



本科生实验报告

实验课程: 计算机网络实验

专业名称: 计算机科学与技术 (超算方向)

学生姓名: 李钰

学生学号: 19335112

实验成绩:

报告时间:

一、实验内容

实验一 Ping 命令

1. ping www.sohu.com

```
C:\Users\16435>ping www.sohu.com

正在 Ping fgzyd.a.sohu.com [2409:8c00:3001::4] 具有 32 字节的数据:
来自 2409:8c00:3001::4 的回复: 时间=72ms
来自 2409:8c00:3001::4 的回复: 时间=77ms
来自 2409:8c00:3001::4 的回复: 时间=73ms
来自 2409:8c00:3001::4 的回复: 时间=73ms

2409:8c00:3001::4 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 = 72ms, 最长 = 77ms, 平均 = 73ms

C:\Users\16435>
```

- ✧ 结果中可显示域名 www.sohu.com IP 地址是 2409:8c00:3001::4, 体现 ping 命令从域名中查找对应 IP 地址的作用。
- ✧ 结果显示一共发送了 4 个数据包, 且全部被接收, 与对方主机往返一次所用的时间最快为 72ms, 最长为 77ms, 平均用时 73ms。

2. ping 2409:8c00:3001::4

```
PS C:\Windows\system32> ping 2409:8c00:3001::4

正在 Ping 2409:8c00:3001::4 具有 32 字节的数据:
来自 2409:8c00:3001::4 的回复: 时间=81ms
来自 2409:8c00:3001::4 的回复: 时间=79ms
来自 2409:8c00:3001::4 的回复: 时间=80ms
来自 2409:8c00:3001::4 的回复: 时间=81ms

2409:8c00:3001::4 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 = 79ms, 最长 = 81ms, 平均 = 80ms

PS C:\Windows\system32>
```

- ✧ 利用第一次 ping 得到的 IP 地址, 更改为 ping IP, 连通成功。

3. ping 118.228.148.143

```
C:\Users\16435>ping 118.228.148.143

正在 Ping 118.228.148.143 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

118.228.148.143 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

✧ 由于搜狐的 IP 地址已作更改，所以此 IP 地址失效，本机发出的四个数据包全部没有在规定时间内(TTL，生存时间)内被接收，所以显示请求超时，本机与 IP 地址为 118.228.148.143 的主机网络连通失败。

4. ping www.sysu.edu.cn -t

```
C:\Users\16435>ping www.sysu.edu.cn -t

正在 Ping pisces-1.sysu.edu.cn [2001:250:3002:10::8] 具有 32 字节的数据:
来自 2001:250:3002:10::8 的回复: 时间=2ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=5ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=7ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=2ms
来自 2001:250:3002:10::8 的回复: 时间=6ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=8ms
来自 2001:250:3002:10::8 的回复: 时间=2ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=10ms
来自 2001:250:3002:10::8 的回复: 时间=8ms
来自 2001:250:3002:10::8 的回复: 时间=2ms
来自 2001:250:3002:10::8 的回复: 时间=5ms
来自 2001:250:3002:10::8 的回复: 时间=12ms
来自 2001:250:3002:10::8 的回复: 时间=2ms
来自 2001:250:3002:10::8 的回复: 时间=2ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=7ms
来自 2001:250:3002:10::8 的回复: 时间=6ms
来自 2001:250:3002:10::8 的回复: 时间=6ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=2ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=2ms
来自 2001:250:3002:10::8 的回复: 时间=5ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
```



```

来自 2001:250:3002:10::8 的回复: 时间=5ms
来自 2001:250:3002:10::8 的回复: 时间=6ms
来自 2001:250:3002:10::8 的回复: 时间=3ms
来自 2001:250:3002:10::8 的回复: 时间=2ms
来自 2001:250:3002:10::8 的回复: 时间=8ms
来自 2001:250:3002:10::8 的回复: 时间=2ms
来自 2001:250:3002:10::8 的回复: 时间=4ms
来自 2001:250:3002:10::8 的回复: 时间=9ms
来自 2001:250:3002:10::8 的回复: 时间=2ms

2001:250:3002:10::8 的 Ping 统计信息:
    数据包: 已发送 = 56, 已接收 = 56, 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 = 12ms, 平均 = 4ms
Control-C
^C
C:\Users\16435>

```

- ✧ 该命令表示连续向 www.sysu.edu.cn 发送 ping 测试报文，直至使用 ctrl + C 键

5. ping -r 6 -l 200 172.18.187.254

```

C:\Users\16435>ping -r 6 -l 200 172.18.187.254

正在 Ping 172.18.187.254 具有 200 字节的数据:
来自 172.18.187.254 的回复: 字节=200 时间=24ms TTL=252
    路由: 10.44.36.202 ->
            10.44.32.201 ->
            10.44.185.201 ->
            172.18.187.254 ->
            10.44.32.202 ->
            10.44.36.201
来自 172.18.187.254 的回复: 字节=200 时间=74ms TTL=252
    路由: 10.44.36.202 ->
            10.44.32.201 ->
            10.44.185.201 ->
            172.18.187.254 ->
            10.44.32.202 ->
            10.44.36.201
来自 172.18.187.254 的回复: 字节=200 时间=63ms TTL=252
    路由: 10.44.36.202 ->
            10.44.32.201 ->
            10.44.185.201 ->
            172.18.187.254 ->
            10.44.32.202 ->
            10.44.36.201
来自 172.18.187.254 的回复: 字节=200 时间=46ms TTL=252
    路由: 10.44.36.202 ->
            10.44.32.201 ->
            10.44.185.201 ->
            172.18.187.254 ->
            10.44.32.202 ->
            10.44.36.201

172.18.187.254 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 = 24ms, 最长 = 74ms, 平均 = 51ms
C:\Users\16435>

```

- ✧ 该命令记录了向 IP 地址为 172.18.187.254 发送 4 个大小为 200 字节的数据包时，所经历的 6 个路由

6. ping -s 4 -l 200 172.18.187.254

```
C:\Users\16435>ping -s 4 -l 200 172.18.187.254

正在 Ping 172.18.187.254 具有 200 字节的数据:
来自 172.18.187.254 的回复: 字节=200 时间=7ms TTL=252
    时间戳: 172.19.63.254 : 30470028 ->
                10.44.36.201 : 59270005 ->
                10.44.32.202 : 59270011 ->
                10.44.185.202 : 30470000
来自 172.18.187.254 的回复: 字节=200 时间=9ms TTL=252
    时间戳: 172.19.63.254 : 30471041 ->
                10.44.36.201 : 59271025 ->
                10.44.32.202 : 59271021 ->
                10.44.185.202 : 30471020
来自 172.18.187.254 的回复: 字节=200 时间=10ms TTL=252
    时间戳: 172.19.63.254 : 30472056 ->
                10.44.36.201 : 59272035 ->
                10.44.32.202 : 59272031 ->
                10.44.185.202 : 30472030
来自 172.18.187.254 的回复: 字节=200 时间=7ms TTL=252
    时间戳: 172.19.63.254 : 30473074 ->
                10.44.36.201 : 59273055 ->
                10.44.32.202 : 59273051 ->
                10.44.185.202 : 30473050

172.18.187.254 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 = 7ms, 最长 = 10ms, 平均 = 8ms

C:\Users\16435>
```

- ✧ 该条命令是指设置了 4 个时间戳，向 IP 地址为 172.18.187.254 的主机发送数据包，得到了接受回复。

实验二 tracert 命令

1. tracert www.sina.com

```
C:\Users\16435>tracert www.sina.com

通过最多 30 个跃点跟踪
到 spool.grid.sinaedge.com [2409:8c34:2000:2::17:73] 的路由:

 1      3 ms      8 ms      2 ms  2001:250:3002:4240::1
 2      4 ms      1 ms      2 ms  fd44:1024::ff01
 3      3 ms      3 ms      3 ms  fd04:110::ff01
 4      6 ms      5 ms      3 ms  fd00:110::ff02
 5      5 ms      5 ms      5 ms  cernet2.net [2001:da8:a2:102::1]
 6      3 ms      3 ms      3 ms  cernet2.net [2001:da8:a2:11::1]
 7      5 ms      4 ms      9 ms  2001:da8:2:104::1
 8     15 ms     15 ms     13 ms  2001:da8:2:17::2
 9     32 ms     30 ms     30 ms  2001:da8:2:2b::1
10     35 ms     35 ms     35 ms  2001:da8:2:13::1
11     53 ms     52 ms     49 ms  2001:da8:2:703::2
12     51 ms     49 ms     50 ms  2409:8080:0:3:2e1:283:1:0
13     50 ms     49 ms     50 ms  2409:8080:0:1:203:2e1:1:0
14     50 ms     48 ms     49 ms  2409:8080:1:2:201:203:1:1
15     65 ms     64 ms     64 ms  2409:8080:1:2:201:1001:0:1
16     66 ms     66 ms     71 ms  2409:8080:1:2:1001:1071:0:1
17     85 ms     83 ms     83 ms  2409:8034:0:166::1
18     81 ms     81 ms     80 ms  2409:8034:3002:205::1
19     80 ms     80 ms     79 ms  2409:8034:3012:501::1
20     85 ms     83 ms     83 ms  2409:8c34:2000:2::17:73

跟踪完成。

C:\Users\16435>
```

- ✧ traert 是路由跟踪实用程序，用于获得 IP 数据报访问目标时从本地计算机到目的主机的路径信息。
- ✧ 本条命令，跟踪了 www.sina.com 路由，结果显示，经过 20 个路由，最终本机与该网络联通成功。

2. tracert -d 172.16.0.88

```
C:\Windows\system32>tracert -d 172.16.0.88

通过最多 30 个跃点跟踪到 172.16.0.88 的路由

 1    <1 毫秒    <1 毫秒    <1 毫秒  172.16.0.1
 2  172.16.0.1  报告：无法访问目标主机。

跟踪完成。

C:\Windows\system32>
```

- ✧ 由结果可知，本机与 IP 地址为 172.16.0.88 的主机网络连接失败。

3. tracert -h 30 2409:8c34:2000:2::17:73

```
PS C:\Windows\system32> tracert -h 30 2409:8c34:2000:2::17:73
通过最多 30 个跃点跟踪到 2409:8c34:2000:2::17:73 的路由

 1      4 ms      2 ms      1 ms  2001:250:3002:4252::1
 2      7 ms      2 ms      1 ms  fd44:1025::ff01
 3      2 ms      2 ms      5 ms  fd04:110::ff01
 4      5 ms      6 ms      4 ms  fd00:110::ff02
 5      7 ms      5 ms      6 ms  cernet2.net [2001:da8:a2:102::1]
 6      3 ms      3 ms      2 ms  cernet2.net [2001:da8:a2:11::1]
 7      7 ms      7 ms      4 ms  2001:da8:2:104::1
 8     15 ms     19 ms     15 ms  2001:da8:2:17::2
 9     31 ms     31 ms     32 ms  2001:da8:2:2b::1
10     35 ms     38 ms     35 ms  2001:da8:2:13::1
11     69 ms     52 ms     54 ms  2001:da8:2:703::2
12     52 ms     49 ms     50 ms  2409:8080:0:3:2e1:283::
13     50 ms     49 ms     55 ms  2409:8080:0:1:204:2e1:1:0
14     50 ms     50 ms     52 ms  2409:8080:1:2:204:204:1:1
15     82 ms     81 ms     82 ms  2409:8080:1:2:204:1002:1:1
16     84 ms     84 ms     85 ms  2409:8080:1:2:1002:1072:0:1
17     85 ms     83 ms     84 ms  2409:8034:0:268::1
18     92 ms     81 ms     82 ms  2409:8034:3002:200::1
19     86 ms     87 ms     86 ms  2409:8034:3012:501::1
20     85 ms     86 ms     85 ms  2409:8c34:2000:2::17:73

跟踪完成。
PS C:\Windows\system32>
```

- ✧ 设置搜索目标的路径中存在的跃点最大数为 30，实际上通过了 20 个路由及跟踪到了目的地址。

实验三 ipconfig

1. ipconfig

```
C:\Users\16435>ipconfig

Windows IP 配置

以太网适配器 以太网:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

以太网适配器 VirtualBox Host-Only Network:

    连接特定的 DNS 后缀 . . . . . :
    本地链接 IPv6 地址. . . . . : fe80::7c4d:bcd5:c9:953%8
    IPv4 地址 . . . . . : 192.168.56.1
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . :

无线局域网适配器 本地连接* 1:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 本地连接* 2:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :

无线局域网适配器 WLAN:

    连接特定的 DNS 后缀 . . . . . :
    IPv6 地址 . . . . . : 2001:250:3002:4240:893e:edc6:780c:97e2
    临时 IPv6 地址. . . . . : 2001:250:3002:4240:84d6:4830:dc7a:c434
    本地链接 IPv6 地址. . . . . : fe80::893e:edc6:780c:97e2%2
    IPv4 地址 . . . . . : 172.19.44.222
    子网掩码 . . . . . : 255.255.192.0
    默认网关. . . . . : fe80::a68:8dff:fea5:1e01%2
                        172.19.63.254

以太网适配器 蓝牙网络连接:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :
```

✧ 如上图，通过 ipconfig 命令，显示了本机所有适配器的基本 TCP/IP 配置

2. ipconfig /all

✧ 如下图，利用 ipconfig /all 命令，显示了所有适配器的完整 TCP/IP 配置

```
C:\Users\16435>ipconfig /all
```

Windows IP 配置

```
主机名 . . . . . : LAPTOP-7CPJJ3CS
主 DNS 后缀 . . . . . :
节点类型 . . . . . : 混合
IP 路由已启用 . . . . . : 否
WINS 代理已启用 . . . . . : 否
```

以太网适配器 以太网:

```
媒体状态 . . . . . : 媒体已断开连接
连接特定的 DNS 后缀 . . . . . :
描述 . . . . . : Realtek PCIe GbE Family Controller
物理地址. . . . . : D0-5F-64-35-F9-5D
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
```

以太网适配器 VirtualBox Host-Only Network:

```
连接特定的 DNS 后缀 . . . . . :
描述 . . . . . : VirtualBox Host-Only Ethernet Adapter
物理地址. . . . . : 0A-00-27-00-00-08
DHCP 已启用 . . . . . : 否
自动配置已启用. . . . . : 是
本地链接 IPv6 地址. . . . . : fe80::7c4d:bcd5:c9:953%8(首选)
IPv4 地址 . . . . . : 192.168.56.1(首选)
子网掩码 . . . . . : 255.255.255.0
默认网关. . . . . :
DHCPv6 IAID . . . . . : 671744039
DHCPv6 客户端 DUID . . . . . : 00-01-00-01-24-FD-16-BA-D0-5F-64-35-F9-5D
DNS 服务器 . . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::2%1
                          fec0:0:0:ffff::3%1
TCP/IP 上的 NetBIOS . . . . . : 已启用
```

无线局域网适配器 本地连接* 1:

```
媒体状态 . . . . . : 媒体已断开连接
连接特定的 DNS 后缀 . . . . . :
描述 . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
物理地址. . . . . : 84-FD-D1-EA-9E-59
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
```

无线局域网适配器 本地连接* 2:

```
媒体状态 . . . . . : 媒体已断开连接
连接特定的 DNS 后缀 . . . . . :
描述 . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
物理地址. . . . . : 86-FD-D1-EA-9E-58
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
```

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. . . . . : 84-FD-D1-EA-9E-58
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
IPv6 地址 . . . . . : 2001:250:3002:4240:893e:edc6:780c:97e2(首选)
临时 IPv6 地址. . . . . : 2001:250:3002:4240:84d6:4830:dc7a:c434(首选)
本地链接 IPv6 地址. . . . . : fe80::893e:edc6:780c:97e2%2(首选)
IPv4 地址 . . . . . : 172.19.44.222(首选)
子网掩码 . . . . . : 255.255.192.0
获得租约的时间 . . . . . : 2021年3月4日 16:09:22
租约过期的时间 . . . . . : 2021年3月4日 18:09:23
默认网关. . . . . : fe80::a68:8dff:fea5:1e01%2
                          172.19.63.254
DHCP 服务器 . . . . . : 125.217.174.123
DHCPv6 IAID . . . . . : 42270161
DHCPv6 客户端 DUID . . . . . : 00-01-00-01-24-FD-16-BA-D0-5F-64-35-F9-5D
DNS 服务器 . . . . . : 10.8.4.4
                          10.8.8.8
TCP/IP 上的 NetBIOS . . . . . : 已启用
```

以太网适配器 蓝牙网络连接:

```
媒体状态 . . . . . : 媒体已断开连接
连接特定的 DNS 后缀 . . . . . :
描述 . . . . . : Bluetooth Device (Personal Area Network)
物理地址. . . . . : 84-FD-D1-EA-9E-5C
DHCP 已启用 . . . . . : 是
自动配置已启用. . . . . : 是
```

```
C:\Users\16435>
```

实验四 netstat 命令

1. netstat -an

✧ 该命令显示了所有活动的 TCP 链接以及计算机侦听的 TCP 和 UCP 端口。

```
C:\Users\16435>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    0.0.0.0:5021       0.0.0.0:0         LISTENING
TCP    0.0.0.0:5040       0.0.0.0:0         LISTENING
TCP    0.0.0.0:8473       0.0.0.0:0         LISTENING
TCP    0.0.0.0:15000      0.0.0.0:0         LISTENING
TCP    0.0.0.0:19531      0.0.0.0:0         LISTENING
TCP    0.0.0.0:29917      0.0.0.0:0         LISTENING
TCP    0.0.0.0:49664      0.0.0.0:0         LISTENING
TCP    0.0.0.0:49665      0.0.0.0:0         LISTENING
TCP    0.0.0.0:49666      0.0.0.0:0         LISTENING
TCP    0.0.0.0:49667      0.0.0.0:0         LISTENING
TCP    0.0.0.0:49670      0.0.0.0:0         LISTENING
TCP    0.0.0.0:49671      0.0.0.0:0         LISTENING
TCP    0.0.0.0:54321      0.0.0.0:0         LISTENING
TCP    0.0.0.0:62990      0.0.0.0:0         LISTENING
TCP    0.0.0.0:62993      0.0.0.0:0         LISTENING
TCP    127.0.0.1:4096     0.0.0.0:0         LISTENING
TCP    127.0.0.1:4301     0.0.0.0:0         LISTENING
TCP    127.0.0.1:4709     0.0.0.0:0         LISTENING
TCP    127.0.0.1:8911     0.0.0.0:0         LISTENING
TCP    127.0.0.1:9080     0.0.0.0:0         LISTENING
TCP    127.0.0.1:62427    0.0.0.0:0         LISTENING
TCP    172.19.44.222:139  0.0.0.0:0         LISTENING
TCP    172.19.44.222:54539 183.232.96.62:80  ESTABLISHED
TCP    172.19.44.222:54553 39.156.80.227:80  CLOSE_WAIT
TCP    172.19.44.222:54684 120.241.186.18:443 CLOSE_WAIT
TCP    172.19.44.222:58445 120.204.10.154:443 CLOSE_WAIT
TCP    172.19.44.222:58451 120.241.186.232:443 CLOSE_WAIT
TCP    172.19.44.222:58472 120.204.10.154:443 CLOSE_WAIT
TCP    172.19.44.222:58480 183.192.199.123:443 CLOSE_WAIT
TCP    172.19.44.222:58512 120.238.157.240:443 CLOSE_WAIT
TCP    172.19.44.222:62997 117.144.237.40:80  ESTABLISHED
TCP    172.19.44.222:63010 117.144.237.151:80  ESTABLISHED
TCP    172.19.44.222:63101 112.60.8.41:14000  CLOSE_WAIT
TCP    172.19.44.222:63163 216.58.221.240:443 TIME_WAIT
TCP    172.19.44.222:63168 120.232.181.162:443 TIME_WAIT
TCP    172.19.44.222:63188 183.232.231.174:443 CLOSE_WAIT
TCP    172.19.44.222:63189 183.232.231.174:443 CLOSE_WAIT
```



```

TCP 172.19.44.222:63191 120.241.147.225:443 ESTABLISHED
TCP 172.19.44.222:63198 183.232.231.174:80 ESTABLISHED
TCP 172.19.44.222:63202 120.92.94.32:80 TIME_WAIT
TCP 172.19.44.222:63218 120.241.16.104:36688 TIME_WAIT
TCP 172.19.44.222:63226 120.241.16.104:36688 TIME_WAIT
TCP 172.19.44.222:63247 120.92.208.196:80 SYN_SENT
TCP 172.19.44.222:63248 120.92.208.196:80 SYN_SENT
TCP 172.19.44.222:63249 183.232.231.172:443 FIN_WAIT_1
TCP 172.19.44.222:63250 122.51.63.211:302 SYN_SENT
TCP 172.19.44.222:63251 183.232.231.172:443 FIN_WAIT_1
TCP 172.19.44.222:63683 106.14.225.219:1883 ESTABLISHED
TCP 172.19.44.222:64100 40.119.211.203:443 ESTABLISHED
TCP 172.19.44.222:64126 222.186.180.89:1015 ESTABLISHED
TCP 172.19.44.222:64324 119.96.205.214:80 CLOSE_WAIT
TCP 172.19.44.222:64338 119.96.205.214:80 CLOSE_WAIT
TCP 172.19.44.222:64382 39.156.80.76:80 CLOSE_WAIT
TCP 172.19.44.222:64403 36.156.49.184:443 CLOSE_WAIT
TCP 172.19.44.222:64463 120.238.157.238:80 CLOSE_WAIT
TCP 172.19.44.222:64649 120.241.179.34:80 ESTABLISHED
TCP 172.19.44.222:64665 120.241.186.232:443 CLOSE_WAIT
TCP 192.168.56.1:139 0.0.0.0:0 LISTENING
TCP [::]:135 [::]:0 LISTENING
TCP [::]:445 [::]:0 LISTENING
TCP [::]:5021 [::]:0 LISTENING
TCP [::]:15000 [::]:0 LISTENING
TCP [::]:49664 [::]:0 LISTENING
TCP [::]:49665 [::]:0 LISTENING
TCP [::]:49666 [::]:0 LISTENING
TCP [::]:49667 [::]:0 LISTENING
TCP [::]:49670 [::]:0 LISTENING
TCP [::]:49671 [::]:0 LISTENING
TCP [::]:54321 [::]:0 LISTENING
TCP [::]:62996 [::]:0 LISTENING
TCP [::1]:62420 [::]:0 LISTENING
TCP [2001:250:3002:4240:84d6:4830:dc7a:c434]:63239 [2409:8c00:8441:f21::688:111]:443 ESTABLISHED
UDP 0.0.0.0:5353 *:
UDP 0.0.0.0:5353 *:
UDP 0.0.0.0:5353 *:
UDP 0.0.0.0:5353 *:
UDP 0.0.0.0:5355 *:
UDP 0.0.0.0:6881 *:
UDP 0.0.0.0:12345 *:
UDP 0.0.0.0:15000 *:
UDP 0.0.0.0:30264 *:
UDP 0.0.0.0:30274 *:
UDP 0.0.0.0:49819 *:
UDP 0.0.0.0:53022 *:
UDP 0.0.0.0:53699 *:

```

```

UDP 0.0.0.0:58355 *:
UDP 0.0.0.0:61208 *:
UDP 0.0.0.0:62183 *:
UDP 0.0.0.0:62717 *:
UDP 0.0.0.0:62719 *:
UDP 0.0.0.0:64057 *:
UDP 127.0.0.1:1900 *:
UDP 127.0.0.1:61651 *:
UDP 127.0.0.1:61652 *:
UDP 127.0.0.1:61653 *:
UDP 127.0.0.1:61654 *:
UDP 127.0.0.1:61655 *:
UDP 127.0.0.1:61656 *:
UDP 127.0.0.1:63272 *:
UDP 127.0.0.1:65169 *:
UDP 172.19.44.222:137 *:
UDP 172.19.44.222:138 *:
UDP 172.19.44.222:1900 *:
UDP 172.19.44.222:63271 *:
UDP 192.168.56.1:137 *:
UDP 192.168.56.1:138 *:
UDP 192.168.56.1:1900 *:
UDP 192.168.56.1:63270 *:
UDP [::]:5353 *:
UDP [::]:5353 *:
UDP [::]:5353 *:
UDP [::]:5355 *:
UDP [::]:12345 *:
UDP [::1]:1900 *:
UDP [::1]:63269 *:
UDP [fe80::7c4d:bcd5:c9:953%8]:1900 *:
UDP [fe80::7c4d:bcd5:c9:953%8]:63267 *:
UDP [fe80::893e:edc6:780c:97e2%2]:1900 *:
UDP [fe80::893e:edc6:780c:97e2%2]:63268 *:

```


2. netstat -e -s

✧ 该命令，显示了以太网统计信息，例如发送和接收的字节数、数据包数

```
C:\Users\16435>netstat -e -s
接口统计

              接收的              发送的
字节          1597119888          88581064
单播数据包    1266072             690104
非单播数据包  472960             3344
丢弃          0                 0
错误          0                 0
未知协议      0

IPv4 统计信息

接收的数据包          = 153911
接收的标头错误        = 0
接收的地址错误        = 5
转发的数据报          = 0
接收的未知协议        = 0
丢弃的接收数据包      = 1983
传送的接收数据包      = 152936
输出请求              = 96238
路由丢弃              = 0
丢弃的输出数据包      = 10
输出数据包无路由      = 25
需要重新组合          = 257
重新组合成功          = 34
重新组合失败          = 0
数据报分段成功        = 0
数据报分段失败        = 0
分段已创建            = 0
```

IPv6 统计信息

| | |
|----------|---------|
| 接收的数据包 | = 82326 |
| 接收的标头错误 | = 0 |
| 接收的地址错误 | = 0 |
| 转发的数据报 | = 0 |
| 接收的未知协议 | = 0 |
| 丢失的接收数据包 | = 0 |
| 传送给接收数据包 | = 82672 |
| 输出请求 | = 36142 |
| 路由丢弃 | = 0 |
| 丢弃的输出数据包 | = 0 |
| 输出数据包无路由 | = 0 |
| 需要重新组合 | = 0 |
| 重新组合成功 | = 0 |
| 重新组合失败 | = 0 |
| 数据报分段成功 | = 0 |
| 数据报分段失败 | = 0 |
| 分段已创建 | = 0 |

ICMPv4 统计信息

| | 已接收 | | 已发送 | |
|--------|-----|-----|-----|-----|
| 消息 | 155 | 0 | 480 | 0 |
| 错误 | 0 | 0 | 0 | 0 |
| 目标不可达 | 100 | 328 | 0 | 0 |
| 超时 | 45 | 0 | 0 | 0 |
| 参数问题 | 0 | 0 | 0 | 0 |
| 源抑制 | 0 | 0 | 0 | 0 |
| 重定向 | 0 | 0 | 0 | 0 |
| 回复 | 9 | 0 | 0 | 0 |
| 显式回复 | 1 | 0 | 0 | 152 |
| 时间戳 | 0 | 0 | 0 | 0 |
| 时间戳回复 | 0 | 0 | 0 | 0 |
| 地址掩码 | 0 | 0 | 0 | 0 |
| 地址掩码回复 | 0 | 0 | 0 | 0 |
| 路由器请求 | 0 | 0 | 0 | 0 |
| 路由器播发 | 0 | 0 | 0 | 0 |

ICMPv6 统计信息

| | | 已接收 | 已发送 |
|---------|-----|------|-----|
| 消息 | | 1147 | 181 |
| 错误 | | 0 | 0 |
| 目标不可达 | 0 | 0 | 0 |
| 数据包太大 | | 0 | 0 |
| 超时 | 57 | 0 | 0 |
| 参数问题 | 0 | 0 | 0 |
| 回显 | | 0 | 120 |
| 回显回复 | | 63 | 0 |
| MLD 查询 | | 0 | 0 |
| MLD 报告 | | 0 | 0 |
| MLD 已完成 | | 0 | 0 |
| 路由器请求 | 0 | 9 | 0 |
| 路由器播发 | 985 | 0 | 0 |
| 邻居请求 | 8 | 33 | 0 |
| 邻居播发 | 34 | 19 | 0 |
| 重定向 | | 0 | 0 |
| 路由器重新编号 | | 0 | 0 |

IPv4 的 TCP 统计信息

| | |
|---------|----------|
| 主动开放 | = 4363 |
| 被动开放 | = 41 |
| 失败的连接尝试 | = 656 |
| 重置连接 | = 570 |
| 当前连接 | = 28 |
| 接收的分段 | = 144646 |
| 发送的分段 | = 85614 |
| 重新传输的分段 | = 3638 |

IPv6 的 TCP 统计信息

| | |
|---------|---------|
| 主动开放 | = 429 |
| 被动开放 | = 6 |
| 失败的连接尝试 | = 106 |
| 重置连接 | = 51 |
| 当前连接 | = 3 |
| 接收的分段 | = 80885 |
| 发送的分段 | = 35015 |
| 重新传输的分段 | = 156 |

IPv4 的 UDP 统计信息

| | |
|--------|--------|
| 接收的数据报 | = 5458 |
| 无端口 | = 1877 |
| 接收错误 | = 17 |
| 发送的数据报 | = 6793 |

IPv6 的 UDP 统计信息

| | |
|--------|-------|
| 接收的数据报 | = 713 |
| 无端口 | = 0 |
| 接收错误 | = 0 |
| 发送的数据报 | = 504 |

实验五 netstat 命令检测端口是否被开放

3. 命令: `netstat -ano -p tcp | find "3389">nul 2>nul &&echo`

3389 端口已开启 || echo 3389 未开启

```
C:\Users\16435>netstat -ano -p tcp | find "3389" > nul2 >nul && echo 3389端口已开启 || echo 3389端口未开启
3389端口未开启
```

