CMPE 472 - Lab 5: ICMP
Bensu Şeker 2251984658

In this lab, we'll explore several aspects of the ICMP protocol:

- ICMP messages generated by the Ping program,
- ICMP messages generated by the Traceroute program,
- the format and contents of an ICMP message.

Useful notes:

- You may review the ICMP material in section 5.6 of the text.

- This lab is presented in the context of the Microsoft Windows operating system. However, it is straightforward to translate the lab to a Unix or Linux environment.

1. 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 the hostname is a host on another continent. 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. From this window, we see that the source ping program sent 10 query packets and received 10 responses. Also note that for each response, the source calculates the round-trip time (RTT), which for the 10 packets is on average 375 msec.

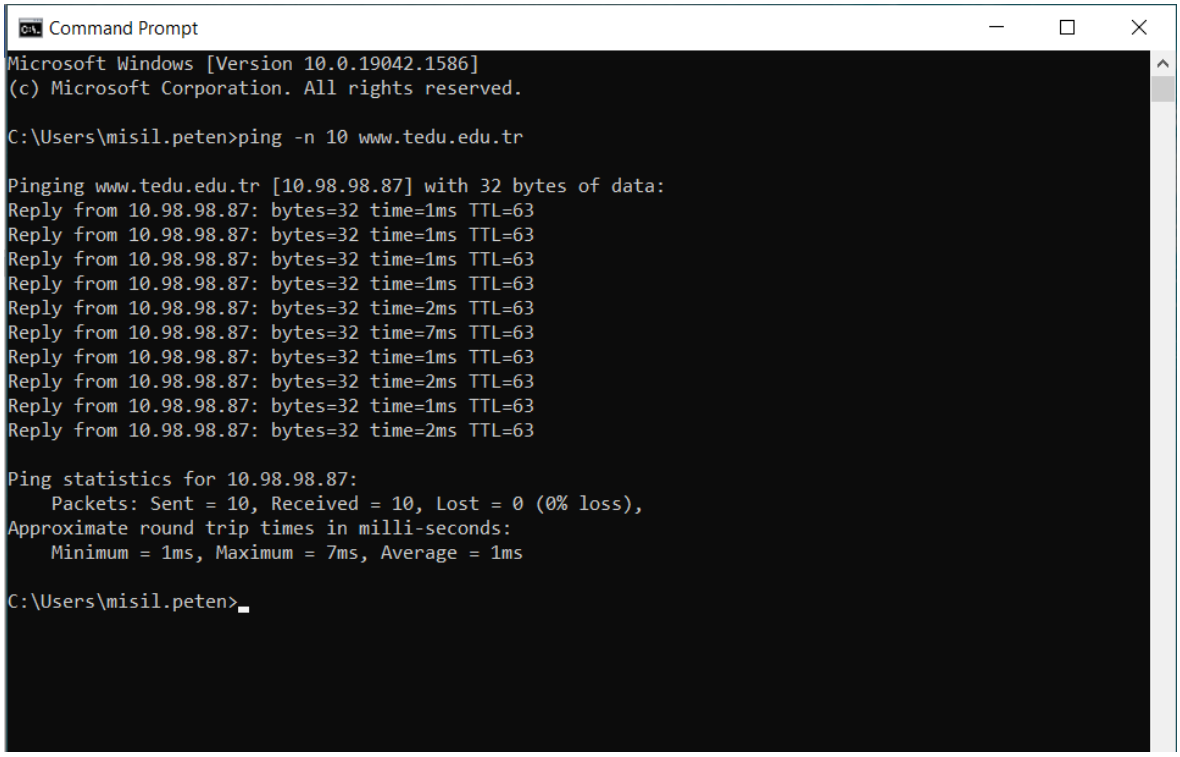


Figure 1 Command Prompt window after entering Ping command.

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. 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.

