

# 浙江大学

## 本科实验报告

课程名称： 计算机网络

姓 名： 秦嘉俊

学 院： 计算机学院

系：

专 业： 计算机科学与技术

学 号： 3210106182

指导教师： 许海涛

2023 年 10 月 1 日

# 浙江大学实验报告

课程名称： 计算机网络 实验类型： 操作实验  
实验项目名称： Wireshark 软件初探和常见网络命令的使用  
学生姓名： 秦嘉俊 专业： 计算机科学与技术 学号： 3210106182  
同组学生姓名： \_\_\_\_\_ 指导老师： \_\_\_\_\_  
实验地点： 计算机网络实验室 实验日期： 2023 年 10 月 1 日

## 一、 实验目的和要求：

- 初步了解 Wireshark 软件的界面和功能
- 熟悉各类常用网络命令的使用

## 二、 实验内容和原理

- 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、Linux 版本和 Mac 版本，可以免费从网上下载
- 初步掌握网络协议分析软件 Wireshark 的使用，学会配置过滤器
- 根据要求配置 Wireshark，捕获某一类协议的数据包
- 在 PC 机上熟悉常用网络命令的功能和用法：Ping.exe, Netstat.exe, Telnet.exe, Tracert.exe, Arp.exe, Ipconfig.exe, Net.exe, Route.exe, Nslookup.exe
- 利用 Wireshark 软件捕捉上述部分命令产生的数据包

## 三、 主要仪器设备

- 联网的 PC 机
- 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
- 配置网络包捕获软件，只捕获特定类型的包
- 在 Windows 命令行方式下，执行适当的命令，完成以下功能(请以管理员身份打开命令行):
  1. 测试到特定地址的连通性、数据包延迟时间
  2. 显示本机的网卡物理地址、IP 地址
  3. 显示本机的默认网关地址、DNS 服务器地址
  4. 显示本机记录的局域网内其它机器 IP 地址与其物理地址的对照表

5. 显示从本机到达一个特定地址的路由
6. 显示某一个域名的 IP 地址
7. 显示已经与本机建立 TCP 连接的端口、IP 地址、连接状态等信息
8. 显示本机的路由表信息，并手工添加一个路由
9. 显示本机的网络映射连接
10. 显示局域网内某台机器的共享资源
11. 使用 telnet 连接 WEB 服务器的端口，输入（<cr>表示回车）获得该网站的主页内容：

GET / HTTP/1.1

Host: www.baidu.com

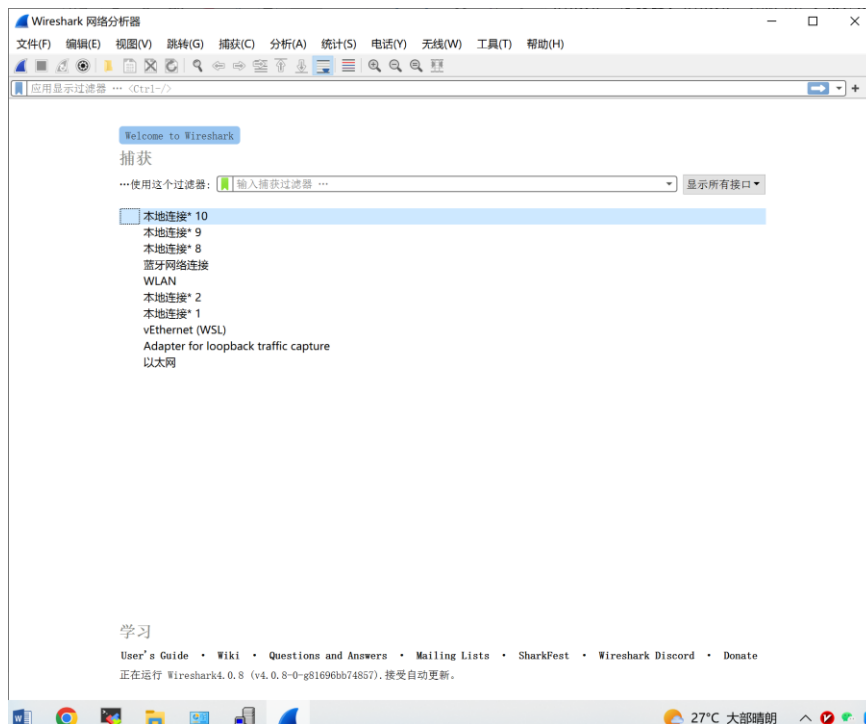
- 利用 Wireshark 实时观察在执行上述命令时，哪些命令会额外产生数据包，并记录这些数据包的种类。

