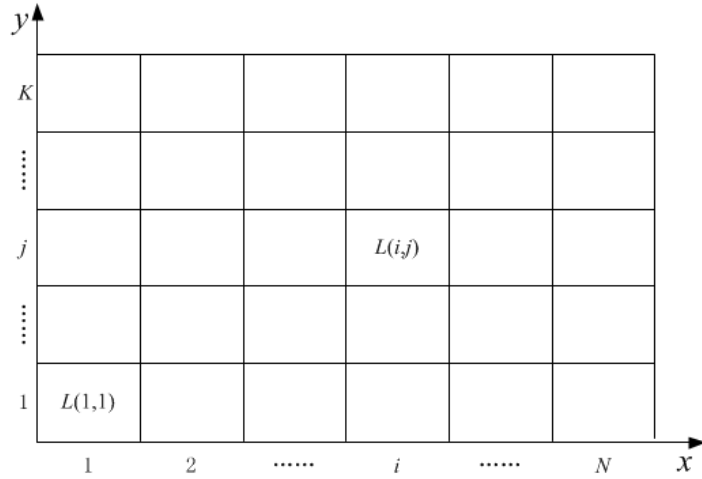


图 4-3 位置区域

2) 其他汽车终端节点收到广播的数据时，会回应它们的地理位置信息给主汽车终端节点。

3) 主汽车终端节点根据收到的地理位置信息来编号，然后再把编号信息发送出去。

在这里，其他汽车终端节点在回应地理位置信息时，它们是什么时候发送，这个问题在这里简单描述一下。一般时隙划分后，前面大部分是分配给各个终端节点的，而后面会有小部分是公共时隙，公共时隙是给新增或新加入节点随机竞争访问的。一开始网络在形成时，数目还是比较小，这种情况下，几个节点在公共时隙上随机竞争，一般很快就能被分配到前面的专有时隙。随着网络的节点增多，这些节点也是慢慢地通过公共时隙获得发言权，然后被分配到专有时隙。而且，在车载网络环境下，汽车终端节点拓扑的变化，远远小于终端节点的通讯速率，可能几分钟才有一辆车载终端节点加进来，而一个完整时隙轮完一遍，可能时间也就在毫秒级别，相比之下，车载终端节点拓扑的变化，可视为缓慢而稳定，在某种程度上，可暂时忽略终端节点拓扑的变化。

一般来说，时隙数目  $M$  是小于  $K*N$  的，所以多个终端节点可能会访问同一时隙，导致冲突发生。位置信息辅助的多路访问方案使用了地理位置信息来帮助终端节点有效地分配时隙。在这个方案中，一个终端节点会以不同的概率来访问时隙，而非随机访问。有以下规则：

定义了一个专用的函数来将汽车终端地理信息  $L(i,j)$  映射到一个随机变量  $Y$ ，随机变量  $Y$  表示被划分的正交的  $M$  个时隙的序列号。函数可以表示为：

$$Y^k(e) = f^k[L(i, j)]; k = 1, 2, \dots, K * N$$

其中,  $Y^k$  表示由  $k$ -th 地理位置所映射的随机变量序列号,  $f^k$  表示  $k$ -th 位置映射用到的函数,  $e$  是随机变量里的一个未知数。  $L(i,j)$  表示车载终端节点的位置。  $K$  表示道路匝道数目,  $N$  表示区域里边位置的数目。

每个随机变量  $Y$  满足一定的概率函数:

$$P\{Y^k = m\} = p_m^k, m = 1, 2, \dots, M$$

$$\sum_{m=1}^M p_m^k = 1;$$

其中,  $M$  表示被划分的时隙数目。  $p_m$  表示汽车终端节点访问  $m$ -th 时隙的概率。

3) 为了满足分布式时隙访问的要求, 位置信息和概率分布必须满足以下条件:

A) 对于任意区域, 映射是唯一。换句话说, 只要区域已形成, 位置就已经确定了, 映射也就同时确定了。

B) 映射只依赖于概率分布和汽车终端节点的相对位置, 与通过 GPS 获得的绝对位置无关。

4) 如果不同位置的车载终端节点访问相同的时隙, 那么冲突发生, 在这种情况下认为没有车载终端节点能成功收发数据。这种情况可以用表达式表达:

$$y^i = y^j; i \neq j$$

其中,  $y^i$  和  $y^j$  是随机变量  $Y^i$ 、 $Y^j$  对应的值。

## 4.3 冲突概率和吞吐量

### 4.3.1 冲突概率

不失一般性, 这里假设情景发生在只有一个车道, 即  $K=1$ 。区域长度  $L$  则取决于能保证该区域中能正常与其他车辆通讯的有效距离。我们假设区域中每个位置都有一个唯一的索引值, 比如  $1, 2, \dots, N$ 。同时, 任意车辆出现在每个位置的概率是相等, 用  $p_e$  表示。

车载终端节点在位置  $N$  处访问第  $M$ -th 的概率用  $p_{nm}$ 。例如图 4-4  $p_{11}$ 、 $p_{31}$ 、 $p_{nm}$ 。

