

# 浙江大学

## 本科实验报告

课程名称： 计算机网络

实验名称： 网络协议分析

姓 名：

学 院： 计算机学院

系： 计算机系

专 业： 计算机科学与技术

学 号：

指导教师： 黄正谦

2022 年 11 月 16 日

# 浙江大学实验报告

## 一、 实验目的

- 学习使用 Wireshark 抓包工具。
- 观察和理解常见网络协议的交互过程
- 理解数据包分层结构和格式。

## 二、 实验内容

- Wireshark 是 PC 上使用最广泛的免费抓包工具，可以分析大多数常见的协议数据包。有 Windows 版本和 Mac 版本，可以免费从网上下载。
- 掌握网络协议分析软件 Wireshark 的使用，学会配置过滤器
- 观察所在网络出现的各类网络协议，了解其种类和分层结构
- 观察捕获到的数据包格式，理解各字段含义
- 根据要求配置 Wireshark，捕获某一类协议的数据包，并分析解读

## 三、 主要仪器设备

- 联网的 PC 机、Windows、Linux 或 Mac 操作系统、浏览器软件
- WireShark 协议分析软件

## 四、 操作方法与实验步骤

- 安装网络包捕获软件 Wireshark
- 配置网络包捕获软件，捕获所有机器的数据包
- 观察捕获到的数据包，并对照解析结果和原始数据包
- 配置网络包捕获软件，只捕获特定 IP 或特定类型的包
- 抓取以下通信协议数据包，观察通信过程和数据包格式
  - ✓ PING：测试一个目标地址是否可达
  - ✓ TRACE ROUTE：跟踪一个目标地址的途经路由
  - ✓ NSLOOKUP：查询一个域名
  - ✓ HTTP：访问一个网页

## 五、实验数据记录和处理

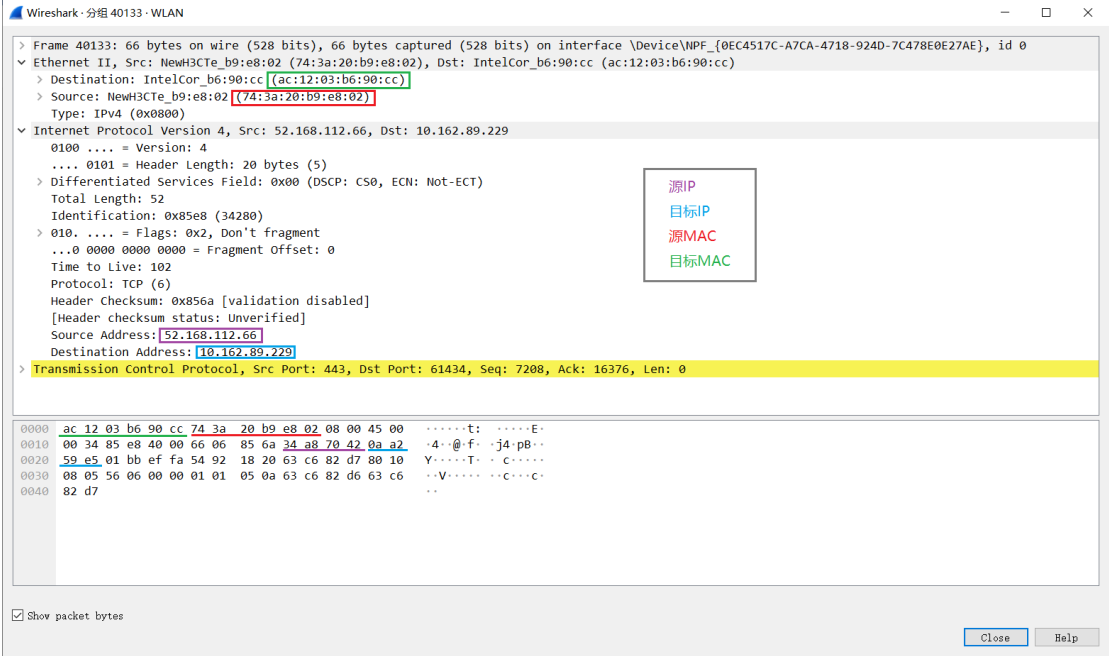
### ✧ Part One

1. 运行 Wireshark 软件，开始捕获数据包，列出你看到的协议名字（至少 5 个）。

协议名：OICQ, TCP, UDP, TLSv1.2, DNS

2. 找一个包含 IP 的数据包，这个数据包有 4 层。最高层协议是 TCP，从 Ethernet 开始往上，各层协议的名字分别为：Ethernet II, IPv4, TCP。

