COMPUTER NETWORKS (CS F303)
LAB-SHEET –
Topic: Learning IP and ICMP using Wireshark

Objectives:

- To learn the working of IP protocol using Wireshark
- To understand the use and working of the ICMP protocol.

In this lab, we will continue doing the experiments with Wireshark to analyze and understand various network layers' protocols. In this lab, we will focus on another fundamental network layer protocol called IP (Internet Protocol) and ICMP (Internet Control Message Protocol).

IP (Internet Protocol):

The Internet Protocol provides the network layer (layer 3) transport functionality in the Internet Protocol Family (Like, ICMP). The IP protocol is used to transfer packets from one IP address to another. The user of this layer will give a packet and a remote IP address, and IP is responsible for transferring the packet to that host. *IP doesn't provide any mechanism to detect Packet Loss* and *Duplicate Packets. IP uses ICMP to transfer control messages to a remote host, such as "Please don't send me more IP packets, I'm full"*. ICMP protocol, which we are going to study in a subsequent lab.

Lab Exercise:

In order to generate a trace of IP datagrams for this lab, we'll use the traceroute program to send datagrams of different sizes towards some destination, X. Recall that traceroute operates by first sending one or more datagrams with the time-to-live (TTL) field in the IP header set to 1; it then sends a series of one or more datagrams towards the same destination with a TTL value of 2; and so on. Recall that a router must decrement the TTL in each received datagram by 1, if the TTL reaches 0, the router returns an ICMP message (type 11 – TTL-exceeded) to the sending host. As a result of this behavior, a datagram with a TTL of 1 (sent by the host executing traceroute) will cause the router one hop away from the sender to send an ICMP TTL-exceeded message back to the sender; the datagram sent with a TTL of 2 will cause the router two hops away to send an ICMP message back to the sender; the datagram sent with a TTL of 3 will cause the router three hops away to send an ICMP message back to the sender; and so on. In this manner, the host executing traceroute can learn the identities of the routers between itself and destination X by looking at the source IP addresses in the datagrams containing the ICMP TTL-exceeded messages.

For Windows:

A Windows traceroute program is pingplotter, available both in free version and shareware versions at http://www.pingplotter.com. Download and install pingplotter, and test it out by performing a few traceroutes to your favorite sites.

Linux/ Unix/ Mac OS traceroute command:

Syntax: traceroute <hostname> <datagram size>

Example: > traceroute www.google.com 2000

Do the following steps:

Step 1: Startup Wireshark and begin packet capture (Capture->Start) and then press OK on the Wireshark Packet Capture Options screen (we'll not need to select any options here).

Step 2: Enter three traceroute commands (as shown above). ,

One with a length of 56 bytes, one with a length of 2000 bytes, and one with a length of 3500 bytes.

Step 3: Stop Wireshark tracing.

Analysis of traced packets:

In your trace, you should be able to see the series of ICMP Echo Requests (in the case of Windows machine) or the UDP segment (in the case of Unix) sent by your computer and the ICMP TTL-exceeded messages returned to your computer by the intermediate routers.

NOTE: If you want to print the packet for later analysis reference, you can perform the following steps: 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.