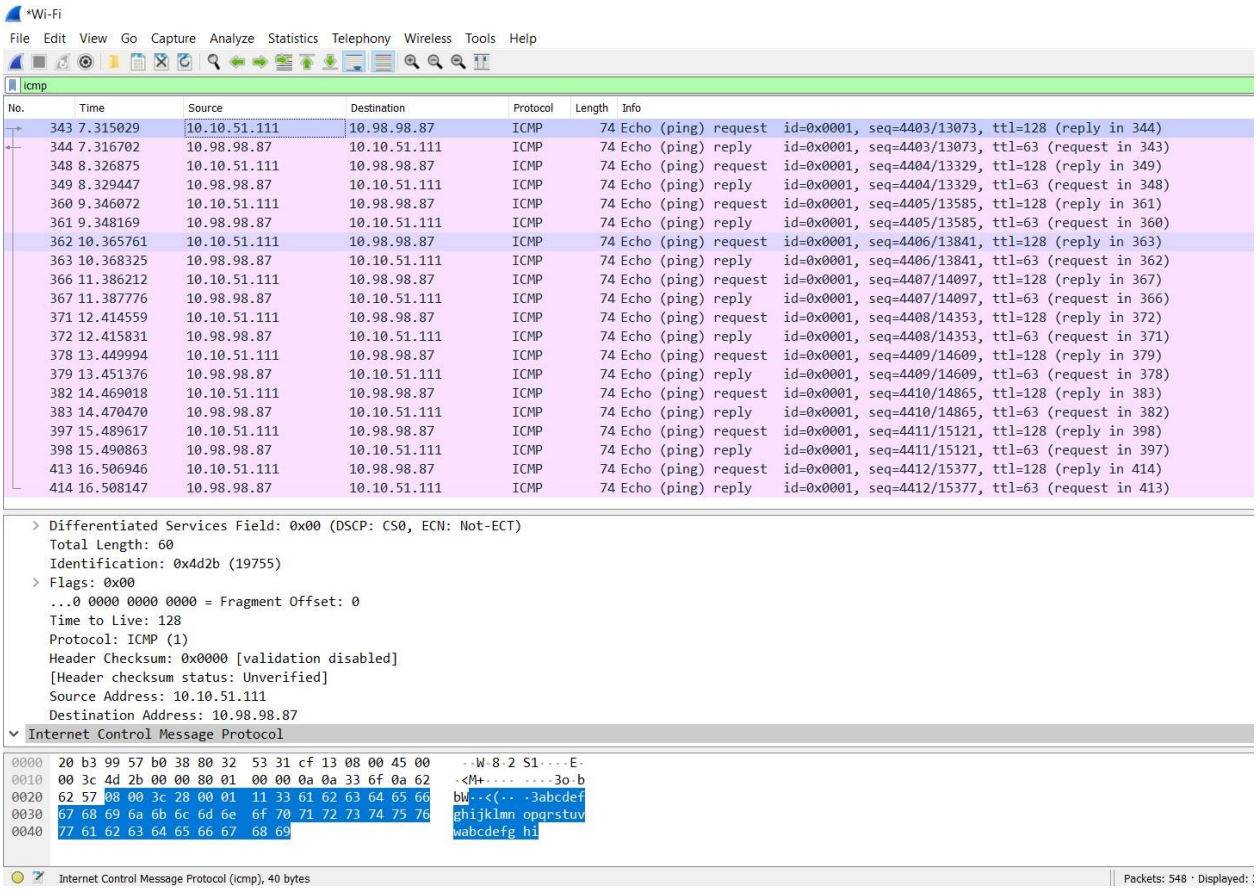


Figure 2 Wireshark output for Ping program with Internet Protocol expanded.

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. (See Figure 5.19 of text.) Also note that this ICMP packet contains a checksum, an identifier, and a sequence number. (Identifier and sequence numbers are given in both big-endian and little-endian formats.)

[1]This lab is adapted from 7th edition of our text, *Computer Networks, A Top-down Approach, 7th ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2016.*

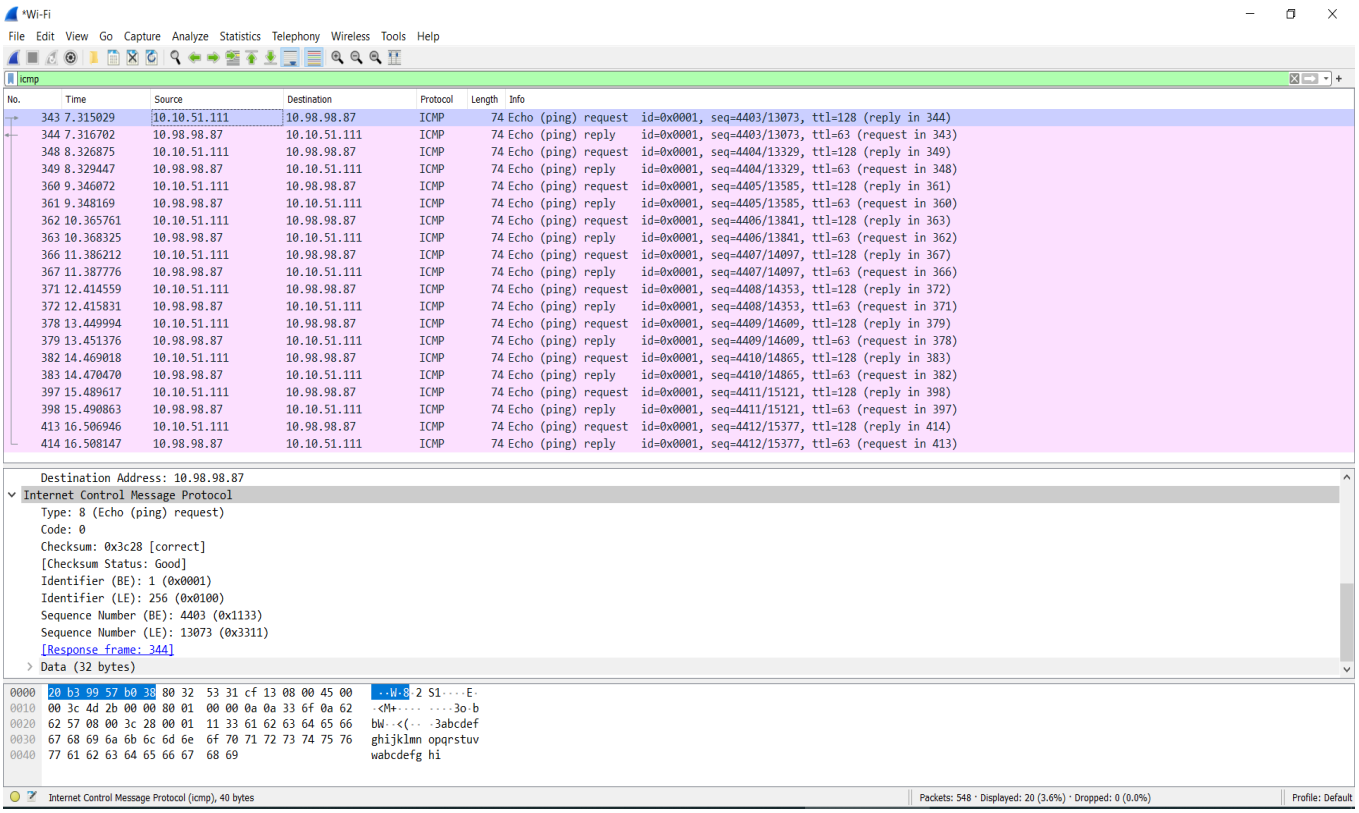


Figure 3 Wireshark capture of ping packet with ICMP packet expanded.

