计算机网络体系结构实验报告

选课序号 44 专业 网络工程 班级 2 学号 2220193816 学生姓名 赤凯

课程名称计算机网络体系结构实验 实验时间7.15 实验名称数据包协议分析实现实验成绩

# 实验内容：

使用pcap库制作一个抓包工具

# 实验目的：

1. 加强对网络协议的理解
2. 锻炼编程能力
3. 培养查阅文献的能力

# 实验要求：

（1）掌据原始套接字（raw socket）或Winpcap的使用方法；掌握数据包捕获和数据包分析的相关知识；

（2）实现数据包的捕获、分析，并保存捕获的数据包；

（3）根据捕获的信息进行统计分析，实时流量监测等；

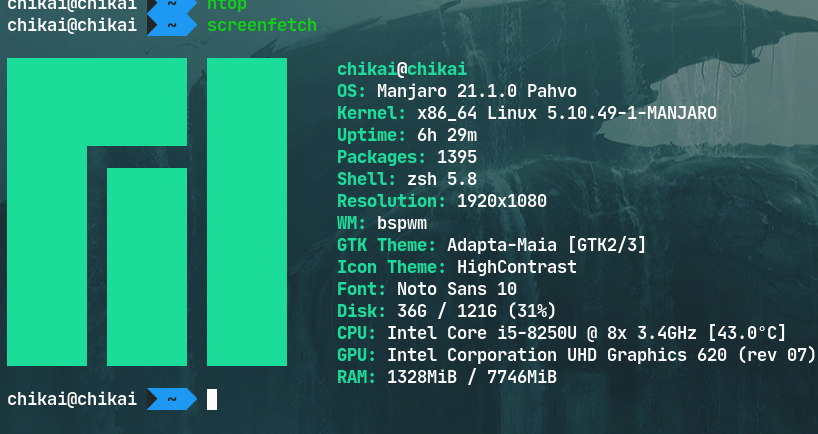
（4）改写数据包内容并发送；

（5）交互界面良好；

（6）提高综合运用文献的能力。

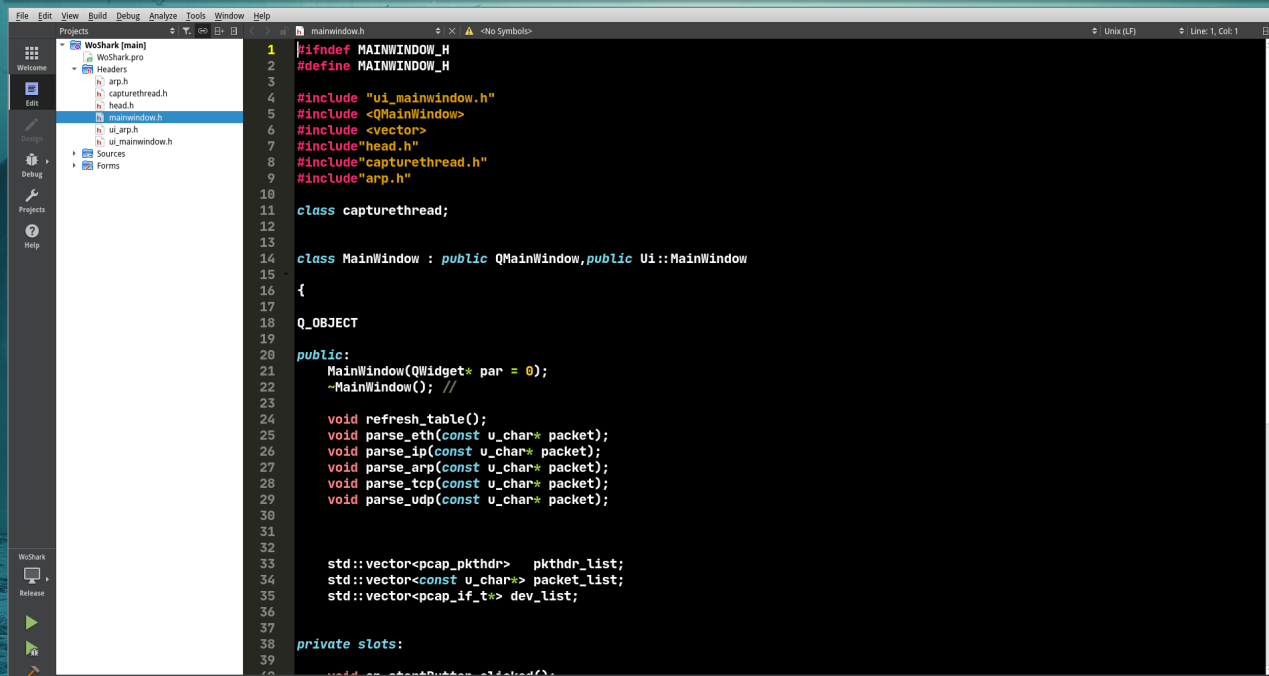
# 四、系统环境、开发工具简介（写清楚开发环境配置步骤，若有需要，可截图说明）

系统环境： Manjaro Linux 5.10



开发工具：

QCreator : Qt 框架的官方IDE ，对Qt应用的编写支持很好。同时也支持C/C++ 语言的编写



Qt Designer : 用来设计 Qt 的 UI 接口



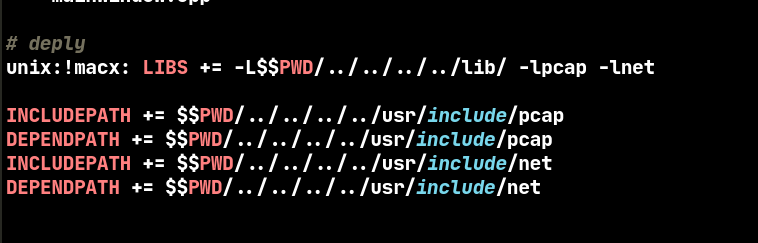
## 环境配置：

在 Linux 下，直接安装 libpcap 库，即可使用。但是在编译时要进行相应的链接

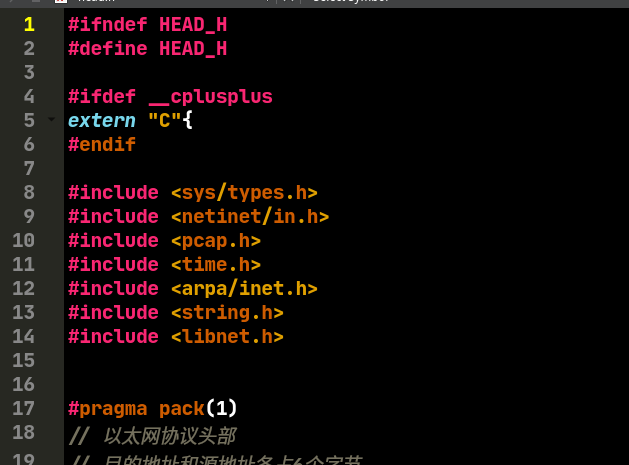
使用gcc 编译的命令如下：



在工程中，要在 makefike 文件中 指明 依赖关系：



