ABSTRACT

ABSTRACT: Aiming at the problem of mass traffic data transmission from mass multipath and different type sensors existed in urban road traffic state obtaining technology, based on the sensor network system to get traffic state suggested in the project, this thesis introduces an technique of information transmission, which is composed of the CAN bus locating lower and Ethernet locating upper, and study the embedded Ethernet system mostly.

Based on the demand of embedded Ethernet from sensor network, some works have been completed. First, transmission demands of mass information are put forward, and tactic of embedded Ethernet are established. Second, hardware system based on DSP and Ethernet Controller is designed. Drive of lower network is realized, including initialization of system, sending and receiving data through programming. Third, TCP/IP is embedded based on hardware system, TCP and UDP are realized aiming at different case. Last, traffic road is simulated in lab, and combining the other function of the network system, functions of Ethernet designed are validated by experiments.

The embedded Ethernet communication functions of the system are verified by sufficient experiments. Feasibility and Reliability of the Ethernet system are validated, when UDP and TCP are used in different case, by different network experiments. The whole process of data collection, detection, processing, and transmission are completed, combining sensors information detection function, CAN bus communication function and human-computer interface software developed by fellows. The sensor network for getting traffic state is formed in the lab, and can be used in road traffic system.

KEYWORDS: traffic state acquisition; sensor network; embedded Ethernet; DSP; TCP/IP

CLASSNO: TP393.03

索引

ARPA	Advanced Research Projects Agency	高级研究计划管理局(美国国防部)
ARP	Address Resolution Protocol	地址解析协议
AUI	Attachment Unit Interface	连接单元接口
CAN	Controller area network	控制器局域网
CSMA/CD	Carrier Sense Multiple Access with Collision Detection	载波监听多路访问及冲突检测技术
DNS	Domain Name System	域名系统
DSP	Digital Signal Processor	数字信号处理器
FSM	Finite State Machine	有限状态机
FTP	File Transfer Protocol	文件传输协议
HTTP	Hypertext Transfer Protocol	超文本传输协议
ICMP	Internet Control Message Protocol	因特网控制报文协议
IGMP	Internet Group Management Protocol	网际组管理协议
IP	Internet Protocol	网际协议
ISA	Industry Standard Architecture	工业标准接口
ITS	Intelligent Transportation Systems	智能交通系统
LAN	Local Area Network	局域网
MAC	Medium Access Control	媒体访问控制
MIPS	Million Instructions Per Second	每秒百万条指令
MTU	Maximum Transfer Unit	最大传送单元
RARP	Reverse Address Resolution Protocol	逆向地址解析协议
RTO	Retransmission Time-Out	重传超时
SCATS	Sydney Coordinated Adaptive Traffic System	悉尼协调自适应交通系统
SCOOT	Split, Cycle and Offset Optimization Technique	绿信比、周期和相位差优化技术
SMTP	Simple Message Transfer Protocol	简单邮件传送协议
SNTP	Simple Network Time Protocol	简单网络定时协议
TCP	Transmission Control Protocol	传输控制协议
UDP	User Datagram Protocol	用户数据报协议

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：　　签字日期：　　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北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：　　导师签名：

签字日期：　　年　月　日　　签字日期：　　年　月　日

致谢

本论文的工作是在我的导师张和生副教授的悉心指导下完成的，张和生副教授严谨的治学态度和科学的工作方法给了我极大的帮助和影响。在此衷心感谢两年半年来张老师对我的关心和指导。

张和生副教授悉心指导我们完成了实验室的科研工作，在学习上和生活上都给予了我很大的关心和帮助，在此向张老师表示衷心的谢意。

张和生副教授对于我的科研工作和论文都提出了许多的宝贵意见，在此表示衷心的感谢。

在实验室工作及撰写论文期间，叶华、郑巨明等同学对我论文中的嵌入式以太网研究工作给予了热情帮助，在此向他们表达我的感激之情。

另外也感谢家人，他们的理解和支持使我能够在学校专心完成我的学业。

1 引言

本章简要叙述了课题的项目背景和研究意义，介绍了交通状态获取技术的国内外研究现状，以及作为本课题核心的嵌入式以太网的技术背景，阐明了针对交通状态获取来构建传感器网络的必要性和可行性。

1.1 课题的项目背景与研究意义

本课题隶属于“科技部国家 863 计划”项目“交通状态获取的新型传感器、传感器网络优化与融合”(项目编号 2006AA11Z231)中的子课题“传感器网络复合节点、组网设备及传感器融合系统”。

随着现代社会经济的高速发展，交通问题日益成为一个突出的社会问题。在智能交通系统中，可靠有效的交通状态信息是进行交通流控制和诱导等交通管理的重要前提，是制定交通安全管理策略、交通事故检测、交通事故致因分析等交通安全保障措施的必要基础，是交通基础设施管理、监控和维护不可或缺的第一手资料。在智能交通系统中，道路交通状态的获取是交通管理、交通安全保障和交通基础设施监控维护的基础性问题[1]。

要更好地实现对城市道路交通的控制，就必须获取大量的交通状态数据来为上层控制策略提供足够的信息，这又要求道路交通控制系统必须构建更加可靠的交通数据网络来实现数据的高效传输。因此，面对智能交通系统构建城市道路交通数据网络的需求，如何更可靠更有效地传输这些大量数据和信息将直接影响到道路交通状态获取的可行性和可靠性，并对智能交通系统的任务和需求提出挑战。

交通状态获取的物理基础是物理传感器。目前道路交通系统中所用传感器类型多、差异大，如用于道路交通管理的传感器包括电感（环形线圈）、雷达（测速仪）、红外（测速仪）、超声（测速仪）等传感器；用于道路交通安全保障的传感器包括红外（测速仪）、雷达（测速仪）、应力应变(测重仪)、超声(测高仪)、气体（酒精检测仪）等传感器；用于道路基础设施监控的传感器包括温度、应力应变(测重仪)等传感器[1][2]。

而道路交通系统所用传感器通常仅面向一两个固定应用，并不完全具备自适应、多用途、可组网的功能，大量不同种类的传感器的存在，就不可避免的造成了底层检测环节在检测处理方式和接口方式上的多样性和复杂性以及由此产生的海量异构数据，而且各类传感器的部署相互独立，分属不同系统，无法直接进行数据交互并提供综合立体的交通信息。

以北京市 SCOOT 系统为例，遍布于城区主要交通干线上的 1000 多个传感器每月产生的数据量达到几十 GB[3][4]。美国圣安东尼奥市附近的高速公路(46km)上的传感器每天产生的数据量为 120MB[5]。在这种情况下，传统的城市道路交通控制系统所采用的中央处理计算机集中接入和控制模式面临着巨大的信息传输和处理压力。

因此，现有的交通状态信息获取模式对数据传输网络的巨大压力、对后台系统处理能力的极高要求以及将导致的日益增加的面向全局综合应用的复杂集成等因素都迫切需要一种新型的道路交通传感器网络系统，来构建新问题下的道路交通控制系统。

“交通状态获取的新型传感器、传感器网络优化与融合”项目正是在此背景下，利用先进的电子检测技术、现场总线技术、以太网技术和微处理器技术等，通过研究和使用新型的网络化智能交通检测传感器、分层式的信息传输网络以及区域交通状态信息融合获取技术使交通检测传感器网络化，实现一种新型的分布式交通状态获取方式。

针对实际道路交通控制系统中传感器数据的有效传输问题，本项目提出了一种网络化分层次的信息传输技术，通过构建现场总线和以太网相结合的数据网络来实现对大量交通状态数据的传输，并通过在上下层网络之间构建前端数据处理节点来减轻上层系统的数据处理压力，实现数据的灵活传输。

交通状态获取的嵌入式以太网系统是针对构建交通数据网络时所面临的系统资源受到限制的情况，对现有的完整的 TCP/IP 的协议进行的简化，它在结合道路交通控制系统的需求基础之上，以提高以太网系统的灵活性和实时性为原则，并考虑到降低成本等因素，保证了海量数据的可靠传输。

由于构建传感器网络系统庞大且实现复杂，因此，本课题论文作为本项目的一个子部分，主要研究构建传感器上层网络的嵌入式以太网通信技术。

1.2 交通状态获取的国内外研究现状

1.2.1 交通状态获取技术

日益严重的交通问题已经成为城市发展的主要瓶颈，智能交通被认为是缓解城市道路交通拥堵问题的有效途径，交通状态获取作为智能交通实现的基础，对于交通预测，拥挤预防，避免交通紊乱等都发挥着重要的作用。道路交通状态的变化本质上是交通流的时空变化，不同的交通流特征将对应不同的交通状态，因此交通状态是伴随着交通系统运行产生的一种动态交通流状态或网络模式[5]。

到目前为止，国际上尚无统一的城市交通状态的判别定义[5]，但交通状态获取技术却很早就出现了。1930 年代初，美国和英国就相继采用气动橡皮管作为车辆检测器，用来获取道路交通状态以控制交通信号灯。随后，雷达、超声波、光电、地磁、电磁、微波、红外传感器和环形线圈等都相继用于道路交通控制中。1970 年代，英国运输研究所开发出了基于交通状态的自适应交通控制系统 SCOOT，目前，在全世界有超过 200 个城市正运行着该系统。进入 1980 年代，由于自动检测技术、数据通讯技术、自动控制技术、智能信息处理与决策等技术的飞速发展，新兴的智能交通系统（Intelligent Transportation Systems，简称 ITS）成为解决交通问题的最佳途径，而交通状态获取则是该系统重要的基础性工作[1][5]。

进入 21 世纪后，一些新技术如下一代互联网、无线传感器网络、移动组网技术、智能车辆传感器技术等的出现，为智能交通系统的快速发展与广泛应用带来了新的活力和机遇，使得原本难以在大范围内获取交通状态信息得以实现。目前美国、欧盟和日本都在研究开发道路交通状态获取的相关技术和产品。

我国道路交通状态获取技术发展虽然起步晚，但也取得了较大成绩。1970 年代中期在上海和北京研制成功环形线圈、超声波、电磁式等多种车辆检测器。进入九十年代中期，我国开始进行智能交通系统的研究和开发。2000 年完成中国 ITS 体系框架和标准规范对我国建设 ITS 奠定了基础[6]。十五期间，我国 ITS 开发和试验由讨论可行性和必要性进入实际开发和试验阶段。在技术研究方面，ITS 关键技术开发和示范工程作为十五国家科技攻关重大专项已经在基础交通信息采集与融合、ITS 数据管理技术、短程通信等关键技术展开了研究，并在北京、上海、天津、重庆等十个城市开展示范工程，形成了一些研究成果和原型系统。此外，国内还研制成功了多种车辆传感器，获得了多项国家专利，投入了工程实践并得到了技术验证，为构建传感器网络奠定了一定基础[1][6]。

从国内外交通状态获取技术发展趋势中可以看出，研制用于交通状态获取的自适应、多功能、可组网的新型传感器；研究如何利用现有传感器和新型传感器构成交通状态获取的传感器网络将成为制约交通状态获取技术发展的前沿问题，而这些技术的发展程度也将极大的影响到智能交通系统下一步的发展和自动公路系统的构成。

1.2.2 传感器网络技术

传感器网络是计算机科学技术的一个新的研究领域，它是由大量部署在作用区域内的、具有有线或无线通信与计算能力的微小传感器节点通过某种方式构成的能根据环境自主完成指定任务的分布式智能化网络系统[7][8]。

最早出现的传感器是模拟传感器，结构简单，功能单一，输出信号为模拟信号，之后随着集成电路的发展而出现的数字传感器将模拟信号数字化处理和传输，提高了性能。但是在现代工业等领域，被控对象分布范围越来越大，控制要求越来越精，而采用传统的分布检测、电缆传输以及主控制器集中接入和控制的方式已经不能够满足大量现场信号的采集、传递和处理，因此20世纪后期出现的适应工业测控应用需求的现场总线(Fieldbus)技术进入了传感器网络领域。作为一种连接智能化现场设备和主控系统之间的全数字、开放式的双向通信网络，在智能传感器的构建中引入现场总线大大减少了传感器与主控系统的连线，有效降低了系统成本和复杂度，并逐渐形成了基于现场总线的传感器网络。

引入现场总线技术后，测控现场中的每一个传感器都将成为网络中的一个节点，于是形成了分布于较大区域的传感器网络，实现了使用不同的传感器协作的监控不同位置或不同对象的运行状态和工作过程。此外，结合控制网络与数据网络来构建传感器网络也成为一个新型的热点，通过将现场总线接入因特网，在一定条件下便可通过因特网监视并控制这些生产系统和现场设备的运行状况和各种参数，通过网络实现被控对象的远程监控。

综合了传感器技术、嵌入式计算技术、现代网络及无线通信技术、分布式信息处理技术等的传感器网络，能够通过各类集成化的微型传感器协作地实时监测、感知和采集各种环境或监测对象的信息，通过嵌入式系统对信息进行处理，并通过随机自组织无线通信网络以多跳中继方式将所感知信息传送到用户终端，从而真正实现“无处不在的计算”理念[7][8]。

传感器网络节点的组成和功能包括四个基本单元，分别为：传感单元（由传感器和模数转换功能模块组成）、处理单元（由嵌入式系统构成，包括CPU、存储器、嵌入式操作系统等）、通信单元（由无线通信模块组成）、以及电源部分。此外，可以选择的其它功能单元包括：定位系统、运动系统以及发电装置等。

在传感器网络中，节点通过各种方式大量部署在被感知对象内部或者附近。这些节点通过自组织方式构成无线网络，以协作的方式感知、采集和处理网络覆盖区域中特定的信息，可以实现对任意地点信息在任意时间的采集，处理和分析，自组织、微型化和对外部世界的感知能力是传感器网络的三大特点[8][9]。

传感器网络是由密集型、低成本、随机分布的节点组成的，自组织性和容错能力使其不会因为某些节点在恶意攻击中的损坏而导致整个系统的崩溃，这一点是传统的传感器技术所无法比拟的，也正是这一特性使得其在交通、军事、环境、健康、家庭和商业领域都有着广泛的应用，而在空间探索和灾难拯救等特殊的领域，传感器网络也有其得天独厚的技术优势。

在道路交通控制系统中，传感器网络主要用于对道路交通状态的获取。随着

新技术的发展，道路交通系统越来越多地使用各类差异巨大的传感器。传感器网络所具有的自适应、多用途、可组网的功能完全适用于对各种道路交通传感器进行组网，并优化物理传感器和虚拟传感器，而在面对不同应用时，传感器网络还能够对各个传感器进行融合优化。传感器网络技术是智能交通系统下一步发展的关键制约技术，是构成自动公路系统的基础。

在军事领域，传感器网络非常适合应用于恶劣的战场环境中，包括监控我军兵力、装备和物资，监视冲突区，侦察敌方地形和布防，定位攻击目标，评估损失，侦察和探测核、生物和化学攻击等。与独立的卫星和地面雷达系统相比，传感器网络在提高型信噪比、降低成本、消除环境噪声、提高探测性、多点联合能等方面都具有潜在的优势。

在环境科学领域，传感器网络为野外随机性的研究数据获取提供了方便，如跟踪候鸟和昆虫的迁移，研究环境变化对农作物的影响，监测海洋、大气和土壤的成分等。同样，传感器网络对森林火灾准确、及时地预报也有帮助。此外，传感器网络也可以应用在精细农业中，以监测农作物中的害虫、土壤的酸碱度和施肥状况等。

在医疗健康领域，在住院病人身上安装特殊用途的传感器节点，如心率和血压监测设备，利用传感器网络，医生就可以随时了解被监护病人的病情，进行及时处理。传感器网络还能被用来长时间地收集人的生理数据，这些数据能够在研制新药品的过程中发挥作用，而安装在被监测对象身上的微型传感器也不会给人的正常生活带来太多的不便。此外，在药物管理等诸多方面，它也有新颖而独特的应用。总之，传感器网络为未来的远程医疗提供了更加方便、快捷的技术实现手段。

此外，在商业领域、足球比赛、在灾难拯救、仓库管理、交互式博物馆、交互式玩具、工厂自动化生产线等众多领域，传感器网络都体现出了全新的设计和应用[7][8][9]。

1.3 以太网技术的国内外研究现状

1.3.1 以太网

以太网(Ethernet)是当今局域网所采用的最通用的通信协议标准，它既是一种计算机接入局域网络的连接标准，又是一种网络互联设备数据共享的通信协议[10][11]。

以太网是美国施乐（Xerox）公司的 Palo Alto 研究中心（简称为 PARC）于 1975

年研制成功的，后经数字设备公司（Dgital Equipment Corp.）、Intel 公司联合扩展，于 1982 年公布的以太网规范，IEEE802.3 就是以这个技术规范为基础制定的。以太网络使用 CSMA/CD(Carrier Sense Multiple Access with Collision Detection，载波监听多路访问及冲突检测技术)技术，它不是一种具体的网络，而是一种技术规范，它定义了在局域网中采用的电缆类型和信号处理方法。以太网在互联设备之间以 10~100Mbps 的速率传送信息包，双绞线电缆 10Base-T 以太网由于其低成本、高可靠性以及 10Mbps 的速率而成为应用最为广泛的以太网技术[10][11]。

按照国际标准化组织开放系统互连参考模型(ISO/OSI)的 7 层结构，以太网标准只定义了数据链路层和物理层。作为一个完整的通信系统，它需要高层协议的支持[12][13]。APARNET（Advanced Research Projects Agency，美国国防部高级研究计划管理局）在制定了 TCP/IP 高层通信协议，并把以太网作为其数据链路层物理层的协议之后，以太网便和 TCP/IP 紧密地捆绑在一起了。现在人们俗称的以太网技术以及工业以太网技术，不仅包含了物理层与数据链路层的以太网规范，而且包含了 TCP/IP 协议组，即包含网络层的网际互联协议 IP、传输层的传输控制协议 TCP、用户数据报协议 UDP 等。有时候甚至把应用层的简单邮件传送协议 SMTP、域名服务 DNS、文件传输协议 FTP、再加上超文本链接 HTTP、动态网页发布等互联网上的应用协议都与以太网这个名词捆绑在一起，因此以太网技术实际上是上述一系列技术的统称[10][12]。

国际因特网作为成功运用以太网技术的实例，即以太网+TCP/IP 模式，具有成本低、传输速率高等诸多优点，进入商业市场 20 多年来，在办公自动化等方面获得了广泛的应用，而且已经成为最受欢迎的通信网络之一[10][12]。

1.3.2 TCP/IP 协议

TCP/IP 协议(Transmission Control Protocol/Internet Protocol，传输控制协议或网际互联协议)是一组包括 IP 协议、UDP（User Datagram Protocol）协议、TCP(Transmission Control Protocol)、ICMP（Internet Control Message Protocol）协议和其他一些协议在内的协议组。TCP/IP 协议组之所以流行，部分原因是因为它可以用在各种各样的信道和底层协议（例如 T1 和 X.25、以太网以及 RS-232 串行接口）之上，如图 1-1 为 TCP/IP 协议的分层体系。

在 TCP/IP 协议组中，属于网络层的协议有：网际互联协议 IP(Internet Protocol)、地址解析协议 ARP(Address Resolution Protocol)和逆向地址解析协议 RARP(Reverse Address Resolution Protocol)、网际控制报文协议 ICMP(Internet Control Message Protocol)与网际组管理协议 IGMP(Internet Group Management

Protocol)[10][11]。

ARP 的功能是将 32 位 IP 地址转换成 48 位的网络连接设备的物理地址；而 RARP 则相反，它将网络连接设备的的物理地址转换为 IP 地址；网际控制报文协议 ICMP 负责因路由问题引起的差错报告和控制；IGMP 则是多目的传送设备间信息交换协议。

运输层包括传输控制协议 TCP 和用户数据报协议 UDP。

应用层的内容十分丰富，包括域名服务 DNS、文件传输协议 FTP、简单的网络管理协议 SNMP、简单邮件传输协议 SMTP、简单网络定时协议 SNTP、超文本传输协议 HTTP 等，他们构成了 TCP/IP 协议组的高层协议[10][11]。

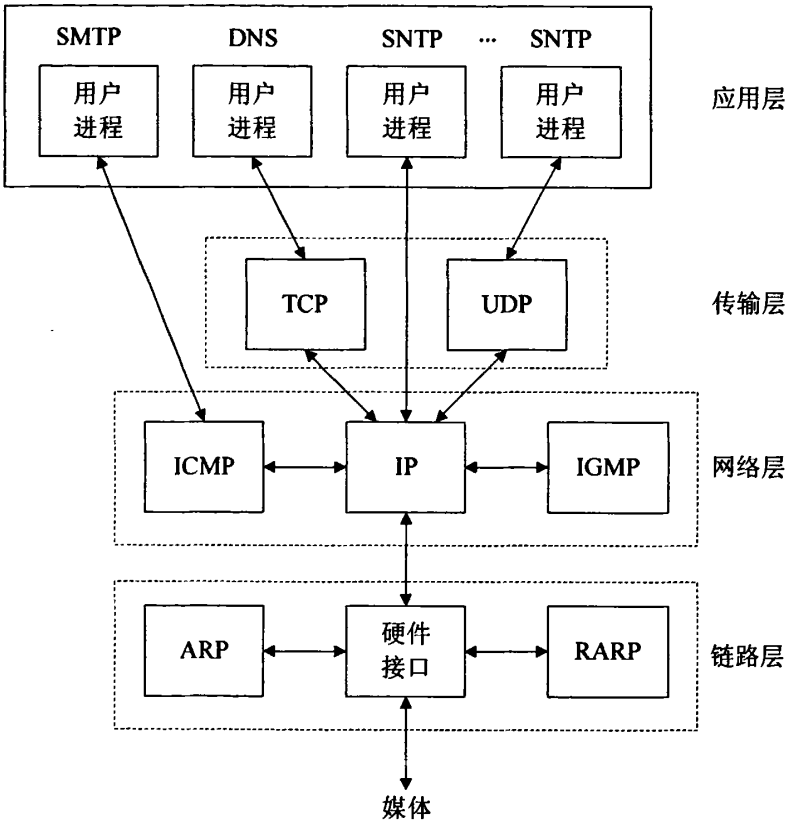


图 1-1 TCP/IP 协议组中不同层次的协议

Figure1-1 Different Layers of TCP/IP Protocol

1.3.3 嵌入式以太网及其发展前景

作为一种通信协议标准，以太网具有速度快、成本低、高实效、高智能及高扩展性等特点。由于计算机技术和网络通信技术的飞速发展，现代工业信息交换

需求已经从管理层内部迅速覆盖到控制、现场设备等层次，因此连接上层网络和现场总线的工业数据通信正在成为工业自动化领域的新兴热点之一[14][15]。

作为当前通信网络领域的核心之一，以太网技术在给办公自动化带来深刻变革的同时，也逐渐渗透到工业自动化领域并迅速增长。几乎所有的现场总线系统最终都可以连接到以太网。很早以前就有将以太网用作现场总线的做法，然而，以太网的资源得不到有效地利用、速度快但传输效率低、不可靠等以太网的传统缺点并不适合现场的应用要求。后来通过对以太网协议的修改，一些新方案的实施大大改善了以太网的缺点。如通过修改分布式报文结构和传输方式，使以太网的高速率传输得到充分应用，拓扑结构更为灵活，方便现场布线；通过设立专用通道保证实时性，完全利用 PC 控制网络资源[10][13][16]。

随着 PC 技术和 IT 的发展，以太网物理层的很多通用部件成本越来越低。随着集成电路的发展，高档的微处理器作为 I/O 处理器和控制器核心的条件逐渐成熟，而在控制器上运行的实时嵌入式操作系统使控制器更易于实现 TCP/IP 协议[16][17]，以太网络更易于接近现场。嵌入式以太网因为速度快，带宽较宽（一般为 10M 或 100M 还有更高速的 1000M），互操作性好，可扩展性强，价格便宜等特点在嵌入式控制领域得到了广泛的应用。

嵌入式以太网已经成为控制网络发展的主要方向，具有很大的发展潜力。在工业现场引入嵌入了 TCP/IP 协议的以太网技术能够使它直接达到现场设备级，它所具有的环境适用性、可靠性、安全性、通信的确定性、网络的可维护性和可恢复性等特点都使它能够在工业自动化领域中异军突起。

过程控制工业和自动化工业，从嵌入式系统到现场总线控制系统，都认识到了以太网和 TCP/IP 的重要性。以太网和 TCP/IP 作为世界上最为广泛应用的网络协议，它将成为过程级和控制级的主要传输技术。带 TCP/IP 协议标准的以太网接口现在已经在智能设备和 I/O 模块中使用，它能够与工厂信息管理系统进行直接地、无缝地连接，而无需任何专用设备。可以说，以太网在工业通信网络中的应用将构建从底层的现场设备到控制层、企业管理决策层的综合自动化网络平台。作为 21 世纪未来工业网络的首选，嵌入式以太网有着广阔的应用和发展前景[15][16][17]。

针对大量交通状态数据的传输需求，在道路交通数据网络的构建中引入嵌入式以太网技术将成为一种必然的趋势。

1.4 课题的主要研究工作、技术路线和论文结构

1.4.1 课题的主要研究工作

结合项目课题，本论文主要做了如下研究工作：

- 对以太网理论、嵌入式系统及 TCP/IP 协议做了较为深入的研究和分析，确定了采用基于 TCP/IP 协议的嵌入式以太网来构建传感器网络上层系统的策略；
- 结合项目其余部分课题，构建了基于数字信号处理器 DSP 的嵌入式以太网节点，进行了硬件选型以及以太网硬件电路的设计；
- 基于以太网硬件基础，在 DSP 工作环境下编程实现了以太网底层驱动，并结合系统实际需求，实现了嵌入式 TCP/IP 协议；
- 结合项目其余部分成果，在实验室搭建道路交通模拟车道，实现了实验室环境下的分层次传感器网络和交通状态信息的获取。

1.4.2 课题研究的技术路线

根据控制网络的研究特点，本节点系统在设计和实现中采用了功能需求——设计实现——实验验证的技术研究路线，先提出节点所需求的功能，然后通过硬软件设计实现，最后通过大量实验来验证本节点系统方案实现的合理性和可行性。如图 1-2 为本嵌入式以太网实现的技术研究路线图。