展开 IP 层协议，标出源 IP 地址、目标 IP 地址及其在数据包中的具体位置，展开 Ethernet 层，标出源 MAC 地址和目标 MAC 地址及其在数据包中的具体位置。



3. 配置应用显示过滤器，让界面只显示某一协议类型的数据包（输入协议名称）。

使用的过滤器：oicq，希望显示的协议类型：OICQ。

截图：

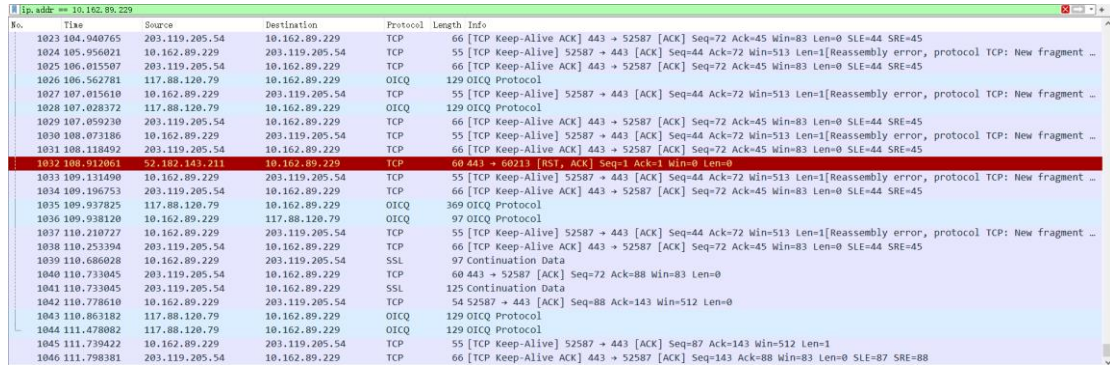
The screenshot shows a Wireshark packet capture window with the filter 'oicq' applied. The packet list on the left shows a list of packets of type OICQ. The packet details pane on the right shows the following layers: OICQ. The packet list is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
224	15.455459	117.88.120.79	10.162.89.229	OICQ	121	OICQ Protocol
225	15.456109	10.162.89.229	117.88.120.79	OICQ	97	OICQ Protocol
240	16.725097	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
251	19.626712	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
255	21.458500	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
256	21.517839	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
283	26.996450	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
291	29.040817	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
294	30.168966	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
295	30.207500	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
304	34.427620	10.162.89.229	117.88.120.79	OICQ	81	OICQ Protocol
305	34.453405	117.88.120.79	10.162.89.229	OICQ	89	OICQ Protocol
310	36.406609	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
314	37.432129	117.88.120.79	10.162.89.229	OICQ	457	OICQ Protocol
315	37.432815	10.162.89.229	117.88.120.79	OICQ	97	OICQ Protocol
343	39.109638	10.162.89.229	117.88.120.79	OICQ	81	OICQ Protocol
344	39.142166	117.88.120.79	10.162.89.229	OICQ	137	OICQ Protocol
352	40.807543	117.88.120.79	10.162.89.229	OICQ	369	OICQ Protocol
353	40.808121	10.162.89.229	117.88.120.79	OICQ	97	OICQ Protocol
368	41.450134	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
373	42.146402	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
377	42.464759	117.88.120.79	10.162.89.229	OICQ	129	OICQ Protocol
381	43.104272	10.162.89.229	117.88.120.79	OICQ	81	OICQ Protocol
382	43.142155	117.88.120.79	10.162.89.229	OICQ	209	OICQ Protocol

