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：　猜猜

学　　院：　计算机学院

系：　计算机学院

专　　业：　数字媒体技术

学　　号：　猜猜

指导教师：

2019　年　10 月　10 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

提醒：为了避免捕获到大量无关数据包，影响实验观察，建议关闭所有无关软件。实验之前可以提前了解下第六部分有哪些问题。

五、 实验数据记录和处理

以下实验记录均需结合屏幕截图，进行文字标注和描述，图片应大小合适、关键部分清晰可见，可直接在图片上进行标注，也可以单独用文本进行描述。

✧ Part One

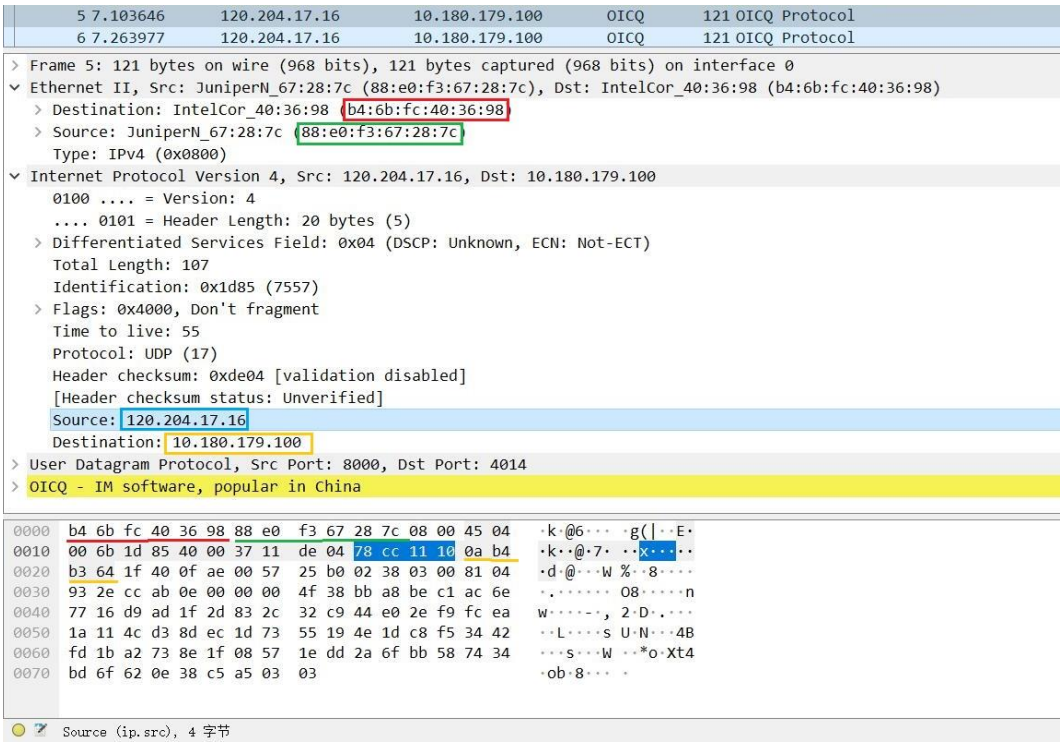
1. **运行 Wireshark 软件，开始捕获数据包，列出你看到的协议名字（至少 5 个）。**

48	40.391928	10.180.177.99	10.180.191.255	NBNS	92	Name query NB WORKGROUP<1d>
49	41.211491	10.180.177.99	10.180.191.255	NBNS	92	Name query NB WORKGROUP<1d>
50	41.866221	10.180.177.99	10.180.191.255	NBNS	92	Name query NB WORKGROUP<1d>
51	41.957422	120.204.17.16	10.180.179.100	OICQ	121	OICQ Protocol
52	43.346563	120.204.17.16	10.180.179.100	OICQ	121	OICQ Protocol
53	44.126942	IntelCor_40:36:98	JuniperN_67:28:7c	ARP	42	Who has 10.180.176.1? Tell 10.180.179.100
54	44.129195	JuniperN_67:28:7c	IntelCor_40:36:98	ARP	56	10.180.176.1 is at 88:e0:f3:67:28:7c
55	44.812919	10.180.80.1	255.255.255.255	DHCP	293	DHCP NAK - Transaction ID 0x2cefcfa5
56	45.368454	120.204.17.16	10.180.179.100	OICQ	121	OICQ Protocol
57	45.409843	120.204.17.16	10.180.179.100	OICQ	121	OICQ Protocol
58	47.095915	120.204.17.16	10.180.179.100	OICQ	121	OICQ Protocol
59	49.408007	120.204.17.16	10.180.179.100	OICQ	121	OICQ Protocol
60	52.764136	10.180.179.100	40.90.189.152	TLSv1.2	97	Application Data
61	52.771989	10.180.179.100	10.10.0.21	DNS	89	Standard query 0x46d6 A v10.events.data.microsoft.com

协议名：OICQ NBNS TCP TLSv1.2 ARP DHCP

2. **找一个包含 IP 的数据包，这个数据包有 5 层？最高层协议是 OICQ，从 Ethernet 开始往上，各层协议的名字分别为：Internet Protocol, User Datagram Protocol, OICQ。**

