

**Department of Computer Engineering**

**Spring 2018**

**CMPE-208: Net Architecture and Protocol**

**Internet Control MessageProtocol**

**Group Lab 3 Report**

**By**

**Team 4**

**Virat Mathur**

**Shivangi Gupta**

**Tejas Chumbalkar**

**Vishal Govindraddi Yarabandi**

Table of Contents

[Overview and Purpose 3](#_Toc506737649)

[Brief Overview of Address Resolution Protocol (ICMP) 3](#_Toc506737652)

[ICMP ARCHITECTURE 4](#_Toc506737653)

[Working of Internet Control Message Protocol (ICMP) 4](#_Toc506737654)

[ICMP MESSAGE FORMAT 6](#_Toc506737655)

[ADDRESS LEASE TIME 7](#_Toc506737656)

[DHCP ATTACK 7](#_Toc506737657)

[Lab Setup 10](#_Toc506737658)

Lab Configuration.................................................................................................................................10

[Lab Execution and Observations. 15](#_Toc506737659)

[Contribution 15](#_Toc506737660)

References............................................................................................................................................22

## **Overview and Purpose**

# **Brief Overview of Internet Control Message Protocol (ICMP)**

## **What is Internet Control Message Protocol (ICMP)**

## **ICMP ARCHITECTURE**

**Working of Internet Control Message Protocol (ICMP)**

**ICMP MESSAGE FORMAT**

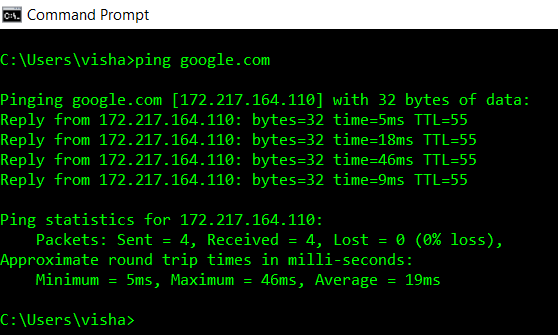
# **Lab Setup**

## **Network Topology**

We observed the behavior of ICMP protocol on a single linux machine (Ubuntu 16.04). Study of various ICMP packets is done using Wireshark tool.

