

四川大学计算机学院、软件学院

实验报告

学号：2022141460176 姓名：杨一舟 专业：计算机科学与技术 第 9 周

|                   |   |      |                  |
|-------------------|---|------|------------------|
| 课程名称              | 计算机网络课程设计   | 实验课时 | 2 课时             |
| 实验项目              | ICMP 协议分析   | 实验时间 | 2024 年 10 月 31 日 |
| 实验目的              | 使用 wire shark 捕获 ping 和 trace route 命令会话的数据包，分析 ICMP 协议特点与原理，了解 ping 和 trace route 的设计原理  |      |                  |
| 实验环境              | Windows 11、wire shark   |      |                  |
| 实验内容（算法、程序、步骤和方法） | <p>一、Ping 数据包捕获及原理分析</p> <p>第一步:ping 数据包的捕获;</p> <p>1.1 启动 Wireshark 并配置捕获:</p> <p>打开 Wireshark 软件。选择 WLAN 网络接口，点击“捕获”按钮或使用快捷键开始捕获网络流量。</p> <p>1.2 执行 ping 命令:</p> <p>打开计算机上的命令提示符或终端窗口。</p> <p>输入命令 <code>ping -n 5 www.scu.edu.cn</code> 并按回车键执行。这条命令将会向 <code>www.scu.edu.cn</code> 发送 5 个 ICMP 回声请求。</p> <div><pre>C:\Users\MountainMist&gt;ping -n 5 www.scu.edu.cn  正在 Ping www.scu.edu.cn [202.115.32.43] 具有 32 字节的数据: 来自 202.115.32.43 的回复: 字节=32 时间=1ms TTL=61 来自 202.115.32.43 的回复: 字节=32 时间=1ms TTL=61 来自 202.115.32.43 的回复: 字节=32 时间=1ms TTL=61 来自 202.115.32.43 的回复: 字节=32 时间=2ms TTL=61 来自 202.115.32.43 的回复: 字节=32 时间=3ms TTL=61</pre></div> <p>其中的 -n 参数用来指定要发送的回声请求数量，在这里设置为 5 次。可以在命令行中输入 <code>ping /?</code> 来查看帮助文档了解关于 ping 命令的更多选项和参数。</p> |      |                  |

```
C:\Users\MountainMist>ping /?
```

用法: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS] [-r count] [-s count] [[-j host-list] | [-k host-list]] [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p] [-4] [-6] target\_name

选项:

- t Ping 指定的主机, 直到停止。  
若要查看统计信息并继续操作, 请键入 Ctrl+Break;  
若要停止, 请键入 Ctrl+C。
- a 将地址解析为主机名。
- n count 要发送的回显请求数。
- l size 发送缓冲区大小。
- f 在数据包中设置“不分段”标记(仅适用于 IPv4)。
- i TTL 生存时间。
- v TOS 服务类型(仅适用于 IPv4。该设置已被弃用, 对 IP 标头中的服务类型字段没有任何影响)。
- r count 记录计数跃点的路由(仅适用于 IPv4)。
- s count 计数跃点的时间戳(仅适用于 IPv4)。
- j host-list 与主机列表一起使用的松散源路由(仅适用于 IPv4)。
- k host-list 与主机列表一起使用的严格源路由(仅适用于 IPv4)。
- w timeout 等待每次回复的超时时间(毫秒)。
- R 同样使用路由标头测试反向路由(仅适用于 IPv6)。  
根据 RFC 5095, 已弃用此路由标头。  
如果使用此标头, 某些系统可能丢弃回显请求。
- S srcaddr 要使用的源地址。
- c compartment 路由隔离舱标识符。
- p Ping Hyper-V 网络虚拟化提供程序地址。
- 4 强制使用 IPv4。
- 6 强制使用 IPv6。

