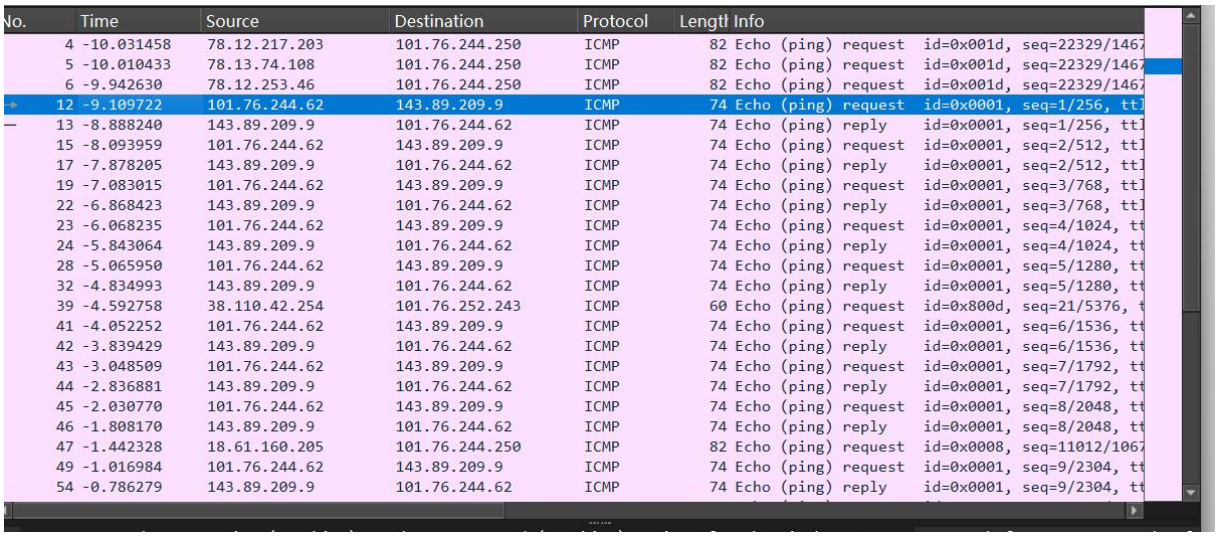
山东大学 计算机 学院

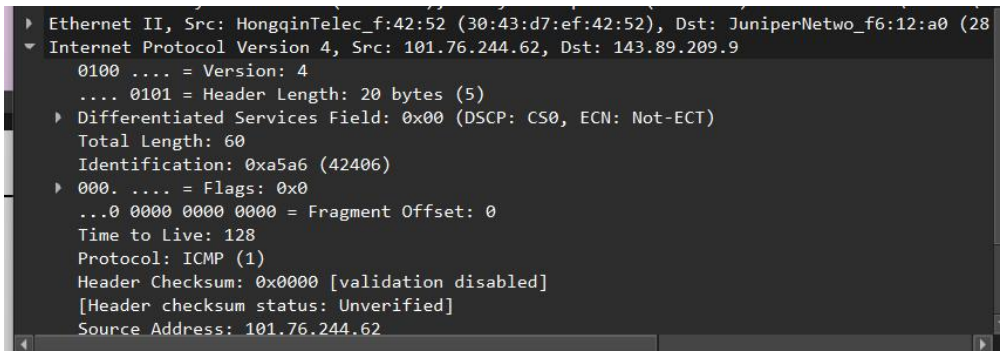
计算机网络 课程实验报告

学号：202400130240	姓名： 贾宗翰	班级： 24.6
实验题目：ICMP		
实验学时：2h	实验日期： 2025.10.28	
实验目的： 了解 ICMP		
硬件环境： 联想拯救者		
软件环境： Wireshark, edge		
实验步骤与内容： 首先打开 cmd 与 wireshark，输入 ping www.ust.hk -n 10：  截取如下：		