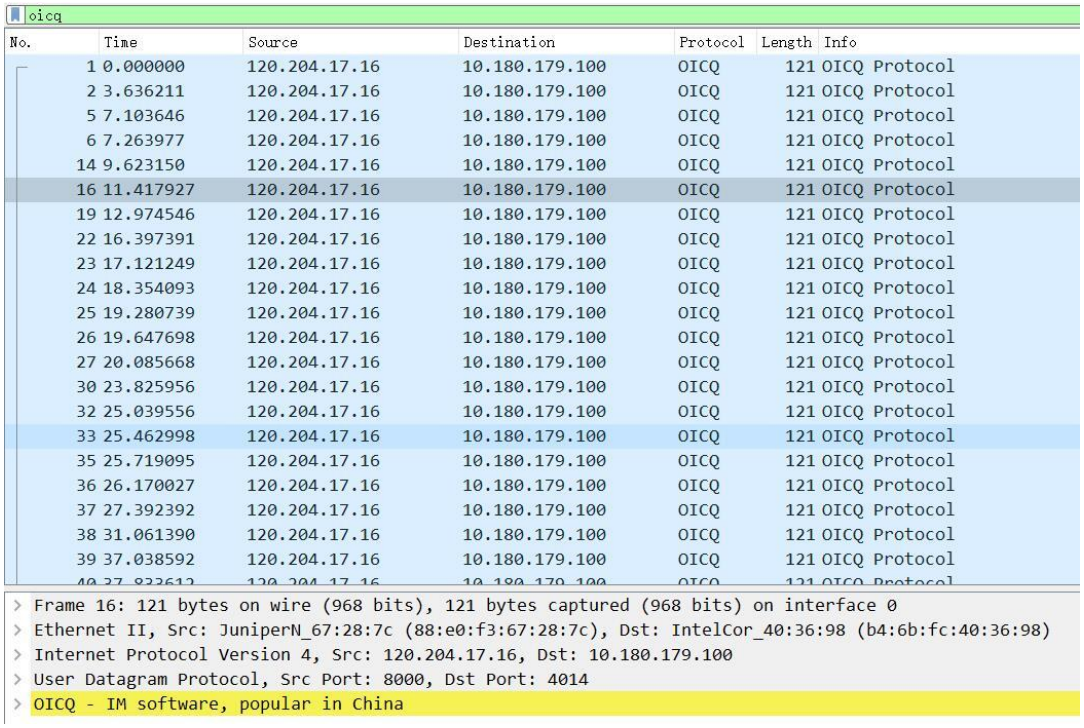
展开 IP 层协议，标出源 IP 地址、目标 IP 地址及其在数据包中的具体位置，展开 Ethernet 层，标出源 MAC 地址和目标 MAC 地址及其在数据包中的具体位置。



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 。

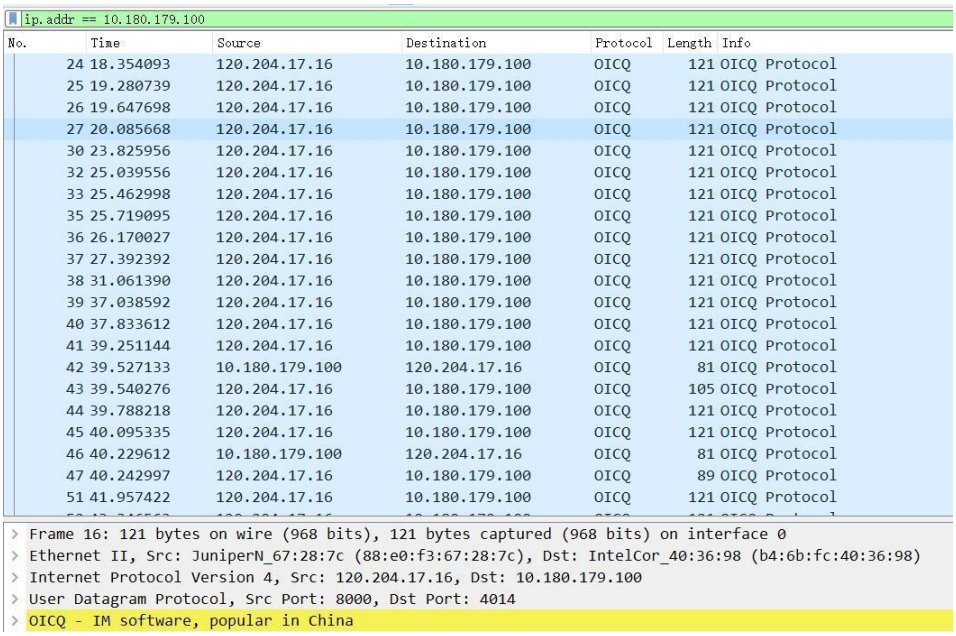
截图：



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 == 10.180.179.100，希望显示的 IP 地址：10.180.179.100。

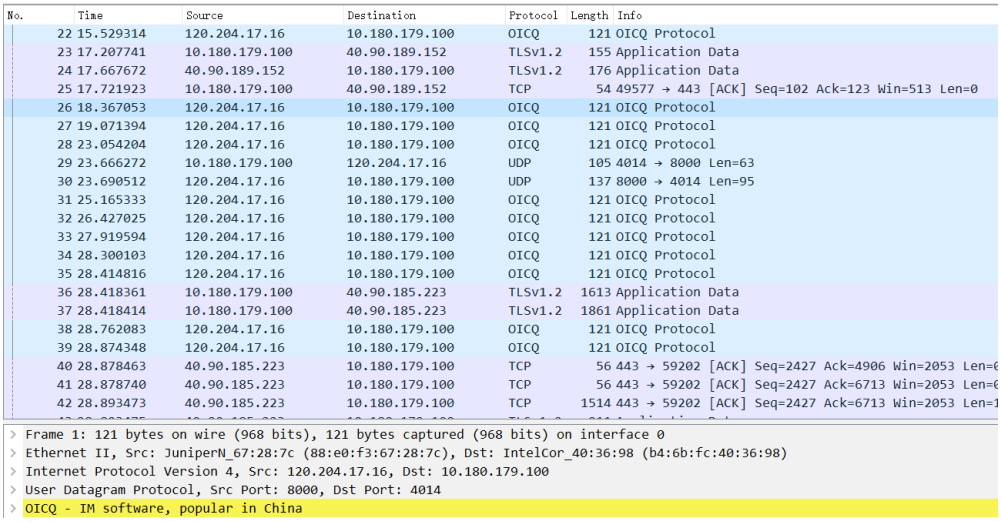
截图：



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 10.180.179.100，希望捕获的 IP 地址：10.180.179.100。