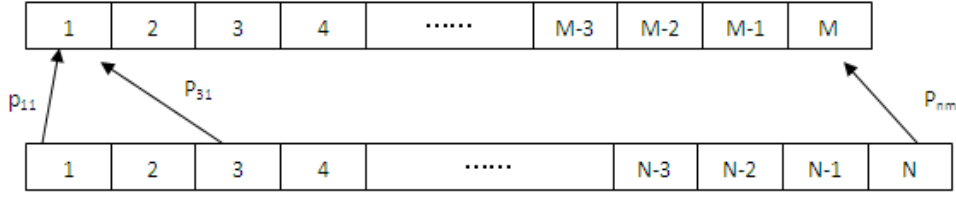


图 4-4 时隙划分和车载终端节点访问

基于以上假设，这里总共会出现  $N \times M$  个未知参数， $p_{ij} (i = 1, 2, \dots, N; j = 1, 2, \dots, M)$ ，我们使用了遗传算法来优化这  $N \times M$  个参数，优化目标是降低这个区域中平均冲突可能性。这个可能性的优化值会被当做车载终端节点访问时隙的基准。换句话说，当车载终端节点需要占用时隙来收发数据时，它必须根据  $p_{ij}$  来决定占用哪个时隙。这里我们和一般随机访问方案作对比。为了两种方案做对比，必须计算一些性能指标来评估对比。我们可以计算它们对应的平均冲突概率和吞吐量。两种方案的平均冲突概率和吞吐量具体的计算方式如下：

位置信息辅助的多路访问方案的平均冲突概率：

按照上述提到的，车载终端节点出现在任何一个地理位置的概率是相等的，在这个方案中，车载终端节点在每个位置中访问不同的时隙的概率是不同的。车载终端节点在访问时隙时，有可能会遇到其他车载终端节点访问同一时隙而引发的冲突。首先，我们计算车载终端节点在每个位置中访问时隙的概率分布，然后把这些加起来得到总的概率分布，最后除以  $N$ ，就得到了平均的概率分布。为了得到平均概率分布的表达式，我们首先计算一个车载终端节点在一个地理位置的概率。在这个位置中，一个车载终端节点可能随机访问一个时隙。如果该车载终端节点访问的时隙也被其他车载终端节点访问，那么冲突发生。车载终端节点在一个地理位置 1 的冲突概率模型可表示为：

$$P_{c1} = 1 - P_{nc1}, \quad \text{式 1}$$

其中， $P_{c1}$  表示车载终端节点在一定位置 1 访问时隙时的冲突概率， $P_{nc1}$  表示车载终端节点在第一个位置访问时隙时非冲突的概率， $N$  表示讨论区域中位置的数目， $M$  表示时隙被划分的数目， $P_e$  表示一个车载终端节点出现在某个位置的概率。

式 2 可表示为

$$\begin{aligned}
 P_{nc1} &= p_{11}[(1-p) + p(1-p_{21})][(1-p) + p(1-p_{31})] \cdots [(1-p) + p(1-p_{N1})] \\
 &+ p_{12}[(1-p) + p(1-p_{22})][(1-p) + p(1-p_{32})] \cdots [(1-p) + p(1-p_{N2})] \\
 &+ p_{13}[(1-p) + p(1-p_{23})][(1-p) + p(1-p_{33})] \cdots [(1-p) + p(1-p_{N3})] \\
 &+ \cdots \\
 &+ p_{1M}[(1-p) + p(1-p_{2M})][(1-p) + p(1-p_{3M})] \cdots [(1-p) + p(1-p_{NM})] \\
 &= p_{11}(1-pp_{21})(1-pp_{31})(1-pp_{41}) \cdots (1-pp_{N1}) \\
 &+ p_{12}(1-pp_{22})(1-pp_{32})(1-pp_{42}) \cdots (1-pp_{N2}) \\
 &+ p_{13}(1-pp_{23})(1-pp_{33})(1-pp_{43}) \cdots (1-pp_{N3}) \\
 &+ \cdots \\
 &+ p_{1M}(1-pp_{2M})(1-pp_{3M})(1-pp_{4M}) \cdots (1-pp_{NM})
 \end{aligned}$$

式 2

$$P_{nc1} = \sum_{i=1}^M p_{1i} \prod_{j=2}^N (1 - p_e p_{ji}); \text{ 式 3}$$

根据上述的表述，车载终端节点在所有位置的总的冲突概率表达式可表示为：

$$P_{tc} = 1 - P_{inc}, \quad \text{式 4}$$

$$P_{inc} = \sum_{k=1}^N P_{nck} = \sum_{k=1}^N \sum_{i=1}^M p_{ki} \prod_{\substack{j=1 \\ j \neq i}}^N (1 - p_e p_{ji}), \quad \text{式 5}$$

$$P_{aver-c} = \frac{1}{N} P_{tc} = \frac{1}{N} (1 - P_{inc}); \quad \text{式 6}$$

其中， $P_{tc}$  表示讨论区域中所有车辆访问时隙时总的冲突概率， $P_{inc}$  表示讨论区域中所有车辆访问时隙时非冲突的概率， $P_{aver-c}$  代表讨论区域中所有车辆访问时隙时的平均冲突概率。

随机访问方案的平均冲突概率：

随机访问方案是一种很简单的车载终端节点访问时隙的方案。在这个方案中，所有的车载终端节点访问时隙的概率都是相等的，即  $1/M$ 。这种方案下车载终端节点访问时隙的平均概率等于车载终端节点在任何一个位置访问时隙的冲突概率。因此，平均冲突概率模型为：

$$\begin{aligned}
 P_{aver-c} &= 1 - [(1 - p_e) + p_e (1 - \frac{1}{M})]^{N-1} \\
 &= 1 - (1 - p_e \frac{1}{M})^{N-1};
 \end{aligned} \quad \text{式 7}$$