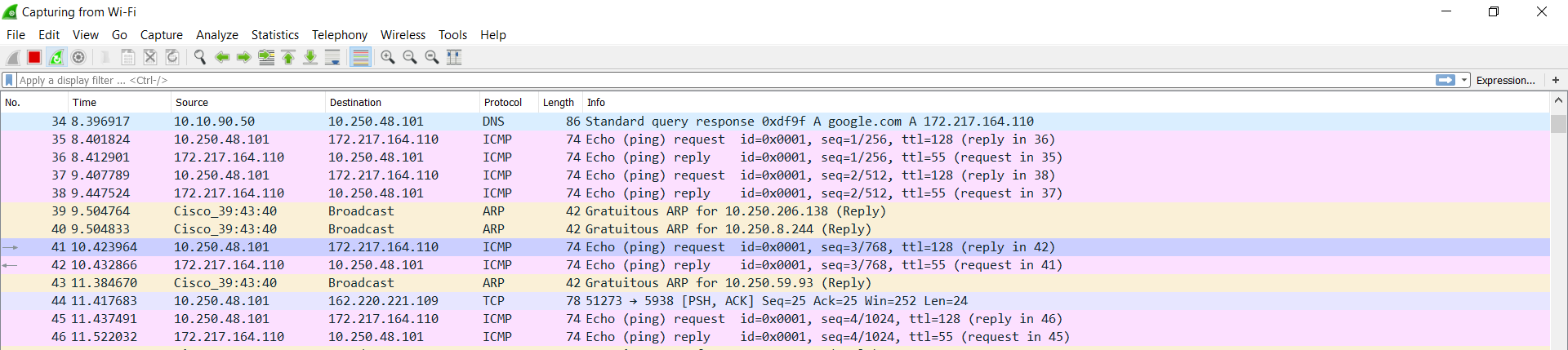
**Lab Execution and Observations:**

**Case 1-: Ping observation**

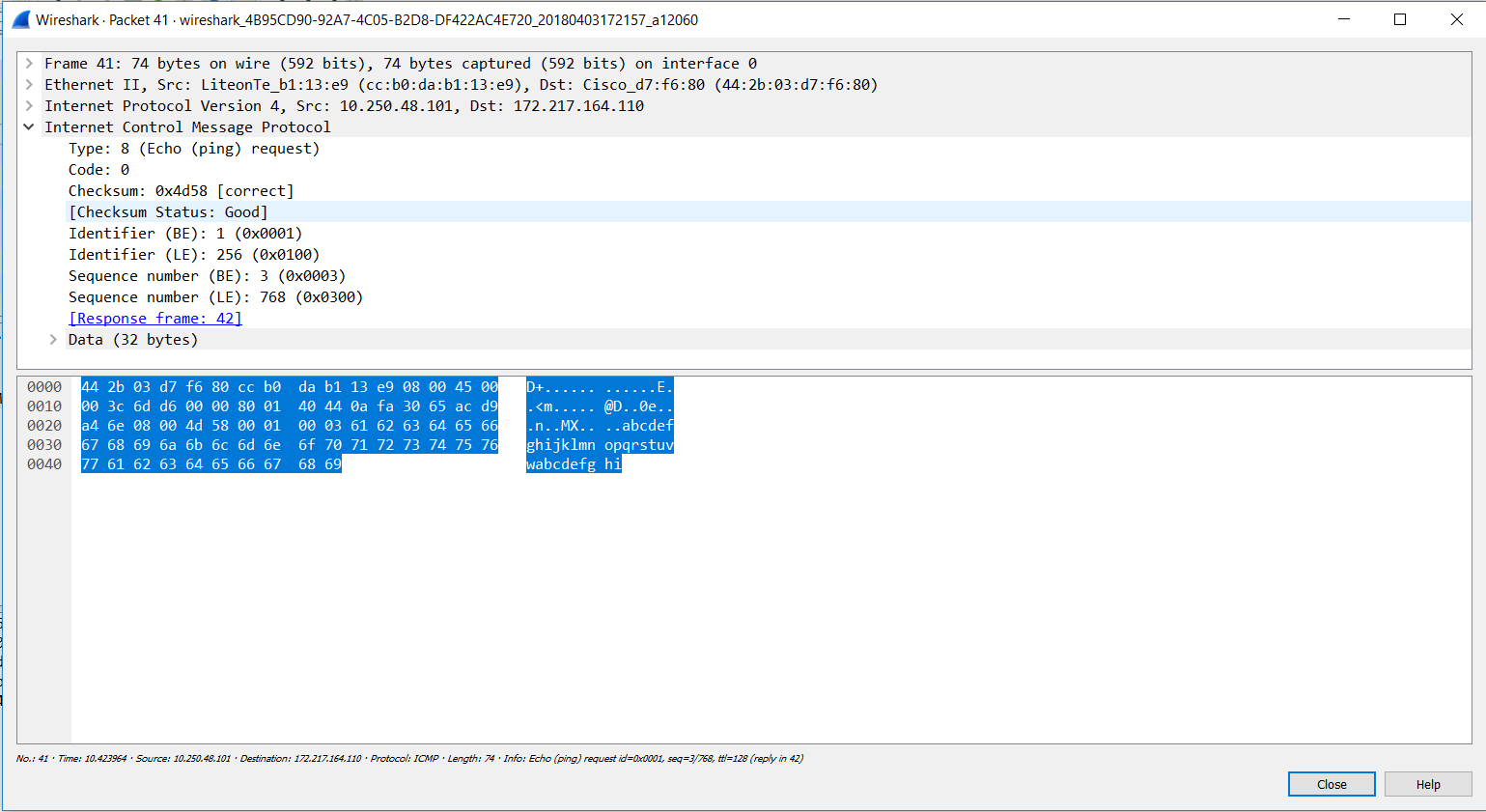
We ping to [www.google.com](http://www.google.com) and observer ICMP packet in Wireshark.

**Obervations:**

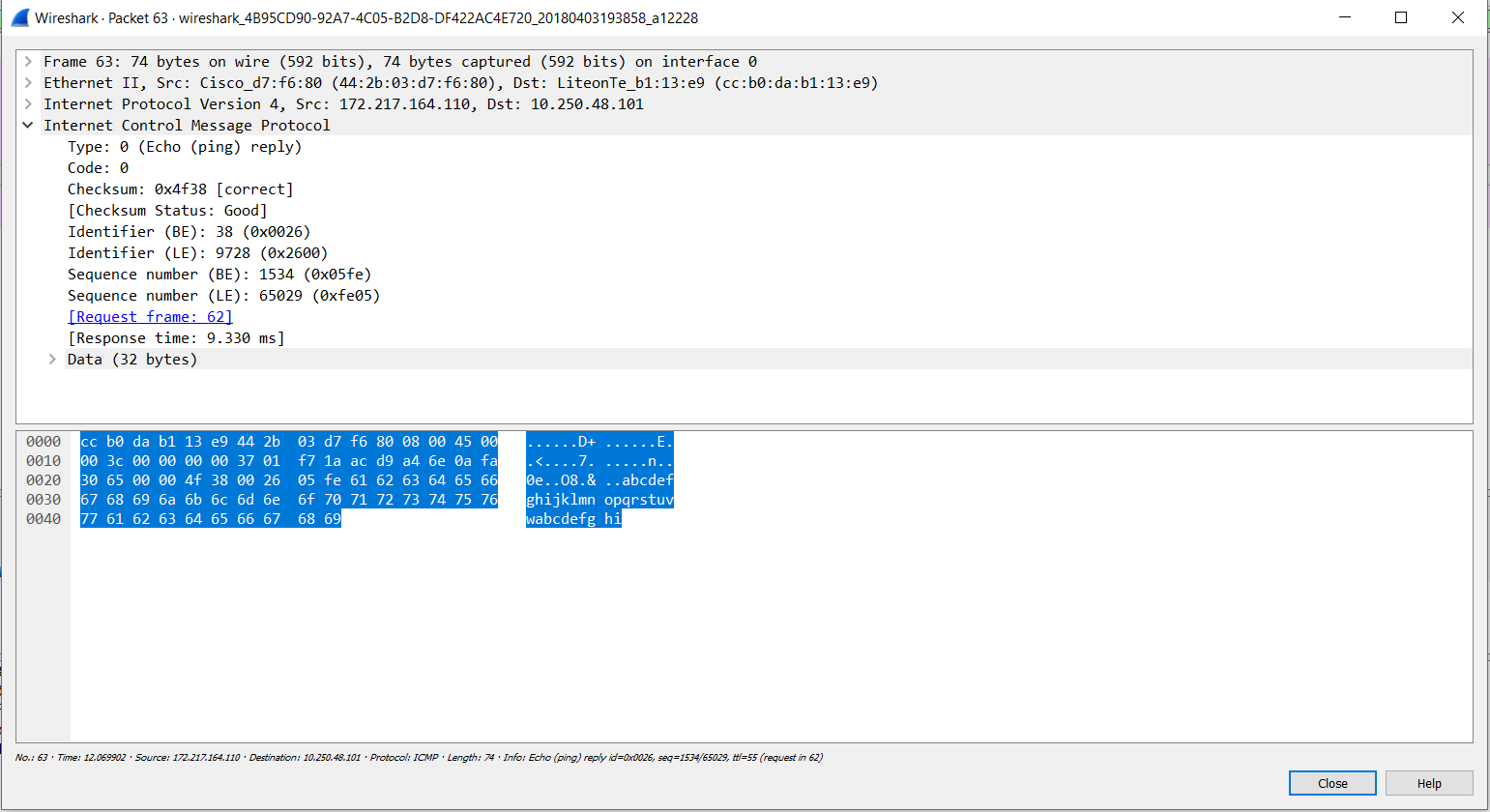
* 32 bytes of data sent per packet
* The domain google.com resolves to IP address 172.217.164.110
* TTL = 55 set by the system and is decremented by 1 each time when a packet is passed through a router.
* time = 5ms, Packet’s roundtrip time.

