

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：	刘轩铭
学 院：	计算机学院
系：	计算机系
专 业：	软件工程
学 号：	3180106071
指导教师：	邱劲松

2020 年 9 月 20 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: ARP (Address Resolution Protocol), UDP (User Datagram Protocol),
TCP (Transmission Control Protocol), TLS (Transport Layer Security), HTTP
(HyperText Transfer Protocol)

部分截图如下：

[illegible]

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 Data，从 Ethernet 开始往上，各层协议的名字分别为：Internet Protocol, User Datagram Protocol, Data。

截图如下：



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。

截图如下:



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: arp , 希望显示的协议类型: ARP。

截图如下：

[illegible]

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 == 36.152.44.96 , 希望显示的 IP 地址: 36.152.44.96。

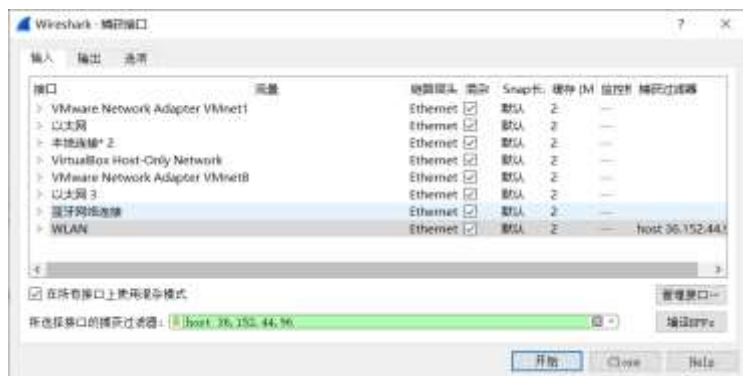
截图如下：

ts	Time	Source	Destination	Protocol	Length	Info
1	8.000000	10.181.217.115	36.152.44.96	TCP	554096	→ 554096 → 443 [ACK] Seq=1 A/R=0 Win=132 Len=1 [TCP segment of a reassembled PDU]
2	8.000000	36.152.44.96	10.181.217.113	TCP	66443	→ 54096 → 54096 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
3	8.000000	10.181.217.115	36.152.44.96	TCP	554096	→ 54096 → 443 [ACK] Seq=1 A/R=0 Win=132 Len=1 [TCP segment of a reassembled PDU]
4	8.000000	36.152.44.96	10.181.217.113	TCP	66443	→ 54096 → 54096 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
5	27.330000	10.181.217.115	36.152.44.96	TCP	554096	→ 54096 → 443 [ACK] Seq=1 A/R=0 Win=132 Len=1 [TCP segment of a reassembled PDU]
6	27.330000	36.152.44.96	10.181.217.113	TCP	66443	→ 54096 → 54096 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
7	27.330000	10.181.217.115	36.152.44.96	TCP	554096	→ 54096 → 443 [ACK] Seq=1 A/R=0 Win=132 Len=1 [TCP segment of a reassembled PDU]
8	27.330000	36.152.44.96	10.181.217.113	TCP	66443	→ 54096 → 54096 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
9	28.000000	10.181.217.115	36.152.44.96	TCP	554096	→ 54096 → 443 [ACK] Seq=1 A/R=0 Win=132 Len=1 [TCP segment of a reassembled PDU]
10	28.000000	36.152.44.96	10.181.217.113	TCP	66443	→ 54096 → 54096 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
11	32.320000	10.181.217.115	36.152.44.96	TCP	55443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
12	32.320000	36.152.44.96	10.181.217.113	TCP	66443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
13	32.320000	10.181.217.115	36.152.44.96	TCP	2282	Ignored (no window extension)
14	32.320000	36.152.44.96	10.181.217.113	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
15	32.320000	10.181.217.115	36.152.44.96	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
16	32.320000	36.152.44.96	10.181.217.113	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
17	32.320000	10.181.217.115	36.152.44.96	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
18	32.320000	36.152.44.96	10.181.217.113	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
19	32.320000	10.181.217.115	36.152.44.96	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
20	32.320000	36.152.44.96	10.181.217.113	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
21	32.320000	10.181.217.115	36.152.44.96	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
22	32.320000	36.152.44.96	10.181.217.113	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
23	32.320000	10.181.217.115	36.152.44.96	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
24	32.320000	36.152.44.96	10.181.217.113	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
25	32.320000	10.181.217.115	36.152.44.96	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
26	32.320000	36.152.44.96	10.181.217.113	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
27	32.320000	10.181.217.115	36.152.44.96	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
28	32.320000	36.152.44.96	10.181.217.113	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
29	32.320000	10.181.217.115	36.152.44.96	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
30	32.320000	36.152.44.96	10.181.217.113	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0
31	32.320000	10.181.217.115	36.152.44.96	TCP	90443	→ 54044 → 54044 [ACK] Seq=1 A/R=0 Win=1384 Len=0 Win=1 Win=0

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 36.152.44.96 , 希望捕获的 IP 地址: 36.152.44.96。

