National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member: **Dr. Hassan Khaliq** **Dated:10/4/2022**

Semester: **6th** **Section:** **D**

EE-357 Computer and Communication Networks

Experiment - 8

Wireshark – ICMP (Internet Control Message Protocol)

		PLO5/ CLO3		PLO5/ CLO3	PLO5/ CLO3	PLO5/ CLO3
Name	**Reg. No**	**Viva / Quiz / Lab Performance** **5 Marks**	**Analysis of data in Lab Report** **5 Marks**	**Modern Tool Usage** **5 Marks**	**Ethics and Safety** **5 Marks**	**Individual and Team Work** **5 Marks**
Myesha Khalil	**305093**					
Noor-ul-Ain Ansar	**284825**					

National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

EXPERIMENT NO 8

Wireshark – ICMP (Internet Control Message Protocol)

1.0 Objective of this lab:

In this lab, we'll explore several aspects of the ICMP protocol. We will specifically study ICMP messages that are used by the Ping and Traceroute programs. We will also study the general format of ICMP messages.

2.0 Instructions:

- Read carefully before starting the lab.
- These exercises are to be done individually.
- You are supposed to provide the answers to the questions listed at the end of this document, paste the screenshots of your working and upload the completed report to your course's LMS site.
- Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work.

3.0 Background:

3.1 Internet protocol:

The Internet Protocol (IP) takes the stage when the datagram service between the hosts is required in an interconnected network. These internetworks are connected using the devices called Gateways. In some occasions where the gateway or the destination host needs to communicate with the source host and these situations arise when there is an error in datagram processing. These situations occur occasionally and in these situations, the ICMP (Internet Control Message Protocol) is used.

Some of the situations where the ICMP messages are sent are

1. The datagram is not transmitted to the destination host because of no routing entry.
2. When the gateway identifies a shorter route that the host can take to send the datagrams to a destination.

3.2 ICMP Message Types:

Based on the type field, the message types are categorized into different groups. Each of these messages carries different fields with respect to ICMP over the network. The summary of these messages and the type is tabularized below.

Type	Message Category
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request
11	Time Exceeded
12	Parameter Problem
13	Time Stamp
14	Time Stamp Reply
15	Information Request
16	Information Reply

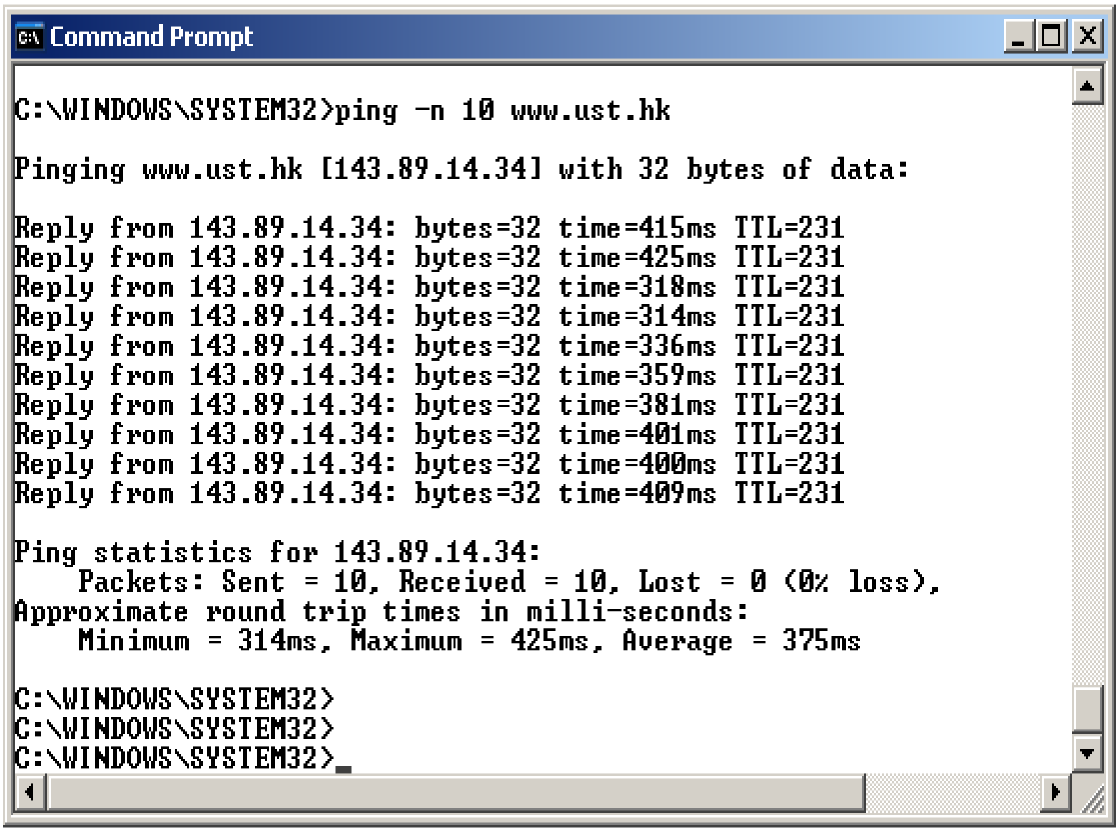
3.3 ICMP and Ping:

Let’s begin our ICMP adventure by capturing the packets generated by the Ping program. You may recall that the Ping program is simple tool that allows anyone (for example, a network administrator) to verify if a host is live or not. The Ping program in the source host sends a packet to the target IP address; if the target is live, the Ping program in the target host responds by sending a packet back to the source host. As you might have guessed (given that this lab is about ICMP), both of these Ping packets are ICMP packets.

Do the following:

- Let’s begin this adventure by opening the Windows Command Prompt application (which can be found in your Accessories folder).
- Start up the Wireshark packet sniffer, and begin Wireshark packet capture.
- The *ping* command is in c:\windows\system32, so type either “*ping –n 10 hostname*” or “*c:\windows\system32\ping –n 10 hostname*” in the MS-DOS command line (without quotation marks), where hostname is a host on another continent. If you’re outside of Asia, you may want to enter www.ust.hk for the Web server at Hong Kong University of Science and Technology. The argument *“-n 10*” indicates that 10 ping messages should be sent. Then run the Ping program by typing return.
- When the Ping program terminates, stop the packet capture in Wireshark.