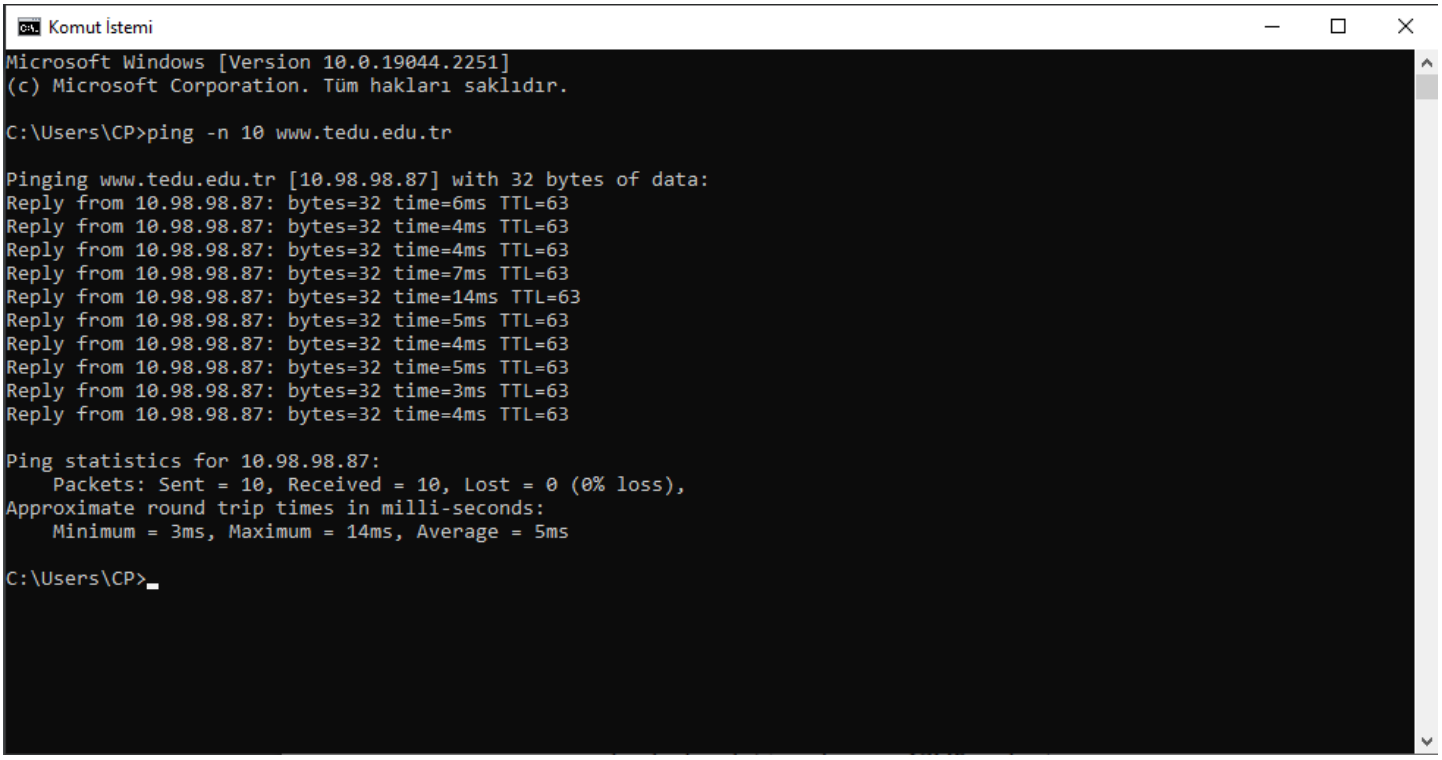
What to Hand In:

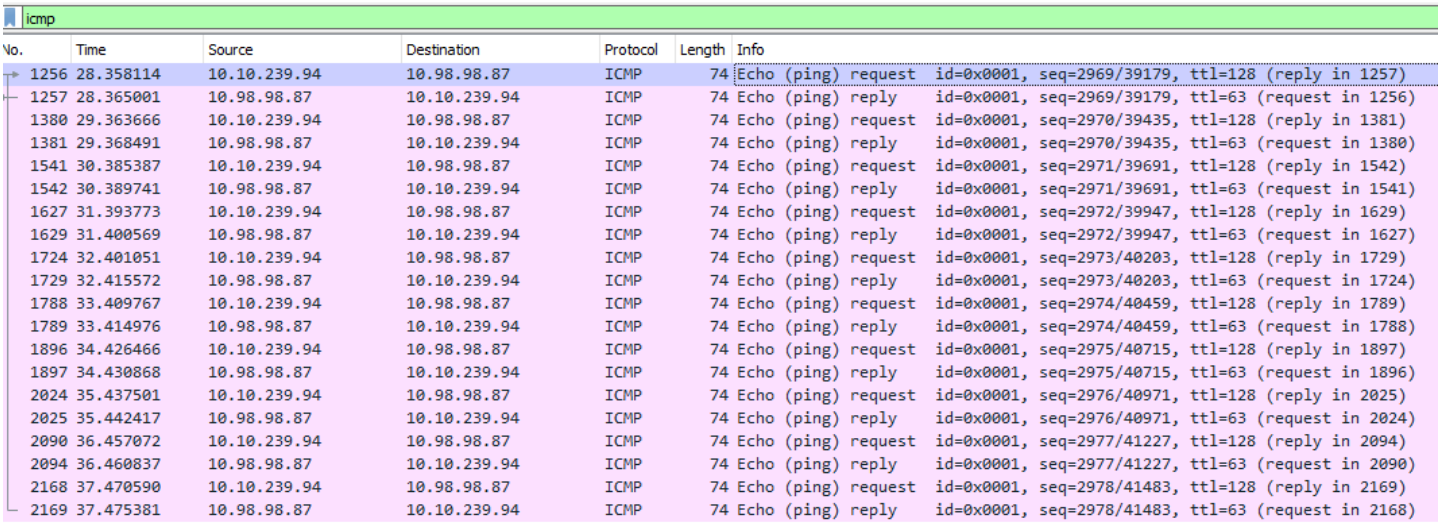
You should hand in a screenshot of the Command Prompt window similar to Figure 1 above. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout[2] to explain your answer. To print a packet, use *File->Print*, choose *Selected packet only*, choose *Packet summary line*, and select the minimum amount of packet detail that you need to answer the question.

