浙江大学

# 本科实验报告

| | |
|---|---|
| 课程名称： | 计算机网络基础 |
| 姓　　名： | 沈子衿 |
| 学　　院： | 计算机学院 |
| 　　系： | 软件工程 |
| 专　　业： | 软件工程 |
| 学　　号： | 3160104734 |
| 指导教师： | 董玮 |

2018 年 9 月 25 日

# 浙江大学实验报告

课程名称：计算机网络基础　　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：沈子衿　　专业：软件工程　学号：3160104734

同组学生姓名：林宇翔　　指导老师：董玮

实验地点：计算机网络实验室　　实验日期：2018 年 9 月 25 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. 测试到特定地址的联通性、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：

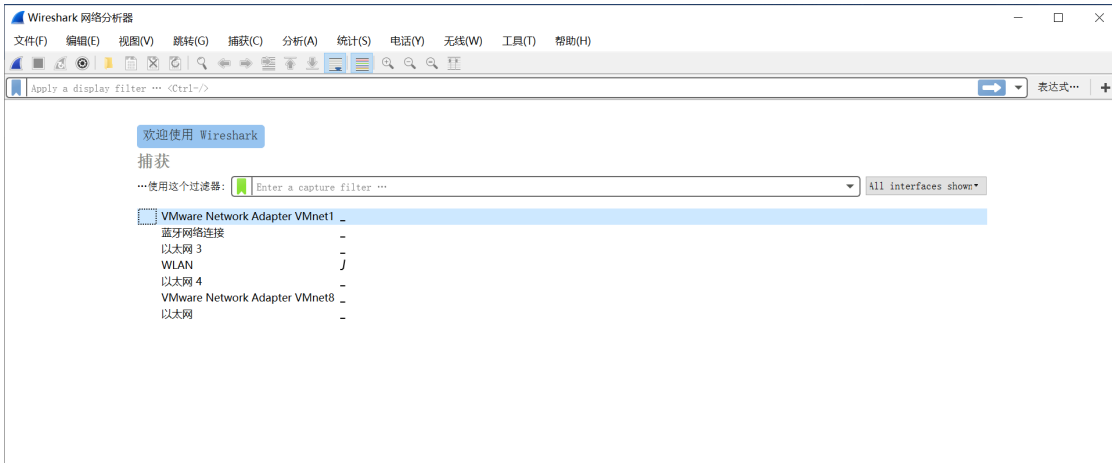
```
GET / HTTP/1.1<cr>
Host: 任意字符串<cr>
<cr>
```

- 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

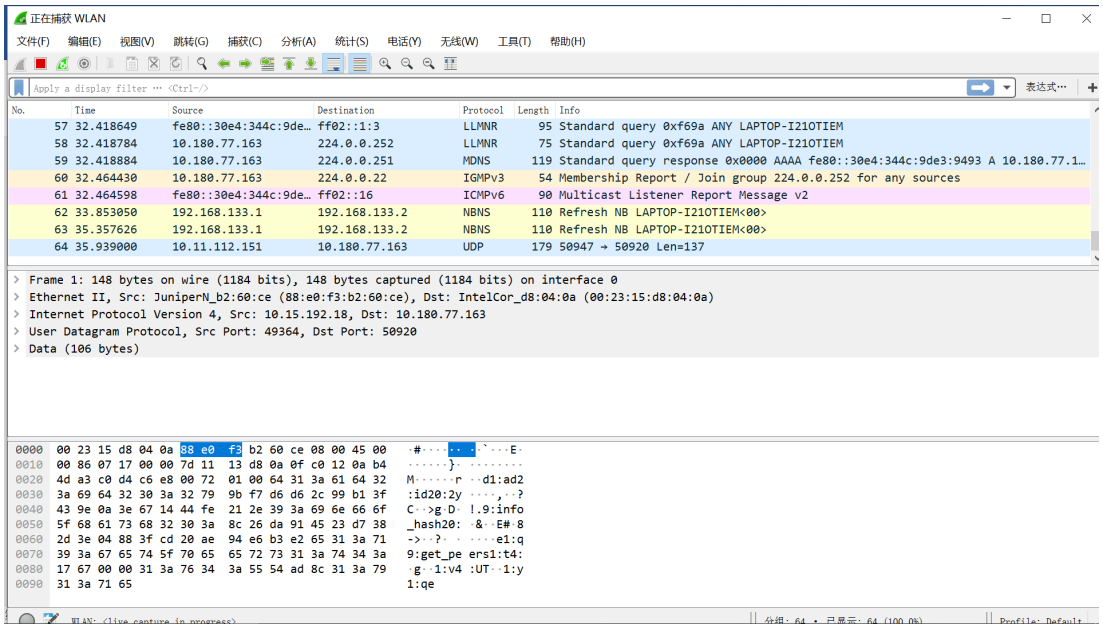
- **运行 Wireshark 软件，界面是由哪几个部分构成？各有什么作用？**

从互联网上下载 Wireshark 软件，安装后双击打开，首先显示的是如下界面：




上面是菜单和工具栏，窗口中则显示了当前电脑各网络适配器的网络状态，由于我当前连接的是无线局域网，因此图中 WLAN 一行显示是活动的（出现折线），而其他的网络则暂无活动迹象。

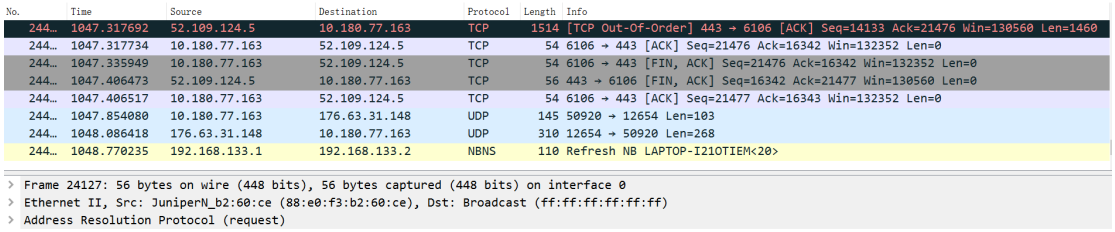
双击一个网络即可进入主界面对其进行细致分析，并开始捕获网络数据包。主界面如下：

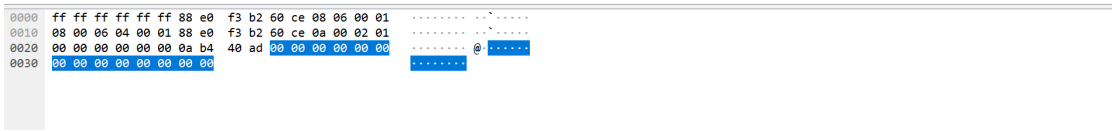



主界面主要由四个部分组成。最上面是菜单栏和众多控制按钮，用于对一些属性进行定制；其下是实时捕获包（帧）的列表，该列表显示了包序号、自本次捕获开始之后的时间戳、包来源、包目的地、协议种类、长度和包所携带的信息。由于网络传输无时无刻不在进行，因此列表是在快速动态刷新的。再下面显示了当前所选中的包的详细信息，包括实际大小（在缆线上传输的大小）、成功捕获的大小、接口、路由和包所携带的具体信息（或承担的具体功能），最底部则显示了这些信息的 16 进制格式化表示。

- **开始捕获网络数据包，你看到了什么？有哪些协议？**

程序运行了一段时间后，捕获包的界面如下：

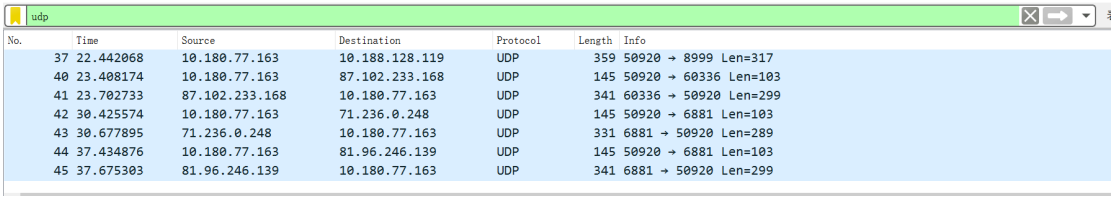



我看到了当前捕获包的列表（包括帧（包）序号、时间戳、）、每一个包的详细信息（实际大小、成功捕获的大小、接口、路由和包所携带的具体信息）以及对应的 16 进制表示。
至于协议，则有 TCP, UDP,NBNS, TLS, ICMP, ICMPv6，IGMPv3，LLMNR 等。

