计算机网络

网络协议仿真教学系统

（基本版）

实验教材

吉林中软吉大信息技术有限公司

目　　录

【实验环境】---------- 1
【网络仿真编辑器的使用说明】---------- 3
【实验一　以太网数据帧的构成】---------- 16
【实验二　IP 地址分类与 IP 数据包的组成】---------- 21
【实验三　ARP 地址解析协议】---------- 26
【实验四　ICMP Internet 控制报文协议】---------- 31
【实验五　UDP 用户数据报协议】---------- 40
【实验六　TCP 传输控制协议】---------- 45
【实验七　TCP 与 UDP 端口扫描】---------- 51
【实验八　TELNET 和 FTP 协议】---------- 54
【实验九　DNS 域名服务协议】---------- 60
【实验十　DHCP 协议】---------- 67
【实验十一　SMTP 和 POP3 协议】---------- 71
【实验十二　SNMP 协议与网络管理】---------- 78
【实验十三　HTTP 协议】---------- 84
【实验十四　WINS 和 NETBIOS 协议】---------- 89
【实验十五　SOCKET 基础网络 程序设计】---------- 95
【实验十六　Ping 实现 程序设计】---------- 108
【实验十七　文件传输工具 程序设计】---------- 109
【实验十八　主机信息获取工具 程序设计】---------- 110
【实验十九　网络性能监测及故障分析】---------- 111
【附录：常见问题及解决方法】---------- 115

【实验概述】

网络协议仿真教学系统结合高校教育的实际情况，将网络方面的理论知识通过软件来实现，让学生在实践的过程中更深入地掌握网络方面的基础理论知识。本系统能够使学生清楚地理解和掌握网络的内部结构和协议，通过编辑各种协议的数据包深入学习计算机网络的内部原理，同时也可以很好地辅助网络编程的调试。网络协议仿真教学系统作为一门独立的课程体系，以实验为主，强调学生的主动性和设计性，能够拓宽学生的思路，达到真正的教学互动。

【实验环境】

每个实验都要求有以下实验环境：

1、仿真端：运行网络协议仿真教学系统仿真编辑器软件；

2、监控端：运行网络协议仿真教学系统协议分析器软件；

3、可以连接 Internet 的多子网环境；

4、服务器一台，装有 FTP、TELNET、MAIL、DNS、DHCP、SNMP 等服务；

5、VC 程序开发环境（Microsoft Visual c++集成开发环境）；

6、VB 程序开发环境（Microsoft Visual Basic 集成开发环境）；

【实验内容】

网络协议仿真教学系统（基本版）实验共有 19 个，每个实验 4 学时，要求学生完成每个实验及编写实验报告。通过一系列实验，希望学生能对 TCP/IP 协议有一个更深入的了解。

实验教学内容按照验证型、设计型和综合型三个层次，共设置了 19 个实验项目。

实验 1　以太网数据帧的构成：该实验类型为验证型，要求学生在了解仿真教学系统平台的基础上，学会使用仿真编辑器编辑以太网数据帧，学会观察并分析地址本中的 MAC 地址，学会分析以太网数据帧的 MAC 首部。实验教学效果达到理解 MAC 地址的作用、MAC 首部中的 LLC－PDU 长度及类型字段。

实验 2　IP 地址分类与 IP 数据包的组成：该实验类型为综合型，要求学生学会分类 IP 地址，掌握 IP 数据报的格式、长度以及各字段的功能。实验教学效果达到学会利用子网掩码确定 IP 地址的网络号、子网号和主机号，学会分析给定数据包的 IP 首部信息，学会手工计算 IP 校验和的方法。

实验 3　ARP 地址解析协议：该实验类型为验证型，要求学生在理解 IP 地址与 MAC 地址的对应关系的基础上，掌握 ARP 协议的作用与工作原理。实验教学效果达到利用 ARP 协议的工作原理，学会在局域网中查找 MAC 地址。

实验 4　ICMP Internet 控制报文协议：该实验类型为综合型，要求学生理解和掌握 ICMP 协议的基本作用与报文格式，理解 ICMP 协议与 IP 协议的封装关系。实验教学效果达到能根据各种响应信息进行出错分析。

实验 5　UDP 用户数据报协议：该实验类型为综合型，要求学生在理解 UDP 协议的基础上，学会 UDP 校验和的计算方法，理解 UDP 协议各字段的作用及工作原理。实验教学效果达到学会应用 C 语言与 WinSock2 进行网络程序设计。

实验 6　TCP 传输控制协议：该实验类型为验证型，要求学生理解 TCP 首部中各字段的含义及作用，理解三次握手的过程。实验教学效果达到能够分析 TCP 协议的建立连接、进行会话和断开连接的全过程，学会 TCP 校验和的计算方法。

实验 7　TCP 与 UDP 端口扫描：该实验类型为验证型，要求学生掌握常用的 TCP、UDP 端口扫描的原理及其各种手段，增强学生的网络安全意识。

实验 8　TELNET 和 FTP 协议： 该实验类型为验证型，要求学生理解 TELNET 和 FTP 协议的工作原理，了解他们的网络安全性和具体使用方法。

实验 9　DNS 域名服务协议：该实验类型为综合型，要求学生学习 DNS 协议的格式，理解 DNS 协议的工作原理。实验教学效果达到学会应用 C 语言与 WinSock2 进行 DNS 程序设计。

实验 10　DHCP 协议：该实验类型为验证型，要求学生学习 DHCP 协议的格式，理解 DHCP 协议的工作原理，验证获取 IP 地址与重新登陆过程。

实验 11　SMTP 和 POP3 协议：该实验类型为验证型，要求学生理解 SMTP 协议、POP3 协议及其作用，学习 SMTP、POP3 协议命令的使用方法。

实验 12　SNMP 协议与网络管理：该实验类型为验证型，要求学生理解 SNMP 协议的作用与格式，了解 SNMP 与 UDP 的关系。实验教学效果达到了解 MIB 及基本的 MIB 变量，了解 ASN.1 的编码标准。

实验 13　HTTP 协议：该实验类型为验证型，要求学生学习 HTTP 协议的格式，理解 HTTP 协议与 TCP 协议的关系，理解 HTTP 协议的无连接、无状态特性。

实验 14　WINS 和 NETBIOS 协议：该实验类型为验证型，通过发送 NETBIOS 名字请求包，学习 WINS 协议和 NETBIOS 协议的原理。学习 NETBIOS 请求的报文格式，了解 NETBIOS 使用的端口。要求学生理解 WINS 协议和 DNS 协议的不同。

实验 15　SOCKET 基础网络 程序设计：该实验类型为设计型，要求学生进一步掌握 UDP 及 TCP 协议的工作原理，掌握 SOCKET 编程的基本方法。实验教学效果达到应用 C 语言与 WinSock2 进行无连接的、面向连接的网络程序设计，实现网络数据传输。

实验 16　Ping 实现 程序设计：该实验类型为设计型，使学生加深对 ICMP 协议的理解，掌握原始套接字的编程方法，并学习设置 TTL 等自定义参数的编程方法。

实验 17　文件传输工具 程序设计：该实验类型为设计型，使学生加深对 TCP 协议的理解，掌握一对多的客户端服务器模型的编程方式。

实验 18　主机信息获取工具 程序设计：该实验类型为设计型，使学生掌握使用套接字获取主机信息的方法，理解 Windows 的 WMI 信息的获取方法，并掌握在 Windows 下的网络函数库 IpHelp 的功能与用法。

实验 19　网络性能监测及故障分析：该实验类型为综合型，要求学生掌握监测网络性能的基本方法，能够根据性能监测结果分析网络性能问题，能够分析常见故障及原因。实验教学效果达到检验学生对 TCP/IP 协议原理的掌握情况，以及综合分析问题的能力。

【仿真编辑器的使用说明】

一、仿真编辑器简介

仿真编辑器主界面如下：

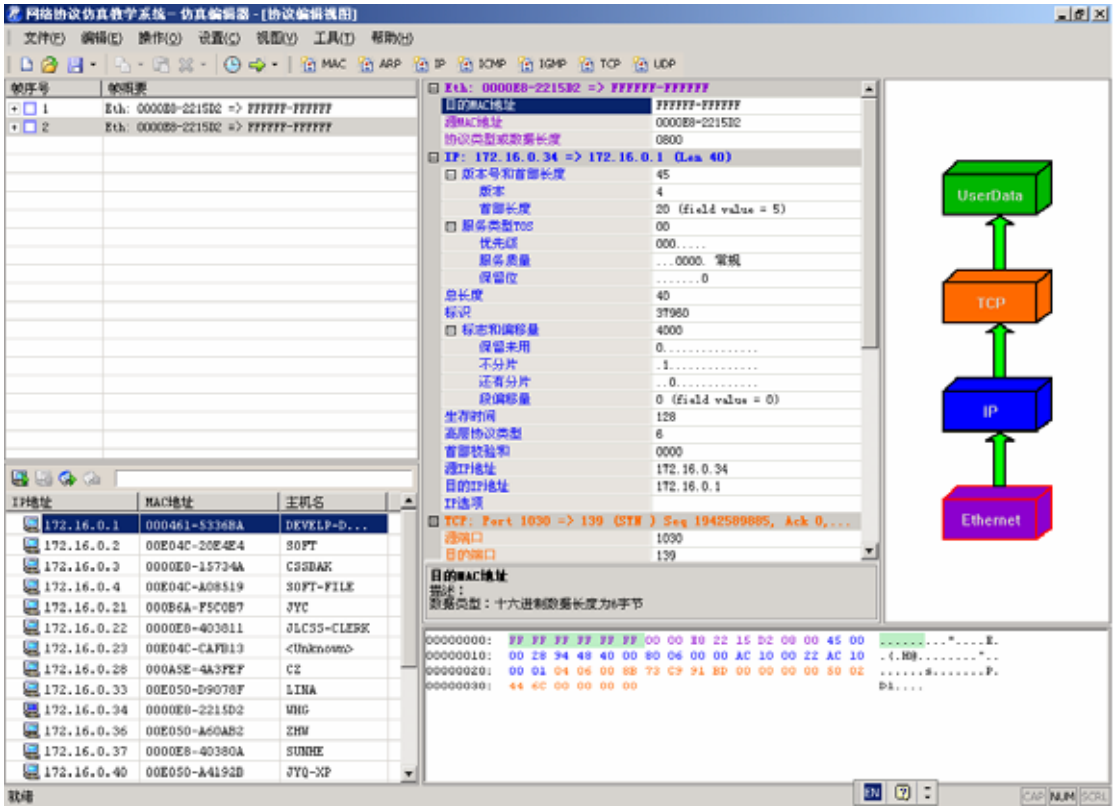


图 1-1　仿真编辑器的主界面

本系统的初始界面显示分为五个部分：多帧编辑器、单帧编辑器、协议模型图、地址本和十六进制显示区。

1、多帧编辑器

界面如图所示：

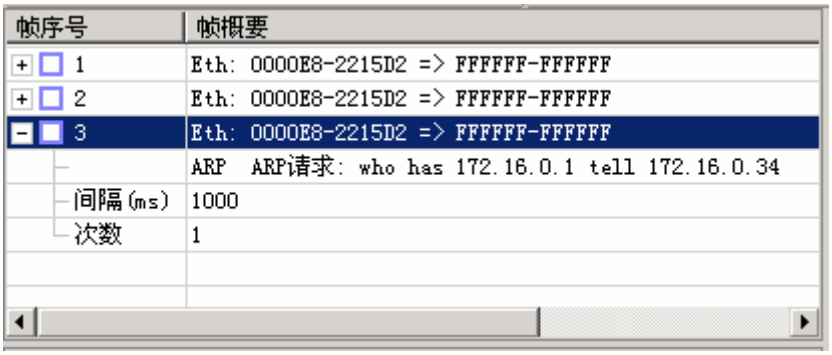


图 1-2　多帧编辑器

①第一列为帧的序号，此序号无其他特殊含义，即为各个帧顺序向下的计数号码。点击“+”可展开为多行，相邻两帧发送的时间间隔默认值为 1000（毫秒），如果点击展开的为第一帧，则为发送延迟时间。发送次数默认为 1 次。

②第二列为帧的概要信息：

MAC　源地址、目的地址、下层协议（如果在协议类型中选择了下拉框中的协议）

LLC　LLC Unnumbered/LLC Information/LLC Supervisor,DSAP,Ctrl

ARP　请求：who has 172.16.0.1 tell 172.16.0.34

ARP　应答：172.16.0.34 is at 0000E8-2215D2

IP　源IP、目的IP、IP总长度、上层协议（如果在协议类型中选择了已有的协议），（如果分片或偏移量≠0）Flagment 偏移量： 偏移量+（总长度-首部长度-1）

TCP　源端口、目的端口、TCP类型（SYN，FIN，ACK，RST）、序列号、确认号和窗口大小。

UDP　源端口、目的端口。

③具体帧的位置排序，可以手工自由拖动。拖动的单帧将插入到当前帧的上方；如果想将一个单帧拖动到最后的位置，则需要点击这个单帧将它拖动到“End”位置上。

2、单帧编辑器

界面如图所示：

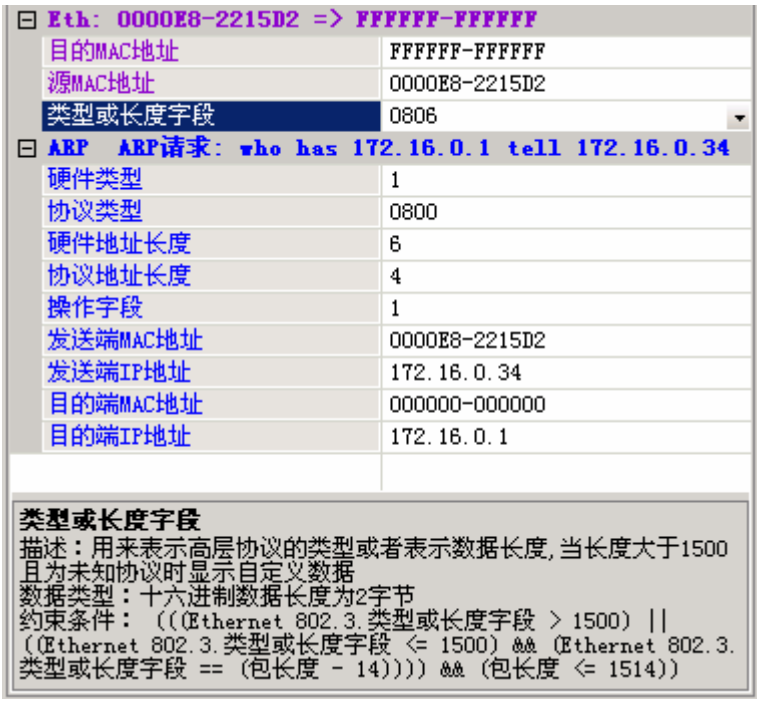


图 1-3　单帧编辑器

单帧编辑器分为帧编辑区和提示区两部分。帧编辑区可以对协议属性进行编辑。提示区可以根据选中的协议属性给出相应的提示，以便于用户了解各属性的含义或范围。

3、协议模型图

在协议模型图中，以图形化模型显示各协议的封装层次，并与单帧编辑器对应的协议层相互联动。当前选中的协议层用红色边框显示，如图所示：

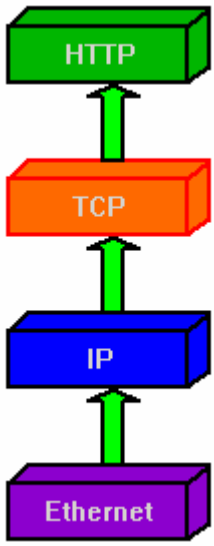


图 1-4　协议模型图

4、地址本

地址本包括两项功能：主机扫描和端口扫描。点击 主机扫描按钮，系统在当前网络内进行主机扫描，并在地址本中列出扫描到的所有主机。选中一台主机后，可以点击端口扫描按钮，对该主机进行 TCP 端口扫描。地址本的主机扫描结果分为三列：IP 地址、MAC 地址和主机名。每台主机的端口扫描结果也分为三列：端口号、端口类型和服务名称。如下图所示：

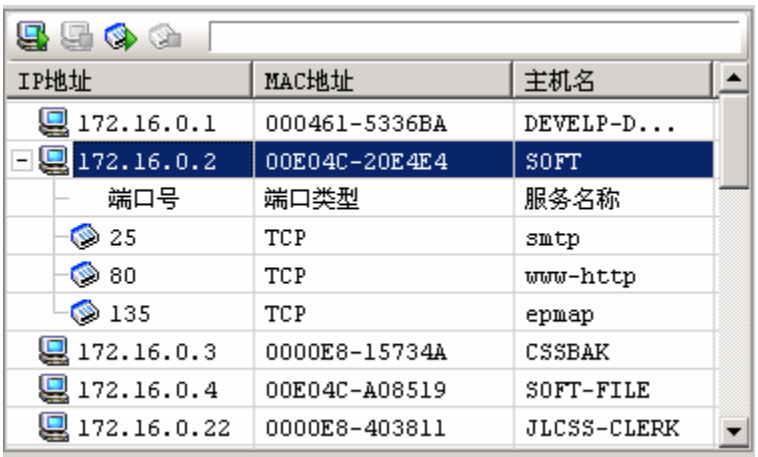


图 1-5　地址本

5、十六进制对照表

十六进制对照表与单帧编辑区是联动的，以绿色做为标识。右键可进行计算校验和、拷贝、计算数据长度操作。如图所示：

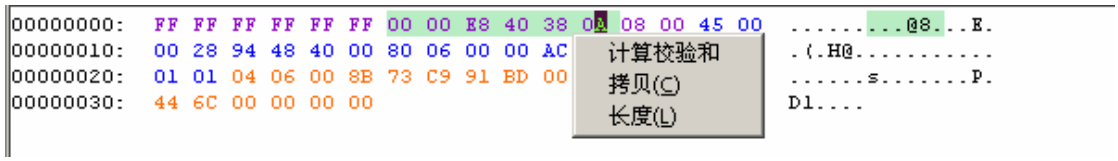


图 1-6　十六进制对照表

6、仿真编辑器的菜单栏

菜单栏包括“文件”、“编辑”、“操作”、“设置”、“视图”、“工具”、“帮助”七项，如图 1-7 所示：

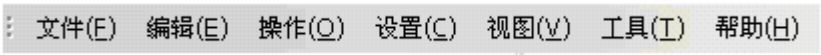


图 1-7 仿真编辑器菜单栏

（1）文件菜单：如下图所示：



图 1-8 文件菜单

新建文件：新建立一个文件；

打开文件：打开原有的文件；

保存文件：保存当前编辑的文件；

保存选中帧：对选中的数据帧进行保存；

退出：退出仿真编辑器。

（2）编辑菜单：如下图所示：



图 1-9 编辑菜单

新建帧：在帧序列的末尾新建一个所选择的帧；

插入帧：在当前帧之前插入一个所选择的帧；

删除当前帧：删除当前编辑的帧；

删除选中帧：删除所有选中的帧；

复制当前帧：将当前帧复制到剪贴版中；

复制选中帧：将所有选中的帧复制到剪贴版中；

粘贴帧：对剪贴版中的帧进行粘贴。

（3）操作菜单：如下图所示：

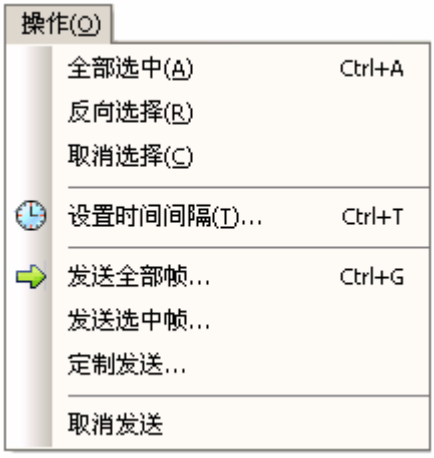


图 1-10 操作菜单

全部选中：对当前所有帧进行选中；

反向选择：对当前选中的帧进行反向选择；

取消选择：对当前选中的帧进行取消选择；

设置时间间隔：对全部帧或选中帧设置发送时间间隔；

发送全部帧：对多帧编辑器中的所有帧进行发送；

发送选中帧：对多帧编辑器选中的帧进行发送；

定制发送：自定义发送。在确认是否修改 IP 标示号、是否修改源 IP 地址、是否重新计

算并修改校验和后，设置发送时间间隔和发送次数，定制发送当前数据帧；

取消发送：停止正在进行的数据帧发送。

（4）设置菜单：如下图所示：

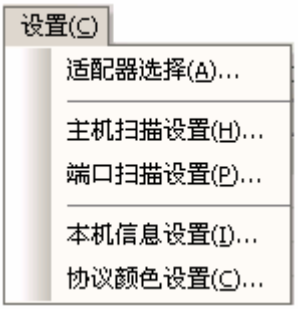


图 1-11 设置菜单

适配器选择：在主机有两个以上的适配器可用时，选择用来发送数据帧的适配器；

主机扫描设置：设置进行主机扫描的网络范围；

端口扫描设置：添加、删除、修改、复位要进行扫描的主机的端口，并可以将自定义的端口信息进行导出和导入；

本机信息设置：选择本地的 MAC、IP 地址，作为默认的 MAC、IP 地址，在新建数据包时使用；

协议颜色设置：可以自定义各个协议的颜色。

（5）视图菜单：

协议编辑视图：切换到协议编辑视图；

TCP 连接视图：切换到 TCP 连接视图；

SNMP 连接视图：切换到 SNMP 连接视图；

工具栏 — 标准工具栏：显示或隐藏标准工具栏；

向导工具栏：显示或隐藏向导工具栏；

状态栏：显示或隐藏状态栏。

（6）工具菜单

命令行：弹出 Windows 命令行窗口；

计算器：打开 Windows 计算器；

组播工具：打开组播工具对话框；

UDP 工具：打开 UDP 工具对话框；

TCP 过滤 — 启动过滤：使本机收不到 TCP 数据包，即将 TCP 协议数据过滤掉；

停止过滤：相对于启动过滤而言，使本机能够再接收 TCP 协议数据。

二、应用实例

系统将自动为用户提供若干实例，用户可以载入这些实例，重新编辑后进行发送。下面以一个例子来进一步说明仿真编辑器的功能与使用方法。

1、编辑 MAC 层

①源、目的 MAC 地址可以在地址本中进行查找，双击后自动添加，这样可以保证填写的 MAC 地址在网络中真实存在，尽量保证模拟的真实性。

②在上层协议中选择 IP 协议。用户可以根据下拉列表框中的条目选择此帧的上层协议；也可以直接填写，系统会根据填写的内容进行字符匹配判断，自动选择上层协议。

③编辑完成时的界面如图 2-1 所示：

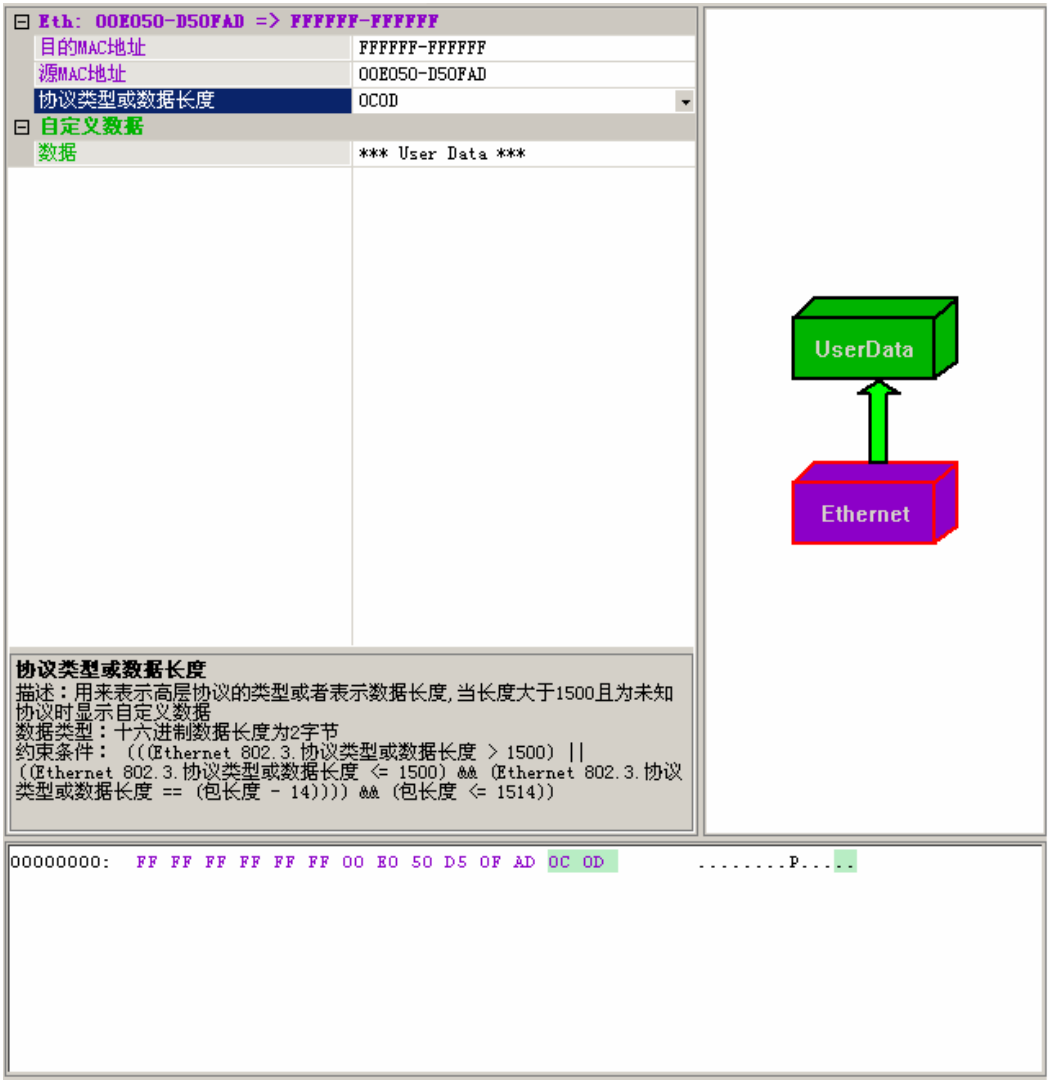


图 2-1　编辑 MAC 层的界面

2、编辑 IP 层

①用户选择 IP 协议后，单帧编辑区和协议模型图相应地刷新显示 IP 层的内容，如图 2-2 所示：

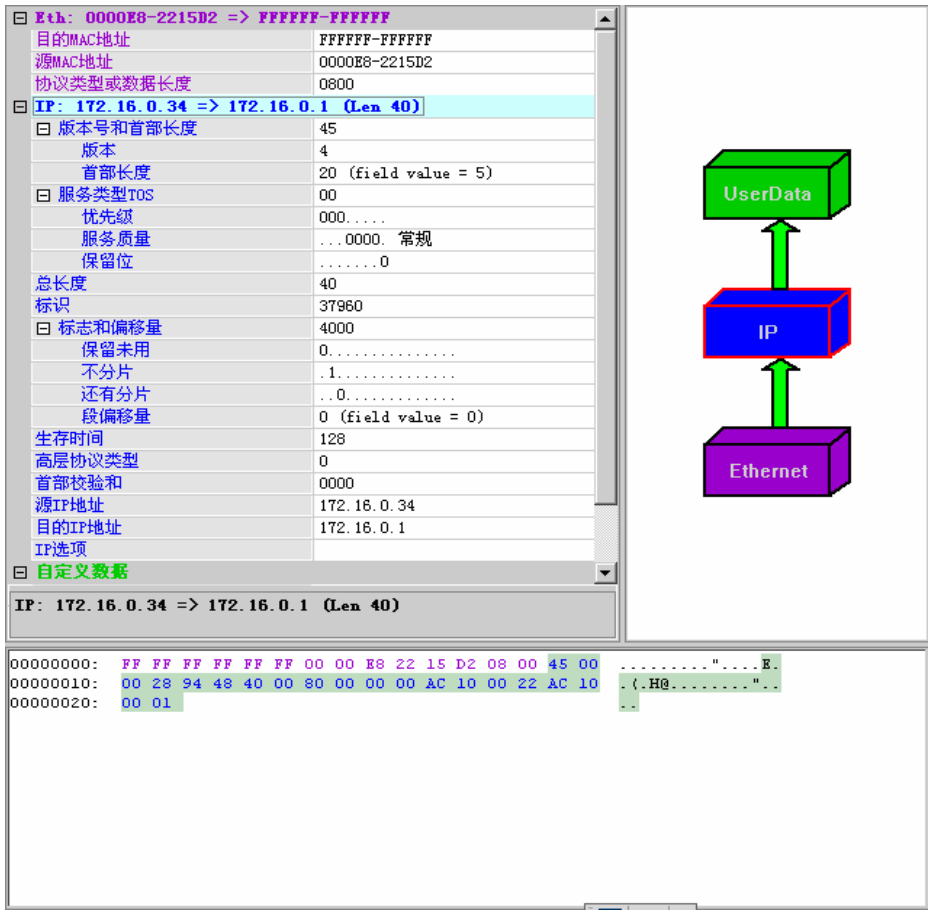


图 2-2　编辑 IP 层协议的界面

②当前 IP 层为选中，协议模型图中被红色线条选中。

③单帧编辑区显示 IP 层用户需要填写的各个属性。

④十六进制显示区显示用户填写的属性的十六进制值。

说明：IP 层涉及到校验和的计算方法，其具体使用如下：

由于单帧编辑区与十六进制显示区为联动的关系，用户可以点击单帧编辑区的某个属性（其变为蓝色），这时十六进制显示区与之相对应的十六进制代码变为绿色，用户可以根据这一方法找到校验和在十六进制显示区的位置，然后选中此段十六进制数值，点击鼠标右键，在弹出的对话框中选择“计算校验和”，用户可以将这个值回填到单帧编辑器 IP 层的“校验和”属性中。在计算校验和前一定要将该字段清零。

