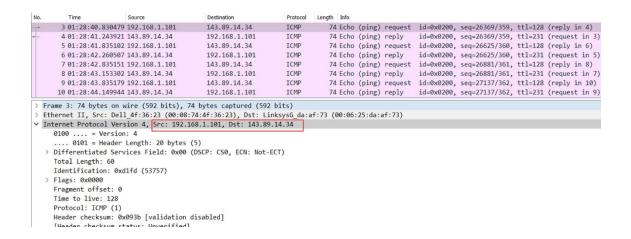


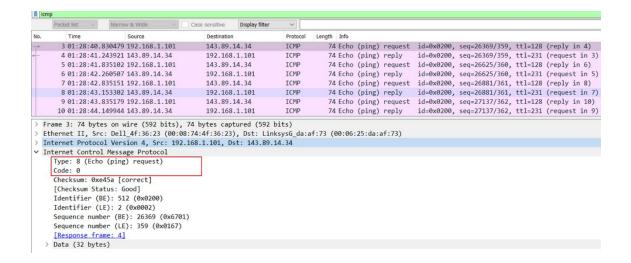


1/ What is the IP address of your host? What is the IP address of the destination host? Answer: The IP address of my host is 192.168.1.101. The IP address of the destination host is 143.89.14.34.



2/ Why is it that an ICMP packet does not have source and destination port numbers? Answer: The ICMP packet does not have source and destination port numbers because it was designed to communicate network-layer information between hosts and routers, not between application layer processes. Instead it uses the type and code message to

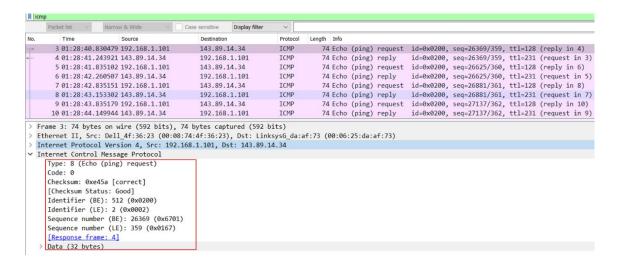
not between application layer processes. Instead it uses the type and code message to specify the message received. The network decodes all ICMP messages so port numbers are unnecessary.





# 3/ Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?

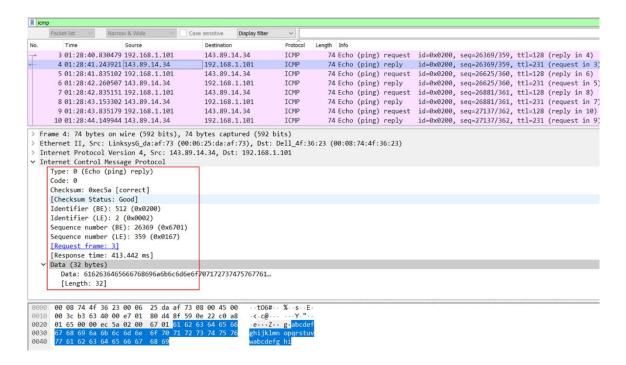
Answer: The ICMP type is 8 and the code number is 0. Other fields in this packet are checksum, identifier, sequence number and data field. The checksum, identifier and sequence number are 2 bytes each. The data field is 32 bytes



4/ Examine the corresponding ping reply packet. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?

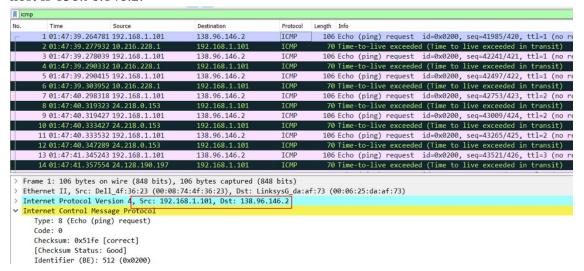
Answer: The ICMP type is 0 and the code number is 0. Other fields in this packet are checksum, identifier, sequence number and data field. The checksum, identifier and sequence number are 2 bytes each. The data field is 32 bytes.





