嗅探器文档

目录

[嗅探器文档 1](#_Toc149900684)

[1.环境配置 1](#_Toc149900685)

[2.总体流程 2](#_Toc149900686)

[3.详细介绍 4](#_Toc149900687)

[3.1协议定义 4](#_Toc149900688)

[3.2主程序介绍(mainwindow.h/mainwindow.cpp) 5](#_Toc149900689)

[3.3抓包线程（cap\_poket） 7](#_Toc149900690)

[3.4包分析（poket\_analyze） 9](#_Toc149900691)

[3.5双击table某项显示 9](#_Toc149900692)

[3.6结束 11](#_Toc149900693)

[4.后续版本开发 11](#_Toc149900694)

1.环境配置

首先，编程之前需要下载winpcap(windows packet capture)，以及winpcap的开发工具。如图1.1和图1.2。然后在 QT工程文件中引入winpcap库，如图1.3。



图1.1

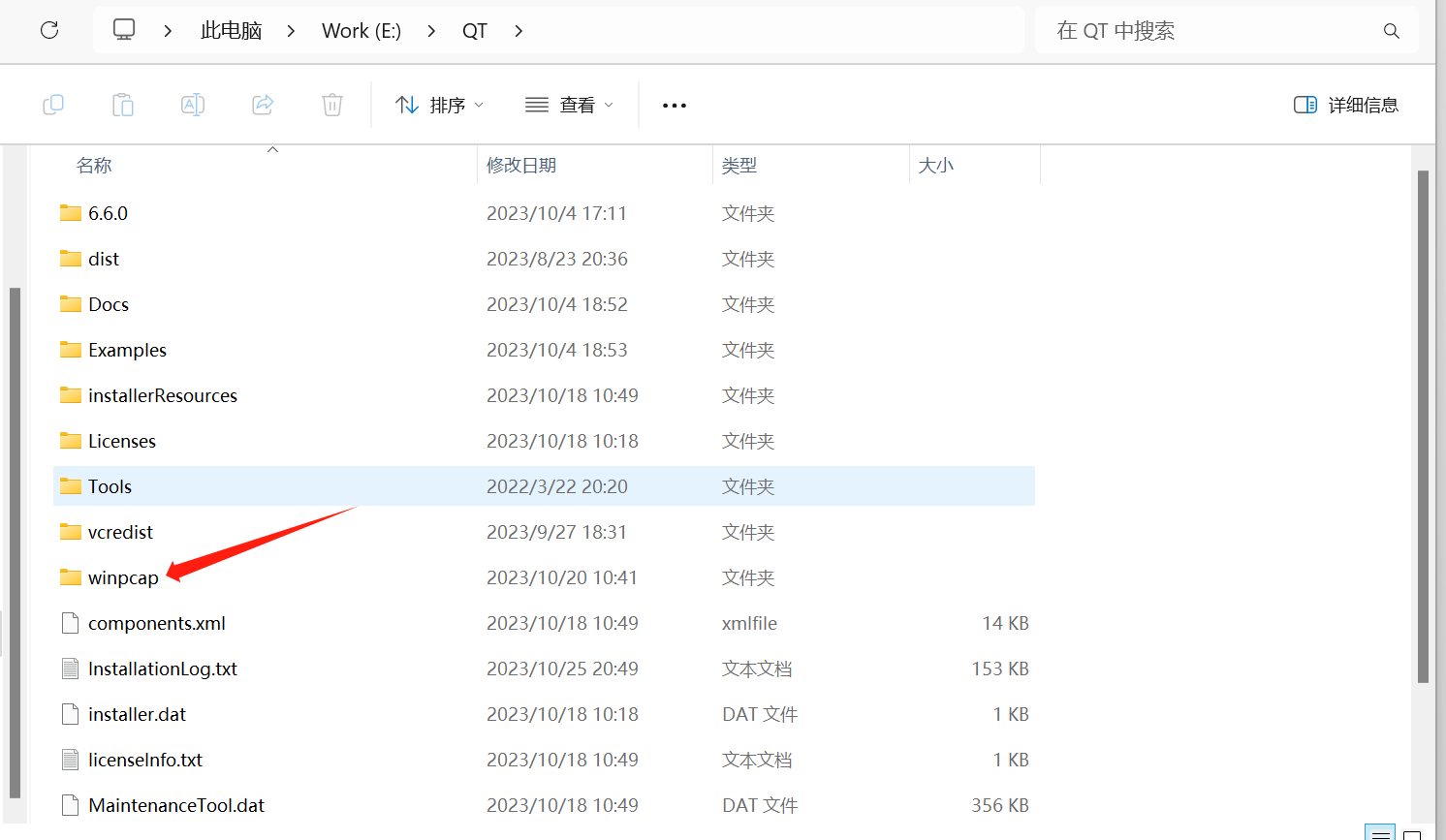


图1.2

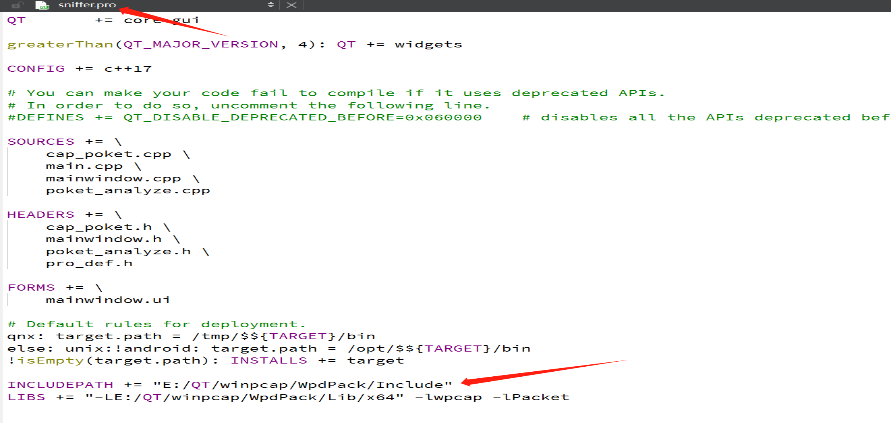


图1.3

2.总体流程

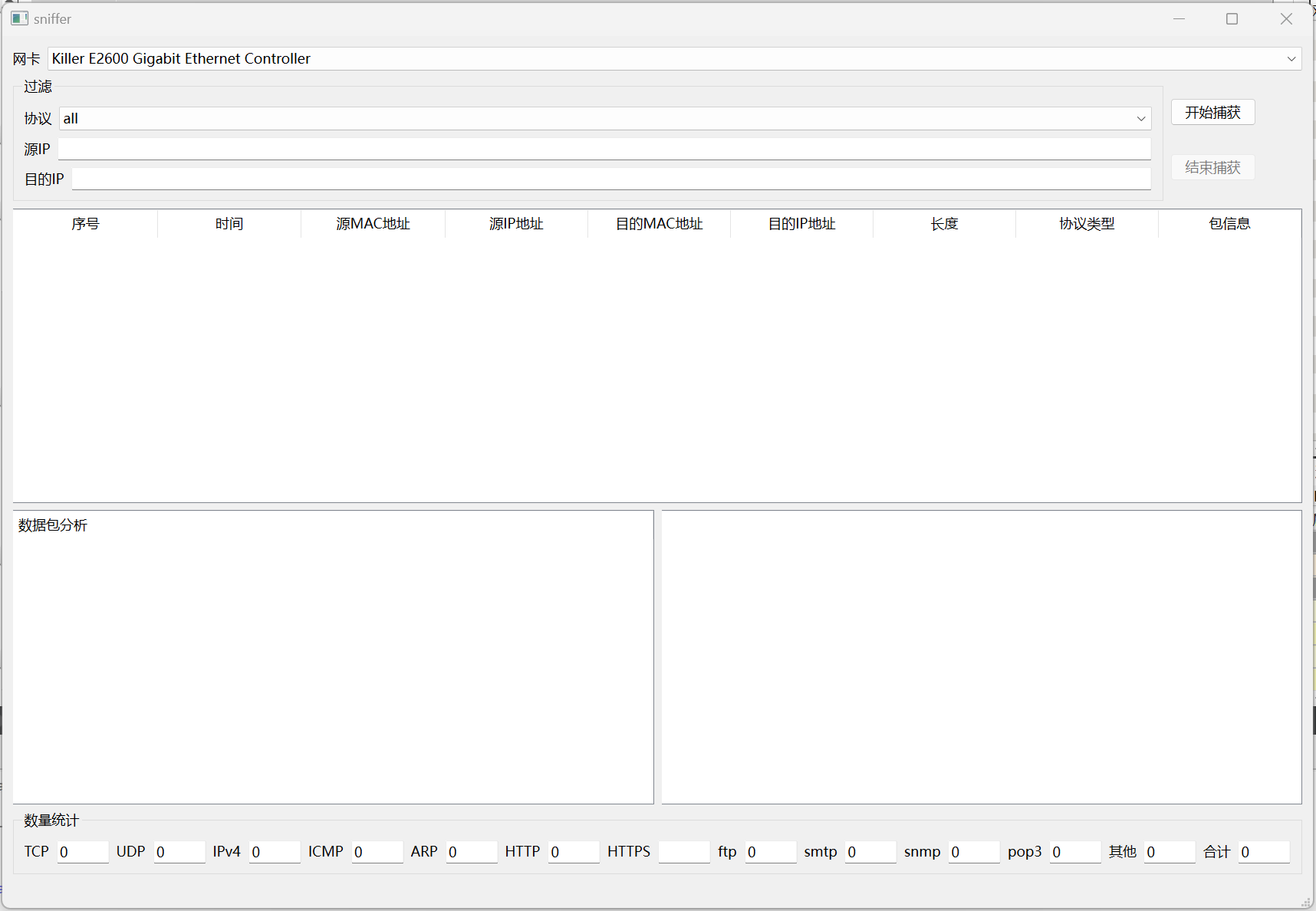


