



**udp** UNIVERSIDAD  
DIEGO PORTALES

Facultad de Ingeniería  
Escuela de Informática y Telecomunicaciones

---

## 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

<b>1. Introducción</b>	<b>2</b>
<b>2. Contenido</b>	<b>3</b>
2.1. Creación de Paquetes . . . . .	3
2.2. Switch . . . . .	4
2.2.1. Envío de un paquete de datos a FF:FF:FF:FF:FF:FF . . . . .	4
2.2.2. Envío de un paquete de datos con MAC específica . . . . .	5
2.2.3. Envío de un paquete de datos con una MAC fuera de la red . . . . .	5
2.3. Hub . . . . .	6
2.3.1. Envío de un paquete de datos a FF:FF:FF:FF:FF:FF . . . . .	6
2.3.2. Envío de un paquete de datos con MAC específica . . . . .	6
2.3.3. Envío de un paquete de datos con una MAC fuera de la red . . . . .	6
<b>3. Conclusión</b>	<b>7</b>

# 1. Introducción

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

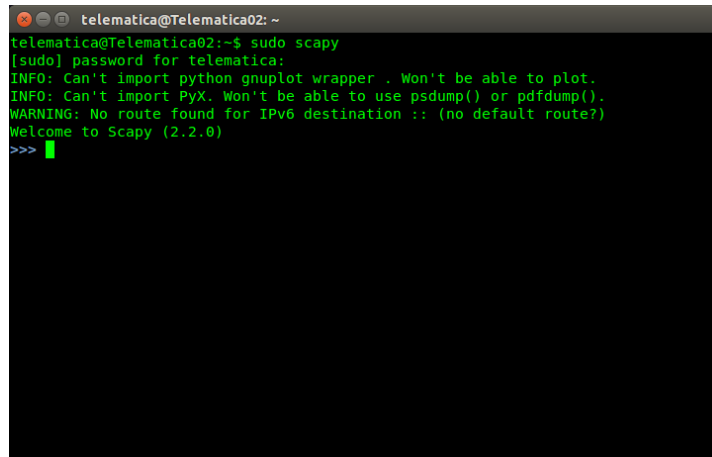
Los paquetes, en este experimento, varían principalmente en la dirección MAC, lo que hace que sean recibidos por distintos equipos. Para esto se ocupa 'Wireshark', programa con el que se puede capturar los paquetes enviados por la red. Una vez creados y enviados los paquetes a través del Switch, se repite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

Este comando iniciara el programa 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.



```
telematica@Telematica02: ~  
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)  
>>>
```

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.

