

Informe Laboratorio III Redes de Datos

Arturo Mantinetti
Manuel Tobar
Diego Vilches
Nicolas Henriquez
`arturo.mantinetti@mail_udp.cl`
`manuel.tobar@mail_udp.cl`
`diego.vilches@mail_udp.cl`
`nicolas.henriquez@mail_udp.cl`

Profesor
Jaime Álvarez
Ayudante
Maximiliano Vega

10 de Abril de 2016

Índice general

1. Introducción	2
2. Contenido	3
2.1. Creación de Paquetes	3
2.2. Hardware utilizado	4
2.3. Envío de un paquete de datos a FF:FF:FF:FF:FF:FF	5
2.3.1. Switch	5
2.3.2. Hub	6
2.4. Envío de un paquete de datos con MAC específica	8
2.4.1. Switch	8
2.4.2. Hub	9
2.5. Envío de un paquete de datos con una MAC fuera de la red	10
2.5.1. Switch	10
2.5.2. Hub	11
3. Conclusión	12

1. Introducción

Este laboratorio consistió en crear paquetes de datos con diferentes parámetros para luego enviarlos por la red, con el fin lo lograr comprender como se conforman y comportan estos según sus características. Esto es posible gracias a un programa llamado 'Scapy' que nos da esas funcionalidades.

Los paquetes, en este experimento, varían principalmente en la dirección MAC, lo que hace que sean recibidos por distintos equipos. Para esto se ocupa 'Wireshark', programa con el que se puede capturar los paquetes enviados por la red. Una vez creados y enviados los paquetes a través del Switch, se repite el procedimiento, sólo que esta vez los equipos están conectados a un Hub.

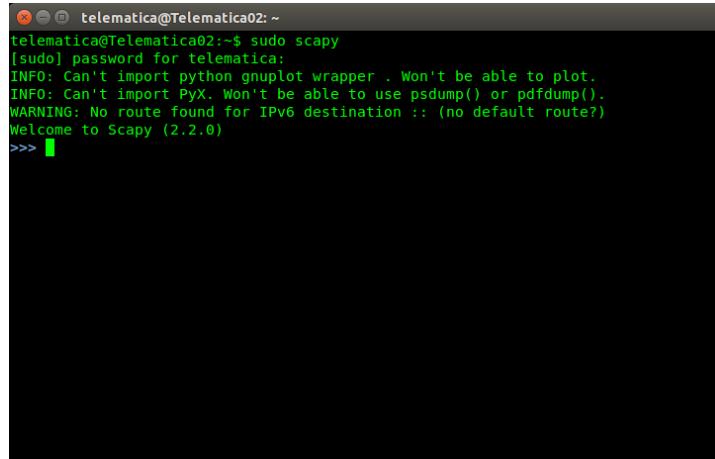
2. Contenido

2.1. Creación de Paquetes

Para crear un paquete con Scapy, este se tiene que ejecutar vía la consola el siguiente comando:

```
sudo scapy
```

Este comando iniciara el programa como superuser donde se podrá usar sus funciones para la creación de los paquetes. Estos se crean en base a las capas del modelo OSI, sin necesidad de seguir un orden específico al crear las capas, las cuales el programa permite su uso desde la Capa 2 hasta la que se necesite para el paquete.

A screenshot of a terminal window titled "telematica@Telematica02: ~". The window shows the command "sudo scapy" being run, followed by a password prompt "[sudo] password for telematica:". The output includes several informational messages: "INFO: Can't import python gnuplot wrapper . Won't be able to plot.", "INFO: Can't import PyX. Won't be able to use psdump() or pdfdump()", and "WARNING: No route found for IPv6 destination :: (no default route?)". The prompt ">>> [redacted]" is visible at the bottom of the window.

```
telematica@Telematica02: ~
[sudo] password for telematica:
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
>>> [redacted]
```

Iniciando con la Capa 2 esta el comando Ether(), este comando permite modificar los parámetros del enlace de datos, en especial las MACs de destino y origen, en este laboratorio se utiliza en demasía esta capa.

