源代码组织架构：

Main.py: GUI部分，负责显示软件页面。

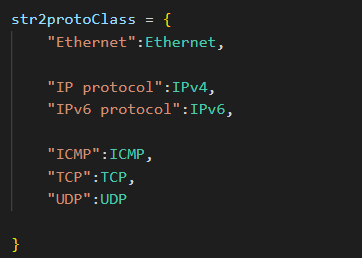
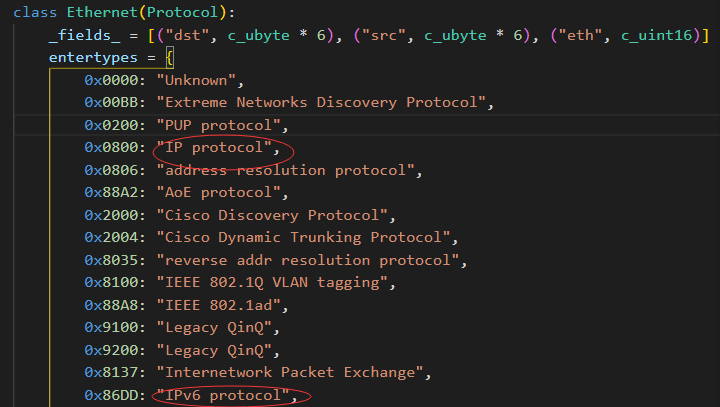
Sniffer.py:后台抓包并解析部分，负责调用winpcap对指定网卡进行抓包。

Layers：包解析部分，以插件的形式添加数据解析支持，在实验中实现了ethernet、icmp、ipv4、ipv6、tcp、udp等协议的识别。如果要添加新的协议支持，只需要在layers文件夹中新建类，该类应该以Protocol为父类，Protocol是一个BigEndianStructure的一个自定义子类。新定义的协议类需要至少填写以下字段：\_fields\_字段，表示协议的数据格式；如果协议还能支持下层协议，添加entertypes字段：表示协议能够识别的上层协议；self.info，表示协议在详细信息页显示出的信息，self.header\_len，表示协议头的长度。我们以udp协议的源码为例：



\_fields\_中以协议格式顺序将数据分割为各个字段，self.header\_len从字段中直接读取，最后构造self.info，在详细信息中显示数据包的源端口和目的端口。因为udp无下层协议字段，没有entertypes字段。

在实现新类后，将类以及其对应的字段添加到\_\_init\_\_.py的str2protoClass中，保证key值与类相匹配，软件即可自动识别新协议。（Key值应与entertypes字段内容一致）