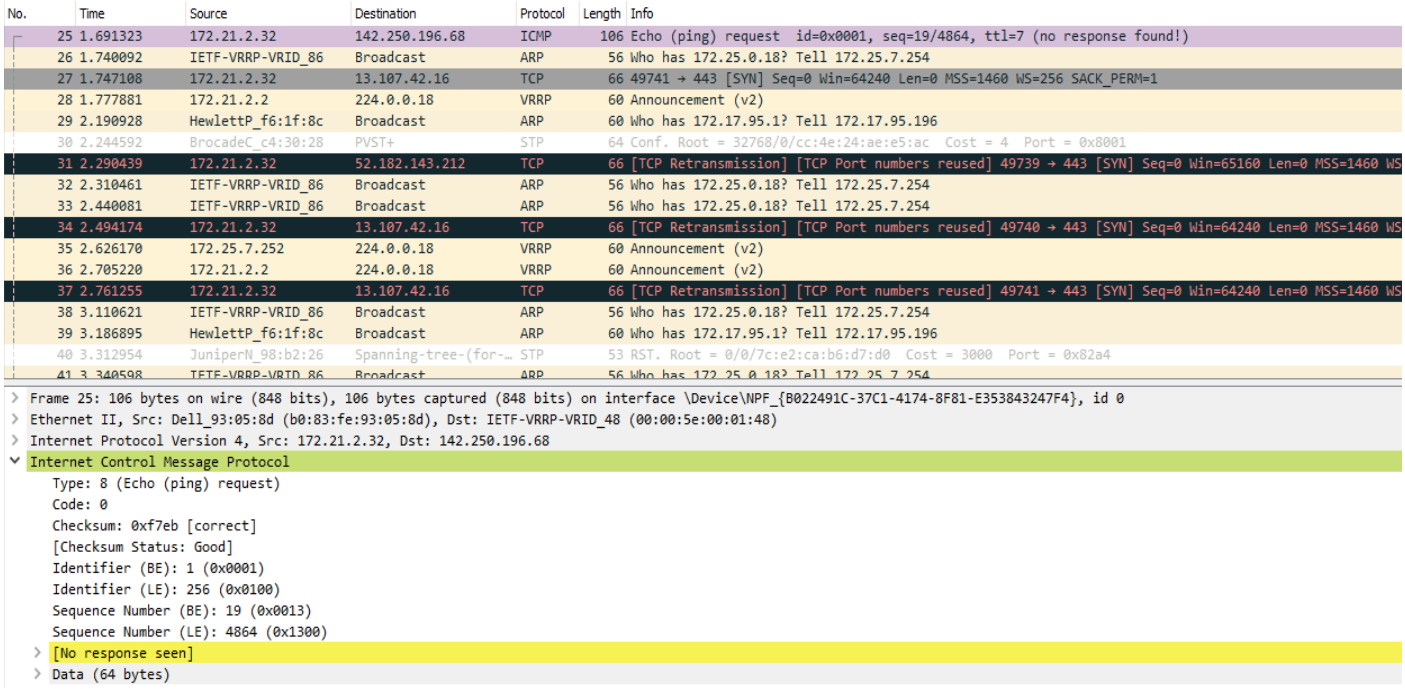

Figure 1: Wireshark window of marked ICMP request packets

1. Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol part of the packet in the packet details window. What is the IP address of your computer?

2. What is a router and network solicitation message?

3. Within the IP packet header, what is the value in the upper layer protocol field?

4. How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.

5. Has this IP datagram been fragmented? Explain how you determined whether or not the datagram has been fragmented.

Next, sort the traced packets according to IP source address by clicking on the Source column header; a small downward pointing arrow should appear next to the word Source. If the arrow points up, click on the Source column header again. Select the first ICMP Echo Request message sent by your computer, and expand the Internet Protocol portion in the "details of selected packet header" window. In the "listing of captured packets" window, you should see the subsequent ICMP messages (perhaps with additional interspersed packets sent by other protocols running on your computer) below this first ICMP. Use the down arrow to move through the ICMP messages sent by your computer.

