****

硕士学位论文

|  |
| --- |
| 面向车联网应用的信息发布系统的  设计与实现 |

|  |  |
| --- | --- |
| 作者姓名 | 张洁柯 |
| 学科专业 | 信号与信息处理 |
| 指导教师 | 傅予力教授 |
|  |  |
| 所在学院 | 电子与信息学院 |
| 论文提交日期 | 2014年4月 |

**Design and Implementation of Information Distribution System based on IOV**

A Dissertation Submitted for the Degree of Msater

**Candidate：Zhang Jieke**

**Supervisor：Prof. Fu Yuli**

SouthChinaUniversity of Technology

Guangzhou, China

**分类号：TP393.09学校代号：10561**

**学号：201120108588**

华南理工大学硕士学位论文

**面向车联网应用的信息发布系统的**

**设计与实现**

作者姓名：张洁柯指导教师姓名、职称：傅予力教授

申请学位级别：工学硕士学科专业名称：信号与信息处理

研究方向：通信信号处理

论文提交日期：2014年 4月25日论文答辩日期：2014年月日

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：

委员：

**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：日期：年月日

**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。学校有权保存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： 日期：

指导教师签名： 日期

作者联系电话： 电子邮箱：

联系地址(含邮编)：

摘要

车联网技术隶属于物联网技术的范畴，主要应用于智能交通系统（ITS，Intelligent Transport System）。作为智能交通系统的典型运用，车联网技术综合运用了信号与信息处理技术、消息分发技术、遥感与识别技术、数据挖掘技术、网络接入与控制技术等，把道路和车辆的各种动、静态信息通过互联网连接起来，从而实现多个维度的系统大规模、大数据的信息融合，最终起到减少道路事故、优化交通资源配置的作用。车联网典型的应用服务主要分为交通道路安全服务、交通数据采集分发服务以及娱乐信息服务三大类。但是目前基于车联网的信息发布系统普遍均存在通信质量差、传输速率低、系统并发访问差，业务单一等问题。智能交通领域上缺乏提供完善服务的车联网信息发布系统。

首先，本文研究了两种有助于提高车联网应用服务质量的关键技术，即WAVE(Wireless Access in Vehicular Environment)通信协议栈技术以及LAMP（Linux+Apache+MySQL+Python）架构的动态网络应用服务平台架设技术，并简要的分析这两种关键技术的特性。其次，在针对车联网信息数量多、数据量大并且通信时间短的特点，设计了三层架构的车联网信息发布系统的构建方案；针对车联网通信环境快速多变、通信质量差的特点，设计了以WAVE协议栈作为信息发布系统的底层通信协议的方案；针对车联网用户数量庞大，车联网网络中心并发访问需求高的特点，设计了基于LAMP架构的Web网络中心的方案。最后，再根据本文提出的系统设计方案，详细描述信息发布系统的网络中心、路侧单元RSU和车载单元OBU三个层次中各个功能子模块具体实现方法。

与传统信息发布系统相比，车联网信息发布系统摒弃了此前常用的C/S架构,使用更适合车联网环境的三层架构体系，以WAVE协议栈作为系统通信基础，采用高性能高并发量的LAMP构建Web网络中心。经测试实验证明，本文所设计的雏形系统，其点对点通信传输带宽约为12Mbits/s，服务器可满足200-500个车载单元OBU同时在线访问。该车联网信息发布系统能满足面向车联网应用的服务需求，对构建新型的智能交通具有一定的实践意义。

**关键词：**车联网；WAVE；智能交通系统；LAMP架构

Abstract

The Internet of Vehicles (IOV), which is subordinate to the Internet of Things (IOT), is generally used in the Intelligent Transportation System (ITS).In order to connect all the static and dynamic information of cars and roads via the Internet, signal and information processing technology, message distribution technology, sensing and recognition technology, data mining technology and network communication technology are synthetically applied to IOV. As a result, a large-scale, large-capacity and multiple-dimension data fusion of the system can be achieved and thus the number of road accidents reduces and the efficiency of traffic resource allocation is improved. Typical IOV services mainly divide into three categories, respectively is: traffic safety service, traffic data collection and distribution service and entertainment information service.However, there are still open issues in IOV at present, such as poor communication quality, low data rate, badconcurrent access and single business. A lack of information distribution system which can provide perfect service is still in IOV.

WAVE (Wireless Access in Vehicular Environment) communication protocol stack technology and LAMP (Linux + Apache + MySQL + Python) architecture, which are play an important role in improving theservice quality of IOV, are studied in this paper, and the analysis of their characteristic is also presented.The design of system can be divided into three steps: focus on a huge quantity information and a short communication time, a three-tier design scheme for the IOV information distribution system is proposed; focus onthe IOV rapidly changing communications environment and the poor communication quality characteristics, theWAVE stack is used as a underlying communication protocol; focus on IOV huge number of Internet users and the high concurrent access, a LAMP-based Web network centersis proposed.Based on the proposed system design scheme, the implementation of IOV information distribution system’s network center, road side unit RSU and vehicle unit OBU and their respective sub-module is investigated.

Compared with traditional information systems, IOV information distribution systemproposed in this paper discards the traditional C/S architecture and uses the three-tier system which is more suitable for the IOV environment.The WAVE communication protocol stack is used as a basis communication protocol in IOV information distribution system, while the high- performance and high-concurrency LAMP architecture is used as building Web network Center.The system was built and tested. And results show that the prototype system can achieve good performance, in which point- to-pointcommunication transmission bandwidth is about 12 Mbit/s, the server meets 200-500 onboard unit OBU online access at the same time. The IOV information distribution system, as an integral part of ITS, has a practical significance to new intelligent transportation systems.

**Keywords:** Internet of Vehicles;Wireless Access on Vehicular Environments; Intelligent Transport System; LAMPArchitecture

**目录**

[摘要 0](#_Toc385623191)

[Abstract 1](#_Toc385623192)

[第一章 绪论 6](#_Toc385623193)

[1.1. 研究背景及意义 6](#_Toc385623194)

[1.2. 国内外研究现状 8](#_Toc385623195)

[1.2.1. 智能交通系统研究现状 8](#_Toc385623196)

[1.2.2. 车联网的信息分发系统研究现状 9](#_Toc385623197)

[1.3. 系统功能 10](#_Toc385623198)

[1.4. 本文研究内容及组织架构 11](#_Toc385623199)

[第二章 信息发布系统技术研究 13](#_Toc385623200)

[2.1. 车联网通信技术 13](#_Toc385623201)

[2.1.1. WAVE技术简介 13](#_Toc385623202)

[2.1.2. WAVE协议栈研究与分析 14](#_Toc385623203)

[2.1.3. WAVE信息传输服务 17](#_Toc385623204)

[2.1.4. 车联网通信技术对比 19](#_Toc385623205)

[2.2. LAMP架构技术体系 22](#_Toc385623206)

[2.2.1. LAMP架构概述 22](#_Toc385623207)

[2.2.2. LAMP架构模型特性 22](#_Toc385623208)

[2.2.3. LAMP架构技术详析 24](#_Toc385623209)

[2.3. 本章小结 28](#_Toc385623210)

[第三章 车联网信息发布系统的整体设计 29](#_Toc385623211)

[3.1 系统功能需求分析 29](#_Toc385623212)

[3.1.1 交通道路安全服务需求 30](#_Toc385623213)

[3.1.2 交通数据采集服务需求 30](#_Toc385623214)

[3.1.3 娱乐信息服务需求 31](#_Toc385623215)

[3.2 三层架构的体系设计 32](#_Toc385623216)

[3.3 系统WAVE底层通信设计 33](#_Toc385623217)

[3.4 系统框架图 34](#_Toc385623218)

[3.5 本章小结 35](#_Toc385623219)

[第四章 网络中心设计与实现 36](#_Toc385623220)

[4.1 网络中心分析与设计 36](#_Toc385623221)

[4.2 基于LAMP架构的服务器实现概要 36](#_Toc385623222)

[4.2.1. 网络中心框架图 36](#_Toc385623223)

[4.2.2. LAMP架构服务器的部署 38](#_Toc385623224)

[4.3 数据库设计 41](#_Toc385623225)

[4.3.1. 数据库表设计 41](#_Toc385623226)

[4.3.2. 数据库接口设计 44](#_Toc385623227)

[4.4 Python CGI网关模块的设计与实现 46](#_Toc385623228)

[4.5 本章小结 50](#_Toc385623229)

[第五章 RSU与OBU的设计与实现 51](#_Toc385623230)

[5.1 RSU与OBU需求分析与设计 51](#_Toc385623231)

[5.2 平台介绍 52](#_Toc385623232)

[5.2.1. 硬件平台介绍 52](#_Toc385623233)

[5.2.2. 软件平台介绍 53](#_Toc385623234)

[5.3 软件设计 54](#_Toc385623235)

[5.3.1. 信息分发功能子模块 57](#_Toc385623236)

[5.3.2. Web客户端子模块 63](#_Toc385623237)

[5.3.3. 系统配置更新子模块 64](#_Toc385623238)

[5.3.4. WAVE底层通信子模块 66](#_Toc385623239)

[5.4 本章小结 67](#_Toc385623240)

[第六章 系统测试 68](#_Toc385623241)

[6.1 测试目的和环境部署 68](#_Toc385623242)

[6.2 系统测试 68](#_Toc385623243)

[6.2.1. 系统功能测试 68](#_Toc385623244)

[6.2.2. 系统性能测试 75](#_Toc385623245)

[6.3 车联网系统对比分析 84](#_Toc385623246)

[6.3.1. 系统功能对比 84](#_Toc385623247)

[6.3.2. 系统性能对比 85](#_Toc385623248)

[6.4 本章小结 86](#_Toc385623249)

[总结与展望 87](#_Toc385623250)

[参考文献 89](#_Toc385623251)

[攻读硕士学位期间取得的研究成果 89](#_Toc385623252)

[致谢 90](#_Toc385623253)

# 绪论

## 研究背景及意义

车联网技术属于物联网技术，主要应用于智能交通系统ITS（Intelligent Transport System）领域中。随着车联网技术研究的深入，IEEE已经在802.11无线通信规范的基础上制订了专门用于车载通信的协议标准----WAVE(Wireless Access in Vehicular Environment)。WAVE协议栈主要制定了车联网中车-车（V2V，Vehicle to Vehicle）和车-路（V2I，Vehicles to Infrastructure）的无线通信规范。作为智能交通系统的典型运用，车联网技术综合运用了各种高新技术，如信号与信息处理技术、射频遥感识别技术、传感器监控技术、互联网与网络接入控制技术等，利用设置在道路两侧或者车辆内部的红外传感器、射频识别设备、视频及音频感应设备、GPS系统等信源感知及传输设备，按照预先设定的协议，把道路以及道路上车辆的各种动、静态信息通过现有的网络互联互通起来。从这个维度上讲，车联网就是通过互联网把现有的多层次系统大规模的融合起来，从而达到对每一条道路和道路上的车辆进行全时空的监测感应，以起到减少交通事故、提高交通资源配置作用的网络与应用技术。它能够有效地解决目前城市普遍存在的道路监管能力低下、机动车辆增长过快、交通管理水平不足等问题。车联网技术是物联网技术面向智能交通系统（ITS）领域应用产生的一大分支，是整个物联网的一个核心部分。因此，研究车联网相关技术，开发车联网相关的核心产品，借此发展与其息息相关汽车类电子产业，推动相关领域的社会和经济变革，势必将有不可估量的社会经济效应。

为了确保在快速移动的车联网通信环境中，车载单元OBU之间，车载单元OBU与路侧单元RSU之间，可以进行稳定的高质量通信，美国电气和电子工程师协会（即IEEE协会）开始在基于无线传输的802.11技术（一种为无线传输服务的通信协议）的基础上定制ASTM -DSRC标准，并对该新协议标准重新命名规定为802.11p协议，其主要涵盖物理层和数据链路层管理规范。车联网的通信基础WAVE协议栈就是在802.11p协议的基础上，包含了对通信协议中的MAC层，网络层，传输层以及资源的安全管理的各种规范协议的集合。WAVE车联网技术能提高车辆在高速移动中的数据传输能力，并且能保证通信系统的稳定性，安全性和可靠性。基于WAVE协议栈的车联网系统可以作为新型的车辆诱导辅助系统和车辆道路服务平台的核心，应用到未来的智能交通，智能公路等领域，拥有十分广泛的应用前景。目前，IEEE已经相继推出了IEEE802.11p标准、1609.3规范、1609.4规范等相关车联网通信的规范标准，并将确定应用于包括车辆自身安全、交通管理服务以及商业娱乐服务中。

伴随互联网技术的快速发展，特别是Web2.0的广泛运用，它导致了传统由网站主导内容产生转变为由用户主导内容产生的交互式的互联网产品模式的诞生。互联网也从一系列的网络站点转变为成熟的为用户提供动态网络应用的服务平台。因此，一种以Web 2.0标准为基础的，用于搭建动态网站的强大应用程序平台的开发架构模型便被发明起来，其中典型的便是商用软件架构Oracle的J2EE和Microsoft的.NET。但紧随着开源学潮的逐渐兴起，开放源码的LAMP(Linux + Apache + Mysql + Perl/PHP/Python)架构体系凭借着其出色的性能，典型的如简易性、高模块化、高安全性、成本低廉、并发性能优异等，使其应用的普及范围不断扩大，已从诞生之初的小型服务应用系统的扩展到复杂、大型服务平台系统上，并开始与 J2EE 和.Net 等商业软件形成分庭抗礼之势[[1]](#endnote-2)。从国内国外网站的流量上来说，LAMP架构的服务平台已经占有了全部70%以上的互联网访问流量，LAMP模型已逐步发展成最热门的Web服务开发基础架构。

本文将基于车联网领域中缺乏完善的信息分发系统现状的前提下，针对车联网通信环境快速多变的特点，分析车联网信息发布系统的服务需求，包括交通道路安全服务需求、交通数据采集分发服务需求以及道路娱乐商业服务需求等方面。并在这基础上，利用IEEE最新的针对车联网提出的WAVE协议栈，设计出一套三层体系的基于LAMP架构的面向车联网应用的信息发布系统。从通信角度上，使用WAVE协议栈保障了车联网可以提供可靠的服务；从业务角度上，LAMP架构的高性能以及高稳定特性，保障了车联网中大数据、大用户量的服务需求得以快速实时的实现。并且，网络中心LAMP模型的实现，使得系统拥有良好的可维护升级和可扩展的能力。本文所述的面向车联网应用的信息发布系统，其第一层网络中心将采用LAMP架构部署服务器，为整个车联网信息发布系统的业务核心，其第二层路侧单元RSU和第三层OBU将采用软件客户端的形式实现。RSU和OBU之间通过WAVE协议进行通信，通过Web客户端的形式与网络中心进行信息交互。

## 国内外研究现状

### 智能交通系统研究现状

智能交通系统（ITS）是一个面向道路交通服务的智能系统，它以物联网技术为基础，涵盖了车辆的监测控制，运营调度，信息通信服务，车辆信息数据挖掘等领域。智能交通系统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，从而实现将公路、铁路、车辆、乘客以及路边的设施完全调动连接起来，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，并为公共交通系统的使用者和监控者提供智能化服务的系统。目前，国际上主要由欧洲的ERITCO、美国的ITS America以及日本的VERTIS等机构在积极的促进智能交通系统向民用化商用化演变。

智能交通系统的通信方式主要分成两种，即“车-路”通信（V2I, Vehicles to Infrastructure）与“车-车”通信（V2V,Vehicle to Vehicle）。国际上对ITS的研究主要集中在“车-路”通信（V2I）中。如日本的VICS系统和美国的IVHS系统，都是利用道路与车辆之间的信息连接，实现对交通路况的管理配置功能[[2]](#endnote-3)。WAVE协议栈作为智能交通系统中的核心通信协议簇，其核心框架结构及具体协议标准正逐步被标准化。符合WAVE协议栈的原型RSU和OBU WAVE模拟测试设备也已经逐步开发出来，其中最典型的有美国的Sirit公司（美国主要的[RFID](http://info.ec.hc360.com/zt/rfid/index.shtml)技术提供商）和日本的RSB公司（瑞萨半导体）已经开发出正在测试阶段的原型WAVE设备。台湾的Unex公司也在最近提出了整套符合WAVE协议栈标准的WAVE解决方案。总而言之，国际上关于“车-路-人”（V2I2P）的通信规范，伴随着最新的车联网通信规范（802.11p协议），自组网技术，网络接入技术，传感器监测技术等技术的不断更新变革，正朝着短程化、网络化、实时化、专用化、信息化的方向发展。除此之外，国外的汽车制造商也在车联网通信方面做出了卓越的贡献，如：通用汽车、西门子、宝马等。集成车载单元OBU的设备Onstar已经在美国上市[[3]](#endnote-4)。

相比国外迅速发展的车联网技术，国内相关领域的研究及智能交通系统的产业化相对比较落后。国内尚未有专门的机构或者公司，像欧洲的ERITCO组织等一样专门从事国际上车联网技术相关协议的标准定义工作。但伴随着国内经济的日益发展，人们物质生活水平的提高，加上国内城镇化的推进，智能交通系统的研究已经逐步被列入了重点。在近十年来，我国已经把智能交通领域的研究列入了“十五”、“十一五”、“十二五”的国家科技攻关计划中。各种国家研究所，高校甚至是企业也开始投入大量的物力和能力到智能交通系统的领域来，并也取得相应的成就，如公路的电子ETC收费系统、城市交通的智能指挥控制系统等。总之，我国虽然在智能交通系统的研究中相对落后，但发展迅速，并且后劲十足。

### 车联网的信息分发系统研究现状

伴随着智能交通系统的高速发展，为高速行驶的移动车载终端提供实时可靠的信息服务是当今ITS中一个十分重要的研究热点。但是以目前的方法和手段收集与发布信息的成本很高，并且针对不同需求开展信息服务的效果与可用性受到特定技术的限制。而能为智能交通服务，功能完善的可用于车联网的信息发布系统基本没有。

现有的车联网信息发布系统，按照通信方式，主要分成以下四种：

* 基于射频识别（RFID）技术的信息发布系统

这类系统的典型代表就是电子ETC收费系统（Electronic Toll Collection）。ETC系统是目前最流行并且逐渐普及的一种用于高速公路等收费路段的电子自助收费系统。这类系统利用射频识别技术，通过非触碰通信的方式进行身份鉴权和数据传输服务。此类系统一般依赖于专用的收费通道进行信息分发，不能完整的部署在所有的道路上，服务覆盖范围较小，并且信息交互的种类有限，一般只能交互身份信息，收费信息等特定的消息类型，无法满足大数据时代高信息容量传输服务要求。并且此系统中的移动设备只能被动的接收路侧单元的信息并作出处理，无法进行信息交换，也即移动车载设备无法向路侧单元主动请求信息。即，信息发布是单向，而非双向，因此它无法满足现今智能交通系统所需要的可靠稳定的交互式信息发布。

* 基于移动基站提供的2G/3G通信方式的信息发布系统

作为这种功能系统的典型代表，便是使用中国移动提供的蜂窝式CDMA或者GSM网络的信息分发系统。但这种系统对移动基站的依赖性较强，在没有基站信号的偏远道路或者基站信号盲点处，信息发布系统作用将大大折扣。加之，对于高速移动的车载终端传输速率较低，实时性较差，无法为车辆提供实时可靠的信息服务。更重要的是，使用移动信号的2G/3G信号信息发布系统的通信成本较高，商用性能较差。