El siguiente comando, el cual se encarga de la Capa 3, es IP() el cual se encarga de los parámetros de enrutamiento incluyendo protocolos y direcciones lógicas del sistema, las direcciones de IP de origen y destino.

A continuación definimos ICMP(), o Internet Control Message Protocol, el cual es un protocolo como UDP o TCP, el cual se encarga de administrar la información relacionada con errores de los equipos en Red, este maneja mensajes de errores y control para los sistemas de la red, informando con ellos a la fuente original para que evite o corrija el problema detectado.

[...] ultimo comando a usar, el cual se encarga de la información a enviar, es Raw() este se tiene un String como parámetro para el envío de información a ser usada por el equipo de destino.

Una vez creado las capas a usar, con las capas que se estimen convenientes, estas son apiladas en orden ascendente separadas con un '/' para que estas formen un solo paquete que luego puede ser enviado, para el envío del paquete se utiliza el comando sendp()

2.2. Hardware utilizado

Para este laboratorio debimos utilizar una red montada con un Switch, para esto fue utilizado un Catalyst 2690 fabricado por Cisco System, y una red montada con un Hub, siendo este un AdvanceStack Switching Hub-12R fabricado por Hawlett Packard.



Los equipos conectados al Switch para realizar las pruebas fueron los equipos del laboratorio de Informática, mientras que los equipos que fueron utilizados para realizar las pruebas con el Hub fueron notebooks.

2.3. Envío de un paquete de datos a FF:FF:FF:FF:FF:FF

Creamos el paquete con la dirección MAC 'FF:FF:FF:FF:FF:FF', ante eso fijamos el valor en el campo de destino, 'dst', con los valores dado por el ejercicio. El resto de los campos son innecesarios para la actividad por lo cual los dejamos por defecto.