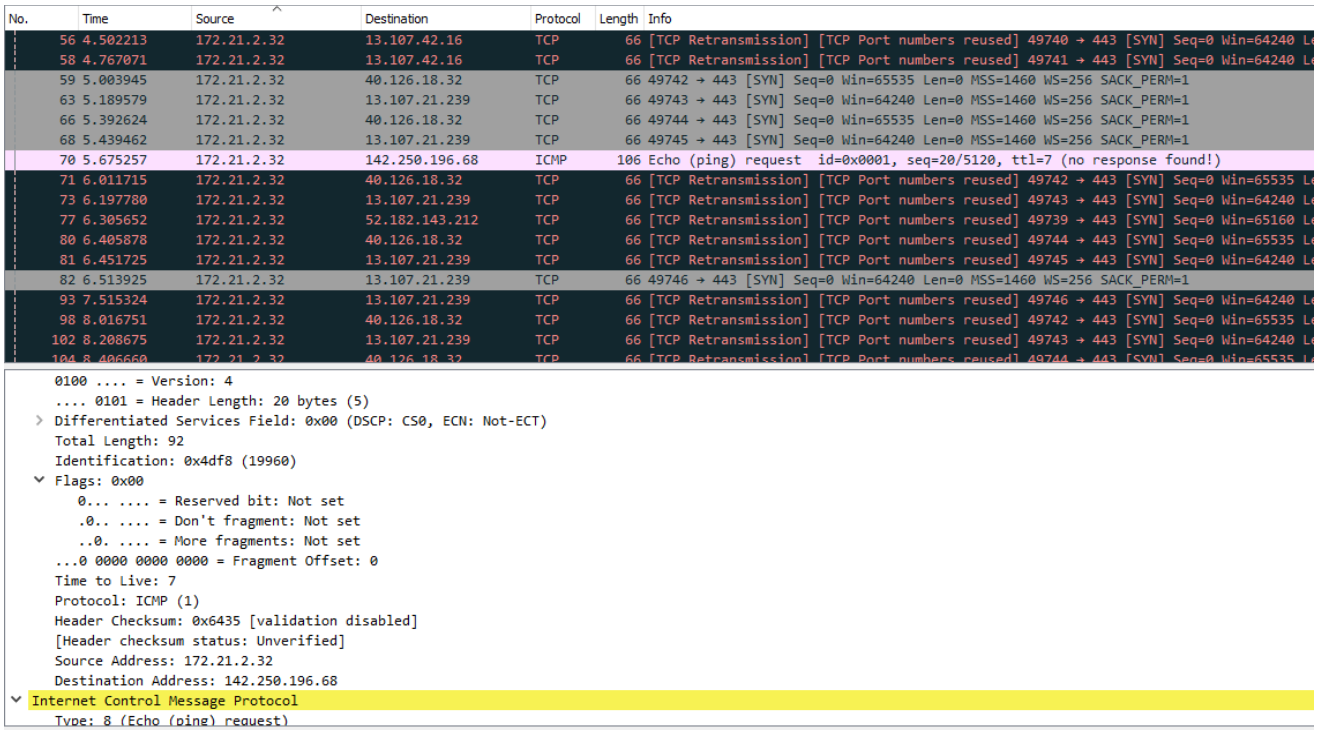

Figure 2: Wireshark window of marked ICMP reply packet detailed fields

6. Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer?

7. Which fields stay constant? Which of the fields must stay constant? Which fields must change? Why?

Next (with the packets still sorted by source address) find the series of ICMP TTL- exceeded replies sent to your computer by the nearest (first hop) router.

8. What is the value in the Identification field and the TTL field?

9. Do these values remain unchanged for all of the ICMP TTL-exceeded replies sent to your computer by the nearest (first hop) router? Why?

Fragmentation with IP:

Sort the packet listing according to time again by clicking on the Time column.

10. Find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 2000. Has that message been fragmented across more than one IP datagram?