#### 4. 配置应用显示过滤器，让界面只显示某个 IP 地址的数据包（ip.addr==x.x.x.x）。

使用的过滤器：ip.addr == 10.162.89.229，希望显示的 IP 地址：10.162.89.229。

截图：

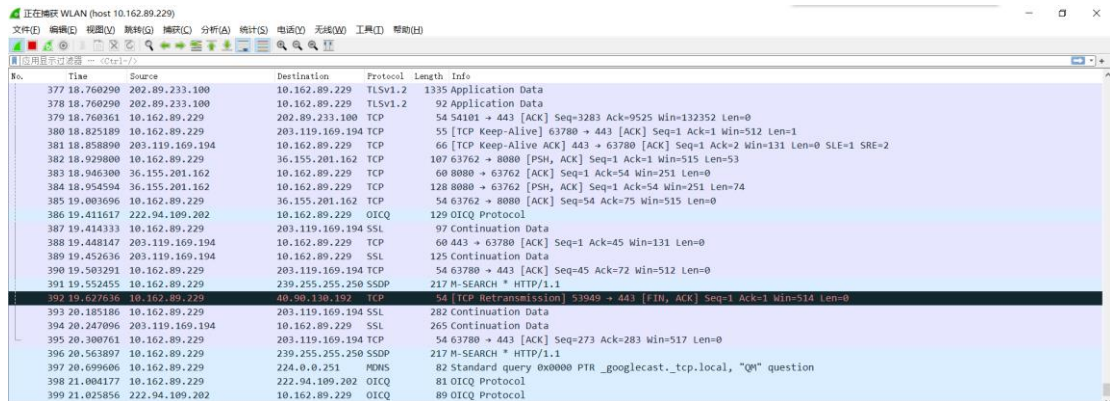


No.	Time	Source	Destination	Protocol	Length	Info
1023	184.940765	203.119.205.54	10.162.89.229	TCP	66	[TCP Keep-Alive ACK] 443 → 52587 [ACK] Seq=72 Ack=45 Win=83 Len=0 SLE=44 SRE=45
1024	185.956021	10.162.89.229	203.119.205.54	TCP	55	[TCP Keep-Alive] 52587 → 443 [ACK] Seq=44 Ack=72 Win=513 Len=1 [Reassembly error, protocol TCP: New fragment ...]
1025	186.015507	203.119.205.54	10.162.89.229	TCP	66	[TCP Keep-Alive ACK] 443 → 52587 [ACK] Seq=72 Ack=45 Win=83 Len=0 SLE=44 SRE=45
1026	186.562781	117.88.120.79	10.162.89.229	ICMP	129	ICMP Protocol
1027	187.015610	10.162.89.229	203.119.205.54	TCP	55	[TCP Keep-Alive] 52587 → 443 [ACK] Seq=44 Ack=72 Win=513 Len=1 [Reassembly error, protocol TCP: New fragment ...]
1028	187.028372	117.88.120.79	10.162.89.229	ICMP	129	ICMP Protocol
1029	187.059230	203.119.205.54	10.162.89.229	TCP	66	[TCP Keep-Alive ACK] 443 → 52587 [ACK] Seq=72 Ack=45 Win=83 Len=0 SLE=44 SRE=45
1030	188.073186	10.162.89.229	203.119.205.54	TCP	55	[TCP Keep-Alive] 52587 → 443 [ACK] Seq=44 Ack=72 Win=513 Len=1 [Reassembly error, protocol TCP: New fragment ...]
1031	188.118492	203.119.205.54	10.162.89.229	TCP	66	[TCP Keep-Alive ACK] 443 → 52587 [ACK] Seq=72 Ack=45 Win=83 Len=0 SLE=44 SRE=45
1032	188.912061	52.182.143.211	10.162.89.229	TCP	60	443 → 60211 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1033	189.131490	10.162.89.229	203.119.205.54	TCP	55	[TCP Keep-Alive] 52587 → 443 [ACK] Seq=44 Ack=72 Win=513 Len=1 [Reassembly error, protocol TCP: New fragment ...]
1034	189.196753	203.119.205.54	10.162.89.229	TCP	66	[TCP Keep-Alive ACK] 443 → 52587 [ACK] Seq=72 Ack=45 Win=83 Len=0 SLE=44 SRE=45
1035	189.937825	117.88.120.79	10.162.89.229	ICMP	369	ICMP Protocol
1036	189.938120	10.162.89.229	117.88.120.79	ICMP	97	ICMP Protocol
1037	110.210727	10.162.89.229	203.119.205.54	TCP	55	[TCP Keep-Alive] 52587 → 443 [ACK] Seq=44 Ack=72 Win=513 Len=1 [Reassembly error, protocol TCP: New fragment ...]
1038	110.253394	203.119.205.54	10.162.89.229	TCP	66	[TCP Keep-Alive ACK] 443 → 52587 [ACK] Seq=72 Ack=45 Win=83 Len=0 SLE=44 SRE=45
1039	110.686028	10.162.89.229	203.119.205.54	SSL	97	Continuation Data
1040	110.733045	203.119.205.54	10.162.89.229	TCP	60	443 → 52587 [ACK] Seq=72 Ack=88 Win=83 Len=0
1041	110.733045	203.119.205.54	10.162.89.229	SSL	125	Continuation Data
1042	110.778610	10.162.89.229	203.119.205.54	TCP	54	52587 → 443 [ACK] Seq=88 Ack=143 Win=512 Len=0
1043	110.863182	117.88.120.79	10.162.89.229	ICMP	129	ICMP Protocol
1044	111.478082	117.88.120.79	10.162.89.229	ICMP	129	ICMP Protocol
1045	111.739422	10.162.89.229	203.119.205.54	TCP	55	[TCP Keep-Alive] 52587 → 443 [ACK] Seq=87 Ack=143 Win=512 Len=1
1046	111.798381	203.119.205.54	10.162.89.229	TCP	66	[TCP Keep-Alive ACK] 443 → 52587 [ACK] Seq=143 Ack=88 Win=83 Len=0 SLE=87 SRE=88