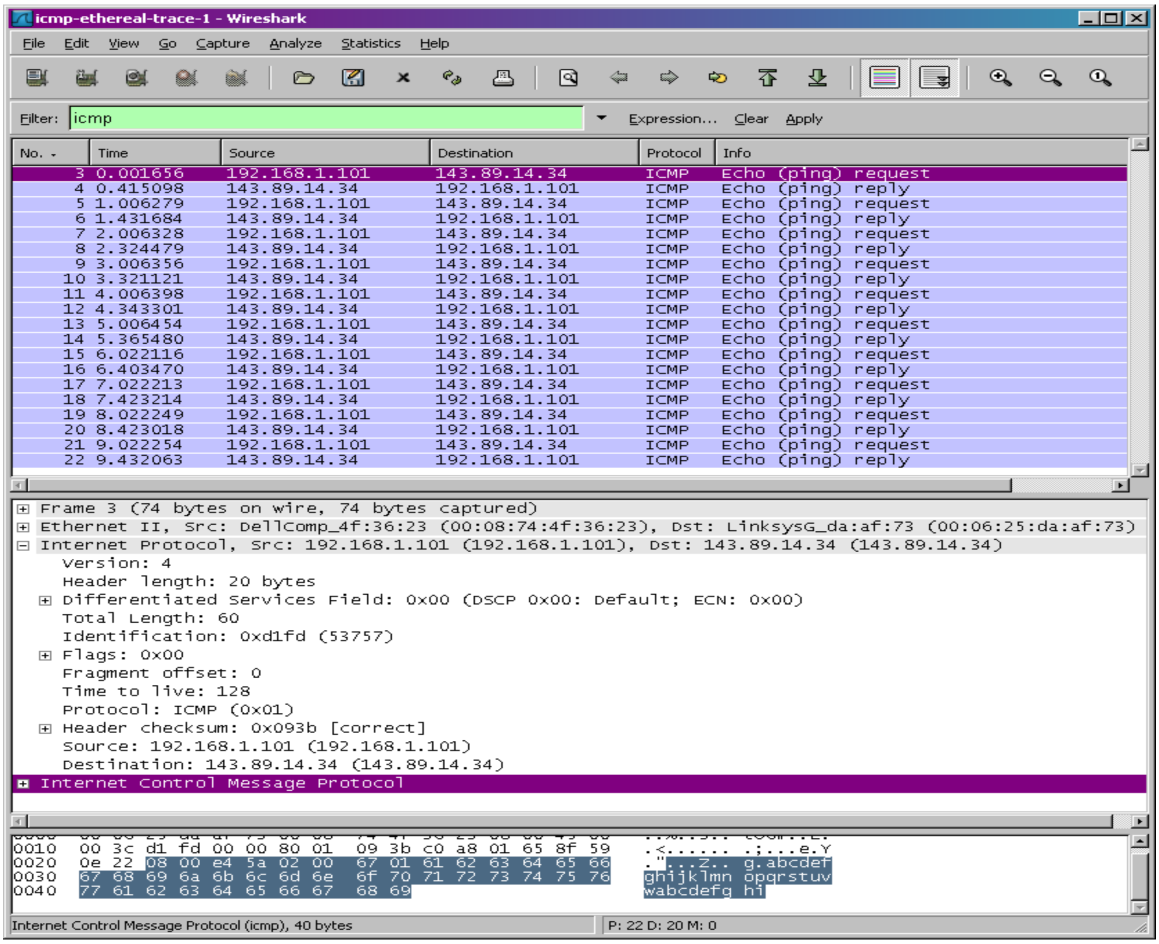
At the end of the experiment, your Command Prompt Window should look something like Figure 1. In this example, the source ping program is in Massachusetts and the destination Ping program is in Hong Kong. From this window we see that the source ping program sent 10 query packets and received 10 responses. Note also that for each response, the source calculates the round-trip time (RTT), which for the 10 packets is on average 375 msec.



```
C:\WINDOWS\SYSTEM32>ping -n 10 www.ust.hk

Pinging www.ust.hk [143.89.14.34] with 32 bytes of data:

Reply from 143.89.14.34: bytes=32 time=415ms TTL=231
Reply from 143.89.14.34: bytes=32 time=425ms TTL=231
Reply from 143.89.14.34: bytes=32 time=318ms TTL=231
Reply from 143.89.14.34: bytes=32 time=314ms TTL=231
Reply from 143.89.14.34: bytes=32 time=336ms TTL=231
Reply from 143.89.14.34: bytes=32 time=359ms TTL=231
Reply from 143.89.14.34: bytes=32 time=381ms TTL=231
Reply from 143.89.14.34: bytes=32 time=401ms TTL=231
Reply from 143.89.14.34: bytes=32 time=400ms TTL=231
Reply from 143.89.14.34: bytes=32 time=409ms TTL=231

Ping statistics for 143.89.14.34:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 314ms, Maximum = 425ms, Average = 375ms

C:\WINDOWS\SYSTEM32>
C:\WINDOWS\SYSTEM32>
C:\WINDOWS\SYSTEM32>_
```

Figure 1 Command Prompt window after entering Ping command.

Figure 2 provides a screenshot of the Wireshark output, after “icmp” has been entered into the filter display window. Note that the packet listing shows 20 packets: the 10 Ping queries sent by the source and the 10 Ping responses received by the source. Also note that the source’s IP address is a private address; the destination’s IP address is that of the Web server at HKUST. Now let’s zoom in on the first packet (sent by the client); in the figure below, the packet contents area provides information about this packet. We see that the IP datagram within this packet has protocol number 01, which is the protocol number for ICMP. This means that the payload of the IP datagram is an ICMP packet.



icmp-ethereal-trace-1 - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: icmp
Expression... Clear Apply

```
No. -  Time      Source         Destination    Protocol Info
   3 0.001656    192.168.1.101  143.89.14.34   ICMP     Echo (ping) request
   4 0.415098    143.89.14.34   192.168.1.101  ICMP     Echo (ping) reply
   5 1.006279    192.168.1.101  143.89.14.34   ICMP     Echo (ping) request
   6 1.431684    143.89.14.34   192.168.1.101  ICMP     Echo (ping) reply
   7 2.006328    192.168.1.101  143.89.14.34   ICMP     Echo (ping) request
   8 2.324479    143.89.14.34   192.168.1.101  ICMP     Echo (ping) reply
   9 3.006356    192.168.1.101  143.89.14.34   ICMP     Echo (ping) request
  10 3.321121    143.89.14.34   192.168.1.101  ICMP     Echo (ping) reply
  11 4.006398    192.168.1.101  143.89.14.34   ICMP     Echo (ping) request
  12 4.343301    143.89.14.34   192.168.1.101  ICMP     Echo (ping) reply
  13 5.006454    192.168.1.101  143.89.14.34   ICMP     Echo (ping) request
  14 5.365480    143.89.14.34   192.168.1.101  ICMP     Echo (ping) reply
  15 6.022116    192.168.1.101  143.89.14.34   ICMP     Echo (ping) request
  16 6.403470    143.89.14.34   192.168.1.101  ICMP     Echo (ping) reply
  17 7.022213    192.168.1.101  143.89.14.34   ICMP     Echo (ping) request
  18 7.423214    143.89.14.34   192.168.1.101  ICMP     Echo (ping) reply
  19 8.022249    192.168.1.101  143.89.14.34   ICMP     Echo (ping) request
  20 8.423018    143.89.14.34   192.168.1.101  ICMP     Echo (ping) reply
  21 9.022254    192.168.1.101  143.89.14.34   ICMP     Echo (ping) request
  22 9.432063    143.89.14.34   192.168.1.101  ICMP     Echo (ping) reply
```