### 1.3 停止捕获

#### 1.4 过滤捕获的数据包:

在 Wireshark 的过滤栏中输入 icmp 并应用该过滤条件。这将使得界面仅显示那些与 ICMP 协议有关的数据包。

根据预期的结果, 可以看到总共出现了 10 个 ICMP 数据包: 这是由于每发送一次请求就会收到一次应答, 因此 5 次请求对应了 5 次应答, 总计 10 个数据包。

|     |           |                |                |      |                        |  |
|-----|-----------|----------------|----------------|------|------------------------|--|
| 153 | 20.432092 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request | id=0x0001, seq=1/256, ttl=128 (reply in 154)   |
| 154 | 20.433853 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply   | id=0x0001, seq=1/256, ttl=61 (request in 153)  |
| 157 | 21.448968 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request | id=0x0001, seq=2/512, ttl=128 (reply in 158)   |
| 158 | 21.450901 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply   | id=0x0001, seq=2/512, ttl=61 (request in 157)  |
| 164 | 22.464482 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request | id=0x0001, seq=3/768, ttl=128 (reply in 165)   |
| 165 | 22.466248 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply   | id=0x0001, seq=3/768, ttl=61 (request in 164)  |
| 166 | 23.474515 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request | id=0x0001, seq=4/1024, ttl=128 (reply in 167)  |
| 167 | 23.476530 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply   | id=0x0001, seq=4/1024, ttl=61 (request in 166) |
| 170 | 24.477457 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request | id=0x0001, seq=5/1280, ttl=128 (reply in 171)  |
| 171 | 24.480503 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply   | id=0x0001, seq=5/1280, ttl=61 (request in 170) |

在 Wireshark 的“信息(Info)”列中, 可以清楚地区分出哪些数据包是请求(Request), 哪些是应答(Reply)。

## 第二步:ICMP 报文内容分析

1) Ping 命令利用了 ICMP 的哪种类型报文,从哪里可以看出来?

"Ping"命令使用了 Internet Control Message Protocol (ICMP) 的回显请求 (Echo Request) 与回显应答 (Echo Reply) 类型的报文。从执行 Ping 命令后返回的信息中可以看出。

|     |           |                |                |      |                        |  |
|-----|-----------|----------------|----------------|------|------------------------|--|
| 153 | 20.432092 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request | id=0x0001, seq=1/256, ttl=128 (reply in 154)   |
| 154 | 20.433853 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply   | id=0x0001, seq=1/256, ttl=61 (request in 153)  |
| 157 | 21.448968 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request | id=0x0001, seq=2/512, ttl=128 (reply in 158)   |
| 158 | 21.450901 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply   | id=0x0001, seq=2/512, ttl=61 (request in 157)  |
| 164 | 22.464482 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request | id=0x0001, seq=3/768, ttl=128 (reply in 165)   |
| 165 | 22.466248 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply   | id=0x0001, seq=3/768, ttl=61 (request in 164)  |
| 166 | 23.474515 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request | id=0x0001, seq=4/1024, ttl=128 (reply in 167)  |
| 167 | 23.476530 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply   | id=0x0001, seq=4/1024, ttl=61 (request in 166) |
| 170 | 24.477457 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request | id=0x0001, seq=5/1280, ttl=128 (reply in 171)  |
| 171 | 24.480503 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply   | id=0x0001, seq=5/1280, ttl=61 (request in 170) |

2) Ping 包发送的 ICMP 报文的数据部分内容是什么?

该数据可以以多种形式表示,若以字符串形式表示,

则数据内容就是 abcdefghijklmn opqrstuvwxyz abcdefg hi

[Request frame: 153]

[Response time: 1.761 ms]

▼ Data (32 bytes)

Data: 6162636465666768696a6b6c6d6e6f7071727374757677616263646566676869

[Length: 32]

|      |   |                    |
|------|---|--------------------|
| 0000 | f4 26 79 a9 5d 85 58 69 6c 4c 47 53 08 00 45 00 | &y.]·Xi lLGS·E·    |
| 0010 | 80 3c 4a 72 00 00 3d 01 bc 5a ca 73 20 2b 0a 87 | ·c·Tr·a··Z·s·+·    |
| 0020 | 01 cf 00 00 55 5a 00 01 00 01 61 62 63 64 65 66 | ·····UZ·····abcdef |
| 0030 | 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 | ghijklmn opqrstuv  |
| 0040 | 77 61 62 63 64 65 66 67 68 69                   | wabcddefg hi       |

3) 第一个 Ping 报返回的准确时间是多少?