* 基于ZigBee等低速率传输设备的信息发布系统

使用ZigBee网络作为交通信息发布系统，其最大的特点就是低功耗，低成本。但低功率带来的代价就是低传输速率和低服务覆盖范围。使用ZigBee网络进行信息分发，传输的数据量较低，只能勉强满足基本的交通信息收集和发布功能，无法满足现今大数据时代高信息容量传输服务要求。对于数据量要求较大的视频图像广告等业务无法满足。

* 基于Wifi通信的客户端-服务器模式的信息发布系统

这类系统使用无线Wifi作为网络覆盖和通信的媒介，但其终端之间通信依赖于网络接入点，在没有网络接入点覆盖区域，网络通信将会中断，并且这类系统仍然使用普通的802.11a，802.11b，802.11g，802.11n等通信协议，身份验证和鉴权耗时复杂，通信效率较低。其次，该系统中所有车载设备的网络接入服务均需要依靠在路侧部署的网络接入点（AP）。但该接入设备一般只起到网关路由器的作用，该系统所发布和存储的信息均需要经过网络中心，网络接入点没有任何信息处理过滤的能力。因此，这类系统信息冗余较多，效率较低，不利于在高速移动的车载环境中进行信息分发。

从上面四类系统的描述中，我们清楚看出，在高速移动的车载环境中，面向车联网业务的信息发布系统乏善可陈。并且每个系统均有其严重的局限性，如何使行驶在公路上通信环境快速变化的车辆有效并且稳定的获得网络资源，并与周围通信系统进行稳定的消息传递是智能交通系统中需要迫切解决的问题。

## 系统功能

从前一小节，可以看出，在智能交通领域中，能使在快速变化的通信环境中的车辆有效并且稳定的获得网络资源的车联网的信息发布系统乏善可陈。

本文主要研究重点是在基于车联网缺乏完善的信息系统现状的前提下，利用WAVE协议栈和LAMP动态网络应用服务平台架构的技术，设计出一种新型的面向车联网应用服务的信息发布系统，以满足智能交通系统中交通道路安全服务，交通数据采集分发服务以及道路娱乐商业服务等服务需求。

本信息发布系统研究内容及主要核心功能包括以下几点：

1. 根据车联网信息发布特点，设计三层架构的信息发布系统

针对车联网信息发布系统，系统数据信息的分发主要基于移动中的车载单元OBU。这种类型的消息具有数据量大，数据多等特点，并且车载单元OBU有效通信时间短的特点，老式的服务器客户端模型（C/S架构）的信息发布系统并不能十分切合车联网通信需求。因此，本文提出了新型的三层架构的面向车联网运用的信息发布系统。

1. 根据车联网通信特点，实现系统的底层WAVE通信功能

针对车联网信息发布系统通信环境快速多变，老式的无线802.11协议拥有复杂繁琐的身份验证机制和鉴权机制，车辆往往尚未身份验证完毕便离开路侧单元服务区。基于WAVE协议栈的通信协议标准，从功能上可以很好的解决该问题，本文将在后面详细描述该协议栈。

1. 根据车联网信息数据量、访问并发量巨大的特点，设计LAMP架构的服务器

由于车联网中OBU的数量庞大，造成对网络中心的并发访问能力要求很高。因此，作为网络中心需要性能强大，并发能力强，系统稳定的服务器。因此，本文将详细叙述使用LAMP架构部署动态网络服务平台实现网络中心功能的原因及其相应的特点。

1. 根据车联网RSU、OBU服务的特点，设计实现RSU、OBU客户端软件系统。

作为车联网信息发布系统最低层的数量庞大的路侧单元RSU和车载单元OBU，它们一方面需要两两之间进行通信的能力，另一方面它们的服务特点一般都具有基于位置特性，并且位置具有多变的特点。因此，本文将在后面根据RSU和OBU的特点设计RSU和OBU的软件客户端系统。

## 本文研究内容及组织架构

本论文研究目标是将WAVE通信技术和LAMP动态网络应用服务平台架构技术应用到车联网通信系统中，从而设计并搭建出能够应付快速变化的车联网通信服务环境的车联网信息发布系统。

本论文各章节的内容安排如下：

* 第一章绪论：

本章简要介绍本论文研究背景，并针对车联网技术国内外发展现状以及目前使用的作车联网信息发份发系统做了一定的研究对比。

* 第二章信息发布系统技术研究：

本章主要阐述了WAVE通信技术以及LAMP架构体系。第一节分别详细的介绍了WAVE协议栈的结构和WAVE通信特点，第二节则概述了LAMP架构体系，LAMP架构模型特性以及LAMP每一层的技术特征。

* 第三章车联网信息发布系统的整体设计：

本章分析了面向车联网应用的信息发布系统的服务需求，并根据该服务需求，针对车联网快速变化的车载环境，提出了底层通信由基于WAVE的MadWfi协议实现，并分成三层架构部署的车联网信息发布系统的整体设计。

* 第四章网络中心设计与实现：

本章描述了网络中心的设计与实现。其中重点描述了网络中心Web服务器的部署实现概要，网络中心数据库的设计（包括数据库表，数据库接口设计），Python CGI网关的设计。

* 第五章 RSU与OBU的设计与实现：

本章先分析路侧单元RSU和车载单元OBU的功能需求，并根据其功能特点，介绍了其软硬件平台，以及软件系统三大模块（信息分发功能子模块、Web客户端子模块和系统配置更新子模块）详细实现方案。

* 第六章系统测试：

本章从系统的功能和性能出发，分别对所设计的雏形系统进行的测试。并根据测试结果与其他同类信息发布系统进行比较，得出本雏形系统的性能优势。

* 第七章总结与展望：

本章对本论文进行详细总结，论述论文所做的研究和工作，以及系统后续需要进一步研究和解决的问题。

# 信息发布系统技术研究

## 车联网通信技术

过去的几十年是通信技术的快速发展的黄金时期，通信信息技术日益影响人们的日常生活。在这种趋势下，一门新的学科---物联网技术（IOT）应运而生。

一般的智能通信设备主要分两类，其一是包含“内部智能”的各种监测设备、工控设备、传感控制器、智能家居设备、音频视频监察系统等；其二包含“外设智能”的如外置无线通信终端的“智能化物件”，或者附带RFID的各种物资产品。我们所谓的物联网技术，就是利用现有的有线或者无线通信网络，把这两类智能设备连接起来，进而构建物联网应用的大融合。其中最典型的就是基于云计算的SaaS运营模式，它能提供各种安全控制，并能进一步精确的定制各种警报监察、监测定位、远程维护控制、在线升级配置、实时决策等管理服务功能。物联网技术目前是社会研究的热点，据国内外专家预测，未来的10年间，物联网就能在人们日常生活中普及，它将成为新兴信息技术产业的龙头技术，并将带来上万亿规模的潜在经济效益。国内外特别是西方发达国家和地区，如美国，欧盟，日本都相继提出和制定了物联网（IOT）技术的发展研究计划。我国前总理温家宝，也紧随国际步伐，提出了“感知中国“的物联网发展技术，并将之确定为国家战略发展产业。眺望未来，可以想象物联网技术将在我国迎来蓬勃发展的好机遇！

车联网技术属于物联网技术，主要应用于智能交通系统ITS领域中。随着车联网技术研究的深入，IEEE已经在旧式的无线通信协议（802.11）的基础上制订了车载通信的专属规范WAVE，专门用于车联网中V2V以及V2I的无线通信应用。

### WAVE技术简介

WAVE(Wireless Access in Vehicular Environment)是指能够实现的宽带车车之间、车路之间无线传输通信，完成实时精确和可靠的图像、语音和数据双向传输的新兴技术。WAVE作为下一代智能交通系统(ITS)的发展方向，它承担着车联网络技术发展的核心构件的作用。作为车载单元与路侧单元提供网路通信服务的WAVE 技术，它使得交管中心能收集车辆的行车数据的同时，也使得车辆能拉取车联网中的各种消息资源。WAVE 技术能提供高速的数据传输，并在此基础上同时保证系统的可靠性与稳定性，以及安全性。就目前的研究状况而言，基于WAVE的运用范围非常广。在本章中，将重点对适用于车联网络通信的WAVE协议栈标准进行重点研究和分析。

WAVE作为下一代智能交通系统的关键技术，其通信特点有以下几点：

* 基于WAVE的移动设备间可以进行信息交互，通信理论距离大概1000 米内，实际运用的有效距离约为300 米；
* 高传输速率：一般可达到 3-27Mbps；
* 与现有的互联网兼容：其网络层采用了 IPv6协议；具体的信道分布图见下图。



图2-1WAVE信道分布及应用

以 WAVE 为代表的下一代车路专用高速通信技术为各种 ITS 应用提供了有力支撑，其典型的服务包括：

* 交通安全的服务，如典型的车辆免碰撞提示、行车道路通告、行车天气实时通知等；
* 公共服务相关的服务，如典型的道路交通监控、智能导航地图服务、区域道路状况采集，停车车位搜索等；
* 商业娱乐性质的服务，包括互联网接入、商业广告服务、酒店住宿餐饮等商业服务的搜索。

### WAVE协议栈研究与分析

#### 协议栈分析

为了确保在快速移动的车联网通信环境中，车载单元OBU之间，车载单元OBU与路侧单元RSU之间，可以进行稳定的高质量通信，IEEE协会开始在旧式无线传输的802.11技术（一种为无线传输服务的通信协议）的基础上定制ASTM DSRC标准。并对该新协议标准重新命名规定为802.11p协议，其主要涵盖物理层和数据链路层管理规范。车联网的通信基础WAVE协议栈就是在802.11p协议的基础上，包含了对通信协议中的MAC层，网络层，传输层以及资源的安全管理的各种规范协议的集合。WAVE车联网技术能提高车辆在高速移动中的数据传输能力，并且能保证通信系统的稳定性。基于WAVE协议栈的通信设备主要分成两类：

* RSU ( Road-Side Unit，即路侧单元)

RSU承担着网络控制接入中心的作用，即AP（Access Point）的角色。它一般通过有线或者无线的形式与网络中心进行互联通信，通常设置在道路两旁。

* OBU ( On-Board Unit，即车载单元 )

OBU一般为车载设备单元，它通过无线通信的模式与RSU或者其他OBU进行互联互通。

RSU和OBU有两种基本的通信方式，在有RSU的情况下，RSU可以和周围覆盖范围的多个OBU以基础通信服务集（WAVE Basic Service Set）的形式提供服务。当周围没有RSU的情况下，OBU之间通过自组网的方式，构建WIBSS (WAVE Independent Basic Service Set)，也即独立结构式的基础服务集，从而实现车-车通信并共享信息。RSU和OBU均是利用WAVE协议栈实现在快速移动的通信环境中完成数据信息的传递。

WAVE协议栈从其诞生历史可以知道，其底层的协议（PHY层和MAC子层）便是基于IEEE 802.11协议的。它借鉴了802.11协议簇中11a，11b，11g等通信规范的特点。因而，也可以说WAVE（或者WAVE模式）就是定义在IEEE 802.11协议之上的一种新的车载通信架构，因而它同时具备了和之前的无线通信设备兼容的性能。

从之前叙述的WAVE设备可以看出，新的WAVE协议栈和当前的无线网络结构类似，它也有自己的BSS和IBSS结构，即WBSS和 WIBSS。对于WAVE设备而言，其主要控制中心节点AP（Access Point）为路侧单元RSU ，移动工作站（STA）为车载单元OBU。

WAVE 协议栈借鉴了传统七层网络结构，并且将信息传输区分为数据层和管理层进行管理。WAVE的数据层担负着数据传输的责任，WAVE的管理层则担负着系统传输行为的配置管理职责。从功能上 WAVE 协议栈包含了以下内容[[4]](#endnote-5)：

1) 802.11p服务：包含了 802.11 数据链路层和物理层，针对快速移动通信环境的特点进行了相应的优化，其频段为 5.8G Hz，频段较为干净，调制模式则为 OFDM；

2)多信道协作功能服务：它隶属于WAVE协议栈的数据链路层。WAVE协议栈规定了WAVE设备支持多物理信道共同工作，即包括控制CCH信道和服务SCH信道。它们在系统时钟的控制下根据车载环境协同工作或自主切换，从而实现增强 WAVE 数据链路层传输质量的功能。

3)信息传输服务：WAVE标准规定了本层有IpV6协议和WSMP协议两个标准可选。为了提高数据传输的效率，WAVE将网络层和传输层合二为一，也即路由功能以及数据传输功能均由此层负责。并且为了增加WAVE协议的扩展性能，使其能与具体应用相切合，WAVE并没有具体规定使用的路由协议。此外，由于WAVE协议栈规定可以实现IP 协议，因而基于WAVE的网络可以好现有网络无缝融合。与此同时，WAVE还设计了WSMP 协议，它简化了传统网络协议的流程，大大的优化了通信效率。

4) 其他：WAVE协议栈为了保证车辆通信中的私有信息的安全，在上层还规定了如Privacy等的各类安全算法。但由于车联网技术还并不成熟，WAVE并没有实现一个成熟统一的的安全算法方案。该部分方案，IEEE组织目前仍在协商中，有望在不久的将来达成一致。

#### 协议栈结构

WAVE协议栈主要包括IEEE 802.11p 以及上层的IEEE P1609. 1、P1609. 2、P 1609. 3以及P 1609. 4规范。其中802.11p为其底层的协议，它主要制定了物理层和介质访问控制层相关的规范。而IEEE P1609的协议簇则是在802.11p底层规范的基础上定制的，主要涵盖了链路层、网络层、传输层以及相应层中的资源配置管理标准以及安全规范。如下图所示：



图2-2WAVE协议栈结构示意图

从上面的结构示意图可以看出WAVE协议栈包括了802.11p和1609.4协议。其中802.11p协议的MAC层和PHY层使用的是5.8GHz的频带。这样，就保证了WAVE数据的可靠性，并且大幅度的优化MAC层，同时还能简化握手流程，提高了通信的效率。与此同时，它还同步使用1609.4协议，这样一来使得WAVE支持多信道操作，包括服务SCH信道和控制CCH信道CCH的操作、信道跳转和路由控制以及多信道选取。同时，它作为数据链路层的一部分，能为数据帧的传输提供高效的QoS保证。

WAVE协议栈的网络层由1609.3规范来负责实现，划分为数据层及管理层，数据层则主要是通信的协议以及发送数据的硬件结构，它由LLC、Ipv6的UDP和TCP以及WSM和它对应的WSMP协议组成；管理层主要是系统的配置和维护的功能，它包括了应用程序的注册、WBSS的管理、信道使用情况的监督、Ipv6的配置以及RCPI的监测和MIB的维护。

WAVE协议栈的安全服务保证是由1609.2规范负责实现。在本规范中，它定义了安全报文格式，以及如何处理这些安全报文的方法，并在此基础上提供对WAVE的应用程式及管理消息的安全加密功能。

WAVE协议栈的上层是由1609.1规范来实现，是一个可选层，它规定了如何将OBU的系统资源规范化的存储。利用该规范，外界应用便可以根据规范约束的接口来访问OBU资源，大大的提高了WAVE设备应用的兼容性能。

### WAVE信息传输服务

WAVE的信息传输服务主要分为两种，传输实际数据的数据传输服务以及维护系统安全性稳定性的管理数据传输服务。

* WAVE管理信息传输服务

WAVE协议栈通过层与层之间的传递信息管理数据进行系统的管理服务，主要包括：管理数据的传输功能、信道的接入和分配服务功能、WAVE服务广播监听功能、IPv6的定制配置功能以及管理信息库MIB的维护功能。

WAVE管理信息传输服务的职责是由WAVE管理实体WME履行的。WME将根据上层应用服务的不同请求，配置系统的服务参数以及提供跨层API，以满足上层应用接入服务SCH信道的要求。同时，WAVE管理实体WME还会定期的广播WMA数据包，告知其他WAVE设备自己的管理信息。例如，要访问系统特定的服务SCH信道，WME通过收发器向数据链路层的管理模块发送访问请求，并在信道规定的时钟内，调整通信设备工作到指定的频段中。

在车联网通信中，WAVE设备的路侧单元和车载单元的职责是固定的。作为服务提供者的RSU（或者在自组网中的某一OBU）将发送管理信息WSA帧，以此向其他设备声明其在某一个或者多个服务SCH信道上具备数据交换能力。其他的OBU所承担的职责则是监听和接收广播中的WSA信息报文，并根据其应用层服务的要求配置到某服务信道SCH上进行信息交互。按照WAVE协议栈规定，对任一特定的WAVE设备，它可以根据自身服务需求选择成为服务提供者或者服务使用者，甚至两者功能兼有。

根据上述描述，可知所有的WAVE设备均是由WAVE管理实体配置管理信道分配的。其基本的接入信道服务的例子可以由下图2-3清晰看出。

从图2-3可以看出，信息传输服务根据WAVE的接入分配信道的种类分为以下四种：只运行在服务信道或控制信道；来回运行不同信道，即在服务信道系统时钟内使用服务信道S，在控制信道系统时钟内使用控制信道CCH；优先使用服务信道，即在服务信道或者控制信道系统时钟内，可以立即对服务信道进行接入，并可适当增加服务信道系统时钟时间；在服务信道或控制信道系统时钟内，对服务信道进行接入控制（不能像上一种一样接收到服务请求后立即控制服务信道）。

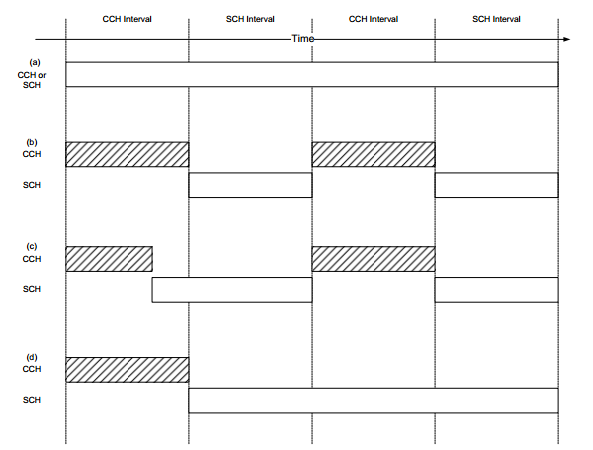


图2-3信道分配方式

* WAVE数据传输服务

从图2-2 WAVE协议栈结构图看以看出，WAVE数据传输从底层往上区分可以划分为三大功能部分：由数据链路层和物理层共同组成的物理信道接收和发送数据功能部分；由IPV6以及上层的TCP/UDP协议组成的兼容现有网络的数据传输功能部分；以及WAVE协议栈设计的替代网络层和传输层的WSMP数据包传输协议功能部分。

从上述三部分可以看出，支持IPv6和TCP/UDP协议使得WAVE协议栈能无缝兼容现有的无线传输网络。标准中新增加的WSMP传输协议，融合了网络层和传输层，并加强了专用的数据传输接口性能，以应对车载通信中的高速率低延时特性。