另外，还要在相关 C 的库前面加上C++ 的宏定义



这样就可以进行开发了

# 五、实验原理

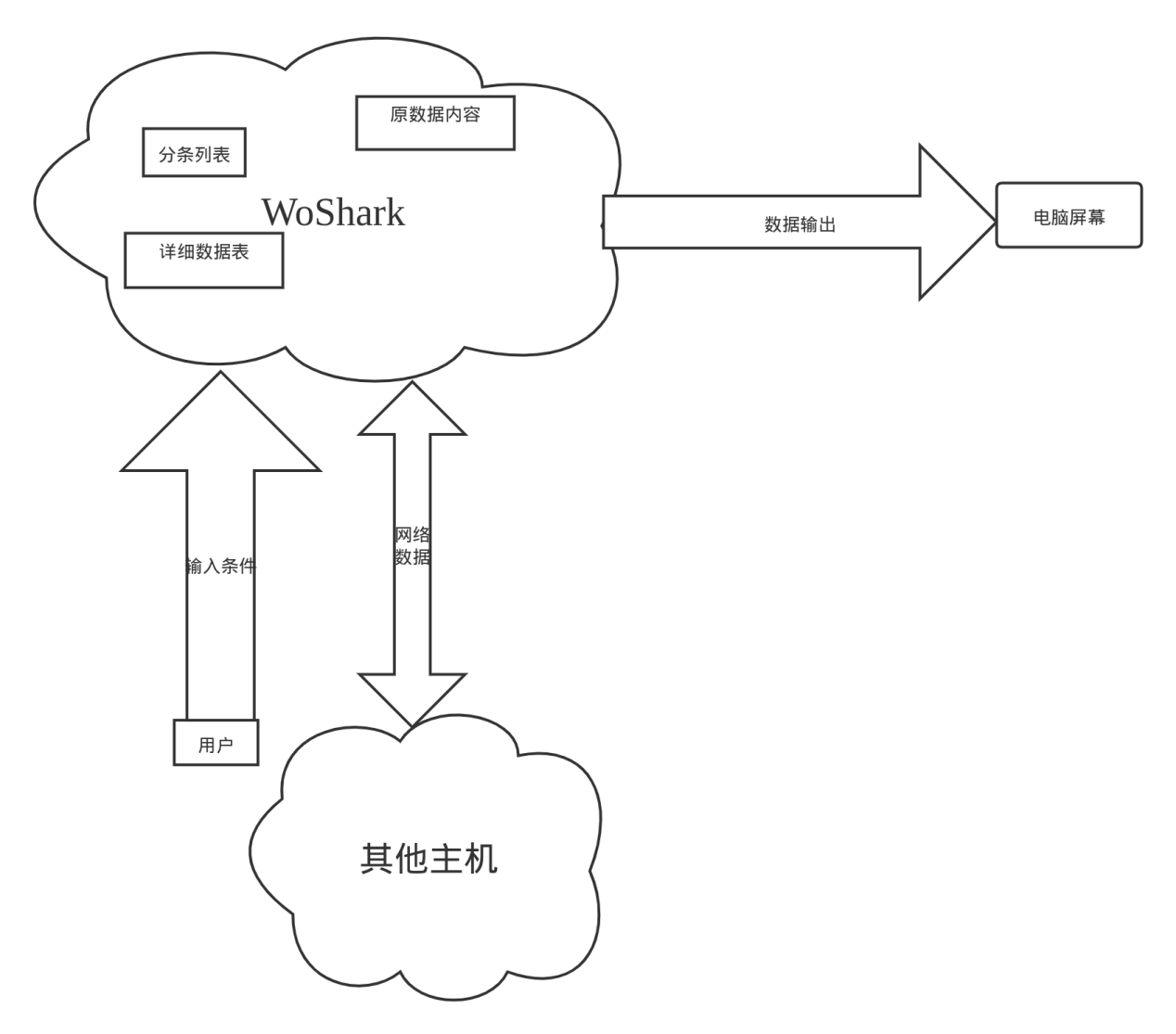
程序多线程进行，一个线程进行数据报的抓取，一个线程进行数据内容的分析和在屏幕的输出。

其中libpcap 库用来进行数据报的抓取，基于Qt的图形化界面用来输出数据。libnet库 进行数据报的发送。

# 六、程序设计

## （1）程序设计思想（数据流图、算法、程序流程图、主要的数据结构等）

数据流图：



### 算法：

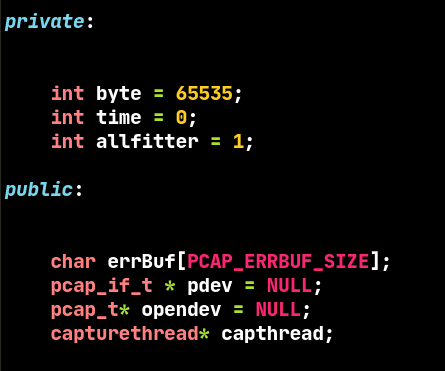
#### 1.主函数的实现

主函数mainwindow 进行UI 界面的绘制和公共数据的设置。

使用可变数组来 存放 数据信息块，捕捉的数据报，以及设备



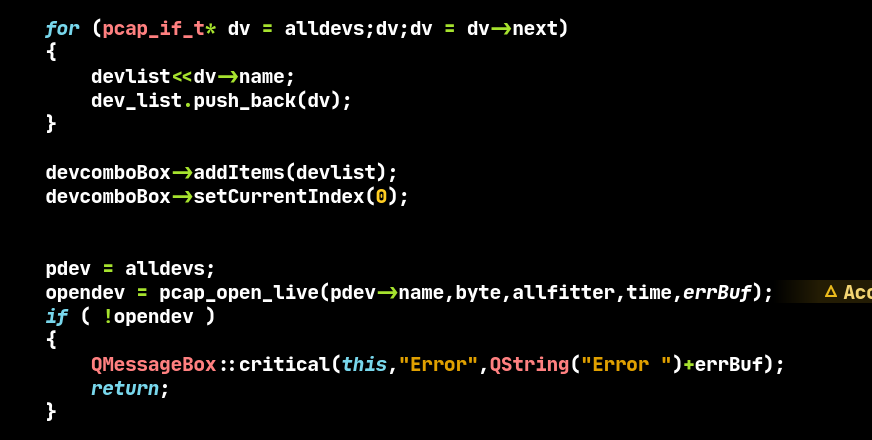
存放一些和 libpacp库抓包有关的数据



初始化界面：



通过API 读取设备，将它们存在一个全局数组里，用于将来切换抓包设备



### 2. 数据报的捕捉

基于libpcap 库的 API 进行相应的操作

这部分主要在 mainwindows 类的 on\_startButton\_clicked()

函数 和 线程类 captureThread 实现。

Qt 的 按键操作 是 用信号函数来进行的，当点击start 按钮时，

执行on\_startButton\_clicked() 函数。



函数首先从 条件栏里 读取过滤条件，通过相关的API 函数进行编译。

这时 条件输入栏变为不可输入，网卡选择栏变为不可改变，start 按钮变为不可点击同时变为”runing’，stop按钮变为 可以点击 .