✧ 图中结果显示: 3389 端口未开启

实验六 arp

1. `arp -a`

```
C:\Users\16435>arp -a

接口: 172.19.44.222 --- 0x2
    Internet 地址      物理地址      类型
    172.19.63.254      08-68-8d-a5-1e-01 动态
    172.19.63.255      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
    239.11.20.1        01-00-5e-0b-14-01 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: 192.168.56.1 --- 0x8
    Internet 地址      物理地址      类型
    192.168.56.255     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
    239.11.20.1        01-00-5e-0b-14-01 静态
    239.255.255.250    01-00-5e-7f-ff-fa 静态

C:\Users\16435>
```

✧ 该指令显示了所有接口的 arp 缓存表, 其 IP 地址、物理地址以及类型

2. `arp -a -N 192.168.1.100`

```
PS C:\Windows\system32>arp -a -N 192.168.1.100
ARP: 错误参数: 192.168.1.100
```

✧ 因为本机无该接口, 所以显示错误参数

✧ 修改命令为 arp -a -N 192.168.56.1

```
PS C:\Windows\system32> arp -a -N 192.168.56.1
接口: 192.168.56.1 --- 0x8
Internet 地址      物理地址      类型
192.168.56.255     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
239.11.20.1        01-00-5e-0b-14-01 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态
PS C:\Windows\system32>
```

✧ 显示了该接口的 ARP 缓存表

3. arp -s 192.168.56.1 01-00-5e-00-00-16

```
连接特定的 DNS 后缀 . . . . .
PS C:\Windows\system32> arp -s 192.168.56.1 01-00-5e-00-00-16
PS C:\Windows\system32>
PS C:\Windows\system32> arp -a
接口: 172.26.63.144 --- 0x2
Internet 地址      物理地址      类型
10.0.0.80          00-aa-00-4f-2a-9c 静态
172.26.127.254     00-74-9c-9f-46-87 动态
172.26.127.255     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
239.11.20.1        01-00-5e-0b-14-01 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: 192.168.56.1 --- 0x8
Internet 地址      物理地址      类型
192.168.56.1       01-00-5e-00-00-16 静态 ←
192.168.56.255     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
239.11.20.1        01-00-5e-0b-14-01 静态
239.255.255.250    01-00-5e-7f-ff-fa 静态
PS C:\Windows\system32>
```

✧ 该指令将本机 IP 地址与物理地址 01-00-5e-00-00-16 绑定, 再次

查看接口缓存表, 对比之前的可以发现多了一条记录

实验七 route

1. route print

```

C:\Users\16435>route print
=====
接口列表
14...d0 5f 64 35 f9 5d .....Realtek PCIe GbE Family Controller
 8...0a 00 27 00 00 08 .....VirtualBox Host-Only Ethernet Adapter
18...84 fd d1 ea 9e 59 .....Microsoft Wi-Fi Direct Virtual Adapter
16...86 fd d1 ea 9e 58 .....Microsoft Wi-Fi Direct Virtual Adapter #2
 2...84 fd d1 ea 9e 58 .....Intel(R) Wi-Fi 6 AX201 160MHz
11...84 fd d1 ea 9e 5c .....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          172.19.63.254  172.19.44.222  45
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
127.255.255.255    255.255.255.255  在链路上      127.0.0.1      331
172.19.0.0          255.255.192.0    在链路上      172.19.44.222  301
172.19.44.222      255.255.255.255  在链路上      172.19.44.222  301
172.19.63.255      255.255.255.255  在链路上      172.19.44.222  301
192.168.56.0        255.255.255.0    在链路上      192.168.56.1   281
192.168.56.1        255.255.255.255  在链路上      192.168.56.1   281
192.168.56.255      255.255.255.255  在链路上      192.168.56.1   281
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      192.168.56.1   281
224.0.0.0           240.0.0.0        在链路上      172.19.44.222  301
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      192.168.56.1   281
255.255.255.255     255.255.255.255  在链路上      172.19.44.222  301
=====
永久路由:
无

IPv6 路由表
=====
活动路由:
接口跃点数网络目标          网关
1    331 ::1/128          在链路上
8    281 fe80::/64        在链路上
2    301 fe80::/64        在链路上
8    281 fe80::7c4d:bcd5:c9:953/128 在链路上
2    301 fe80::893e:edc6:780c:97e2/128 在链路上
1    331 ff00::/8          在链路上
8    281 ff00::/8          在链路上
2    301 ff00::/8          在链路上
=====
永久路由:
无
C:\Users\16435>_

```

✧ 通过该指令，显示了 IP 路由表的完整内容。

2. route print 10.*

✧ 通过该指令，显示了 IP 路由表中以 10. 开头的路由

```
C:\Users\16435>route print 10.*
=====
接口列表
14...d0 5f 64 35 f9 5d .....Realtek PCIe GbE Family Controller
8...0a 00 27 00 00 08 .....VirtualBox Host-Only Ethernet Adapter
18...84 fd d1 ea 9e 59 .....Microsoft Wi-Fi Direct Virtual Adapter
16...86 fd d1 ea 9e 58 .....Microsoft Wi-Fi Direct Virtual Adapter #2
2...84 fd d1 ea 9e 58 .....Intel(R) Wi-Fi 6 AX201 160MHz
11...84 fd d1 ea 9e 5c .....Bluetooth Device (Personal Area Network)
1.....Software Loopback Interface 1
=====

IPv4 路由表
=====
活动路由:
无
永久路由:
无

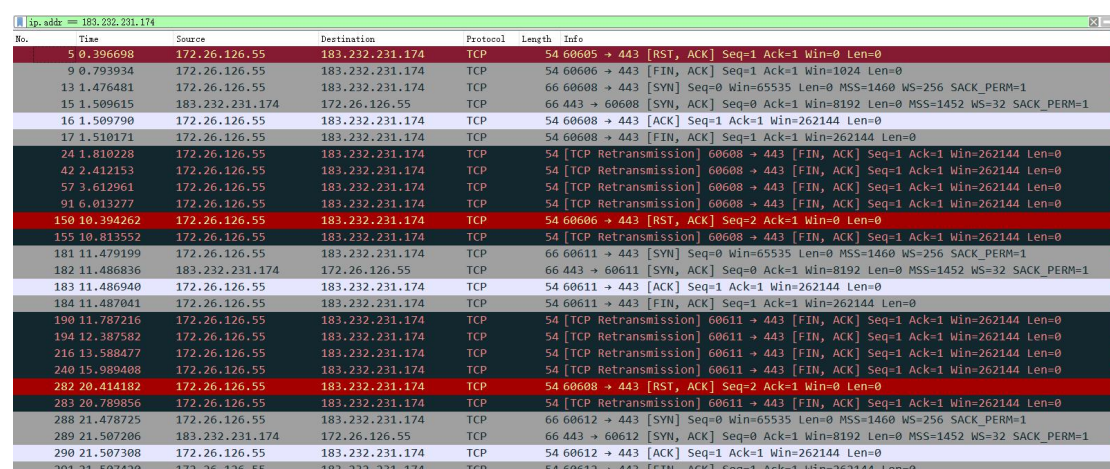
IPv6 路由表
=====
活动路由:
无
永久路由:
无

C:\Users\16435>
```