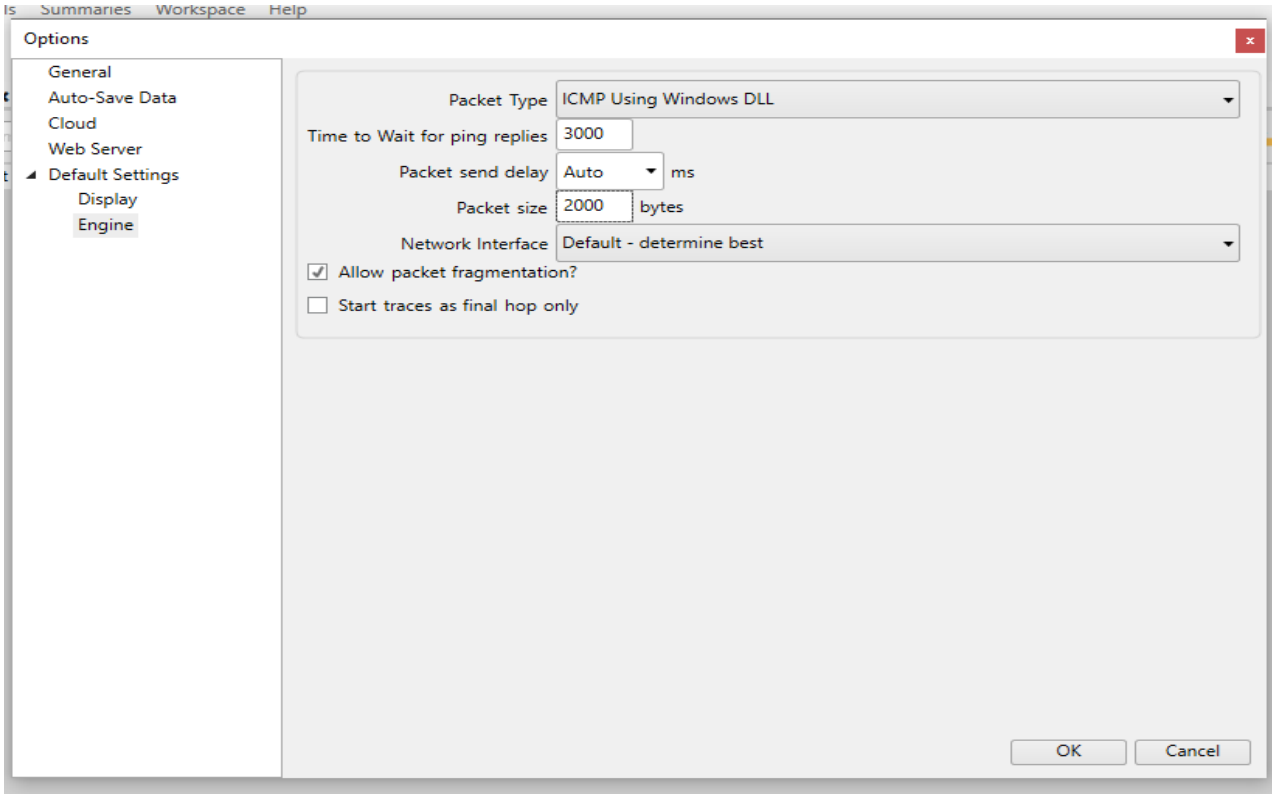

Figure 3: Pingplotter window for editing packet size (Edit -> options -> edit packet size)

11. Print out the first fragment of the fragmented IP datagram. What information in the IP header indicates that the datagram has been fragmented? What information in the IP header indicates whether this is the first fragment versus a latter fragment? How long is this IP datagram?