## 五、实验数据记录和处理运行 Wireshark 软件，界面是由哪几个部分构成？各有什么作用？

Wireshark 软件界面主要由下面几个部分构成：

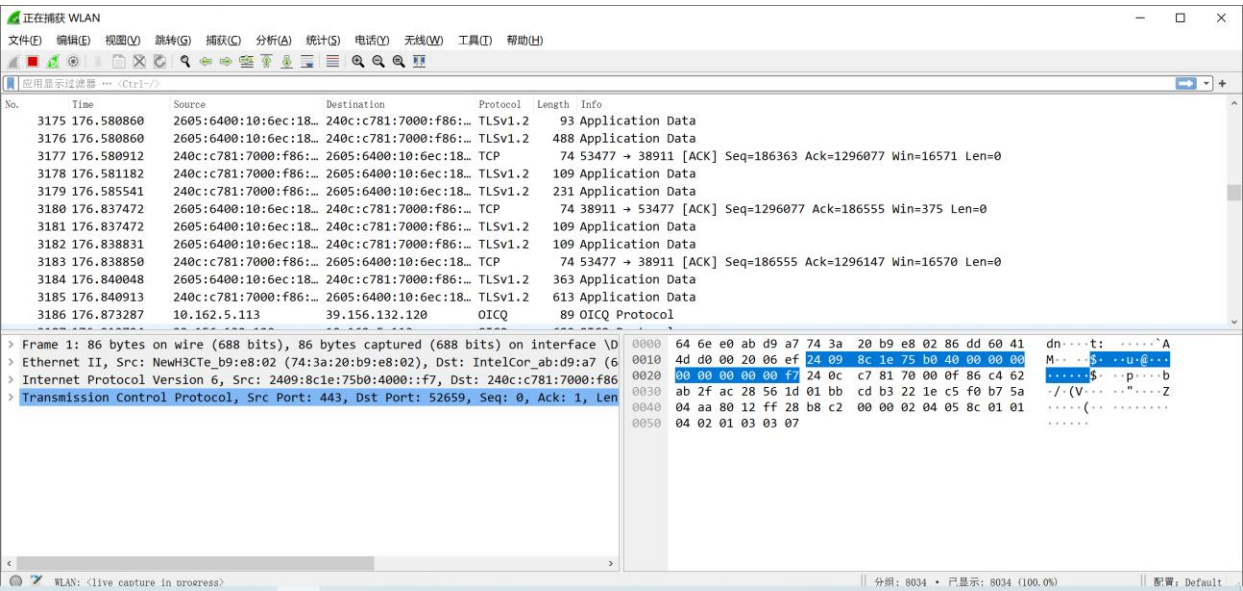
初始界面如下：

主界面上有打开历史捕获流量文件和新建捕获两个部分。上方主工具栏有捕获、暂停、重新开始、捕获选项、打开、保存、关闭、重新加载、查找、一系列跳转选项，工具栏下面是过滤器工具栏，以及一些调整视图的选项。下面的大部分空白区域是主界面，可以选择筛选特定类型的数据包。（如 WIFI 相关的数据包）



随后开始捕获，看到界面如下。这里上方的工具栏保持不变，但主界面显示捕获或加载

的数据包列表以及数据包的详细信息。主界面底部是数据包详情，显示了所选数据包的详细信息，包括协议分解、十六进制数据以及其他相关信息。**Wireshark** 窗口底部是状态栏，通常显示有关捕获进程、过滤器状态、数据包数量等信息的状态栏。



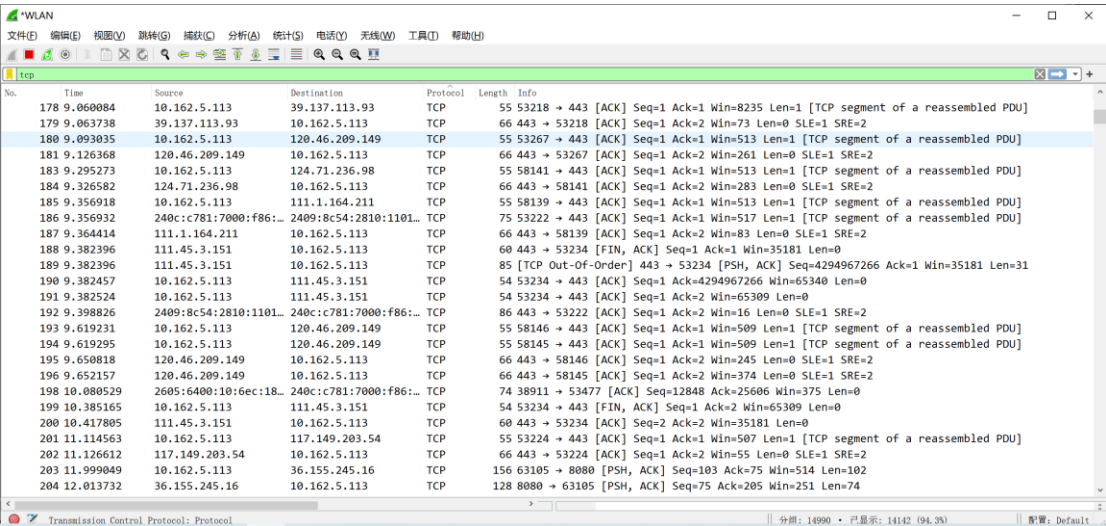
## ● 开始捕获网络数据包，你看到了什么？有哪些协议？

看到一系列网络数据包的信息，这些信息包括：源地址和目标地址、时间戳、协议、数据包大小、数据包信息。捕获截图见上。

其中我们看到的协议有 TLSv1.2, TCP, OICQ。

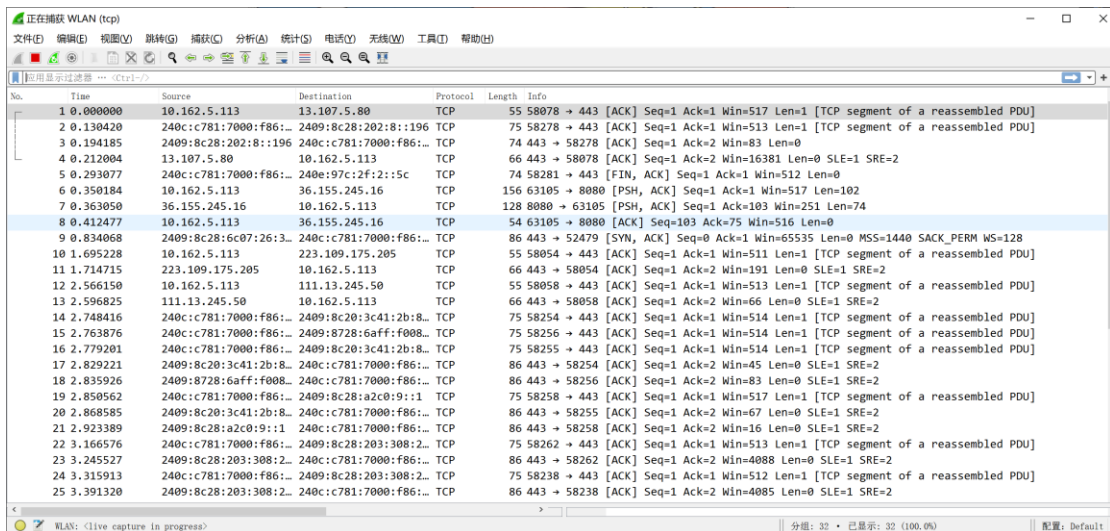
## ● 配置应用显示过滤器，让界面只显示某一协议类型的数据包。

这里我们设置显示过滤器为 TCP，可以看到此时显示的都是 TCP 协议类型的数据包，



- 配置捕获过滤器，只捕获某类协议的数据包。

这里我们在捕获过滤器中设置 `tcp.port==80`，即可捕获到 TCP 协议类型的数据包。



- 利用 Ping.exe, Netstat.exe, Telnet.exe, Tracert.exe, Arp.exe, Ipconfig.exe, Net.exe, Route.exe 命令完成在实验步骤中列举的 11 个功能。