截图：



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：

tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
157	63.542375	10.180.179.100	183.192.196.205	TCP	66	59220 → 80 [SYN] Seq=0 Win=64240 Le
158	63.556862	183.192.196.205	10.180.179.100	TCP	66	80 → 59220 [SYN, ACK] Seq=0 Ack=1 W
159	63.557011	10.180.179.100	183.192.196.205	TCP	54	59220 → 80 [ACK] Seq=1 Ack=1 Win=66
160	63.571820	10.180.179.100	183.192.196.205	TCP	533	59220 → 80 [PSH, ACK] Seq=1 Ack=1 W
161	63.573478	10.180.179.100	183.192.196.205	TCP	10294	59220 → 80 [PSH, ACK] Seq=480 Ack=1
162	63.585020	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=480 Win=
163	63.587318	10.180.179.100	183.192.196.205	TCP	4374	59220 → 80 [ACK] Seq=10720 Ack=1 Wi
164	63.589671	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=1920 Win
165	63.589723	10.180.179.100	183.192.196.205	TCP	2934	59220 → 80 [ACK] Seq=15040 Ack=1 Wi
166	63.589998	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=3360 Win
167	63.589998	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=4800 Win
168	63.589998	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=6240 Win
169	63.589998	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=7680 Win
170	63.589998	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=9120 Win
171	63.589998	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=10720 Wi
172	63.590011	10.180.179.100	183.192.196.205	TCP	3094	59220 → 80 [PSH, ACK] Seq=17920 Ack
173	63.600961	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=12160 Wi
174	63.601849	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=13600 Wi
175	63.601849	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=15040 Wi
176	63.602916	10.180.179.100	183.192.196.205	TCP	10294	59220 → 80 [PSH, ACK] Seq=20960 Ack
177	63.606140	183.192.196.205	10.180.179.100	TCP	56	80 → 59220 [ACK] Seq=1 Ack=16480 Wi

请在下面的每次捕获任务完成后，保存 Wireshark 抓包记录（.pcap 格式），随报告一起提交。每一个任务一个单独文件（如 dns.pcap、ping.pcap、tracert.pcap）。

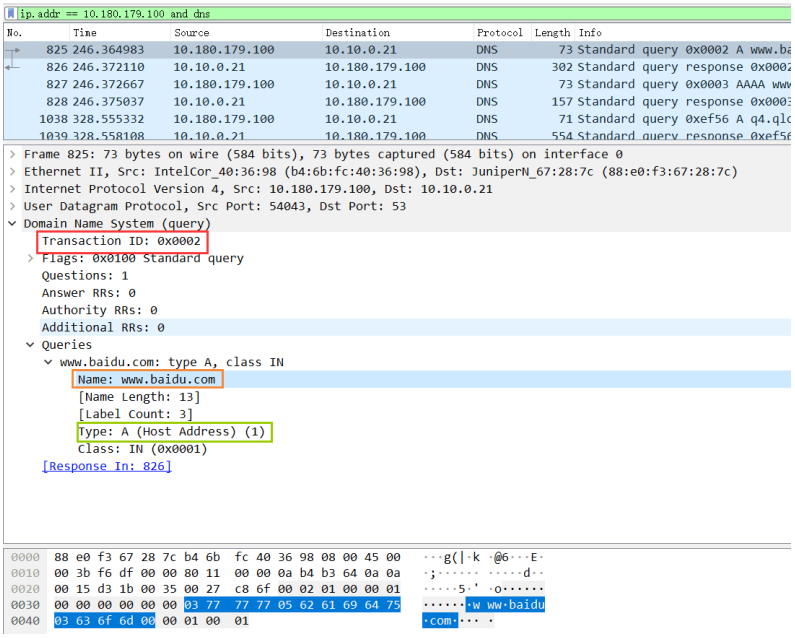
✧ Part Two

任务 1：使用 nslookup 命令，查询某个域名，并捕获这次的数据包。DNS 数据包由哪几层协议构成？ Frame, Ethernet, Internet Protocol Version 4, User Datagram Protocol, Domain Name System 。 使用的服务方端口是： 53 。

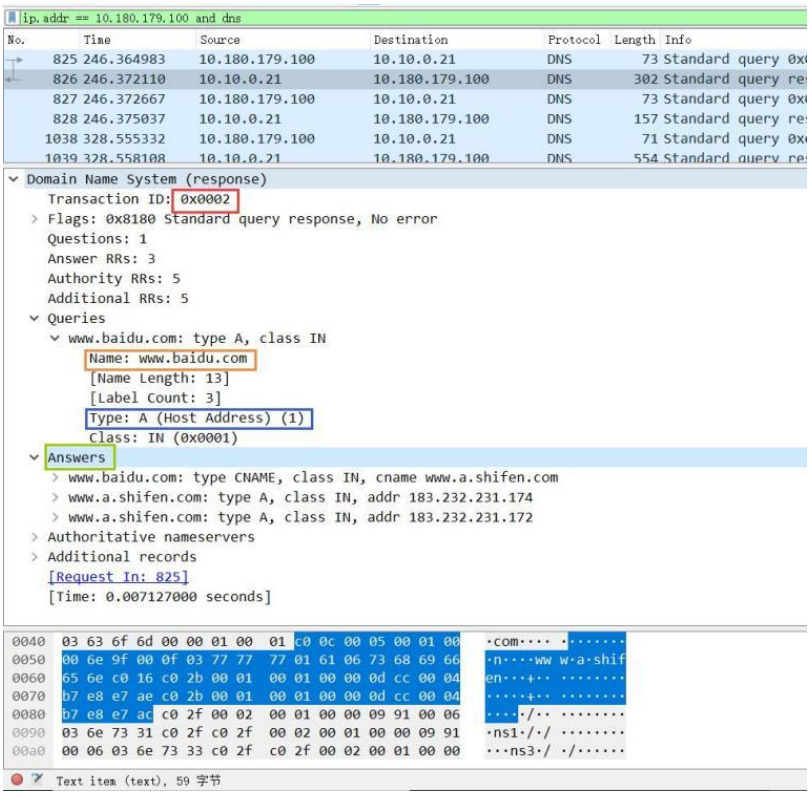
```
C:\Users\18367>nslookup www.baidu.com
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

非权威应答:
名称:    www.a.shifen.com
Addresses:  183.232.231.174
          183.232.231.172
Aliases:  www.baidu.com
```