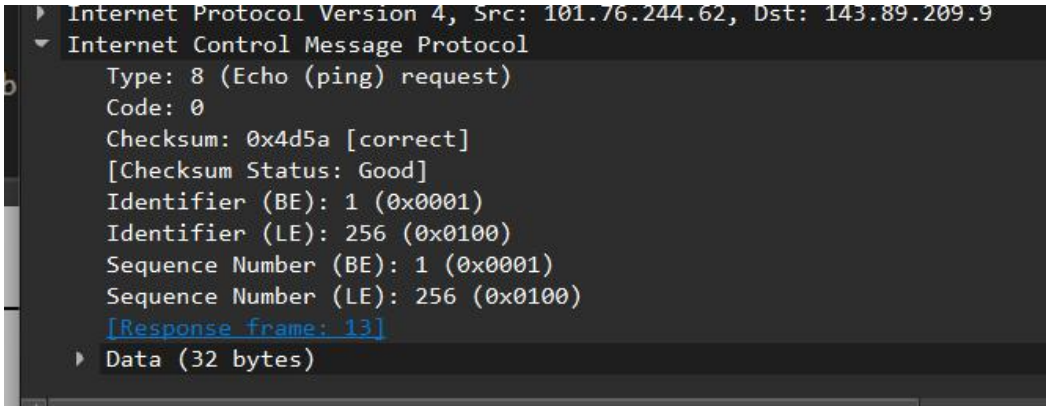
```
正在 Ping www.ust.hk [143.89.209.9] 具有 32 字节的数据:
来自 143.89.209.9 的回复: 字节=32 时间=217ms TTL=39
来自 143.89.209.9 的回复: 字节=32 时间=211ms TTL=39
来自 143.89.209.9 的回复: 字节=32 时间=210ms TTL=39
来自 143.89.209.9 的回复: 字节=32 时间=208ms TTL=39
来自 143.89.209.9 的回复: 字节=32 时间=207ms TTL=39
来自 143.89.209.9 的回复: 字节=32 时间=208ms TTL=39
来自 143.89.209.9 的回复: 字节=32 时间=208ms TTL=39
来自 143.89.209.9 的回复: 字节=32 时间=207ms TTL=39
来自 143.89.209.9 的回复: 字节=32 时间=208ms TTL=39
来自 143.89.209.9 的回复: 字节=32 时间=208ms TTL=39

143.89.209.9 的 Ping 统计信息:
    数据包: 已发送 = 10, 已接收 = 10, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 207ms, 最长 = 217ms, 平均 = 209ms
```



No.	Time	Source	Destination	Protocol	Length	Info
4	-10.031458	78.12.217.203	101.76.244.250	ICMP	82	Echo (ping) request id=0x001d, seq=22329/1467
5	-10.010433	78.13.74.108	101.76.244.250	ICMP	82	Echo (ping) request id=0x001d, seq=22329/1467
6	-9.942630	78.12.253.46	101.76.244.250	ICMP	82	Echo (ping) request id=0x001d, seq=22329/1467
12	-9.109722	101.76.244.62	143.89.209.9	ICMP	74	Echo (ping) request id=0x0001, seq=1/256, ttl
13	-8.888240	143.89.209.9	101.76.244.62	ICMP	74	Echo (ping) reply id=0x0001, seq=1/256, ttl
15	-8.093959	101.76.244.62	143.89.209.9	ICMP	74	Echo (ping) request id=0x0001, seq=2/512, ttl
17	-7.878205	143.89.209.9	101.76.244.62	ICMP	74	Echo (ping) reply id=0x0001, seq=2/512, ttl
19	-7.083015	101.76.244.62	143.89.209.9	ICMP	74	Echo (ping) request id=0x0001, seq=3/768, ttl
22	-6.868423	143.89.209.9	101.76.244.62	ICMP	74	Echo (ping) reply id=0x0001, seq=3/768, ttl
23	-6.068235	101.76.244.62	143.89.209.9	ICMP	74	Echo (ping) request id=0x0001, seq=4/1024, tt
24	-5.843064	143.89.209.9	101.76.244.62	ICMP	74	Echo (ping) reply id=0x0001, seq=4/1024, tt
28	-5.065950	101.76.244.62	143.89.209.9	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, tt
32	-4.834993	143.89.209.9	101.76.244.62	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, tt
39	-4.592758	38.110.42.254	101.76.252.243	ICMP	60	Echo (ping) request id=0x800d, seq=21/5376, t
41	-4.052252	101.76.244.62	143.89.209.9	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, tt
42	-3.839429	143.89.209.9	101.76.244.62	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, tt
43	-3.048509	101.76.244.62	143.89.209.9	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, tt
44	-2.836881	143.89.209.9	101.76.244.62	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, tt
45	-2.030770	101.76.244.62	143.89.209.9	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, tt
46	-1.808170	143.89.209.9	101.76.244.62	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, tt
47	-1.442328	18.61.160.205	101.76.244.250	ICMP	82	Echo (ping) request id=0x0008, seq=11012/1067
49	-1.016984	101.76.244.62	143.89.209.9	ICMP	74	Echo (ping) request id=0x0001, seq=9/2304, tt
54	-0.786279	143.89.209.9	101.76.244.62	ICMP	74	Echo (ping) reply id=0x0001, seq=9/2304, tt



```
▸ Ethernet II, Src: HongqinTelec_f:42:52 (30:43:d7:ef:42:52), Dst: JuniperNetwo_f6:12:a0 (28
▾ Internet Protocol Version 4, Src: 101.76.244.62, Dst: 143.89.209.9
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0xa5a6 (42406)
  ▸ 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: ICMP (1)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 101.76.244.62
```



```
▸ Internet Protocol Version 4, Src: 101.76.244.62, Dst: 143.89.209.9
▾ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4d5a [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 1 (0x0001)
    Sequence Number (LE): 256 (0x0100)
    [Response frame: 13]
  ▸ Data (32 bytes)
```