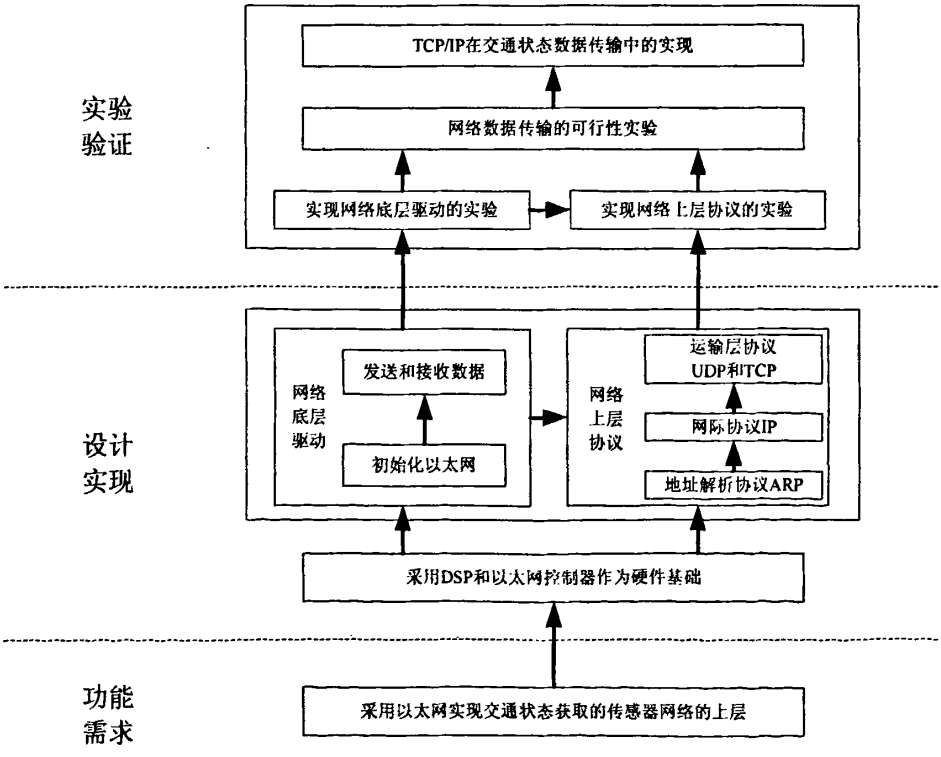


图 1-2 本嵌入式以太网的技术研究路线图

Figure1-2 Technology Roadmap of the embedded Ethernet

1) 功能需求

本嵌入式以太网功能的设计目标是实现一个完整的以太网通信节点，结合整个传感器网络系统，本部分的功能需求主要有二：

其一，结合系统整体设计规划，本部分采用数字信号处理器 DSP 控制以太网控制器的形式来构建以太网硬件系统。

其二，在以太网硬件基础之上，需要嵌入相应的网络协议来实现信息的传输功能，在本节点的设计中，需要实现基于 TCP/IP 协议的以太网通信功能。

2) 设计实现

结合整体系统需求，需要从硬件和软件两部分对系统进行设计并实现功能。首先需要进行硬件的选型，尤其是以太网控制器的选型，然后根据相应芯片的操作控制规则实现硬件的连接。在硬件实现的基础之上，系统需要完成实现通信的基本功能的设计，包括网络系统的初始化配置、底层驱动以及 TCP/IP 协议的实现，这是本系统功能设计的核心部分，也是难点所在，主要通过基于 DSP 操作软件 CCS3.1 的编程来实现。

由于网络系统需要传输大量的交通状态数据，在系统上层的以太网，同时采用了 UDP 和 TCP 作为 TCP/IP 的运输层协议。UDP 为无连接协议，传输控制都较简单，但是不保证数据可靠传输，因此用 UDP 来实现对大量交通状态数据的传输，而面向连接的协议 TCP，实现起来较为复杂，但是可靠性高，因此用来实现少量重要数据的可靠传输。

3) 实验验证

实验验证是本设计最为重要的部分，本课题的任务在于从软硬件上实现嵌入式以太网对交通状态数据的传输功能，因此，需要大量的实验来验证节点系统设计的正确性。

在本课题中，基于 DSP 软件 CCS3.1，编程实现了网络系统的底层驱动及上层协议，结合相关交通理论，针对面向交通状态获取的嵌入式以太网系统的特点，设计了大量相关的网络实验，通过基于网络实验平台和模拟道路交通环境的软件编程实现了网络系统的各预期功能，包括作为网络节点需要实现的帧填充和 UDP 数据报分片等功能，以及 TCP 连接的建立和释放，大量实验最终了证明了本嵌入式以太网系统设计的合理性和可行性。

通过选择合理的技术研究路线将能使课题研究工作更好的开展，并最终实现了预期的设计功能。

1.4.3 论文结构

本论文共分了五个章节对课题研究进行了叙述。

第一章为引言部分，主要介绍了交通状态获取技术、传感器网络技术及嵌入式以太网技术在国内外的研究现状和发展前景，阐述了交通状态信息获取的过程中，引入嵌入式以太网的可行性和必要性。

第二章介绍了嵌入式以太网系统的硬件设计，主要介绍了 DSP 和以太网控制器的硬件选型，及所选器件的特性，并介绍了本嵌入式以太网系统中主要硬件的连接方式。

第三章为软件实现部分，主要介绍了以太网底层驱动及上层嵌入式 TCP/IP 协议，阐述了在本系统中对 TCP/IP 协议的简化，重点介绍了运输层的两个重要协议 TCP 及 UDP，并针对它们的不同应用进行了比较。

第四章为实验部分，主要介绍了对 ARP、UDP 和 TCP 等协议的实验设计，并通过大量实验进行了验证，给出了各个实验的实验现象图，以及数据包分析图，结合项目其余成果，在实验室的模拟道路交通环境下，实现了对交通状态信息的获取。

第五章为结论部分，对本嵌入式以太网课题进行了总结，对后续的研究工作进行了展望，并指出了本论文研究的不足之处。

2 嵌入式以太网硬件设计

在本项目中，本课题部分主要负责传感网络上层的以太网通信部分，因此本文主要介绍网络上层的以太网通信功能的设计和实现，而下层 CAN 总线网络及其余外围功能不在本文讨论范围之列。

在本节点系统的设计中，以太网通信功能的实现主要是通过 DSP 对以太网控制器 CS8900A 的控制操作来实现的，本章首先介绍了传感器网络系统的总体结构，其次重点介绍了以太网节点的硬件设计，包括主要硬件的选型和电路连接。

2.1 网络系统的总体结构

根据构建网络化分层次的信息传输技术的需求，本项目采用现场总线和以太网相结合的方式来构建道路交通数据网络。在采集传感器信号的底层道路现场，数据传输量较小，但是对实时性和灵活性的要求较高，因此采用现场总线来实现数据传输，而在上层系统中，数据传输量较大，通信距离长但实时性要求并不高，采用以太网来实现通信功能[16][18][19]。

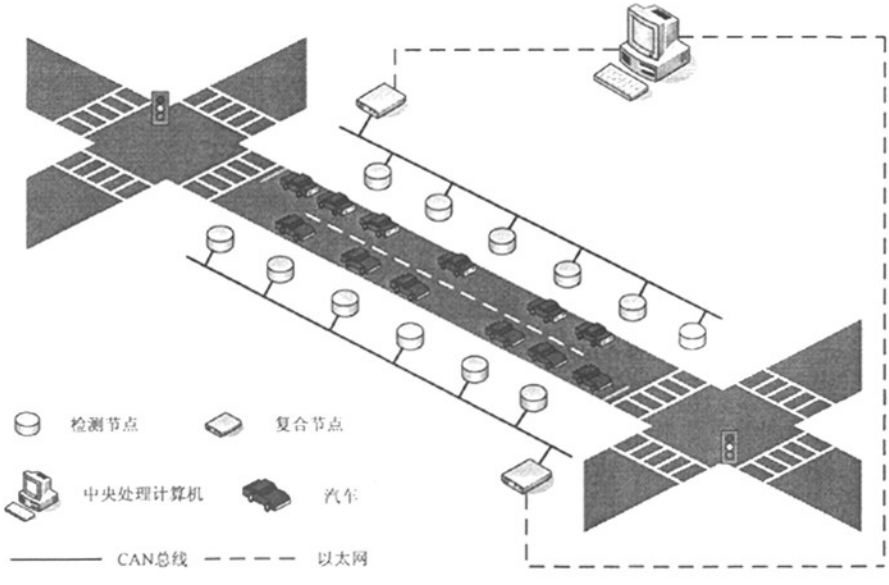


图 2-1 道路传感器网络的系统总体框图

Figure2-1 Configuration of Road Sensor Networks System

如图 2-1 所示为道路交通控制系统获取交通状态的传感器网络部署结构图，在传感器网络系统中，首先要在一定的道路路段铺设各种连接在检测节点上的传感

器，检测节点采集来自各路段传感器的交通信息并做较为简单的前端处理，下层CAN 总线网络通过对各个检测节点进行检测和轮询来获取底层交通信息，并将其送到每个路口的 DSP 复合节点。在复合节点，需要对来自 CAN 总线的大量交通数据进行处理和融合，以减轻网络系统的压力，最后通过以太网将处理后的数据送入中央处理计算机进行上层系统的处理和控制，从而完成道路交通网络的构建，实现对交通状态的获取。

根据系统各个部分功能的不同，道路交通网络节点系统可以分为三个部分：DSP 处理器及相关外设、下层 CAN 总线通信模块和上层以太网通信模块。

从整个项目构建传感器网络的要求出发，选用 TI 公司生产的 TMS320F2812 DSP 芯片作为本传感器网络节点的微处理器，作为一种低功耗的 32 位定点数字信号处理器，它所具有的强大的数字信号处理功能和方便的接口模块完全能够胜任本项目对网络节点功能的要求[20][21]。

由于在本系统应用中可能会面临大量的实时动态交通状态数据处理问题，因此需要对 DSP 进行外部存储区的扩展，以增加装置的数据处理潜力。这里选用了两片 ISSI 公司生产的高速静态 RAM 芯片 IS61LV25616，每片容量为 256K*16 位，共计 512K*16 位。

由于本课题设计的复合节点装置所采用的主控器 TMS320F2812 中已集成了 CAN 总线控制器 eCAN 模块，因此只需增加 CAN 总线驱动器的相关电路就可实现 CAN 总线的通信功能。在这里选用 Philips 公司生产的较为常用的 PCA82C250 作为 CAN 总线收发器。为增加复合节点硬件的抗干扰能力，在 CAN 总线驱动器 PCA82C250 与 DSP 的 eCAN 模块之间加入了快速光藕 6N137 进行光电隔离，该器件能够切断主控制电路与外部总线电气环境之间的联系，有效地抑制尖峰脉冲和各种噪声干扰。

在以太网通信模块，采用 CS8900A 芯片作为以太网控制器，通过对 DSP 的编程控制来实现以太网的通信功能，其中 CS8900A 与 RJ45 的连接需要通过一个带厄流线圈的脉冲隔离变压器 HR601627 进行隔离，它的作用主要是将外部线路与 CS8900A 隔开，防止干扰和烧坏元器件，实现带电的插拔功能。作为本论文的重点，本部分将在后面的内容中进行重点介绍。

此外，在系统的实现中，还设计连接一片 CPLD 芯片，采用 ALTERA 公司 MAX II 系列的 CPLD 芯片，型号为 EPM1270T144C5N，它主要负责板上各设备的片选译码控制，以及矩阵键盘和液晶显示模块的人机接口控制。

如图 2-2 为本道路交通系统网络节点的总体硬件连接框图。

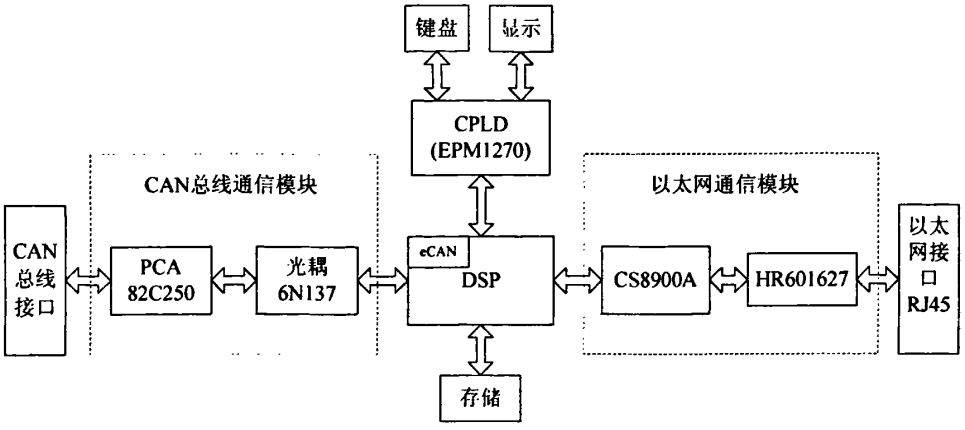


图 2-2 网络节点的总体硬件连接框图

Figure2-2 Hardware Configuration of Network Node

2.2 DSP 及其外围电路

在本传感器网络系统中，采用数字信号处理器 TMS320F2812 作为网络复合节点的核心处理器，它主要用来连接并控制上层太网系统和下层 CAN 总线系统之间的数据通信，并在复合节点对大量交通数据进行融合处理，因此作为整个系统的核心，本部分首先主要介绍数字信号处理器，及其相关的外围电路，包括时钟电路和片外扩展存储器电路。

2.2.1 数字信号处理器 TMS320F2812

TMS320F2812是美国Texas Instruments(TI)公司生产的面向测控应用的32位定点型DSP（数字信号处理器，Digital Signal Processor）芯片，是目前最先进、功能最强大的数字信号处理器之一[20][21]。该系列处理器基于TMS320C2xx内核的定点数字信号处理器，其上集成了多种先进的外设，同时代码和指令与F24x系列数字信号处理器完全兼容。

该型号DSP采用改进的哈佛结构，程序存储器和数据存储器具有各自独立的总线结构，指令执行速度为150MIPS，并支持C/C++编程[22][23]。另外，该DSP具有丰富的片内外设资源，包括AD转换模块、增强型CAN总线控制器、事件管理器以及串行通信接口等，使其非常适用于有大批量数据处理的测控场合。

TMS320F2812主要性能特点如下：

- 高性能静态CMOS技术，功耗低(内核电压1.8V，I/O电压3.3V)，编程电压

3.3V；

- 高性能的32位CPU，采用改进的哈佛总线结构，统一的编址方式和强大的寻址能力；
- 丰富的片内存储器资源，包括8KX16位的Flash存储器,1K*16位的OTP型只读存储器，两块4K*16位的SARAM，一块8K*16位的SARAM和两块1K*16的SARAM；
- 内部锁相环支持动态改变时钟倍频系数；
- 丰富的外部中断源，包括12组每组8个，共96个外部中断；
- 16通道的12位ADC；
- 丰富的串行通信接口设备，包括SPI、SCI及eCAN；
- 56个可编程、多用途的GPIO引脚；
- 3个32位的CPU定时器；
- 两个事件管理器模块(EVA,EVB)，提供高效的电机控制方案。

2.2.2 时钟电路

在本系统设计中，TMS320F2812 的时钟信号采用一个外部有源晶体振荡器来产生，选用有源外部晶振可以有效降低噪声和提高系统的抗干扰能力，并保证足够的驱动能力。TMS320F2812 芯片具有锁相环的功能，主要用来从一个较低频率的外部时钟获得较高的内部时钟频率，倍频数可为 0.5-5 倍，由编程实现，在这里使用 30MHz 的外部时钟，系统最高工作频率可达 150MHz，如图 2-3 所示。

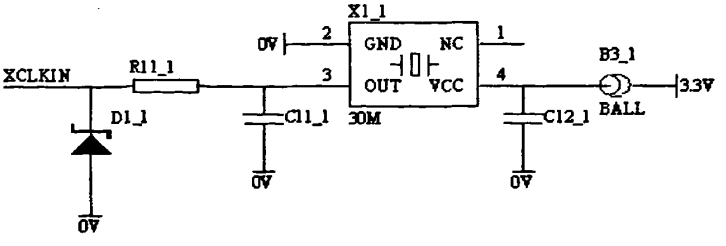


图 2-3 时钟电路

Figure2-3 the Clock Circuit

由于 TMS320F2812 芯片的内核工作电压为 1.8V，因此在时钟电路中连接一个 1.8V 的稳压二极管，以提供不超过 1.8V 的适当时钟信号驱动电平。

2.2.3 片外扩展存储器

由于本系统是应用于大量交通状态的获取和传输的，需要面对大量实时动态交通状态数据处理问题，因此需要对处理器进行外部存储区的扩展，以增加装置的数据处理潜力。这里选用了两片 ISSI 公司生产的高速静态 RAM 芯片 IS61LV25616，每片容量为 256K*16 位，共计 512K*16 位。该芯片为 3.3V 电源供电，高性能、低功耗，能够直接与 DSP 连接并满足 DSP 高速运行的要求。如图 2-4 所示。

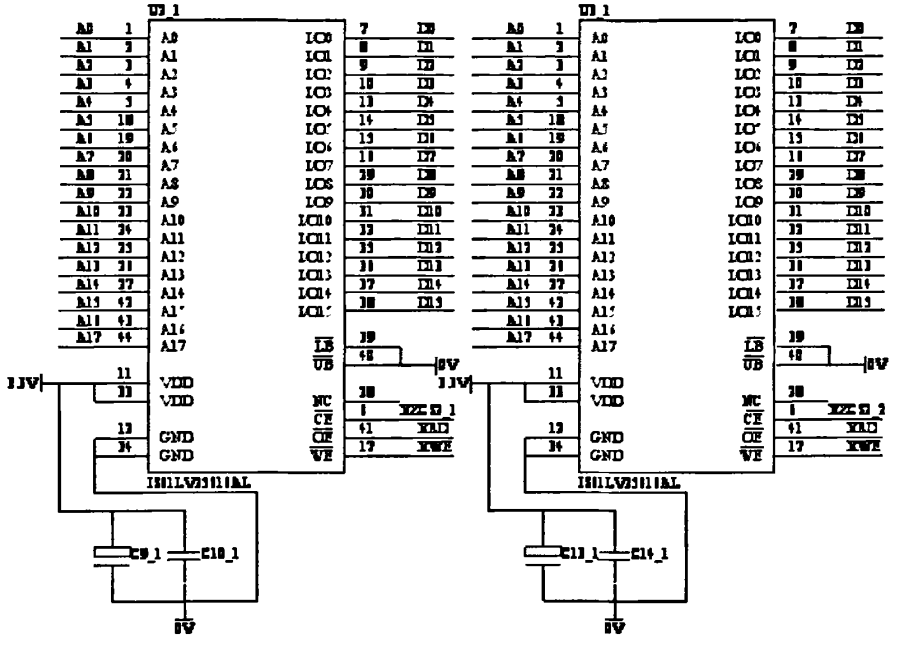


图 2-4 片外扩展 RAM 电路

Figure2-4 the Circuit of External Extended RAM

2.3 以太网节点的硬件设计

根据以太网系统的实际需求，从节约能耗和设计可靠性的角度考虑，选用 CIRRUS LOGIC 公司生产的以太网控制器 CS8900A 来实现本节点的以太网通信功能。

CS8900A 与仅支持 5V 供电的以太网接口芯片相比，最大的优点在于它有 3.3V 供电型号，因此在本以太网节点，它能够直接与 DSP 相连，而无需电平转换电路，这极大地简化了节点的硬件设计。

2.3.1 以太网控制器 CS8900A

CS8900A 是 CIRRUS LOGIC 公司生产的一种低功耗、性能优越的 16 位的 10M 以太网控制器。它功能强大，为 100 脚 TQFP(薄四方扁平封装)封装。该器件的突出特点是应用灵活，其物理接口、数据传输模式和工作模式都能够根据需要进行动态调整、并能通过对内部寄存器的设置来适应不同的应用环境[24][25]。

CS8900A内部结构框图如图2-5所示，主要有ISA(Industry Standard Architecture，工业标准接口)总线接口、EEPROM接口、RAM、存储管理单元、802.3MAC单元以及完整的物理层接口。CS8900A的总线接口逻辑包括16位数据总线、0-19根地址总线和控制总线。

CS8900A 内部功能模块主要是 802.3 介质访问控制块(MAC)。它支持全双工操作，完全依照 IEEE 802.3 以太网标准，它负责处理有关以太网数据帧的发送和接收，包括：冲突检测、帧头的产生和检测、CRC 校验码的生成和验证。通过对发送控制寄存器(TxCMD)的初始化配置，MAC 能自动完成帧的冲突后重传。如果帧的数据部分少于 46 个字节，它能生成填充字段使数据帧达到 802.3 所要求的最短长度[24]。

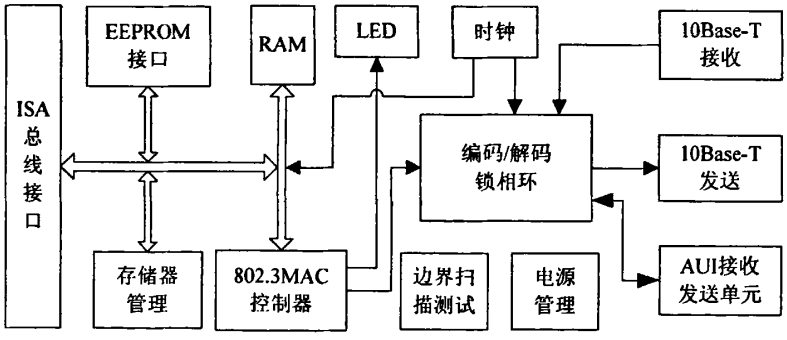


图 2-5 CS8900A 内部结构框图

Figure2-5 Interior Configuration of CS8900A

1) CS8900A 的特点

作为一种性能优越的以太网控制器，CS8900A 有很多特点：符合 IEEE802.3 以太网标准，并带有 ISA 接口；片内 4K 字节 RAM；适用于 I/O 操作模式、存储器操作模式和 DMA 操作模式；带有传送和接收低通滤波的 10BASE-T 的连接端口；支持 10BASE-2、10BASE-5 和 10BASE-T 的 AUI(Attachment Unit Interface)接口；自动生成报头，自动进行 CRC 检验，冲突后自动重发；最大电流消耗为 55mA(5V 电源)；全双工操作；支持外部 EEPROM。

除此之外，CS8900A还有不同于其他以太网控制器的独有特点，如：工业级温度低（-40~80C)；3.3V工作电压，功耗低；高度集成设计；独特的PacketPage结构等。

由于 CS8900A 功能强大，集成度高，因此在不到 10cm 的电路板上能够集成片上 RAM、10BASE-T 传输和接收滤波、AUI 单元、ISA 总线接口等，从而能够极大的减少电路板面积，节约成本。

CS8900A 的内存结构组织采用独特的 PacketPage 结构[24]。采用这种结构，不论 CS8900A 采用 I/O 模式还是存储模式，都可以以一种统一的方式对其进行控制，并能够自动调整网络传输方式和所需的系统资源，这样既减少了 CPU 的开销，又简化了软件编程，提高了系统灵活性和效率，适应嵌入式系统开发原则。

2) CS8900A 的工作原理

CS8900A 对数据的传输非常灵活，有三种数据传输模式，分别为 I/O 模式、MEMORY 模式和 DMA 模式。其中，I/O 模式是访问 CS8900A 存储区的默认模式，简单易用[24]。

在 I/O 模式下，I/O 地址空间映射为 8 个端口(偶数偏移地址)，由于对内部 RAM 的地址采用统一编码，通过对 PacketPage 指针的低 12 位赋值，可以对整个 4K RAM 进行访问。状态和控制寄存器的低 6 位分别为其各自内部寄存器地址，因此通过对中断状态队列端口访问，便可确知哪个中断寄存器响应。

在 I/O 操作模式下，处理器通过 8 个 16 位的 I/O 端口访问 CS8900A，这个端口与主机系统的 I/O 空间的 16 个相邻的 I/O 存储单元相连，表 2-1 为 CS8900A 在 I/O 下的端口映射。I/O 模式是 CS8900A 默认的配置，而且始终都是使能的，上电时，I/O 基地址的默认值为 0x0300H（0x0300H 是局域网典型的外设地址），它可改变成任何可用的 XXX0H 的存储单元，也可以从 EEPROM 载入数据，或者通过系统复位进行[24]。

表2-1 CS8900A在I/O下的端口映射

Table2-1 I/O Mode Mapping of CS8900A

端口	读写类型	功能
0000H	读/写	接收/发送数据端口 0
0002H	读/写	接收/发送数据端口 1
0004H	只写	TxCMD（传送命令寄存器）
0006H	只写	TxLength(传输数据长度寄存器)
0008H	只写	ISQ（中断状态队列）
000AH	读/写	PacketPage 地址指针
000CH	读/写	PacketPage 数据（端口 0）
000EH	读/写	PacketPage 数据（端口 1）

CS8900A在I/O模式下的端口映射如表2-1所示，其各个端口的功能如下：

(1) 接受/发送数据（端口0和1）

这两个端口用于CS8900A传送数据和接受数据。端口0可用于16比特操作，端口0和端口1可用于32比特操作。

(2) TxCMD端口

主机在每次传送操作时向这个端口写入TxCMD指令，传送指令告诉CS8900A有一帧将被传送以及如何传送。

(3) TxLength端口

帧的长度在传送指令写入后立即被写入。

(4) 中断状态序列端口

该端口包含中断状态序列的实时值。

(5) PacketPage地址指针端口

无论何时，该端口被写入主机想要访问的任何CS8900A的内部寄存器，前1(0到B)比特在实时操作时提供目的寄存器内部地址，后三个比特为只读且始终为011b。任何合适的值写到指示端口时将写入这些比特，最后的比特指示是否PacketPage地址指针会自动增加到下一字符的存储单元。

(6) PacketPage数据端口（端口0和1）

PacketPage数据端口用于传送数据且数据来自于CS8900A的内部寄存器，端口0可用于16比特操作，端口0和端口1可用于32比特操作。

CS8900A 的基本工作原理比较简单。在收到由主机发来的数据报（从目的地址域到数据域）后，CS8900A 侦听网络线路。如果线路忙，它就等到线路空闲为止，否则立即发送该数据帧。在发送过程中，它首先添加以太网帧头（包括先导字段和帧开始标志），然后生成 CRC 校验码，最后将此数据帧发送到以太网上。接收时，它将从以太网收到的数据帧在经过解码、去掉帧头和地址检验等步骤后缓存在片内。在 CRC 校验通过后，它会根据初始化配置情况，通知主机 CS8900A 收到了数据帧，最后，用某种传输模式将数据帧传到主机的存储区中。在这里，先导字段、帧开始标志和 CRC 校验序列仅供以太网控制器本身使用，并不传输到 DSP，也可以在初始化时选择是否传输 CRC 校验到 DSP[24][26]。

2.3.2 以太网接口的硬件连接

在 CS8900A 的三种数据传输模式中，I/O 模式是访问 CS8900A 存储区的默认模式，简单易用[24]。考虑到系统内存资源的限制，为了简化软硬件设计，采用 I/O 模式来进行操作控制和数据传输，将 CS8900A 的寻址空间安排在 DSP 的 I/O 地址空间，同时不使用 EEPROM 对 CS8900A 进行初始化，以减少嵌入式系统板上器件数量，CS8900A 仅通过 10BASE-T 与网络连接[27][28]。如图 2-6 为本节点系统设

计中 DSP 与 CS8900A 的硬件连接方式。

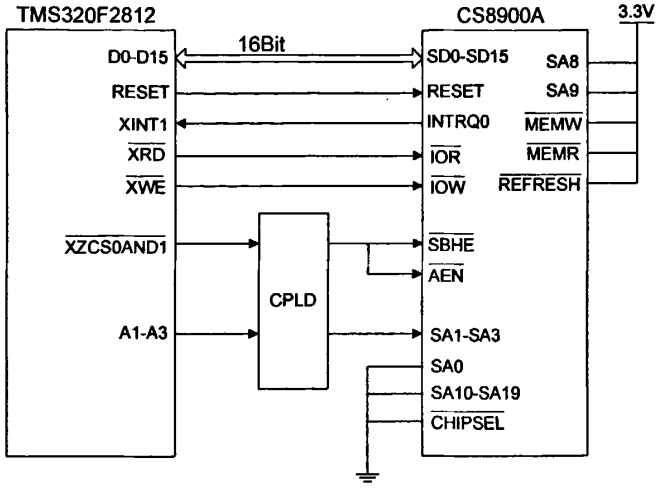


图 2-6 DSP 与 CS8900A 的硬件连接方式

Figure2-6 Connection of DSP and CS8900A

在本节点的硬件连接中，CS8900A 的 I/O 基地址取默认值 0x0300H，读写管脚/MEMW 和/MEMR 被置高，以关闭 Memory 方式，在此模式下，用 3 根地址线就可以实现对所有寄存器的访问。将 SA0 以及 SA10…SA19 一同接地，SA8、SA9 接 3.3V 正电压，这样仅需将 3 根地址线 SA1、SA2 和 SA3 对应接到 CPU 的地址线 A1、A2 和 A3 上即可控制。AEN 低电平有效，可作为片选信号，连接到 DSP 的内存空间 0 区和 1 区的片选线 XZCS0AND1 上。(/SBHE)为 CS8900A 数据总线高 8 位使能信号线，在每次复位后，进行 I/O 或存储模式访问前，需要用一个由高到低、再由低到高变化的电平信号重新进行触发，当 CS8900A 作为 16 位通信时，此信号为低有效，因此可以将 AEN 与 SBHE 相连接用同一个片选信号控制。为了有效分配 I/O 地址空间，选用一片 CPLD 芯片 EPM1270 对 CS8900A 控制信号进行地址译码，将其地址映射为从 0x2000H 起始的 DSP 空间内。DSP 通过/IOW 和/IOR 等控制信号线实现对 CS8900A 工作方式的控制和读写操作。