WAVE协议栈的一个创新之处就是使用了替代传输层和网络层协议的WSMP传输协议。基于WSMP协议的数据传输服务，传输的每一个数据包称为WSM数据包，并由服务的应用层根据需求设置其传输信道、发送功率和传输速率。由于WAVE协议栈同时支持IPv6网络传输服务和WSMP传输服务，因而符合WAVE通信标准的移动设备，可以根据自身服务的需求，选择兼容IPv6网络传输服务或者兼容WSMP传输服务，甚至两种协议兼之。

WAVE协议栈在设计之初，便是为了尽可能减少在高速移动的车载通信环境中，WAVE设备传输数据的延时时间，保证数据传输的通信质量，以满足各种基于车联网的应用服务需求。故使用单层的WAMP协议取代传统的网络层和传输层服务协议，以简化数据传输时的处理时间。

### 车联网通信技术对比

目前，现有的车联网信息发布系统，主要有四种类型，它们各自使用的车联网通信技术各不相同，分别是基于射频识别（RFID）技术、基于移动基站提供的2G/3G通信方式、基于ZigBee等低速率传输设备以及基于旧式的802.11的WiFi方式。

使用射频识别技术的车联网应用系统，其主要特点就是信息安全性能好，适合各种需要安全保密性的收费业务，其典型代表就是电子ETC收费系统。但其缺点同样明显，通信距离短，通信速率低，并且严重依赖道路上的射频识别设备。这类信息分发系统，信息的交互一般是单向的，并非双向。

使用ZigBee等低速率低功率的车联网应用系统，其主要特点就是功耗低，成本低，系统自愈能力强，持续运营时间长，适用于车联网信息分发系统中的路况中各种传感器的数据采集分发业务，其典型代表就是公路两侧的道路状况监测，如滑坡监测等。但其缺点明显，低速率，低服务覆盖范围，无法应用到车载单元OBU通信中，无法满足现今车联网高信息容量的服务要求。

使用移动基站提供的2G/3G通信方式的车联网应用系统，其主要特点就是通信服务覆盖范围广，通信速率较高，其典型代表就是各种基于车载单元OBU的自动报警及定位系统。但此类系统缺点明显，它对移动基站的依赖性较强，在没有基站信号的偏远道路或者基站信号盲点处，信息发布系统作用将大大折扣。并且，该类通信系统通信成本较高，无法大规模应用。

使用WiFi通信方式的客户端-服务器模式的信息发布系统，其主要特点是通信速率快，覆盖范围较大，适合高信息容量的各种多媒体业务。但其缺点同样明显，使用传统的802.11的无线通信方式，身份验证和鉴权繁琐，无法在高速移动的车联网中进行可靠的传输服务。C/S架构的系统部署，使服务器端压力繁重，对系统的稳定性和并发性是严重的考验，无法利用车联网业务中OBU基于地区聚合等特点，信息分发系统冗余较多，效率较低，实用性不强。

从上面四类系统的描述中，我们清楚看出，在高速移动的车载环境中，面向车联网业务的信息发布系统乏善可陈。并且每个系统均有其严重的局限性，如何使行驶在公路上通信环境快速变化的车辆有效并且稳定的获得网络资源，并与周围网络进行消息传递是智能交通系统中需要迫切解决的瓶颈。

但本系统使用的以WAVE协议栈为系统底层通信的信息发布系统，能够很好的解决上述的问题。

总所周知传统 IEEE 802.11 协议都使用了比较复杂的鉴权和关联机制，从而实现一个服务集中新站点的加入。然而在高速行驶的车载环境下，这些繁琐的鉴权、认证机制无疑将会增加车辆接入车载通信网的时间，在某些极端情况下，可能鉴权过程尚未结束车辆却早已驶离原来通信服务集所覆盖的区域。WAVE协议栈中的IEEE 802.11p规范在这方面进行了相当大的简化，引入了全新的模式——WAVE 模式下的基础服务集（WAVEMode BSS）。

在具体实现上，IEEE 802.11p 定义了服务提供者（Provider）和用户（User）两种角色。当系统中的两个设备（两辆汽车）想建立通信连接时，由其中一个WAVE设备充当服务提供者，另一个设备充当用户。首先，Provider 开启广播 WAVE 服务信息，宣告一个 WAVE 基础服务集（WBSS）的存在。WAVE 服务信息中包含了信道使用信息、应用程序支持类型以及各服务的配置参数等信息。此过程与传统 IEEE 802.11 中广播 Beacon 信息的情形非常相似。当用户（User）进入服务提供者（Provider）提供WAVE基础服务集的通信覆盖区域后，收到服务提供者广播的服务配置帧。此时，用户将利用预先保存的密钥解密该服务配置帧，以获取传输的管理信息（主要为获得基本集标识符BSSID）。接着用户根据BSSID判断是否和自身预设的BSSID匹配。如果BSSID符合，则用户被允许加入该WAVE 基础服务集中，并根据服务配置帧中的信息配置通信信息（如使用的频段等信息）。一旦服务提供者和用户之间建立起了一条通信信道，服务提供者和用户就转为对等服务，它们之间开始平等交换数据信息。当WAVE 基础服务集有用户需要结束通信时，由该用户向 WAVE相关的管理实体传输结束通信的消息，并离开该WAVE基础服务集；在极端情况下，当用户监测到已经离开当前的WAVE基础服务集时，用户同样会自下往上的异步传输结束通信的消息，进而结束当前的通信。在最新版本的 IEEE 802.11p 草案中又将上述流程进行了进一步的简化，规定移动节点的通信不需要预先加入到任何 BSS 中，而只需在数据帧中加入身份鉴权标识符BSSID 即可。在信道设置相同的情况下，任何两辆车只要拥有相同的身份鉴权标识符BSSID 便可以进行免验证的通信。在这样的方案，在同一信道通信的车辆只需要拥有同样的身份鉴权标识符BSSID，并且这些车辆完全避免了额外的开销便能直接通信。这一改进与上一版本相比，节约了服务提供者向用户广播WAVE BSS 等一系列流程，更为简洁，使得WAVE协议栈相比旧式的802.11无线通信协议更适合于快速移动的车载通信环境。

因此，本文设计的面向车联网应用的信息发布系统，由于其底层设计使用的WAVE协议栈，使得本系统能更适合车联网快速移动的通信环境，为上层各种基于车联网通信的应用服务提供高速率可靠的稳定的通信质量保证。

## LAMP架构技术体系

### LAMP架构概述

目前，国际上最流行的开源Web架构当属于LAMP（Linux+Apache+MySQL+Python /PHP/Perl）架构。LAMP架构一般由类Linux操作系统、Apache Web服务器、MySQL管理数据库以及Python等脚本语言四部分组成。LAMP架构包含的这四部分结构，均是开源的产品，是国际上目前比较成熟的架构，如新浪，搜狐，百度，腾讯等国内知名企业均使用这个架构。和传统的其他著名的Java/J2EE或者Microsoft的.Net架构相比较，LAMP模型是轻量级架构，其Web资源相对丰富，具有高快发，高性能以及跨平台等优势。因此，LAMP架构成为了大多数企业搭建Web服务器的首选，典型的LAMP网站如下表所示：

表2-1经典LAMP架构网站

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | L: System | A:WebServer | M: Storage | P: Script |
| Yahoo | FreeBSD+Linux | Apache | MySQl | PHP |
| Facebook | FreeBSD | Apache | MySQl+Memcached | PHP |
| Wikimedia | Linux | Apache+Lighttpd | MySQl+Memcached | PHP |
| Fickr | RedHat Linux | Apache | MySQl+Memcached | PHP+Perl |
| Sina | FreeBSD+Solaris | Apache+Nginx | MySQl+Memcached | PHP |
| Audiogalaxy | Linux | Apache | MySQl | PHP |
| Firendster | Linux | Apache | MySQL | PHP+Perl |
| YouTube | Suse Linux | Apache+Lighttpd | MySQl | Python |
| Mixi.jp | Linux | Apache | MySQl+Memcached | Perl |
| TypePad | Linux | Apache | MySQl+Memcached | Perl |

### LAMP架构模型特性

紧随着互联网技术的急速发展，基于HTTP协议的的WEB结构在互联网技术中起着举足轻重的作用。目前，市场上的WEB架构的开发模型主要有三种，即ASP.NET架构（Windows Server + IIS + SQL Server + ASP组合），J2EE架构(UNIX + Tomcat + Orecle + JSP组合)以及本文提到的LAMP架构（Linux + Apache + MySQL + PYTHON的组合）开发模型。

三种架构模型的性能对比，如下对比表格所示：

表2-2Web架构性能比较

|  |  |  |  |
| --- | --- | --- | --- |
| 性能比较 | LAMP | J2EE | ASP.NET |
| 运行速度 | 较快 | 快 | 快 |
| 开发速度 | 快 | 慢 | 中等 |
| 扩展性能 | 好 | 好 | 较差 |
| 安全性能 | 好 | 好 | 较差 |
| 建设成本 | 较低 | 高 | 中等 |
| 运行平台 | Linux/UINX/Windows | 绝大多数平台均可 | Windows平台 |

从上表看可以清晰看出，与ASP.NET架构，J2EE架构相对比，LAMP架构模型一般具有以下的这些特性：

* 灵活的开发特性

作为目前商业软件的两大阵营ASP.NET和J2EE，都是由单一公司控制，由顶层控制的。系统开发人员可以控制的范围非常有限，例如，J2EE是源于Java程序的框架，.NET框架则是ASP.NET开发的唯一选择， Rails则必须基于Ruby框架。但是，LAMP架构的开发模型，系统开发人员可以根据自身系统的需要选择不同的Web框架。此外，开发者因为可以获取全部的源代码。从而可以根据自身需要，在LAMP模型中增加或者删除特殊的模块（如身份验证模块）。因此，LAMP架构具有极其灵活的开发特性。

* 高模块化特性

LAMP模型的灵活性和稳定性是兼容的，除了拥有优异的灵活特性外，还一向以稳定性而著称。LAMP模型的各个开源软件，均能稳定的向后兼容。即便是系统更换了或者升级了新版本，开发人员一般也不需要重写代码。LAMP模型的稳定性源于其具有高度模块化的特点。LAMP架构的模块化是松散耦合的，并且模块间是可组合的并且自治，它依赖系统间的配置文件组合而成，因此LAMP模型的维护成本也比较低廉。

* 开发成本低，系统可扩展性能优异

面向车联网应用的信息发布系统，在系统部署阶段，开发部署成本肯定是最首要因素。与ASP.NET和J2EE等商业平台不同，LAMP架构模型，系统从高往下架设的所有产品均是开源免费的。并且，LAMP对运行的硬件条件要求也更为低廉。在相同条件下，对于Web服务的托管领域，相比ASP.NET架构以及J2EE架构的托管服务，LAMP模型的要低很多。如此同时，由于LAMP模型的开源特性，开发者可以根据自身的需求，对LAMP模型做特殊的定制，使得LAMP模型具有优良的可扩展特性。

* 安全性能高

作为目前最流行的Web框架开发模型，LAMP架构的部署量是巨大的，国内许多大的门户网站均使用LAMP模型架构服务器。但由于LAMP模型的开源特性，LAMP模型的开源特性，LAMP模型的安全问题也在不断的被发现和被优化中。就目前而言，相比于ASP.NET和J2EE开发模型，LAMP模型单位访问故障次数是最小的。

### LAMP架构技术详析

#### LAMP架构基础:Linux

LAMP架构的”L”一般泛指Linux操作系统，但一般认为它包括但不仅限于Linux，即我们一般是认为“L”可以代表各种Liunx/Unix系统，包括熟知的各种Debian、FreeBSD、Redhat以及Suse等系统。Liunx/Unix系统，在LAMP模型处于基础作用，所有的其他开源软件均运行在Liunx/Unix系统上，它的高效开放性等特点切实影响着LAMP架构服务器的运行效率。

Linux作为类UNIX的操作系统，并且服从可移植操作系统接口POSIX标准的开源系统。它属于自由的开源软件，通常通过各种包含了完整的安装包管理系统进行发布。作为LAMP架构的操作系统，截止至2013年10月1日止，Debian和Ubuntu之间共享市场中58.5%的web服务器市场份额,而RHEL,Fedora和CentOS则占据了剩余的37.3%。

作为UNIX体系中最重要的分支Linux，选择它作为LAMP模型的操作系统，一般是因为它具备以下四大的特点：

* 系统的高效开放性：

Linux系统遵守国际的标准规范，特别是POSIX接口规范，以及开放系统的互连（OSI）国际标准。Linux系统本质上是用C语言重写了Unix系统，但在其基础上做出了许多优化，因此Linux系统占用的计算机资源更少，其系统的效能更高。

* 多用户，多任务

Linux系统可以并行的被多用户使用，不同系统用户对操作系统的资源拥有不同的权限，用户之间互不影响。并且由于Linux系统的开源特性，为了保护系统的安全性，系统采用了各种安全加密技术，以及严格的权限管理体系，如严格的读写权限控制，跟踪审计，身份鉴权等。因而，它能有效的保证用户在多用户的工作环境中有合理的安全保障。

* 系统的可移植性

Linux系统在其设计之初便考虑了系统的可移植性，它能从一个平台无缝的移植到另一个平台中，无论是微型计算机系统或者是大型机系统，Linux系统均能运行。并且Linux系统内置了各种标准化的网络协议，这使得它能完美的兼容融入现代的通信网络系统中。

#### LAMP架构核心:Apache

LAMP架构的”A”一般泛指Apache服务器，Apache Web服务器在整套LAMP架构中起着承上启下的作用，它是连接着Linux系统和Python等脚本语言的桥梁。

Apache 服务器是一种高性能的开源服务器，占了全球72%的市场使用份额。Apache服务器在整个LAMP架构中起着极其重要的作用，它是整个Web系统的核心处理单元。它的性能直接决定着网络中心LAMP框架的性能高低。LAMP架构之所以选择Apache服务器作为其Web服务器，是因为它具有以下的一些特性：

* 跨平台性

Apache服务器是一款跨平台的Web服务器，它不经能运行于UNIX系统中，还能广泛的部署于Windows系统中。

* 稳定性:

Apache服务器使用的是多进程方式提供服务，每个进程均不能访问其他进程的资源，因此Apache服务器非常稳定，不容易出现死锁或者假死的情况。目前互联网中过半的服务器均使用Apache服务器。LAMP架构的高效稳定特性很大程度是因为它使用了Apache服务器。

* 高性能的处理动态请求

Apache在处理动态请求具有先天的优势，他比其他的Web服务器在准确性和速度方面有着优势。并且Apache服务器功能组件丰富，它比其他的IIS或者Nginx提供更多的功能组件，方便用户扩展功能。

#### LAMP架构数据中心:MySQL

LAMP架构的”M”一般泛指关系型数据库MySQL，MySQL是目前应用最广泛的数据库，其市场占有量正逐步上升，为大多数企业网络系统的企业级数据库的首选。

不过LAMP架构的“M”的概念，从广义上可以扩展为Memcached，即内存缓存系统，从这种观点出发，MySQL只是Memcached范例中最典型的应用而已。因此，”M”可以扩展到一系列的产品，包括MySQL数据库的memcache\_engine/memcachedb\_engine、MemcacheQ以及Sharedance等产品。

从LAMP模型出发，其“M”在整个架构中起着数据中心的作用。车载单元OBU访问的资源，OBU的各种监测信息都存储在数据库中。从服务里的角度出发，它起着一个数据服务器的作用，它辅助Apache服务器为用户进行数据存储和数据处理的作用。

本文主要讨论的”M”则主要代指关系数据库MySQL，作为开源免费的数据库，MySQL和商业数据库如Oracle, Db2, Sybase等，一样具备数据库的通用特点。其一，作为数据库管理系统，它们本质上都是一些结构化的数据的结合体，都需要向用户提供对数据库中的一般化处理接口，如CRUD操作（创建Create，读取Read，更新Update，删除Delete）等。其二，它们均是关系化数据库管理系统，均支持国际上标准的数据库语言SQL。众所周知，在数据库发展的历程中，曾经出现过各种不同类型的数据库系统，但因为关系型数据库有着其他数据库系统没有的优越性而被广泛的使用。

除了拥有一般数据库所拥有的特性外，MySQL还具备则以下特性：

* 开源数据库

MySQL最大的特点，就是其开源特性。和其他商业性的数据库相比，只要用户遵循GPL规定的规范，就可以按照自身的服务需求精确定制MySQL源码，开发自己需求的MySQL版本。同时，开源的特性使得MySQL能跨平台运行，并且MySQL还拥有许多数量庞大的便于开发的API接口。

* C/S架构的数据库管理系统**。**

MySQL数据库实行客户端/服务器运行模式，其中其服务器是为多线程运行模式。而客户端上除了提供类似MySQLadmin，mysql等客户端管理工具外，还为用户提供了不同的链接库和各式的接口。这使得MySQL支持C/C++, JAVA, PERL, PYTHON, PHP等语言。并且，MySQL使用多线程运行模式，这使其能充分发挥系统多处理器的特点，避免在某些特殊情况下仅使用单CPU的现象。

* 高性能的内存和磁盘管理技术

MySQL采用高性能的内存申请技术，临时表在内存中用hash表的形式存在；而对于磁盘表，则采用加密的B+树索引的方法，使用户可以迅速定位到搜索的数据。MySQL具有优秀的内存管理机制，它能严格的管理内存泄露，该功能保证其能通过了严格的Purify测试。

#### .LAMP架构粘结剂: PYTHON/PHP/PERL

LAMP架构的”P”一般泛指Python, PHP, Perl等解释性脚本语言。这些”P”类型的解释性语言，能很好的与Apache服务器相结合，使得传统的静态Web网页服务得到动态交互的能力。并且，这些“P”类脚本语言，一般都自带了丰富的并且功能强大的数据库操作库函数。利用这些库函数，Apache服务器能够简便的操作MySQL等数据库，进行相关的数据存储处理工作。”P”类型的解释性语言，在整个LAMP架构中起着粘合剂的作用，它的存在使得LAMP模型的服务器各个部分能够无缝的对接起来，是系统运行的更为流畅。

本系统将采用Python作为服务器的CGI的脚本语言。这里选择Python最大的原因是，和前面叙述的三类系统一样，不仅功能强大，而且开源免费，是一种通用型的解释性编程语言。作为”P”的代表，Python语言可以跨平台运行，包括：以Linux为内核的（Debian, Rehat, Ubuntu）,微软的Windows系统系列（xp/win7/NT等），Marcintosh (包括OSX)，以及几乎所有的类Unix系统。跟Java或者Perl语言相比，Python拥有许多显著的优势，利用Python语言可以实现许多功能强大的程序来完成服务器的各种任务。

## 本章小结

本章主要介绍了信息发布系统中的关键技术，即车联网通信技术以及LAMP架构技术。首先介绍了系统底层的通信协议WAVE协议栈，包括了WAVE协议栈的结构、WAVE通信特点、WAVE信息传输服务以及车联网通信技术对比。其次，再介绍了LAMP架构模型体系，包括了LAMP架构的结构特点，系统特性，优缺点等，为下文面向车联网应用的信息发布系统的设计奠定了基础。