3、编辑 TCP 层

①与前两层相同，单帧编辑区和协议模型图进行相应的刷新后，用户就可以填写其相应的属性值了。

②在每一层，用户均可以填写“自定义数据”这一选项，但是一旦填写了这一项，就意味着这个单帧已经编辑完毕，并且按照用户填写的数据发送出去。

③TCP 的上层协议是通过端口来判断的。用户填写不同的端口，上层协议类型会不同。

【协议分析器的使用说明】

协议分析器负责捕获网络上的各种数据包，分析其中包含的各层协议，提供辅助教学功能，主要包含两个功能模块：会话分析和协议解析。

进入协议分析器，单击开始捕获数据包按钮，进行数据捕获。刷新显示后，在会话分析和协议解析视图中显示对数据的分析。

1、会话分析

会话分析功能将捕获到的常用协议的数据帧加入会话列表，直观的看出一次完整的会话过程，并且有会话次序、数据报传输方向的图示。如图所示：

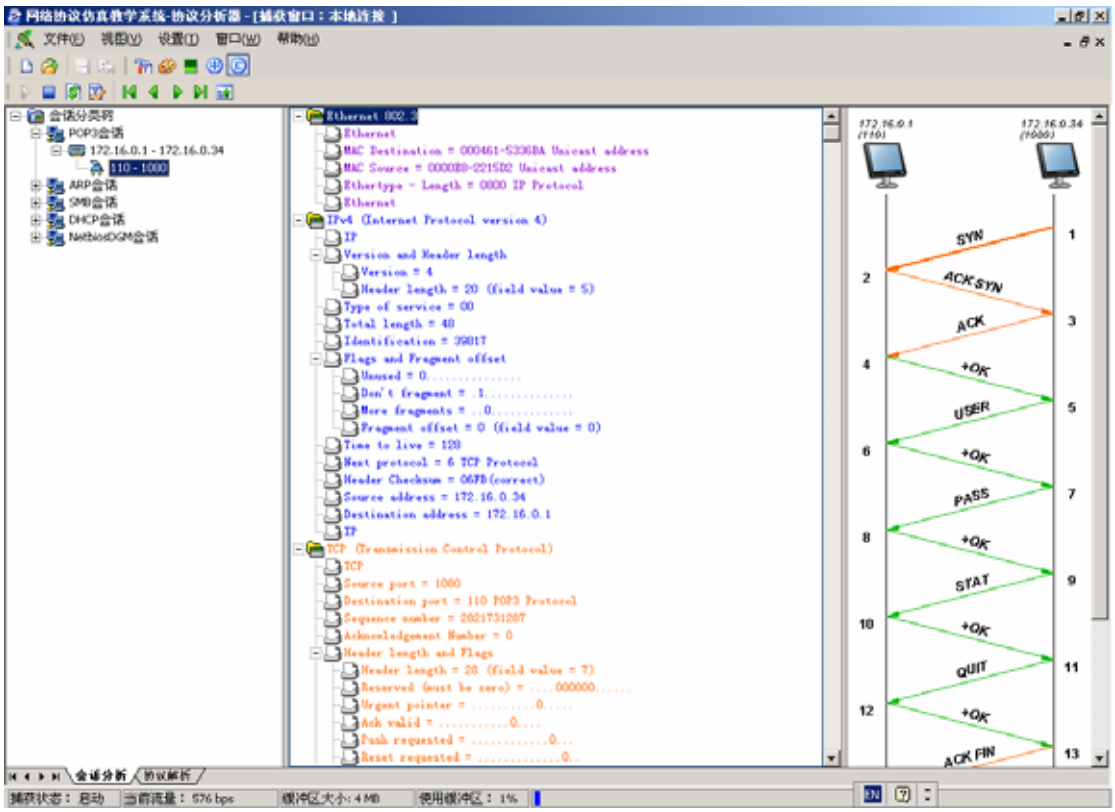


图 3-1 会话分析界面

会话分析功能主要用于有完整会话过程的实验，例如：ARP、ICMP、HTTP、DNS、SMTP、POP3 等。会话列表包含有通信双方的 IP 地址、端口号、会话类别。通过一次完整会话的分析，使学生加深对协议内容的理解。

2、协议解析

协议解析界面如图 3-2 所示：

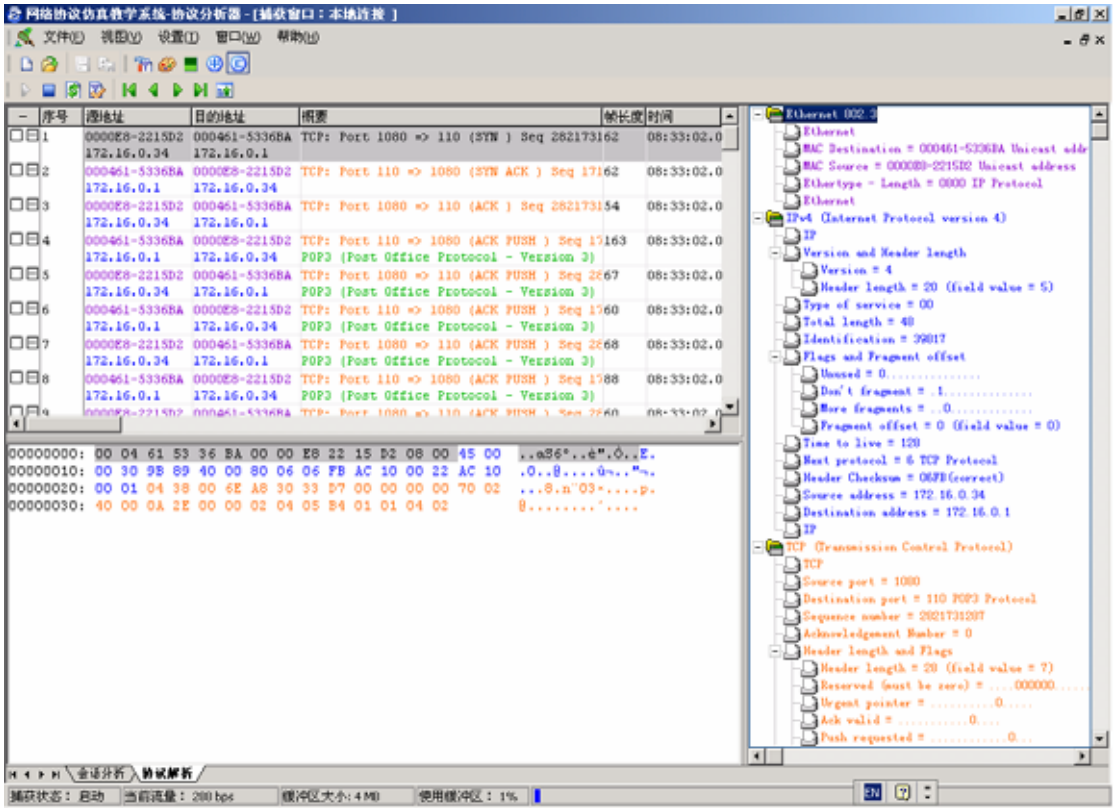


图 3-2 协议解析界面

该窗口主要有三个显示区：概要解码显示区、详细解码区和原始数据显示区，原始数据显示区包括十六进制数据显示和字符显示两种方式。

用户可以从详细协议解码显示中获得非常详细的解释和说明，不同栏目的数据和解释可以按照用户的选择实现动态跟踪显示，方便用户对数据的分析。

定义过滤器：

当观察的数据量很多而且比较杂乱时，想要从中提取有用或关心的数据就显得比较费力，为此本系统提供了定义过滤器的功能，可以对符合过滤条件的数据进行提取或滤除。对于定义的过滤条件，可保存成过滤器配置文件。当应用程序重新启动时，上一次保存应用的过滤器配置仍然有效，这一点要注意。过滤器可以实现“网络地址”、“数据模式”和“协议过滤”三种过滤，在首界面中显示当前定义的过滤条件概要信息。设置界面如图所示：

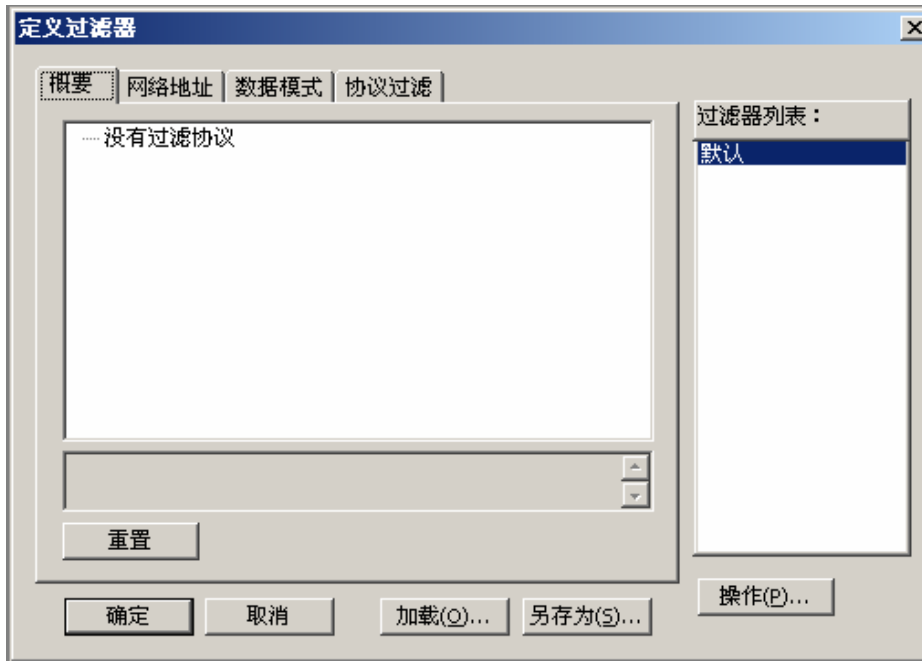


图 3-3　定义过滤器窗口

网络地址过滤：网络地址包括 IPv4、MAC 两种地址类型，设置界面如下图所示：

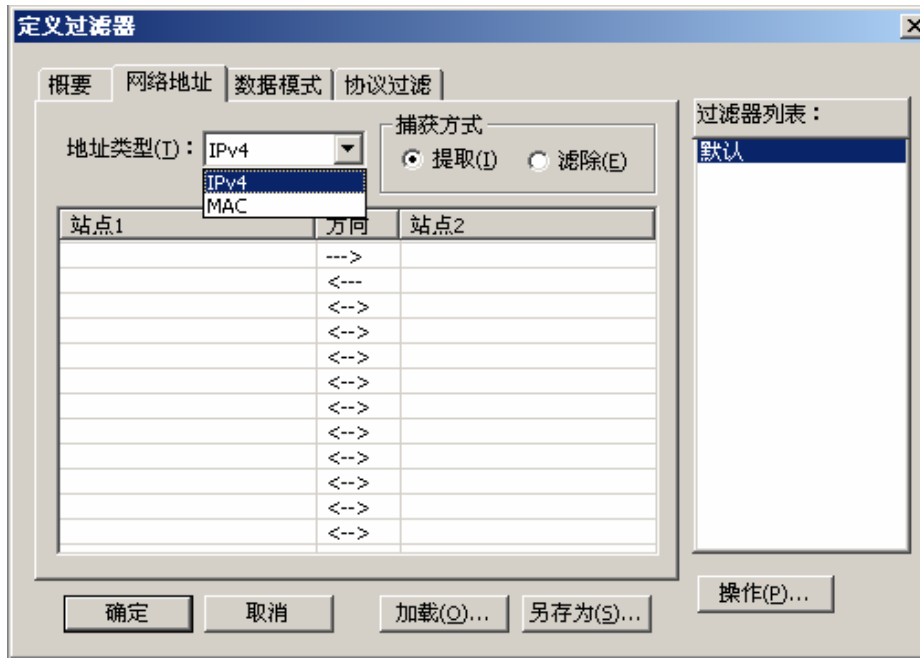


图 3-4　网络地址过滤设置

填写网络地址，中间的编辑框表示数据方向选择，在这里可以选择需要过滤地址数据的传输方向，*站点 1* 中填写传输的一端地址，*站点 2* 中填写传输的另一端地址，如果不填写*站点 2*，系统缺省设为 Any，表示过滤*站点 1* 到任意地址的数据。

数据模式过滤：设置界面如图所示：

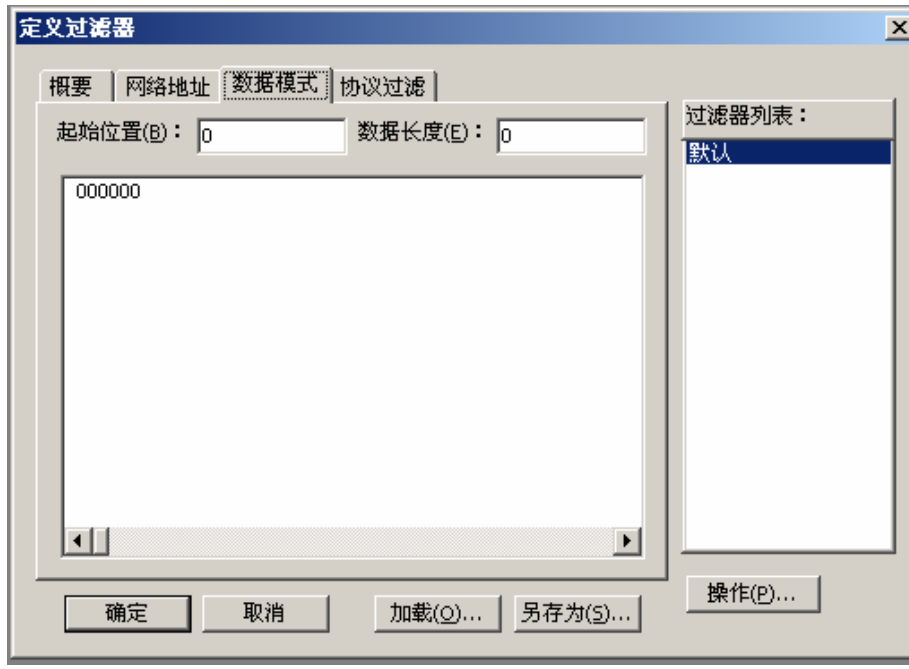


图 3-5　数据模式过滤

数据模式过滤，是针对十六进制数据而言的，起始位置指的是十六进制数据中的第几个字节，数据长度是指从起始位置算起共有多少个字节。下面的文本框中要输入过滤的十六进制数据，长度要与填入的“数据长度”一致。

协议过滤：协议过滤可以针对关心的协议进行设置，如果针对该协议有封装类型复选框，则需要进行选择，不做选择则过滤设置无效。设置界面如图所示：



图 3-6　协议过滤设置

单击 [操作] 按钮显示如图 3-7 的对话框：

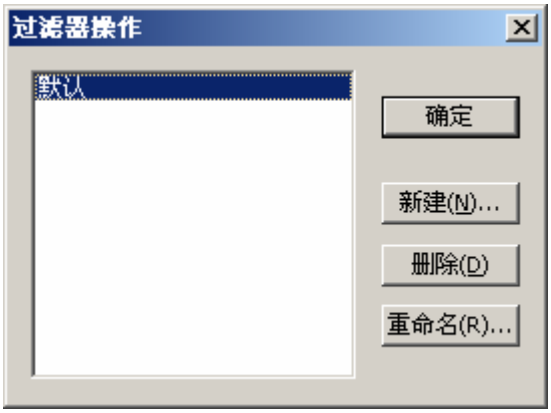


图 3-7　过滤器操作设置

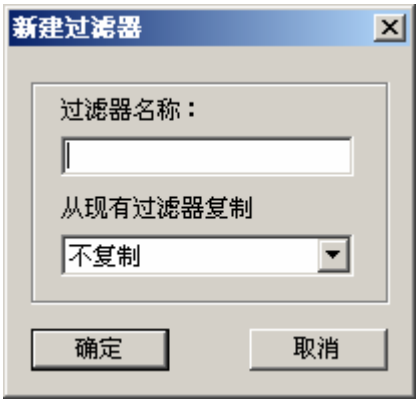


图 3-8　新建过滤器

在过滤器操作对话框中点击“新建”按钮，弹出“新建过滤器”对话框，如图 3-8 所示。在“过滤器名称”中输入过滤条件名称，确认后进行过滤条件设置。也可以通过“删除”，“重命名”按钮对用户已定义的过滤条件进行删除、重命名操作，但不可对“缺省过滤条件”（过滤器名称为“默认”）进行这样的操作。

在定义过滤器窗口中，单击“另存为”可保存该过滤条件，通过“加载”按钮可打开保存过的过滤条件，方便用户的使用。

【实验一　以太网数据帧的构成】

【实验目的】

1、掌握以太网帧的构成，了解各个字段的含义；
2、能够识别不同的 MAC 地址并理解 MAC 地址的作用；
3、掌握协议分析器的基本使用方法；
4、掌握仿真编辑器的基本使用方法。

【实验学时】

4 学时；

【实验类型】

验证型；

【实验内容】

1、学习仿真编辑器的各组成部分及其功能；
2、学习协议分析器的各组成部分及其功能；
3、学会使用仿真编辑器编辑以太网帧，包括单帧和多帧；
4、学会分析以太网帧的 MAC 首部和 LLC-PDU 的内容；
5、理解 MAC 地址的作用；
6、理解 MAC 首部中的 LLC-PDU 长度/类型字段的功能；
7、学会观察并分析地址本中的 MAC 地址。

【实验原理】

局域网（LAN）是在一个小的范围内，将分散的独立计算机系统互联起来，实现资源的共享和数据通信。局域网的技术要素包括了体系结构和标准、传输媒体、拓扑结构、数据编码、媒体访问控制和逻辑链路控制等。其中主要的技术是传输媒体、拓扑结构和媒体访问控制方法。局域网的主要的特点是：地理分布范围小、数据传输速率高、误码率低和协议简单等。

1、三个主要技术

⑴**传输媒体：**双绞线、同轴电缆、光缆、无线。
⑵**拓扑结构：**总线型拓扑、星型拓扑和环型拓扑。
⑶**媒体访问控制方法：**载波监听多路访问/冲突检测（CSMA/CD）技术。

2、IEEE 802 标准的局域网参考模型

IEEE 802 参考模型包括了 OSI/RM 最低两层（物理层和数据链路层）的功能，OSI/RM

的数据链路层功能，在局域网参考模型中被分成媒体访问控制 MAC（Medium Access Control）和逻辑链路控制 LLC（Logical Link Control）两个子层。由于局域网采用的媒体有多种，对应的媒体访问控制方法也有多种，为了使数据帧的传送独立于所采用的物理媒体和媒体访问控制方法，IEEE 802 标准特意把 LLC 独立出来形成单独子层，使 LLC 子层与媒体无关，仅让 MAC 子层依赖于物理媒体和媒体访问控制方法。LLC 子层中规定了无确认无连接、有确认无连接和面向连接三种类型的链路服务。媒体访问控制技术是以太网技术的核心。以太网不提供任何确认收到帧的应答机制，确认必须在高层完成。

3、以太网帧结构

以太网中传输的数据包通常被称为“帧”，以太网的帧结构如下：



图 1-1　以太网帧结构

各字段的含义：

目的地址：6 个字节的目的物理地址，标识帧的接收结点。

源地址：6 个字节的源物理地址，标识帧的发送结点。

帧类型/长度（TYPE/LEN）：该字段的值大于或等于 0x0600 时，表示上层数据使用的协议类型。例如 0x0806 表示 ARP 请求或应答，0x0800 表示 IP 协议。该字段的值小于 0x0600 时表示以太网用户数据的长度字段，上层携带 LLC-PDU。

数据字段：这是一个可变长度字段，用于携带上层传下来的数据。

帧校验 FCS：以太网采用 32 位 CRC 冗余校验。校验范围是目的地址、源地址、长度/类型、数据字段。

4、LLC-PDU 部分

当长度/类型字段的值小于 0x0600 时，说明数据字段携带的是 LLC-PDU。

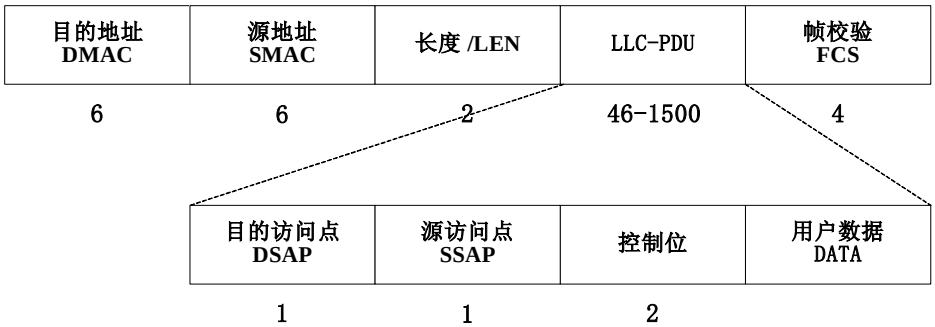


图 1-2　LLC-PDU 格式

三种 LLC 帧的控制位：

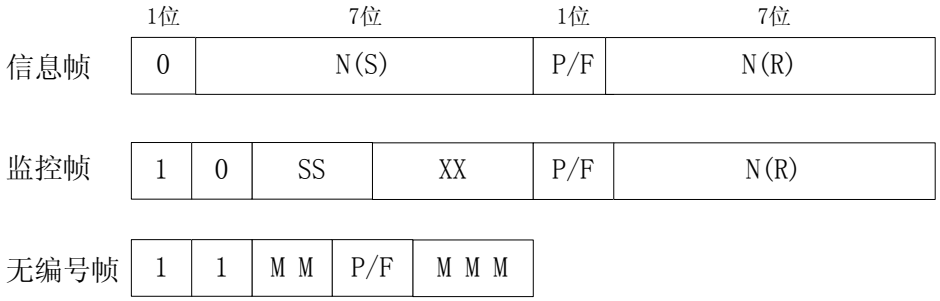


图 1-3　LLC 控制位的示意图

N（S）：是发送帧的序列号。

N（R）：是接收帧的序列号。

P/F：是 POLL/FINAL 轮询/终止位。

SS：是监控帧类型指示。标识监控帧的具体类型，即：RR（接收就绪）、RNR（接收未就绪）和 REJ（拒绝）。

MM-MMM：无编号帧的修饰段，定义发送无编号帧的具体类型。

3、详细原理请参考教材

【实验步骤】

一、仿真编辑器端

练习一：运行 ipconfig 命令

1、打开仿真编辑器，选择“工具”菜单中的“命令行”，出现命令提示符后输入：

C:>ipconfig/all

2、观察运行结果，获得本机的主机名及以太网地址。

3、在地址本中找到本机的相关信息。

练习二：编辑 LLC 信息帧并发送

1、打开仿真编辑器，在工具栏选择“新建”，建立一个以太网帧（也可以利用仿真编辑器打开时默认建立的以太网帧进行编辑）；此时在仿真编辑器的各部分会显示出该新帧的信息。

⑴多帧编辑器中显示：新帧的序号、概要信息；

⑵协议树中显示以太网 MAC 层协议；

⑶单帧编辑器中显示新帧各字段的默认值；

⑷十六进制对照表中显示新帧对应的十六进制信息；

2、编辑以太网帧

在单帧编辑器中编辑该帧，如图所示：

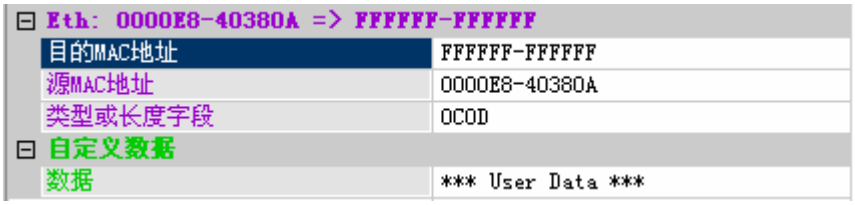


图 1-4 单帧编辑器中以太网帧示意图

具体步骤为：

⑴填写“目的 MAC 地址”字段；

方法一：手工填写。

方法二：选择地址本某主机的 MAC 地址，双击自动添加。

⑵填写“源 MAC 地址”字段，方法同上；

注意：仿真编辑器可以仿真本机发送的 MAC 帧，也可以仿真另一台主机发送 MAC 帧，所以，源 MAC 地址字段可以填写本机 MAC 地址，也可以填写其他主机的 MAC 地址。但要注意协议分析器显示过滤器的设置，否则会捕获不到数据包。

⑶填写“类型或长度”字段：可以填写 0X0043；同时在 LLC 的 DATA 中填写 0X0040 个数据；例如：在 LLC 的 DATA 中添入：

AAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBCCCCCCCCCCCCCCCCDDDDDDDDDDDDDDDD；

注意：填写的类型或长度字段为十六进制数，填写的数值应等于 LLC 层与数据字段的长度和。

⑷协议树中会显示 LLC；其中的 SSAP 和 DSAP 是源和目的访问点，可以不变。

⑸修改 LLC 的控制字段，最低位为零，表示信息帧。

3、单帧编辑器与十六进制对照表是联动的，观察十六进制对照表中该帧的信息；

4、选中单帧编辑器中的某一字段，该字段对应的十六进制值会相应地改变颜色；

5、点击工具栏或菜单栏中的“发送”，在弹出的对话框上选择“按范围发送”或“自定义发送”，正确填写发送的范围后，选择“发送”按钮，发送该帧；

6、协议分析器一端捕获新发送的帧并分析:该帧的发送序号 N（S）是多少？接收序号 N（R）是多少？是什么类型的帧？

7、考虑“类型和长度”字段分别填写以下十六进制值的含义:

⑴0x0800：对应的上层协议是什么？在监控端捕获该帧，捕获到帧的长度是多少？

⑵0x0806：对应的上层协议是什么？

⑶0x0100：该值的具体含义是什么？在协议分析器端捕获该帧，帧的长度是多少？

⑷0x0001：这样填写该字段是否正确？为什么？

练习三：编辑 LLC 监控帧和无编号帧，并发送和捕获：

1、编辑帧的步骤同练习二，只是修改 LLC-PDU 的控制字段。

2、监控帧和无编号帧时的控制字段的内容应该是什么？

3、在监测端捕获数据包，确定是否是监控帧和无编号帧？

练习四：编辑并发送帧序列

1、按照单帧编辑的步骤编辑多个帧；

2、对每一帧可以重新设置发送的时间间隔和发送的重复次数，如不设置，则采用默认值；

方法为：点击某一帧左边的“+”，即可展开为两行，在第一行内填写此帧与其上一帧的发送时间间隔(单位：ms， 默认值：1000),展开项的第二行为设置此帧的重复发送次数（默认值为 1）。

3、编辑好以后，发送帧序列；

点击菜单栏或工具栏中的“发送”按钮，选择弹出对话框中的“按范围发送”，并编辑起始帧序号和结束帧序号。

如果选择“自定义发送”，则在“发送序号”编辑框中选择要发送的单帧序号，并根据需要修改发送“间隔时间”和“发送次数”。点击“发送”按钮，即完成帧的发送。

4、观察主界面底部“状态”栏中的各项数据信息。

5、在协议分析器一端捕获数据并分析。

6、将编辑好的以太网帧保存到文件中，文件名为：MAC.pkt。

二、协议分析器端

练习一：捕获数据并分析

1、启动协议分析器协议解析功能，获得若干以太网帧。对其中的 5-10 个帧的以太网首部进行观察和分析；分析的内容为：源 MAC 地址、目的 MAC 地址、上层协议类型。

2、以 MAC 地址为过滤条件捕获若干以太网帧。

如：捕获源 MAC 地址为：00-09-74-A5-6F-4C（选择地址本中的一个 MAC 地址）的帧。对每一帧的以太网首部进行观察分析；分析的内容为：源 MAC 地址、目的 MAC 地址、上层协议类型。

3、以本机 MAC 地址为过滤条件捕获相应的以太网帧（多个）；对其中目的 MAC 地址为本机 MAC 地址的以太网帧进行观察分析:

⑴这些帧的源 MAC 地址都是什么？

⑵对照地址本观察这些 MAC 地址对应的 IP 地址有何特点？

⑶说明：为什么能收到来自这些源 MAC 地址的帧？

⑷将捕获到的以太网帧保存到文件中，文件名为：实验一.zdt。

练习二：学习协议分析器端其他的协议分析功能

1、在会话分析视图中,观察会话列表中的会话种类，并通过协议详细解析树、协议分析视图了解会话过程；

2、在协议解析视图中，观察概要解析及十六进制对照表。

3、学习使用过滤器的设置和调用方法。

【思考问题】

结合实验过程中的实验结果，回答下列问题：

1、在网络的分层体系结构中，MAC 层的作用是什么？

2、以太网的最小帧长度是（　　），　最大帧长度是（　　）？

3、为什么 IEEE802 标准将数据链路层分割为 MAC 子层和 LLC 子层？

【实验报告要求】

1、对以太网数据帧各字段的理解，怎样编辑 U 帧、S 帧和 I 帧？

2、为什么以太网有最小帧长度的要求？

3、回答练习中提出的问题。

【实验二　IP 地址分类与 IP 数据包的组成】

【实验目的】

1、理解 IP 层的作用以及 IP 地址的分类方法；

2、理解子网的划分和子网掩码的作用；

3、掌握 IP 数据包的组成和网络层的基本功能。

【实验学时】

4 学时

【实验类型】

综合型

【实验内容】

1、学会根据 IP 地址的分类方式区分各类 IP 地址；

2、掌握 IP 数据报的格式、长度以及各字段的功能；

3、学会利用子网掩码确定 IP 地址的网络号、子网号和主机号；

4、学会分析给定数据包的 IP 首部信息；

5、学会手工计算 IP 校验和的方法；

6、学习通过简单的 VC 编程，读取主机的 IP 地址。

【实验原理】

1、IP（Internet Protocol）协议和 IP 地址的分类

IP 协议用于管理客户端和服务器端之间的报文传送，是互联网最重要的网际协议。IP 地址是互联网主机的唯一标识。网络通信时使用 IP 地址来指定接收端的主机地址。IPv4 版本的 IP 地址占 32 位，由类别、网络地址和主机地址共 3 个部分组成。网络地址用于区分不同的网络，主机地址用于区分一个网段内主机。IP 地址分成 5 类：A 类，B 类，C 类，D 类和 E 类。其中 A、B 和 C 类是分配给用户使用的，D 类用于多点广播，E 类保留。

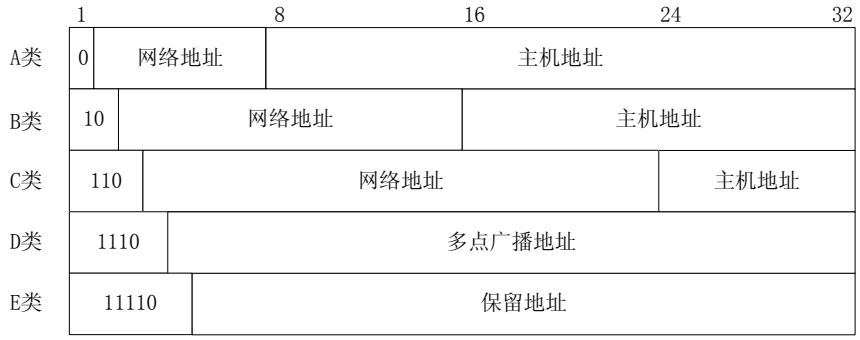


图 2-1　IP 地址分类示意图

2、各类 IP 地址的范围

A 类：0.0.0.0 到 127.255.255.255

B 类：128.0.0.0 到 191.255.255.255

C 类：192.0.0.0 到 223.255.255.255

3、特殊的 IP 地址

①网络地址：主机地址部分为全“0”的 IP 地址是作为网络本身的标识。

②直接广播地址：主机地址为全“1”的 IP 地址不分配给任何主机，用作广播地址。

③有限广播地址：32 位为全“1”的 IP 地址(255.255.255.255)称为有限广播地址。

④专用 IP 地址：　10.0.0.0 — 10.255.255.255；

172.16.0.0 — 172.31.255.255；

192.168.0.0 — 192.168.255.255

4、子网划分

子网是将网络地址中的主机标识分离出若干位作为子网地址位。**掩码**是一个 32 位的二进制数。网络地址部分为 1，主机地址部分为 0。子网的设计是为了便于网络的管理、提高系统性能。给定 IP 地址和子网掩码以后，主机就可以确定 IP 数据报的目的是：①本子网上的主机；②本网络中其他子网中的主机；③其他网络上的主机。根据 IP 地址可以知道它是 A 类、B 类或 C 类地址，同时也知道网络号和子网号之间的分界线。根据子网掩码可以知道子网号与主机号之间的分界线。

5、IP 数据报格式



图 2-2　IP 分组格式

【实验步骤】

练习一：利用仿真编辑器编辑并发送 IP 数据包

1、打开仿真编辑器，在工具栏选择“新建”，建立一个以太网帧（也可以使用系统打开时默认建立的以太网帧进行编辑）；

2、手动填写“源 MAC 地址”，或在地址本中双击主机自动填入。

3、手动填写“目的 MAC 地址”，或在地址本中双击主机自动填入。

4、填写“类型或长度”：该字段值为 0800（即 IP 协议的类型值）或者选择工具栏中的 IP 协议， 此时，单帧编辑区中自动添加 IP 协议的各属性字段，并有默认值。右侧的协议模型图也会相应增加 IP 层显示。