CS8900A 与以太网接口，采用 10BASE-T 连接，其中 CS8900A 与 RJ45 的连接需要通过一个带厄流线圈的脉冲隔离变压器 HR601627（发送变比为 1：2.5，接收变比为 1：1）进行隔离，它的作用主要是将外部线路与 CS8900A 隔开，防止干扰和烧坏元器件，实现带电的插拔功能。如图 2-7 为 DSP 与以太网节点的总体硬件连接框图。

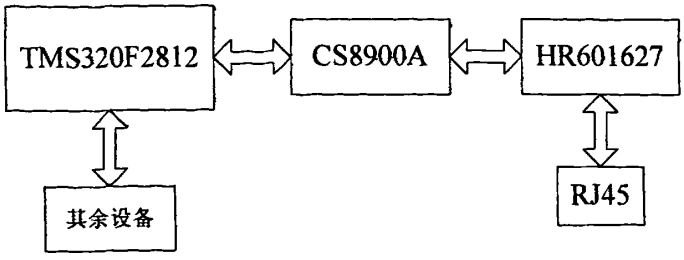


图2-7 DSP与以太网节点的总体硬件连接框图

Figure2-7 Connection of DSP and Ethernet Interface

3 嵌入式以太网软件设计

要完成以太网的通信功能，需要在硬件连接的基础上实现软件的设计，在本网络节点的设计中，以太网的软件实现主要包括以太网控制器 CS8900A 的初始化配置、以太网驱动程序的实现和嵌入式以太网协议的实现。

3.1 CS8900A 的初始化

在实现以太网通信功能之前，需要根据节点系统自身功能的要求先对以太网控制器 CS8900A 进行初始化，在实现中，这可以通过对相关寄存器的设置来达到目的[24][25]。

CS8900A 初始化的主要内容如下：

- 软件复位，并检查复位完成标志是否置位；
- 设定 8/16 位工作模式，一般设为 16 位模式；
- 设定临时使用的以太网物理地址，真实地址需要向权威机构申请；
- 设定接收帧的类型，一般要能接收广播；
- 确定数据的传送方向，可设为全双工或半双工；
- 中断允许；
- 使能接收中断；
- 确定 CS8900A 的中断管脚号，根据硬件线路使用情况来确定；
- 接收发送使能。

在本网络节点中，DSP 与 CS8900A 是按照 16 位方式连接，CS8900A 复位后默认的工作方式为 I/O 连接，基地址为 0x0300H[24][25]。

3.2 CS8900A 主要工作寄存器的配置

在 DSP 控制 CS8900A 实现以太网通信的过程中，任何控制都是通过对芯片内部寄存器的操作来实现的，下面对它的几个主要工作寄存器进行介绍（寄存器后括号内的数字为寄存器地址相对于基地址 0x0300H 的偏移量）。

(1) LINECTL(0112H)

决定 CS8900A 的基本配置和物理接口。初始值为 00D3H，选择物理接口为 10Base-T，并使能设备的发送和接收控制位。

(2) RXCTL(0104H)

控制 CS8900A 接收特定的数据报．初始值为 0D05H，接收网络上的广播或者目标地址同本地物理地址相同的正确数据报。

(3) RXCFG(0102H)

控制 CS8900A 接收特定的数据报后会引发中断，可设置为 0103H，当收到一个正确的数据包后，CS8900A 会产生中断。

(4) BUSCT(0116H)

控制芯片的 I/O 接口的一些操作。设置初始值为 8017H，打开 CS8900A 的中断总控制位。

(5) ISQ(0120H)

ISQ 是 CS8900A 的中断状态寄存器，内部映射接收中断状态寄存器和发送中断状态寄存器的内容。

(6) PORT0(0000H)

发送和接收数据时，DSP 通过 PORT0 传递数据。

(7) TXCMD(0004H)

发送控制寄存器，如果写入数据 00C0H，那么以太网控制器在全部数据写入后开始发送数据。

(8) TXLENG(0006H)

发送数据长度寄存器，发送数据时，首先写入发送数据长度，然后将数据通过 PORT0 写入芯片。

以上主要工作寄存器的设置并不是一定的，在不同的系统传输控制要求下，可以通过对寄存器设置不同的值来实现各自的功能。

3.3 以太网底层驱动的实现

以太网驱动程序的编写和实现是系统软件设计的关键，它直接关系着对硬件的底层操作和对上层协议的连接并实现相应的通信功能。

以太网驱动程序主要包括 CS8900A 的初始化程序、数据的发送和接收程序、以及相应的中断处理子程序等。由于之前已经对初始化配置作了叙述，因此此处只对 DSP 控制 CS8900A 进行数据的接收和发送过程进行叙述。

在对数据包的发送和接收的操作中，有中断和查询两种模式。在查询方式中，主程序需要不断地将要查询的内部寄存器的地址（PacketPage 地址）写入 0x000AH 口（以太网 I/O 地址），然后再从 0x000CH 口读出或写入相关的寄存器内容，来判断是否接收或者发送报文。对于能够主动控制的数据包的发送，可以依照查询方式来操作，而对于处于被动地位的数据包的接收而言，这种操作不符合嵌入式系

统实时性的要求，不可取[29]。

因此，在本节点的设计中，采用查询方式进行数据包的发送，而采用中断方式进行数据包的接收。在进行发送或者接收数据时，首先要根据系统的需求情况对以太网控制芯片进行初始化。

3.3.1 CS8900A 的发送驱动

在发送数据时，首先向 CS8900A 的 TxCMD 寄存器写入发送命令，然后把要发送帧的长度写入到 TxLength 寄存器中，查询 BusST 寄存器的 Rdy4Tx 位是否置 1，是则把要发送的数据帧写入到 CS8900A 的 0x0000H 口，否则继续查询，直到该位置位为止，完成一次数据包的发送。如图 3-1 为查询方式发送流程示意图。

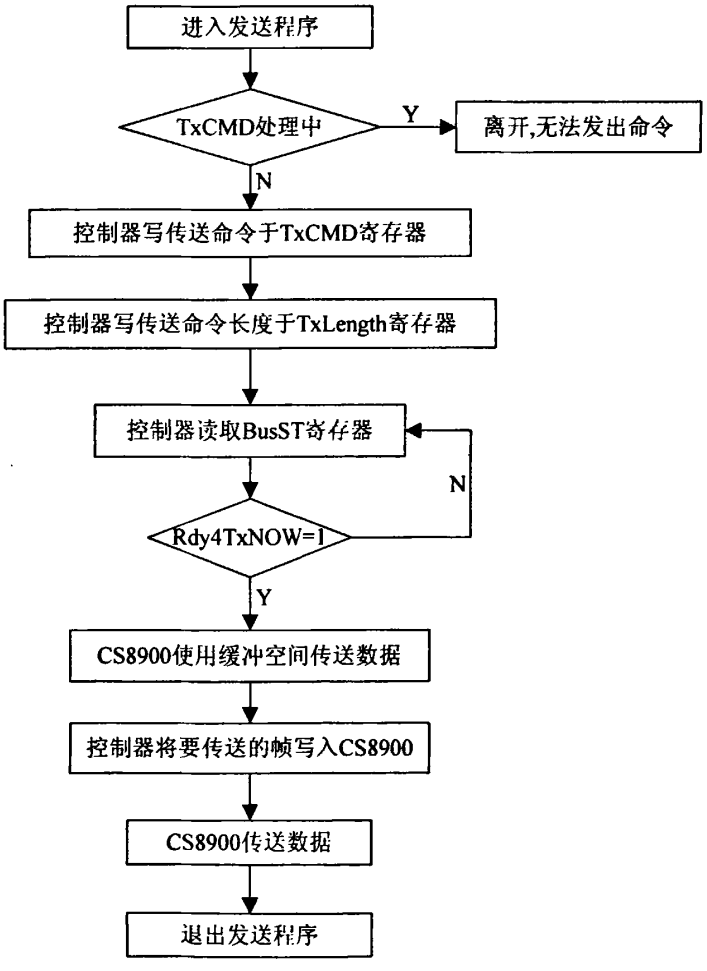


图 3-1 查询方式发送数据流程图

Figure3-1 Flow Chart of Sending Dada in Poll Mode

3.3.2 CS8900A 的接收驱动

在中断方式，CS8900A 先由中断端口向主控制器产生中断信号，主控制器进入以太网通信中断程序，循环读中断状态队列寄存器 ISQ，判断是哪个中断寄存器触发，并进入相应的中断服务子程序中，直到 ISQ 值为 0x0000H。在 CS8900A 中有 5 个寄存器能够产生中断，按优先级顺序分别为：RxEvent(接收事件寄存器)、TxEvent(发送事件寄存器)、BufEvent(缓存事件寄存器)、RxMISS(接收丢失计数器)、TxCOL(发送冲突计数器)。每次主控制器读 ISQ，相应寄存器中产生中断的位将清零，下一个中断报告将会自动移到中断队列最前面。

在中断方式的接收中，一旦有数据包进入 CS8900A 前端，则立即产生中断信号，该中断信号通知主机对 ISQ 进行处理，然后进入接收中断子程序对数据包进行接收。在接收中断处理子程序中，只需要连续读相应的 0x000AH 口(系统 I/O 地址)即可。如图 3-2 为中断方式接收数据流程图。

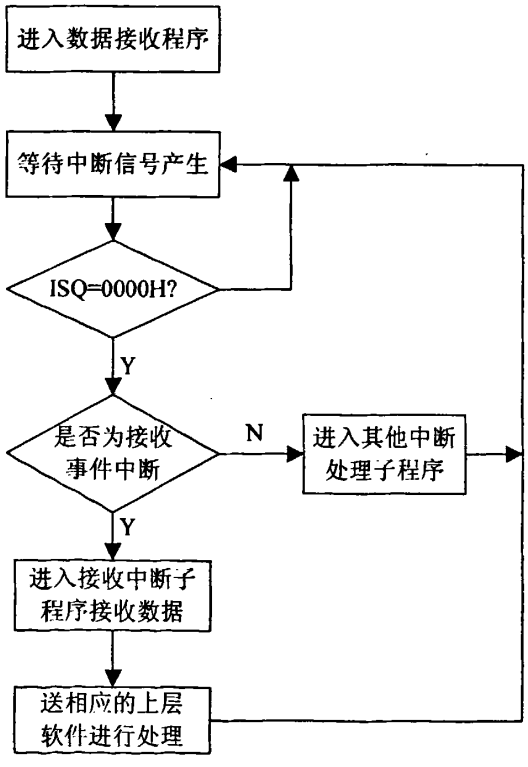


图 3-2 中断方式接收数据流程图

Figure3-2 Flow Chart of Receiving Dada in Interrupt Mode

由于有 5 种中断方式，因此可以通过对相应寄存器的操作来设置是否开放接收中断以外的其余中断，如果设置为仅开放接收中断，那么一旦出现中断，即可认为是接收中断，可以直接进入中断子程序，而无需判断中断方式，因此能在调

试中简化过程，突出重点。

3.4 嵌入式以太网协议的实现

在硬件驱动程序之上是以太网 TCP/IP 协议的实现，它主要负责对要传输的数据按照相应的协议规则进行封装，然后通过底层驱动进行传输。

在本以太网节点网络协议的实现中，同时采用了 UDP 和 TCP 作为 TCP/IP 的运输层协议。UDP 为无连接协议，传输控制都较简单，无法保证数据的可靠传输，因此用 UDP 来实现对大量交通状数据的传输。而 TCP 为面向连接的协议，实现起来较为复杂，但是可靠性高，因此用 TCP 来实现少量重要数据的可靠传输[10][11]。

3.4.1 TCP/IP 协议分层的简化

和工业控制现场类似，在道路交通控制系统中，交通状态获取的传感器网络必然面临着嵌入式系统资源受到限制的问题。完全的TCP/IP协议需要占用大量的系统资源，而道路交通控制系统必须面对海量数据的传输和处理，系统资源并不能够满足完全的TCP/IP协议的应用需求。

因此，针对这个问题，为了提高系统的灵活性并降低成本，以太网接口部分采用了嵌入式解决方案，在不违背协议标准的前提下对传统的 TCP/IP 加以改进，并进行了合理的简化。文献[30]分析了 TCP/IP（Transmission Control Protocol/Internet Protocol）在嵌入式以太网中的速度，代码空间的大小。文献[31][32]则在嵌入式以太网中使用 UDP（User Datagram Protocol）来实现数据的传输。

对TCP/IP协议的适当简化不但能够减少协议对存储空间的占用并提高程序代码运行效率，同时也提高了系统的实时性，并保证了海量交通数据的可靠性传输，从而满足了本传感器网络的嵌入式系统的交通状态获取的应用要求。

这种用于嵌入式系统联网的，根据实际需求简化了的TCP/IP就称为嵌入式TCP/IP，其主要特点是实时性、简单性和灵活性[33][34]。

嵌入式系统的TCP/IP协议通常采用一种简化的四层模型，分别为：应用层、运输层、网络层、链路层，它的实现按不同层次开发，每一层分别负责不同的通信功能，其协议的层次结构如图3-3所示[10][11][35]。

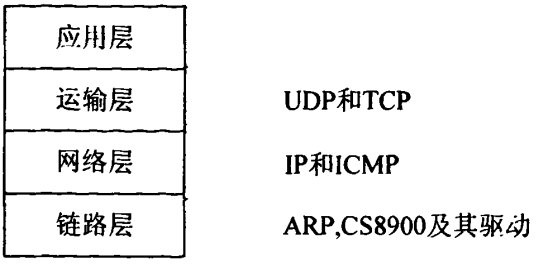


图 3-3 嵌入式 TCP/IP 的层次结构

Figure3-3 Layers Configuration of Embedded TCP/IP

链路层 主要处理对CS8900A芯片的驱动，使其可以和Internet进行数据交换，并实现ARP协议对地址进行解析。

网络层 主要处理数据包在网络中的活动，该层中主要实现IP协议和ICMP协议。

运输层 主要为通信双方的应用程序提供端到端的通信，在本设计中使用UDP协议来实现数据传输。

应用层 主要对要发送的数据或接收到的数据进行上层处理，这部分内容不在本论文讨论范围之列。

在本节点的设计的实现方案中，采用的嵌入式TCP/IP协议主要有：网际协议IP(Internet Protocol)网际协议、网络控制报文协议ICMP(Internet Control Message Protocol)网络控制报文协议、地址解析协议ARP(Address Resolution Protocol)地址解析协议、用户数据包协议UDP(User Datagram Protocol)以及传输控制协议TCP(Transmission Control Protocol)，它们分别对应了网络分层的数据链路层、网络层和运输层[5][21]。在整个系统中，这些协议是通过对DSP进行软件编程来实现的。

3.4.2 地址解析协议 ARP

在任何时候，当主机或路由器需要找出这个网络上的另一个主机或者路由器的物理地址时，它必须有接收段的 IP 地址。但是 IP 地址并不能直接用来进行通信，这是因为 IP 地址只是主机在抽象的网络层中的地址，若要把网络层中传送的数据交给目的主机，还要传到链路层转变成 MAC 帧后才能发送到实际的网络上。因此，不管网络层使用的是什么协议，在实际网络的链路上传送数据帧时，最终还是必须使用物理地址。

这就表示，发送端必须有接收端的物理地址。因此在主机中，需要存放一个从 IP 地址到物理地址的映射表，并且这个映射表还必须能够经常动态更新。

地址解析协议 ARP(Address Resolution Protocol)很好的解决了这个问题，它能

把 IP 地址和它的物理地址关联起来。任何时候当主机或者路由器需要找出这个网络上的另一个主机或路由器时，它就发送 ARP 查询组。这个分组包括发送端的物理地址和 IP 地址，以及接收端的 IP 地址。因为发送端不知道接收端的物理地址，查询就在网络上广播。

在网络上的每一个主机或路由器都接收和处理这个 ARP 查询分组，但只有意图中的接收者才识别 ARP 分组的 IP 地址，并发回 ARP 响应分组。这个响应分组包含有接收者的 IP 地址和物理地址。这个分组用单播直接发送给查询者，它使用收到的查询分组中所有的物理地址。

如图 3-4 为 ARP 的分组格式，主要字段有：硬件类型、协议类型、硬件长度、协议长度、操作、发送端硬件地址、发送端协议地址、目标硬件地址和目标协议地址。其中对于 ARP 请求报文，目标硬件地址字段为 0，因为发送端发送 ARP 时并不知道目标的物理地址[10][11]。

<table>
<tr><td colspan="2">硬件类型</td><td>协议类型</td></tr>
<tr><td>硬件长度</td><td>协议长度</td><td>操作（请求1、回答2）</td></tr>
<tr><td colspan="3">源硬件地址（例如：对以太网是6字节）</td></tr>
<tr><td colspan="3">源协议地址（例如：对以太网是6字节）</td></tr>
<tr><td colspan="3">目标物理地址（例如：对以太网是6字节）（请求时不填）</td></tr>
<tr><td colspan="3">目标协议地址（例如：对以太网是6字节）</td></tr>
</table>

图 3-4 ARP 的分组格式

Figure3-4 Format of ARP

一个典型的 ARP 主要步骤如下：

(1) 发送端知道目标 IP 地址。

(2) IP 请求 ARP 产生 ARP 请求报文，并填入源物理地址、源 IP 地址以及目标 IP 地址。目标物理地址则填入 0。

(3) 将这个 ARP 报文发送给数据链路层，然后被封装成数据帧，将发送端的物理地址作为源地址，而把物理广播地址作为目标地址。

(4) 每一个主机或路由器都收到这个帧。因为这个帧使用了广播目的地址，所有的站都把这个报文交给 ARP。除了目标主机外，所有的主机都丢弃这个分组，

因为只有目标主机识别这个 IP 地址。

(5) 目标主机用 ARP 回答报文进行回答，这个报文包含它的物理地址。ARP 回答报文使用单播的方式进行传送。

(6) 发送端收到这个回答报文后，就知道了目标主机的物理地址。

(7) 源端口将 IP 数据报用单播的方式发送给目的主机。

需要指出的是，ARP 是解决同一个局域网上的主机或路由器的 IP 地址和物理地址的映射问题。如果所要找的主机不在同一个局域网上，那么主机就无法解析，这种情况下，它需要先解析所要传送 IP 数据报的中间路由器的物理地址，并通过该路由器转发 IP 数据报。

除了地址解析协议 ARP 外，还有逆地址解析协议 RARP(Reverse Address Resolution Protocol)。和 ARP 相反，RARP 能使一个只知道自己的物理地址的主机知道其 IP 地址。

3.4.3 网际协议 IP

网际协议 IP 是 TCP/IP 体系中两个最主要的协议之一，也是最重要的因特网标准协议[RFC 791]之一，它是为计算机网络相互连接进行通信而设计的协议。在因特网中，它能使连接到网上的所有计算机网络实现相互通信，并且规定了计算机在因特网上进行通信时应当遵守的规则。IP 协议的责任是把数据从源传送到目的地。它不负责保证传送可靠性、流控制、包顺序和其它主机之间的协议[10][11]。

IP 协议是整个 TCP/IP 协议中最为核心的协议，TCP，UDP，ICMP 以及 IGMP 数据报都以 IP 数据报格式传输。

与 IP 协议配套使用的还有四个协议：

- 地址解析协议 ARP(Address Resolution Protocol)
- 逆地址解析协议 RARP(Reverse Address Resolution Protocol)
- 因特网控制报文协议 ICMP(Internet Control Message Protocol)
- 因特网组管理协议 IGMP(Internet Group Management Protocol)

1) IP 协议的功能

IP 协议提供的是不可靠、无连接的数据报传送服务。它只提供最好的传输服务，但不保证 IP 数据报能成功地到达目的地。如发生某种错误时，IP 仅将数据报丢弃，然后发送 ICMP 消息给信源端。而且 IP 协议对数据报的处理是相互独立的，每个数据报都独立的进行路由选择。

IP 协议的功能，是对要发送的数据进行加 IP 报头的处理，再把打好包的 IP 数据报发送给 MAC 层，通过数据报在一个个 IP 协议模块间传送，直到数据报达

到目的模块。同时，IP 层接收由更低层发来的 IP 报，对接收到的数据报进行报头校验和拆报处理，并把数据发送到更高层——TCP 层或 UDP 层。网络中每个主机和网关设备上都有 IP 模块，数据报在一个个模块间通过路由处理网络地址传送到目的地址。IP 在提供网络层服务时，采用了统一的报头，以使处于各子网中的 IP 都能根据报头对数据作出相应的处理。

每个数据报都有报头和报文这两个部分，报头中有目的地址等必要内容，使每个数据报不经过同样的路径都能准确地到达目的地。在目的地重新组合还原成原来发送的数据。这就要 IP 具有分组打包和集合组装的功能。在实际传送过程中，数据报还要能根据所经过网络规定的分组大小来改变数据报的长度，IP 数据报的最大长度可达 65535 个字节。

IP 协议中还有一个非常重要的内容，那就是给因特网上的每台计算机和其它设备都规定了一个唯一的地址，叫做“IP 地址”。由于有这种唯一的地址，才保证了用户在连网的计算机上操作时，能够高效而且方便地从千万台计算机中选出自己所需的对象来。

2) IP 协议的格式

IP 数据报的格式能够说明 IP 协议都具有什么功能，在 TCP/IP 的标准中，各种数据格式常常以 32bit(即 4 字节)为单位来描述。图 3-5 是完整的 IP 数据报格式。

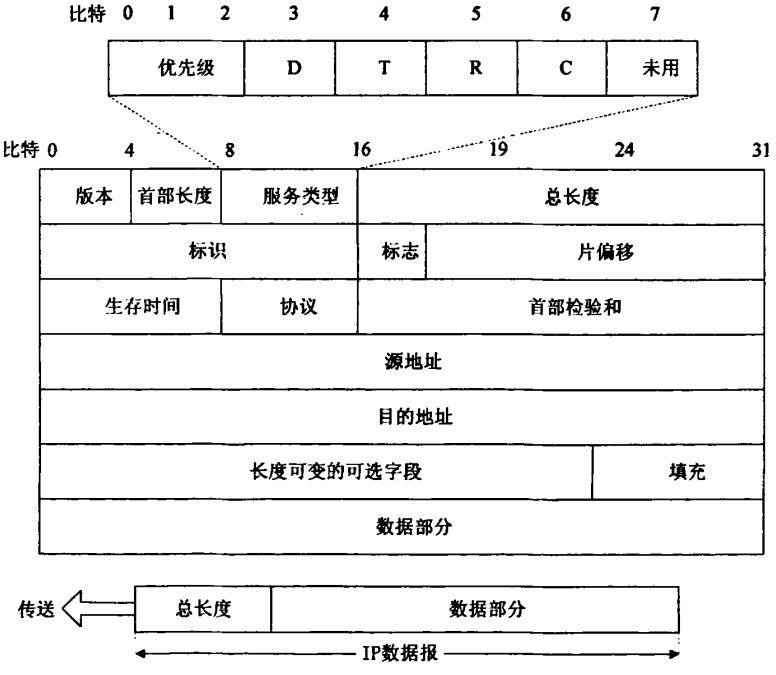


图3-5 IP数据报的格式

Figure3-5 Format of IP

从图3-5中可以看出，一个IP数据报由首部和数据两部分组成。首部的前一部分是固定长度的20个字节，首部的后一部分是长度可变。下面介绍首部各字段的意义：

(1) 版本：占4bit，指IP协议的版本。通信双方适用的IP协议的版本必须一致。目前广泛使用的IP版本号为4(IPv4)。

(2) 首部长度：占4bit，可表示的最大数值是15个单位(一个单位为4字节)，因此IP的首部长度的最大值是60字节。当IP分组的首部长度不是4字节的整数倍时，必须利用一个填充字段加以填充。因此数据部分永远在4字节的整数倍开始，这样实现IP协议时较为方便。首部长度限制为60字节的缺点是有时不够用。但这样做是希望尽量减少开销。最常用的首部长度是20字节，即不使用任何选项。

(3) 服务类型：占8bit，用来获得更好的服务，其意义如图8上面的部分所示。

- 前3个比特表示优先级，它可使数据报具有8个优先级中的一个。
- 第4个比特是D比特，表示要求有更低的时延。
- 第5个比特是T比特，表示要求有更高的吞吐量。
- 第6个比特是R比特，表示要求有更高的可靠性。
- 第7个比特是C比特，是新增加的，表示要求选择费用更低的路由。

(4) 总长度：总长度指首部和数据之和的长度，单位为字节。总长度段为16bit，因此数据报的最大长度为65535字节(即64KB)

(5) 标识：占16bit，它是一个计数器，用来产生数据报的标识。但这里的“标识”并没有序号的意思，因为IP是无连接服务，数据报不存在按序接收的问题。当IP协议发送数据时，它就将这个计数器的当前值复制到标识字段中。

(6) 标志：占3bit，目前只有前两个比特有意义。

- 标志字段中的最低位记为MF。MF=1，表示后面“还有分片”的数据报。MF=0，表示这已是若干数据报片中的最后一个。
- 标志字段中间的一位记为DF，意思是“不能分片”。只有当DF=0时才允许分片。

(7) 片偏移：较长的分组在分片后，某片在原分组中的相对位置。也就是说，相对于用户数据字段的起点，该片从何处开始。片偏移以8个字节为偏移单位。

(8) 生存时间：数据报在网络中的寿命。

(9) 协议：占8bit，协议字段指出此数据报携带的数据是使用何种协议，以便使目的主机的IP层知道应将此数据报上交给哪个进程。

(10) 首部检验和：此字段只检验数据报的首部，不包括数据部分。

(11) 源地址：占4字节，这是首部中最重要的字段。

(12) 目的地址：占4字节，这也是首部中最重要的字段。

3.4.4 用户数据报协议 UDP

运输层是整个网络体系结构中的关键层次之一，它属于面向通信部分的最高层，同时也是用户功能中的最低层，主要负责向其上的应用层提供通信服务。

根据应用的不同，运输层有两种不同的运输协议，即面向连接的传输控制协议 TCP(Transmission Control Protocol)和无连接的用户数据报协议 UDP(User Datagram Protocol)。

UDP 在传送数据之前不需要先建立连接。远地主机的运输层在收到 UDP 报文后，不需要给出任何确认。虽然 UDP 不提供可靠交付，但在某些情况下 UDP 却是一种最有效的工作方式[10][11][12]。

TCP 则提供面向连接的服务。在传送数据之前必须先建立连接，数据传送结束后要释放连接。TCP 不提供广播或者多播服务。由于 TCP 要提供可靠的、面向连接的运输服务，因此不可避免地增加了许多的开销，如确认、流量控制、计时器以及连接管理等。这不仅使协议数据单元的首部增大很多，还有占用许多的处理器资源。