图2.1

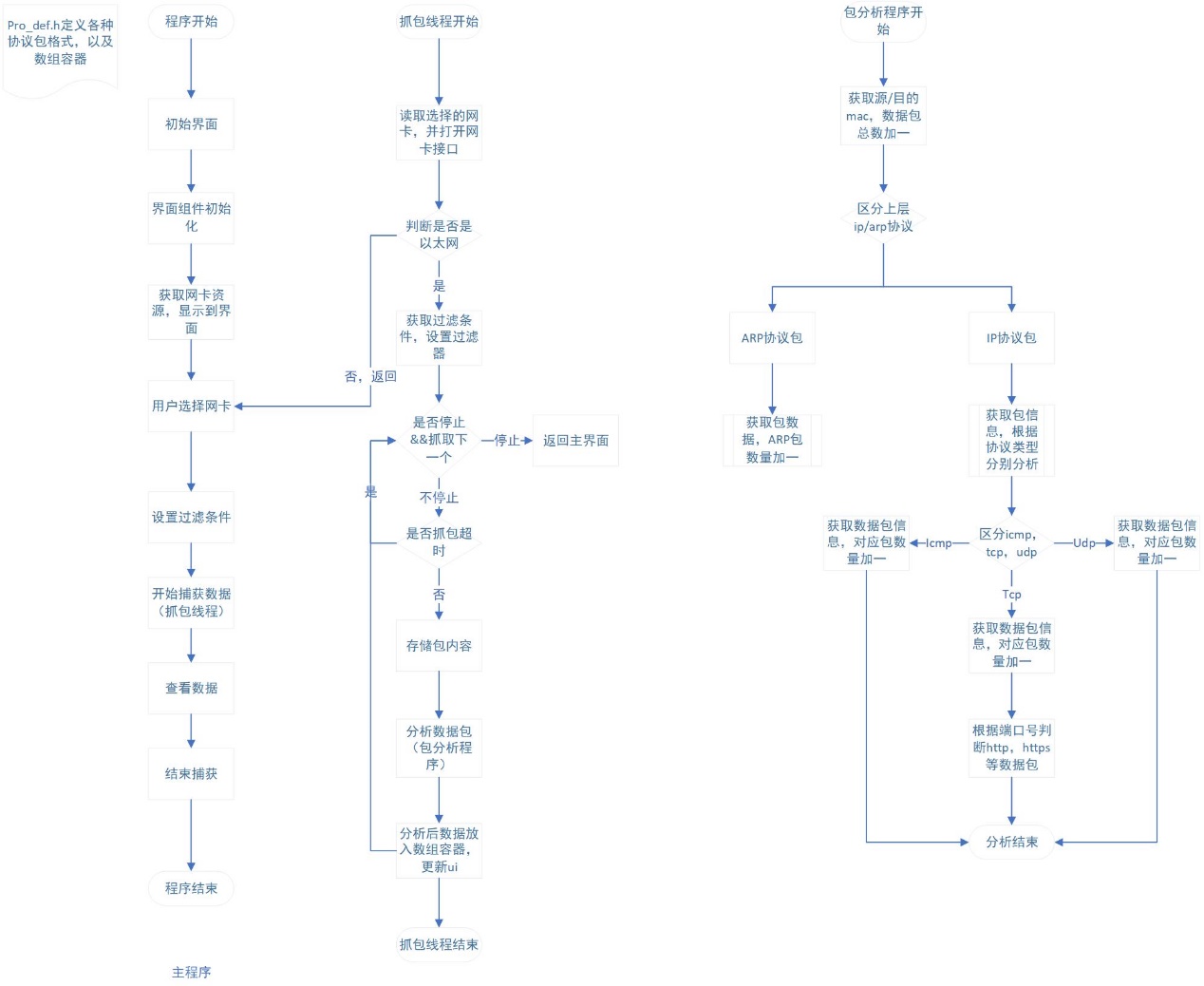


图2.2

如图2.1，主界面包含网卡选择，设置抓包过滤条件，开始和结束珠宝按钮，抓包展示表，单个包分析树，包具体内容表以及数据包统计。程序主要流程如图2.2所示。

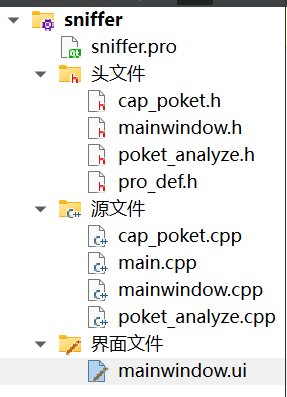


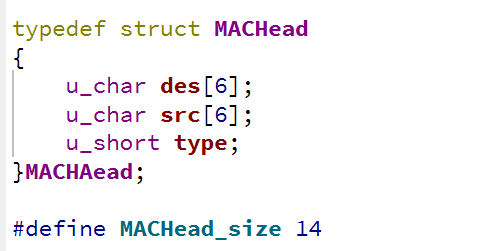
图2.3

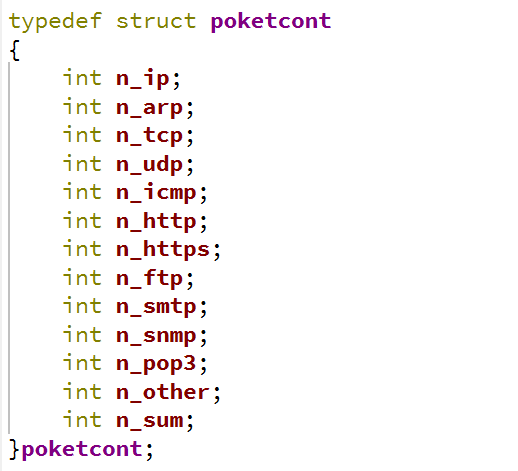
如图2.3为该项目的文件结构，分为主程序(mainwindow),包抓取线程（cap\_poket）和包分析程序（poket\_analyze）。还包括协议定义文件（pro\_def）。

注：文中poket，cont等英文为拼写错误，正确的是包（packet）数量（count），因为使用广泛难以更改了。

3.详细介绍

## 3.1协议定义







首先定义mac，arp，ip，tcp，udp，icmp协议的包头字段。然后定义了一个结构体poketcont(单词打错了)来记录各种协议包的数量，并且定义一个poketdata结构体来记录每次抓到的包的内容。最后，定义了两个vector容器来存储每次直接抓取到的数据包字符串和包分析后的内容（datavec存储每次抓到的包的字符串，poketdatavec存储每次抓到的包分析后的内容）。