#### 5. 配置捕获过滤器，只捕获某个 IP 地址的数据包（host x.x.x.x）。

使用的过滤器：host 10.162.89.229，希望捕获的 IP 地址：10.162.89.229。

截图：



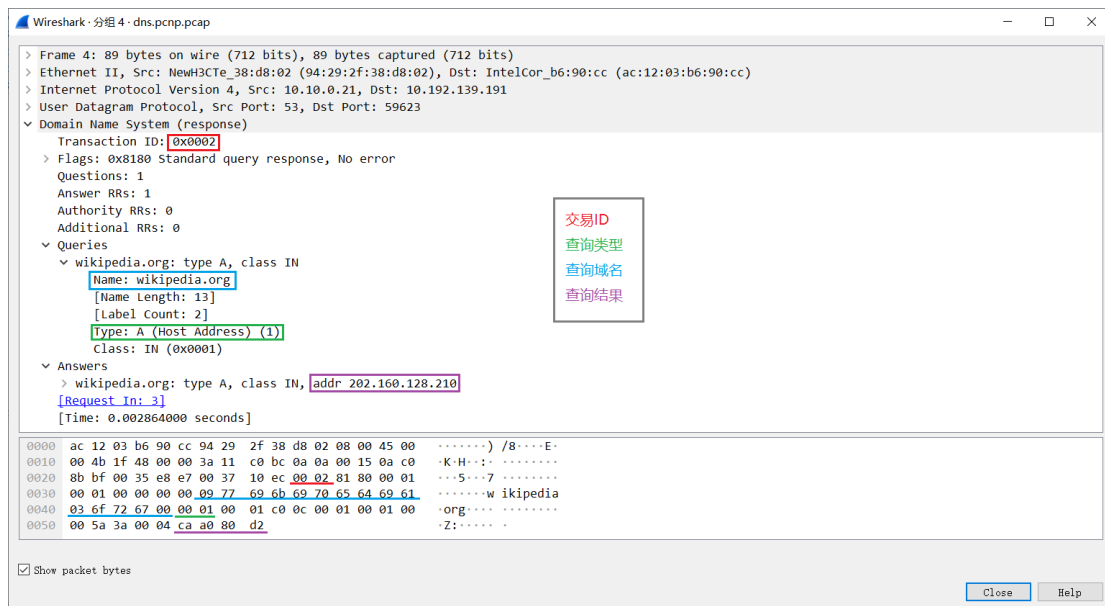
No.	Time	Source	Destination	Protocol	Length	Info
377	18.760290	202.89.233.100	10.162.89.229	TLSv1.2	1335	Application Data
378	18.760290	202.89.233.100	10.162.89.229	TLSv1.2	92	Application Data
379	18.760361	10.162.89.229	202.89.233.100	TCP	54	54101 → 443 [ACK] Seq=3283 Ack=9525 Win=132352 Len=0
380	18.825189	10.162.89.229	203.119.169.194	TCP	55	[TCP Keep-Alive] 63780 → 443 [ACK] Seq=1 Ack=1 Win=512 Len=1
381	18.858890	203.119.169.194	10.162.89.229	TCP	66	[TCP Keep-Alive ACK] 443 → 63780 [ACK] Seq=1 Ack=2 Win=131 Len=0 SLE=1 SRE=2
382	18.929800	10.162.89.229	36.155.201.162	TCP	107	63762 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=515 Len=53
383	18.946300	36.155.201.162	10.162.89.229	TCP	60	8080 → 63762 [ACK] Seq=1 Ack=54 Win=251 Len=0
384	18.954594	36.155.201.162	10.162.89.229	TCP	128	8080 → 63762 [PSH, ACK] Seq=1 Ack=54 Win=251 Len=74
385	19.003690	10.162.89.229	36.155.201.162	TCP	54	63762 → 8080 [ACK] Seq=54 Ack=75 Win=515 Len=0
386	19.411617	222.94.109.202	10.162.89.229	ICMP	129	ICMP Protocol
387	19.414333	10.162.89.229	203.119.169.194	SSL	97	Continuation Data
388	19.448147	203.119.169.194	10.162.89.229	TCP	60	443 → 63780 [ACK] Seq=1 Ack=45 Win=131 Len=0
389	19.452636	203.119.169.194	10.162.89.229	SSL	125	Continuation Data
390	19.503291	10.162.89.229	203.119.169.194	TCP	54	63780 → 443 [ACK] Seq=45 Ack=72 Win=512 Len=0
391	19.552455	10.162.89.229	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
392	19.627636	10.162.89.229	40.90.130.192	TCP	54	[TCP Retransmission] 53949 → 443 [FIN, ACK] Seq=1 Ack=1 Min=514 Len=0
393	20.185180	10.162.89.229	203.119.169.194	SSL	282	Continuation Data
394	20.247096	203.119.169.194	10.162.89.229	SSL	265	Continuation Data
395	20.300761	10.162.89.229	203.119.169.194	TCP	54	63780 → 443 [ACK] Seq=273 Ack=283 Win=517 Len=0
396	20.563897	10.162.89.229	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
397	20.699606	10.162.89.229	224.0.0.251	MDNS	82	Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
398	21.004177	10.162.89.229	222.94.109.202	ICMP	81	ICMP Protocol
399	21.025856	222.94.109.202	10.162.89.229	ICMP	89	ICMP Protocol