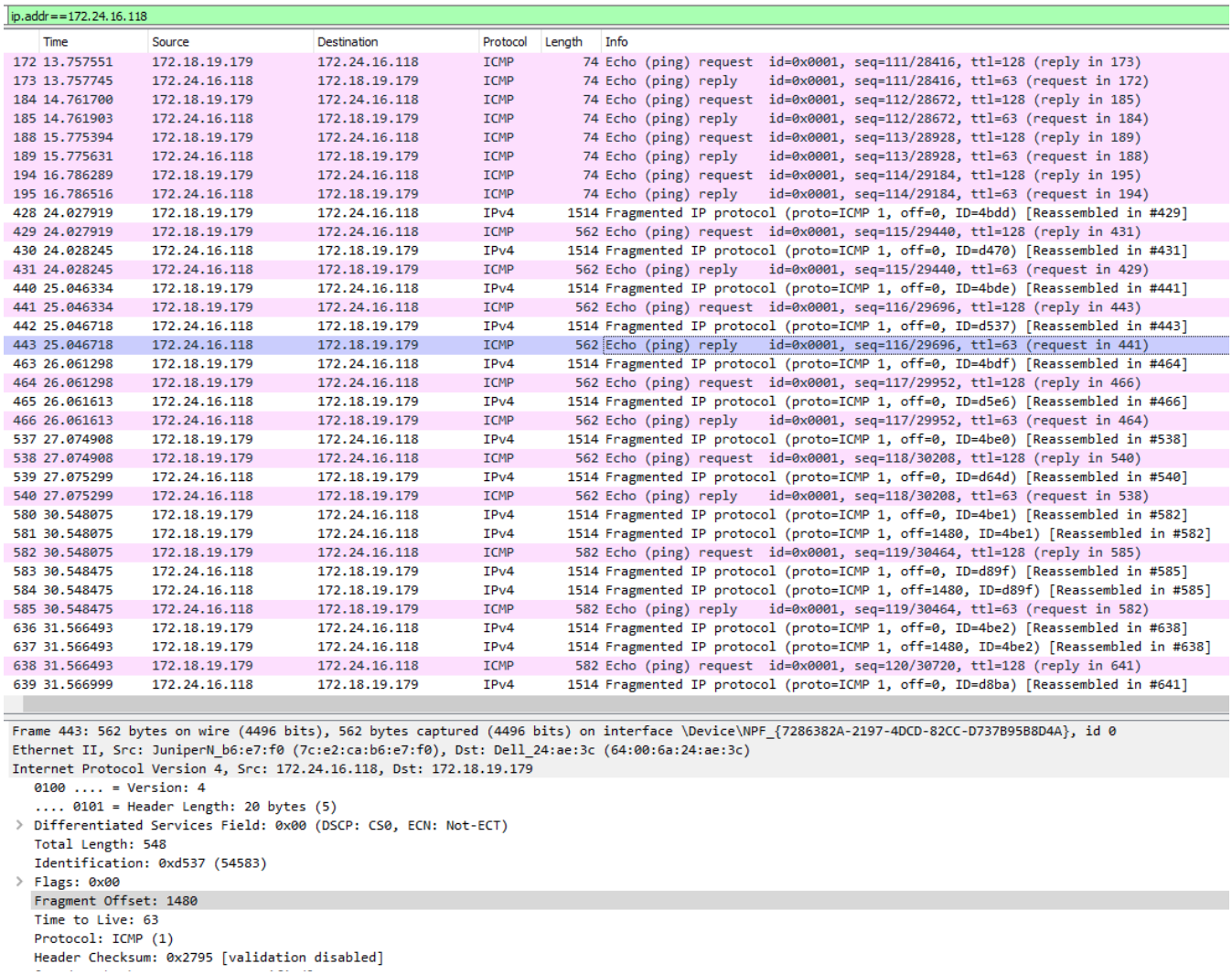

Figure 4: Ping reply

12. Print out the second fragment of the fragmented IP datagram. What information in the IP header indicates that this is not the first datagram fragment? Are there more fragments? How can you tell?

13. What fields change in the IP header between the first and second fragment? Now find the first ICMP Echo Request message that was sent by your computer after you changed the Packet Size in pingplotter to be 3500.

14. How many fragments were created from the original datagram?

15. What fields change in the IP header among the fragments?

ICMP

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

1. ICMP messages generated by the Ping program;
2. ICMP messages generated by the Traceroute program; and

The format and contents of an ICMP message.

ICMP: Internet Control Message Protocol

- ICMP or Internet Control Message Protocol is Internet or Network layer protocol. In general, it is used to check the reachability of a host or router in a network.
- Ping or traceroute uses ICMP as the inner protocol. Ping uses ICMP echo requests and ICMP echo reply messages to check whether the destination host is reachable or not.
- ICMP packet types:
    - ICMP echo request messages.
    - ICMP echo reply messages.

We present this lab in the context of the Microsoft Windows operating system. However, it is straightforward to translate the lab to a Unix or Linux environment.