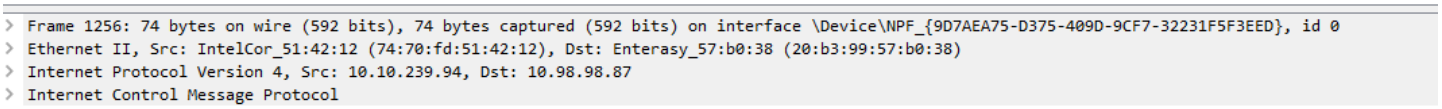
You should answer the following questions:

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

   Src: 10.10.239.94
   Dst: 10.98.98.57







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

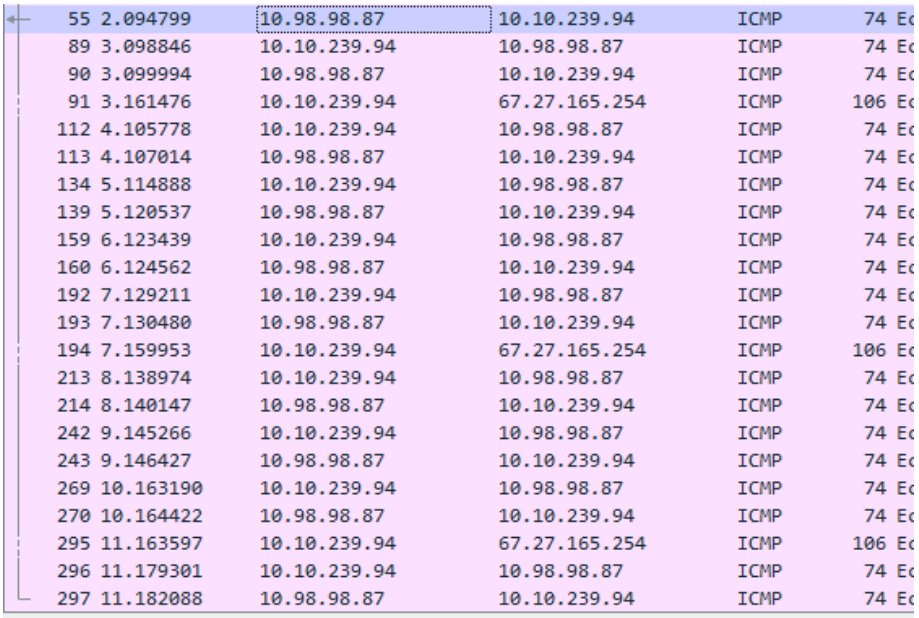
It would have source and destination port numbers if this were an application layer. These are not required because the network layer is being used.

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?

```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x41c2 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 2969 (0x0b99)
    Sequence Number (LE): 39179 (0x990b)
    [Response frame: 1257]
```

Type:8
Code: 0

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?



55	2.094799	10.98.98.87	10.10.239.94	ICMP	74 Ec
89	3.098846	10.10.239.94	10.98.98.87	ICMP	74 Ec
90	3.099994	10.98.98.87	10.10.239.94	ICMP	74 Ec
91	3.161476	10.10.239.94	67.27.165.254	ICMP	106 Ec
112	4.105778	10.10.239.94	10.98.98.87	ICMP	74 Ec
113	4.107014	10.98.98.87	10.10.239.94	ICMP	74 Ec
134	5.114888	10.10.239.94	10.98.98.87	ICMP	74 Ec
139	5.120537	10.98.98.87	10.10.239.94	ICMP	74 Ec
159	6.123439	10.10.239.94	10.98.98.87	ICMP	74 Ec
160	6.124562	10.98.98.87	10.10.239.94	ICMP	74 Ec
192	7.129211	10.10.239.94	10.98.98.87	ICMP	74 Ec
193	7.130480	10.98.98.87	10.10.239.94	ICMP	74 Ec
194	7.159953	10.10.239.94	67.27.165.254	ICMP	106 Ec
213	8.138974	10.10.239.94	10.98.98.87	ICMP	74 Ec
214	8.140147	10.98.98.87	10.10.239.94	ICMP	74 Ec
242	9.145266	10.10.239.94	10.98.98.87	ICMP	74 Ec
243	9.146427	10.98.98.87	10.10.239.94	ICMP	74 Ec
269	10.163190	10.10.239.94	10.98.98.87	ICMP	74 Ec
270	10.164422	10.98.98.87	10.10.239.94	ICMP	74 Ec
295	11.163597	10.10.239.94	67.27.165.254	ICMP	106 Ec
296	11.179301	10.10.239.94	10.98.98.87	ICMP	74 Ec
297	11.182088	10.98.98.87	10.10.239.94	ICMP	74 Ec

```
Frame 55: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interf
Ethernet II, Src: Enterasy_57:b0:38 (20:b3:99:57:b0:38), Dst: IntelCor_51:42:
Internet Protocol Version 4, Src: 10.98.98.87, Dst: 10.10.239.94
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x496c [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 3055 (0x0bef)
    Sequence Number (LE): 61195 (0xef0b)
    [Request frame: 54]
    [Response time: 6,603 ms]
    Data (32 bytes)
```

The code number and ICMP type are both 0. The ICMP packet also includes data, checksum, identification, and sequence number fields. The fields for the checksum, sequence number, and identification are each two bytes long.

[2]What do we mean by "annotate"? Please highlight where in the printout you've found the answer and add some text (preferably with a colored pen) noting what you found in what you 've highlighted.

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 discussed in Section 1.4 and in Section 5.6 of the text.

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*".)
- 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 two 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.

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.

