

# 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：	张 溢 弛
学 院：	计算机科学与技术学院
系：	软件工程
专 业：	软件工程
学 号：	3180103772
指导教师：	邱劲松

2020 年 9 月 25 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

### ✧ 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, HTTP, ARP, DCPH, OICQ 等等

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

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。

```
> Frame 47: 293 bytes on wire (2344 bits), 293 bytes captured (2344 bits) on interface 0
✖ Ethernet II, Src: JuniperN_b2:60:52 (88:e0:f3:b2:60:52), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: JuniperN_b2:60:52 (88:e0:f3:b2:60:52)
  Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 10.181.128.1, Dst: 255.255.255.255
  > User Datagram Protocol, Src Port: 67, Dst Port: 68
  > Bootstrap Protocol (NAK)
```

IP 层的标注如下，红色表示源 IP 地址和其在数据包中对应的位置，蓝色为目标 IP 地址

```
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
  Total Length: 279
  Identification: 0x5f99 (24473)
> Flags: 0x0000
  Time to live: 255
  Protocol: UDP (17)
  Header checksum: 0xcfc6 [validation disabled]
  [Header checksum status: Unverified]
  Source: 10.181.128.1
  Destination: 255.255.255.255
> User Datagram Protocol, Src Port: 67, Dst Port: 68
> Bootstrap Protocol (NAK)
```

0010	01 17 5f 99 00 00 ff 11	cf c6 0a b5 80 01 ff ff	..C.D..
0020	ff ff 00 43 00 44 01 03	00 00 02 01 06 00 14 9a	.....HI!
0030	e2 1e 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
0040	00 00 00 00 00 00 1c 91	48 49 21 d5 00 00 00 00	.....
0050	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
0060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
0070	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
0080	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
0090	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00a0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00c0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00d0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00e0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
00f0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
0100	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	.....
0110	00 00 00 00 00 00 63 82	53 63 35 01 06 36 04 0a	.....c Sc5 6

Enthenet 层的标注如下所示

▼ Ethernet II, Src: JuniperN\_b2:60:52 (88:e0:f3:b2:60:52), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Destination: Broadcast (ff:ff:ff:ff:ff:ff)

> Source: JuniperN\_b2:60:52 (88:e0:f3:b2:60:52)

Type: IPv4 (0x0800)

> Internet Protocol Version 4, Src: 10.181.128.1, Dst: 255.255.255.255

> User Datagram Protocol, Src Port: 67, Dst Port: 68

> Bootstrap Protocol (NAK)

0000ff ff ff ff ff ff 88 e0 f3 b2 60 52 08 00 45 c0.....`R.E

001001 17 5f 99 00 00 ff 11 cf c6 0a b5 80 01 ff ff.....

0020ff ff 00 43 00 44 01 03 00 00 02 01 06 00 14 9a...C.D.....

0030e2 1e 00 00 00 00 00 00 00 00 00 00 00 00 00.....

004000 00 00 00 00 00 00 1c 91 48 49 21 d5 00 00 00 00.....HI!.....

005000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00.....

006000 00 00 00 00 00 00 00 00 00 00 00 00 00 00.....

007000 00 00 00 00 00 00 00 00 00 00 00 00 00 00.....

008000 00 00 00 00 00 00 00 00 00 00 00 00 00 00.....

009000 00 00 00 00 00 00 00 00 00 00 00 00 00 00.....

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：

arp

No.

Time

Source

Destination

Protocol

Length

Info

132.432364JuniperN\_67:28:52BroadcastARP60who has 10.181

142.432365JuniperN\_67:28:52BroadcastARP60who has 10.181

229.906601JuniperN\_67:28:52BroadcastARP60who has 10.181

3212.672049JuniperN\_67:28:52BroadcastARP60who has 10.181

4715.025692JuniperN\_67:28:52BroadcastARP60who has 10.181

9823.423207JuniperN\_67:28:52BroadcastARP60who has 10.181

10224.447358JuniperN\_67:28:52BroadcastARP60who has 10.181

13033.764503JuniperN\_67:28:52BroadcastARP60who has 10.181

13133.764504JuniperN\_67:28:52BroadcastARP60who has 10.181

> Frame 13: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

> Ethernet II, Src: JuniperN\_67:28:52 (88:e0:f3:67:28:52), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Address Resolution Protocol (request)

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 == 10.181.128.1，希望显示的 IP 地址：10.181.128.1。