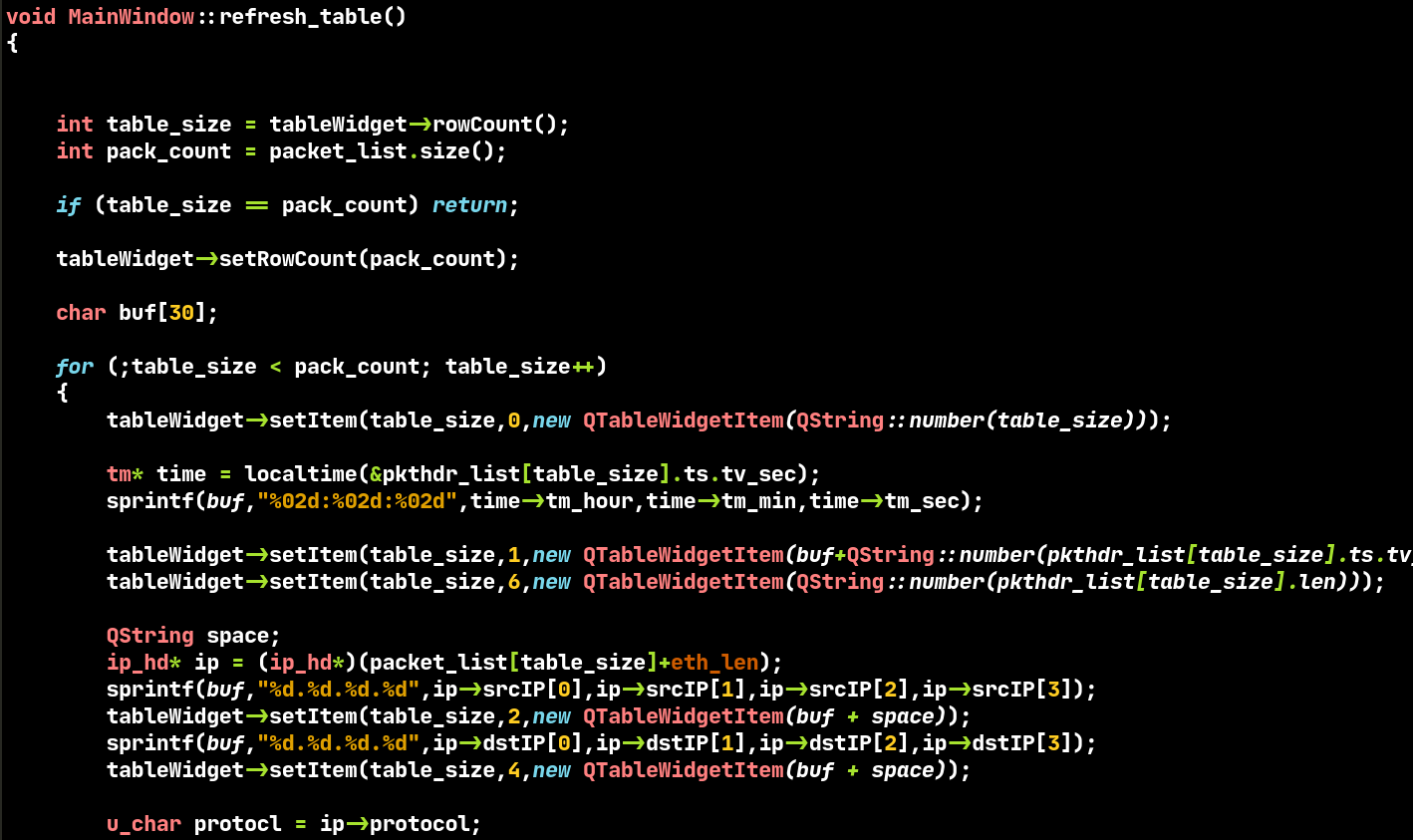
进程标识isrun变为1,抓包进程启动。



captureThread 类继承自 QThread,重写了run()函数。**这里设置一个运行判断条件 isrun,用来判断进程是否运行**。

当isrun 标识符 为1 时，从mainwindow类中读取设备，循环执行pcap\_next\_ex函数，抓取数据报并将它和它的头部信息循环保存在数组里，然后刷新视图。这里设置了错误信息，用来提示。

### 3. 数据报的解析和视图的刷新。





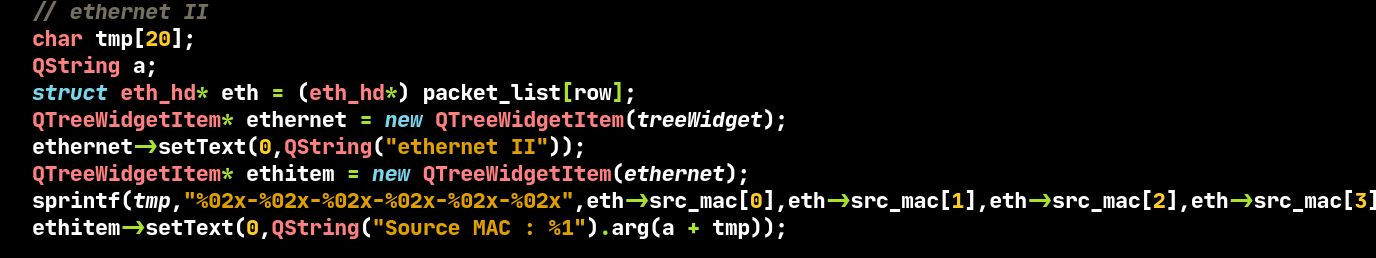


当执行刷新操作的同时进行数据报的解析。

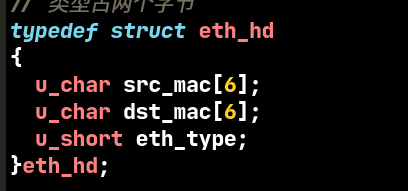
如果数据队列比视图列表大，循环读取数据报分析，从 pkthdr-list 读取一个 pacp\_pkthdr结构体，这是 API 中封装好的，从中读取数据包的长度，捕获长度，以及时间，将他们插入到表格中。

从数据报队列中读取相应的数据报，类型为u\_char\* ,此时指针在数据报的首部，即以太网首部位置，此时将它强制类型转化为 eth\_hd 结构体指针，这个结构体是事先定义好的，用来划分首部，之后从结构体中直接找出以太网首部信息插入列表。