### 4.3.2 吞吐量

为了进一步比较这两种方案,我们提出了吞吐量这个性能指标。这里有些前提假设,首先是时隙条件状况是稳定和统一的。第二,吞吐量是通过车载终端成功占用某一时隙平均传送的消息来估计的。基于以上假设,我们这里可以得到吞吐量的表达式:

$$T_{aver} = \frac{C}{M}(1 - P_{aver-c}); \quad \text{式 8}$$

其中,  $T_{aver}$  表示时隙的平均吞吐量,  $C$  表示时隙带宽容量,  $M$  表示时隙的数目,  $P_{aver-c}$  表示一个车载终端节点访问时隙时的冲突概率。

### 4.4 建模求解

根据式 5 和式 6, 我们可以看到平均冲突概率是依赖于位置-时隙的映射。我们应该优化位置-时隙映射的参数来获得最小的平均冲突概率。优化问题的目标函数和约束函数可以改写成:

目标函数

$$P_{tc} = 1 - P_{inc} = 1 - \sum_{k=1}^N \sum_{i=1}^M p_{ki} \prod_{\substack{j=1 \\ j \neq i}}^N (1 - p_e p_{ji}); \quad \text{式 9}$$

(2) 约束条件

$$s.t. \begin{cases} \sum_{j=1}^M p_{ij} = 1 & i = 1, 2, \dots, N \\ 0 \leq p_{ij} \leq 1 & j = 1, 2, \dots, M \end{cases} \quad (M < N) \quad \text{式 10}$$

位置信息辅助的多路访问方案中的总冲突概率表达式中含有多个未知变量、高阶、非线性的特征。这些特征使得优化变得更加困难。同时, 常见的优化解决方法无法达到这个模型的优化目标。遗传算法优化是一个以自然选择进化论的自适应、启发式的算法, 在一定搜索范围内利用智能随机搜索来解决优化问题, 能搜索到一个近乎全局优化的解法。遗传算法的基本思想是基于达尔文的适者生存的原则。按照这个原则, 遗传算法分为以下步骤<sup>[50][51]</sup>:

- a) 初始化: 进化代数初始值为  $gen=0$ , 确定最大进化代数  $MAXGEN$ , 随机生成  $M$  个个体作为初始群体  $P(0)$ 。
- b) 个体评价: 按照一定的适应度函数计算群体  $P(gen)$  的适应度。

c)选择运算: 在群体适应度评价基础上, 按照一定选择算法, 选出优质的个体遗传到下一代, 适应度评价值高地有比较大的机会较大。

d)交叉运算: 按照一定的算法和概率交叉某两个个体之间的染色体, 从而得到新的个体。

e)变异运算: 单个个体按照一点算法概率进行随机编译。

群体  $P(\text{gen})$  经过选择、交叉、变异运算之后得到下一代群体  $P(\text{gen}+1)$ 。

f)终止条件: 若  $\text{gen}=\text{MAXGEN}$ , 则表明已经达到设定的进化代次数了, 这时将可以终止进化和计算, 把最大适应度评价值的个体输出, 作为遗传算法的最优解。

通过上述的遗传算法的迭代, 将获得一组个体, 使得总的冲突概率获得最小值。在上述过程中, 将获得总冲突概率函数的各个参数的值, 这些值使得函数达到近乎全局优化。

在这次 matlab 仿真中, 使用了雷英杰的 genetic 遗传算法库<sup>[52]</sup>, 定义了群体中个体数目 (Number of individuals) 为  $\text{NIND}=80$ , 变量数目  $\text{NVAR}$  则由时隙数目和位置数目乘积决定, 最大遗传代数 (Maximum number of generations) 为  $\text{MAXGEN}=4000$ , 变量的二进制位数 (Precision of variables) 为  $\text{PRECI}=20$ , 代沟 (Generation gap) 为  $\text{GGAP}=0.9$ 。仿真程序的主体框架为:

```
Chrom = crtpb(NIND, NVAR*PRECI); %初始种群
gen=0; %代计数器
trace = zeros(2, MAXGEN);
X = bs2rv(Chrom, FieldD); %计算初始种群的十进制转换 X(NIND,NVAR);
ObjV = objV_calc(X, M, N, p); %计算初始种群的目标函数值
while gen < MAXGEN
    FitnV = ranking(ObjV); %分配适应度值 (Assign fitness values)
    SelCh = select('sus', Chrom, FitnV, GGAP); %选择
    SelCh = recomb('xovsp', SelCh, 0.7); %交叉
    SelCh = mut(SelCh); %变异
    X = bs2rv(SelCh, FieldD); %子代个体的十进制转换
    ObjVSel = objV_calc(X, M, N, p); %计算子代的目标函数值
    [Chrom ObjV] = reins(Chrom, SelCh, 1, 1, ObjV, ObjVSel); %重插入子代的新种
    gen = gen + 1; %代计数器增加
%输出最优解及其序号, 并在目标函数图像中标出, Y 为最优解, I 为种群的序号
```

```
[Y,I]= min(ObjVSel);
trace(1, gen) = min(ObjV); %遗传算法性能跟踪
trace(2, gen) = sum(ObjV) / length(ObjV);
end
```