1) UDP 的优点

用户数据报协议 UDP 只在 IP 的数据报服务之上增加了很少一点的功能，这就是端口的功能（有了端口，运输层就能进行复用和分用）和差错检测的功能。虽然 UDP 数据报只能提供不可靠的交付，但 UDP 在某些方面有其特殊的优点：

(1) UDP 是无连接的，即发送数据前不需要建立连接（当然发送数据结束时也没有连接可以释放），因此减少了开销和发送数据之前的时延。

(2) UDP 使用尽最大努力交付，即不保证可靠交付，同时也不使用拥塞控制，因此主机不需要维持具有许多参数的、复杂的连接状态表。

(3) UDP 没有拥塞控制，因此网络出现的拥塞不会使源主机的发送速率降低。这对某些实时应用是很重要的。很多的实时应用（如 IP 电话、实时视频会议等）要求源主机以恒定的速率发送数据，并且允许在网络发送拥塞时丢失一些数据，但却不允许数据有太大的时延，UDP 正好适合这种要求。

(4) UDP 是面向报文的。这就是说，UDP 对应用程序交下来的报文不再划分为若干个分组来发送，也不把收到的若干个报文合并后再交付给应用程序。应用程序交给 UDP 一个报文，UDP 就发送这个报文；而 UDP 收到一个报文，就把它交付给应用程序。因此，应用程序必须选择合适大小的报文。若报文太长，UDP 把它交给 IP 层后，IP 在传送时可能要进行分片，这会降低 IP 层的效率。反之，若报文太短，UDP 把它交给 IP 层后，会使 IP 数据报的首部相对变大，这也会降低 IP 层的效率。

(5) UDP 支持一对一、一对多和多对多的交互通信。用户数据报只有 8 个字节的首部开销，比 TCP 的 20 个字节的首部要短。

虽然某些实时应用需要使用没有拥塞控制的 UDP，但当很多的源主机同时都要向网络发送高速率的实时视频流时，网络就有可能发生拥塞，结果大家都无法正常接收。因此 UDP 不使用拥塞控制功能可能会引起网络产生严重的拥塞问题。

2) UDP 报文的结构

根据 TCP/IP 协议的分层规范，UDP 数据帧可以分为四层结构，其具体封装结构如图 3-6 所示。该图清晰地显示出了 UDP 数据帧的各分层结构以及各层数据字段的内容：分别为 MAC 层（以太网层）、IP 层（因特网协议）、UDP 层（用户数据报协议）以及所要发送的数据（额外数据）[12][34]。FCS（帧校验序列）是由 CS8900A 所产生的 CRC 校验，并不是数据帧的一部分，可以通过对 CS8900A 的相应寄存器的设置而屏蔽该字段。

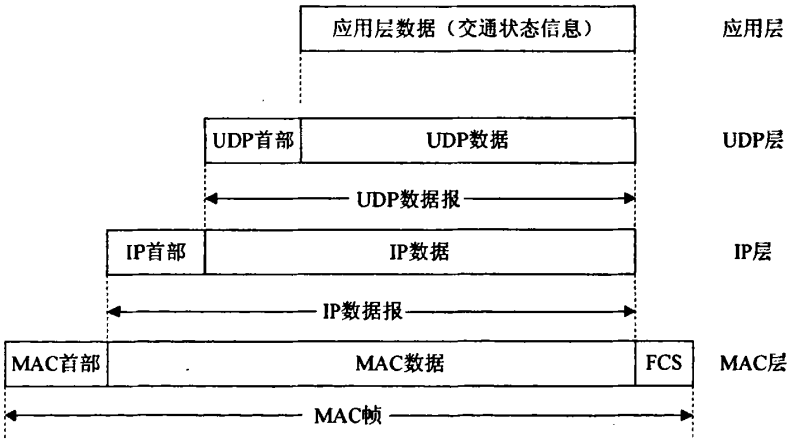


图3-6 UDP数据帧的以太网分层封装结构

Fig3-6 Encapsulation of UDP

用户数据报UDP有两个字段：数据字段和首部字段。如图3-7所示，UDP的首部仅包括四个数据字段，分别为源端口、目的端口、UDP数据报长度和检验和，每个字段都是两个字节，各字段意义如下所示：

(1) 源端口字段和目的端口字段

包含了16bit的UDP协议端口号，主要用于寻找发送端和接收端应用进程，通过这两个值再加上IP首部的源IP地址和目的IP地址可以唯一确定一个UDP数据的传送，UDP的端口号和TCP的端口号是相互独立的。其中源端口字段是可选的。若选用，则字段值是应答报文应该回送的端口值，若不选用时其值为零。

(2) 长度字段

记录了该UDP数据报的字节数，这个长度包括了首部和用户数据段，因此长

度字段的最小值是8，即首部的长度。

(3) 检验和字段

UDP数据的校验和覆盖UDP首部和UDP数据，UDP数据的校验和是一个端到端的校验和。由发送端进行计算而由接收端进行验证，主要目的是为了发现UDP首部和数据在发送端和接收端之间的发生的任何改变，UDP的校验和是可选的，但是协议还是建议它们应该总是在用，如果该字段值为零就说明不进行校验。设计者把这个字段作为可选项的目的，是为了那些在可靠性很好的局域网上使用UDP的实现者尽量减少开销。

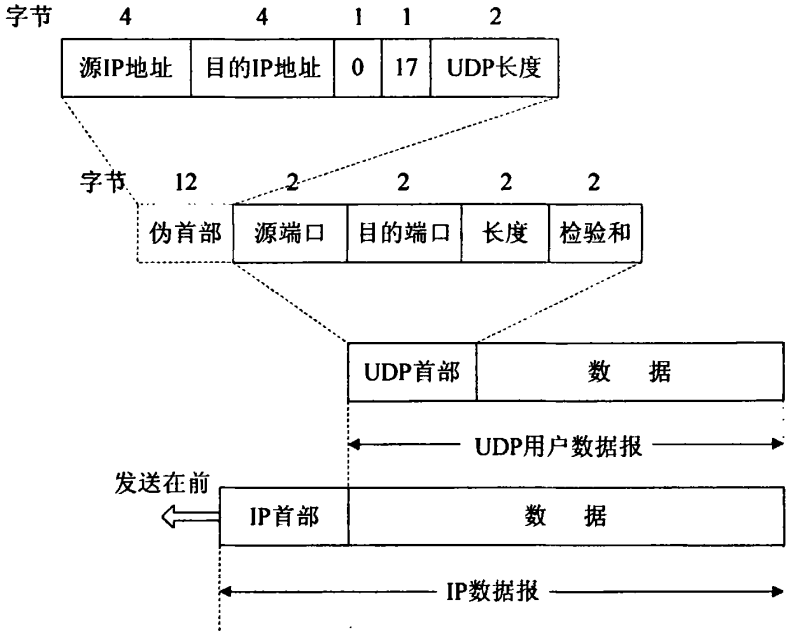


图3-7 UDP用户数据报的首部和伪首部

Figure3-7 Head and Fake Head of UDP

UDP用户数据报首部中检验和的计算方法有些特殊。在计算检验和时在UDP用户数据报之前要增加12个字节伪首部。所谓“伪首部”是因为这种伪首部并不是UDP用户数据报真正的首部。只是在计算检验和时，临时和UDP用户数据报连接在一起，得到一个过渡的UDP用户数据报。检验和就是按照这个过渡的UDP用户数据报来计算的。伪首部既不向下传送，也不向上递交。图3-7给出了伪首部各字段的内容。

UDP计算检验和的方法和计算IP首部检验方法相类似，在发送端，首先是将全零放入检验和字段，再将伪首部以及UDP用户数据报(包括数据字段)看成是许多16bit的字串接起来。若UDP用户数据报的数据部分不是偶数个字节，则要填入一个全零字节(但此字节不发送)，然后按二进制反码计算出16bit字的和，将此和的二

进制反码写入检验和字段后，发送此UDP数据报。在接收端，将收到的UDP用户数据报连同伪首部(以及可能的填充全零字节)一起，按二进制反码计算出这些16bit字的和。当无差错时其结果应为全1，否则就表明有差错出现，接受端就应将此UDP用户数据报丢弃[10][12]。

3.4.5 传输控制协议 TCP

传输控制协议TCP(Transmission Control Protocol)是TCP/IP协议中面向连接的运输层协议，它提供全双工的和可靠交付的服务。TCP与运输层中的另一个协议用户数据报协议UDP最大的区别就是TCP是面向连接的，而UDP是无连接的[10][11][12]。

TCP的数据传输过程较为简单，发送方的应用进程按照自己产生数据的规律，不断地把数据块（其长短可能各异）陆续写入到TCP的发送缓存中。TCP再从发送缓存中取出一定数量的数据，将其组成TCP报文段（segment）逐个传送给IP层，然后发送出去。接收方从IP层收到TCP报文段后，先把它暂存在接收3缓存中，然后让接收方的应用程序从接收缓存中将数据块逐个读取[10]。

由于运输层的通信是面向连接的，因此TCP每一条连接上的通信只能是一对一的，而不可能是一对多、多对一或者多对多的。

TCP具有确认和重传机制，这就可以在IP层的通信是不可靠的情况下，使运输层之间的通信成为可靠的。

1) TCP可靠通信实现的复杂性

TCP具有确认和重传机制，这就可以在IP层的通信是不可靠的情况下，使运输层之间的通信成为可靠的。但是TCP可靠通信的具体实现要复杂很多，主要有如下四个原因：

(1) TCP的报文段长度是不确定的，以报文段作为确认的单位不够方便，TCP是用字节序号进行确认的。

(2) TCP能够提供全双工通信，即双方可同时发送数据。因此，不必专门发送确认报文段，而可以在发送自己的数据文段的同时，捎带地把确认信息附上。

(3) 为了提高通信传输效率，发送数据报文段的一方，可以连续发送多个数据报文段，而不需要在收到一个确认后才发送下一个报文段。但运输层的一方究竟可以连续发送多少个报文段，这又取决于网络当时的拥塞程度以及对方当时的接收能力，这也是一个比较复杂的问题。

(4) 要使用特定的算法来估算比较合适的重传时间。

2) TCP报文的结构

一个TCP的报文段分为首部和数据两部分，TCP的全部功能都体现在它首部中

各个字段的作用。TCP报文段的前20个字节是固定的，后面有4N字节是根据需要而增加的选项（N必须是整数）。因此TCP首部的最小长度是20字节。

首部固定部分各字段分配较多，各字段内容意义相对复杂，由于论文篇幅限制，因此关于各字段的内容，本处只简要地给出与本系统实验设计相关的关键字段解释，其余内容不再赘述，TCP报文段的排列结构如图3-8所示。

在TCP的数据字段中，序号为本段报文数据的第一个字节的序号，确认号是期望收到对方下一个报文的段第一个数据字节的序号，数据偏移指出TCP报文段的数据起始处距离TCP报文段的起始处有多远，这实际上是TCP报文段首部的长度。由于四位二进制数能够表示的最大十进制数字是15，因此数据偏移的最大值是60字节，这也是TCP首部的最大长度。

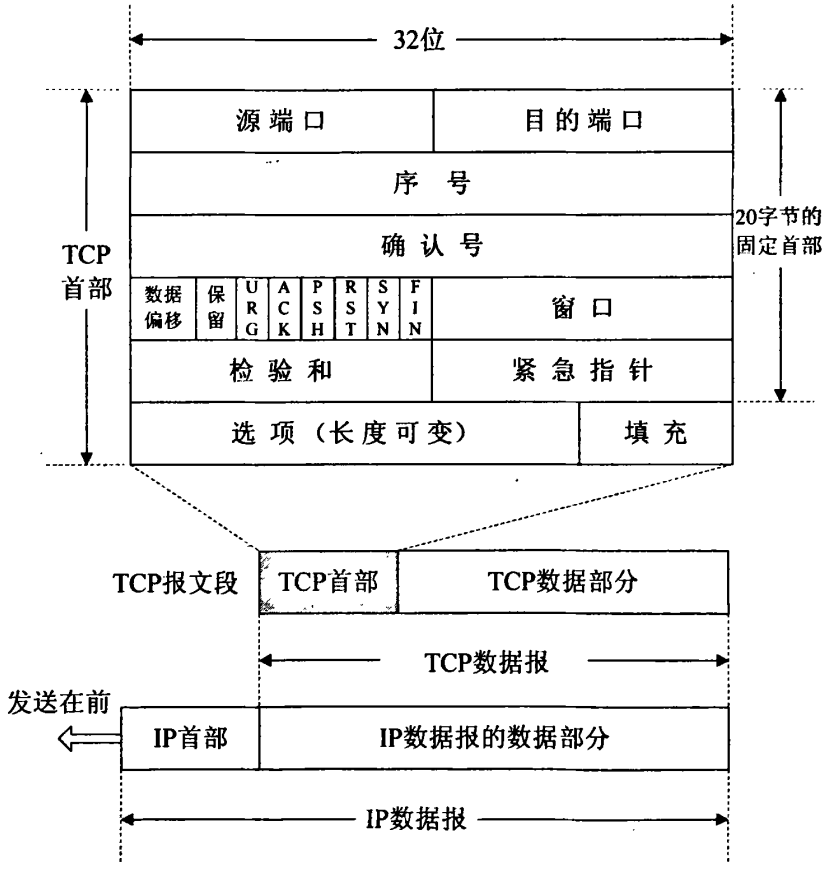


图3-8 TCP报文段的首部

Figure3-8 Head of TCP

在TCP的数据字段中，序号为本段报文数据的第一个字节的序号，确认号是期望收到对方下一个报文的段第一个数据字节的序号，数据偏移指出TCP报文段的数据起始处距离TCP报文段的起始处有多远，这实际上是TCP报文段首部的长度。由

于四位二进制数能够表示的最大十进制数字是15，因此数据偏移的最大值是60字节，这也是TCP首部的最大长度。

在数据字段中，有6个控制位说明本报文字段的性质，它们也是TCP复杂的连接功能的体现。

- 紧急URG=1时，表明紧急指针字段有效，它告诉系统此报文段汇总有紧急数据，应尽快传送（相当于高优先级的数据），而不是按原来的排队顺序来传送。
- 确认ACK=1时，表明确认字段有效；当ACK=0时，确认号无效。
- 推送PSH=1时，表明该应用进程在接收端被接收到时需要立即响应发送端，而不是等到整个缓存都填满后再向上交付。
- 复位RST=1时，表明TCP连接中出现严重差错（如由于主机崩溃等原因），必须释放连接，然后再重新建立运输连接。RST置1还用来拒绝一个非法的报文段或拒绝打开一个连接。RST也可称为重建位或重置位。
- 同步SYN用来在连接建立时用来同步序号，SYN=1而ACK=0时，表明这是一个连接请求或连接接受报文。
- 终止FIN用来释放一个连接。当FIN=1时，表明此报文段的发送方的数据已发送完毕，并要求释放运输连接。

3) TCP的运输连接管理

作为面向连接的协议，运输连接是用来传送TCP报文的，它的建立和释放是每一次面向连接的通信中必不可少的过程。

TCP运输的连接有三个阶段，即：连接建立、数据传送和连接释放[10][11][12]。运输连接的管理就是使运输连接的建立和释放都能正常地进行。

连接建立过程中要解决以下三个问题：

- 要使每一方能够确知对方的存在；
- 要允许双方协商一些参数（如最大报文段长度，最大窗口大小，服务质量等）；
- 能够对运输实体资源（如缓存大小，连接表中的项目等）进行分配。

TCP的连接和建立都是采用客户服务器方式。主动发起连接建立的应用进程叫做客户(client)。被动等待连接建立的应用进程叫做服务器(server)。

TCP建立连接采用的过程叫做三次握手(three-way handshake)，或三次联络。

在数据传输结束后，通信双方都可以发出释放连接的请求，在关闭过程中，有半关闭和全关闭两种状态，从一个主机到另一个主机的释放叫半关闭，两个主机都释放连接叫全关闭。连接释放的过程和建立连接时的三次握手在本质上是一致的。

通信的连接和释放是本论文TCP实现部分的重点，为了更好的描述本系统中的TCP功能，将该部分的叙述放到第四章试验设计部分，结合系统试验，以使描述更加清楚。

3.5 网络协议在 DSP 中的实现

由于各种不同帧的信息包含在帧的各层格式当中，因此以太网协议的具体实现主要通过发送数据前的打包和发送数据后的解包程序来完成。在发送帧时，先对要发送的数据按照所要封装的协议格式进行分层打包，通过在数据前加各层的首部实现，然后送入发送程序进行发送。在接收帧时，先接收由网络发送来的帧，然后通过对该帧各层首部信息的解读，确定该帧种类，提取所需要的数据信息，并进行相应的处理。

在进行ARP请求帧的发送时，需要注意帧格式中最后一个字段的内容是目的IP地址，在填写该字段之前需要先判断要询问硬件地址的IP地址是否在本地IP地址所在的局域网内，如果是，则直接写入该硬件地址作为目的IP地址，如果不是，则需要写入相应的网关地址，通过路由器的转发功能来交付数据帧。

在进行以太网协议的软件实现时，需要严格按照各种帧的分层格式进行打包，根据各个字段的定义写入相应的值。由于本设计中对CS8900A采用16位的操作方式，因此，在写入帧数据时，需要对每个16位字的前后两个8位的字节进行交换，同样在接收到数据时也需要进行相应的交换操作。

值得注意的是，在用CS8900A进行以太网数据传输时，存在对整个数据帧的CRC检验，但是该检验过程只存在于以太网控制器中，用来判断发送和接收的数据帧是否有效，并不进入DSP参与上层的数据处理。

4 实验设计及结果分析

在硬件和软件设计的基础之上，针对大量交通数据传输的需要，设计了相应的以太网功能以满足系统的需求，本部分将通过相应的实验来证明这些设计的可行性和合理性。

本节点实验主要分为三个部分：其一是实现了基础的ARP、UDP的数据发送和接收功能，包括ARP和IP数据报的填充，以及IP数据报的分片功能，根据本系统需求，还实现了简化了状态机的TCP通信功能；其二是设计和实现了相应的以UDP为核心的数据传输流程来证明本软硬件设计的可行性；其三是通过模拟道路交通现场数据传输网络来证明在本设计基础上采用TCP/IP协议传输大量交通状态数据的合理性。

在本实验中，技术实现平台主要通过PC机和复合节点之间的通信来进行，采用科来网络分析系统来实现对网络数据的监控和分析，采用网络调试助手NetAssist作为PC机的发送和接收界面。

在科来网络分析系统及调试助手监控到的信息中，会包含有相关的IP地址及其端口信息，其中，202.112.155.148为PC机的IP地址，202.112.155.147为设置的网络复合节点的IP地址，qw-4e2253df4fle、69b58e1bf6eb4d1或qw为PC机名称，在科来监控系统中，会直接采用PC机名称来代替PC机的IP地址。在IP地址后是端口号，本节点在ARP和UDP的相关实验中，设置PC机端口号为5000或6000，网络复合节点端口号为1025。而在TCP的相关实验中，PC机的端口号是1024，网络复合节点的端口号是1028。PC机MAC地址为00-0D-61-9E-1C-CC，网络复合节点的MAC地址为02-E0-4C-A0-7E-7A。由于这些网络设置在后面实验中都要用到，并且设置都相同，因此此处先做说明，后面将不再赘述。

由于是点对点的通信，因此，在RJ45的连接上，需要交换网络连接线的线序，如果采用集线器(hub)则无需改变。

4.1 底层驱动及相关协议的实现

4.1.1 数据包的发送和接收

数据的发送和接收是以太网节点底层驱动设计是否正确的基本体现，在节点设计功能的逐步实现过程中，首先需要完成数据的发送和接收过程。本实验设计以封装较为简单的UDP数据包的收发为例，以验证本节点在硬件和软件的底层设

计是否正确，从而为上层网络协议的实现奠定基础。

1) UDP 数据报的发送

本实验的主要过程为从网络复合节点向 PC 机循环发送 UDP 数据包，用科来网络分析系统监控 UDP 数据包信息，通过网络调试助手对节点发送的数据包进行接收，从而验证节点接收驱动的正确性。

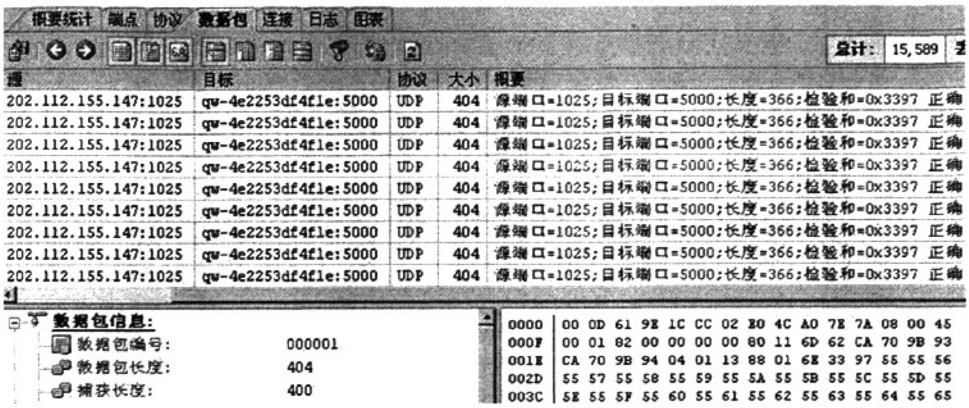


图 4-1 网络线路上的 UDP 数据帧

Figure4-1 UDP in Network

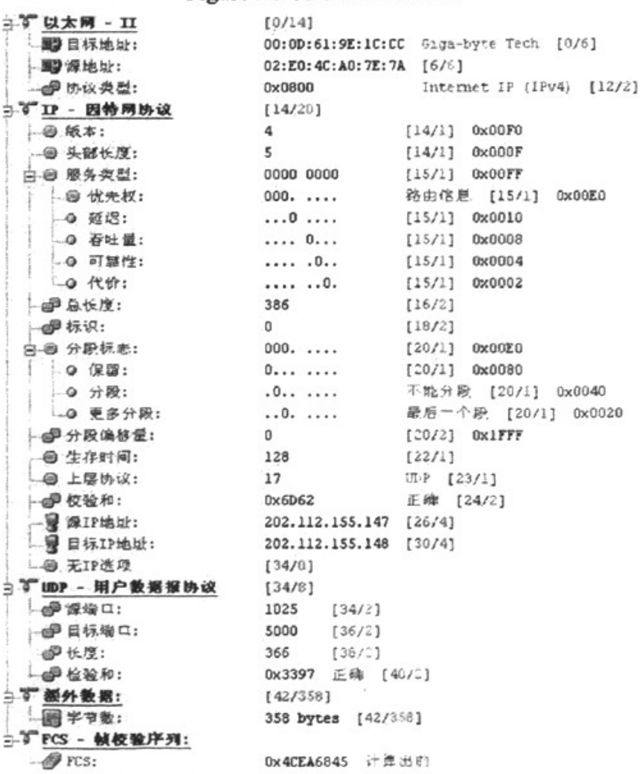


图 4-2 UDP 数据帧的解析图

Figure4-2 Analysis of UDP

数据发送程序流程如 3.3.1 中图 3-1 所示，只需要持续查询相应的寄存器，循环操作即可实现 UDP 数据包的持续发送，此处不再赘述。

在本实验中，设置发送数据包数据字段长度为 358 字节，再加上 8 字节的 UDP 首部，UDP 的数据帧总长度为 366 字节。如图 4-1 为科来网络监控软件所监控到的网络线路上 UDP 数据帧发送的界面截图，图 4-2 为其中任意一个 UDP 帧所包含的信息，其中，IP 层和 UDP 层的检验和都显示正确，上层协议为 17 表示采用的协议为 UDP。如图 4-3 为网络调试助手接收到的 UDP 数据帧界面图，其中各数据包信息都符合发送的 UDP 信息，证明 PC 机正确接收到了网络节点发送的 UDP 帧。

图 4-3 PC 机接收到的 UDP 数据报的界面

Figure4-3 Data Interface of UDP Received by PC

2) UDP 数据报的接收

本实验主要过程为从 PC 机向网络复合节点循环发送 UDP 数据包，通过科来网络分析系统监控数据包信息，通过查看网络复合节点的相应接收寄存器内的数据，来验证节点接收驱动的正确性。

在实验设计中，PC 需要连续向复合节点发送数据包，这通过网络调试助手

NetAssist 来实现，如图 4-4 所示，其中界面下方的窗口为要发送的数据窗口。

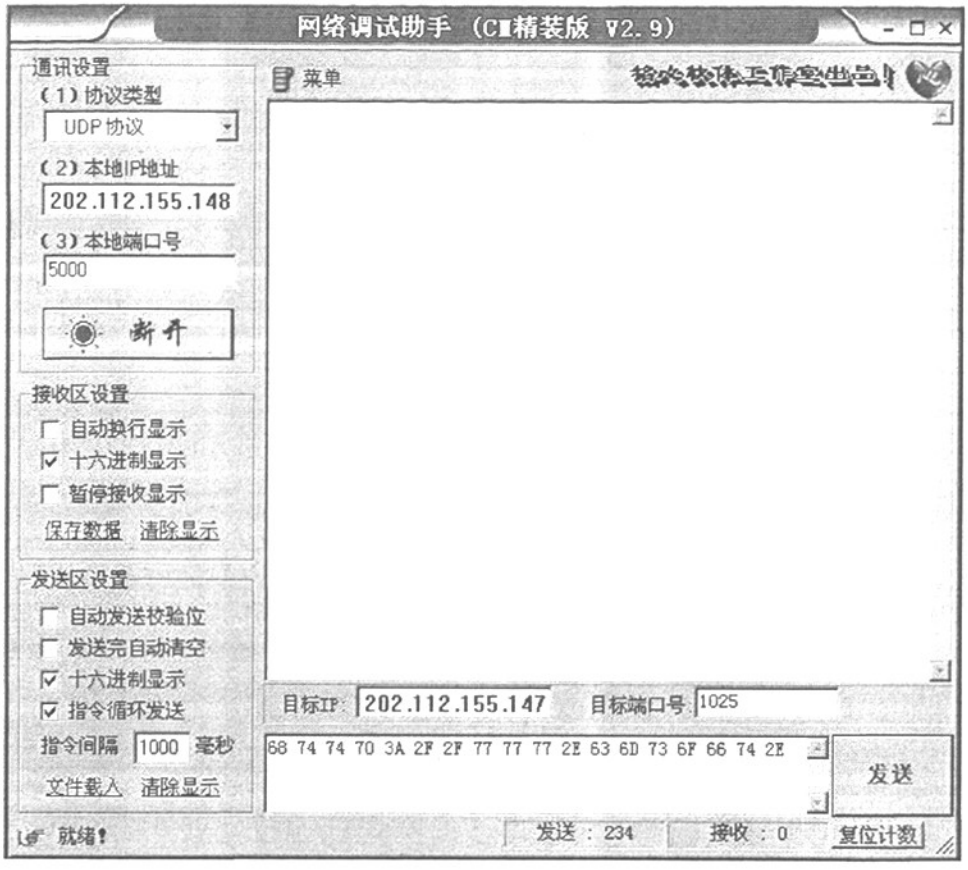


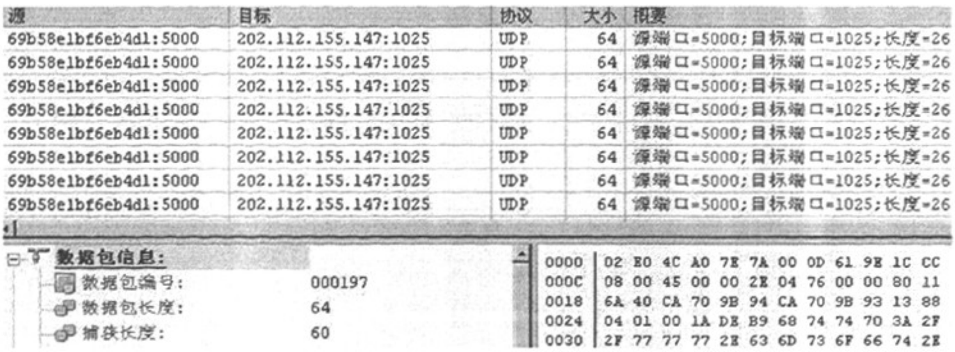
图 4-4 PC 机发送 UDP 数据报的界面

Figure4-4 Data Interface of UDP Sent by PC

在 UDP 数据包的接收过程中，可以采用查询和中断两种方式来实现的，前者通过持续查询 0x0000H 来实现查询方式的持续接收，而后者需要通过中断状态队列 ISQ 判断是否为接收事件，然后进入接收中断子程序进行数据包的接收。