如图：



首部定义如下：

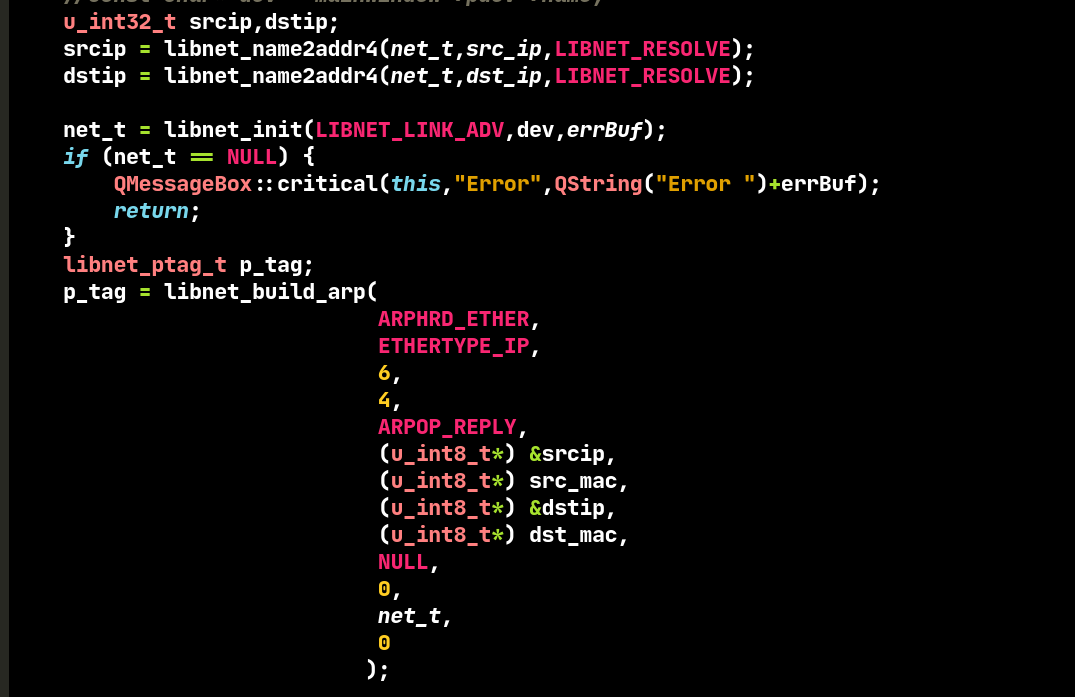


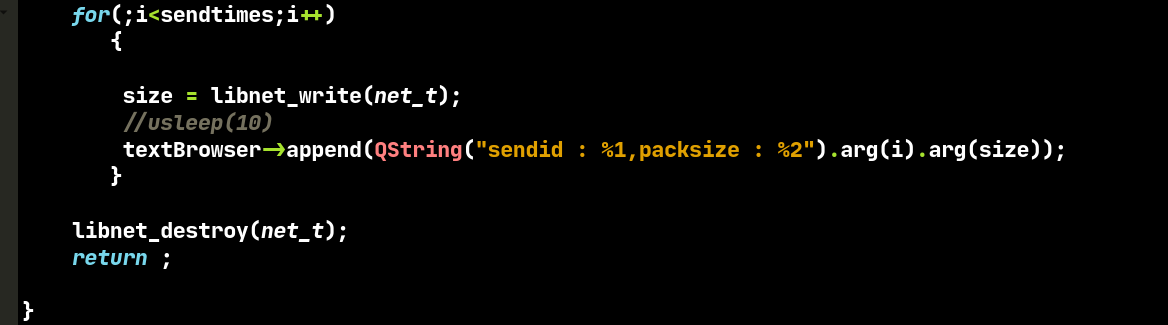
类似的，处理各部分首部信息，将其插入视图。

## 发送arp报文

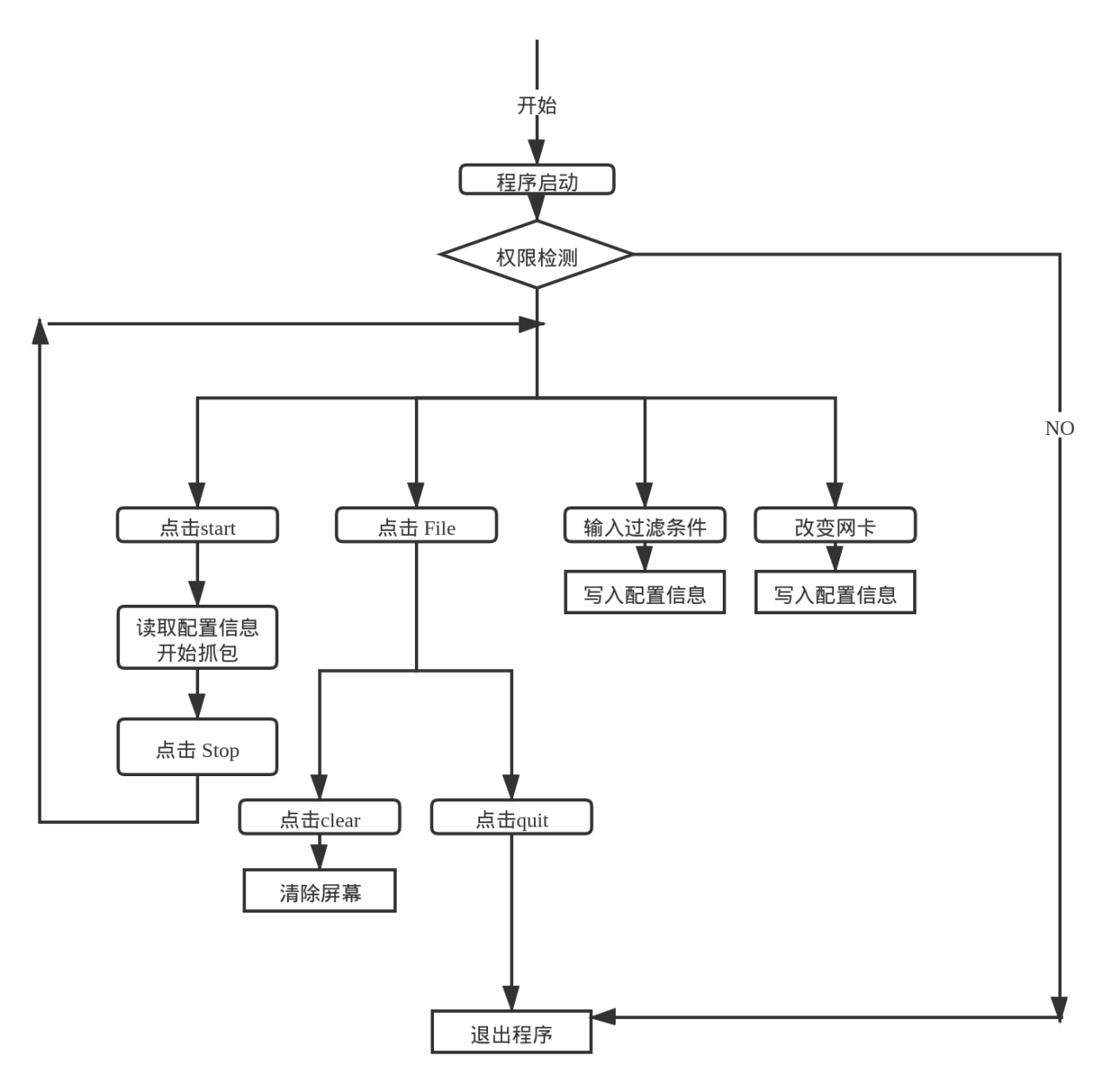
使用 libnet 库进行数据报的封装，可以指定源IP和MAC ，目的IP和MAC