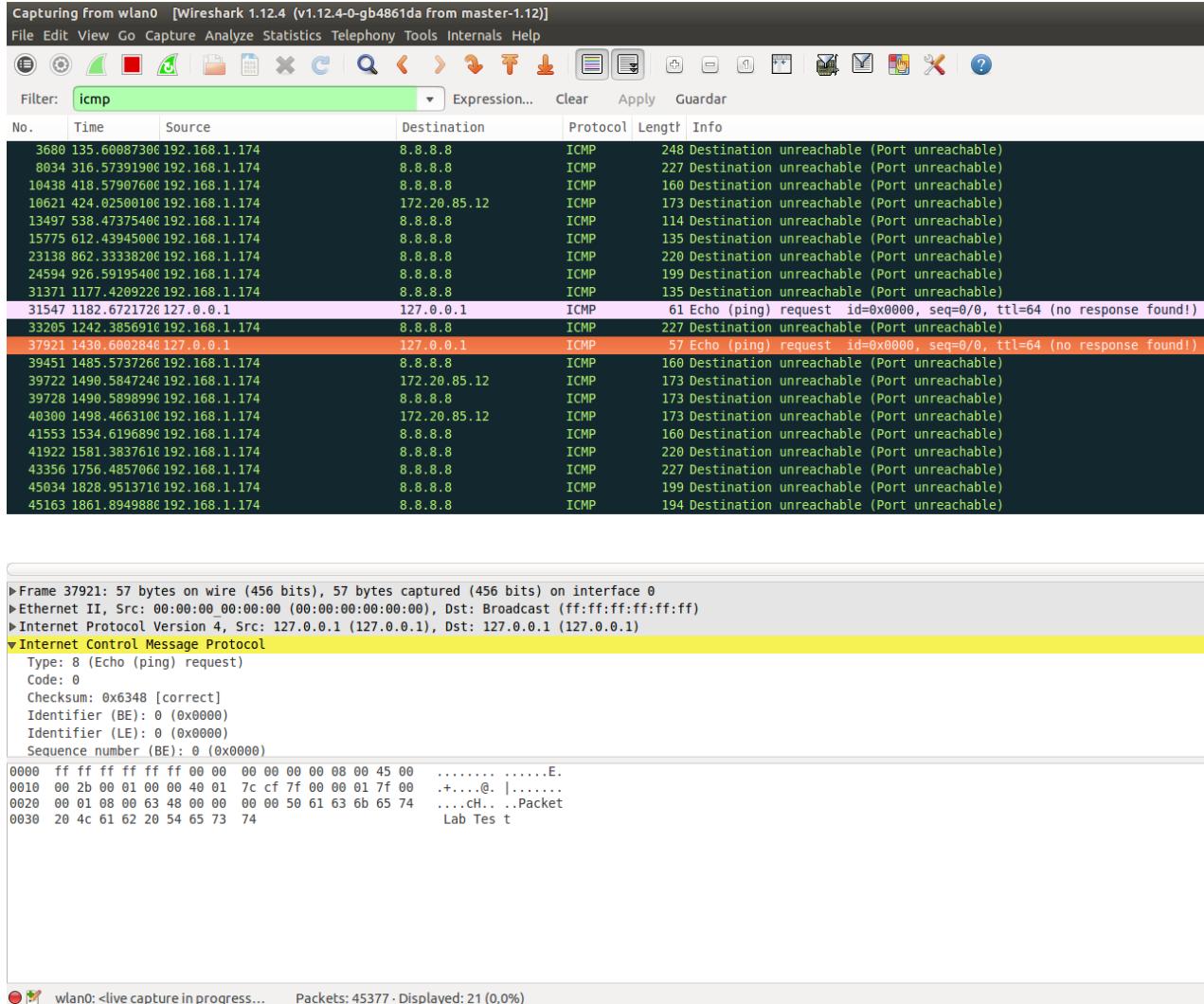
```
telematica@Telematica02:~$ sudo scapy
[sudo] password for telematica:
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> link
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'link' is not defined
>>> link=Ether()
>>> ls(link)
WARNING: Mac address to reach destination not found. Using broadcast.
dst      : DestMACField      = 'ff:ff:ff:ff:ff:ff' (None)
src      : SourceMACField    = '00:00:00:00:00:00' (None)
type     : XShortEnumField   = 0          (0)
>>> link.dst="FF:FF:FF:FF:FF:FF"
>>> ts(link)
dst      : DestMACField      = 'FF:FF:FF:FF:FF:FF' (None)
src      : SourceMACField    = '00:00:00:00:00:00' (None)
type     : XShortEnumField   = 0          (0)
>>> █
```

Al no necesitar ningún parámetro extra en las capas superiores solo las definimos, aunque esto no son necesarias para el funcionamiento del paquete . Solo modificamos el parámetro de la función Raw() para poder identificar el paquete que nosotros enviamos, una vez hecha la modificación a ese parámetro apilamos el paquete y procedemos el envío de este.

```
amantinetti@Manti-Buntu-Note:~$ sudo scapy
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> mac = Ether()
>>> mac.dst="FF:FF:FF:FF:FF:FF"
>>> ip=IP()
>>> raw=Raw()
>>> icmp=ICMP()
>>> ls(raw)
load    : StrField           = ''          ('')
>>> raw.load="Packet Lab Test"
>>> paquete=mac/ip/icmp/raw
>>> sendp(paquete)
.
Sent 1 packets.
>>> ~█
```

2.3.1. Switch

Al ser la dirección MAC 'FF:FF:FF:FF:FF:FF' el Switch no reenvía el paquete a ningún equipo que se encuentre dentro de la red. Adicionalmente a esto si estamos usando Wireshark en modo Promiscuo podemos capturar el paquete se a enviado.



