**中国矿业大学计算机学院**

**2018级本科生计算机网络实验报告**

实验内容 协议报文分析

学生姓名 孙正雨 学 号 08183039

专业班级 信安2班

学 院 计算机科学与技术学院

任课教师

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **课程基础理论掌握程度** | 熟练 🞏 | 较熟练 🞏 | 一般 🞏 | 不熟练 🞏 |
| **综合知识应用能力** | 强 🞏 | 较强 🞏 | 一般 🞏 | 差 🞏 |
| **报告内容** | 完整 🞏 | 较完整 🞏 | 一般 🞏 | 不完整 🞏 |
| **报告格式** | 规范 🞏 | 较规范 🞏 | 一般 🞏 | 不规范 🞏 |
| **实验完成状况** | 好 🞏 | 较好 🞏 | 一般 🞏 | 差 🞏 |
| **工作量** | 饱满 🞏 | 适中 🞏 | 一般 🞏 | 欠缺 🞏 |
| **学习、工作态度** | 好 🞏 | 较好 🞏 | 一般 🞏 | 差 🞏 |
| **抄袭现象** | 无 🞏 | 有 🞏 姓名: | | |
| **存在问题** |  | | | |
| **总体评价** |  | | | |

综合成绩： 任课教师签字：

年 月 日

**实验编号：03**

**实验名称：协议报文分析**

**实验内容：**

1. 分别获取不同互联网访问情形下的本机网卡数据包；过滤捕获和过滤显示不同条件的数据包。
2. 获取ARP协议，进行报文格式解析。
3. 获取ping、tracert、netstat命令对应的交互数据，进行报文格式解析。
4. 连续获取面向连接的互联网访问情形下的本机网卡数据包；对连续获取的数据包找到执行面向连接过程的报文，给出实现面向连接过程（TCP三次握手）的详细分析。
5. 分别对不同互联网访问情形下的数据包进行逐层分析，给出各层协议的主要参数及意义；要求分别获取DNS服务、WWW服务、Email服务、QQ通信、微信、迅雷文件下载等六种不同网络应用服务运行过程中的协议数据。
6. 对每种应用服务的协议数据需要从应用层、运输层、网络层、数据链路层四个层次对相关协议格式、字段取值进行解析。

**实验要求：**

（1）运用抓包工具，实时抓包，记录包状态变化；

（2）给出不同应用情境下的不同层次数据包的分析结果。

（3）对基于TCP的应用层协议，需要获取TCP协议的工作过程，验证连接建立的三次握手过程，以及滑动窗口工作机制。

（4）验证IP数据包、TCP报文段和UDP数据报的校验和。

（5）验证数据链路层的CRC冗余校验。

**预习要求：**

提前通过互联网或在实验室开始实验前登录实验管理服务器，点击预习链接，阅览或下载实验指导书——预习\网络协议\进阶-IP分组基本报文分析。

**操作与观察：**

正确按照实验指导书步骤操作，观察记录下操作结果。

**实验报告要求：**

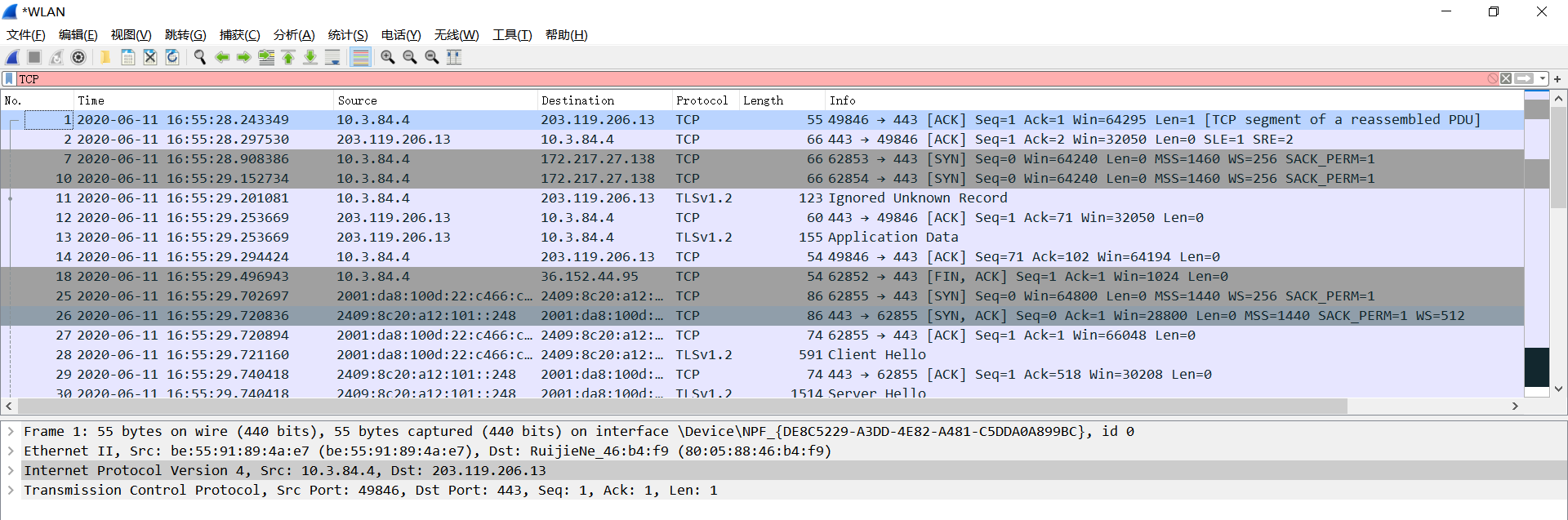
1. 按照实验要求，完成全部实验内容
2. 在标准实验报告书上填写全部实验操作记录和观察结果
3. 登录实验管理服务器，提交实验报告电子档。
4. 提交纸质版实验报告。