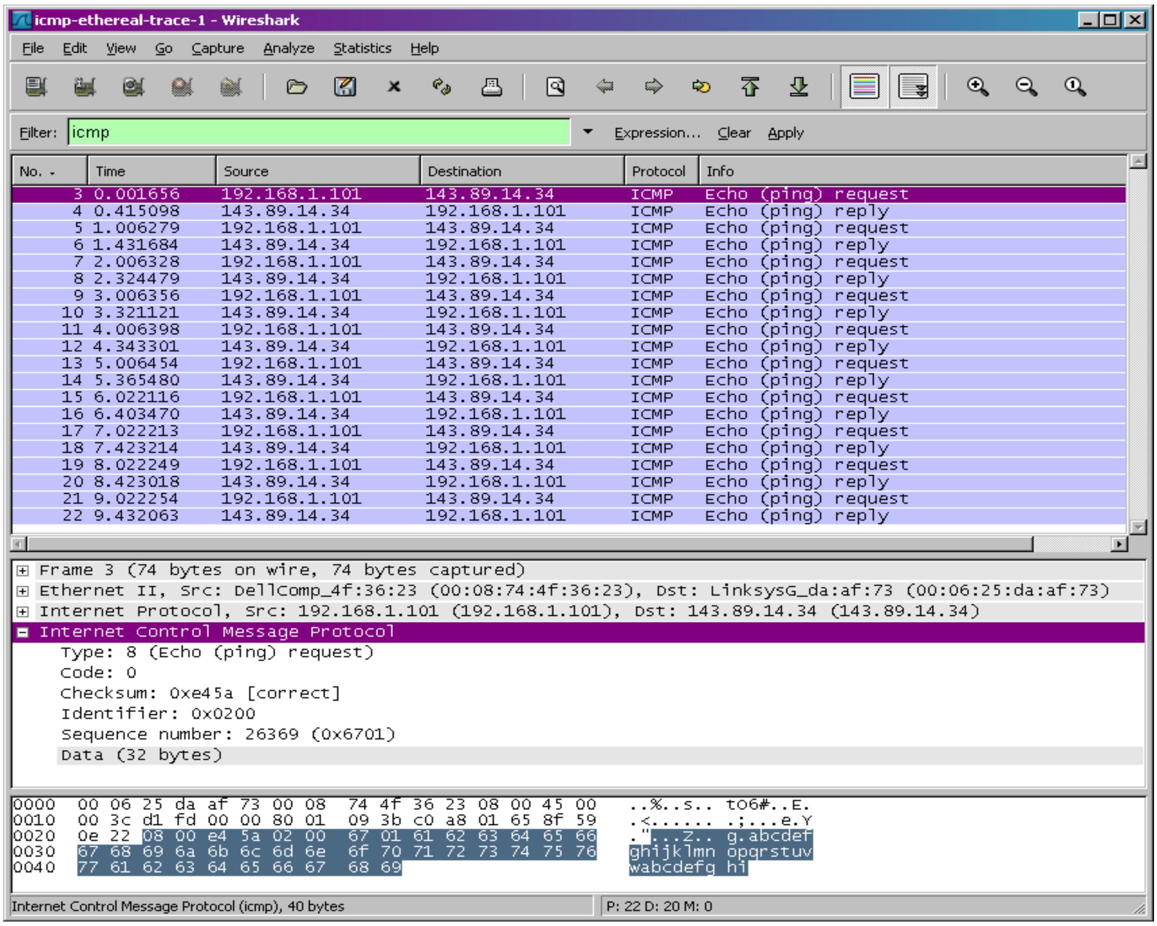
```
⊞ Frame 3 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: DellComp_4f:36:23 (00:08:74:4f:36:23), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
⊟ Internet Protocol, Src: 192.168.1.101 (192.168.1.101), Dst: 143.89.14.34 (143.89.14.34)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 60
    Identification: 0xd1fd (53757)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0x093b [correct]
    Source: 192.168.1.101 (192.168.1.101)
    Destination: 143.89.14.34 (143.89.14.34)
⊞ Internet Control Message Protocol
```

```
0010  00 3c d1 fd 00 00 80 01  09 3b c0 a8 01 65 8f 59   .<...... .;...e.Y
0020  0e 22 08 00 e4 5a 02 00  67 01 61 62 63 64 65 66   ."...Z.. g.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                     wabcdefg hi
```

Internet Control Message Protocol (icmp), 40 bytes | P: 22 D: 20 M: 0

Figure 2 Wireshark output for Ping program with Internet Protocol expanded.

Figure 3 focuses on the same ICMP but has expanded the ICMP protocol information in the packet contents window. Observe that this ICMP packet is of Type 8 and Code 0 - a so-called ICMP “echo request” packet. Also note that this ICMP packet contains a checksum, an identifier, and a sequence number.



icmp-ethereal-trace-1 - Wireshark

File Edit View Go Capture Analyze Statistics Help
Filter: icmp
Expression... Clear Apply

```
No.  Time      Source          Destination     Protocol Info
  3 0.001656  192.168.1.101   143.89.14.34    ICMP     Echo (ping) request
  4 0.415098  143.89.14.34    192.168.1.101   ICMP     Echo (ping) reply
  5 1.006279  192.168.1.101   143.89.14.34    ICMP     Echo (ping) request
  6 1.431684  143.89.14.34    192.168.1.101   ICMP     Echo (ping) reply
  7 2.006328  192.168.1.101   143.89.14.34    ICMP     Echo (ping) request
  8 2.324479  143.89.14.34    192.168.1.101   ICMP     Echo (ping) reply
  9 3.006356  192.168.1.101   143.89.14.34    ICMP     Echo (ping) request
 10 3.321121  143.89.14.34    192.168.1.101   ICMP     Echo (ping) reply
 11 4.006398  192.168.1.101   143.89.14.34    ICMP     Echo (ping) request
 12 4.343301  143.89.14.34    192.168.1.101   ICMP     Echo (ping) reply
 13 5.006454  192.168.1.101   143.89.14.34    ICMP     Echo (ping) request
 14 5.365480  143.89.14.34    192.168.1.101   ICMP     Echo (ping) reply
 15 6.022116  192.168.1.101   143.89.14.34    ICMP     Echo (ping) request
 16 6.403470  143.89.14.34    192.168.1.101   ICMP     Echo (ping) reply
 17 7.022213  192.168.1.101   143.89.14.34    ICMP     Echo (ping) request
 18 7.423214  143.89.14.34    192.168.1.101   ICMP     Echo (ping) reply
 19 8.022249  192.168.1.101   143.89.14.34    ICMP     Echo (ping) request
 20 8.423018  143.89.14.34    192.168.1.101   ICMP     Echo (ping) reply
 21 9.022254  192.168.1.101   143.89.14.34    ICMP     Echo (ping) request
 22 9.432063  143.89.14.34    192.168.1.101   ICMP     Echo (ping) reply
```

```
⊞ Frame 3 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: DellComp_4f:36:23 (00:08:74:4f:36:23), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
⊞ Internet Protocol, Src: 192.168.1.101 (192.168.1.101), Dst: 143.89.14.34 (143.89.14.34)
⊟ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xe45a [correct]
    Identifier: 0x0200
    Sequence number: 26369 (0x6701)
    Data (32 bytes)
```

```
0000   00 06 25 da af 73 00 08  74 4f 36 23 08 00 45 00   ..%..s.. tO6#..E.
0010   00 3c d1 fd 00 00 80 01  09 3b c0 a8 01 65 8f 59   .<...... .;...e.Y
0020   0e 22 08 00 e4 5a 02 00  67 01 61 62 63 64 65 66   ."...Z.. g.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                     wabcdefg hi
```