截图：

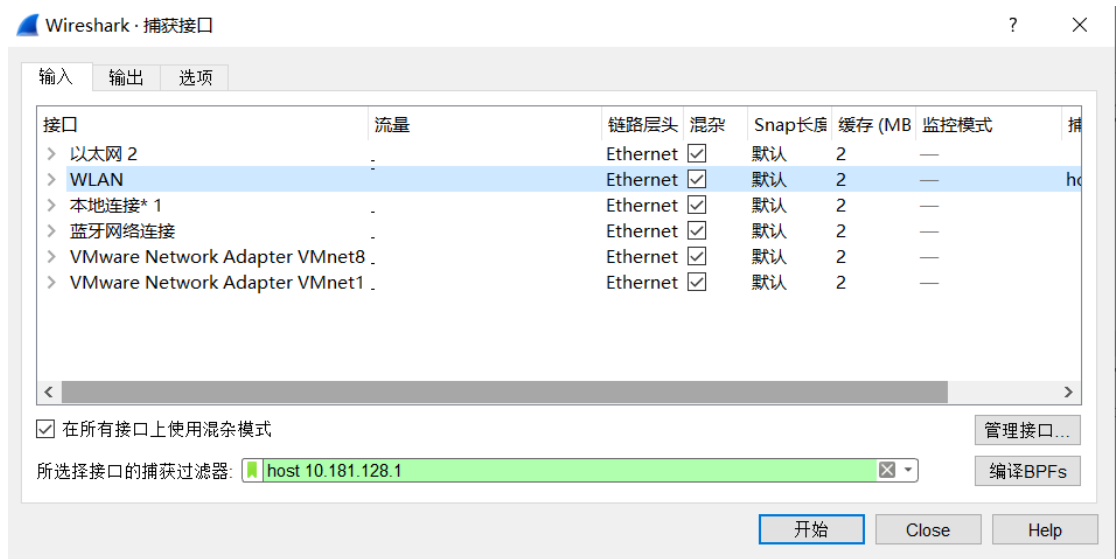
ip.addr == 10.181.128.1						
No.	Time	Source	Destination	Protocol	Length	Info
7	0.690901	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
8	0.895593	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
9	1.099614	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
10	1.202837	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
11	1.407433	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
17	7.346940	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
21	9.700631	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
28	12.672045	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
30	12.672047	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK

## 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 10.181.128.1 ， 希望捕获的 IP 地址： 10.181.128.1 。

截图：

在捕获接口中进行相应的设置



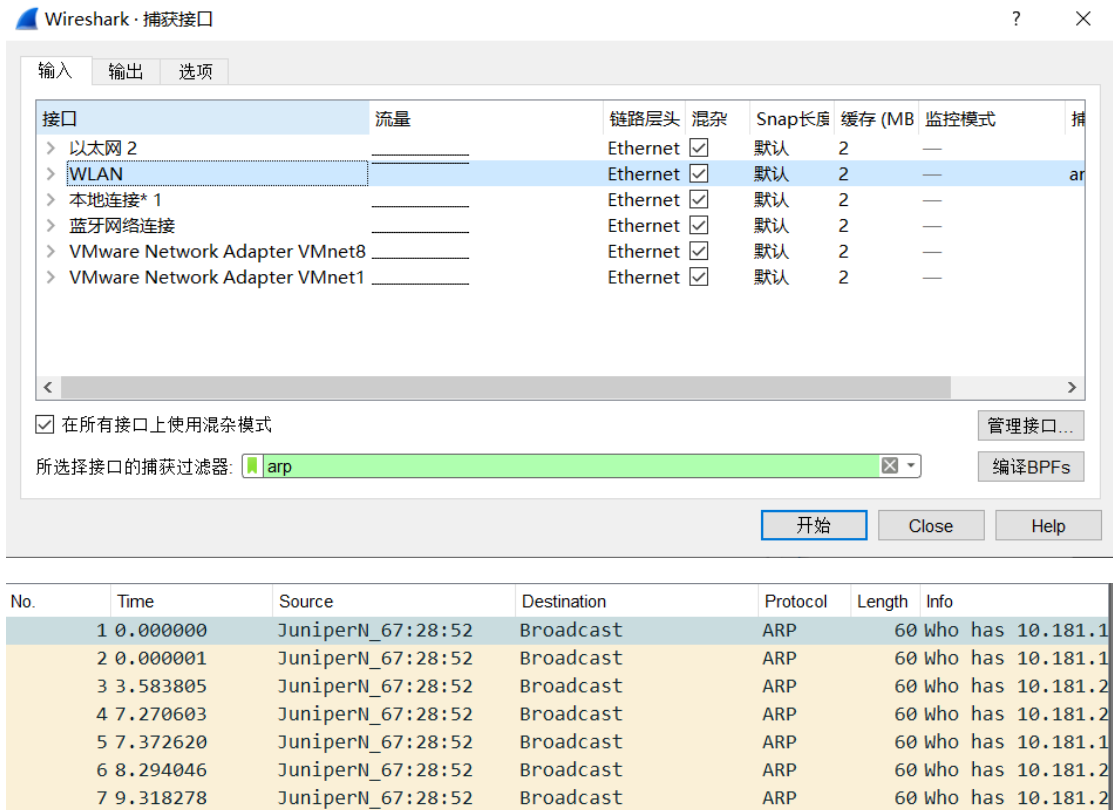
查看捕获的结果，抓到了一系列从 10.181.128.1 地址发出的包

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
2	0.716830	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
3	1.228812	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
4	1.434633	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
5	2.149887	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
6	2.355436	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
7	2.355438	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
8	3.073017	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK
9	3.175346	10.181.128.1	255.255.255.255	DHCP	293	DHCP NAK