#### 1. 测试到特定地址的连通性、数据包延迟时间

```
PS C:\Windows\system32> ping 10.12.86.210

正在 Ping 10.12.86.210 具有 32 字节的数据:
来自 10.12.86.210 的回复: 字节=32 时间=2ms TTL=59
来自 10.12.86.210 的回复: 字节=32 时间=3ms TTL=59
来自 10.12.86.210 的回复: 字节=32 时间=2ms TTL=59
来自 10.12.86.210 的回复: 字节=32 时间=3ms TTL=59

10.12.86.210 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 = 2ms, 最长 = 3ms, 平均 = 2ms
PS C:\Windows\system32>
```

这里我们输入 `ping 10.12.86.210` 测试到这个地址的连通性和数据包延迟时间，结果如上图。这里可以看到发送的 ICMP 包得到回复，说明联通；平均延时时间为 2ms。

#### 2. 显示本机的网卡物理地址、IP 地址

在终端输入 `IPCONFIG /ALL`，即可查看本机的网卡物理地址和 IP 地址，如下图所示。

```
无线局域网适配器 WLAN:

    连接特定的 DNS 后缀 . . . . . : 
    描述 . . . . . : Intel(R) Wi-Fi 6 AX201 160MHz
    物理地址. . . . . : 64-6E-E0-AB-D9-A7
    DHCP 已启用 . . . . . : 是
    自动配置已启用. . . . . : 是
    IPv6 地址 . . . . . : 240c:c781:7000:f86:592b:5dc:ed6c:db09(首选)
    临时 IPv6 地址. . . . . : 240c:c781:7000:f86:c462:ab2f:ac28:561d(首选)
    本地连接 IP 地址. . . . . : 10.10.0.21
    IPv4 地址 . . . . . : 10.10.0.21(首选)
    子网掩码 255.255.255.0
    默认网关. . . . . : 10.10.0.1
    获得租约的时间 . . . . . : 2023年10月1日 15:51:40
    租约过期的时间 . . . . . : 2023年10月2日 15:51:40
    默认网关. . . . . : fe80::763a:20ff:feb9:e802%14
    DHCP 服务器 . . . . . : 10.10.0.1
    DHCPv6 IAID . . . . . : 107245280
    DHCPv6 客户端 DUID . . . . . : 00-01-00-01-2C-34-21-8C-64-6E-E0-AB-D9-A7
    DNS 服务器 . . . . . : 10.10.0.21
    NetBIOS 上的 NetBIOS . . . . . : 已启用
```

#### 3. 显示本机的默认网关地址、DNS 服务器地址

指令同 2，结果如图。

```
无线局域网适配器 WLAN:
    连接特定的 DNS 后缀 . . . . . : 
    描述. . . . . : Intel(R) Wi-Fi 6 AX201 160MHz
    物理地址. . . . . : 64-6E-E0-AB-D9-A7
    DHCP 已启用 . . . . . : 是
    自动配置已启用. . . . . : 是
    IPv6 地址. . . . . : 240c:c781:7000:f86:592b:5dc:ed6c:db09(首选)
    临时 IPv6 地址. . . . . : 240c:c781:7000:f86:c462:ab2f:ac28:561d(首选)
    本地连接 IPv6 地址. . . . . : fe80::fc2d:64dd:7646:2072%14(首选)
    IPv4 地址. . . . . : 10.162.5.113(首选)
    子网掩码. . . . . : 255.255.0.0
    获得租约的时间. . . . . : 2023年10月1日 15:51:40
    租约过期的时间. . . . . : 2023年10月2日 15:51:40
    默认网关. . . . . : fe80::763a:20ff:feb9:e802%14
    DHCP 服务器. . . . . : 10.162.0.1
    DHCPv6 IAID . . . . . : 107245280
    DHCPv6 客户端 DUID . . . . . : 00-01-00-01-2C-34-21-8C-64-6E-E0-AB-D9-A7
    DNS 服务器 . . . . . : 10.10.0.21
    TCP/IP 上的 NetBIOS . . . . . : 已启用
```

#### 4. 显示本机记录的局域网内其它机器 IP 地址与其物理地址的对照表

在终端输入 `arp -a` 命令,即可查看本机记录的局域网内其他机器 IP 地址与其物理地址的对照表,如下图所示。

```
PS C:\Windows\System32> arp -a

接口: 10.162.5.113 --- 0xe
Internet 地址      物理地址      类型
10.162.0.1         74-3a-20-b9-e8-02 动态
10.162.255.255     ff-ff-ff-ff-ff-ff 静态
224.0.0.22         01-00-5e-00-00-16 静态
224.0.0.251        01-00-5e-00-00-fb 静态
224.0.0.252        01-00-5e-00-00-fc 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态
255.255.255.255    ff-ff-ff-ff-ff-ff 静态

接口: 172.30.0.1 --- 0x33
Internet 地址      物理地址      类型
172.30.15.255      ff-ff-ff-ff-ff-ff 静态
224.0.0.22         01-00-5e-00-00-16 静态
224.0.0.251        01-00-5e-00-00-fb 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态

PS C:\Windows\System32>
```