是 1.761ms

|     |           |                |                |      |   |                          |
|-----|-----------|----------------|----------------|------|---|--------------------------|
| 153 | 20.432092 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request                        | id=0x0001, seq=1/256, tt |
| 154 | 20.433853 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply                          | id=0x0001, seq=1/256, tt |
| 157 | 21.448968 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request                        | id=0x0001, seq=2/512, tt |
| 158 | 21.450901 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply                          | id=0x0001, seq=2/512, tt |
| 164 | 22.464482 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request                        | id=0x0001, seq=3/768, tt |
| 165 | 22.466248 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply                          | id=0x0001, seq=3/768, tt |
| 166 | 23.474515 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request                        | id=0x0001, seq=4/1024, t |
| 167 | 23.476530 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply                          | id=0x0001, seq=4/1024, t |
| 170 | 24.477457 | 10.135.129.207 | 202.115.32.43  | ICMP | 74 Echo (ping) request                        | id=0x0001, seq=5/1280, t |
| 171 | 24.480503 | 202.115.32.43  | 10.135.129.207 | ICMP | 74 Echo (ping) reply                          | id=0x0001, seq=5/1280, t |
| 172 | 26.088013 | 10.135.135.254 | 10.135.129.207 | ICMP | 94 Destination unreachable (Host unreachable) |                          |
| 173 | 26.088013 | 10.135.135.254 | 10.135.129.207 | ICMP | 94 Destination unreachable (Host unreachable) |                          |
| 174 | 26.088179 | 10.135.135.254 | 10.135.129.207 | ICMP | 94 Destination unreachable (Host unreachable) |                          |
| 175 | 26.088179 | 10.135.135.254 | 10.135.129.207 | ICMP | 94 Destination unreachable (Host unreachable) |                          |
| 176 | 26.088179 | 10.135.135.254 | 10.135.129.207 | ICMP | 94 Destination unreachable (Host unreachable) |                          |

> Frame 154: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF\_{EDF25D4C-1883-4468-AE33-B3}

> Ethernet II, Src: RuijieNetwork\_4c:47:53 (58:69:6c:4c:47:53), Dst: Intel\_09:5d:85 (f4:26:79:a9:5d:85)

> Internet Protocol Version 4, Src: 202.115.32.43, Dst: 10.135.129.207

> Internet Control Message Protocol

Type: 0 (Echo (ping) reply)

Code: 0

Checksum: 0x555a [correct]

[Checksum Status: Good]

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence Number (BE): 1 (0x0001)

Sequence Number (LE): 256 (0x0100)

[Request frame: 153]

[Response time: 1.761 ms]

▼ Data (32 bytes)

Data: 6162636465666768696a6b6c6d6e6f7071727374757677616263646566676869

[Length: 32]

4) IP 数据报头部已经有 checksum 字段,为什么 ICMP 还有 checksum 字段?

IP 头部的校验和仅验证 IP 头部的完整性,而不检查 IP 数据报中的数据部分。ICMP 作为 IP 数据报的一部分,为了确保自身报文的完整性和正确性,需要独立的校验和字段。这样可以独立验证 ICMP 报文,覆盖整个 ICMP 消息的错误检测,并增加安全性。

Internet Protocol Version 4, Src: 202.115.32.43, Dst: 10.135.129.207

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 60

Identification: 0x4a72 (19058)

> 000. .... = Flags: 0x0

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 61

Protocol: ICMP (1)

Header Checksum: 0xbc5a [validation disabled]

[Header checksum status: Unverified]

Source Address: 202.115.32.43

Destination Address: 10.135.129.207

[Stream index: 17]

Internet Control Message Protocol

Type: 0 (Echo (ping) reply)

Code: 0

Checksum: 0x555a [correct]

[Checksum Status: Good]

Identifier (BE): 1 (0x0001)

Identifier (LE): 256 (0x0100)

Sequence Number (BE): 1 (0x0001)

Sequence Number (LE): 256 (0x0100)

[Request frame: 153]

[Response time: 1.761 ms]

(接上) 实验内容 (算法、程序、步骤和方法)

## 二、Trace route 数据包捕获及原理分析

第一步:Trace route 数据包的捕获;

2.1 打开 Wireshark, 启动 Wireshark 分组俘获器;

2.2 在命令行中输入 "tracert/d www.scu.edu.cn", 回车

输入命令 tracert /d www.scu.edu.cn 后按回车键。这里的 /d 参数用于禁止名称解析, 加快追踪速度, 因为直接使用 IP 地址代替主机名可以减少 DNS 查询的时间。

通过最多 30 个跃点跟踪  
到 www.scu.edu.cn [202.115.32.43] 的路由:

跟踪完成。

此命令将追踪到目标网站（本例中为 `scu.edu.cn`）的路径，并显示沿途每个路由器的响应时间。

## 第二步:Trace route 报文内容分析

Traceroute 使用的是 ICMP 回声请求数据报，通过设置不同的 TTL 值来探测路径上的路由器。每个路由器在 TTL 为 0 时返回 ICMP 超时消息，Traceroute 记录这些路由器的 IP 地址，直到达到目标主机。

2) Traceroute 发送的回显请求数据包和 ping 发送的数据包数据部分有什么差异?