**Wireshark Activity**

**ICMP request packet**



* Source IP address: 10.250.48.101
* Destination IP address: 172.217.164.110
* Type: 8, as it is a ICMP request packet
* Code: 0

**ICMP reply packet**

* Source IP address: 172.217.164.110
* Destination IP address: 10.250.48.101
* Type: 0, as it is a ICMP reply packet
* Code: 0

**ICMP Message types:**

We will have a brief overview of ICMP message types. **Hping**, a packet generator and analyzer for TCP/IP protocol is used to generate ICMP packets.

**Type 0 – Echo Reply:** Explained in the above scenario.

**Type 1 – Unassigned:** This type is unassigned.

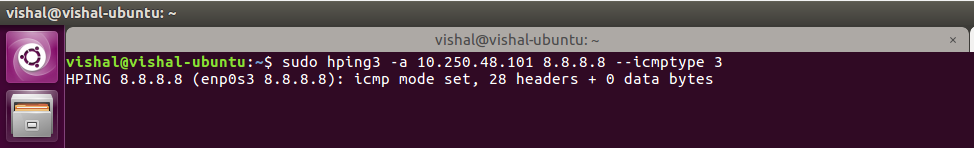
**Type 2 – Unassigned:** This type is unassigned.

**Type 3 – Destination Unreachable:**

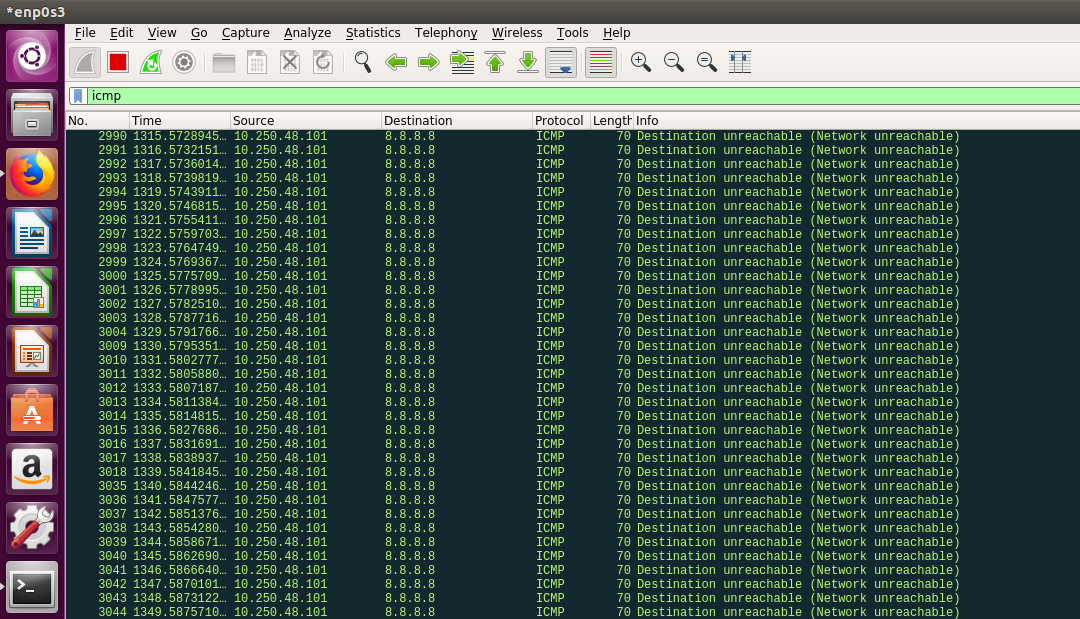
IT alerts the source machine of delivery problems faced while reaching the destination. Destination unreachable uses various codes to describe the specific problem faced.

Following is the ICMP packets for various ICMP codes generated using hping.

**Code 0: Destination Unreachable**

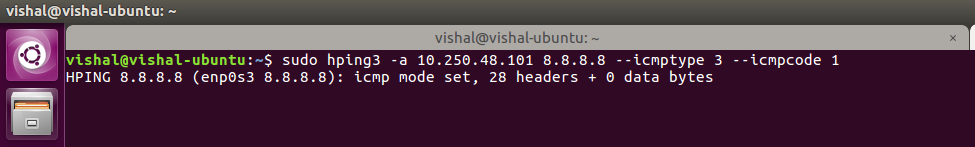
Router cannot find the destination network or there is no route to the network.