5、填写 IP 协议头信息：

⑴版本号和首部长度：版本号为 4，首部长度为 5（即 20 个字节）；

⑵服务类型 TOS：00；

⑶总长度：该值为 IP 首部长度加上数据部分的长度；如果没有数据该字段应为 20，否则加上数据的长度；

⑷标识号：可以采用默认值，或任意值，例如 12345；

⑸标志字段：通过修改各种标志值，在监控端捕获数据包分析理解各种标志含义；

⑹生存时间：可以采用默认值 128；

⑺高层协议类型：即 IP 携带的上层协议类型（例如：TCP 为 6，UDP 为 17，ICMP 为 1）；本实验填 0（没有上层协议数据）；

⑻首部校验和；先添 0，等全部字段填完后再计算；

⑼源 IP 地址：注意，仿真编辑器可以仿真本机发送 IP 数据包，也可以仿真另一台机器发送 IP 数据包，所以，源 IP 地址字段可以填写本机 IP 地址，也可以填写其它机器的 IP 地址（注意协议分析器的显示过滤器设置）；

⑽目的 IP 地址：从地址本中选择一个目的主机的 IP 地址，双击自动填加；

⑾当上述各字段值均已填写完毕后，可以通过两种方法计算首部校验和的值；

6、计算“校验和”

⑴**方法一：**手工计算

首先把“校验和”字段置为 0，单击 M 按钮，弹出手动计算校验和窗口，单击 IP 协议首部，十六进制对照表中对应的十六进制数据会以绿色标注。右键拷贝，粘贴到 IP 首部文本框中，单击计算按钮，得出校验和，或在十六进制中右键直接计算校验和。

⑵**方法二：**自动校验

单击 A 按钮实现自动计算并填充。

7、如果要编辑多个 IP 数据包，可重复上述步骤，并可在多帧编辑器中设置每一帧的发送次数和发送每帧的时间间隔；

8、点击工具栏或菜单栏中的“发送”，在弹出的对话框中选择“发送”按钮，发送帧序列；

9、协议分析器一端截获数据包并分析；

10、将编辑正确的 IP 包保存到文件：IP.pkt。

练习二：运行 ipconfig 命令

1、在“工具”菜单中选择“命令行”，出现提示符后输入 ipconfig/all

2、观察运行结果，获得本机的 IP 地址及子网掩码；

3、分析本主机属于哪一类 IP 地址，网络号、子网号和主机号分别是什么；

4、利用本机的 IP 地址及子网掩码，找出地址本中与本机属于同一子网的主机。

练习三：运行 netstat 命令

Netstat 命令用于显示与 IP、ICMP、TCP 和 UDP 协议相关的统计数据，一般用于检验本机各端口的网络连接情况。

1、在命令行提示符下运行：

C:>netstat －r ；显示本机路由表，记录本机的缺省网关的 IP 地址；

2、在命令行提示符下运行：

C:>netstat －s ；观察 IP 协议部分，查看本机已经接收和发送的 IP 报文个数。

练习四：读本机 IP 地址的 VC 程序设计

1、启动 Visual C++，进入集成开发环境；

2、建立一个“Win32 Console Application”类型的工程,工程名自拟；

3、在该工程下，新建一个源程序文件，即选择“C++ Source File”，文件名自拟；

4、输入以下源程序；

```
#include "windows.h"
#include "winsock.h"
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
```

```
#pragma comment ( lib, "ws2_32.lib" )
void print_ip(void)
{
  WORD wVersionRequested;        //*初始化 windows sockets API*//
  WSADATA wsaData;
  char name[255];
  PHOSTENT hostinfo;
  wVersionRequested =MAKEWORD(2,0);
  if(WSAStartup(wVersionRequested,&wsaData)==0)
  {
    if(gethostname(name,sizeof(name))==0)
    {
      if((hostinfo=gethostbyname(name))!=NULL)
        { LPCSTR ip=inet_ntoa(*(struct in_addr*)*hostinfo->h_addr_list);
        printf("%s\n",ip);
        }
    }
    WSACleanup();
  }
}
int main(int argc, char* argv[])
{
  print_ip();
  return 0;
}
```

5、编译、连接并运行程序。

练习五：运行 Ping 命令

1、在地址本中选择与本主机在同一子网中另一主机的 IP 地址（假设为：172.16.1.46）；

⑴在命令行方式下运行：ping 172.16.1.46；

⑵在协议分析器一端捕获数据，记录源、目的 MAC 地址和源、目的 IP 地址；

⑶按照地址本中的记录，分析捕获数据的 MAC 地址与 IP 地址的对应关系；

⑷在协议分析器一端,查看“会话分析”中的“会话图解”，了解 PING 程序的会话过程。

2、选择与本主机属于不同子网另一主机的 IP 地址（假设为：172.16.0.34）；

⑴在命令行方式下运行：ping 172.16.0.34；

⑵协议分析器端捕获数据，记录源、目的 MAC 地址和源、目的 IP 地址；

⑶分析捕获数据的 MAC 地址与 IP 地址是否具有对应关系。

⑷在协议分析器端,查看“会话分析”中的“会话图解”，分析 PING 程序的会话过程。

3、比较上面两个实验的结果，分析二者有何不同？

【思考问题】

结合实验过程中的实验结果，回答下列问题：

1、实验所用主机的 IP 地址、子网掩码、网络号、子网号分别是多少？该主机的 IP 地址属于哪类？

2、分析地址本中所有的 IP 地址，说明实验环境中所有机器的子网划分情况。

3、IP 数据包在从源主机出发到达目的主机的过程中，IP 首部中的 IP 源地址和目的地址字段是否发生变化？

【实验报告要求】

1、IP 数据报各字段格式的含义。

2、通过实验理解 MAC 地址和 IP 地址在网络传输中的作用及区别。

3、回答练习中提出的问题。

【实验三　ARP 地址解析协议】

【实验目的】

1、掌握 ARP 协议的作用和格式；

2、理解 IP 地址与 MAC 地址的对应关系；

3、了解 ARP 命令。

【实验学时】

4 学时

【实验类型】

验证型

【实验内容】

1、学习 ARP 协议的工作原理；

2、掌握 ARP 协议的作用和使用方法；

3、理解 IP 地址与 MAC 地址的对应关系；

4、学习使用 ARP 命令。

【实验原理】

IP 数据包常通过以太网传输，在以太网中是以 48 位的物理地址传输数据包而不是 32 位的 IP 地址。因此当我们知道 IP 地址不知道 MAC 地址时，就要用到 ARP 协议。ARP 协议的功能是将网络层的 IP 地址转换为数据链路层的 MAC 地址。ARP 协议的请求包是以广播方式发送的，网段中的所有主机都会接收到这个包。如果一个主机的 IP 地址和 ARP 请求中的目的 IP 地址相同,该主机会对这个请求数据包作出 ARP 应答,将其 MAC 地址发送给请求端。

1、ARP 请求或应答的分组格式

<table>
<tr><td colspan="2">硬件类型(1)</td><td>协议类型(0800)</td></tr>
<tr><td>硬件长度</td><td>协议长度</td><td>操作(1;2)</td></tr>
<tr><td colspan="3">发送方MAC(八位组0-3)</td></tr>
<tr><td colspan="2">发送方MAC(八位组4-5)</td><td>发送方IP地址(八位组0-1)</td></tr>
<tr><td colspan="2">发送方IP地址(八位组2-3)</td><td>目标MAC(八位组0-1)</td></tr>
<tr><td colspan="3">目标MAC(八位组2-5)</td></tr>
<tr><td colspan="3">目标IP地址(八位组0-3)</td></tr>
</table>

图 3-1　ARP 协议的分组格式

字段说明：

硬件类型：表示硬件类型，例如：1 表示以太网。

协议类型：表示要映射的协议类型，例如 0x0800 表示 IP 地址。

硬件长度：指明硬件地址长度，单位是字节，MAC 是 48 位，6 个字节。

协议长度：高层协议地址的长度，对于 IP 地址，长度是 4 个字节。

操作字段：共有二种操作类型，1 表示 ARP 请求，2 表示 ARP 应答。

发送方 MAC：6 个字节的发送方 MAC 地址。

发送方 IP： 4 个字节的发送方 IP 地址。

目的 MAC：6 个字节的目的 MAC 地址。

目的 IP： 4 个字节的目的 IP 地址。

2、ARP 缓存和 ARP 命令

为减少广播通信量和提高 ARP 解析速度，每个主机都有 ARP 高速缓存，用于存放解析过的 MAC 和 IP 的映射关系。ARP 缓存表是可以通过命令查询的。在命令提示符下，输入“arp -a”可以查看 ARP 缓存表中的内容，“arp -d”清除 ARP 缓存。

3、ARP 工作过程

假设网络中的计算机 A 要和计算机 B 交换数据，首先计算机 A 要得到计算机 B 的 IP 地址和 MAC 地址的映射关系，工作过程如下：

①计算机 A 检查自己的高速缓存中的 ARP 表，判断 ARP 表中是否存有计算机 B 的 IP 地址与 MAC 地址的映射关系。如果找到，则完成 ARP 地址解析；如果没有找到，则转至②。

②计算机 A 广播含有自身 IP 地址与 MAC 地址映射关系的请求信息包，请求解析计算机 B 的 IP 的地址与 MAC 地址映射关系。

③包括计算机 B 在内的所有计算机接收到计算机 A 的请求信息，然后将计算机 A 的 IP 地址与 MAC 地址的映射关系存入各自的 ARP 表中。

④计算机 B 发送 ARP 响应信息，通知自己的 IP 地址与 MAC 地址的对应关系。

⑤计算机 A 收到计算机 B 的响应信息，并将计算机 B 的 IP 地址与 MAC 地址的映射关系存入自己的 ARP 表中，从而完成计算机 B 的 ARP 地址解析。

【实验步骤】

注意：没有强调在“协议分析器”一端操作时，以下操作均是在“仿真编辑器”上进行的。

练习一：仿真发送 ARP 请求报文（同网段内）

本练习的目的是理解 ARP 协议的数据包格式和协议的功能。

1、在命令提示符下运行：

C:>arp －d　　清空 ARP 高速缓存。

2、编辑并发送 ARP 请求帧：

⑴打开仿真编辑器，在界面初始状态下，程序会自动新建一个单帧，可以利用仿真编辑器打开时默认的以太网帧进行编辑；

⑵填写该以太网帧的各字段值；

①源 MAC 地址：可以为本机的 MAC 地址，也可以仿真其他主机发送 ARP 请求报文。其他主机的 MAC 地址可根据主机名在地址本中获得。如果仿真其他主机发送 ARP 请求报文，应注意协议分析器的过滤器设置。

②目的 MAC 地址：对于 ARP 请求来说，目的 MAC 地址应该是什么？我们常说 ARP 广播帧，所以目的 MAC 应该是广播地址：FFFFFF-FFFFFF。

③类型或长度字段：0806H（即：上层是 ARP 协议类型值）或者单击工具栏中的 ARP 协议选项，此时，单帧编辑区中将显示出 ARP 协议的各属性字段，并有默认的属性值。

⑶填写 ARP 协议信息：

①硬件类型：1，以太网地址；

②协议类型：0800，要映射的协议地址类型为 IP 地址类型；

③硬件地址长度：6，MAC 地址是 6 个字节 48 位；

④协议地址长度：4，IP 地址是 4 个字节 32 位；

⑤操作字段：1，ARP 请求；2，ARP 应答；

⑥发送端 MAC 地址：发送 ARP 包的源主机的 MAC 地址；

⑦发送端 IP 地址：发送 ARP 包的源主机的 IP 地址；

⑧目的端 MAC 地址：任意，可以填 0；

⑨目的端 IP 地址：选择目的主机的 IP 地址；

⑷点击工具栏或菜单栏中的“发送”，在弹出对话框中选择“发送”按钮；

⑸在协议分析器端捕获该请求数据包及 ARP 响应包并分析，ARP 响应包的源地址是多少？

⊟ Eth: 0000E8-40380A => FFFFFF-FFFFFF	
目的MAC地址	FFFFFF-FFFFFF
源MAC地址	0000E8-40380A
类型或长度字段	0806
⊟ ARP ARP请求: who has 172.16.1.1 tell 172.16.1.16	
硬件类型	1
协议类型	0800
硬件地址长度	6
协议地址长度	4
操作字段	1
发送端MAC地址	0000E8-40380A
发送端IP地址	172.16.1.16
目的端MAC地址	000000-000000
目的端IP地址	172.16.1.1

图 3-2　ARP 协议树的示意图

3、将编辑正确的 ARP 帧保存到文件：ARP.pkt。

4、在仿真端和分析端命令提示符下运行：

C:>arp　-a　　；检查 ARP 高速缓存中的内容，是否发生了变化？

注：若仿真端接收响应信息后缓存表里没有目的主机的信息列出，则说明操作系统未读取显示目的主机的地址解析。

练习二：仿真发送 ARP 请求报文（不同网段内）

1、参照练习一，编辑一个 ARP 数据包，目的主机的 IP 选择其他网络（例如：172.16.0.56）；

2、在协议分析器一端捕获该数据包及 ARP 响应包并分析，观察 ARP 协议的会话过程。

3、记录 ARP 响应包的源地址 MAC 是多少？该 MAC 是否是目的 IP 的 MAC？理解 ARP 路由代理的过程。

练习三：运行 Ping 命令（同一子网内）

1、在命令提示符下运行 arp　-d 命令，清空 ARP 高速缓存；

2、在命令提示符下运行 arp　-a 命令，检查 ARP 高速缓存。

3、从地址本中选择与本机属于同一子网的某主机的 IP 地址（例如：172.16.1.56）作为 Ping 命令的参数，然后在命令提示符下运行：

C：>Ping 172.16.1.56

4、在协议分析器一端捕获数据报文并分析:

⑴其中是否有 ARP 请求及应答报文?

⑵分析各 ARP 报文：源、目的 MAC 地址都是什么？源、目的 IP 地址都是什么？

⑶为什么产生这样的 ARP 报文？

5、在命令提示符下运行：arp －a 命令，检查 ARP 高速缓存中的内容是否发生变化？分析原因。

6、清空协议分析器的 ARP 高速缓存，再次运行 Ping 172.16.1.56，在协议分析器一端捕获的数据报文是否和上一次一样？

练习四：运行 Ping 命令（经网关连接的不同子网上的主机）

1、在命令提示符下运行：arp －d 命令，清空 ARP 高速缓存

2、在命令提示符下运行：arp －a 命令，检查 ARP 高速缓存

3、从地址本选择与本机属于不同子网的主机的 IP 地址（例如：172.16.0.34），然后在命令提示符下运行：

C:>Ping 172.16.0.34

4、在协议分析器一端捕获数据报文:

⑴其中是否有 ARP 请求及应答报文数据包?

⑵分析各 ARP 报文：源、目的 MAC 地址是什么？源、目的 IP 地址是什么？

5、在命令提示符下运行 arp －a 命令，检查 ARP 高速缓存中的内容是否发生变化，是否存在 172.16.0.34 和其 MAC 的“地址对”？分析原因。

6、比较练习三和练习四两实验中得到的 ARP 报文有何不同？分析产生二者区别的原因？

【思考问题】

结合实验过程中的实验结果，回答下列问题：

1、观察实验过程中捕获网络上的多个 ARP 请求帧，观察这些帧的以太网目的地址是否相同，分析其原因。

2、观察实验过程中捕获网络上的多个 ARP 应答帧，观察这些帧的以太网目的地址是否相同，分析其原因。

3、ARP 缓存的作用。

【实验报告要求】

1、通过实验，解释 ARP 在网络中的作用，区分同一网段和不同网段响应包的区别。

2、回答练习中提出的问题。

【实验四　ICMP Internet 控制报文协议】

【实验目的】

1、了解 ICMP 协议的格式和响应方式及作用；
2、了解常见的网络故障。

【实验学时】

4 学时

【实验类型】

综合型

【实验要求】

1、学习和掌握 ICMP 协议的基本作用；
2、掌握 ICMP 报文的格式；
3、理解 ICMP 协议与 IP 协议的封装关系；
4、学会根据各种响应信息进行出错分析的方法；
5、学会使用 ping、netstat 和 tracert 命令。

【实验原理】

IP 协议是一种不可靠无连接的包传输，当数据包经过多个网络传输后，可能出现错误、目的主机不响应、包拥塞和包丢失等。为了处理这些问题，在 IP 层引入了一个子协议 ICMP（Internet Control Message Protocol）。ICMP 数据报有两种形式：差错数据报和查询数据报。ICMP 数据报封装在 IP 数据报里传输。ICMP 报文可以被 IP 协议层、传输层协议（TCP 或 UDP）和用户进程使用。ICMP 与 IP 一样，都是不可靠传输，ICMP 的信息也可能丢失。为了防止 ICMP 信息无限制的连续发送，对 ICMP 数据报传输中问题不能再使用 ICMP 传输。

1、ICMP 报文的封装

ICMP 有两种报文：差错报文和查询报文。两种报文都是封装在 IP 报文中进行传输的，具体的封装格式见图 4-1：

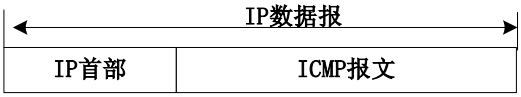


图 4-1　ICMP 的封装

2、ICMP 报文格式

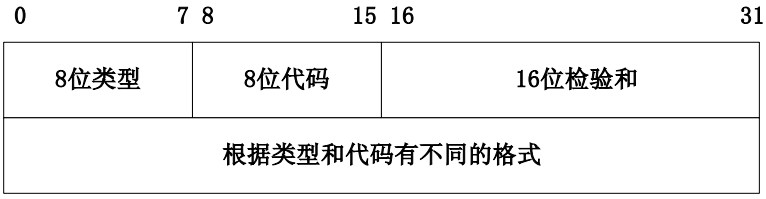


图 4-2　ICMP 协议的报文格式

字段说明：

ICMP 类型和代码字段：8 位的类型字段有 15 个不同值，它与 8 位代码字段共同决定各种类型的 ICMP 报文。

校验和字段：对 ICMP 整个报文中每个 16 bit 进行二进制反码求和。

3、ICMP报文的主要类型

表4-1　ICMP的主要类型

类型	代码	描述	查询	差错
0	0	回显应答(Ping应答)	•	
3		目的不可达		•
	0	网络不可达		•
	1	主机不可达		•
	2	协议不可达		•
	3	端口不可达		•
	4	需要进行分片但设置了不分片比特		•
4	0	源端被关闭（基本流控制）		•
5	0	对网络重定向		•
	1	对主机重定向		•
8	0	请求回显（Ping请求）	•	
9	0	路由器通告	•	
10	0	路由器请求	•	
11	0	传输期间生存时间为0（Traceroute）		•
	1	在数据报组装期间生存时间为0		•
12	0	坏的IP首部（包括各种差错）		•
13	0	时间戳请求	•	
14	0	时间戳应答	•	
17	0	地址掩码请求	•	
18	0	地址掩码应答	•	

4、ICMP 时间戳请求和应答报文格式

ICMP 时间戳请求允许系统向另一个系统查询当前的时间。返回的结果是自午夜开始计算的毫秒数。这种提供时间服务的 ICMP 报文的优点是它提供了毫秒级的分辨率。由于返回

的时间是从午夜开始计算的，因此调用者必须通过计算方法获取当前的时间。请求端填写发起时间戳，应答系统收到请求报文时填写接收时间戳，在发送应答时填写发送时间戳。

类型（13或14）	代 码（0）	校 验 和
标 识 符		序 列 号
发起时间戳		
接收时间戳		
传送时间戳		

图 4-3　ICMP 时间戳请求和应答报文格式

5、ICMP 回显请求和应答报文格式

网络测试程序 Ping 就是利用 ICMP 的回显请求报文完成的，回显请求报文的类型是 8，代码是 0。其格式如下：

类型=8	代 码=0	检 验 和
标 识 符		序 列 号
数　　据		

图 4-4　ICMP 回显请求报文格式

6、详细内容请参考教材

【实验步骤】

练习一：仿真发送回显请求 ICMP 报文

本练习的目的是学习最常用的 ICMP **回显请求**报文的格式，学习 ICMP 的一般使用方法。

1、从地址本中选择一台主机，向该主机仿真发送 ICMP 回显请求报文；

⑴打开仿真编辑器，在界面初始状态下，程序会自动新建一个单帧，可以利用仿真编辑器打开时默认的以太网帧进行编辑；

⑵单击工具栏上的 ICMP 协议，自动添加 IP 协议及 ICMP 协议（ICMP 封装在 IP 中）。

⑶填写该帧的以太网协议首部：

①源 MAC 地址：源 MAC 地址是本机的 MAC，可以在地址本中选择双击；

②目的 MAC 地址：目的 MAC 地址是目的主机的 MAC 地址，可以在地址本中选中一个主机，然后双击（注意：当选择的目的主机与本机不在同一子网的情况，目的 MAC 地址该怎样填写？）；

③类型或长度字段：该字段值为 0800（即 IP 协议的类型值，ICMP 封装在 IP 中）；

2、填写 20 个字节的 IP 协议头信息；

（1）填写源、目的 IP 地址；

（2）总长度：总长度应为 IP 长度+ICMP 长度；如果没有数据，长度是 28；

（3）计算 IP 首部校验和：首先将该字段清 0，单击 A 按钮，自动计算并填充；单击 M 按钮手动计算；

（4）其他字段可以使用默认值；

3、填写 ICMP 协议信息，见图 4-5：

⑴类型：8，回显请求类型；

⑵代码：0；

⑶标识符：任选 16 位二进制数（如 512）；

⑷序列号：任选 16 位二进制数（如 256）；

⑸填充位：即数据字段。可填充任意数据（注意：如果添加了 DATA，要修改 IP 总长度）；

⑹校验和：首先将该字段清 0，单击 A 按钮，自动计算并填充；单击 M 按钮手动计算；

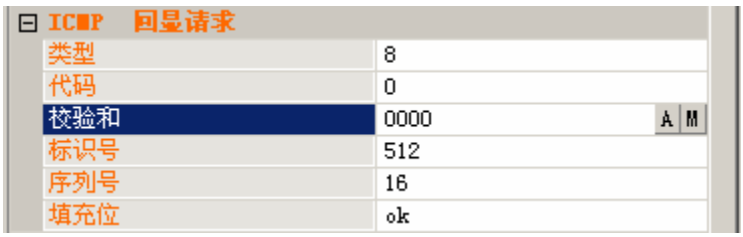


图 4-5 ICMP 协议信息

4、点击工具栏或菜单栏中的“发送”，在弹出对话框中选择“发送”按钮，发送 ICMP 报文；

5、在监控机一端截获相应的请求报文及应答报文并分析（注意当源 MAC 与源 IP 属于另一台主机的时候，响应报文的情况）。

6、将编辑好的 ICMP 报文保存到文件：ICMP.pkt。

7、编辑 IP 数据包并主动引入一些错误：

⑴向目标主机发送一个只有 IP 头的 IP 数据包；

⑵向目标主机发送一个数据包，版本号字段填入 7；

⑶向目标主机发送一个数据包，协议类型字段填入 120；

⑷向目标主机发送一个数据包，填写不正确的 IP 报头长度；

向目标主机发送上述数据包；在监控机一端捕获响应 ICMP 报文并分析。对目标主机发送一个错误的 ICMP 报文，观察该主机是否会产生对错误 ICMP 报文的响应报文。注意观察“会话分析”中的 ICMP 会话图解。

练习二：仿真发送 ICMP 时间戳请求报文

1、从地址本中选择一台主机，向该主机仿真发送 ICMP 时间戳请求报文。

2、操作步骤同练习一。注意对于 ICMP 时间戳请求报文，IP 总长度字段的值，ICMP 的类型字段值为 13，代码字段值为 0，见图 4-6 所示。在协议分析器一端截获相应的请求报文及应答的 ICMP 报文并分析；查看时间戳的值。

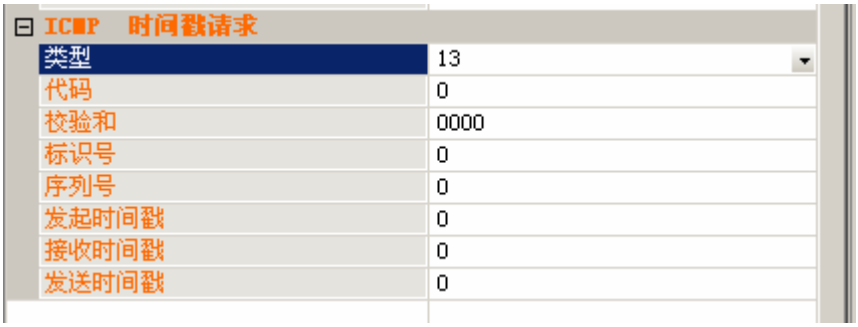


图 4-6　ICMP 时间戳请求报文示意图

练习三：运行 ping 命令

关闭本网段所有主机的防火墙，在命令行提示符下运行 ping 命令：

1、从地址本中选择与本机属于同一子网的某主机的主机名或 IP 地址，作为 ping 命令的参数；例如：ping 172.16.1.56

2、选择与本机属于不同子网的某主机的主机名或 IP 地址，作为 ping 命令的参数；例如：ping 172.16.0.34

3、对于上述两种情况，在仿真编辑器和协议分析器端分别进行观察和分析：

⑴在命令窗口观察：

①是否收到响应包？

②如果收到，观察响应包的个数，并对每个响应包的各信息字段进行解释；

③比较两次实验中 TTL 字段值是否相同，为什么？

⑵在协议分析器一端捕获数据并分析：

①是否捕获到响应包？

②如果捕获到，观察并分析每个响应包的 ICMP 协议头部分；

③比较两次实验中协议分析器一端捕获到的响应包的个数是否相同？

⑶同时打开多个命令行提示符窗口，在每个窗口中运行一个 ping 程序：

①在协议分析器一端捕获数据，分析各程序如何识别属于自己的返回信息？

②在“会话分析”中观察多个 ping 程序产生的多个“会话图解”。

练习四：运行 netstat 命令

在命令行环境下运行 netstat　-s，观察 ICMP 协议部分，记录本机已经接收和发送的 ICMP 报文类型及个数。

练习五：运行 tracert 命令

Tracert 程序的实现方法是：主机先发出一个 TTL（Time To Live，生存时间）为 1 的数据包。接着数据包经过第一跳之后到达一个路由器，此时路由器将发回一个 ICMP 超时错误消息，以说明这个数据包不能继续向前传送，因为其 TTL 已经过期了。然后 tracert 程序又发出一个 TTL 为 2 的数据包，这个数据包经过两次跳跃后其 TTL 过期。第 2 个路由器又将发回一个 ICMP 超时错误消息。这个过程不断重复，直到数据包到达目的主机。此过程的目

的就是要记录下所有发出 ICMP 超时消息的路由器，由此就可以确定出数据包到达目的主机的路径了。本练习的目的是学习 tracert 命令的原理，tracert 过程中使用了哪些协议。步骤如下：

仿真端，在命令提示符下运行以下命令：

C:>tracert oa.jlu.edu.cn

1、观察运行结果，分析解释路由路线；

2、在监控端捕获数据包，分析 tracert 程序都利用了什么协议？

3、在上面的例子中，tracert 程序共发送了几个数据包，每次发送包的 TTL 值是多少？

4、根据 TTL 值，说明 tracert 程序发现路由的原理。

5、再运行:

C:>tracert 202.198.16.80

C:>tracert d61.jlu.edu.cn

观察结果。

练习六：VC 编程实现 PING 功能

本实验利用微软的 ICMP.DLL 实现“ping”另一台主机的功能，具体实现如下:

1、启动 Visual C++，进入集成开发环境；

2、建立一个“Win32 Console Application”类型的工程,工程名自拟；

3、在该工程下，新建一个源程序文件，即选择“C++ Source File”，文件名自拟；

4、输入以下源程序（注释部分可不输入）：

```
#include <iostream.h>
#include <winsock.h>
#include <windowsx.h>
#include "icmpdefs.h"
#pragma comment ( lib, "ws2_32.lib" )

//==================ping 的实现部分=================//
int doit(int argc, char* argv[])
{
  // 装载 ICMP.DLL 连接库//
  HINSTANCE hIcmp = LoadLibrary("ICMP.DLL");
  if (hIcmp == 0)
  {
   cerr << "Unable to locate ICMP.DLL!" << endl;
   return 2;
  }
```

```
// 查找给定机器的 IP 地址信息//
struct hostent* phe;
if ((phe=gethostbyname(argv[1])) == 0)
{
 cerr << "Could not find IP address for " << argv[1] << endl;
 return 3;
}
// 定义函数三个指针类型//
typedef HANDLE (WINAPI* pfnHV)(VOID);
typedef BOOL (WINAPI* pfnBH)(HANDLE);
typedef DWORD (WINAPI* pfnDHDPWPipPDD)(HANDLE, DWORD, LPVOID, WORD,
        PIP_OPTION_INformATION, LPVOID, DWORD, DWORD); // evil, no?
//定义三个指针函数//
pfnHV pIcmpCreateFile;
pfnBH pIcmpCloseHandle;
pfnDHDPWPipPDD pIcmpSendEcho;
//从 ICMP.DLL 中得到函数入口地址//
pIcmpCreateFile = (pfnHV)GetProcAddress(hIcmp,  "IcmpCreateFile");
pIcmpCloseHandle = (pfnBH)GetProcAddress(hIcmp, "IcmpCloseHandle");
pIcmpSendEcho = (pfnDHDPWPipPDD)GetProcAddress(hIcmp, "IcmpSendEcho");
if ((pIcmpCreateFile == 0) || (pIcmpCloseHandle == 0) || (pIcmpSendEcho == 0))
{
 cerr << "Failed to get proc addr for function." << endl;
 return 4;
}
// 打开 ping 服务//
HANDLE hIP = pIcmpCreateFile();
if (hIP == 0)
{
 cerr << "Unable to open ping service." << endl;
 return 5;
}
// 构造 ping 数据包//
char acPingBuffer[64];
memset(acPingBuffer, '\xAA', sizeof(acPingBuffer));
PIP_ECHO_REPLY pIpe = (PIP_ECHO_REPLY)GlobalAlloc( GMEM_FIXED | GMEM_ZEROINIT,
```

```
            sizeof(IP_ECHO_REPLY) + sizeof(acPingBuffer));
  if (pIpe == 0)
  {
   cerr << "Failed to allocate global ping packet buffer." << endl;
   return 6;
  }
  pIpe->Data = acPingBuffer;
  pIpe->DataSize = sizeof(acPingBuffer);
  // 发送 ping 数据包//
  DWORD dwStatus = pIcmpSendEcho(hIP, *((DWORD*)phe->h_addr_list[0]),
          acPingBuffer, sizeof(acPingBuffer), NULL, pIpe,
          sizeof(IP_ECHO_REPLY) + sizeof(acPingBuffer), 5000);
  if (dwStatus != 0)
  {
    cout << "Addr: " <<
              int(LOBYTE(LOWORD(pIpe->Address))) << "." <<
              int(HIBYTE(LOWORD(pIpe->Address))) << "." <<
              int(LOBYTE(HIWORD(pIpe->Address))) << "." <<
              int(HIBYTE(HIWORD(pIpe->Address))) << ", " <<
              "RTT: " << int(pIpe->RoundTripTime) << "ms, " <<
              "TTL: " << int(pIpe->Options.Ttl) << endl;
  }
  else
  {
   cerr << "Error obtaining info from ping packet." << endl;
  }
  GlobalFree(pIpe);
  FreeLibrary(hIcmp);
  return 0;
}
//==================主函数=================//
int main(int argc, char* argv[])
{
  WSAData wsaData;
  if (WSAStartup(MAKEWORD(1, 1), &wsaData) != 0)
  {
```

```
    return 255;
  }
  if (argc < 2)
  {
   cerr << "usage: ping <host>" << endl;
   return 1;
  }
  int retval = doit(argc, argv);
  return retval;
}
```