Ping 的数据部分是 32bytes, 且含有有效信息

- Internet Control Message Protocol

```
Code: 0
Checksum: 0x4d59 [correct]
[Checksum Status: Good]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence Number (BE): 2 (0x0002)
Sequence Number (LE): 512 (0x0200)
```

▼ Data (32 bytes)  
Data: 6162636465666768696a6b6c6d6e6f70717273747576776162636465  
[Length: 32]

- Internet Control Message Protocol

```
Code: 0
Checksum: 0xf7f4 [correct]
[Checksum Status: Good]
Identifier (BE): 1 (0x0001)
Identifier (LE): 256 (0x0100)
Sequence Number (BE): 10 (0x000a)
Sequence Number (LE): 2560 (0x0a00)
```

[illegible]



3)发送的报文,出现了什么错误,错误原因是什么?

出现了“Time-to-live exceeded”，即 ICMP 超时未收到回复的错误，其 type 为 11，code 为 0 可能有以下几个原因：

目标主机不可达：目标主机可能已经关机或网络连接中断。

防火墙或安全设备阻止：中间的防火墙或安全设备可能拦截了 ICMP 请求或响应。

路由问题：路径上的某个路由器可能配置错误或故障，导致数据包无法到达目标主机。

目标主机配置：目标主机可能配置为不响应 ICMP 请求，例如禁用了 ICMP 回显功能。

网络拥塞：网络拥塞可能导致数据包丢失或延迟过大，超过了超时时间。

|       |           |                |                |      |                           |  |
|-------|-----------|----------------|----------------|------|---------------------------|--|
| 6991  | 31.074217 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=6/1536, ttl=1 (no response found!)  |
| 10333 | 34.991582 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=7/1792, ttl=1 (no response found!)  |
| 10334 | 34.993245 | 10.135.135.254 | 10.135.129.207 | ICMP | 134 Time-to-live exceeded | Time to live exceeded in transit                   |
| 10335 | 34.993447 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=8/2048, ttl=1 (no response found!)  |
| 13101 | 38.996353 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=9/2304, ttl=2 (no response found!)  |
| 13102 | 38.998313 | 202.115.39.33  | 10.135.129.207 | ICMP | 134 Time-to-live exceeded | Time to live exceeded in transit                   |
| 13103 | 38.998691 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=10/2560, ttl=2 (no response found!) |
| 13106 | 39.000554 | 202.115.39.33  | 10.135.129.207 | ICMP | 134 Time-to-live exceeded | Time to live exceeded in transit                   |
| 13107 | 39.001068 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=11/2816, ttl=2 (no response found!) |
| 13108 | 39.002790 | 202.115.39.33  | 10.135.129.207 | ICMP | 134 Time-to-live exceeded | Time to live exceeded in transit                   |
| 13574 | 40.024452 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=12/3072, ttl=3 (no response found!) |
| 13575 | 40.037011 | 202.115.39.102 | 10.135.129.207 | ICMP | 70 Time-to-live exceeded  | Time to live exceeded in transit                   |
| 13576 | 40.037524 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=13/3328, ttl=3 (no response found!) |
| 13577 | 40.041565 | 202.115.39.102 | 10.135.129.207 | ICMP | 70 Time-to-live exceeded  | Time to live exceeded in transit                   |
| 13578 | 40.041906 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=14/3584, ttl=3 (no response found!) |
| 13579 | 40.045167 | 202.115.39.102 | 10.135.129.207 | ICMP | 70 Time-to-live exceeded  | Time to live exceeded in transit                   |
| 14743 | 41.046982 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=15/3840, ttl=4 (reply in 14748)     |
| 14748 | 41.048626 | 202.115.32.43  | 10.135.129.207 | ICMP | 106 Echo (ping) reply     | id=0x0001, seq=15/3840, ttl=61 (request in 14743)  |

Internet Control Message Protocol  
Type: 11 (Time-to-live exceeded)  
Code: 0 (Time to live exceeded in transit)  
Checksum: 0xf4ff [correct]  
[Checksum Status: Good]  
Unused: 00000000

4)第一个 TTL 超时报文时由谁发出的?

