**计算机网络技术第二次作业**

IP流量分析程序

姓名： 周宝航

学号：2120190442

日期： 2019.11.13

专业：计算机科学与技术

**目录**

[一、作业要求 3](#_Toc24745956)

[二、IP数据包格式 3](#_Toc24745957)

[三、开发环境 5](#_Toc24745958)

[四、程序流程图 6](#_Toc24745959)

[五、关键问题 6](#_Toc24745960)

[1．数据包捕获 6](#_Toc24745961)

[2．数据包解析 7](#_Toc24745962)

[六、测试截图 8](#_Toc24745963)

[1．主页面展示 8](#_Toc24745964)

[2．IP数据包获取 8](#_Toc24745965)

[3．IP数据包过滤 10](#_Toc24745966)

[4．IP数据包导出 11](#_Toc24745967)

# 作业要求

**实现一个流量分析程序，具体要求：**

1. Windows平台上，基于WinPcap，图形用户界面，编程语言不限；
2. 输入捕获条件（IP地址、时间段），输出IP分组主要字段（版本、协议、源地址与目的地址），实现IP流量排序（按协议或IP地址）；
3. 撰写说明文档，包括：编程环境、关键问题、程序流程、测试截图等；
4. 提交全部程序，包括：源代码、可执行程序、说明文档等。

# 二、IP数据包格式

IP协议位于网络层，是TCP/IP协议簇中的核心协议，提供数据传输的最基本服务，是实现网络互联的基本协议。IP分组的结构如表2.1所示。RFC791是最早的IP协议的文本，它对IP分组结构做出了明确的规定。IP数据报 是一个与硬件无关的虚拟包，由首部和数据两部分组成。首部的前一部分是固定长度，共 20 字节，是所有IP数据报必须具有的。在首部的固定部分的后面是一些可选字段，其长度是可变的。首都中的源地址和目的地址都是 IP 协议地址。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本（4位） | 头长（4位） | 服务类型（8位） | 封包总长度（16位） | |
| 封包标识（16位） | | | 标志（3位） | 片断偏移地址（13位） |
| 存活时间（8位） | | 协议（8位） | 校验和（16位） | |
| 来源IP地址（32位） | | | | |
| 目的IP地址（32位） | | | | |
| 选项（可选） | | | 填充（可选） | |
| 数据 | | | | |

**表2.1 IP包结构**

IP分组结构的各个字段含义如下：

**（1）版本**：占4位（bit），指IP协议的版本号。目前的主要版本为IPV4，即第4版本号，也有一些教育网和科研机构在使用IPV6。在进行通信时，通信双方的IP协议版本号必须一致，否则无法直接通信。

**（2）首部长度：**占4位（bit），指IP报文头的长度。最大的长度（即4个bit都为1时）为15个长度单位，每个长度单位为4字节（TCP/IP标准，DoubleWord），所以IP协议报文头的最大长度为60个字节，最短为上图所示的20个字节。

**（3）服务类型**：占8位（bit），用来获得更好的服务。其中的前3位表示报文的优先级，后面的几位分别表示要求更低时延、更高的吞吐量、更高的可靠性、更低的路由代价等。对应位为1即有相应要求，为0则不要求。

**（4）总长度**：16位（bit），指报文的总长度。注意这里的单位为字节，而不是4字节，所以一个IP报文最大长度为2^16 - 1=65535字节。

**（5）标识（**identification**）**：该字段标记当前分片为第几个分片，在数据报重组时很有用。

**（6）标志（**flag**）**：该字段用于标记该报文是否为分片（有一些可能不需要分片，或不希望分片），后面是否还有分片（是否是最后一个分片）。

**（7）片偏移：**指当前分片在原数据报（分片前的数据报）中相对于用户数据字段的偏移量，即在原数据报中的相对位置。

**（8）生存时间：**TTL（Time to Live）。该字段表明当前报文还能生存多久。每经过1ms或者一个网关，TTL的值自动减1，当生存时间为0时，报文将被认为目的主机不可到达而丢弃。使用过Ping命令的用户应该有印象，在windows中输入ping命令，在返回的结果中即有TTL的数值。