5、在该工程下，新建一个头文件，即选择“C++ Header File”，文件名：icmpdefs.h；头文件内容如下：

```
//ICMP.DLL 函数中需要的结构//
typedef struct {
  unsigned char Ttl;                          // Time To Live
  unsigned char Tos;                          // Type Of Service
  unsigned char Flags;                        // IP header flags
  unsigned char OptionsSize;                  // Size in bytes of options data
  unsigned char *OptionsData;                 // Pointer to options data
} IP_OPTION_INformATION, * PIP_OPTION_INformATION;
typedef struct {
  DWORD Address;                              // Replying address
  unsigned long  Status;                      // Reply status
  unsigned long  RoundTripTime;               // RTT in milliseconds
  unsigned short DataSize;                    // Echo data size
  unsigned short Reserved;                    // Reserved for system use
  void *Data;                                 // Pointer to the echo data
  IP_OPTION_INformATION Options;              // Reply options
} IP_ECHO_REPLY, * PIP_ECHO_REPLY;
```

6、编译、连接并生成 EXE 程序；

7、进入命令行提示符下运行程序（注意：格式同 Ping 程序，例如：ping oa.jlu.edu.cn）。

【思考问题】

结合实验过程中的实验结果，回答下列问题：

1、根据练习一的实验结果说明：实验室环境所使用操作系统默认的 TTL 最大值是多少？

2、运行 ping 127.0.0.1，再运行 ping 本机名（或本机 IP 地址），在监控机端是否捕获到相应的 ICMP 回显请求报文？

【实验报告要求】

1、说明 ICMP 协议的用途和产生 ICMP 的原因，理解协议主要字段。

2、说明网络中为什么要使用 ICMP 协议？

3、回答每个练习中提出的问题。

【实验五　UDP 用户数据报协议】

【实验目的】

1、理解 UDP 协议的工作原理及工作过程；

2、掌握 UDP 的数据包格式；

3、了解基本的端口分配。

【实验学时】

4 学时

【实验类型】

综合型

【实验内容】

1、学习 UDP 协议的通信过程；

2、学会手工计算 UDP 校验和；

3、初步学会应用 C 语言与 WinSock2 进行网络程序设计。

【实验原理】

UDP 是面向非连接的协议，发送端与接收端在传输数据包之前不建立连接，而只是简单地把数据包发送到网络上，或者从网络上接收数据包。UDP 提供不可靠的数据传输服务。

1、UDP 封装在 IP 数据报

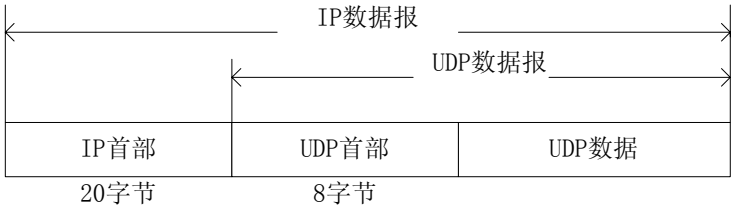


图 5-1　UDP 的封装

2、UDP 字段组成

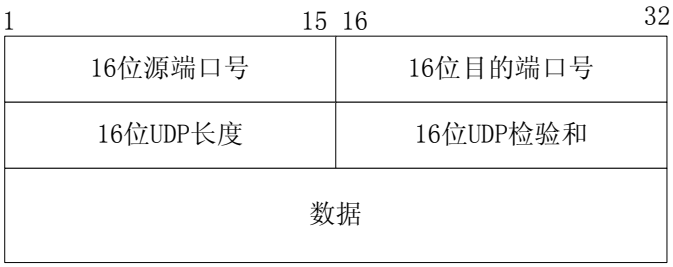


图 5-2　UDP 分组格式

3、关于 UDP“校验和”的计算方法

尽管 UDP 校验和的基本计算方法与 IP 首部“校验和”的计算方法类似（16bit 的二进制反码和），但是它们之间存在不同的地方。

首先，UDP 数据报长度可以为奇数字节，但是“校验和”的算法是把若干个 16bit 字相加。解决方法是必要时在最后增加填充字节 0，这只是为了“校验和”的计算（也就是说，可能增加的填充字节不被传送）。

其次，UDP 数据报包含一个 12 字节长的伪首部，它是为了计算“校验和”而设置的。伪首部包含 IP 首部一些字段。其目的是让 UDP 两次检查数据是否已经正确到达目的地。发送时不用传送伪首部，但接收时将 IP 头中的信息构建伪首部，然后计算“校验和”用以判错。

UDP“校验和”是一个端到端的“校验和”。它由发送端计算，然后由接收端验证。其目的是为了发现 UDP 首部和数据在从发送端到接收端之间发生的任何改动。

UDP 数据报中的伪首部格式如图 5-3：



图 5-3　UDP 分组的伪首部

【实验步骤】

练习一：编辑并发送 UDP 数据包

1、打开仿真编辑器，在工具栏中单击 UDP 协议。

2、填写该帧的以太网协议首部：

⑴源 MAC 地址：发送方的 MAC 地址；

⑵目的 MAC 地址：接收方的 MAC 地址；

注意:当选择的目的主机与本机不在同一子网的情况，目的 MAC 地址该怎样填写？

⑶类型或长度：该字段值应为 0800（即 IP 协议的类型值）。

3、填写 IP 协议头信息：

⑴高层协议类型：即上层协议类型，应为 17（UDP 协议的类型为 17）；

⑵总长度：应包括 UDP 长度；

⑶填写源、目的 IP 地址；

⑷计算首部校验和。

4、填写 UDP 协议信息，见图 5-4：

⑴源端口：自选；

⑵目的端口：自选（对于本实验，选择大于 1024）；

⑶UDP 长度：UDP 报头长度（8 个字节）与 UDP 数据长度之和；

⑷校验和：覆盖 UDP 首部（含伪首部）和数据两部分；

⑸数据：可有可无；

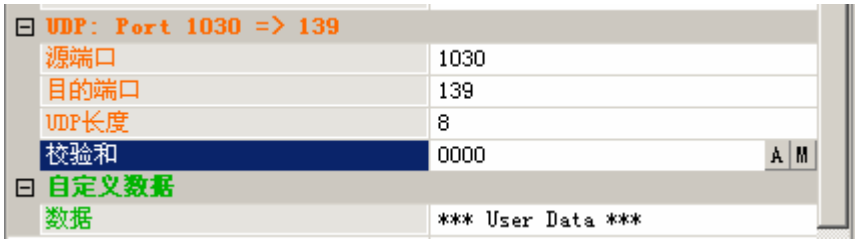


图 5-4　填写 UDP 协议信息

⑹校验和：

方法一：自动计算

在校验和中先填入一个非 0 的数，单击 A 按钮，实现自动填充（UDP 的校验和可以为 0，所以自动计算时应先填入一非 0 数）。

方法二：手动计算

步骤为：

1. 将 UDP 的校验和置为 0；
2. 单击 M 按钮，弹出校验和对话框（在此填充的数据均为十六进制，不足两个字节按双字节补齐）；
3. 在协议字段填入 0011（UDP 协议为 17）；
4. 以后各项对应的十六进制数，均可在十六进制对照表中找到，右键复制粘贴到对应的文本框中；
5. 单击计算按钮得出校验和。

如果要发送多个 UDP 数据报，可重复上述步骤，并可在多帧编辑器中设置每一帧的发送次数和发送每帧的时间间隔。

5、 点击工具栏或菜单栏中的“发送”，在弹出对话框中选择“发送”按钮，发送 UDP 报文；

6、在协议分析器一端截获相应的报文并分析，注意观察“会话分析”中的 UDP 会话过程。

7、将编辑正确的 UDP 数据报保存到文件：UDP.pkt。

练习二：编辑并发送 UDP 广播包

1、实验步骤同上；注意：以太网协议首部中的目的 MAC 地址应怎样填写？IP 首部中的目的 IP 地址该怎样填写？

2、在协议分析器一端截获相应的报文并分析。

练习三：运行 netstat 命令

netstat 命令是用于显示网络使用协议的统计；

1、在命令行方式下运行：

netstat －s　　；显示每个协议的使用状态，观察 UDP 协议的使用情况。

2、在命令行方式下运行：

netstat －a　　；显示所有主机的端口号，观察与 UDP 协议相关的内容。

练习四：VB 编程练习

应用 VB 开发环境和 Winsock 控件，编制简单的点对点聊天程序。通过练习四，使学生能了解 VB 网络编程的基础。本实验要求一个同学在仿真编辑器上，另一个同学在协议分析器上，同时编程实现双机通信。

步骤如下：

1、启动 VB 开发环境，新建标准 EXE 文件，工程 1--Form1；

2、添加 WINSOCK 控件，方法是：工程→部件，选择 Microsoft Winsock control 6.0，按确定，

工具箱里会增加一个计算机网络的图标；

3、将 Winsock 图标添加到 FORM1 中，另外添加两个 text，text1 是接收窗口，text2 是发送窗口；添加两个 command 按钮；

4、修改 command1 的 caption 属性为 send，修改 command2 的 caption 属性为 clear,修改 winsock 的 LocalPort=2000，RemoteHost=对方的 IP 地址，RemotePort=2000，修改 protocol 属性为 1-sckUDPProtocol。见图 5-5 所示；

5、对方计算机同样作以上内容，不同的是 RemoteHost IP 地址；

6、双击 send，在 Private Sub Command1_Click()和 End Sub 之间填写下列代码：

Winsock1.SendData Text2.Text

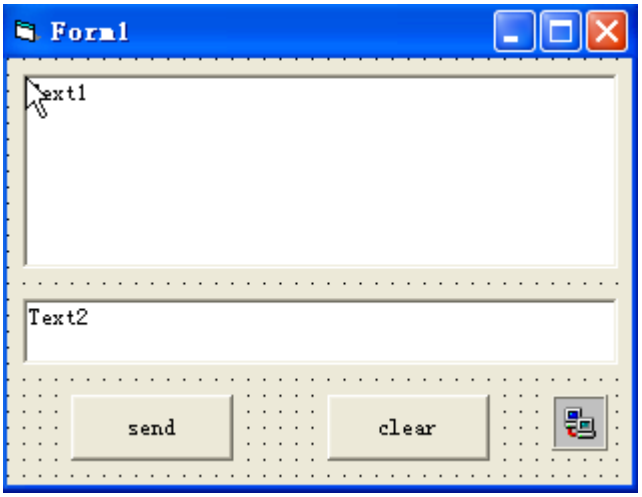


图 5-5　简单的 VB 聊天程序界面

7、双击 clear，在 Private Sub Command2_Click()和 End Sub 之间填写下列代码：

```
Text1.Text = ""
Text2.Text = ""
```

8、双击 form1 窗体，在 Private Sub Form_Load()和 End Sub 之间填写下列代码：

```
Winsock1.Bind 2000
```

9、双击 form1 窗体，进入代码窗体，在左上角的控件表中选择 winsock1，在右上角的子程序中选择 DataArrival，在代码窗体中会生成一个子程序：Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)，在这个子程序中添加以下代码：

```
Dim sss As String
Winsock1.GetData sss
Text1.Text = sss
```

10、点击工具栏上的三角号，运行 form1，两个计算机之间就可以聊天了！

11、双方在 text2 文本框中随便输入什么会在对方的 text1 文本框中出现。

12、对方程序未运行时，发送数据会成功吗？为什么？

13、在协议分析器端截获数据包，分析结果。

【思考问题】

1、协议分析器一端截获网络中的多个广播报文，观察其源 MAC 地址及源 IP 地址，是否都来自同一子网？为什么？

2、为什么 UDP 的“校验和”要包含伪首部？

【实验报告要求】

1、说明 UDP 协议的工作原理。

2、解释 UDP 的各字段功能。

3、回答练习中提出的问题。

【实验六　TCP 传输控制协议】

【实验目的】

1、掌握 TCP 协议的工作原理；

2、理解 TCP 协议的通信过程。

【实验学时】

4 学时

【实验类型】

验证型

【实验内容】

1、理解 TCP 首部中各字段的含义及作用；

2、理解三次握手的过程；

3、能够分析 TCP 协议的建立连接、会话和断开连接的全过程；

4、学会计算 TCP 校验和的方法；

5、了解 TCP 的标志字段的作用。

【实验原理】

TCP 协议是面向连接的、端到端的可靠传输协议，它支持多种网络应用程序。TCP 必须

解决可靠性，流量控制的问题，能够为上层应用程序提供多个接口，同时为多个应用程序提供数据，TCP 也必须能够解决通信安全性的问题。

1、TCP 的封装

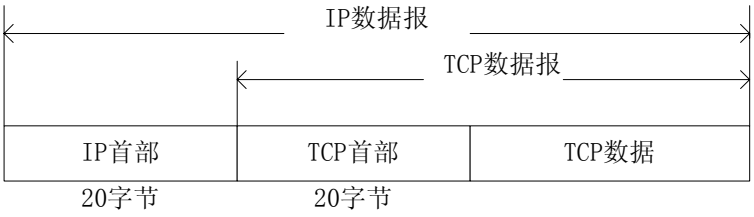


图 6-1　TCP 的封装

2、TCP 首部格式：

各字段含义：

16 位源端口号和 16 位目的端口号：端口号通常也称为进程地址。

32 位序号：序号用来标识从 TCP 发送端向 TCP 接收端发送的数据字节流。

32 位确认序列号：表示一个准备接收包的序列号。

4 位首部长度：首部长度指出了首部中 32 bit 字的数目。正常的 TCP 首部长度是 20 字节。

6 个标志字段：URG 紧急指针；ACK 确认序号；PSH 推标志；RST 重建连接；SYN 同步序号；FIN 结束标志。

16 位窗口：TCP 的流量控制由连接的每一端通过声明的滑动窗口大小来提供，窗口大小为字节数。

16 位检验和：检验和字段覆盖了 TCP 首部和 TCP 数据。TCP 检验和的计算方法和 UDP 检验和的计算方法一样，计算时需要考虑伪报头。

16 位紧急指针：URG 标志置 1 时紧急指针才有效。

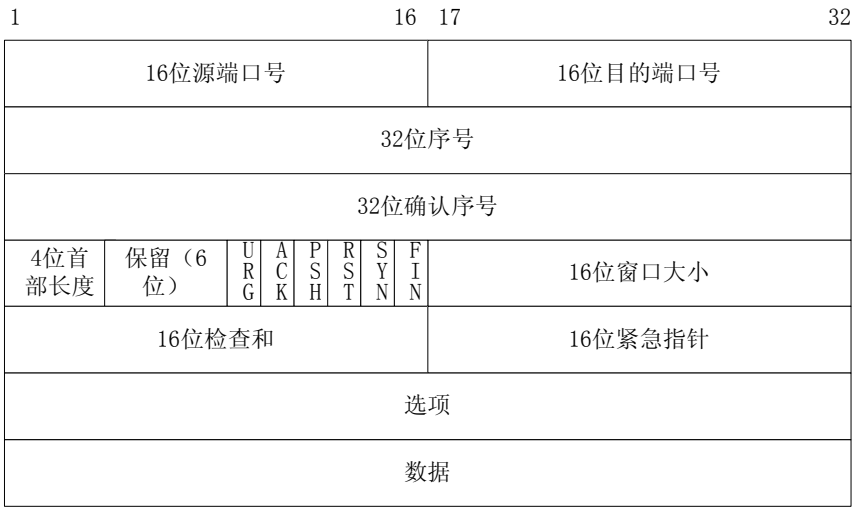


图 6-2　TCP 分组格式示意图

【实验步骤】

注意：本实验要求服务器端启动 TELNET 服务。

练习一：运行 netstat 命令

1、在浏览器的地址栏输入：www.jlu.edu.cn，在命令行下输入：netstat -n 回车 。
2、观察 TCP 状态，记录 Local Address、Foreign Address 和 State。
3、在浏览器的地址栏输入：www.baidu.com，在命令行下输入：netstat -n 回车 。
4、观察 TCP 状态，记录 Local Address、Foreign Address 和 State。
5、再开一个命令行窗口，输入：telnet bbs.jlu.edu.cn。
6、观察 TCP 状态，记录 Local Address、Foreign Address 和 State。
7、三次记录有什么不同，为什么？
8、使用带其他参数的 netstat 命令，查看网络状态。例如：

```
netstat -a
netstat -b
```

练习二：利用仿真编辑器编辑并发送 TCP 数据包

说明：

●在本实验中由于 TCP 连接有超时时间的限制，故仿真编辑器端和协议分析器端的两位同学要默契配合，某些步骤（如计算 TCP 校验和）要求熟练、迅速。

●为了实现 TCP 三次握手过程的仿真，发送第一个连接请求帧之前，应该暂停仿真编辑器端的 TCP/IP 协议，否则计算机系统的网络会对该请求帧的应答帧发出拒绝响应。具体操作为在仿真端主机上运行“开始/程序/网络协议仿真教学系统 基本版/TCP 过滤/启动过滤”。

●通过手工编辑 TCP 数据包实验，要求理解实现 TCP 连接建立、数据传输以及断开连接的全过程。在编辑的过程中注意体会 TCP 首部中的序列号和标志位的作用。

首先选择服务器主机上的一个进程作服务器进程，并向该服务器进程发送一个建立连接请求报文,对应答的确认报文和断开连接的报文也编辑发送。其步骤如下：

1、打开仿真编辑器，单击工具栏中的 TCP 协议选项；
2、填写该帧的以太网协议首部：
⑴源 MAC 地址：本机 MAC 地址；
⑵目的 MAC 地址：服务器的 MAC 地址；
3、填写 IP 协议头信息：
⑴协议字段：协议类型值为 6，即上层协议类型应为 TCP；
⑵总长度： IP 头部+TCP 头部+数据长度；
⑶首部校验和：应用前面学到的知识进行计算；
⑷源 IP 地址：本机 IP 地址；
⑸目的 IP 地址：服务器的 IP 地址；

⑹其它字段默认；

4、填写 TCP 协议信息：

⑴源端口：任意大于 1024 的数，不要用下拉选择；

⑵目的端口：可以设 80（HTTP 协议）；

⑶序号：选择一个序号 ISN（假设 1942589885），以后的数据都按照这个来填写；

⑸首部长度和标志位：首部长度和标志位设为 5002（即长度 20 字节，标志 SYN=1）；

⑹窗口大小：任意；

⑻校验和：覆盖 TCP 首部（包含伪首部）和 TCP 数据两部分，计算方法同 UDP 协议一样。值得注意的是伪首部中的协议号是 0006，而不是 UDP 时的 0011；

⑼将设置完成的数据帧复制 3 份；修改第二帧的 TCP 首部长度和标志位为 5010（即标志位 ACK=1），TCP 层序号为 1942589885+1；修改第三帧的 TCP 首部长度和标志位为 5011（即标志位 ACK=1、FIN=1），TCP 层序号为 1942589885+1；

5、在发送该 TCP 连接请求之前，先运行一次 ping 目标服务器，目的是让目标服务器知道自己的 MAC；

6、在工具菜单栏中选择 TCP 三次握手下的启动过滤，以暂停 TCP/IP 协议；

7、点击菜单栏中的“发送”，在弹出对话框中选择发送第一个帧；

8、在协议分析器一端截获相应的应答报文，可以设置协议端口（80）过滤。这里要求协议分析器一端的同学及时准确地捕获应答报文并迅速从中获得应答报文的接收字节序号，并告之仿真编辑器一端的同学；

9、我们假设接收字节序号为：3246281765，修改第二帧和第三帧的 TCP 应答序号中的值：3246281766；

10、计算第二帧的 TCP 校验和，发送；对服务器的应答报文进行确认；

11、计算第三帧的 TCP 校验和，发送；断开连接，完成 TCP 连接的全过程；

12、协议分析器一端截获相应的请求及应答报文并分析，注意观察“会话分析”中的会话过程。

注意：实验结束后启动仿真编辑器端的 TCP/IP 协议。

具体步骤是：工具—>TCP 三次握手→暂停过滤。

练习三：使用 TCP 连接客户端工具测试各种 TCP 连接建立和断开的过程

1、首先选择一个 **TELNET 服务器**，并建立一个 TCP 连接，步骤如下：

⑴选择一个 TELNET 服务器，比如服务器 IP 为 172.16.1.177，启动协议分析器的捕获功能；

⑵在仿真编辑器的 TCP 连接试图中，填写所选择服务器的 IP 地址及服务器进程的端口号，选择 23 端口，点击“连接”，建立本机与 TELNET 服务器的 TCP 连接； 在 TCP 连接客户端程序的返回窗口中会有返回信息；

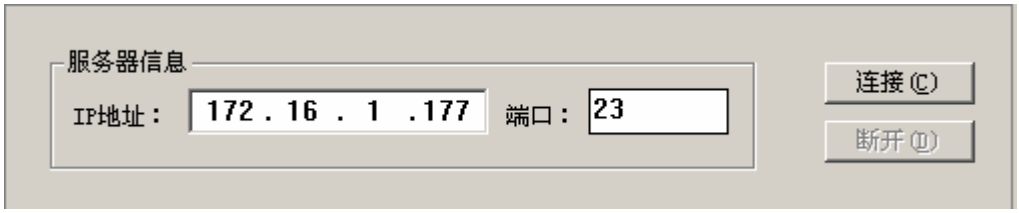


图 6-3　TCP 连接视图界面

2、断开连接：

⑴点击 “断开”按扭，即可断开与服务器端的连接；

⑵在协议分析器一端刷新截获报文，从“会话分析”中找出此连接的三次握手、服务器返回的信息、拆除连接的几个数据包；

3、由 1 步和 2 步，画出 TELNET 连接建立、拆除连接的会话过程，标记出每步的序号、确认序号、SYN、ACK、FIN 等标志位。

4、选择一个 **WWW 服务器**，并建立一个 TCP 连接，步骤如下：

⑴在地址本中选择扫描到的服务器，点击端口扫描按钮，开始扫描该服务器开放的端口。

⑵双击该主机 IP，将其 IP 填加到服务器信息 IP 地址中；

⑶双击端口号 80，也会自动填加到服务器信息端口中（也可以手动输入）；

⑷单击“连接”按钮，建立连接；



图 6-4 TCP 连接视图界面

5、断开连接：

⑴单击 “断开”按扭，即可断开与服务器端的连接；

⑵在协议分析器一端截获报文，从 HTTP 会话中找出此连接的三次握手、拆除连接的几个数据包；

6、由 4 和 5 步，画出 HTTP 连接建立、拆除连接的过程，标记出每步的序号、确认序号、SYN、ACK、FIN 等标志位。

7、选择一个 **FTP 服务器**，并建立一个 TCP 连接，步骤如下：

⑴选择一个 FTP 服务器，比如：实验室的 FTP 服务器（IP=172.16.1.177）；

⑵填写所选择服务器的 IP 地址及服务器进程的端口号 21，点击“连接”，建立本机与 FTP 服务器的 TCP 连接；



图 6-5 TCP 连接视图界面

8、断开连接：

⑴点击“断开”按扭，即可断开与服务器端的连接；

⑵在协议分析器一端截获报文，从“会话分析”中找出此连接的三次握手、服务器返回的信息、拆除连接的几个数据包；

9、由 7 和 8 步，总结 FTP 连接建立的过程和拆除连接的过程，记录序号、确认序号、SYN、ACK、FIN 、PSH 等标志位。

10、上述 3 个 TCP 连接操作的过程可以用下图描述，写出每一步的序号和确认序号及标志位变化情况。

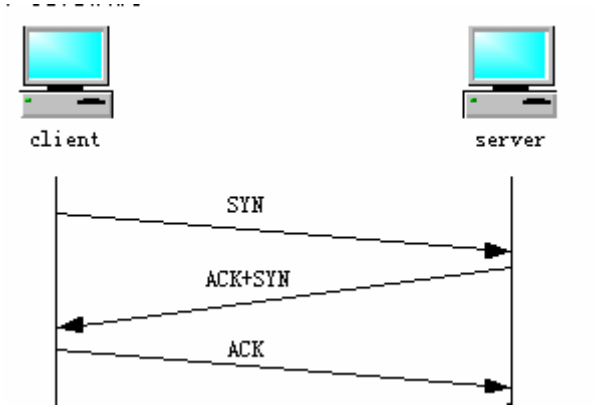


图 6-6 TCP 连接过程的示意图

练习四：多 TELNET 客户端同时运行

1、在仿真编辑器端同时打开两个或更多命令提示符窗口，在这些窗口上同时运行：

C：>telnet 服务器 IP 地址

2、在协议分析器一端捕获数据，分析仿真编辑器发送和接收的 TCP 数据包。说明哪些包是属于同一 TCP 连接上的，你是根据什么进行区分的？

练习五：察看会话状况

1、在协议分析器的 “会话分析”视图窗口中，查看多个 TCP 连接，并与自己的分析结果进行比较，注意观察标志位。

2、捕获大量数据包，分析其中完整的 TCP 会话。

【思考问题】

结合实验过程中的实验结果，回答下列问题

1、如果在练习二发送建立连接请求帧之前，仿真编辑器端不运行“ping 服务器”命令，而又停止了仿真编辑器端的 TCP/IP 协议，可能出现什么情况？通过实验来验证你的答案。

2、根据练习五说明：在仿真编辑器端，系统是如何把接收到的数据显示到相应的窗口的。

3、根据实验中窗口的变化和确认序号，分析 TCP 协议所使用的滑动窗口协议机制。

【实验报告要求】

1、说明 TCP 协议三次握手的工作过程和拆除连接的过程。

2、回答练习中提出的问题。

3、提出新的 TCP 协议仿真实验的方法。

【实验七　TCP 与 UDP 端口扫描】

【实验目的】

1. 了解常用的 TCP、UDP 端口扫描的原理及其各种手段；
2. 增强网络安全意识。

【实验学时】

4 学时

【实验类型】

验证型

【实验内容】

1、学习 TCP Connect 端口扫描方法；

2、学习 TCP SYN 端口扫描方法；

3、学习 UDP 端口扫描方法。

【实验原理】

一. TCP/UDP 报文头格式

UDP 报文格式见实验六(用户数据报协议 UDP)，TCP 报文格式见实验七(传输控制协议 TCP)实验原理。在 TCP 首部中 6 个标志位的用法依次为：

URG 紧急指针(urgent pointer)有效。

ACK 确认序号有效。

PSH 接收方应该尽快将这个报文段交给应用层。

RST 重建连接。

SYN 同步序号用来发起一个连接。

FIN 发端完成发送任务。

二. ICMP 端口不可达报文

ICMP 端口不可达报文格式见实验四(Internet 控制报文协议 ICMP)实验原理。 类型域为 3，代码域为 3 表示该 ICMP 报文为端口不可达报文，它是 ICMP 目的不可达报文的一种。

三. TCP connect() 扫描

这种方法最简单，直接连到目标端口并完成一个完整的三次握手过程(SYN, SYN/ACK, 和 ACK)。操作系统提供的“connect（）”函数完成系统调用，用来与每一个感兴趣的目标计算机的端口进行连接。如果端口处于侦听状态，那么“connect（）”函数就能成功。否则，这个端口是不能用的，即没有提供服务。这个技术的一个最大的优点是不需要任何权限，系统中的任何用户都有权利使用这个调用。另一个好处是速度。如果对每个目标端口以线性的方式，使用单独的“connect（）”函数调用，那么将会花费相当长的时间，你可以通过同时打开多个套接字，从而加速扫描。使用非阻塞 I/O 允许你设置一个低的时间用尽周期，同时观察多个套接字。但这种方法的缺点是很容易被发觉，并且很容易被过滤掉。目标计算机的日志文件会显示一连串的连接和连接时出错的服务消息，目标计算机用户发现后就能很快使它关闭。

四. TCP SYN 扫描

这种技术也叫“半开式扫描”（half-open scanning），因为它没有完成一个完整的 TCP 协议连接。这种方法向目标端口发送一个 SYN 分组（packet），如果目标端口返回 SYN/ACK 标志，那么可以肯定该端口处于检听状态；否则，返回的是 RST/ACK 标志。这种方法比第一种更具隐蔽性，可能不会在目标系统中留下扫描痕迹。但这种方法的一个缺点是，必须要有 root 权限才能建立自己的 SYN 数据包。

五. UDP 端口扫描

这种方法向目标端口发送一个 UDP 协议分组。如果目标端口以“ICMP port unreachable”消息响应，那么说明该端口是关闭的； 反之，如果没有收到“ICMP port unreachable”响应消息，则可以肯定该端口是打开的。由于 UDP 协议是面向无连接的协议，这种扫描技术的精确性高度依赖于网络性能和系统资源。另外，如果目标系统采用了大量分组过滤技术，那么 UDP 协议扫描过程会变得非常慢。如果你想对 Internet 进行 UDP 协议扫描，那么你不能指望得到可靠的结果。

【实验步骤】

练习一：TCP Connect 扫描

1. 在分析端上启动协议分析器进行数据捕获并设置过滤条件(提取 TCP 协议)。
2. 在分析端上使用命令“netstat -a”，显示本机可用的 TCP、UDP 端口。
3. 在仿真端上使用 TCP 连接工具，扫描分析端的某一端口：
   a. 分析端将在步骤 2 得到的某一 TCP 端口告知仿真端。
   b. 仿真端上填入分析端的 IP、被告知的端口号，点击“连接”按钮进行连接。
   c. 观察提示信息，是否连接上。
   d. 分析端将在步骤 2 没有得到的某一 TCP 端口告知仿真端。
   e. 重复 b、c。
4. 察看分析端捕获的数据，分析连接成功与失败的数据包差别。
   - 结合捕获数据的差别，说明 TCP Connect 扫描的实现原理。

练习二：TCP SYN 扫描

1. 在分析端上启动协议分析器进行数据捕获并设置过滤条件(提取 TCP 协议)。
2. 在分析端上使用命令“netstat -a”，显示本机可用的 TCP、UDP 端口。
3. 仿真端编辑 TCP 数据包：
   a. 源 MAC、IP 地址为本机 MAC、IP。
   b. 目的 MAC、IP 地址为分析端的 MAC 和 IP。
   c. 源端口为本机可用端口。
   d. 目的端口为分析端在步骤 2 得到的 TCP 可用端口。
   e. 标志 SYN 置为 1，其他标志置为 0。
4. 发送仿真端编辑好的数据包。
5. 修改仿真端编辑的数据包(将目的端口置为分析端上非可用的 TCP 端口)，将其发送。
6. 察看分析端捕获的数据，找到仿真端发送的两个数据包对应的应答包。
   - 分析两个应答包的不同之处，说明 TCP SYN 扫描的实现原理。

练习三：UDP 端口扫描

1. 在分析端上启动协议分析器进行数据捕获并设置过滤条件(提取仿真端的 IP←>分析端的 IP)。
2. 在分析端上使用命令“netstat -a”，显示本机可用的 TCP、UDP 端口。
3. 仿真端编辑 UDP 数据包：
   a. 源 MAC、IP 地址为本机 MAC、IP。
   b. 目的 MAC、IP 地址为分析端的 MAC、IP。
   c. 源端口为本机可用端口。
   d. 目的端口为分析端在步骤 2 得到的 UDP 可用端口。
4. 发送仿真端编辑好的数据包。
5. 修改仿真端编辑的数据包(将目的端口置为分析端上非可用的 UDP 端口)，将其发送。

6. 察看分析端捕获的数据。
   - 仿真端发送的数据包，哪个收到目的端口不可达的 ICMP 数据报。
   - 说明这种端口扫描的原理。
   - 使用这种端口扫描得到的结果准确吗？说明理由。