分别选择一个请求包和一个响应包，展开最高层协议的详细内容，标出交易 ID、查询类型、查询的域名内容以及查询结果。

请求包：



响应包：



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：ICMP

Ping 域名时：ICMP，ARP

ICMP 数据包分别由哪几层协议构成？Frame, Ethernet, Internet Protocol Version

4, Internet Control Message Protocol

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。

请求包：

```
> Frame 1922: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
> Ethernet II, Src: Shenzhen_ed:03:80 (d4:5f:25:ed:03:80), Dst: IntelCor_8c:db:a6 (f4:96:34:8c:db:a6)
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Shenzhen_ed:03:80 (d4:5f:25:ed:03:80)
    Sender IP address: 192.168.123.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.123.96
```

```
0000  f4 96 34 8c db a6 d4 5f  25 ed 03 80 08 06 00 01   ··4····_ %······ 
0010  08 00 06 04 00 01 d4 5f  25 ed 03 80 c0 a8 7b 01   ······ %·····{· 
0020  00 00 00 00 00 00 c0 a8  7b 60                     ······ {`
```

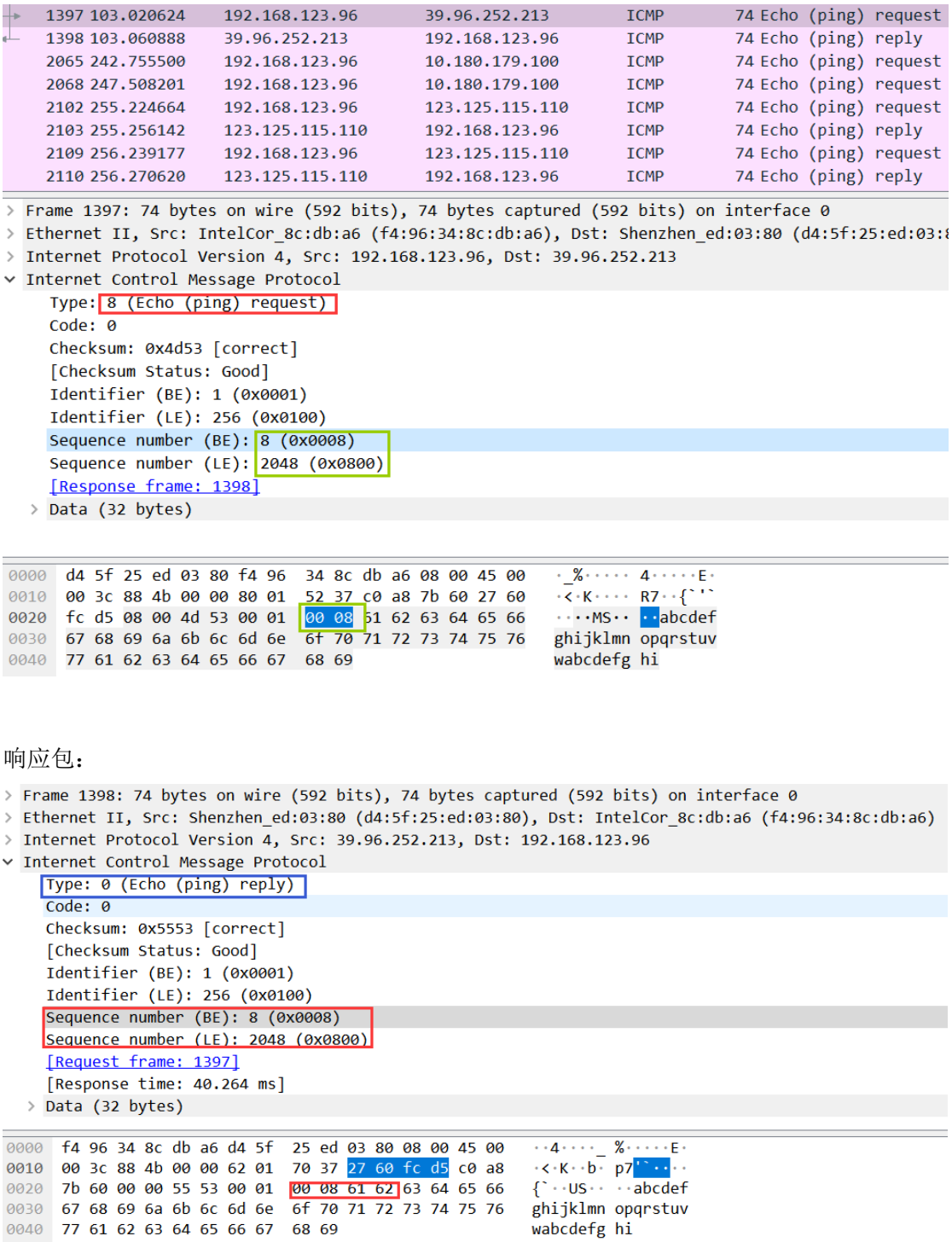
响应包：

```
> Frame 1923: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
> Ethernet II, Src: IntelCor_8c:db:a6 (f4:96:34:8c:db:a6), Dst: Shenzhen_ed:03:80 (d4:5f:25:ed:03:80)
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: IntelCor_8c:db:a6 (f4:96:34:8c:db:a6)
    Sender IP address: 192.168.123.96
    Target MAC address: Shenzhen_ed:03:80 (d4:5f:25:ed:03:80)
    Target IP address: 192.168.123.1
```

```
0000  d4 5f 25 ed 03 80 f4 96  34 8c db a6 08 06 00 01   ·_%····· 4······ 
0010  08 00 06 04 00 02 f4 96  34 8c db a6 c0 a8 7b 60   ········ 4·····{` 
0020  d4 5f 25 ed 03 80 c0 a8  7b 01                     ·_%····· {·
```