**实验报告内容：**

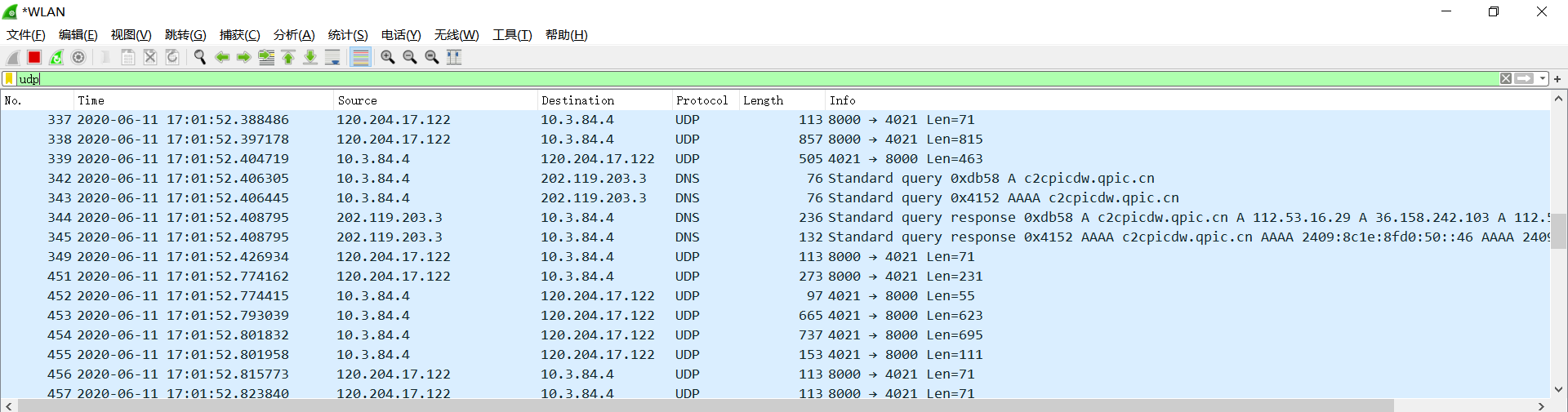
**实验一**

**通过设置过滤器，可选择自己想要获取的数据包类型**

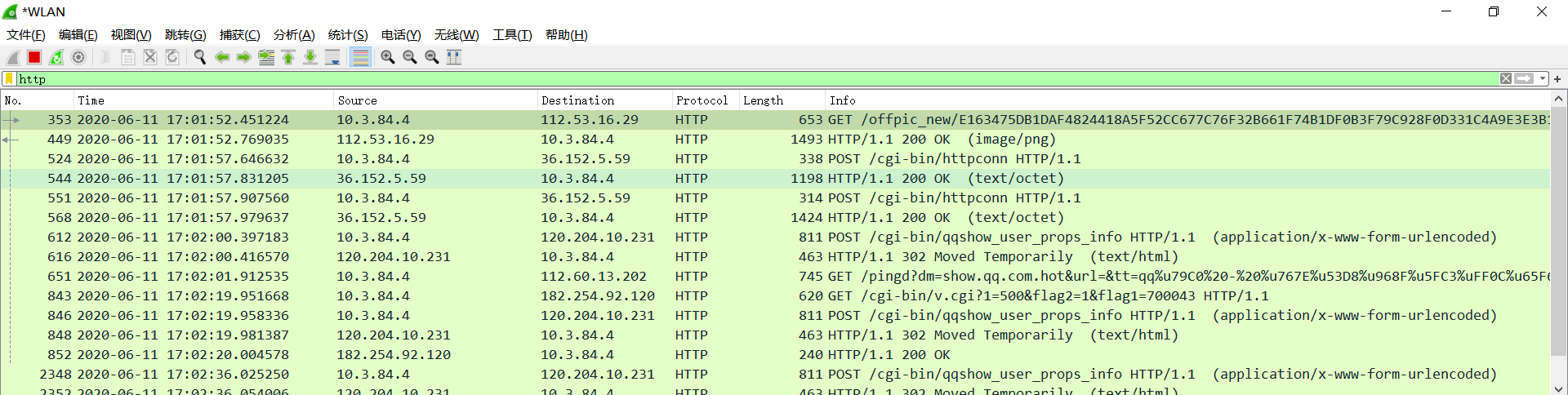
TCP



UDP



HTTP

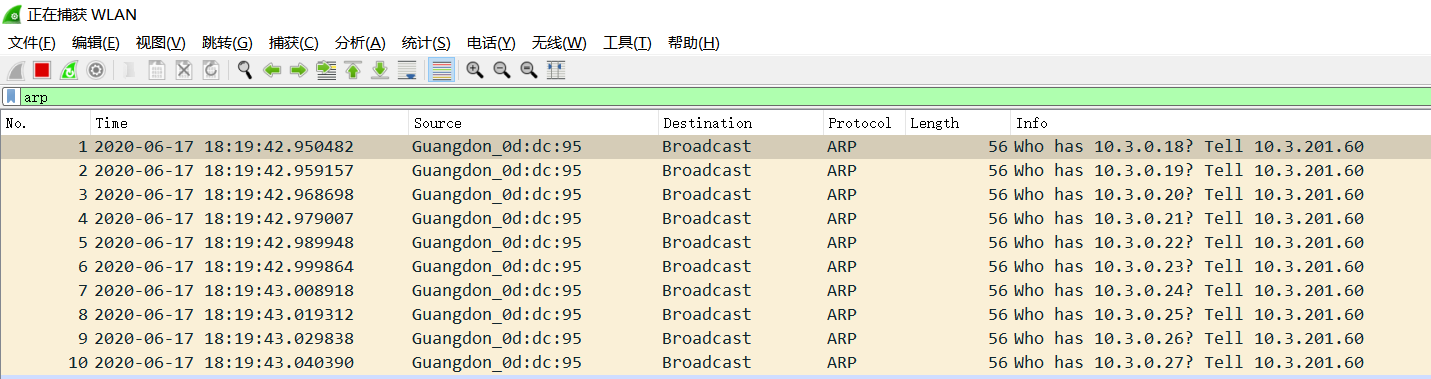


实验二

ARP帧结构



ARP抓包截图



数据包组成介绍

1. 以太网目的地址（6Byte）：当发送 ARP 请求时此处全为1 (FF:FF:FF:FF:FF:FF)，即为广播地址。当发送 ARP 响应时，此处即为目的端 MAC 地址。

2.以太网源地址（6Byte）：发送 ARP 请求的 MAC 地址，为本机 MAC 地址。

3.帧类型（2Byte）：表示的是后面的数据类型，ARP 请求和 ARP 应答这个值为 0x0806 。

4.硬件类型（2Byte）：硬件地址不只以太网一种，是以太网类型时此值为1。

5.协议类型（2Byte）：要映射的协议地址的类型，要对IPv4地址进行映射，此值为0x0800。

6.硬件地址长度（1Byte）：即为 MAC 地址长度 6 Byte。

7.协议地址长度（1Byte）：即为 IP 地址长度 4 Byte。

8.操作（2Byte）：值为1，表示进行ARP请求；值为2，表示进行ARP应答；值为3，表示进行RARP请求；值为4，表示进行RARP应答。

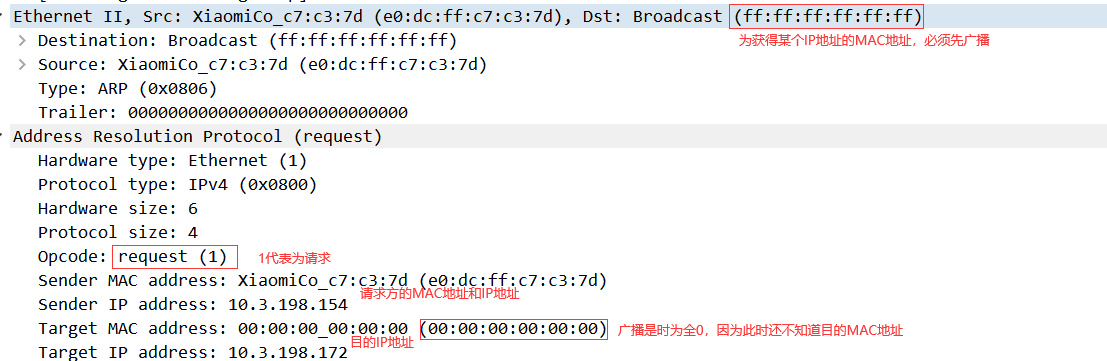
9.发送者硬件地址（6Byte）：这是本机的 MAC 地址，与第二个字段相同。

10.发送者 IP 地址（4Byte）：这是本机的 IP 地址。

11.目标硬件地址（6Byte）：在发送 ARP 请求时，还不知道目的端的 MAC 地址，所以此处全为 0 ( 00:00:00:00:00:00 )。当发送 ARP 请求报文是，此处即为目的端 MAC 地址。

