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：　卢雨洁

学　　院：　计算机学院

系：　计算机科学与技术学院

专　　业：　计算机科学与技术

学　　号：　3150105267

指导教师：　邱劲松

年　　月　　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。

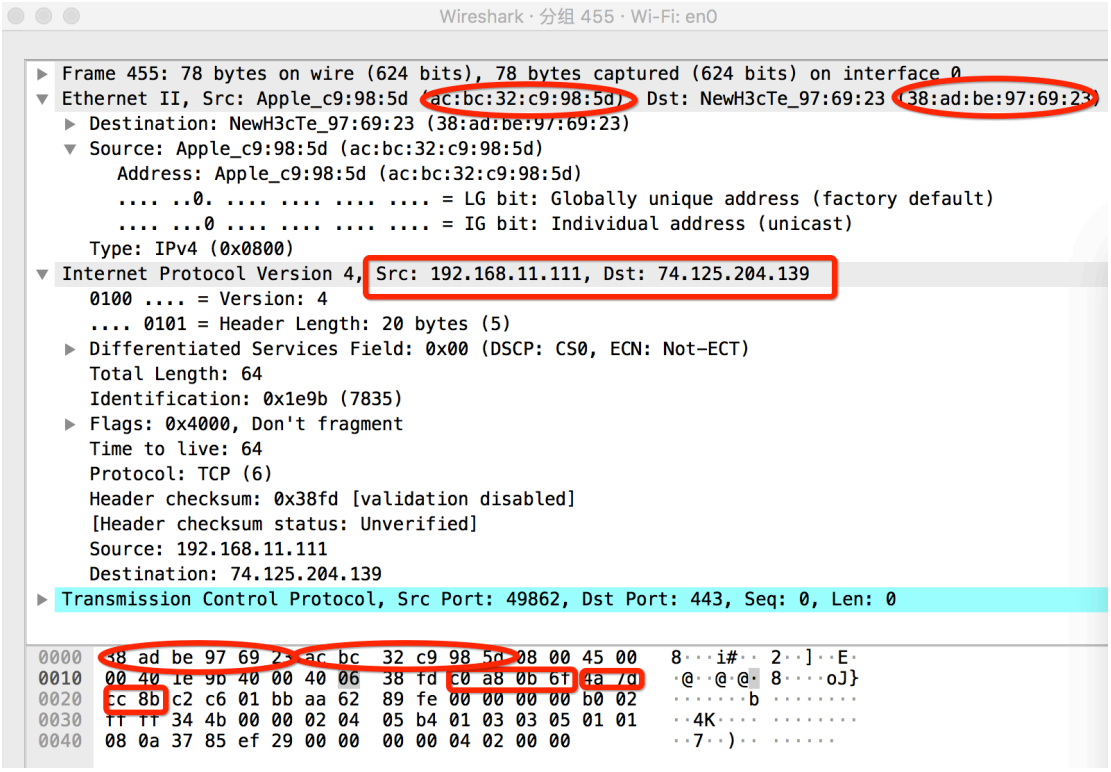
✧ Part One

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）。**

   协议名： TCP DNS MDNS ARP HTTP

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 传输层，从 Ethernet 开始往上，各层协议的名字分别为： 数据链路层，网络层， 传输层 。**

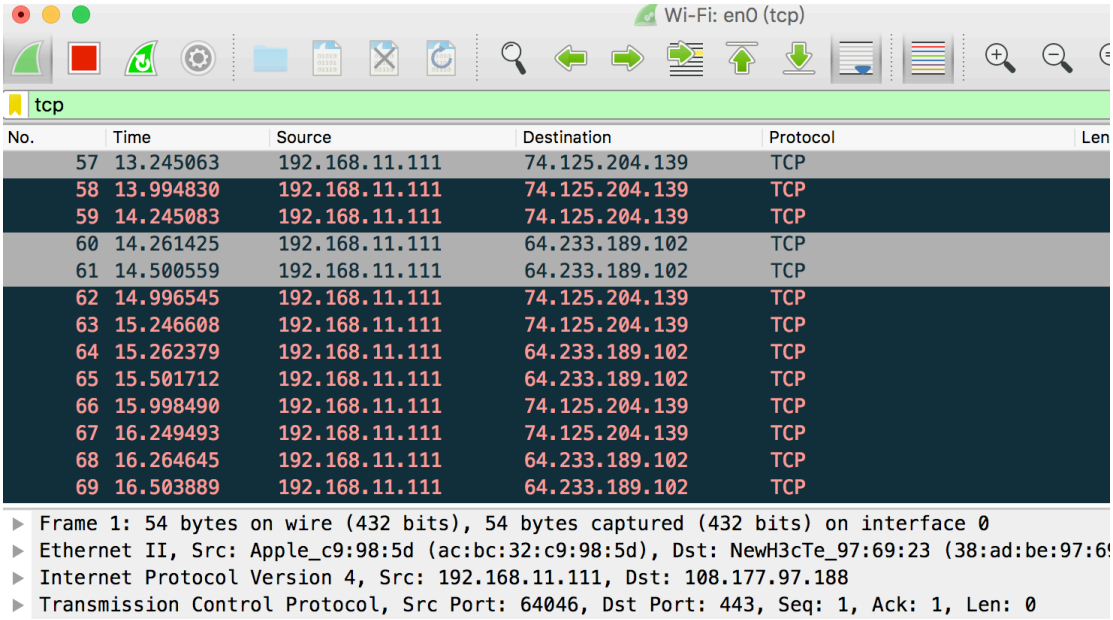
   展开 IP 层协议，标出源 IP 地址、目标 IP 地址及其在数据包中的具体位置，展开 Ethernet 层，标出源 MAC 地址和目标 MAC 地址及其在数据包中的具体位置。