在这个遗传算法库中，创建种群的方法有 **ctrbase** 创建基向量、**ctrbp** 创建任意离散随机种群和 **crtrp** 创建实值初始种群，在这里使用了 **crtbp** 函数创建了任意离散随机种群。

适应度计算中提供了 **ranking** 基于秩的适应度计算和 **scaling** 比率适应度计算，在这里使用了 **ranking** 基于秩的适应度计算函数。

选择函数中提供了 **reins** 一致随机和基于适应度的冲插入、**rws** 轮盘选择、**select** 高级选择例程和 **sus** 随机遍历采样，在这里使用了 **sus** 随机遍历采样。

交叉算子中提供了较多的选择，有 **recdis** 离散重组、**recint** 中间重组、**reclin** 线性重组等，在这里使用过了 **xovsp** 单点交叉算子。

变异算子有 **mut** 离散变异、**mutate** 高级变异和 **mutbga** 实值变异，在这里使用过了 **mut** 离散变异。

适应度的计算过程为：

% 把随机生成的概率值单位化，使其满足：  $pn_1+pn_2+...+pn_M = 1, n = 1...N$ ;

```
for row = 1:NIND
    for n = 1:N
        for m = 1:M
            sum = sum + X( row, (n-1)*M + m);
        end
        for m = 1:M
            Prob_Mtrx( row, (n-1)*M + m ) = X( row, (n-1)*M + m ) / sum;
        end
        sum = 0;
    end
end
% 计算个体的目标函数值
for i = 1:NIND
```

```
objV(i,1) = dest_fun(Prob_Mtrx(i,:), M, N, p);
end
```

## 4.5 仿真结果

在这里，将比较两种不同方案的性能。在仿真中，泊松点过程用来描述车载终端节点的位置分布。车载拓扑变化的频率远远低于无线通信的速率。因此我们可以在一个快照中观察到分布。假设车载在这个区域中随机分布了。每个车载在任何时刻都有一个唯一的位置信息（即：不会发生位置信息重叠现象），车载出现在一个地理位置的概率是一个常数（即  $p_e$ ）。同时假设车载之间的连接是由车载之间的距离。为了改进无线时隙的利用率，时隙的划分数目  $M$  应该尽可能小。但是，越小的  $M$  值将带来更大的平均冲突。当吞吐量达到最大值，时隙数目  $M$  就是最优值。由于实际上车辆之间要保持一定的安全距离，所以设置了每辆车的长度为 10 米。同时，为了保证车与车之间能通讯，设置了区域长度为 500 米，这是 IEEE802.11p 的经典通讯距离。参考 DSRC，设置了信道的通信容量  $C$  为 20Mbps。所以，位置的数目  $N$  为 50。 $p_e$  值的变化，会很大地影响到平均概率。这里挑选了两个值。小的  $p_e$  值代表了道路上比较少的车，大的  $p_e$  值代表了车路上比较拥挤。在这仿真中，分别设置为 0.3 和 0.7。在遗传优化算法的过程中，使用了格雷码编码方法来实现参数编码，一个参数使用了 20bit 来表示。还有，我们选择了总冲突概率函数最为最适应函数，基于排序方法来赋予最佳值。交叉和变异的概率分别设置为 0.7 和 0.035。

首先，我们先设置车载终端节点出现在一个位置的概率为  $p_e = 0.3$ 。我们观察时隙划分数目  $M$  的变化时，时隙的平均吞吐量和平均冲突概率的变化。从图 5 中，我们可以看到，在随机访问方案中，当  $M$  取 15 的时候，平均吞吐量达到最大值。而在位置信息辅助的多路访问方案中，平均吞吐量要高出 16%。

从另一方面，从图 4-6 我们可以观察到位置信息辅助的多路访问方案中平均冲突概率总是低于随机访问方案。特别是，在吞吐量最大值（即  $M$  取 17 时）， $P_{avr-c}$  相比降低了 13.7%，同时，位置信息辅助的多路访问方案的吞吐量比随机方案高出 19%。

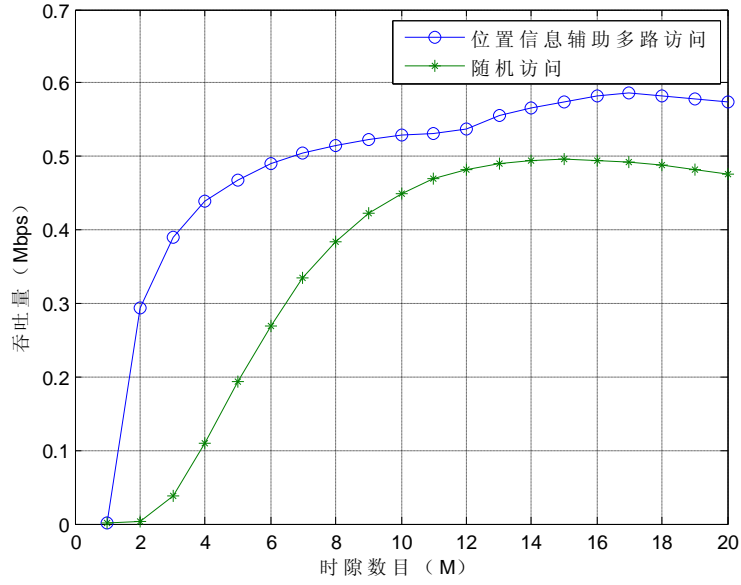


图 4-5 吞吐量仿真图 ( $N=50$ ,  $p_e = 0.3$ )