发送一个空的arp报文，对同一个局域网主机可以发送大量arp报文，使其arp缓存混乱，造成断网。





### 程序流程图：



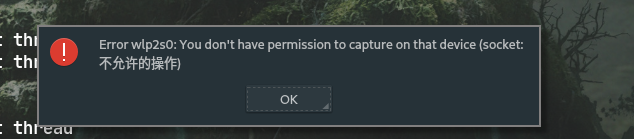
### 主要的数据结构：

可变数组： vector 用来储存抓包数据

### 容错处理及可移植性

每个输入加入了错误检测

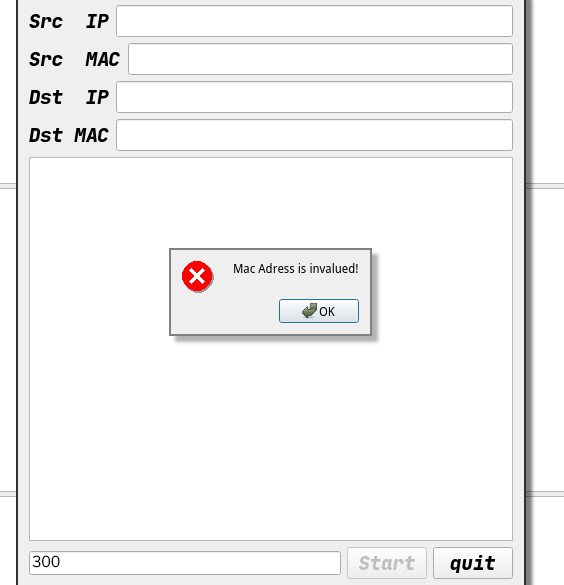
在没有网卡权限时，不能打开程序



如在运行时不能改变 过滤条件 和 网卡设备，也不能重复点击start



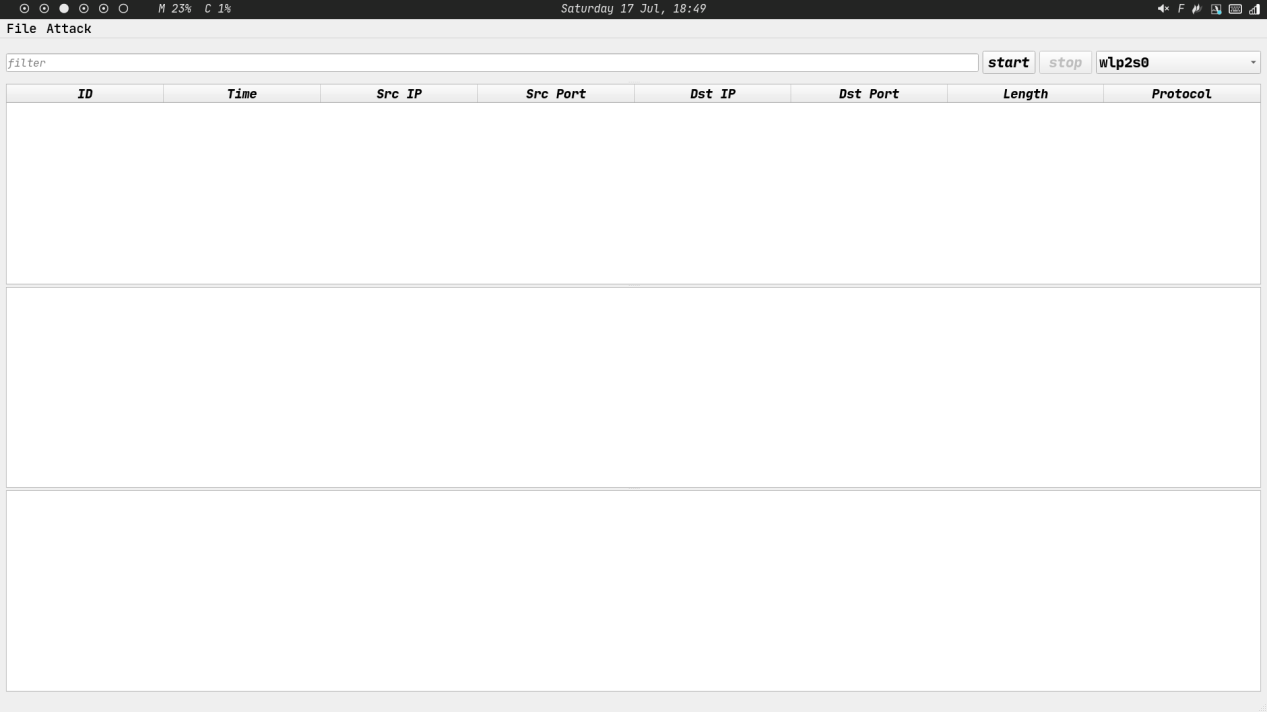
在输入MAC 和 IP 的时候，不符合格式会报错，不能运行



使用 Qt 框架做图形化界面，可以较好的在不同平台进行移植

## 主要的程序模块

主界面：



条件输入栏：



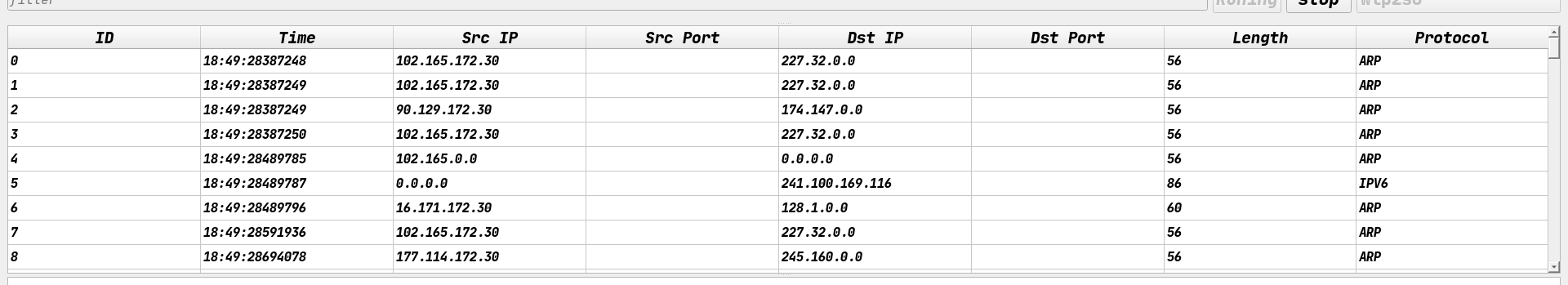
开始，停止按钮：



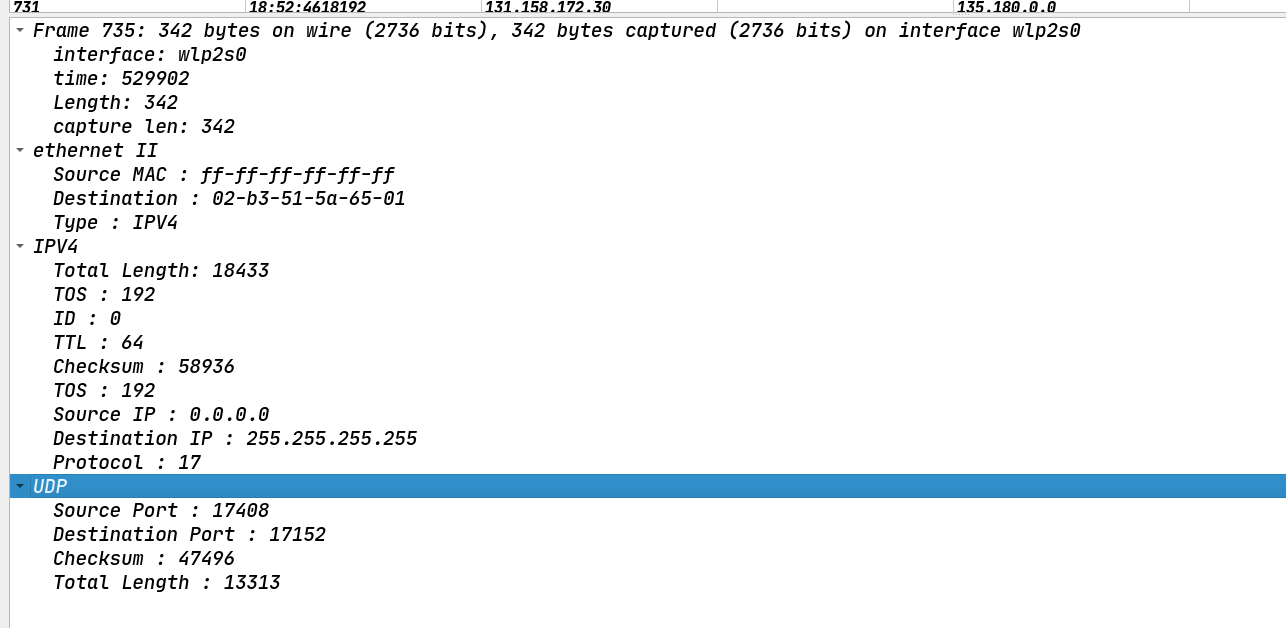
网卡选择栏：



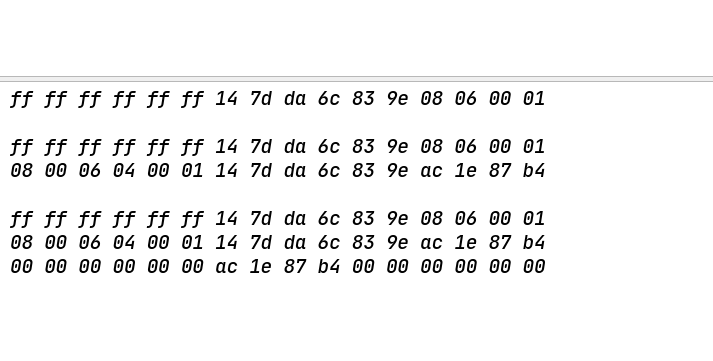