1.您的主机的 IP 地址是什么？目标主机的 IP 地址是什么？

可以看出，我的 ip 是 101.76.244.62，目标是 143.89.209.9。

2. 为什么 ICMP 数据包没有源端口号和目标端口号？

因为 ICMP 是一种在 IP 上运行的网络层协议，不是传输层协议，因此不存在端口的概念。

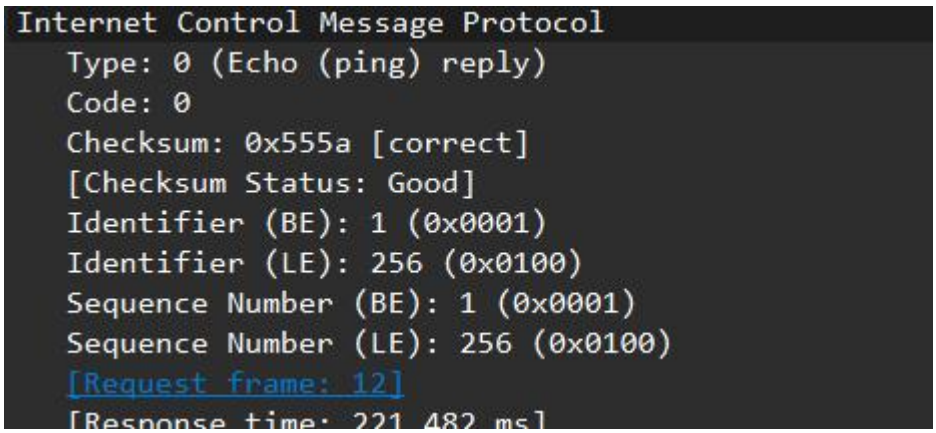
3. 检查主机发送的其中一个 ping 请求数据包。ICMP 类型和代码有哪些？此 ICMP 数据包还有哪些其他字段？校验和、序列号和标识符字段有多少字节？

Type: 8 (Echo (ping) request)