#### 5/ What is the IP address of your host? What is the IP address of the target destination host?

Answer: The IP address of my host is 192.168.1.101. The IP address of the destination host is 138.96.146.2.



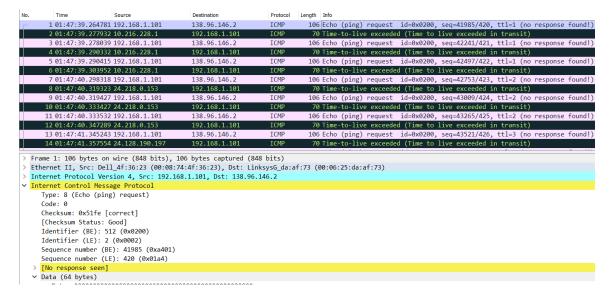


### 6/ If ICMP sent UDP packets instead (as in Unix/Linux), would the IP protocol number still be 01 for the probe packets? If not, what would it be?

Answer: It would actually be different if ICMP sent UDP packets. Instead of 01 it would be switched to 0X11.

# 7/ Examine the ICMP echo packet in your screenshot. Is this different from the ICMP ping query packets in the first half of this lab? If yes, how so?

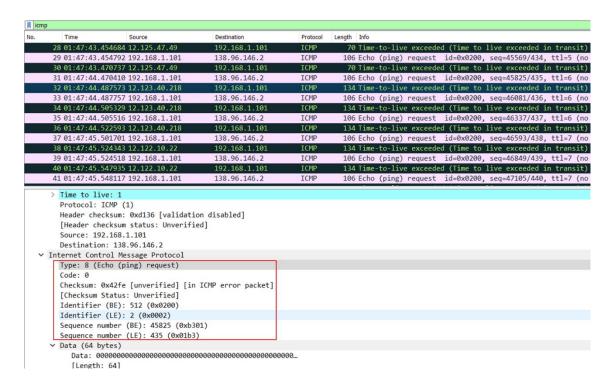
Answer: The echo packets have the same ones as the ping query.



### 8/ Examine the ICMP error packet in your screenshot. It has more fields than the ICMP echo packet. What is included in those fields?

Answer: The ICMP error packet is not the same as the ping query packets. It has the IP header and the first 8 bytes of the original ICMP packet.

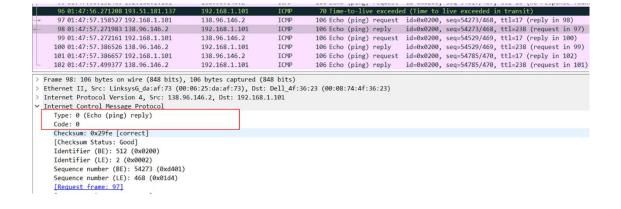




# 9/ Examine the last three ICMP packets received by the source host. How are these packets different from the ICMP error packets? Why are they different?

Answer: Instead of being type 11, the last three packets are all type 0. They are different because they have all arrived before the TTL expires.