#### 6. 配置捕获过滤器，只捕获某类协议的数据包（tcp port xx 或者 udp port xx）。

使用的过滤器：tcp port 80，希望捕获的协议类型：TCP。

截图：





任务 2: 使用 Ping 命令，分别测试某个 IP 地址和某个域名的连通性，并捕获数据包。

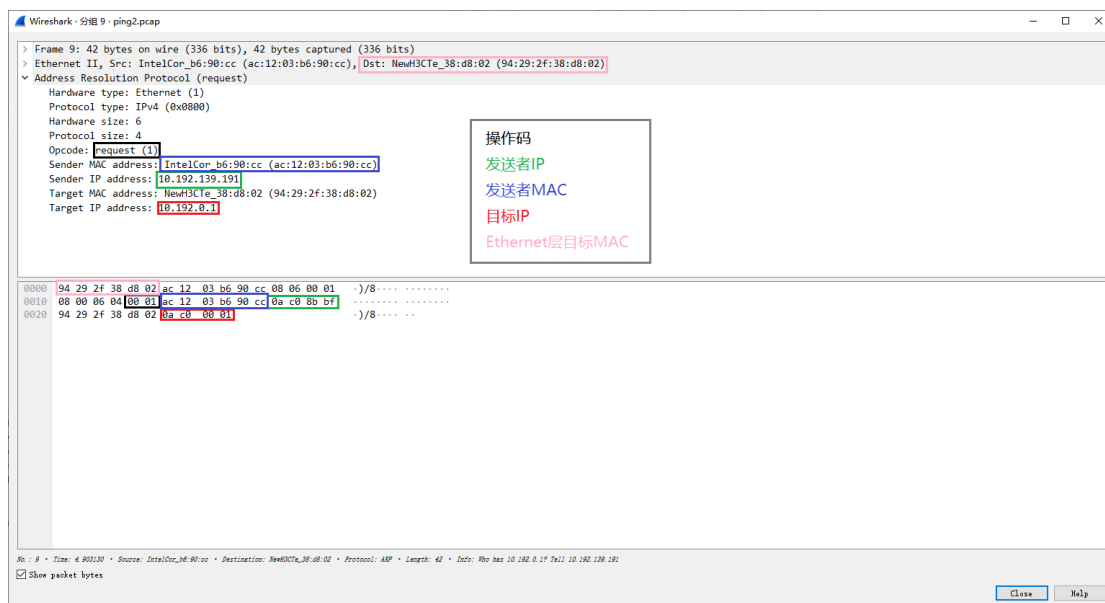
捕获到了哪些相关协议数据包？

Ping IP 地址时: DNS, ICMP

Ping 域名时: PCMP, ARP