报文列表框：



报文分析框：

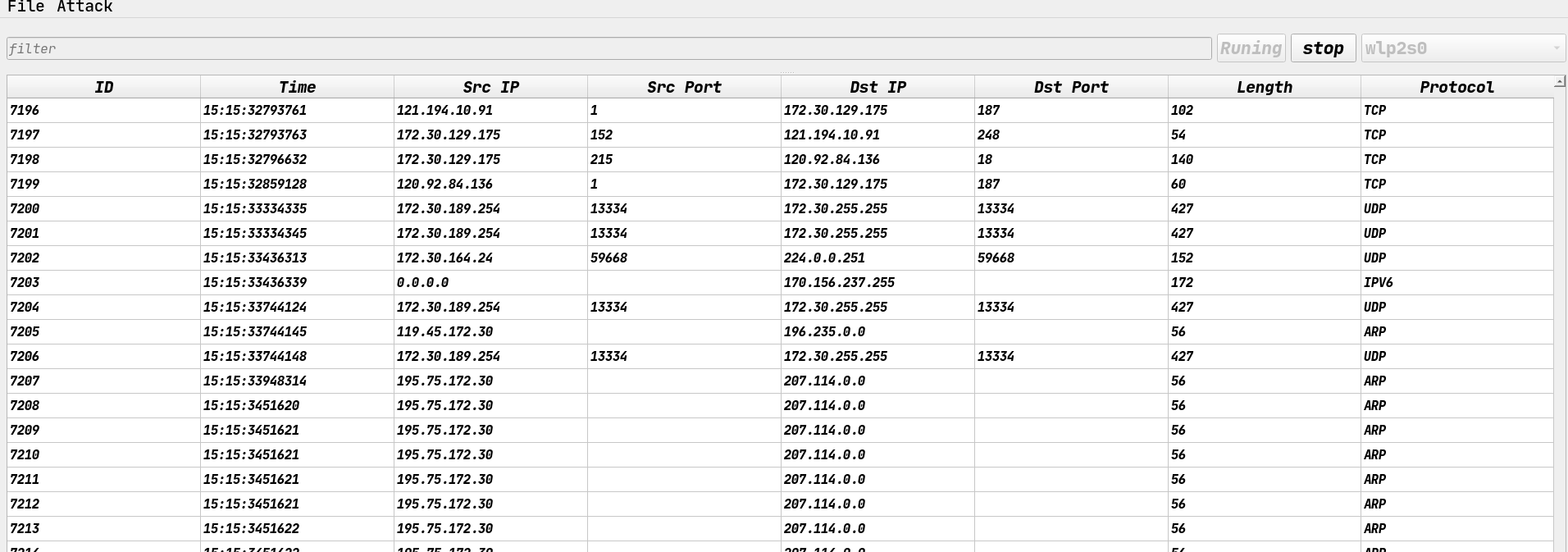


原始数据框：

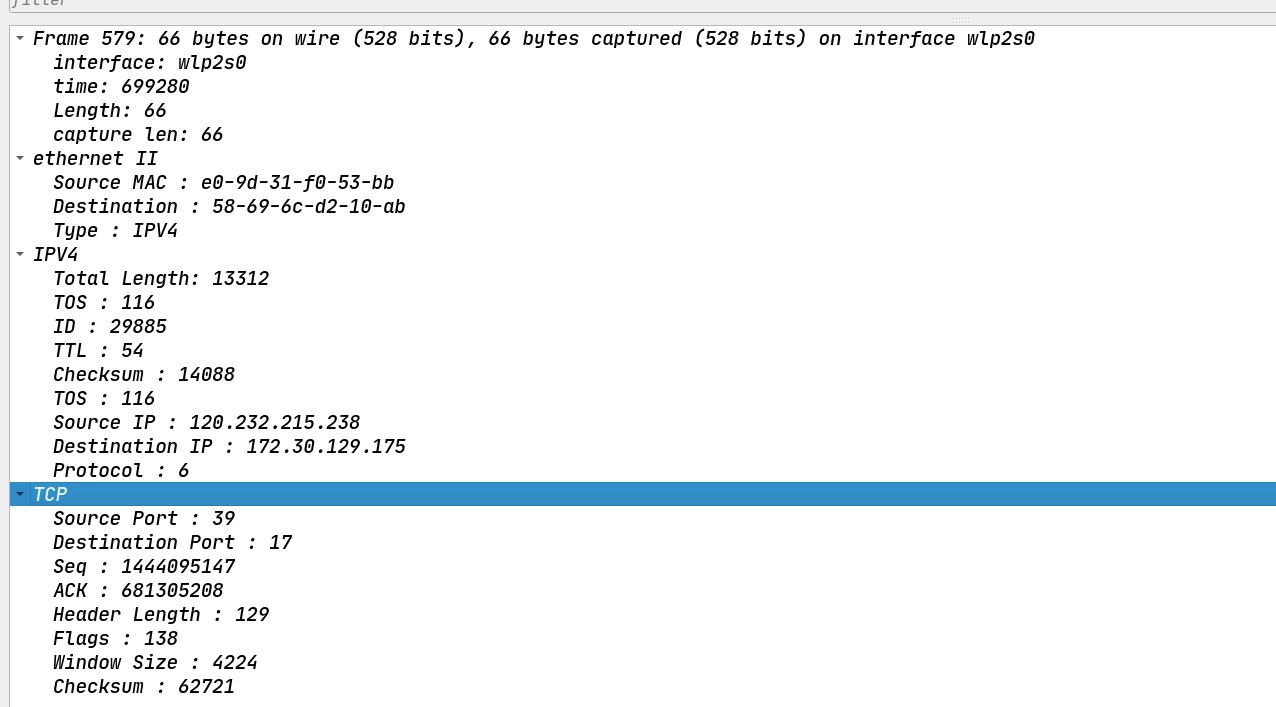


## 程序实现功能

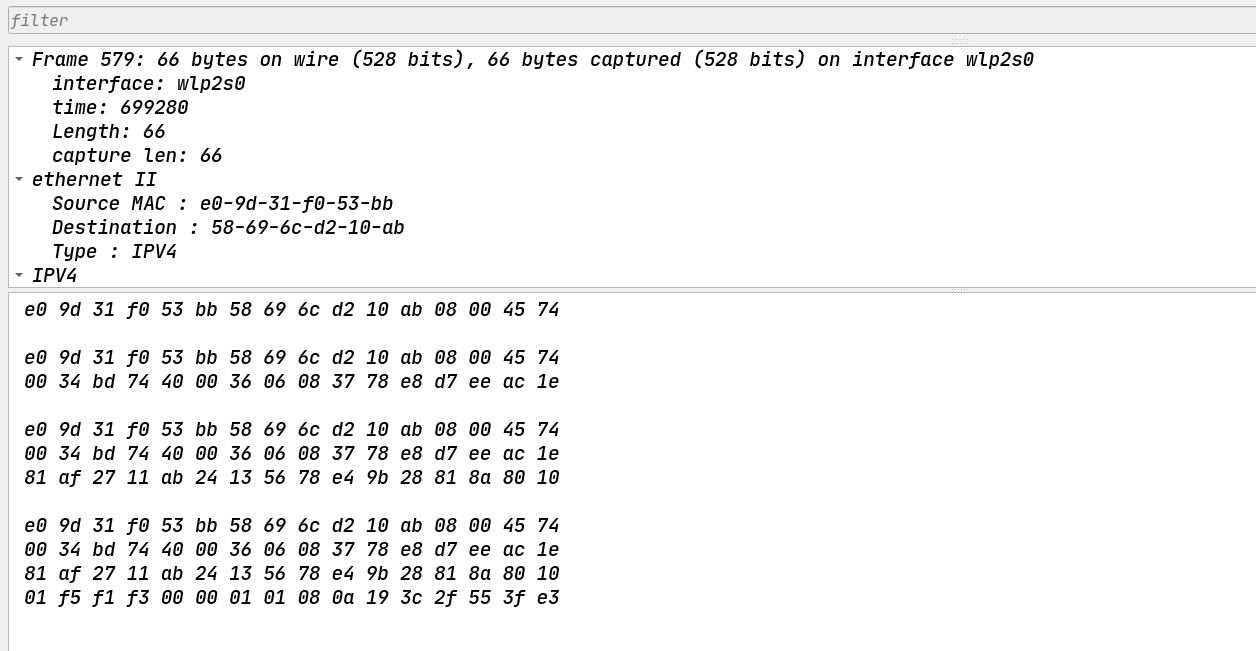
1. 抓取数据报，并列表显示



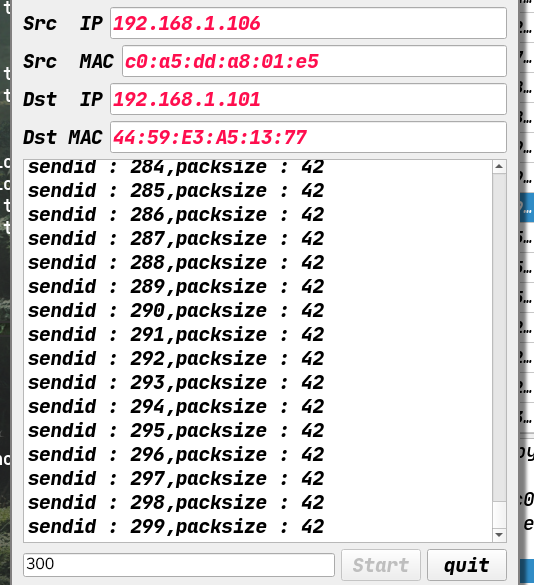
1. 分析数据报

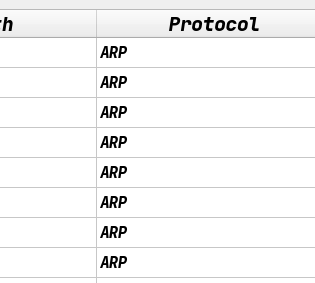


#### 显示数据报内容



### 发送arp报文

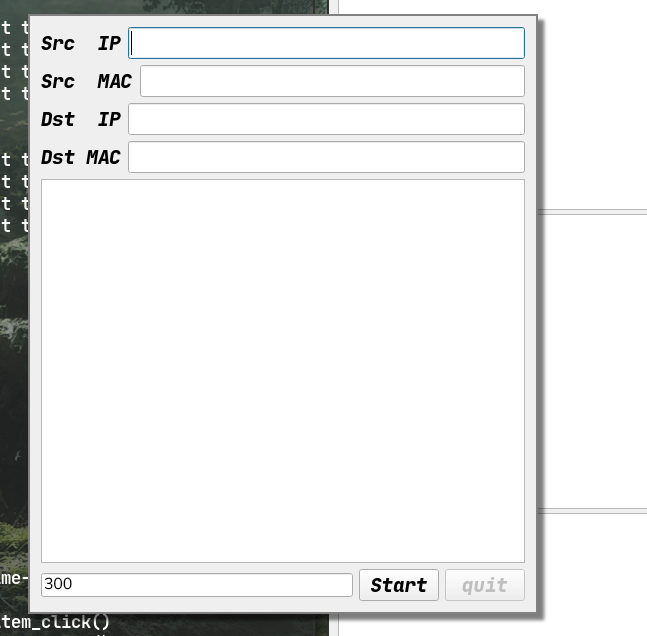




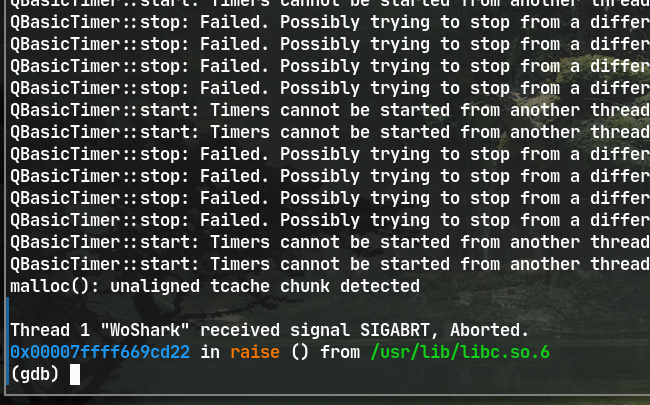
### （4）程序调试过程

#### 1. 使用gdb 调试 内存泄漏 闪退问题

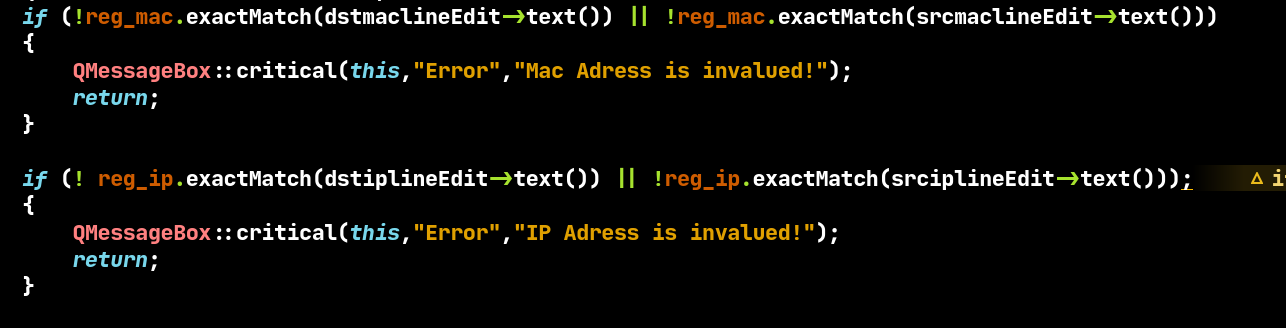
当进行arp泛洪攻击时，出现莫名卡退



进行 gdb调试

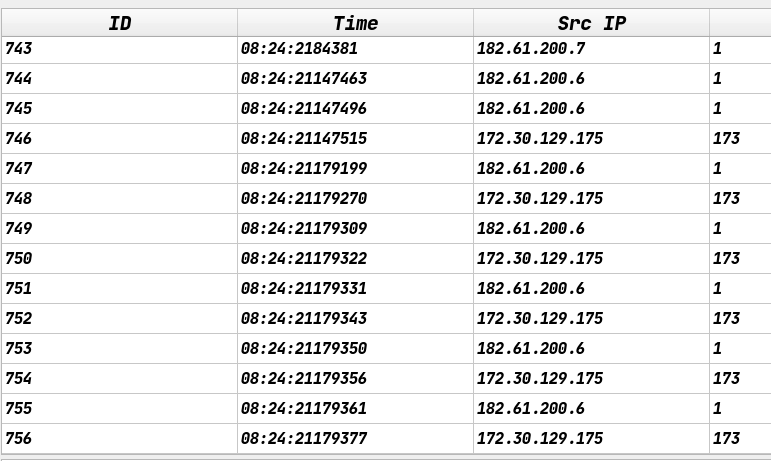