#### 5. 显示从本机到达一个特定地址的路由

在终端输入 `tracert xxx.com` 即可查看本机到一个特定地址的路由,这里我们通过 `tracert note.hobbitqia.cc` 命令查看。(note.hobbitqia.cc 是本人的公开笔记本)

```
PS C:\Windows\System32> tracert note.hobbitqia.cc

通过最多 30 个跃点跟踪
到 hobbitqia.github.io [2606:50c0:8001::153] 的路由:

 1  3 ms  2 ms  1 ms  2001:da8:e000:191::2
 2 14 ms  2 ms  2 ms  2001:da8:e000:191::1
 3  2 ms  1 ms  3 ms  2001:da8:e000:189::1
 4  8 ms 10 ms  2 ms  2001:da8:e000:75::2
 5  5 ms  4 ms  4 ms  2001:da8:e000:1::1
 6  2 ms  3 ms  4 ms  2001:da8:b4:1::1
 7  3 ms  4 ms  2 ms  2001:da8:2:115::1
 8 10 ms  9 ms  8 ms  2001:da8:2:13::1
 9 26 ms 25 ms 25 ms 2001:da8:2:11::2
10 29 ms 29 ms 30 ms 2001:da8:2:27::1
11 34 ms 32 ms 32 ms 2001:da8:2:2::1
12 31 ms 47 ms 34 ms cernet2.net [2001:252:0:2::101]
13 33 ms 33 ms 34 ms cernet2.net [2001:252:0:109::2]
14 * * * 请求超时。
15 * * * 请求超时。
16 * * * 请求超时。
17 246 ms 245 ms 245 ms 54113.sgw.equinix.com [2001:de8:4::5:4113:1]
18 113 ms 101 ms 101 ms 2606:50c0:8001::153

跟踪完成。
PS C:\Windows\System32>
```

#### 6. 显示某一个域名的 IP 地址

通过 `ping` 命令来显示域名的 IP 地址。这里我们通过 `ping baidu.com` 来查询百度的 IP 地址,如下图所示。

```
PS C:\Windows\System32> ping baidu.com

正在 Ping baidu.com [39.156.66.101] 具有 32 字节的数据:
来自 39.156.66.10 的回复: 字节=32 时间=27ms TTL=51
来自 39.156.66.10 的回复: 字节=32 时间=27ms TTL=51
来自 39.156.66.10 的回复: 字节=32 时间=27ms TTL=51
来自 39.156.66.10 的回复: 字节=32 时间=28ms TTL=51

39.156.66.10 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 = 27ms, 最长 = 28ms, 平均 = 27ms
PS C:\Windows\System32>
```

也可以使用 `nslookup baidu.com` 来查看 IP 地址,结果如下。



```
PS C:\Windows\System32> nslookup baidu.com
服务器: dns1.zju.edu.cn
Address: 10.10.0.21

非权威应答:
名称: baidu.com
Addresses: 39.156.66.10
          110.242.68.66
```

7. 显示已经与本机建立 TCP 连接的端口、IP 地址、连接状态等信息  
在终端输入 `netstat -an` 命令，并通过管道传给 `findstr "tcp"` 和 `findstr "ESTABLISHED"` 即可看到已经与本机建立 TCP 连接的端口、IP 地址、连接状态等信息，如下图所示。  
(篇幅限制这里仅展示了部分连接信息)

```
PS C:\Windows\System32> netstat -an | findstr "ESTABLISHED" | findstr "TCP"
TCP 10.162.5.113:49414 20.198.162.76:443 ESTABLISHED
TCP 10.162.5.113:57944 120.241.130.216:8080 ESTABLISHED
TCP 10.162.5.113:57984 36.155.205.230:8080 ESTABLISHED
TCP 10.162.5.113:65425 20.197.71.89:443 ESTABLISHED
TCP 127.0.0.1:6000 127.0.0.1:50445 ESTABLISHED
TCP 127.0.0.1:6000 127.0.0.1:50446 ESTABLISHED
TCP 127.0.0.1:6000 127.0.0.1:50447 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:49729 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:50412 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:50414 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:50420 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:50496 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:50503 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:50505 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:50510 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:50548 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:50600 ESTABLISHED
TCP 127.0.0.1:10809 127.0.0.1:50601 ESTABLISHED
TCP 127.0.0.1:49729 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:50412 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:50414 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:50420 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:50445 127.0.0.1:6000 ESTABLISHED
TCP 127.0.0.1:50446 127.0.0.1:6000 ESTABLISHED
TCP 127.0.0.1:50447 127.0.0.1:6000 ESTABLISHED
TCP 127.0.0.1:50496 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:50503 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:50505 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:50510 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:50548 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:50600 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:50601 127.0.0.1:10809 ESTABLISHED
TCP 127.0.0.1:54533 127.0.0.1:56636 ESTABLISHED
TCP 127.0.0.1:56636 127.0.0.1:54533 ESTABLISHED
TCP 127.0.0.1:56637 127.0.0.1:56638 ESTABLISHED
TCP 127.0.0.1:56638 127.0.0.1:56637 ESTABLISHED
TCP 127.0.0.1:56680 127.0.0.1:56681 ESTABLISHED
TCP 127.0.0.1:56681 127.0.0.1:56680 ESTABLISHED
TCP [240c:c781:7000:f86:40f:9fdc:363b:4965]:50504 [2402:4f00:4001::df77:3292]:443 ESTABLISHED
TCP [240c:c781:7000:f86:40f:9fdc:363b:4965]:50507 [2402:4f00:4001::df77:3292]:443 ESTABLISHED
TCP [240c:c781:7000:f86:40f:9fdc:363b:4965]:53495 [2409:8c54:1003:1019::121]:443 ESTABLISHED
TCP [240c:c781:7000:f86:40f:9fdc:363b:4965]:55953 [2408:4001:f10:183]:443 ESTABLISHED
TCP [240c:c781:7000:f86:40f:9fdc:363b:4965]:65339 [2605:6400:10:6ec:186f:736b:c73a:66ee]:38911 ESTABLISHED
PS C:\Windows\System32>
```

同时我们再使用 `netstat -an`，可以看到表头：

```
PS C:\Windows\System32> netstat -an

活动连接

 协议 本地地址           外部地址           状态
TCP    0.0.0.0:135         0.0.0.0:0          LISTENING
TCP    0.0.0.0:445         0.0.0.0:0          LISTENING
TCP    0.0.0.0:5040        0.0.0.0:0          LISTENING
TCP    0.0.0.0:5091        0.0.0.0:0          LISTENING
```