## 3.2主程序介绍(mainwindow.h/mainwindow.cpp)



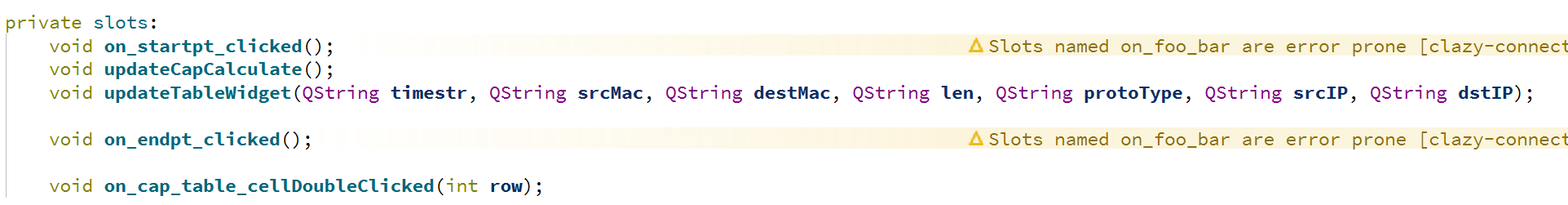
图3.1

主程序定义了如图3.1所示的全局变量，作用如图注释所示。



主函数的程序如图3.2所示，clear\_tab\_tree()函数用于在每次使用组件之前把组件清空。Getfilter函数用于生成过滤条件语句，并根据所选择的不同过滤协议调用profilter来生成协议相关的字符串。

其余函数作用如图3.2备注所示。



主程序定义了如上槽函数，on\_startpt\_clicked()为开始按钮点击槽函数。updateCapCalculate()为更新包统计数据槽函数。updateTableWidget为每次抓包后更新table组件中的信息的槽函数。on\_endpt\_clicked（）为结束按钮点击槽函数。on\_cap\_table\_cellDoubleClicked(int row)为双击table中某一项后的在tree和detail\_table中现实所点击项的具体信息。

主程序逻辑为：



首先关闭结束按钮的可点击属性。然后对过滤条件进行初始化，对tablewidget和treewidget组件进行初始化，然后调用setNetdevs（）获取网卡资源，并把获取的资源初始化到网卡选择combobox组件中供用户选择。最后还需要将数据包统计的lineedit设置为不可编辑状态。