本部分的实现中，采用了中断的方式来进行数据的接收，这是由嵌入式系统的实时性要求所决定的(详见 3.3 节)，其数据接收程序流程如 3.3.2 中图 3-2 所示，具体内容参照 3.3.2 节前面，此处不再赘述。

各 IP 地址和端口的设置都和发送过程相同。在本实验中，设置 PC 机的发送数据包数据字段长度为 18 字节，再加上 8 字节的 UDP 首部，UDP 的数据帧总长度为 26 字节。如图 4-5 所示，为科来网络监控软件所监控到的 UDP 数据帧的界面截图，其概要中的信息表明监控到的 UDP 数据报的长度为 26 字节。图 4-6 所示为任意一个 UDP 数据报所包含的信息，其中，IP 层和 UDP 层的检验和都显示正确，证明本 UDP 数据帧在传输过程中完全正确。



源	目标	协议	大小	概要
69b58e1bf6eb4d1:5000	202.112.155.147:1025	UDP	64	源端口=5000;目标端口=1025;长度=26
69b58e1bf6eb4d1:5000	202.112.155.147:1025	UDP	64	源端口=5000;目标端口=1025;长度=26
69b58e1bf6eb4d1:5000	202.112.155.147:1025	UDP	64	源端口=5000;目标端口=1025;长度=26
69b58e1bf6eb4d1:5000	202.112.155.147:1025	UDP	64	源端口=5000;目标端口=1025;长度=26
69b58e1bf6eb4d1:5000	202.112.155.147:1025	UDP	64	源端口=5000;目标端口=1025;长度=26
69b58e1bf6eb4d1:5000	202.112.155.147:1025	UDP	64	源端口=5000;目标端口=1025;长度=26
69b58e1bf6eb4d1:5000	202.112.155.147:1025	UDP	64	源端口=5000;目标端口=1025;长度=26
69b58e1bf6eb4d1:5000	202.112.155.147:1025	UDP	64	源端口=5000;目标端口=1025;长度=26

数据包信息:
数据包编号: 000197
数据包长度: 64
捕获长度: 60

```
0000  02 E0 4C A0 7E 7A 00 0D 61 9E 1C CC
000C  08 00 45 00 00 2E 04 76 00 00 80 11
0018  6A 40 CA 70 9B 94 CA 70 9B 93 13 88
0024  04 01 00 1A DE B9 68 74 74 70 3A 2F
0030  2F 77 77 77 2E 63 6D 73 6F 66 74 2E
```

图 4-5 网络线路上的 18 字节数据长度的 UDP 帧

Figure4-5 18 Bytes Data Length UDP in Network

图 4-6 18 字节数据长度 UDP 数据帧的解析图

Figure4-6 Analysis of 18 Bytes Data Length UDP

如图 4-7 所示为 CCS3.1 界面所显示的 DSP 接受缓存所接收到的数据，该 30 个字的数组中的数据为 DSP 接收到的 UDP 帧，其中数组 0 到 20 中的 21 字长的数据（即 42 字节数据）为 UDP 的帧头，包括 MAC、IP 和 UDP 帧头，从 21 到 29 的 9 字长的数据为接收到的 UDP 数据，可以对应于图 4-4 和图 4-5 中的数据，在

DSP 缓存中，每字数据的存放是两高低位字节前后交换的，因此 DSP 接收界面显示和发送显示的数据相反。

Name	Value	Type	Radix
RxEthnetFrameBuffer	0x003F9640	unsigned int[759]	hex
[0]	0xE002	Uint16	hex
[1]	0xA04C	Uint16	hex
[2]	0x7A7E	Uint16	hex
[3]	0x0D00	Uint16	hex
[4]	0x9E61	Uint16	hex
[5]	0xCC1C	Uint16	hex
[6]	0x0008	Uint16	hex
[7]	0x0045	Uint16	hex
[8]	0x2E00	Uint16	hex
[9]	0x4404	Uint16	hex
[10]	0x0000	Uint16	hex
[11]	0x1180	Uint16	hex
[12]	0x726A	Uint16	hex
[13]	0x70CA	Uint16	hex
[14]	0x949B	Uint16	hex
[15]	0x70CA	Uint16	hex
[16]	0x939B	Uint16	hex
[17]	0x8813	Uint16	hex
[18]	0x0104	Uint16	hex
[19]	0x1A00	Uint16	hex
[20]	0xB9DE	Uint16	hex
[21]	0x7468	Uint16	hex
[22]	0x7074	Uint16	hex
[23]	0x2F3A	Uint16	hex
[24]	0x772F	Uint16	hex
[25]	0x7777	Uint16	hex
[26]	0x632E	Uint16	hex
[27]	0x736D	Uint16	hex
[28]	0x666F	Uint16	hex
[29]	0x2E74	Uint16	hex
[30]	0x0000	Uint16	hex

图 4-7 CCS3.1 中 DSP 接收缓存所接收到的 UDP

Figure4-7 UDP in DSP Receive Buffer of CCS3.1

4.1.2 ARP 协议的实现

地址解析协议 ARP 是用来询问已知 IP 地址的网络节点的物理地址的一种链路层协议，它是以广播帧的形式发送出去的。各个网络节点或路由器都要接收并处理这个 ARP 数据帧，但只有 IP 地址符合的节点才用单播形式回发 ARP 响应，该响应包含有接收节点的 IP 地址和物理地址。

在本实验中，网络复合节点向 PC 机发送 ARP 广播帧，询问 PC 机的物理地址，PC 机响应该询问，回发单播的响应 ARP 帧，网络复合节点得到 PC 机的物理地址。

如图 4-8 为科来网络系统监控到的网络线路上的 ARP 帧的界面，其中包括节

点询问和主机响应两个 ARP 信息的传输过程。首先由 IP 地址为 202.112.155.147（物理地址为 02-E0-4C-A0-7E-7A）的源节点发送广播询问帧，询问 IP 地址为 202.112.155.148 的目标节点（PC 机）的物理地址，然后目标节点（PC 机）发送相应的响应 ARP，告诉源节点其物理地址为 00-0D-61-9E-1C-CC。

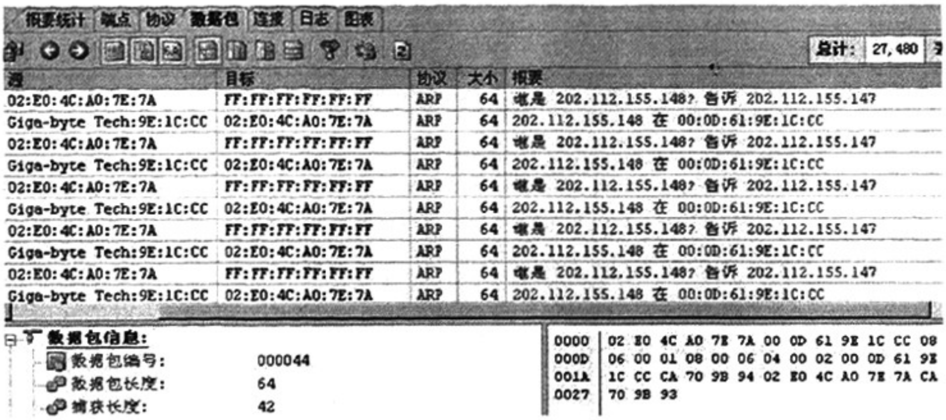


图 4-8 网络线路上的 ARP 帧

Figure4-8 ARP of Network

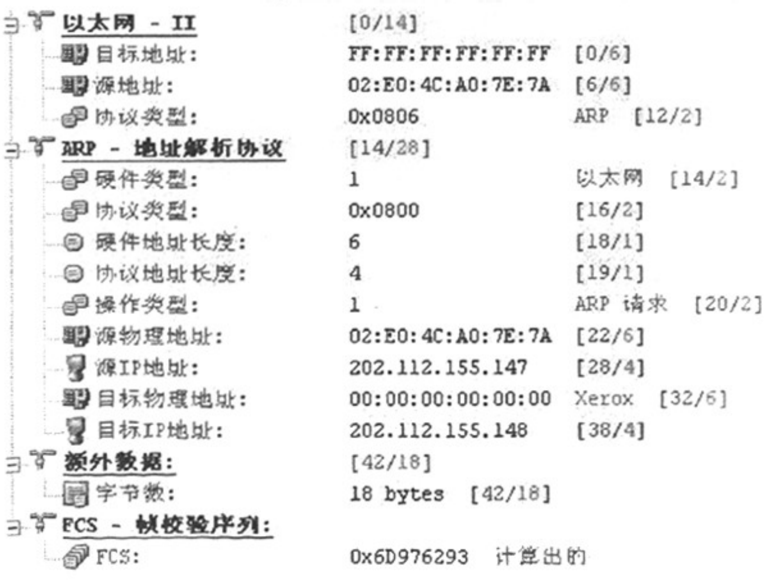


图 4-9 ARP 询问帧的解析图

Figure4-9 Analysis of Asking ARP

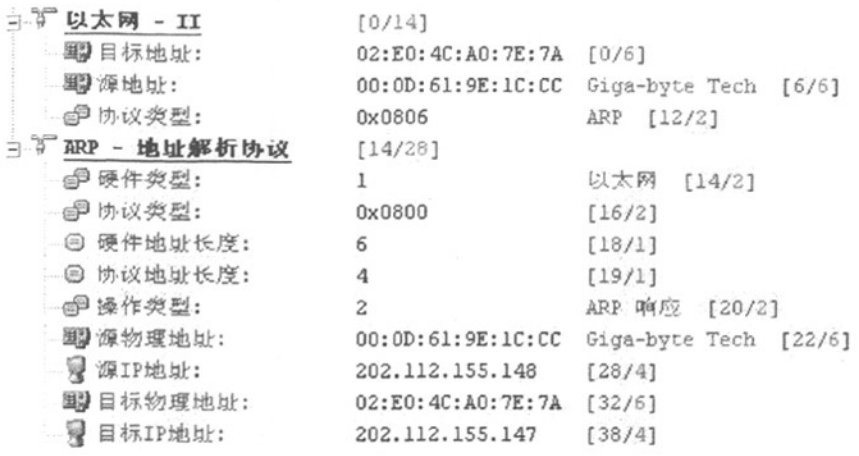


图 4-10 ARP 响应帧的解析图

Figure4-10 Analysis of Answering ARP

图 4-9 和图 4-10 分别为 ARP 询问数据包和 ARP 响应数据包的解析图，由于 ARP 直接封装在数据链路层内，因此，它并没有高层协议，包含信息也相对较少。在以太网控制器 CS8900A 的初始化设置中已经设定发送帧的有效长度为 64 字节到 1518 字节之间，因此帧长度为 42 字节的 ARP 数据帧会被自动添加数据 0 直到使长度符合链路层的发送要求：最短 64 字节。

4.2 数据报的填充

4.2.1 帧界定

帧界定是数据链路层的基本问题，数据链路层的发送方应当让接收方的数据链路层知道，所发送的帧是从什么地方开始到什么地方结束，这就是帧界定。

帧界定就是确定帧的界限。在发送帧时，发送方的数据链路层在帧前后都各加入事先商定好的标记，使得接收方在收到这个帧后，就能根据这种标记识别帧的开始和结束，以及帧里面装入的数据的准确位置。

在发送帧时，是从帧首部开始发送，各种数据链路层协议都要对帧首部和帧尾部的格式有非常明确的规定。显然为了提高帧传送效率，应当使帧的数据部分的长度尽可能的大于首部和尾部的长度。但是每一种链路层协议都规定了帧的数据部分的长度上限----最大传送单元 MTU（Maximum Transfer Unit）。当一个 IP 数据报封装成链路层的帧时，此数据报的总长度（即首部加上数据部分）一定不能超过下面的数据链路层的 MTU 值。如图 4-11 所示为最大传送单元 MTU。

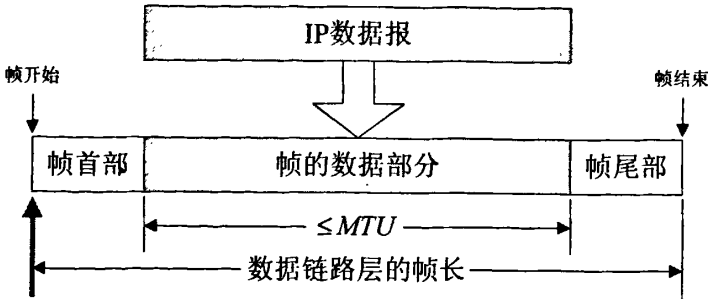


图 4-11 MTU

Figure4-11 MTU

4.2.2 MAC 帧的有效长度

虽然尽可能使用长的数据报会使传输效率提高，但由于以太网的普遍应用，所以实际上使用的数据报长度很少有超过 1500 字节的，而有时数据报长度还被限制在 576 字节。当数据报长度超过网络所容许的最大传送单元 MTU 时，就必须将过长的数据报进行分片后才能在网络上传送。这时数据报首部中的“总长度”字段不是指未分片前的数据报长度，而是指分片后的首部长度与数据长度的总和。

在 IEEE802.3 标准中，规定了凡出现下列情况之一即为无效的 MAC 帧：

- MAC 客户数据字段的长度与长度字段的值不一致。
- 帧的长度不是整数个字节。
- 收到的帧检验序列 FCS 查出有差错。
- 收到的帧的 MAC 客户数据字段的长度不在 46----1500 字节之间。考虑到 MAC 帧首部的长度是 18 字节，可以得出有效的 MAC 帧长度为 64----1518 字节。

对于检查出的无效 MAC 帧就简单地丢弃，以太网不负责重传丢弃的帧。当 MAC 客户数据字段的长度小于 46 字节时，则应加以填充（内容不限），这样，整个 MAC 帧（包括 14 字节首部和 4 字节尾部）的最小长度是 64 字节，或 512 位。

在一个完整的 MAC 数据帧中，除了 18 字节的 MAC 首部，还包含 20 字节的 IP 的首部和 8 字节的 UDP 首部，因此，除去该 28 字节首部，在 MAC 数据帧中，实际数据字段的长度范围应当在 18----1472 之间。在传送数据的过程中，应当将要发送的数据长度设在该范围之间，但有时候往往不能严格保证数据长度符合该要求，因而会出现不同的发送情况。

一般情况下，如果数据长度小于 64 字节，MAC 子层就会在数据字段的后面加入一个整数字节的填充字段，以保证以太网的 MAC 帧的长度不小于 64 字节。

而在大于 1472 字节的情况下，必须将过长的数据报进行分片后才能传输，本节首先将通过相关的实验来验证 IP 数据字段的填充并证明本功能设计的正确性。

4.2.3 ARP 数据报的填充

不论是何种数据帧，只有 MAC 帧的最小长度达到 64 字节才能进行传输，该情况最典型的是 ARP 帧的传输，因为 ARP 数据帧的总长度为 46 字节，因此要达到最小发送帧长度 64 字节，需要在其后填充至少 18 字节的额外数据才能发送。

实验分为两个过程：其一是由节点向 PC 机发送 ARP 询问帧，然后节点接收来自网络节点回发的 ARP 响应帧；其二是由 PC 机向网络节点发送 ARP 询问帧，然后 PC 机接收来自网络节点回发的 ARP 响应帧。

分别监控两个过程中来自节点的两个 ARP 帧，第一个过程中节点发出的 ARP 为询问帧，第二个过程中节点发出的 ARP 为响应帧。如图 4-12 和图 4-13 分别为两个过程中来自网络节点的 ARP 询问帧和 ARP 响应帧的解析图，由图可以看出，不论是询问还是响应，由网络节点发出的 ARP 帧都需要添加 18 字节的数据字段以使数据帧长度达到 MAC 帧发送的最小长度 64 字节，而该添加字段值为随机数。

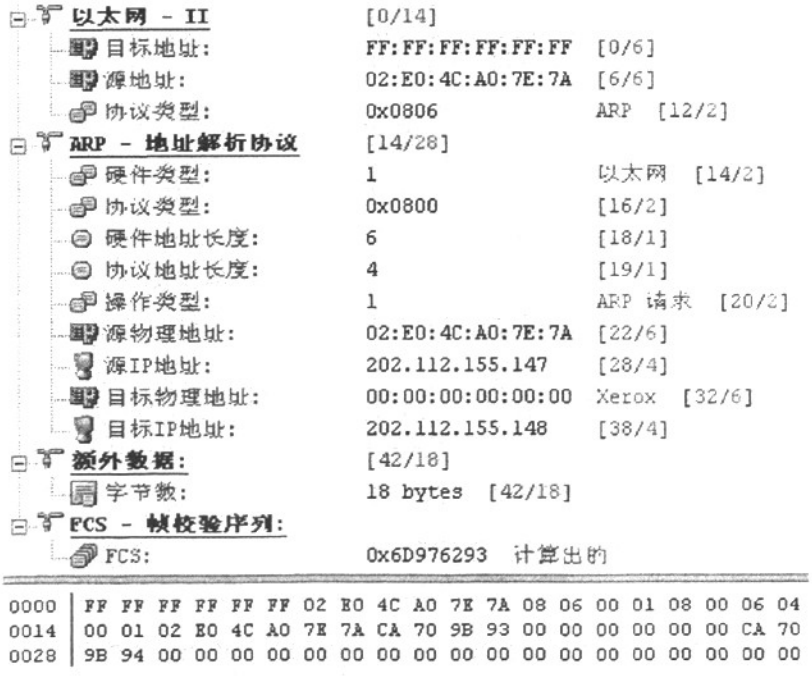


图 4-12 来自网络节点的 ARP 询问帧解析图

Figure4-12 Analysis of Asking ARP from Network Node

在 ARP 询问和响应中，需要返回的是被询问 IP 地址的 MAC 地址，这只

需要在 ARP 帧的相应地址位写入即可，ARP 并不包含附加信息及数据字段，因此不论是 ARP 帧是询问还是响应，都为 46 字节的长度，都需要在发送中填充 18 字节，以补足最少 64 字节的发送帧长度。

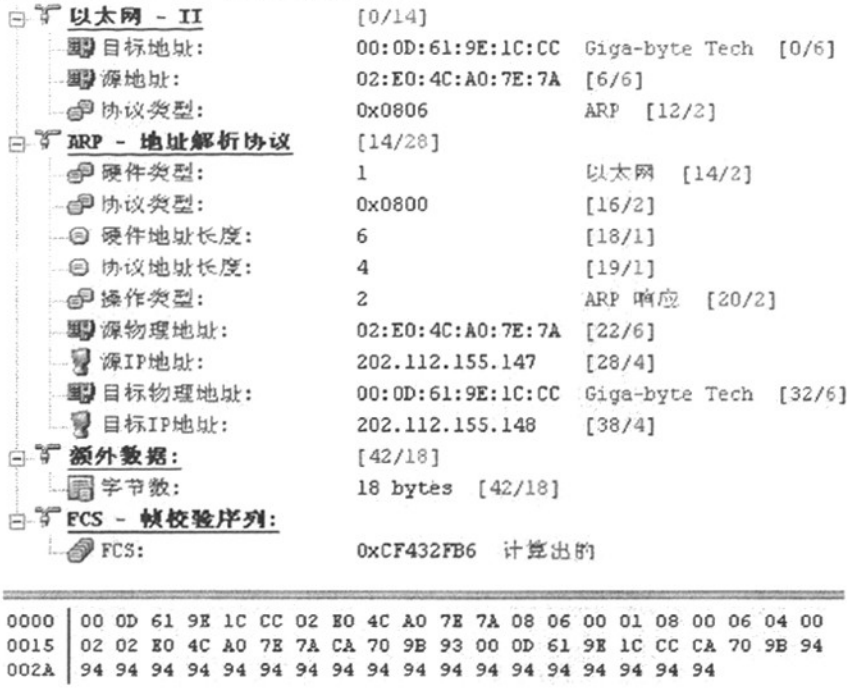


图 4-13 来自网络节点的 ARP 响应帧解析图

Figure4-13 Analysis of Answering ARP from Network Node

4.2.4 IP 数据报的填充

与 ARP 相同，IP 数据报在数据字段小于所规定的要发送的数据帧长度时，同样需要填充相应的数据字段以使之完整，具体填充多少字节长度的数据字段，由设置的发送数据包长度决定。

本部分以 UDP 为例，通过三个实验来实现 IP 数据报的填充：其一为网络节点规定要发送的 UDP 长度为 64 字节时，发送 8 字节数据；其二为网络节点规定要发送的 UDP 长度大于 64 字节时，发送 8 字节数据；其三为网络节点规定的要发送的 UDP 数据长度大于 64 字节，且要发送的数据字节长度也大于 64 字节，但是该发送长度小于规定的 UDP 发送长度，在此实验中，规定的 UDP 数据长度为 400 字节，实际发送的数据长度为 200 字节。

Name	Value	Type	Radix
TxEthnetFrameBu...	0x003F900C	unsigned int...	hex
[0]	0x0D00	Uint16	hex
[1]	0x9E61	Uint16	hex
[2]	0xCC1C	Uint16	hex
[3]	0xE002	Uint16	hex
[4]	0xA04C	Uint16	hex
[5]	0x7A7E	Uint16	hex
[6]	0x0008	Uint16	hex
[7]	0x0045	Uint16	hex
[8]	0x3200	Uint16	hex
[9]	0x0000	Uint16	hex
[10]	0x0000	Uint16	hex
[11]	0x1180	Uint16	hex
[12]	0xB26E	Uint16	hex
[13]	0x70CA	Uint16	hex
[14]	0x939B	Uint16	hex
[15]	0x70CA	Uint16	hex
[16]	0x949B	Uint16	hex
[17]	0x0104	Uint16	hex
[18]	0x8813	Uint16	hex
[19]	0x1E00	Uint16	hex
[20]	0xCAC0	Uint16	hex
[21]	0x5555	Uint16	hex
[22]	0x5556	Uint16	hex
[23]	0x5557	Uint16	hex
[24]	0x5558	Uint16	hex
[25]	0x0000	Uint16	hex
[26]	0x0000	Uint16	hex
[27]	0x0000	Uint16	hex

图 4-14 CCS 3.1 发送缓存中要发送的数据

Figure4-14 Data Prepared to Send in Buffer of CCS3.1

1) 要发送的 UDP 长度为 64 字节，发送 8 字节数据

本填充实验定义了 64 字节长度的数据包的发送，但是其中仅仅给出 8 字节的数据字段，因此需要填充后面的数据字段到 64 字节，发送中该填充字段用 0 补足。

图 4-14 为 DSP 软件 CCS 3.1 发送缓存中显示的要发送的数据，其显示为每字数据需要前后 8 位进行交换（见 4.4.5 节），0——20 字为 UDP 首部，21——24 字数据为要发送的 8 字节数据，分别为 55 55 56 55 57 55 58 55（CCS3.1 的发送中，一个字内的两字节是需要交换排列），其余 25 字以后为填充数据，都为 0。

如图 4-15 为网络线路上发送 8 字节的 UDP 数据包时，科来网络软件监控到的数据报的解析图，图的右半部分为数据报的所有数据内容。发送的 8 字节数据分别为 55 55 56 55 57 55 58 55，其后的 10 个字节全部填充为数据 0。

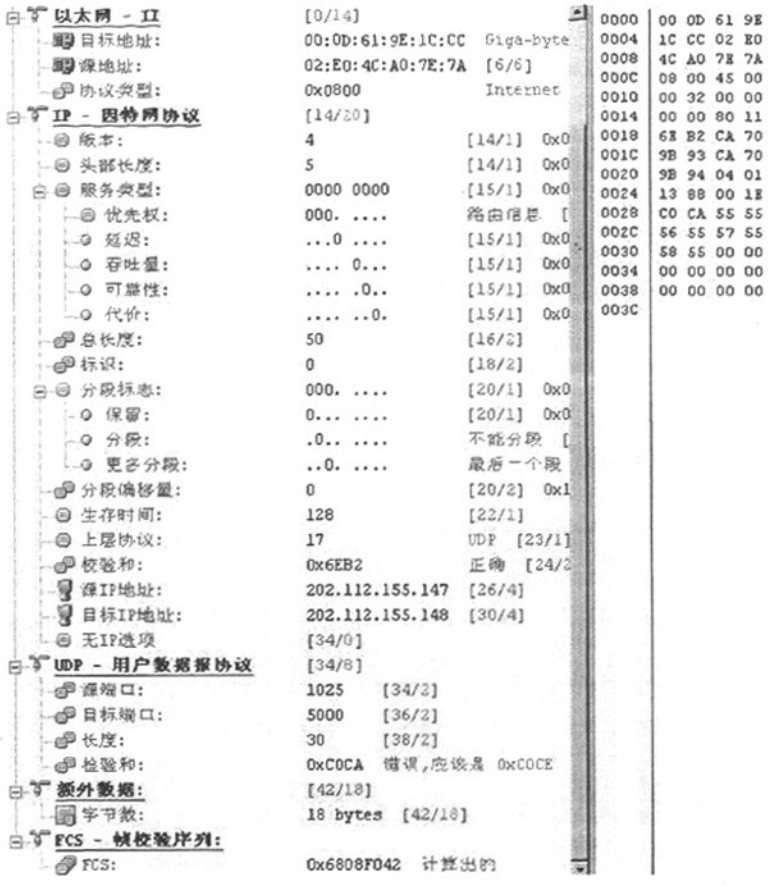


图 4-15 帧长度 64 字节时网络线路上发送 8 字节数据的 UDP 的解析图

Figure4-15 Analysis of UDP with 8 Bytes Data Sent And 64 Bytes Length of Frame in Network

2) 要发送的 UDP 长度大于 64 字节，发送 8 字节数据

在网络节点对数据的发送时，需要先在程序中规定 UDP 数据帧的长度，一般都取在 64 字节——1518 字节之间。本实验旨在验证在网络节点定义的数据报发送长度大于 64 字节时，实际有效数据字段小于 64 字节的 UDP 数据报在发送过程中是否依然要符合程序中定义的数据字段的长度，将 UDP 数据报填充至定义的发送长度，而不是仅仅填充至 64 字节。

本填充实验定义了 400 字节长度的数据报的发送，但是其中仅仅给出 8 字节的数据字段，因此需要填充后面的数据字段至少到 64 字节，发送中该填充字段用 0 补足。

由于定义的要发送的数据为 400 字节，因此发送缓存设置较大，但其有效部分显示与前一实验相同，即都为 0——20 字为 UDP 首部，21——24 字数据为要发送的 8 字节数据，与前实验相同，为 55 55 56 55 57 55 58 55，其余 25 字以后为填