截图如下：

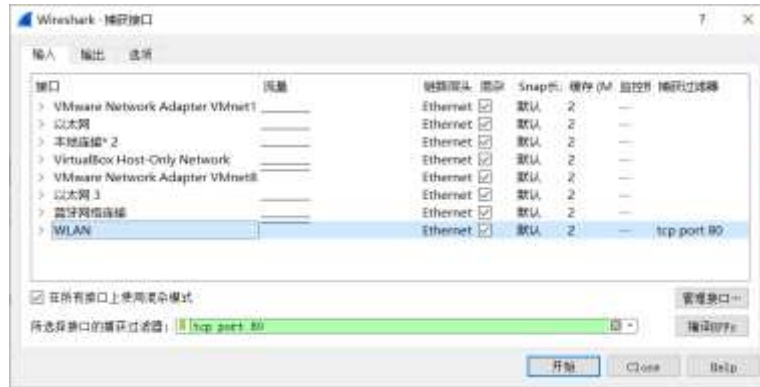


No.	Time	Source	Destination	Protocol	Length	Info.
1	0.000000	10.0.0.1:217.215	0.0.0.0:42.00	TCP	60	51821 → 423 [RST] Seq=0 Win=0 Len=0 Win=0 Len=0 RST=0 Seq=0
2	0.000010	10.0.0.1:217.215	0.0.0.0:42.00	TCP	60	51821 → 443 [RST] Seq=0 Win=0 Len=0 Win=0 Len=0 RST=0 Seq=0
3	0.000012	10.0.0.1:217.215	10.0.0.1:217.112	TCP	60	443 → 51821 [RST] Seq=0 Win=0 Len=0 Len=0 RST=0 Seq=0
4	0.000024	10.0.0.1:217.215	0.0.0.0:42.00	TCP	60	51821 → 443 [ACK] Seq=1 Win=0 Len=0 Len=0
5	0.000090	10.0.0.1:217.215	0.0.0.0:42.00	TCP	60	51821 → 443 [ACK] Seq=1 Win=0 Len=0 Len=0
6	0.001170	10.0.0.1:217.215	10.0.0.1:217.112	TCP	60	443 → 51821 [RST] Seq=0 Win=0 Len=0 Len=0 RST=0 Seq=0
7	0.001303	10.0.0.1:217.215	0.0.0.0:42.00	TCP	60	51821 → 443 [ACK] Seq=1 Win=0 Len=0 Len=0
8	0.001601	10.0.0.1:217.113	0.0.0.0:42.00	TCP	60	51821 → 443 [ACK] Seq=1 Win=0 Len=0 Len=0
9	0.002001	0.0.0.0:42.00	10.0.0.1:217.113	TCP	60	443 → 51821 [ACK] Seq=1 Win=0 Len=0 Len=0
10	0.002157	0.0.0.0:42.00	10.0.0.1:217.113	TCP	60	443 → 51821 [ACK] Seq=1 Win=0 Len=0 Len=0
11	0.002157	0.0.0.0:42.00	10.0.0.1:217.113	TCP	60	443 → 51821 [ACK] Seq=1 Win=0 Len=0 Len=0
12	0.002519	0.0.0.0:42.00	10.0.0.1:217.113	TCP	60	443 → 51821 [ACK] Seq=1 Win=0 Len=0 Len=0
13	0.002519	0.0.0.0:42.00	10.0.0.1:217.113	TCP	60	443 → 51821 [ACK] Seq=1 Win=0 Len=0 Len=0
14	0.002760	10.0.0.1:217.215	0.0.0.0:42.00	TCP	60	51821 → 443 [ACK] Seq=1 Win=0 Len=0 Len=0

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。

截图如下：



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.181.217.115	110.43.81.33	TCP	60	55523 → 80 [SYN] Seq=0 Win=0 Len=0 MSS=1460 Window=0
2	0.000000	110.43.81.33	10.181.217.115	TCP	60	80 → 55523 [SYN, ACK] Seq=0 Ack=1 Win=0 Len=0 MSS=1460 Window=0
3	0.000000	10.181.217.115	110.43.81.33	TCP	60	55523 → 80 [ACK] Seq=1 Ack=1 Win=12800 Len=0
4	0.000000	10.181.217.115	110.43.81.33	HTTP	287	GET /query?2550001 HTTP/1.1 (application/x-www-form-urlencoded)
5	0.000000	110.43.81.33	10.181.217.115	TCP	60	80 → 55523 [ACK] Seq=1 Ack=234 Win=1460 Len=0
6	0.000000	110.43.81.33	10.181.217.115	TCP	60	55523 → 80 [FIN, ACK] Seq=234 Ack=233 Win=12800 Len=0
7	0.000000	110.43.81.33	10.181.217.115	TCP	60	80 → 55523 [FIN, ACK] Seq=235 Ack=233 Win=1460 Len=0
8	0.000000	10.181.217.115	110.43.81.33	TCP	60	55523 → 80 [ACK] Seq=235 Ack=235 Win=12800 Len=0
9	0.000000	10.181.217.115	111.48.110.196	TCP	60	55523 → 80 [FIN, ACK] Seq=235 Ack=1 Win=1460 Len=0
10	0.000000	10.181.217.115	111.48.110.196	TCP	60	80 → 55523 [FIN, ACK] Seq=1 Ack=1 Win=1460 Len=0
11	0.000000	10.181.217.115	111.48.110.196	TCP	60	55523 → 80 [FIN, ACK] Seq=235 Ack=1 Win=1460 Len=0
12	0.000000	10.181.217.115	111.48.110.196	TCP	60	80 → 55523 [FIN, ACK] Seq=1 Ack=1 Win=1460 Len=0
13	0.000000	10.181.217.115	111.48.110.196	TCP	60	55523 → 80 [FIN, ACK] Seq=235 Ack=1 Win=1460 Len=0
14	0.000000	111.48.110.196	10.181.217.115	TCP	60	80 → 55523 [SYN] Seq=1 Win=0 Len=0
15	0.000000	111.48.110.196	10.181.217.115	TCP	60	55523 → 80 [SYN, ACK] Seq=1 Ack=1 Win=1460 Len=0

No.	Time	Source	Destination	Protocol	Length	Info
10	0.905908	10.101.217.115	10.10.0.21	DNS	82	Standard query 000001 PTR 21.0.10.10.in-addr.arpa
11	0.908111	10.10.0.21	10.101.217.115	DNS	342	Standard query response 000001 PTR 21.0.10.10.in-addr.arpa PTR druid.rja.edu.cn 10 druid.rja.edu.cn 0 10 10 0 0
12	0.909107	10.101.217.115	10.10.0.21	DNS	80	Standard query 000002 A ba5da.com
13	0.971695	10.10.0.21	10.101.217.115	DNS	271	Standard query response 000002 A ba5da.com 0 0 116 00 70 0 220 101 30 140 00 004 ba5da.com 004 ba5da.com 0
14	0.972118	10.101.217.115	10.10.0.21	DNS	85	Standard query 000003 AAAA ba5da.com
15	0.975451	10.10.0.21	10.101.217.115	DNS	312	Standard query response 000003 AAAA ba5da.com 004 004 ba5da.com
v Details pane (response)						
Transaction ID: 000001						
Flags: NoError Standard query response, no error						
Questions: 1						
Answer RRs: 1						
Authority RRs: 1						
Additional RRs: 1						
= Query						
= 21.0.10.10.in-addr.arpa type PTR, class IN						
Name: 21.0.10.10.in-addr.arpa						
Name length: 25						
Label count: 0						
Type: PTR (Domain Name Pointer) (12)						
Class: IN (000001)						
= Answer						
= 21.0.10.10.in-addr.arpa type PTR, class IN, druid.rja.edu.cn						
Authoritative name server						
Additional records						
[Resource ID: 10]						
[Time: 0.002210000 seconds]						
0000	20 10 00 05 1a 97 00 00 f3 63 28 52 00 00 25 00	p(R-)				
0010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00000000000000000000000000000000				
0020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00000000000000000000000000000000				
0030	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
0040	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
0050	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
0060	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
0070	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
0080	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
0090	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
00a0	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
00b0	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
00c0	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
00d0	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
00e0	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				
00f0	00 01 00 01 00 01 00 01 00 01 00 01 00 01 00 01	00000000000000000000000000000000				