#### ICMP packet sent by the host:





#### ICMP error packet:

```
70 Time-to-live exceeded (Time to live exceeded in transit)
     97 01:47:57.158527 192.168.1.101
98 01:47:57.271983 138.96.146.2
                                                                                          106 Echo (ping) request id=0x0200, seq=54273/468, ttl=17 (reply in 98)
106 Echo (ping) reply id=0x0200, seq=54273/468, ttl=238 (request in 97)
                                                     138.96.146.2
                                                                              ICMP
                                                    192.168.1.101
                                                                              ICMP
                                                                                          106 Echo (ping) request id=0x0200, seq=54529/469, ttl=17 (reply in 100)
     99 01:47:57.272161 192.168.1.101
                                                                                          106 Echo (ping) reply id=0x0200, seq=54529/469, ttl=238 (request in 99) 106 Echo (ping) request id=0x0200, seq=54785/470, ttl=17 (reply in 102)
    100 01:47:57.386526 138.96.146.2
                                                    192,168,1,101
                                                                              ICMP
    101 01:47:57.386657 192.168.1.101
                                                    138.96.146.2
                                                                              ICMP
   102 01:47:57.499377 138.96.146.2
                                                    192.168.1.101
                                                                              ICMP
                                                                                                                        id=0x0200, seq=54785/470, ttl=238 (request in 101
Frame 96: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
Ethernet II, Src: Linksys6_da:af:73 (00:06:25:da:af:73), Dst: Dell_4f:36:23 (00:08:74:4f:36:23)
Internet Protocol Version 4, Src: 193.51.181.137, Dst: 192.168.1.101
Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
   Code: 0 (Time to live exceeded in transit)
    Checksum: 0x2c16 [correct]
   [Checksum Status: Good]
    Unused: 00000000
   Internet Protocol Version 4, Src: 192.168.1.101, Dst: 138.96.146.2

✓ Internet Control Message Protocol

       Type: 8 (Echo (ping) request)
       Code: 0
       Checksum: 0x22fe [unverified] [in ICMP error packet]
       [Checksum Status: Unverified]
       Identifier (BE): 512 (0x0200)
       Identifier (LE): 2 (0x0002)
```

10/Within the tracert measurements, is there a link whose delay is significantly longer than others? Refer to the screenshot in Figure 4, is there a link whose delay is significantly longer than others? On the basis of the router names, can you guess the location of the two routers on the end of this link?

Answer: There is a link between steps 9 and 10 that has a significantly longer delay. The location of the link is from New York City to Pastourelle, France.

```
Command Prompt
C:\WINDOWS\SYSTEM32>
C:\WINDOWS\SYSTEM32>
C:\WINDOWS\SYSTEM32>
C:\WINDOWS\SYSTEM32>tracert www.inria.fr
Tracing route to www.inria.fr [138.96.146.2]
over a maximum of 30 hops:
                                                                                      10.216.228.1
24.218.0.153
bar01-p4-0.wsfdhe1.ma.attbb.net [24.128.190.197]
bar02-p6-0.ndhmhe1.ma.attbb.net [24.128.0.101]
12.125.47.49
12.123.40.218
                   13 ms
               13 ms
21 ms
12 ms
16 ms
15 ms
17 ms
22 ms
23 ms
26 ms
98 ms
97 ms
104 ms
                                                                   13 ms
13 ms
13 ms
15 ms
15 ms
17 ms
22 ms
                                          12 ms
14 ms
11 ms
16 ms
15 ms
17 ms
23 ms
                                                                                      tbr2-c11.n54ny.ip.att.net [12.122.10.22]
ggr2-p3120.n54ny.ip.att.net [12.123.3.109]
att-gw.nyc.opentransit.net [192.205.32.138]
P4-0.PASCRI.Pastourelle.opentransit.net [193.251.241.133]
P9-0.AUUCRI.Aubervilliers.opentransit.net [193.251.243.29]
                                           21 ms
98 ms
98 ms
  10
11
12
13
14
15
                                                                    96 ms
98 ms
                                           98 ms
                                                                                      P6-0.BAGCR1.Bagnolet.opentransit.net [193.251.241.93]
193.51.185.30
                                                                108 ms
                                                                103 ms
117 ms
114 ms
                                        106 ms
                                        114 ms
115 ms
114 ms
114 ms
                                                                                      grenoble-pos1-0.cssi.renater.fr [193.51.179.238]
nice-pos2-0.cssi.renater.fr [193.51.180.34]
inria-nice.cssi.renater.fr [193.51.181.137]
www.inria.fr [138.96.146.2]
                114 ms
114 ms
               129 ms
113 ms
Trace complete.
C:\WINDOWS\SYSTEM32>_
```