Capturing from wlan0 [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: icmp

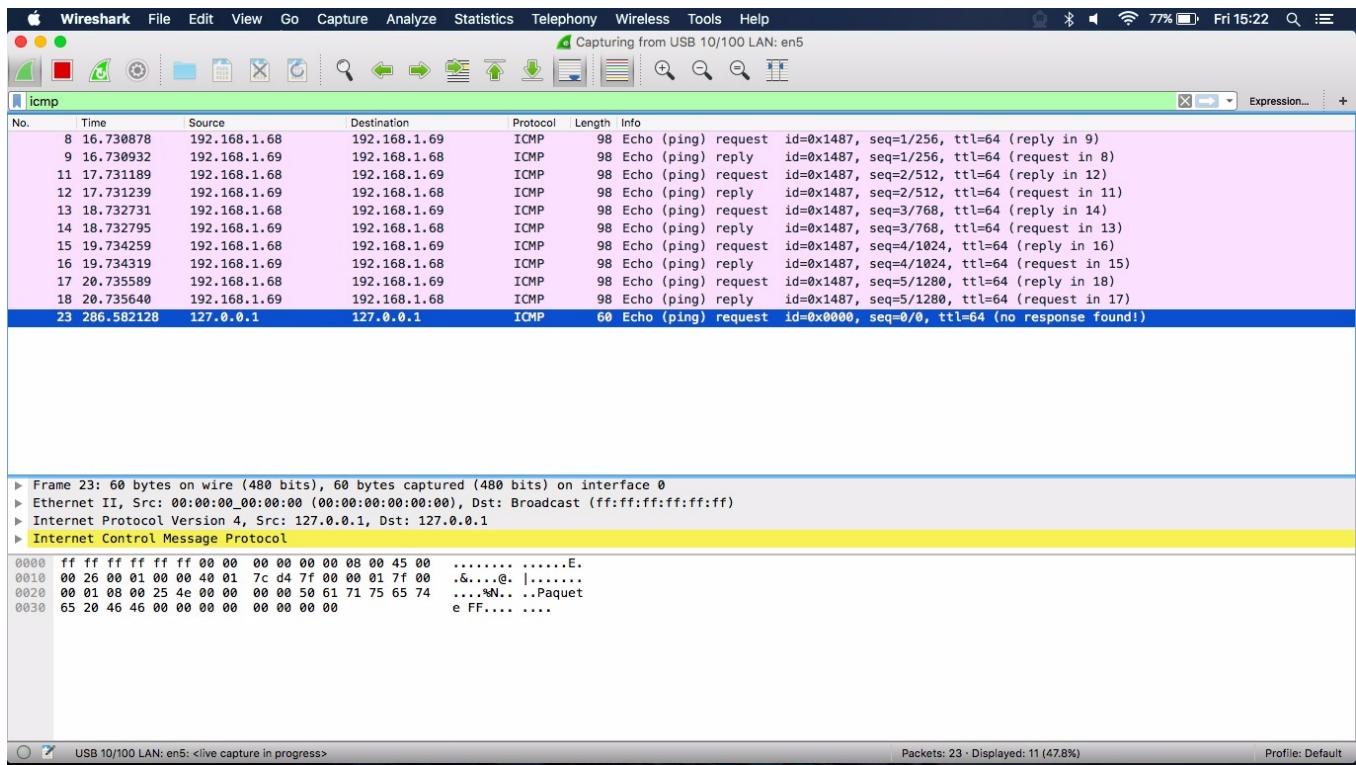
No.	Time	Source	Destination	Protocol	Length	Info
3680	135.600873000	192.168.1.174	8.8.8.8	ICMP	248	Destination unreachable (Port unreachable)
8034	316.573919000	192.168.1.174	8.8.8.8	ICMP	227	Destination unreachable (Port unreachable)
16438	418.579076000	192.168.1.174	8.8.8.8	ICMP	166	Destination unreachable (Port unreachable)
16621	424.025001000	192.168.1.174	172.20.85.12	ICMP	173	Destination unreachable (Port unreachable)
13497	538.473754000	192.168.1.174	8.8.8.8	ICMP	114	Destination unreachable (Port unreachable)
15775	612.439450000	192.168.1.174	8.8.8.8	ICMP	135	Destination unreachable (Port unreachable)
23138	862.333382000	192.168.1.174	8.8.8.8	ICMP	220	Destination unreachable (Port unreachable)
24594	926.591954000	192.168.1.174	8.8.8.8	ICMP	199	Destination unreachable (Port unreachable)
31371	1177.420922600	192.168.1.174	8.8.8.8	ICMP	135	Destination unreachable (Port unreachable)
31547	1182.672172600	127.0.0.1	127.0.0.1	ICMP	61	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
33205	1242.385691600	192.168.1.174	8.8.8.8	ICMP	227	Destination unreachable (Port unreachable)
37921	1430.660284600	127.0.0.1	127.0.0.1	ICMP	57	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response found!)
39451	1485.573726000	192.168.1.174	8.8.8.8	ICMP	166	Destination unreachable (Port unreachable)
39722	1490.584724600	192.168.1.174	172.20.85.12	ICMP	173	Destination unreachable (Port unreachable)
39728	1490.589899600	192.168.1.174	8.8.8.8	ICMP	173	Destination unreachable (Port unreachable)
40300	1498.466310600	192.168.1.174	172.20.85.12	ICMP	173	Destination unreachable (Port unreachable)
41553	1534.619689600	192.168.1.174	8.8.8.8	ICMP	160	Destination unreachable (Port unreachable)
41922	1581.383761600	192.168.1.174	8.8.8.8	ICMP	220	Destination unreachable (Port unreachable)
43356	1756.485706600	192.168.1.174	8.8.8.8	ICMP	227	Destination unreachable (Port unreachable)
45034	1828.951371600	192.168.1.174	8.8.8.8	ICMP	199	Destination unreachable (Port unreachable)
45163	1861.894988600	192.168.1.174	8.8.8.8	ICMP	194	Destination unreachable (Port unreachable)