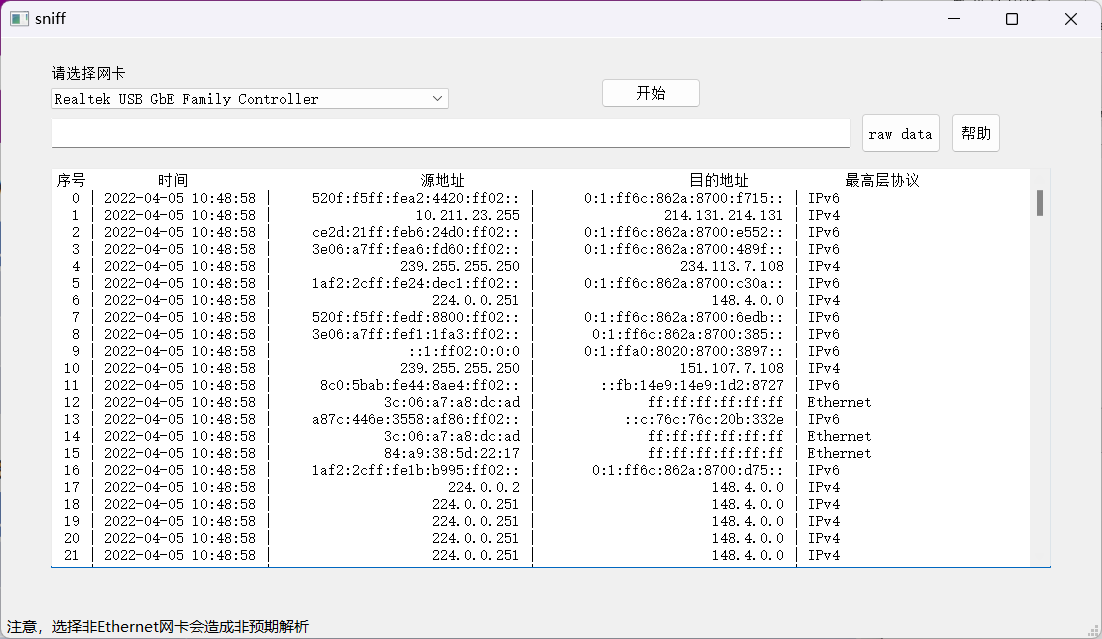
 

软件使用指南：

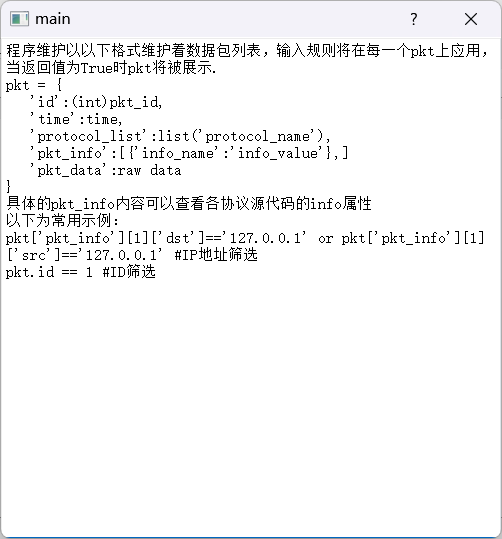
打开软件，其默认界面如图：



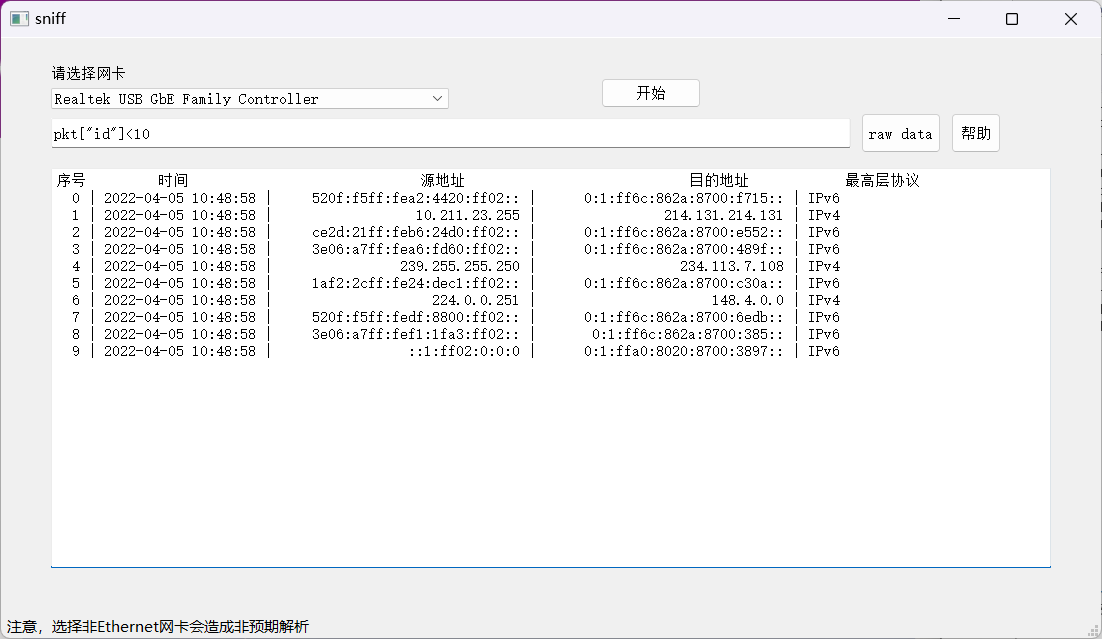
首先需要选择网卡，我们选择Realtek USB GbE Family Controller，而后点击开始即可捕包，再次点击停止捕包，下方信息框中将显示数据包的序号、时间、源地址、目的地址（若有IP地址显示IP地址，否则显示物理地址）以及可以解析出的最高层协议：