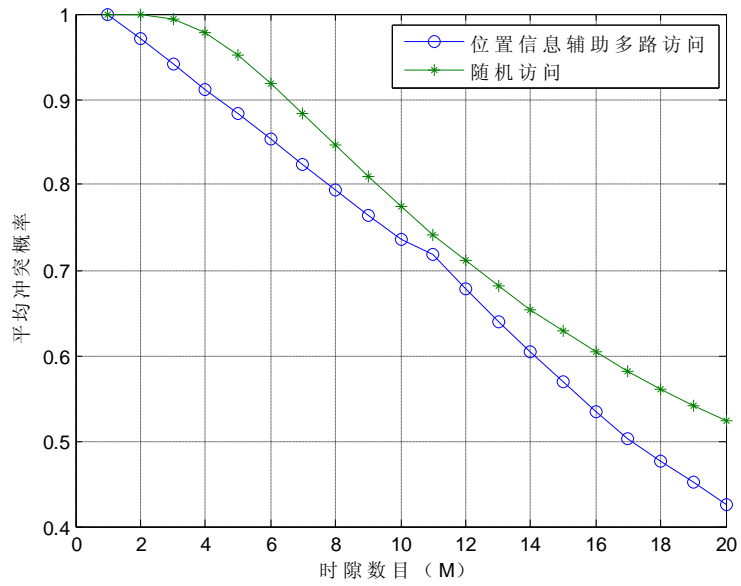
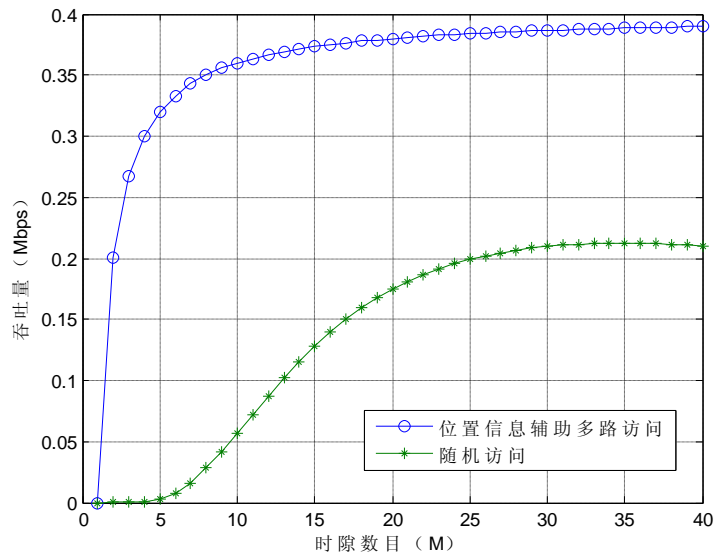
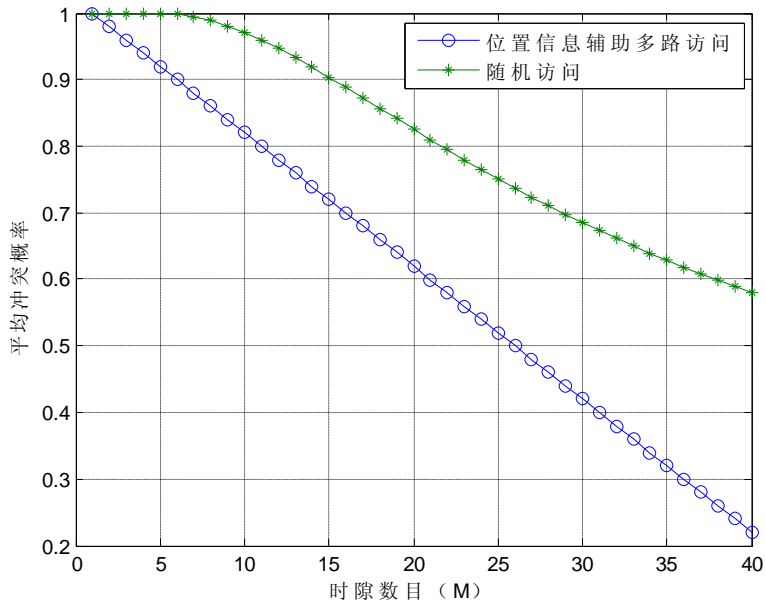


图 4-6 平均冲突概率图 ( $N=50$ ,  $p_e = 0.3$ )

其次，我们修改了车载出现在一个位置的概率（即  $p_e = 0.7$ ），然后观察随着时隙数目  $M$  的变化，平均吞吐量和平均冲突概率的变化。

从图 4-7 看到，随机访问方案中，最大的平均吞吐量出现在  $M$  取 35 的时候。在这个地方，位置信息辅助的多路访问方案的平均吞吐量高出了 83%。两个方案的最大值均小于  $p_e = 0.3$  时的情况。

图 4-7 吞吐量仿真图 ( $N=50$ ,  $p_e = 0.7$ )图 4-8 平均冲突概率图 ( $N=50$ ,  $p_e = 0.7$ )