其中红色为交易 ID，蓝色为查询对象内容，绿色为查询类型，黄色为查询结果

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

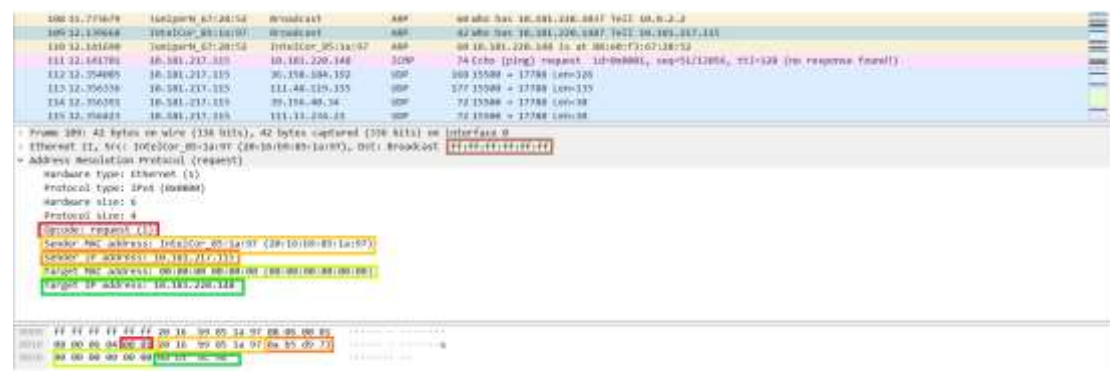
Ping 域名时: ICMP, ARP

ICMP 数据包分别由哪几层协议构成? 3 层, 分别是 Ethernet II, IPv4, Internet Control Message Protocol

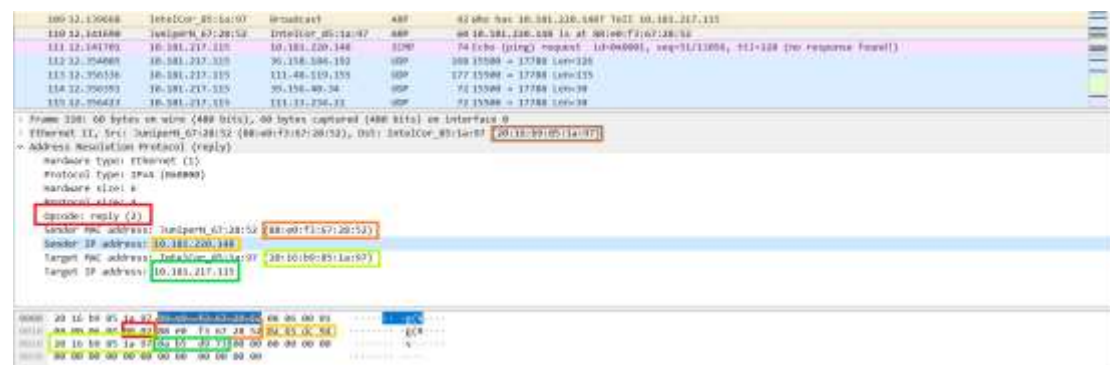
分别选择一个 ARP 请求和响应数据包, 展开最高层协议的详细内容, 标出操作码、发送者 IP 地址、发送者 MAC 地址、查询的目标 IP 地址、Ethernet 层的目标 MAC 地址以及查询结果。

截图如下:

ARP 请求:



ARP 响应:



以上两者中, 红色为操作码, 淡黄色为源 MAC, 橙色为源 IP, 黄色为目标 MAC, 绿色为目标 IP, 褐色为 Ethernet 层目标 MAC

以上两者中，红色为类型，绿色为序号。

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? 3 层, 分别为 Ethernet, IPv4, ICMP(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 16。附上截图:

截图如下:

```
PS C:\Users\Dell\Desktop> ping baidu.com

正在 Ping baidu.com [220.181.38.148] 具有 32 字节的数据:
来自 220.181.38.148 的回复: 字节=32 时间=31ms TTL=51
来自 220.181.38.148 的回复: 字节=32 时间=32ms TTL=51
来自 220.181.38.148 的回复: 字节=32 时间=31ms TTL=51
来自 220.181.38.148 的回复: 字节=32 时间=31ms TTL=51

220.181.38.148 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 31ms, 最长 = 32ms, 平均 = 31ms
PS C:\Users\Dell\Desktop> tracert 220.181.38.148

通过最多 30 个跃点跟踪到 220.181.38.148 的路由

    1      18 ms      2 ms      3 ms     10.0.2.2
    2       2 ms      1 ms      2 ms     10.3.1.46
    3       *         *         *        请求超时。
    4       4 ms      3 ms      2 ms     115.236.178.233
    5       3 ms      3 ms      3 ms     61.130.126.125
    6       9 ms      6 ms      6 ms     61.164.8.120
    7      29 ms     29 ms     29 ms     202.97.26.113
    8       *         *         *        请求超时。
    9       *         *         *        请求超时。
   10      *         *         *        请求超时。
   11     31 ms     31 ms     31 ms     220.181.182.170
   12      *         *         *        请求超时。
   13      *         *         *        请求超时。
   14      *         *         *        请求超时。
   15      *         *         *        请求超时。
   16     31 ms     32 ms     31 ms     220.181.38.148

跟踪完成。
PS C:\Users\Dell\Desktop>
```