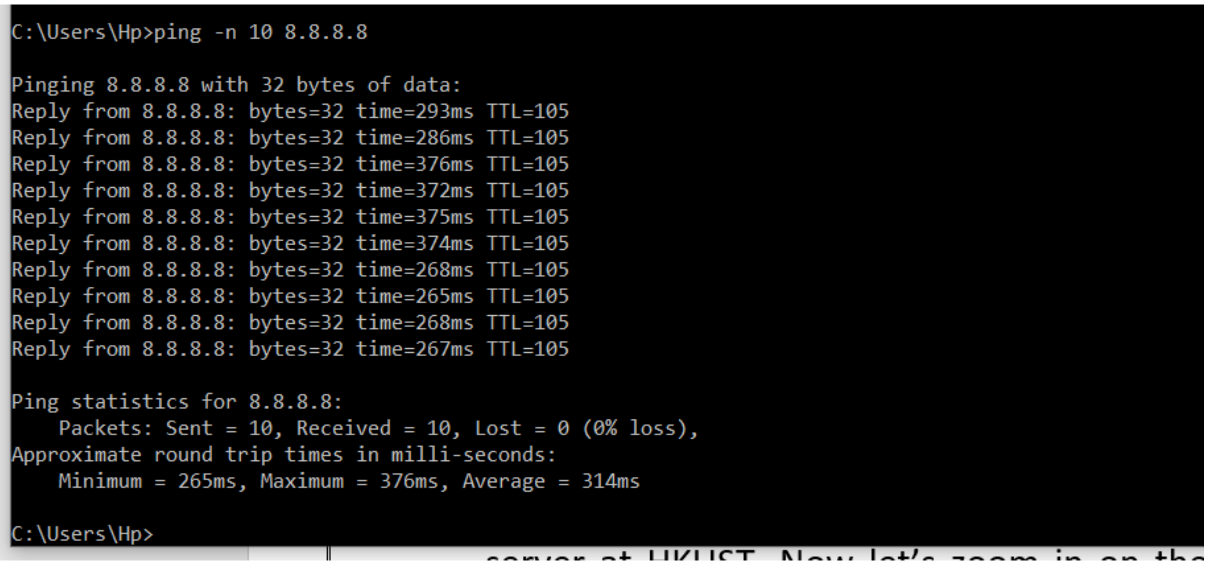
Internet Control Message Protocol (icmp), 40 bytes | P: 22 D: 20 M: 0

Figure 3: Wireshark capture of ping packet with ICMP packet expanded.

What to Hand In:

You should hand in a screen shot of the Command Prompt window similar to Figure 1 above. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked.

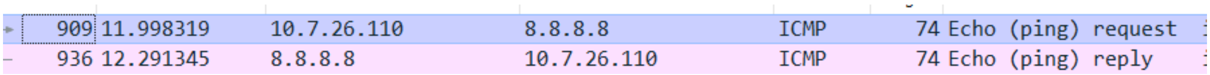
You should answer the following questions:



```
C:\Users\Hp>ping -n 10 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=293ms TTL=105
Reply from 8.8.8.8: bytes=32 time=286ms TTL=105
Reply from 8.8.8.8: bytes=32 time=376ms TTL=105
Reply from 8.8.8.8: bytes=32 time=372ms TTL=105
Reply from 8.8.8.8: bytes=32 time=375ms TTL=105
Reply from 8.8.8.8: bytes=32 time=374ms TTL=105
Reply from 8.8.8.8: bytes=32 time=268ms TTL=105
Reply from 8.8.8.8: bytes=32 time=265ms TTL=105
Reply from 8.8.8.8: bytes=32 time=268ms TTL=105
Reply from 8.8.8.8: bytes=32 time=267ms TTL=105

Ping statistics for 8.8.8.8:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 265ms, Maximum = 376ms, Average = 314ms

C:\Users\Hp>
```

1. **What is the IP address of your host? What is the IP address of the destination host?**

   Answer:



```
  909 11.998319   10.7.26.110    8.8.8.8        ICMP    74 Echo (ping) request
  936 12.291345   8.8.8.8        10.7.26.110    ICMP    74 Echo (ping) reply
```

   Host; 10.7.26.110

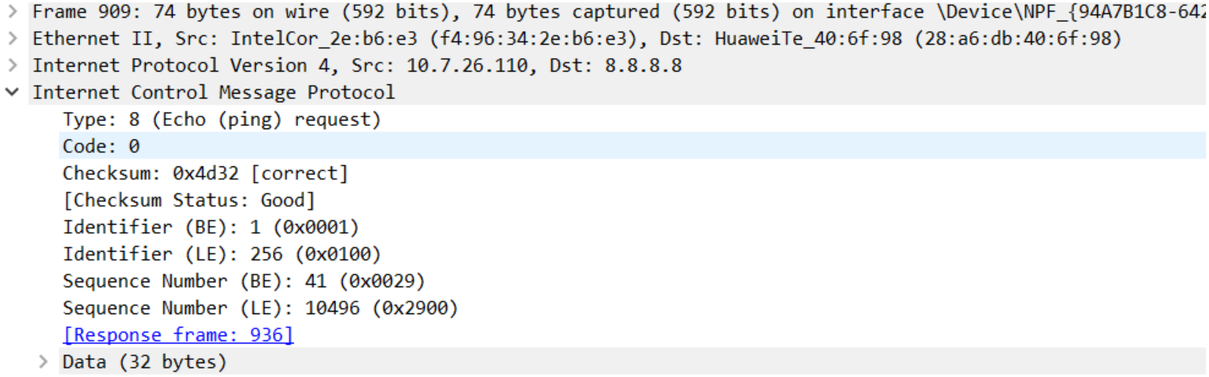
   Destination; 8.8.8.8

2. **Why is it that an ICMP packet does not have source and destination port numbers?**