分别选择一个 ICMP 请求和响应数据包，展开最高层协议的详细内容，标出类型、序号。

请求包：

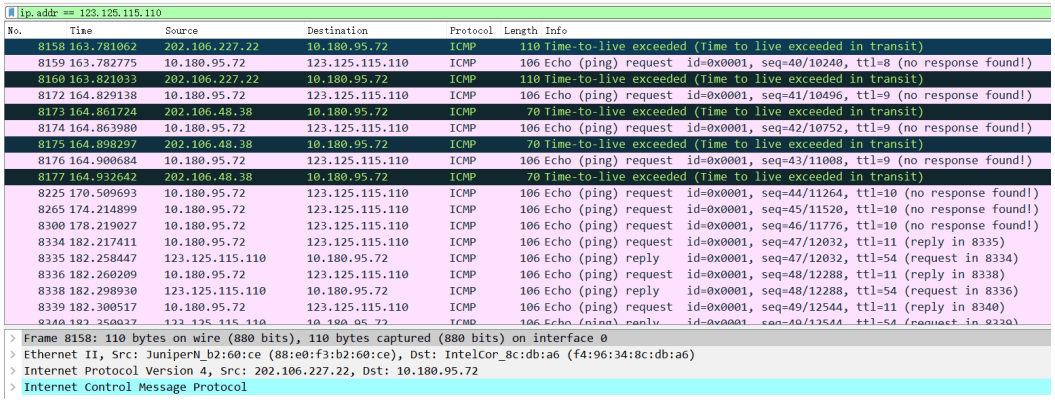


响应包：

任务 3：使用 Tracert 命令（Mac 下使用 Traceroute 命令），跟踪某个外部 IP 地址的路由，并捕获这次的数据包。跟踪路由使用的数据包协议类型是：_____ICMP_____，数据包由几层协议构成？____四层 Frame, Ethernet, Internet Protocol Version 4, Internet Control Message Protocol_____。

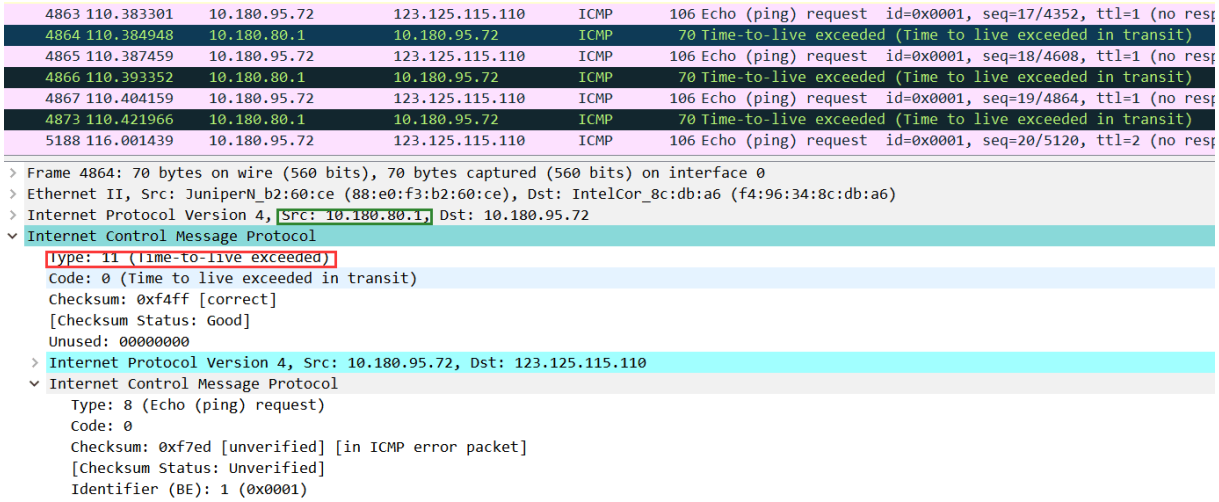
观察并记录请求包中 IP 协议层的 TTL 字段变化规律，第一个请求的 TTL 等于__1__，同样 TTL 的请求连续发送了__3__个，然后每次 TTL 增加了__1__，最后一个请求的 TTL

等于 **11** 。附上截图：

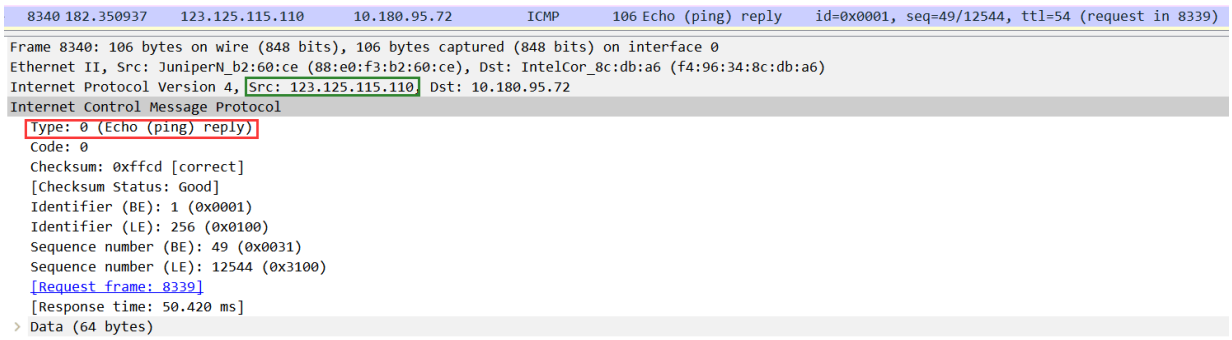


观察并记录响应包的信息，第一组响应包的发送者 IP 是：**10.180.80.1**，标记 ICMP 层的类型字段。最后一组响应包的发送者 IP 是：**123.125.115.110**，标记 ICMP 层的类型字段。附上截图：

第一组：



最后一组：



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。

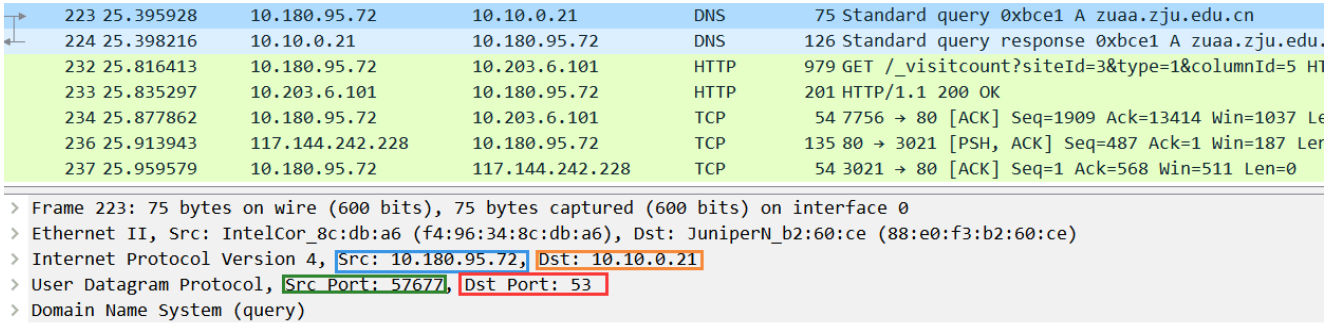
   捕获到的这些最高层的协议数据包分别由哪几层协议构成？

DNS：**Frame, Ethernet, Internet Protocol Version 4, User Datagram, Protocol, Domain Name System**