# 车联网信息发布系统的整体设计

在第一章我们提到了车联网中信息分发系统的现状，其主要的问题是现有的信息发布系统在快速移动的通信环境下，通信质量差，传输速率低，系统并发访问能力不足，业务单一，并不能满足现今大数据高信息容量传输服务要求。在本章，我们将以此为主要解决目标，并跟据的车联网信息发布系统的实地应用环境，设计出一种具有一定实用价值的新型面向车联网应用的信息发布系统。该系统主要根据上章介绍的WAVE协议栈和LAMP架构为技术核心，实现车-路互通，车-车互通等形式车辆网络的互联通信，并以此作为车联网通信基础，实现了各种基于车联网信息发布服务为基础的交通应用服务功能。

## 系统功能需求分析

面向车联网应用的信息发布系统，其典型的运用场景便是车-车通信（V2V）, 车-路通信（V2I）。支持WAVE的车载设备OBU，将通过V2V或者V2R的方式在快速移动的车载通信环境下，依靠路侧设备RSU或者不依靠RSU的自组网的形式，组成车联网络。其典型的运用场景图如下所示：



图3-1车联网信息发布系统运行场景图

根据从车联网信息发布系统的运行场景图可以清晰得出，面向车联网应用的信息发布系统，其潜在的应用服务大致可以分成三种，即交通道路安全服务，交通数据采集分发服务以及娱乐信息服务。

### 交通道路安全服务需求

根据车联网信息发布系统的特点，每个车载单元OBU的用户，都会成为整个车联网信息发布系统数据采集者和接收者。OBU定时的接收来自其他OBU或者RSU的信息的同时，也会不断把自己的信息整理上传。基于这个基础，关于道路交通安全服务大致可以分为以下二种：

* 道路交通安全广播信息服务。

利用车联网信息发布系统，当系统覆盖范围中的某一位置发生如交通事故，物理道路灾害或者人为道路封闭等道路安全状况时，系统将根据车辆的位置，有特定针对性的实时发布道路安全广播信息。

* 车辆行驶自身安全预警服务。

路侧单元RSU可以与其覆盖范围的OBU或者其他RSU进行信息交互。RSU通过采集到附近的OBU的传感器信息后，便可以知道其服务范围内行驶车辆的行车状况。利用RSU强大的计算处理能力，便可以针对每个OBU的特定状况发布行车指引，如变道，减速等。

### 交通数据采集服务需求

从3.1的应用场景图可以得出，系统的第一层架构网络中心，不断和分布在各个位置的路侧单元RSU进行数据交互，每个路侧RSU又实时性的采集该服务覆盖范围中的OBU中的传感器实时数据，典型的应该包括车辆GPS位置，车速，油箱存储量，车辆类型，行驶方向等等。

交通管理部门，通过网络中心的特定API接口，可以方便实时性获得车联网系统中的车辆信息，并对这些信息进行处理存储或者下行分发，这就是车联网信息发布系统的交通数据采集分发服务。其典型的运用需求可以有如下的两种：

* 道路状态动态地图监测和绘制服务

交管部门可以通过网络中心的数据库，获得所有道路OBU的传感器信息，例如GPS位置信息，车辆类型，车速，行车方向等信息，然后通过系统建模，利用地理信息系统或者交通地理信息，动态地绘制出基于车联网系统的交通状况动态地图。利用绘制的动态地图，交通管理部门可以根据地图模型，监测道路拥堵状态，并可以根据系统的数据进行交通优化，大大的减少城市拥堵问题。

* 道路状况监控收费服务。

对于高速路段等收费路段，常常由于停车收费的原因而在高速路段的出入口发生交通拥堵。利用车联网信息发布系统，交管部门可以实时准确的测算OBU的行车距离，并可以通过网络实时扣费。车辆可以快速听过高速路段的出入口，而无需在收费站中停车等待。

### 娱乐信息服务需求

从3.1的应用场景图可以知道，每个车载单元OBU均通过RSU或者OBU自组网的形式连接到互联网中。所以，面向车联网应用的信息发布系统，从需求出发，可以实现以下的娱乐信息服务：

* 基于位置的商业服务

利用面向车联网应用的信息发布系统，可以知道用户OBU的地理位置信息。因此，系统便可以引入各种基于用户位置的商业服务，如基于用户位置的精确投放式广告服务，基于用户位置的语音向导式服务等。

* 基于互联网的用户上网服务

路侧单元RSU可以通过有线网络接入到外部网络中，这样车联网信息发布系统便可以利用WAVE协议栈在快速的移动车载环境中稳定高速的传输速率，提供的可靠的，远比3G/4G通信代价低的互联网接入服务，如在线视频，语音聊天等。

* 基于自组网的用户自娱乐服务

利用WAVE协议栈的通信功能，OBU与OBU之间可以实现自组网式的互联互通。OBU之间可以利用Adhoc模式建立连接通信，可以自己组成独立的集群网组。基于区域的组网方式，车载用户们可以实现自组网的各种服务，如多人游戏，多人视频相片共享浏览等自娱乐服务。

## 三层架构的体系设计

面向车联网应用的信息发布系统，其设计的目的是实现一套基于快速移动的运行环境中车载单元信息交互的车联网信息发布系统。由于车联网通信系统时刻处在快速移动的通信环境中，为了增加车载单元OBU的信息交互的实时性和可靠性，减少网络时延等造成的信息延误，因此本系统设计成三层架构的体系。即，在传统的服务器-客户端（C/S）结构的通信系统增加了一个路侧单元和区域服务器的中间层。此时，路侧单元RSU并不是单纯的一个网关作用，RSU还肩负着处理和分发信息的作用。

因此，本系统的架构整体上分为三层。即，第一层：网络中心；第二层：区域服务器和路侧单元RSU（在本文论述的雏形系统中，RSU将同时肩负区域服务器的功能）；第三层：车载单元OBU。同时，为了应对车联网网络快速变化的特点以及车辆获得信息具有位置性的特征，所述的系统每一层获得的信息类型和信息量均不同。第一层的网络中心对外可以获得外部信息源（如交通管理局，广告发布商等）输入的信息，对内可以获得所有的区域服务器的管理信息，以及大部分数据信息的索引。第二层区域服务器和路侧单元RSU，对上获得网络中心针对区域位置的发布的特定信息（包括管理信息和数据信息），对下获得下层车载单元OBU上传的数据信息。第三层的车载单元OBU可以获得针对车辆所在道路的公共信息以及针对车辆自身的特定信息。

位于信息发布系统第一层的网络中心，系统内部将部署Web服务器以及数据存储服务器等核心设备。通过网络中心提供的远程客户访问接口，外界信息源（如道路交通管理局，广告发布商等）可以直接从网络中心获得管理信息。网络中心还通过有线网络与所有的区域服务器进行通信，从而实现信息的从上而下的发布工作。

位于信息发布系统第二层的区域服务器，按地域分布，用于进行基于车辆位置数据的存储和调度管理工作，通过有线媒介与网络中心和路侧单元RSU进行通信。

与区域服务器同时处在信息发布系统第二层的路侧单元RSU，分设在道路两旁，实现有线网络与无线网络之间的转换工作，为车载单元OBU提供短距离无线网络的接入服务，并且存储基于车辆位置的信息数据（如针对位置的广告、针对特定道路安全信息等）。RSU实现车载单元OBU网络接入点的工作。

位于信息发布系统第三层的车载单元OBU，为设置在行驶车辆上的通信处理装置，它使用短距离无线通信的方式（WAVE）与路侧单元RSU进行通信。车载单元OBU为用户界面终端，它主要负责车载用户的数据采集，数据传输，以及自主选择网络功能。位于这一层的车载单元OBU使用两套选路功能，基于快速变化的车载环境的网络性能进行切换。这两套选路功能如下：

* 直接访问网络范围的路侧单元RSU，直接进行网络资源的申请和使用。
* 在网络性能较差的情况或者行驶道路附近没有路侧单元覆盖的情况下，使用独立型网络结构（Ad hoc模式），即基于车与车之间多跳进行网络资源的获得与使用。

本文所设计的信息发布系统，信息发布将动态的采用拉取或推送的形式进行信息的分发。即网络中心，区域服务器，路侧单元RSU，车载单元OBU，以及OBU之间的数据的获取方式均是双向的，并按照实际需求进行信息分发。如，对重复的公共信息数据系统将动态采用推送的形式进行分发，对针对特定OBU或者RSU需要的特定的信息则采用拉取的形式进行分发。

## 系统WAVE底层通信设计

车联网通信环境是一个十分恶劣而且变化迅速的高速移动环境，由于车辆行驶速度较快，所以多普勒效应特别严重。传统的基于802.11协议（包括11a,、11b以及11n等）的无线通信方式，均绑定有十分复杂而且耗时的身份验证和鉴权的过程，这务必将造成车载单元OBU处在不停身份验证和切换网络的死循环中。其典型的场景是：

当某OBU驶入某RSU服务覆盖范围时OBU需要和RSU进行通信，OBU将请求RSU发起身份验证。由于旧的802.11协议繁琐的身份验证和鉴权流程，当RSU确认该OBU身份，并允许OBU进行通信访问时，OBU由于高速移动的特性，已经使出该RSU的服务范围，进入其他RSU的覆盖范围。此时，由于OBU通信的需求没有得到解决，它将重新发起身份验证请求。至此，OBU通信资源将耗尽在不断的身份验证中和RSU网络服务切换中。

为了解决这类情况，面向车联网应用的信息发布系统，拟采用以WAVE协议为基础的MadWifi驱动作为底层通信驱动。该驱动采用的是双网卡模拟WAVE的服务信道SCH和控制信道CCH，并且符合802.11p协议的通信流程。

利用修改后的MadWifi驱动协议，用户将先通信后身份验证。即每次RSU底层接收到OBU 的数据包，先检查数据链路层的数据包包头是否携带允许通行的BSSID,如果携带则允许通信，如果没有则丢弃该数据包。此时用户的身份验证和鉴权将由上层服务实现。

因此，本系统的底层通信驱动均采用基于WAVE定制的MadWifi驱动，它能大大减少车载单元OBU的身份验证及鉴权时间，大大增加车联网信息发布系统的通信效率。MadWifi驱动是Linux下基于802.11协议簇的（包括11a，11b，11n）无线通信的网卡驱动。由于WAVE驱动协议簇使用的802.11p底层通信协议，所以需要定制MadWifi驱动，使其符合WAVE通信规则。

由于MadWifi驱动中的802.11n协议与最新的802.11p协议类似，均是使用正交频分复用技术OFDM作为其调制解调技术。利用正交频分复用技术将系统的通信[信道](http://baike.baidu.com/view/26456.htm)分成若干个正交的子信道，并在此基础上将系统高传输速率的数据流转变成若干个低速率的数据流。这些正交的子信号可以通过相关的接收机在接收端利用正交特性来区分。使用OFDM作为调制解调技术，将大大的提高车联网通信的传输速率和抗干扰性能。

由于MadWifi驱动是支持802.11n协议的，因此定制满足于WAVE通信标准的系统底层通信协议标准，只需要在802.11n的MadWifi驱动中做少量的定制修改。根据标准，WAVE在传输中是支持IP和WSMP两套协议的，由于本文设计实现的只是雏形系统，故这里我们保留MadWifi驱动原有的IP模块，使用IP作为网络层的协议标准。在数据链路层上，802.11n有着繁杂的身份验证和鉴权的模块，在本系统中将直接删除该功能模块，并相应的增加数据链路层中监测报文头部的BSSID域的功能模块，以替代MadWifi驱动的数据链路层的身份验证功能。这将大大减少OBU和RSU之间通信身份验证和鉴权消耗的通信时间，符合WAVE协议栈规定。

## 系统框架图

根据车联网信息发布系统的需求分析，得出其对应的系统框架图，如下图所示：



图3-2系统框图

## 本章小结

本章首先进行了面向车联网应用的信息发布系统的功能需求分析，详细的阐述了交通道路安全服务，交通数据采集分发服务以及娱乐信息服务等三大车联网服务需求。其次，在根据车联网通信环境变化快，质量差，可靠通信时间短，车辆时刻处在移动中的通信特点，创造性的设计出一种底层通信以WAVE协议栈为基础，三层架构的新型车联网信息发布系统，使其能满足智能交通系统中对车联网信息发布系统高速率、高可靠性以及高实时性的要求。

# 网络中心设计与实现

## 网络中心分析与设计

根据第三章所设计的车联网信息发布系统的框架图，可以看出系统网络中心的功能与地位。位于系统第一层的网络中心对外可以获得外部信息源（如交通管理局、广告商等）输入的信息，对内可以获得所有的区域服务器的管理信息以及大部分车载单元OBU的数据索引信息。由于网络中心需要实时获得交管中心或者广告商输入的数据（如广告视频，图片等）以及处理车载单元OBU的资源访问请求，因此网络中心可以部署一个高性能高并发量的web服务器，本系统拟使用用的是目前运用最广泛的并且十分稳定的Apache服务器。

此外，由于网络中心还需要存储所有区域服务器的管理信息以及OBU的数据信息，因此，网络中心还必须部署专用的数据库，进行数据的并发存储管理。在这里我们采用开源的基于多线程的关系型数据库MySQL。

鉴于以上两点原因，网络中心需要拥有高并发量、高性能以及高稳定性的处理能力，因此利用LAMP架构技术部署网络中心非常切合。

## 基于LAMP架构的服务器实现概要

第二章相关技术中重点提到了LAMP架构，本节将基于LAMP架构着重描述网络中心的服务器的设计与实现；并将主要通过LAMP框架图，以及LAMP架构服务器的实现过程两部分详细讲解基于LAMP架构的Web服务器的实现问题。

### 网络中心框架图

面向车联网应用的信息发布系统，其网络中心将使用基于LAMP架构的服务器，也即为部署了Linux（Debian）操作系统+Apache高性能并发服务器+关系型数据库MySQL+脚本解释性语言Python的物理机器。

如下图所示，网络中心架构图中最低层的是Debian操作系统，在其上端为Debian系统运行的Apache服务器，Apache通过Python脚本和MySQL数据库以及外部的Web应用进行交互。



图4- 1网络中心LAMP架构框架图

当外部的用户，如交通管理中心或者广告发布商需要访问网络中心，获取网络中心收集的数据，或者内部的OBU用户，需要上传或者拉取网络中心的数据时，它们仅需要使用包含了浏览器（Browser）的客户端即可实现数据的获取。当用户通过web浏览器访问服务器时，其本质经历了下图所示的流程：

当客户端（Client）通过浏览器向Apache发送HTTP请求时，运行在服务器（Server）中的Apache服务器，通过监听80端口获得该请求消息。此时，Apache服务器通过预设的Python网关程序，利用Python脚本解释器，运行SQL命令。与此同时，同样部署在服务器（Server）中的关系型数据库MySQL将会根据Python网关给出的SQL语句，处理并返回数据给Python解释器。Python解释器根据返回的Data数据，生成浏览器可以预览的HTML格式数据，返回给Apache服务器。最终，由Apache服务器将用户的HTTP请求生成HTML文件，通过响应推送到客户端（Client）的浏览器中显示。服务器（Server）通过Apache服务器的多线程运行模式，能够高效稳定的同时处理来自不同客户端成千上万的HTTP服务请求。



图4-2 Web服务器工作流程图

### LAMP架构服务器的部署

#### Debian安装部署

Debian与其他的linux发行版本，如Readhat, Foredo, Ubuntu一样，均属于Linux内核的操作系统。本系统所设计的网络中心将采用Debian的稳定版（Debian Wheezy）作为操作系统。

Debian Wheezy 的安装很简单，只需要登入Debian的官网下载Debian系统光盘镜像便可以安装使用了。安装完成后，便可以看到下面的登入界面：



图4-3Debian系统登入界面

Debian系统安装完成后含需要进行环境配置，以便为后续的LAMP架构服务器的运行做必要的准备。Debian系统的环境配置包括以下几点：

* 更改源：

通过命令：vi /etc/apt/sources.list 修改Debian系统的源，修改成国内的源。

|  |
| --- |
| deb http://mirrors.163.com/debian/ wheezy main non-free contrib  deb http://mirrors.163.com/debian/ wheezy-proposed-updates main non-free contrib  deb-src http://mirrors.163.com/debian/ wheezy main non-free contrib  deb-src http://mirrors.163.com/debian/ wheezy-proposed-updates main non-free contrib |

* 配置编译开发环境

新安装的系统必须通过apt-get update、apt-get upgrade以及apt-get install build\_essential命令更新和升级配置开发环境，这样才能让Apache以及MySQL完美运行。

#### Apache安装部署

Apache在Debian上的部署十分简便，只需要的终端中输入命令：sudo apt-get install apache2即可。通过命令:/etc/init.d/apache2 start 便可以启动Apache服务器。Apache服务器安装完成后，打开任意浏览器，在浏览器中输入：<http://127.0.0.1>或者<http://localhost>,看到“It work！”的页面，则表示安装成功。

Apache服务器静态网页默认的根目录：/var/www/，因此只要把静态的网页文件放在该目录下，便可以通过Apache服务器直接访问。

为了帮助区分Apache服务器不同的http业务，我们在/var/www/目录下，新建目录：/var/www/cgi-bin/存在车联网信息发布系统相关的python网关程序。然后并配置相关的Apache配置文件apache.conf，使得服务器的的URL地址/cgi-bin/能直接映射到/var/www/cgi-bin/目录下。

在该配置文件上增加如下内容：

|  |
| --- |
| ScriptAlias /cgi-bin/ /var/www/cgi-bin/ #浏览器路径映射关系  <Directory /var/www/cgi-bin/>  Options +ExecCGI  AddHandler cgi-script .cgi .py  AddHandler default-handler .jpg .png  </Dirctory> |

该配置内容的意思是，在路径/var/www/cgi-bin/下Apache服务器支持cgi网关脚本为cgi文件或者py文件（python文件），支持直接访问该目录的后缀为jpg或者png的文件。

#### MySQL安装部署

在Debian系统中安装MySQL数据库很简单，只需要通过命令：

|  |
| --- |
| apt-get install mysql-server |

MySQL在安装过程中输入用户登录命令，这里请注意，MySQL的登录密码不能和管理员登录密码相同，否则出错。MySQL数据库的一些启动方法如下所示：

|  |
| --- |
| service mysql start #启动  chkconfig mysql on #开机启动mysql  service mysql restart #重启 |

通过命令：mysql –uroot –ppassword（-u后面表示用户名，-p表示用户密码）便可以登入mysql数据库。

#### Python安装部署

Debian系统默认是安装了Python解释器的，用户只需要输入：python -V便可以查看自己安装的python解释器的版本：



图4-4Python解释器版本

如果python –V没有输出，则系统没有安装python 解释器，只需要通过sudo apt-get install python-dev 命令安装即可。

## 数据库设计

### 数据库表设计

面向车联网应用的信息发布系统，其数据均存储在网络中心的数据库中。为了实现信息发布系统的三类服务需求，即交通道路安全信息服务，交通数据采集分发服务以及娱乐信息服务。网络中心的MySQL数据库需要建立下列表：

1. 系统用户信息表

网络中心通过Web服务器为交管中心或者其他商业用户提供数据访问服务，因此需要建立系统用户信息表，以记录外部调用的用户的基本信息及其访问权限。当系统用户在系统注册时，把信息添加到该表中，如新的广告发布商入驻系统时更新表单。