► Frame 37921: 57 bytes on wire (456 bits), 57 bytes captured (456 bits) on interface 0
► Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
► Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▼ Internet Control Message Protocol
 Type: 8 (Echo (ping) request)
 Code: 0
 Checksum: 0x6348 [correct]
 Identifier (BE): 0 (0x0000)
 Identifier (LE): 0 (0x0000)
 Sequence number (BE): 0 (0x0000)
0000 ff ff ff ff ff ff 00 00 00 00 00 00 00 45 00E.
0010 00 2b 00 01 00 00 40 01 7c cf 7f 00 00 01 7f 00 .+....@. |.....
0020 00 01 08 00 63 48 00 00 00 00 50 61 63 6b 65 74CH.. .Packet
0030 20 4c 61 62 20 54 65 73 74 Lab Tes t

wlan0: <live capture in progress... Packets: 45377 · Displayed: 21 (0,0%)

2.3.2. Hub

2.3. ENVÍO DE UN PAQUETE DE DATOS A FF:FF:FF:FF:FF:FF



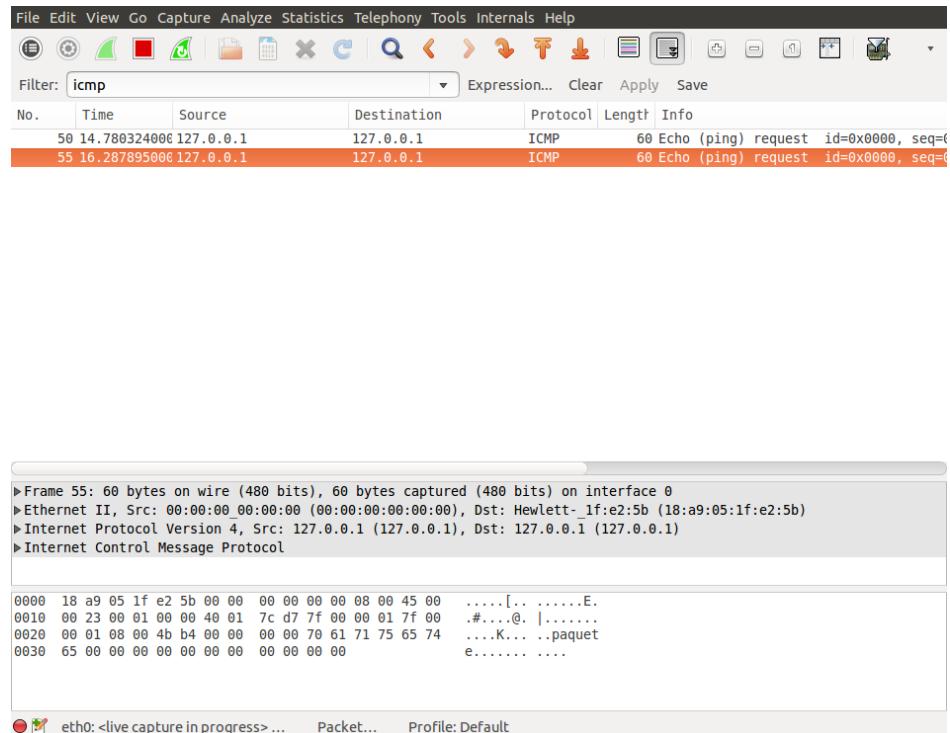
2.4. Envío de un paquete de datos con MAC específica