前几次 ping 截图:



5880 238.762166	10.181.217.115	238.101.28.140	100%	100 tchko [ping] request: id=000001, seq=200/26880, ttt=15 (no response found)
5873 149.700001	10.181.217.115	238.101.28.140	100%	100 tchko [ping] request: id=000001, seq=200/27118, ttt=15 (no response found)
5876 243.423286	Scripted, 07/18/02	Broadcast	400%	60 who has 10.181.205.252? Tell: 10.0.0.2
5924 344.700034	10.181.217.115	238.101.28.140	100%	100 tchko [ping] request: id=000001, seq=200/27362, ttt=15 (no response found)
5940 149.900073	Scripted, 07/18/02	Broadcast	400%	60 who has 10.181.254.252? Tell: 10.0.0.2
5950 344.701112	10.181.217.115	238.101.28.150	100%	100 tchko [ping] request: id=000001, seq=200/27642, ttt=16 (reply 38 times)
5960 348.701288	238.101.28.142	10.181.217.115	100%	100 tchko [ping] reply: id=000001, seq=200/27642, ttt=16 (request in 1900)
5991 248.317135	10.181.217.115	238.101.28.140	100%	100 tchko [ping] request: id=000001, seq=200/27860, ttt=16 (reply 16 times)
5992 148.800163	238.101.28.142	10.181.217.115	100%	100 tchko [ping] reply: id=000001, seq=200/27860, ttt=16 (request in 1900)
5993 144.800064	10.181.217.115	238.101.28.140	100%	100 tchko [ping] request: id=000001, seq=200/27860, ttt=16 (reply 16 times)
5994 248.800460	238.101.28.142	10.181.217.115	100%	100 tchko [ping] reply: id=000001, seq=200/27860, ttt=16 (request in 1900)
5995 254.005156	Scripted, 07/18/02	Broadcast	400%	60 who has 10.181.254.252? Tell: 10.0.0.2

第一组:

[illegible]

The image shows a Wireshark packet capture of an ICMP Echo (ping) reply. The packet list at the top shows packet 100, selected. The packet details pane shows the following structure:

- Frame 100: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0
- Ethernet II, Src: Janipart_67:2d:52 (88:ad:6d:67:2d:52), Dst: IntelReal_IClax:97 (20:16:0b:0c:1a:97)
- Internet Protocol Version 4, Src: 10.10.1.10, Dst: 10.10.1.1
- Internet Control Message Protocol**
 - ICMP Echo (ping) reply**
 - Type: 0
 - Checksum: 0xffff [correct]
 - Checksum Status: Good
 - Identifier (ID): 1 (0x0001)
 - Sequence Number (ID): 100 (0x00000064)
 - Sequence Number (ID): 21818 (0x00005552)
 - Request frame: 10001
 - Response time: 0.134 ms
 - Data (04 bytes)

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII. The data is 04 bytes long, as indicated by the 'Data (04 bytes)' label. The hexadecimal data is 00 00 00 00, and the ASCII data is four null characters (00).

其中红色为 ICMP 的类型字段

✧ 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: 以太网协议, IPv4 协议, UDP 协议, DNS 协议

HTTP: 以太网协议, IP 协议, TCP 协议, HTTP 协议

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）

截图如下：

DNS:



HTTP:




其中红色为源 IP，橙色为目标 IP，黄色为源端口，绿色为目标端口

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.beian.miit.gov.cn, content-autofill.googleapis.com, toutiao.com, weibo.com, baidu.com, cuepa.cn, qq.com 以及一些 zju.edu.cn 的子域名

部分截图如下：

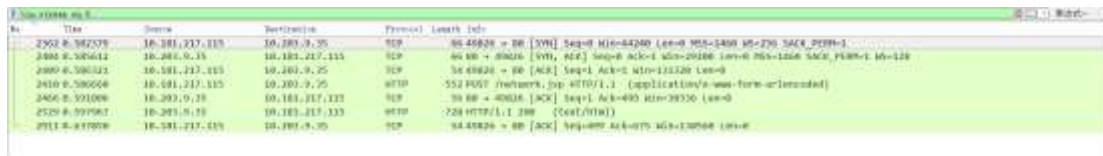


No.	Time	Source	Destination	Protocol	Length	Info
1196	0.470598	10.181.217.115	10.10.0.21	DNS	87	Standard query 0x2f9 A www.beian.miit.gov.cn
1198	0.480880	10.181.217.115	10.10.0.21	DNS	87	Standard query 0x2f9 A www.toutiao.com
1199	0.491040	10.181.217.115	10.10.0.21	DNS	84	Standard query 0x2f9 A www.qq.com
1200	0.500308	10.181.217.115	10.10.0.21	DNS	81	Standard query 0x2f9 A www.beian.miit.gov.cn
1201	0.509600	10.181.217.115	10.10.0.21	DNS	81	Standard query 0x2f9 A content-autofill.googleapis.com
1202	0.518805	10.181.217.115	10.10.0.21	DNS	78	Standard query 0x2f9 A www.cuepa.cn
1203	0.527997	10.181.217.115	10.10.0.21	DNS	75	Standard query 0x2f9 A www.zju.edu.cn
1204	0.537191	10.181.217.115	10.10.0.21	DNS	78	Standard query 0x2f9 A www.zju.edu.cn
1205	0.546385	10.181.217.115	10.10.0.21	DNS	71	Standard query 0x2f9 A www.zju.edu.cn

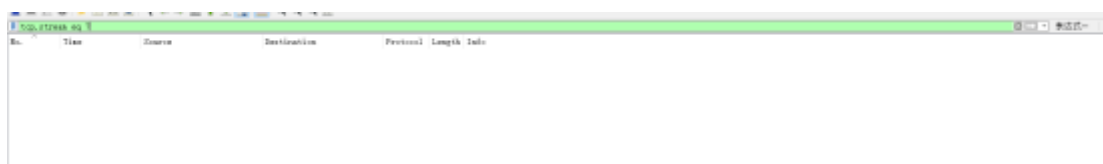
3. 使用显示过滤器 tcp.stream eq X，让 X 从 0 开始变化，直到没有数据。分析浏览器为了获取网页数据，总共建立了几个连接？（一个 TCP 流对应一个 TCP 连接）