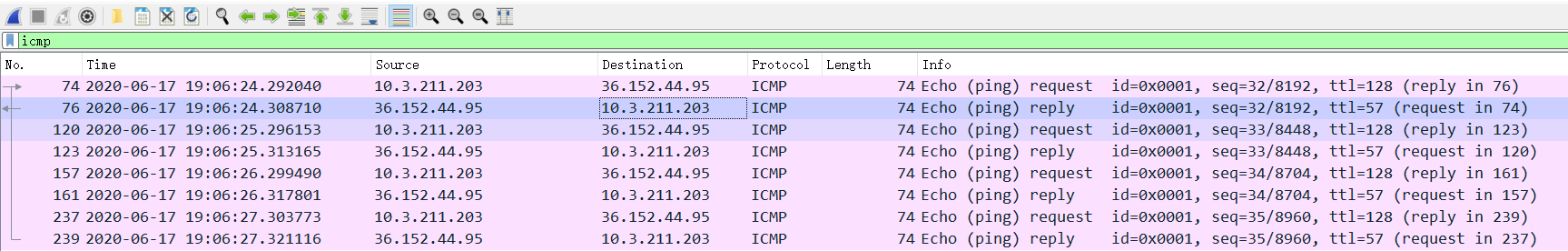
12.目标 IP 地址（4Byte）：目标端 IP 地址。

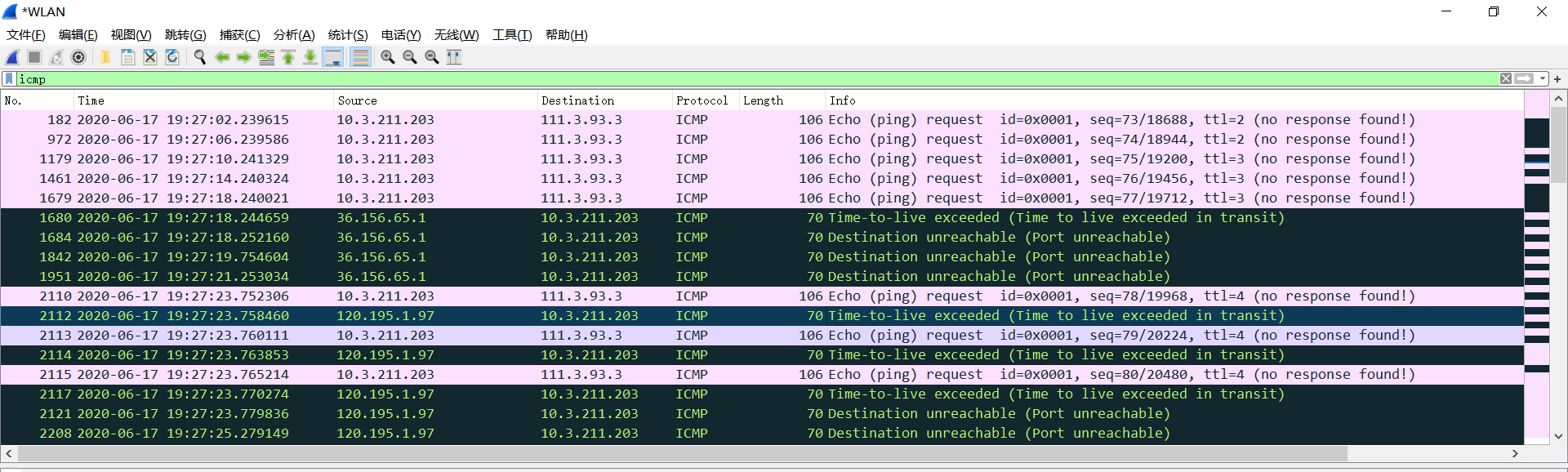
ARP请求数据包



实验三

Ping和tarcert命令的数据包类型为ICMP，netstat命令是数据包为TCP





协议类为ICMP

      类型（Type）： ICMP信息基于RFC规范的类型分类

      代码（Code）： ICMP信息基于RFC规范的子类型

     检验和（CheckSum）：用来保证ICMP数据的头部和数据部分的完整

     数据部分：依赖于类型和代码域的部分

ICMP请求包



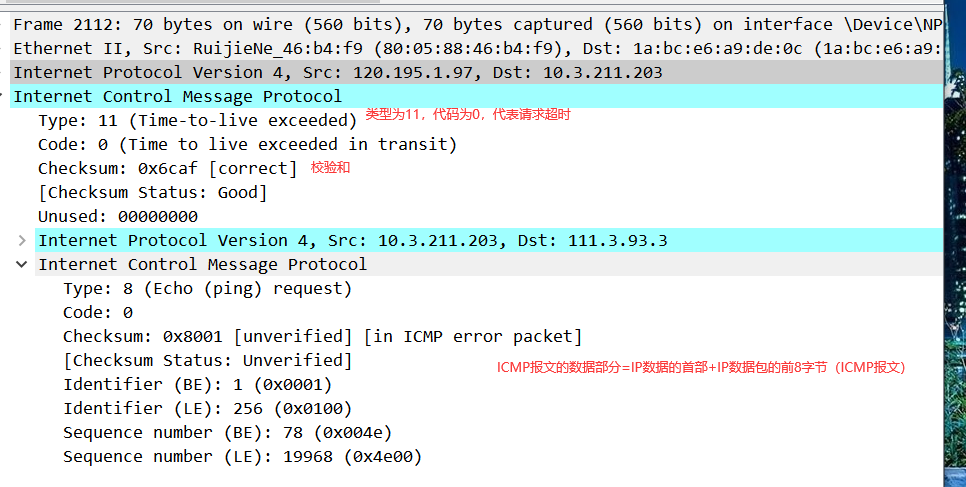
ICMP响应包



ICMP 请求超时包

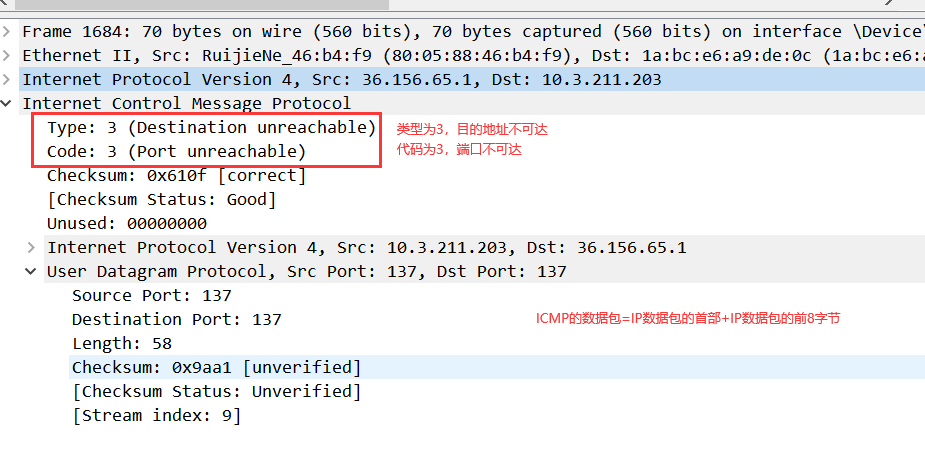


ICMP差错报文的格式是：把收到的IP数据报的首部和数据字段的前8个字节提取出来（也就是ICMP报文的格式）。作为ICMP报文的数据字段。在加上ICMP差错报文的前8个字节，就构成了ICMP差错报告报文。



ICMP不可达数据包





TCP

工作原理：

首先，客户机发送一个特殊的TCP报文段；

其次，服务器用另一个特殊的TCP报文段来响应；

最后，客户机再用第三个特殊报文段作为响应。

TCP协议格式



**源和目标端口**

用于多路复用/多路分解来自或送至上层应用的数据，可以这样理解，端口用来标识同一台计算机的不同进程。

**序列号和确认号**

这两个字段是TCP可靠传输服务的关键部分，序列号是该报文段首字节的字节流编号(**TCP把数据看成是有序的字节流**，TCP隐式地对数据流的每个字节进行编号)

**首部长度(4位)**

因为选项是不定长的，这就需要标识整个首部字段的长度(单位是32位字)，即5+选项个数。

**标志**

**URG**

指示报文段里存在着被发送方的上层实体标记为”紧急”数据，当URG=1时，其后的紧急指针指示紧急数据在当前数据段中的位置(相对于当前序列号的字节偏移量)，TCP接收方必须通知上层实体。

**ACK**

当ACK=0时，表示该数据段不包含确认信息，当ACK=1时，表示该报文段包括一个对已被成功接收报文段的确认。

**PSH**

当PSH=1时，接收方在收到数据后立即将数据交给上层，而不是直到整个缓冲区满。

**RST**

用于重置一个已经混乱的连接(如主崩溃)，也可用于拒绝一个无效的数据段或者拒绝一个连接请求。一般而言，如果你得到的数据段被设置了RST位，那说明你这一端有问题了。

**SYN**

用于建立连接过程，在连接请求中，**SYN=1和ACK=0**表示该数据段没有使用**捎带**的确认域，而连接应答捎带一个确认，即**SYN=1和ACK=1**。