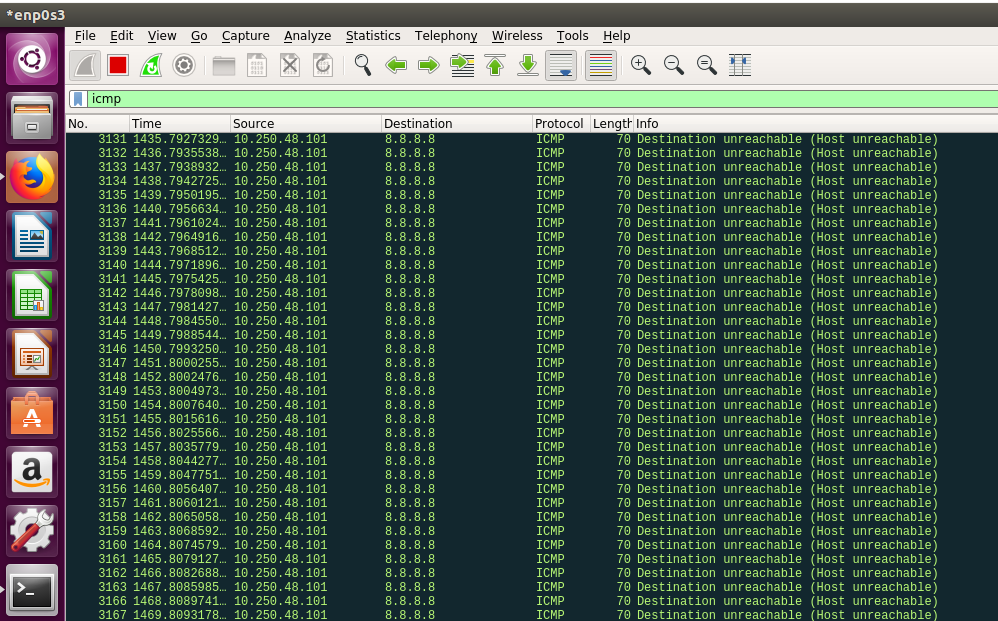
Wireshark capture:



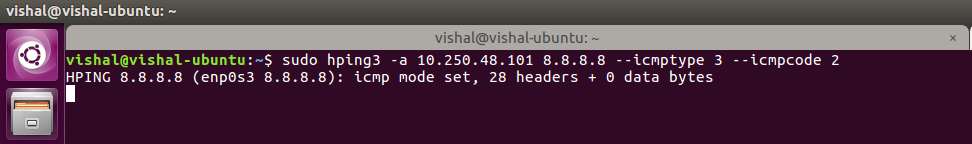
**Code 1: Host Unreachable**

It alerts the sender that the destination host is not found. It can happen if the host machine is turned off or not in existence.

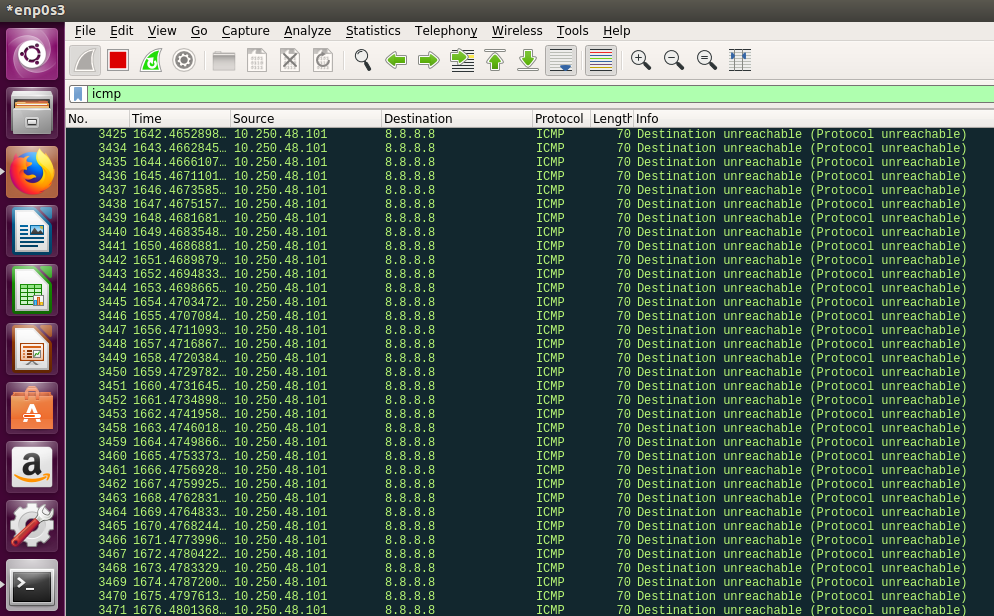


Wireshark capture:

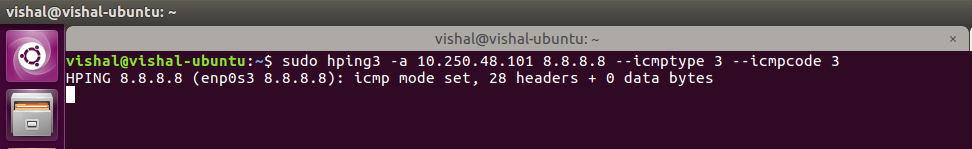
**Code 2: Protocol Unreachable**

It indicated that the protocol underlying transport protocol such as TCP or UDP is unavailable.