第一个 TTL 超时报文是由路径上的第一个路由器发出的。当 Traceroute 发送的第一个 ICMP 回声请求数据包 (TTL 设置为 1) 到达第一个路由器时，路由器会将 TTL 减 1，此时 TTL 变为 0。路由器会丢弃该数据包，并向发送方返回一个 ICMP 超时 (Time Exceeded) 消息。

5)在 Traceroute 的过程中,发送方一共发送了多少个不同 TTL 的报文? (相同的 TTL 算一个)

四个, 因为 TTL 从 1 增加至 4

|       |           |                |                |      |                           |  |
|-------|-----------|----------------|----------------|------|---------------------------|--|
| 6991  | 31.074217 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=6/1536, ttl=1 (no response found!)  |
| 10333 | 34.991582 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=7/1792, ttl=1 (no response found!)  |
| 10334 | 34.993245 | 10.135.135.254 | 10.135.129.207 | ICMP | 134 Time-to-live exceeded | Time to live exceeded in transit                   |
| 10335 | 34.993447 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=8/2048, ttl=1 (no response found!)  |
| 13101 | 38.996353 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=9/2304, ttl=2 (no response found!)  |
| 13102 | 38.998313 | 202.115.39.33  | 10.135.129.207 | ICMP | 134 Time-to-live exceeded | Time to live exceeded in transit                   |
| 13103 | 38.998691 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=10/2560, ttl=2 (no response found!) |
| 13106 | 39.000554 | 202.115.39.33  | 10.135.129.207 | ICMP | 134 Time-to-live exceeded | Time to live exceeded in transit                   |
| 13107 | 39.001068 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=11/2816, ttl=2 (no response found!) |
| 13108 | 39.002790 | 202.115.39.33  | 10.135.129.207 | ICMP | 134 Time-to-live exceeded | Time to live exceeded in transit                   |
| 13574 | 40.024452 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=12/3072, ttl=3 (no response found!) |
| 13575 | 40.037011 | 202.115.39.102 | 10.135.129.207 | ICMP | 70 Time-to-live exceeded  | Time to live exceeded in transit                   |
| 13576 | 40.037524 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=13/3328, ttl=3 (no response found!) |
| 13577 | 40.041565 | 202.115.39.102 | 10.135.129.207 | ICMP | 70 Time-to-live exceeded  | Time to live exceeded in transit                   |
| 13578 | 40.041906 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=14/3584, ttl=3 (no response found!) |
| 13579 | 40.045167 | 202.115.39.102 | 10.135.129.207 | ICMP | 70 Time-to-live exceeded  | Time to live exceeded in transit                   |
| 14743 | 41.046982 | 10.135.129.207 | 202.115.32.43  | ICMP | 106 Echo (ping) request   | id=0x0001, seq=15/3840, ttl=4 (reply in 14748)     |
| 14748 | 41.048626 | 202.115.32.43  | 10.135.129.207 | ICMP | 106 Echo (ping) reply     | id=0x0001, seq=15/3840, ttl=61 (request in 14743)  |

6)这四种不同 TTL 的数据包 TTL 字段的特点是什么?

每个数据包 的 TTL 值从 1 开始逐渐增加, 每次增加 1。通过逐步增加 TTL 值, Traceroute 可以逐跳记录路径上的每个路由器。

7)Traceroute 到达目的地的判断方法是什么？

Traceroute 通过接收目标主机返回的 ICMP 回应消息来判断是否到达目的地。如果收到目标主机的 ICMP 回应消息 (Echo Reply)，或达到最大 TTL 值仍未收到回应，则结束。

| Time  | Source    | Destination    | Protocol | Length | Info   |
|-------|-----------|----------------|----------|--------|--|
| 13576 | 40.037524 | 10.135.129.207 | ICMP     | 78     | Time to live exceeded (time to live exceeded in transit) |
| 13577 | 40.041565 | 202.115.32.43  | ICMP     | 78     | Time to live exceeded (time to live exceeded in transit) |
| 13578 | 40.041906 | 10.135.129.207 | ICMP     | 78     | Time to live exceeded (time to live exceeded in transit) |
| 13579 | 40.045167 | 202.115.39.102 | ICMP     | 78     | Time to live exceeded (time to live exceeded in transit) |
| 14743 | 41.046982 | 10.135.129.207 | ICMP     | 78     | Time to live exceeded (time to live exceeded in transit) |
| 14748 | 41.048626 | 202.115.32.43  | ICMP     | 78     | Time to live exceeded (time to live exceeded in transit) |
| 14749 | 41.049287 | 10.135.129.207 | ICMP     | 78     | Time to live exceeded (time to live exceeded in transit) |