   Answer:

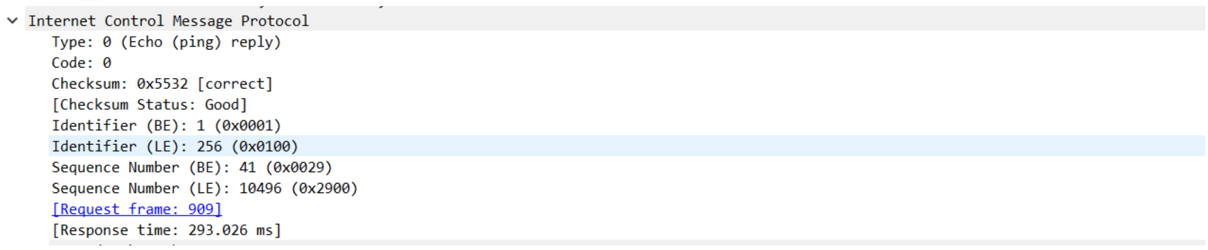
   The ICMP packet does not have source and destination port numbers because it was designed to communicate network-layer information between hosts and routers, not between application layer processes. Each ICMP packet has a "Type" and a "Code". The Type/Code combination identifies the specific message being received.

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?**



```
> Frame 909: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{94A7B1C8-642
> Ethernet II, Src: IntelCor_2e:b6:e3 (f4:96:34:2e:b6:e3), Dst: HuaweiTe_40:6f:98 (28:a6:db:40:6f:98)
> Internet Protocol Version 4, Src: 10.7.26.110, Dst: 8.8.8.8
v Internet Control Message Protocol
     Type: 8 (Echo (ping) request)
     Code: 0
     Checksum: 0x4d32 [correct]
     [Checksum Status: Good]
     Identifier (BE): 1 (0x0001)
     Identifier (LE): 256 (0x0100)
     Sequence Number (BE): 41 (0x0029)
     Sequence Number (LE): 10496 (0x2900)
     [Response frame: 936]
   > Data (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?**



```
v Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x5532 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 41 (0x0029)
    Sequence Number (LE): 10496 (0x2900)
    [Request frame: 909]
    [Response time: 293.026 ms]
```

2. ICMP and Traceroute:

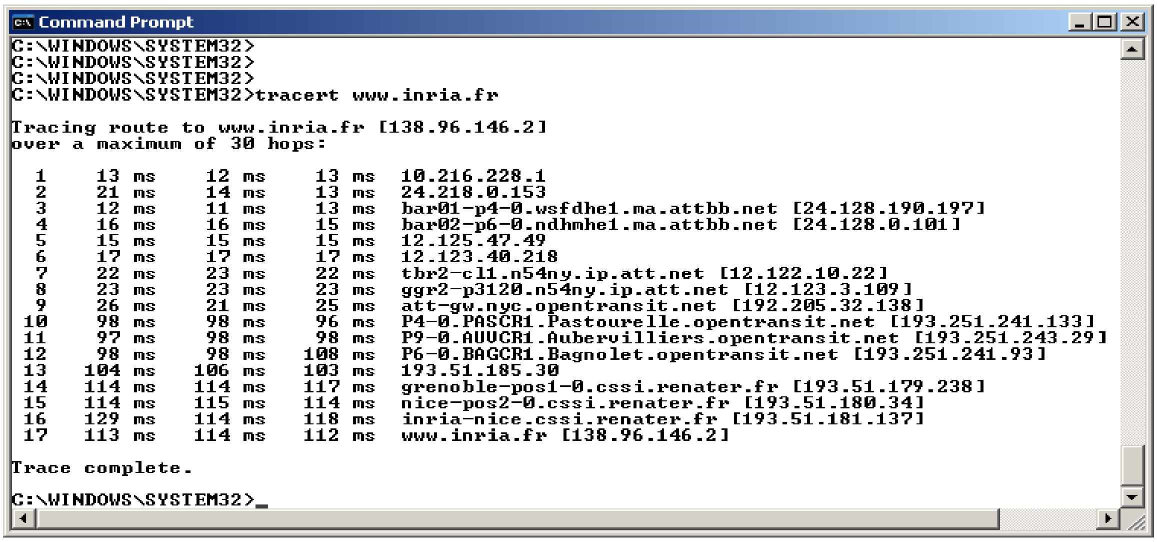
Let’s now continue our ICMP adventure by capturing the packets generated by the Traceroute program. You may recall that the Traceroute program can be used to figure out the path a packet takes from source to destination.

Traceroute is implemented in different ways in Unix/Linux/MacOS and in Windows. In Unix/Linux, the source sends a series of UDP packets to the target destination using an unlikely destination port number; in Windows, the source sends a series of ICMP packets to the target destination. For both operating systems, the program sends the first packet with TTL=1, the second packet with TTL=2, and so on. Recall that a router will decrement a packet’s TTL value as the packet passes through the router. When a packet arrives at a router with TTL=1, the router sends an ICMP error packet back to the source. In the following, we’ll use the native Windows *tracert* program.

Do the following:

- Let’s begin by opening the Windows Command Prompt application (which can be found in your Accessories folder).
- Start up the Wireshark packet sniffer, and begin Wireshark packet capture.
- The *tracert* command is in c:\windows\system32, so type either “*tracert hostname*” or “*c:\windows\system32\tracert hostname*” in the MS-DOS command line (without quotation marks), where hostname is a host on another continent. (Note that on a Windows machine, the command is ***“tracert”*** and not ***“traceroute”***.) If you’re outside of Europe, you may want to enter www.inria.fr for the Web server at INRIA, a computer science research institute in France. Then run the Traceroute program by typing return.
- When the Traceroute program terminates, stop packet capture in Wireshark.

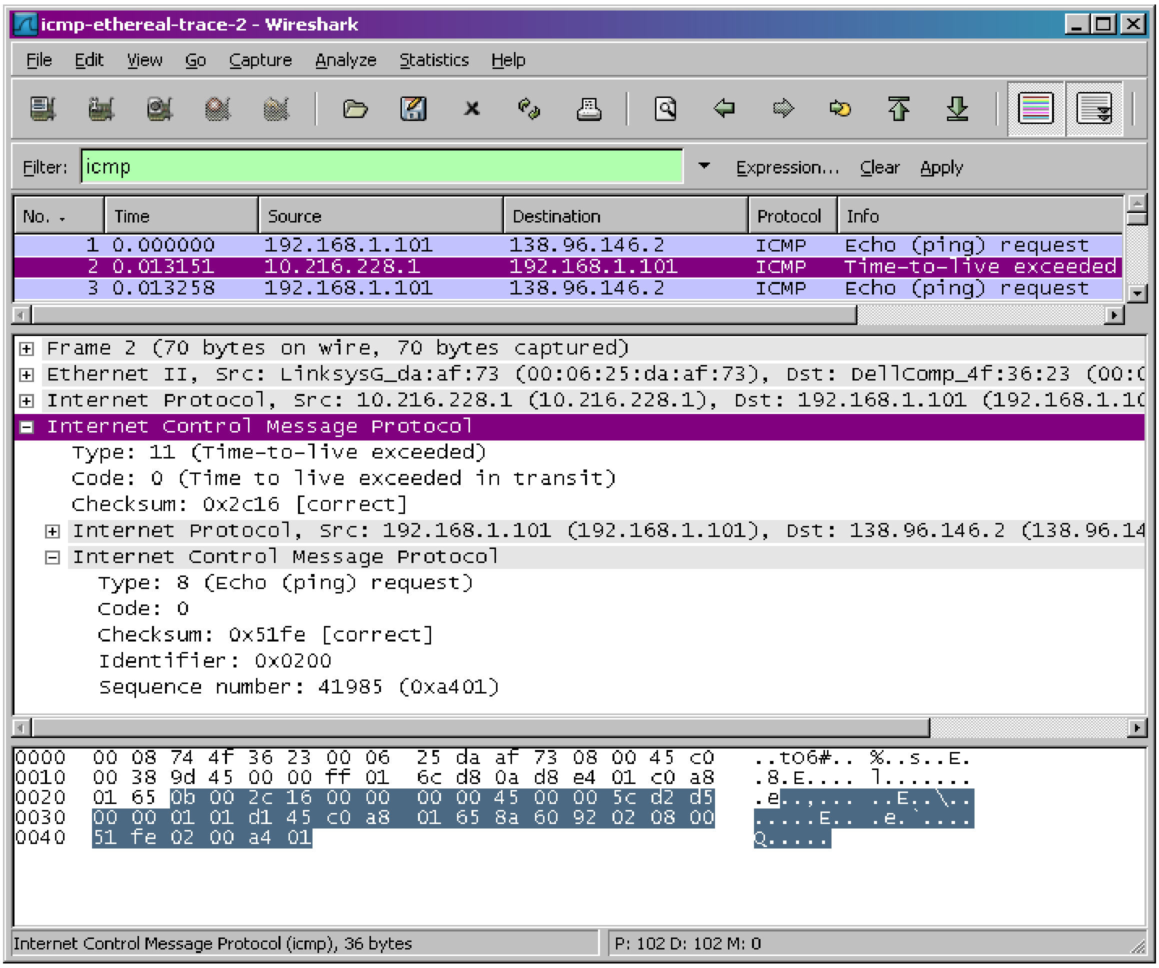
At the end of the experiment, your Command Prompt Window should look something like Figure 4. From this figure we see that for each TTL value, the source program sends three probe packets. Traceroute displays the RTTs for each of the probe packets, as well as the IP address (and possibly the name) of the router that returned the ICMP TTL-exceeded message.