**（9）协议**：该字段指出在上层（网络7层结构或TCP/IP的传输层）使用的协议，可能的协议有UDP、TCP、ICMP、IGMP、IGP等。

**（10）首部校验和**：用于检验IP报文头部在传播的过程中是否出错，主要校验报文头中是否有某一个或几个bit被污染或修改了。

源IP地址：32位（bit），4个字节，每一个字节为0～255之间的整数，及我们日常见到的IP地址格式。

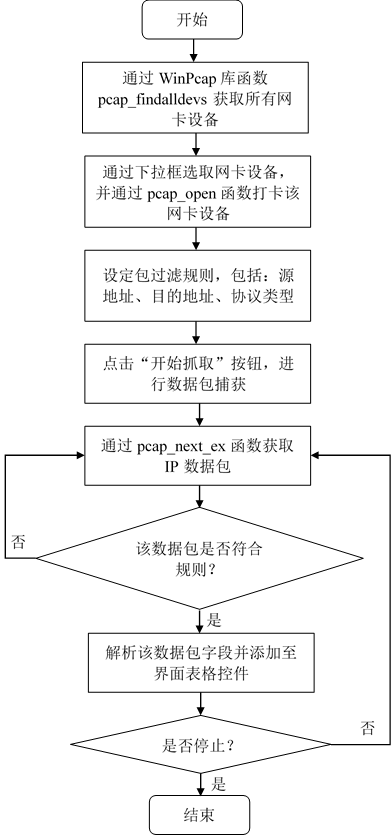
**（11）源IP地址**：32位（bit），4个字节，每一个字节为0～255之间的整数，及我们日常见到的IP地址格式

**（12）目的IP地址**：32位（bit），4个字节，每一个字节为0～255之间的整数，及我们日常见到的IP地址格式。

# 三、开发环境

|  |  |
| --- | --- |
| **硬件环境** | Intel(R) Core(TM)i7 CPU, 16.00GB DDR |
| **操作系统** | Microsoft Windows 10 |
| **开发环境** | Qt Creator 4.10.0 |
| **编程语言** | C++ |
| **界面框架** | Qt |

# 四、程序流程图



# 五、关键问题

## 1．数据包捕获

程序抓取数据包的主要流程为：

1）通过pcap\_findalldevs函数获取主机当前所有的网卡设备名称；

2）用户选择网卡后便确定了网卡名，通过pcap\_open函数打开指定网卡；

3）在获取网卡设备后通过pcap\_compile函数编译过滤表达式；

4）将上一步设置的过滤器通过pcap\_setfilter函数绑定至该网卡，以便网卡按照既定规则进行数据包的抓取；

5）通过pcap\_next\_ex函数即可进行数据包的捕获，该函数的返回值为按字节存储的数据。

## 2．数据包解析



在上一部分中，我们介绍了如何通过WinPcap库提供的函数进行数据包的捕获。为了解析数据包的各个字段，我们定义了“以太网首部”结构体以及“IP数据包”结构体。在得到pcap\_next\_ex函数的返回值后，我们将原始包数据强制类型转换为上述两种结构体即可通过’->’操作符获取相应的字段值。

# 六、测试截图

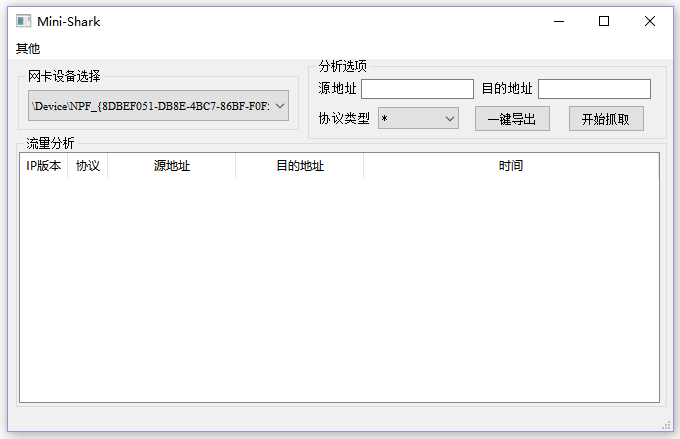
## 1．主页面展示

程序界面主要包括三部分：网卡设备选择、分析选项以及流量分析。

“网卡设备选择”主要给用户提供选择要监听的网卡。

“分析选项”给用户提供了筛选数据包的功能，可筛选的字段包括：源地址、目的地址以及传输层协议类型。“一键导出”功能可以供用户方便地将抓取的流量数据导出为csv格式文件。

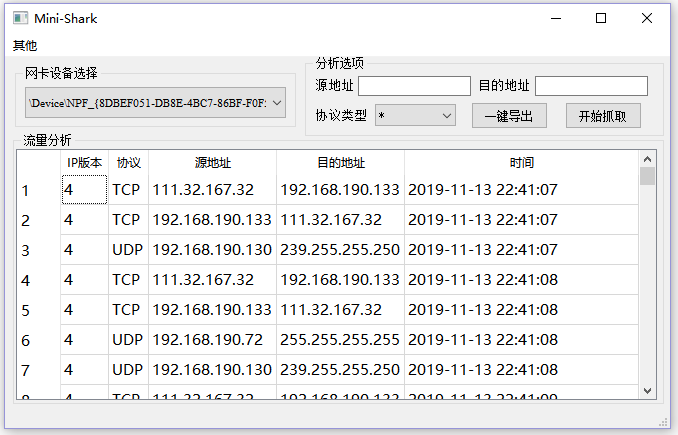
“流量分析”将捕获的数据包以列表形式展示给用户，解析字段包括：IP版本号、传输层协议、源地址、目的地址以及捕获时间。