ICMP 数据包分别由哪几层协议构成? 链路层 Ethernet II, 网络层 IPv4, ICMP

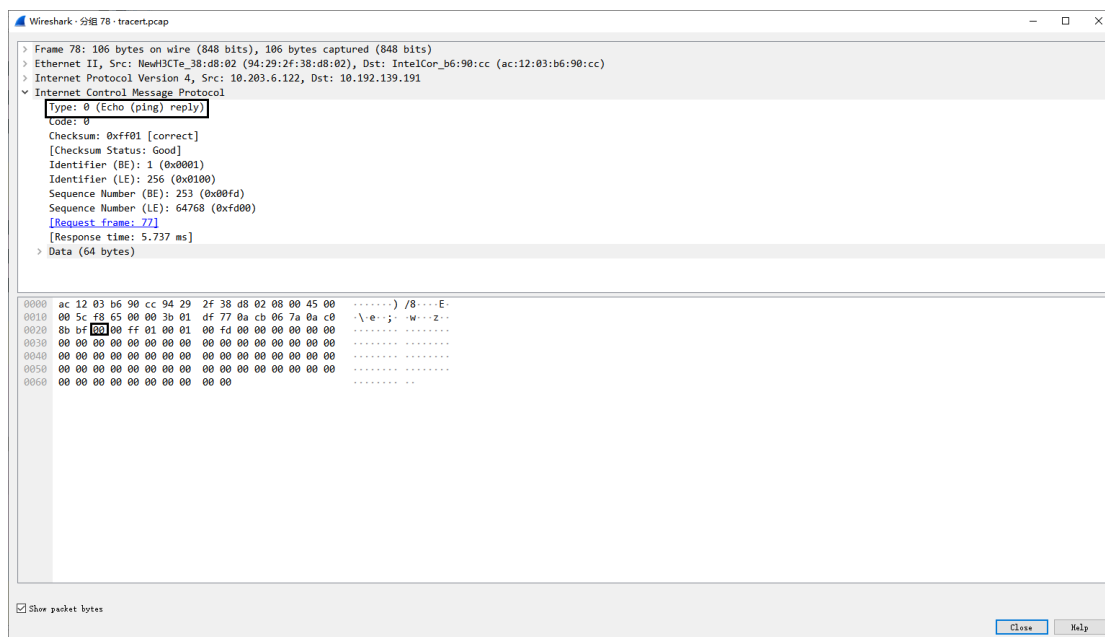
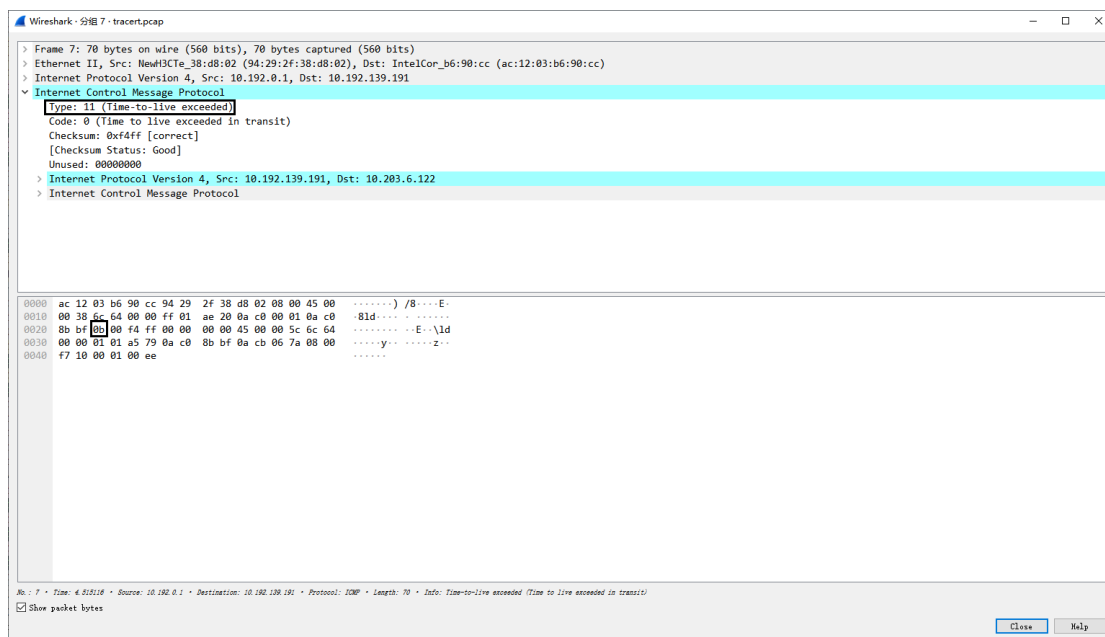
分别选择一个 ARP 请求和响应数据包，展开最高层协议的详细内容，标出操作码、发送者 IP 地址、发送者 MAC 地址、查询的目标 IP 地址、Ethernet 层的目标 MAC 地址以及查询结果。