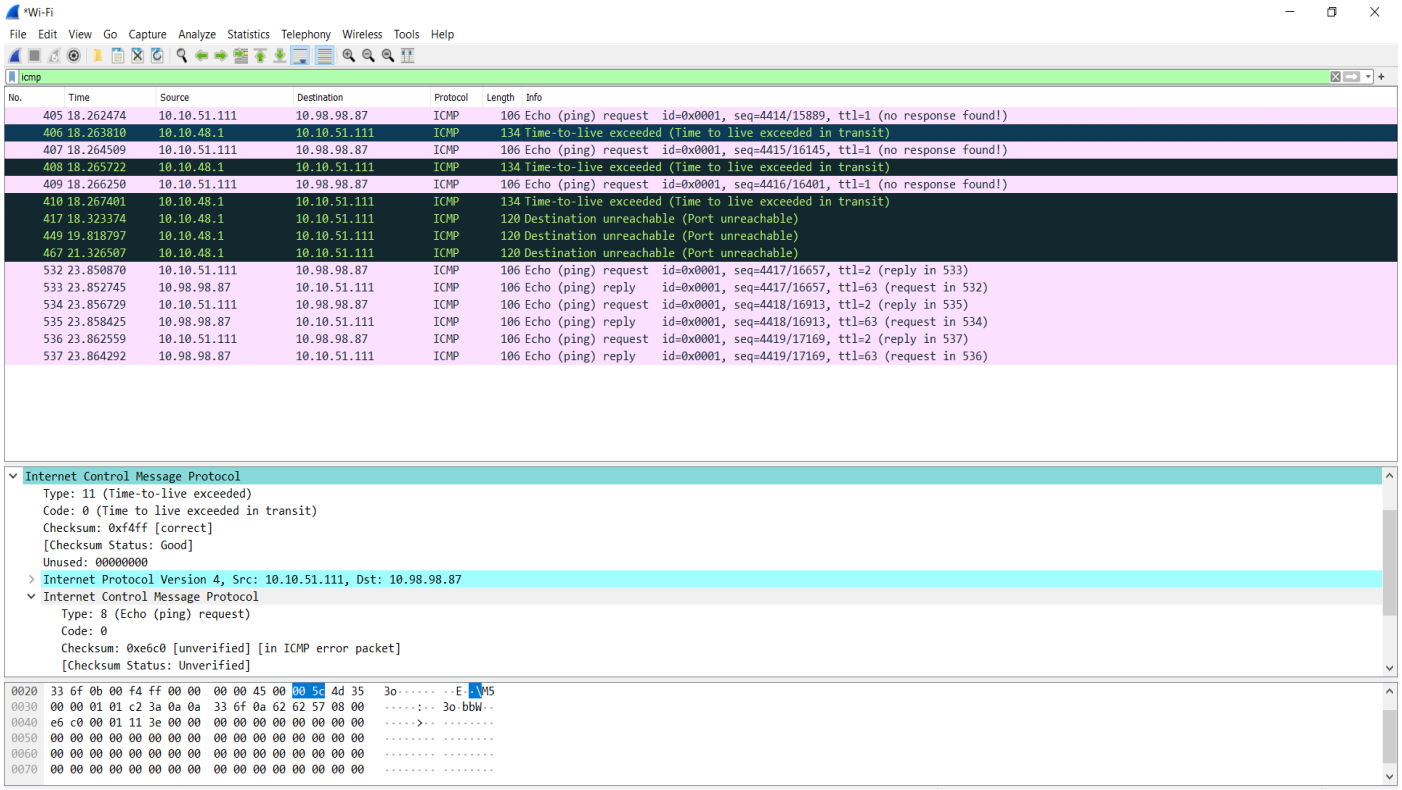
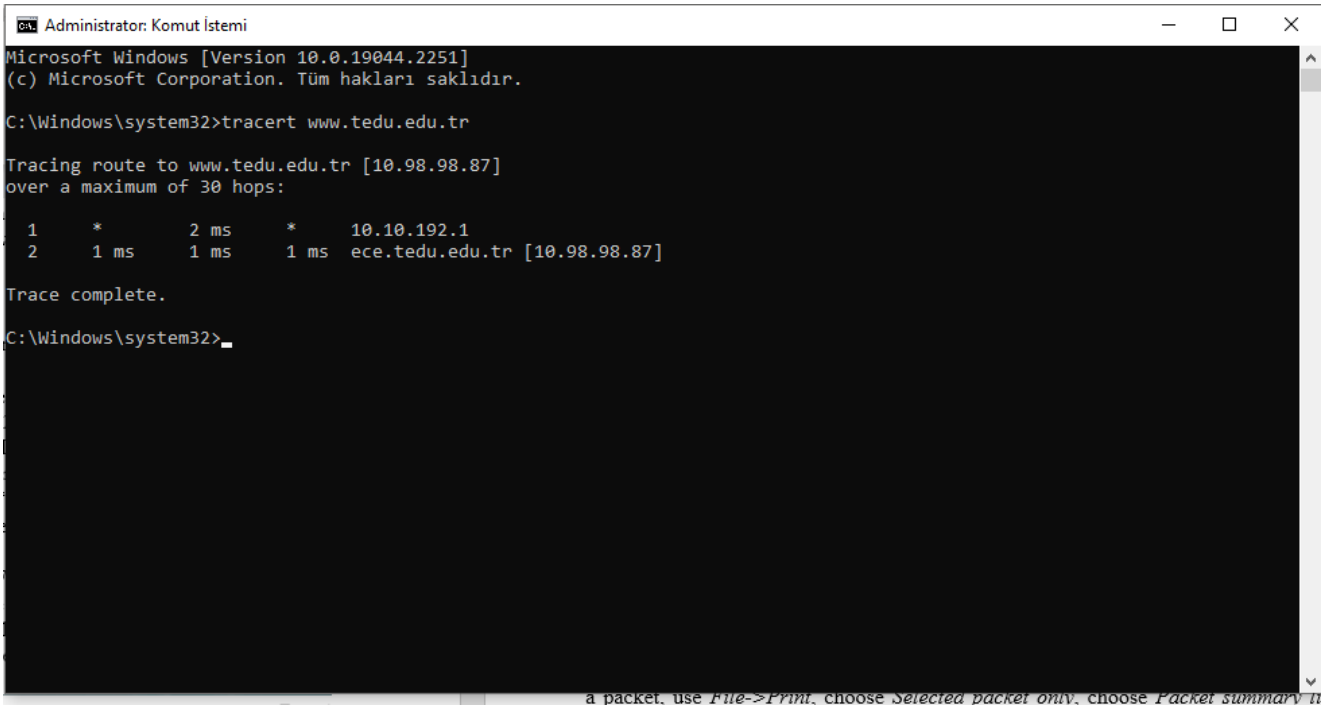


Figure 5 Wireshark window of ICMP fields expanded for one ICMP error packet.