即刚刚的输出也是按照协议-本地地址-外部地址-状态来输出的。

8. 显示本机的路由表信息，并手工添加一个路由

首先我们在终端中输入 `route print` 指令查看所有路由，结果如下图所示。

```
PS C:\Windows\System32> route print

接口列表
8...64 6e e0 ab d9 a8 .....Microsoft Wi-Fi Direct Virtual Adapter
3...66 6e e0 ab d9 a7 .....Microsoft Wi-Fi Direct Virtual Adapter #2
14...64 6e e0 ab d9 a7 .....Intel(R) Wi-Fi 6 AX201 160MHz
11...00 ff 89 0c af 17 .....Sangfor SSL VPN CS Support System VNIC
15...64 6e e0 ab d9 ab .....Bluetooth Device (Personal Area Network)
1.....Software Loopback Interface 1
51...00 15 5d f0 a1 70 .....Hyper-V Virtual Ethernet Adapter

IPv4 路由表

活动路由:
网络目标      网络掩码      网关      接口      跃点数
0.0.0.0        0.0.0.0        10.162.0.1  10.162.5.113  35
10.162.0.0     255.255.0.0    在链路上    10.162.5.113  291
10.162.5.113   255.255.255.255 在链路上    10.162.5.113  291
10.162.255.255 255.255.255.255 在链路上    10.162.5.113  291
127.0.0.0     255.0.0.0      在链路上    127.0.0.1     331
127.0.0.1     255.255.255.255 在链路上    127.0.0.1     331
172.30.0.0    255.255.240.0  在链路上    172.30.0.1    5256
172.30.0.1    255.255.255.255 在链路上    172.30.0.1    5256
172.30.15.255 255.255.255.255 在链路上    172.30.0.1    5256
224.0.0.0     240.0.0.0      在链路上    127.0.0.1     331
224.0.0.0     240.0.0.0      在链路上    10.162.5.113  291
224.0.0.0     240.0.0.0      在链路上    172.30.0.1    5256
255.255.255.255 255.255.255.255 在链路上    127.0.0.1     331
255.255.255.255 255.255.255.255 在链路上    10.162.5.113  291
255.255.255.255 255.255.255.255 在链路上    172.30.0.1    5256

永久路由:
无

IPv6 路由表

活动路由:
接口跃点数网络目标      网关
14 291 :::/0      fe80::763a:20ff:feb9:e802
1 331 ::1/128     在链路上
14 291 240c:c781:7000:f86::/64 在链路上
14 291 240c:c781:7000:f86:592b:5dc:ed6c:db09/128 在链路上
14 291 240c:c781:7000:f86:c462:ab2f:ac28:561d/128 在链路上
14 291 fe80::/64 在链路上
51 5256 fe80::/64 在链路上
51 5256 fe80::8e59:b6e9:5570:d9d3/128 在链路上
14 291 fe80::fc2d:64dd:7646:2072/128 在链路上
1 331 ff00::/8 在链路上
14 291 ff00::/8 在链路上
51 5256 ff00::/8 在链路上

永久路由:
无
```

我们通过 `route add 10.253.251.0 mask 255.255.255.0 -p 192.254.1.1` 命令手工添加路由，这里 10.253.251.0 是源地址，255.255.255.0 是源地址掩码，192.254.1.1 是目标地址。接着我们通过 `route print` 查看路由，可以手工添加路由成功。

```
PS C:\Windows\System32> route print

接口列表
8...64 6e e0 ab d9 a8 .....Microsoft Wi-Fi Direct Virtual Adapter
3...66 6e e0 ab d9 a7 .....Microsoft Wi-Fi Direct Virtual Adapter #2
14...64 6e e0 ab d9 a7 .....Intel(R) Wi-Fi 6 AX201 160MHz
11...00 ff 89 0c af 17 .....Sangfor SSL VPN CS Support System VNIC
15...64 6e e0 ab d9 ab .....Bluetooth Device (Personal Area Network)
1.....Software Loopback Interface 1
51...00 15 5d f0 a1 70 .....Hyper-V Virtual Ethernet Adapter

IPv4 路由表

活动路由:
网络目标      网络掩码      网关      接口      跃点数
0.0.0.0        0.0.0.0        10.162.0.1  10.162.5.113  40
10.162.0.0     255.255.0.0    在链路上    10.162.5.113  296
10.162.5.113   255.255.255.255 在链路上    10.162.5.113  296
10.162.255.255 255.255.255.255 在链路上    10.162.5.113  296
10.253.251.0   255.255.255.0  192.254.1.1  10.162.5.113  41
127.0.0.0     255.0.0.0      在链路上    127.0.0.1     331
127.0.0.1     255.255.255.255 在链路上    127.0.0.1     331
172.30.0.0    255.255.240.0  在链路上    172.30.0.1    5256
172.30.0.1    255.255.255.255 在链路上    172.30.0.1    5256
172.30.15.255 255.255.255.255 在链路上    172.30.0.1    5256
224.0.0.0     240.0.0.0      在链路上    127.0.0.1     331
224.0.0.0     240.0.0.0      在链路上    10.162.5.113  296
224.0.0.0     240.0.0.0      在链路上    172.30.0.1    5256
255.255.255.255 255.255.255.255 在链路上    127.0.0.1     331
255.255.255.255 255.255.255.255 在链路上    10.162.5.113  296
255.255.255.255 255.255.255.255 在链路上    172.30.0.1    5256

永久路由:
网络地址      网络掩码      网关地址      跃点数
10.253.251.0  255.255.255.0  192.254.1.1    1
```

## 9. 显示本机的网络映射连接

在终端输入 `net use` 命令来显示本机的网络映射连接。起初显示没有网络连接。



```
PS C:\Windows\System32> net use
会记录新的网络连接。
列表是空的。
```

在执行完第十题之后，再次输入 net use 命令，可以看到不同的结果。

```
PS C:\windows\system32> net use
会记录新的网络连接。

状态          本地          远程          网络
-----
已断开连接          \\SK-20210817HXXZ\IPC$  Microsoft Windows Network
命令成功完成。

PS C:\windows\system32>
```