表4-1SystemUserInfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 空值 | 说明 |
| SystemID | varchar(50) | 否 | 系统用户唯一ID |
| Password | varchar(50) | 否 | 登录密码 |
| UserName | varchar(50) | 否 | 用户名称 |
| ServiceName | varchar(50) | 否 | 提供商提供的服务名称 |
| ServiceAbstract | varchar(50) | 否 | 服务摘要 |
| AccessLevel | int | 否 | 用户访问权限 |
| Tel | varchar(50) | 否 | 联系电话 |
| Email | varchar(50) | 否 | 联系邮箱 |
| RegisterTime | datetime | 否 | 注册时间 |
| LoginTime | datatime | 否 | 用户登入时间 |

1. 车载单元信息表

网络中心数据库中，除了要保存系统用户信息外，还需要保存每个车载单元OBU的用户信息，与系统用户信息表类似，我们需要建立如下的表单：

表4-2 OBUInfo表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 空值 | 说明 |
| OBUID | varchar(50) | 否 | 车载单元唯一ID |
| UserName | varchar(50) | 否 | 车载单元用户名 |
| CarNumber | varchar(50) | 否 | 车载单元所在车牌号 |
| RegisterTime | datetime | 否 | 注册时间 |
| LoginTime | datatime | 否 | 用户登入时间 |

1. 系统服务信息表

系统服务信息表通过外键SystemID与系统用户信息表关联，同一个系统用户可以允许多个服务内容。系统服务信息表主要记录了服务内容，服务提交文档，服务的有效范围以及服务的提供商（SystemID）等信息。

表4-3ServiceContent表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 空值 | 说明 |
| ContentID | int | 否 | 服务内容内容唯一ID |
| SystemID | varchar(50) | 否 | 服务提供者ID，与系统用户信息表SystemUserInfo的SystemID关联 |
| ContentFileName | varchar(50) | 否 | 提交的内容文件名 |
| Abstract | varchar(500) | 否 | 提交内容摘要 |
| RelateRegion | int | 否 | 内容所在区域 |
| CreateTime | datetime | 否 | 创建时间 |

1. 车载单元监测信息表

车载单元监测信息表主要记录了网络中心每次由区域服务器中收到的车载单元OBU传感器的监测信息，如GPS的位置信息，车辆速度，湿度等。每条记录均有唯一的ID标记。该表通过OBUID与车载单元信息表中的OBU外键关联。

表4-4OBUData表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 空值 | 说明 |
| ID | int | 否 | 记录ID |
| GPSV | float | 否 | 车载单元所在位置经度 |
| GPSH | float | 否 | 车载单元所在位置纬度 |
| Speed | float | 否 | 车辆速度 |
| Temperature | float | 否 | 所在位置温度 |
| Humidity | float | 否 | 所在位置湿度 |
| Time | datetime | 否 | 记录获取时间 |
| OBUID | varchar(50) | 否 | 关联的车载单元ID |

1. 车载单元服务关联表

车载单元服务关联表ServiceRelation，主要记录了服务类型，服务提供商以及服务OBU对象的多对多关系。网络中心，只需要根据服务内容ID即，ContentID便可以快速的获得服务的OBUID集合。

表4-5ServiceRelation表

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 空值 | 说明 |
| ContentID | int | 否 | 服务内容内容唯一ID |
| SystemID | varchar(50) | 否 | 服务提供者ID，与系统用户信息表SystemUserInfo的SystemID关联 |
| OBUID | varchar(50) | 否 | 关联的车载单元ID |
| Abstract | varchar(500) | 否 | 服务关联摘要 |
| CreateTime | datetime | 否 | 创建时间 |

面向车联网应用的信息发布系统的网络中心，主要用到了上述的五个表，表与表之间的关系如下图所示：



图4-5数据库表关系图

### 数据库接口设计

从LAMP架构框架图中我们可以清楚看到，网络中心的数据库是经过Python脚本调用的。要利用Python使用MySQL数据库，必须下载安装Python的MySQL库，即MySQLdb。在使用MySQLdb提供的API之前，只需要import MySQLdb即可。

本信息发布系统利用Python脚本对数据库中五个表单的操作是通过定义五个数据库访问类（5个类对应5个表），五个类均封装了对相应的表进行CRUD（Create，Read，Update, Delete）操作。即：

* 系统用户信息表SystemUserInfo : SysUserDB类
* 车载单元信息表OBUInfo: OBUUserDB类
* 系统服务信息表ServiceContent： ServContentDB类
* 车载单元监测信息表OBUData: OBUDataDB类
* 车载单元服务关联表ServiceRelation：ServRelationDB类

由于五个类的操作大同小异，本文只要通过描述对系统用户信息表SystemUserInfo的接口来介绍，网络中心数据库的接口设计。

数据库表SystemUserInfo对应的操作类为class SysUserDB,其定义如下图所示：

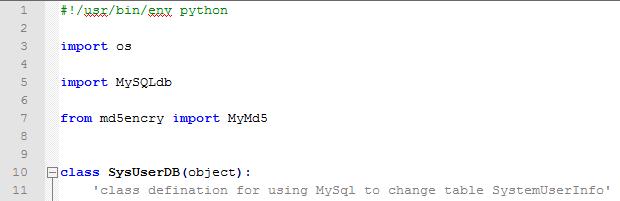


图4- 6数据库操作类SysUserDb

当网络中心需要访问系统用户信息表SystemUserInfo时，只需要import SysUserDB,并初始化一个SysUserDB实例便可。Class SysUserDB类提供了以下一些接口，方便系统对数据库表SystemUserInfo进行访问：

* \_\_init\_\_（）

当类SysUserDB实例化的时候，将会默认执行该方法。在该方法中，系统在会从MySQL.txt文件中读取数据库的用户名，密码，默认数据库表等配置信息，并初始化数据库（获取MySQL数据的cursor，self.cur = self.conn.cursor()），为用户的进一步操作做准备。

* insert\_user(self, SystemID, Password, UserName, ServiceName, ServiceAbstract,AccessLevel,Tel,Email,RegisterTime,LoginTime,)

插入操作，将系统需要在SystemUserInfo表中增加新的用户信息时，可以直接调用本方法，实现插入操作。

* delete\_user(self, user, passwd)

与插入操作相反，当系统需要在SytemUserInfo表中删除用户信息时，将可以直接调用本方法，实现删除操作。

* update\_status(self, column, status):

更新操作接口，当系统需要在SystemUserInfo更新用户某列数据状态时，将列名字以及对应的值传入本接口，从而实现状态更新操作。如，用户需要更改访问级别状态AccessLevel时，只需要调用updata\_status(“AccessLevel”,1)，即可以将用户的访问权限更改为1。

* select\_user(self, user, passwd):

信息查询接口，当系统用户需要查询系统用户信息时，通过传入用户名字和密码，将可以实现用户系统信息查询功能。

* check\_name(self, user):

用户名字检测接口，利用本接口，用户可以快速的检测到用户名是否已经被使用，一般在注册时使用本接口。

此外，为了保护用户个人隐私，对用户输入的秘密，均使用MD5加密算法加密后再保存进数据库中，系统无须知道用户真正的秘密值，只需知道用户密码加密后的MD5值即可。

## Python CGI网关模块的设计与实现

在本论文的第二章我们详细介绍了LAMP结构，其中“P”在本系统我们采用的是Python脚本语言。Python解释器，在LAMP结构体系中，其本质是作为一个CGI网关（通用网关接口，即定义在Web服务器和自定义脚本之间用户交换信息的一组定义标准。）存在。CGI网关在整个Web浏览服务的作用可以通过下图（CGI结构图）清晰看出：



图4-7CGI架构图

一个典型的CGI脚本执行流程如下所示：

1. 首先，Web服务器会根据URL地址映射相应CGI脚本。
2. 服务器接收到该请求，找到该特定的CGI脚本，并执行脚本。
3. 该特定的CGI脚本执行时，将根据用户传入的数据进行操作，包括数据处理、数据运用，并根据数据处理的结果调用系统数据库或者继续执行其它脚本。
4. 运行该脚本将输出某种Web服务器能理解的输出数据（一般为HTML格式的数据）。
5. 服务器接收来自目标脚本的输出数据，并把它回传到浏览器或者Web客户端，让用户预览到该结果。

CGI脚本通常被Web服务器调用，其功能用于处理用户通过HTML <FORM>或者 <ISINDEX>元素提交的数据。一般而言，CGI脚本设置在服务器的特殊目录下（如本系统设置在/var/www/cgi-bin）。Web服务器在该目录下，处理各式各样的客户端访问请求（如客户机的主机名,请求的URL,查询字符串等等）以脚本的shell环境，执行脚本，并将脚本的输出返回给客户端。

Python CGI脚本的输入一般通过直接连接的客户端获得，在某些情况下客户端表单的数据也是通过这种方式读取的，当然更一般的情况下是通过URL的请求字符串传递表单数据。Python CGI网关模块，其功能本质上就是为了处理不同的数据请求功能，并为我们的Web服务器提供一个简单的访问接口。目前，通过Python CGI 网关的形式，我们能轻易的调试脚本，并获得文件上传表单跟服务器常用功能。

Python CGI网关在本系统中，主要承担以下的作用：

* 接收Web浏览器系统用户（交管中心，广告发布商等）登入请求，并按照预设规则读取或者存取系统用户信息，并返回相关的HTML数据格式。
* 处理Web客户端（主要是OBU）系统发送的请求，处理OBU端的数据，并按要求返回规定的响应信息。
* 处理HTML页面跳转的逻辑功能。
* 处理MySQL数据库的业务逻辑功能。

本系统最典型的Python CGI脚本是logintest.py。它主要负责系统用户身份鉴权并显示鉴权结果的功能，其大致的源码如下所示（只保存关键部分）：

* logintest.py源码

|  |
| --- |
| #!/usr/bin/python  import cgi  import os  from md5encry import MyMd5  import db  header = 'Content-Type: text/html\n\n'  url = '/cgi-bin/login.py'  loginout\_url = '/cgi-bin/loginout.py'  logintest\_url = '/cgi-bin/logintest.py'  reshtml = '''<HTML><HEAD><TITLE>  （省略）  </BODY></HTML>'''  scripthtml = '''  <script type="text/javascript">  function Upload(){  var url = getFileUrl("idChange");  var dataSend = "image=" + url;  alert(url)  $.ajax(  {  type: "POST",  url: "/cgi-bin/upload.py",  data:dataSend,  success:function(){  alert("send success!");  }  });  }  </script>'''  errhtml = '''<HTML><HEAD><TITLE>  （省略）  </BODY></HTML>'''  def process():  form =cgi.FieldStorage()  if form.has\_key('personName'):  name = form['personName'].value  else:  name = ''  （省略）  python\_db = db.SysUserDB()  result = python\_db.select\_user(name, passwd)  （省略）  #show HTML  if \_\_name\_\_ == '\_\_main\_\_':  process() |

从logintest.py的源码我们可以看出，一般的Python网关模块，主要由两部分构成：

第一，包含了HTML格式输出部分。即Python脚本中包含了输出HTML格式的字符串常量如：reshtml，errhtml，headhtml，scripthtml等。其中scripthtml主要负责生成HTML中的JavaScript源码。JavaScript源码能让静态网页具有动态执行的功能，其典型功能便是能够调用异步数据传输的Ajax接口函数。

第二，包含了Web服务器业务逻辑功能部分。在logintest.py源码中，业务逻辑功能主要通过调用process()函数实现。process()接口的典型作用就是获取用户的表单输入，再根据用户传入的数据，利用Ajax的异步执行功能，从MySQL数据库中查询获取用户数据，并在页面上返回显示。它先通过：

|  |
| --- |
| form =cgi.FieldStorage() #python的cgi模块用户获得用户输入的表单数据 |

获得用户输入，经过输入有效性判断后，实例化SysUserDB数据库操作类，然后利用SysUserDB对系统用户进行身份验证，并获取相应的用户数据，再通过服务器回调给用户的客户端显示。

## 本章小结

本章主要了根据第三章提出的设计方案，从具体实现出发，详细的阐述了网络中心的设计与实现方案。本章首先根据网络中心的功能需求出发，即高并发量、高性能以及高稳定性特点，阐述了使用LAMP架构部署网络中心的必要性。其次，再根据网络中心的架构，详细的描述了包括基于LAMP架构Web服务器的搭建步骤，网络中心数据库设计实现方案（包括数据库表单设计以及数据库接口设计）以及Python CGI网关模块的作用和具体接口实现方案。

# RSU与OBU的设计与实现

## RSU与OBU需求分析与设计

根据第三章所设计的车联网信息发布系统的框架图，可以看出系统第二层路侧单元RSU以及系统第三层车载单元OBU的功能与地位。位于系统第二层的路侧单元RSU起承上启下的作用，它对上通过有线网络连接到服务器群中，对下负责与覆盖范围的车载单元OBU进行信息交互，承担着网络接入点（AP）的管理工作。因此，路侧单元RSU，应该存在以下两大功能模块：

* 通信功能模块。

该模块对上承接网络中心，负责和网络中心沟通交流，对下承接与车载单元OBU模块采集和分发信息功能。

* Web客户端模块。

该模块负责通过HTTP协议链接到网络中心或者RSU的Web服务器中，以获得各种广告发布商发布的广告消息。

位于系统第三层的车载单元OBU，起着收集分发行驶车辆信息的任务。它同时也为整个信息发布系统的基础用户。车载单元OBU一方面负责记录车辆的各种传感信息，如车速，GPS位置灯，另一方面还承担着与路侧单元RSU的互联互通功能。

与路侧单元RSU功能相似，在没有路侧单元RSU覆盖的情况下，车载单元OBU同样要承担网络接入中心AP的作用（即Adhoc自组网模式），因此车载单元也同样包含了通信功能模块与Web客户端模块。

从这点出发，由于RSU和OBU功能类似，均包含通信功能模块和Web客户端模块，故RSU和OBU的软件客户端可以设计为同一软件系统，用户只需要根据设备的类型配置客户端启动不同的功能即可。

## 平台介绍

### 硬件平台介绍

对于路侧单元RSU和车载单元OBU，本信息发布系统将采用一款基于ARM架构的开发板—Tiny6410作为硬件平台，并匹配一款高性能的NETSYS-9800000N型号的USB网卡。其硬件实物图如下所示：

****

图5-1硬件实物图



图5-2硬件整体实物图

* Tiny6410开发板

基于ARM11架构的Tiny6410开发板，其处理器为S3C6410的ARM11架构芯片。Tiny6410开发板的资源特性如下表所示：

表5-1 Tiny6410开发板资源

|  |  |
| --- | --- |
| **Tiny6410SDK**开发板  资源特性如下：**Item** | **Description** |
| CPU | Samsung S3C6410A(ARM1176JZF-S) |
| 频率 | 运行频率533Mhz, 最高可达667Mhz |
| RAM | 128M DDR RAM，可升级至256M |
| Nand Flash | 128M/256M/512M/1GB, 缺省为256M |
| 多媒体 | 支持Mpeg4, H.264, H.263, VC1硬件编解码，高达30fps@SD |
| 3D | 支持3D硬件加速处理 |
| 2D | 支持图形图像无极缩放，旋转，翻转 |
| 指示灯 | 4 x User LED(在核心板), 1 x Power LED |
| USB Slave | 1 x mini USB(底板没有设计OTG功能) |
| USB Host | 通过USB HUB芯片，实现4个USB Host接口 |
| 网络接口 | 10/100M MB以太网，RJ-45接口 |
| 音频输入输出 | 3.5mm 标准双声道音频输输入输出口 |
| SD卡 | 普通SD卡座 |
| 串口 | 4 x RS232 DB9串口，4 x TTL电平串口座 |
| 红外 | 1路红外接收头 |
| 温度传感器 | 1路DS18B02温度传感器 |

* NETSYS-9800000N型 USB网卡

NETSYS-9800000N型号的USB网卡，外设18DBI的全向天线。该USB网卡工作功率为4200mw，工作频段为2.4GHz，具有网络覆盖范围大，传输速率高，连接稳定性好等特点。

### 软件平台介绍

路侧单元RSU和车载单元OBU采用的软件开发平台均使用Qt，开发工具为Qt Creator，是一款跨平台的可移植性的C++界面开发平台。从功能上讲，Qt同微软Windows平台上著名的MFC，ATL以及X Window上的GTK，Mofi等图形界面库是相似的，但Qt具备以下三大有点：

* 跨平台特性:

使用Qt库开发的程序具有良好的跨平台特性，Qt程序均能在以下操作系统中运行: Linux、 Solaris、 Microsoft Windows NT、 SunOS、 Digital UNIX (OSF/1)以及 FreeBSD等。

* 优良的面向对象封装

Qt对许多特定功能都进行了面向对象的封装（如服务于TCP连接的socket通信TcpServert类和TcpSocket类）。Qt的这种机制使得 Qt库设计的各个功能类模块化程度非常高，具有良好的维护和重用特性。除此之外，Qt 提供了一种特殊的信号槽机制，并用它来替代 callback方法，这使得Qt模块之间的进行消息传递和访问非常安全和方便。

* 丰富的 API

Qt 库提供了数量庞大的各种 C++功能类，其典型的代表有TCP服务的QTcpServer ,QTcpSocket,,IO操作的I/O device，甚至还提供了包括复杂的正则表达式的处理功能，系统开发者可以利用这些API进行快速开发工作。

由于Qt库的这三大特点，特别是其强大的跨平台移植特性，故开发者可以方便的在PC上开发测试程序，然后无缝的移植到Tiny6410等嵌入式开发板中。利用Qt的这种特性，可以大大的减少了系统的开发时间和测试成本。

## 软件设计

从路侧单元RSU和车载单元OBU的功能模块可以看出，其软件系统功能上大同小异，只存在侧重点不同。因此，从系统设计实现出发，为了提高开发效率，增加软件客户端系统的可扩展性，故设计RSU和OBU的软件为同一个系统，只是启动时根据设备角色不同，配置不同的启动功能。

由于本系统软件部分除了实现信息发布功能外，还必须提供给系统用户实时交流使用的UI界面，故本系统采用观察者模式实现全系统UI功能。系统的程序结构关系图如下所示：



图5-3系统的程序结构关系图

从程序类结构关系图中，我们可以看出，系统的软件部分主要由以下的一些功能类组成，每个的功能类的典型作用如下所示：

* Class MainWindow

MainWindow类，为整个系统UI界面的最高级父类。信息发布系统的所有功能都是包含在该类中。由该类生成四个主要TabWidget显示子类，每一类负责一个主要功能的显示，分别为信息发布的Server部分，信息发布的Client部分，系统更新消息显示的MesDef部分，以及Web客户端部分。

* Class GlobalConfig

系统软件部分全局变量的定义处，定义了端口，IP地址，最大缓存大小等常量。

* Class CControl

系统最重要的控制类，它包含了Class TCPThread，ClassUDPThread，Class TCPThread\_c, Class WebTab类。MainWindow实例包含了一个CControl类，并利用该类开启TCP，UDP服务器线程以及和网络中心进行管理消息通信的TCP客户端线程。

* Class Show

负责系统消息的打印工作，本质上是调用MainWindow各个子控件的show接口打印显示消息。

* Class TCPThread

继承于Qt的QThread类，负责系统TCP服务器的监听和处理连接工作。TCP服务器采用异步IO的方式进行通信（即采用轮询方式），利用select函数处理新连接以及已有连接的消息接收任务。并提供m\_client[]数组保存已有连接的套接字描述符，让MainWindow类可以直接调用并发送消息给TCP的客户端。