- **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。**

直接在 display filter 上输入需要筛选的协议类型，即可让界面只显示某一协议类型的数据包，如图：



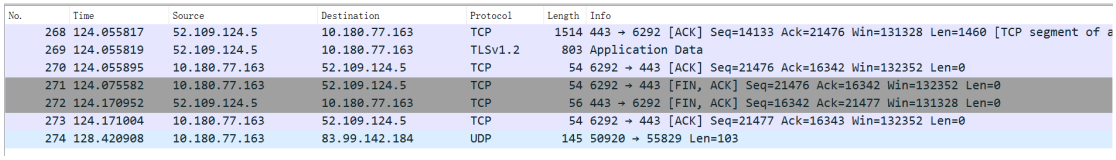
udp

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 37 | 22.442068 | 10.180.77.163 | 10.188.128.119 | UDP | 359 | 50920 → 8999 Len=317 |
| 40 | 23.408174 | 10.180.77.163 | 87.102.233.168 | UDP | 145 | 50920 → 60336 Len=103 |
| 41 | 23.702733 | 87.102.233.168 | 10.180.77.163 | UDP | 341 | 60336 → 50920 Len=299 |
| 42 | 30.425574 | 10.180.77.163 | 71.236.0.248 | UDP | 145 | 50920 → 6881 Len=103 |
| 43 | 30.677895 | 71.236.0.248 | 10.180.77.163 | UDP | 331 | 6881 → 50920 Len=289 |
| 44 | 37.434876 | 10.180.77.163 | 81.96.246.139 | UDP | 145 | 50920 → 6881 Len=103 |
| 45 | 37.675303 | 81.96.246.139 | 10.180.77.163 | UDP | 341 | 6881 → 50920 Len=299 |

包列表上只剩下 UDP 协议的包了。

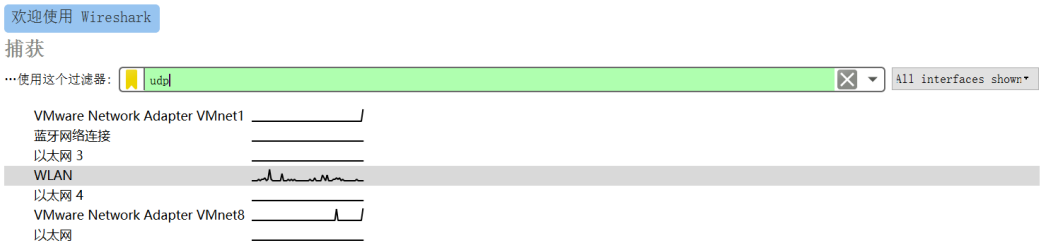
- **配置捕获过滤器，只捕获某类协议的数据包。**

关闭应用显示过滤器，列表恢复默认状态：

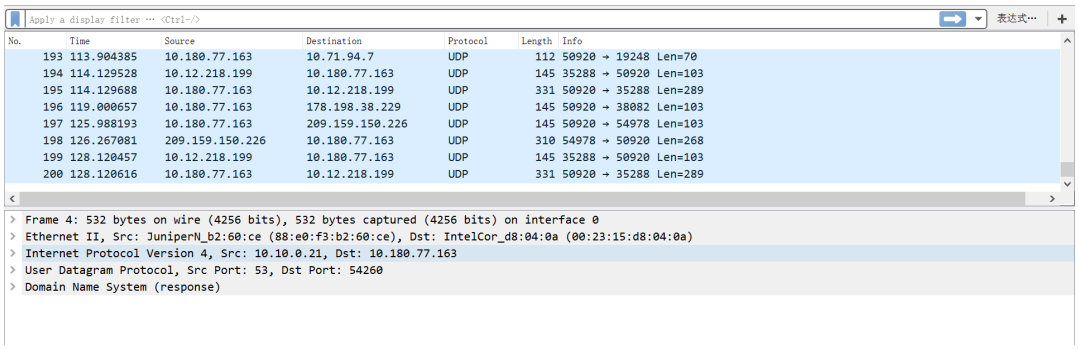


| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 268 | 124.055817 | 52.109.124.5 | 10.180.77.163 | TCP | 1514 | 443 → 6292 [ACK] Seq=14133 Ack=21476 Win=131328 Len=1460 [TCP segment of a |
| 269 | 124.055819 | 52.109.124.5 | 10.180.77.163 | TLSv1.2 | 803 | Application Data |
| 270 | 124.055895 | 10.180.77.163 | 52.109.124.5 | TCP | 54 | 6292 → 443 [ACK] Seq=21476 Ack=16342 Win=132352 Len=0 |
| 271 | 124.075582 | 10.180.77.163 | 52.109.124.5 | TCP | 54 | 6292 → 443 [FIN, ACK] Seq=21476 Ack=16342 Win=132352 Len=0 |
| 272 | 124.170952 | 52.109.124.5 | 10.180.77.163 | TCP | 56 | 443 → 6292 [FIN, ACK] Seq=16342 Ack=21477 Win=131328 Len=0 |
| 273 | 124.171004 | 10.180.77.163 | 52.109.124.5 | TCP | 54 | 6292 → 443 [ACK] Seq=21477 Ack=16343 Win=132352 Len=0 |
| 274 | 128.420908 | 10.180.77.163 | 83.99.142.184 | UDP | 145 | 50920 → 55829 Len=103 |

此时可以在先前选择网络适配器的页面使用特定语法进行捕获过滤：




如此处只显示 UDP 包，双击进入捕获列表界面：



Apply a display filter … <Ctrl-/>

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 193 | 113.904385 | 10.180.77.163 | 10.71.94.7 | UDP | 112 | 50920 → 19248 Len=70 |
| 194 | 114.129528 | 10.12.218.199 | 10.180.77.163 | UDP | 145 | 35288 → 50920 Len=103 |
| 195 | 114.129688 | 10.180.77.163 | 10.12.218.199 | UDP | 331 | 50920 → 35288 Len=289 |
| 196 | 119.000657 | 10.180.77.163 | 178.198.38.229 | UDP | 145 | 50920 → 38082 Len=103 |
| 197 | 125.988193 | 10.180.77.163 | 209.159.150.226 | UDP | 145 | 50920 → 54978 Len=103 |
| 198 | 126.267081 | 209.159.150.226 | 10.180.77.163 | UDP | 310 | 54978 → 50920 Len=268 |
| 199 | 128.120457 | 10.12.218.199 | 10.180.77.163 | UDP | 145 | 35288 → 50920 Len=103 |
| 200 | 128.120616 | 10.180.77.163 | 10.12.218.199 | UDP | 331 | 50920 → 35288 Len=289 |

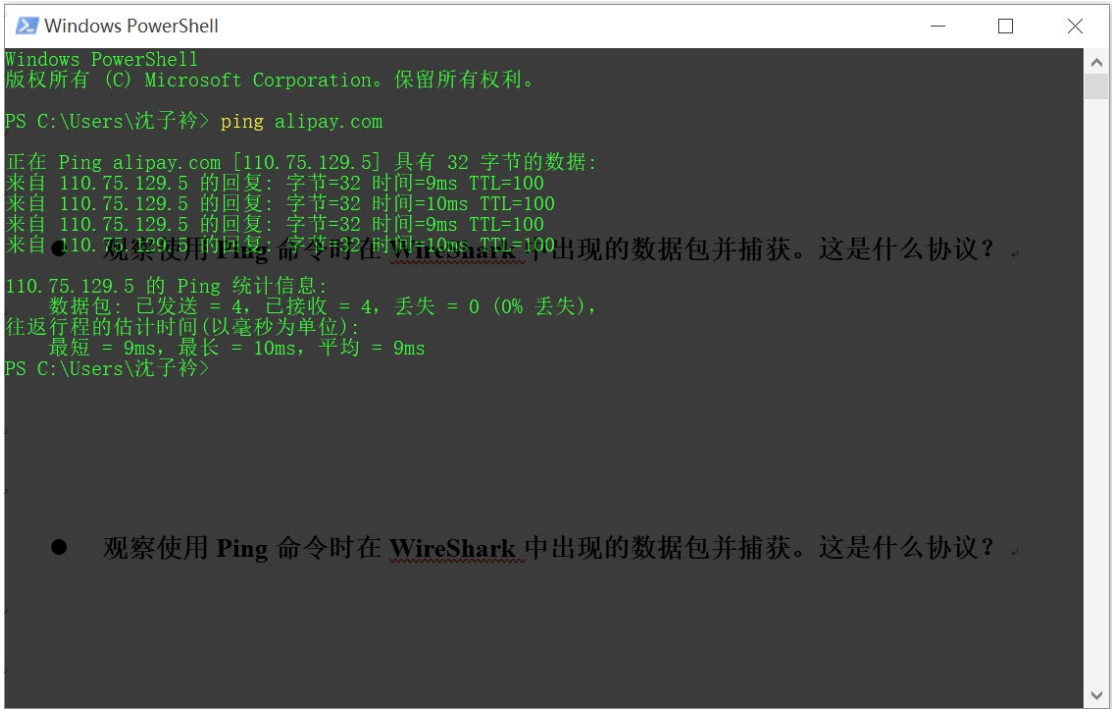
```
> Frame 4: 532 bytes on wire (4256 bits), 532 bytes captured (4256 bits) on interface 0
> Ethernet II, Src: JuniperN_b2:60:ce (88:e0:f3:b2:60:ce), Dst: IntelCor_d8:04:0a (00:23:15:d8:04:0a)
> Internet Protocol Version 4, Src: 10.10.0.21, Dst: 10.180.77.163
> User Datagram Protocol, Src Port: 53, Dst Port: 54260
> Domain Name System (response)
```