实验八 使用 wireshark 捕获数据包，设置 2 至多种不同过滤条件

1. ip.addr == 183.232.231.174

✧ 第一个过滤条件，仅显示与指定 IP 地址（183.232.231.174）通信的记录

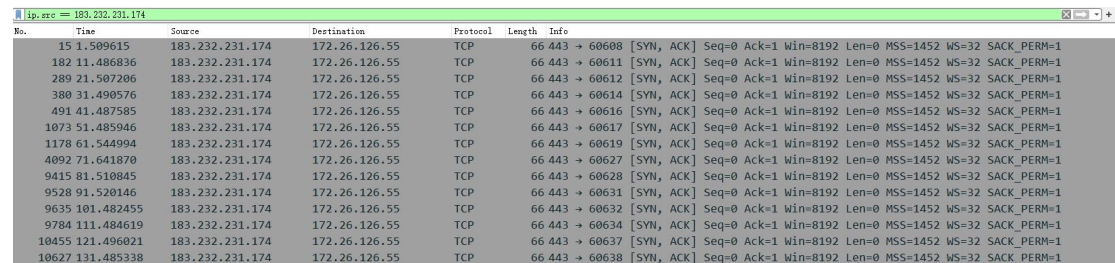


| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|-----------------|-----------------|----------|--------|--|
| 5 | 0.396698 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | 60605 → 443 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0 |
| 9 | 0.793934 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | 60606 → 443 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0 |
| 13 | 1.476481 | 172.26.126.55 | 183.232.231.174 | TCP | 66 | 60608 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 15 | 1.509615 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60608 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 16 | 1.509790 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | 60608 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 17 | 1.510171 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | 60608 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 24 | 1.810228 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | [TCP Retransmission] 60608 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 42 | 2.412153 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | [TCP Retransmission] 60608 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 57 | 3.612961 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | [TCP Retransmission] 60608 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 91 | 6.013277 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | [TCP Retransmission] 60608 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 150 | 10.394262 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | 60606 → 443 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0 |
| 155 | 10.813552 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | [TCP Retransmission] 60608 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 181 | 11.479199 | 172.26.126.55 | 183.232.231.174 | TCP | 66 | 60611 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 182 | 11.486836 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60611 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 183 | 11.486940 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | 60611 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 184 | 11.487041 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | 60611 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 190 | 11.787216 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | [TCP Retransmission] 60611 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 194 | 12.387582 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | [TCP Retransmission] 60611 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 216 | 13.588477 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | [TCP Retransmission] 60611 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 240 | 15.989408 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | [TCP Retransmission] 60611 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 282 | 20.414182 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | 60608 → 443 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0 |
| 283 | 20.789956 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | [TCP Retransmission] 60611 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 288 | 21.478725 | 172.26.126.55 | 183.232.231.174 | TCP | 66 | 60612 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 289 | 21.507206 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60612 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 290 | 21.507308 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | 60612 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0 |
| 291 | 21.507420 | 172.26.126.55 | 183.232.231.174 | TCP | 54 | 60612 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0 |

✧ 如图，显示的所有记录来源或目的地为 183.232.231.174

2. ip.src == 183.232.231.174

✧ 如图，将过滤条件设置为 ip 地址的来源为 183.232.231.174



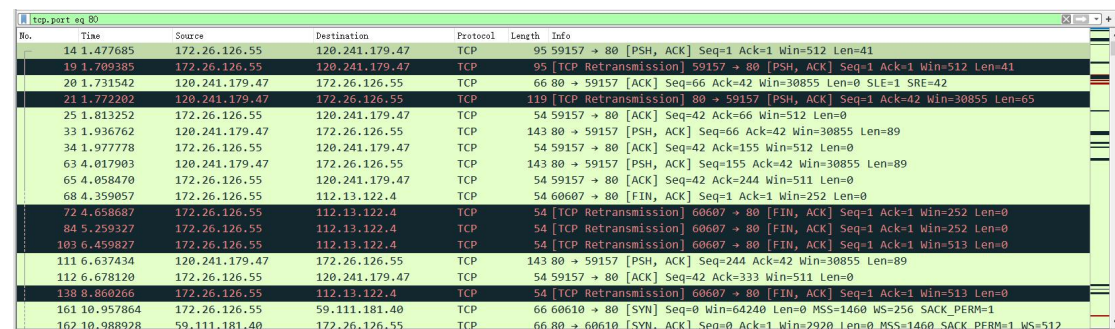
The screenshot shows a Wireshark interface with the filter 'ip.src == 183.232.231.174' applied. The packet list displays 18 packets, all originating from 183.232.231.174 and destined for 172.26.126.55. All packets are TCP with a length of 66 bytes. The packet details pane shows the first packet (No. 15) as a SYN-ACK with sequence 60608, acknowledgment 1, and window 8192.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-------|------------|-----------------|---------------|----------|--------|--|
| 15 | 1.509615 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60608 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 182 | 11.486836 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60611 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 289 | 21.507206 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60612 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 380 | 31.490576 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60614 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 491 | 41.487585 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60616 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 1073 | 51.485946 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60617 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 1178 | 61.544994 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60619 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 4092 | 71.641870 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60627 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 9415 | 81.510845 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60628 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 9528 | 91.520146 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60631 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 9635 | 101.482455 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60632 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 9784 | 111.484619 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60634 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 10455 | 121.496021 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60637 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |
| 10627 | 131.485338 | 183.232.231.174 | 172.26.126.55 | TCP | 66 | 443 → 60638 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1 |

✧ 图中显示的均是来自 183.232.231.174 的记录

3. tcp.port eq 80

✧ 该过滤条件是按端口过滤的，不管端口是源还是目标的都只显示满足 tcp.port == 80 条件的包。

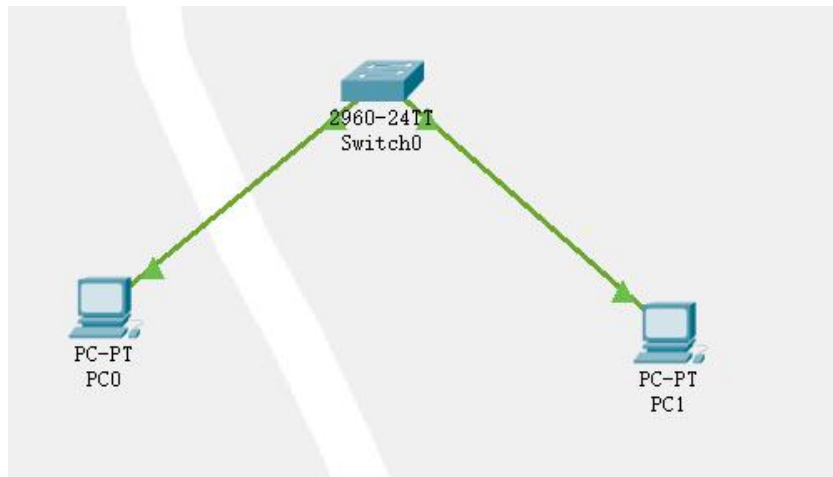


The screenshot shows a Wireshark interface with the filter 'tcp.port eq 80' applied. The packet list displays 20 packets, all related to port 80. The packet details pane shows the first packet (No. 14) as a PSH-ACK with sequence 59157, acknowledgment 80, and window 512.