当用户点击界面中的开始抓包后，将开启抓包线程。

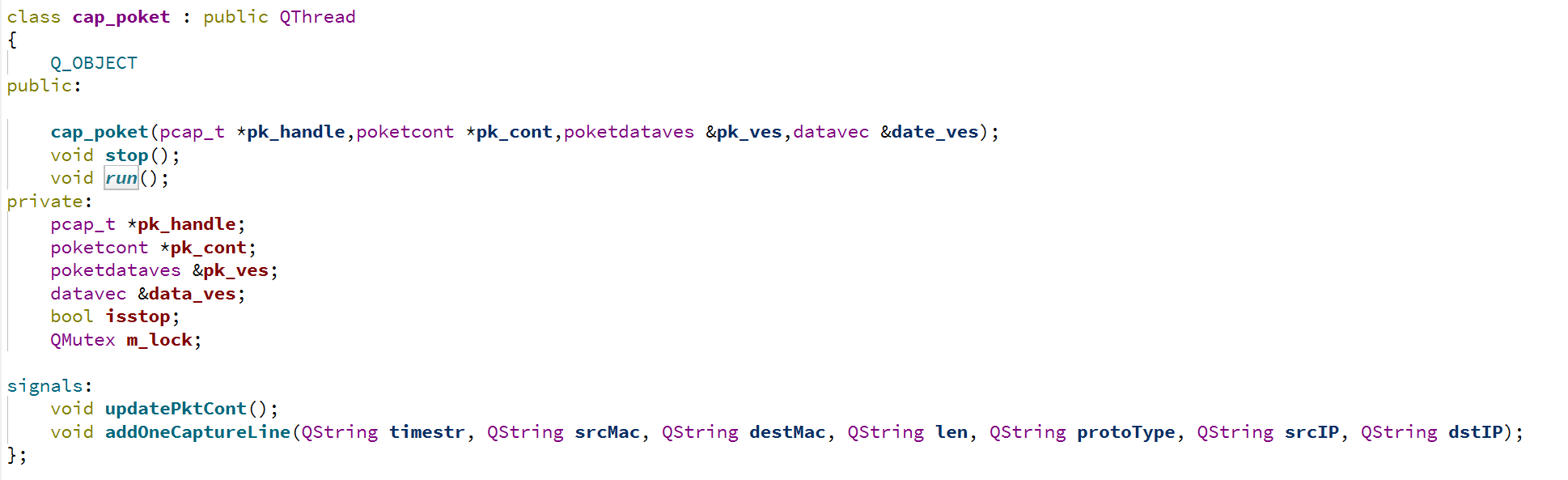
## 3.3抓包线程（cap\_poket）



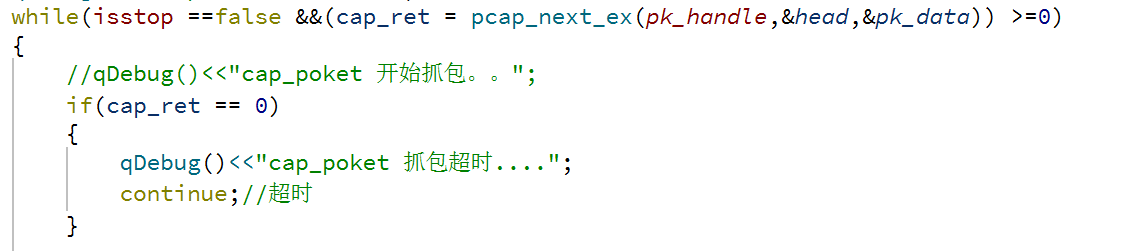
点击开始按钮后，首先对组件进行设置，关闭过滤组件，开始按钮；打开结束按钮。然后清楚上一次捕获的数据，并清空ui中组件的内容。最后检查设备和抓包对象的指针，如果没有问题就开始抓包（调用startcap（）进行抓包）。

Startcap首先获取用户选中的网卡，然后打开网卡，获取数据包捕获描述字。然后获取过滤条件，并设置过滤器。最后，创建cap\_poket对象进行抓包。具体函数见源代码。

注:winpcap只支持抓取以太网数据包。对于无线网，会一直抓包超时，在后续版本中可以使用Npcap进行对无线网包数据的捕获。



如图为cap\_poket类的内容，主要函数为run（）函数。

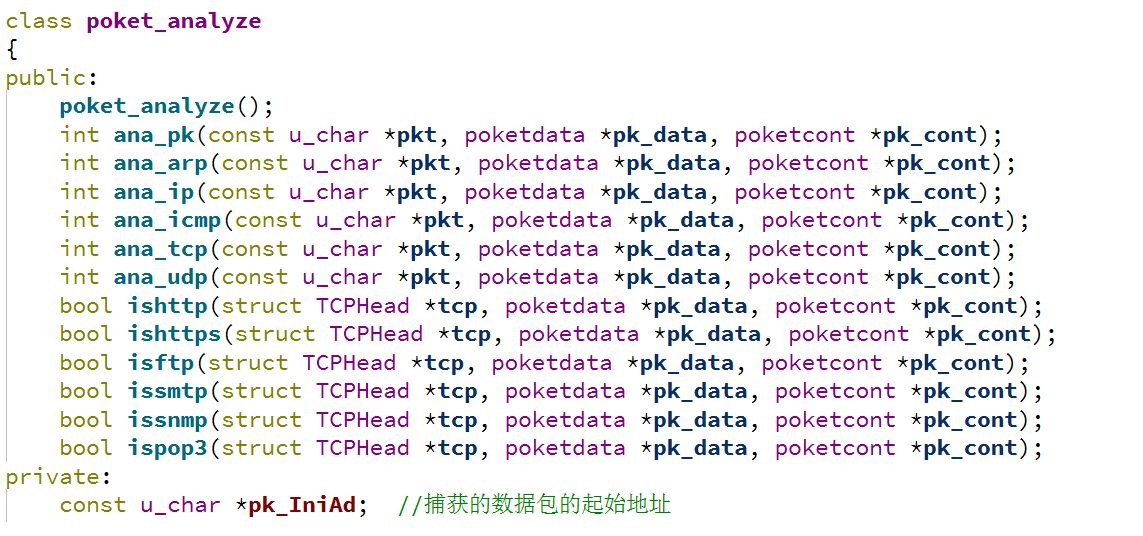


首先，调用pcap\_next\_ex函数循环捕获数据包。



然后，将包信息传输到包分析函数中，对包的协议进行分析，分析结束后将捕获到的字符串数据和分析后的结构体数据放入数组容器中。之后用信号更新ui中的数据包统计信息和table中的显示信息。