[...] 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. Switch

### 2.2.1. 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' ...

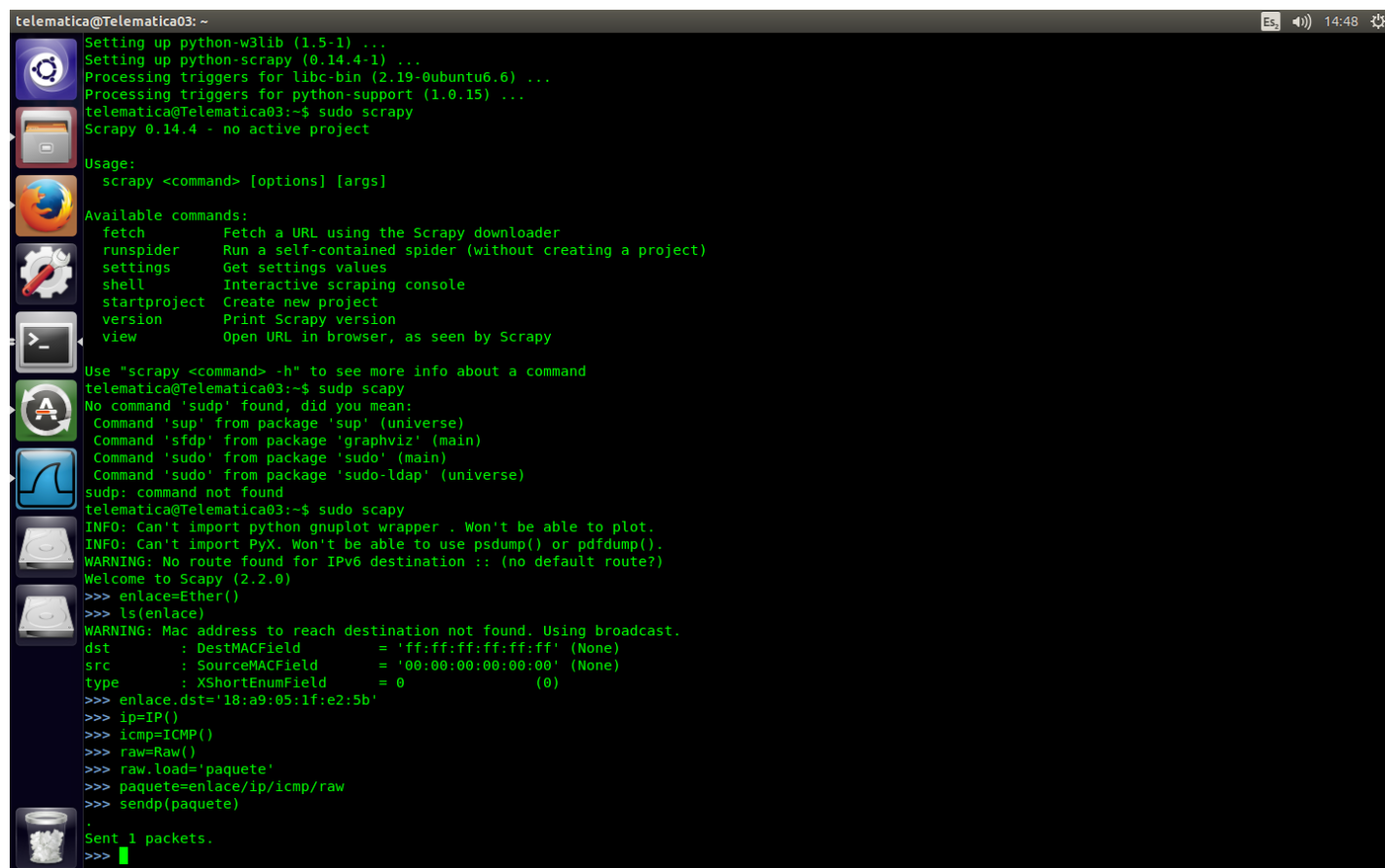
```
telematica@Telematica02:~$ sudo scapy
[sudo] password for telematica:
INFO: Can't import python gnuplot wrapper . Won't be able to plot.
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)
Welcome to Scapy (2.2.0)
>>> 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"
>>> ls(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/instanciamos, aunque esto no es necesario 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.

```
Terminal telematica@Telematica02: ~
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"
>>> ls(link)
dst      : DestMACField      = 'FF:FF:FF:FF:FF:FF' (None)
src      : SourceMACField    = '00:00:00:00:00:00' (None)
type     : XShortEnumField   = 0                (0)
>>> ip=IP()
>>> icmp=ICMP()
>>> rawlin=raw()
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'raw' is not defined
>>> rawlin=Raw()
>>> rawlin.load="este es un paquete a FF:FF:FF:FF:FF:FF"
>>> packet=link/ip/icmp/rawlin
>>> sendp(packet)
Sent 1 packets.
```

### 2.2.2. Envío de un paquete de datos con MAC específica

Para este segundo experimento, 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 el experimento anterior no necesitamos usar parametros adicionales de las capas superiores a excepción de la ultima capa que la usamos para identificar facilmente nuestro paquete.



```

telematica@Telematica03: ~
Setting up python-w3lib (1.5-1) ...
Setting up python-scrapy (0.14.4-1) ...
Processing triggers for libc-bin (2.19-0ubuntu6.6) ...
Processing triggers for python-support (1.0.15) ...
telematica@Telematica03:~$ sudo scrapy
Scrapy 0.14.4 - no active project