这也就实现了协议过滤。除此之外，使用正确的标识符和原语，还可以实现主机过滤和端口过滤等。

- **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) 测试到特定地址的联通性、数据包延迟时间

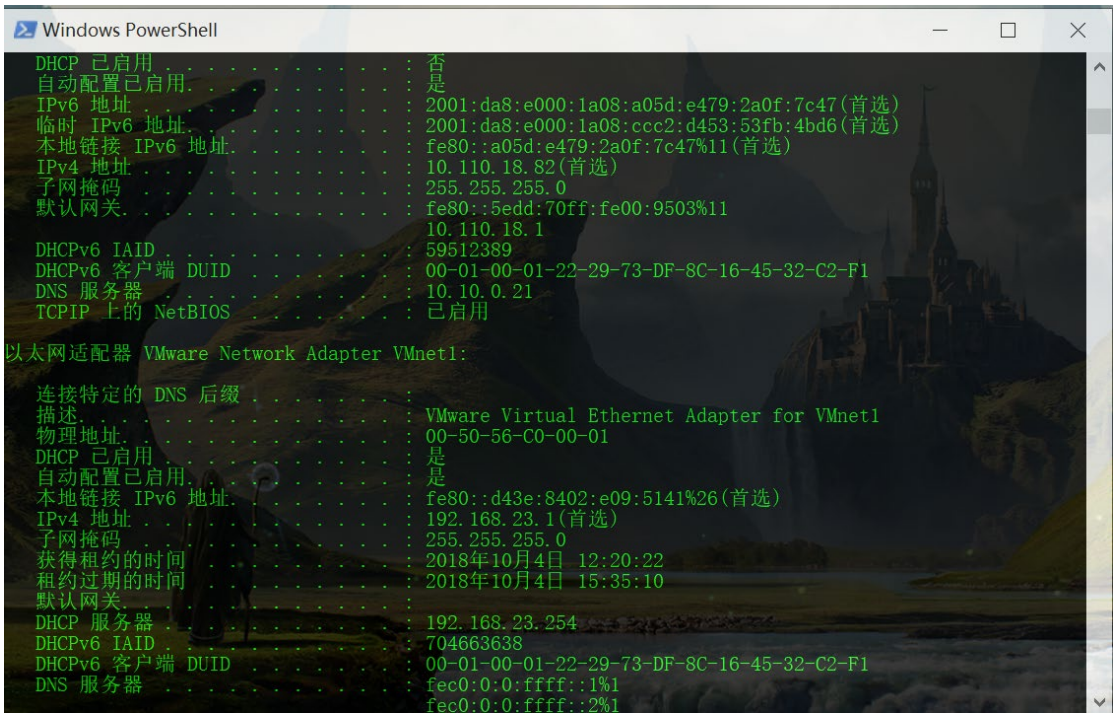
打开 Windows Powershell，使用 ping 命令尝试 ping 支付宝官网(Alipay.com)：




由于 Windows 系统中的默认设置，只发送/接收了 4 个数据包。如果需要长 ping，可以在命令后加上‘-t’，并使用 Ctrl+C 控制 ping 进程结束。从 ping 命令我们可以清楚地看到到该地址地数据包延迟时间，并由此判断网络连通性。

2) 显示本机的网卡物理地址、IP 地址

打开 Windows Powershell，使用 ipconfig /all 命令，显示本机所有适配器的网卡地址：



截图以适配器“以太网”为例（网络环境：浙大玉泉校区学生宿舍有线网）：

```
以太网适配器 以太网:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Intel(R) Ethernet Connection (4) I219-V
   物理地址. . . . . . . . . . . . . : 8C-16-45-32-C2-F1
   DHCP 已启用 . . . . . . . . . . . : 否
   自动配置已启用. . . . . . . . . . : 是
   IPv6 地址 . . . . . . . . . . . . : 2001:da8:e000:1a08:a05d:e479:2a0f:7c47(首选)
   临时 IPv6 地址. . . . . . . . . . : 2001:da8:e000:1a08:ccc2:d453:53fb:4bd6(首选)
   本地链接 IPv6 地址. . . . . . . . : fe80::a05d:e479:2a0f:7c47%11(首选)
   IPv4 地址 . . . . . . . . . . . . : 10.110.18.82(首选)
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : fe80::5edd:70ff:fe00:9503%11
                                       10.110.18.1
   DHCPv6 IAID . . . . . . . . . . . : 59512389
   DHCPv6 客户端 DUID . . . . . . . . : 00-01-00-01-22-29-73-DF-8C-16-45-32-C2-F1
   DNS 服务器 . . . . . . . . . . . : 10.10.0.21
   TCPIP 上的 NetBIOS . . . . . . . : 已启用
```

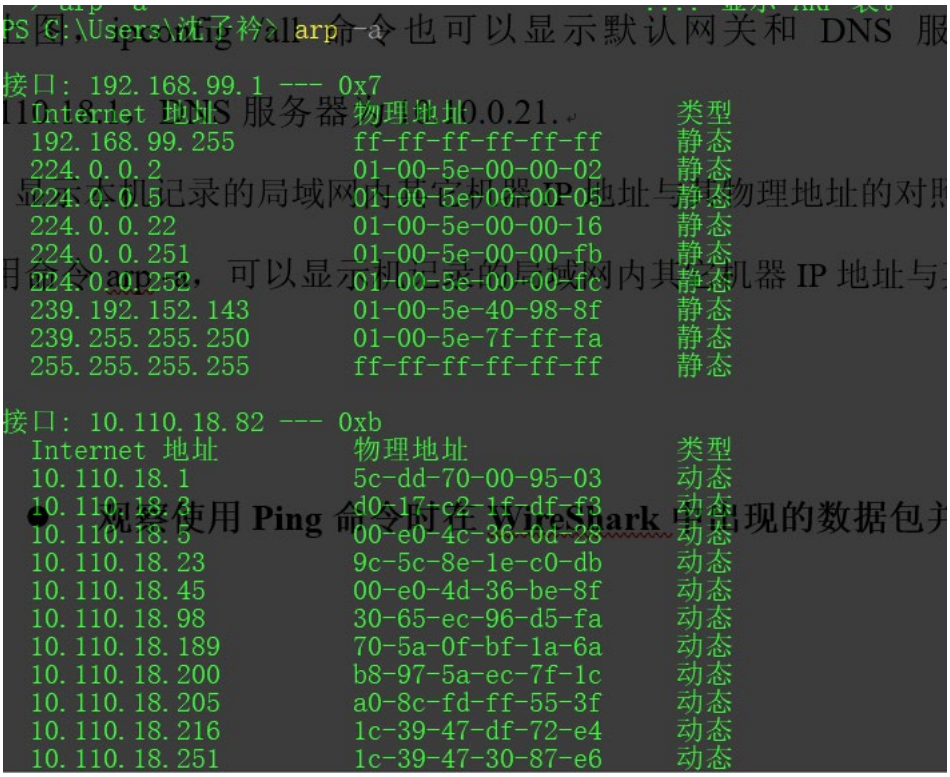
可以看出该适配器网卡物理地址为 8C-16-45-32-C2-F1，本地 IPv4 地址为 10.110.18.82（浙大内网的私有 IP）。由于玉泉有线网支持 IPv6，因此本地 IPv6 地址也显示了出来。

3) 显示本机的默认网关地址、DNS 服务器地址

如上图，ipconfig /all 命令也可以显示默认网关和 DNS 服务器的地址。默认网关为 10.110.18.1，DNS 服务器为 10.10.0.21.