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： arp ，希望捕获的协议类型： ARP 。

截图：先设置过滤器为 arp，再查看抓包的结果，抓到的数量比较少



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？ 5 层，Frame，Ethernet II，Internet Protocol Version4，User Datagram Protocol，Domain Name System 。 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。

```
C:\Users\74096>nslookup cc98.org
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

非权威应答:
名称:     cc98.org
Address:  192.64.119.35
```

捕获请求包和相应包的截图

8785	266.281305	10.181.248.76	10.10.0.21	DNS	68 Standard query 0x0003 AAAA cc98.org
8786	266.285180	10.10.0.21	10.181.248.76	DNS	141 Standard query response 0x0003 AAAA cc98.org SOA dns1.registrar-servers.com
9188	365.296973	10.181.248.76	10.10.0.21	DNS	94 Standard query 0x165d A tile-service.weather.microsoft.com
9189	365.301759	10.10.0.21	10.181.248.76	DNS	549 Standard query response 0x165d A tile-service.weather.microsoft.com CNAME wildcard.weather.microsoft.com...
9528	450.704605	10.181.248.76	10.10.0.21	DNS	94 Standard query 0x1de8 A msftspeechmodelsprod.azureedge.net

> Frame 8785: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0  
> Ethernet II, Src: IntelCor\_78:a1:54 (14:4f:8a:78:a1:54), Dst: JuniperH\_67:28:52 (88:e0:f3:67:28:52)  
> Internet Protocol Version 4, Src: 10.181.248.76, Dst: 10.10.0.21  
> User Datagram Protocol, Src Port: 59358, Dst Port: 53  
v Domain Name System (query)  
Transaction ID: 0x0003  
Flags: 0x0100 Standard query  
Questions: 1  
Answer RRs: 0  
Authority RRs: 0  
Additional RRs: 0  
Queries  
[Response In: 8786]

请求包的内容展开和标注

v Domain Name System (query)	
Transaction ID:	0x0003
> Flags: 0x0100 Standard query	
Questions:	1
Answer RRs:	0
Authority RRs:	0
Additional RRs:	0
v Queries	
v cc98.org: type AAAA, class IN	
Name:	cc98.org
[Name Length:	8]
[Label Count:	2]
Type:	AAAA (IPv6 Address) (28)
Class:	IN (0x0001)
[Response In: 8786]	

0000	88 e0 f3 67 28 52	14 4f 8a	78 a1 54 08 00 45 00	...	g(R·O·x·T·E·
0010	00 36 3a be 00 00 80 11	f2 d8 0a b5 f8 4c 0a 0a		·6:·	·L·
0020	00 15 e7 de 00 35 00 22	93 42 00 03 01 00 00 01		·5·"	·B·
0030	00 00 00 00 00 00 04 63	63 39 38 03 6f 72 67 00		·c	·c98·org·
0040	00 1c 00 01			·	

相应包的内容展开和标注

Domain Name System (response)		
Transaction ID: 0x0003		
Flags: 0x8180 Standard query response, No error		
Questions: 1		
Answer RRs: 0		
Authority RRs: 1		
Additional RRs: 0		
Queries		
cc98.org: type AAAA, class IN		
Name: cc98.org		
[Name Length: 8]		
[Label Count: 2]		
Type: AAAA (IPv6 Address) (28)		
Class: IN (0x0001)		
Authoritative nameservers		
[Request In: 8785]		
[Time: 0.003875000 seconds]		

0000	14 4f 8a 78 a1 54 88 e0 f3 67 28 52 08 00 45 00	·O·x·T· ·g(R·E·
0010	00 7f 86 a6 00 00 3b 11 eb a7 0a 0a 00 15 0a b5	·····;· ······
0020	f8 4c 00 35 e7 de 00 6b fe df 00 03 81 80 00 01	·L·S· ··k ······
0030	00 00 00 01 00 00 04 63 63 39 38 03 6f 72 67 00	·····c c98·org·
0040	00 1c 00 01 c0 0c 00 06 00 01 00 00 01 ac 00 3d	·····=
0050	04 64 6e 73 31 11 72 65 67 69 73 74 72 61 72 d	·dns1·re gistrar·
0060	73 65 72 76 65 72 73 03 63 6f 6d 00 0a 68 6f 73	servers· com·hos
0070	74 6d 61 73 74 65 72 c0 2b 5f 57 94 c8 00 00 a8	tmaster· + W·····
0080	c0 00 00 0e 10 00 09 3a 80 00 00 0e 11	·····: ·····