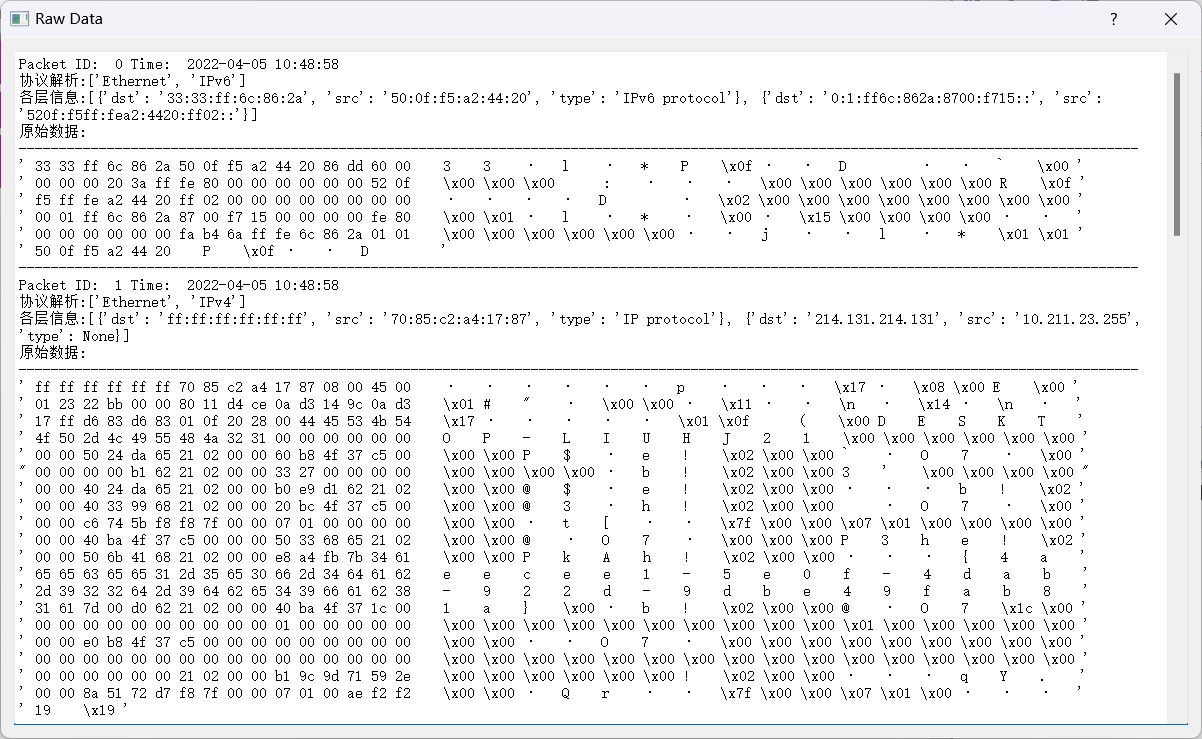
此时可以在中间空白条中输入筛选规则对抓到的数据包进行筛选，详见help：



我们向其中输入：pkt[‘id’]<10筛选出前10个数据包，输入回车后应用筛选规则：



在筛选完成后，我们可以点击raw data显示所选数据包的详细信息，详细信息包括数据包的ID，时间，解析出的各层协议、各层协议解析所给出的信息以及原始数据：



在演示视频中，我们还会给出各种用户不好的操作时带来的软件提示。