* Class TCPThread\_c

继承于Qt的QThread类，负责系统作为客户端模式下，connect到中心节点的TCP服务器的连接工作。连接建立完成后，则负责等待系统的消息推送和显示工作。提供了一个套接字描述符m\_client的成员变量，MainWindow实例能利用该socket描述符与网络中心实现异步通信。

* Class UDPThread

与TCPThread类似，继承于Qt的QThread类，负责系统的UDP服务器的建立和维护工作。UDPThread包含了CUDPServer类，该类包含了一个发送广播消息的messageList，CUDPServer类维护该广播消息队列，并按照先进先出的方式进行消息发送。UDPThread类提供了ProduceMessage接口给CControl类，利用该接口CControl类可以为该消息队列添加发送的广播消息，并按照顺序把消息发送出去。

* Class TCPThreadCenter

继承于Qt的QThread类，是客户端软件负责管理消息更新的TCP客户端线程操作类。该软件启动的一开始，随着TCP服务器启动的时候，由CControl类控制开启。它首先连接到负责配置管理消息定义的网络中心，建立与网络中心的TCP连接。如果有消息重定义发生。TCPThreadCenter类将收到网络中心的配置文件。该类的实例在获得新的配置文件后，将发送update信号给MainWindow，更新显示新的消息定义。如果没有消息重定义，则关闭该线程。

* Class WebTab

该类是位于MainWindow界面类实例其中的一个控制类。该类主要包含了Qt 的QWebView类，利用该QWebView类，该控件WebTab类可以简单的实现Web客户端功能，即从RSU中获得多媒体资源的URL地址，并循环显示。

* 其他

包括了Class CarInfoAbstract，Class WaveMessage，Class WaveImpleMessage, Class WaveLocation等其他消息或者资源类等。其中

CarInfoAbstract 负责获取车辆消息的接口定义的抽象类；WaveMessage负责系统的Wave消息定义；WaveImpleMessage 负责要发送的消息类型的定义；WaveLocation负责地理位置消息的类型定义。

此外，RSU和OBU的客户端软件，如果从功能上划分，系统软件部分还可以分成三大模块:

* 信息分发功能模块
* Web客户端模块
* 系统配置更新模块

### 信息分发功能子模块

#### 作为服务器Server状态配置启动时的信息发布子模块

路侧单元RSU和车载单元OBU均需要信息发布功能模块。但两者的功能略有不同。当客户端软件作为服务器的Server状态时，信息发布功能模块具有以下的功能：

* UDP广播信息通信功能

在用户启动系统的服务器版本时启动UDP服务器。UDP服务器，其功能主要分为两点：第一，广播告知发现功能。路侧单元RSU或者在没有RSU的通信环境下OBU打算通过自组网通信并且愿意成为临时的AP中心时，都会通过UDP广播的形式，发送消息给所有的OBU告知自身的存在。第二，交通安全信息广播功能。当路侧单元接到网络中心或者附近其他RSU传输的安全消息时，将通过广播的形式将消息传送给覆盖范围的所有车载单元OBU。UDP广播信息通信模块，其程序流程图如下所示：



图5-4UDP广播信息通信模块流程图

从UDP广播通信模块，可以看出，只要往UDP服务器的广播消息队列中增加消息，UDPServer线程将会自动的广播出去。因此，我们在系统中还增加了一个UDP服务器的异步消息广播接口（线程安全）：

|  |
| --- |
| void CControl::ProduceMessage(BroadObj mes)  {  this->m\_udpthread->Produce(mes);  } |

从代码可以看出，该接口本质上还是调用了udpthead线程的Produce接口函数，为UDP服务器消息队列增加广播消息。利用该接口，客户端在需要广播消息时，只需要简单的调用该线程安全的广播API接口，便可以随意使用UDP广播方式发送消息。

* TCP服务器通信功能

在用户启动客户端软件的服务器版本时启动TCP服务器。和其他TCP服务器类似，该服务器有两大功能：一是绑定端口，然后监听来自其他客户端的connect连接请求，二是利用已有的套接字socket，同时和其他客户端软件进行消息收发工作。不过和传统的TCP服务器不一样，本系统的TCP服务器通信模块并非采用多进程或者多线程模式，（即一个进程或者一个线程在不停的监听TCP服务器的端口等待连接，然后由该进程或者线程根据客户端的请求建立新的子进程或者线程进行通信），而是采用单线程IO异步的形式建立TCP服务器。

传统的TCP服务器是基于多线程的阻塞模式，也即阻塞式IO操作，它们两个一般有如下的区别：

表5- 2同步IO和异步IO对比

|  |  |
| --- | --- |
| 同步IO操作（特点） | 异步IO操作（特点） |
| 利用多线程提高吞吐量 | 单线程即可实现高吞吐量 |
| 通过时间片分割和线程调度利用多核CPU | 通过功能划分利用多核CPU |
| 需要由OS调度多线程使用多核CPU | 可以将单进程绑定到单核CPU |
| 难以充分利用CPU资源 | 可以充分利用CPU资源 |
| 内存轨迹大，数据局部性弱 | 内存轨迹小，数据局部性强 |

TCP服务器通信模块的流程图如下所示：



图5-5TCP服务器通信功能流程图

由于TCP服务器监听端口和接收数据的IO操作等待时间较长，故在本系统中交由select异步IO操作线程进行管理操作。但由于服务器的消息分发具有不可意料性，故除了提供select异步IO操作线程之外，本系统还提供了一个TCP服务器端发送接口：

|  |
| --- |
| void MainWindow::on\_bnt\_send\_message\_clicked()  {  QString mes,number;  mes=ui->lineSend\_s->text();  number=ui->lineClientNumber->text();  int number\_client = number\_client=number.toInt()-1;  int sockfd=this->m\_pControl->m\_tcpthread->m\_client[number\_client];  int n =write(sockfd,mes.toLatin1().data(),mes.length());  if (n!=0)  {  number=QString("Send to Client %1:").arg(number\_client,0,10);  ui->textEdit->append(number);  ui->textEdit->append(mes);  }  ui->lineSend\_s->clear();  } |

该接口本质上是利用了CControl类中包含了select子线程的套接字描述符的接口，利用该接口可以获得select子线程中所存储的和客户端建立连接的套接字描述符。然后再利用socket的write函数，将信息发送给TCP客户端（在本系统即为OBU）。

#### 作为服务器Client状态配置启动时的信息发布子模块

当系统用户为车载单元OBU时，并且OBU不作为自组网的中心节点的情况下，将启动客户端Client状态的软件系统。在本状态下，软件系统只需启动TCP客户端，与中心节点的RSU或者OBU进行点对点的通信即可。其TCPClient线程的程序流程图如下所示：

该线程主要负责消息的接收工作，由于客户端发送消息的异步性以及不可预料性，故额外提供了一个用户发送消息的接口函数：

|  |
| --- |
| void MainWindow::on\_bnt\_Send\_clicked()  {  QString mes;  int sockfd=this->m\_pControl->m\_tcpthread\_client->m\_sockfd;  if(sockfd==-1)  {  ui->textEdit\_2->append("Send error,Socket haven't open!:");  }  else  {  mes=ui->lineEdit\_send->text();  int i = write(sockfd,mes.toLatin1().data(),mes.length());  if(i!=0)  {  ui->textEdit\_2->append("Send to Server:");  ui->textEdit\_2->append(mes);  }  else  {  ui->textEdit\_2->append("Send Error!");  }  ui->lineEdit\_send->clear();  }  } |

和服务器Server模式下的消息发送类似，on\_bnt\_Send\_clicked接口本质是用获得客户端线程所得到的套接字描述符socketfd，然后利用该描述符进行异步消息的发送。



图5-6Client模式下信息发布子模块流程图

### Web客户端子模块

Web客户端模块，其本质是一个移动端的浏览器，其主要功能就是访问网络中心或者RSU的Web服务器，以获得广告发布商等发布的基于位置的多媒体资讯。Web客户端模块本质上是通过Qt自带的封装功能完善的QWebView类实现。

QWebView类实现Web服务器的访问只需要经过三个步骤，即：

1. 新建一个QWebView对象,如:QWebView \* webClient = new QWebView(parent);
2. 调用QWebView对象的load或setUrl方法，设置访问网页的URL地址。
3. 调用QWebView对象的show接口，获得已经设置的URL对应的的网页资源并显示。



图5-7Web客户端子模块流程图

从上图可以看出Web客户端模块的运作流程，它先从指定服务器获取静态XML配置文件（主要是保存了车载单元OBU需要访问的基于位置的网页URL地址）。获取到XML的配置信息后，Web客户端模块便利用模块的cycle接口，循环访问Web服务器的网页资源。

|  |
| --- |
| void WebTab::cycle()  {  QVector<QString>::iterator it=reader.urlvector.begin();  for(it;it!=reader.urlvector.end();it++)  {  QString str=\*it;  QUrl url2(str);  web1->load(url2);  web1->show();  QTest::qWait(5000);  }  } |

### 系统配置更新子模块

本模块负责RSU和OBU配置更新功能。由于车联网信息发布系统内部管理的需要，需要设置信息发布系统之间，能够传递基于服务管理配置的消息。而消息的定义是由网络中心确定，并通过RSU，一级级的传递给每个信息发布系统的用户。消息的定义，消息的类型，消息的增加和删除均处于时刻变化过程中。因此，路侧单元RSU以及车载单元OBU均需要增加系统配置更新模块。利用该模块，RSU和OBU均能通过TCP通信直接或者间接的从网络中心处接收新的消息定义文件，并根据该文件更新消息定义。其获取更新的流程如下图所示。

系统配置更新模块本质上是一个TCP客户端的线程，它通过TCP连接直接连接到网络中心，获得新的消息定义配置文件，然后再更新系统的消息定义。该模块最主要的功能是通过socket连接获得的数据保存到配置文件message.dat中，该功能通过mesSave接口实现：

|  |
| --- |
| void TcpThread\_center:: mesSave ()  {  ofstream outfile("message.dat",ios::app);//输出方式打开，写入数据文件末尾  for(int i=0;i<6;i++)  {  memset(this->buf, 0, MAXDATASIZE);//每次调用，确保收到消息正确。  if((nbytes =read(sockfd, buf, MAXDATASIZE-1))==-1)  {  perror("read error");  exit(1);  }  if(nbytes == 0)  {  printf("receive EOF...\n");  break;  }  else  {  this->str\_dispose=buf;  if((str\_dispose.find("Stop")!=-1) || (str\_dispose.find("File#%#") != -1))  {  exist=true;  }  else  {  exist=this->m\_TabWidget->chbuff\_fromctcp(buf);  }  if(exist)  {  cout<<"message exist."<<endl;  }  else  {  buf[nbytes]='\n';  outfile<<buf;  }  }  }  outfile.close();  } |



图5-8系统配置更新子模块流程图

### WAVE底层通信子模块

WAVE底层通信子模块，本质上为RSU和OBU的通信驱动模块，在这里我们采用MadWifi协议为模板，定制其WAVE通信功能。WAVE底层通信子模块主要工作如下：

1. 检测设备

当设备接入到操作系统，OBU核心板会调用一个通信驱动程序中的探测函数来检测此驱动程序与新接入的设备是否匹配。探测函数根据OBU核心传入的 pci\_dev 结构体和记录着本驱动程序所支持设备的 pci\_device\_id 结构体来判断。如果两者不匹配，那么OBU核心将会调用另一个驱动程序来进行相同的操作。

1. 生成并填充 net\_device 数据结构

通过调用 alloc\_netdev函数来生成 net\_device 数据结构,并通过 ether\_setup函数完成对于net\_device 数据结构的部分初始化。在这里WAVE底层通信子模块设置保留MadWifi驱动原有的IP模块，使用IP作为网络层的协议标准。在数据链路层上，802.11n有着繁杂的身份验证和鉴权的模块，本本模块直接在该结构体中删除该功能，并相应的增加数据链路层中监测报文头部的BSSID域的功能函数，以替代MadWifi驱动的数据链路层的身份验证。

1. 注册中断处理程序

通过指定中断号和中断处理程序，在初始化函数中完成对于中断处理程序的注册工作。中断处理程序在无线网卡驱动中起到重要的作用。例如接收功能就是通过中断函数来完成的。

## 本章小结

本章主要了根据第三章提出的设计方案，从具体实现出发，详细的阐述了路侧单元RSU和车载单元OBU的客户端软件设计与具体实现。本章首先从路侧单元RSU和车载单元OBU的需求出发，介绍了RSU和OBU的软硬件平台以及开发环境Qt Creator。其次，再根据给出的系统的程序结构关系图，从系统功能类的角度介绍了客户端系统的构成。最后，在从具体功能出发，详细的介绍系统的信息分发功能子模块、Web客户端子模块、系统配置更新子模块以及WAVE底层通信子模块的具体实现，包括各子模块的功能介绍，程序流程图，及关键接口的介绍。

# [系统测试](#_Toc369890263)

## [测试目的和环境部署](#_Toc369890264)

面向车联网应用的信息发布系统，其设计的目的便是实现一个可以在道路上部署服务的用于车辆用户和路侧系统之间交流通信的信息发布系统。然则由于场地以及硬件系统所限，本文所设计并且实现部署的为车联网信息发布系统的雏形系统。部署本演示雏形系统的最主要目的，是验证本论文所设计的车联网通信系统的可行性以及其提供服务的可用性。本系统采用如下硬件设施进行部署：

1. 安装了Debian（Linux）系统的服务器一台（网络中心）

网络中心为一台Intel双核主频2.20 GHz，内存为1G的服务器，其操作系统为Debian系统，并安装部署了Apache2.2服务器，MySQL5.5数据库服务器，Python2.7的脚本解释器。

1. 安装了Ubuntu（Linux）系统的笔记本一台（路侧单元RSU）

路侧单元RSU的操作系统为Ubuntu系统，安装部署了Apache 2.2服务器，并配备NETSYS-网卡一块

1. 安装了Linux系统的Tiny6410开发板四台（车载单元OBU），

每个Tiny6410开发板操作系统为内核为Linux 2.6的嵌入式系统，通过串口外接了一个GPS传感器。

由于本系统只是简单的演示雏形系统，故车载单元外接的传感器只限于传输位置的GPS传感器，其他的如温度，车速传感器等均与GPS传感器的工作原理一致，待后续开发升级时再加入到OBU中。

## [系统测试](#_Toc369890265)

### 系统功能测试

在本节中，将对面向车联网应用的信息发布系统的三层结构，即网络中心、路侧单元RSU以及车载单元OBU分别进行功能测试。

#### 网络中心功能测试

网络中心的功能测试分为两部分，即Web服务器功能测试以及MySQL数据库功能测试。

* Web服务器功能测试

车联网信息发布系统的Web服务器主页地址为：<http://192.168.0.110/cgi-bin/homepage.py>。通过Chrome浏览器打开的界面如下所示：



图6-1车联网信息发布系统主页

主页提供了车联网系统的简介以及各类功能导航，包括：

* 交管中心：基于交通管理服务的系统配置管理的服务功能
* 广告发布：基于广告发布的资源更新管理的服务功能
* 监测管理：基于车载单元OBU监测管理的服务功能
* 服务配置：基于新服务类型管理配置的服务功能
* 系统用户登录：基于车联网Web服务器系统用户的登入管理功能

下图为用户点击了监测管理用的浏览器界面图，它能根据用户输入的车载单元OBU的ID或者路侧单元RSU的ID从数据库中调出该ID所代表的用户上传到网络中心的道路监控消息，并根据其上传的位置，在地图中标记出来。标记分成四类，分别代表四种传输消息的不同的优先级，从高往低分别是红色（A），橙色(B)，黄色(C)以及蓝色(D)。其典型功能验证案例如下图显示：

而交管中心，广告发布是进入对应系统类型的用户配置阅读并定制其特定专属服务的Web控制界面，服务配置是系统用户配置管理服务类型的Web控制界面，系统用户登录为系统用户注册或者修改系统用户消息的控制界面。



图6-2车联网信息发布系统监测管理页面

* MySQL数据库功能测试

网络中心部署的MySQL数据库，其功能的核心是存储路侧单元RSU以及车载单元OBU上传的道路消息，以及系统用户（目前只有交管中心和广告发布商）的管理配置消息，并配合Apache服务器提供基于HTTP协议的Web远程服务。

从MySQL服务器端可以看到，本系统建立了名为IOV（Internet of Vehicles）的数据库，数据库中包含了如第四章所叙述的数据库表单：系统用户信息表SystemUserInfo、车载单元信息表OBUInfo、系统服务信息表ServiceConten、车载单元监测信息表OBUData以及车载单元服务关联表ServiceRelation。MySQL数据库中的表如图所示：

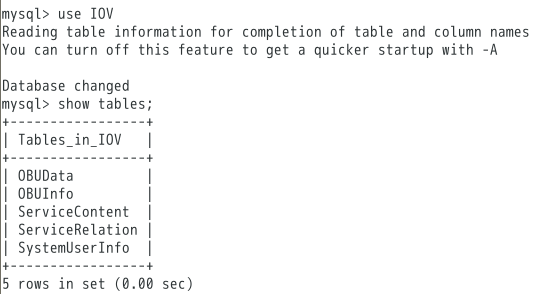
****

图6-3MySQL数据库表

#### 路侧单元RSU和车载单元OBU功能测试

位于信息发布系统第二层的路侧单元RSU（在雏形系统中包含了区域服务器功能）和第三层车载单元OBU负责整个车联网信息发布系统的消息推送和拉取工作，其本质为整个系统的消息收集者以及消息的接收者。启动的客户端软件的RSU和OBU如下图6-4所示。目前，面向车联网应用的信息发布系统主要实现了以下四类功能，即路侧单元UDP广播功能，路侧单元RSU与车载单元OBU通信功能，车载单元OBU消息上传功能以及车载单元OBU基于位置的服务功能。在本小节里面将分别对着四种典型应用做功能性测试。