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? 以太网协议， IPv4 和 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。

ip.addr == 192.64.119.35							
No.	Time	Source	Destination	Protocol	Length	Info	
45	1.459438	10.185.246.153	192.64.119.35	ICMP	74	Echo (ping) request	id=0x0001, seq=69/17664, ttl=128 (reply in 54)
54	1.688646	192.64.119.35	10.185.246.153	ICMP	74	Echo (ping) reply	id=0x0001, seq=69/17664, ttl=46 (request in 45)
71	2.474345	10.185.246.153	192.64.119.35	ICMP	74	Echo (ping) request	id=0x0001, seq=70/17920, ttl=128 (reply in 76)
76	2.747376	192.64.119.35	10.185.246.153	ICMP	74	Echo (ping) reply	id=0x0001, seq=70/17920, ttl=46 (request in 71)
85	3.490915	10.185.246.153	192.64.119.35	ICMP	74	Echo (ping) request	id=0x0001, seq=71/18176, ttl=128 (reply in 90)
90	3.736541	192.64.119.35	10.185.246.153	ICMP	74	Echo (ping) reply	id=0x0001, seq=71/18176, ttl=46 (request in 85)
100	4.501144	10.185.246.153	192.64.119.35	ICMP	74	Echo (ping) request	id=0x0001, seq=72/18432, ttl=128 (reply in 105)
105	4.772558	192.64.119.35	10.185.246.153	ICMP	74	Echo (ping) reply	id=0x0001, seq=72/18432, ttl=46 (request in 100)



```
> Ethernet II, Src: 48:89:e7:c5:1f:16 (48:89:e7:c5:1f:16), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 48:89:e7:c5:1f:16 (48:89:e7:c5:1f:16)
    Sender IP address: 10.185.170.20
    Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
    Target IP address: 169.254.15.227

0000  ff ff ff ff ff ff 48 89 e7 c5 1f 16 08 06 00 01  .....H.....
0010  08 00 06 04 00 01 48 89 e7 c5 1f 16 0a b9 aa 14  .....H.....
0020  00 00 00 00 00 00 a9 fe 0f e3 00 00 00 00 00 00  .....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
```

所捕获的 ARP 协议的标注如上图所示

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。

在 cmd 中输入 ping cc98.org 可以查到该网站的 IP 地址是 192.64.119.35，因此可以在过滤器中输入 ip.addr == 192.64.119.35 来获得所有相关的数据包，因此实验的结果如下：

```
> Frame 92223: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: 94:29:2f:38:d8:02 (94:29:2f:38:d8:02), Dst: IntelCor_78:a1:54 (14:4f:8a:78:a1:54)
> Internet Protocol Version 4, Src: 192.64.119.35, Dst: 10.192.172.162
  Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x553c [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 31 (0x001f)
    Sequence number (LE): 7936 (0x1f00)
    [Request frame: 92052]
    [Response time: 434.881 ms]
  Data (32 bytes)

0000  14 4f 8a 78 a1 54 94 29 2f 38 d8 02 08 00 45 48  ·O·x·T·) /8····EH
0010  00 3c c3 8d 00 00 00 2e 01 da 25 c0 40 77 23 0a c0  ·<····· ··%·@w#··
0020  ac a2 00 00 55 3c 00 01 00 1f 51 62 63 64 65 66  ····U<··· ··abcdef
0030  67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76  ·ghijklmn opqrstuv
0040  77 61 62 63 64 65 66 67 68 69  ·wabcdefg hi
```

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？ 三层，以太网协议，IPv4 和 ICMP 。

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 20。附上截图：

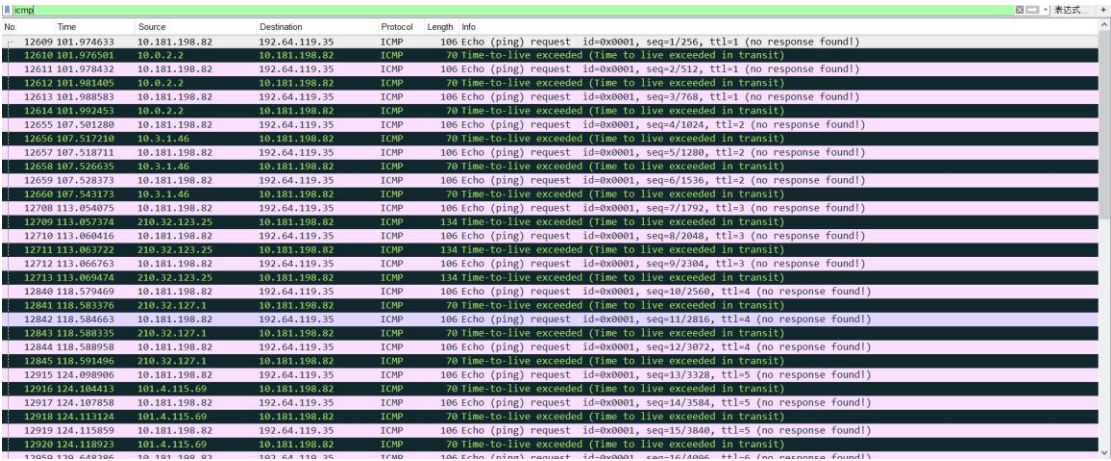
在 cmd 中输入如下命令开始捕获

```
C:\WINDOWS\system32>net start npf
NetGroup Packet Filter Driver 服务已经启动成功。

C:\WINDOWS\system32>tracert 192.64.119.35
通过最多 30 个跃点跟踪到 192.64.119.35 的路由

  1    2 ms    3 ms    4 ms    10.0.2.2
  2   16 ms   8 ms   14 ms   10.3.1.46
  3    3 ms    3 ms    2 ms   210.32.123.25
  4    4 ms    3 ms    2 ms   210.32.127.1
  5    5 ms    5 ms    3 ms   101.4.115.69
  6    7 ms    6 ms    7 ms   101.4.117.45
  7   30 ms   27 ms   30 ms   101.4.117.30
  8   33 ms   31 ms   31 ms   101.4.116.118
  9    *     34 ms   32 ms   101.4.112.69
 10   41 ms   32 ms   32 ms   101.4.113.110
 11   36 ms   34 ms   36 ms   101.4.116.78
 12   39 ms   33 ms   35 ms   101.4.117.98
 13  191 ms  189 ms  186 ms   101.4.117.170
 14  307 ms  294 ms  283 ms  te0-15-0-7-3.ccr41.lax04.atlas.cogentco.com [38.88.196.185]
 15  266 ms  231 ms  223 ms  ctl.lax04.atlas.cogentco.com [154.54.10.198]
 16    *     280 ms  284 ms  ae-3-3.ear4.LosAngeles1.Level3.net [4.69.215.129]
 17  225 ms  223 ms  215 ms  4.7.26.34
```