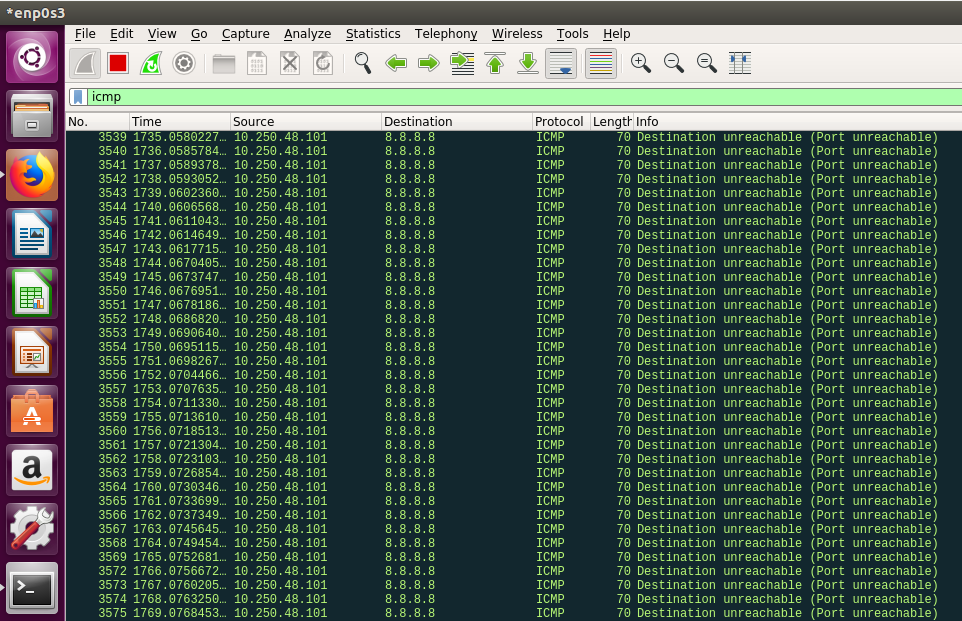
Wireshark capture:



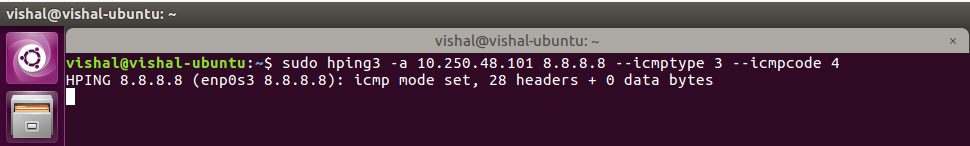
**Code 3: Port Unreachable**

It indicates that the application on the destination host is not active.

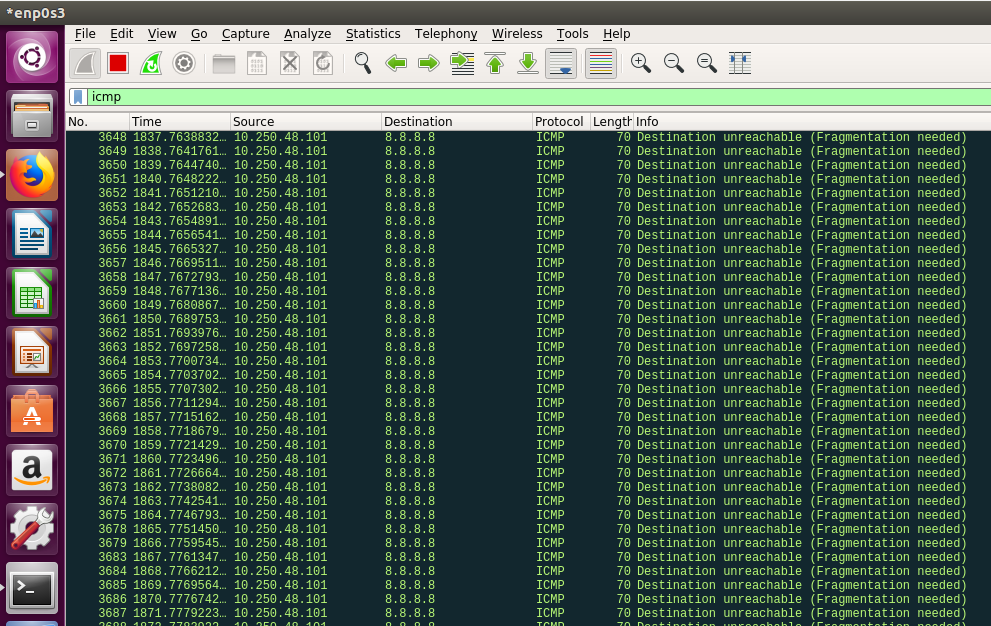
Wireshark



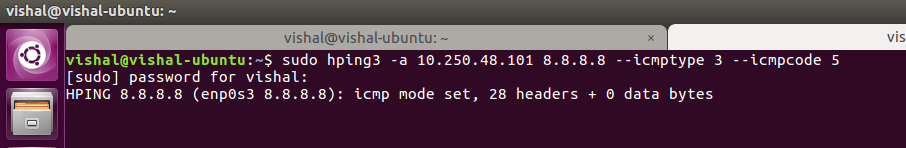
**Code 4: Cannot Fragment**

This occurs when router has datagram for fragmentation but cannot fragment as DF bit is on. Router cannot forward this datagram and will discard it, alerting the sender with this code.