8)从捕获的数据包中分析,源主机收到了哪些不同 IP 发送的 ICMP 报文？

收到了 10.135.135.254 202.115.39.33 202.115.39.102 202.115.32.43 这四个 IP 的 ICMP 报文，它们恰是追踪目标 IP 路程中的路由器

```
C:\Users\MountainMist>tracert /d www.scu.edu.cn

通过最多 30 个跃点跟踪
到 www.scu.edu.cn [202.115.32.43] 的路由：

  1      *          1 ms      *          10.135.135.254
  2      2 ms      1 ms      1 ms      202.115.39.33
  3     12 ms      4 ms      3 ms      202.115.39.102
  4      1 ms      1 ms      1 ms      202.115.32.43

跟踪完成。

C:\Users\MountainMist>
```

| Time  | Source    | Destination    | Protocol |
|-------|-----------|----------------|----------|
| 10333 | 34.991582 | 10.135.129.207 | ICMP     |
| 10334 | 34.993245 | 10.135.135.254 | ICMP     |
| 10335 | 34.993447 | 10.135.129.207 | ICMP     |
| 13101 | 38.996353 | 10.135.129.207 | ICMP     |
| 13102 | 38.998313 | 202.115.39.33  | ICMP     |
| 13103 | 38.998691 | 10.135.129.207 | ICMP     |
| 13106 | 39.000554 | 202.115.39.33  | ICMP     |
| 13107 | 39.001068 | 10.135.129.207 | ICMP     |
| 13108 | 39.002790 | 202.115.39.33  | ICMP     |
| 13574 | 40.024452 | 10.135.129.207 | ICMP     |
| 13575 | 40.037011 | 202.115.39.102 | ICMP     |
| 13576 | 40.037524 | 10.135.129.207 | ICMP     |
| 13577 | 40.041565 | 202.115.39.102 | ICMP     |
| 13578 | 40.041906 | 10.135.129.207 | ICMP     |
| 13579 | 40.045167 | 202.115.39.102 | ICMP     |
| 14743 | 41.046982 | 10.135.129.207 | ICMP     |
| 14748 | 41.048626 | 202.115.32.43  | ICMP     |

|             |  |
|-------------|--|
| 数据记录<br>和计算 | 实验过程及抓包数据如截图所示   |
| 结论<br>(结果)  | 通过使用 Wireshark 捕获并分析 ping 和 traceroute 命令的数据包，我们深入了解了 ICMP 协议的特点与原理。ping 命令通过发送 ICMP 回声请求 (Echo Request) 数据包并接收 ICMP 回应 (Echo Reply) 数据包，来测试目标主机的可达性和网络延迟。每个数据包包含时间戳和序列号，用于计算往返时间和检测数据包丢失。traceroute 命令通过发送具有不同 TTL 值的 ICMP 回声请求数据包，逐跳记录路径上的每个路由器。当 TTL 减为 0 时，路由器会返回 ICMP 超时 (Time Exceeded) 消息，直到目标主机返回 ICMP 回应消息，表示已到达目的地。通过分析这些数据包，我们可以清晰地看到网络路径上的每一个节点，以及各个节点的响应时间，从而诊断网络问题和优化网络性能。 |
| 小结          | 通过本次实验，我对 ICMP 协议的工作原理有了更深入的理解。使用 Wireshark 捕获和分析数据包，不仅让我掌握了基本的网络抓包技术，还帮助我直观地看到了 ping 和 traceroute 命令的具体实现过程。ping 命令的简单高效令人印象深刻，通过简单的 ICMP 回声请求和回应，就能快速检测网络连通性和延迟。而 traceroute 命令通过巧妙利用 TTL 机制，逐跳记录路径上的每个路由器，展示了网络路径的详细信息。这些工具在实际网络管理和故障排除中非常实用，通过这次实验，我不仅巩固了理论知识，还提高了实际操作能力，为今后的网络管理打下了坚实的基础。   |
| 指导老师<br>评议  | 成绩评定：<br><br>指导教师签名：   |