最终捕获到的数据如下图所示(比较多，一次截不全)，第一次做忘记保存数据包了，提交的数据包是后面重做的。



观察并记录响应包的信息，第一组响应包的发送者 IP 是： 10.0.2.2，标记 ICMP 层的类型字段。最后一组响应包的发送者 IP 是： 100.65.240.35，标记 ICMP 层的类型字段。附上截图：

第一组：

```
> Frame 12610: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: JuniperN_67:28:52 (88:e0:f3:67:28:52), Dst: IntelCor_78:a1:54 (14:4f:8a:78:a1:54)
> Internet Protocol Version 4, Src: 10.0.2.2, Dst: 10.181.198.82
< Internet Control Message Protocol
  Type: 11 (Time-to-live exceeded)
  Code: 0 (Time to live exceeded in transit)
  Checksum: 0xf4ff [correct]
  [Checksum Status: Good]
> Internet Protocol Version 4, Src: 10.181.198.82, Dst: 192.64.119.35
< Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xf7fd [unverified] [in ICMP error packet]
  [Checksum Status: Unverified]
  Identifier (BE): 1 (0x0001)
  Identifier (LE): 256 (0x0100)
  Sequence number (BE): 1 (0x0001)

0000  14 4f 8a 78 a1 54 88 e0 f3 67 28 52 08 00 45 00  ·O·X·T··· ·g(R··E·
0010  00 38 00 00 00 00 ff 01 de bb 0a 00 02 02 0a b5  ·8··········
0020  c6 52 0b 00 f4 ff 00 00 00 00 45 00 00 5c ea e5  ·R·········E··\··
0030  00 00 01 01 c6 50 0a b5 c6 52 c0 40 77 23 08 00  ·····P···R·@w#··
0040  f7 fd 00 01 00 01  ······
```

最后一组:

```
> Frame 26128: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: JuniperN_67:28:52 (88:e0:f3:67:28:52), Dst: IntelCor_78:a1:54 (14:4f:8a:78:a1:54)
> Internet Protocol Version 4, Src: 100.65.240.35, Dst: 10.181.198.82
< Internet Control Message Protocol
  Type: 11 (Time-to-live exceeded)
  Code: 0 (Time to live exceeded in transit)
  Checksum: 0xf4ff [correct]
  [Checksum Status: Good]
> Internet Protocol Version 4, Src: 10.181.198.82, Dst: 192.64.119.35
< Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xf7c5 [unverified] [in ICMP error packet]
  [Checksum Status: Unverified]
  Identifier (BE): 1 (0x0001)
  Identifier (LE): 256 (0x0100)
  Sequence number (BE): 57 (0x0039)

0000  14 4f 8a 78 a1 54 88 e0 f3 67 28 52 08 00 45 48  ·O·X·T··· ·g(R··EH
0010  00 38 05 93 40 00 ee 01 61 7d 64 41 f0 23 0a b5  ·8··@··· a}dA·#··
0020  c6 52 0b 00 f4 ff 00 00 00 00 45 28 00 5c eb 1d  ·R·········E(·\··
0030  00 00 01 01 c5 f0 0a b5 c6 52 c0 40 77 23 08 00  ······R·@w#··
0040  f7 c5 00 01 00 39  ······9
```

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: 以太网协议, IPv4 协议, UDP 协议, DNS 协议