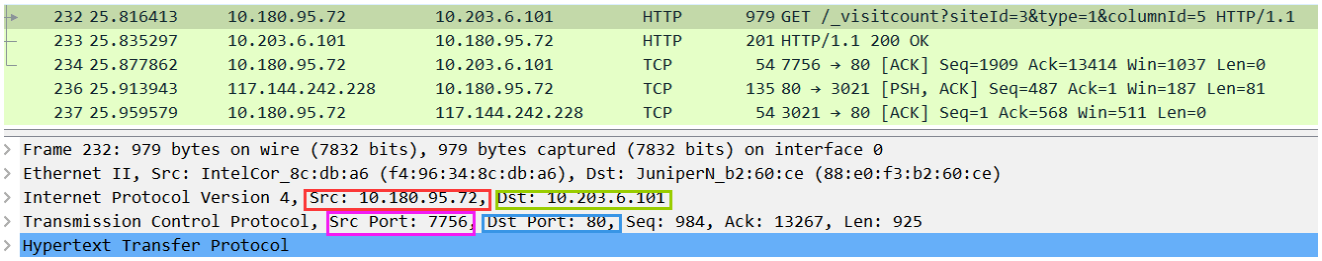
HTTP: **Frame, Ethernet, Internet Protocol Version 4,Transmission Protocol, Hypertext Transfer Protocol**

每种协议选取一个代表展开后截图，并标出源和目标 IP 地址、源和目标端口）

DNS:



HTTP:



2. 为了打开网页，浏览器查询了哪些相关的域名？

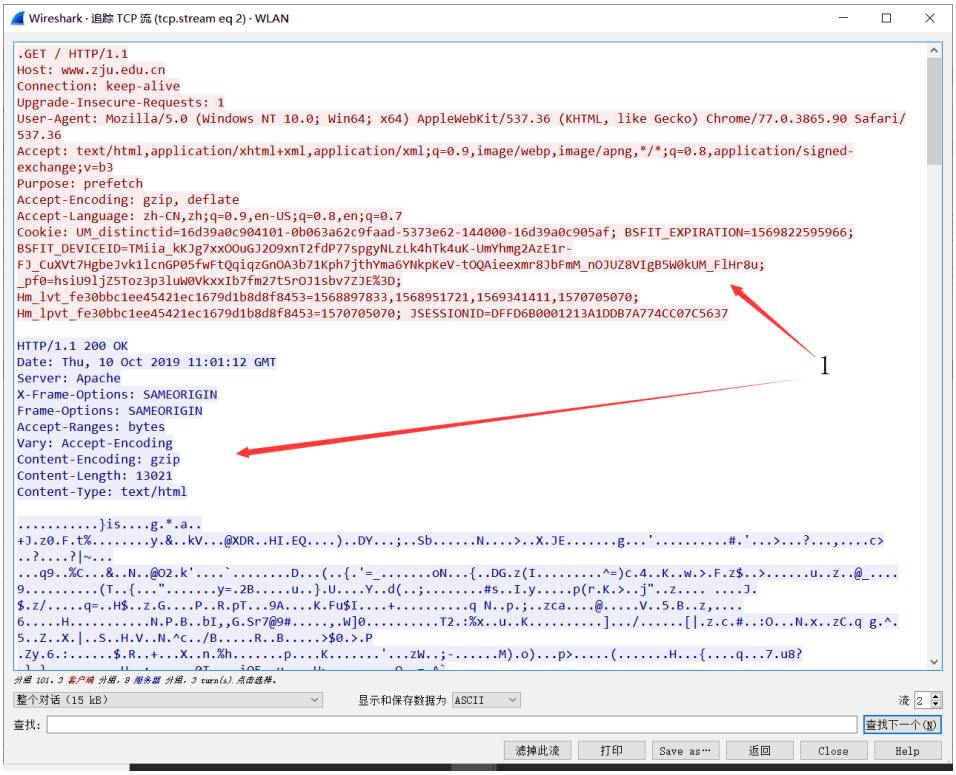
   域名列表：bksy.zju.edu,cn, culture.zju.edu.cn, grs.zju.edu.cn, iczu.zju.edu.cn, map.zju.edu.cn, my.zju.edu.cn, person.zju.edu.cn, ocw.zju.edu.cn, piclib.zju.edu.cn, rd.zju.edu.cn, rwsk.zju.edu.cn, ugrs.zju.edu.cn……