同时，从图 4-8 看到，位置信息辅助的多路访问方案中平均冲突概率仍然总是低于随机访问方案。两个方案的平均冲突概率下降均  $p_e = 0.3$  时的情况。

随着  $M$  的增长，平均冲突概率的降低就越明显。随着  $p_e$  的增大，说明车况越来越拥堵，平均冲突概率增大，平均吞吐量下降。

## 4.6 本章小结

在这里，提出了位置信息辅助的多路访问方案。在这个方案中，每个车载终端节点是根据位置-时隙映射表来选择时隙的，平均冲突概率分布降低了很多，平均吞吐量得到明显的提高。通过 `matlab` 遗传算法仿真结果表明，位置信息辅助的多路访问方案的平均吞吐量优于随机访问方案，随着时隙数目  $M$  的变化，其平均吞吐量和平均冲突概率也有不同的变化。

## 总结与展望

当前, 世界各国研究机构和厂商都在开展有关车联网系统的研究。车联网系统是未来发展的一个趋势, 是促进物联网发展的重要方面, 它将改变人类的生活和生产。因此, 研究车联网, 对于智能交通、智能城市的发展等具有非常重要的意义。

IEEE802.11p 和 IEEE1609 协议族做为车联网的主要通信协议栈, 正在逐步完善。虽然要在实际中大规模应用和部署还有很长的路要走, 更多的研究工作要完成, 但是这也是一个发展的机会。

本文在学习和理解 IEEE802.11p 和 IEEE1609 协议族情况下, 重点抓住多信道协调机制, 在现有硬件条件尚不完善的情况下, 使用 Tiny6410 开发板和 RT3070 无线网卡, 移植 libpcap 库, 利用 libpcap 开发框架实现了一个双无线网卡通信平台系统, 取得良好的效果, 为车联网应用系统提供了一个强有力和可靠的支撑系统, 取得性能上的优化。同时针对双网卡中任一网卡在抢占信道上使用 CSMA/CA 所面临的缺点, 文章最后提出位置信息辅助的多路访问 TDMA 理论方案, 给出平均冲突概率和吞吐量两个性能指标, 使用了遗传算法求解优化结果, 通过仿真验证了所提方案在两个性能指标方面的优越性。

当然, 本文所做的工作有限, 车联网是一个庞大的系统, 需要投入更多的研究, 系统还需更多完善:

(1) 为了提高车载的通讯能力, 需要选用专门针对车辆网的无线网卡, 在硬件上实现相关的处理, 然后把信道协调等功能实现在内核中, 提高效率, 抽象化接口, 简化上层开发应用人员开发工作。

(2) 本文没有对路由方面做相关研究, 而实际应用中, 路由算法也是一个很关键的方面, 一个好的路由算法, 能更快地做出一个更优的路由选择, 减少车载中的不必要的路由和转发。

(3) 位置信息辅助的多路访问方案中, 需要考虑一个更实际的车辆的随机分布模型, 同时, 无线电信号的衰减和衰弱模型也必须考虑进去。

总之, 本论文研究了车联网应用的一部分, 为整个车联网的研究和实现提供了一定的参考, 未来可以此基础上继续完善和丰富, 最终实现车联网, 实现智能交通。



## 参考文献

- [1] 李政裔. 浅析信息技术在物流教学过程中的作用[J].消费导刊.2013,09.
- [2] 百度百科.物联网. [EB/OL]. <http://baike.baidu.com/view/1136308.htm>
- [3] 谢鹏.基于 WAVE 协议栈的车联网通信终端研究及定位系统实现[D].华南理工大学.2013
- [4] 邱团准. 基于 SOA 的车联网应用系统设计与实现[D]. 华南理工大学, 2013
- [5] 宋俊德. 浅谈物联网的现状和未来[J]. 移动通信. 2010,15
- [6] 孙其博,刘杰,黎彝,范春晓,孙娟娟. 物联网:概念、架构与关键技术研究综述[J].北京邮电大学学报.2010,03
- [7] 百度百科.智能交通系统[EB/OL]. <http://baike.baidu.com/view/1488750.htm>
- [8] 王笑京,沈鸿飞,汪林. 中国智能交通系统发展战略研究[J]. 交通运输系统工程与信息. 2006,04
- [9] 马芸芸. 基于 802.11 p 协议的平台实现. [D]. 北京邮电大学, 2013
- [10] 王建强, 吴辰文,李晓军. 车联网架构与关键技术研究[J].微计算机信息.2011,04
- [11] Grafing, Sebastian, Petri Mahonen, Janne Riihijarvi. Performance evaluation of IEEE 1609 WAVE and IEEE 802.11 p for vehicular communications. Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on. IEEE, 2010.
- [12] 常促宇,向勇,史美林.车载自组网的现状与发展.[J]. 通信学报.2007,11
- [13] 刘富强,项雪琴,邱冬.车载通信 DSRC 技术和通信机制研究.[J].上海汽车.2007,08
- [14] 迟铁军,高鹏.国外智能交通系统发展状况分析及对我国的启示.[J].黑龙江交通科技.2009,02
- [15] 美国 ITS 的发展历程与现状[EB/OL].  
<http://www.itschina.org/article.asp?articleid=2143>.
- [16] 张丽. 基于小波变换的交通图像预处理与特征提取. [D]. 河北大学, 2007
- [17] 曹颖荣,林小玲. IEEE 802.11p 无线车载自组网络协议的性能分析与模拟[J]. 仪表技术, 2011,02
- [18] 杜英田. 基于 IEEE802.11p/1609 协议的智能交通无线车载通信协议优化研究[D].北京邮电大学, 2011