请在下面的捕获任务完成后，保存 Wireshark 抓包记录（.pcap 格式），随报告一起提交。文件名 **http.pcap**。

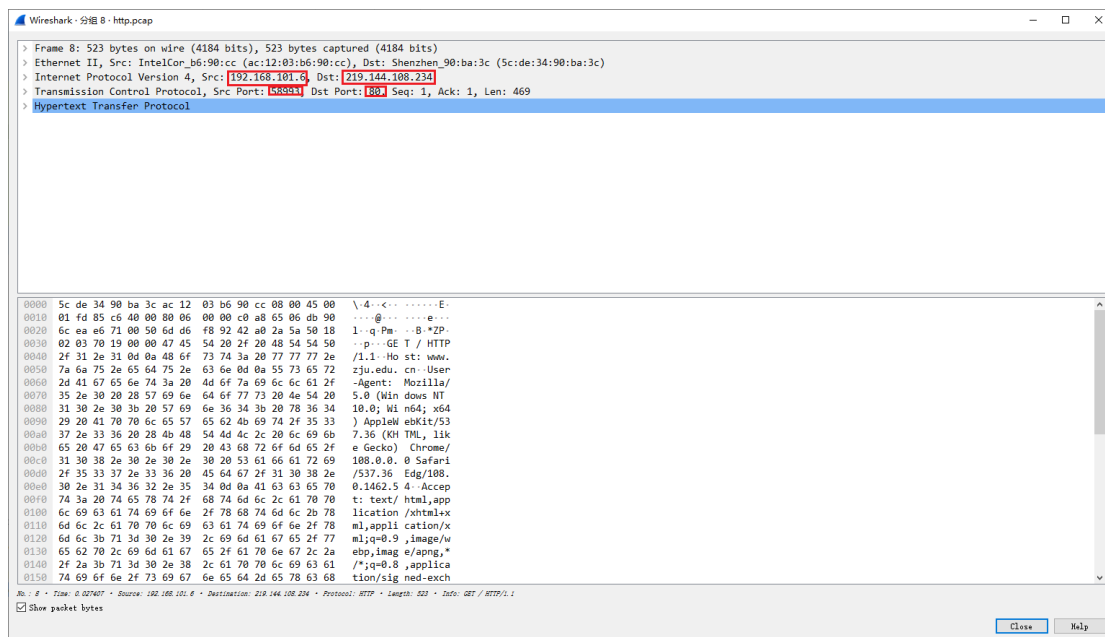
### ✧ Part Three

1. 运行 `ipconfig /flushdns` 命令清空 DNS 缓存，然后打开浏览器，访问 **www.zju.edu.cn**，并使用捕获过滤器只捕获访问该网站的数据（过滤器设置：**tcp port 80 or udp port 53**），网页完全打开后，停止捕获。