Para esta segunda actividad, esta vez usamos la siguiente dirección MAC '18:A9:05:1F:E2:5B' que pertenece a un equipo dentro de la red LAN, al igual que en la actividad anterior no necesitamos usar parámetros adicionales de las capas superiores a excepción de la ultima capa que la usamos para identificar fácilmente nuestro paquete.

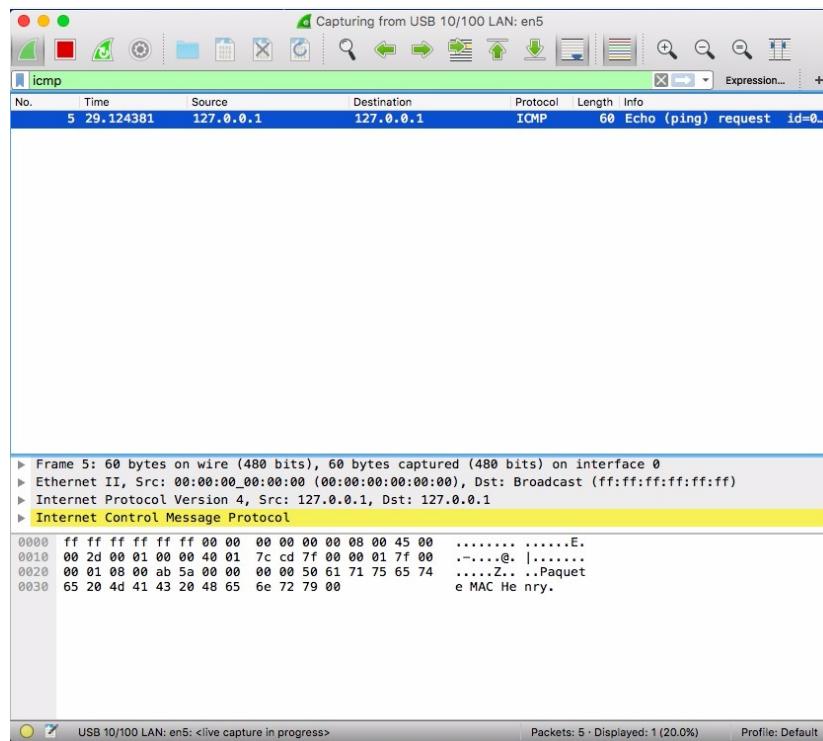
```
telematica@Telematica03:~$ sudo scapy
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> enlace=Ether()
>>> ls(enlace)
WARNING: Mac address to reach destination not found. Using broadcast.
dst      : DestMACField      = 'ff:ff:ff:ff:ff:ff' (None)
src      : SourceMACField    = '00:00:00:00:00:00' (None)
type     : XshortEnumField   = 0           (0)
>>> enlace.dst='18:a9:05:1f:e2:5b'
>>> ip=IP()
>>> icmp=ICMP()
>>> raw=Raw()
>>> raw.load='paquete'
>>> paquete=enlace/ip/icmp/raw
>>> sendp(paquete)
.
Sent 1 packets.
>>> █
```

2.4.1. Switch

Luego buscamos nuestro paquete en Wireshark para ver si se envió correctamente y llegó a destino.