## 2．IP数据包获取

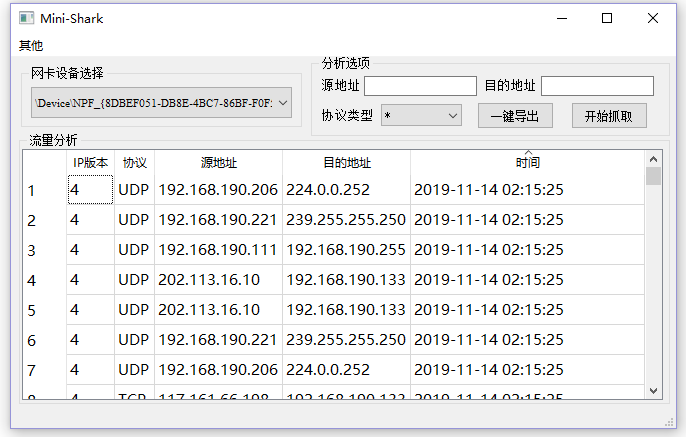
2.1 数据包列表

用户通过点击界面上的“开始抓取”按钮即可开始抓包工作。当抓取到数据包后，列表控件将解析后的数据包字段填充至新一行，并且自动滚动至列表最下方。如此一来，用户即可实时查看到抓取数据包的情况。



2.1 数据包排序

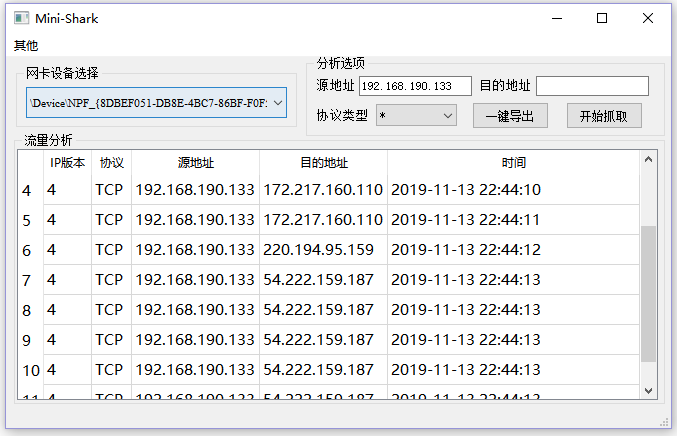
用户通过点击“流量分析”表格控件上方的“协议”或“时间”条目，即可对表格中的数据包按照相应的条目进行排序显示。当点击“协议”条目时，表格中的数据项会自动按照相同协议组合在一起。当点击“时间”条目时，表格中的数据项会按照相同时间组合在一起，并可按照时间的升序或降序排列。



## 3．IP数据包过滤

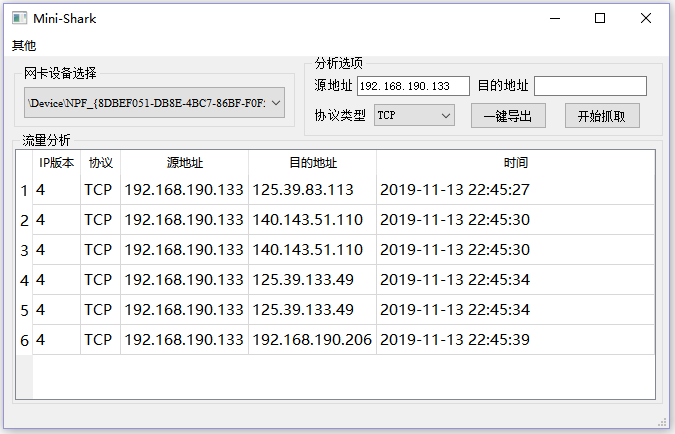
3.1 地址筛选

用户可以通过在“分析选项”中的“源地址”、“目的地址”和“协议类型”设置抓取数据包的规则。当用户设置了“源地址”规则，程序抓取数据包时会匹配其“源地址”字段，若符合该规则，那么表格控件会添加上该数据包的详细信息。“目的地址”规则同样会过滤数据包的“目的地址”字段，满足规则的数据包会呈现在表格控件。



3.2 协议筛选

“协议类型”规则提供了四种传输层协议，其会过滤数据包的“协议”字段，若满足规则就将该数据包添加至表格控件。如下图，设置了“源地址”为“192.168.190.133”，协议类型为“TCP”，那么程序会按照规则匹配数据包的相应字段是否符合规则，若满足条件则该数据包被添加至表格控件。



## 4．IP数据包导出

用户可以点击“分析选项”中的“一键导出”按钮，若此时表格控件存在捕获的数据包信息，那么会弹出“保存文件”提示框进行保存操作。导出的文件格式为.csv，包含表格控件中的所有信息，如：表头、对应列的所有数据包信息。具体保存信息情况如下图所示。