注：捎带是指对客户机到服务器数据的**确认**被装载在一个承载服务器到客户机的数据报文段中。

**FIN**

用于释放一个连接，表示发送方已经没有数据要传输了。此时，接收方可能继续接收数据，好在SYN和FIN数据段都有序列号，从而保证了这两种数据段以正确顺序被处理。

**窗口大小**

用于流控制(确保连接的任何一方都不会过快地发送过量的分组而淹没另一方)，窗口大小指定了从**被确认的字节**算起可以发送多少个字节。

**校验和**

提供了额外可靠性，在计算检验和的时候，TCP的Checksum域设为0，如果数据域的字节数为奇数，则数据域填补一个额外的0字节。校验和算法：将所有的16位字按1的补码形式累加起来，取累加结果的补码。因此，当接收方执行同样计算时(包括Checksum域)，结果应该是0。

**紧急指针**

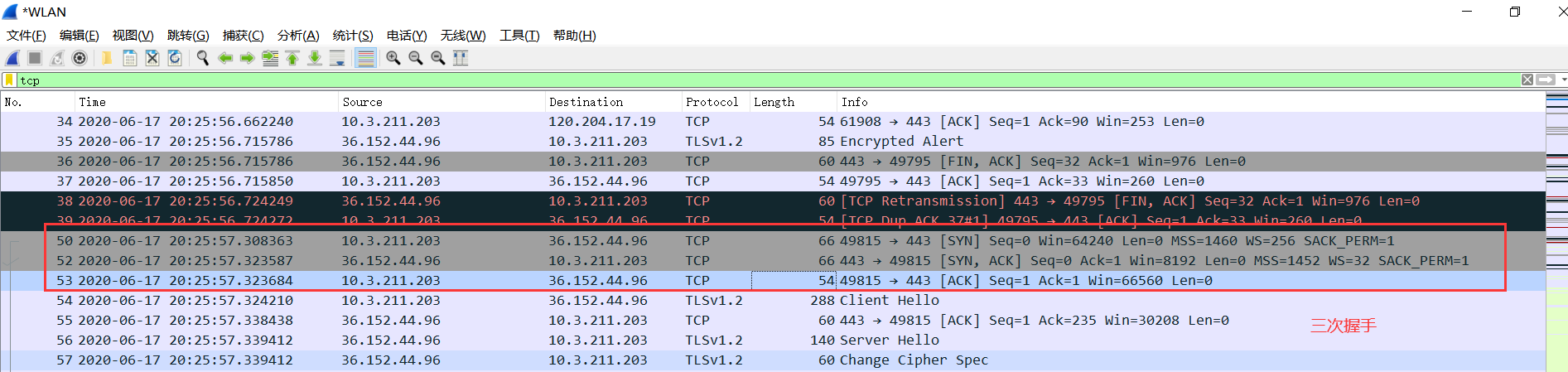
参考标志字段的URG位。

**选项**

选项部分是为了适合复杂网络环境和更好地服务于应用层设计的。TCP选项最长是40字节。

**数据**

无任何数据的TCP段也是合法的，通常用于确认和控制信息。

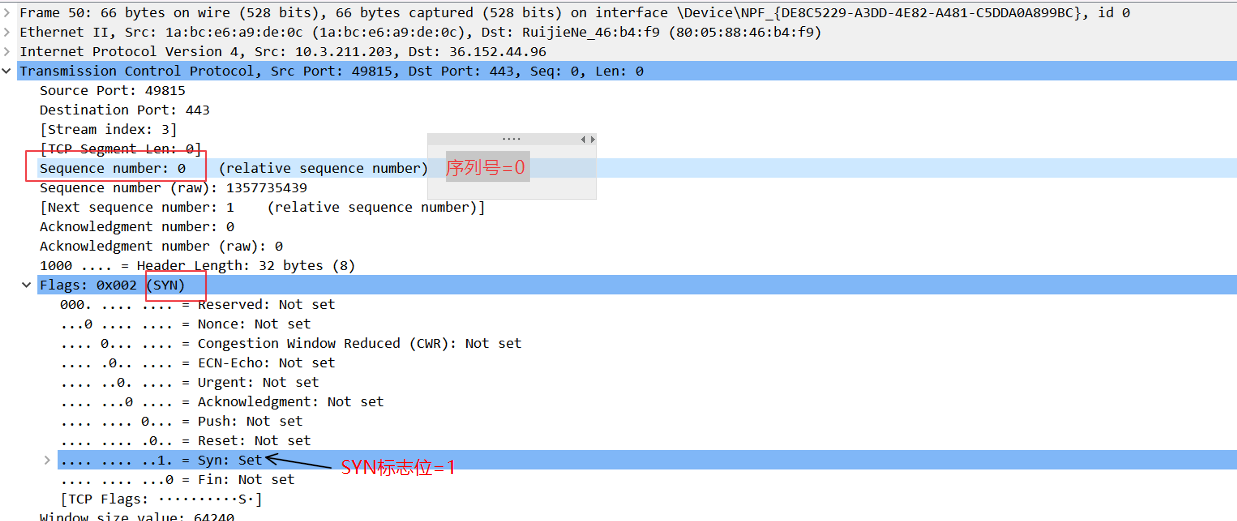


实验四

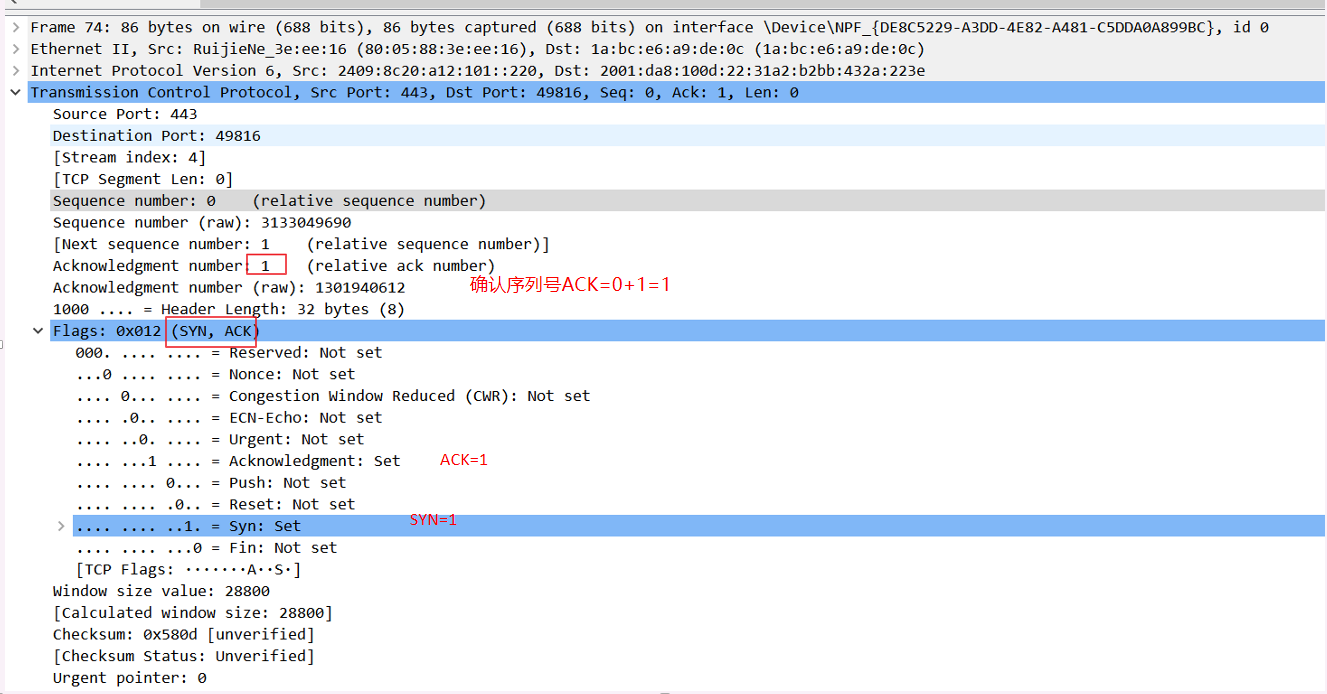
根据上述报文格式我们可以将wireshark捕获到的TCP包中的每个字段与之对应起来，更直观地感受一下TCP通信过程。先看三次握手，下图中的3条数据包就是一次TCP建立连接的过程。