【思考问题】

根据各种端口扫描的原理，设计出不同端口扫描的预防措施。

【实验八　TELNET 和 FTP 协议】

【实验目的】

1、理解 TELNET、FTP 协议；
2、了解应用层协议与传输层协议的关系。

【实验学时】

4 学时

【实验类型】

验证型

【实验内容】

1、学习 TELNET 协议及作用；
2、学习 TELNET 的使用方法；
3、学习 FTP 工作原理和使用方法。

【实验原理】

1、TELNET 协议

TELNET 是一个简单的远程终端协议。用户可以通过 TCP 连接登录到远程的一个主机上，好象使用远程主机一样。

TELNET 采用客户机/服务器计算模式。在本地系统上运行 TELNET 客户机进程，在远程主机上运行 TELNET 服务器进程。

TELNET 一般用于远程管理一台主机，例如：服务器、路由器和交换机等。

2、FTP 协议

FTP 文件传输协议是 Internet 网络上传输文件的通用方法。FTP 位于 TCP/IP 模型的应用层。FTP 的下层使用 TCP 服务。FTP 是一个客户机/服务器系统。用户通过一个支持 FTP 协议的客户机程序，连接到在远程主机上的 FTP 服务器程序。用户通过客户机程序向服务器程序发出命令，服务器程序执行用户所发出的命令，并将执行的结果返回到客户机。当 FTP 客户端与服务器建立 FTP 连接时，将与服务器上的两个端口建立联系：端口 20 和 21。FTP 使用不同的端口号传输不同的内容，见下面的流程：

```
[ ftp Client ]                          [ ftp Server ]
Port xxxx ----- |SYN|-----------------> Port 21 [TCP]
Port xxxx <-----|SYN+ACK | ------------ Port 21
Port xxxx ----- |ACK|-----------------> Port 21 [TCP]

……  控制操作: 用户列目录或传输文件……
Port xxxx ----- |ACK+PORT 命令|--------> Port 21 [TCP]
Port yyyy <-----|SYN|------------------  Port 20
  Port yyyy ----|SYN+ACK|-------------> Port 20
  Port yyyy <---| ACK |---------------  Port 20
```

具体的 FTP 工作流程请在实验中仔细的记录。

FTP 的命令可以在命令提示符下键入：

ftp 10.60.48.200（实验室FTP服务器的IP地址）

用户被提示输入用户名和口令（用匿名登入），确认口令输入正确后，用户将获得FTP提示符“ftp>”。

FTP常用的命令：

```
Binary              ；将文件传输模式设为二进制方式
Bye                 ；退出，结束远程连接
Case                ；打开case功能
Close               ；关闭远程连接
cd<directory>       ；改变远程系统目录名
cd up               ；进入父目录
debug <level>       ；设置调试级别
dir                 ；打印路径名
delete <filename>   ；在远程计算机中删除文件
get <filename>      ；将文件从远程系统下载到本地系统
glob                ；在文件传输时使用通配符
hash                ；每传输1 0 2 4字节打印一个“#”
help                ；打印帮助文件
lcd <directory>     ；改变本地系统路径
```

ls	；打印远程主机目录
lpwd	；打印本地主机工作路径
mget <filenames>	；将多个文件从远程系统下载到本地系统
mkdir <directory>	；在远程计算机上创建路径
mput <filenames>	；将多个文件上载到远程计算机
prompt	；交互式提示
put <filename>	；将文件上载到远程计算机
pwd	；显示当前工作路径
user	；登录到远程计算机系统

【实验步骤】

本实验要求：

1、服务器端已经启动 telnet 服务，并且提供一个公共账号：guest,口令：guest；

2、也可以利用校园网的 BBS 作为服务主机。

练习一：运行 TELNET 命令，捕获数据并分析

1、实验环境中的服务器(假设 IP 地址：172.16.1.177)上的 telnet 服务已经启动，并且提供一个公共帐号，其用户名:guest；口令：无；

在命令行提示符下运行：

⑴telnet 服务器 IP（例如：telnet 172.16.1.177）；

⑵在“Login:”提示符后输入用户名（guest），然后在“Password:”提示符后输入口令（无）；

⑶在虚拟终端上进行一些简单的操作（可不作）；

⑷按“CTRL+]”回到 telnet 提示符下；

⑸输入“quit”退出 telnet。

2、在协议分析器端捕获数据，从协议的角度分析该次 telnet 操作的完整过程。

⑴在协议分析器端捕获数据，查找用户名和密码。是否可以捕获到其它网段上的信息？

⑵仿真编辑器端在 TELNET 状态下，运行各种操作，在协议分析器端上是否可以捕获到数据，理解 TELNET 明文传输的不安全性，了解网络监听的可能。

练习二：FTP 工作流程

1、设置协议分析器端过滤器，按 TCP 协议和 MAC 地址过滤。源 MAC 是本机，目的 MAC 是 FTP 服务器，即本实验室的 FTP 服务器（172.16.1.177）。启动协议分析器捕获；

2、登录 FTP 服务器：

在实验环境中的 FTP 服务器（172.16.1.177）已经启动，并提供一个公共帐号，用户名是：anonymous，口令：无；或用户名：demo，口令：demo。

在命令行提示符下运行：

⑴C：>ftp 172.16.1.177

⑵在“LOGIN:”提示符后输入用户名：Anonymous；

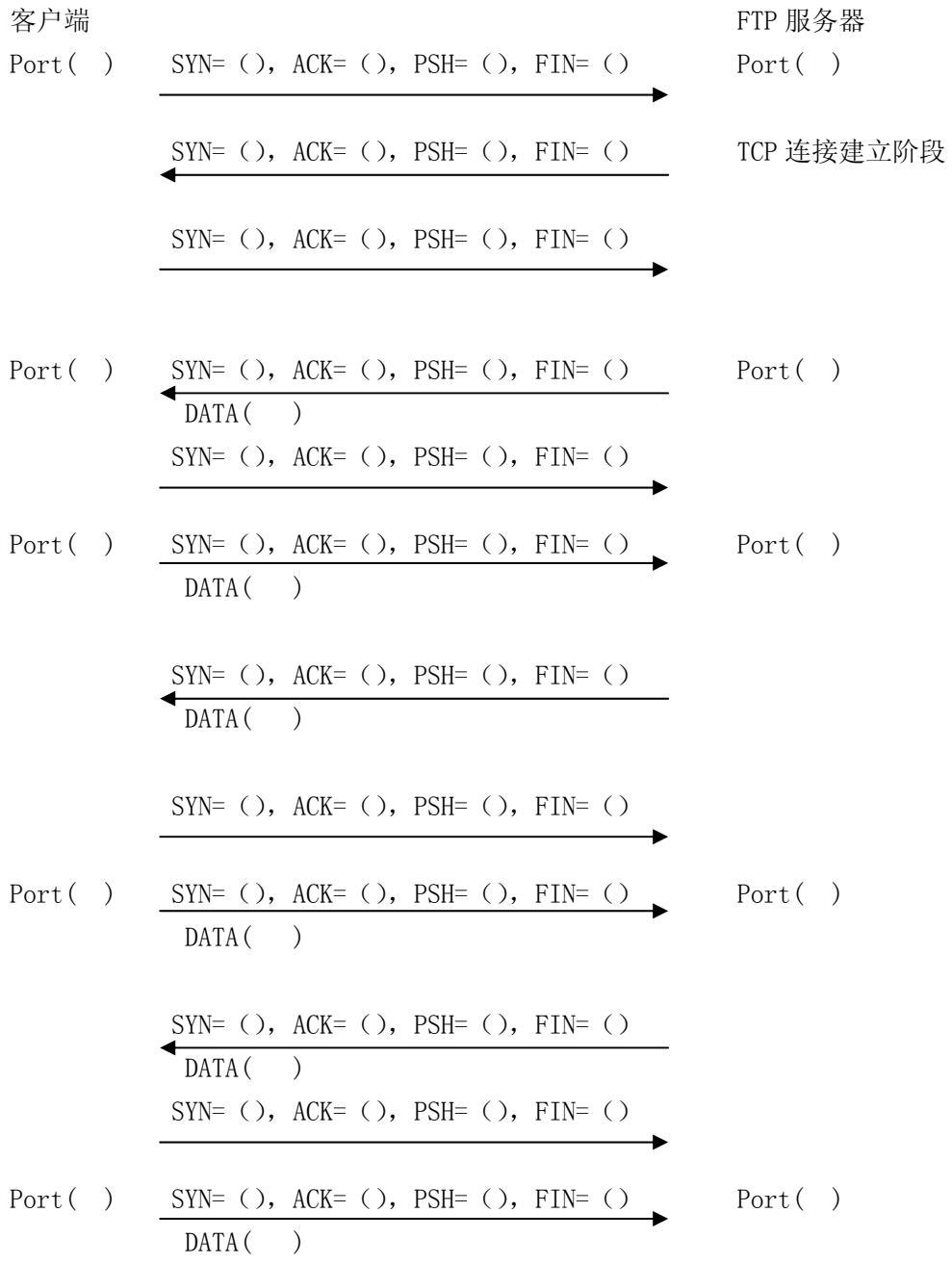
⑶在“PASS” 提示符后输入密码：无；

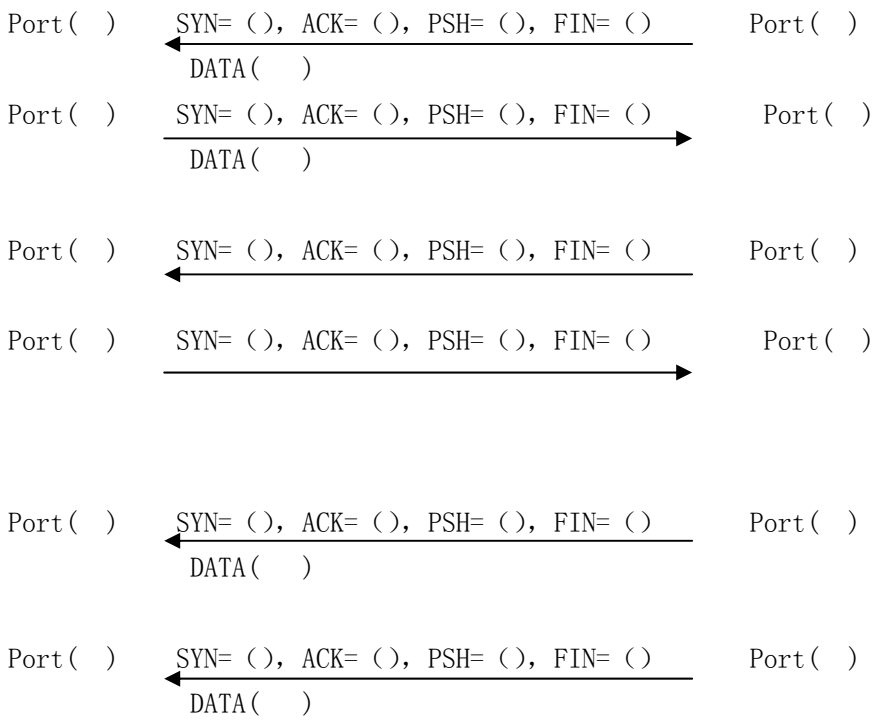
⑷在客户端上运行一个简单的操作，如：

ftp> dir

⑸在 FTP 提示符下输入“quit”退出 FTP。

3、暂停协议分析器的捕获，刷新显示，在协议分析器端已捕获的数据中，对该过程中仿真编辑器发送和接收的全部 TCP 数据包记录：





下面应该是传输数据的一些数据包。

4、根据实验结果总结：

⑴FTP 工作时客户端和服务器的两个端口工作过程。

⑵根据标志字段分析 FTP 的两个端口的连接建立、会话和断开连接的全部过程。

⑶分析 FTP 的工作过程。

⑷分析该过程中的源、目的端口号。

⑸是否可以捕获到用户名和密码？说明 FTP 的安全性？

练习三：使用 TCP 连接客户端工具

1、首先选择一个 FTP 服务器，并建立一个 TCP 连接：

⑴选择一个 FTP 服务器，比如：172.16.1.177，或其他的 FTP 服务器；

⑵在仿真编辑器端，打开 TCP 连接视图，填写所选择的 FTP 服务器的 IP 地址及 FTP 服务器进程的端口号 21，然后点击“连接”，建立与 FTP 服务器的 TCP 连接；

⑶连接成功，在接收窗口会显示成功连接的信息；若不成功，再次尝试进行连接，直到成功。如图：

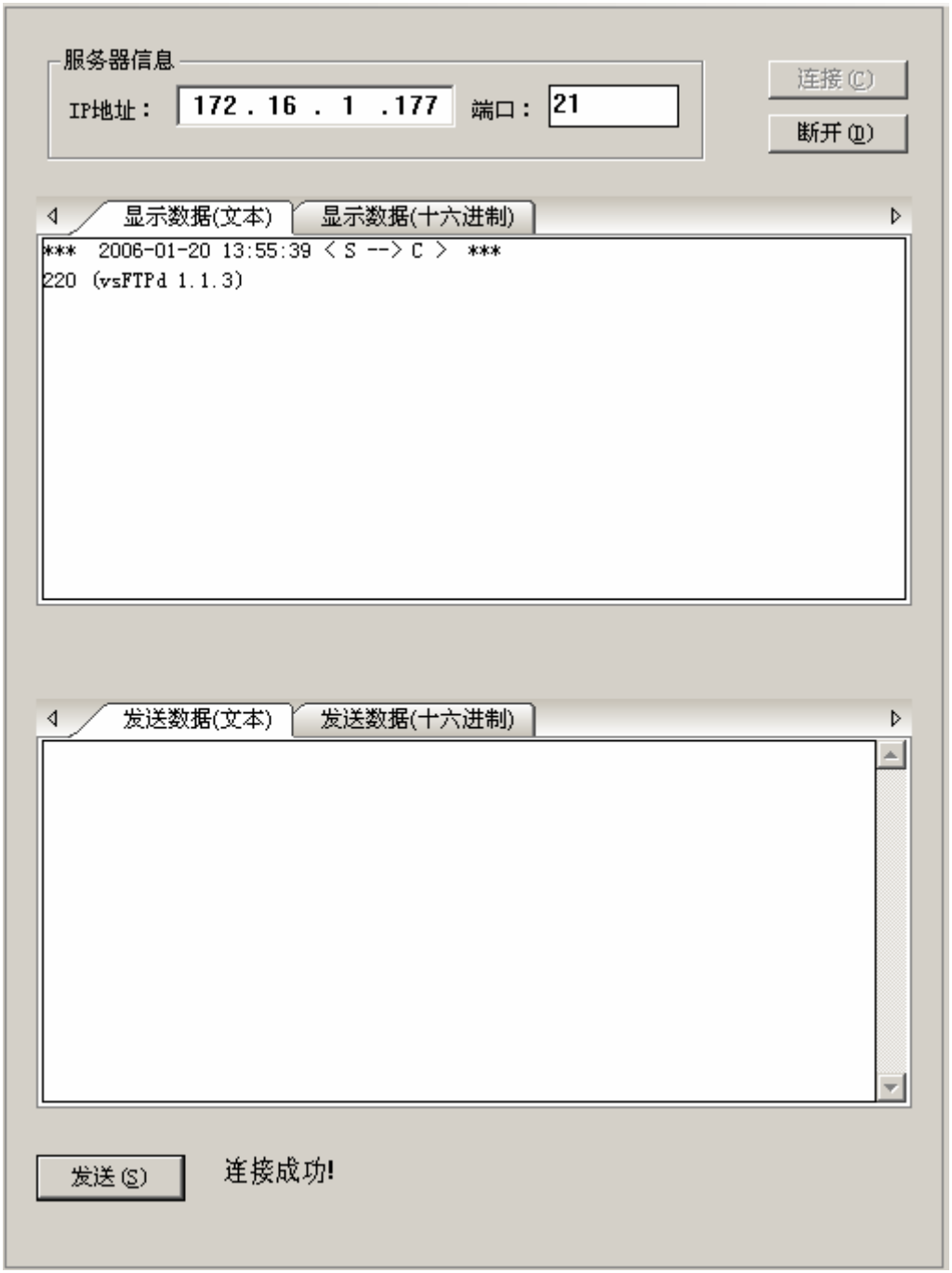


图 8-1　TCP 连接客户端工具界面

2、在发送区输入下列命令：

USER username〈CRLF〉　　点击“发送”；

PASS password〈CRLF〉　　点击“发送”；

进入 FTP 服务器，使用下列命令，观察结果；

NOOP〈CRLF〉　　点击“发送”；　　（空操作）

HELP〈CRLF〉　　点击“发送”；　　（命令帮助）

STAT〈CRLF〉　　点击“发送”；　　（当前连接状态）

等等；

3、断开连接：

⑴点击 TCP 连接视图上的“断开”按扭，即可断开与服务器端的连接。

⑵在协议分析器端截获报文，分析连接建立和拆除连接的过程，记录标志位的变化。

【思考问题】

1、通过实验说明你的 TELNET 帐户密码在网络上传输是安全的吗？

2、在 ACTIVE 模式的 FTP 中，服务器端为何使用 20 和 21 两个端口，其作用分别是什么？客户端与这两个端口建立起的连接分别是谁（客户端还是服务器端）主动发起的？

【实验报告要求】

1、说明 TELNET、FTP 协议的使用，分析 TELNET 和 FTP 的安全性。

2、画出 FTP 连接、传输和拆除连接的工作流程。

3、回答练习中提出的问题。

【实验九　DNS 域名服务协议】

【实验目的】

1、理解 DNS 实现的原理；

2、掌握 DNS 报文格式。

【实验学时】

4 学时

【实验类型】

综合型

【实验内容】

1、学习 DNS 协议的原理和实现方法；

2、学习 DNS 的报文格式；

3、通过编辑 DNS 请求数据包，了解响应包的格式。

【实验原理】

1、DNS 协议

DNS 域名系统是服务器和客户程序相互通信的一种协议。它提供了主机域名和 IP 地址之间的转换。域名服务器使用固定的端口号 53，支持 UDP 和 TCP 访问。

（1）DNS 的报文格式：

1	15 16	32
标识	标志	
问题数	资源记录数	
授权资源记录数	额外资源记录数	
查询问题		
回答（资源记录数可变）		
授权（资源记录数可变）		
额外信息（资源记录数可变）		

图 9-1　DNS 的报文格式

上图是DNS查询和响应的报文格式。DNS查询和响应的报文由12字节长的首部和4个可变长度的字段组成。标识字段由客户程序设置并由服务器返回结果。

（2）DNS 工作流程：

域名服务分为客户端和服务器端，客户端提出请求，询问一个 Domain Name 的 IP 地址，服务器端必须回答客户端的请求。本地 DNS 首先查询自己的数据库，如果自己的数据库中没有对应的 IP 地址，则向本地 DNS 所设的上一级 DNS 询问，得到结果之后，将收到的结果保存在高速缓冲区，并回答给客户端。

待查询的域名放在查询问题中，查询结果放在回答的资源记录中。

【实验步骤】

练习一：利用仿真编辑器编辑 DNS 请求包

利用仿真编辑器编辑 DNS 请求包，请求的域名是：www.baidu.com.cn，在协议分析器端观察结果。

1、打开仿真编辑器，在工具栏选择“新建”，建立一个以太网帧（或利用仿真编辑器打开时默认的以太网帧进行编辑）；

2、填写该帧的以太网协议首部：

（1）源 MAC 地址：仿真编辑器的 MAC 地址；

（2）目的 MAC 地址：目的 MAC 地址应该是网关的 MAC，应用前面学习的知识获得；

（3）类型或长度：该字段值应为 0800（即 IP 协议的类型值）；

3、填写 IP 协议头信息：

（1）高层协议类型：即上层协议类型应为 17（UDP 协议的类型为 17）；

（2）总长度：IP 长度+UDP 长度；

（3）源 IP 地址：仿真编辑器的 IP 地址；

（4）目的 IP 地址：DNS 服务器的 IP 地址；

（5）计算 IP 首部校验和；

4、填写 UDP 协议的各个字段信息：

（1）源端口：自选，例如：2000；

（2）目的端口：选择 DNS 端口 53；

（3）UDP 长度：UDP 报头长度+DNS 长度；

注：以上步骤可简化为：（其它实验也是如此）

1)、打开仿真编辑器，在工具栏中选择 UDP 协议，将 UDP 目的端口改为 53；

2)、目的 MAC 地址：网关的 MAC；

3)、源 IP 地址：仿真编辑器的 IP 地址；

4)、目的 IP 地址：DNS 服务器的 IP 地址；

建议将全部字段都填充完毕后，在填充 IP/UDP 的长度及校验和。

5、填写 DNS 协议的各个字段信息，见图 9-2：

DNS 查询	
标识	1291
标志	0100
报文类型	0.............. 查询
查询响应类型	.0000.......... 标准查询
授权应答	.....0.....
截断	......0.....
要求递归	.......1.....
可用递归	........0.....
保留位	.........000....
响应中的错误状态码	............0000 无错误
问题记录数	1
应答记录数	0
授权记录数	0
附加记录数	0
问题记录	
Loop Block	
域名循环体	
Loop Block	
隐藏长度	3
域标记	www
Loop Block	
隐藏长度	5
域标记	baidu
Loop Block	
隐藏长度	3
域标记	com
Loop Block	
隐藏长度	2
域标记	cn
Loop Block	
长度	0
问题类型	1
问题类别	1
回答记录	
名字服务器资源记录	
附加资源记录	

图 9-2 DNS 各字段的信息

⑴标识：1291；

⑵标志：0100；

⑶问题记录数：1；

⑷应答记录数：0；

⑸授权记录数：0；

⑹附加记录数：0；

⑺问题记录：问题记录是填写 DNS 请求的域名部分，按下列步骤进行；

①域名循环体：鼠标移至域名循环体点，击鼠标右键，追加块 5 块（LoopBlock）；

②改写第一块的长度为 3，在域标记中添加 www；

③改写第二块的长度为 5，在域标记中添加 baidu；

④改写第三块的长度为 3，在域标记中添加 com；

⑤改写第四块的长度为 2，在域标记中添加 cn；

⑥第五块长度为 0；

⑦问题类型：1 A 主机类型；

⑧问题类别：1 IN （INTERNET）；

6、计算 UDP 校验和：覆盖 UDP 首部（含伪首部）和数据（DNS）两部分；

7、点击“发送”；

8、在协议分析器端截获报文，从会话分析中找出 DNS 请求包和 DNS 响应包；

9、在响应包中，找出 Answer section 字段并记录：

回答类型=（ ）；

回答等级=（ ）；

生存时间=（ ）；DNS 的有效时间

资源数据长度=（ ）；

鼠标点击下一项 Internet 地址[这 4 个字节是资源数据]；

在 16 进制对应的地方有 4 个字节的数据=（ ）；

这 4 个字节的数据就是我们要的 www.baidu.com.cn 的 IP 地址。

10、改变问题记录数据，比如：发送域名是 www.sina.com.cn 的 DNS 请求 ，在协议分析器截获报文，查出对应的 IP 地址是多少？

练习二：nslookup 工具的使用

nslookup 命令是查询域名对应 IP 的工具，其用法是：nslookup 域名

1、在命令提示符下输入：

c:>ipconfig /flushdns

c:>nslookup www.baidu.com.cn

2、观察运行结果，在协议分析器端截获相应的数据包；

3、查询其他域名的 IP 地址。

练习三：DNS 解析

1、打开 IE 浏览器，访问某站点，如在地址栏中写入：www.python.org 并回车。在协议分析器一端截获这一过程中传输的数据包，是否有 DNS 请求？

2、关闭 IE 浏览器，重新打开 IE 浏览器，再次访问 www.python.org。在协议分析器一端截获这一过程中传输的数据包，是否有 DNS 请求？为什么？

3、通过实验理解 DNS 协议的使用过程，从协议分析器端截获的数据包中找出本地的域名服务器 IP 地址是多少？域名服务进程的端口号是什么？

练习四：运行 ipconfig 命令

1、打开 IE 浏览器，访问某站点，如在地址栏中写入：www.163.com 并回车。

2、在命令提示符下运行：ipconfig /displaydns 显示本机缓冲区中的 DNS 解析内容。

3、观察运行结果。

练习五：Winsock 网络程序设计

设计一个简单的 Winsock 网络程序，模拟 UDP 数据报，加深对 DNS 作用的理解。按照以下步骤完成该练习：

1、启动 Visual C++，进入集成开发环境；

2、建立一个“Win32 Console Application”类型的工程,工程名自拟；

3、在该工程下，新建一个源程序文件，即选择“C++ Source File”，文件名自拟；

4、输入以下源程序（注释部分可不必输入）：

```
//////////////////////////////////////////////////////////
//本程序的功能为：                                      //
//（1）获取本机名（gethostname()函数）                  //
//（2）获得给定主机名的 IP 地址(gethostbyname()函数)    //
//（3）获得给定 IP 地址的主机名(gethostbyaddr()函数)    //
//////////////////////////////////////////////////////////
//头文件//
#include <winsock.h>
#include <stdio.h>
#include <string.h>
#pragma comment(lib, "ws2_32.lib")

//主文件//
int main()
{

    WORD wVersionRequested = MAKEWORD(1, 1);
    WSADATA wsaData;            //初始化 windows sockets API//
    if (WSAStartup(wVersionRequested, &wsaData)) {
```

```
        printf("WSAStartup failed %s\n", WSAGetLastError());
        return -1;
    }
    char hostname[256];                 //获得本主机名//
    int res = gethostname(hostname, sizeof(hostname));
    if (res != 0) {                     //错误处理//
        printf("Error: %u\n", WSAGetLastError());
        return -1;
    }
                                        //打印本主机名字//
    printf("本主机名为：%s\n", hostname);
    printf("请输入一个主机域名：");    //输入一个主机的域名//
    scanf("%s",hostname);          //利用主机名获得主机的地址//
    hostent* pHostent = gethostbyname(hostname);
    if (pHostent==NULL) {               //错误处理//
        printf("Error: %u\n", WSAGetLastError());
        return -1;
    }
//解析返回的主机地址信息：别名、地址类型、地址长度，并打印//
    hostent& he = *pHostent;
    printf("name=%s\naliases=%s\naddrtype=%d\nlength=%d\n",
        he.h_name, he.h_aliases, he.h_addrtype, he.h_length);

    sockaddr_in sa;          //打印主机每一个网卡的 IP 地址//
    for (int nAdapter=0; he.h_addr_list[nAdapter]; nAdapter++) {
        memcpy ( &sa.sin_addr.s_addr, e.h_addr_list[nAdapter], he.h_length);
        printf("Address: %s\n", inet_ntoa(sa.sin_addr));
    }                                   //显示 IP 地址//
    printf("\n");
    unsigned long addr;
    char hostaddr[50];
    printf("请输入服务器 IP 地址:"); //输入一个服务器的 IP 地址//
    scanf("%s",hostaddr);
    addr=inet_addr(hostaddr);   //将 IP 地址转化为网络字节序//
    pHostent=gethostbyaddr((char *)&addr,4,AF_INET);
                                //利用 IP 地址获得主机名//
```

```
    if (pHostent==NULL) {              //错误处理//
        printf("Error: %u\n", WSAGetLastError());
        return -1;
    }
    hostent& he1 = *pHostent;
        printf("主机名为：%s\naliases=%s\naddrtype=%d\nlength=%d\n",
        he.h_name, he.h_aliases, he.h_addrtype, he.h_length);
                                       //打印返回的主机信息//
    WSACleanup();              // 结束 windows sockets API//
    return 0;
}
```

5、编译、连接并运行程序；

6、观察程序运行的结果，体会程序中几个主要函数的作用。

练习六：反复运行上述程序，可以输入以下域名

www.jlu.edu.cn

mail.jlu.edu.cn

www.sina.com

www.microsoft.com

……

在协议分析器端捕获数据并分析说明：

1、域名解析使用传输层的哪一协议来处理查询请求？

2、本地的域名服务器 IP 地址是什么？

3、域名服务进程的端口号是什么？

4、随便输入一个不带后缀的主机名（如：aaa），观察捕获到的数据。

【思考问题】

结合实验过程中的实验结果，回答下列问题：

1、当仿真编辑器上同时打开多个浏览器窗口并访问同一 WEB 站点的不同页面时，系统是根据什么把返回的页面正确地显示到相应窗口的？

2、域名与 IP 地址之间是否有一一对应的关系？

【实验报告要求】

1、根据练习 1 说明 DNS 协议的工作流程。

2、根据实验结果，回答练习中提出的问题。

【实验十　DHCP 协议】

【实验目的】

1、 掌握 DHCP 的报文格式；
2、 掌握 DHCP 的工作原理。

【实验学时】

建议 2 学时

【实验类型】

验证型

【实验内容】

1、学习 DHCP 报文格式；
2、使用 DHCP 获取 IP 地址；
3、模拟重新登陆。

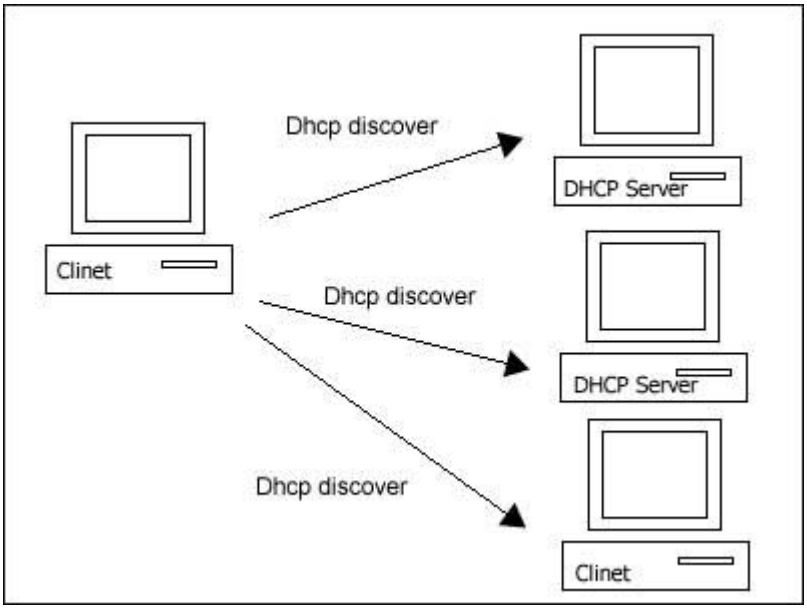
【实验原理】

一. DHCP 报文格式

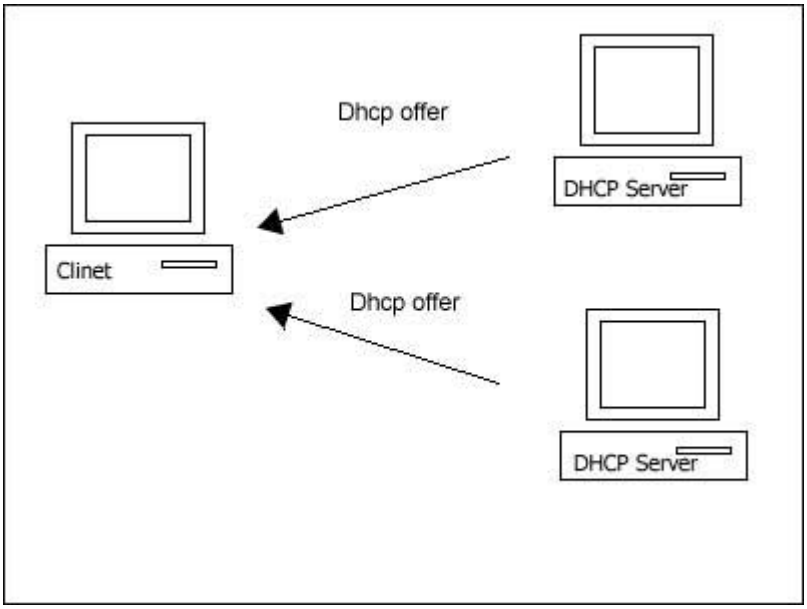
操作代码 (1 字节)	硬件类型 (1 字节)	硬件长度 (1 字节)	跳数 (1 字节)
事务 ID (4 字节)			
秒 (2 字节)		标志 (2 字节)	
客户端 IP 地址 (4 字节)			
您(客户端)的 IP 地址 (4 字节)			
服务器 IP 地址 (4 字节)			
网关 IP 地址 (4 字节)			
客户端硬件地址 (16 字节)			
服务器名 (64 字节)			
引导文件名 (128 字节)			
选项 (64 字节)			