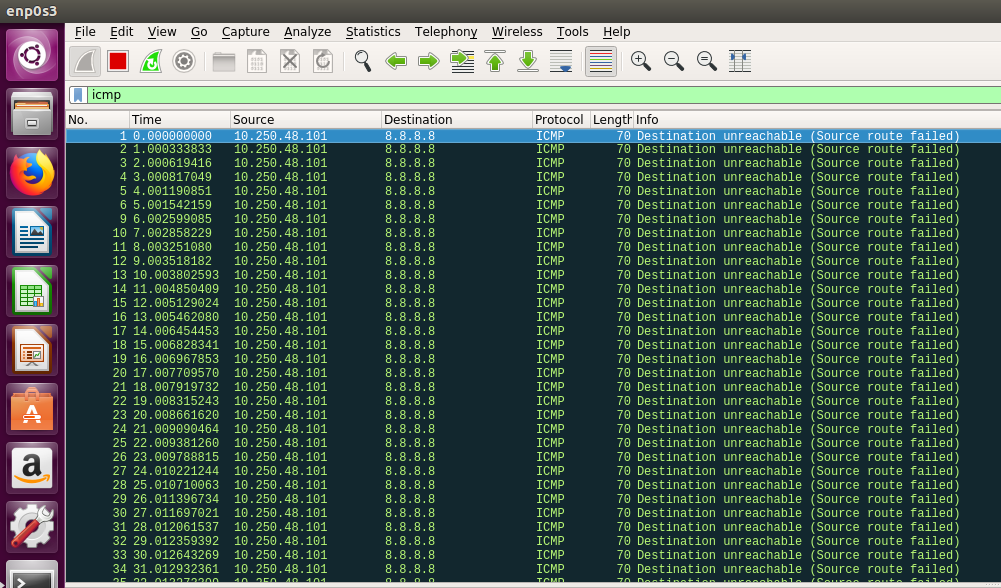
Wireshark



**Code 5: Source Route Failed**

This message is observed when a router finds a next hop in the route that is not on the directly connected network.

Wireshark



**Type 4 – Source Quench:**

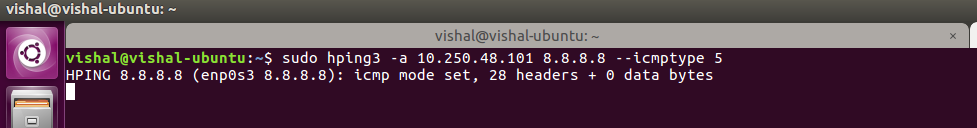
A receiver can generate such message when the received datagram cannot be processed at the respective speed due to insufficient memory or resources. It acts as a flow control mechanism that alerts sending host to slow down the transmission speed.

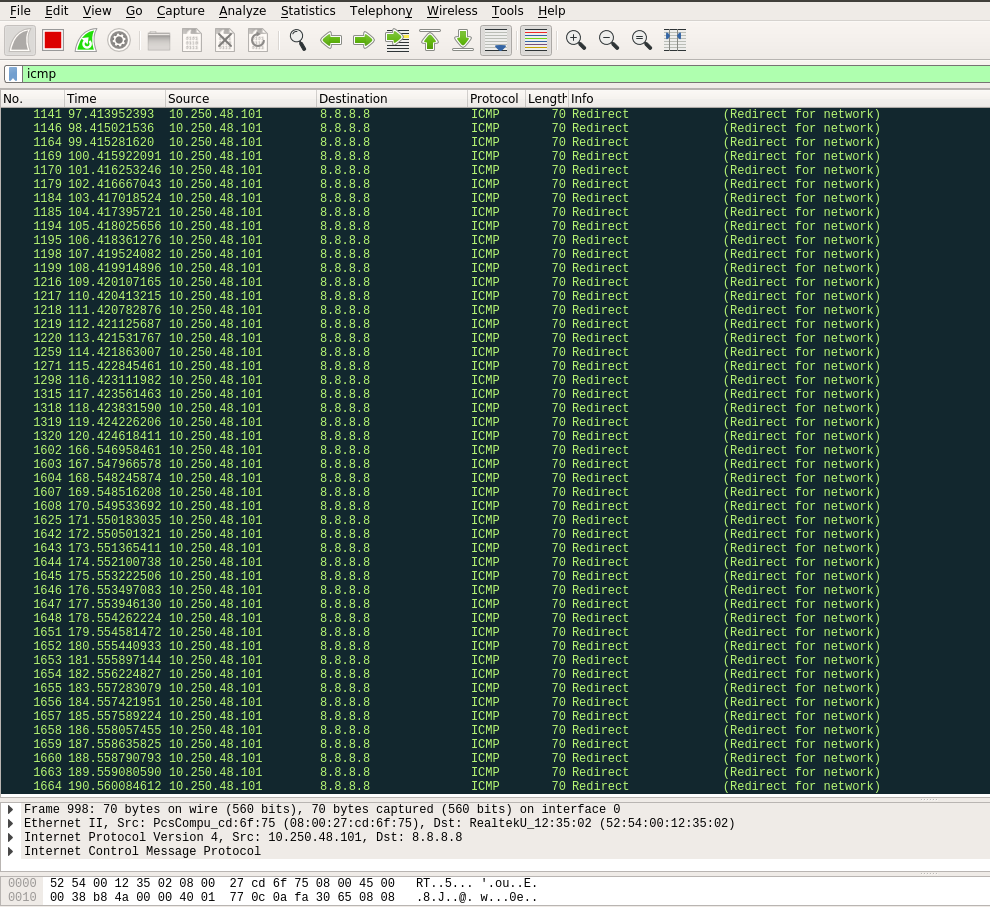
**Type 5 – Redirect:**

A router generates such redirect message to sender when the sender can send this datagram to different router or host. It guides the sender host for the correct route. Only gateways can generate such redirect messages.

Four types of redirect error codes are:

* 0 : Redirect for network.
* 1: Redirect for host.
* 2: Redirect for Type of service and network.
* 3: Redirect for Type of service and host.



Wireshark

**Type 6 – Alternate Host Address:**

This type has been deprecated.