第一次握手

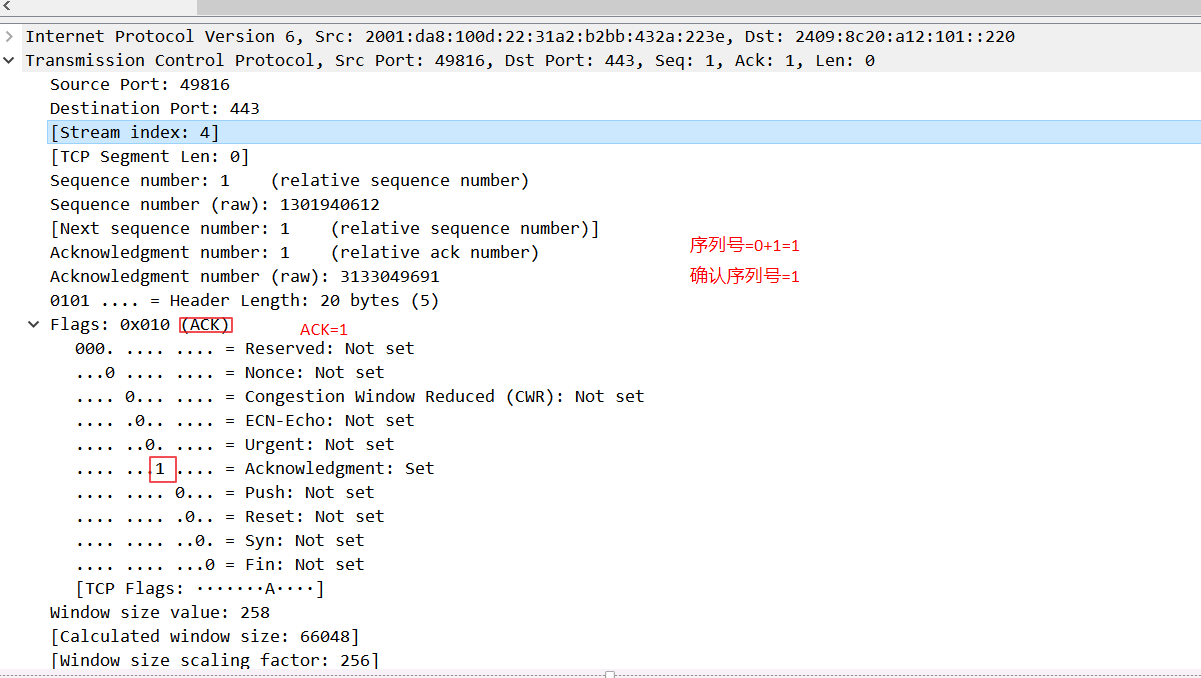
第一次握手，客户端发送一个TCP，标志位为SYN=1，序号seq为Sequence number=0， 49815-> 443，代表客户端请求建立连接；



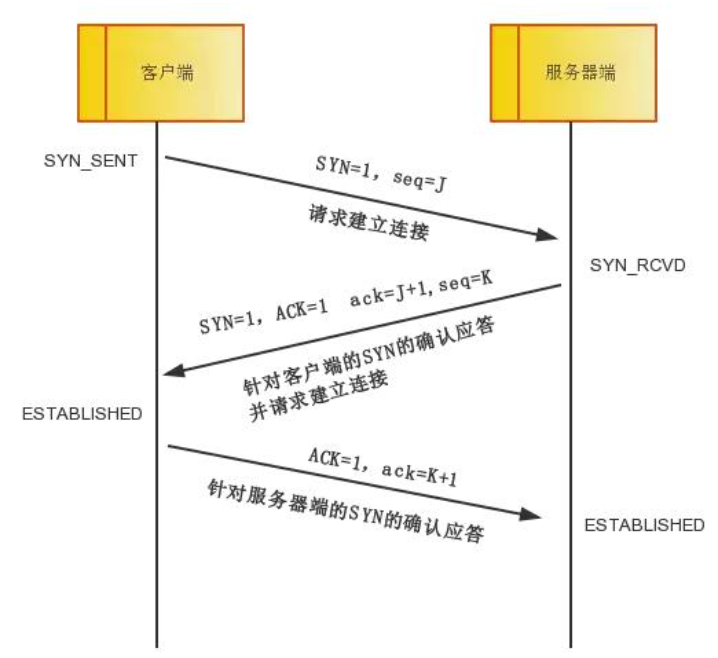
第二次握手，服务器向客户端返回一个数据包，SYN=1，ACK=1，443 -> 49815，将确认序号(Acknowledgement Number)设置为客户的序号seq(Sequence number)加1，即0+1=1；



第三次握手，客户端收到服务器发来的包后检查确认序号(Acknowledgement Number)是否正确，即第一次发送的序号seq加1（X+1= 0+1=1）。以及标志位ACK是否为1。若正确，客户端会再向服务器端发送一个数据包，SYN=0，ACK=1，确认序号(Acknowledgement Number)=Y+1=0+1=1，并且把服务器发来ACK的序号seq(Sequence number)加1发送给对方，发送序号seq为X+1= 0+1=1。客户端收到后确认序号值与ACK=1，49816 -> 443，至此，一次TCP连接就此建立，可以传送数据了。