Code: 0

分析此图，还有校验和、序列号和标识符字段，各是 2 个字节。

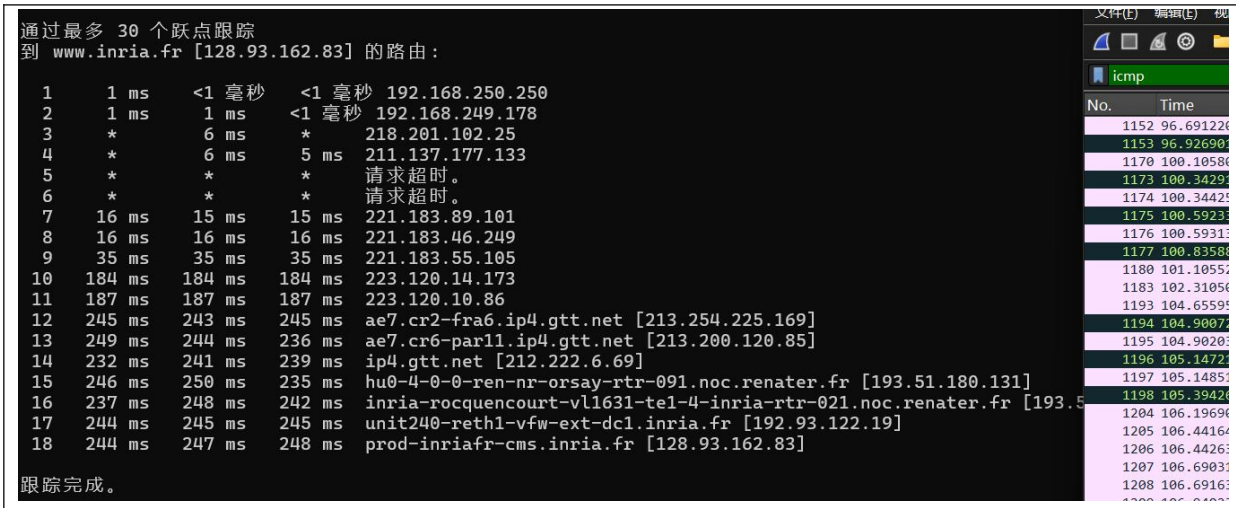
4. 检查相应的 ping 回复数据包。ICMP 类型和代码有哪些？此 ICMP 数据包还有哪些其他字段？校验和、序列号和标识符字段有多少字节？



Type：0(Echo (ping) reply)
Code: 0
还有校验和、序列号和标识符字段，各是 2 个字节。

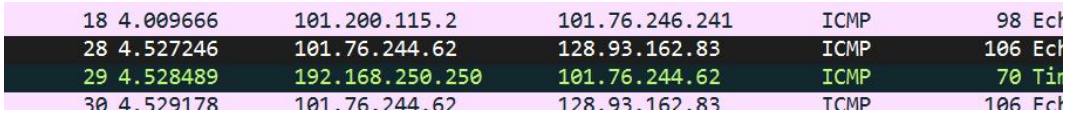
现在让我们通过捕获 Traceroute 程序生成的数据包来继续我们的 ICMP 冒险。您可能还记得，Traceroute 程序可用于计算数据包从源到目的地的路径。Traceroute 在文本的 Section 1.4 和 Section 5.6 中讨论。Traceroute 在 Unix/Linux/MacOS 和 Windows 中以不同的方式实现。在 Unix/Linux 中，源使用不太可能的目标端口号将一系列 UDP 数据包发送到目标目标;在 Windows 中，源将一系列 ICMP 数据包发送到目标目标。对于这两个作系统，程序都会发送 TTL=1 的第一个数据包，TTL=2 的第二个数据包，依此类推。回想一下，当数据包通过路由器时，路由器将递减数据包的 TTL 值。当数据包到达 TTL=1 的路由器时，路由器会将 ICMP 错误数据包发送回源。在下文中，我们将使用本机 Windows tracert 程序。一个非常好的 Windows Traceroute 程序的共享软件版本是 pingplotter（www.pingplotter.com ）。我们将在 Wireshark IP 实验室中使用 pingplotter，因为它提供了我们在那里需要的额外功能。