TCP 连接数：7

截图如下：



No.	Time	Source	Destination	Protocol	Length	Info
2362	0.582578	10.181.217.115	10.10.0.25	TCP	60	45828 → 80 [SYN] Seq=0 Win=0 Len=0
2363	0.582612	10.10.0.25	10.181.217.115	TCP	60	80 → 45828 [SYN, RST] Seq=0 Win=0 Len=0
2364	0.583121	10.181.217.115	10.10.0.25	TCP	60	45828 → 80 [ACK] Seq=131320 Win=0
2410	0.586560	10.181.217.115	10.10.0.25	HTTP	552	POST /webwork.jsp HTTP/1.1 (application/x-www-form-urlencoded)
2426	0.591080	10.10.0.25	10.181.217.115	TCP	60	80 → 45828 [ACK] Seq=400 Win=28320 Len=0
2529	0.599987	10.10.0.25	10.181.217.115	HTTP	224	HTTP/1.1 200 (text/html)
2711	0.617030	10.181.217.115	10.10.0.25	TCP	60	45828 → 80 [ACK] Seq=400 Win=0 Len=0



No.	Time	Source	Destination	Protocol	Length	Info
-----	------	--------	-------------	----------	--------	------

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：67

解释：各个 TCP 流分别有 12, 15, 9, 8, 5, 17, 1 个对话，共 67 个

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）：

截图如下：

其中红色部分为请求，蓝色部分为响应。

选择序号为 4（共 5 次对话）的 TCP 流进行跟踪：

```
GET /_js/_portletPlugs/datepicker/css/datepicker.css HTTP/1.1
Host: www.zju.edu.cn
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36
Accept: text/css,*/*;q=0.1
Referer: http://www.zju.edu.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: _ga=GA1.3.459161614.1598533995; route=66f625e2c030628cd6133593896946a;
Hm_lvt_fe30bbc1ee45421ec1679d1b8df8453=1599722077; Hm_lpv_fe30bbc1ee45421ec1679d1b8df8453=1599722077;
JSESS10R1D=D206D37B342CF408D99495B4EAD180B2; Hm_lvt_eaa57ca47dab4ad4f5a257801a3457c=1599722083;
Hm_lpv_eaa57ca47dab4ad4f5a257801a3457c=1599722083;
sudy_ck=1217F1E1E8F8488B79051BF9DBFE854617F7BD3AE3ED60F950B3FF4AD1F932C0197BB888D83E820D113F64FB704C89FDEC2CE0ECAD163D1E
C30CB0F1C01E6FDB8DAB459CE4A3555F1BD08082335671; _csrftoken=Vj9Kw0F2uQ0T1CeAFecQycKqZp0M4HfZCrmG3D;
_pvt=2HP43kqisgKGn0o0w0d2FbqLEpOG0083e5MygMl
%2F1valhnmwz2c1LX1J6Ed5F3y5G3EH1RZ1ZrA4201VE50A1T06H3xeyY9ah00H5dCP425nu58uJ06W
%2Bwadd05KAYHT7PFREFK108e6UuJooVZP4mVVCjH5HX3f6TjHs7PcaUGTphk92PCceV5SWtG5d6pKXTZdZ3LwEwLj3%2FXP1AN
%2Bupcd5Fp1tWc11Tqst3mFRGZjG31KKnhAunnJyl1we9dShubQQR11pUG0sekurneBx11Z17Cm4ek3mgeEct01Ka1rC5MK1qerIghy76U6OL60y24Acf
%2F2F3pA3Lwgaeshw1d01n21RSastD5akeDqocwYjAKCM0wae%2F6FTFJQnaBhh91lgH5TbngMc1fC11ZC1Ag05Z2Zx0KAR05P03D;
_pft=UOL5P1Z7%2BEXgmkdtbkgyw7m8ZFKeyHCCgK7BhnaQ03D; _pc0=Cugc1c150T0KAKGQLBqB0DyW55Bj038EeqCC%2Fw5nZhcAw00a
%2FwBVE2E9ef1jX0K; iPlanetDirectoryPro=bbqDg5gRVR6G0Arkvyv1HcggmXh0v73wRwFkyBz2U7YzC9V0qjmc2B5BvK0DLn5K0M94hg1nQ3cb1D
%2B1XV1Z0X7QhtKjrn1UHktxfRfK%2B0PT7P2Hsy3nv%2Fcy90FCput2e3fKZ2q0Vh1Zm00ohGChqY8n5eLcAXZ01T3Fd
%2B97U7921B081wX7p1mVnZuc0W8E651Tb0%2Fw1D0Kw7F7YtZpWYPG1ESfB3KvYnhGfGhagc11w0280v1E9ve0X530W61Zxx%2F12kdZw
%2Bgfs4pZrKcGX5fX28FbF1B0W03EeknNAW03B0M45A00awCOTeAEbfbDUjWY%2B031dG04ANZ%2B0y7jW1Ag3p1JwMgo1Z8E82033C

HTTP/1.1 200 OK
Server: nginx
Date: Mon, 21 Sep 2020 03:28:17 GMT
Content-Type: text/css
Content-Length: 1456
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Frame-Options: SAMEORIGIN
Last-Modified: Mon, 21 Mar 2016 05:58:10 GMT
ETag: "1500-52e08c000c000-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding

.....0.....\.....0.....GET /_js/jquery.sudy.wp.visitcount.js HTTP/1.1
Host: www.zju.edu.cn
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36
Accept: */*
Referer: http://www.zju.edu.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: _ga=GA1.3.459161614.1598533995; route=66f625e2c030628cd6133593896946a;
Hm_lvt_fe30bbc1ee45421ec1679d1b8df8453=1599722077; Hm_lpv_fe30bbc1ee45421ec1679d1b8df8453=1599722077;
JSESS10R1D=D206D37B342CF408D99495B4EAD180B2; Hm_lvt_eaa57ca47dab4ad4f5a257801a3457c=1599722083;
Hm_lpv_eaa57ca47dab4ad4f5a257801a3457c=1599722083;
sudy_ck=1217F1E1E8F8488B79051BF9DBFE854617F7BD3AE3ED60F950B3FF4AD1F932C0197BB888D83E820D113F64FB704C89FDEC2CE0ECAD163D1E
C30CB0F1C01E6FDB8DAB459CE4A3555F1BD08082335671; _csrftoken=Vj9Kw0F2uQ0T1CeAFecQycKqZp0M4HfZCrmG3D;
_pvt=2HP43kqisgKGn0o0w0d2FbqLEpOG0083e5MygMl
%2F1valhnmwz2c1LX1J6Ed5F3y5G3EH1RZ1ZrA4201VE50A1T06H3xeyY9ah00H5dCP425nu58uJ06W
%2Bwadd05KAYHT7PFREFK108e6UuJooVZP4mVVCjH5HX3f6TjHs7PcaUGTphk92PCceV5SWtG5d6pKXTZdZ3LwEwLj3%2FXP1AN
%2Bupcd5Fp1tWc11Tqst3mFRGZjG31KKnhAunnJyl1we9dShubQQR11pUG0sekurneBx11Z17Cm4ek3mgeEct01Ka1rC5MK1qerIghy76U6OL60y24Acf
%2F2F3pA3Lwgaeshw1d01n21RSastD5akeDqocwYjAKCM0wae%2F6FTFJQnaBhh91lgH5TbngMc1fC11ZC1Ag05Z2Zx0KAR05P03D;
_pft=UOL5P1Z7%2BEXgmkdtbkgyw7m8ZFKeyHCCgK7BhnaQ03D; _pc0=Cugc1c150T0KAKGQLBqB0DyW55Bj038EeqCC%2Fw5nZhcAw00a
%2FwBVE2E9ef1jX0K; iPlanetDirectoryPro=bbqDg5gRVR6G0Arkvyv1HcggmXh0v73wRwFkyBz2U7YzC9V0qjmc2B5BvK0DLn5K0M94hg1nQ3cb1D
%2B1XV1Z0X7QhtKjrn1UHktxfRfK%2B0PT7P2Hsy3nv%2Fcy90FCput2e3fKZ2q0Vh1Zm00ohGChqY8n5eLcAXZ01T3Fd
%2B97U7921B081wX7p1mVnZuc0W8E651Tb0%2Fw1D0Kw7F7YtZpWYPG1ESfB3KvYnhGfGhagc11w0280v1E9ve0X530W61Zxx%2F12kdZw
%2Bgfs4pZrKcGX5fX28FbF1B0W03EeknNAW03B0M45A00awCOTeAEbfbDUjWY%2B031dG04ANZ%2B0y7jW1Ag3p1JwMgo1Z8E82033C

HTTP/1.1 200 OK
Server: nginx
Date: Mon, 21 Sep 2020 03:28:17 GMT
Content-Type: application/javascript
Content-Length: 1309
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Frame-Options: SAMEORIGIN
Last-Modified: Fri, 05 May 2017 06:39:56 GMT
ETag: "1938-54ec128570700-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
```

```
..k.p5.....GET /_upload/tpl/05/e5/1509/template1509/js/plugin.js HTTP/1.1
Host: www.zju.edu.cn
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36
Accept: */*
Referer: http://www.zju.edu.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: _ga=GA1.3.459161614.1598533995; route=66f625e2c030628c4d6133593896946a;
Hm_lvt_fe30bbc1ee45421ec1679d1b8df8453=1599722077; Hm_lpvtf_fe30bbc1ee45421ec1679d1b8df8453=1599722077;
JSESSIOHID=D206D378342CFA08D99495BAEAD100B2; Hm_lvt_eaa57ca47d4cb4ad4f5a257001a3457c=1599722083;
Hm_lpvtf_eaa57ca47d4cb4ad4f5a257001a3457c=1599722083;
sudy_ck=1217FE1E8F8488B79051BF9DBFE85E617F78D3AE3ED60F95083FF4AD1F932C0197886880B3EB200113F64F8704C89FDEECB7C7E8EAD1630E
C30CB8F1CD1E6F08B0ABA59CE4A3555F18D00882335671; _csrf=S8mnp1V19KMoF2MQ0TLCeAFe9CqycKRQZp8MhrfzCrHX3D;
_pvo=2MP43kqHsgKgn8odwHd2FbqlEp000n83e5MyghL
Q2FlvahYwGzcLlX136Ed5F3yS63EhLRZIZrAW20LVE50A1T0BH3XeYY9ah0DN5dCP42Utu5BmJUEM
Q2BMMWNSKAYNT7PFRfEfk10Bm6UuJooyZP4mVvcJHSHx3f6t1Hs7PcauGTpchK9ZPCceV5N5wtgSdGpKXtZdZ3LUEWlj3X2FXPIAH
Q2Bupcd5FpItWc11TqstJmFRGZjG31KKnbAunnuy11ue9dShubQ0r11pUG08skurmeBx11z17CPMeK3mgeEc61Ka1rC9H81qerTghMyZ6U6L6By24ALfca
Q2FP1M0A31wgAexhWIdu1zn21RSastDsAkeDqco4WYJAKCHWneK2F6ftF3QmABHh911gh5TbhgmC1fc1I2CIAg05zz2xDKAR05FK3D;
_pfo=UOLSP127%2BEKXgpkdtkbgya3m82FKevhKc9YC7BHNzQ03D; _pc0=Cugcic150TKaKGLBq8DyW55Bj038EeqwC2Fw5n2HCaWnHa
Q2FwVIE2E9eFjJMXi; IPlanetDirectoryPro-hbqDgSgsRVR6G0ArKvYy1HcgghHbHbvh73wRufkvBzh2UTVzC9VBqjnd2B5Bvk00Ln5K0uP4bg1nQ1c1b
Q2B1xV12HX7QHTKjrn1UHktxf9FKS2BDPT7P2H5y7nV%2Fcy90FCput2e3fKz2q6Vh1Zmdoo0H6ChqYBn5eLCAXZ0IT3Fd
Q2B97U7021B801wK7p1wH2UC0NG8E651tBb%2FW1DGkw7FYmZyPwYPGJESFb2KVMYhgfGhagc1luK2B6v1E9V8X536w632xx2F12kd24
Q2BgfShpZrkCXsFK2Bfbf1BwV03EekHnARQ2B4d5A0oac0teAEDbf8DUjwYK2Bm3ldG0Hn2K28ny7JwIAag3pl3wMgo128e828X3D

HTTP/1.1 200 OK
Server: nginx
Date: Mon, 21 Sep 2020 03:28:17 GMT
Content-Type: application/javascript
Content-Length: 6213
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Frame-Options: SAMEORIGIN
Last-Modified: Tue, 30 Jul 2019 15:40:43 GMT
ETag: "6029-S8ee7d4d6f4c0-gzip"
```

```
$$......RD[.NO[.L.N.R.I][.C.G[.L.B[.C.C[.L.V.R[.C.C[.L.P[.C.C[.L.AK,)...GET /_upload/article/
images/f5/cb/5bb0958748d1ac785b3577d59bee/f936b334-c909-4b6f-bb2a-993f91b1e6e.jpg HTTP/1.1
Host: www.zju.edu.cn
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/*,*/*;q=0.8
Referer: https://www.zju.edu.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: _ga=GA1.3.459161614.1598533995; route=66f625e2c030628c4d6133593896946a;
Hm_lvt_fe30bbc1ee45421ec1679d1b8df8453=1599722077; Hm_lpvtf_fe30bbc1ee45421ec1679d1b8df8453=1599722077;
JSESSIOHID=D206D378342CFA08D99495BAEAD100B2; Hm_lvt_eaa57ca47d4cb4ad4f5a257001a3457c=1599722083;
Hm_lpvtf_eaa57ca47d4cb4ad4f5a257001a3457c=1599722083;
sudy_ck=1217FE1E8F8488B79051BF9DBFE85E617F78D3AE3ED60F95083FF4AD1F932C0197886880B3EB200113F64F8704C89FDEECB7C7E8EAD1630E
C30CB8F1CD1E6F08B0ABA59CE4A3555F18D00882335671; _csrf=S8mnp1V19KMoF2MQ0TLCeAFe9CqycKRQZp8MhrfzCrHX3D;
_pvo=2MP43kqHsgKgn8odwHd2FbqlEp000n83e5MyghL
Q2FlvahYwGzcLlX136Ed5F3yS63EhLRZIZrAW20LVE50A1T0BH3XeYY9ah0DN5dCP42Utu5BmJUEM
Q2BMMWNSKAYNT7PFRfEfk10Bm6UuJooyZP4mVvcJHSHx3f6t1Hs7PcauGTpchK9ZPCceV5N5wtgSdGpKXtZdZ3LUEWlj3X2FXPIAH
Q2Bupcd5FpItWc11TqstJmFRGZjG31KKnbAunnuy11ue9dShubQ0r11pUG08skurmeBx11z17CPMeK3mgeEc61Ka1rC9H81qerTghMyZ6U6L6By24ALfca
Q2FP1M0A31wgAexhWIdu1zn21RSastDsAkeDqco4WYJAKCHWneK2F6ftF3QmABHh911gh5TbhgmC1fc1I2CIAg05zz2xDKAR05FK3D;
_pfo=UOLSP127%2BEKXgpkdtkbgya3m82FKevhKc9YC7BHNzQ03D; _pc0=Cugcic150TKaKGLBq8DyW55Bj038EeqwC2Fw5n2HCaWnHa
Q2FwVIE2E9eFjJMXi; IPlanetDirectoryPro-hbqDgSgsRVR6G0ArKvYy1HcgghHbHbvh73wRufkvBzh2UTVzC9VBqjnd2B5Bvk00Ln5K0uP4bg1nQ1c1b
Q2B1xV12HX7QHTKjrn1UHktxf9FKS2BDPT7P2H5y7nV%2Fcy90FCput2e3fKz2q6Vh1Zmdoo0H6ChqYBn5eLCAXZ0IT3Fd
Q2B97U7021B801wK7p1wH2UC0NG8E651tBb%2FW1DGkw7FYmZyPwYPGJESFb2KVMYhgfGhagc1luK2B6v1E9V8X536w632xx2F12kd24
Q2BgfShpZrkCXsFK2Bfbf1BwV03EekHnARQ2B4d5A0oac0teAEDbf8DUjwYK2Bm3ldG0Hn2K28ny7JwIAag3pl3wMgo128e828X3D

HTTP/1.1 200 OK
Server: nginx
Date: Mon, 21 Sep 2020 03:28:18 GMT
Content-Type: image/jpeg
Transfer-Encoding: chunked
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Frame-Options: SAMEORIGIN
Last-Modified: Wed, 19 Aug 2020 02:09:06 GMT
ETag: "992cf-Sad317f66ce00-gzip"
```

```
202 /_upload/article/images/6b/3d/fc98b2124a94a729ec9ab4738dci/9f53a3b8-3988-495c-be67-8ec8296d6427.jpg HTTP/1.1
Host: www.zju.edu.cn
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.102 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://www.zju.edu.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: _ga=GA1.3.459161614.1598533095; route=66f625e2c038628c4d6133593896046a;
hm_lvt_fa30bbc1e9a5421e61670d1b8d8f8453-1599722077; JSESSIONID=02060378342CFA88D904058AEAD188B2;
hm_lvt_aaa57ca47dacb4ad4f5a257001a3457c-1599722083; hm_lvt_aaa57ca47dacb4ad4f5a257001a3457c-1599722083;
sudy_ck=1217f1e1f8f88879a51bf90dfe85e617f7803ae3ed60f95083ff4ad1f932c019788688083EB200113f64f8704c89fDEECB2C7E0F3AD163D1
C30C8BF1CD1E6FDB8DABA59CE4A3555F18008882335671; _csrf=58awp1V10KMOF2wQ0T1CeAF-e9CqyKq7p8BhrfzCrr03D;
_pvh=2MP4JkqHsgG6n8oW4d2FbqLEpOGm83w5MygmI
%2F1valenVWzcl1X1JGE05F3y5G3EH1R2I2rAw201VES0A1T0bH3xeYY9ah00N5dcP42sNu5BefJ0u4
%2BMM95KAYNT7PFREfkIDb6uuJooY2P4mVycJH5Hx3Fot1Hs7Pcau6TpcHk92PCceV5H5wtgSdopKxT2dZ3LMFWLJ3ZJFXP1AN
%2Bapcd5fpITWc11TqstJmFRGZJG3IKKbAunmuy1Jue9dShubQQe11p0G8sekuame8x1IzI7CMAek3eggEc161Ka17C9Hk1q0RTghMyZ6U60L60Y4AL11T
%2FP3mdA3CvgAocHw1du12n21R5aktDsAkeDqcodAvJAckHwde32F6ftF3QwAbH011gn5TbqHc1fc11ZC1Ag05z12XDKA8052G3D;
_pfb=UOL5P127X20EXgwdtBkywJm8ZfKevHkGPK78HuaQd3D; _pfb=Cugc1c150TKwAGGtLhQDDYw558j038eEqaC%2Fw5n2HKA08u
%2F8bVE2E9ef1J0K1; LP1anetDrectoryPro-bbqBgsSrvr6GoArKvY1hcggR00thdv7JwbuFkvBzh2UTYzC9VbqJnG285BvkDDLn5K0wP4h3Inq7cbJ
%2B1xV1ZHX7QHTKJro1uHkxf9FK32BDPT7P205yJmVZfCy90fCpUT2e13FK2Zq6Vh1ZedooCh6ChqY8NSeLC4X20Tt1Fd
%2B97J921080JwZp1wvZuCO8G0E651180Z2FwIDGkww7FmZyPwVPGJf5f6ZXwvthdFhagc11wG2B6v1E9VeRXS3m4612xxZf1Zkd2w
%2BgfSip2rKcGXsFX2Bfbf1MvV03EknHwPQZ8H4d5A0oawcotEAE0DFR0UjWYX28w31dG0HAnZS28My7JwIAag3p13wMgo128e8203D;
hm_lvt_fa30bbc1e9a5421e61670d1b8d8f8453-1600658508

HTTP/1.1 200 OK
Server: nginx
Date: Mon, 21 Sep 2020 03:28:18 GMT
Content-Type: image/jpeg
Transfer-Encoding: chunked
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Frame-Options: SAMEORIGIN
```


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
- 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；
 3. ping 一个域名和 ping 一个 IP 地址出现数据包的区别在于：当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.
 - ① 首先从源地址发出一个 UDP 探测包到目的地址，并将 TTL 设置为 1，每次到达路由器的时候将 TTL 减 1，当 TTL 变为 0 时，包被丢弃，路由器向源地址发回一个 ICMP 超时通知 (ICMP Time Exceeded Message)，内含发送 IP 包的所有内容及路由器的 IP 地址；
 - ② 当源地址收到该 ICMP 包时，显示这一跳路由信息；
 - ③ 每次发送时将设置的 TTL 值加 1，依次如此进行，直到抵达最终的目的地。
- 如何理解 TCP 连接和 HTTP 会话？他们之间存在什么关系？
 1. 首先 HTTP 协议是在 TCP 协议之上建立的，HTTP 在发起请求时通过 TCP 协议建立起连接服务器的通道，请求结束后，立即断开 TCP 连接
 2. 其次在结构方面，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。

3. 同时 HTTP 是无状态的短连接，而 TCP 是有状态的长连接

- 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？

1. 使用基于 UDP 的 DNS 协议只要一个请求和一个应答就可以完成；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；所以基于 TCP 协议的 DNS 更浪费网络资源。

2. DNS 数据包不是那种大数据包，所以使用 UDP 不需要考虑分包，如果丢包那么就是全部丢包，如果收到了数据，那就是收到了全部数据。所以只需要考虑丢包的情况，丢包后只需要重新请求一次就好了。而且 DNS 的报文允许填入序号字段，对于请求报文和其对应的应答报文，这个字段是相同的，通过它可以区分 DNS 应答是对应的哪个请求

3. 同时，UDP 最多只有 512Byte，只能传输小型数据。而 HTTP 一般是对大量数据进行操作，所以应该使用 TCP 协议。

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：

该实验较为简单，主要是针对 wireshark 抓包的使用。filter 语法并不复杂，但是想真正拿到自己想要的东西，还并不容易。我在网上进行了相关知识的大量学习，发现其实本实验中我们用到的 filter 知识只是冰山一角，还有正则表达式等更加精妙的用法等着我们去学习。

同时我也意识到了书本上的网络协议和网络上真实运行流动的各种各样的分层协作的协议是不一样的。不同的指令需要的网络协议是不一样的。在我们专业中，更多的知识应该通过实践去获取，而不能简单停留在书本层次。

在本次实验中我学会了使用 wireshark 软件捕获各协议报文内容，并展开对其内部数据的传输做一定的了解。一些过滤统计的技巧也让实验变得容易。通过对报文格式的分析，我也更加深入地了解 ARP 协议，ICMP 协议，HTTP 协议的工作过程。

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：

对于 filter 语法的使用需要进一步的学习。计算机网络实验需要不断自我超越，动手操作。这些实验可以提高自己动手解决问题的能力。

一些 filter 的语法可以参照这篇文章：

<https://www.cnblogs.com/laoxiajiadeyun/p/10365073.html>

另外对于 TCP 流我不是很熟悉，也查找了大量的知识进行学习。，可以参照这篇文章进行学习：

<https://blog.csdn.net/bcbobo21cn/article/details/91349077>

你对本实验安排有哪些更好的建议呢？欢迎献计献策：

我觉得本实验较为基础，目前的安排已经很不错了。一个建议是，报告中有一些词语不是很通俗易懂，比如外部 IP 等，或许可以给出明确的指示，比如告知学生：先 ping 指定域名，然后获得一个 IP，然后 tracert 该 IP 来进行实验。或许更便于学生理解。