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

使用命令 arp -a，可以显示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



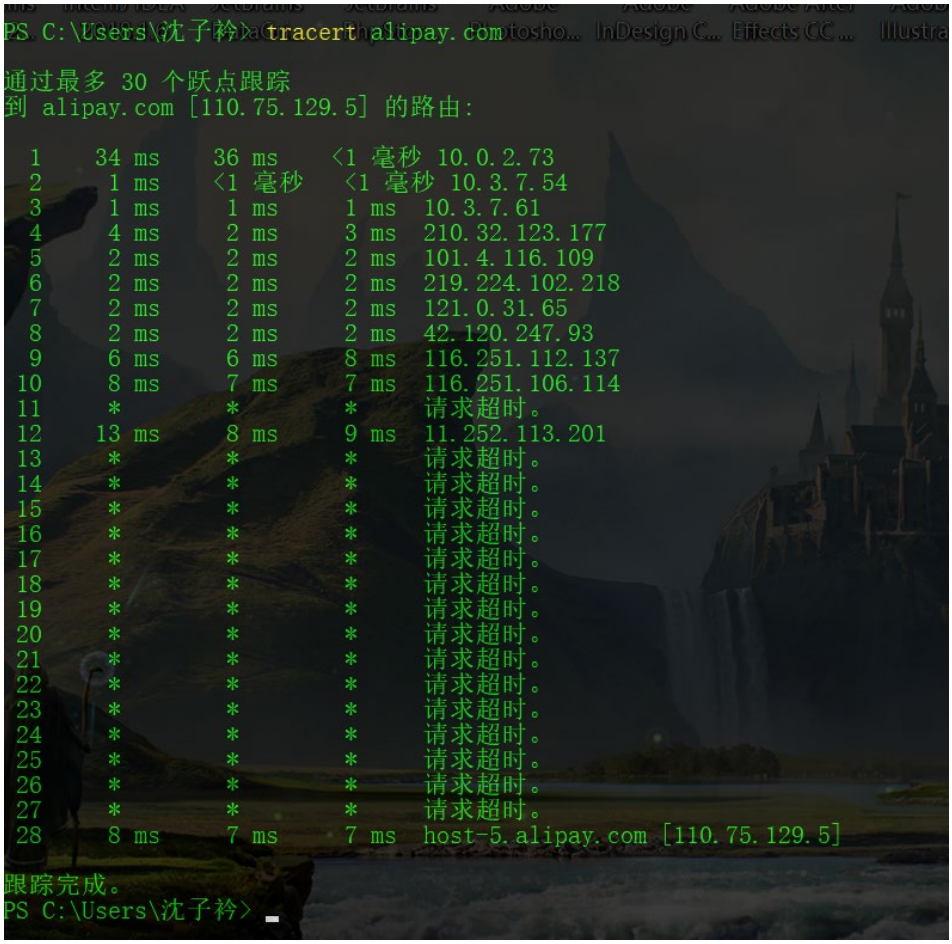
```
PS C:\Users\沈子衿> arp -a

接口: 192.168.99.1 --- 0x7
  Internet 地址         物理地址              类型
  192.168.99.255        ff-ff-ff-ff-ff-ff     静态
  224.0.0.2             01-00-5e-00-00-02     静态
  224.0.0.5             01-00-5e-00-00-05     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
  239.192.152.143       01-00-5e-40-98-8f     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: 10.110.18.82 --- 0xb
  Internet 地址         物理地址              类型
  10.110.18.1           5c-dd-70-00-95-03     动态
  10.110.18.3           d0-17-c2-1f-df-f3     动态
  10.110.18.5           00-e0-4c-36-0d-28     动态
  10.110.18.23          9c-5c-8e-1e-c0-db     动态
  10.110.18.45          00-e0-4d-36-be-8f     动态
  10.110.18.98          30-65-ec-96-d5-fa     动态
  10.110.18.189         70-5a-0f-bf-1a-6a     动态
  10.110.18.200         b8-97-5a-ec-7f-1c     动态
  10.110.18.205         a0-8c-fd-ff-55-3f     动态
  10.110.18.216         1c-39-47-df-72-e4     动态
  10.110.18.251         1c-39-47-30-87-e6     动态
```

5) 显示从本机到达一个特定地址的路由

可以使用 tracert 命令来监视到达一个特定地址的路由，需要花费一定的时间。此处以 alipay.com 为例：



```
PS C:\Users\沈子衿> tracert alipay.com

通过最多 30 个跃点跟踪
到 alipay.com [110.75.129.5] 的路由:

  1    34 ms    36 ms    <1 毫秒 10.0.2.73
  2     1 ms    <1 毫秒   <1 毫秒 10.3.7.54
  3     1 ms     1 ms     1 ms  10.3.7.61
  4     4 ms     2 ms     3 ms  210.32.123.177
  5     2 ms     2 ms     2 ms  101.4.116.109
  6     2 ms     2 ms     2 ms  219.224.102.218
  7     2 ms     2 ms     2 ms  121.0.31.65
  8     2 ms     2 ms     2 ms  42.120.247.93
  9     6 ms     6 ms     8 ms  116.251.112.137
 10     8 ms     7 ms     7 ms  116.251.106.114
 11     *        *        *     请求超时。
 12    13 ms     8 ms     9 ms  11.252.113.201
 13     *        *        *     请求超时。
 14     *        *        *     请求超时。
 15     *        *        *     请求超时。
 16     *        *        *     请求超时。
 17     *        *        *     请求超时。
 18     *        *        *     请求超时。
 19     *        *        *     请求超时。
 20     *        *        *     请求超时。
 21     *        *        *     请求超时。
 22     *        *        *     请求超时。
 23     *        *        *     请求超时。
 24     *        *        *     请求超时。
 25     *        *        *     请求超时。
 26     *        *        *     请求超时。
 27     *        *        *     请求超时。
 28     8 ms     7 ms     7 ms  host-5.alipay.com [110.75.129.5]

跟踪完成。
PS C:\Users\沈子衿>
```

6) 显示某一个域名的 IP 地址

可以使用 nslookup 来查询一个域名的 ip 地址，当然，在 ping 一个域名的时候，也会显示它的 ip 地址：

```
PS C:\Users\沈子衿> nslookup alipay.com
服务器:  dns1.zju.edu.cn
Address:  10.10.0.21

非权威应答:
名称:    alipay.com
Addresses:  110.75.139.5
          110.75.129.5

PS C:\Users\沈子衿>
```

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

使用 netstat 命令可以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

```
活动连接

  协议   本地地址          外部地址          状态
  TCP    127.0.0.1:1580         LAPTOP-I210TIEM:1581   ESTABLISHED
  TCP    127.0.0.1:1581         LAPTOP-I210TIEM:1580   ESTABLISHED
  TCP    127.0.0.1:3178         LAPTOP-I210TIEM:54530  ESTABLISHED
  TCP    127.0.0.1:3179         LAPTOP-I210TIEM:3180   ESTABLISHED
  TCP    127.0.0.1:3180         LAPTOP-I210TIEM:3179   ESTABLISHED
  TCP    127.0.0.1:54530        LAPTOP-I210TIEM:3178   ESTABLISHED
  TCP    222.205.75.57:1968     17.252.156.71:5223     ESTABLISHED
  TCP    222.205.75.57:2170     52.230.7.59:https      ESTABLISHED
  TCP    222.205.75.57:2487     101.226.103.123:http   CLOSE_WAIT
  TCP    222.205.75.57:2667     101.227.139.187:8080   ESTABLISHED
  TCP    222.205.75.57:2976     i-db3p-cor001:https    ESTABLISHED
  TCP    222.205.75.57:3171     183.136.212.96:https   CLOSE_WAIT
  TCP    222.205.75.57:3175     219.146.244.169:http   CLOSE_WAIT
  TCP    222.205.75.57:3176     219.146.244.169:http   CLOSE_WAIT
  TCP    222.205.75.57:3177     219.146.244.169:http   CLOSE_WAIT
  TCP    222.205.75.57:3183     183.136.212.96:https   CLOSE_WAIT
  TCP    222.205.75.57:3185     219.146.244.169:http   CLOSE_WAIT
  TCP    222.205.75.57:3186     219.146.244.169:http   CLOSE_WAIT
  TCP    222.205.75.57:3187     183.134.56.22:http     CLOSE_WAIT
  TCP    222.205.75.57:3188     183.134.56.22:http     CLOSE_WAIT
  TCP    222.205.75.57:3189     219.146.244.169:http   CLOSE_WAIT
  TCP    222.205.75.57:3504     122.228.251.113:https  ESTABLISHED
  TCP    222.205.75.57:3530     115.231.141.190:http   ESTABLISHED
  TCP    222.205.75.57:3536     47.107.24.57:https     ESTABLISHED
  TCP    222.205.75.57:3554     101.89.125.211:https   ESTABLISHED
```