Lab exercise:

1. ICMP and Ping: Ping program is a 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 site or 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.

How to get an ICMP packet in Wireshark?

Step 1: Open the command line or terminal in Windows or Linux respectively.

Step 2: Start up the Wireshark packet sniffer, and begin Wireshark packet capture.

Step 3: Run the command: **Syntax:** ping <hostname>

Example: > ping www.google.com or > ping –n 10 www.google.com or > ping 192.168.1.1 *(using IP address)*

In the terminal or 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.

Step 4: When the Ping program terminates, stop the packet capture in Wireshark.

```
C:\Users\lenovo>ping www.google.com

Pinging www.google.com [172.217.167.132] with 32 bytes of data:
Reply from 172.217.167.132: bytes=32 time=11ms TTL=55
Reply from 172.217.167.132: bytes=32 time=21ms TTL=55
Reply from 172.217.167.132: bytes=32 time=12ms TTL=55
Reply from 172.217.167.132: bytes=32 time=17ms TTL=55

Ping statistics for 172.217.167.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 21ms, Average = 15ms
```

Figure 5: terminal window after entering the Ping command.

At the end of the experiment, your Terminal Window should look something like Figure 5. Make sure you have an internet connection or ping will fail. Here is the snapshot for the successful ping to Google. Note also that for each response, the source calculates the ***round-trip time (RTT)***, which for the 04 packets is on average 15 msec. We can see 0% ***loss.*** That means ***ICMP request packets = ICMP reply packets***. There are certain other details like IP address, the amount of data, min and max amount of round trip time.

Step 5: Stop Wireshark and put "ICMP" as a filter in Wireshark.

Let's check what happens in Wireshark when we ping to Google or 192.168.1.1.

Here is the ICMP request and reply packets for Google ping.

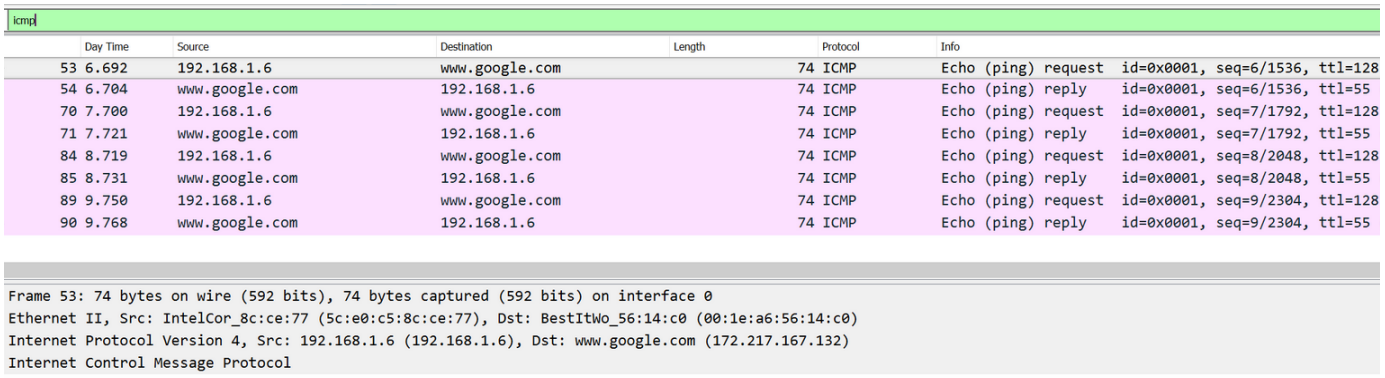

Figure 6: Wireshark window after entering ICMP in the search/ filter box.

Number of ICMP requests: From the capture, we can see there are 4 ICMP request packets. Check the marked packets.