#### 10. 显示局域网内某台机器的共享资源

这里我将自己的电脑和室友 A 的电脑同时连入手机热点，并将 A 电脑上的部分文件设为共享，创建实验环境。接着在终端输入 NET USE[computername]\IPC\$ "password" /USER:"user name"建立连接，随后 net view \\[computername]即可。

```
PS C:\windows\system32> net use \\SK-20210817HXXZ\IPC$ " " /USER:"蚀"
命令成功完成。

PS C:\windows\system32> net view \\SK-20210817HXXZ
在 \\SK-20210817HXXZ 的共享资源

共享名  类型  使用为  注释
-----
Users   Disk
命令成功完成。

PS C:\windows\system32>
```

#### 11. 使用 telnet 连接 WEB 服务器的端口，输入 (<cr>表示回车) 获得该网站的主页内容：

GET / HTTP/1.1

Host: [www.baidu.com](http://www.baidu.com)

这里我们首先在终端输入 telnet baidu.com 80，随后敲入 ctrl+l，并敲击回车键，随后输入 GET / HTTP/1.1 和 Host:www.baidu.com 两行命令，可以看到如下输出。

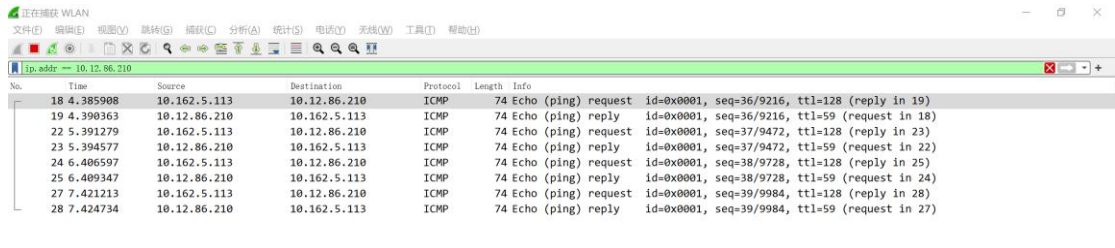
```
GET / HTTP/1.1
Host: www.baidu.com

HTTP/1.1 200 OK
Date: Sun, 01 Oct 2023 11:36:51 GMT
Server: Apache
Last-Modified: Tue, 12 Jan 2010 13:48:00 GMT
ETag: "51-47cf7e6ee8400"
Accept-Ranges: bytes
Content-Length: 81
Cache-Control: max-age=86400
Expires: Mon, 02 Oct 2023 11:36:51 GMT
Connection: Keep-Alive
Content-Type: text/html

<html>
  <meta http-equiv="refresh" content="0;url=http://www.baidu.com/">
</html>
```

#### ● 观察使用 Ping 命令时在 WireShark 中出现的数据包并捕获。这是什么协议？

通过设置显示捕获器，我们捕获了执行 ping 10.12.86.210 时出现的数据包，可以看到这是 ICMP 协议的数据包。ICMP 用于在 IP 网络中传递控制消息和错误消息。Ping 命令实际上是 ICMP Echo 请求和响应的发送和接收过程。

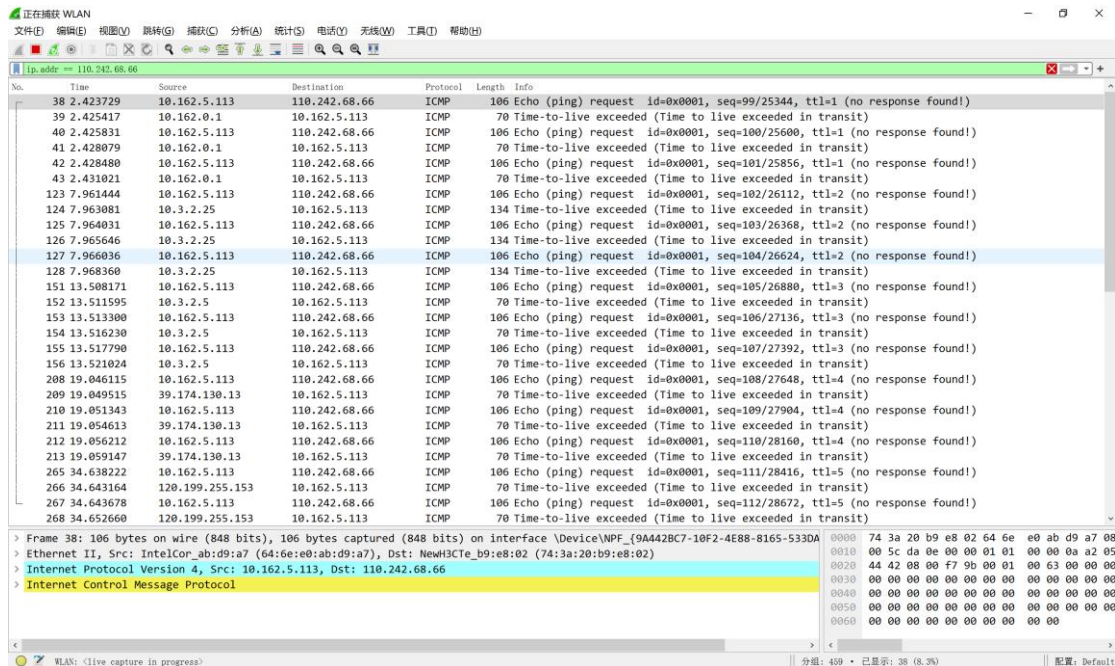


No.	Time	Source	Destination	Protocol	Length	Info
18	4.385908	10.162.5.113	10.12.86.210	ICMP	74	Echo (ping) request id=0x0001, seq=36/9216, ttl=128 (reply in 19)
19	4.390363	10.12.86.210	10.162.5.113	ICMP	74	Echo (ping) reply id=0x0001, seq=36/9216, ttl=59 (request in 18)
22	5.391279	10.162.5.113	10.12.86.210	ICMP	74	Echo (ping) request id=0x0001, seq=37/9472, ttl=128 (reply in 23)
23	5.394577	10.12.86.210	10.162.5.113	ICMP	74	Echo (ping) reply id=0x0001, seq=37/9472, ttl=59 (request in 22)
24	6.406597	10.162.5.113	10.12.86.210	ICMP	74	Echo (ping) request id=0x0001, seq=38/9728, ttl=128 (reply in 25)
25	6.409347	10.12.86.210	10.162.5.113	ICMP	74	Echo (ping) reply id=0x0001, seq=38/9728, ttl=59 (request in 24)
27	7.421213	10.162.5.113	10.12.86.210	ICMP	74	Echo (ping) request id=0x0001, seq=39/9984, ttl=128 (reply in 28)
28	7.424734	10.12.86.210	10.162.5.113	ICMP	74	Echo (ping) reply id=0x0001, seq=39/9984, ttl=59 (request in 27)