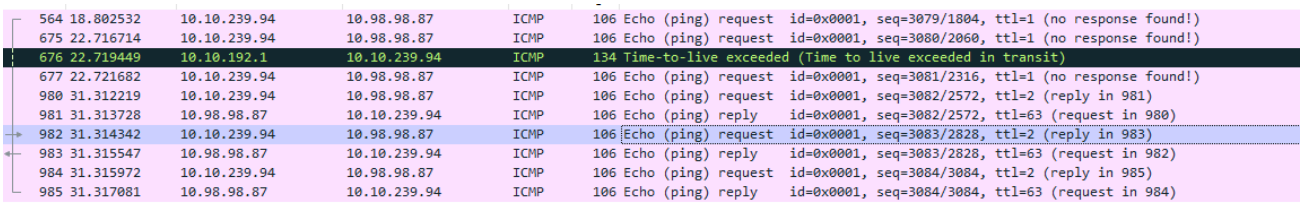
What to Hand In:

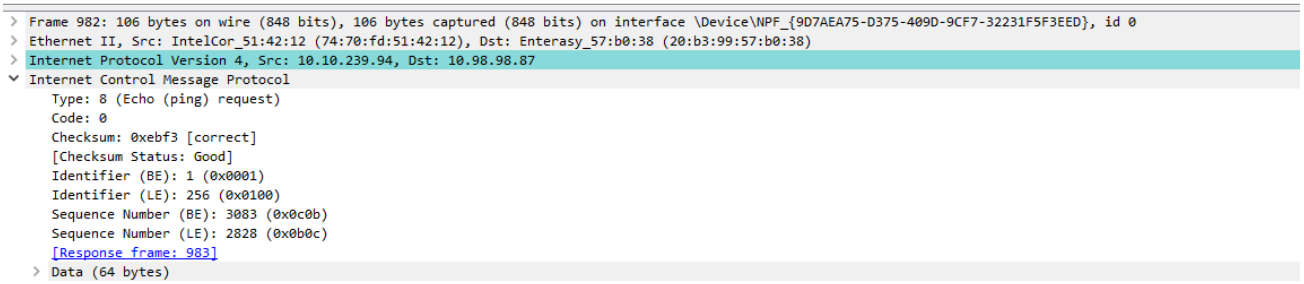
For this part of the lab, you should hand in a screenshot of the Command Prompt window. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout to explain your answer. To print a packet, use *File->Print*, choose *Selected packet only*, choose *Packet summary line*, and select the minimum amount of packet detail that you need to answer the question.

Answer the following questions:



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



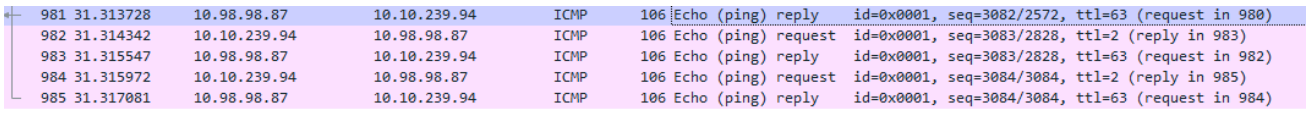


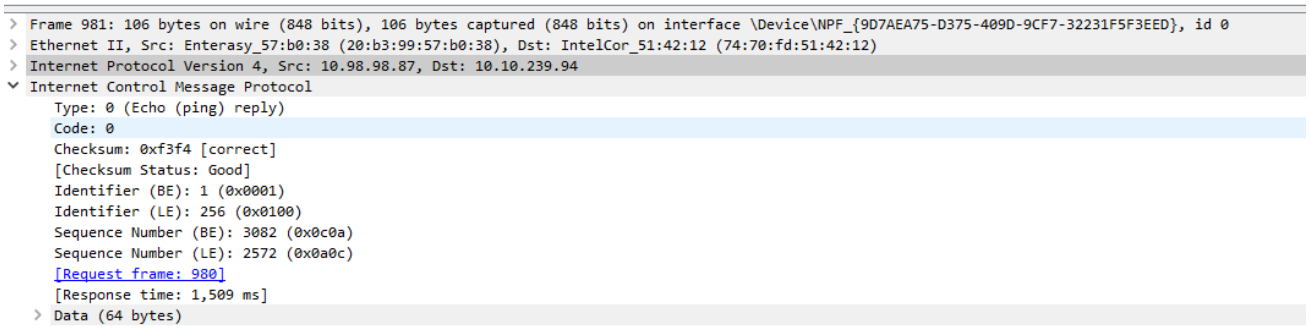
Src: 10.10.239.94
Dst: 10.98.98.87

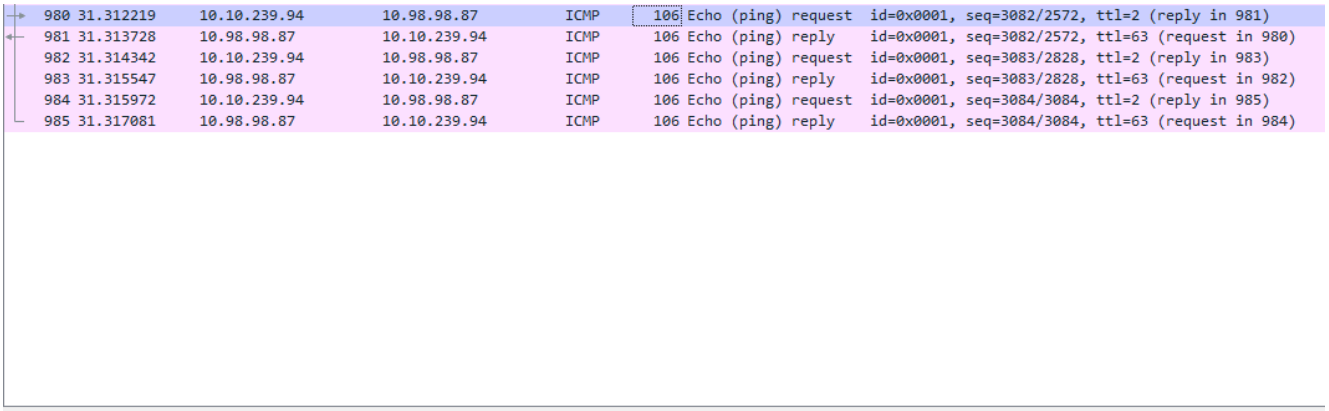
6. If ICMP sent UDP packets instead (as in Unix/Linux), would the IP protocol number still be 01 for the probe packets? If not, what would it be?

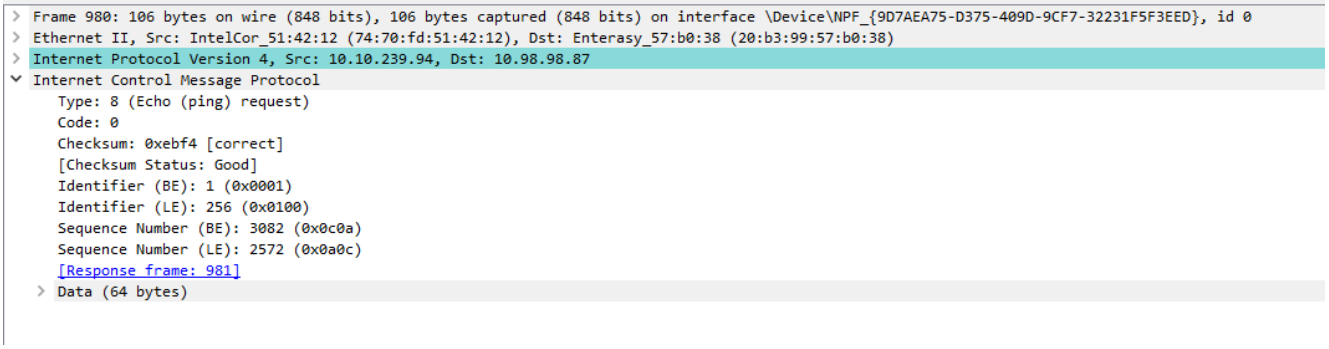
The situation would change if ICMP sent UDP packets. It would be changed from 01 to 0 X 11.

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?



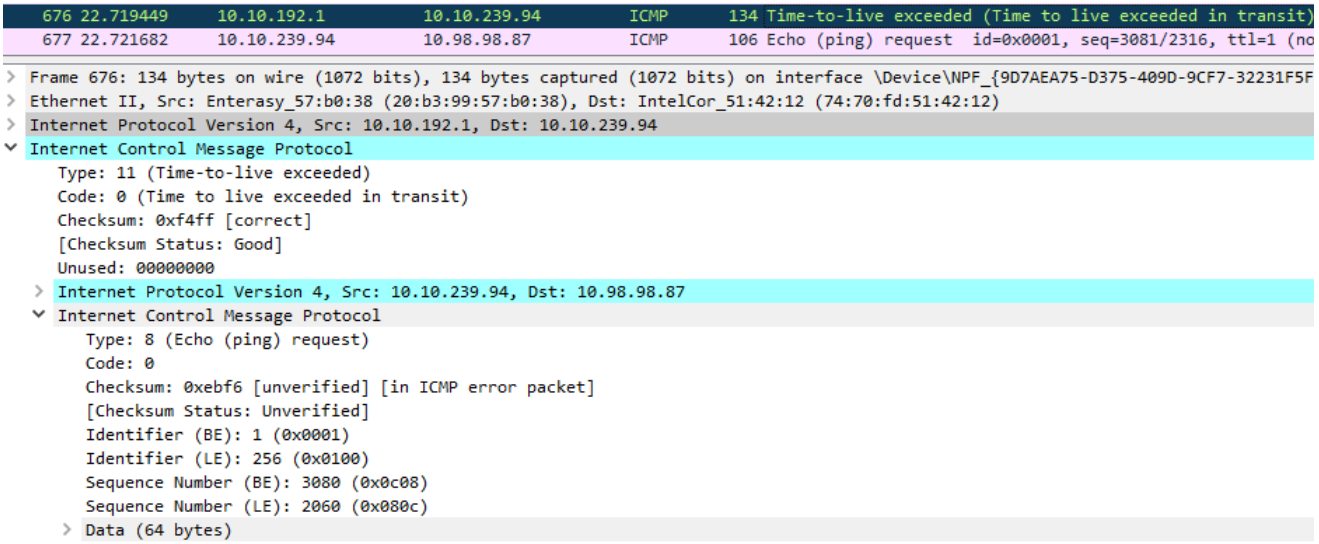






The fields in ICMP echo packets are identical as those in ping query packets.

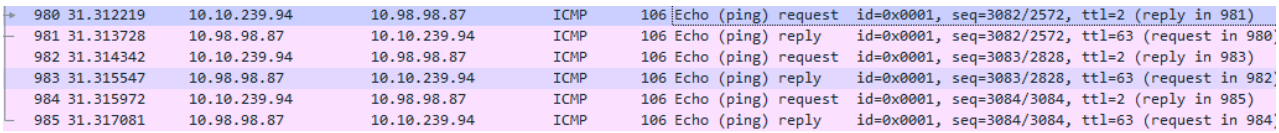
8. Examine the ICMP error packet in your screenshot. It has more fields than the ICMP echo packet. What is included in those fields?