## 3.4包分析（poket\_analyze）



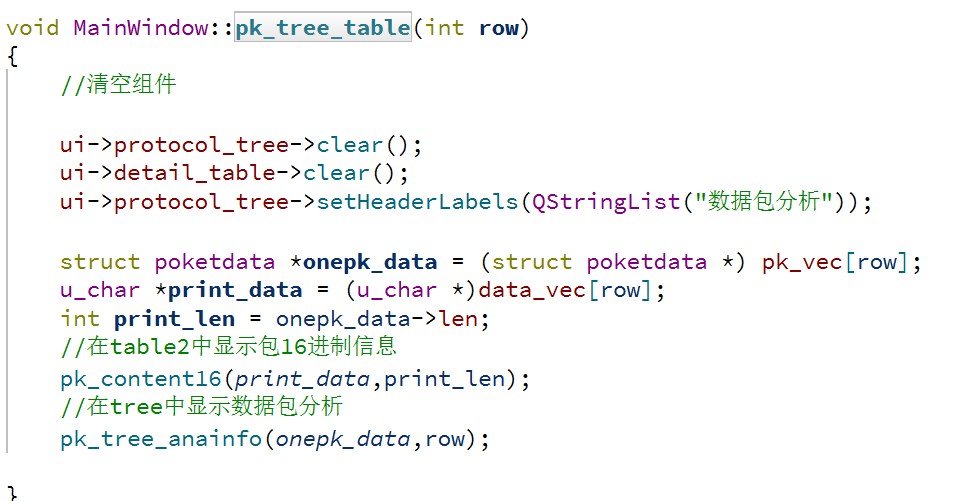
包分析类如图所示，其中，ana\_pk对数据链路层进行分析，并把包区分为arp和ip协议包。Ip分析时将ip包区分为icmp，tcp和udp协议包。



最后在tcp协议分析中通过端口将包区分为http，https，ftp，smtp，snmp和pop3协议。

## 3.5双击table某项显示





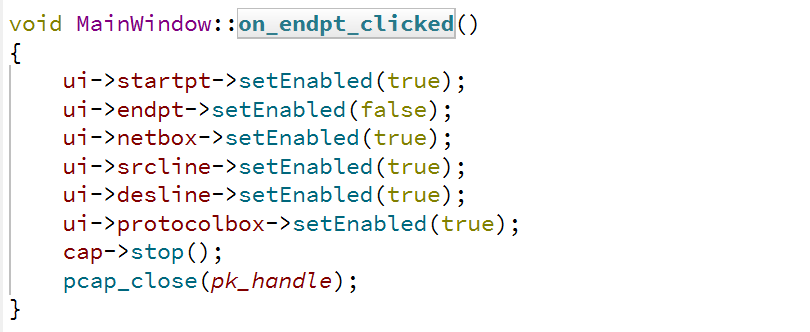
在双击表中某项后会在下面的tree和table中分别显示数据包16进制内容（pk\_content16(print\_data,print\_len);）和具体的包信息（pk\_tree\_anainfo(onepk\_data,row);）。

pk\_content16函数中，每行显示16个16进制数和内容，将不能打印的内容显示为“.”其他直接显示。



pk\_tree\_anainfo函数中，首先处理数据链路层信息显示。然后通过区分arp和ip协议来进行网络层信息显示。然后在ip基础下进一步显示tcp，udp和icmp信息。最后在tcp基础下显示http等数据信息。

## 3.6结束



如图点击结束按钮之后，打开过滤和开始按钮，关闭结束按钮，结束抓包线程。

4.后续版本开发

首先，本程序基本实现了包的抓取和对不同协议的过滤。后续将开发：

* 与文件相关的功能，如保存成“pcap”文件和从pcap文件中导入信息。
* 完善根据端口，包内容中特征判断包协议的代码
* 根据syn等信息分析那些tcp包属于同一连接，将其显示到cap\_table中包信息的部分。
* 分析其他协议中是否有可利用信息，同样显示到cap\_table中。
* 其他再说………

GitHub可见各个版本迭代更新，地址：