**Type 7 – Unassigned**

**Type 8 – Echo Reply:** Explained in the first above scenario.

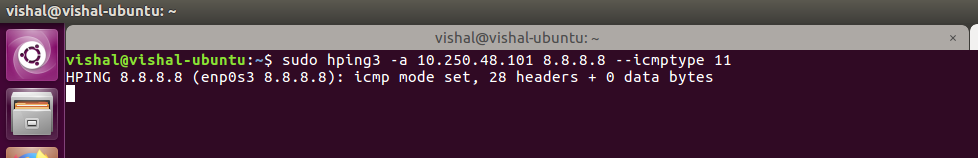
**Type 9 and 10 – Router Advertisement and selection:**

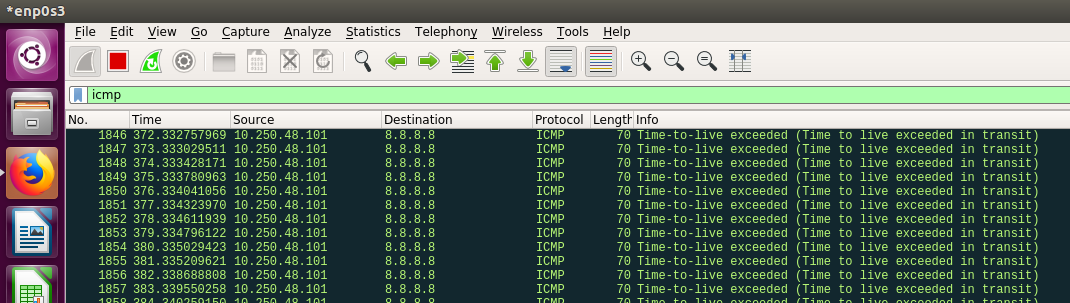
Router advertisement and solicitation messages can be used for updating the routing table. A host can send a multicast or broadcast solicitation message and receive an router advertisement from the router.

**Type 11 – Time Exceeded:**

This message occurs when a TTL =1 or 0 datagram is received by the router. Router discards the datagrams with such TTL values.

Two time exceeded error codes are:

* 0: TTL equals 0 during transit.
* 1: TTL equals 1 during reassembly.

Wireshark

**Type 12 – Parameter Problem:**

This message indicates that a receiving host could not interpret an invalid or misunderstood parameter. In most cases this message indicates some type of implementation error occurred due to vendor incompatibility issues.

Two types of parameter problem error messages:

* 0 = IP bad header
* 1 = Required Option Missing

**Type 13 and 14 – Time Stamp Request and Reply:**

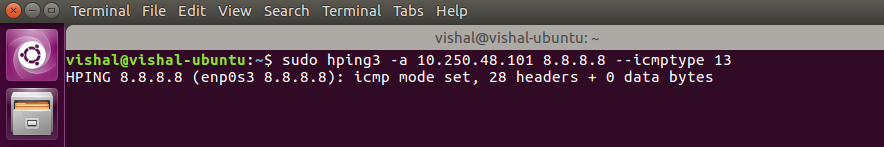
Timestamp request and reply messages work together.

A timestamp request allows to query another for the current time. The value returned is the number of milliseconds since midnight, Coordinated Universal Time.

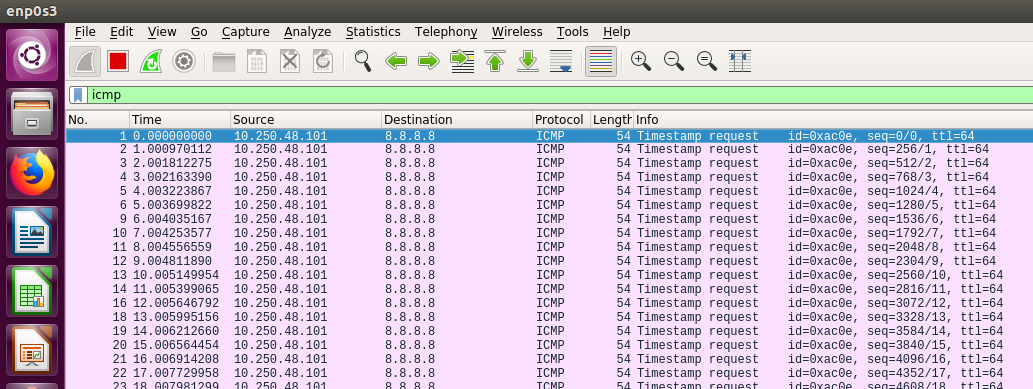
The process for time resolution:

* The requestor stamps the originate time and sends the query.
* The replying system stamps the receive time when it receives the query.
* The replying system stamps the transmit time when it sends the reply to the query.

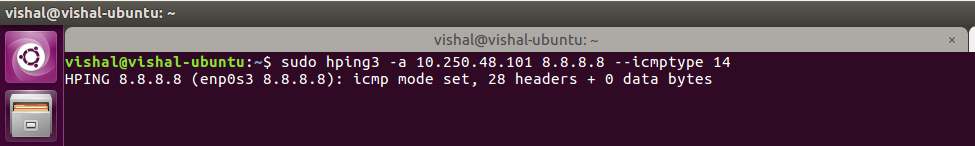
**Timestamp Request**



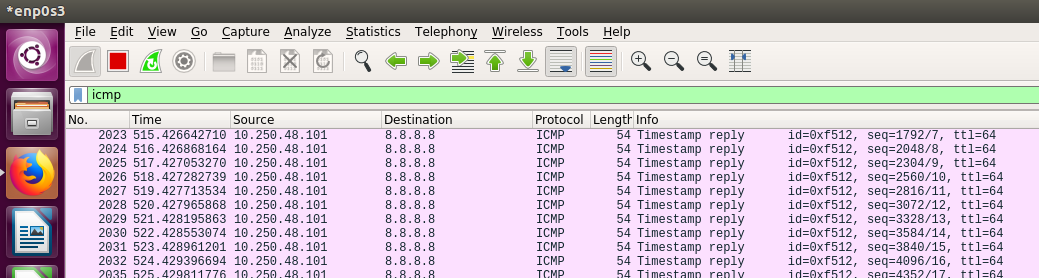
Wireshark



**Timestamp Request**



Wireshark



**Type 15 and 16 – Information Request and Reply:**

Obsolete ICMP messages type. A host can request information like what network it was attached.