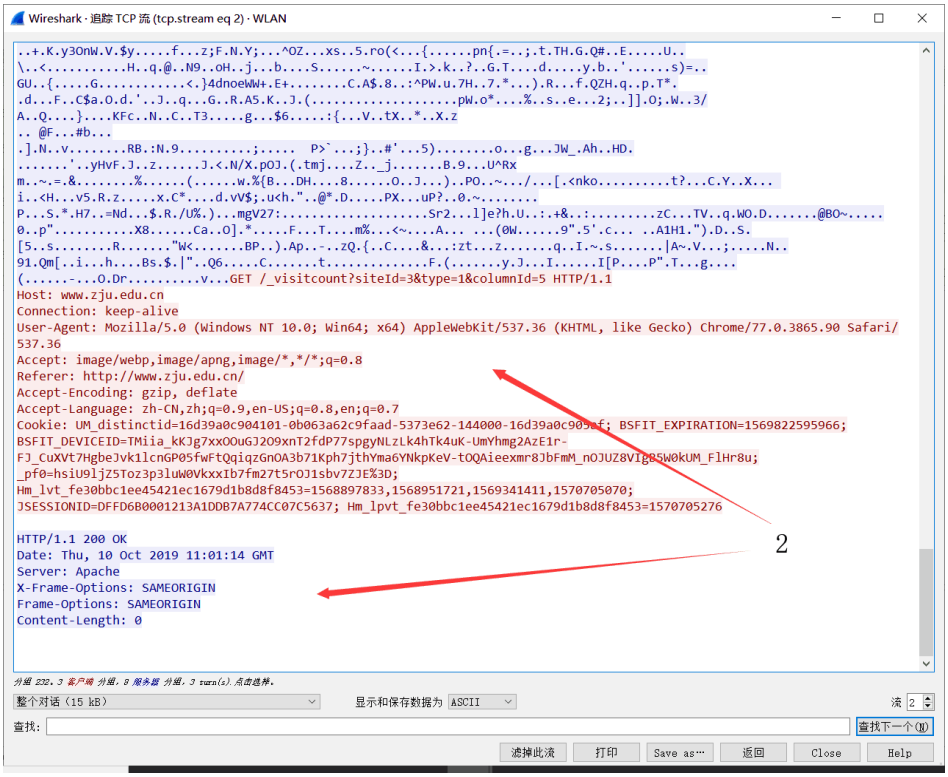
3. 使用显示过滤器 **tcp.stream eq X**，让 **X** 从 **0** 开始变化，直到没有数据。分析浏览器为了获取网页数据，总共建立了几个连接？（一个 **TCP** 流对应一个 **TCP** 连接）

TCP 连接数：　　10　　

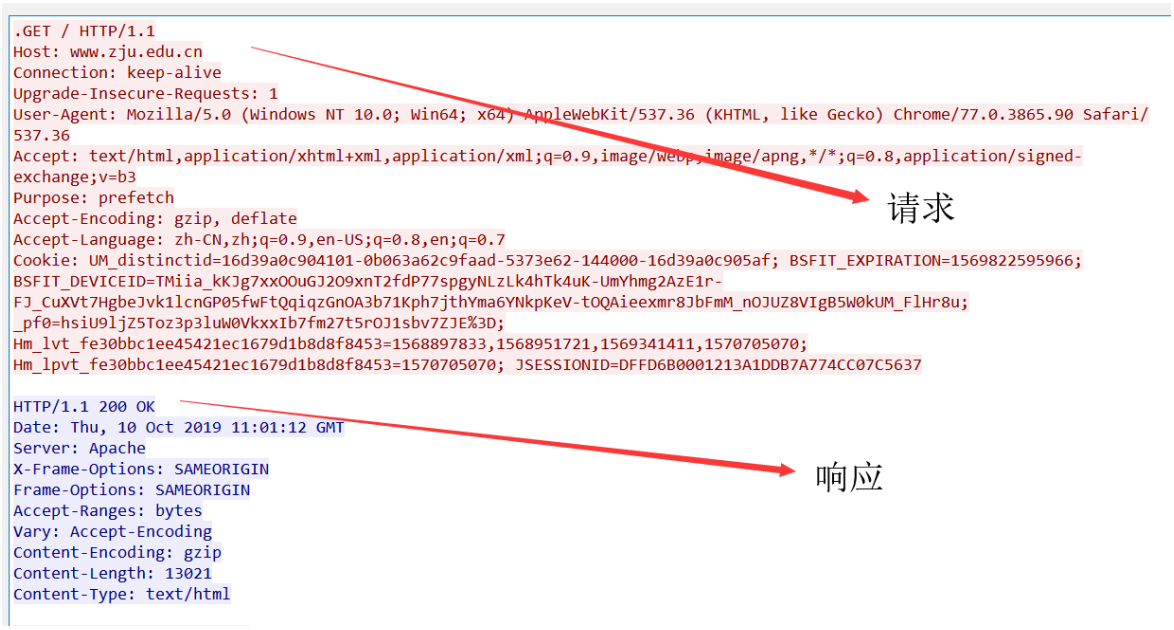
4. **右键点击某个 HTTP 数据包，选择跟踪 TCP 流，可以看到 HTTP 会话的数据。分析浏览器与 WEB 服务器之间进行了几次 HTTP 会话（一对 HTTP 请求和响应对应一次 HTTP 会话）？注意：一个 TCP 流上可能存在多个 HTTP 会话。**