- [19] Architectures, Challenges, Standards and Solutions[J]. Communications Surveys & Tutorials, IEEE, Fourth Quarter 2011: 584-616
- [20] 百度百科. OSI 模型[EB/OL]. <http://baike.baidu.com/view/571842.htm>
- [21] 东南大学. 802.11 无线网络[M]. 东南大学出版社, 2006
- [22] 百度百科. IEEE802[EB/OL]. <http://baike.baidu.com/view/126269.htm>
- [23] 唐波. WAVE 架构及相关协议设计与实现[D]. 华东理工大学, 2009
- [24] 谢仕云. 无线局域网视频流组播自适应机制的研究. [D]. 电子科技大学, 2011
- [25] 常亚楠. 无线 mesh 网络中基于功率控制和定向天线的低时延广播研究. [D]. 武汉大学, 2013
- [26] S. Eichler. Performance evaluation of the IEEE 802.11p WAVE communication standard[C]. Proc. IEEE Vehicular Technology Conf., Baltimore, MD, US, Oct. 2007: 2199-2203
- [27] Karagiannis G.. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions[J]. Communications Surveys & Tutorials, IEEE, Fourth Quarter 2011: 584-616
- [28] IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE) [EB/OL]. [http://www.standards.its.dot.gov/fact\\_sheet.asp?f=80](http://www.standards.its.dot.gov/fact_sheet.asp?f=80)
- [29] IEEE Std 1609.0-2012. IEEE Draft Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture. IEEE P1609.0/D5[S]. IEEE, 2012
- [30] IEEE Std 1609.1 -2006. IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)—Resource Manager. IEEE Vehicular Technology Society (VTS) [S]. October 2006
- [31] IEEE Std 1609.3-2010, IEEE Standard for Wireless Access in Vehicular Environments (WAVE)-Networking Services[S]. IEEE, 2010.
- [32] IEEE Std 1609.4-2010, IEEE Standard for Wireless Access in Vehicular Environments (WAVE)-Multi-channel Operation[S]. IEEE, 2011
- [33] 王力轩. 基于嵌入式系统工业串口集线器的设计与测试. [D]. 北京邮电大学, 2008
- [34] 冀谦祥. 基于 Linux 的嵌入式网络视频监控系统设计与实现. [D]. 电子科技大学, 2010
- [35] 余沛. 基于 Linux 的嵌入式系统软件平台设计. [D]. 华中科技大学, 2004

- [36]张静. 嵌入式 ARM9200 与 MCP2510 的通信研究. [D]. 太原科技大学,2008
- [37]马力遥. 移动数据网格的设计与实现. [D]. 电子科技大学,2006
- [38]丁弋宇. 基于嵌入式以太网的 LED 显示屏控制系统应用研究. [D]. 东南大学,2006
- [39]韦东山. 嵌入式 Linux 应用开发[M]. 人民邮电出版社,2003
- [40]俞贵涛. 旋转机械的网络化在线监测诊断系统研究. [D]. 电子科技大,2007
- [41]魏永明,耿岳,钟书毅. LINUX 设备驱动程序(译)[M]. 中国电力出版社,2006
- [42]宋浩. 无线视频监控系统及其关键问题的研究. [D]. 天津大学硕士,2011
- [43]胡海. 嵌入式 TCP/IP 协议栈研究与实现. [D]. 西南交通大学, 2005
- [44]易小林. 基于 Linux 的网络测量技术研究与实现. [D]. 河北工业大学,2010
- [45]张旭. 智能化网络安全监控系统技术研究. [D]. 北京工业大学,2012
- [46]戴维. 基于 IRC 协议的僵尸网络检测系统的实现. [D]. 电子科技大学,2010
- [47]Richard Steve. Advanced Propramming in the Unix Environment[M]. 人民邮件出版社  
2006
- [48]Csdn 博客.libpcap-mmap  
[EB/OL].<http://blog.csdn.net/aaa6695798/article/details/4451582>
- [49]Yu, Fan, and Subir Biswas. Self-configuring TDMA protocols for enhancing vehicle  
safety with DSRC based vehicle-to-vehicle communications. IEEE Journal. 2007,25
- [50]高建兵. 基于遗传算法的模糊推理控制系统的参数优化研究. [D]. 辽宁工程技术大  
学,2010
- [51]张远春. 多品种混合装配车间关键资源建模与优化. [D] 上海交通大学,2011
- [52]雷英杰,张善文,李续武,周创明 MATLAB 遗传算法工具箱与应用[M]. 西安电子科技  
大学出版社,2005

## 攻读硕士学位期间取得的研究成果

一、已发表（包括已接受待发表）的论文，以及已投稿、或已成文打算投稿、或拟成文投稿的论文情况（只填写与学位论文内容相关的部分）：

序号	作者（全体作者，按顺序排列）	题目	发表或投稿刊物名称、级别	发表的卷期、年月、页码	相当于学位论文的哪一部分（章、节）	被索引收录情况
----	----------------	----	--------------	-------------	-------------------	---------