HTTP: 以太网协议, IPv4 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）

## DNS 协议

249	2.101984	10.185.246.153	10.10.0.21	DNS	74 Standard query 0x55f3 A tel.zju.edu.cn
250	2.102171	10.185.246.153	10.10.0.21	DNS	74 Standard query 0x57ba AAAA tel.zju.edu.cn
251	2.105989	10.10.0.21	10.185.246.153	DNS	125 Standard query response 0x55f3 A tel.zju.edu.cn A 10.203.9.35 NS dns1.zju.edu.cn A 10.10.0.9
> Frame 249: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0					
> Ethernet II, Src: IntelCor_78:a1:54 (14:4f:8a:78:a1:54), Dst: 84:46:fe:26:2f:2e (84:46:fe:26:2f:2e)					
▼ Internet Protocol Version 4, Src: 10.185.246.153, Dst: 10.10.0.21					
0100 .... = Version: 4					
.... 0101 = Header Length: 20 bytes (5)					
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)					
Total Length: 60					
Identification: 0x581a (22554)					
> Flags: 0x0000					
Time to live: 128					
Protocol: UDP (17)					
Header checksum: 0xd725 [validation disabled]					
[Header checksum status: Unverified]					
Source: 10.185.246.153					
Destination: 10.10.0.21					
> User Datagram Protocol, Src Port: 56556, Dst Port: 53					
> Domain Name System (query)					

## HTTP 协议

256	2.126643	10.185.246.153	10.203.9.35	HTTP	570 POST /network.jsp HTTP/1.1 (application/x-www-form-urlencoded)
258	2.138715	10.203.9.35	10.185.246.153	HTTP	728 HTTP/1.1 200 (text/html)
318	2.217205	10.185.246.153	10.203.6.126	HTTP	141 GET /_visitcount?siteId=590&type=1&columnId=32642 HTTP/1.1
322	2.226255	10.203.6.126	10.185.246.153	HTTP	224 HTTP/1.1 200 OK

> Frame 256: 570 bytes on wire (4560 bits), 570 bytes captured (4560 bits) on interface 0					
> Ethernet II, Src: IntelCor_78:a1:54 (14:4f:8a:78:a1:54), Dst: 84:46:fe:26:2f:2e (84:46:fe:26:2f:2e)					
▼ Internet Protocol Version 4, Src: 10.185.246.153, Dst: 10.203.9.35					
0100 .... = Version: 4					
.... 0101 = Header Length: 20 bytes (5)					
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)					
Total Length: 556					
Identification: 0xfd79 (64889)					
> Flags: 0x4000, Don't fragment					
Time to live: 128					
Protocol: TCP (6)					
Header checksum: 0xe611 [validation disabled]					
[Header checksum status: Unverified]					
Source: 10.185.246.153					
Destination: 10.203.9.35					
> Transmission Control Protocol, Src Port: 58764, Dst Port: 80, Seq: 1, Ack: 1, Len: 516					
> Hypertext Transfer Protocol					
> HTML Form URL Encoded: application/x-www-form-urlencoded					

## 2. 为了打开网页，浏览器查询了哪些相关的域名？

域 名 列 表 ： iczu.zju.edu.cn, grs.zju.edu.cn, 123.zju.edu.cn, my.zju.edu.cn,  
map.zju.edu.cn, piclib.zju.edu.cn, person.zju.edu.cn, rwsz.zju.edu.cn, rd.zju.edu.cn,  
www.arv.zju.edu.cn, ugrs.zju.edu.cn, www.tyys.zju.edu.cn, www.zdxqn.zju.edu.cn,  
www.zdzsc.zju.edu.cn, zuits.zju.edu.cn, xwfw.zju.edu.cn, zuaa.zju.edu.cn,  
www.libweb.zju.edu.cn, courses.zju.edu.cn, ac.zju.edu.cn, helps.zju.edu.cn,  
tel.zju.edu.cn 等等

## 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: 8

tcp.stream eq 8						
No.	Time	Source	Destination	Protocol	Length	Info
336	2.404471	10.185.246.153	172.217.27.138	TCP	66	58766 → 443 [SYN
371	3.404619	10.185.246.153	172.217.27.138	TCP	66	[TCP Retransmiss
454	5.404780	10.185.246.153	172.217.27.138	TCP	66	[TCP Retransmiss