二. DHCP 工作原理

发现阶段：即 DHCP 客户机寻找 DHCP 服务器的阶段。DHCP 客户机以广播方式（因为 DHCP 服务器的 IP 地址对于客户机来说是未知的）发送 DHCP discover 发现报文来寻找 DHCP 服务器（如下图）。

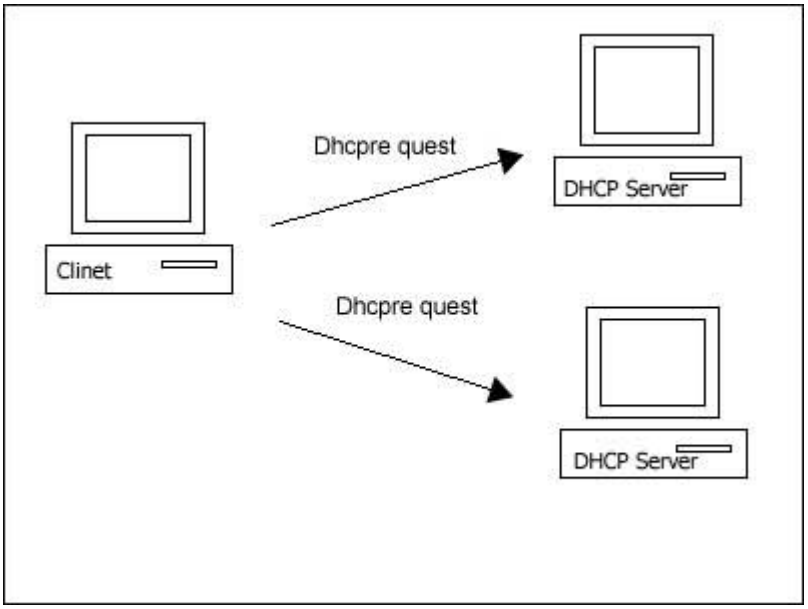


提供阶段：即 DHCP 服务器为 DHCP 客户机提供 IP 地址的阶段。在网络中接收到 DHCP discover 发现报文的 DHCP 服务器都会做出响应，它从尚未出租的 IP 地址中挑选一个分配给 DHCP 客户机，向 DHCP 客户机发送一个包含出租的 IP 地址和其他设置的 DHCP offer 提供报文（如下图）。

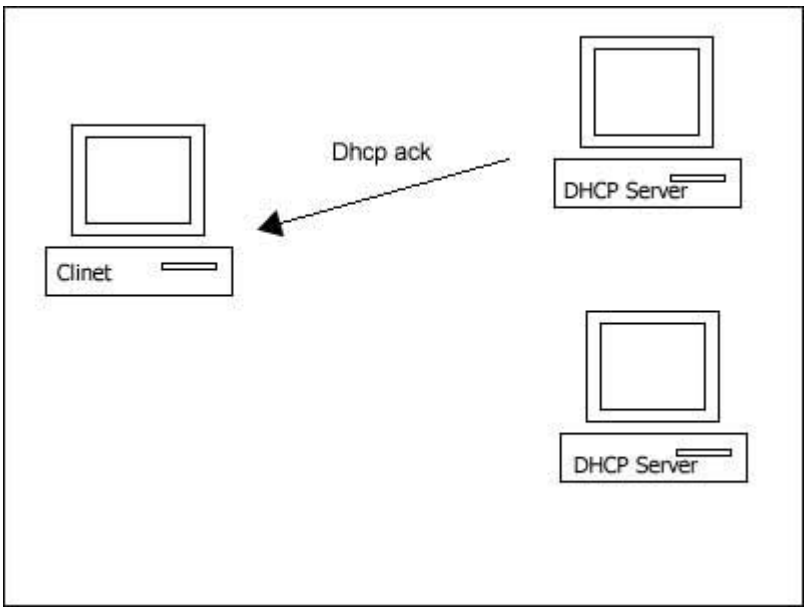


选择阶段：即 DHCP 客户机选择某台 DHCP 服务器提供的 IP 地址的阶段。如果有多台 DHCP

服务器向 DHCP 客户机发来的 DHCP offer 提供报文，则 DHCP 客户机只接受第一个收到的 DHCP offer 提供报文，然后它就以广播方式回答一个 DHCP request 请求报文，该报文中包含向它所选定的 DHCP 服务器请求 IP 地址的内容。之所以要以广播方式回答，是为了通知所有的 DHCP 服务器，他将选择某台 DHCP 服务器所提供的 IP 地址（如下图）。



确认阶段：即 DHCP 服务器确认所提供的 IP 地址的阶段。当 DHCP 服务器收到 DHCP 客户机回答的 DHCP request 请求报文之后，它便向 DHCP 客户机发送一个包含它所提供的 IP 地址和其他设置的 DHCP ack 确认报文，告诉 DHCP 客户机可以使用它所提供的 IP 地址。然后 DHCP 客户机便将其 TCP/IP 协议与网卡绑定，另外，除 DHCP 客户机选中的服务器外，其他的 DHCP 服务器都将收回曾提供的 IP 地址（如下图）。



重新登录：以后 DHCP 客户机每次重新登录网络时，就不需要再发送 DHCP discover 发现报文了，而是直接发送包含前一次所分配的 IP 地址的 DHCP request 请求报文。当 DHCP 服务器收到这一报文后，它会尝试让 DHCP 客户机继续使用原来的 IP 地址，并回答一个 DHCP ack 确认报文。如果此 IP 地址已无法再分配给原来的 DHCP 客户机使用时（比如此 IP 地址已分配给其它 DHCP 客户机使用），则 DHCP 服务器给 DHCP 客户机回答一个 DHCP nack 否认报文。当原来的 DHCP 客户机收到此 DHCP nack 否认报文后，它就必须重新发送 DHCP discover 发现报文来请求新的 IP 地址。

更新租约：DHCP 服务器向 DHCP 客户机出租的 IP 地址一般都有一个租借期限，期满后 DHCP 服务器便会收回出租的 IP 地址。如果 DHCP 客户机要延长其 IP 租约，则必须更新其 IP 租约。DHCP 客户机启动时和 IP 租约期限过一半时，DHCP 客户机都会自动向 DHCP 服务器发送更新其 IP 租约的报文。

【实验步骤】

练习一：使用 DHCP 获取 IP 地址

本练习由每一位组员独自完成。将分析端未用的网卡禁用。

1. 记下本机的 IP 地址，在命令行方式下，输入下面的命令：
   “netsh interface ip set address name=“*本机可用网卡的接口名*” source=dhcp”。
2. 在命令行方式下，输入命令“ipconfig -release”。
3. 启动协议分析器捕获数据并设置过滤条件(提取 DHCP 协议)。
4. 在命令行方式下，输入命令“ipconfig -renew”。
5. 察看 DHCP 会话分析，填写下表。

报文序号	Boot record type 的值	Message Type 的值	Lease Time 的值(若有)	源 IP 地址	目的 IP 地址
…					

6. 等待，超过租用时间(上表中的 Lease Time 的值)的 50%后，察看捕获的数据包。
   - 各报文中字段“Boot record type”、“Message Type”的值分别是多少？该请求报文的作用是什么？

练习二：模拟重新登录

1. 启动仿真编辑器，编辑一个 DHCP request 数据包，其中：

   MAC 层：“源 MAC 地址”设置为本机 MAC 地址

   “目的 MAC 地址”设置为服务器 MAC 地址

   IP 层：“源 IP 地址”设置为本机 IP 地址

   “目的 IP 地址”设置为服务器 IP 地址(172.16.0.10)

   UDP 层：“源端口”设置为 68

   “目的端口”设置为 67

   DHCP 层：“代码”设置为 1

   “标志”设置为 0

   “客户端 IP 地址”设置为分析端的 IP 地址

   “你的 IP 地址”设置为 0.0.0.0

   “客户端硬件地址”设置为本机的 MAC 地址

   追加选项块

   “选项代码”设置为 53

   “长度”设置为 1

   “DHCP 消息类型”设置为 3

2. 启动协议分析器捕获数据并设置过滤条件(提取 DHCP 协议)。
3. 发送编辑好的数据包。
4. 察看捕获的数据。
   - 各报文中字段“Boot record type”、“Message Type”的值分别是多少？
   - 在捕获的报文中是否有 DHCP discover 报文，为什么？

【思考问题】

1. DHCP 为何使用 67、68 两个熟知端口进行 UDP 通信？
2. DHCP 协议适合于什么情况下使用？请举例说明。

【实验十一　SMTP 和 POP3 协议】

【实验目的】

1、理解 POP3、SMTP 协议的基本原理；

2、了解 SMTP 和 POP3 的基本命令格式；

3、了解应用层协议与传输层协议的关系。

【实验学时】

4 学时

【实验类型】

验证型

【实验内容】

1、学习邮件服务的工作原理；
2、理解 SMTP 协议和 POP3 协议；
3、学习 POP3 协议的命令格式和使用方法；
4、学习 SMTP 协议的命令和使用方法；
5、学习 Outlook 的使用方法。

【实验原理】

1、POP3（邮局协议 3）

POP 是一个脱机协议，它是一个具有存储转发功能的中间服务器。POP3 是邮局协议的第 3 个版本。POP3 协议采用客户/服务器工作模式。

表 11-1　POP3 的命令码

命令	描述
USER〈用户邮件地址〉	用来区分用户正在连接的邮箱
PASS〈口令〉	此命令将邮件箱的口令发送给 POP3 服务器。缺省情况用明码文本发送
STAT	请求服务器发回关于邮箱的统计资料，如邮件总数和总字节数
LIST<邮件编号>	返回邮件数量和每个邮件的大小
RETR<邮件编号>	返回由参数标识的邮件的全部文本
DELE<邮件编号>	服务器将由参数标识的邮件标记为删除，由 quit 命令执行
RSET	服务器将重置所有标记为删除的邮件，用于撤消 DELE 命令
NOOP	服务器返回一个肯定的响应
QUIT	更新

2、SMTP（简单的邮件传送协议）

SMTP 协议是两个报文传输代理之间的通信协议。它规定了两个相互通信 MTA 之间如何交换信息。SMTP 协议的最大特点是简单，它规定了发送程序和接收程序之间的命令和应答

格式。

表 11-2 常用 SMTP 命令码

命 令	描 述
HELO<domain><CRLF>	发送一个 SMTP 的 HELLO 命令，开始会话
MAIL FROM:〈发送者邮件地址〉〈CRLF〉	此命令开始一个邮件传输处理
RCPT TO:〈接收者邮件地址〉<CRLF>	标识邮件接收者的地址
DATA 〈CRLF〉	接收 SMTP 将 DATA 命令后面的数据看作邮件内容处理，以＜CRLF＞.＜CRLF＞标识数据的结尾。
REST 〈CRLF〉	退出/复位当前的邮件传输
NOOP 〈CRLF〉	要求接收 SMTP 仅做 OK 应答。（用于测试）
QUIT 〈CRLF〉	要求接收 SMTP 返回一个 OK 应答并关闭传输。
HELP 〈CRLF〉	查询服务器支持什么命令

注：＜CRLF＞为回车、换行

表 11-3 SMTP 应答码

应答码	描述	应答码	描述
501	参数格式错误	251	用户非本地，将转发向＜forward-path＞
502	命令不可实现	450	要求的邮件操作未完成，邮箱不可用
503	错误的命令序列	550	要求的邮件操作未完成，邮箱不可用
504	命令参数不可实现	451	放弃要求的操作；处理过程中出错
211	系统状态或系统帮助响应	551	用户非本地，请尝试＜forward-path＞
214	帮助信息	452	系统存储不足，要求的操作未执行
220	＜domain＞服务就绪	552	过量的存储分配，要求的操作未执行
221	＜domain＞服务关闭	553	邮箱名不可用，要求的操作未执行
421	＜domain＞服务未就绪	354	开始邮件输入，以".″结束
250	要求的邮件操作完成	554	操作失败

注：第一位数字为 2 时表示命令成功；为 5 时表示失败；为 3 时表示没有完成。

【实验步骤】

本实验要求：

每台仿真编辑器上都安装有 outlook express 软件，学生可以进行账号的配置并利用其进行收发邮件的操作。

邮件服务器域名为 JServer.NetLab。

邮件服务器的用户名和密码的分配规则按 Group1、Group2、Group3 …… GroupN 递增，用户名和密码相同。每台主机拥有一个用户。下面仅以用户名 Group1 密码 Group1 为例说明实验步骤。

练习一：利用“TCP 工具”和 SMTP 命令编辑邮件并发送

本练习的目的是学习 SMTP 协议的命令用法，以及一个邮件发送的过程。

1、首先要与邮件服务器建立一个 TCP 连接：

⑴打开工具菜单栏中的“TCP 连接视图”；

⑵填写邮件服务器的 IP 地址及邮件服务器 SMTP 协议的端口号（25），点击“连接”，从而建立本机与邮件服务器 SMTP 端口的 TCP 连接（从协议分析器一端可以捕获到三次握手的过程）。

⑶若连接成功，在接收窗口会显示成功连接的信息:

220 ......

若不成功，查看 IP 地址和端口号是否有错，再次尝试进行连接，直到成功。

2、用 SMTP 命令编辑并发送邮件：

⑴在发送窗口编辑发送 SMTP 协议的命令，在接收窗口会返回服务器端信息：

发送窗口：helo group1<CRLF>　　点击“发送”；

接收窗口：250 ...... pleased to meet you

发送窗口：mail from:Group1@JServer.NetLab<CRLF>　　点击“发送”；

接收窗口：250 2.1.0 ...... Sender ok

发送窗口：rcpt to: Group1@JServer.NetLab <CRLF>　　点击“发送”；

接收窗口：250 2.1.5 ...... Recipient ok

发送窗口：data<CRLF>　　点击“发送”；

接收窗口：354 Enter mail, end with ″.″ on a line by itself

发送窗口：my email<CRLF> (My email 是邮件内容)　　点击“发送”；

接收窗口：

发送窗口：. <CRLF>　　点击“发送”；

接收窗口：250 2.0.0 ...... Message accepted for delivery

发送窗口：quit<CRLF>　　点击“发送”。

接收窗口：221 2.0.0 ...... closing connection

(注：<CRLF>是回车换行)

⑵观察 TCP 连接视图的接收信息窗口中返回的信息；

⑶在协议分析器一端截获命令报文及应答报文，分析序号和确认序号，标志位，报文内容等。

3、断开连接：

在协议分析器一端截获报文，分析 TCP 拆除连接的过程。

练习二：利用 TCP 连接客户端工具和 POP3 命令接收电子邮件

1、选择邮件服务器，建立一个 TCP 连接：

⑴打开“工具”菜单中的 TCP 连接视图；

⑵填写邮件服务器的 IP 地址及邮件服务器 POP3 协议的端口号 110，点击“连接”，从而建立本机与邮件服务器的 TCP 连接；

⑶若连接成功，在接收窗口会显示成功连接的信息（+OK POP3 ..... server ready）；若不成功，再次尝试进行连接，直到成功。

2、察看邮件信息：

⑴在发送窗口编辑发送 POP3 协议的命令，在接收窗口会返回服务器端信息：

发送窗口：user Group1<CRLF>　　点击“发送”；

接收窗口：+OK User name accepted, password please

发送窗口：pass Group1<CRLF>　　点击“发送”；

接收窗口：+OK mailbox open, 1 messages

发送窗口：stat<CRLF>　　点击“发送”；

接收窗口：+OK 1 387

发送窗口：uidl<CRLF>　　点击“发送”；

接收窗口：+OK Unique-ID listing follows

1 4427b358000000001

发送窗口：list<CRLF>　　点击“发送”；

接收窗口：+OK Mailbox scan listing follows

1 387

发送窗口：retr 1 <CRLF> （查看第一封信件内容）　　点击“发送”；

接收窗口：+OK 387 octets

按下一帧查看第 1 封信的内容；

发送窗口：dele 1<CRLF>　　点击“发送”；

接收窗口：+OK message deleted

发送窗口：quit<CRLF>　　点击“发送”；

接收窗口：+OK Sayonara

⑵观察 TCP 连接视图的接收信息窗口中返回的信息；

⑶在协议分析器一端截获命令报文及应答报文，分析序号和确认序号，标志位，报文内容等。

3、断开连接：

在协议分析器一端截获报文，分析 TCP 拆除连接的过程。

练习三：利用仿真编辑器编辑 SMTP 问候数据包

本练习的目的是理解 SMTP 的工作过程和数据包封装格式。SMTP 是应用层协议，在传输层使用 TCP 协议建立连接，因此不能通过编辑 SMTP 数据帧来实现 SMTP 问候数据包。我们可以利用仿真编辑器的 TCP 连接工具，先建立一个 TCP 连接，然后通过捕获响应数据包，查出端口号和序号。步骤如下：

1、参考练习一的内容，利用 TCP 连接视图工具和邮件服务器建立一个 TCP 连接；

2、在协议分析器一端捕获数据包，找出三次握手的过程，记录仿真编辑器端启动的端口号、三次握手后的最后一个 ACK 应答包的序列号和确认号，记录窗口大小。

3、在仿真编辑器一端建立一个 SMTP 问候包，步骤如下：

⑴打开仿真编辑器，在界面初始状态下，程序会自动新建一个单帧，可以利用仿真编辑器打开时默认的以太网帧进行编辑；

⑵填写该帧的以太网协议首部：

①源 MAC 地址：仿真编辑器的 MAC 地址；

②目的 MAC 地址：目的 MAC 地址应该是邮件服务器的 MAC 地址；

③类型或长度字段：该字段值应为 0800（即 IP 协议的类型值）；

⑶填写 IP 协议头信息：

①高层协议类型：即上层协议类型应为 6（TCP 协议的类型）；

②总长度：应包括 TCP 总长度；

③源 IP 地址：仿真编辑器的 IP 地址；

④目的 IP 地址：邮件服务器的 IP 地址；

⑤首部校验和,确定长度以后计算；

⑷填写 TCP 协议头信息：

①源端口：记录下来的端口号，即 TCP 连接工具进程的端口号；

②目的端口：25（SMTP 协议）；

③序列号：记录下来的确认号+1；

④确认号：记录下来的序列号；

⑤首部长度和标志位：5018（即长度 20 字节，标志 ACK=1，PSH=1）；

⑥窗口大小：用记录值；

⑦紧急指针：0；

⑧校验和：确定 SMTP 内容以后计算。计算方法参见实验 6-TCP 协议；

⑸填写 SMTP 协议信息：

①右键单击协议树中“SMTP 协议数据”选择“追加块”；在单帧编辑器中“首部字段”中写：HELO+主机名

②计算 IP 总长度，添加到 IP 长度字段；计算 TCP 总长度，添加到 TCP 长度字段；

③分别计算 IP、TCP 校验和，添加到相应字段；

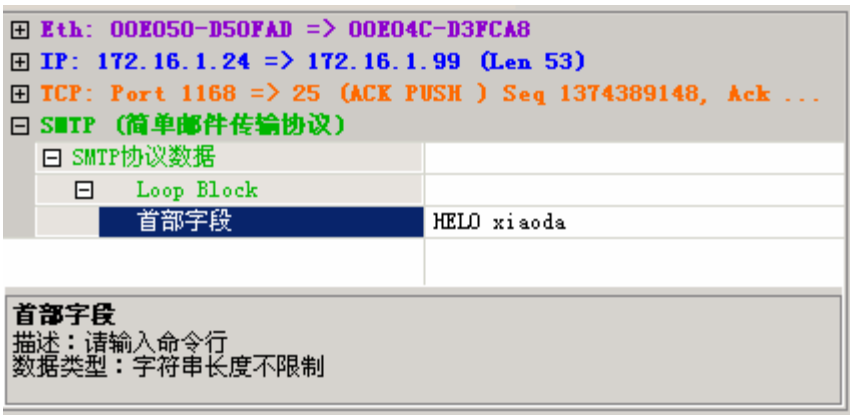


图 11-1 SMTP 协议信息编辑

4、编辑正确后，发送；

5、点击 TCP 连接工具的“断开”，断开和邮件服务器的连接；

6、在协议分析器一端截获数据，找出 SMTP 问候包和 SMTP 响应包，记录响应结果。理解 SMTP 的工作过程和数据包封装格式。同理，可以编辑 POP3 协议的命令数据包。

练习四：利用 Outlook 发送电子邮件

1、打开邮件工具：Outlook Express；

2、设置一个邮件帐号；

3、编辑并发送一个电子邮件；

4、在协议分析器一端捕获数据，联系练习二的步骤；

5、找出这次发送邮件的所有数据包；

6、找出传输用户帐号及密码的数据包，能不能看到用户密码是什么？

7、发送邮件的过程传输层采用什么协议？

8、发送端的端口号是多少？代表什么协议？

练习五：利用 Outlook 接收电子邮件

1、打开邮件工具：Outlook Express；

2、接收电子邮件；

3、在协议分析器一端捕获数据，联系练习二的步骤；

4、找出这次接收邮件的所有数据包；

5、找出传输用户帐号及密码的数据包，能不能看到用户密码是什么？

6、找出所接收邮件中的某一邮件内容；

7、接收邮件的过程传输层采用什么协议？

8、发送端的端口号是多少？代表什么协议？

【思考问题】

1、说明为什么要借助 TCP 连接工具才能用仿真编辑器编辑 SMTP 协议的命令？

【实验报告要求】

1、通过实验说明你的电子邮件在网络上传输是安全的吗？为什么？如果不安全，你认为实现邮件安全传输的最好的办法是什么？

2、回答练习中提出的问题。

【实验十二　SNMP 协议与网络管理】

【实验目的】

1、理解 SNMP 协议的工作原理；

2、理解 SNMP 协议的作用。

【实验学时】

2 学时

【实验类型】

验证型

【实验内容】

1、通过仿真 SNMP 数据包，学习 SNMP 协议的格式；

2、学习使用 SNMP 工具，了解 MIB 以及基本的 MIB 变量；

3、学习 SNMP 协议的作用；

4、了解 SNMP 与 UDP 的关系。

【实验原理】

用于网络管理站与被管设备的网管代理之间交互管理信息的协议。管理信息的结构使用一种概念树。树叶表示各种对象，帮助用户了解互连网络的结构。网络管理站通过 SNMP 协议向被管设备的网管代理发出各种请求报文，网管代理则接收这些请求后完成相应的操作。SNMP 为应用层协议，是 TCP/IP 协议簇的一部分，它是通过 UDP 用户数据报协议来操作的。

网络管理员对网络及其设备的管理有三种方式：本地终端方式、远程 Telnet 命令方式和基于 SNMP 的代理/服务器方式。

1、SNMP 的报文格式：

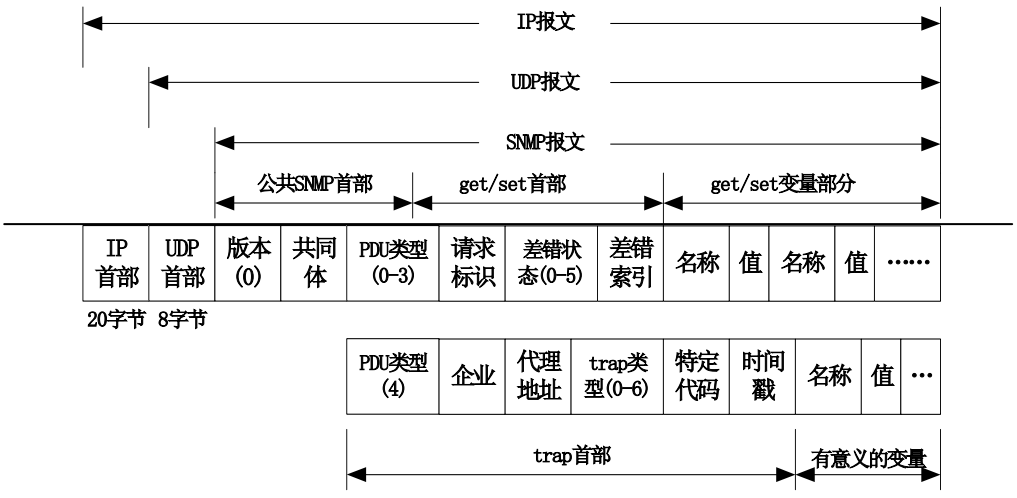


图 12-1　SNMP 报文格式

字段说明：

共同体：

共同体字段是一个字符串。这是管理进程和代理进程之间的口令，是明文格式。默认值是 public。

表 12-1 PDU 类型

PDU 类型	名 称	描 述
0	get-request	从代理进程提取一个或多个参数值
1	get-next-request	从代理进程提取一个或多个参数的下一个参数值
2	get-response	应答一个取操作
3	set-request	设置代理进程的一个或多个参数值
4	trap	代理进程主动发出的报文，通知管理进程有某些事情发生

请求标识：

请求标识由管理进程设置，由代理进程在 get-response 中返回。管理进程通过它来确定代理进程发出的响应与管理进程发出的查询是否匹配。

表 12-2 差错状态

差错状态	名 称	描 述
0	noError	没有错误
1	tooBig	代理进程无法把响应放在一个 SNMP 消息中发送
2	noSuchName	操作不存在的变量
3	badValue	set 操作的值或语义有错误
4	readOnly	管理进程试图修改一个只读变量
5	genErr	其他错误

差错索引：

是一个整数偏移量，指明当有差错发生时，差错发生在哪一个参数。它是由代理进程标注的，只有在发生 noSuchName、badValue、readOnly 差错时才标注。

表 12-3 trap 类型

trap 类型	名 称	描 述
0	coldStart	代理进程对自己初始化
1	warmStart	代理进程对自己重新初始化
2	linkDown	一个接口已经从工作状态改变为故障状态，报文中的第一个变量标识此接口
3	linkUp	一个接口已经从故障状态改变为工作状态，报文中的第一个变量标识此接口
4	authenticationFailure	从 SNMP 管理进程收到无效共同体的报文
5	egpNeighborLoss	一个 EGP 邻站已改变为故障状态。报文中的第一个变量包含此邻站的 IP 地址
6	enterpriseSpecific	在这个特定的代码字段中查找 trap 信息

SNMP 管理站用 Get-Request 消息从拥有 SNMP 代理的网络设备中检索信息，而 SNMP 代理则用 Get-Response 消息响应。Get-Next-Request 用于和 Get-Request 组合起来查询特定的表对象中的列元素。如：

首先通过下面的原语获得所要查询的设备的接口数：

{iso org(3) dod(6) internet(1) mgmt(2) mib(1) interfaces(2) ifNumber(2)}

然后再通过下面的原语，进行查询(其中第一次用 Get-Request，其后用 Get-Next-Request)：

{iso org(3) dod(6) internet(1) mgmt(2) mib(1) interfaces(2) ifTable(2)}

2、详细原理请参考教材

【实验步骤】

练习一：查看网络信息库（MIB）

本练习的目的是学习了解 MIB2 库的内容。实验步骤如下：

1、打开仿真编辑器，在视图菜单中选择“SNMP 连接视图“；

2、在 SNMP 主机扫描地址本中双击一台主机 IP，自动填加到 IP 文本框中，例如：172.16.0.55

3、依次展开节点：（在下部显示对不同信息的描述）

-iso（国际标准化组织）1

-org（国际组织）3

-dod（美国国防部）6

-internet（互联网）1

-mgmt（管理组织机构）2

-mib2（管理信息库）1

得到 mib2 节点：1.3.6.1.2.1

4、查看节点 mib2 下的 8 个组：

-system（系统）1

-interface（接口）2

-at（地址翻译）3

-ip（IP 信息）4

-icmp（ICMP 信息）5

-tcp（TCP 信息）6

-udp（UDP 信息）7

-egp（外部网关协议信息）8

单击获取按钮在右侧信息框中会有相应的变量信息；

例如：点击 system 组节点；会列出该组节点下的 7 个变量；

- sysDescr ；系统的文字描述
- sysObjectID ；在子树 1.3.6.1.2.1.1.2 中的厂商标识
- sysUpTime ；从系统的网管部分启动以来运行的时间（百分之一秒为单位）

- sysContact ；联系人的名字及联系方式
- sysName ；节点的完全合格的名称
- sysLocation ；节点的物理位置
- sysServices ；指示节点提供的服务的值