充数据，都为 0。由于缓存数据部分内容相似，因此 DSP 软件 CCS 3.1 发送缓存中要发送数据结构可参照前一实验图 5-13，本处将不再给出。

如图 4-16 为总长度大于 64 字节（400 字节）时网络线路上发送 8 字节的 UDP 数据报的解析图，其中右半部分为数据报的所有数据内容，而图 4-17 为 PC 机上的接收界面。由图可以看出，在接收到发送的 8 字节数据后，其余所有字节全部填充为数据 0，并且填充超过 64 字节直到达到节点发送时所定义的 UDP 数据报字节长度 400。

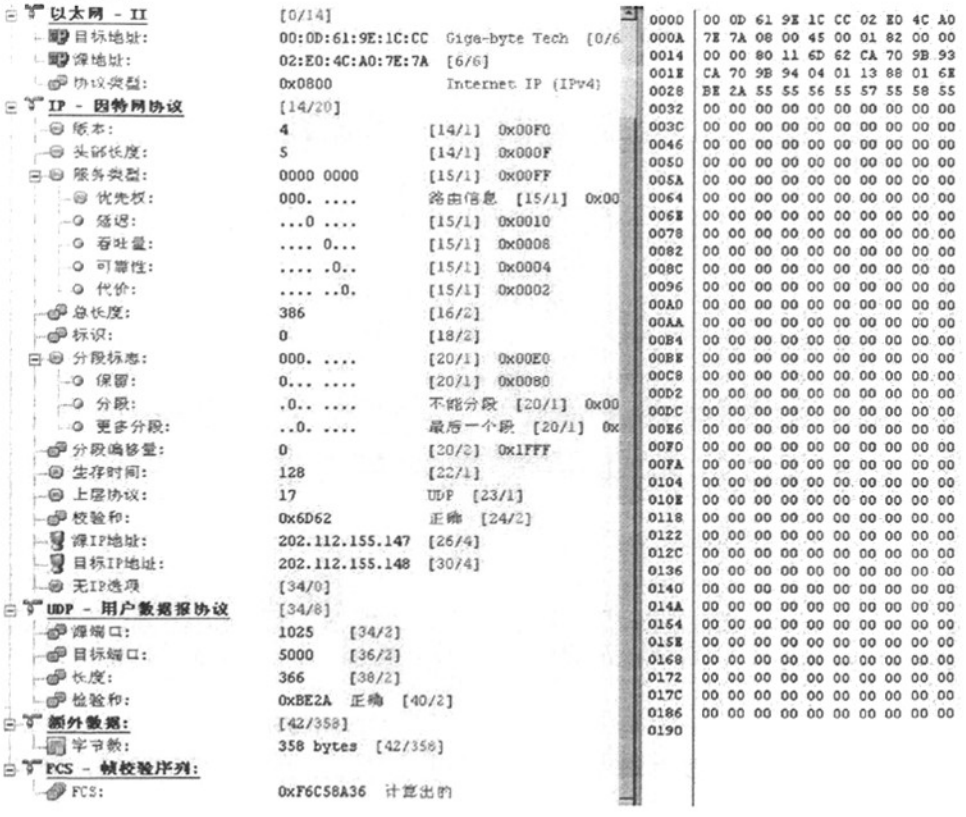


图 4-16 帧长度 400 字节时网络线路上发送 8 字节数据的 UDP 的解析图

Figure4-16 Analysis of UDP with 8 Bytes Data Sent and 400 Bytes Length of Frame in Network

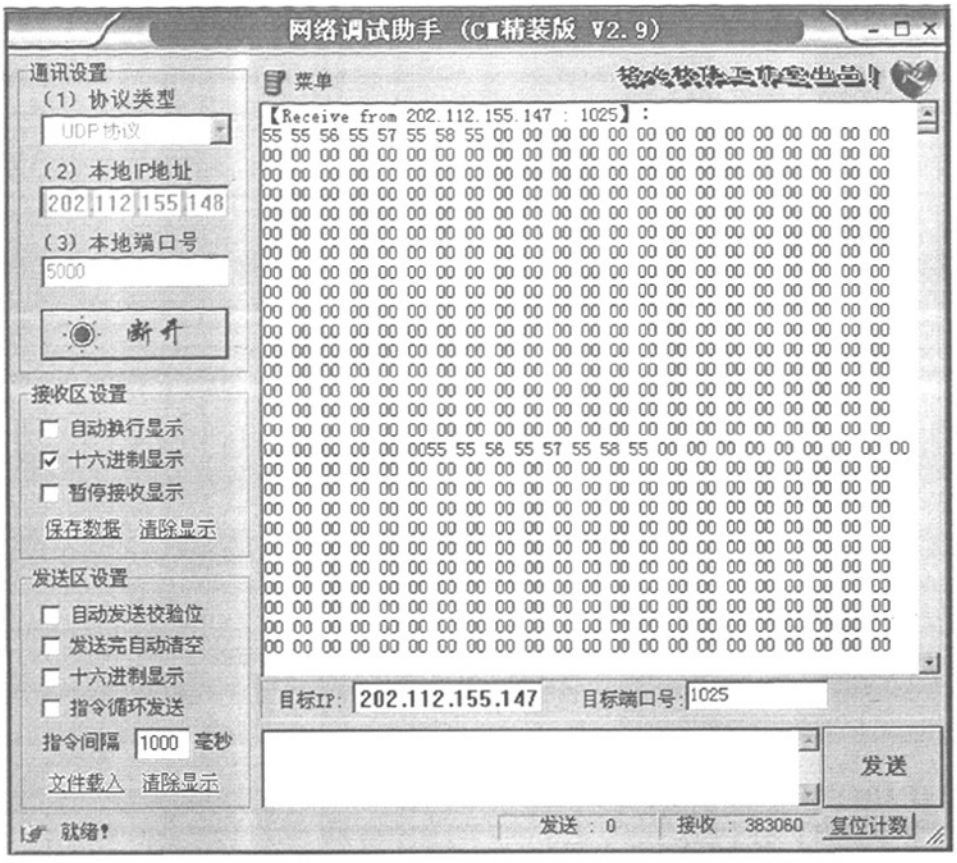


图 4-17 总长度 400 字节时网络上发送 8 字节的 PC 机上的 UDP 数据接收界面

Figure4-17 Data Interface of UDP Received by PC with 8 Bytes Data Sent and 400 Bytes Length of Frame in Network

3) 要发送的 UDP 长度大于 64 字节，发送大于 64 字节但小于要发送的 UDP 长度的数据

在本实验中，网络节点定义要发送的 UDP 数据报的长度为 400 字节，实际发送的数据长度为 200 字节。本实验旨在验证在要发送的数据报长度和实际发送的数据报长度都在 64——1518 范围之内时，如果实际发送的数据报长度小于要发送的数据报长度，是否还能够对 UDP 数据报进行填充，并达到所定义的要发送的 UDP 长度。

如图 4-18 为总长度大于 64 字节（400 字节）时网络线路上发送 200 字节的 UDP 数据报的解析图，其中右半部分为数据报的所有数据内容，而图 4-19 为 PC 机上的数据接收界面。由图可以看出，在接收到发送的 200 字节数据之后，其余所有字节全部都填充为数据 0，直到填充至节点发送时定义的 UDP 数据报字节数 400，这证明了定义的数据报和实际数据报长度都大于 64 字节时的数据报填充依

然需要填充至定义的发送字节，和前面情况结果相似。

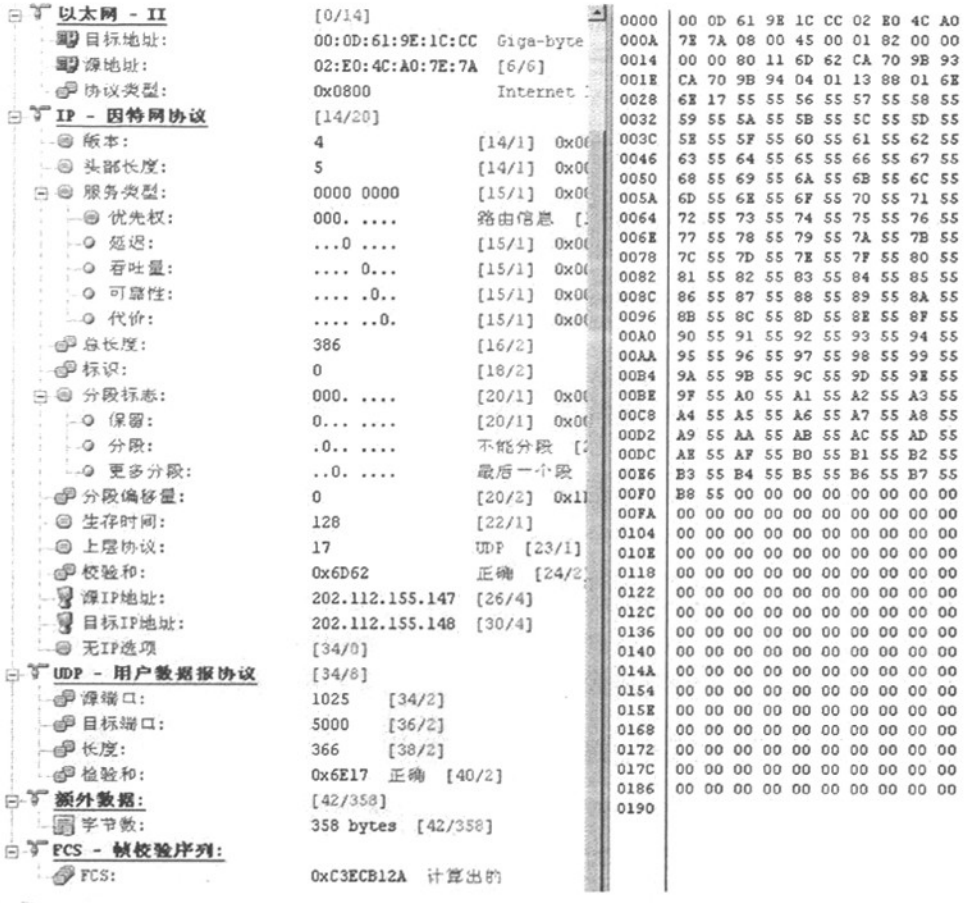


图 4-18 总长度 400 字节时网络线路上发送 200 字节数据的 UDP 的解析图

Figure4-18 Analysis of UDP with 200 Bytes Data Sent and 400 Bytes Length of Frame in Network

4.2.5 IP 数据报的填充结论

从本部分三个实验可以得出关于 IP 数据报的填充结论：

在实际发送的数据字段长度（此处仅指小于 1472 字节，不含分片功能）小于节点定义的要发送的数据字段长度（定义在数据报首部中）的情况下，不论节点实际要发送的有效数据字段长度是大于等于或小于 18 字节，也不论节点定义的要发送的数据字段的长度是等于或者大于 18 字节，在数据的传输过程中，都要自动进行填充，且填充后的数据字段长度等于节点定义的要发送的数据字段的长度，即肯定大于或等于 18 字节。

图 4-19 总长度 400 字节时网络上发送 200 字节的 UDP 的接收界面

Figure4-19 Data Interface of UDP Received by PC with 200 Bytes Data Sent and 400 Bytes Length of Frame in Network

4.3 IP 数据报的分片

在 IP 数据报的传输过程中，如果数据字段长度小于 18 字节，MAC 子层就会在数据字段的后面加入一个整数字节对它进行填充，以保证以太网的 MAC 帧的长度不小于 64 字节。而如果数据字段的长度大于 1472 字节，则必须将过长的数据报进行分片后才能传输，本节将通过相关的实验来验证 UDP 数据字段的分片并证明本功能设计的正确性。

4.3.1 数据报的分片

前面已经提到了帧的数据部分的上限----最大传送单元 MTU，当一个 IP 数据

报封装成链路层的帧时，此数据报的总长度（即首部加上数据部分）必须在 MTU 范围之内。如果数据报的长度超过网络所容许的最大传送单元 MTU 时，就必须将过长的数据报进行分片后才能在网络上传送。这时，数据报的“总长度”字段不是指未分片前的数据报长度，而是指分片后的每片首部长度与数据长度的总和。

一个数据报可以通过多个网络。每一个主机或者路由器都要把收到的帧进行拆装，再进行处理，然后又封装成另一个帧。由于每个网络所使用的协议不同，因此该网络发送出去的帧的格式与长度也不完全相同，不同的网络对分片的要求都不一样。当数据报被分片时，每一个数据报片都有它自己的首部，其中大部分的字段都是重复的，但是有些是变化的。如果已经分片的数据报遇到了更小 MTU 网络，那么这些已经分片的数据报还可以再进行分片。总之，数据在到达终点之前可以经过多次的分片。

一个数据报可以被源主机或者在其路径上的任何路由器进行分片，但是重装只能在目的主机上进行，因为每个分片都是独立的数据报。由于被分片的数据报可以是不同的路由，因此我们永远无法控制或者保证被分片的数据报应当走哪一条路径，而属于同一个数据报的所有数据报片应当最终到达目的主机，因此逻辑上，重装应当在最后终点。

4.3.2 IP 数据报分片的相关字段

当 IP 数据报被分片后，首部中的某些部分需要被复制到所有的分片中。与数据报的分片和重装有关的三个字段的值分别为标识(identification)、标志(flag)和片偏移，除了这三个数据字段，其余各字段必须被复制。但是不论是否分片，检验和的值必须要重新计算。

(1) 标识

这是个从源主机发出的 16 位字段，它是一个计数器，用来产生数据报的标识。但这里的“标识”并没有序号的意思，因为 IP 是无连接的服务，数据报不存在按序接收的问题。当 IP 发送数据报时，它就把这个数据报的当前值复制到标识字段中。当数据报由于长度超过网络的 MTU 而必须分片时，这个标识字段的值使分片后的各数据报片最后能正确的重装成为原来的数据报。

(2) 标志

占 3 位的数据字段，但目前只有两位有意义，第一位为保留为以后用。

标志字段中的最低位记为 MF(More Fragment)。MF=1 即表示后面还有“还有分片”的数据报。MF=0 标识这已是若干数据报片中的最后一个。

标志字段中间的一位记为 DF(Don't Fragment),意思是“不能分片”。只有当

DF=0 时才允许分片。

(3) 片偏移

占 13 位的数据字段。片偏移指出，较长的分组在分片后，某片在原分组中的相对位置。也就是说，相对于用户数据字段的起点，该片从何处开始。片偏移以 8 个字节为偏移单位，这就是说，每个分片的长度一定是 8 字节（64 位）的整数倍。

4.3.3 IP 数据报分片的实现

在本设计中，设计了相应的实验来实现源主机对 IP 数据报的分片，主要通过对各报片的编程来实现，其中的关键是对以上三个和分片有关的数据字段进行设置。在本实验中，以 UDP 为例，，要发送的数据字段长度为 704 字节，网络源节点规定的要发送的每一个 UDP 数据帧的数据字段的长度为 352 字节，因此，要发送的数据字段需要分成两片进行发送，每片数据字段分别为 352 字节。

在分片的实现中，需要设置 IP 数据报的相应字段：标识字段通过对每次要发送的数据进行计数来实现；对于第一个分片数据报，标志字段中的最后一位 MF 需要设置为 1，因为后面还有分片，而对于后面没有分片的第二个数据报，MF 则为 0；标志字段中间的一位 DF 设置为 0，标识允许分片；对于片偏移，第一个字段设置为 0，因为它的数据字段相对于用户数据报的起点是 0，而其后的的数据字段相对于用户数据报的起点则为之前所有分片的数据字段的字节数除以 8。

如图 4-20 和 4-21 分别为任意两个分片 UDP 数据报的结构解析图，由于他们为同一个数据报的两个分片，因此标识相同，都为 2857。第一个分片中分段标志的更多分段的值为 1 证明其后还有分片，而第二个分片中该字段为 0 则说明其后没有分片，图中分段偏移量即为片偏移。在第一个分片中，分段偏移量值为 0，说明该分片是第一个分片，而第二个字段中，该分片值为 44，即该分片部分的分段数据之前的字节数为 44 乘以 8 字节，即为 352 字节，该值也正好是前一分片的数据字段的字节数。因此，通过对源节点编程可以实现本数据报的分片功能，而通过实验证明本设计是完全正确。

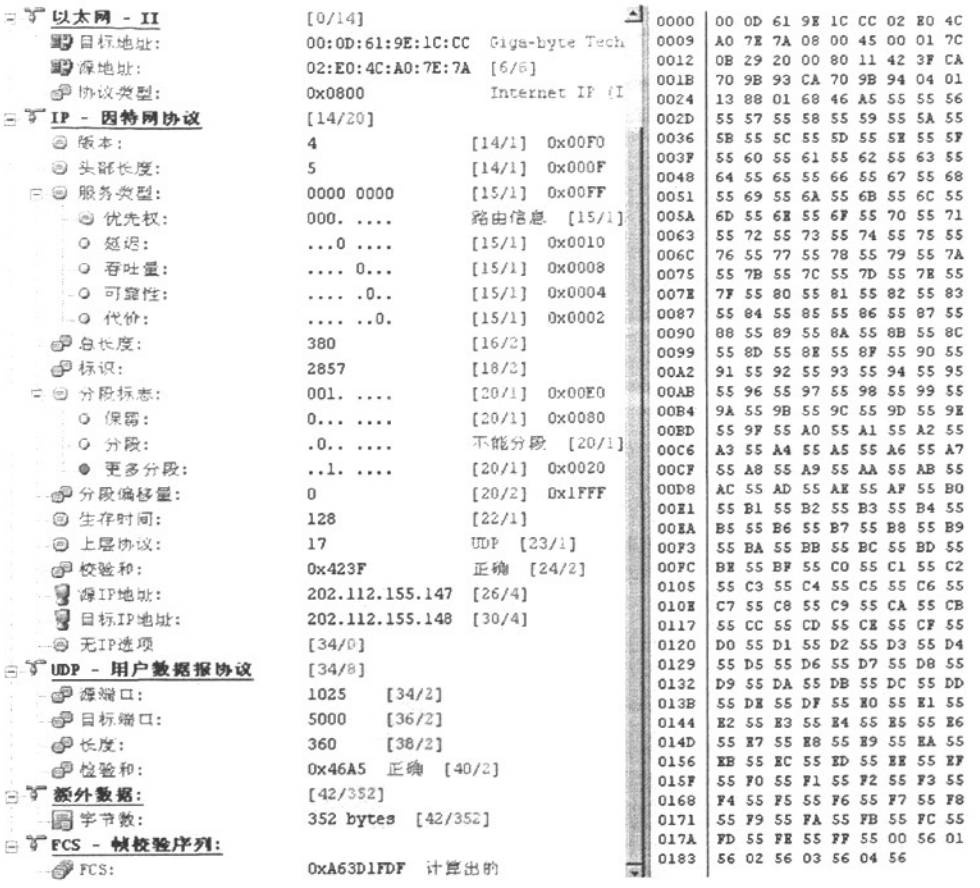


图 4-20 第一个分片 UDP 解析图

Figure4-20 Analysis of the First UDP Divided

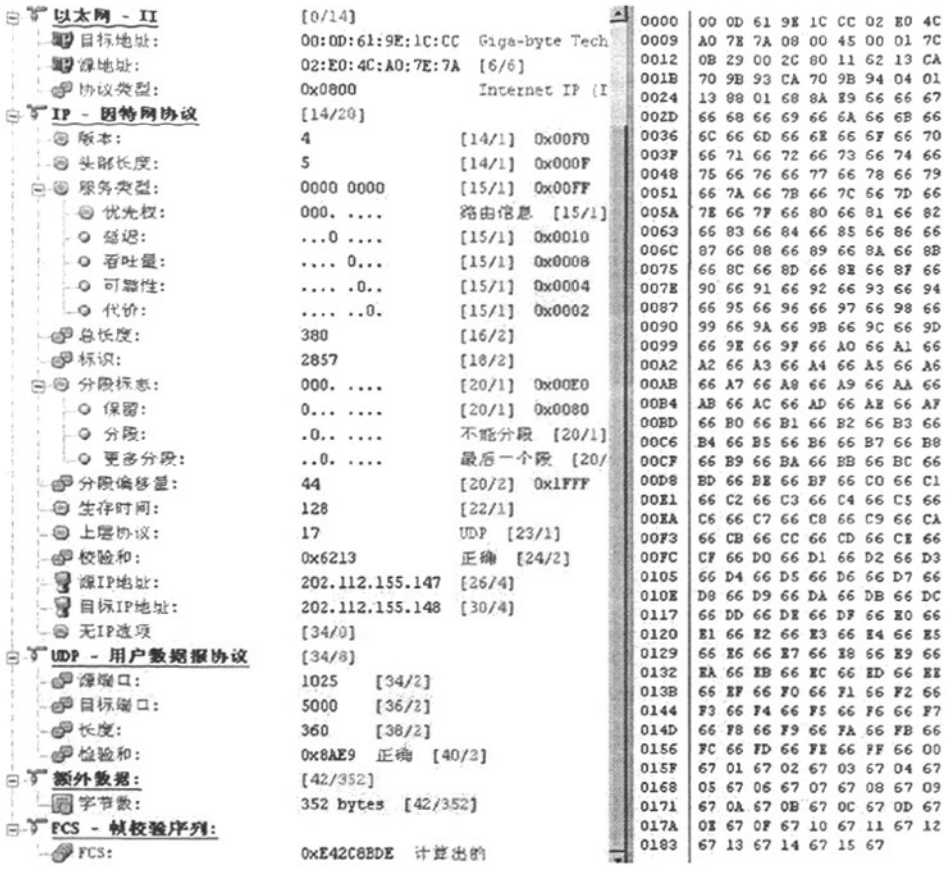


图 4-21 第二个分片 UDP 解析图

Figure4-21 Analysis of the Second UDP Divided

4.4 UDP 数据传输的可行性实验

针对 UDP 的特点，本系统设计了 UDP 与 ARP 相结合的完整的实验过程，通过编程实现了 UDP 数据帧的传输，并包含了部分 ICMP 的发送应答功能。

在实验时，PC 首先机向网络复合节点发送 UDP 帧，同时启动节点程序。由于在首次发送数据帧时，PC 机并不知道网络节点模块的 MAC 地址，因此会先自动向网络节点发送 ARP 询问帧进行询问，而此时对应 IP 的节点将发送 ARP 响应帧到 PC 机。PC 机在收到来自该网络节点的 ARP 响应帧后，就在其高速缓存中记入节点 IP 地址和硬件地址的映射，此后 PC 机将连续发送 UDP 帧，而不再出现 ARP 帧。

图 4-22 为网络节点的数据处理流程，网络节点首先通过查询相应寄存器的位来判断是否有帧需要被接收，而对于进入 CS8900A 前端的数据帧，需要先对协议

类型进行判断，协议类型错误的帧将被丢弃，如果是 ARP 帧，则进入 ARP 处理，确定是否需要回发 ARP 响应帧，如果是 IP 数据报，则需要在 IP 层之上进一步判断 IP 类型字段为 ICMP 还是 UDP（此处暂时不涉及 TCP 帧），如果是 ICMP 帧，则需要判断为何种报文类型，然后决定是否发送 ICMP 应答，如果是 UDP 帧，则可以直接读取，并对数据进行处理。

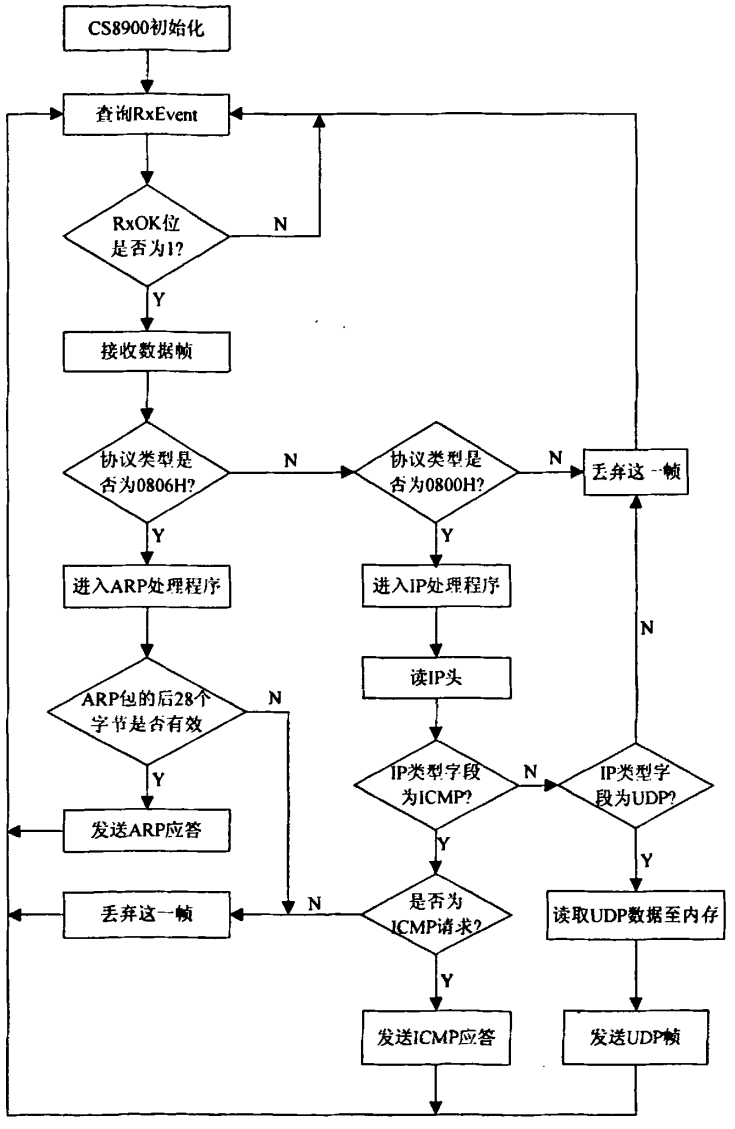


图 4-22 系统的整体数据处理流程图

Figure4-22 Flow Chat of Data Process in System

图 4-23 为科来网络软件所监控到的数据帧的界面截图，在这个监控界面中可以看到，PC 机连续广播发送 ARP 询问帧，在得到来自节点的一个 ARP 响应帧后，

停止发送 ARP 帧，然后开始连续发送 UDP 数据帧。

图 4-24 为来自该监控界面的某一 UDP 数据包的结构解析图，该图清晰地显示出了 UDP 数据帧的各分层结构以及各层数据字段的内容：分别为 MAC 层（以太网层）、IP 层（因特网协议）、UDP 层（用户数据报协议）以及所要发送的数据（额外数据）。FCS（帧校验序列）是由 CS8900A 所产生的 CRC 校验，并不是数据帧的一部分，可以通过对 CS8900A 的相应寄存器的设置而屏蔽该字段。在图 4-23 中，上层协议值为 17 表示该数据帧为 UDP，而 IP 层和 UDP 层中的校验和都显示正确，证明该 UDP 数据帧的打包和发送是正确的，其中，源端口和目的端口分别为 6000 和 1025。

以太网 MAC 帧的最大长度为 1518 字节，除去 MAC 层、IP 层和 UDP 层数据字段首部，实际传输的数据最大长度为 1472 字节。在本设计中选取最大长度 1472 字节数据（图 8 中额外数据字节）进行 UDP 传输，实验结果证明对最大数据字节的传输完全可行，因此，从对大量数据的传输需求来说，UDP 完全能够应用于交通状态获取的数据网络。