```
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:

  1    13 ms    12 ms    13 ms  10.216.228.1
  2    21 ms    14 ms    13 ms  24.218.0.153
  3    12 ms    11 ms    13 ms  bar01-p4-0.wsfdhe1.ma.attbb.net [24.128.190.197]
  4    16 ms    16 ms    15 ms  bar02-p6-0.ndhmhe1.ma.attbb.net [24.128.0.101]
  5    15 ms    15 ms    15 ms  12.125.47.49
  6    17 ms    17 ms    17 ms  12.123.40.218
  7    22 ms    23 ms    22 ms  tbr2-cl1.n54ny.ip.att.net [12.122.10.22]
  8    23 ms    23 ms    23 ms  ggr2-p3120.n54ny.ip.att.net [12.123.3.109]
  9    26 ms    21 ms    25 ms  att-gw.nyc.opentransit.net [192.205.32.138]
 10    98 ms    98 ms    96 ms  P4-0.PASCR1.Pastourelle.opentransit.net [193.251.241.133]
 11    97 ms    98 ms    98 ms  P9-0.AUVCR1.Aubervilliers.opentransit.net [193.251.243.29]
 12    98 ms    98 ms   108 ms  P6-0.BAGCR1.Bagnolet.opentransit.net [193.251.241.93]
 13   104 ms   106 ms   103 ms  193.51.185.30
 14   114 ms   114 ms   117 ms  grenoble-pos1-0.cssi.renater.fr [193.51.179.238]
 15   114 ms   115 ms   114 ms  nice-pos2-0.cssi.renater.fr [193.51.180.34]
 16   129 ms   114 ms   118 ms  inria-nice.cssi.renater.fr [193.51.181.137]
 17   113 ms   114 ms   112 ms  www.inria.fr [138.96.146.2]

Trace complete.

C:\WINDOWS\SYSTEM32>_
```

Figure 4: Command Prompt window displays the results of the Traceroute program.

Figure 5 displays the Wireshark window for an ICMP packet returned by a router. Note that this ICMP error packet contains many more fields than the Ping ICMP messages.



Filter: icmp

```
No. .  Time       Source          Destination     Protocol  Info
  1 0.000000   192.168.1.101   138.96.146.2    ICMP      Echo (ping) request
  2 0.013151   10.216.228.1    192.168.1.101   ICMP      Time-to-live exceeded
  3 0.013258   192.168.1.101   138.96.146.2    ICMP      Echo (ping) request
```

```
⊞ Frame 2 (70 bytes on wire, 70 bytes captured)
⊞ Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: DellComp_4f:36:23 (00:0
⊞ Internet Protocol, Src: 10.216.228.1 (10.216.228.1), Dst: 192.168.1.101 (192.168.1.10
⊟ Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0x2c16 [correct]
  ⊞ Internet Protocol, Src: 192.168.1.101 (192.168.1.101), Dst: 138.96.146.2 (138.96.14
  ⊟ Internet Control Message Protocol
      Type: 8 (Echo (ping) request)
      Code: 0
      Checksum: 0x51fe [correct]
      Identifier: 0x0200
      Sequence number: 41985 (0xa401)
```