- 观察使用 Tracert 命令时在 WireShark 中出现的数据包并捕获。这是什么协议？

通过设置显示捕获器，我们捕获了执行 `tracert baidu.com` (IP 地址为 110.242.68.66) 时出现的数据包。可以看到这也是 ICMP 协议的数据包。

当使用 Tracert (或 Traceroute) 命令时，在 Wireshark 中出现的数据包属于 ICMP (Internet Control Message Protocol) 协议。Tracert 是一种网络诊断工具，用于跟踪数据包在网络中的路径，并显示每个路由器或中间节点的延迟时间。



No.	Time	Source	Destination	Protocol	Length	Info
38	2.423729	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=99/25344, ttl=1 (no response found!)
39	2.425417	10.162.0.1	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
40	2.425831	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=100/25600, ttl=1 (no response found!)
41	2.428079	10.162.0.1	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
42	2.428480	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=101/25856, ttl=1 (no response found!)
43	2.431021	10.162.0.1	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
123	7.961444	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=102/26112, ttl=2 (no response found!)
124	7.963081	10.3.2.25	10.162.5.113	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
125	7.964031	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=103/26368, ttl=2 (no response found!)
126	7.965646	10.3.2.25	10.162.5.113	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
127	7.966036	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=104/26624, ttl=2 (no response found!)
128	7.968360	10.3.2.25	10.162.5.113	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
151	13.508171	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=105/26880, ttl=3 (no response found!)
152	13.511595	10.3.2.5	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
153	13.513300	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=106/27136, ttl=3 (no response found!)
154	13.516230	10.3.2.5	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
155	13.517790	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=107/27392, ttl=3 (no response found!)
156	13.521024	10.3.2.5	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
208	19.046115	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=108/27648, ttl=4 (no response found!)
209	19.049515	39.174.130.13	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
210	19.051343	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=109/27904, ttl=4 (no response found!)
211	19.054613	39.174.130.13	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
212	19.056212	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=110/28160, ttl=4 (no response found!)
213	19.059147	39.174.130.13	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
265	34.638222	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=111/28416, ttl=5 (no response found!)
266	34.643164	120.199.255.153	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
267	34.643678	10.162.5.113	110.242.68.66	ICMP	106	Echo (ping) request id=0x0001, seq=112/28672, ttl=5 (no response found!)
268	34.652660	120.199.255.153	10.162.5.113	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)

- 观察使用 Nslookup 命令时在 WireShark 中出现的数据包并捕获。这是什么协议？

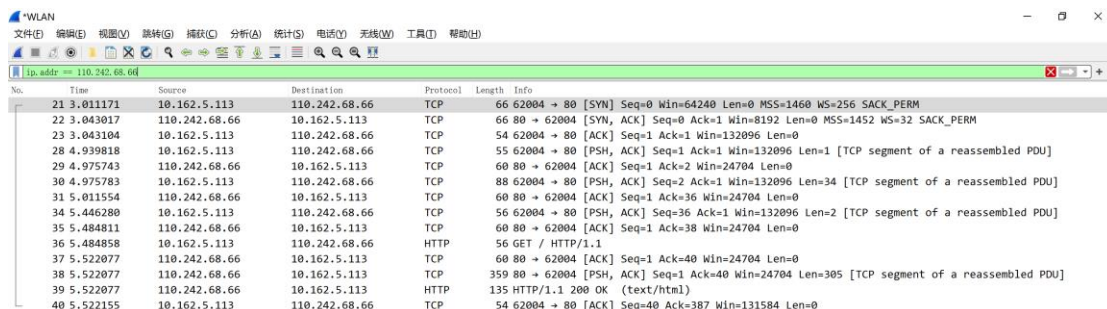
执行指令 `nslookup baidu.com` 时，可以看到 wireshark 中出现了 DNS 协议的数据包，且 info 里包含了 baidu.com。

DNS 是一种用于域名解析的协议，它将域名映射到 IP 地址，以便计算机能够在网络上找到其他设备。当你运行 `nslookup` 命令并指定要查询的域名时，计算机会向 DNS 服务器发送 DNS 查询请求，以获取该域名对应的 IP 地址。

12 1.311409	10.162.5.113	10.10.0.21	DNS	83 Standard query 0x0001 PTR 21.0.10.10.in-addr.arpa
13 1.315839	10.10.0.21	10.162.5.113	DNS	142 Standard query response 0x0001 PTR 21.0.10.10.in-addr.arpa PTR dns1.zju.edu.cn NS dns1.zju...
14 1.316298	10.162.5.113	10.10.0.21	DNS	69 Standard query 0x0002 A baidu.com
15 1.322214	10.10.0.21	10.162.5.113	DNS	359 Standard query response 0x0002 A baidu.com A 110.242.68.66 A 39.156.66.10 NS dns.baidu.com ...
16 1.322593	10.162.5.113	10.10.0.21	DNS	69 Standard query 0x0003 AAAA baidu.com
17 1.325596	10.10.0.21	10.162.5.113	DNS	112 Standard query response 0x0003 AAAA baidu.com SOA dns.baidu.com

- 观察使用 Telnet 命令时在 Wireshark 中出现的数据包并捕获。这是什么协议？

在执行上面的第十一个功能时，我们可以看到 wireshark 中捕获了 TCP 协议和 HTTP 协议的数据包。Wireshark 可以捕获并显示这些 Telnet 通信的 TCP 数据包，包括 TCP 连接的建立、HTTP 请求的发送以及从 Web 服务器返回的 HTTP 响应。



No.	Time	Source	Destination	Protocol	Length	Info
21	3.011171	10.162.5.113	110.242.68.66	TCP	66	62004 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
22	3.043017	110.242.68.66	10.162.5.113	TCP	66	80 → 62004 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM
23	3.043104	10.162.5.113	110.242.68.66	TCP	54	62004 → 80 [ACK] Seq=1 Ack=1 Win=132096 Len=0
28	4.930918	10.162.5.113	110.242.68.66	TCP	55	62004 → 80 [PSH, ACK] Seq=1 Ack=1 Win=132096 Len=1 [TCP segment of a reassembled PDU]
29	4.975743	110.242.68.66	10.162.5.113	TCP	60	80 → 62004 [ACK] Seq=1 Ack=2 Win=24704 Len=0
30	4.975783	10.162.5.113	110.242.68.66	TCP	88	62004 → 80 [PSH, ACK] Seq=2 Ack=1 Win=132096 Len=34 [TCP segment of a reassembled PDU]
31	5.011554	110.242.68.66	10.162.5.113	TCP	60	80 → 62004 [ACK] Seq=1 Ack=36 Win=24704 Len=0
34	5.446280	10.162.5.113	110.242.68.66	TCP	56	62004 → 80 [PSH, ACK] Seq=36 Ack=1 Win=132096 Len=2 [TCP segment of a reassembled PDU]
35	5.484811	110.242.68.66	10.162.5.113	TCP	60	80 → 62004 [ACK] Seq=1 Ack=38 Win=24704 Len=0
36	5.484858	10.162.5.113	110.242.68.66	HTTP	56	GET / HTTP/1.1
37	5.522077	110.242.68.66	10.162.5.113	TCP	60	80 → 62004 [ACK] Seq=1 Ack=40 Win=24704 Len=0
38	5.522077	110.242.68.66	10.162.5.113	TCP	359	80 → 62004 [PSH, ACK] Seq=1 Ack=40 Win=24704 Len=305 [TCP segment of a reassembled PDU]
39	5.522077	110.242.68.66	10.162.5.113	HTTP	135	HTTP/1.1 200 OK (text/html)
40	5.522155	10.162.5.113	110.242.68.66	TCP	54	62004 → 80 [ACK] Seq=40 Ack=387 Win=131584 Len=0

## 六、实验结果与分析

- Wireshark 的两种过滤器有什么不同？

### 1. 显示过滤器

显示过滤器用于在 Wireshark 的用户界面中过滤和显示已经捕获的数据包。它们仅影响你在 Wireshark 中看到的数据包，而不会影响数据包的实际捕获。

用于在已捕获的数据包中筛选、搜索、过滤或突出显示特定类型的数据包，以帮助用户更容易地分析网络流量。

### 2. 捕获过滤器

捕获过滤器用于在数据包捕获开始之前定义要捕获的数据包类型。它们影响数据包的实际捕获，可以用来减少捕获的数据量，以节省存储和分析时间。

用于过滤出感兴趣的数据包类型，避免记录大量不必要的数据包。

- 哪些网络命令会产生在 Wireshark 中产生数据包，为什么？

1. Ping 命令：当使用 ping 命令测试到特定主机的连接性时，它会发送 ICMP Echo 请求数据包到目标主机，并接收来自目标主机的 ICMP Echo 响应数据包。
2. Tracert 命令：使用 tracert 命令来跟踪数据包从源到目标的路径。它会发送一系列的 ICMP Echo 请求数据包。
3. Telnet 命令：当使用 Telne 等远程登录协议连接到远程主机时，会建立 TCP 连接，

并在该连接上传输命令和响应。

4. **HTTP 或 HTTPS 请求**（如 GET / HTTP/1.1）：当你使用 Web 浏览器或其他 HTTP 客户端发送 HTTP 或 HTTPS 请求时，会触发 TCP 连接并发送 HTTP 请求数据包到 Web 服务器。Web 服务器会返回 HTTP 响应数据包。这些数据包可见于 Wireshark，用于查看 Web 通信，包括请求和响应。
5. **Nslookup 命令**：当使用 nslookup 命令查询域名时，通常会触发 DNS 查询，因此会生成 DNS 查询数据包。这些 DNS 查询数据包将用于获取域名的 IP 地址或其他相关信息。
6. **Arp 命令**：如果 ARP 缓存表中不存在对应的目标网络，源计算机就会发出 ARP 请求，ARP 请求就是将自己的 IP 地址和希望得到的 MAC 地址的目标计算机的 IP 地址包装成数据包通过广播发出去，当目标计算机接收到这个数据包后会将源 IP 地址取出来，将自己的 MAC 地址包装成数据包发送回去。

- **Ping 发送的是什么类型的协议数据包？什么时候会出现 ARP 消息？Ping 一个域名和 Ping 一个 IP 地址出现的数据包有什么不同？**

Ping 发送的是 ICMP 协议数据包。

当 Ping 目标主机的 IP 地址时，如果本地计算机不知道目标主机的 MAC 地址，它会发送一个 ARP 请求以获取目标主机的 MAC 地址。这是为了构建要发送到目标主机的数据包。

Ping 一个域名和 Ping 一个 IP 地址出现的数据包有什么不同？

- **Ping 一个域名**：当你使用 Ping 命令并提供一个域名时，Ping 会首先使用 DNS 解析该域名，将域名转换为 IP 地址。然后，它会发送 ICMP Echo 请求数据包到得到的 IP 地址。在这个过程中，你可能会看到 DNS 请求和响应的数据包，以及 ICMP Echo 请求和响应的数据包。
- **Ping 一个 IP 地址**：如果你直接提供了目标主机的 IP 地址，Ping 命令会直接发送 ICMP Echo 请求数据包到这个 IP 地址。在这种情况下，没有 DNS 请求和响应的数据包，只有 ICMP Echo 请求和响应的数据包。

## 七、 讨论、心得

最大的困难在于实验没有任何指导，全靠自己摸索，而且看上去和理论没什么关系，相关的知识都没有讲，希望以后每个步骤能有指导，而且理论能和实验配套。

助教姐姐捞捞（哭