## 基于Winpcap的网络流量统计分析系统的设计与实现

## 实验内容描述

本实验是用java实现的网络抓包程序，在windows环境下安装winpcap4.0和jpcap6.0后，下载eclipse和jigloo插件（一种在eclipse底下作图形化开发的工具），将其安装好，然后就可以进行java的网络抓包图形化开发了。

基础功能：

(1) 完成局域网数据包的捕获和统计，能够识别并统计各类数据包，包括TCP、UPD、ICMP、ARP、广播数据包等；

(2) 能够捕获一段时间的数据包，分析统计各类数据包的数量；

(3) 能够图形化显示数据包统计结果；

(4) 进行简单的流量分析

扩展功能：

1. 可自定义过滤器过滤条件，自定义捕获数据包最大字长，而有选择地捕获数据包；
2. 对捕获的数据包可查看数据报时间、源IP地址、目的IP地址、首部长度、数据长度、是否分段、分段偏移量、首部内容、数据内容等属性。
3. 对捕获结果进行简单流量分析并柱状图显示，并可保存结果至本地。

## 原理与关键技术

2.1 网络抓包技术原理

网络层上有各种各样的数据包，它们以不同的帧格式在网络层上进行传输，但是在传输时它们都遵循相同的格式，即有相同的长度，如果一种协议的帧格式达不到这种长度，就让其补齐，以达到我们的要求。

2.2 网络抓包关键技术

无论是在windows操作系统下还是在linux操作系统下，要想捕获网络上的数据包，必须要对网卡进行控制，因为本机的数据报从网络上来到本机是通过网卡然后再保存到本地缓冲区上的，所以要抓获网包就必须调用网卡驱动中的对外函数，在linux系统中有net.h文件，可以调用net.h文件中的函数来操作网卡，可以直接编程实现，但为了更方便的使用，可以安装一个叫libpcap的软件，这样调用函数更好用，而在windows系统中，因为源代码不对外公开，所以要安装一个叫winpcap的软件，这样用C或VC++就可以实现了，但因为我用的是java语言来实现的，所以无论是在哪个系统都要安装一个叫jpcap的软件，它本身就把底层的函数又封装了一下，这样就可以让java来使用了。

## 三、设计与实现

3.1 基于java的设计方案

我的这个网络抓包程序是图形化操作界面，在菜单栏点击抓包按钮后选择网卡和过滤字还有最长字长，点击开始，然后就可以开始抓包了，在主界面中就会显示出一行又一行的数据，这些数据就是抓获到的数据包。

3.2 具体实现

1、安装winpcap4.0和jpcap6.0

2、下载eclipse3.3和jigloo，jigloo是eclipse底下的插件，是用来支持eclipse底下的java 图形化开发的。

3、编写java抓包程序：

核心文件有四个：主程序；抓包程序；抓包选项程序对话框程序；流量统计

非核心文件有两个：柱状图显示分析；文件自定义保存

**主程序文件**：提供主界面及其控件的显示，并可通过菜单功能点击间接调用其它程序文件，

文件名：JFrameMain.java

在这里定义了一个向量r，当有数据包产生时，便将数据包赋值给r，rows.AddElement（r）语句便将r添加到向量rows中，然后tabledisplay.addNotify();语句就会刷新界面将新的数据包显示出来。

**抓包程序**：抓包核心程序，由Jcapturedialog调用，包括“开始”、“停止”等功能

文件名：Netcaptor.java

定义一个抓包对象JpcapCaptor jpcap = null;但点击开始时调用private void startCaptureThread()方法开始抓包，jpcap.processPacket(1, handler);语句能够反复调用handler所指向的方法，这个方法中定义的packet就是网络上抓到的数据包，经过frame.dealPacket(packet);就可以显示在主界面上。

**抓包选项程序对话框**：提供用户自定义过滤条件，选择是否为混杂模式，自定义捕获字长，选择网卡等数据包捕获属性自定义接口。由主程序调用

文件名：Jcapturedialog.java

这一部分主要是界面操作，根据jigloo插件对不同的按钮和文本框还有其他的组件设置监听操作，以激发不同的函数操作，主要是devices = JpcapCaptor.getDeviceList();语句和

jpcap=JpcapCaptor.openDevice(devices[netJComboBox.getSelectedIndex()],caplen,

CheckBox.isSelected(),50);语句要选择一下监听的网卡，比如说笔记本就有两个网卡，一个无线一个有线，选择一下就会监听相应的网卡。函数

publicstatic JpcapCaptor getJpcap(JFrame parent){

new Jcapturedialog(parent).setVisible(true);

returnjpcap;

}

返回jpcap，这个jpcap就是对应的选择上的网卡对象，接下来就从对应的网卡对象jpcap上不断得到数据包。

**文件自定义保存**：提供文件自定义保存功能

文件名：SaveFile.java

对newCount中数据进行保存

**柱状图显示**：提供柱状图可视化分析接口

文件名：newChart1.java

使用Jcommon.jar及jfreechart.jar函数调用柱状图对newCount中数据图形化显示

**流量统计**：提供已捕获数据包数量共有访问接口，及流量分析

文件名：newCount.java

使用Jcommon.jar及jfreechart.jar函数调用柱状图对newCount中数据图形化显示