```
0000  00 08 74 4f 36 23 00 06  25 da af 73 08 00 45 c0   ..tO6#.. %..s..E.
0010  00 38 9d 45 00 00 ff 01  6c d8 0a d8 e4 01 c0 a8   .8.E.... l.......
0020  01 65 0b 00 2c 16 00 00  00 00 45 00 00 5c d2 d5   .e..,... ..E..\..
0030  00 00 01 01 d1 45 c0 a8  01 65 8a 60 92 02 08 00   .....E.. .e.`....
0040  51 fe 02 00 a4 01                                  Q.....
```

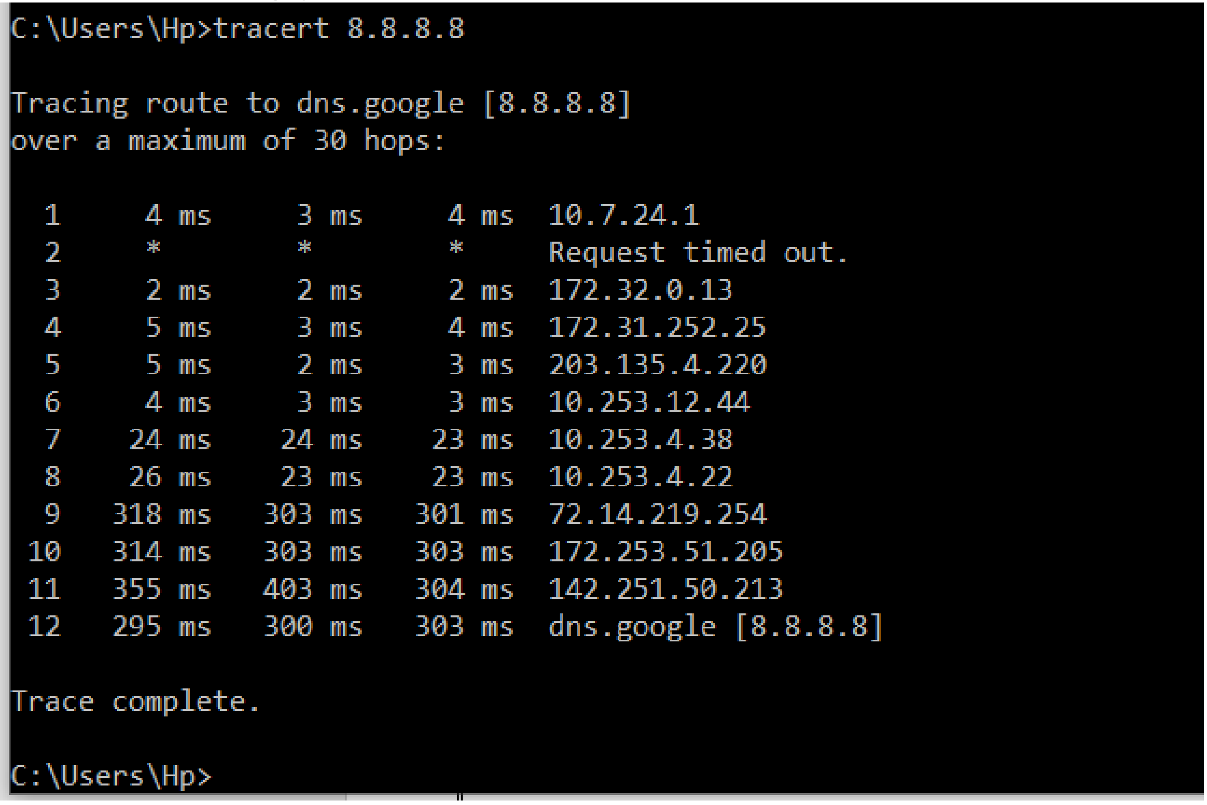
Internet Control Message Protocol (icmp), 36 bytes   P: 102 D: 102 M: 0

Figure 5: Wireshark window of ICMP fields expanded for one ICMP error packet.

What to Hand In:

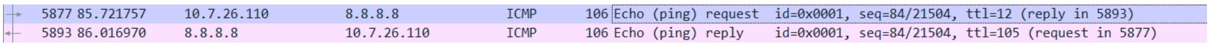
For this part of the lab, you should hand in a screen shot of the Command Prompt window. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked.

Answer the following questions:



```
C:\Users\Hp>tracert 8.8.8.8

Tracing route to dns.google [8.8.8.8]
over a maximum of 30 hops:

  1     4 ms     3 ms     4 ms  10.7.24.1
  2     *        *        *     Request timed out.
  3     2 ms     2 ms     2 ms  172.32.0.13
  4     5 ms     3 ms     4 ms  172.31.252.25
  5     5 ms     2 ms     3 ms  203.135.4.220
  6     4 ms     3 ms     3 ms  10.253.12.44
  7    24 ms    24 ms    23 ms  10.253.4.38
  8    26 ms    23 ms    23 ms  10.253.4.22
  9   318 ms   303 ms   301 ms  72.14.219.254
 10   314 ms   303 ms   303 ms  172.253.51.205
 11   355 ms   403 ms   304 ms  142.251.50.213
 12   295 ms   300 ms   303 ms  dns.google [8.8.8.8]

Trace complete.