8) 显示本机的路由表信息，并手工添加一个路由

使用 route PRINT 命令可以显示当前本机的路由表情况：

```
PS C:\Users\沈子衿> route PRINT
===========================================================================
接口列表
  9...00 ff 3c 31 6e da ......Sangfor SSL VPN CS Support System VNIC
  7...0a 00 27 00 00 07 ......VirtualBox Host-Only Ethernet Adapter #3
 10...00 23 15 d8 04 0a ......Intel(R) Dual Band Wireless-AC 8265
 27...02 23 15 d8 04 0a ......Microsoft Wi-Fi Direct Virtual Adapter
 28...00 23 15 d8 04 0b ......Microsoft Wi-Fi Direct Virtual Adapter #3
 11...8c 16 45 32 c2 f1 ......Intel(R) Ethernet Connection (4) I219-V
 26...00 50 56 c0 00 01 ......VMware Virtual Ethernet Adapter for VMnet1
 17...00 50 56 c0 00 08 ......VMware Virtual Ethernet Adapter for VMnet8
 23...00 ff 9c d0 e5 e9 ......TAP-Windows Adapter V9
 62.............................浙大VPN(玉泉校区)
 24...00 23 15 d8 04 0e ......Bluetooth Device (Personal Area Network)
  1...........................Software Loopback Interface 1
===========================================================================

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     10.110.18.1     10.110.18.82     4516
          0.0.0.0          0.0.0.0         在链路上     222.205.75.57       36
         10.0.0.0        255.0.0.0     10.110.18.1     10.110.18.82     4261
       10.0.2.73  255.255.255.255     10.110.18.1     10.110.18.82     4261
      10.110.18.0    255.255.255.0         在链路上     10.110.18.82     4516
     10.110.18.82  255.255.255.255         在链路上     10.110.18.82     4516
    10.110.18.255  255.255.255.255         在链路上     10.110.18.82     4516
        127.0.0.0        255.0.0.0         在链路上        127.0.0.1     4556
        127.0.0.1  255.255.255.255         在链路上        127.0.0.1     4556
  127.255.255.255  255.255.255.255         在链路上        127.0.0.1     4556
    192.168.23.0    255.255.255.0         在链路上      192.168.23.1     4516
    192.168.23.1  255.255.255.255         在链路上      192.168.23.1     4516
  192.168.23.255  255.255.255.255         在链路上      192.168.23.1     4516
    192.168.99.0    255.255.255.0         在链路上      192.168.99.1     4506
    192.168.99.1  255.255.255.255         在链路上      192.168.99.1     4506
  192.168.99.255  255.255.255.255         在链路上      192.168.99.1     4506
  192.168.133.0   255.255.255.0         在链路上     192.168.133.1     4516
```

使用 route ADD 命令可向计算机添加指定路由。值得注意的是，添加路由需要在管理员环境下才能实现：

```
PS C:\WINDOWS\system32> route add 157.0.0.0 MASK 255.0.0.0  157.55.80.1
 操作完成!
PS C:\WINDOWS\system32>
```

9) 显示本机的网络映射连接

```
PS C:\Users\沈子衿> net use
会记录新的网络连接。

列表是空的。
```

在没有参数的情况下，使用 net use 命令可以查看本机的网络映射连接。由于当前并没有网络映射连接，因此列表是空的。

10. 显示局域网内某台机器的共享资源

使用命令 net view \\<HostName>可以查看局域网中某台机器的共享资源列表。此时我们以本地机器为例，使用 net view \\localhost ：

```
PS C:\Users\沈子衿> net view \\localhost
列表是空的。
```

发现列表是空的。此时我们尝试使用 net share 命令为本地机器添加一个共享。注意，这里需要使用管理员模式运行：

```
PS C:\WINDOWS\system32> NET SHARE d.share=D:\share
d.share 共享成功。

PS C:\WINDOWS\system32> NET SHARE

共享名       资源                        注解

-------------------------------------------------------------------------------
C$           C:\                         默认共享
D$           D:\                         默认共享
IPC$                                     远程 IPC
E$           E:\                         默认共享
ADMIN$       C:\WINDOWS                  远程管理
d.share      D:\share
命令成功完成。

PS C:\WINDOWS\system32> net view
PS C:\WINDOWS\system32> net view \\localhost
在 \\localhost 的共享资源



共享名   类型   使用为   注释

-------------------------------------------------------------------------------
d.share  Disk
命令成功完成。
```

这个时候再执行 net view \\localhost，就能看到之前设置共享的 d.share 文件夹已经在共享名单上了，

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<cr>

Host: 任意字符串<cr>

<cr>

此处以 www.baidu.com 为例：

首先输入 telnet www.baidu.com 80 命令，进入百度的 telnet 页面

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

PS C:\Users\沈子衿> telnet www.baidu.com 80
```

然后输入 GET / HTTP/1.1 并两次回车即可。值得注意的是，windows 的 teinet 客户端是看不到输入的：

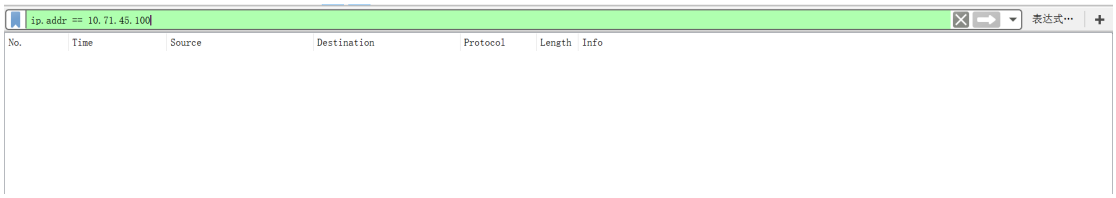
```
HTTP/1.0 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Content-Length: 14615
Content-Type: text/html
Date: Thu, 04 Oct 2018 13:26:08 GMT
Etag: "5ba38008-3917"
Last-Modified: Thu, 20 Sep 2018 11:10:00 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: BWS/1.1
Set-Cookie: BAIDUID=26D18367E1CF3C77C4B97CD79BE6EA6B:FG=1; expires=Thu, 31-Dec-37 23:55:55 GMT; max
age=2147483647; path=/; domain=.baidu.com
Set-Cookie: BIDUPSID=26D18367E1CF3C77C4B97CD79BE6EA6B; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age
2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1538659568; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domai
=.baidu.com
Vary: Accept-Encoding
X-Ua-Compatible: IE=Edge,chrome=1