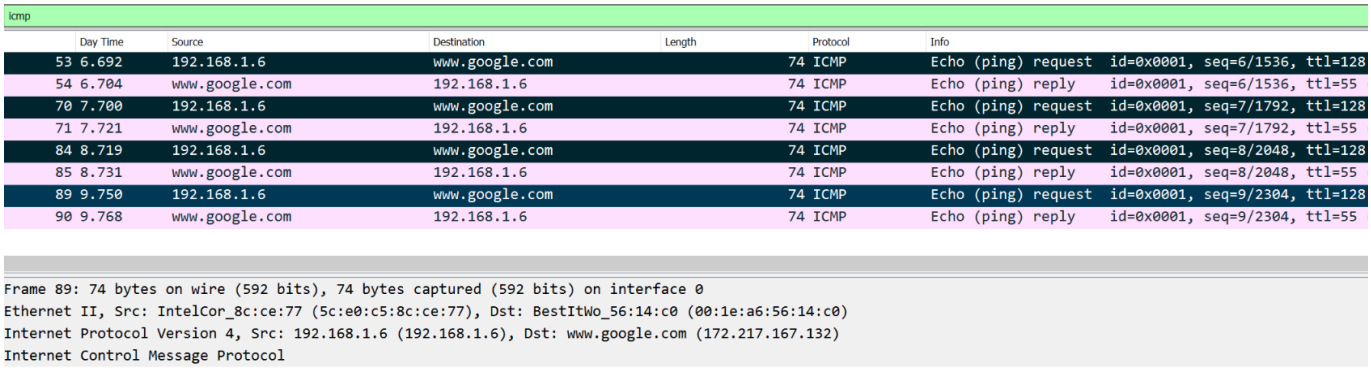

Figure 7: Wireshark window of marked ICMP request packets.

Number of ICMP replies: From the capture, we can see there are 4 ICMP reply packets. Check the marked packets.

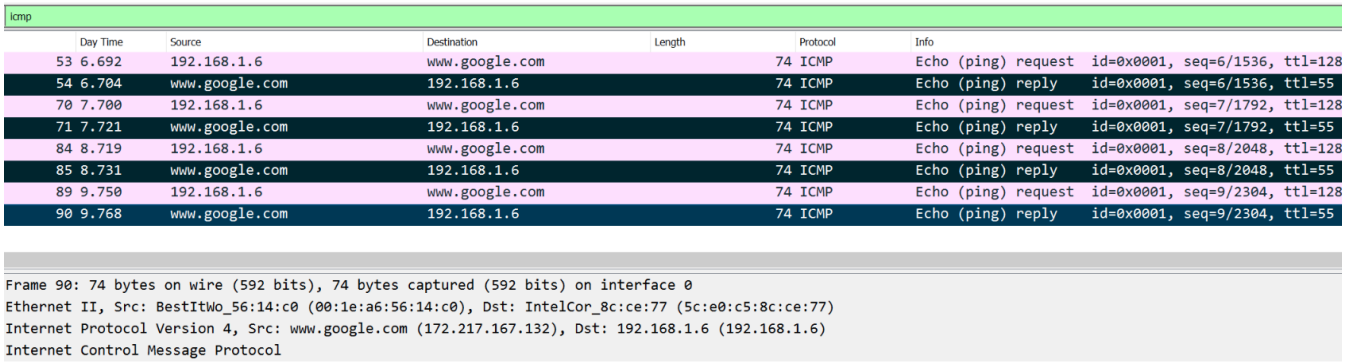

Figure 8: Wireshark window of marked ICMP reply packets.

Analysis of ICMP Request: Now select the ICMP request/ reply packet in Wireshark and look into the IPv4 layer. As this is an ICMP request packet, we can see the source IP as my system IP address and the destination IP as Google's one IP address. Also, the IP layer mentioned the protocol as ICMP.

Now, for the same packet, select the ICMP part in Wireshark. We can see below the important fields:

```
Type: 8 [Means its ICMP request]
Code: 0 [Always 0 for ICMP packets]
Identifier (BE): 1
Identifier (LE): 256
Sequence Number (BE): 6
Sequence Number (LE): 1536
*BE -> Big Endian
*LE -> Little Endian
Data -> Data present in ICMP packet.
```

Figure 9: Important fields of ICMP packet

Answer the following questions by observing your Wireshark screen:

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

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

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?

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?

Lab Exercise:

2. ICMP and Traceroute: Let's now continue our ICMP adventure by capturing the packets generated by the Traceroute program. 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.

Do the following steps:

Step 1: Start up the Wireshark packet sniffer, and begin Wireshark packet capture.

Step 2: Open the terminal and write the tracert command:

Syntax: tracert <hostname> **Example:** > tracert www.google.com

Step 3: When the Traceroute program terminates, stop packet capture in Wireshark.

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 10 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.

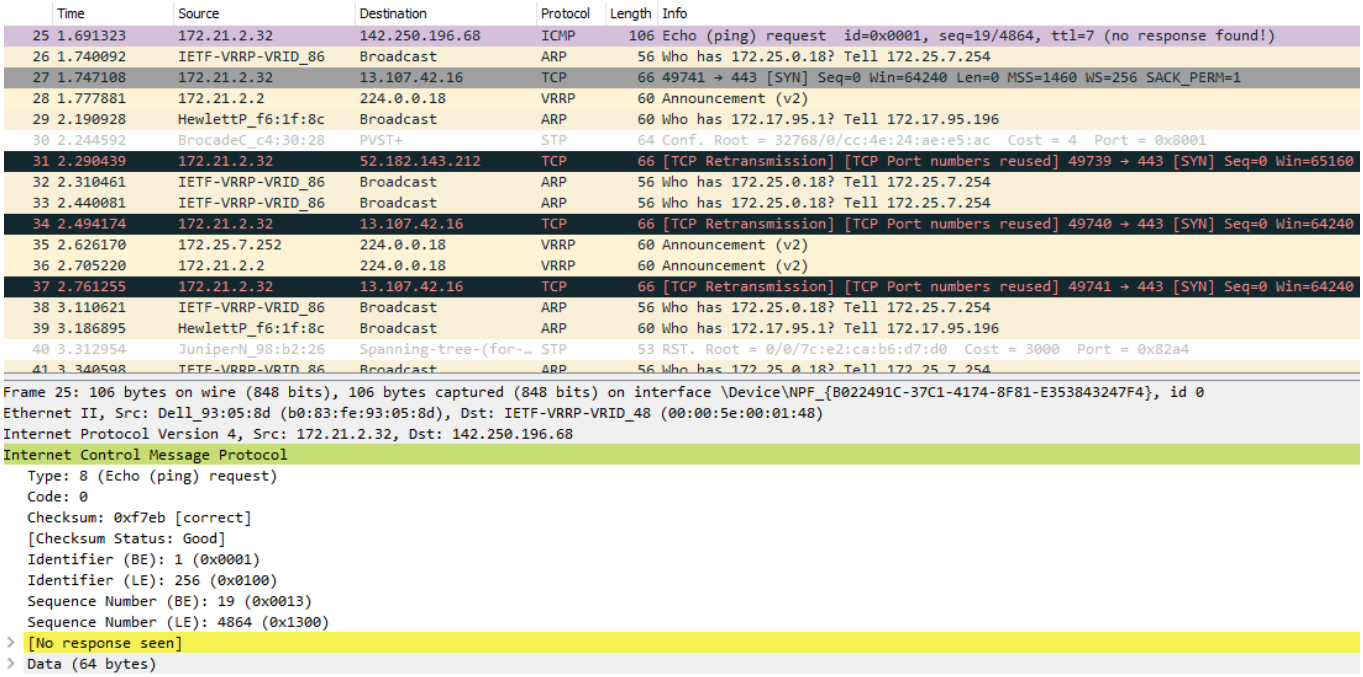

Figure 10: Wireshark window of ICMP fields expanded for one ICMP error packet.

Answer the following questions:

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

2. 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?

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?

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

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?

6. Within the tracert measurements, is there a link whose delay is significantly longer than others? (Refer the traceroute window).