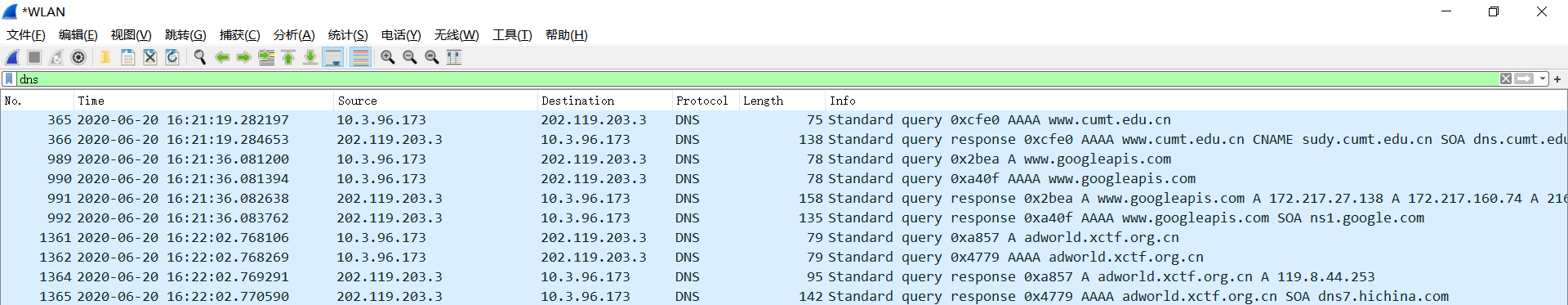
**三次握手总结**



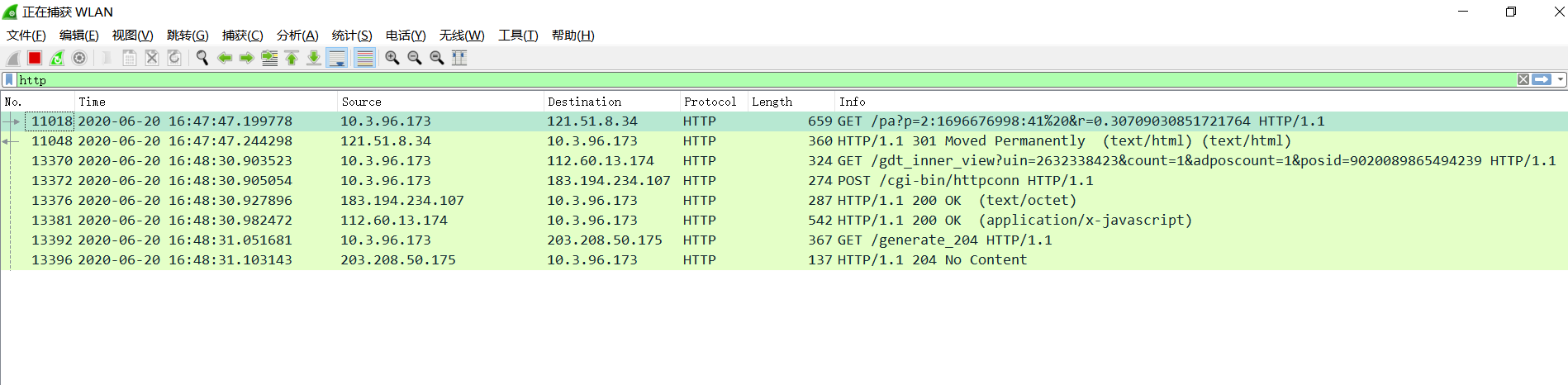
**实验五**

**DNS报文分析**

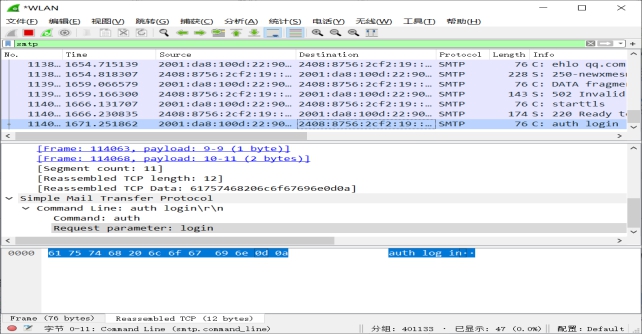
**Ping** [**www.cumt.edu.cn**](http://www.cumt.edu.cn)