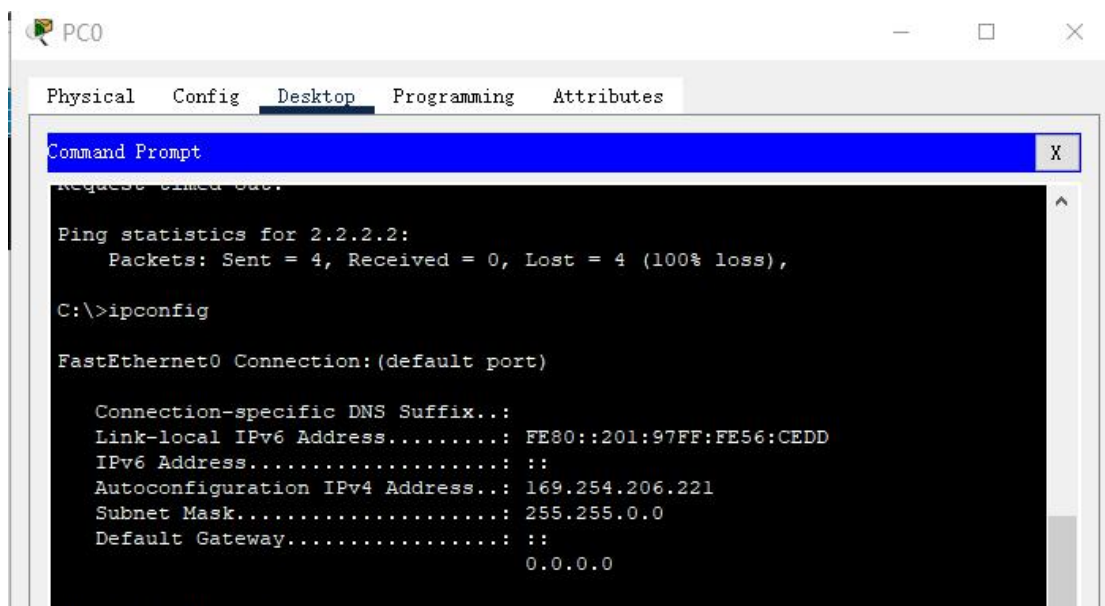
| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-----------|----------------|----------------|----------|--------|--|
| 14 | 1.477685 | 172.26.126.55 | 120.241.179.47 | TCP | 95 | 59157 → 80 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=41 |
| 19 | 1.709385 | 172.26.126.55 | 120.241.179.47 | TCP | 95 | [TCP Retransmission] 59157 → 80 [PSH, ACK] Seq=1 Ack=1 Win=512 Len=41 |
| 20 | 1.731542 | 120.241.179.47 | 172.26.126.55 | TCP | 66 | 80 → 59157 [ACK] Seq=66 Ack=42 Win=30855 Len=0 SLE=1 SRE=42 |
| 21 | 1.772202 | 120.241.179.47 | 172.26.126.55 | TCP | 119 | [TCP Retransmission] 80 → 59157 [PSH, ACK] Seq=1 Ack=42 Win=30855 Len=65 |
| 25 | 1.813252 | 172.26.126.55 | 120.241.179.47 | TCP | 54 | 59157 → 80 [ACK] Seq=42 Ack=66 Win=512 Len=0 |
| 33 | 1.936762 | 120.241.179.47 | 172.26.126.55 | TCP | 143 | 80 → 59157 [PSH, ACK] Seq=66 Ack=42 Win=30855 Len=89 |
| 34 | 1.977778 | 172.26.126.55 | 120.241.179.47 | TCP | 54 | 59157 → 80 [ACK] Seq=42 Ack=155 Win=512 Len=0 |
| 63 | 4.017903 | 120.241.179.47 | 172.26.126.55 | TCP | 143 | 80 → 59157 [PSH, ACK] Seq=155 Ack=42 Win=30855 Len=89 |
| 65 | 4.058470 | 172.26.126.55 | 120.241.179.47 | TCP | 54 | 59157 → 80 [ACK] Seq=42 Ack=244 Win=511 Len=0 |
| 68 | 4.359057 | 172.26.126.55 | 112.13.122.4 | TCP | 54 | 60607 → 80 [FIN, ACK] Seq=1 Ack=1 Win=252 Len=0 |
| 72 | 4.658687 | 172.26.126.55 | 112.13.122.4 | TCP | 54 | [TCP Retransmission] 60607 → 80 [FIN, ACK] Seq=1 Ack=1 Win=252 Len=0 |
| 84 | 5.259327 | 172.26.126.55 | 112.13.122.4 | TCP | 54 | [TCP Retransmission] 60607 → 80 [FIN, ACK] Seq=1 Ack=1 Win=252 Len=0 |
| 103 | 6.459827 | 172.26.126.55 | 112.13.122.4 | TCP | 54 | [TCP Retransmission] 60607 → 80 [FIN, ACK] Seq=1 Ack=1 Win=513 Len=0 |
| 111 | 6.637434 | 120.241.179.47 | 172.26.126.55 | TCP | 143 | 80 → 59157 [PSH, ACK] Seq=244 Ack=42 Win=30855 Len=89 |
| 112 | 6.678120 | 172.26.126.55 | 120.241.179.47 | TCP | 54 | 59157 → 80 [ACK] Seq=42 Ack=333 Win=511 Len=0 |
| 138 | 8.860266 | 172.26.126.55 | 112.13.122.4 | TCP | 54 | [TCP Retransmission] 60607 → 80 [FIN, ACK] Seq=1 Ack=1 Win=513 Len=0 |
| 161 | 10.957864 | 172.26.126.55 | 59.111.181.40 | TCP | 66 | 60610 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 162 | 10.988928 | 59.111.181.40 | 172.26.126.55 | TCP | 66 | 80 → 60610 [SYN, ACK] Seq=0 Ack=1 Win=2920 Len=0 MSS=1460 SACK_PERM=1 WS=512 |

实验九 使用 packet tracer 实时、仿真两个操作模式呈现网络的行为

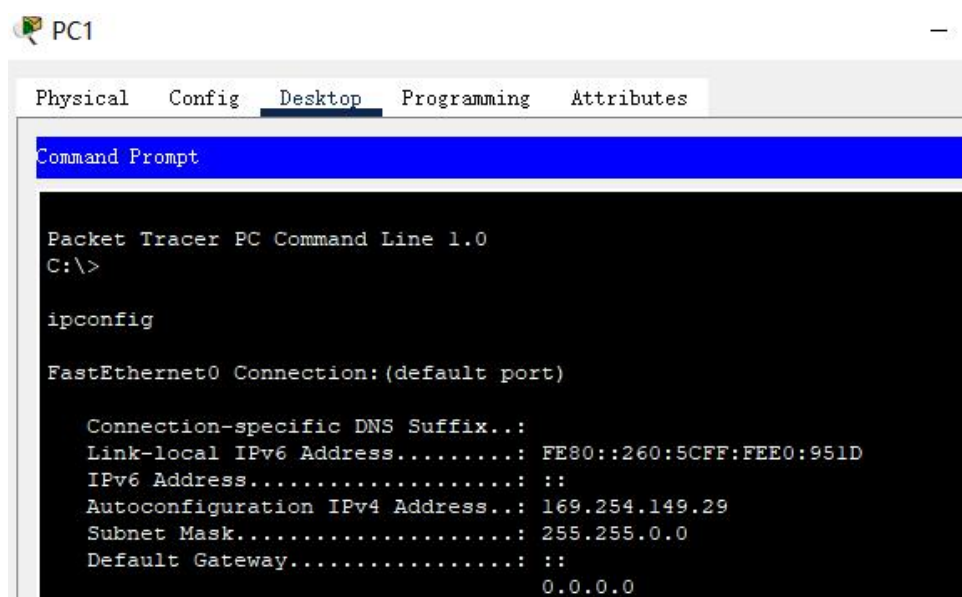
1. 实时模式



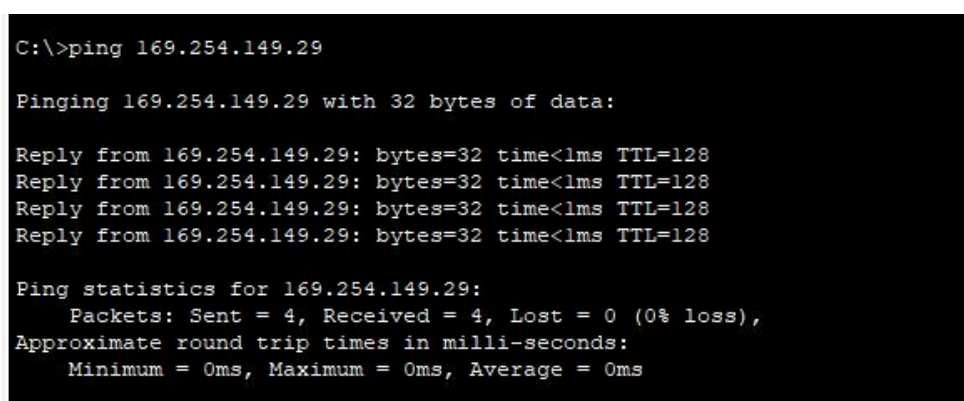
✧ 通过 ipconfig 命令，得知 PC0 的 IP 地址为 169.254.206.221



