**网络技术实践**

**实验报告**

**学号：1120141945**

**姓名：高子恺**

**班级：07111406**

**北京理工大学**

**计算机学院**

**2017年10月**

**实验七 WinPcap 捕获数据帧程序**

1. **设计要求**

**1．实验目的**

网络数据包捕获工具是进行网络协议分析、故障检测的主要手段，WinPcap是网络包捕获工具包，可以实现网卡级数据帧的捕获。

* 理解以太网数据帧的格式
* 掌握 WinPcap 库的编程方法
* 理解网络各层包之间的包涵关系

**2．实验内容**

利用 WinPcap 库函数编写网络数据帧（以太网）捕获程序，首先捕获通过本机网卡的数据链路帧，根据以太网帧格式显示不同的字段，也可以进一步分析IP、TCP/UDP 上层包格式。

**3．实验环境**

程序运行环境为以太网，采用 TCP/IP 协议栈，网络操作系统为 Windows。程序开发环境为 Visual C++6.0 版本，支持库采用 WinPcap。

**4．实验步骤**

**步骤 1** 需求分析

WinPcap 捕获数据帧程序功能为：

（1） 捕获程序具有捕获本机网卡数据帧的能力，并显示以太帧头各个字段值

（2） 捕获程序具有进一步分析上层协议数据单元的能力

（3） 将网卡设置为混杂模式，以接收网段的数据帧并进行分析

**步骤 2** 捕获帧程序开发

用 Visual C++编写数据帧捕获程序，利用 WinPcap 库函数实现网卡数据链路帧的嗅探。数据帧捕获程序首先获取本机网卡的适配器号，将其设置为混杂模式，以接收本网段的所有数据帧，在驱动程序中设置缓冲，定位、初始化设备后，开始捕获数据帧，按照以太帧格式显示各个字段，进一步显示上层协议数据单元的格式。

**步骤 3** 编译和执行程序

将 WinPcap 数据帧捕获程序编译、连接成执行程序，运行数据帧捕获程序。程序循环显示捕获的数据帧和上层协议的帧格式。

1. **系统总体设计说明**

安装WinPcap包后，通过调用WinPcap 库函数实现网卡数据链路帧的嗅探。数据帧捕获程序首先获取本机网卡的适配器号，将其设置为混杂模式，以接收本网段的所有数据帧，在驱动程序中设置缓冲，定位、初始化设备后，开始捕获数据帧，按照以太帧格式显示各个字段，进一步显示上层协议数据单元的格式。

**三、编程/开发环境**

操作系统：Windows 10专业版

处理器：Intel(R) Xeon(R) CPU E5-2660 v2 @2.20GHz

内存：32GB

系统类型：64位操作系统，基于x64的处理器

IDE：Microsoft Visual Studio Community 2017 版本 15.2 (26430.15) Release

**四、系统设计详细说明**

**步骤1**需求分析

WinPcap 捕获数据帧程序功能为：

1. 捕获程序具有捕获本机网卡数据帧的能力，并显示以太帧头各个字段值；
2. 捕获程序具有进一步分析上层协议数据单元的能力；
3. 将网卡设置为混杂模式，以接收网段的数据帧并进行分析。

**步骤2** 捕获帧程序开发

用 Visual Studio2013编写数据帧捕获程序，利用WinPcap 库函数实现网卡数据链路帧的嗅探。数据帧捕获程序首先获取本机网卡的适配器号，将其设置为混杂模式，以接收本网段的所有数据帧，在驱动程序中设置缓冲，定位、初始化设备后，开始捕获数据帧，按照以太帧格式显示各个字段，进一步显示上层协议数据单元的格式。

**步骤3** 编译和执行程序

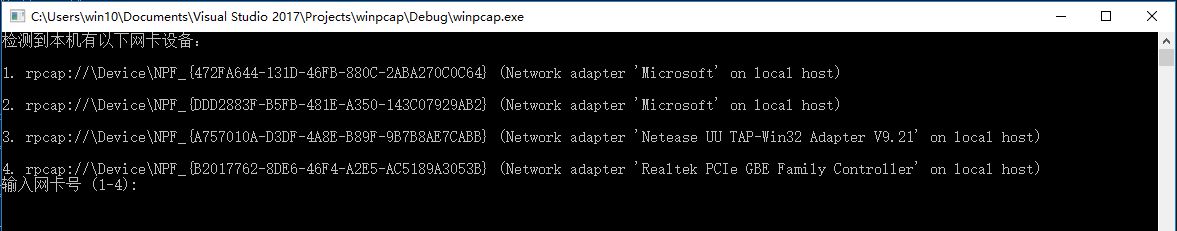
将 WinPcap 数据帧捕获程序编译、连接成执行程序，运行数据帧捕获程序。程序循环显示捕获的数据帧和上层协议的帧格式。

1. **主要函数说明**

|  |  |
| --- | --- |
| 函数名 | 函数功能 |
| pcap\_findalldevs\_ex() | 获取本机设备列表 |
| pcap\_open() | 打开网卡适配器文件、准备记录数据 |
| pcap\_compile() | 编译过滤规则 |
| pcap\_setfilter() | 设置过滤规则 |
| pcap\_loop() | 抓取网络数据报文 |
| packet\_handler() | 包捕捉回调函数 |
| pcap\_freealldevs() | 释放设备 |
| pcap\_close() | 关闭pcap库 |

1. **实验结果**

执行winpcap.exe文件，获取本机网卡设备：



监听结果：