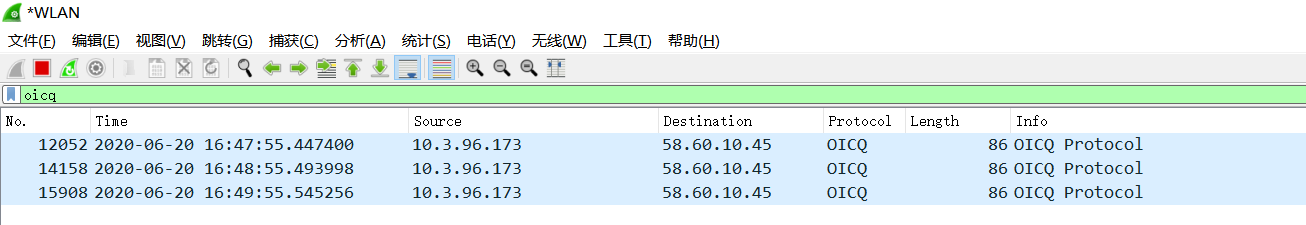
**WWW服务数据包**



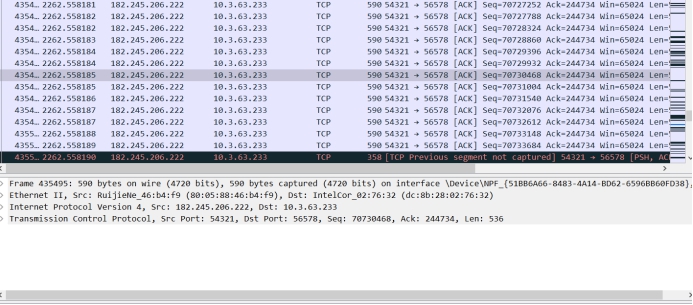
**邮箱服务**



**QQ通信服务**



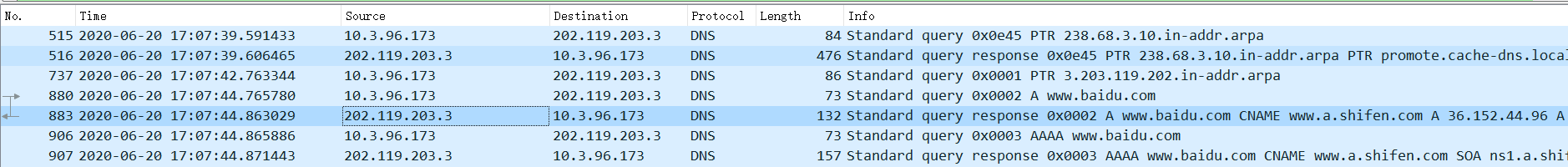
**迅雷下载报文**



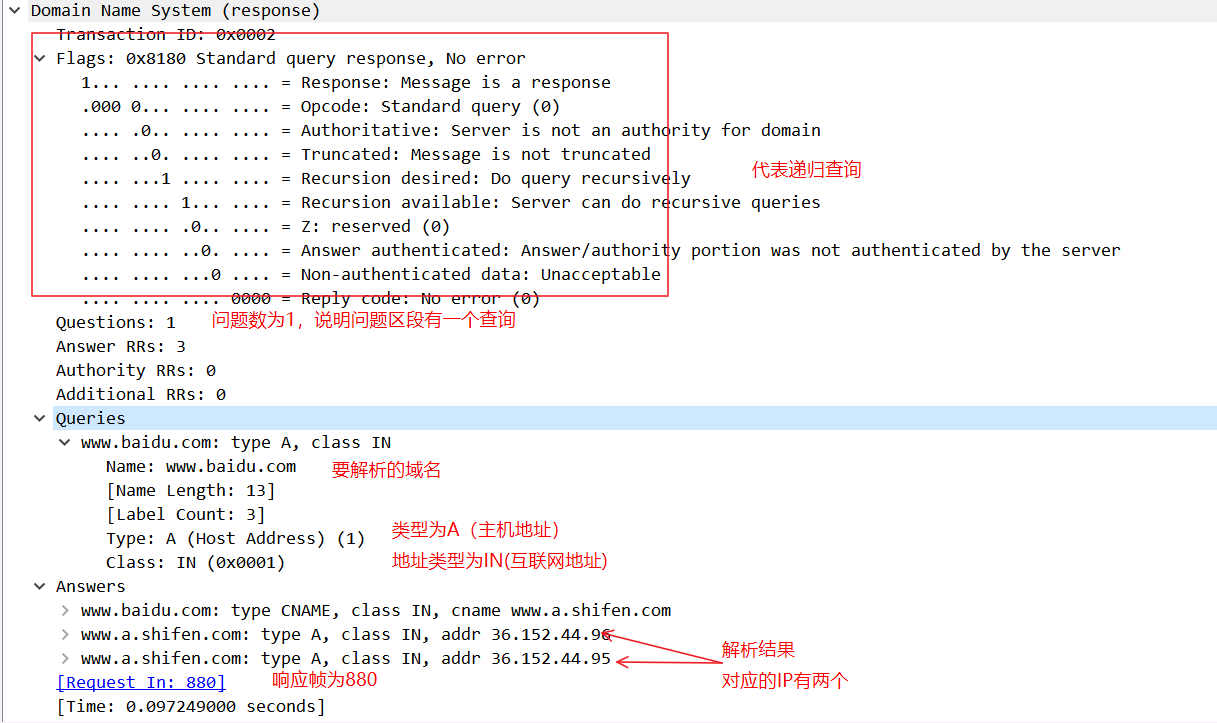
实验六

**DNS报文分析**

**Nslookup** [**www.baidu.com**](http://www.baidu.com)



1. 应用层

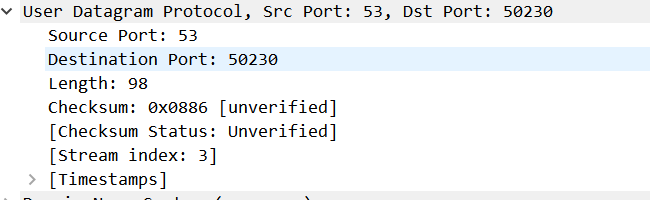


1. 运输层

使用的UDP协议

源端口为53，目的端口为50230

数据长度为98



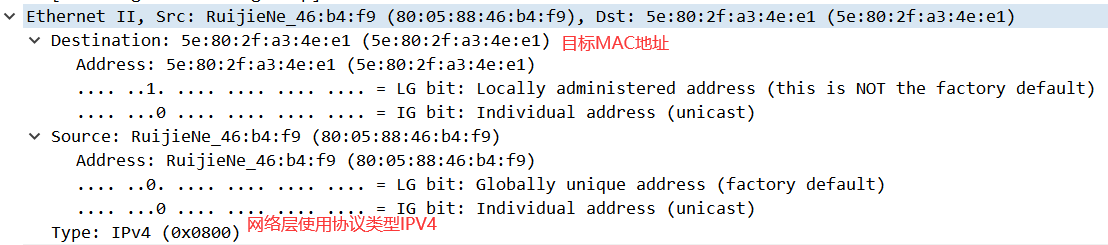
1. 网络层

源IP地址为202.119.203.3 目的IP地址 10.3.96.173



1. 数据链路层

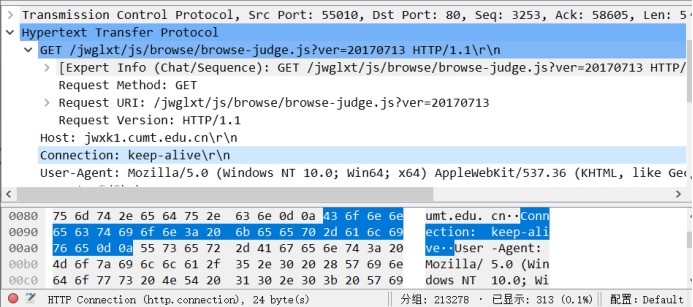
源MAC地址80:05:88:46:b4:f9 目标MAC地址5e:80:2f:a3:4e:e1



**www服务报文分析**

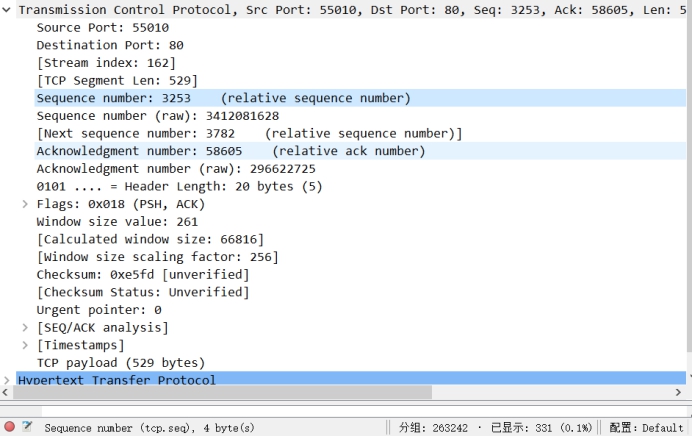
**对于http数据进行分析，发现五层协议，物理层，数据链路层，网路层，传输层，还有应用层**

**1.传输层**



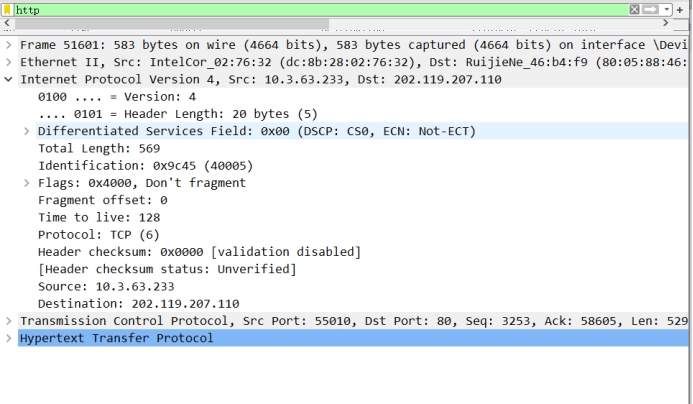
**这一次应用层以http报文显示，里面传递的东西表明，通过GET方式访问网站资源，访问的主机名为jwxk1.cumt.edu.cn,还包含有浏览器类型，连接方式等http报文信息**

1. **TCP层**



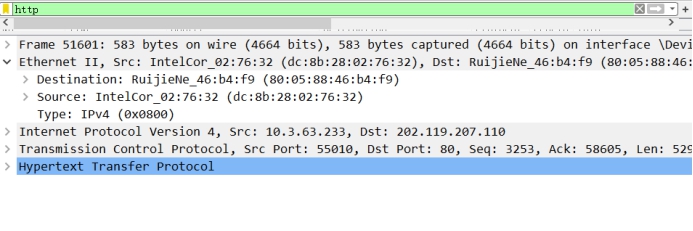
**http协议使用的运输层协议是TCP，源端口为55010，目的端口为80，还有它的序号3253以及确认信号，还可以看到标志位Flags：0x018（Push操作就是指在数据包到达接收端以后，立即传送给应用程序，而不是在缓冲区中排队，ACK此标志表示应答域有效，就是说前面所说的TCP应答号将会包含在TCP数据包中；有两个取值：0和1，为1的时候表示应答域有效，反之为0；），窗口大小:261，首部长度为5，检验和是0xe5fd，紧急指针Urgent pointer置0**

1. **IP层**



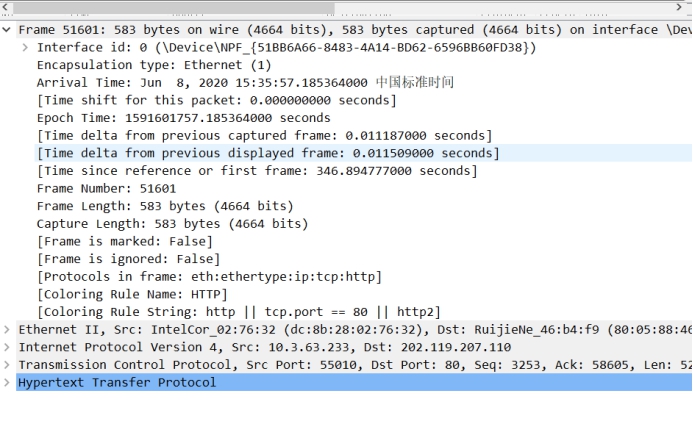
**IP层的内容，可以清楚分析出他的首部信息，版本号为4，首部长为20字节，还有标识40005，共569个字节，标识0x9c45(40005), 片偏移0个单位, TTL为128 ，协议字段6，代表了上层使用TCP，header checksum= 0校验失败（TCP和UDP校验和会由网卡计算，因此wireshark抓到的本机发送的TCP/UDP数据包的校验和都是错误的） 源IP为10.3.63.233，目的IP为202.119.207.110**