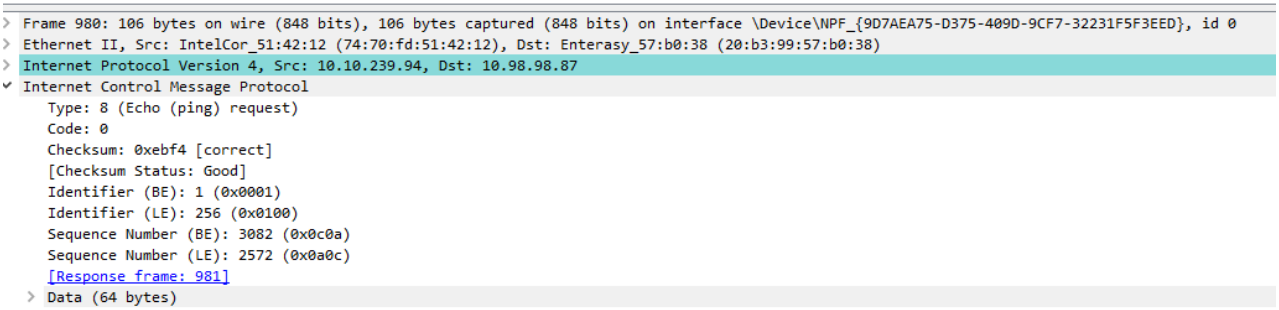


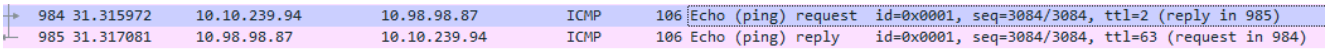
The ping query packets are different from the ICMP error packets. The error's original ICMP packet's first 8 bytes are also included, along with the IP header.

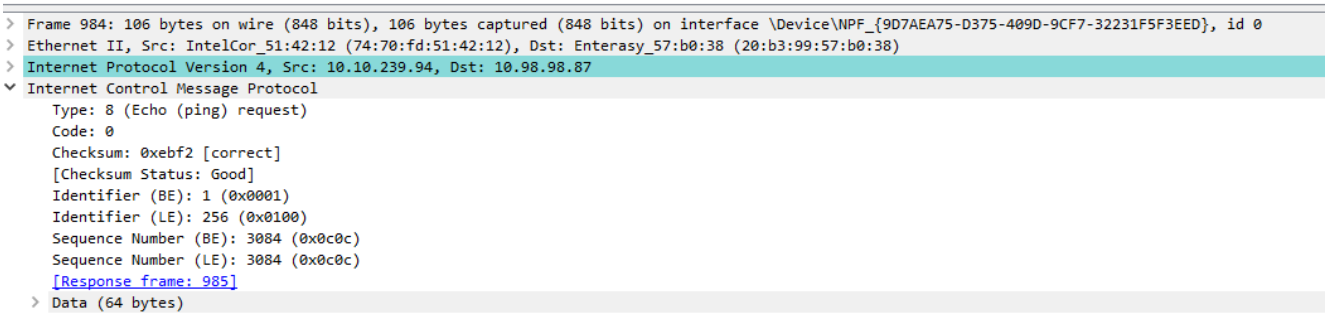
9. Examine the last three ICMP packets received by the source host. How are these packets different from the ICMP error packets? Why are they different?

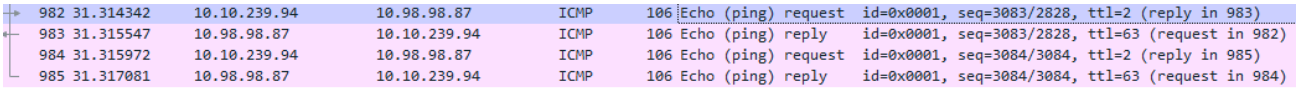
Message type 8 is seen in the past three ICMP packets (echo reply). They differ due to the fact that the datagrams arrived at the target host before the TTL ran out.

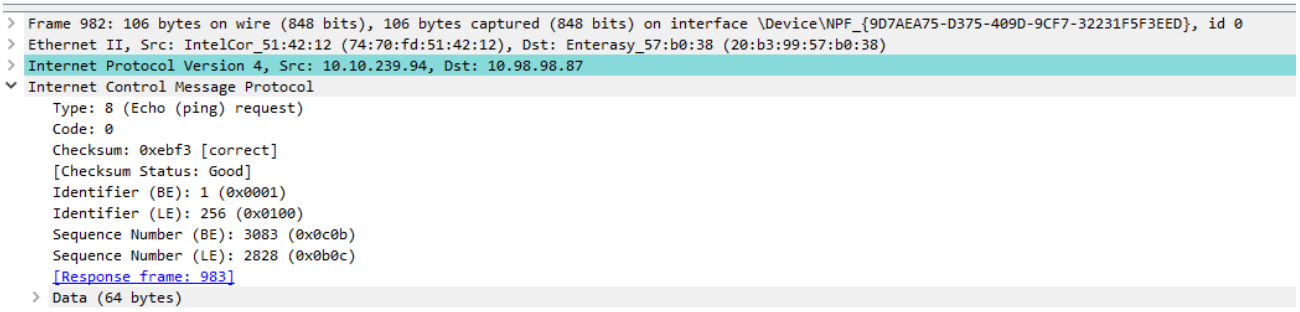












10. Within the tracert measurements, is there a link whose delay is significantly longer than others? Refer to the screenshot in Figure 4, is there a link whose delay is significantly longer than others? Based on the router names, can you guess the location of the two routers on the end of this link?

Since my delay value is one, I could not compare, but if I could compare, I would comment by looking at the transmitic link forms.