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：tcp ，希望显示的协议类型：TCP 。

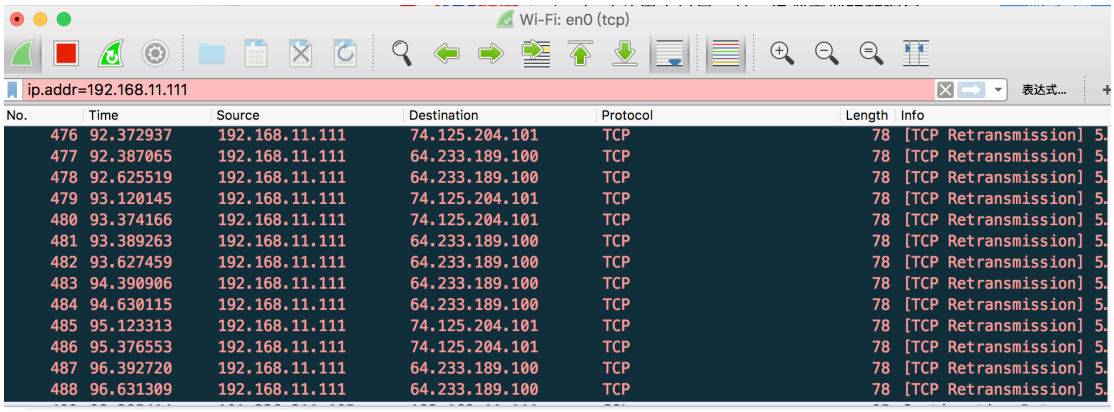
截图：



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=192.168.11.111 ，希望显示的 IP 地址：192.168.11.111 。

截图：



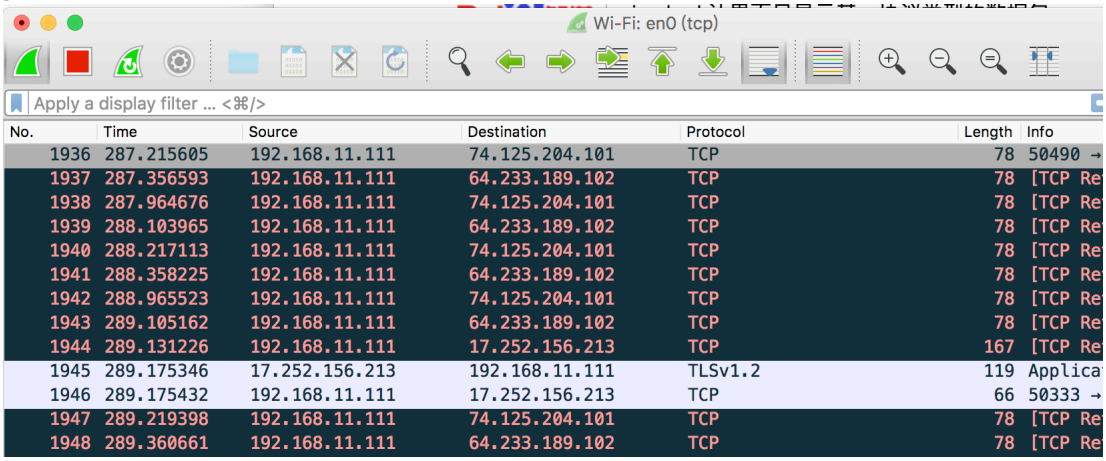
No.	Time	Source	Destination	Protocol	Length	Info
476	92.372937	192.168.11.111	74.125.204.101	TCP	78	[TCP Retransmission] 5.
477	92.387065	192.168.11.111	64.233.189.100	TCP	78	[TCP Retransmission] 5.
478	92.625519	192.168.11.111	64.233.189.100	TCP	78	[TCP Retransmission] 5.
479	93.120145	192.168.11.111	74.125.204.101	TCP	78	[TCP Retransmission] 5.
480	93.374166	192.168.11.111	74.125.204.101	TCP	78	[TCP Retransmission] 5.
481	93.389263	192.168.11.111	64.233.189.100	TCP	78	[TCP Retransmission] 5.
482	93.627459	192.168.11.111	64.233.189.100	TCP	78	[TCP Retransmission] 5.
483	94.390906	192.168.11.111	64.233.189.100	TCP	78	[TCP Retransmission] 5.
484	94.630115	192.168.11.111	64.233.189.100	TCP	78	[TCP Retransmission] 5.
485	95.123313	192.168.11.111	74.125.204.101	TCP	78	[TCP Retransmission] 5.
486	95.376553	192.168.11.111	74.125.204.101	TCP	78	[TCP Retransmission] 5.
487	96.392720	192.168.11.111	64.233.189.100	TCP	78	[TCP Retransmission] 5.
488	96.631309	192.168.11.111	64.233.189.100	TCP	78	[TCP Retransmission] 5.

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 192.168.11.111 ，希望捕获的 IP 地址：192.168.11.111 。

截图：

名称	过滤器
新建捕获过滤器	host 192.168.11.111
Ethernet address 00:00:5e:00:53:00	ether host 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	ether proto 0x0806
No Broadcast and no Multicast	not broadcast and not multicast
No ARP	not arp
IPv4 only	ip

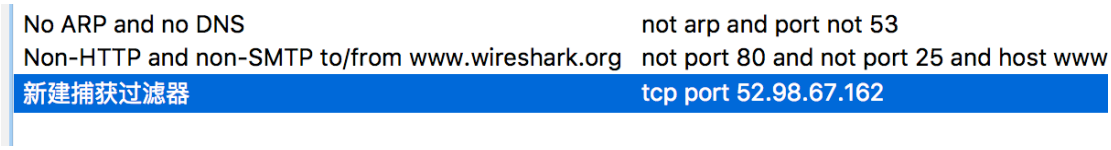


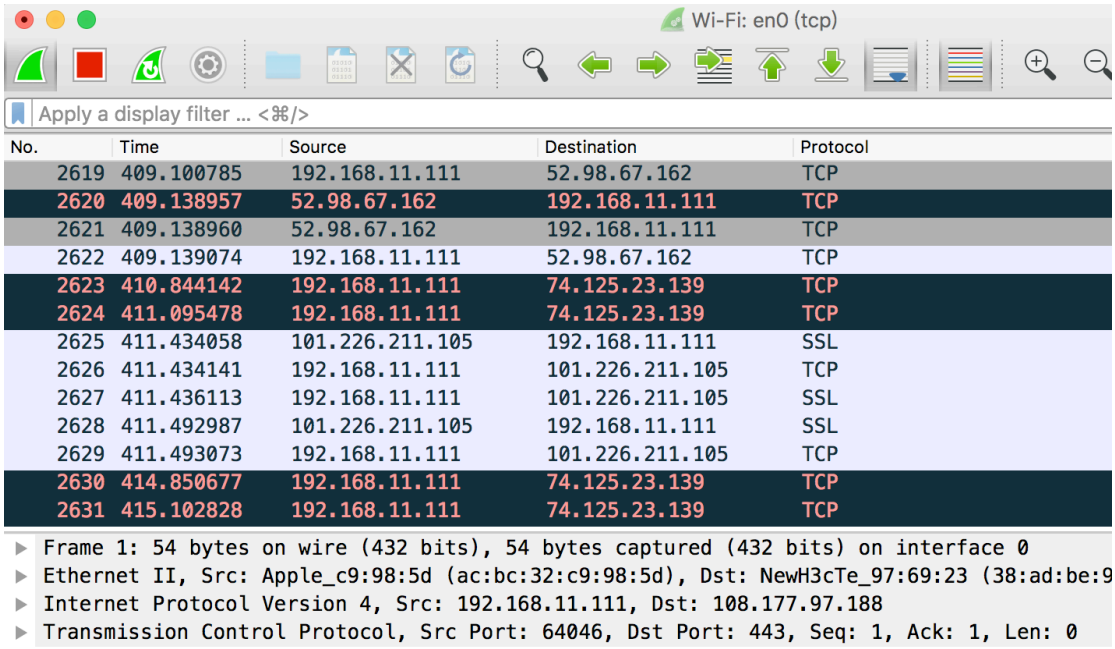
No.	Time	Source	Destination	Protocol	Length	Info
1936	287.215605	192.168.11.111	74.125.204.101	TCP	78	50490 →
1937	287.356593	192.168.11.111	64.233.189.102	TCP	78	[TCP Re
1938	287.964676	192.168.11.111	74.125.204.101	TCP	78	[TCP Re
1939	288.103965	192.168.11.111	64.233.189.102	TCP	78	[TCP Re
1940	288.217113	192.168.11.111	74.125.204.101	TCP	78	[TCP Re
1941	288.358225	192.168.11.111	64.233.189.102	TCP	78	[TCP Re
1942	288.965523	192.168.11.111	74.125.204.101	TCP	78	[TCP Re
1943	289.105162	192.168.11.111	64.233.189.102	TCP	78	[TCP Re
1944	289.131226	192.168.11.111	17.252.156.213	TCP	167	[TCP Re
1945	289.175346	17.252.156.213	192.168.11.111	TLSv1.2	119	Applica
1946	289.175432	192.168.11.111	17.252.156.213	TCP	66	50333 →
1947	289.219398	192.168.11.111	74.125.204.101	TCP	78	[TCP Re
1948	289.360661	192.168.11.111	64.233.189.102	TCP	78	[TCP Re

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 52.98.67.162 ，希望捕获的协议类型：52.98.67.162 。

截图：





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？ 物理层，数据链路层，网络层，传输层，应用层 **。 使用的服务方端口是：** 53 **。**

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

```
▶ User Datagram Protocol, Src Port: 65406, Dst Port: 53
▼ Domain Name System (query)
    Transaction ID: 0x1ef2
  ▶ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▼ Queries
    ▼ wu.apple.com.akadns.net: type A, class IN
        Name: wu.apple.com.akadns.net
        [Name Length: 23]
        [Label Count: 5]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 10]
```

```
0000  88 e0 f3 b2 60 ce ac bc  32 c9 98 5d 08 00 45 00   ····`··· 2··]··E·
0010  00 45 06 a7 00 00 ff 11  57 8c 0a b4 48 a2 0a 0a   ·E····· W···H···
0020  00 15 ff 7e 00 35 00 31  67 78 1e f2 01 00 00 01   ···~·5·1 gx······
0030  00 00 00 00 00 00 02 77  75 05 61 70 70 6c 65 03   ·······w u·apple·
0040  63 6f 6d 06 61 6b 61 64  6e 73 03 6e 65 74 00 00   com·akad ns·net··
0050  01 00 01                                          ···
```

响应包

```
▶ Internet Protocol Version 4, Src: 10.10.0.21, Dst: 10.180.72.162
▶ User Datagram Protocol, Src Port: 53, Dst Port: 65406
▼ Domain Name System (response)
    Transaction ID: 0x1ef2
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 10
    Additional RRs: 5
  ▼ Queries
    ▼ wu.apple.com.akadns.net: type A, class IN
        Name: wu.apple.com.akadns.net
        [Name Length: 23]
        [Label Count: 5]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
```

```
0000  ac bc 32 c9 98 5d 88 e0  f3 b2 60 ce 08 00 45 48   ··2··]·· ··`···EH
0010  01 9b 3e f5 00 00 3b 11  e1 a0 0a 0a 00 15 0a b4   ··>···;· ········
0020  48 a2 00 35 ff 7e 01 87  02 0a 1e f2 81 80 00 01   H··5·~·· ········
0030  00 02 00 0a 00 05 02 77  75 05 61 70 70 6c 65 03   ·······w u·apple·
0040  63 6f 6d 06 61 6b 61 64  6e 73 03 6e 65 74 00 00   com·akad ns·net··
0050  01 00 01 c0 0c 00 05 00  01 00 00 00 20 00 09 06   ········ ···· ···
0060  77 75 2d 6e 77 6b c0 0f  c0 35 00 01 00 01 00 00   wu-nwk·· ·5······
0070  00 12 00 04 11 fe 20 10  c0 19 00 02 00 01 00 02   ······ · ········
0080  06 c1 00 14 07 61 31 38  2d 31 32 38 06 61 6b 61   ·····a18 -128·aka
0090  64 6e 73 03 6f 72 67 00  c0 19 00 02 00 01 00 02   dns·org· ········
00a0  06 c1 00 0a 07 61 31 32  2d 31 33 31 c0 62 c0 19   ·····a12 -131·b··
00b0  00 02 00 01 00 02 06 c1  00 0a 07 61 32 38 2d 31   ········ ···a28-1
00c0  32 39 c0 62 c0 19 00 02  00 01 00 02 06 c1 00 09   29·b···· ········
00d0  06 61 35 2d 31 33 30 c0  62 c0 19 00 02 00 01 00   ·a5-130· b·······
00e0  02 06 c1 00 09 06 61 37  2d 31 33 31 c0 19 c0 19   ·····a7 -131·····
00f0  00 02 00 01 00 02 06 c1  00 09 06 61 33 2d 31 32   ········ ···a3-12
0100  39 c0 19 c0 19 00 02 00  01 00 02 06 c1 00 09 06   9······· ········
0110  61 31 2d 31 32 38 c0 19  c0 19 00 02 00 01 00 02   a1-128·· ········
0120  06 c1 00 09 06 61 39 2d  31 32 38 c0 19 c0 19 00   ·····a9- 128·····
0130  02 00 01 00 02 06 c1 00  0a 07 61 31 31 2d 31 32   ········ ··a11-12
0140  39 c0 19 c0 19 00 02 00  01 00 02 06 c1 00 0a 07   9······· ········
0150  61 31 33 2d 31 33 30 c0  62 c0 e5 00 01 00 01 00   a13-130· b·······
```

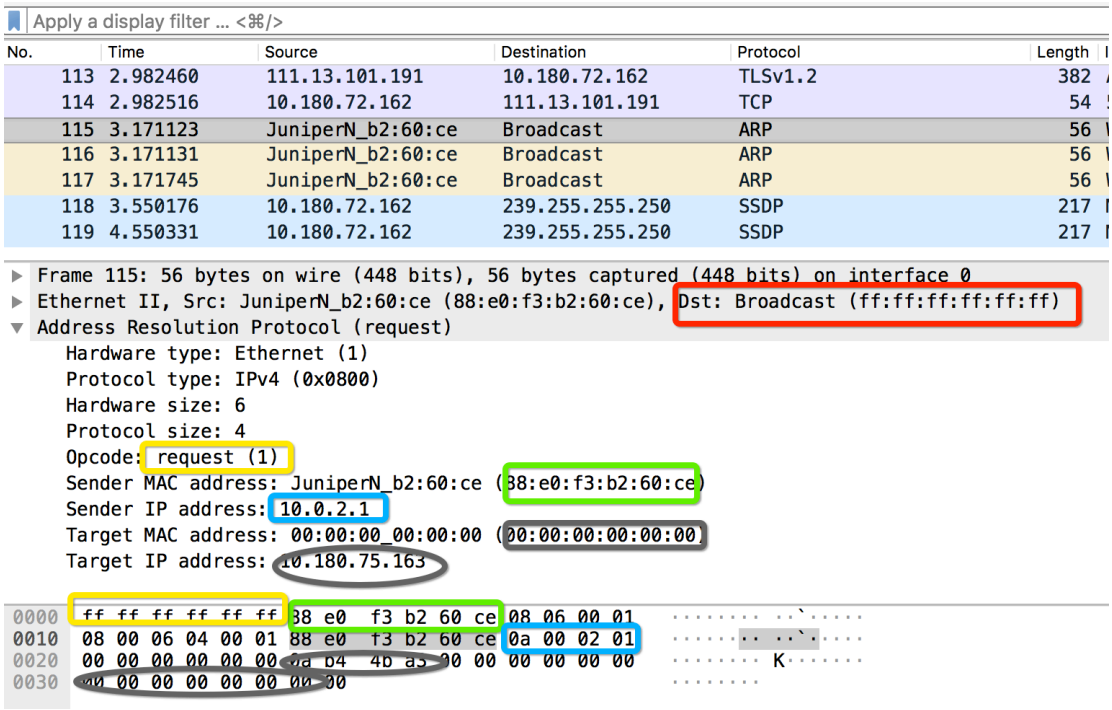
任务 2：使用 Ping 命令，分别测试某个 IP 地址和某个域名的连通性，并捕获数据包。捕获到了哪些相关协议数据包？

Ping IP 地址时：_____

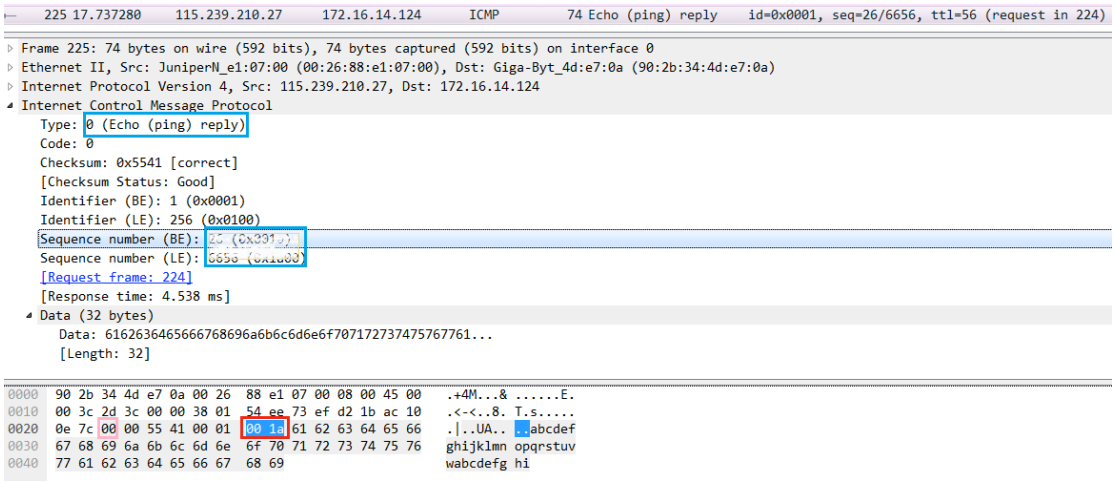
Ping 域名时：_____DNS_____

ICMP 数据包分别由哪几层协议构成？_____

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。

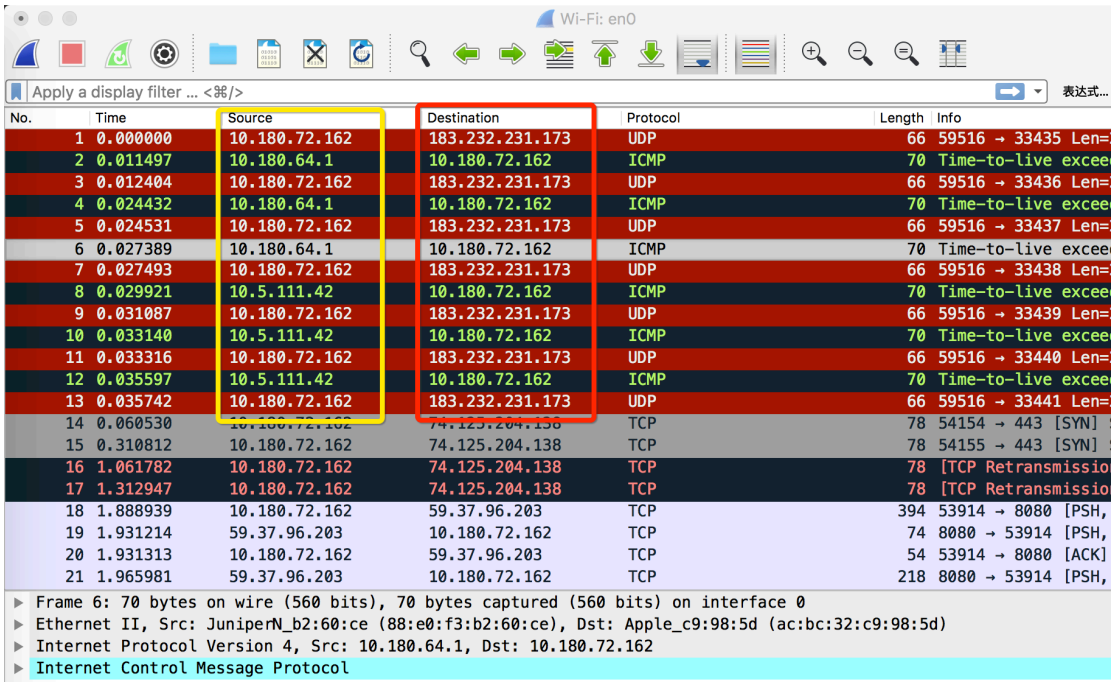


分别选择一个 ICMP 请求和响应数据包，展开最高层协议的详细内容，标出类型、序号。



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。

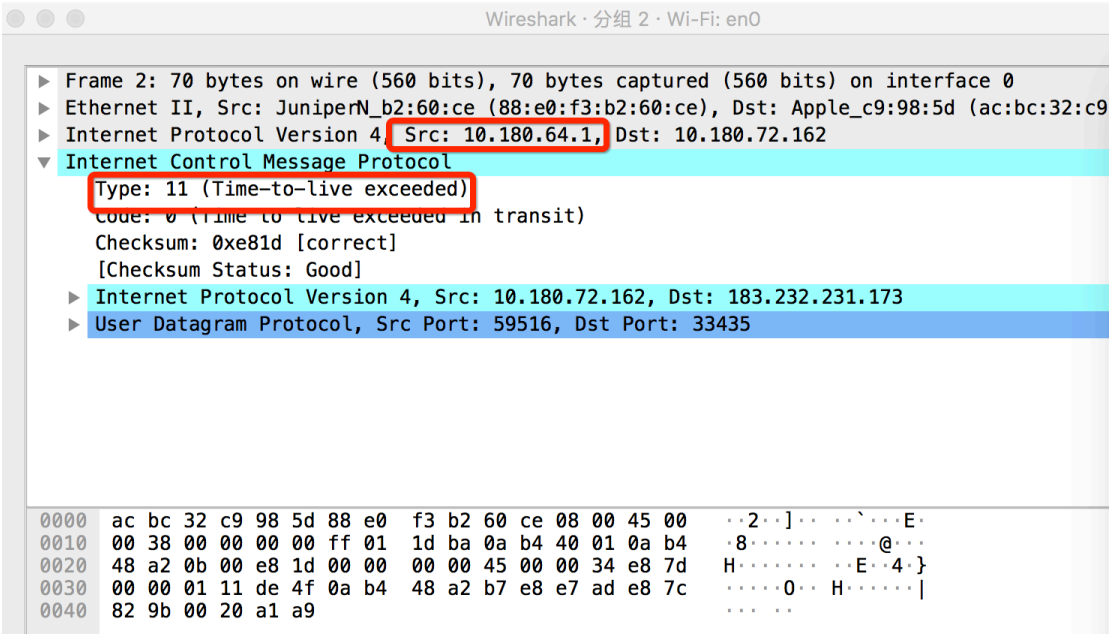
观察并记录请求包中 IP 协议层的 TTL 字段变化规律，第一个请求的 TTL 等于 253，同样 TTL 的请求连续发送了 3 个，然后每次 TTL 增加了 1，最后一个请求的 TTL 等于 255。附上截图：



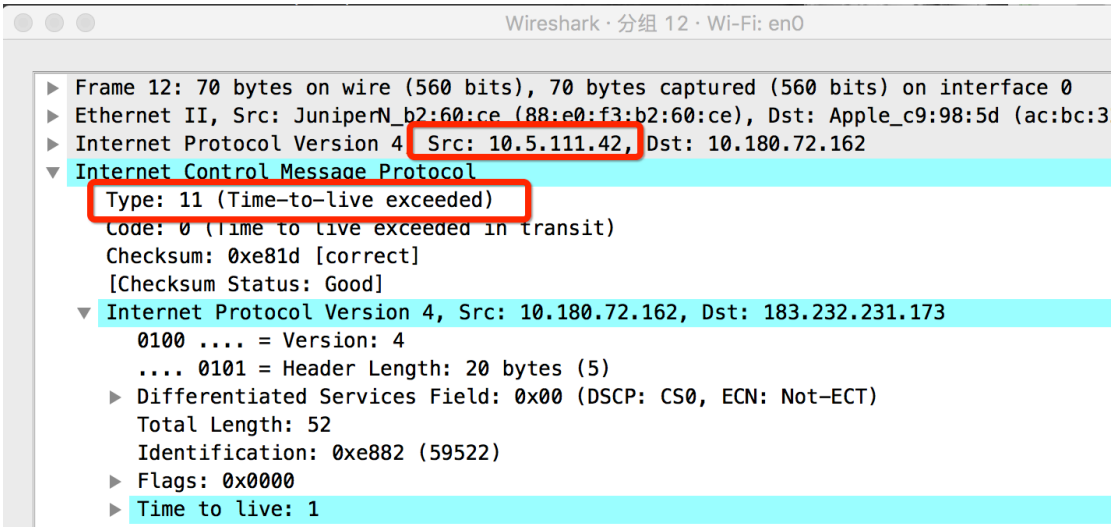
观察并记录响应包的信息，第一组响应包的发送者 IP 是：10.180.64.1，标记 ICMP 层的类型字段。最后一组响应包的发送者 IP 是：10.5.111.42，

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。

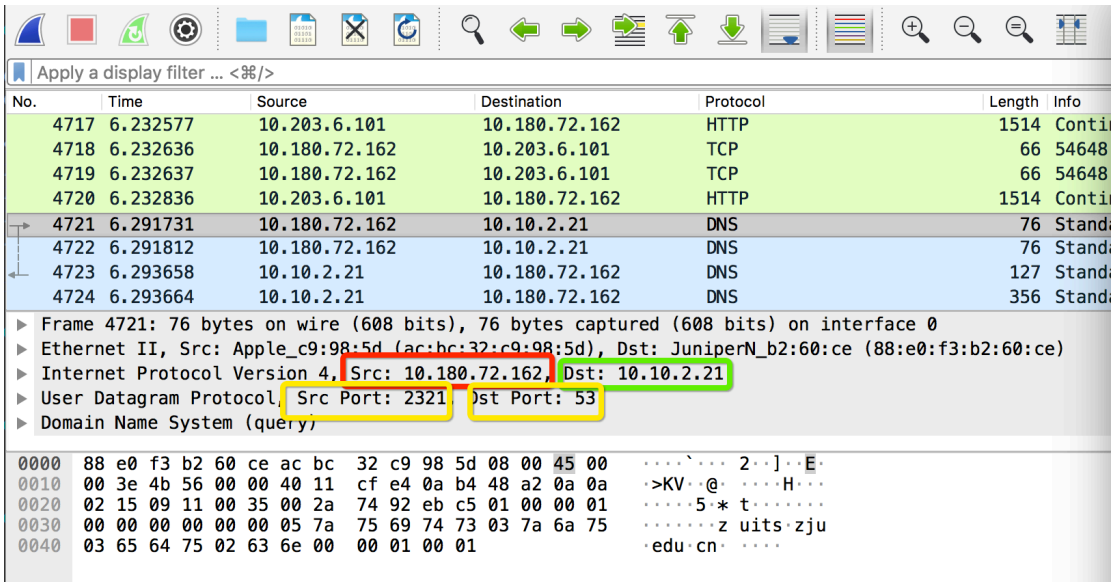
捕获到的这些最高层的协议数据包分别由哪几层协议构成？

DNS： 物理层 数据链路层 网络层 传输层 用户层

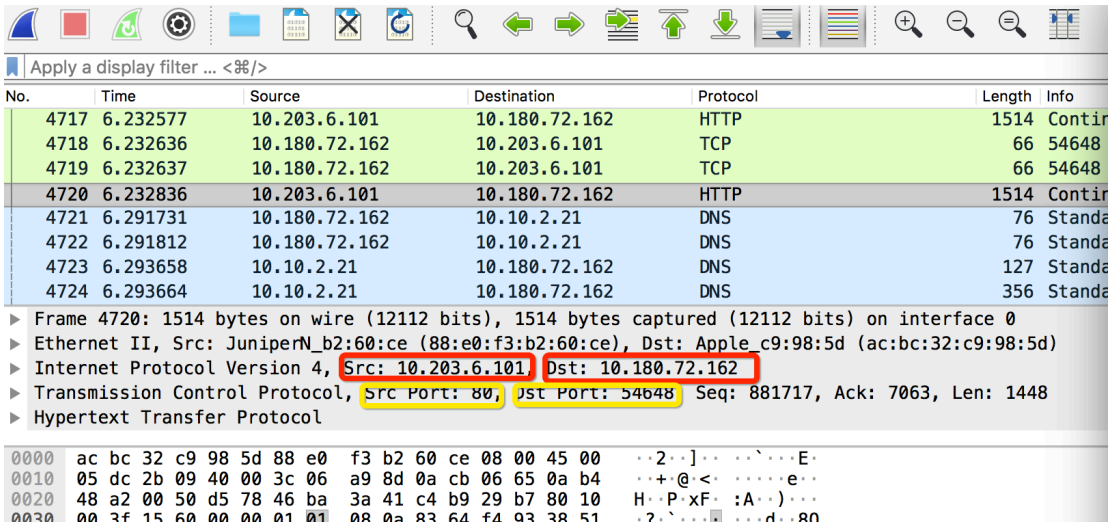
HTTP: 物理层 数据链路层 网络层 传输层

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：

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：____2____

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____

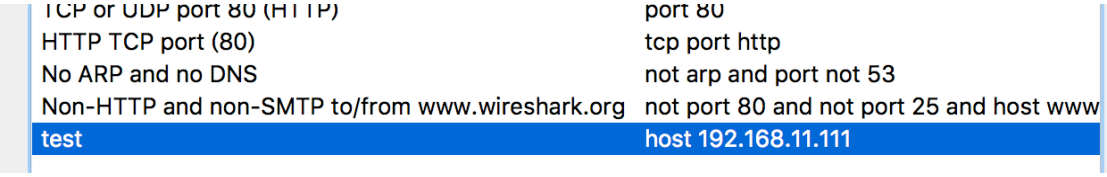
5. **选择一个 HTTP 的 TCP 流进行截图，标出请求和响应部分（最好有多个 HTTP 会话的）**

六、 实验结果分析与思考

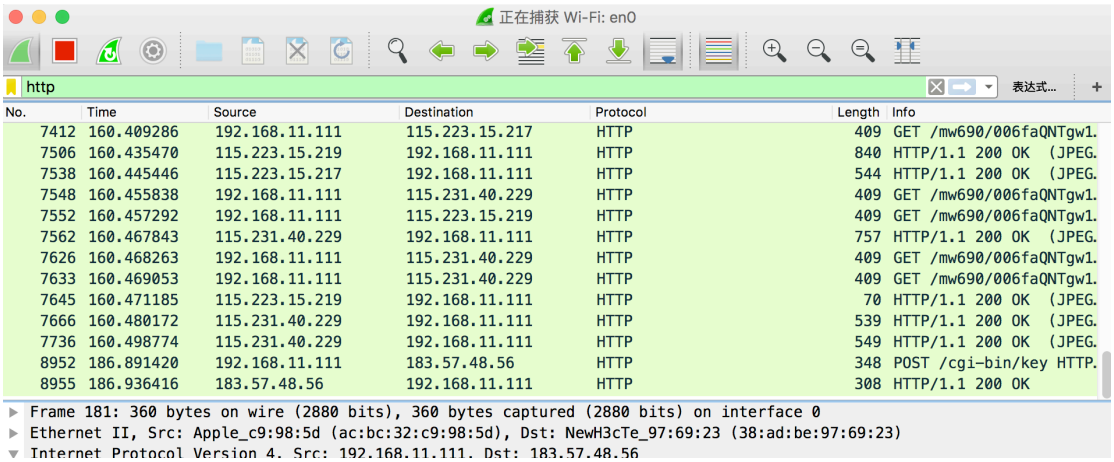
- 如果只想捕获某个特定 WEB 服务器 IP 地址相关的 HTTP 数据包，捕获过滤器应该怎么写？