对于从复合节点向 PC 机进行 UDP 传输的实验，实现简单，由于此前在 4.1.1 节已作介绍，此处不再赘述。

图 4-23 网络监控软件所监控到的数据帧的界面截图

Figure4-23 Interface of Data Monitored by Network Monitor

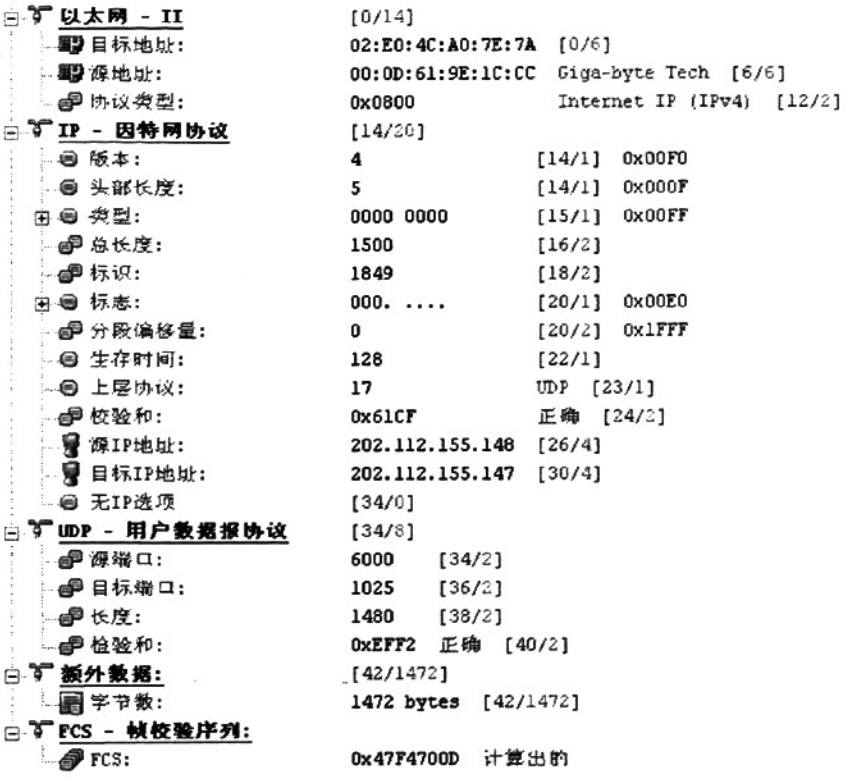


图 4-24 来自监控界面的某一 UDP 帧结构解析图

Figure4-24 Analysis of UDP from Monitor Interface

4.5 TCP 通信协议的实现

TCP 是面向连接的网络运输层协议，由于有复杂的线路连接的建立和释放过程，和 UDP 相比，它的实现较为复杂，但是可靠性高，因此针对少量较为重要的交通状态数据，采用 TCP 来实现少量重要数据的可靠传输。

单个的 TCP 数据包在传输过程中所需要的网络底层驱动，以及数据包的封装过程都和 UDP 数据包的传输原理相似，因此此处将不再赘述，此处只针对 TCP 所特有的连接和释放过程以及状态机的转换进行描述，并通过相应的实验来验证关于 TCP 功能设计的正确性。

4.5.1 TCP 协议的简化原则

在交通状态获取的嵌入式网络中，TCP 所具有的编号系统，差错控制等特点能够保证来自道路交通现场的传感器数据的可靠上传[36][37]。但如果将复杂的 TCP

协议直接移植到嵌入式系统中，势必会提高系统开销，增大延迟时间，因此，这就需要对现有的 TCP 进行简化，主要有以下两个方面：

(1) 在系统拓扑结构较为简单的网络中，网络出错和发生拥塞的可能性较低，使用 TCP 选项中选择确认（SACK）来选择性重传丢失的 TCP 报文段，经济性并不高。而 10M 网络无需考虑防止信号绕回 PAWS（Protection Against Wrapped Sequence numbers）问题，所以在 TCP 报头中使用选项的意义不大，并且过长的 TCP 报头还会降低数据传输的效率。所以忽略紧急指针和选项及填充字段的值，首部长度选定为 20 字节。

(2) 因为本系统下层使用的是通信实时性好，传输速度较快，数据量小的 CAN 总线网络，而 10Mbps 的以太网传输一般不会发生阻塞，这将会使得以太网的上位机有足够的反应时间来及时处理通信数据。所以在简化系统中，可以固定超时与重传的时间，而无需在程序中添加算法和在 DSP 中增加定时器来计算重传超时 RTO(Retransmission Time-Out)值，从而提高 DSP 的运行效率。

4.5.2 简化的 TCP 状态机

在 TCP 的实现过程中，为了清楚的掌握在连接建立、连接终止以及数据传输时所发生的所有不同事件，TCP 是以有限状态机（Finite State Machine，简写为 FSM）的形式来实现的。TCP 把连接可能处于的状态及各状态可能发生的变迁，画成图 4-25 所示的有限状态机，在图中每个方框是 TCP 可能具有的状态，方框中的字是 TCP 标准使用的状态名。一个 TCP 连接有两个端点，在 TCP 的有限状态机中，同时表示这两个端点的状态[10][11][12]。

有限状态机是指输出取决于过去输入部分和当前输入部分的时序逻辑电路。一般来说，除了输入部分和输出部分外，有限状态机还含有一组具有“记忆”功能的寄存器，这些寄存器的功能是记忆有限状态机的内部状态，它们常被称为状态寄存器。在有限状态机中，状态寄存器的下一个状态不仅与输入信号有关，而且还与该寄存器的当前状态有关，因此，有限状态机又可以认为是组合逻辑和寄存器逻辑的一种组合。其中，寄存器逻辑的功能是存储有限状态机的内部状态；而组合逻辑又可以分为次态逻辑和输出逻辑两部分，次态逻辑的功能是确定有限状态机的下一个状态，输出逻辑的功能是确定有限状态机的输出。

在 TCP 的实现中，有限状态机通常包含客户端和服务器端。在本网络系统中，上位机只作为数据的接收方来接收来自复合节点的传感器数据，即 TCP 的每次连接都由复合节点主动发起，它在 TCP 的连接过程中充当客户，所以在本系统设计中，简化的 TCP 有限状态机只涉及到完全 TCP 状态机中的 6 种状态，它们分别为：

TCP_CLOSED、TCP_SYN_SENT、TCP_ESTABILSHED、TCP_FIN_WAIT1、TCP_FIN_WAIT2、TCP_TIME_WAIT。各状态如表 4-1 所示，简化的有限状态机如图 4-25 所示。

表 4-1 复合节点 TCP 的各种状态

Table4-1 TCP State of Compound Node

状态	说明
TCP_CLOSED	没有连接
TCP_SYN_SENT	已发送 SYN，等待 ACK
TCP_ESTABILSHED	连接已建立
TCP_FIN_WAIT1	第一个 FIN 已发送，等待 ACK
TCP_FIN_WAIT2	等待第二个 FIN
TCP_TIME_WAIT	收到第一个 FIN，已发送

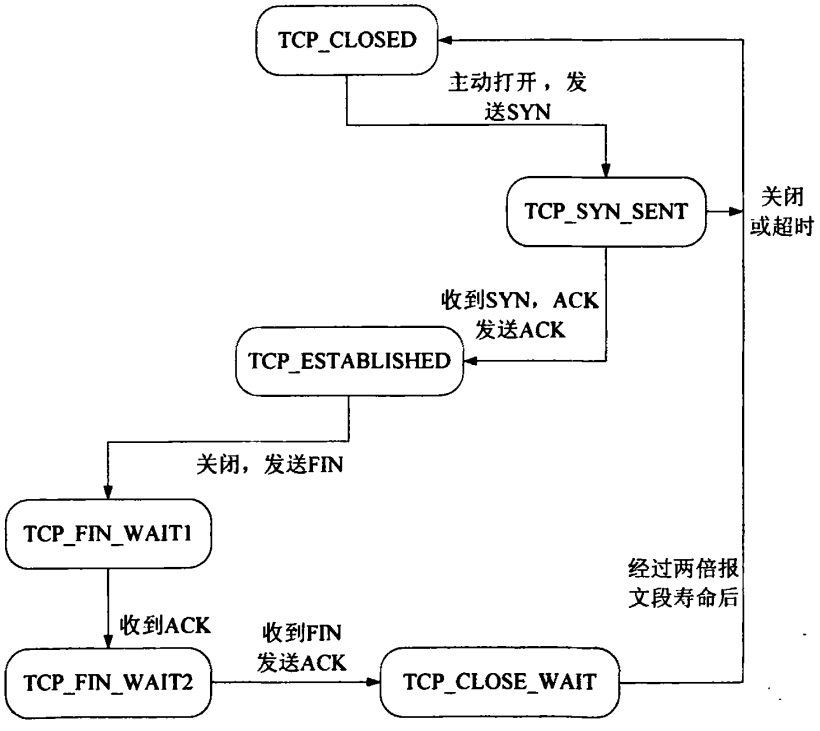


图 4-25 复合节点的 TCP 有限状态机

Figure4-25 FSM of Compound Node

为了平滑的完成 TCP 操作，使用定时器来实现 TCP 的连接和数据的可靠传输。为了防止在通信中出现由于网络的原因而导致的 TCP 报文丢失，需要设置重传计时器。当计时器截止时间到之前收到了对特定报文段的确认，则撤销这个计时器，

否则就重传该报文段，并将计时器复位。同时为了防止因为丢失最后一个 ACK（应答）报文段而导致服务器无法正常关闭的问题，设置时间等待计时器。这里定时器采用的是 DSP 的 32 位的 CPU 定时器 0。

4.5.3 TCP 连接的建立和释放

TCP 连接的建立和释放是 TCP 协议的重点，也是它和 UDP 的区别之处，只有建立连接，TCP 才能保证报文的可靠传输，而只有释放使用过的连接，TCP 连接才能结束一次数据通信，并释放信道[10][11][12]。

1) TCP 连接的建立

如图 4-26 为本网络系统设计中，TCP 连接的建立过程，上位机只作为数据的接收方接收来自复合节点的传感器数据，因为 TCP 的每次连接都由复合节点主动发起。

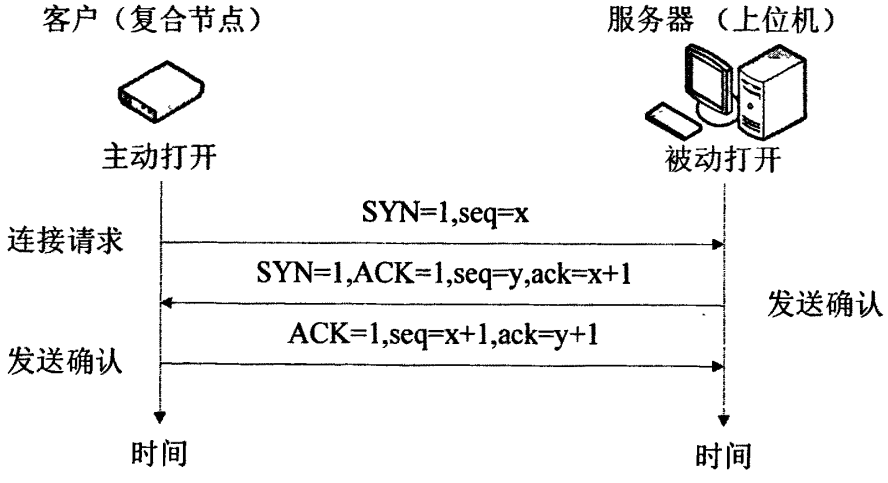


图 4-26 TCP 连接的建立过程

Figure4-26 Establishment of TCP Connection

TCP 连接步骤如下：

- 复合节点首先向上位机发出 TCP 连接请求报文段，其首部中的同步比特 SYN 应置为 1，并选择序号为 x，表明传送数据时的第一个数据字节的序号是 x。
- 上位机收到 TCP 连接请求报文段后，如同意，则发回确认。上位机在确认报文段中应将 SYN 置为 1，其确认号应为 x+1，同时也为自己选择序号 y。
- 复合节点收到此报文段后，向上位机给出确认，其确认号应为 y+1。

- 复合节点的 TCP 通知上层应用进程，连接已经建立。
- 当上位机收到复合节点的 TCP 确认后，也通知其上层应用进程，连接已经建立。

2) TCP 连接的释放

在数据传输结束后，通信的双方都可以发出释放连接的请求，如图 4-27 所示。

在连接释放过程中，复合节点的应用进程先向上位机发送 TCP 释放请求，并且不再发送数据。复合节点的 TCP 通知上位机要释放从复合节点到上位机两个站点之间的连接，把发往主机的报文段首部 FIN 置 1，其序号为 seq=x，等于前面已传送过的最后一个字节的序号加 1。

上位机收到 TCP 释放连接后发出确认通知，即 ack=x+1，而这个报文段的序号设置为 y，上位机通知高层应用进程。这样从复合节点到上位机的连接就释放了，此后上位机不再接收来自复合节点的数据，而上位机还可以继续向复合节点发送数据，此时为半关闭(half-close)状态。

若上位机不再向主机发送数据，则其应用进程就通知 TCP 释放连接。发送报文段使 FIN=1，并使其序号仍为 y，还必须重复上次已发送过的确认号 ack=x+1，而自己的序号是 seq=x+1。这样才把上位机到复合节点的反方向连接释放掉。复合节点的 TCP 再发送应用进程报告，整个连接全部释放。

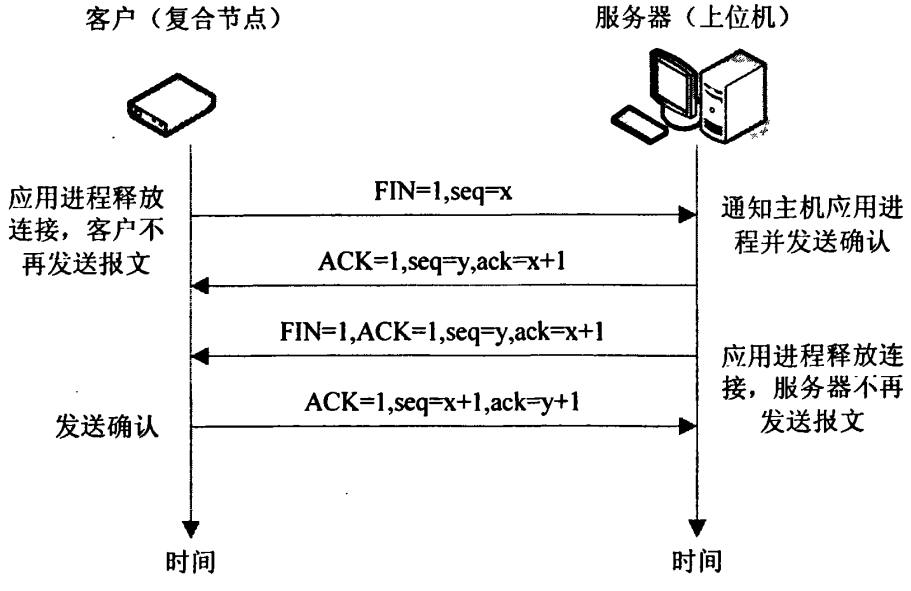


图 4-27 TCP 连接的释放过程

Figure4-27 Finish of TCP Connection

3) TCP 连接的建立和释放实验

针对在本系统中，基于简化的状态机，设计了相应的实验过程，通过对 DSP

编程实现了 TCP 连接的建立和释放过程。

图 4-28 为上位机和复合节点之间正常的 TCP 通信的截图。其中，202.112.155.148 为 PC 机的 IP 地址，202.112.155.147 为本系统设置的网络复合节点的 IP 地址，qw 为 PC 机名称，在 TCP 的相关实验中，PC 机的端口号是 1024，网络复合节点的端口号是 1028。在该图概要部分的内容中，S 代表 SYN，A 代表 ACK，P 代表 PSH，F 代表 FIN，它们是 TCP 数据报的控制位，分别表示了该 TCP 数据报的性质。

报表统计 端点 协议 数据包 连接 日志 图表

总计 71 重新 0 已捕获

源	目标	协议	大小	概要
02:E0:4C:A0:7E:7A	FF:FF:FF:FF:FF:FF	ARP	64	谁是 202.112.155.148? 告诉 202.112.155.147
Giga-byte Tech:9E:1C:CC	02:E0:4C:A0:7E:7A	ARP	64	202.112.155.149 在 00:00:61:9E:1C:CC
202.112.155.147:1024	qw:1028	TCP	66	序列号=0000000000,确认号=0000000000,状态=....S.,长度= 0,...
qw:1028	202.112.155.147:1024	TCP	64	序列号=0129344609,确认号=0000000001,状态=.A..S.,长度= 0,...
202.112.155.147:1024	qw:1028	TCP	64	序列号=0000000001,确认号=0129344610,状态=.A....,长度= 0,...
202.112.155.147:1024	qw:1028	TCP	78	序列号=0000000001,确认号=0129344610,状态=.AP...,长度= 20,...
qw:1028	202.112.155.147:1024	TCP	64	序列号=0129344610,确认号=0000000021,状态=.A....,长度= 0,...
202.112.155.147:1024	qw:1028	TCP	64	序列号=0000000001,确认号=0129344610,状态=.A...F,长度= 0,...
qw:1028	202.112.155.147:1024	TCP	64	序列号=0129344610,确认号=0000000021,状态=.A....,长度= 0,...

图 4-28 网络线路上的 TCP 数据帧

Figure4-28 TCP of Network

由图 4-28 可以看出，复合节点首先通过 ARP 解析上位机的物理地址。复合节点的 TCP 处于 TCP_CLOSED 状态，然后发送 SYN=1（图 4-28 中为 S）的 TCP 报文请求连接，状态变为 TCP_SYN_SENT。若收到上位机发送来的 SYN=1，ACK=1（图 4-28 中为 A）的 TCP 报文段，则发送 ACK=1 的 TCP 报文，连接建立，复合节点状态为 TCP_ESTABLISHED。

源IP地址:	202.112.155.147	[26/4]	
目标IP地址:	202.112.155.148	[30/4]	
无IP选项	[34/0]		
TCP - 传输控制协议	[34/20]		
源端口:	1024	[34/2]	
目标端口:	1028	[36/2]	
序列号:	0	[38/4]	
确认号:	0	[42/4]	
TCP偏移量:	7	[46/1]	0x00F0
标志:	..00 0010	[47/1]	0x003F
紧急位:	..0.	[47/1]	0x0020
确认位:	...0	[47/1]	0x0010
急迫位:	.... 0...	[47/1]	0x0008
重置位:	.... .0..	[47/1]	0x0004
同步位:	.... ..1.	[47/1]	0x0002
终止位:	0	[47/1]	0x0001

图 4-29 复合节点发送的同步位

Figure4-29 SYN from Compound Node

此后，双向的数据传输开始。复合节点发送含 PSH=1（图 4-28 中为 P）的数

据，字节长度为 20。复合节点每发送完一个 TCP 报文段，就等待上位机的确认，若超时则重发该报文段，如果重发 6 次依然收不到确认，则删除本次连接，复合节点回到 TCP_CLOSED 状态。

当收到上位机的应答时，则发送下一个 TCP 报文段。当所有的 TCP 数据发送完毕，则复合节点发送 FIN=1（图 4-28 中为 F）的 TCP 报文，并将自己的状态置为 TCP_FIN_WAIT1。上位机收到后发送 ACK=1 的 TCP 报文。在这里由于上位机还有数据要继续发送，因此并没有发送 FIN=1 的 TCP 报文，复合节点将不再发送数据，连接处于半关闭状态。

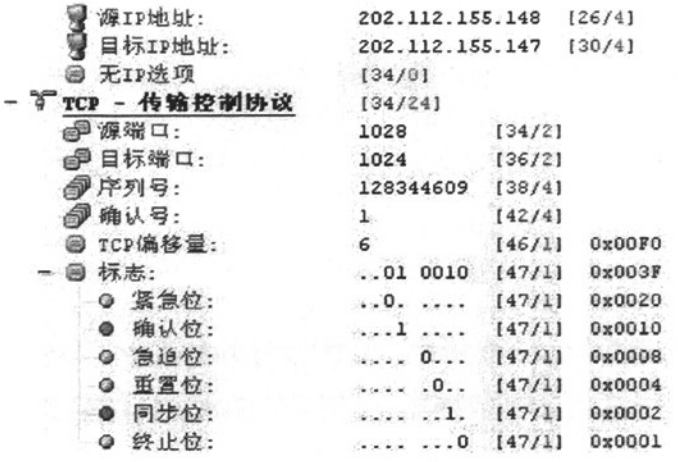


图 4-30 上位机发送的同步位

Figure4-30 SYN from PC

图 4-31 复合节点发送的确认位

Figure4-31 ACK from Compound Node

如图 4-29 到 4-35 所示分别为从科来网络监控软件监控到的从连接建立到释放，节点半关闭，各个 TCP 数据包解析图的关键部分截图，其中该次 TCP 发送帧

的控制位用绿色表示，完全符合 TCP 传输控制的需求和顺序，因此证明本实验结果完全正确。

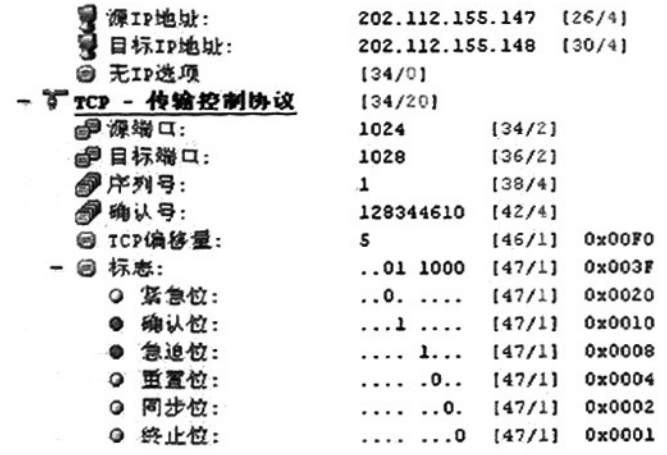


图 4-32 复合节点发送的急迫位

Figure4-32 PSH from Compound Node

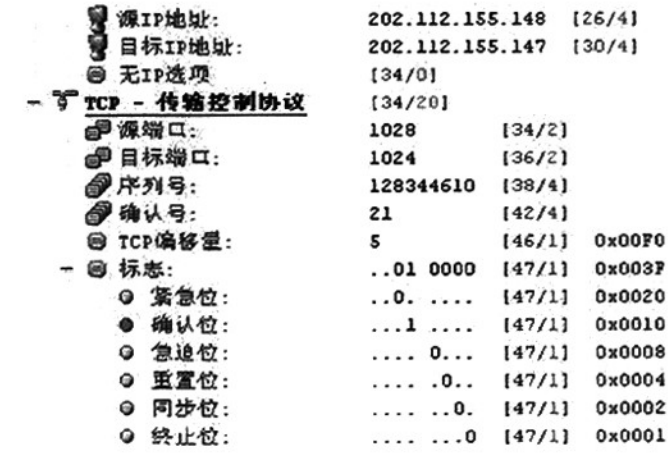


图 4-33 上位机发送认可急迫的确认位

Figure4-33 ACK from PC for PSH

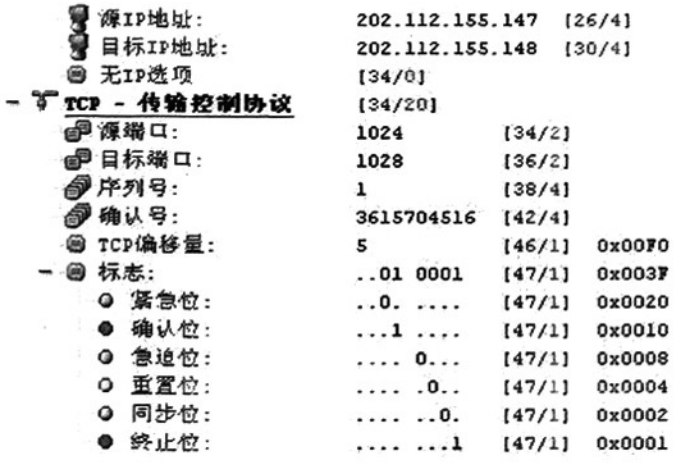


图 4-34 复合节点发送的终止位

Figure4-34 FIN from Compound Node

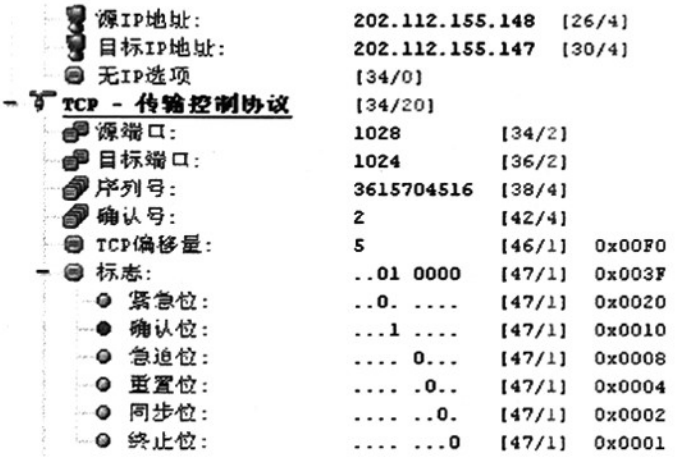


图 4-35 上位机发送同意终止的确认位

Figure4-35 SYN from PC for FIN

4) TCP 数据传输的可靠性

针对点对点数据传输的可靠性，本系统通过科来网络软件得出了 TCP 传输时的正确包与错误包的对比图，如图 4-36 所示。可以看出，在该次点对点的通信中，没有出现错误包，数据包的出错率接近于 0。

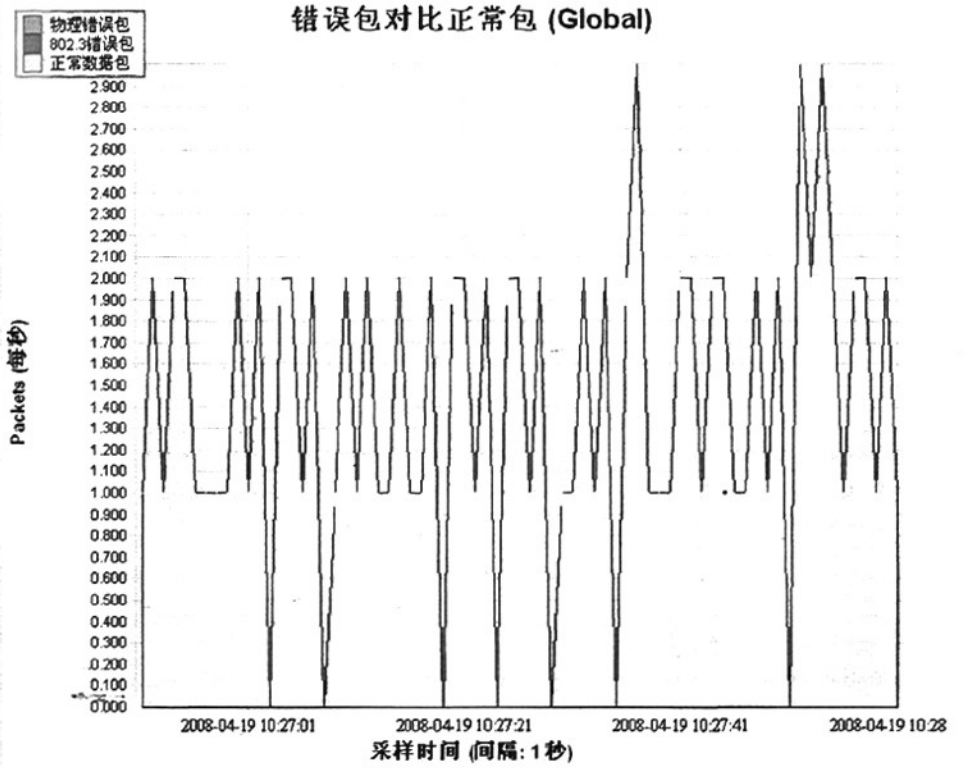


图 4-36 正确数据包与错误数据包对比

Figure4-36 Contrast between Right Data Packet and Wrong Data Packet

5) TCP 的差错控制

在本系统的状态机转换过程中，当上位机不存在相应的 TCP 服务程序时，复合节点发送 TCP 帧，将会收到一个来自上位机的包含置 1 的控制字段 RST（连接复位标志）的 TCP 报文段，这表明 TCP 连接中出现严重差错（如主机崩溃等原因），必须释放连接。于是复合节点停止 TCP 的连接，TCP 状态回到 TCP_CLOSED。