此时看出 malloc 函数出了问题，在根据运行时进行的操作，推断出是输入的数据超过数组内存造成内存溢出，于是对相应的数组进行修改，增加正则表达式和差错处理进行避免



#### 2.线程问题（1）

采用多线程操作，一个线程控制主页面，一个线程控制抓包程序，在停止抓包线程时，无法真正停止，再次点击 Start 时发现仍在后台运行。

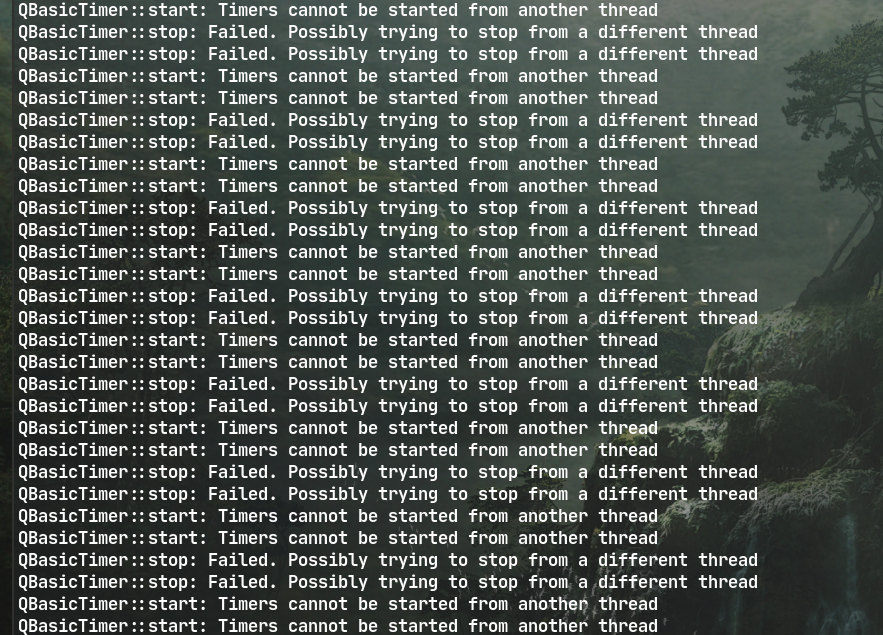


经过网上查阅，原来使用 quit（）关闭线程，实际没有完全关闭，要等待线程工作结束才停止。另一方面，可能是因为视图刷新和抓包速率不一致，造成在抓包停止时，视图刷新还没有完全结束，存在一定延时。

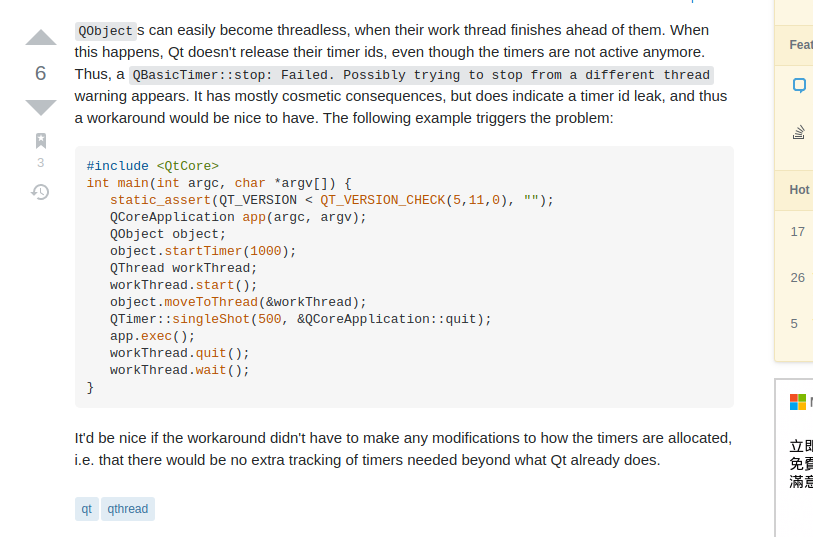
解决方法是，将quit()改为wait(),当结束线程时，将其阻塞，这样可以使其完全关闭，但是会造成页面卡顿一段时间。

3 线程问题（2）

线程运行时，终端会出现



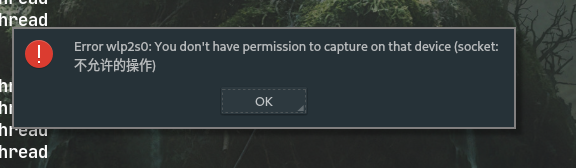
以为是线程的问题，经查阅资料，Stackoverflow 上得到回答说这是 Qt 线程的问题，不影响运行