点击 sysName 变量，在右边的窗口出现该变量的值。

记录 system 组下的变量的对象标识都是多少？

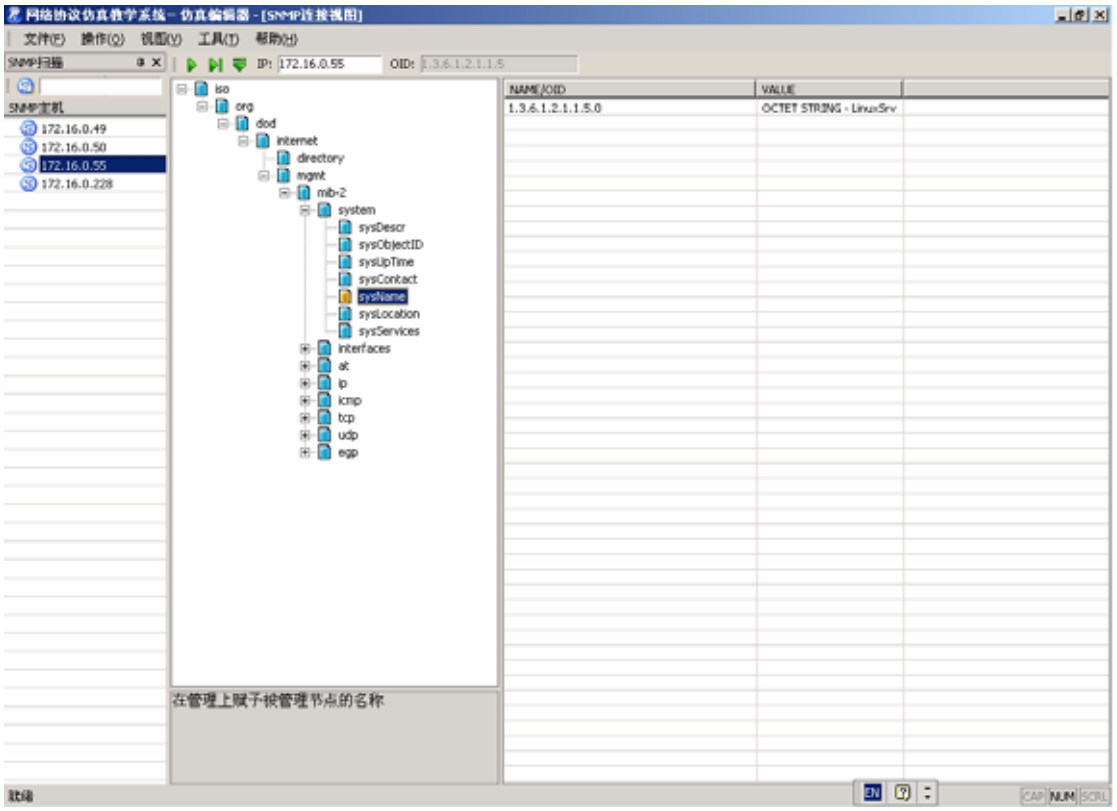


图 12-2 SNMP 编辑视图

5、点击其他的组节点，查看相应的变量。

练习二：编辑并发送 SNMP 数据包

SNMP 服务是建立在 UDP 协议之上的，SNMP 数据封装在 UDP 数据包内，故对 SNMP 数据的编辑即是对 UDP 数据字段内容的编辑。

1、打开仿真编辑器，在界面初始状态下，程序会自动新建一个单帧，可以利用仿真编辑器打开时默认的以太网帧进行编辑；

2、填写该帧的以太网协议首部：

⑴目的 MAC 地址:网关的 MAC 地址；

源 MAC 地址:本机的地址

⑵类型或长度字段：0800（即 IP 协议的类型值）；

3、填写 IP 协议头信息：

⑴总长度：IP 长度+UDP 长度+SNMP 长度；

⑵高层协议类型：选择上层协议类型为 UDP 协议，其类型值为 17；

⑶目的 IP 地址：为代理主机（待管理设备）的 IP 地址，可以设为：202.198.156.254；

⑷源 IP 地址：本机 IP ；

⑸计算 IP 头的校验和；

4、填写 UDP 协议信息：

⑴源端口：自选；

⑵目的端口：161（SNMP 代理进程的 UDP 端口号）；

⑶UDP 长度：UDP 报头长度与 UDP 数据长度（SNMP 部分）之和；

⑷编辑 39 个字节的 SNMP 数据字段：点击 SNMP 的数据出现编辑框，添入下列信息：

30 25 02 01 00 04 06 70 75 62 6C 69 63 A1 18 02

01 01 02 01 00 02 01 00 30 0D 30 0B 06 07 2B 06

01 02 01 01 05 05 00

⑸校验和：覆盖 UDP 首部（含伪首部）和数据（SNMP）两部分，计算方法参考实验五；

5、点击工具栏或菜单栏中的“发送”，在弹出对话框上选择“发送”按钮，发送 SNMP 报文；

6、在协议分析器一端截获该 SNMP 请求报文及 SNMP 响应报文，并记录分析响应报文的内容。

练习三：SNMP SET 实验

服务器配置：

1、将 SNMP 的节点都设为可读：

⑴打开/etc/snmp/snmpd.conf 文件，命令：vi /etc/snmp/snmpd.conf

⑵在文件中找到以 view 开头的行，例如

view systemview included .1.3.6.1.2.1.1

view systemview included .1.3.6.1.2.1.5

……

⑶删除所有找到行。添加如下行：

view systemview included .1.3.6.1.2.1

⑷在文件中找到以 access 开头的那一行，例如：

access notConfigGroup ……

⑸将找到的行改为如下所示：

access notConfigGroup “” any noauth exact systemview none none

⑹重新启动 snmp 服务。（命令：**/etc/rc.d/init.d/snmpd restart**）

2、将 SNMP 的节点都设为可读写：

⑴打开/etc/snmp/ snmpd.conf 文件，命令：vi /etc/snmp/snmpd.conf

⑵在文件中找到以 view 开头的行，例如

view systemview included .1.3.6.1.2.1.1

view systemview included .1.3.6.1.2.1.5

……

⑶删除所有找到行。添加如下行：

view systemview included .1.3.6.1.2.1

⑷在文件中找到以 access 开头的那一行，例如：

access notConfigGroup “” any noauth exact systemview writable none

⑸将找到的行改为如下所示：

access notConfigGroup “” any noauth exact systemview systemview none

⑹重新启动 snmp 服务。（命令：**/etc/rc.d/init.d/snmpd restart**）

3、设置 SNMP 节点值

本练习的目的是为了学习如何设置 SNMP 节点值，以 sysName 节点为例，其实验步骤如下：

⑴在仿真端打开仿真编辑器，在视图菜单中选择“SNMP 连接视图”；

⑵展开节点：mib-2 —> system —> sysName，选中该节点并获取该节点的值；

⑶在“操作”菜单下选择“设置”选项，弹出“设置 SNMP 节点值”对话框；

⑷在“公共体名称”中有两个选项“public”和“private”，在本练习中选择“public

⑸在“节点值”中输入新的名称，例如：xyz；

⑹在“变量绑定类型”中有 8 种不同的数据类型，可根据不同节点选择相应的数据类型，在本练习中选择“Octet String”；

⑺在分析端，打开协议分析器，设置过滤器为 snmp，新建捕获窗口，选择该过滤器并开始捕获；

⑻在仿真端，设置完成后点击“确定”按钮，然后重新获取 sysName 节点的值。

⑼查看捕获的数据包。

练习四：SNMP TRAP 实验

服务器配置：

设置 trap 的公用体和目的主机：

1、打开/etc/snmp/snmpd.conf 文件，命令：vi /etc/snmp/snmpd.conf

2、在文件中找到以 access 开头的那一行，例如：

access notConfigGroup “” any noauth exact systemview systemview none

3、在 access 行后，另起一行，输入如下配置信息：

trapcommunity public

trapsink IP

(注意：IP 为设置的接收 trap 报文的目的主机，不是服务器)

4、重新启动 snmp 服务。（命令：**/etc/rc.d/init.d/snmpd restart**）

练习：

1、在分析端打开协议分析器，设置过滤器为 SNMP；

2、新建捕获窗口，应用过滤器 SNMP，开始捕获；

3、教师在服务器上重新启动 SNMP 服务：

在命令提示符下输入命令：

/etc/rc.d/init.d/snmpd restart

(注意：此命令使 snmp 服务重启，触发 trap 报文发送到目的主机)

4、在分析端捕获并分析 trap 数据包。

【思考问题】

结合实验过程中的实验结果，回答下列问题：

1、MIB2 的 UDP 节点组下有哪些变量？

2、MIB2 的 TCP 节点组下有哪些变量？

【实验报告要求】

1、说明 SNMP 协议的用途。

2、解释 SNMP 服务为什么选择 UDP 协议而不是 TCP 协议作为下层协议？

【实验十三　HTTP 协议】

【实验目的】

1、掌握 HTTP 协议的使用；

2、了解 HTTP 协议的实现方式。

【实验学时】

2 学时

【实验类型】

验证型

【实验内容】

1、学习 HTTP 协议的格式和与 TCP 协议的关系

2、了解 HTTP 协议的通信过程；

3、理解 HTTP 协议的无连接、无状态特性。

【实验原理】

1、HTTP 协议

HTTP 报文共有两种通用类型：请求报文，响应报文。

（1）请求报文

请求报文包括请求行、首部，以及有时出现的主体。如图 13-1：



图 13-1　HTTP 请求报文

请求行：包括请求类型、空格、URL、空格以及 HTTP 版本。请求行的格式如下：



图 13-2　请求行格式

请求类型：此字段定义了几种不同方法的报文。

常用的几种方法：

①GET：客户要从服务器读取文档时使用；

②HEAD：客户想得到关于文档的某些信息，但并不是要这个文档时使用；

③POST：当客户要给服务器提供某些信息时使用；

④PUT：当客户将新的或更换的文档存储在服务器上时使用；

⑤COPY：当需要将文件复制到另一个位置时使用；

⑥MOVE：当需要将文件移到另一个位置时使用；

⑦DELETE：当需要将服务器上的文档移走时使用；

⑧LINK：当需要创建从一个文档到另一个位置的链接时使用；

⑨UNLINK：当需要删除由 LINK 方法创建的链接时使用；

⑩OPTION：当客户向服务器询问到一些可用的选项时使用。

统一资源定位符（URL）：有 4 个要素：方法、主机、端口、路径，



图 13-3　URL 示意图

①方法：用来读取文档的协议；

②主机：放置信息的计算机；

③端口：服务器的端口号，可选；

④路径：放置文件的路径名。

（2）响应报文

响应报文包括状态行、首部、有时包括主体。

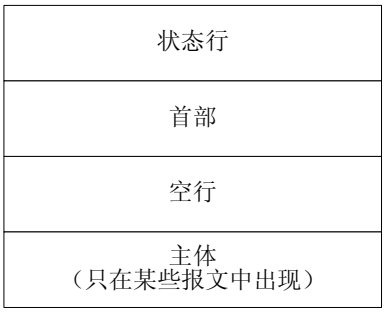


图 13-4　HTTP 响应报文示意图

状态行：包括 HTTP 版本、空格、状态码、空格、状态短语。

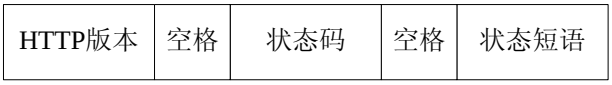


图 13-5　HTTP 响应报文状态行

状态码：

100 系列：提供信息的；

200 系列：指示成功的请求；

300 系列：把客户重定向到另一个 URL；

400 系列：指示客户端差错；

500 系列：指示服务器端差错；

状态短语：用来解释状态码。

2、详细原理请参考教材

【实验步骤】

练习一：使用 TCP 连接客户端工具，发送 HTTP 数据；

实验步骤如下：

1、首先要选择一个 WWW 服务器，建立一个 TCP 连接：

⑴选择一个 WWW 服务器，比如 www.jlu.edu.cn，利用前面所学的知识获得它的 IP 地址；

⑵打开 TCP 连接客户端，填写所选择的 WWW 服务器的 IP 地址及 WWW 服务器进程的端口号 80；

2、编辑并发送 HTTP 数据

⑴在发送窗口编辑发送数据信息，如：

GET / HTTP / 1.0<CRLF>（或 GET / HTTP / 1.1<CRLF>）；

Host: www.jlu.edu.cn<CRLF>

<CRLF>

注意“/”前后的空格；<CRLF>代表回车换行。

⑵点击“连接”，从而建立与 WWW 服务器的 TCP 连接。HTTP 连接成功时“断开”按钮变为有效；若不成功，再次尝试进行连接或换其他服务器。

⑶点击“发送”，若成功，服务器会返回信息，见下图；否则，点击“停止” ，重新进行⑵、⑶步；

（注意:发送 HTTP 的数据包含回车符，点击“连接”和“发送”的连贯性）。



图 13-6 HTTP 服务器对请求的回应显示

⑷在协议分析端截获相应的报文并分析；

⑸继续发送其他数据，如：

HEAD / HTTP/ 1.0<CRLF> （或 HEAD / HTTP / 1.1<CRLF>）

Host: www.jlu.edu.cn<CRLF>

<CRLF>

是否可以发送成功？为什么？由此理解 HTTP 的无连接性、无状态特性。若要继续发送必须重新建立连接。

练习二：使用 TCP 连接客户端工具，测试 HTTP 的连接、断开过程；

测试 HTTP 的连接、断开连接的过程，可以帮助我们更好的理解 HTTP 的无连接性，无状态性，同时，了解 HTTP 的 TCP 连接过程。

1、选择一个 WWW 服务器，比如 www.jlu.edu.cn；确定它的 IP 地址。

⑴启动 TCP 连接视图工具，填写 IP 地址和 80 端口号；

⑵点击“连接”按钮建立连接；

⑶在协议分析端截获相应的报文及应答报文；

⑷完成下面的填空：

客户端 **WWW 服务器**

Port( ) SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( ) Port( )

SEQ=( ),ACK SEQ=( )

SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( ) TCP 连接建立阶段

SEQ=( ),ACK SEQ=( )

SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( )
SEQ=( ),ACK SEQ=( )

2、发送 HTTP 数据

⑴在发送数据窗口编辑欲发送的数据信息，如：GET / HTTP / 1.0<CRLF>；

⑵点击“发送”按扭，即完成上述数据信息的发送；

⑶观察显示数据窗口中返回的信息；如果不成功，重新连接并发送。

⑷完成下面的填空：

客户端 **WWW 服务器**

Port( ) SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( ) Port( )
SEQ=( ),ACK SEQ=( )

SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( )
SEQ=( ),ACK SEQ=( )

SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( ) TCP 数据传输阶段
SEQ=( ),ACK SEQ=( )

SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( )
SEQ=( ),ACK SEQ=( )

3、断开连接

⑴在协议分析端截获报文，分析 TCP 断开连接的过程。

⑵完成下面的填空：

客户端 **WWW 服务器**

Port( ) SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( ) Port( )
SEQ=( ),ACK SEQ=( )

SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( ) TCP 连接断开阶段
SEQ=( ),ACK SEQ=( )

SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( )
SEQ=( ),ACK SEQ=( )

SYN= ( ), ACK= ( ), PSH= ( ), FIN= ( )
SEQ=( ),ACK SEQ=( )

练习三：通过浏览器学习 HTTP 协议

1、在仿真编辑端主机上打开 IE 浏览器；

2、在浏览器的地址栏中输入一个 WEB 站点的 URL 如：http://www.jlu.edu.cn；

3、在协议分析端捕获数据，分析数据包的传输过程，记录 TCP 连接的三次握手、序号、确认序号、TCP 数据长度和仿真机的进程端口号；一个网页是否只有一个进程？记录本次 HTTP 请求过程 TCP 断开连接的过程。

4、通过协议分析器捕获数据，在上面的网页上继续浏览一个链接，是否需要新的 TCP 连接？

5、说明 HTTP 与 TCP 之间是什么关系？说明 HTTP 的无连接、无状态性。

6、说明 HTTP 协议的格式是什么样的？

【思考问题】

结合实验过程中的实验结果，回答下列问题：

1、当仿真机上同时打开多个浏览器窗口并访问同一 WEB 站点的不同页面时，系统是根据什么把返回的页面正确地显示到相应窗口的？

2、一个主页是否只有一个连接？

3、为什么 HTTP 不保持与客户端的 TCP 连接？

【实验报告要求】

1、说明 HTTP 协议的应用，结合实验说明 HTTP 协议的无连接、无状态特性。

2、对于域名是 www.jlu.edu.cn 的 WWW 服务器，你利用什么手段获得它的 IP 地址？

3、根据实验结果，回答练习中提出的问题。

【实验十四　WINS 和 NETBIOS 协议】

【实验目的】

1、了解 Windows 网络协议架构；

2、了解 NETBIOS 协议的使用；

3、理解 WINS 协议及 SMB 协议。

【实验学时】

2 学时

【实验类型】

验证型

【实验内容】

1、通过发送 NETBIOS 名字请求包，学习 WINS 协议和 NETBIOS；
2、学习 NETBISO 请求的报文格式，了解 NETBIOS 使用的端口；
3、理解 WINS 协议和 DNS 协议的不同；
4、了解 CIFS 协议；
5、了解 SMB 协议。

【实验原理】

1、WINS 协议

WINS(Windows Internet Name Server)是 Windows 网际名字服务。网络上的 WINS 服务器提供了一个将 NetBIOS 名解析为 IP 地址的动态数据库。WINS 为 NetBIOS 提供名字注册、更新、释放和解析服务。WINS 服务器收集和维护网络上可用的 NetBIOS 名字并处理名字注册、名字释放及从 WINS 客户机来的名字解析请求。名字解析过程:

（1）当 WINS 客户端想要解析一个名字时，它首先检查本地 NetBIOS 名字缓存器。

（2）如果名字不在本地 NetBIOS 名字缓存器中，便发送一个 WINS 查询到首选 WINS 服务器（每隔 15 秒发送一次，共发三次），如果请求失败，则向次选 WINS 发送同样的请求。

2、NETBIOS

NetBIOS（NETwork Basic Input/Output System）网络基本输入输出系统。NetBIOS 名称和计算机名是一样的，在一个网络内只能使用一次。这些名称是在计算机启动、服务开始或用户登录时动态注册的。

3、SMB/CIFS 协议

SMB(Server Message Block) 简称服务信息块协议，主要用于计算机间局域网共享文件系统、打印机和其他资源的协议。CIFS（common internet file system）是通用互联网文件系统，是 SMB 的一个公共版本。CIFS 和 SMB 解析计算机名的方法不同。SMB 使用 NetBIOS 名的广播和 WINS 解析计算机名，而 CIFS 则使用 DNS。Windows 2000 以前版本中使用 NetBIOS 协议，而 Windows2000 以后的版本使用 CIFS。

（1）SMB 报文的格式：

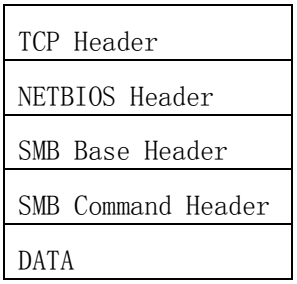


图 14-1 SMB 报文格式

（2）SMB 报文首部格式：

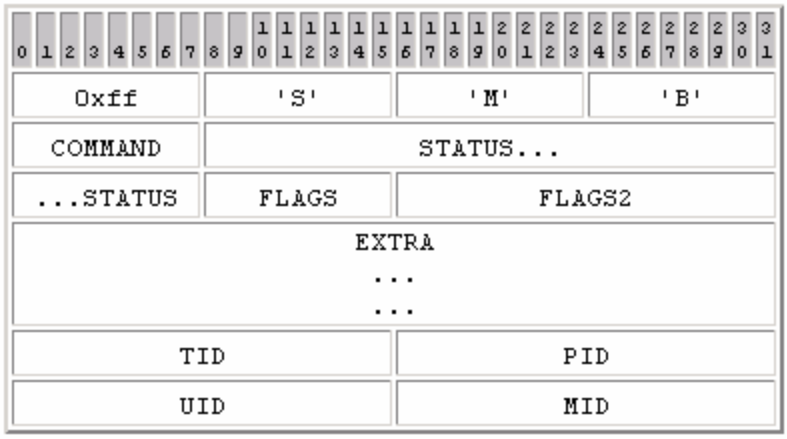


图 14-2　SMB 报文首部格式

协议字段：前四个字节是协议标识字符串，其值总为“0xffSMB”；

COMMAND：命令的参数；

STATUS：错误代码；

FLAGS：标志使用 UNICODE 编码；

EXTRA：扩充；

TID：客户端成功和一台 SMB 服务器上的资源建立连接后被使用的，TID 数字用来鉴别资源；

PID：使用 PID 数字用来鉴别客户端成功在服务器上创建一个进程；

UID：UID 数字用来鉴别一个用户被成功通过验证；

MID：和 PID 同时使用鉴别客户端拥有几个请求(进程，线程，文件访问……)。

4、详细原理请参考教材

【实验步骤】

练习一：利用仿真编辑器编辑 NETBIOS 请求名字服务数据包

利用仿真编辑器编辑 NETBIOS 请求包，在协议分析器端观察结果，理解 WINS 原理。在地址本中找任意一台主机，步骤如下：

1、打开仿真编辑器，在界面初始状态下，程序会自动新建一个单帧，可以利用仿真编辑器打开时默认的以太网帧进行编辑；

2、填写该帧的以太网协议首部：

⑴源 MAC 地址：仿真编辑器的 MAC 地址；

⑵目的 MAC 地址：目的主机 MAC 地址（选定的主机 MAC 地址）；

⑶类型或长度字段：该字段值应为 0800（即 IP 协议的类型值）。

3、填写 IP 协议头信息：

⑴高层协议类型：即上层协议类型应为 17（UDP 协议的类型为 17）；

⑵总长度：应包括 UDP 总长度（20IP+8UDP+50NETBIOS=78）；

⑶源 IP 地址：仿真编辑器的 IP 地址；

⑷目的 IP 地址：目的主机 IP 地址（选定主机的 IP 地址）；

⑸计算 IP 首部校验和。

4、填写 UDP 协议的各个字段信息：

⑴源端口：自选，例如：3092；

⑵目的端口：选择目的主机的 137 端口；

⑶UDP 长度：UDP 报头长度+NETBIOS 长度（58），见图 14-3。

⑷校验和：UDP“校验和”的计算包括 UDP 头、NETBIOS 信息段和 UDP 伪首部；

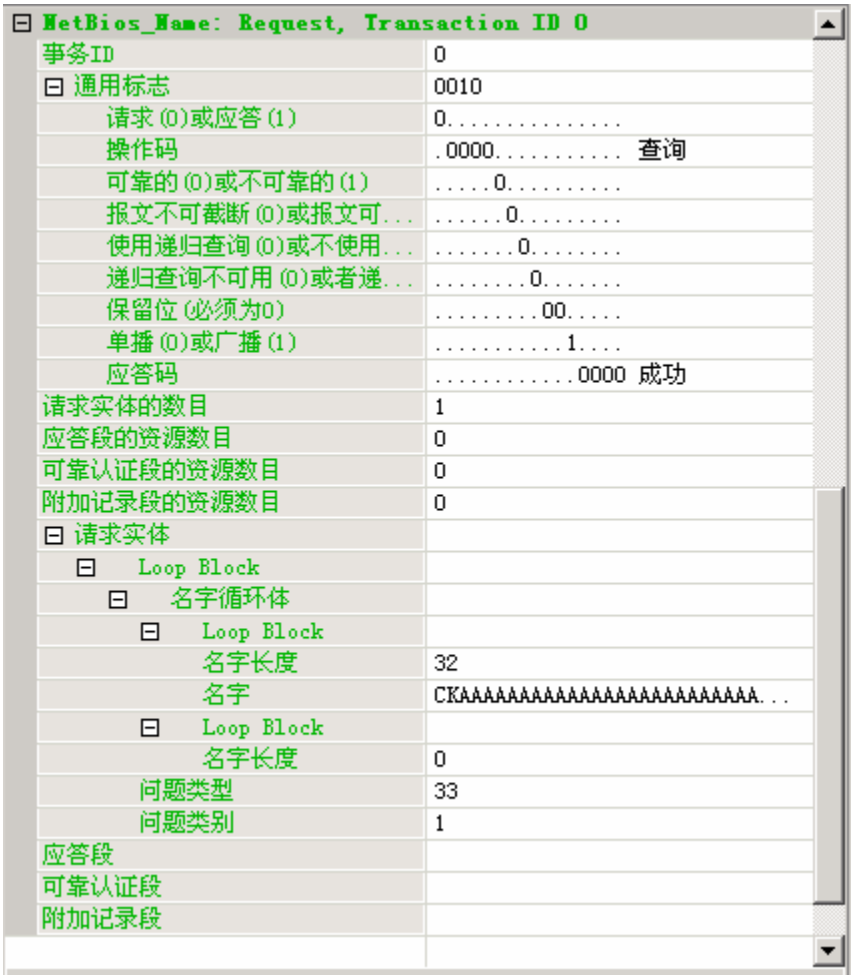


图 14-3 NetBios 协议信息编辑

5、填写 NETBIOS 协议的各个字段信息：

⑴事务 ID：0；

⑵通用标志：0010；

⑶请求实体的数目：1；

⑷应答段的资源数目：0；

⑸可靠认证段的资源数目：0；

⑹附加记录段的资源数目：0；

⑺名字长度：32(在名字循环体中右键追加块，可显示该项)；

⑻名字：CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA；

⑼问题类型：33，NETBIOS 节点状态；

⑽问题类别：1，INTERNET；

6、将编辑好的 NETBIOS 数据帧发送，在协议分析器端截获数据，查找请求包和响应包，分析响应包的内容。

7、从响应包中找到查询主机的主机名称。

练习二：运行 nbtstat 命令

NBTSTAT 命令可以用来查询涉及到 NetBIOS 信息的网络机器。另外，它还可以用来消除 NetBIOS 高速缓存器和预加载 LMHOSTS 文件。这个命令在进行安全检查时非常有用。使用这个命令可以让我们通过网络中的任一台计算机来查看网络中其他计算机的 IP 地址及网卡 MAC 地址。命令格式：

NBTSTAT [[-a RemoteName] [-A IP address] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [interval]]

1、从地址本中找出本网段的一台主机的主机名，在命令提示符界面输入：

c:>nbtstat -a 主机名（或 IP 地址）；

例如：

C:\Documents and Settings\Administrator>nbtstat -a A200

会显示：

本地连接:

```
Node IpAddress: [172.16.1.177] Scope Id: []
          NetBIOS Remote Machine Name Table
          Name          Type          Status
    ---------------------------------------------
    MHX            <00>  UNIQUE      Registered
    MHX            <20>  UNIQUE      Registered
    WORKGROUP      <00>  GROUP       Registered
    WORKGROUP      <1E>  GROUP       Registered
    MHX            <03>  UNIQUE      Registered
    ADMINISTRATOR  <03>  UNIQUE      Registered
    MAC Address = 00-09-44-20-40-0A
```

2、在协议分析器一端捕获数据报文，分析运行 nbtstat 时都使用了什么协议？

记录：

⑴请求包的目的 IP=（　　）；

⑵请求包的源端口号 port=(　　)和目的端口号 port=（　　）；

⑶UDP 长度=（　　）；

⑷请求包 NETBIOS 的 Transaction ID=（　　）；

⑸请求包的 NETBIOS 的类型 Type=（　　）；

⑹响应包的源端口号 port=（　　)和目的端口号 port=(　　)；

⑺响应包 NETBIOS 的 Time To Live=（　　）；

⑻响应包 NETBIOS 的 Transaction ID=（　　）；

⑼响应包 NETBIOS 的 General Flag=（　　）；

⑽响应包 NETBIOS 的名字=（　　）；

3、应用其它的参数运行 nbtstat，观察运行结果：

nbtstat -a　；Lists the remote machine's name table given its name;

nbtstat－c　；Lists NBT's cache of remote names and their IP addresses;

nbtstat－n　；Lists local NetBIOS names;

nbtstat－r　；Lists names resolved by broadcast and via WINS ;

nbtstat -s　；Lists sessions table converting destination IP;

练习三：在网络上“搜索”计算机

1、点击仿真编端主机上的:“开始——>搜索——>网络上的一台计算机”；

2、在“计算机名”一栏输入一个计算机名，可以是地址本中存在的一个计算机名，例如：A200;

3、点击“搜索”，开始在网络上搜索该计算机的过程；

4、在协议分析器一端截获该过程中仿真编辑器发送的请求包和目标主机的响应包，分析该过程中涉及的协议？分析计算机“搜索”功能的实现。

5、根据会话分析记录数据包传输的过程及序号、确认序号、标志位的变化等。

练习四：在计算机之间拷贝文件

1、仿真端在本地磁盘下新建一个共享文件夹，并在文件夹中放置一个文件。

2、分析端访问仿真端的共享文件夹：

方法一：在 IE 浏览器的地址栏中输入：\\仿真端 IP;

方法二：点击:“开始->运行”,并在地址栏中输入：\\仿真端 IP;

2、找到仿真端后，打开其新建的共享文件夹；

3、分析端从仿真端的共享文件夹中将文件拷贝下来，粘贴到本机；

4、从分析端选择一文件拷贝下来，粘贴到仿真端的共享文件夹中；

5、在协议分析器一端捕获数据，分析该过程应用的各种协议；

6、将捕获的相关数据帧保存为文件：实验十四.zdt。

【思考问题】

1、WINS 协议的主要功能是什么？
2、SMB 协议的主要功能是什么？
3、通过实验说明 Windows 网络协议的架构情况。

【实验报告要求】

1、说明 WINS 协议的原理。
2、通过实验，回答练习中的问题。

【实验十五　SOCKET 基础网络 程序设计】

【实验目的】

1、理解进程通信的原理及通信过程；
2、掌握基本的网络编程方法。

【实验学时】

4 学时

【实验类型】

设计型

【实验内容】

1、进一步学习 UDP 及 TCP 协议的工作原理；
2、学习 SOCKET 编程的基本方法；
3、学习应用 C 语言与 WinSock2 进行简单的无连接的网络程序设计，实现网络数据传输；
4、学习应用 C 语言与 WinSock2 进行简单的面向连接的网络程序设计，实现网络数据传输。

【实验原理】

1、关于使用套接字编程的一些基本概念

（1）半相关

网络中用一个三元组可以在全局唯一标志一个进程：（协议，本地地址，本地端口号）。这样一个三元组，叫做一个半相关（half-association），它指定连接的每半部分。

（2）全相关

一个完整的网间进程通信需要由两个进程组成，并且只能使用同一种高层协议。也就是说，不可能通信的一端用 TCP 协议，而另一端用 UDP 协议。因此一个完整的网间通信需要一个五元组来标识：（协议，本地地址，本地端口号，远地地址，远地端口号）。这样一个五元组，叫做一个全相关（association），即两个协议相同的半相关才能组合成一个全相关。

（3）TCP/IP 协议的地址结构为

```
struct sockaddr_in
  {
    short sin_family;                /*AF_INET*/
    u_short sin_port;                /*16 位端口号，网络字节顺序*/
    struct in_addr sin_addr;         /*32 位 IP 地址，网络字节顺序*/
    char sin_zero[8];                /*保留*/
  }
```

（4）套接字类型

TCP/IP 的 socket 提供下列三种类型套接字：

①流式套接字（SOCK_STREAM）

提供了一个面向连接、可靠的数据传输服务，数据无差错、无重复地发送，且按发送顺序接收。内设流量控制，避免数据流超限；数据被看作是字节流，无长度限制。文件传送协议（FTP）即使用流式套接字。

②数据报式套接字（SOCK_DGRAM）

提供了一个无连接服务。数据包以独立包形式被发送，不提供无错保证，数据可能丢失或重复，并且接收顺序混乱。网络文件系统（NFS）使用数据报式套接字。

③原始式套接字（SOCK_RAW）

该接口允许对较低层协议，如 IP、ICMP 直接访问。常用于检验新的协议实现或访问现有服务中配置的新设备。

（5）基本套接字系统调用

为了更好地说明套接字编程原理，下面给出几个基本套接字系统调用说明。

①创建套接字——socket()

应用程序在使用套接字前，首先必须拥有一个套接字，系统调用 socket()向应用程序提供创建套接字的手段，其调用格式如下：

SOCKET socket(int af, int type, int protocol);

该调用要接收三个参数：af、type、protocol。参数 af 指定通信发生的区域，UNIX 系统支持的地址族有：AF_UNIX、AF_INET、AF_NS 等，而 DOS、WINDOWS 中仅支持 AF_INET，它是网际网区域。因此，地址族与协议族相同。参数 type 描述要建立的套接字的类型。参数 protocol 说明该套接字使用的特定协议，如果调用者不希望特别指定使用的协议，则置为 0，使用默认的连接模式。根据这三个参数建立一个套接字，并将相应的资源分配给它，同时返

回一个整型套接字号。因此，socket()系统调用实际上指定了相关五元组中的“协议”这一元。

②指定本地地址——bind()

当一个套接字用 socket()创建后,存在一个名字空间(地址族),但它没有被命名。bind()将套接字地址（包括本地主机地址和本地端口地址）与所创建的套接字号联系起来，即将名字赋予套接字，以指定本地半相关。其调用格式如下：

int bind(SOCKET s, const struct sockaddr FAR * name, int namelen);

参数 s 是由 socket()调用返回的并且未作连接的套接字描述符(套接字号)。参数 name 是赋给套接字 s 的本地地址（名字），其长度可变，结构随通信域的不同而不同。namelen 表明了 name 的长度。

如果没有错误发生，bind()返回 0。否则返回值 SOCKET_ERROR。

地址在建立套接字通信过程中起着重要作用，作为一个网络应用程序设计者对套接字地址结构必须有明确认识。

③建立套接字连接——connect()与 accept()

这两个系统调用用于完成一个完整相关的建立，其中 connect()用于建立连接。无连接的套接字进程也可以调用 connect()，但这时在进程之间没有实际的报文交换，调用将从本地操作系统直接返回。这样做的优点是程序员不必为每一数据指定目的地址，而且如果收到的一个数据报，其目的端口未与任何套接字建立“连接”，便能判断该端口不可操作。而 accept()用于使服务器等待来自某客户进程的实际连接。

connect()的调用格式如下：

int connect(SOCKET s, const struct sockaddr FAR * name, int namelen);

参数 s 是欲建立连接的本地套接字描述符。参数 name 指出说明对方套接字地址结构的指针。对方套接字地址长度由 namelen 说明。

如果没有错误发生，connect()返回 0。否则返回值 SOCKET_ERROR。在面向连接的协议中，该调用导致本地系统和外部系统之间连接实际建立。

由于地址族总被包含在套接字地址结构的前两个字节中，并通过 socket()调用与某个协议族相关。因此 bind()和 connect()无须协议作为参数。

accept()的调用格式如下：

SOCKET accept(SOCKET s, struct sockaddr FAR* addr, int FAR* addrlen);

参数 s 为本地套接字描述符，在用做 accept()调用的参数前应该先调用过 listen()。addr 指向客户方套接字地址结构的指针，用来接收连接实体的地址。addr 的确切格式由套接字创建时建立的地址族决定。addrlen 为客户方套接字地址的长度（字节数）。如果没有错误发生，accept()返回一个 SOCKET 类型的值，表示接收到的套接字的描述符。否则返回值 INVALID_SOCKET。

accept()用于面向连接服务器。参数 addr 和 addrlen 存放客户方的地址信息。调用前，参数 addr 指向一个初始值为空的地址结构，而 addrlen 的初始值为 0；调用 accept()后，