注：在“发表的卷期、年月、页码”栏：

- 1 如果论文已发表，请填写发表的卷期、年月、页码；
  - 2 如果论文已被接受，填写将要发表的卷期、年月；
  - 3 以上都不是，请据实填写“已投稿”，“拟投稿”。
- 不够请另加页。

二、与学位内容相关的其它成果（包括专利、著作、获奖项目等）  
在硕士研究生期间，本人以第一学生作者身份申请国家发明专利一篇：

余荣，谢胜利，陈钦波等. 一种无线网卡的通讯方法  
（发明专利，已受理，申请号：CN201210588784.5

## 致谢

白驹过隙，转瞬即逝，依稀记得刚进实验室时懵懵懂懂的我，在实验室老师们的指导下，不断地学习与成长。三年中，个人经历了各种起伏，如今的技术能力，也是集中在这三年中学成锻炼。如今三年的研究生生活即将结束，我衷心感谢所有帮助过和关心过我的人们。

首先感谢我的研究生导师吴宗泽教授，吴老师是我进入科研领域的领路人，他对科学研究的热情和孜孜不倦的研究精神大大感染了我，使得我对科研产生了兴趣。吴老师严谨认真的学术态度，矢志不渝的学术追求，创新求是，扎扎实实的学术精神，思维活跃的学术观点，营造了一种良好的学术氛围，使得置身其中的我被深深感染。在此，衷心地感谢吴老师，是吴老师您让我进入到这个充满温暖互相帮助团结的集体中。

其次，感谢实验室的余老师，他是我们项目组的指导老师，他认真负责的工作态度、专业的解决问题的能力，以及对学生的耐心指导，让我在做项目的过程中收获颇丰。余老师也不乏幽默，为人亲切，对我们学生就像学长般一样，让我感到亲切。

感谢实验室其他的各位老师，他们为我的学习和发展提供了一个很好的平台。感谢实验室的众多师兄师姐们，他们带领我入门，为我提供许多指导与帮助。

感谢实验室的同学们（Jshen，妍蓉，容姐，D神，Y神，星爷，坚子，沈大，波导，罗涛，爽霸，文龙，芳芳等）以及室友们，在学习上，大家营造了良好的学习环境和学术氛围，在生活上，大家相互关心相互帮助，正是你们，才使我的三年研究生生活收获丰富、充满精彩。与你们同学三年，我亲身感受到大牛是怎么炼成的，你们也是一直以来我的学习榜样。感谢实验室的师弟师妹们，有你们的努力付出，才有我们的功成身退。

衷心感谢我的父母兄姐，他们一直都在背后默默地支持着我，他们从来尊重我的选择，有了他们的支持，我才能安心地完成所有的大学生涯，他们的关怀和支持将是我永远的精神支柱。

“路漫漫其修远兮，吾将上下而求索”，研究生阶段的结束只是新生活的开始，以后的日子里充满未知，但是我总记得有那么一些好老师、好同学，伴我度过了难以忘怀的三年，教会了我许许多多的东西，让我能够更加从容面对未来



#### IV - 2 答辩委员会对论文的评定意见

陈钦波同学的硕士学位论文《车载环境下媒体接入控制协议优化设计》，在深入研究 WAVE 协议栈的基础上，针对多信道协调机制的特点，提出了一套双网卡媒体控制接入系统平台；同时针对单个网卡的信道抢占和使用，做了 TDMA 理论优化方案。这对于 WAVE 协议栈的研究以及车联网应用的研究都具有很好的实践指导意义。

论文的主要工作包括：

1、WAVE 协议栈研究，介绍了 WAVE 协议栈架构及其主要组成协议特点，其中重点介绍了 IEEE802.11p 和 IEEE1609.3、IEEE1609.4 协议。

2、双网卡媒体控制接入系统平台设计，即基于 WAVE 协议栈多信道协调机制，利用友善之臂 Tiny6410 和 RT3070 无线网卡，移植 Libpcap，设计相关数据帧格式和系统软件架构，实现了系统设计，最后还进行了 Libpcap 和 TCP/UDP 的性能测试和比较。

3、基于 TDMA 的媒体接入控制理论设计，提出位置辅助随机访问方案，并定义了冲突概率和吞吐量两个性能指标，利用遗传算法在 Matlab 上仿真求解性能，验证了该方案相比普通的随机访问方案有着较好的性能。

论文工作表明作者已经掌握本学科坚实宽广的基础知识，具备较强的独立从事科研工作的能力。论文结构合理，论述清楚，逻辑性强，层次清晰，试验数据较为充实，已达到硕士学位论文的要求。答辩过程中表述清楚，回答问题基本正确。答辩委员会一致同意通过陈钦波同学的硕士学位论文答辩，并建议授予工学硕士学位。

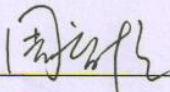
论文答辩日期：2014 年 6 月 7 日

答辩委员会委员共 5 人，到会委员 5 人

表决票数：优秀（0）票；良好（5）票；及格（0）票；不及格（0）票

表决结果（打“√”）：优秀（ ）；良好（√）；及格（ ）；不及格（ ）

决议：同意授予硕士学位（√） 不同意授予硕士学位（ ）

答辩 委员 会成 员签 名	 (主席)	