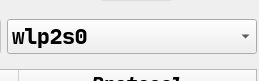
# 程序使用说明、运行结果

1. 使用管理员权限打开程序 （因为要使用网卡的设备）

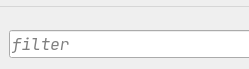
否则会无法打开：



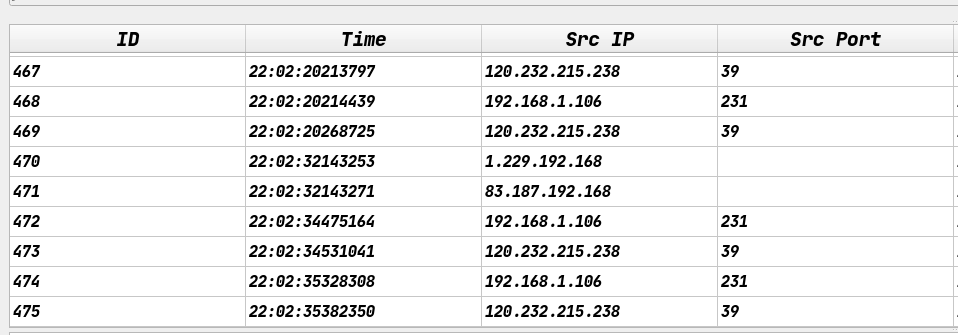
可以选择网卡，默认为当前网卡



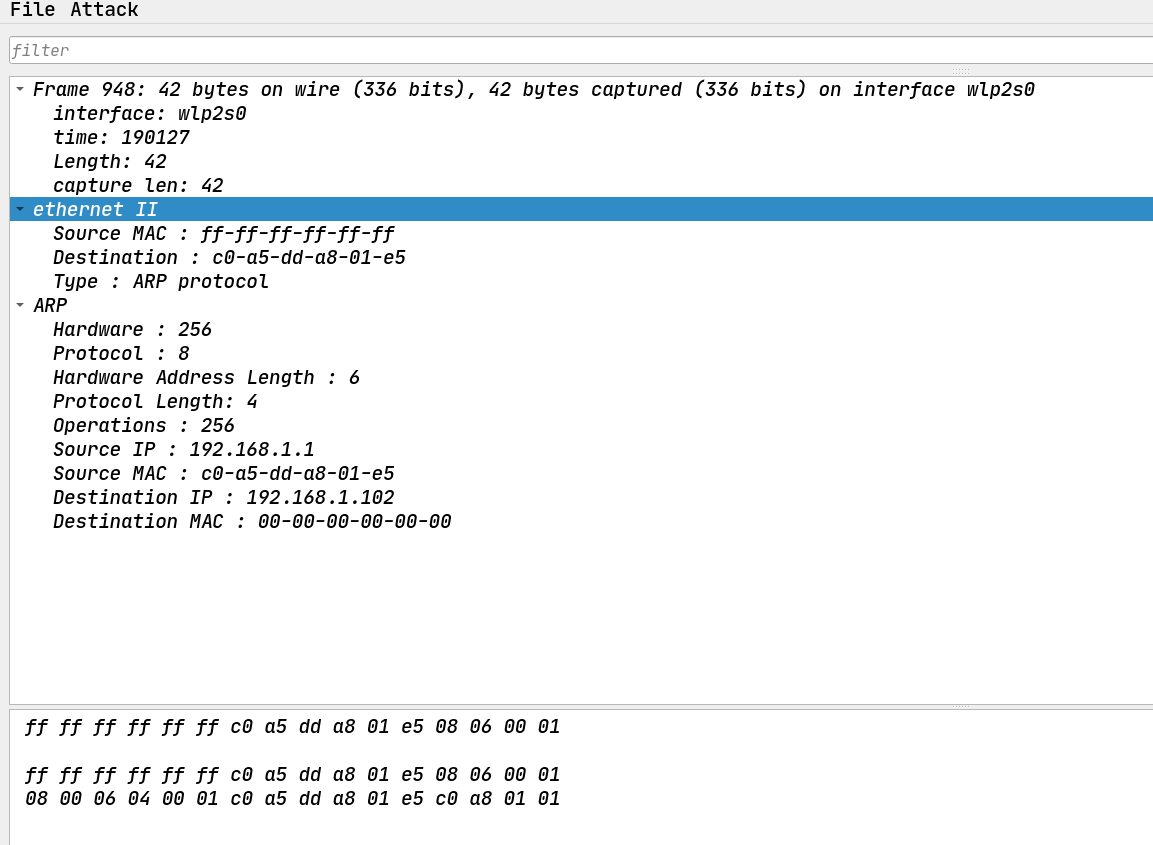
可以输入过滤条件



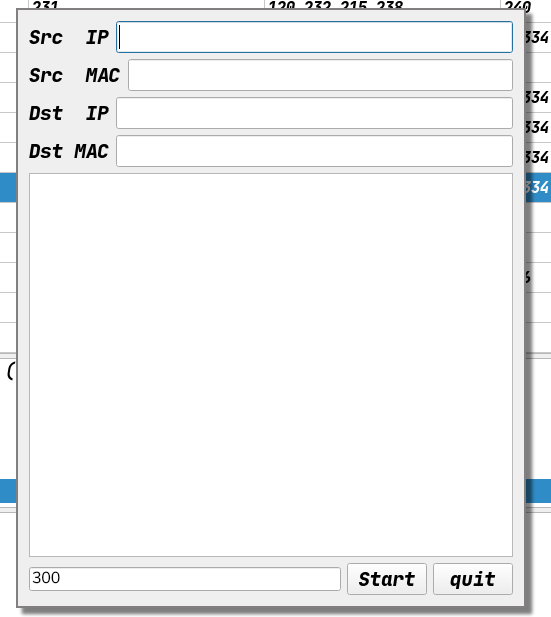
##### 选择好配置条件后，点击start 开始抓包，抓到的报文在列表里显示基本信息

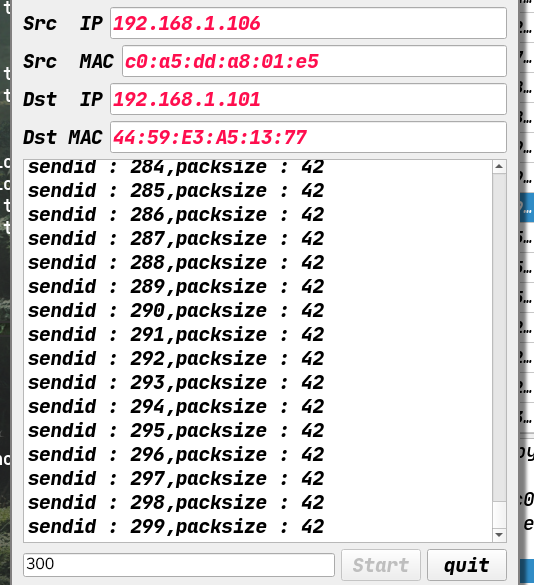


1. 点击相关单个项，可以查看详细的解析和原文件



1. 发ARP报文





# 实验总结

（1）学习资料：

winpacp 文档

Qt 文档

Qt 学习之路2

计算机网络相关知识

## 实验收获

1. 通过此次实验，我巩固了计算机网络所学的知识，特别是网络报文的结构，对网络报文原码的分析。
2. 通过此次实验我学会了使用libpacp(winpacp)库进行抓包工具的开发，学会了使用Qt进行简单图形化界面的开发
3. 在遇到Bug,特别是内存方面的Bug时，使用gdb进行调试，学会了gdb的简单使用和一些调试方法
4. 这次实验查阅了大量的文档资料，锻炼了阅读文档的能力，以及检索资料的能力。遇到一些问题在stack overflow 进行查阅能得到更好的效果

# 参考文献

<https://www.devbean.net/2012/08/qt-study-road-2-catelog/>

<https://www.tcpdump.org/manpages/pcap.3pcap.html>

<https://blog.csdn.net/ptmozhu/article/details/78743126>

《计算机网络 第六版》 谢希仁