<!DOCTYPE html><!--STATUS OK-->
<html>
<head>
        <meta http-equiv="content-type" content="text/html;charset=utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=Edge">
        <link rel="dns-prefetch" href="//ss1.bdstatic.com">
        <link rel="dns-prefetch" href="//t1.baidu.com"/>
        <link rel="dns-prefetch" href="//t2.baidu.com"/>
        <link rel="dns-prefetch" href="//t3.baidu.com"/>
        <link rel="dns-prefetch" href="//t10.baidu.com"/>
        <link rel="dns-prefetch" href="//t11.baidu.com"/>
```

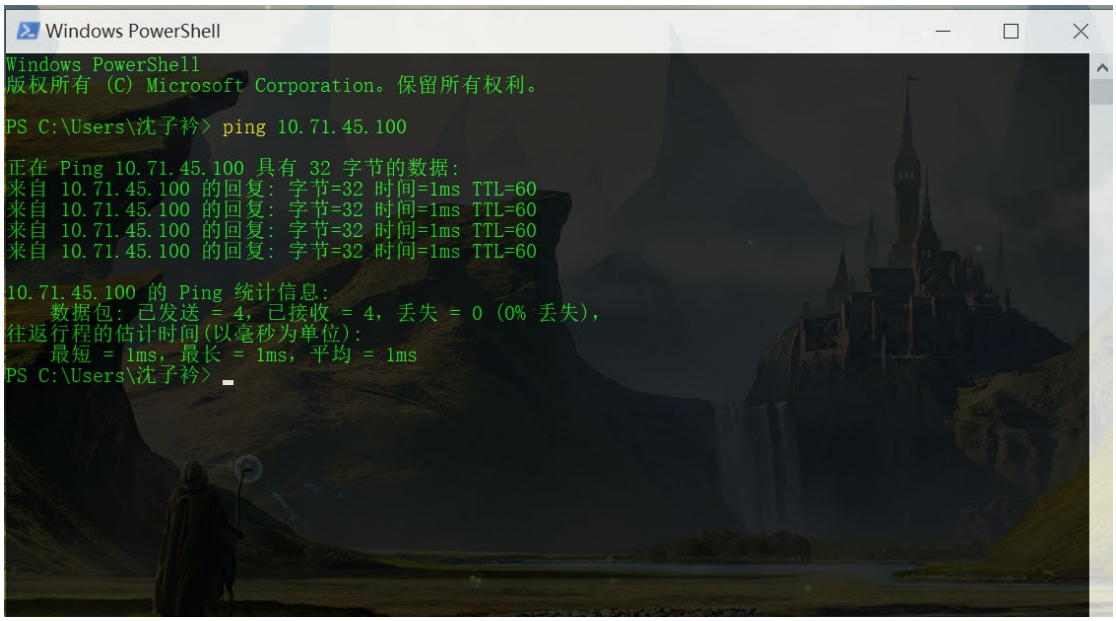
发现确实成功输出了 www.baidu.com 的 html 源代码，值得注意的是，由于编码问题，部分中文出现了乱码。

- **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？**

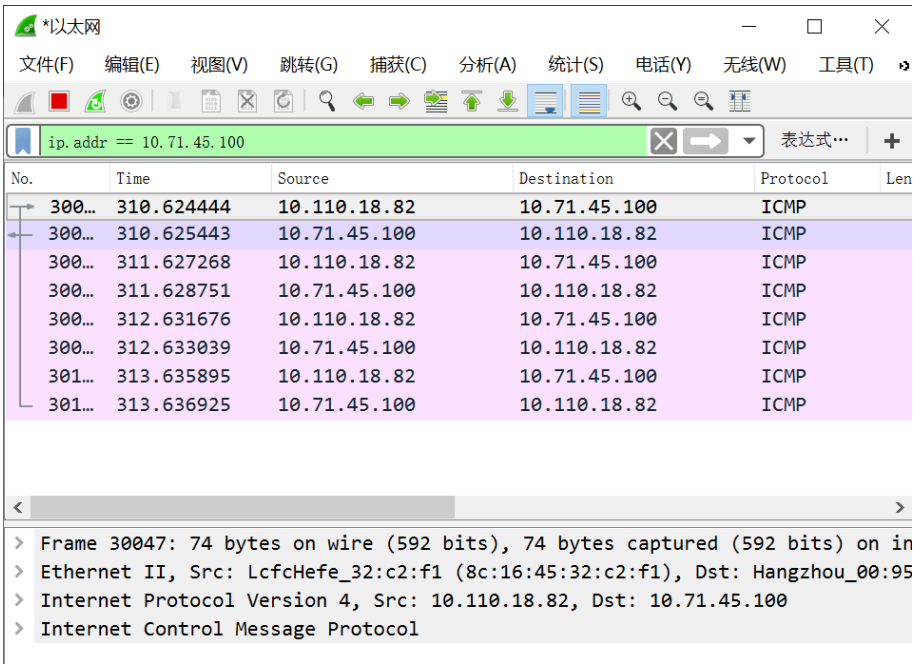
**答：**为防止 L2TP VPN 对实验结果造成干扰，我们以位与内网的计算机学院基础教学课程网站（10.71.45.100）为例。首先，在 wireshark 中配置显示过滤，只显示目的地为 10.71.45.100 的数据包信息。语法为：ip.addr == 10.71.45.100




此时，列表中还没有数据包。然后打开命令行，ping 10.71.45.100：



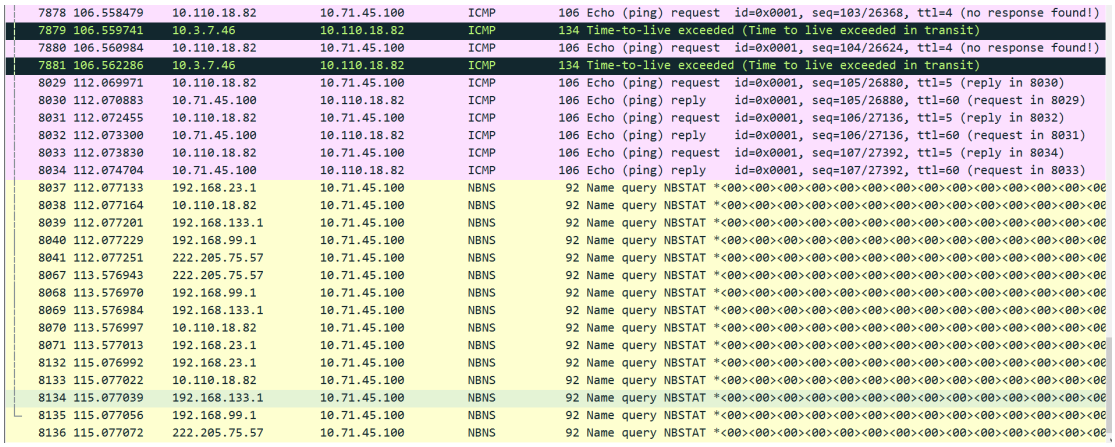

然后发现数据包列表上出现了 8 个被显示的包，分别是本地主机发送到远程主机上的包和远程主机响应的包：



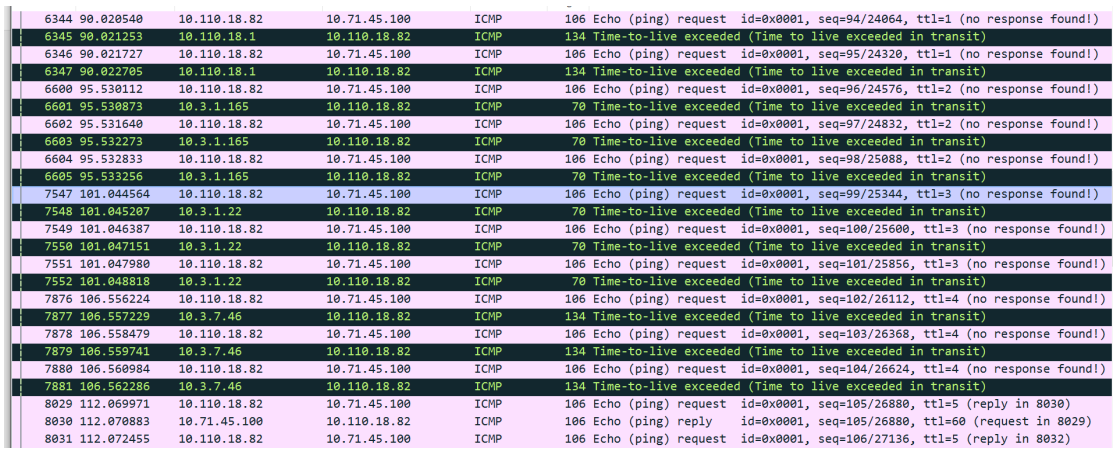

它们使用的协议均为 ICMP 协议。


- **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？**

以 10.71.45.100 为例：



| | | | | | |
|---|---|---|---|---|---|
| 7878 | 106.558479 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=103/26368, ttl=4 (no response found!) |
| 7879 | 106.559741 | 10.3.7.46 | 10.110.18.82 | ICMP | 134 Time-to-live exceeded (Time to live exceeded in transit) |
| 7880 | 106.560984 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=104/26624, ttl=4 (no response found!) |
| 7881 | 106.562286 | 10.3.7.46 | 10.110.18.82 | ICMP | 134 Time-to-live exceeded (Time to live exceeded in transit) |
| 8029 | 112.069971 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=105/26880, ttl=5 (reply in 8030) |
| 8030 | 112.070883 | 10.71.45.100 | 10.110.18.82 | ICMP | 106 Echo (ping) reply id=0x0001, seq=105/26880, ttl=60 (request in 8029) |
| 8031 | 112.072455 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=106/27136, ttl=5 (reply in 8032) |
| 8032 | 112.073300 | 10.71.45.100 | 10.110.18.82 | ICMP | 106 Echo (ping) reply id=0x0001, seq=106/27136, ttl=60 (request in 8031) |
| 8033 | 112.073830 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=107/27392, ttl=5 (reply in 8034) |
| 8034 | 112.074704 | 10.71.45.100 | 10.110.18.82 | ICMP | 106 Echo (ping) reply id=0x0001, seq=107/27392, ttl=60 (request in 8033) |
| 8037 | 112.077133 | 192.168.23.1 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8038 | 112.077164 | 10.110.18.82 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8039 | 112.077201 | 192.168.133.1 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8040 | 112.077229 | 192.168.99.1 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8041 | 112.077251 | 222.205.75.57 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8067 | 113.576943 | 222.205.75.57 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8068 | 113.576970 | 192.168.99.1 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8069 | 113.576984 | 192.168.133.1 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8070 | 113.576997 | 10.110.18.82 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8071 | 113.577013 | 192.168.23.1 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8132 | 115.076992 | 192.168.23.1 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8133 | 115.077022 | 10.110.18.82 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8134 | 115.077039 | 192.168.133.1 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8135 | 115.077056 | 192.168.99.1 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |
| 8136 | 115.077072 | 222.205.75.57 | 10.71.45.100 | NBNS | 92 Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00 |

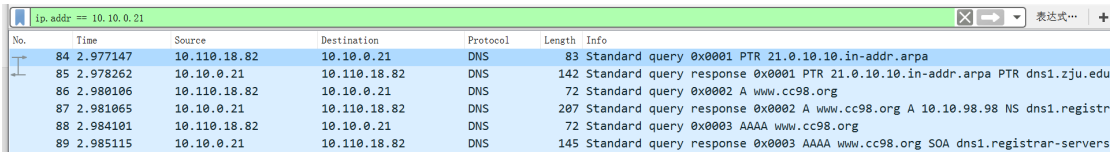


| | | | | | |
|---|---|---|---|---|---|
| 6344 | 90.020540 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=94/24064, ttl=1 (no response found!) |
| 6345 | 90.021253 | 10.110.18.1 | 10.110.18.82 | ICMP | 134 Time-to-live exceeded (Time to live exceeded in transit) |
| 6346 | 90.021727 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=95/24320, ttl=1 (no response found!) |