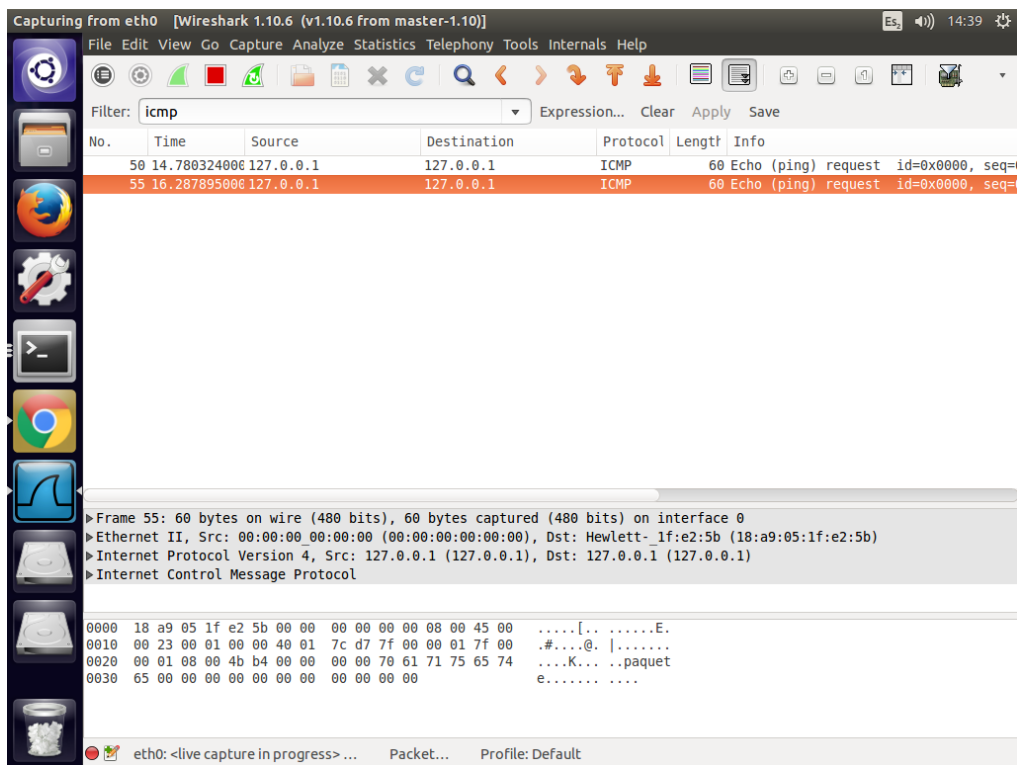
Usage:
  scrapy <command> [options] [args]

Available commands:
  fetch          Fetch a URL using the Scrapy downloader
  runspider      Run a self-contained spider (without creating a project)
  settings       Get settings values
  shell          Interactive scraping console
  startproject   Create new project
  version        Print Scrapy version
  view           Open URL in browser, as seen by Scrapy

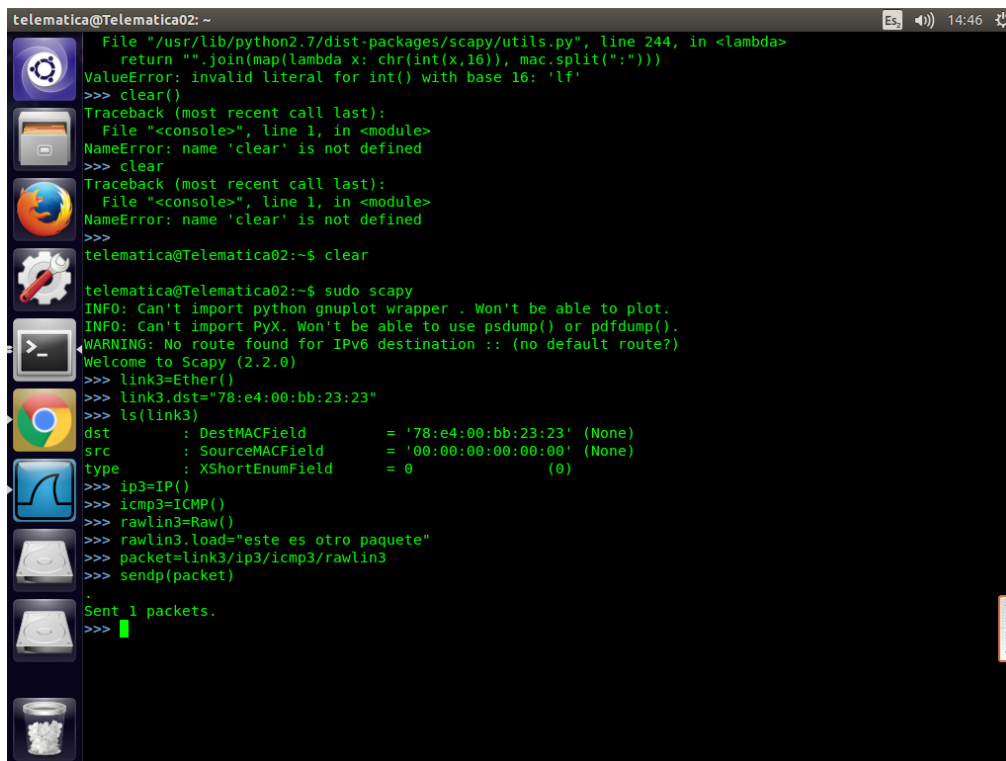
Use "scrapy <command> -h" to see more info about a command
telematica@Telematica03:~$ sudo scapy
No command 'sudp' found, did you mean:
  Command 'sup' from package 'sup' (universe)
  Command 'sfdp' from package 'graphviz' (main)
  Command 'sudo' from package 'sudo' (main)
  Command 'sudo' from package 'sudo-ldap' (universe)
sudp: command not found
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)
>>> enlance=Ether()
>>> ls(enlance)
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)
>>> enlance.dst='18:a9:05:1f:e2:5b'
>>> ip=IP()
>>> icmp=ICMP()
>>> raw=Raw()
>>> raw.load='paquete'
>>> paquete=enlance/ip/icmp/raw
>>> sendp(paquete)
.
Sent 1 packets.
>>> █