tcp.stream eq 9

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: 3 次

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）：

```

GET / HTTP/1.1
Host: www.zju.edu.cn
Connection: keep-alive
Upgrade-Insecure-Requests: 1
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/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh,zh-CN;q=0.9,en;q=0.8,ja;q=0.7
Cookie: ga-GA1.3.1262959460.1600333556; ?tUser=X78%2Zaccount%2X23A%2231801377%2Z2X2%21d%2X3A%2240%2Z2%22tenant_id%2X3A11%2X70; _gid-GA1.3.1756244369.1600443413; token=597ad862757158e4ec1c8f747ebda6a88e048eb3fddaf65df04a823b943a32X3A7B13XA0%3B3XA0%3A22_token%2X3B13XA13B%3A5753A3A22_ythbc10131UZ11m15Inm5C16KpXVC9.eYjH2Vndvd601jo1mE4MDWwz3M15ImvTvlw1s1oj1yZVjY2HQHfXLMvbs15Imv4C16TmYwMDU20jgHs4b16gn4w5eXU1l1jo1ZGVmYVsdC15Imyb2x1cy16w31y21jcnm25 Ie15mvdwZ910131jcnvhd042v2r1jo1jYmYgCwM105s4w7Cv0dWv1I1mL1Zhy4x0Bw0q1jo15a2m55F1w1ZG1Zc4eHv9VpU11jo15a2m55F1w1a0q1jYtsw1Xk2Zhdw01jo1M3Y9X5b0cnv1dmyZ2M101C1XWvW2Z923161mYwYAJN1Zm1Z2c3pVjYjHJ1J3D0wGZ723YU311w1c0hVwU11m13wYvMDW04W2M51511YkxvW11jo15ybsgr15ybb11w13V1j0mYjYHNDAS1m1Bmf49PZC21T0YtYfQ.gvAgV15u4wD051B1Q67YHmKqVH75q7G15G36K2K38Z3B7D; pfo=60s52FKOK13x51y94etqtUQ0xA4Z7F4u4Dp0XK6rvvK0K3D; route=42b6ca244cF9f819080162c27232518; Hm_lvt_fe30bbc1ee45421ec1679d1b8df8453=1600511501; J5E55IONID=1D5192B2745CF5D4700C2690E53DC
Hm_lvt_fe30bbc1ee45421ec1679d1b8df8453=1600511501; J5E55IONID=1D5192B2745CF5D4700C2690E53DC

HTTP/1.1 200 OK
Server: nginx
Date: Sat, 19 Sep 2020 10:41:26 GMT
Content-Type: text/html
Content-Length: 12790
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Frame-Options: SAMEORIGIN
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip

.....CV5S.....A1S..H.....S..c@P.....H.....J.....g.....t.....f.....O.....M.....L.....V.A.A>0S.C.h.(.h)?.....PT.....2H.9.....CM2.....&..T.....QZ.....1.9*.....F..84.G6.<.....6.Cc.
7.%W.....c.g..yA..f.....
.....N.....1.....I..A.y.....0P.....9J.....w0.....pNjD.....
W'.....H].....N.....D.....X..av.6..uBEE.....5..Ky.....9.....D.....H'f.2.U.U.....D.....F..7F..Q8.....[.....]h..hO.....u.i..6.g>.....7uu.M.....f.....9YX
.....so.....JZ.....C.....I.D4.....h.YH%5Y.....g.dg.....[.....]Rp.....Ch.....[.....]bv.v.....b.7'54B.7G.....k
.....Y.....0.....V.....c.....d(qc.....2.....61.....C.....8.....T.....C.....C.....vQbG6.p.X.....[.....]H.....2.....[.....]$.59.3.....F.....A.....I.....
.....8.....B.....N.....M.....S.....S.....F.....E.....M.....[.....]
.....5Y.....XBTN.....yF.....D..Xg.8.5L.I.V.....M.....V.....U.....S.....c.D.....[.....]t.j.Oy.....V.Z.....tT7.N
.....M2A1H1'.....D.....S.....4.....3.....J.....R.....
.....I.....7.b/h.....uI05.....c.ZP.HB.....Q.....[.....]T.....S.....g.....X.....X.....o.....q.....[.....]Zt.L'.....1.eUf'.3O.....[.....]H.z7HB.....wYf.0.....[.....].
.....a'4..[91
.....y.....j..d59Q.....
.....iW.....0.....GET /_visitcount?siteId=590&type=1&columnId=32642 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/57.36
Accept: image/avif,image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://www.zju.edu.cn/
Accept-Encoding: gzip, deflate
Accept-Language: zh,zh-CN;q=0.9,en;q=0.8,ja;q=0.7
Cookie: ga-GA1.3.1262959460.1600333556; ?tUser=X78%2Zaccount%2X23A%2231801377%2Z2X2%221d%2X3A%2240%2Z2%22tenant_id%2X3A11%2X70; _gid-GA1.3.1756244369.1600443413; token=597ad862757158e4ec1c8f747ebda6a88e048eb3fddaf65df04a823b943a32X3A7B13XA0%3B3XA0%3A22_token%2X3B13XA13B%3A5753A3A22_ythbc10131UZ11m15Inm5C16KpXVC9.eYjH2Vndvd601jo1mE4MDWwz3M15ImvTvlw1s1oj1yZVjY2HQHfXLMvbs15Imv4C16TmYwMDU20jgHs4b16gn4w5eXU1l1jo1ZGVmYVsdC15Imyb2x1cy16w31y21jcnm25 Ie15mvdwZ910131jcnvhd042v2r1jo1jYmYgCwM105s4w7Cv0dWv1I1mL1Zhy4x0Bw0q1jo15a2m55F1w1ZG1Zc4eHv9VpU11jo15a2m55F1w1a0q1jYtsw1Xk2Zhdw01jo1M3Y9X5b0cnv1dmyZ2M101C1XWvW2Z923161mYwYAJN1Zm1Z2c3pVjYjHJ1J3D0wGZ723YU311w1c0hVwU11m13wYvMDW04W2M51511YkxvW11jo15ybsgr15ybb11w13V1j0mYjYHNDAS1m1Bmf49PZC21T0YtYfQ.gvAgV15u4wD051B1Q67YHmKqVH75q7G15G36K2K38Z3B7D; pfo=60s52FKOK13x51y94etqtUQ0xA4Z7F4u4Dp0XK6rvvK0K3D; route=42b6ca244cF9f819080162c27232518; Hm_lvt_fe30bbc1ee45421ec1679d1b8df8453=1600511501; J5E55IONID=1D5192B2745CF5D4700C2690E53DC; Hm_lvt_fe30bbc1ee45421ec1679d1b8df8453=1600511703