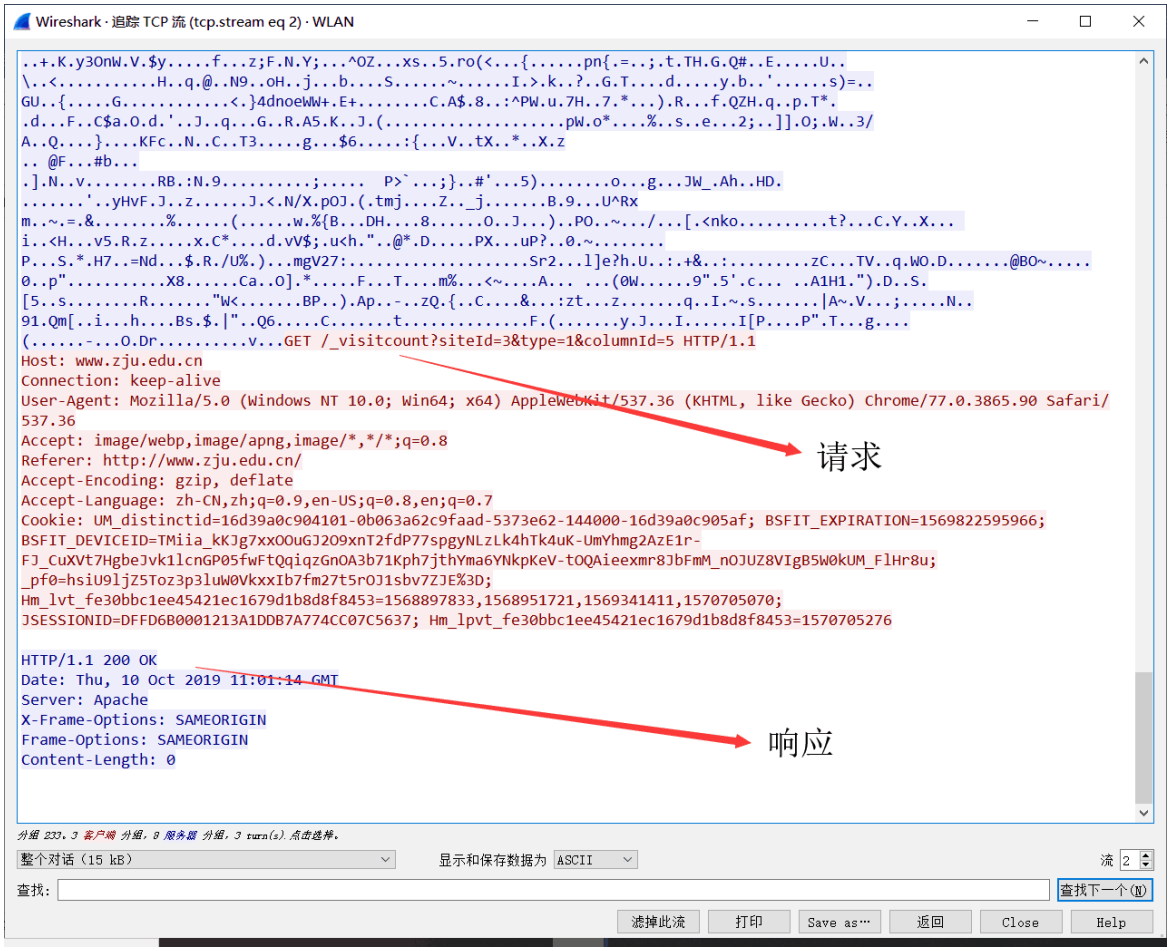
HTTP 会话数：　　2　　





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）：





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？

  ip.addr ==该 WEB 服务器的 ip 地址 and http

  如：ip.addr == 123.125.115.110 and http

- 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？

  1. Ping 发送的是 ICMP 协议数据包。

  2.当 Ping 一个计算机内无缓存的域名时会出现 ARP 消息进行域名解析：当主

机 A 想要同本局域网上的某个主机 B 发送 IP 数据包时，若其 ARP 高速缓存中无主机 B 的 IP 地址，就会向外发送一个 ARP 广播包；交换机收到这个 ARP 后，检索自己有没有保存主机 B 的 MAC 地址，如果没有，就会向所有端口发送 ARP 广播，主机 B 收到报文后立即响应，并按同样的 ARP 报文格式返回给主机 A。

3. Ping 一个域名和 Ping 一个 IP 地址出现的数据包的不同之处在于： 当前计算机内不存在所 Ping 域名对应的 IP 地址时会先进行域名解析。

- 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？

  1.Tracert/Traceroute 发送的是 ICMP 类型的协议数据包。

  2.路由追踪过程如下：

  ① 构造 UDP 数据包，设置 TTL = 1

  ② 发送 UDP 数据包，记录发送时间 t1

  ③ 接收 ICMP 差错包，如果是超时报文，则是经过的中间路由，记录路由信息，记录接收时间 t2，计算时间(t2 - t1)；如果是目标不可达报文，则抵达目的主机，记录势头收时间 t2，打印信息，退出

  ④ 构造 UDP 数据包，设置 TTL += 1，返回第二步

  其中，TTL 的每个数值(如 TTL = 1)发送 3 次 UDP 包，即重复 2~3 步 3 次；接收超时，打印"*"表示报文丢失

- 如何理解 TCP 连接和 HTTP 会话？他们之间存在什么关系？

  HTTP 的长连接和短连接本质上是 TCP 长连接和短连接。HTTP 属于应用层协议，在传输层使用 TCP 协议，在网络层使用 IP 协议。IP 协议主要解决网络路由和寻址问题，TCP 协议主要解决如何在 IP 层之上可靠的传递数据包，使在网络上的另一端收到发端发出的所有包，并且顺序与发出顺序一致。TCP 有可靠，面向连接的特点。

- 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？

TCP 为可靠的面向连接协议，建立 TCP 连接需要两端三次握手协商（双方在线）。 UDP 为不可靠的无连接协议，数据传输出去，无需连接两端，确认是否到达（对端可不在线）。基于 UDP 的 DNS 协议只要一个请求、一个应答，而使用基于 TCP 的 DNS 协议要三次握手、发送数据以及应答、四次挥手 DNS 。故 DNS 选择使用 UDP 传输时效率高，域名解析的时间少，采用 TCP 协议消耗时间长。HTTP 使用 TCP 传输相较于 UDP 传输出错几率小，安全性高。

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：

我们在实验中抓取到了各种各样的数据包，但其中有许多协议并没有了解过，比如 NBNS, TLSv1.2, SSDP 等。另外，抓到的数据包中包含许多的信息，我们能看懂的有源 IP 地址、目标 IP 地址、类型等等，但其中也包含许多我们不明其意的字段，比如 TCP 数据包中的 SEQ/ACK analysis 等。

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：

本次实验在个人 PC 上进行，由于电脑没有网线接口，我们直接在 WLAN 连接状态下开始实验，但这样的连接似乎不够稳定，实验中常常有奇怪的错误发生。

最初，我们打开 Wireshark 时显示找不到接口，经过查询相关资料，发现是由于其自带的 Winpcap 不支持 Win10，重新在官网下载安装了 Winpcap 之后问题得到了解决。

在 Part One 中，我们设置过滤器为 host ipaddress 时，Wireshark 一直报错，不知道是出于什么原因。最后我们退出重启，在开始界面就输入捕获过滤器，捕获成功。

在进行使用 Ping 命令的任务二时，我们最初只能抓到 ICMP 协议的数据包，并不能抓到 ARP 数据包。做到后来的问题回答部分，才发现这是因为我们 Ping 的域名都已经有过缓存，所以不需要进行域名解析，也就没有 ARP 包了。

你对本实验安排有哪些更好的建议呢？欢迎献计献策：

实验安排的比较好，因为实验七已经有了一些 Wireshark 基础，使得这次实验做起来比较容易上手。