2.4.2. Hub



2.5. Envió de un paquete de datos con una MAC fuera de la red

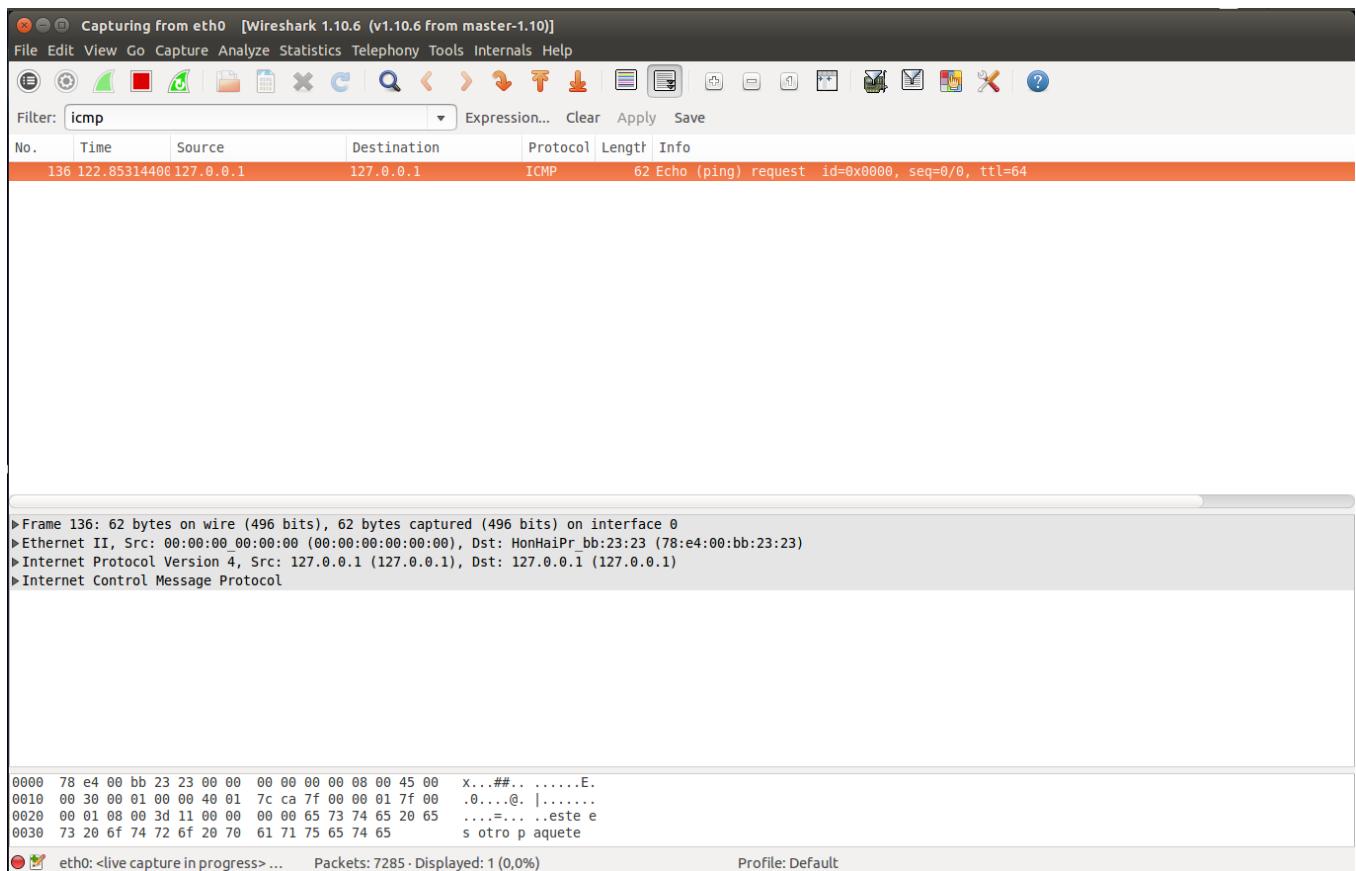
En esta ocasión se utilizo una dirección MAC de destino escrita al azar que no coincidiera con la de ninguno de los equipos pertenecientes a la red LAN, esta dirección fue '78:E4:00:BB:23:23', correspondiente a un notebook perteneciente a uno de los miembros del grupo. Luego se creó un paquete tras haber agregado un mensaje en la última capa, sin ningún cambio adicional el paquete fue enviado a la dirección ya mencionada.

```
File "/usr/lib/python2.7/dist-packages/scapy/utils.py", line 244, in <lambda>
    return "".join(map(lambda x: chr(int(x,16)), mac.split(":")))
ValueError: invalid literal for int() with base 16: 'lf'
>>> clear()
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'clear' is not defined
>>> clear
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'clear' is not defined
>>>
telematica@Telematica02:~$ clear

telematica@Telematica02:~$ sudo scapy
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> link3=Ether()
>>> link3.dst="78:e4:00:bb:23:23"
>>> ls(link3)
dst      : DestMACField      = '78:e4:00:bb:23:23' (None)
src      : SourceMACField   = '00:00:00:00:00:00' (None)
type     : XShortEnumField  = 0          (0)
>>> ip3=IP()
>>> icmp3=ICMP()
>>> rawlin3=Raw()
>>> rawlin3.load="este es otro paquete"
>>> packet=link3/ip3/icmp3/rawlin3
>>> sendp(packet)
.
Sent 1 packets.
>>> █
```