执行以下作 4： • 让我们首先打开 Windows 命令提示符应用程序（可以在您的 Accessories 文件夹中找到）。• 启动 Wireshark 数据包嗅探器，然后开始 Wireshark 数据包捕获。跟踪器命令位于 c：windowssystem32 中，因此请在 MS-DOS 命令行中键入“tracert hostname”或“c：windowssystem32tracert hostname”（不带引号），其中主机名是另一个大洲的主机。（请注意，在 Windows 计算机上，命令是 “tracert” 而不是“traceroute”。如果您在欧洲以外，则可能需要为法国计算机科学研究机构 INRIA 的 Web 服务器输入 www.inria.fr。然后通过键入 return 运行 Traceroute 程序。• 当 Traceroute 程序终止时，停止 Wireshark 中的数据包捕获。在实验结束时，您的 Command Prompt Window 应如图 4 所示。在此图中，客户端 Traceroute 程序位于马萨诸塞州，目标目的地位于法国。从该图中，我们可以看到，对于每个 TTL 值，源程序发送三个探测数据包。Traceroute 显示每个探测数据包的 RTT，以及返回 ICMP TTL-exceeded 消息的路由器的 IP 地址（可能还有名称）。



```
通过最多 30 个跃点跟踪
到 www.inria.fr [128.93.162.83] 的路由：

  1     1 ms    <1 毫秒    <1 毫秒 192.168.250.250
  2     1 ms     1 ms    <1 毫秒 192.168.249.178
  3     *        6 ms     *      218.201.102.25
  4     *        6 ms     5 ms   211.137.177.133
  5     *        *        *      请求超时。
  6     *        *        *      请求超时。
  7    16 ms    15 ms    15 ms   221.183.89.101
  8    16 ms    16 ms    16 ms   221.183.46.249
  9    35 ms    35 ms    35 ms   221.183.55.105
 10   184 ms   184 ms   184 ms   223.120.14.173
 11   187 ms   187 ms   187 ms   223.120.10.86
 12   245 ms   243 ms   245 ms   ae7.cr2-fra6.ip4.gtt.net [213.254.225.169]
 13   249 ms   244 ms   236 ms   ae7.cr6-par11.ip4.gtt.net [213.200.120.85]
 14   232 ms   241 ms   239 ms   ip4.gtt.net [212.222.6.69]
 15   246 ms   250 ms   235 ms   hu0-4-0-0-ren-nr-orsay-rtr-091.noc.renater.fr [193.51.180.131]
 16   237 ms   248 ms   242 ms   inria-rocquencourt-vl1631-te1-4-inria-rtr-021.noc.renater.fr [193.5
 17   244 ms   245 ms   245 ms   unit240-reth1-vfw-ext-dc1.inria.fr [192.93.122.19]
 18   244 ms   247 ms   248 ms   prod-inriafr-cms.inria.fr [128.93.162.83]

跟踪完成。
```

5. 您的主机的 IP 地址是什么？目标主机的 IP 地址是什么？



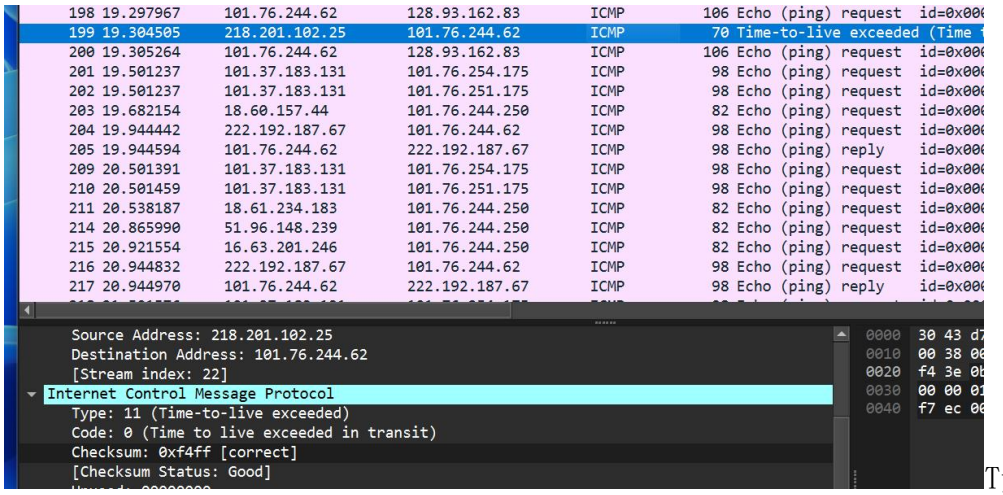
18	4.009666	101.200.115.2	101.76.246.241	ICMP	98 Ech
28	4.527246	101.76.244.62	128.93.162.83	ICMP	106 Ech
29	4.528489	192.168.250.250	101.76.244.62	ICMP	70 Tir
30	4.529178	101.76.244.62	128.93.162.83	ICMP	106 Ech