**Type 17 and 18 – Address Mask Request and Reply:**

Address mask request and reply messages work together. It support the function to dynamically obtain a subnet mask. Hosts can use this type of ICMP request to acquire subnet masks during bootstrap from a remote host. If there is no response from the remote host, the source host assumes a classful mask.

**ICMP Attacks**

**ICMP Smurf attack:**

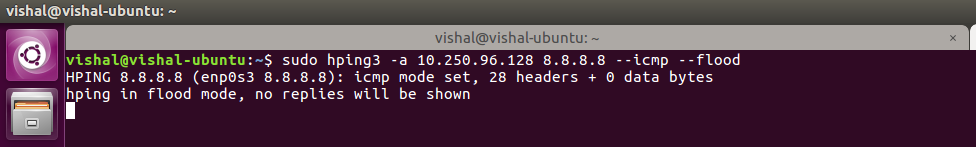
The Smurf attack is type of Ddos attack in which large numbers of ICMP packets with the intended victim's spoofed source IP are broadcast to a computer network using an IP broadcast address. Most devices on a network respond to this by sending a reply to the source IP address. If the number of machines on the network that receive and respond to these packets is very large, the victim's computer will be flooded with traffic. This can slow down the victim's computer to the point where it becomes impossible to work on.

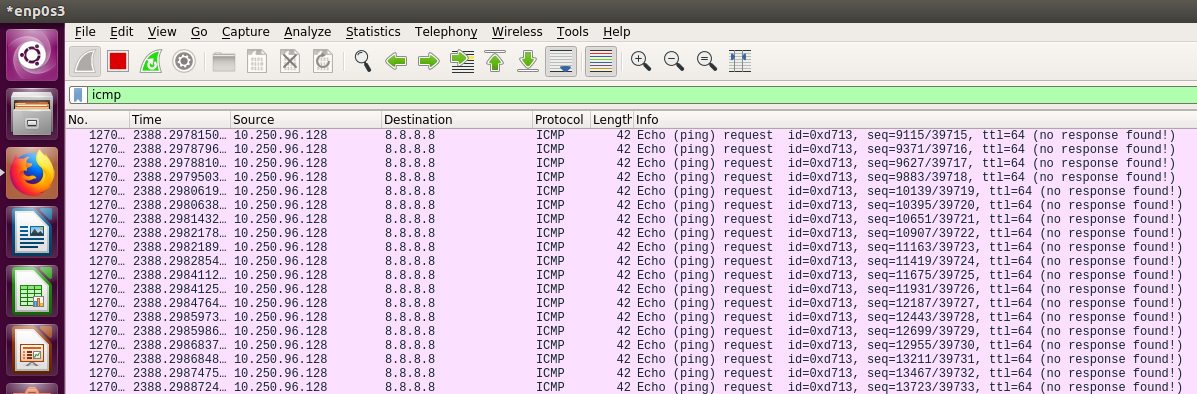
Here we will spoof source address of an ICMP packet to use an IP address of another system.

Original Address: 10.250.48.101

Spoofed Address: 10.250.96.128

Destination Address: 8.8.8.8



Wireshark

**ICMP Tunneling:**

An ICMP tunnel can establish a channel between the client and server such that a firewall will not be able to detect the connection. ICMP tunnelling is a connection between two endpoint host using ICMP echo requests and reply packets. So by using ICMP tunnelling, one can inject arbitrary data into an echo packet and send to a remote host. The remote host can inject data into another ICMP packet and sends it into the tunnel. This type of communication traffic can be undetectable for a proxy-based firewall. These mechanisms can be used to bypass firewalls’ rules.

**Teardrop attack:**

A Ddos attack carried out using ICMP packets. It is intended to crash the operating system of the victim host. It involves sending fragmentation packets to the victim host such that the victim host cannot reassemble them due to a bug in TCP/IP reassembly, packets overlap over one other resulting in operating system crash.

fragment offset field in the IP header, indicating the starting position of the data contained in a fragmented packet relative to the data in the original packet. If the sum of the offset and size of one fragmented packet differs from that of the next fragmented packet, the packets overlap. When such situation occurs, a server vulnerable to teardrop attacks is unable to reassemble the packets - resulting in a denial-of-service condition.

# **Contribution**

|  |  |
| --- | --- |
| Name | Contribution |
| Tejas Chumbalkar | * Documentation and report formatting * ICMP attacks |
| Vishal Govindraddi Yarabandi | * Wireshark observation * ICMP message format |
| Shivangi Gupta | * Documentation and report formatting * ICMP Architecture |
| Virat Mathur | * Set up lab in GNS3 * Working of ICMP |

**References**

* <http://www.networksorcery.com/enp/protocol/icmp/msg8.htm>
* <https://en.wikipedia.org/wiki/Hping>
* <http://www.informit.com/articles/article.aspx?p=26557&seqNum=5>
* <https://en.wikipedia.org/wiki/Smurf_attack>
* <https://security.radware.com/ddos-knowledge-center/ddospedia/teardrop-attack/>