  比如，捕获特定 ip 地址 192.168.11.111

  1.捕获过滤器 host 192.168.11.111



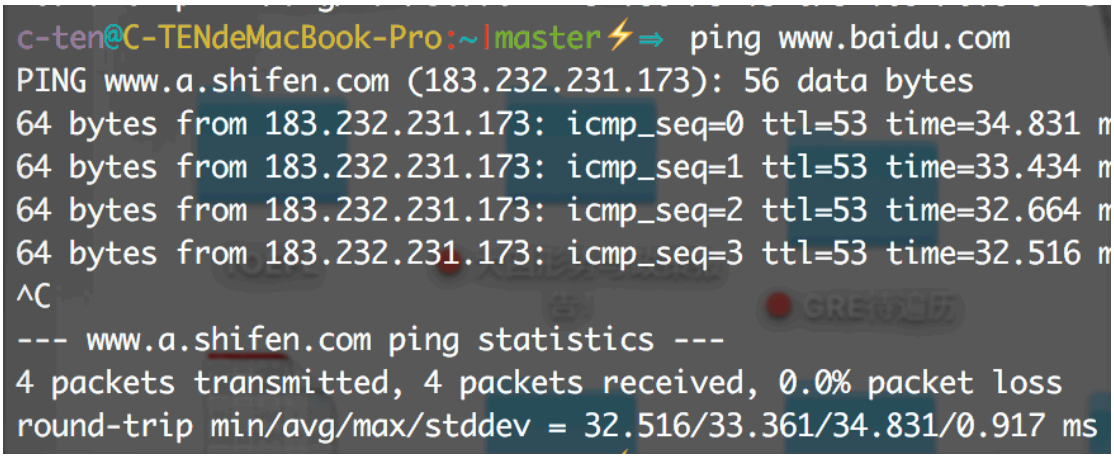
  2.显示过滤器 http



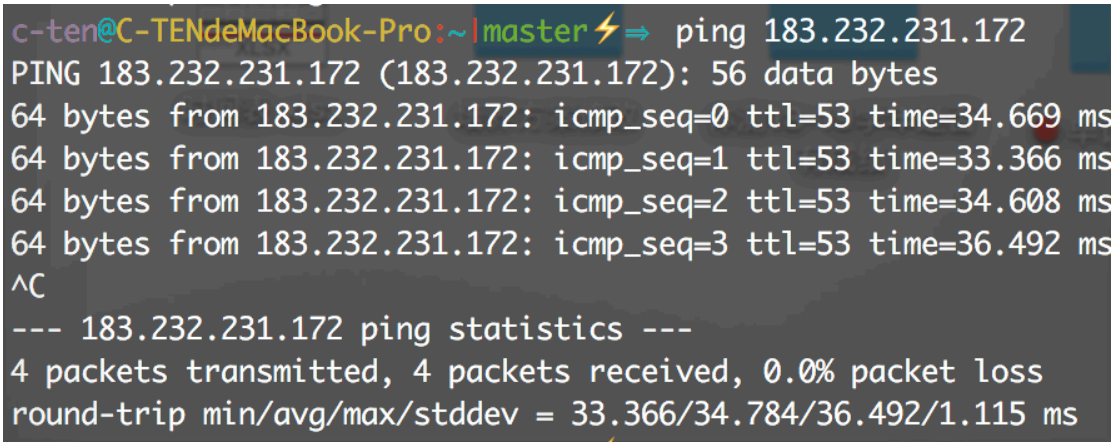
- Ping 发送的是什么类型的协议数据包？什么情况下会出现 ARP 数据包？ Ping 一个域名和 Ping 一个 IP 地址出现的数据包有什么不同？

  Ping 发送 icmp 类型的协议数据包

  Ping 域名



  Ping 一个 IP 地址



- 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？

  ICMP 类型的协议数据包

从源地址发出一个 UDP 探测包到目的地址，并将 TTL 设置为 1；

到达路由器时，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 包的源地址，IP 包的所有内容及路由器的 IP 地址；

当源地址收到该 ICMP 包时，显示这一跳路由信息；

重复 1～5，并每次设置 TTL 加 1；

直至目标地址收到探测数据包，并返回端口不可达通知（ICMP Port Unreachable）；

当源地址收到 ICMP Port Unreachable 包时停止 traceroute。

- 如何理解 TCP 连接和 HTTP 会话？他们之间存在什么关系？

  tcp 是运输层，http 是应用层。

  利用 http 协议传送数据时建立在 tcp/ip 协议的基础之上

- 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？

  Dns 用 UDP

客户端向 DNS 服务器查询域名，一般返回的内容都不超过 512 字节，用 UDP 传输即可。不用经过三次握手，这样 DNS 服务器负载更低，响应更快

  HTTP 用 TCP

  UDP 链接不可靠，网页源文件传输后容易出错

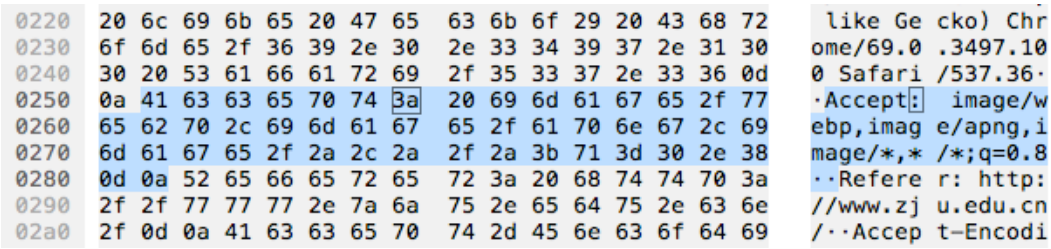
七、 讨论、心得

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：

在做 Part1 实验时，在找数据包中 dst 与 src 具体位置时，是根据对应数据串是否相等来判断的，假若数据包中有两串数据串都匹配，如何判断是哪一个，这是我实验中的一个小疑问，也许有一些偏移量的信息我忽略了，根据偏移量定位 dst 与 src 位置才对。

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：

分析数据包时仍有疑问



你对本实验安排有哪些更好的建议呢？欢迎献计献策：

做实验前，若每个部分实验有个主题，可能就更好理解了，一些任务做的好像比较零散。实验结果分析与思考的确是帮助检测运用能力，检测是否真的理解了每个步骤的含义。