主机的 IP 是 101.76.244.62，目标是 128.93.162.83

6. 如果 ICMP 发送的是 UDP 数据包（如在 Unix/Linux 中），探测数据包的 IP 协议号是否仍为 01？如果不是，那会是什么？

发送请求路由跟踪的数据包是 UDP 数据包，因此 IP 承载上层协议号时 17。

7. 检查屏幕截图中的 ICMP 回声数据包。这与本实验前半部分的 ICMP ping 查询数据包不同吗？如果是，如何？



198	19.297967	101.76.244.62	128.93.162.83	ICMP	106 Echo (ping) request id=0x000
199	19.304505	218.201.102.25	101.76.244.62	ICMP	70 Time-to-live exceeded (Time t
200	19.305264	101.76.244.62	128.93.162.83	ICMP	106 Echo (ping) request id=0x000
201	19.501237	101.37.183.131	101.76.254.175	ICMP	98 Echo (ping) request id=0x000
202	19.501237	101.37.183.131	101.76.251.175	ICMP	98 Echo (ping) request id=0x000
203	19.682154	18.60.157.44	101.76.244.250	ICMP	82 Echo (ping) request id=0x000
204	19.944442	222.192.187.67	101.76.244.62	ICMP	98 Echo (ping) request id=0x000
205	19.944594	101.76.244.62	222.192.187.67	ICMP	98 Echo (ping) reply id=0x000
209	20.501391	101.37.183.131	101.76.254.175	ICMP	98 Echo (ping) request id=0x000
210	20.501459	101.37.183.131	101.76.251.175	ICMP	98 Echo (ping) request id=0x000
211	20.538187	18.61.234.183	101.76.244.250	ICMP	82 Echo (ping) request id=0x000
214	20.865990	51.96.148.239	101.76.244.250	ICMP	82 Echo (ping) request id=0x000
215	20.921554	16.63.201.246	101.76.244.250	ICMP	82 Echo (ping) request id=0x000
216	20.944832	222.192.187.67	101.76.244.62	ICMP	98 Echo (ping) request id=0x000
217	20.944970	101.76.244.62	222.192.187.67	ICMP	98 Echo (ping) reply id=0x000

```
Source Address: 218.201.102.25
Destination Address: 101.76.244.62
[Stream index: 22]
Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xf4ff [correct]
    [Checksum Status: Good]
```

Type 是 11，说明这个是在 Traceroute 程序中，路由器检查到 Traceroute 发出的 IP 数据报中 TTL 正好过期，因此路由器就需要丢包并且发送该警告报文返回源主机。这个与 Ping 程序中所要达成的目的不同，Ping 程序是为了请求响应。

8. 检查屏幕截图中的 ICMP 错误数据包。它的字段比 ICMP 回应数据包多。这些字段包括哪些内容？

发现这些多出的字段是 ICMP 请求数据包的内容。

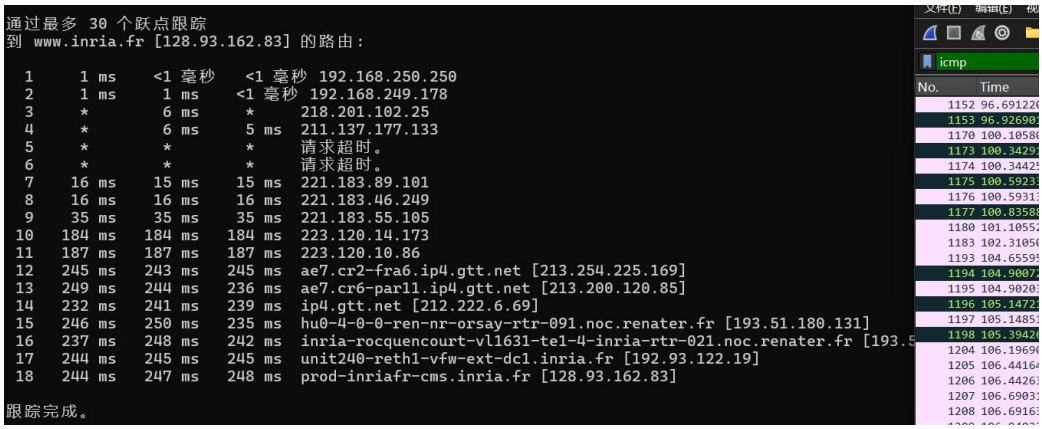
```
[Stream index: 22]
Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xf4ff [correct]
    [Checksum Status: Good]
    Unused: 00000000
    Internet Protocol Version 4, Src: 101.76.244.62, Dst: 128.93.162.83
        0100 .... = Version: 4
        .... 0101 = Header Length: 20 bytes (5)
      ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        Total Length: 92
        Identification: 0xe179 (57721)
      ▸ 000. .... = Flags: 0x0
        ...0 0000 0000 0000 = Fragment Offset: 0
      ▾ Time to Live: 1
          [Expert Info (Note/Sequence): "Time To Live" only 1]
        Protocol: ICMP (1)
        Header Checksum: 0x5bec [validation disabled]
        [Header checksum status: Unverified]
        Source Address: 101.76.244.62
        Destination Address: 128.93.162.83
        [Stream index: 4]
    Internet Control Message Protocol
        Type: 8 (Echo (ping) request)
        Code: 0
        Checksum: 0xf7ec [unverified] [in ICMP error packet]
        [Checksum Status: Unverified]
```