图 4-37 为 TCP 差错控制的截图，其中复合节点在发送了含 PSH 的 TCP 帧后，收到一个含 RST 的帧，则停止 TCP 的连接。图 4-38 为含有重置位的数据包的解析图，图中第二个绿色的控制位为重置位。

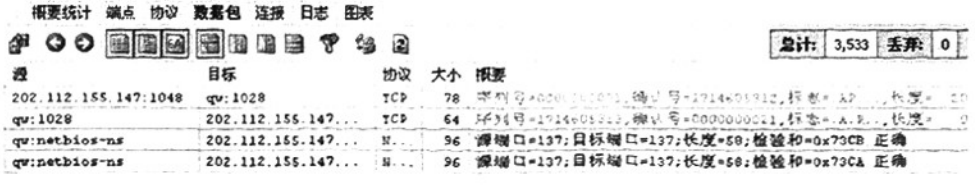


图 4-37 TCP 的差错控制

Figure4-37 Error Control of TCP

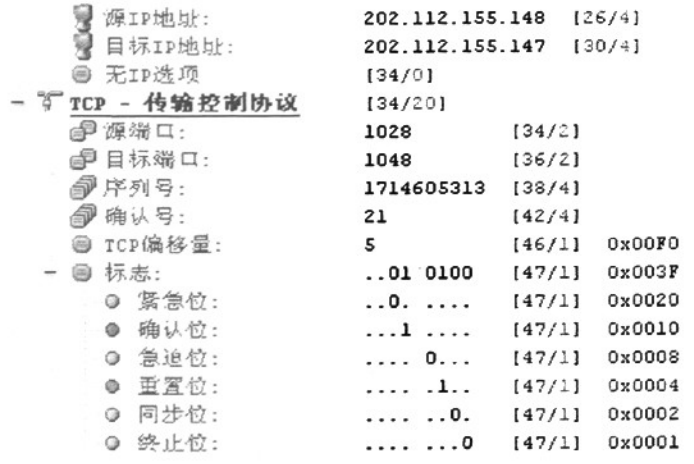


图 4-38 复合节点发送的重置位

Figure4-38 RST from Compound Node

4.6 以太网在交通状态数据传输中的实现

结合项目实际需求，本实验系统搭建了道路现场模型对实际的道路交通状态进行模拟：根据道路现场的实际情况铺设了相应的道路传感器，通过连复合节点和前端节点构建了完整的 CAN 总线和以太网的数据网络（如图 2-1），从而实现了在实验室中对道路交通状态的获取和监测的模拟。

针对本数据网络的实验需求，设计了实验流程，该流程能够完成从 CAN 总线对底层传感器信号的采集到上层以太网的数据传输过程，如图 4-39 所示为系统数据传输实验整体流程图。

在如图 4-39 所示的系统数据传输的整体网络流程中，首先初始化复合节点，然后查看是否有来自上层以太网的数据，如果有，则要先处理以太网数据，传送以太网的信息给各个 CAN 总线节点，如果没有，则用轮询的方式检查各个总线节点是否有数据，每次轮询都要对所有的前端总线节点进行检测，并分别将每个有数据的总线节点的数据进行提取并存储。在一次轮询结束后，对各个节点的数据进行处理，并通过网络协议进行上层传输。

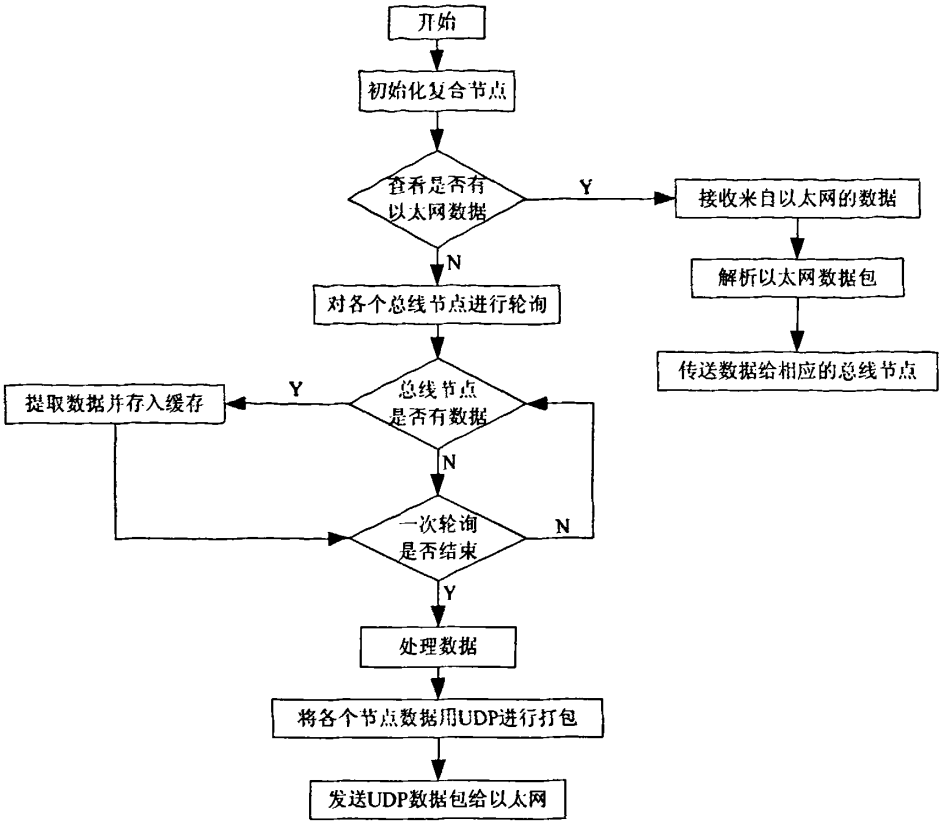


图 4-39 系统整体数据传输实验流程图

Figure4-39 Flow Chart of Data Transmission in System Experiences

参照实际道路现场各传感器及其应用，在设计中，本道路模拟系统采用了涡流传感器、光电传感器和温度传感器来对道路传感器进行模拟。复合节点在对来自各传感器的原始数据进行处理后得到如下七个交通流状态参数：节点状态、周期、流量、占有率、平均速度、地表温度和日期，根据道路交通控制理论，它们在一定程度上能够反映某段道路截面的交通状态。如图 4-40 为以太网应用层中来自 CAN 网络下层的交通流状态数据字段的具体内容及其所占的字节数。

节点号	状态	周期	流量	占有率	平均速率	地表温度	日期
2字节	2字节	2字节	2字节	2字节	2字节	2字节	4字节

图 4-40 交通流状态数据字段的分配

Figure4-40 Encapulation of Traffic States Data

在本系统中，复合节点下层的传感器信号采集及 CAN 总线网络通信部分为本项目中的另一子部分，不属于本文讨论范围，此处涉及只为介绍完整的数据网络实验，以及本以太网部分在整个网络中的应用，因此，对于前者，本文不加叙述。

DSP 复合节点通过对所有总线节点进行轮询来获取底层数据并进行相应的处理，得到 4-40 所示的表征特定交通流状态的数据字段。在对所有节点的一次轮询结束后，复合节点按总线节点序号排列处理后的数据信息并用 TCP/IP 协议进行封装打包，最后通过以太网发送到上位机进行监控。上位机中的应用软件对数据帧进行解析，并将路段上各位置的交通状态检测结果实时显示在人机界面中。

如图 4-41 所示为上位机所显示的人机界面，该界面参数是通过以太网从复合节点得到的。在该图中，有交通状态数据显示的三个绿色节点表示在一次轮询过程中该节点处于工作状态，能够获取到相应的交通状态数据，而其余无数据显示的红色节点则表示该节点没有工作，为故障状态，在该次轮询中无法获取到相应的交通状态数据。

在上层系统对下层总线网络发送控制命令时，采用中断方式，复合节点响应中断并提取数据包中的命令数据，再将其发送到相应的 CAN 总线节点。

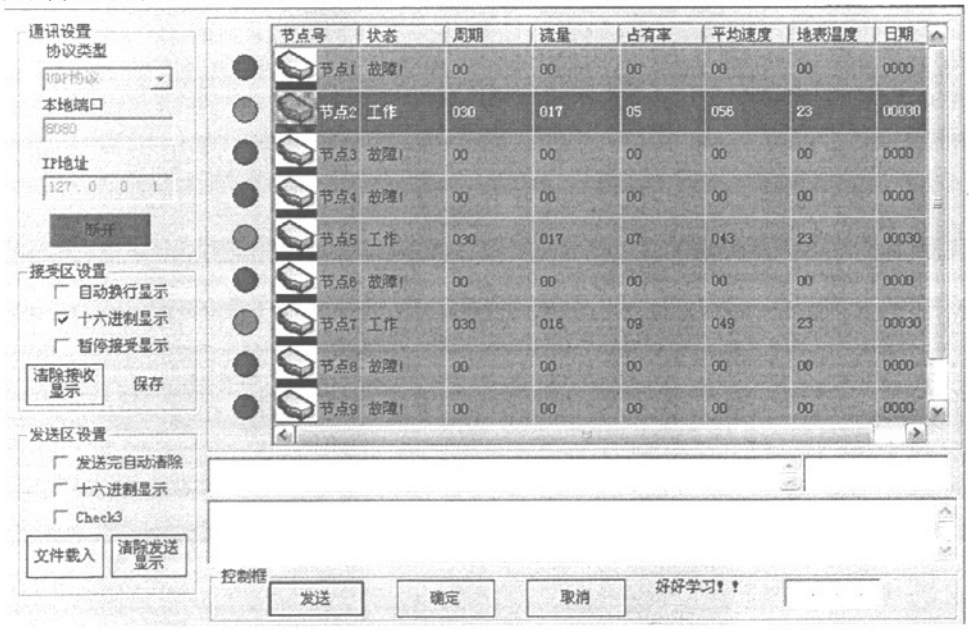


图4-41 以太网上层人机界面

Figure4-41 Human-Computer Interface of Internet

5 结论

5.1 总结

针对道路交通状态获取和传输的需求，基于“科技部国家863计划”项目“交通状态获取的新型传感器、传感器网络优化与融合”提出的分布式全时空城市道路交通状态获取系统的背景，构建了基于CAN总线和以太网的传感器数据通信网络，通过对网络复合节点功能的设计实现了对道路交通数据网络系统的构建。

根据道路交通数据网络的传输需求，在网络的上层，本文提出了一种由数字信号处理器DSP和以太网控制器CS8900A构建的以太网通信方案，介绍了硬件和软件的实现，采用了简化的嵌入式TCP/IP协议，并通过大量实验验证了本设计的合理性和可行性。

在网络上层的以太网系统，本部分主要实现了以下功能：

- 基于DSP器件 TMS320F2812和以太网控制器CS8900A，完成了以太网系统的硬件设计，并结合系统其余功能构建了网络复合节点的硬件基础，实验证明该硬件连接方案是完全正确。
- 在硬件设计的基础之上，实现了以太网的底层驱动功能，主要包括网络系统的初始化，数据的发送和接收功能。
- 在底层硬件驱动之上，首先采用了简单的无连接的UDP作为网络协议进行数据传输，针对大量数据传输的需求，它能够减少网络开销，并保证传输效率。对于少量重要交通数据的传输，采用了面向连接且可靠性强的TCP协议来实现，实验证明本系统完全能够实现相应的嵌入式TCP/IP协议，并用于交通状态数据的传输。
- 在实验室环境中搭建了模拟车道，利用现有设备和研究成果，结合网络复合节点的其他功能，以及上层人机界面软件，构建了完整的传感器数据通信网络，实现了从数据的采集、检测、处理到传输的整个过程，形成了一套较为完整的实验系统和演示模型。

5.2 展望

交通状态获取的数据通信网络是一个针对性强集成度高的数据通信系统，要完全实现该系统的功能，需要耗费更长时间，进行更多功能的开发和研究。本文

针对数据网络上层的以太网通信部分，主要实现了网络的底层驱动和基于嵌入式TCP/IP 协议的数据通信，进而完成了在该条件下对道路交通状态获取系统的模拟，以及数据网络基本功能的构建。

尽管本设计功能取得了一定的成果，但是这主要是基于实验室环境所进行的工作，后续工作尚需要在实际现场应用环境下进行。

本节点在硬件设计上完全正确，因此在本部分的进一步工作中不需要再对硬件系统进行变化，进一步深入的研究工作主要在软件的上层协议进行。

UDP 为无连接的传输服务，它在传输实时性要求高的数据报时更为方便，但是出于其不保证可靠交付，传输效率会相对较低。TCP 提供面向连接的运输层服务，它的实时性较差，但是可靠性好，因此尽管 TCP 实现复杂，且系统开销大，但是对于一些重要的交通状态参数及数据的可靠传输却有着非常重要的意义。本节点系统实现了完全的 UDP 协议，但对于 TCP，仅实现了部分基于简化了的状态机的协议内容。TCP 功能强大且实现复杂，在本文的研究基础上，它还有着更广阔的研究空间，它和 UDP 的互补作用将共同构成完整的 TCP/IP 协议。

结合道路交通控制理论，在 TCP/IP 协议基础之上，还需要借鉴美国国家运输通信协议 NTCIP，研究适应于城市道路交通系统的数据通信网络协议，实现区域交通状态获取网络的互联和优化。

鉴于时间仓促及本人技术理论水平的实际情况，在研究工作和本论文中难免存在一些疏漏和不足之处，敬请各位老师提出宝贵的指导意见。

参考文献

[1]段里仁.道路交通自动控制[M].北京：中国人民公安大学出版社，1991.6

[2]彭春华，刘建业，刘岳峰，晏磊，郑江华.车辆检测传感器综述[J].传感器与微系统，2007，26(6)：4-7，11

[3]J Ren et aLAn.Architecture for ITS Information Application Systems[C].In：IEEE ITS Conference Proceedings,2001-08：996-1001

[4]绍杰.北京市二、三环路交通流信息动态实时检测系统的功能[J].道路交通与安全，2003，(2)：2-6

[5]戴晓峰.城市道路交通状态分析方法回顾与展望[J].道路交通与安全，2008，8(3):11-15

[6]陆化普，李瑞敏，朱茵.智能交通系统概论[M].北京：中国铁道出版社，2004，10

[7]李志刚，周兴社.传感器网络[J].计算机应用研究，2004(12)：9-12

[8]庄庆德.传感器网络的研究现状[J].国外电子测量技术，2005(4)：3-7

[9]宋光明,葛运建.智能传感器网络研究与发展[J].传感技术学报，2003(2)：7-12

[10]Behrouz A.Forouzan,Sophia Chung Fegan.TCP/IP 协议族[M]（TCP/IP Suite）.北京：清华大学出版社，2006

[11](美)Kenneth D.Reed.TCP/IP 基础[M].北京：电子工业出版社，2005

[12]谢希仁.计算机网络教程[M].北京：人民邮电出版社，2006

[13]Michael A.Gallo,William M.Hancock.计算机通信和网络技术[M].北京：人民邮电出版社，2003

[14]阳宪惠.工业数据通信与控制网络[M].北京：清华大学出版社，2003

[15]Chen Gang,Ye Dong，Che Rensheng.Developing Trend of Industrial Fieldbus Control System[J].The Eighth International Conference on Electronic Measurement and Instruments，2007：765-768

[16]陈凌凌，陈以.工业以太网在工业控制网络中的应用与发展综述[J].信息科技，2007，18：147-148

[17]Imran Ahmed，Hong Wang，Vikram Kapia.Internet-Based Remote Control using a Microcontroller and Embedded Ethernet[J].Proceeding of the 2004 American Control Conference，Boston，Massachusetts，2004：1329-1334

[18]Jean Luc Scharbarg, Marc boyer, Jerome Ermont. TTCAN over mixed CAN/Switched Ethernet architecture[A].In:Emerging Technologies and Factory Automation,ETFA:2005：19-22

[19]Scharbarg Jean-Luc,Boyer Marc,Fraboul Christian.CAN-Ethernet architectures for real-time applications[A].In:10th IEEE International conference on Emerging Technologies and Factory

Automation,catania:2005：245-252

[20]苏奎峰，吕强等.TMS320F2812 原理与开发[M]，电子工业出版社，2005

[21]Texas Instrument Incorporated.TMS320C28X 系列 DSP 的 CPU 与外设[M]，北京：清华大学出版社，2005

[22]刘和平，张卫宁等编译.TMS320C28x 系列 DSP 指令和编程指南[M]，北京：清华大学出版社，2005.3

[23]彭启琮，张诗雅，常冉等编译.TI DSP 集成化开发环境（CCS）使用手册[M]，北京：清华大学出版社，2006.12

[24]Cirrus Logic Inc.CS8900A Product Data Sheet.http://cirrus.com，2001

[25]Cirrus Logic Inc.CS8900A Application Note.http://cirrus.com，2001

[26]陈鹏，张爱民，李杰.基于以太网控制器 CS8900 实现嵌入式系统的网络互连.电子工程师，2002，12(28):4-6

[27]叶崧.基于 TMS320LF2407 嵌入式硬件平台的结构及设计.金陵科技学院学报，2004，20(12):12-16

[28]杨小平，牛秦洲.嵌入式系统网络接口模块设计.桂林工业学院学报，2005，25(1):104-105

[29]刘韶华，费树岷，叶崧.基于 DSP 的嵌入式系统的网络接口设计.控制工程，2004，11(1):66-69

[30]Ri-Kun Liao, Yue-Feng Ji, Hui Li.Optimized design and implementation of TCP/IP software architecture base on embedded system [A]. In: Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian: 2006.590-594

[31]孙兵,何瑾,陈广厦.基于 DSP 的 CAN 总线与以太网互联系统研制[J].仪器仪表学报,2008,29(2):378-380

[32]Ahmed Imran, Wong Hong, kapila Vikram. Internet-based remote control using a microcontroller and an embedded Ethernet [A]. In: Proceedings of the 2004 American Control conference(AAC), Boston: 2004.1329-1334

[33]Yanjun FANG，Bo XI，Meicheng CHEN，Jingyu LIU，Implementation of Industrial Ethernet Communication Based on Embedded Systems.Industrial Electronics and Application[J]，2006 1ST IEEE Conference，2006：1-4

[34]范诚亮，基于 AT89C55WD 单片机的网络通信设计[J].现代电子技术，2006，9(224):56-57

[35]唐富年，殷少锋.一种嵌入式 TCP/IP 协议栈的设计与实现.网络通讯与安全，2007:947-949

[36]Andro Milanovic, Sinisa Srbljic,Vlado Sruk.Performance of UDP and TCP Communication on Personal Computers[A].In:10th Mediterranean Electrotechnical Conference, MEleCon 2000.286-289

[37]Abdul Razaque Rind, Khurram Shahzad, M.Abdul Qadir. Evaluation and comparison of TCP and UDP over Wired-cum-wireless LAN[A].In: Multitopic Conference, Islamabad:2006.337-342

附录 A 课题相关实物照片

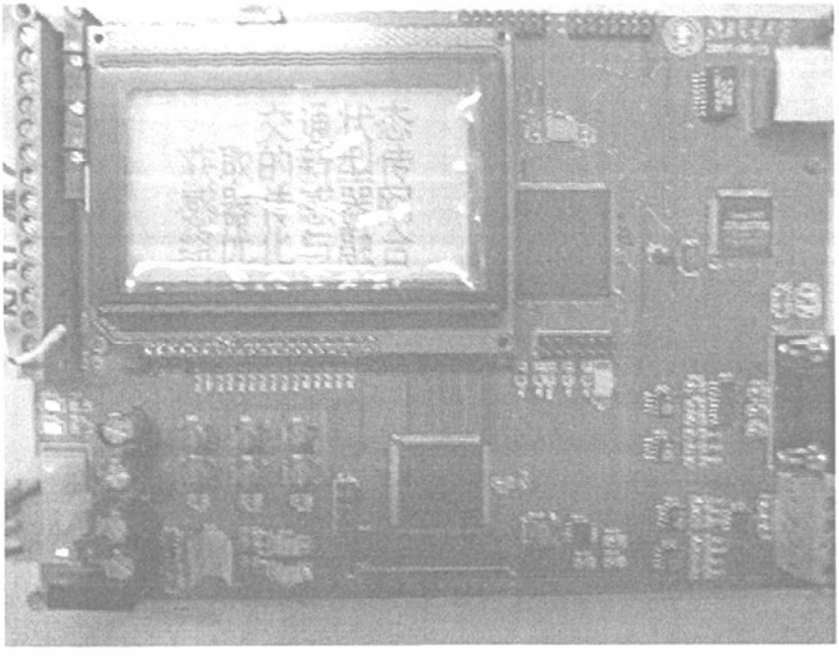


图 1 复合节点样机板卡

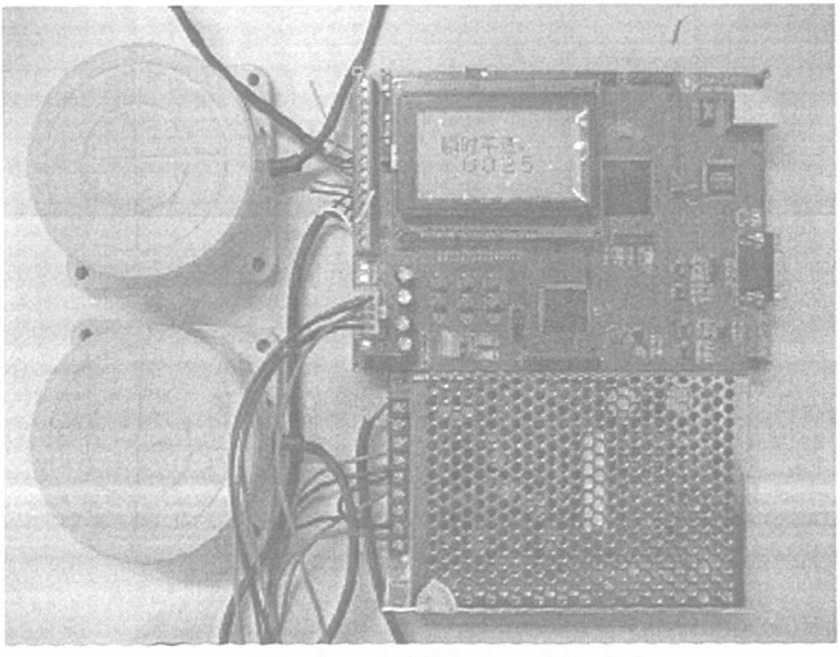


图 2 复合节点样机与涡流传感器

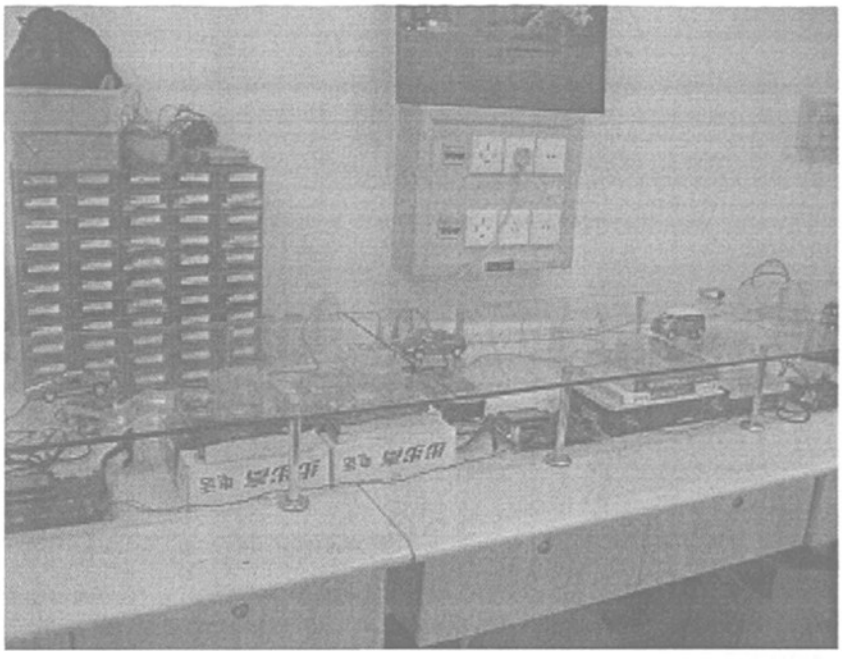


图 3 实验室环境下的模拟道路交通系统

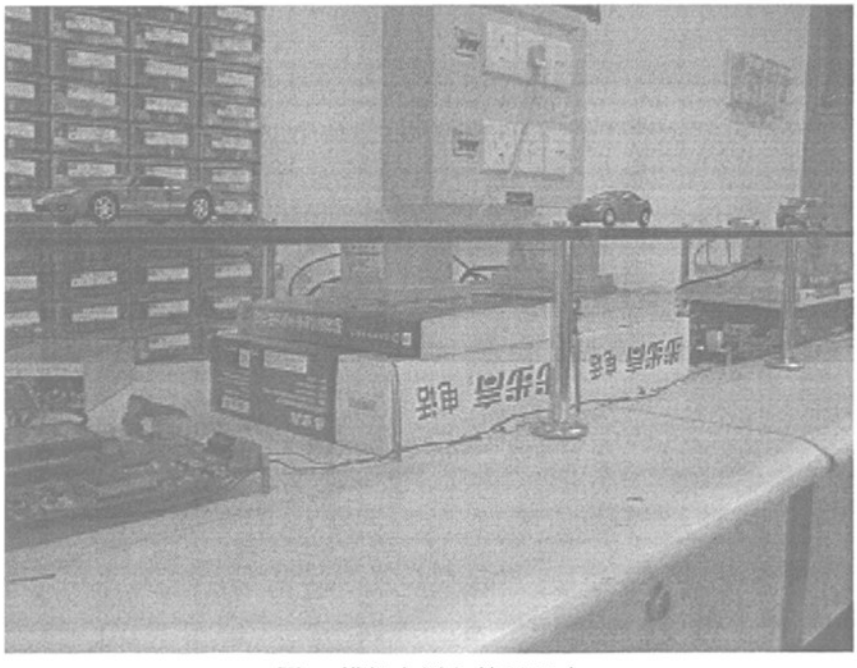


图 4 模拟车道与检测节点

作者简历

王强，1985 年生，甘肃天水人，2006 年获北京交通大学工学学士学位，2006-2008 年就读于北京交通大学电气工程学院，攻读工学硕士学位，研究方向为控制网络与检测技术。

攻读学位期间在国内期刊上发表的论文：

王强，张和生，叶华.交通状态获取的 UDP 通信方法[J]. 电子测量与仪器学报(已录用)