C:\Users\Hp>
```

1. **What is the IP address of your host? What is the IP address of the target destination host?**



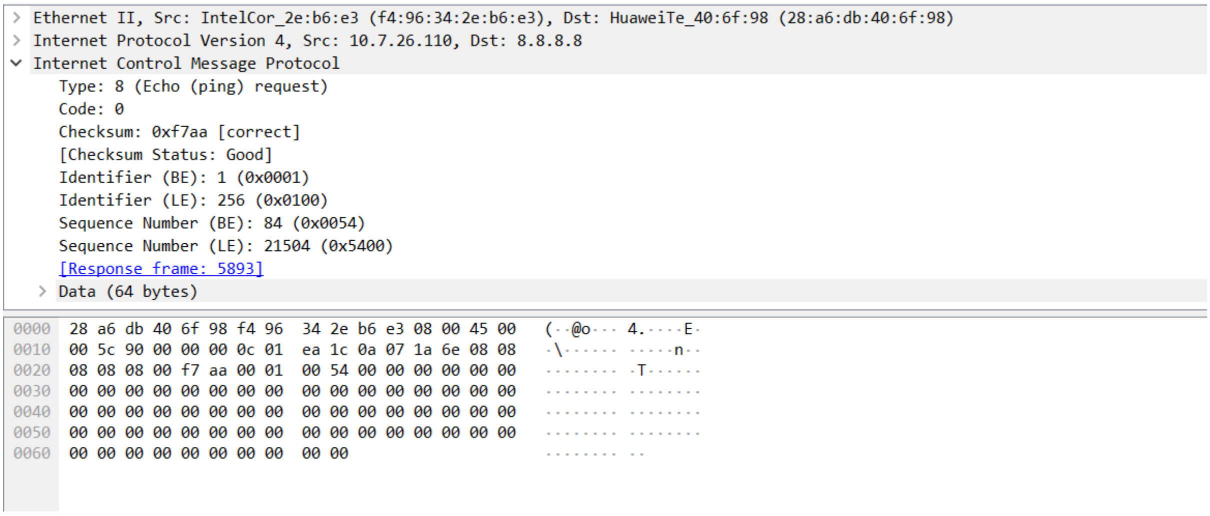
```
5877 85.721757   10.7.26.110   8.8.8.8       ICMP   106 Echo (ping) request  id=0x0001, seq=84/21504, ttl=12 (reply in 5893)
5893 86.016970   8.8.8.8       10.7.26.110   ICMP   106 Echo (ping) reply    id=0x0001, seq=84/21504, ttl=105 (request in 5877)
```

2. **If ICMP sent UDP packets instead (as in Unix/Linux), would the IP protocol number still be 0x01 for the probe packets? If not, what would it be?**

   Answer:

   It would be 0x11, the assigned protocol number for UDP.

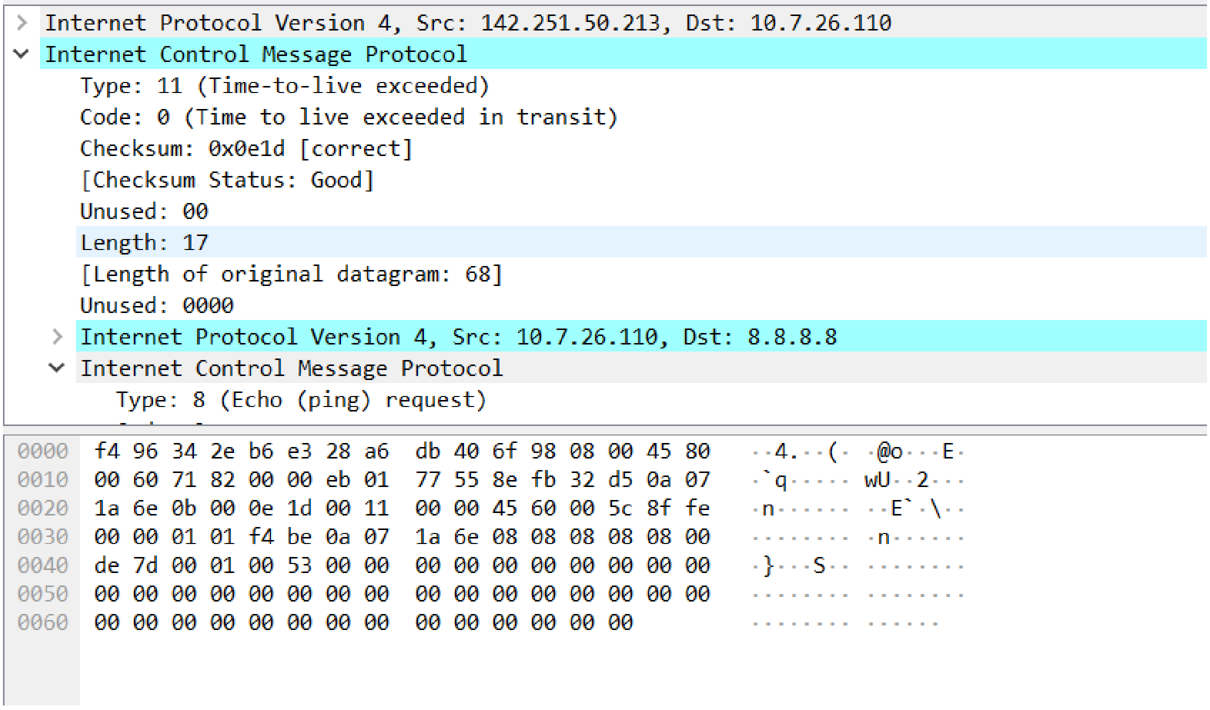
3. **Examine the ICMP echo packet in your screenshot. Is this different from the ICMP ping query packets in the first half of this lab? If yes, how so?**



```
Ethernet II, Src: IntelCor_2e:b6:e3 (f4:96:34:2e:b6:e3), Dst: HuaweiTe_40:6f:98 (28:a6:db:40:6f:98)
Internet Protocol Version 4, Src: 10.7.26.110, Dst: 8.8.8.8
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xf7aa [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 84 (0x0054)
    Sequence Number (LE): 21504 (0x5400)
    [Response frame: 5893]
    Data (64 bytes)

0000  28 a6 db 40 6f 98 f4 96  34 2e b6 e3 08 00 45 00   (··@o··· 4.····E·
0010  00 5c 90 00 00 00 0c 01  ea 1c 0a 07 1a 6e 08 08   ·\······ ·····n··
0020  08 08 08 00 f7 aa 00 01  00 54 00 00 00 00 00 00   ········ ·T······
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00                     ········ ··
```

The ICMP echo packet has the same fields as the ping query packets

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



```
Internet Protocol Version 4, Src: 142.251.50.213, Dst: 10.7.26.110
Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0x0e1d [correct]
    [Checksum Status: Good]
    Unused: 00
    Length: 17
    [Length of original datagram: 68]
    Unused: 0000
    Internet Protocol Version 4, Src: 10.7.26.110, Dst: 8.8.8.8
    Internet Control Message Protocol
        Type: 8 (Echo (ping) request)

0000  f4 96 34 2e b6 e3 28 a6  db 40 6f 98 08 00 45 80   ··4.··(· ·@o···E·
0010  00 60 71 82 00 00 eb 01  77 55 8e fb 32 d5 0a 07   ·`q····· wU··2···
0020  1a 6e 0b 00 0e 1d 00 11  00 00 45 60 00 5c 8f fe   ·n······ ··E`·\··
0030  00 00 01 01 f4 be 0a 07  1a 6e 08 08 08 08 08 00   ········ ·n······
0040  de 7d 00 01 00 53 00 00  00 00 00 00 00 00 00 00   ·}···S·· ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00         ········ ······
```

The ICMP error packet is not the same as the ping query packets. It contains both the IP header and the first 8 bytes of the original ICMP packet that the error is for.

5. **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:

The last three ICMP packets are message type 0 (echo reply) rather than 11 (TTL expired). They are different because the datagrams have made it all the way to the destination host before the TTL expired.

6. **Explain how ICMP helps in working of traceroute application?**

Answer:

Traceroute uses ICMP's Ping command to find out how many different devices are between the computer initiating the traceroute and the target. This command works by manipulating the packets time to live value or TTL.

Some useful tutorials:

https://www.youtube.com/watch?v=glPuwhMNQ2s&ab_channel=internet-class
https://www.youtube.com/watch?v=G05y9UKT69s&ab_channel=internet-class
https://www.youtube.com/watch?v=AGUrTwIX7b8&ab_channel=internet-class
https://searchnetworking.techtarget.com/definition/big-endian-and-little-endian

Conclusion: