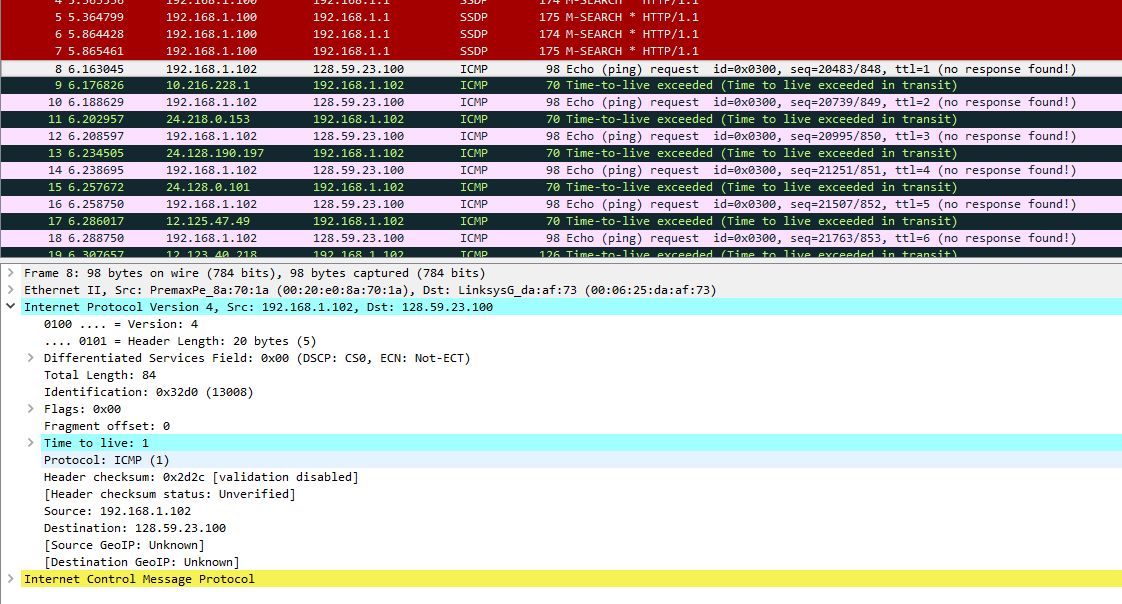
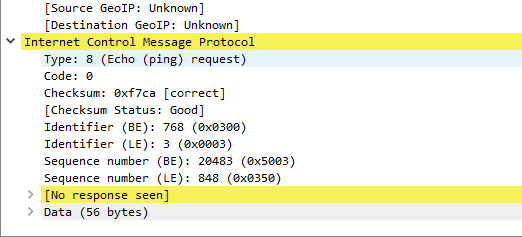
|  |  |
| --- | --- |
|  | LAB 4: Analysis of IP packets through Wireshark and Introduction to static routing through packet tracer. |
|  |  |
|  | Akshar Panchani ID- 202101522  IT304 Computer Networks  8/29/23 |



**Exercise:**

**1.2:**





Answering below questions from the above screenshots

1. What is the IP address of your computer?

IP Address: 192.168.1.102.

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

Protocol: ICMP(1).

3. 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.

The IP header length= 20 bytes.

The total length of the packet = 84 bytes

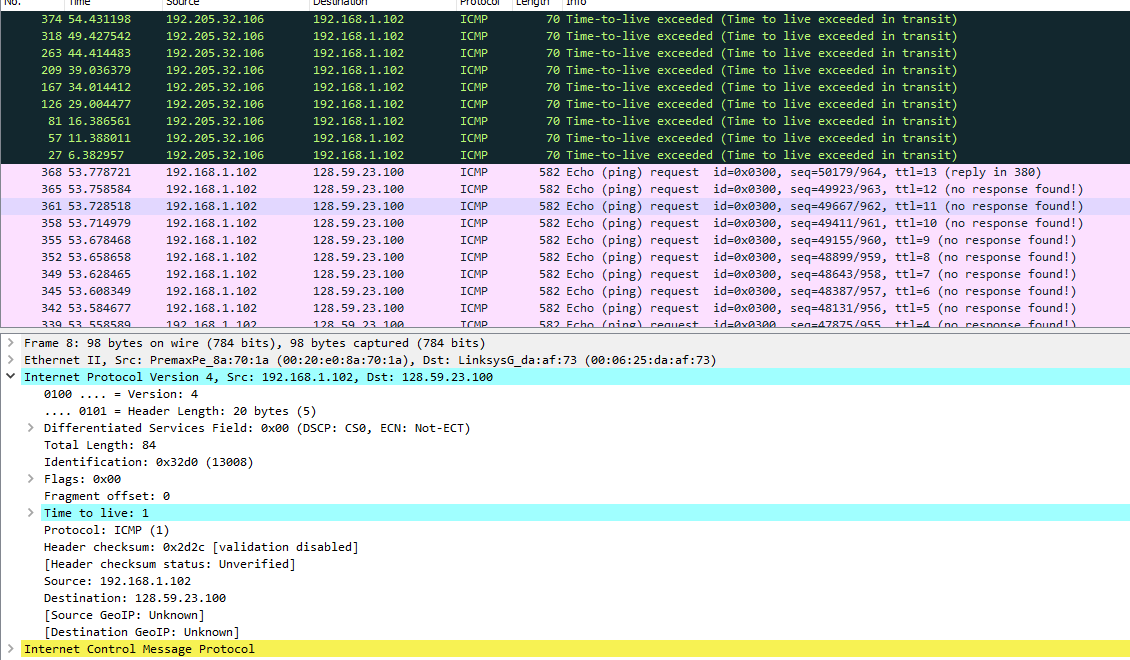
Payload -> 84 - 20 = 64 bytes.

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

Fragment Offset: 0

Therefore there is no fragmentation.

**1.3:**



**1.4:**

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

3 fields change from one datagram to another: Identification , Time to live and Header Checksum.

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

The fields that stay and must remain constant are:

- Source IP: Source is the same computer.

- Destination IP: Destination remains same.

-Protocol: ICMP for all.

- Differentiated Services Field: All the packets are ICMP(they use the same

   types of services)

- Version: All the packets areIPv4

- Header Length: As we are using ICMP.

The fields that change are:

1. Identification:There should be a unique packet id.

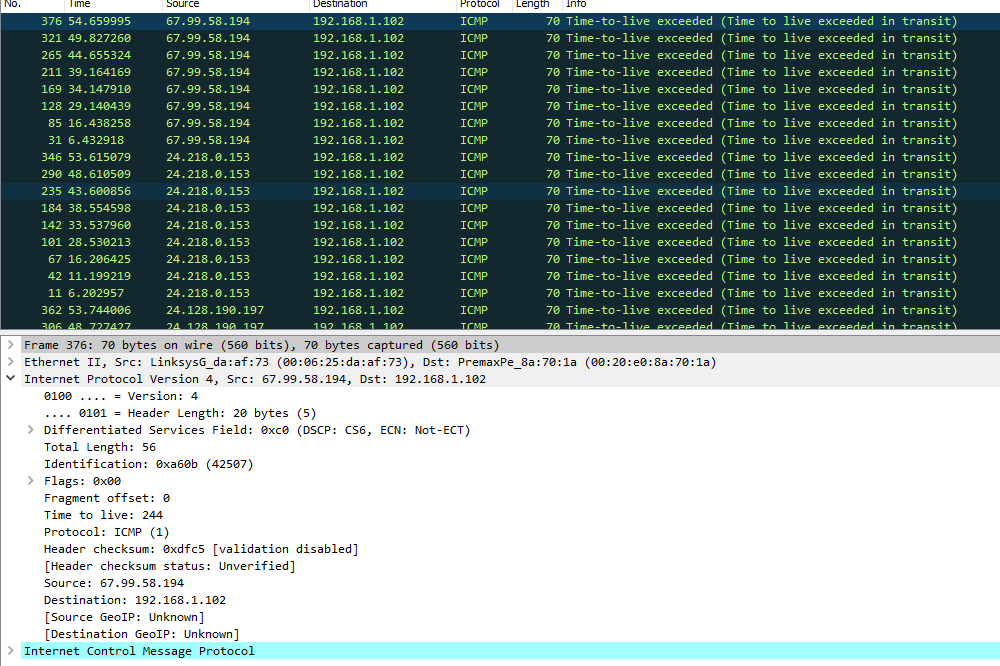
2. Time to live: It decreases as it is in descending order.

3. Header checksum: Checksum changes as headers change.

3. Describe the pattern you see in the values in the identification field of the IP datagram.

The values change by 1 in the identification field.

4. 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.



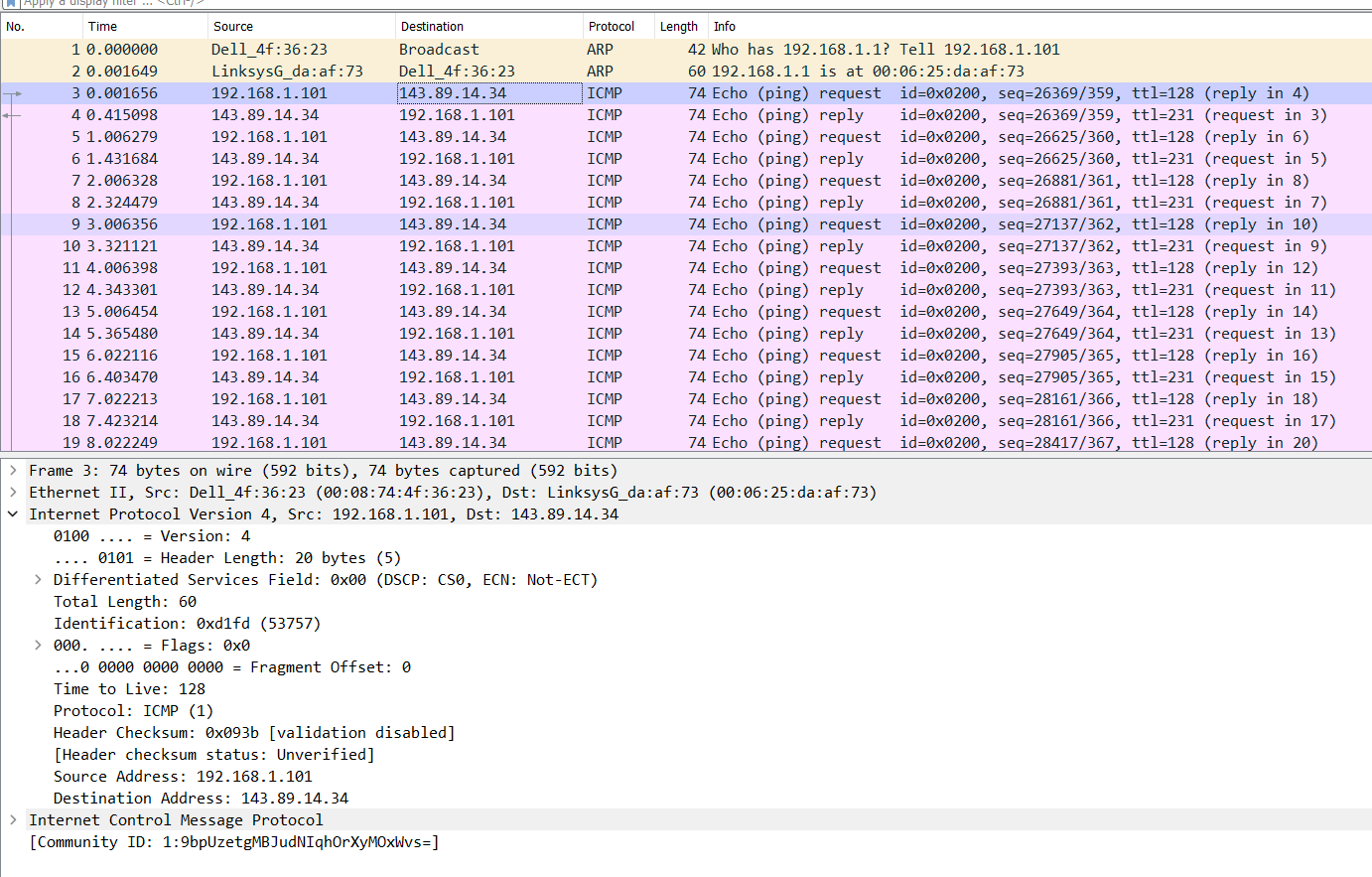
5. What is the value in the Identification field and the TTL field?

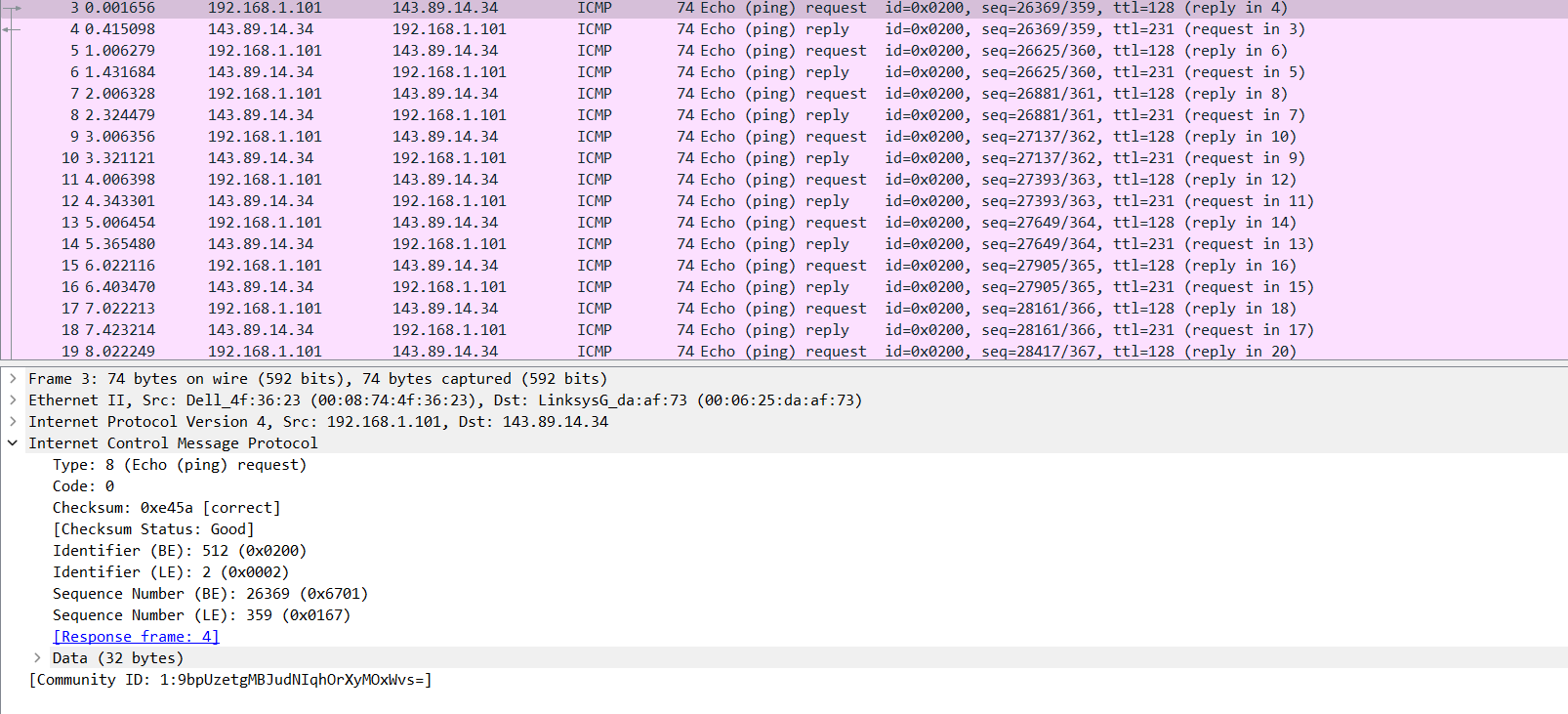
TTL= 244 and Identification = 0xa60b (42507)

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

Yes, for all ICMP TTL-exceeded answers from the closest router, the TTL and identification values stay unaltered. Since the first hop router is constant, the TTL does not vary. Since these IP datagrams are parts of a larger IP datagram, they all have the same identification value.

**1.5:**





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

IP address of host = 192.168.1.101

IP address of destination host = 143.89.14.34.

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

The ICMP packet lacks source and destination port information since it was not intended to be used for application layer processes to exchange network-layer data. Its purpose was to facilitate information exchange at the network layer between hosts and routers. Each ICMP packet have type and code attached to it. They help to identify the message that is being received. No port numbers are required to route an ICMP message to an application layer process because the network software interprets all ICMP signals.

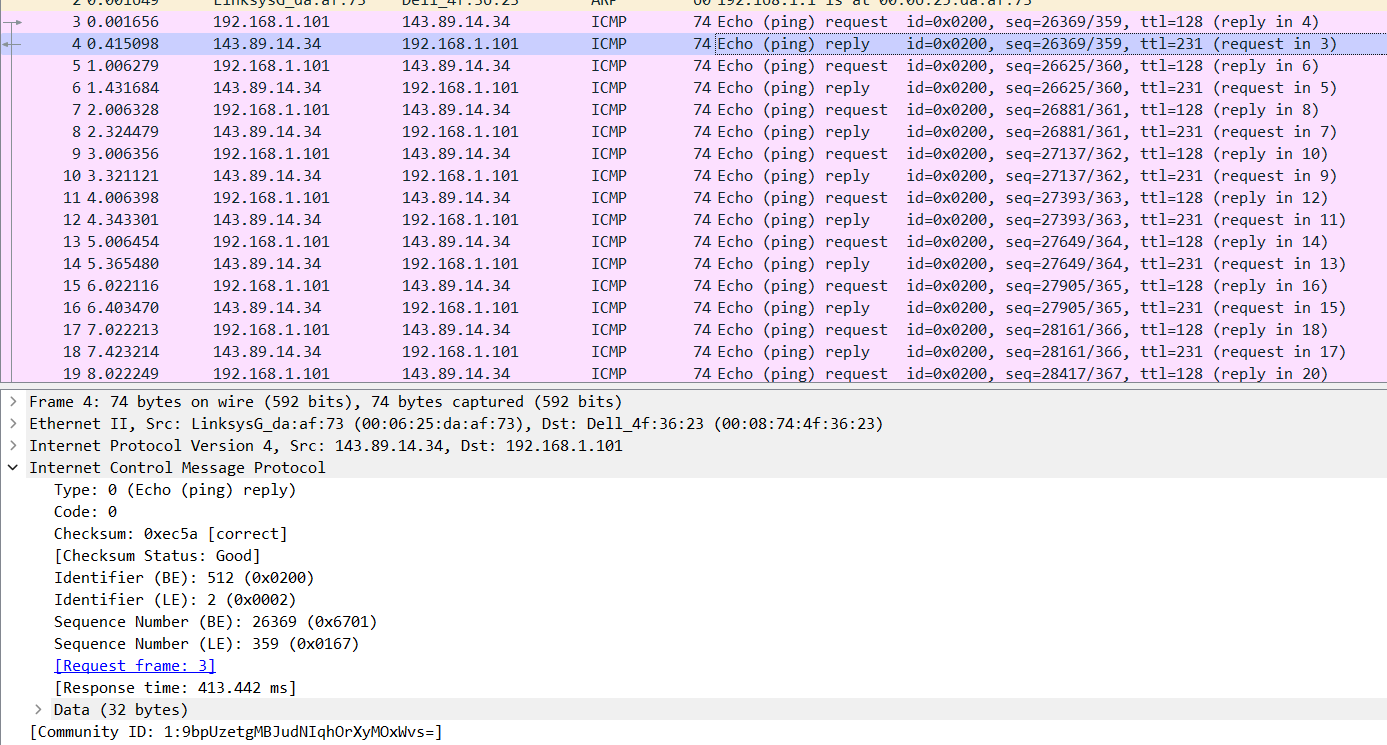
3. Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?

ICMP type = 8 and code number =0.

Some other fields in this ICMP packet are:

Checksum, identifier, sequence number, and data fields, each of 2 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?

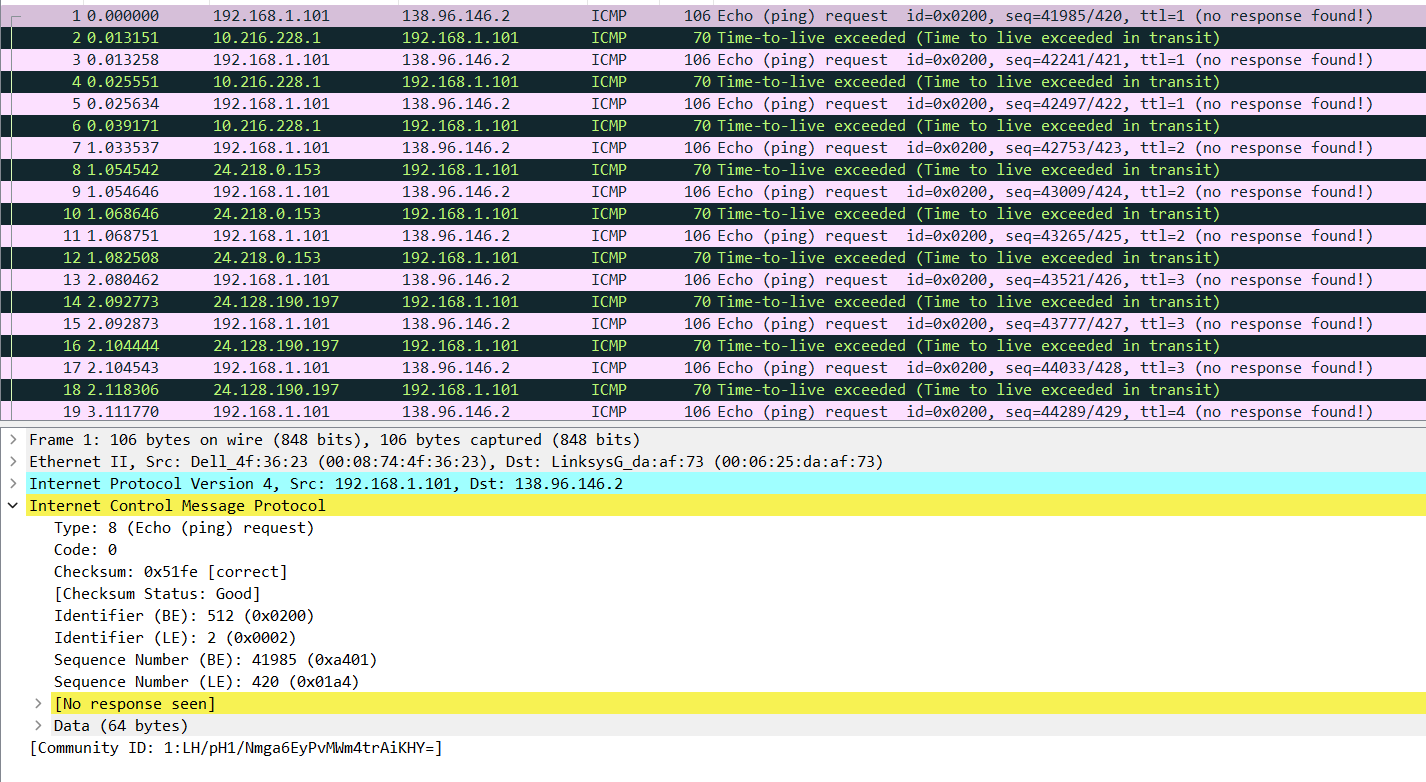


ICMP type =0 and code number = 0.

Some other fields in this ICMP packet are:

Checksum, identifier, sequence number, and data fields, each of 2 bytes

**1.6**



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

The IP address of our host= 192.168.1.101

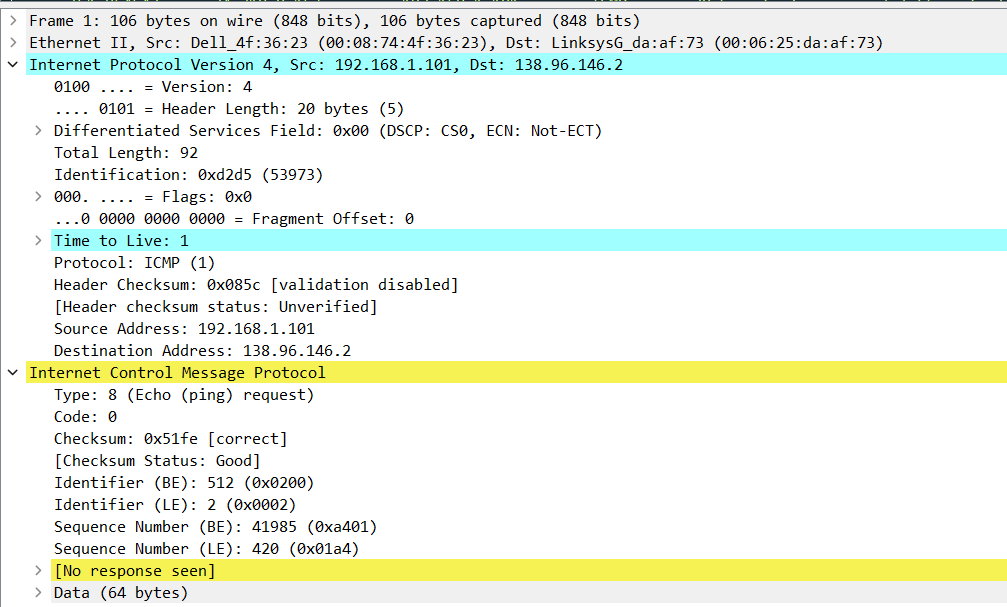
IP address of the target = 138.96.146.2.

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?

No. The IP Protocol number would change to 0x11 if ICMP instead transmitted UDP packets.

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?

The ICMP echo packet is not different and has the same fields.



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

Yes it is more than the echo packet as it contains both the IP header and the first 8 bytes of the original ICMP packet.

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?

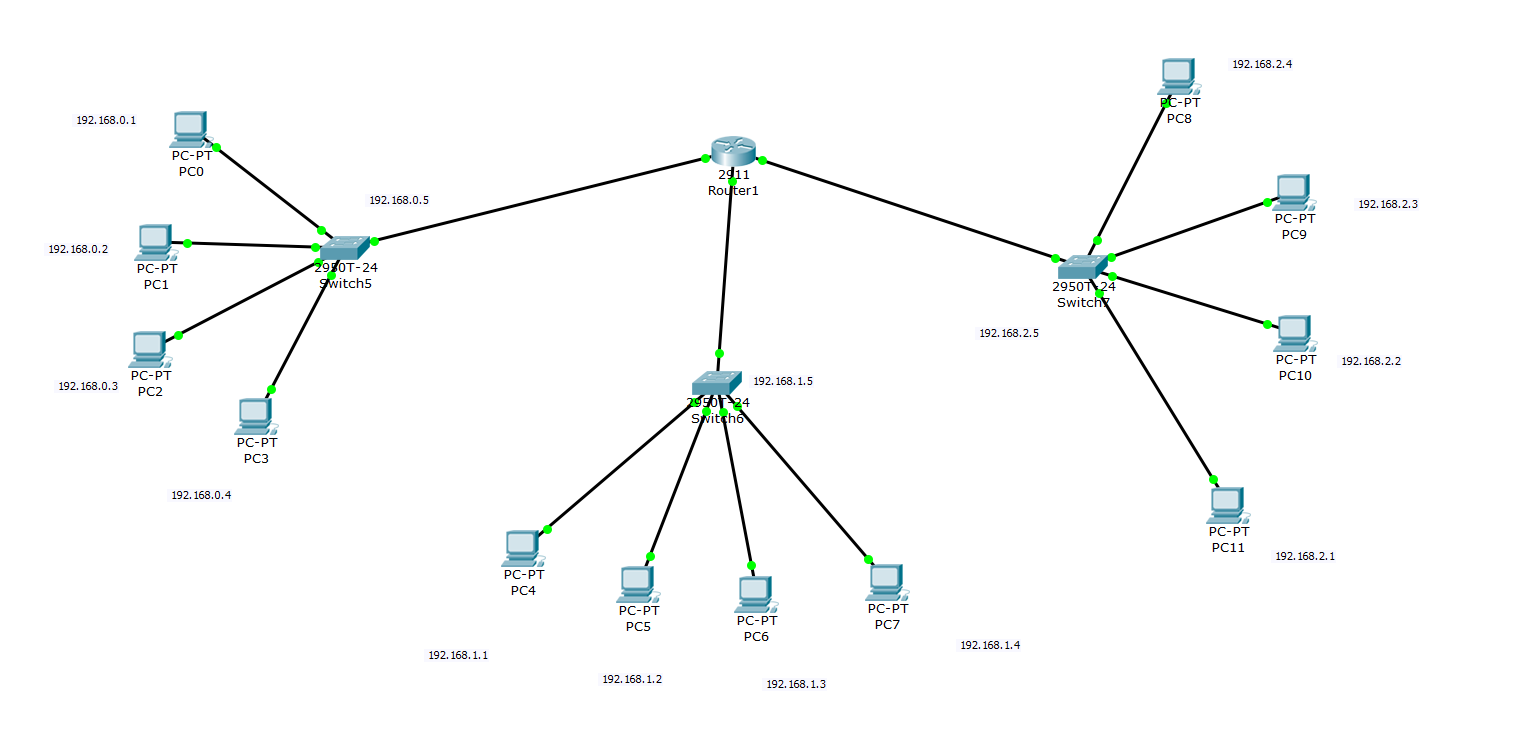
Type field = 0 (echo reply) and not 11 (TTL expired)

It also does not contain additional IP header information.

ICMP packets reach the destination before the TTL happens.

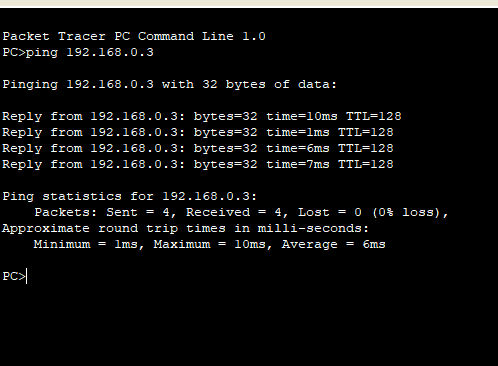
**2.1**

Design a topology which have three networks. Each network has 4 PCs and all three network are connected to each other. The suggested IP ranges are 192.168.0.1 to 192.168.2.4. All IP addresses of all network should be from the given range. Run the experiment and ping from each network to every other Network. Take a snapshot and submit. Also submit the snapshot of topology with IP assigned to each PC.

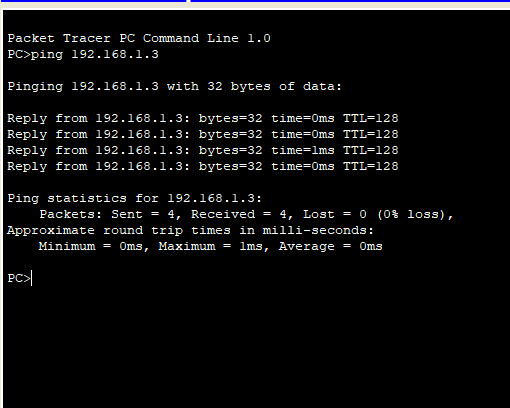


Here are the 3 cases of Ping from all the three different PC connected across another PC through the router.

Case 1



Case 2



Case 3