9. 检查源主机收到的最后三个 ICMP 数据包。这些数据包与 ICMP 错误数据包有何不同？为什么它们不同？

1194 104.900723	192.93.122.19	101.76.244.62	ICMP	70	Time-to-live exceeded (Time to live exceeded
1195 104.902033	101.76.244.62	128.93.162.83	ICMP	106	Echo (ping) request id=0x0001, seq=60/15360,
1196 105.147219	192.93.122.19	101.76.244.62	ICMP	70	Time-to-live exceeded (Time to live exceeded
1197 105.148516	101.76.244.62	128.93.162.83	ICMP	106	Echo (ping) request id=0x0001, seq=61/15616,
1198 105.394263	192.93.122.19	101.76.244.62	ICMP	70	Time-to-live exceeded (Time to live exceeded
1204 106.196902	101.76.244.62	128.93.162.83	ICMP	106	Echo (ping) request id=0x0001, seq=62/15872,
1205 106.441646	128.93.162.83	101.76.244.62	ICMP	106	Echo (ping) reply id=0x0001, seq=62/15872,
1206 106.442634	101.76.244.62	128.93.162.83	ICMP	106	Echo (ping) request id=0x0001, seq=63/16128,
1207 106.690312	128.93.162.83	101.76.244.62	ICMP	106	Echo (ping) reply id=0x0001, seq=63/16128,
1208 106.691638	101.76.244.62	128.93.162.83	ICMP	106	Echo (ping) request id=0x0001, seq=64/16384,
1209 106.940236	128.93.162.83	101.76.244.62	ICMP	106	Echo (ping) reply id=0x0001, seq=64/16384,
1216 108.298941	203.178.148.19	101.76.252.207	ICMP	60	Echo (ping) request id=0xea85, seq=314/14849

源主机收到的最后三个 ICMP 数据包是目的主机发送给我的 ICMP 回应数据包，因为路由查询是使用逐渐递增 TTL 的查询数据包，最后的 ICMP 查询数据包的 TTL 已经大于到达目的主机中间路由跃点数，因此不会被目标主机丢弃来发送 ICMP 超时的数据包，所以只会收到 ICMP 响应数据包。也就是说而最后一组 3 个数据报时可以到达目的主机的，这时由于是被目的主机接收，目的主机不会丢包，而是确确实实收到的这个探测的数据报并进行了响应。

10. 在 tracert 测量中，是否存在延迟明显长于其他链路的链路？参考图 4 中的屏幕截图，是否有链路的延迟明显长于其他链路？根据路由器名称，您能猜出此链路末端两台路由器的位置吗？



如图，9-10 是突然延迟上升的，分析原因在于连接到了亚洲转欧洲的分界路由器，具体来看

你的外网IP地址是：101.76.244.62

请输入IP或网站域名：221.183.55.105

IP 地址:	221.183.55.105
IP Long:	3719772009

高精归属地定位（IP数据云）
中国 北京 北京

归属地(纯真数据):	中国 广东 广州 移动/骨干网
归属地(ipip):	中国 中国 -
归属地(淘宝数据):	
归属地(IP2REGION):	中国 移动
归属地(GeoLite2):	China -
归属地(DbIp):	China -

你的外网IP地址是：101.76.244.62

请输入IP或网站域名：223.120.14.173

IP 地址:	223.120.14.173
IP Long:	3749187245

高精归属地定位（IP数据云）
德国 Hesse Frankfurt am Main

归属地(纯真数据):	中国 移动
归属地(ipip):	CHINAMOBILE.COM 骨干网 CHI
归属地(淘宝数据):	
归属地(IP2REGION):	中国 移动
归属地(GeoLite2):	Hong Kong -
归属地(DbIp):	Hong Kong -

发现确实进行了比较长的跳转

结论分析与体会：

通过本次 ICMP 实验，我对 ICMP 协议的工作原理和实际应用有了更深入的理解。