| 6347 | 90.022705 | 10.110.18.1 | 10.110.18.82 | ICMP | 134 Time-to-live exceeded (Time to live exceeded in transit) |
| 6600 | 95.530112 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=96/24576, ttl=2 (no response found!) |
| 6601 | 95.530873 | 10.3.1.165 | 10.110.18.82 | ICMP | 70 Time-to-live exceeded (Time to live exceeded in transit) |
| 6602 | 95.531640 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=97/24832, ttl=2 (no response found!) |
| 6603 | 95.532273 | 10.3.1.165 | 10.110.18.82 | ICMP | 70 Time-to-live exceeded (Time to live exceeded in transit) |
| 6604 | 95.532833 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=98/25088, ttl=2 (no response found!) |
| 6605 | 95.533256 | 10.3.1.165 | 10.110.18.82 | ICMP | 70 Time-to-live exceeded (Time to live exceeded in transit) |
| 7547 | 101.044564 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=99/25344, ttl=3 (no response found!) |
| 7548 | 101.045207 | 10.3.1.22 | 10.110.18.82 | ICMP | 70 Time-to-live exceeded (Time to live exceeded in transit) |
| 7549 | 101.046387 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=100/25600, ttl=3 (no response found!) |
| 7550 | 101.047151 | 10.3.1.22 | 10.110.18.82 | ICMP | 70 Time-to-live exceeded (Time to live exceeded in transit) |
| 7551 | 101.047980 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=101/25856, ttl=3 (no response found!) |
| 7552 | 101.048818 | 10.3.1.22 | 10.110.18.82 | ICMP | 70 Time-to-live exceeded (Time to live exceeded in transit) |
| 7876 | 106.556224 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=102/26112, ttl=4 (no response found!) |
| 7877 | 106.557229 | 10.3.7.46 | 10.110.18.82 | ICMP | 134 Time-to-live exceeded (Time to live exceeded in transit) |
| 7878 | 106.558479 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=103/26368, ttl=4 (no response found!) |
| 7879 | 106.559741 | 10.3.7.46 | 10.110.18.82 | ICMP | 134 Time-to-live exceeded (Time to live exceeded in transit) |
| 7880 | 106.560984 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=104/26624, ttl=4 (no response found!) |
| 7881 | 106.562286 | 10.3.7.46 | 10.110.18.82 | ICMP | 134 Time-to-live exceeded (Time to live exceeded in transit) |
| 8029 | 112.069971 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=105/26880, ttl=5 (reply in 8030) |
| 8030 | 112.070883 | 10.71.45.100 | 10.110.18.82 | ICMP | 106 Echo (ping) reply id=0x0001, seq=105/26880, ttl=60 (request in 8029) |
| 8031 | 112.072455 | 10.110.18.82 | 10.71.45.100 | ICMP | 106 Echo (ping) request id=0x0001, seq=106/27136, ttl=5 (reply in 8032) |

发现在整个过程中，存在 NBNS 和 ICMP 两种协议。

- **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？**

以 www.cc98.org 为例，由于可以认为 nslookup 只涉及本地主机和 DNS 服务器的交互，因此将显示过滤配置为 DNS 服务器的 IP 即可：

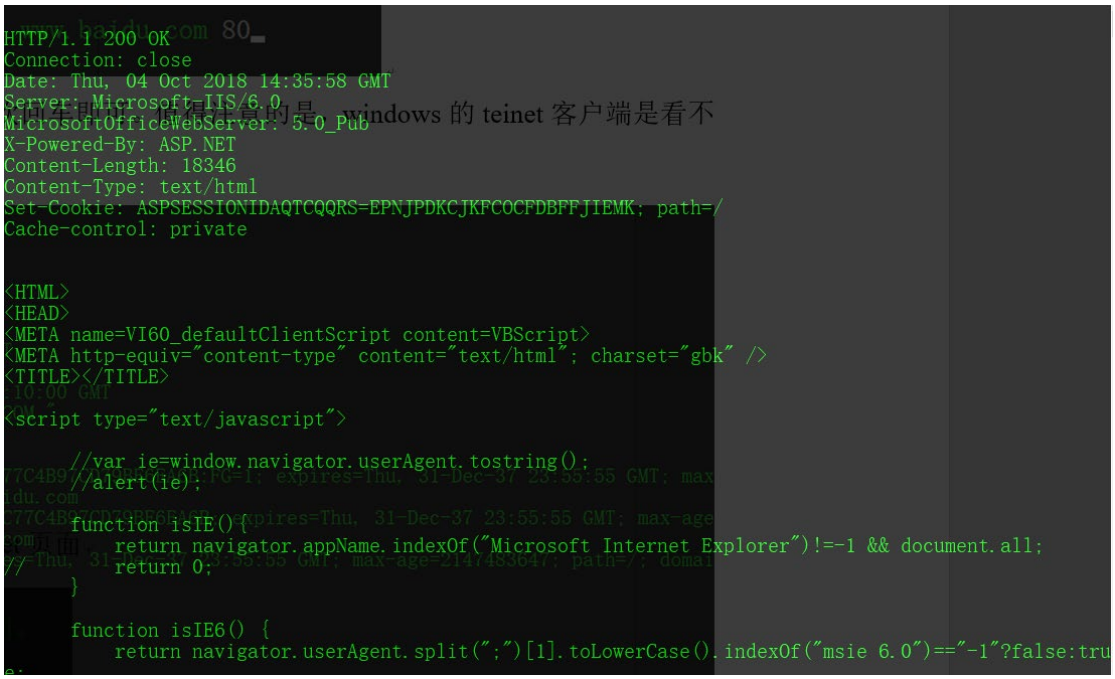
ip.addr == 10.10.0.21

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 84 | 2.977147 | 10.110.18.82 | 10.10.0.21 | DNS | 83 | Standard query 0x0001 PTR 21.0.10.10.in-addr.arpa |
| 85 | 2.978262 | 10.10.0.21 | 10.110.18.82 | DNS | 142 | Standard query response 0x0001 PTR 21.0.10.10.in-addr.arpa PTR dns1.zju.edu |
| 86 | 2.980106 | 10.110.18.82 | 10.10.0.21 | DNS | 72 | Standard query 0x0002 A www.cc98.org |
| 87 | 2.981065 | 10.10.0.21 | 10.110.18.82 | DNS | 207 | Standard query response 0x0002 A www.cc98.org A 10.10.98.98 NS dns1.registr |
| 88 | 2.984101 | 10.110.18.82 | 10.10.0.21 | DNS | 72 | Standard query 0x0003 AAAA www.cc98.org |
| 89 | 2.985115 | 10.10.0.21 | 10.110.18.82 | DNS | 145 | Standard query response 0x0003 AAAA www.cc98.org SOA dns1.registrar-servers |

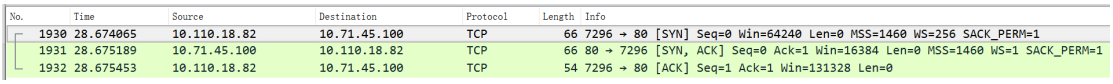
发现本地主机和 DNS 服务器互相发送了六个数据包，其协议为 DNS 协议。

- **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？**

以 10.71.45.100 为例：




在一开始建立 talnet 连接时，以下包被捕获：



| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1930 | 28.674065 | 10.110.18.82 | 10.71.45.100 | TCP | 66 | 7296 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 1931 | 28.675189 | 10.71.45.100 | 10.110.18.82 | TCP | 66 | 80 → 7296 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 WS=1 SACK_PERM=1 |