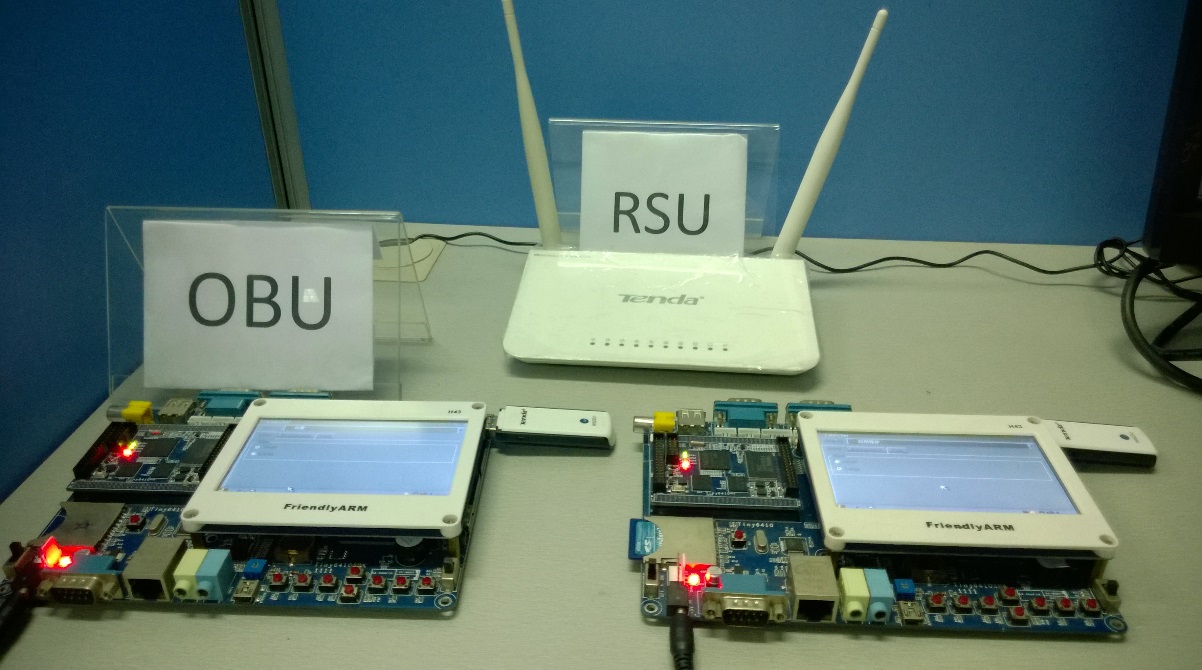


图6-4启动客户端软件的RSU和OBU

* 路侧单元UDP广播功能测试

路侧单元RSU和车载单元OBU在作为网络接入点AP时，将会担任子网络的点协调中心的重任。这时候，系统用户只需要启动客户端软件的服务端模式即可。客户端程序将自动对所覆盖范围的所有其他OBU发送UDP广播包（如下图所示）。接收到UDP广播包的车载单元OBU便知道附近有RSU或者OBU愿意充当网络接入点的职能。此时，其他的车载单元OBU将根据用户自身的需要选择是否加入该网络中。

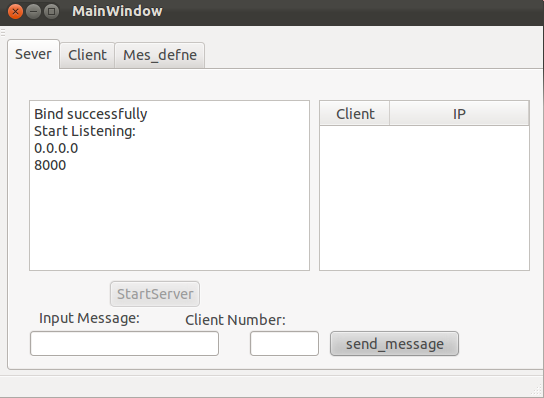


图6-5 RSU启动Server模式

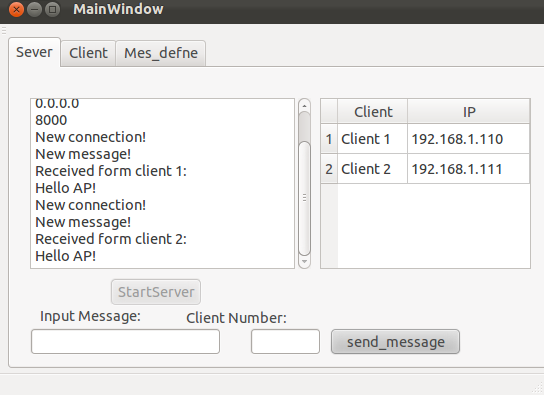


图6-6 RSU收到其他OBU的接入请求

* 路侧单元RSU与车载单元OBU通信功能测试

路侧单元RSU和车载单元OBU之间最基本的功能便是信息交互，当RSU通过UDP广播告诉周围的OBU自身存在的消息之后，周围的OBU便可以根据自身的需要和中心节点RSU建立TCP连接，并进行消息推送与拉取工作。

如下图所示：OBU1为ID为B20140001的车载单元，其IP地址为192.168.1.110；OBU2为ID为B20140002的车载单元，其IP地址为192.168.1.111；RSU1为ID为R20140001的路侧单元，其IP地址为192.168.1.109。首先，当RSU通过UDP广播自己的存在及相关信息之后，周围的存在OBU1，OBU2便与之建立TCP连接，并通过该连接同时与RSU进行信息交互。

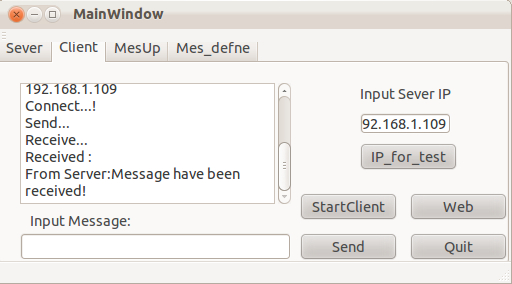


图6-7 OBU1运行图

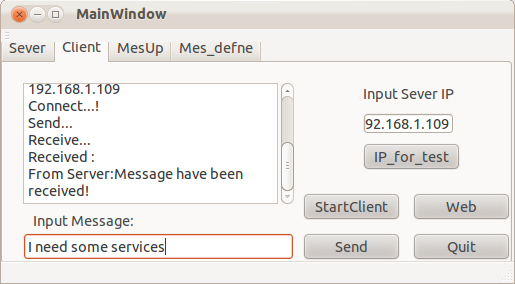


图6-8 OBU2运行图

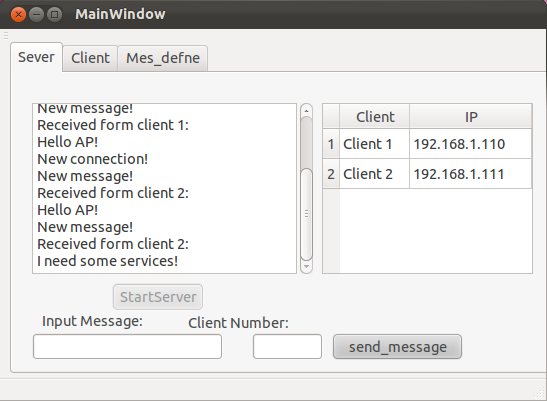


图6-9 RSU1运行图

* 车载单元OBU消息上传功能测试

我们知道车联网信息发布系统中，每个OBU是信息发布系统的用户的同时，还是该系统的信息采集者。OBU在运行中，除了会定时定量的上传车辆的一些行驶消息外，还会出现如故障、事故等一些交通事件的发生。此时，OBU可以通过软件系统的消息上传功能进行消息的推送。网络中心可以迅速的获得该安全消息，并转发给周围的其他车载单元OBU。

如下图所示，OBU推送的安全消息分成三种事件类型，即交通事故、道路状态以及车辆故障，消息的安全等级分成四种，即A特大、B严重、C重要以及D普通。OBU还可以选择是否上传自身的位置信息以及为该安全消息增加必要的摘要。

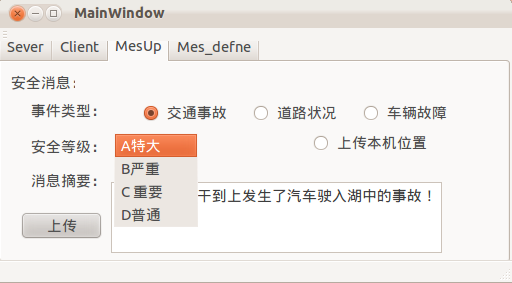


图6-10 OBU上传事故消息

* 车载单元OBU基于位置的服务功能测试

车载单元OBU可以通过客户端系统上的Web按钮，打开浏览器功能，如第五章提到的：OBU首先基于位置的从RSU中获取自身需要的资源的URL地址，然后通过定时器的形式循环访问基于位置分发的Web资源。

如下图所示：图一中OBU1获得了基于自身位置的安全消息，其中红色小车表示OBU1的位置，红色和黄色的标签为周围其他OBU上传的安全消息。图二中，OBU1通过RSU的位置访问到该RSU附近的典型娱乐场所，如餐馆、电影院、酒店等信息。



图6-11 OBU获取周围的安全消息



图6-12 OBU获取周围的娱乐信息

### 系统性能测试

在本节中，将对面向车联网应用的信息发布系统分别进行车联网网络性能测试、Web服务性能测试以及数据库访问性能测试。

#### 车联网网络性能测试

本节将利用Iperf软件工具测试本文所设计的信息发布系统网络性能。

Iperf是一个基于TCP/IP或者UDP/IP协议的网络性能测试工具，它能利用C/S通信模式，获取传输网络的吞吐容量、丢包率、传输时延等统计信息。通过Iperf获得的统计数据，可以清晰的反应出车联网信息发布系统的通信性能以及网络瓶颈。Iperf网络性能工具软件的工作方式和大多数网络性能测试工具的工作方式相同，都是以Client/Server方式工作，服务器端和客户端都使用同一程序“iperf”，服务器端使用“-s”选项，而客户端则使用“-c”选项。首先，Iperf在服务器和客户端之间建立一个测试连接，并利用该连接进行测试参数配置；紧接着它会在服务器和客户端之间不停的来回传输测试数据包，以获得测试网络的性能参数。

本测试的样机是两台Tiny6410开发板OBU，它们分别作为Iperf的客户端和服务器端启动，它们之间直线距离相隔50米。利用Iperf工具，使两开发板之间进行UDP报文通信，通过不断的增加报文数据量，测试他们的UDP延迟时间、传输带宽以及丢包率。测试的数据量分别为1Mb，5Mb，10Mb，20Mb和30Mb。重复测试两次。

此时，路侧单元RSU和车载单元OBU底层驱动均是使用定制后的MadWifi驱动。定制后的MadWifi驱动将切合WAVE通信标准。

下面是通过服务器收到的测试报告，得到以下的

第一轮测试数据：

|  |
| --- |
| [ 3] local 192.168.6.14 port 5001 connected with 192.168.6.13 port 46612  [ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams  [ 3] 0.0- 9.9 sec 1.19 MBytes 1.00 Mbits/sec 0.283 ms 3/ 852 (0.35%)  [ 4] local 192.168.6.14 port 5001 connected with 192.168.6.13 port 60888  [ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams  [ 4] 0.0-10.0 sec 5.96 MBytes 5.00 Mbits/sec 0.409 ms 0/ 4253 (0%)  [ 3] local 192.168.6.14 port 5001 connected with 192.168.6.13 port 37456  [ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams  [ 3] 0.0-10.0 sec 11.9 MBytes 10.0 Mbits/sec 0.415 ms 0/ 8505 (0%)  [ 4] local 192.168.6.14 port 5001 connected with 192.168.6.13 port 51514  [ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams  [ 4] 0.0-10.2 sec 12.9 MBytes 10.7 Mbits/sec 2.158 ms 4709/13940 (34%)  [ 3] local 192.168.6.14 port 5001 connected with 192.168.6.13 port 39281  [ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams  [ 3] 0.0-10.2 sec 13.4 MBytes 11.0 Mbits/sec 2.399 ms 4460/13998 (32%)  [ 4] local 192.168.6.14 port 5001 connected with 192.168.6.13 port 49532  **…** |

第二次测试数据：

|  |
| --- |
| [ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams  [ 4] 0.0-10.0 sec 1.19 MBytes 1.00 Mbits/sec 0.313 ms 0/ 852 (0%)  [ 3] local 192.168.6.14 port 5001 connected with 192.168.6.13 port 59349  [ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams  [ 3] 0.0-10.0 sec 5.96 MBytes 5.00 Mbits/sec 0.188 ms 0/ 4253 (0%)  [ 4] local 192.168.6.14 port 5001 connected with 192.168.6.13 port 37254  [ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams  [ 4] 0.0-10.0 sec 11.9 MBytes 10.0 Mbits/sec 0.421 ms 0/ 8505 (0%)  [ 3] local 192.168.6.14 port 5001 connected with 192.168.6.13 port 45139  [ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams  [ 3] 0.0-10.3 sec 13.1 MBytes 10.7 Mbits/sec 6.178 ms 5078/14409 (35%)  … |

从上面的测试数据，我们绘出了UDP报文测试图表，UDP延迟分析图，UDP丢包速率分析图（见下面三图）。

图6-13 UDP报文测试图

图6-14 UDP延迟分析图

图6-15 UDP丢包速率分析图

从这三幅图可以看出，使用了定制的MadWifi驱动的车载单元OBU之间，点对点的通信中，传输数据量大概为13.1 MBytes的时候有很好的传输质量保证，丢包率基本为0%，数据传输延迟为0.1ms左右。但随着数据传输量的增加，在大于13.1 MBytes的数据时，延迟开始明显上升，其实际传输带宽保持在12Mbits/s左右不变，丢包率则在30%左右，通信质量大大降低。

也就是说，本论文设计的面向车联网应用的信息发布系统，在使用了定制的MadWifi驱动，在达到符合WAVE通信标准后，系统的点对点通信带宽，在理想状态下约为12Mbits/s（WAVE协议标准为3 MBits/s~27 MBits/s）。系统在低于该带宽下才能保证数据的有效传输，高于12Mbits/s后数据就出现明显的丢包和延时现象。

#### Web服务性能测试

利用Apache服务器附带的压力测试工具ApacheBench对系统进行Web服务器进行性能测试。ApacheBench工具能根据用户设置的命令，模拟HTTP请求，向网络中心的服务器发起并发连接请求。

下面是利用ApacheBench模拟100个用户向服务器并发100的HTTP请求（请求URL为：http://localhost/cgi-bin/login.py）后，ApacheBench的输出结果：

|  |
| --- |
| This is ApacheBench, Version 2.3 <$Revision: 655654 $>  Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/  Licensed to The Apache Software Foundation, http://www.apache.org/  Benchmarking localhost (be patient)  Server Software: Apache/2.2.22  Server Hostname: localhost  Server Port: 80  Document Path: /cgi-bin/login.py  Document Length: 566 bytes  Concurrency Level: 100  Time taken for tests: 0.353seconds  Complete requests: 100  Failed requests: 0  Write errors: 0  Total transferred: 72100 bytes  HTML transferred: 56600 bytes  Requests per second: 283.08 [#/sec] (mean)  Time per request: 353.329 [ms] (mean)  Time per request: 3.533[ms] (mean, across all concurrent requests)  Transfer rate: 204.06 [Kbytes/sec] received  Percentage of the requests served within a certain time (ms)  50% 252  66% 304  75% 322  80% 335  90% 346  95% 350  98% 351  99% 352  100% 352 (longest request) |

从上面的输出结果可知，此次压力测试，共发送72100字节的数据，实际传输HTML字节56600，服务器吞吐量为283.08请求/秒。用户平均请求等待时间353.329ms，服务器平均处理时间为3.533毫秒，用户每秒接收的数据量为204.06KB。

根据上面的结果，我们可以得出HTTP请求处理时间完成状态图，如下图所示。根据该图表，我们可以看出100个并发请求，50%都在250ms内完成，其平均的完成时间在300ms左右。

图6-16 HTTP请求处理完成时间图

从上述的压力测试可以看出，服务器的平均等待时间约在300ms，服务器平均吞吐量为280请求/秒，能满足车联网信息发布系统Web服务的基本要求。按照单个OBU的正常访问频率，每台服务器能够支撑200~500台OBU同时在线访问。

#### 数据库访问性能测试

利用MySQL数据库自带的压力测试工具mysqlslap，与ApacheBench工具类似，mysqlslap可以模拟用户并发登入使用数据库，生成scheme以及查询更改数据。为了精确的对MySQL数据库进行用户流量上限以及抗压性进行性能测试，本系统将利用mysqlslap分别模拟25,50,75,100,125,150,175,200,225,250,275,300的系统并发访问量对MySQL数据库进行压力测试，测试的命令如下所示：

|  |
| --- |
| [root@localhost ~]# mysqlslap -uroot -porange  --concurrency=25,50,75,100,125,150,175,200,225,250,275,300  --iterations=10  --auto-generate-sql  --auto-generate-sql-load-type=mixed  --auto-generate-sql-add-autoincrement  --engine=myisam  --number-of-queries=10  #该命令的含义是，利用mysqlslap，模拟并发量用户为25-300，系统自动生成sql命令对系统表数据进行读写测试，重复访问10次，每次执行10次查询命令，测试引擎为myisam |

重复压力测试十次，得到的MySQL性能测试数据如下表所示：

表6-1 MySQL并发测试响应时间表

|  |  |  |  |
| --- | --- | --- | --- |
| 并发量 | 平均时间（单位：s） | 最短时间(单位：s) | 最长时间(单位：s) |
| 25 | 0.066 | 0.032 | 0.288 |
| 50 | 0.081 | 0.076 | 0.105 |
| 75 | 0.123 | 0.119 | 0.126 |
| 100 | 0.167 | 0.154 | 0.206 |
| 125 | 0.199 | 0.189 | 0.208 |
| 150 | 0.252 | 0.238 | 0.261 |
| 175 | 0.293 | 0.283 | 0.307 |
| 200 | 0.326 | 0.311 | 0.342 |
| 225 | 0.371 | 0.36 | 0.381 |
| 250 | 0.426 | 0.404 | 0.5 |
| 275 | 0.465 | 0.441 | 0.508 |
| 300 | 0.503 | 0.476 | 0.515 |

根据上表可以画出MySQL不同并发量的访问时间散点图，如下所示：

图6-17 MySQL并发测试处理时间图

其次，位于网络中心部署的MySQL数据库，使用频率最高的莫过于OBU上传存取车载单元监测信息的OBUData表。为了测试MySQL库中OBU频繁访问查询MySQL数据库的情况下，OBUData表表单大小对系统性能的影响。系统在此将利用mysqlslap分别模拟25,50,75,100,125,150,175,200,225,250,275,300的系统并发访问量对MySQL数据库OBUdata表的进行查询压力测试，表单的大小分别为10,100,1000,10000，测试的命令如下所示：

|  |
| --- |
| [root@localhost ~]# mysqlslap -uroot -porange --concurrency=25,50,75,100,125,150,175,  200,225,250,275,300 --iterations=10–create-schema=”IOV” –query=”select \* from OBUData where Humidity= “56C” --number-of-queries=10  #该命令的含义是，利用mysqlslap，模拟并发量用户为25-300，对数据库IOV中的OBUData 表执行Select查询命令，重复访问10次，每次执行10次查询命令 |

通过测试前，调整OBUData表单的大小，得到的MySQL性能测试数据如下表所示：

表6-2 MySQL并发测试响应时间表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 并发量 | 查询时间  表单大小10  （单位：s） | 查询时间  表单大小100  （单位：s） | 查询时间  表单大小1000  （单位：s） | 查询时间  表单大小1000  （单位：s）) |
| 25 | 0.016 | 0.017 | 0.026 | 0.203 |
| 50 | 0.018 | 0.021 | 0.035 | 0.373 |
| 75 | 0.028 | 0.03 | 0.059 | 0.545 |
| 100 | 0.037 | 0.041 | 0.072 | 0.773 |
| 125 | 0.049 | 0.052 | 0.092 | 1 |
| 150 | 0.059 | 0.065 | 0.11 | 1.217 |
| 175 | 0.071 | 0.078 | 0.128 | 1.349 |
| 200 | 0.081 | 0.091 | 0.146 | 1.459 |
| 225 | 0.094 | 0.101 | 0.167 | 1.521 |
| 250 | 0.107 | 0.116 | 0.186 | 1.666 |
| 275 | 0.116 | 0.126 | 0.209 | 1.8 |
| 300 | 0.13 | 0.14 | 0.23 | 1.885 |