**3.数据链路层**



**上层IP类型为IPv4，源MAC地址: dc:8b:28:02:76:32,目的MAC地址80:05:88:46:b4:f9**

**4.物理层**



**接口id为0捕获日期为： Jun 8 2020，帧序号为51601，帧长度为583字节，捕获了583字节，帧内封装的协议层次结构：eth:ethertype:ip:tcp:http**

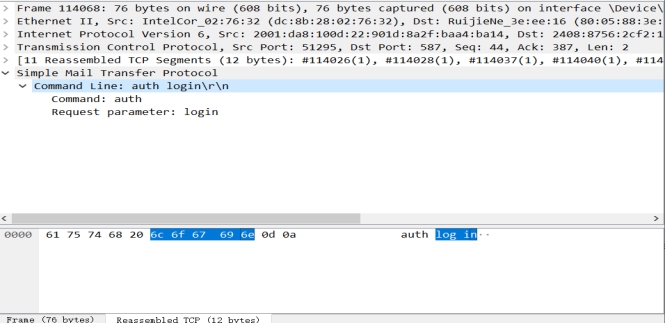
**邮箱报文分析**

**对于邮箱数据报进行分析，发现五层协议，物理层，数据链路层，网路层(IP)，传输层(TCP)，SMTP**

**五层协议，物理层，数据链路层，网路层(IP)，传输层(TCP)的分析·和上面一样，下面只看smtp协议**

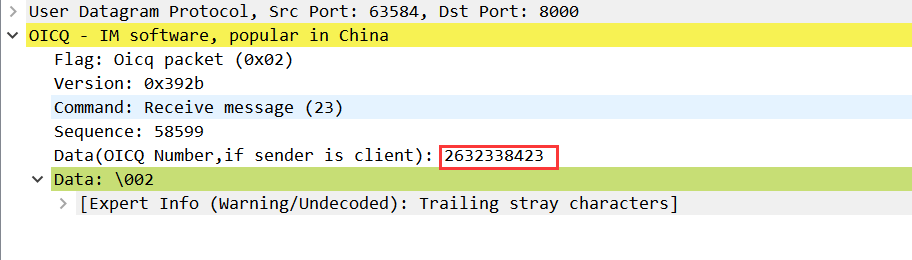
**1.应用层**

**Smtp采用交互式的方式，报文的内容是有意义的ascll码，可以直接读懂，不需要计算。这里的命令是auth login**



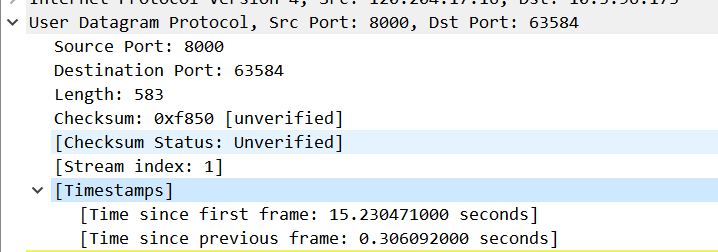
**对于QQ的包可以得出来应用层是使用了OICQ的协议，这是腾讯自己设计的协议，里面可以看出发送这个报文的QQ号(2632338423)，命令(get statue of friend),以及加密过了的数据**

**1.应用层**



**2.运输层**

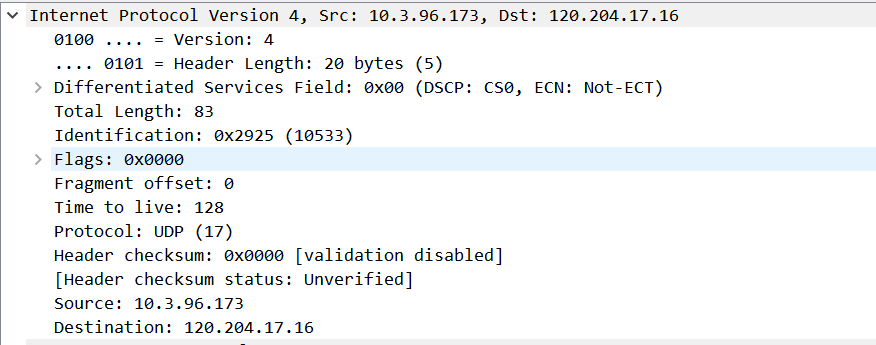
使用的是UDP的传输协议，源端口是8000，目的端口是63584，总和检验码0xf850



**3.IP层**

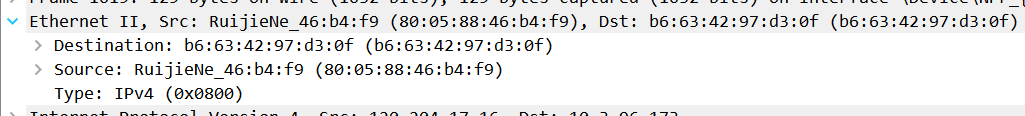
**版本号为4，首部长为20字节，还有标识10533，**

**TTL为55，协议字段17，代表了上层使用UDP，下面就是源IP为10.3.96.173，目的IP为120.204.17.16**

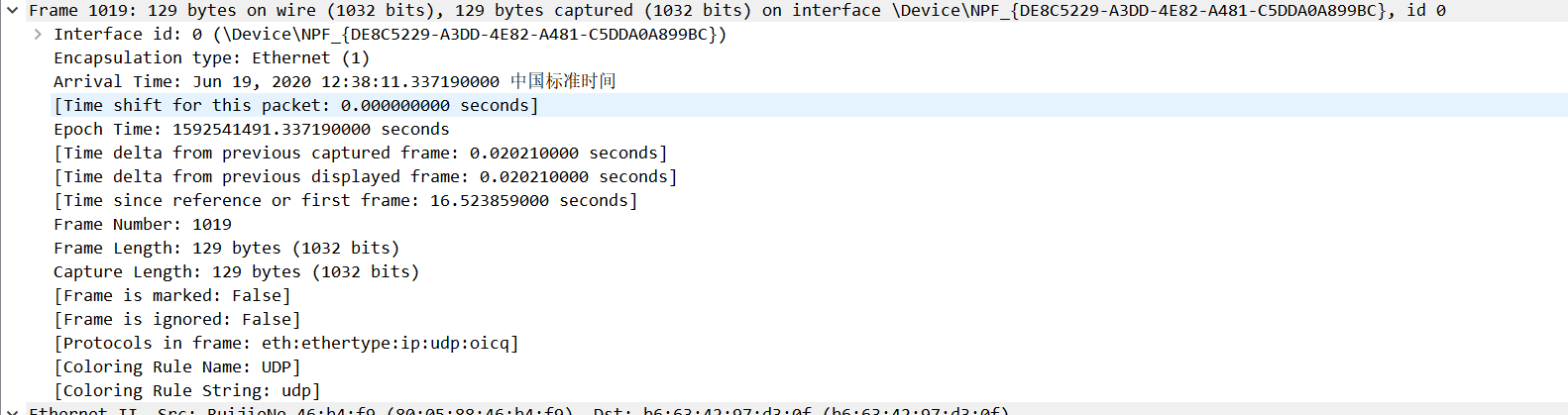


**4.链路层**

**上层IP类型为IPv4，目的MAC地址:** **80:05:88:46:b4:f9,源MAC地址b6:63:42:97:d3:0f**



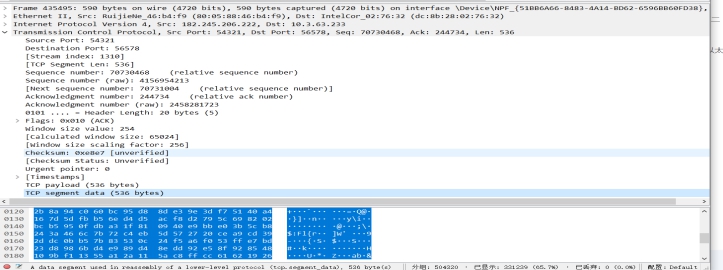
**5.物理层**



**迅雷报文分析**

1.运输层

**这里使用TCP协议，源端口为54321，目的端口为56578，还有它的序号70730468以及确认信号，还可以看到标志位Flags：0x010，窗口大小:254，首部长度为20，检验和是0xe8e7，紧急指针Urgent pointer置0.和前面不一样的是这里的TCP协议还多了时间戳这个字段**



迅雷的其它层协议的分析和上面的分析思路一样

**实验体会：**

在这之前对报文分析并不是很了解，做这个任务的时候不得不去查找各种资料，学习怎么去分析，到现在算是懂了点东西。学会了探测拓扑结构，以及测试互联网的接入路径，学会了运用wireshark抓包