```

3

- (i) 首先从源地址发出 UDP 包到目的地址, 设置 TTL 为 1, 每次到达路由器的時候 TTL 减小 1, 当 TTL 变成 0 的时候数据包被丢弃, 路由器向源

地址发回一个超时通知 (TCPM Time Exceeded Message), 包含了发送包的内容和路由器的 IP 地址

(ii) 当源地址收到这个超时通知的时候, 会显示这个路由信息

(iii) 每次发送的时候将设置的 TTL 增加 1, 按照上述步骤进行, 直到送达最终的目标地址

- 如何理解 TCP 连接和 HTTP 会话? 他们之间存在什么关系?

- (1) HTTP 的长连接和短连接实际上就是 TCP 的长连接和短连接, HTTP 属于应用层的网络协议, 在传输层使用 TCP 协议而在网络层使用 IP 协议。
- (2) IP 协议主要解决网络路由和寻址的问题。
- (3) TCP 协议主要解决如何在 IP 层之上传递数据包, 使得目标地址完整有序地收到源地址发出的所有信息。TCP 协议非常可靠, 并且面向连接。

- 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) 因为使用基于 UDP 的 DNS 协议只要一个请求和一个应答, 而基于 TCP 的 DNS 协议需要三次握手, 发送数据和应答, 四次挥手等过程, 相比于 UDP 协议需要更久的时间和更多的网络资源。
- (2) DNS 包的体量较小, 使用 UDP 协议的时候不需要考虑分包, 如果丢包就会失去全部内容, 如果收到了包就会收到所有内容, 哪怕丢包了也可以重新发送一次, 因此适合使用 UDP 协议传输。
- (3) 而 HTTP 包需要传输的数据量比较大, 需要分包, 需要更严谨, 更安全的 TCP 协议来进行数据传输, 可以保证数据传输的稳定和安全性。

## 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：

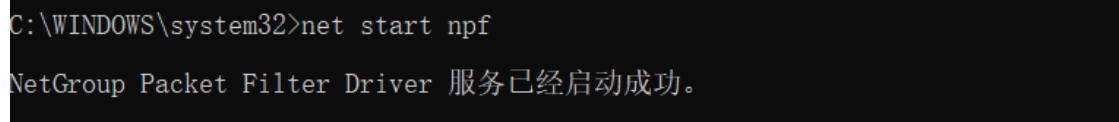
问题 1：wireshark 软件的工作原理是什么？

问题 2：在保存的 pcap 数据中，有一些数据包特别大，是不是因为没有及时停止抓包，导致抓到了一系列对实验没有用的数据包？

问题 3：为什么在打开网页的过程中浏览器回访问这么多相关的域名？因为是第一周做的实验因此对这个问题不是很理解。

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：

1.刚开始做实验的时候发现打开 wireshark 软件后什么也没有捕捉到，查阅资料后得知是因为没有启动 NPF 服务，因此在 Win10 系统下需要以管理员身份运行 cmd，输入命令 net start npf 来启动对应的服务，之后打开 Wireshark 软件后才能正常使用，截图如下：



```
C:\WINDOWS\system32>net start npf
NetGroup Packet Filter Driver 服务已经启动成功。
```

2.每次重新打开 Wireshark 软件的时候都需要重复 1 中的操作，否则会抓不到包，而过滤器的过滤范围比较大的时候可能在刚开始的一段时间内还抓不到对应的包，比如任务 1 中的第 5 和第 6 步，此时需要耐心等待

3.IP 地址指的是 IP 层中的源地址和目标地址，因此找 IP 地址不能去别的协议层找

4.第二三部分实验中要先打开 wireshark 然后在 cmd 中输入命令，这样才能捕捉到对应的数据包，然后可以使用过滤器来筛选出需要的包，常见的筛选方法是根据协议类型来筛选或者根据已知的 IP 地址来筛选



5.filter 中区分大小写，虽然协议平时一般都用大写来表示，但是在这一栏中所有内容都要小写表示，另外可以用 **and** 等逻辑关系词来连接不同的查询条件

6.每次做完实验记得保存 **pcap** 数据文件，不然就需要推倒重做一次，这种情况在实验中出现了好几次，导致我的工作量大幅度提高

你对本实验安排有哪些更好的建议呢？欢迎献计献策：

实验的难度不大，但是会遇到各种各样的问题，**Wireshark** 软件的入门存在一定的难度和门槛，建议在实验报告的前几个部分增加更多的教学内容，方便学生上手，实验量略大，需要比较长的时间完成实验