图6-18 MySQL查询OBUdata表单性能测试图

从MySQL并发测试处理时间图可以看到，数据库响应时间随着系统的并发量的增大而增大，其平均访问时间成线性增长模式。由于模拟的每个用户分别进行10次读写操作，当并发量为100-200时，用户的平均每次读写操作的耗时大概为15ms-31ms，平均等待时间为0.15s到0.31s。系统的平均等待时间，符合车载通信中对时延的容忍的合理范围。通过压力测试数据表明，每台服务器上的MySQL数据库能支持100-200个OBU并发持续的读写数据库操作。

当数据库并发量为25时，最长处理时间出现一极端值，从该值可以推测MySQL数据库对重复查询做了优化。因此，对某些相似的数据库查询命令可由上层软件系统进行优化，集中进行数据库查询访问，减少系统访问处理时间。

从OBUdata表单性能测试图可以看出，随着OBUData表单的性能随着表单的大小逐步降低，MySQL处理时间逐步升高。如在并发量200的情况下，遍历OBUData表查询1次消耗的时间，分别为8.1 ms, 9.1 ms, 14.6 ms, 145.9 ms。因此，表单的数量在维持在1000的数量级以内，数据的遍历查询响应时间均非常短，特别是100-1000时，查询效率最高，但数量级超过万以后，表单的查询质量将急剧下降。因此，网络中心的数据库中的表单应尽可能保持在100-1000范围内，超过1000的表单可以根据OBUID的哈希值进行分表操作，以提高MySQL的并发处理速度，以满足车联网通信系统实时性的要求。

## 车联网系统对比分析

在上一节主要对本文所提出的面向车联网应用的信息发布系统进行了功能层面和性能层面的测试。在本节，将会根据上节的功能测试结果和性能测试结果，与传统的车联网系统进行对比，分析本文所设计的车联网信息发布系统的优点。

### 系统功能对比

在功能层面上，上节先对系统的网络中心Web服务器的用户登入登出、交通服务配置、广告资源配置以及OBU监测管理等进行了验证。其次，再对系统的RSU和OBU之间的TCP数据传输、UDP广播接收、安全信息上传查阅以及基于OBU位置的商业服务功能进行了验证。从功能层面看，传统的车联网信息发布系统主要有以下两点缺陷：

* 业务单一

每类系统均只注重单一的服务，如基于射频识别技术的信息发布系统，主要专用于高速收费路段的不停车收费业务；基于移动基站提供的2G/3G通信方式的信息发布系统，主要专用于交通信息查询业务；基于ZigBee等低速率传输设备的信息发布系统，主要专用于道路交通状况实时监测业务；基于Wifi通信的客户端-服务器模式的信息发布系统，主要专用于车载单元娱乐信息服务业务。

* 信息传输单向性

传统车联网信息发布系统，消息的流向非常单一，信息是单向传输的，如现在运用最广泛的ETC电子收费系统，信息单一的由车辆的用户信息流向收费站，收费站无法传输信息给用户；又如交通信息查询业务，只能由车载用户查询服务器的交通信息，车载用户无法上传自身的消息。

通过6.2.1小节的的车联网信息发布系统功能验证，可以清晰的看出，本文设计的基于三层架构的车联网信息发布系统，涵盖了交通道路安全服务、交通数据采集分发服务以及娱乐信息服务三大业务。它除了能提供与传统的信息发布系统相似的电子自助收费服务、道路状况监测服务等之外，还由于多了一层路侧单元和区域服务器（本雏形系统将区域服务器的功能合并到RSU中），使其可以方便的为OBU提供各种基于位置的服务。如本系统所验证的基于位置的交通安全提醒功能以及基于位置的广告服务功能。并且，本文设计的车联网信息发布系统，信息的流向是双向的，OBU可以随时上传自己的信息给网络中心或者RSU，OBU也可以随时向网络中心或者RSU拉取自己需要的信息资源。

由此可见本文所设计的车联网信息发布系统，从功能角度出发，突破了旧式车联网系统功能单一以及信息传输单向性的局限，服务功能具有多样性，且信息为双向传递的。系统中的车载单元OBU既是系统信息的产生者也是系统信息的消费者。

### 系统性能对比

在性能层面上，本文首先对系统RSU和OBU端WAVE底层通信子模块的网络性能进行了性能测试，其次再对网络中心的Web服务性能和MySQL访问性能进行了压力测试。从测试结果得出，本论文设计的面向车联网应用的信息发布系统，在使用了符合WAVE通信标准定制MadWifi驱动后，系统的点对点通信带宽，在理想状态下约为12Mbits/s，传输时延约为10 ms。网络中心配置的每台服务器（CPU2.20GHz，1G内存）均能同时处理200-500个OBU并发访问，其Web服务器的平均等待时间在100ms级别，平均吞吐量为280请求/秒，MySQL用户的平均每次读写操作的耗时大约处在10 ms 级别，平均等待时间为0.15s到0.31s。

下表6-3是传统的车联网信息发布系统与本论文设计的车联网信息系统的性能对比，通过该表可以清晰的看出，本文所设计的车联网信息发布系统，在保持低延时、高稳定性、高并发量的前提下，能通过WAVE通信标准大大增加的车联网的网络传输质量，为上层交通道路安全服务、交通数据采集分发服务以及娱乐信息服务三大业务提供了可靠的数据传输保证。与其他车联网系统相比，本文所设计的系统其网络通信质量与网络中心的高并发处理能力是其最大的优势。

表6-3车联网信息发布系统性能对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 通信模式 | 传输速率 | 传输距离 | 系统稳定性 | 系统并发性能 |
| 基于ZigBee | 20～40 kbp | 10～100 m | 一般 | 中心节点模型（基于轮询，无需并发） |
| 基于高频RFID | 100kbps | 1 m以内 | 高 | 并发能力差 |
| 基于2G/3G | 144kbps~384kpbs | 1000m以上 | 低 | 并发性能高（基于基站） |
| 基于WAVE | 12Mpbs | 200m以内 | 高 | 并发性能高 |

## 本章小结

本章主要了根据第四、五章的信息发布系统的设计和实现，从功能以及性能出发，对整个面向车联网应用的信息发布系统进行基本的测试。

在功能测试方面，首先，从系统外部通过浏览器访问网络中心Web服务器，验证其功能的可用性。其次，再通过RSU和两个OBU进行通信访问，测试系统RSU和OBU客户端软件的功能可用性。

在性能测试方面，首先，利用Iperf工具测试OBU的网络性能。Iperf通过OBU之间进行不断的增加报文数据量的UDP报文通信，来检测网络的UDP延迟时间，传输带宽以及丢包率；其次，利用ApacheBench和mysqlslap压力测试工具，模拟用户高并发量对系统进行压力访问。最后，在通过与旧式的车联网系统进行对比，分析本文论述的面向车联网应用的信息发布系统的优势所在。通过测试结果数据，可以看出本文提出的面向车联网应用的信息发布系统，网络传输能力足以满足大数据时代高信息容量传输服务要求，并且单个服务器性能稳定高效，能同时满足并发量为200-500的用户持续的访问。

# 总结与展望

本文在基于车联网研究领域中，车联网信息系统目前普通存在通信质量差，传输速率低，系统并发访问能力不足，业务单一的现状下，立足于智能交通系统中车联网应用三大服务需求，即交通道路安全服务，交通数据采集分发服务以及娱乐信息服务，并根据车联网通信环境快速多变的特点，以及车联网信息分发的特征和需求，研究一种新型的面向车联网应用，适应车联网业务环境的的信息发布系统。

本文的研究工作主要围绕着车联网应用的服务需求展开，主要包括WAVE协议栈研究、LAMP架构模型研究、车联网信息发布系统的总体架构与功能研究，信息发布系统各层次核心模块的设计这几个方面。本文详细论述了WAVE协议栈框架、WAVE通信特点、WAVE传输功能、LAMP架构特性以及LAMP各层次的技术特征，并结合车联网服务业务需求，分析了融合WAVE技术和LAMP架构技术搭建车联网信息发布系统的众多优点，如稳定性、高效性、高并发量并且与车联网通信环境的高适用性等，并在本文的末尾通过本雏形系统的功能测试和性能测试对比验证了该结果。

本文主要针对车联网应用信息发布系统与WAVE协议栈、LAMP架构等技术的结合做深入的研究探讨。本文论述的系统作为车联网信息发布系统的雏形，对于实现真正商业化运营的车联网应用系统具有一定的参考价值。本文所论述的车联网雏形系统主要取得如下的成果：

1）利用WAVE协议栈，作为车联网信息发布系统移动设备的无线通信协议，解决了车联网信息发布系统移动设备在快速移动中通信质量差的问题，保证了信息发布系统信息分发的可靠性与稳定性。

2）摈弃了传统车联网信息发布系统采用服务器-客户端（C/S）架构，采用了针对移动车载环境的网络中心-路侧单元-车载单元的三层架构体系，使车联网信息发布系统更贴合实际的车载移动环境，增加了系统进行信息分发的效率和精确度，并使得系统能精确有效的实现各种基于OBU位置的服务业务。

3）利用LAMP架构部署网络中心的动态网络应用服务平台,大大增加了网络中心的并发性能与可维护可扩展性能，解决了后续商业化车联网信息发布系统后，大数据量的OBU对网络中心的负担。

但是本文所提出的雏形系统，由于作者本人水平的限制，系统的实现并不是十分完善，系统的稳定性、可靠性仍然需要进一步修正。加之系统并没有真正部署到高速公路等原因，系统的功能内容丰富度仍然未达到商用级别的要求。本文所提出的面向车联网应用的信息发布系统，只是一个雏形系统，离真正能商业级别的应用系统还有很大的差距，系统很多方面还需要不断的完善，主要有以下几点：

1）目前上，并没有直接硬件上实现WAVE协议栈，大多数均是从软件程度上模拟WAVE通信协议，因此需要后续从硬件上开发出基于WAVE协议栈的无线网卡，并把它运用到本系统中，进一步提高车联网无线质量。

2）由于本系统只是雏形系统，车载单元OBU功能相对单一，只有访问Web广告，接收和上传的安全警报等简单功能。因此，系统的后续需要基于本文提出的交通道路安全服务，交通数据采集分发服务以及娱乐信息服务需求进行功能优化。此外本系统需要与现有的智能交通系统，如ETC电子收费系统等系统无缝对接，以最大程度的整合交通资源。

3）由于车联网信息发布系统，涉及了许多车联网用户的敏感信息，如车辆的牌照，车辆的位置，收费账户等。因此，本系统需要在后续开发基于敏感信息的加密传输模块。

4）本系统真正商用后，将涉及海量交通数据的分发处理工作。基于海量数据的的分布式存储算法，以及相关的以海量数据为基础的数据挖掘算法和交通控制算法将会是下一步系统的研究重点，这将会给车联网的用户和管理者带来极大的收益。

总之，本文为车联网信息发布系统的发展和应用提供了一定的实用参考价值，未来可以从上述的四个方面对车联网信息发布系统进行完善和升级，使之能真正的应用到智能交通领域中，为智能交通系统的发展起到促进作用。

# 参考文献

1. 宋俊德.浅谈物联网的现状和未来[J]. 移动通信,2010 年第 15 期:8-10
2. 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年 第 2期:111-114
3. IEEE Std 802.11p, IEEE Standard for Information technology—Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications-Amendment 6: Wireless Access in Vehicular Environments [S]. IEEE，2010
4. 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
5. IEEE Std 1609.0-2012,IEEE Draft Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture. IEEE P1609.0/D5[S]. IEEE, 2012
6. 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
7. IEEE Std 1609.2-2006,IEEE Trial- Use Standard for Wireless Access in Vehicular Environments —Security Services for Applications andManagement Messages[S]. IEEE,2006
8. IEEE Std 1609. 3-2010,IEEE Standard for Wireless Access in Vehicular Environments (WAVE)-Networking Services[S]. IEEE, 2010.
9. IEEE Std 1609. 4-2010,IEEE Standard for Wireless Access in Vehicular Environments (WAVE)-Multi-channel Operation[S]. IEEE, 2011
10. 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,2011，2:19-22
11. 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
12. 物联网信息技术与产业化省部院产学研联盟 《广州市物联网交通应用示范建议》 2010
13. Mattbew S.Gast. 802.11Wirelss Networks：The Definitive Guide [M]. O'Reilly,2007.12
14. 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第28卷第11期：116-124
15. 韦东山.嵌入式Linux应用开发[M]. 人民邮电出版社,2003
16. 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
17. 东南大学.802.11无线网络[M]. 东南大学出版社,2006
18. 唐波. WAVE架构及相关协议设计与实现[D]. 华东理工大学,2009
19. 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
20. 百度文库.Madwifi详解[EB/0L].[http://wenku.baidu.com/view/541d8307e87197](http://wenku.baidu.com/view/541d8720af45b307e87197)
21. Chrisian Benvenuti,Understanding Linux Network Internals[M],O'Reilly,2005
22. 刘富强,单联海. 车载移动异构无线网络架构及关键技术研究[J].中兴通讯技术，2010 Vol.16 No3
23. 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
24. 杨瑞，工程车辆联网系统及软件平台设计[D]，浙江大学，2012.05
25. 艾利锋，Web Services技术在行车安全综合监控系统整合中应用的研究[D]，铁道科学研究院，2006
26. 顾振飞，车联网系统架构及其关键技术研究[D]，南京邮电大学，2012
27. Hartenstein, H.; Laberteaux, K.P. A tutorial survey on vehicular ad hoc networks[J]. Communications Magazine, IEEE,vol.46, no.6, pp.164-171, June 2008
28. Daniel Q.Chen, 面向服务的行业解决方案[M]，电子工业出版社
29. 杨涛，刘锦德.Web Services技术综述一种面向服务的分布式计算模式[J]，计算机应用，2004，24(8):1-4.
30. 杨斌，张卫冬，张利欣，章立军，时鹏，基于SOA的物联网应用基础框架[J]，计算机工程，2010，36(17)
31. 郭秀娟，基于SOA体系结构的服务组件实现相关技术研究[D]，沈阳理工大学，2011
32. Schroth, C.; Janner, T.; , "Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services," IT Professional , vol.9, no.3, pp.36-41, May-June 2007
33. Christian Nagel, Bill Evjen, Jay Glynn等著，李敏波译，C#高级编程[M]，清华大学出版社
34. 维基百科，LAMP[Z/OL]，[http://zh.wikipedia.org/wiki/](http://zh.wikipedia.org/wiki/Xml)LAMP
35. 维基百科，MySQL[Z/OL]，http://zh.wikipedia.org/wiki/MySQL
36. 维基百科，Debian[Z/OL]，[http://zh.wikipedia.org/wiki/](http://zh.wikipedia.org/wiki/UDDI) DEBIAN
37. Wikipedia，Intelligent Transport System, [EB/OL], http://en.wikipedia.org/wiki/Intelligent\_transportation\_system，2008
38. Minglu Li; , "Metropolitan VANET: Services on the Road," High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on , vol., no., pp.17, 25-27 Sept. 2008
39. Jameel, A.; Stuempfle, M.; Jiang, D.; Fuchs, A.; , "Web on wheels: toward Internet-enabled cars," Computer , vol.31, no.1, pp.69-76, Jan 1998
40. 艾利锋，刘春煌，蒋荟，Web Services在行车安全综合监控系统中的应用研究[J]，铁路计算机应用，2006,15(4):4-7
41. Yi Wang, Akram Ahmed, Bhaskar Krishnamachari. IEEE 802.11p Performance Evaluation and Protocol Enhancement[C]. 2008 IEEE International Conference on
42. Vehicular Electronics and SafetyColumbus, OH, USA. September 22-24, 200
43. L. Stibor, Y. Zang and H-J. Reumermann. Evaluation of communication distance of broadcast messages in a vehicular ad-hoc network using IEEE 802.11p[C]. in Proc. IEEE Wireless Communications and Networking Conf., Hong Kong, China, Mar. 2007, pp. 254-257.
44. S. Eichler. Performance evaluation of the IEEE 802.11p WAVE communication standard[C]. in Proc. IEEE Vehicular Technology Conf., Baltimore, MD, US, Oct. 2007, pp. 2199-2203
45. Katrin Bilstrup,Elisabeth Uhlemann, Erik G. Ström. Evaluation of the IEEE 802.11p MAC method for Vehicle-to-Vehicle Communication[C]. [Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4656831)
46. Stephan Eichler. Performance Evaluation of the IEEE 802.11pWAVE Communication Standard[C]. [Vehicular Technology Conference](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4349648), Sept. 30 2007-Oct. 3 2007
47. Jong-Moon Chung. Time Coordinated V2I Communications andHandover for WAVE Networks[J]. [Selected Areas in Communications,March 2011](http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=49)
48. Performance Evaluation of IEEE 1609 WAVEand IEEE 802.11p for Vehicular Communications[C]. [Ubiquitous and Future Networks (ICUFN), 16-18 June 2010](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5535006)
49. [Karagiannis, G.](http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Karagiannis,%20G..QT.&searchWithin=p_Author_Ids:37339824600&newsearch=true)  Vehicular Networking: A Survey and Tutorial on Requirements,Architectures, Challenges,Standards and Solutions[J]. [Communications Surveys & Tutorials, IEEE](http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=9739), Fourth Quarter 2011

攻读硕士学位期间取得的研究成果

在硕士研究生期间，本人以第一学生作者身份申请发明专利一篇：

1. 余荣，谢胜利，张洁柯， etc. 一种基于车联网的信息发布系统

（发明专利，已受理，申请号：201210588415.6）

致谢

蓦然回首，三年的硕士生涯转瞬即过。三年来，有过憧憬，也有过茫然。已经记不清经历过多少困难与挑战，记不清经历过多少次通宵搭通宵达旦阅读文献、部署系统、调试程序。如果要用一句话来总结我的研究生生涯的话，我想“一分耕耘，一分收获”再贴切不过了。忘不了老师们的谆谆教诲，忘不了同学们的深厚友谊，更忘不了家人无微不至的关怀。是你们，让我从一个懵懵懂懂的少年蜕变成如今些许成熟些许稳重的我。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傅予力教授，傅老师是一位德才兼备的老师，颇受学生的爱戴，我很幸运能够在傅老师的指导下进行研究生阶段的学习。傅老师渊博的知识、严谨的治学态度、低调的处事作风，都深深地影响了我，他不仅教给我专业知识，还教会我许多做人的道理，这些都将让我受用终生，在此，衷心地感谢傅老师。

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，感谢您让我参与到“车联网”的研究项目当中，让我第一次接触到了当时刚刚兴起的WAVE协议栈，踏入了智能交通系统这片广阔的天地中。

感谢实验室的谢胜利教授和其他的各位老师，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。

在论文写作期间，参考了许多相关书籍和文献，在此向这些作者表示谢意。此外，整个车联网系统雏形的搭建，还要感谢邱团准师兄、周芳芳、陈钦波们一起共同努力。

感谢实验室的同学们（城少、钦波、老蒋、阿波、段哥、妍蓉、小强、文龙、启辰、奎文、芳芳），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。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。

特别感谢我的家人和女朋友，感谢你们的支持、理解与包容，正是这无微不至的关怀和鼓舞，给予我无比的动力与力量，坚持撑过了赶论文的日子，是你们教会了我责任与担当，沉稳与踏实！

“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！

1. [↑](#endnote-ref-2)
2. [↑](#endnote-ref-3)
3. [↑](#endnote-ref-4)
4. [↑](#endnote-ref-5)