捕获到的这些最高层的协议数据包分别由哪几层协议构成？

HTTP: 链路层 Ethernet II, 网络层 IPv4, 传输层 TCP, 应用层 HTTP

截图参考（此处应替换成实际截获的数据）：



2. 为了打开网页，浏览器查询了哪些相关的域名？

域名列表：[www.zju.edu.cn](http://www.zju.edu.cn), [tel.zju.edu.cn](http://tel.zju.edu.cn)

3. 使用显示过滤器 `tcp.stream eq X`，让 X 从 0 开始变化，直到没有数据。分析浏览器为了获取网页数据，总共建立了几个连接？（一个 TCP 流对应一个 TCP 连接）

TCP 连接数： 1

4. 右键点击某个 HTTP 数据包，选择跟踪 TCP 流，可以看到 HTTP 会话的数据。分析浏览器与 WEB 服务器之间进行了几次 HTTP 会话（一对 HTTP 请求和响应对应一次 HTTP 会话）？注意：一个 TCP 流上可能存在多个 HTTP 会话。

HTTP 会话数： 1

5. 选择一个 HTTP 的 TCP 流进行截图，标出请求和响应部分（最好有多个 HTTP 会话的）：

截图示例（此处应替换成实际截获的数据）：

```
GET / HTTP/1.1
Host: www.zju.edu.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.54
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 301 Moved Permanently
Server: Tengine
Content-Type: text/html
Content-Length: 162
Connection: keep-alive
Date: Thu, 22 Dec 2022 13:49:04 GMT
Location: https://www.zju.edu.cn/
X-Frame-Options: SAMEORIGIN
Ali-Swift-Global-Savetime: 1671716944
Via: cache26.12cn2630[48,48,301-0,M], cache40.12cn2630[50,0], cache17.cn3242[66,66,301-0,M], cache7.cn3242[67,0]
X-Cache: MISS TCP_MISS dirn:-2:-2
X-Swift-SaveTime: Thu, 22 Dec 2022 13:49:04 GMT
X-Swift-CacheTime: 0
Timing-Allow-Origin: *
EagleId: db906c9b16717169446673657e

<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

请求

响应

## 六、实验结果分析与思考

- 如果只想捕获某个特定 WEB 服务器 IP 地址相关的 HTTP 数据包，捕获过滤器应该怎么写？

host <ip> and port 80

- Ping 发送的是什么类型的协议数据包？什么情况下会出现 ARP 数据包？ Ping 一个域名和 Ping 一个 IP 地址出现的数据包有什么不同？

a) ICMP.

b) 需要与局域网主机通信时，本机的 ARP 表中没有目标 IP 地址。

c) 前者会需要先解析域名，那之后和 ping ip 的结果一致。

- Tracert/Traceroute 发送的是什么类型的协议数据包，整个路由跟踪过程是如何进行的？

a) tracert 发送的是 ICMP 包，traceroute 发送的是 UDP 包。

b) 从本机发送一个 TTL=1 的 ICMP 包到目标地址，每次在路由器让 TTL-1。如果 TTL=0，路由器向本机发送一个 ICMP 包，告知自身 IP 地址。重复以上步骤，每次让 TTL+1，直到目标地址能够答复则停止。

- 如何理解 TCP 连接和 HTTP 会话？他们之间存在什么关系？

a) TCP 连接是一种可靠连接，用以解决 IP 上的数据包传递。HTTP 会话通过与服务器建立 TCP 连接完成数据的交流。

b) HTTP 是应用层，TCP 是传输层，HTTP 会话建立在 TCP 的基础之上。

- DNS 为什么选择使用 UDP 协议进行传输？而 HTTP 为什么选择使用 TCP 协议？

a) DNS 使用 UDP 是因为 UDP 无需建立连接，速度较快。

b) HTTP 使用可靠性较高的 TCP 可以避免网页中的错误。

## 七、 讨论、心得

在完成本实验后，你可能会有很多待解答的问题，你可以把它们记在这里，接下来的学习中，你也许会逐渐得到答案的，同时也可以让老师了解到你有哪些困惑，老师在课堂可以安排针对性地解惑。等到课程结束后，你再回头看看这些问题时你或许会有不同的见解：

打开网页时，可能得不到 HTTP 数据包

在实验过程中你可能会遇到的困难，并得到了宝贵的经验教训，请把它们记录下来，提供给其他人参考吧：

浏览器需要清理网页缓存

你对本实验安排有哪些更好的建议呢？欢迎献计献策：

无