✧ 同样，得知 PC1 的 IP 地址为 169.254.149.29

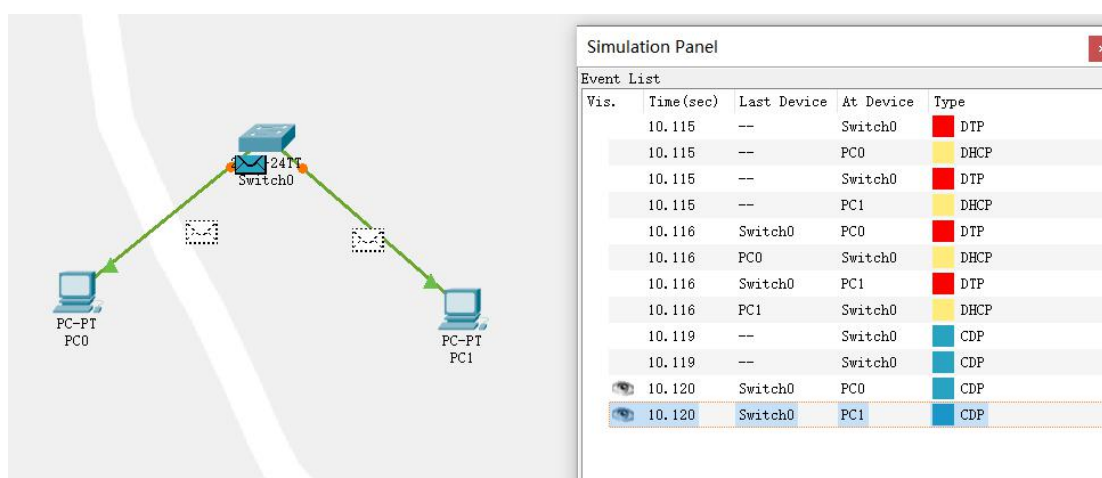


✧ 在 PC0 上操作，ping 169.254.149.29，连接成功



✧ 在实时模式中，网络行为和真实设备一样，对所有网络行为及时响应。

2. 仿真模式



✧ 在仿真模式中，用户可以看到和控制时间间隔、数据传输的内部流程。

实验十

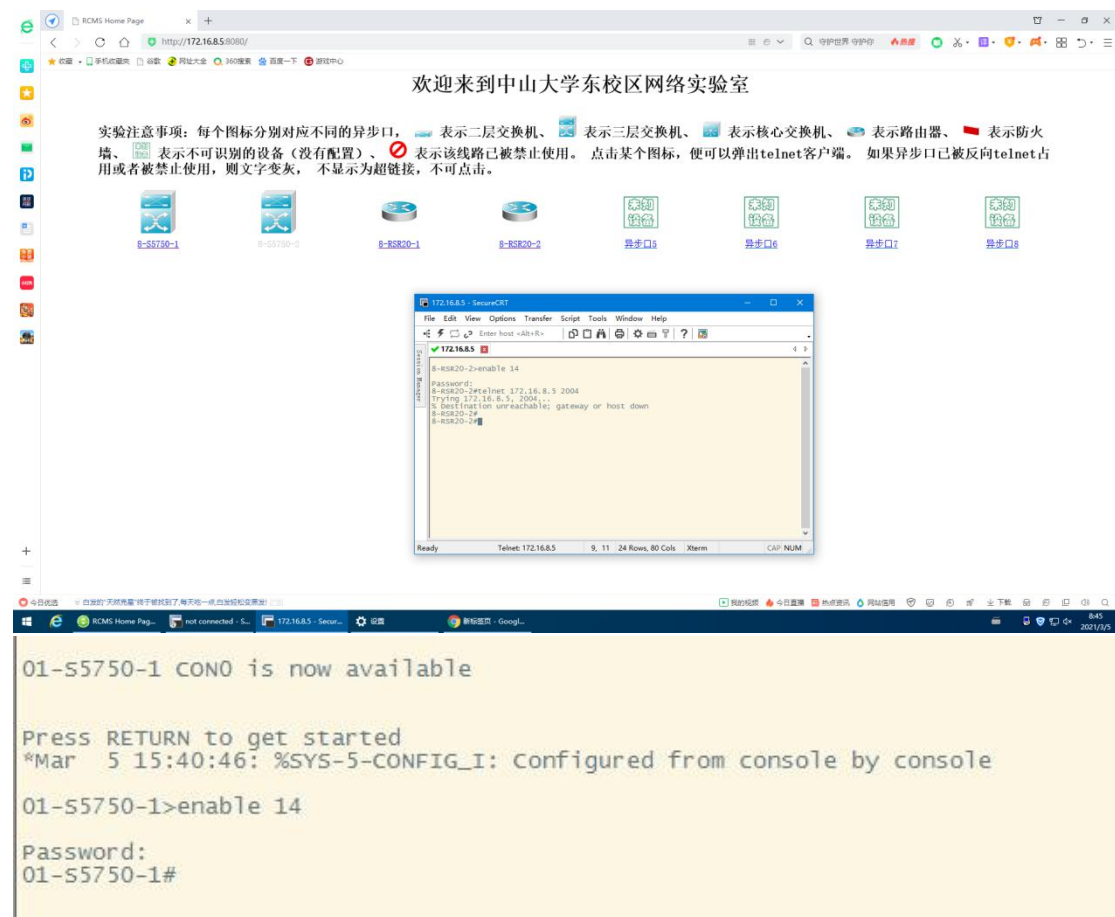
1. 通过 web 访问 RCMS

✧ 打开浏览器，输入 `http://172.16.8.5:8080` 即可进入

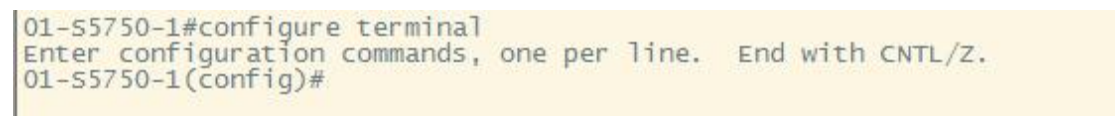
✧ 选择实验设备

✧ Enable 14 进入特权模式


✧ 输入密码 b402



✧ 进入全局模式



2. 通过 telnet 访问网络设备



```
05/03/2021 09:34.29 /home/mobaxterm telnet 172.16.8.5
Trying 172.16.8.5...
Connected to 172.16.8.5.
Escape character is '^['.

TEXT
User Access Verification

Password:

RCMS-8>
```

✧ 按正确形式输入命令，可见显示已连接

二、总结

通过这次实验，我学习到了若干网络命令，对他们的功能、用途、格式，有了一定得了解；其次还学习应用了两个软件 wire shark 抓包工具以及 packet tracer，体会到了在实时和仿真两种模式下不同的网络行为呈现，为后续实验奠定了一定的基础。