服务器等待从编号为 s 的套接字上接受客户连接请求，而连接请求是由客户方的 connect()调用发出的。当有连接请求到达时，accept()调用将请求连接队列上的第一个客户方套接字地址及长度放入 addr 和 addrlen，并创建一个与 s 有相同特性的新套接字号。新的套接字可用于处理服务器并发请求。

四个套接字系统调用，socket()、bind()、connect()、accept()，可以完成一个完全五元相关的建立。socket()指定五元组中的协议元，它的用法与是否为客户或服务器、是否面向连接无关。bind()指定五元组中的本地二元，即本地主机地址和端口号，其用法与是否面向连接有关：在服务器方，无论是否面向连接，均要调用 bind()；在客户方，若采用面向连接，则可以不调用 bind()，而通过 connect()自动完成。若采用无连接，客户方必须使用 bind()以获得一个唯一的地址。

以上讨论仅对客户/服务器模式而言，实际上套接字的使用是非常灵活的，唯一需遵循的原则是进程通信之前，必须建立完整的相关。

④监听连接——listen()

此调用用于面向连接服务器，表明它愿意接收连接。listen()需在 accept()之前调用，其调用格式如下：

```
int listen(SOCKET s, int backlog);
```

参数 s 标识一个本地已建立、尚未连接的套接字号，服务器愿意从它上面接收请求。backlog 表示请求连接队列的最大长度，用于限制排队请求的个数，目前允许的最大值为 5。如果没有错误发生，listen()返回 0。否则它返回 SOCKET_ERROR。

listen()在执行调用过程中可为没有调用过 bind()的套接字 s 完成所必须的连接，并建立长度为 backlog 的请求连接队列。

调用 listen()是服务器接收一个连接请求的四个步骤中的第三步。它在调用 socket()分配一个流套接字，且调用 bind()给 s 赋于一个名字之后调用，而且一定要在 accept()之前调用。

⑤数据传输——send()与 recv()

当一个连接建立以后，就可以传输数据了。常用的系统调用有 send()和 recv()。

send()调用用于在参数 s 指定的已连接的数据报或流套接字上发送输出数据，格式如下：

```
int send(SOCKET s, const char FAR *buf, int len, int flags);
```

参数 s 为已连接的本地套接字描述符。buf 指向存有发送数据的缓冲区的指针，其长度由 len 指定。flags 指定传输控制方式，如是否发送带外数据等。如果没有错误发生，send()返回总共发送的字节数。否则它返回 SOCKET_ERROR。

recv()调用用于在参数 s 指定的已连接的数据报或流套接字上接收输入数据，格式如下：

```
int recv(SOCKET s, char FAR *buf, int len, int flags);
```

参数 s 为已连接的套接字描述符。buf 指向接收输入数据缓冲区的指针，其长度由 len

指定。flags 指定传输控制方式，如是否接收带外数据等。如果没有错误发生，recv()返回总共接收的字节数。如果连接被关闭，返回 0。否则它返回 SOCKET_ERROR。

⑥输入/输出多路复用——select()

select()调用用来检测一个或多个套接字的状态。对每一个套接字来说，这个调用可以请求读、写或错误状态方面的信息。请求给定状态的套接字集合由一个 fd_set 结构指示。在返回时，此结构被更新，以反映那些满足特定条件的套接字的子集，同时， select()调用返回满足条件的套接字的数目，其调用格式如下：

```
int select(int nfds, fd_set FAR * readfds, fd_set FAR * writefds, fd_set FAR * exceptfds, const struct timeval FAR * timeout);
```

参数 nfds 指明被检查的套接字描述符的值域，此变量一般被忽略。

参数 readfds 指向要做读检测的套接字描述符集合的指针，调用者希望从中读取数据。参数 writefds 指向要做写检测的套接字描述符集合的指针。exceptfds 指向要检测是否出错的套接字描述符集合的指针。timeout 指向 select()函数等待的最大时间，如果设为 NULL 则为阻塞操作。select()返回包含在 fd_set 结构中已准备好的套接字描述符的总数目，或者是发生错误则返回 SOCKET_ERROR。

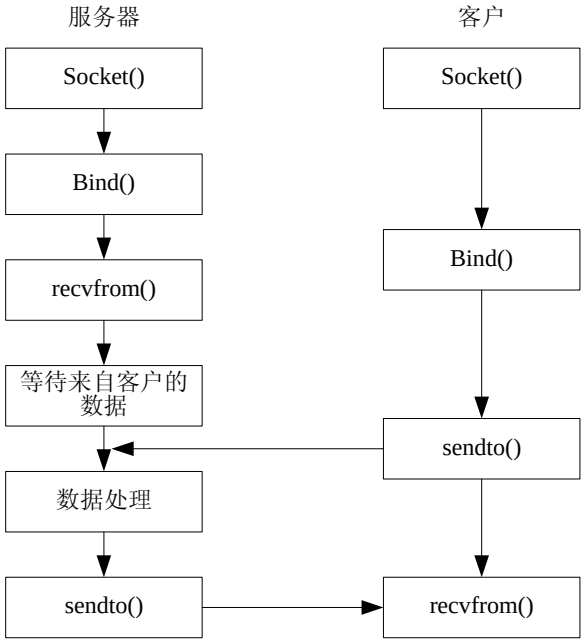
⑦关闭套接字——closesocket()

closesocket()关闭套接字 s，并释放分配给该套接字的资源；如果 s 涉及一个打开的 TCP 连接，则该连接被释放。closesocket()的调用格式如下：

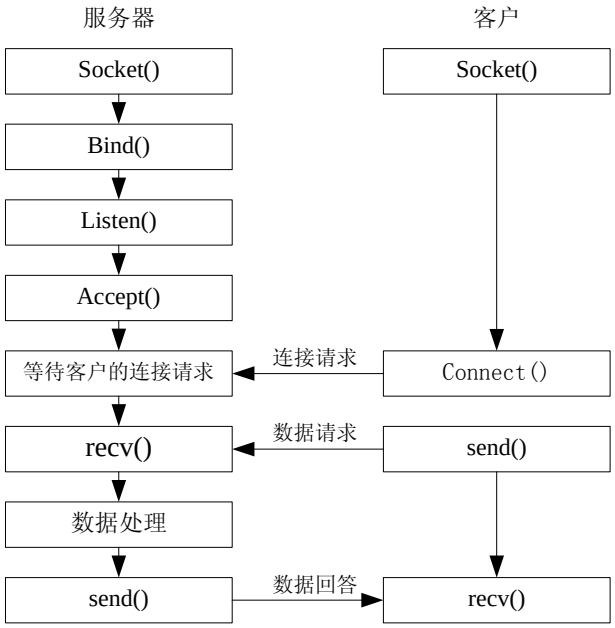
BOOL closesocket(SOCKET s);

参数 s 待关闭的套接字描述符。如果没有错误发生，closesocket()返回 0。否则返回值 SOCKET_ERROR。

2、用于无连接协议（如 UDP）的 SOCKET 系统调用流程框图



3、用于面向连接协议（如 TCP）的 SOCKET 系统调用流程框图



4、详细原理请参考教材

【实验步骤】

练习一：使用 UDP 协议的无连接客户-服务员程序设计

根据实验原理中介绍的内容，设计一个无连接的客户-服务员系统，实现二者之间的数

据传递。下面是一个简单的 UDP 客户-服务员程序的实例，作为参考。

说明：下述服务员程序已经在服务器端运行，在仿真机端仅需编写相应的客户程序并运行。

服务员端程序：

```
#include <stdio.h>
#include <winsock.h>
#pragma comment(lib, "ws2_32.lib")
#define  SERV_UDP_PORT 6000              /*服务员进程端口号，视具体情况而定*/
#define  SERV_HOST_ADDR "10.60.46.58"    /*服务员地址，视具体情况而定*/
/* 宏定义用来打印错误消息*/
#define PRINTERROR(s)
        fprintf(stderr,"\n%: %d\n", s, WSAGetLastError())

////////////////////////////////////////////////////////
//数据报通信的服务员端子程序                          //
////////////////////////////////////////////////////////
void DatagramServer(short nPort)
{
    SOCKET theSocket;
    /*创建一个数据报类型的 socket*/
    theSocket = socket(AF_INET,     // 地址族
                       SOCK_DGRAM,  // socket 类型
                       IPPROTO_UDP);// 协议类型：UDP
    /*错误处理*/
    if (theSocket == INVALID_SOCKET)
    {
        PRINTERROR("socket()");
        return;
    }
    /*填写服务员地址结构*/
    SOCKADDR_IN saServer;
    saServer.sin_family = AF_INET;
    saServer.sin_addr.s_addr = INADDR_ANY; // 由 WinSock 指定地址
    saServer.sin_port = htons(nPort);      // 服务员进程端口号
    /* 将服务员地址与已创建的 socket 绑定*/
    int nRet;
```

```
nRet = bind(theSocket,                  // Socket 描述符
            (LPSOCKADDR)&saServer,  // 服务员地址
            sizeof(struct sockaddr) //地址长度
            );
/*错误处理*/
if (nRet == SOCKET_ERROR)
{
    PRINTERROR("bind()");
    closesocket(theSocket);
    return;
}
/* 等待来自客户端的数据*/
SOCKADDR_IN saClient;
char szBuf[1024];
int nLen = sizeof(saClient);
while(1)
{
    /*准备接收数据*/
    memset(szBuf, 0, sizeof(szBuf));
    nRet = recvfrom(theSocket,          // 已绑定的 socket
                szBuf,                  // 接收缓冲区
                sizeof(szBuf),          // 缓冲区大小
                0,                      // Flags
                (LPSOCKADDR)&saClient,  // 接收客户端地址的缓冲区
                &nLen);                 // 地址缓冲区的长度
    if( nRet > 0 )
    {
        /*打印接收到的信息*/
        printf("\nData received: %s", szBuf);
        /* 发送数据给客户端*/
        strcpy(szBuf, "From the Server");
        sendto(theSocket,                   // 已绑定的 socket
            szBuf,                          // 发送缓冲区
            strlen(szBuf),                  // 发送数据的长度
            0,                              // Flags
            (LPSOCKADDR)&saClient,          // 目的地址
```

```
                nLen);                          // 地址长度
        }

    }
    closesocket(theSocket);
    return;
}
////////////////////////////////////////////////////////
//数据报服务员端主程序                                    //
////////////////////////////////////////////////////////
void main()
{
    WORD wVersionRequested = MAKEWORD(1,1);
    WSADATA wsaData;
    int nRet;
    short nPort;

    nPort = SERV_UDP_PORT;
    /* 初始化 Winsock*/
    nRet = WSAStartup(wVersionRequested, &wsaData);
    if (wsaData.wVersion != wVersionRequested)
    {
        fprintf(stderr,"\n Wrong version\n");
        return;
    }
    /*调用数据服务员子程序*/
    DatagramServer(nPort);
    /*结束 WinSock*/
    WSACleanup();
}
```

上述服务员程序已经在服务器端运行，请学生认真阅读分析后，然后根据实验原理二中介绍的内容，在仿真机端编写相应的无连接的客户端程序并运行。从而实现客户和服务器间的数据传输。

在仿真机一端运行客户端进程，在协议分析端捕获数据并进行分析。

练习二：使用 TCP 协议的面向连接的客户-服务员程序设计

根据实验原理中介绍的内容，设计一个面向连接的客户-服务员系统，实现二者之间的

数据传递。

下面是一个简单的 TCP 客户-服务员程序的服务员程序：

面向连接的服务员程序：

```
#include <PROCESS.H>
#include <windows.h>
#include <winsock.h>
#include <sys/types.h>
#include <fcntl.h>
#include <wsipx.h>
#include <wsnwlink.h>
#include <stdio.h>
#define SERV_TCP_PORT 6000                    /*服务员进程端口号，视具体情况而定*/
#define SERV_HOST_ADDR "10.60.46.40"          /*服务员 IP，视具体情况而定*/
int sockfd;
#pragma comment (lib, "ws2_32.lib")

//////////////////////////////////////////////////////////////////////
//线程用来处理客户端的请求                                          //
//服务员主进程每与某客户端建立一个连接之后，便启动一个新的线程来处理接下//
//客户端的请求，参数为服务员与该客户端的连接点：socket             //
//////////////////////////////////////////////////////////////////////

DWORD ClientThread(void *pVoid)
{
    int nRet;
    char szBuf[1024];
    memset(szBuf, 0, sizeof(szBuf));
    /*接收来自客户端的数据信息*/
    nRet = recv((SOCKET )pVoid,                  //与客户端连接的 socket
                szBuf,                           // 接收缓冲区
                sizeof(szBuf),                   // 缓冲区长度
                0);                              // Flags
    /*错误处理*/
    if (nRet == INVALID_SOCKET)
    {
        printf("recv()");
```

```
        closesocket(sockfd);
        closesocket((SOCKET)pVoid);
        return 0;
    }
    /*显示接收到的数据*/
    printf("\nData received: %s\n", szBuf);
    /* 发送数据给客户端*/
    strcpy(szBuf, "From the Server");        //发送内容
    nRet = send((SOCKET)pVoid,               //与客户端连接的 socket
                szBuf,                       // 数据缓冲区
                strlen(szBuf),               // 数据长度
                0);                          // Flags
    /*结束连接，释放 socket*/
    closesocket((SOCKET)pVoid);
    return 0;
}
//////////////////////////////////////////////////////////
//服务员主程序：                                        //
//在一个众所周知的端口上等待客户的连接请求              //
//有请求到来时建立与客户端的连接，并启动一个线程处理该请求  //
//////////////////////////////////////////////////////////

int main()
{
    int  clilen;
    int pHandle=-1;
    struct sockaddr_in  serv_addr;
    SOCKET      socketClient;
    DWORD   ThreadAddr;
    HANDLE      dwClientThread;
    SOCKADDR_IN SockAddr;

    /*初始化 Winsock API，即连接 Winsock 库*/
    WORD wVersionRequested = MAKEWORD(1, 1);
    WSADATA wsaData;
    if (WSAStartup(wVersionRequested, &wsaData)) {
```

```
    printf("WSAStartup failed %s\n", WSAGetLastError());
    return -1;
}
/*打开一个 TCP SOCKET */
if((sockfd=socket(AF_INET, SOCK_STREAM,0))<0)
    printf("server:can't open stream socker\n");
/*绑定本地地址，以便客户端连接*/
    memset((char *)&serv_addr,0,sizeof(struct sockaddr_in));
    serv_addr.sin_family=AF_INET;
    serv_addr.sin_addr.s_addr=htonl(INADDR_ANY);
    serv_addr.sin_port=htons(SERV_TCP_PORT);
if(bind(sockfd,(struct sockaddr *)&serv_addr, sizeof(serv_addr))<0)
        printf("server: can't bind local address");
/*设置服务员的最大连接数为 15*/
listen(sockfd,5);
/*循环等待来自客户端的连接请求*/
while(1)
{
    /*阻塞等待一个请求的到来*/
    clilen = sizeof(SOCKADDR_IN);
    socketClient = accept(sockfd,
                          (LPSOCKADDR)&SockAddr,
                          &clilen);
    /*出错处理*/
    if (socketClient == INVALID_SOCKET)
    {
        printf("accept failed!\n");
        break;
    }
    /* 打印已建立的连接信息*/
     printf("Connection accepted on socket:%d from:%s\n",
             socketClient,
             inet_ntoa(SockAddr.sin_addr));
    /*启动一个新线程处理该请求*/

    dwClientThread =CreateThread(NULL,
```

```
                                        0,
                                        (LPTHREAD_START_ROUTINE)&ClientThread,
                                        (void *)socketClient,
                                        0,
                                        &ThreadAddr);
        /*错误处理*/
        if(!dwClientThread)
            printf("Cannot start client thread...");
         /*线程建立以后，主程序里不再使用线程 handle,将其关闭，但线程继续运行*/
        CloseHandle((HANDLE)dwClientThread);
    }

    /*结束 windows sockets API*/
    WSACleanup();
    return 0;
}
```

上述服务员程序已经在服务器端运行，请学生认真阅读，然后根据实验原理二中介绍的内容，设计面向连接的客户端程序，实现客户与服务员间的数据传输。

在仿真机一端运行客户端进程，在协议分析端捕获数据并分析。

练习三：使用 TCP 协议进行复杂的客户-服务员程序设计

上述的例子程序比较简单，有一些进程中往往同时存在几条连接，这样的进程在有报文到来时，可以在它处理的任何 socket 上执行 recv 调用，但它不知道哪个 socket 上已有报文，哪个上没有，可以使用 select（）系统调用来解决这样的问题。有兴趣的同学可自行编写这样较复杂的客户-服务员程序。

【思考问题】

结合实验过程中的实验结果，回答下列问题：

1、根据编程练习实验中记录的客户和服务器程序的端口号并结合程序，说明：

在客户/服务器模型当中，客户进程的端口号和服务器进程的端口号都是由程序给出说明的吗？为什么？

2、在 TCP/IP 网络中，当客户与服务员进程建立了一条 TCP 连接以后，是否属于该连接的所有包都是经过同一路径（即一条虚电路）传递的？为什么？

【实验报告要求】

1、网络程序设计体会。

【实验十六　Ping 实现 程序设计】

【实验目的】

1. 加深对 ICMP 协议的理解；
2. 掌握原始套接字进行网络程序设计的方法。

【实验学时】

建议 4 学时

【实验类型】

设计型

【实验内容】

1、需求分析；
2、学习原始套接字（Raw Socket）的使用；
3、ICMP 回显请求数据包的定义及发送；
4、ICMP 回显应答数据包的接收。

【实验步骤】

1. 需求分析：

使用原始套接字实现 ping 命令的功能(程序语言不限，应用程序采用控制台、GUI 方式均可)。

- 可以指定发送报文数。
- 可以指定发送报文长度。
- 可以指定发送报文的 TTL 值。

2. 代码实现：

按需求编写程序代码。参考代码见：在服务器上程序范本。

3. 执行程序，检验程序的正确性：

a. 启动协议分析器捕获数据；
b. 运行已编译好的程序，ping 对方 IP；
c. 察看运行结果，分析捕获的数据。

【实验十七　文件传输工具 程序设计】

【实验目的】

1. 加深对 TCP 协议的理解；
2. 掌握面向连接并发服务器模型。

【实验学时】

建议 4 学时

【实验类型】

设计型

【实验内容】

1、需求分析；
2、C/S 模式的程序设计；
3、一对多通迅模式（多线程）。

【实验步骤】

1. 需求分析：

使用套接字实现基于 TCP 协议的文件传输工具--客户端向服务器端传输文件(程序语言不限，应用程序采用控制台、GUI 方式均可)。

- 服务器指定开放端口，在同一时刻可以和多个客户端建立连接，即多个客户端可以同时上传文件。
- 客户端指定上传文件的本地路径和远程路径名。文件上传完毕自动关闭连接。

2. 客户端代码实现：

仿真端按对客户端的需求编写客户端程序代码。参考代码见：在服务器上程序范本。

3. 服务器段代码实现：

分析端按对服务器端的需求编写服务器端程序代码。参考代码见：在服务器上程序范本。

4. 执行程序，检验程序的正确性：

a. 启动协议分析器捕获数据；
b. 仿真端使用客户端程序向分析端传输文件，分析端使用服务器端程序进行接收；
c. 察看运行结果，分析捕获的数据。

【实验十八　主机信息获取工具 程序设计】

【实验目的】

1. 掌握使用套接字获取主机信息的方法；
2. 掌握使用 WMI 获取主机信息的方法；
3. 掌握 IP 助手函数(IP Help)的使用方法。

【实验学时】

建议 4 学时

【实验类型】

设计型

【实验内容】

1、 需求分析；
2、 本地、远程主机信息获取；
3、 IP 助手的使用；
4、 本机路由表的操作。

【实验步骤】

按照拓扑结构图连接网络，使用拓扑验证检查连接的正确性。

练习一：利用套接字或 WMI 等方法获取主机基本信息

本练习将仿真端和分析端双方相互获取信息。

1. 需求分析：

   使用套接字、WMI 或其它方法获取主机信息(程序语言不限，应用程序采用控制台、GUI 方式均可)。

   - 本机：主机名、CPU 类型、网卡信息、磁盘分区信息、内存容量及使用状况等信息。
   - 其它主机：主机名、CPU 类型、网卡信息、磁盘分区信息、内存容量及使用状况等信息。

2. 代码实现：

   按需求编写程序代码。参考代码见：在服务器上程序范本。

3. 执行程序，检验程序的正确性：

   运行程序将结果与实际信息进行对比。

练习二：利用 IP 助手管理路由表

1. 需求分析：

使用 IP 助手管理本机路由表。

- 可以显示本机路由表。
- 可以增加本机路由表条目。
- 可以删除本机路由表条目。

2. 代码实现：

   按需求编写程序代码。参考代码见：在服务器上程序范本。

3. 执行程序，检验程序的正确性：

   运行程序将结果与运行命令“route print”得出结果进行对比。

【实验十九　网络性能监测及故障分析】

【实验目的】

1、检验学生对 TCP/IP 协议原理的掌握情况，以及综合分析问题的能力；
2、掌握监测网络性能的基本方法；
3、掌握网络性能问题的基本分析方法。

【实验学时】

2 学时

【实验类型】

综合型

【实验内容】

1、学习使用协议分析器进行网络性能监测的基本方法；
2、能够根据性能检测结果分析网络性能问题。

【实验原理】

1、网络检测工具：

（1）netstat

安装了 TCP/IP 协议的 Windows 系统可以使用 netstat 来检查网络的一些情况，给网络安全维护带来了方便。

命令的用法：NETSTAT [-a] [-e] [-n] [-s] [-p proto] [-r] [interval]

-a ；显示所有连接和侦听端口。此命令可以显示出你的计算机当前所开放的所有端口，其中包括 TCP 端口和 UDP 端口。管理员使用它来查看计算机的系统服务是否正常，是否被“黑客”留下后门，木马等。在刚刚装了系统配置好服务器以后，运行一下 netstat -a 看看系统开放了什么端口，并记录下来，以便以后作为参考使用，当发现有不明的端口时就可以及时的做出对策。由于这个参数同时还会显示出当前计算机有什么人的 IP 正连接着你的服务器，所以也是一种实时入侵检测工具，如发现有 IP 连接着不正常的端口，可以及时做出有效对策。例如：

C:\>netstat -a

C:\Documents and Settings\Administrator>netstat -a

Active Connections

Proto	Local Address	Foreign Address	State
TCP	lenovo-b80d9bfe:epmap	lenovo-b80d9bfe:0	LISTENING
TCP	lenovo-b80d9bfe:microsoft-ds	lenovo-b80d9bfe:0	LISTENING
TCP	lenovo-b80d9bfe:netbios-ssn	lenovo-b80d9bfe:0	LISTENING
TCP	lenovo-b80d9bfe:microsoft-ds	ws70-45.jlu.edu.cn:1558	ESTABLISHED
TCP	lenovo-b80d9bfe:1202	mail.jlu.edu.cn:http	CLOSE_WAIT

-e ；显示以太网统计。

-n ；以数字格式显示地址和端口号 C:\>netstat

-r ；显示路由表的内容。

-s ；显示每个协议的统计。默认情况下，显示 TCP、UDP、ICMP 和 IP 的统计。

（2）nbtstat 命令

该诊断命令使用 NBT（TCP/IP 上的 NetBIOS）显示协议统计和当前 TCP/IP 连接。命令格式：

NBTSTAT [[-a RemoteName] [-A IP address] [-c] [-n] [-r] [-R] [-RR] [-s] [-S] [interval]]

-a ；使用远程计算机的名称列出其名称表。此参数可以通过远程计算机的 NetBios 名来查看他的当前状态。例 ：

C:\>nbtstat -a 10.60.48.200

-c；给定每个名称的 IP 地址并列出 NetBIOS 名称缓存的内容。

-n；列出本地 NetBIOS 名称。

-r；列出 Windows 网络名称解析的名称解析统计。

-R；清除 NetBIOS 名称缓存中的所有名称后，重新装入 Lmhosts 文件。

-S；显示客户端和服务器会话，只通过 IP 地址列出远程计算机。

-s ；显示客户端和服务器会话。将远程计算机 IP 地址转换成使用主机文件的名称。

（3）ping 命令

用于测试网络连通性。

命令格式：ping IP 地址或主机域名

（4）tracert 命令

tracert 是 traceroute 的 Windows 版本。通过利用 IP 分组的 TTL 值，从源站到目的路径中路由器产生 ICMP 超时差错报文，从而了解路由的具体情况。

命令格式：tracert IP 地址或主机域名

2、网络故障分析和监测工具

网络发生故障以后可以利用一些工具和设备去分析检查故障原因。常见的工具有：

(1) Fluke Networks 的 OptiViwe

(2) SolarWinds2002

(3) MRTG 网络流量监控软件

(4) 协议分析器教学软件

【实验步骤】

练习一：netstat 命令

netstat 是有用的网络工具之一，能列出与网络有关的核心数据结构的内容。可以根据 TCP 和 UDP 来分析通信状态，发现网络中的异常。

在命令行方式下运行：

1、netstat -s ；显示协议统计信息，包括 TCP、UDP、ICMP、IP 协议的使用情况。

2、netstat -e ；显示以太网统计信息。

3、netstat ？；查看其他参数的功能并运行。

练习二：nbtstat 命令

在命令行方式下运行：

1、C:\>nbtstat -a 服务器 IP;

2、C:\>nbtstat ？；查看其他参数的功能并运行。

练习三：tracert 命令

在命令行方式下运行：

1、tracert d61.jlu.edu.cn

2、tracert bbs.jlu.edu.cn

练习四：编辑并发送帧序列，捕获数据并分析

1、打开仿真编辑器，编辑帧序列，通过设置发送次数及发送时间间隔来控制发包的速度和数量，可发送大批量的数据。

2、在协议分析器一端捕获已发送的数据，并利用协议分析器的统计功能进行分析。

3、当仿真编辑器不停地向网络上发送大量的数据包时，在协议分析器端观察网络的流量变化。

练习五：性能分析

在协议分析器上捕获局域网上所有的数据包，并利用协议分析器的各种统计功能进行实时分析和判断。

1、切换数据采集器上的按钮，将协议分析器连接到整个局域网当中。

2、会话分析：点击协议分析器的“会话分析”功能，察看会话情况。

3、协议解析：点击协议分析器的“协议解析”功能，察看主机发送和接收信息的情况，分析网络性能。

练习六：大量数据帧的分析

1、切换数据采集器上的按钮，将协议分析器连接到局域网当中；

2、启动协议分析端的协议分析器，捕获 2 分钟内流经局域网上的所有数据帧；

3、在这 2 分钟之内，在部分仿真机上运行几种不同的网络应用程序（如 WWW 服务、邮件服务等）；

4、将捕获到的数据帧保存到文件：综合实验.zdt。

5、对捕获到的所有数据帧进行以下分析：

①其中有多少 MAC 会话？

②其中有多少 IP 会话？

③其中有多少 UDP 会话？

④其中有多少 TCP 会话？

⑤在捕获期间，存在多少 TCP 连接？有多少 TCP 连接失败？有多少 TCP 连接正常中断？在结束捕获时，有多少 TCP 连接是活跃的？

⑥共使用了多少种不同的协议？各协议都属于 TCP/IP 协议族的哪一层次？

⑦对 IP 的上层协议进行统计，计算各协议（TCP、UDP 等协议）所占的百分比，从而对 TCP 和 UDP 协议的使用情况进行比较。

【思考问题】

1、在实验的过程中，网络上的峰值流量是多少？

2、网络是否出现性能严重下降的情况，如果有，分析产生的原因。

3、请分析在真实的广域网环境中，UDP 与 TCP 协议的使用情况。

【实验报告要求】

1、分析影响网络性能的各个参数指标？

2、分析常见故障的原因，并提出解决方案。

【附录：常见问题及解决方法】

常见问题：

1、主机无法上网：

①主机的 TCP/IP 协议设置是否正确选中（有时实验需要取消 TCP/IP 协议，但实验结束时应该将 TCP/IP 协议选中）。

②IP 地址设置是否正确。

③可以使用 PING 命令测试，如：PING 10.60.48.200。

2、监控机无法收到数据包：

①实验环境下，关闭仿真机和监控机的防火墙，否则无法收到数据。

②查看分析端主机的两个网卡是否都已启动。

③检查分析端主机的过滤器设置是否正确。

④数据采集器（TAP）的开关按下时是接入仿真机，开放时是接入主控设备。当仿真机无法直接和分析端主机通信时，可能是由于开关的位置问题。

⑤发送的数据报没有回应时，可能出现的问题是：目标地址（IP、MAC）错、目标端口没开放或错、校验和错、不应该回应信息的协议等。

3、无法得到目的主机的响应：

①可能是源地址（MAC、IP）填错，导致数据包无法返回。

②DNS 实验时，域名服务器不响应，可能是：目标地址错、校验和错、源端口不能为 0、目的端口错等等。

③目的主机的端口没有开放。

4、实验环境：

①实验室主控设备 IP 地址由学校自行设定，在吉大是 10．60．48．200。

②开放了 FTP 服务：用户：guest，密码：空；

③TELNET 服务：用户：guest，密码：空；

④EMAIL 服务： ID=groupX, PW=groupX ； groupX@NetLab （X 为 1 到 60）

常见网络专用名词的中英文对照：

ACK (ACKnowledgment) TCP 首部中的确认标志

CRC (Cyclic Redundancy Check) 循环冗余检验

ARP (Address Resolution Protocol) 地址解析协议

CSMA (Carrier Sense Multiple Access) 载波侦听多路存取

DHCP (Dynamic Host Configuration Protocol) 动态主机配置协议

DNS (Domain Name System) 域名服务解析协议

DSAP (Destination Service Access Point) 目的服务访问点

FIN (FINish) TCP 首部中的结束标志

FTP (File Transfer Protocol) 文件传送协议

HDLC (High-level Data Link Control) 高级数据链路控制
HTTP (Hypertext Transfer Protocol) 超文本传输协议
ICMP (Internet Control Message Protocol) Internet 控制报文协议
IGMP (Internet Group Management Protocol) Internet 组管理协议
IMAP (Internet Message Access Protocol) Internet 报文存取协议
ISN (Initial Sequence Number) 初始序号
IP (Internet Protocol) 网际协议
ISO (International Organization for Standardization) 国际标准化组织
LAN (Local Area Network) 局域网
LLC (Logical Link Control) 逻辑链路控制
LCP (Link Control Protocol) 链路控制协议
MAC (Media Access Control) 媒体访问控制
MIB (Management Information Base) 管理信息库
MIME (Multipurpose Internet Mail Extensions) 通用 Internet 邮件扩充
PDU(Protocol Data Unit)协议数据单元
POP3 (Post Office Protocol-Version 3) 邮局协议
PPP (Point-to-Point Protocol) 点对点协议
SACK (Selective ACKnowledgment) 有选择的确认
SLIP (Serial Line Internet Protocol) 串行线路 Internet 协议
SMTP (Simple Mail Transfer Protocol) 简单邮件传送协议
SNMP (Simple Network Management Protocol) 简单网络管理协议
SSAP (Source Service Access Point) 源服务访问点
SYN (SYNchronous) TCP 首部中的同步序号标志
TCP (Transmission Control Protocol) 传输控制协议
TTL (Time-To-Live) 生存时间或寿命
Telnet 远程终端协议
UDP (User Datagram Protocol) 用户数据报协议
WAN(Wide Area Network)广域网
WWW (World Wide Web) 万维网
URG (URGent) TCP 首部中的紧急指针标志
TFTP (Trivial File Transfer Protocol) 简单文件传送协议
UA (User Agent) 用户代理