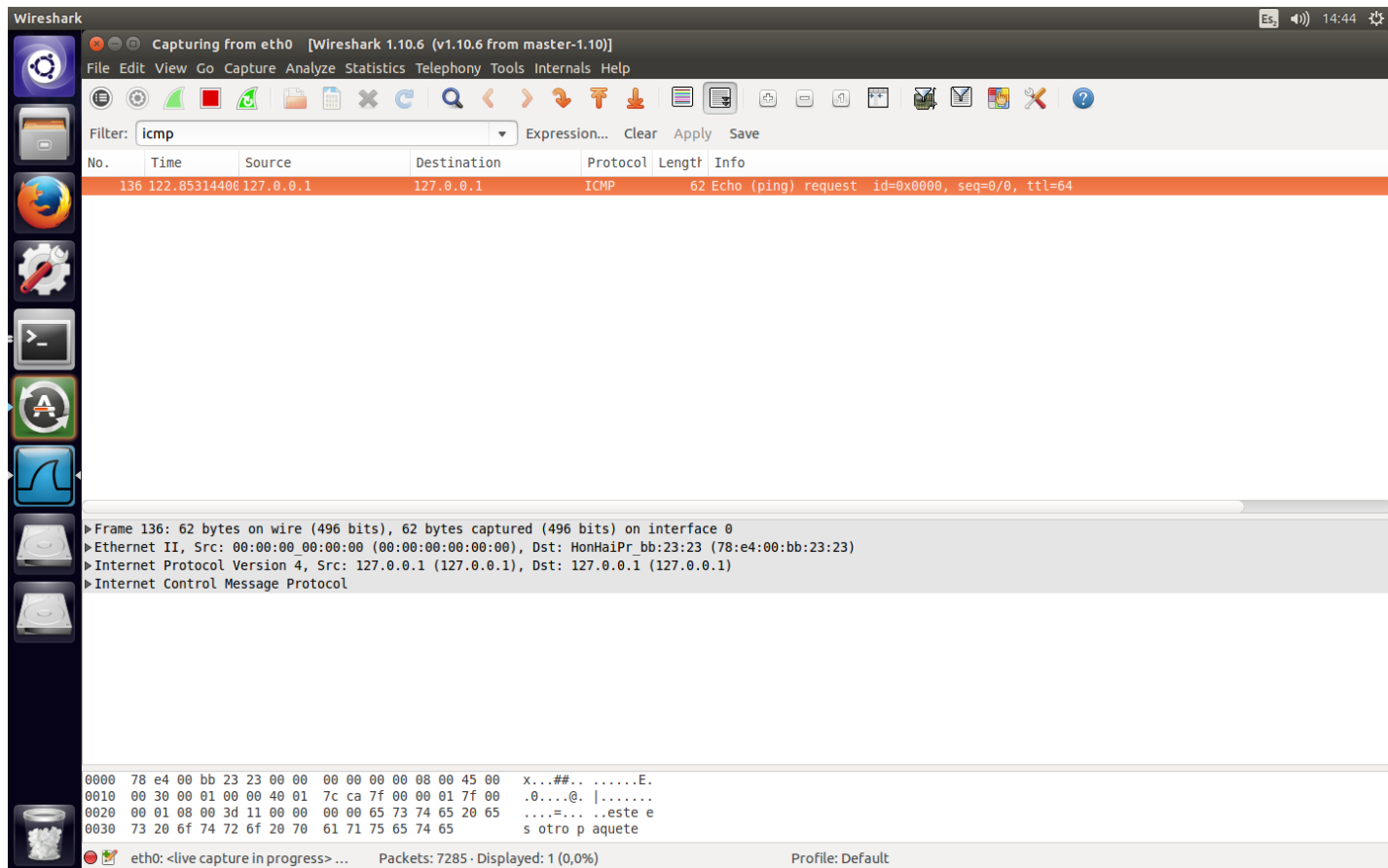
```

Luego buscamos nuestro paquete en Wireshark para ver si se envio correctamente y llego a destino.



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







## 2.3. Hub

- 2.3.1. Envío de un paquete de datos a FF:FF:FF:FF:FF:FF
- 2.3.2. Envío de un paquete de datos con MAC específica
- 2.3.3. Envío de un paquete de datos con una MAC fuera de la red

### **3. Conclusión**