| 1932 | 28.675453 | 10.110.18.82 | 10.71.45.100 | TCP | 54 | 7296 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0 |

其协议均为 TCP 协议；

在请求 html 文本时，以下包被捕获：

| | | | | | | |
|---|---|---|---|---|---|---|
| 6850 | 104.711613 | 10.110.18.82 | 10.71.45.100 | TCP | 55 | 7296 → 80 [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=1 [TCP segment of a reasse |
| 6853 | 104.858631 | 10.71.45.100 | 10.110.18.82 | TCP | 60 | 80 → 7296 [ACK] Seq=1 Ack=2 Win=65534 Len=0 |
| 6863 | 105.177587 | 10.110.18.82 | 10.71.45.100 | TCP | 55 | 7296 → 80 [PSH, ACK] Seq=2 Ack=1 Win=131328 Len=1 [TCP segment of a reasse |
| 6867 | 105.360882 | 10.71.45.100 | 10.110.18.82 | TCP | 60 | 80 → 7296 [ACK] Seq=1 Ack=3 Win=65533 Len=0 |
| 6871 | 105.527769 | 10.110.18.82 | 10.71.45.100 | TCP | 55 | 7296 → 80 [PSH, ACK] Seq=3 Ack=1 Win=131328 Len=1 [TCP segment of a reasse |
| 6875 | 105.764060 | 10.71.45.100 | 10.110.18.82 | TCP | 60 | 80 → 7296 [ACK] Seq=1 Ack=4 Win=65532 Len=0 |
| 6881 | 106.065886 | 10.110.18.82 | 10.71.45.100 | TCP | 55 | 7296 → 80 [PSH, ACK] Seq=4 Ack=1 Win=131328 Len=1 [TCP segment of a reasse |
| 6883 | 106.266726 | 10.71.45.100 | 10.110.18.82 | TCP | 60 | 80 → 7296 [ACK] Seq=1 Ack=5 Win=65531 Len=0 |
| 6884 | 106.272219 | 10.110.18.82 | 10.71.45.100 | TCP | 55 | 7296 → 80 [PSH, ACK] Seq=5 Ack=1 Win=131328 Len=1 [TCP segment of a reasse |
| 6888 | 106.568005 | 10.71.45.100 | 10.110.18.82 | TCP | 60 | 80 → 7296 [ACK] Seq=1 Ack=6 Win=65530 Len=0 |
| 6889 | 106.568032 | 10.110.18.82 | 10.71.45.100 | TCP | 55 | 7296 → 80 [PSH, ACK] Seq=6 Ack=1 Win=131328 Len=1 [TCP segment of a reasse |
| 6891 | 106.776831 | 10.71.45.100 | 10.110.18.82 | TCP | 60 | 80 → 7296 [ACK] Seq=1 Ack=7 Win=65529 Len=0 |
| 8127 | 121.691701 | 10.110.18.82 | 10.71.45.100 | TCP | 55 | 7296 → 80 [PSH, ACK] Seq=7 Ack=1 Win=131328 Len=1 [TCP segment of a reasse |
| 8137 | 121.958113 | 10.71.45.100 | 10.110.18.82 | TCP | 60 | 80 → 7296 [ACK] Seq=1 Ack=8 Win=65528 Len=0 |
| 8144 | 122.183896 | 10.110.18.82 | 10.71.45.100 | TCP | 55 | 7296 → 80 [PSH, ACK] Seq=8 Ack=1 Win=131328 Len=1 [TCP segment of a reasse |
| 8160 | 122.360720 | 10.71.45.100 | 10.110.18.82 | TCP | 60 | 80 → 7296 [ACK] Seq=1 Ack=9 Win=65527 Len=0 |

他们均使用 TCP 协议。易知这些包所承载的是输入的文本，他们不保存在本地的键盘缓冲区上，直接发送给了远程主机，这里我们也了解了 Windows Telnet 客户端不显示输入文本的原因。

在发送 HTTP 请求时，以下包被捕捉：

| | | | | | | |
|---|---|---|---|---|---|---|
| 8325 | 128.799924 | 10.110.18.82 | 10.71.45.100 | TCP | 54 | 7296 → 80 [ACK] Seq=19 Ack=16061 Win=131328 Len=0 |
| 8326 | 128.799622 | 10.71.45.100 | 10.110.18.82 | TCP | 1514 | 80 → 7296 [ACK] Seq=16061 Ack=19 Win=65517 Len=1460 [TCP segment of a reas |
| 8327 | 128.799638 | 10.110.18.82 | 10.71.45.100 | TCP | 54 | 7296 → 80 [ACK] Seq=19 Ack=17521 Win=131328 Len=0 |
| 8328 | 128.799922 | 10.71.45.100 | 10.110.18.82 | HTTP | 1179 | HTTP/1.1 200 OK (text/html) |
| 8329 | 128.799962 | 10.110.18.82 | 10.71.45.100 | TCP | 54 | 7296 → 80 [ACK] Seq=19 Ack=18647 Win=130048 Len=0 |
| 8335 | 128.971415 | 10.110.18.82 | 10.71.45.100 | TCP | 54 | 7296 → 80 [RST, ACK] Seq=19 Ack=18647 Win=0 Len=0 |

这里同时使用了 HTTP 协议和 TCP 协议。

## 六、 实验结果与分析

- **WireShark 的两种过滤器有什么不同？**

答：捕获过滤器启用时，WireShark 将忽略不满足条件的包，不会将其保存到列表中，之后通过显示过滤器定向检索也是找不到的；但显示过滤器只是在窗口上根据条件对捕获结果进行了筛选，本质上依然捕获了这些包，只要显示过滤器的条件改变，这些包还是可以被检索并分析内容的。

- **哪些网络命令会产生在 WireShark 中产生数据包，为什么？**

答：在 Ping.exe，Netstat.exe， Telnet.exe，Tracert.exe，Arp.exe，Ipconfig.exe，Net.exe，Route.exe 这些命令中， ping，netstat，telnet，tracert 是会产生数据包的，因为他们需要远程主机的响应或需要从远程主机上获取数据；ipconfig、route 是不会产生数据包的，因为设配器设置、子网内各主机 IP/MAC 和路由表等数据是存储或缓存在本地主机里的，不需要通过向远程主机发送请求的方式获取。Net 命令有些是会产生数据包的，如 net view 查询远程主机共享文件或文件夹时；有些则不会产生数据包，如 net share 配置自身共享文件夹时。Arp 命令在缓存中不存在某一 IP 地址对应的 MAC 地址时会发出 Arp 请求在局域网中查询，也会产生数据包。

- **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？**

答：①Ping 发送的是一个 ICMP 数据包；

②当缓存中不存在某一 IP 地址对应的 MAC 地址时，才会发送 ARP 请求到局域网查询，产生 ARP 消息；

③ping 一个 IP 地址只会在起点、终点及其中继节点之间产生 ICMP 数据包。但 ping 一个域名时，因解析域名需要，会在主机和 DNS 服务器之间产生 ICMP 数据包。

## 七、 讨论、心得

本次实验是计算机网络的第一次实验，难度不大，但量较大。通过本次实验，我了解了 Wireshark 的功能和基本用法、过滤器原语的使用以及基本网络命令的功能与使用方法，受益匪浅。

本次实验对自学的要求较高。为了做好每一题，我不得不查阅大量参考文献，但这也帮助我对基本概念有了更好的理解。

因为学校特殊的网络环境，本次实验也遇到了不少困难，但我凭借查阅资料和生活经验妥善解决了它们。我想，这将有助于未来工作的开展。

在未来的实验中，我会再接再厉。