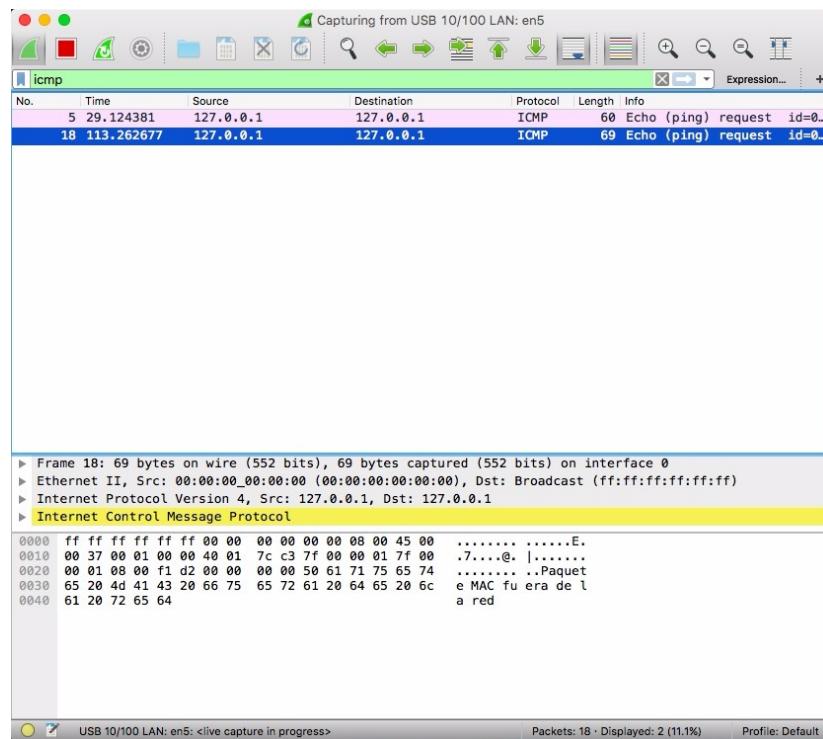
2.5. ENVIÓ DE UN PAQUETE DE DATOS CON UNA MAC FUERA DE LA RED

2.5.1. Switch



2.5. ENVIÓ DE UN PAQUETE DE DATOS CON UNA MAC FUERA DE LA RED

2.5.2. Hub



3. Conclusión