

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	5
2.4. Envío de un paquete de datos con MAC específica	5
2.4.1. Switch	5
2.4.2. Hub	6
2.5. Envío de un paquete de datos con una MAC fuera de la red	6
2.5.1. Switch	6
2.5.2. Hub	8
3. Conclusión	9

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.

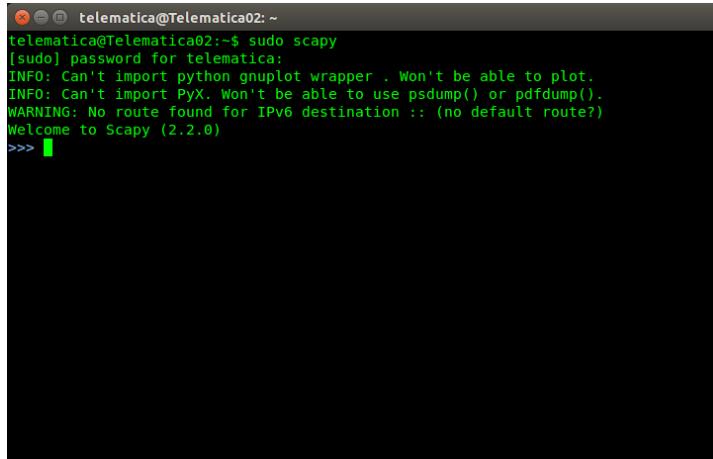
2. Contenido

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 super-user donde se podrá usar sus funciones para la creación de los paquetes y enviarlos por la red. 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 several informational messages from 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()", and "WARNING: No route found for IPv6 destination :: (no default route?)". Below these messages, the text "Welcome to Scapy (2.2.0)" is displayed, followed by a prompt ">>>". The rest of the terminal window is blank, indicating a new line of input.

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' ...

```
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.

```
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.
```

2.3.1. 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.

2.3.2. Hub

2.4. Envío de un paquete de datos con MAC específica

2.4.1. Switch

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

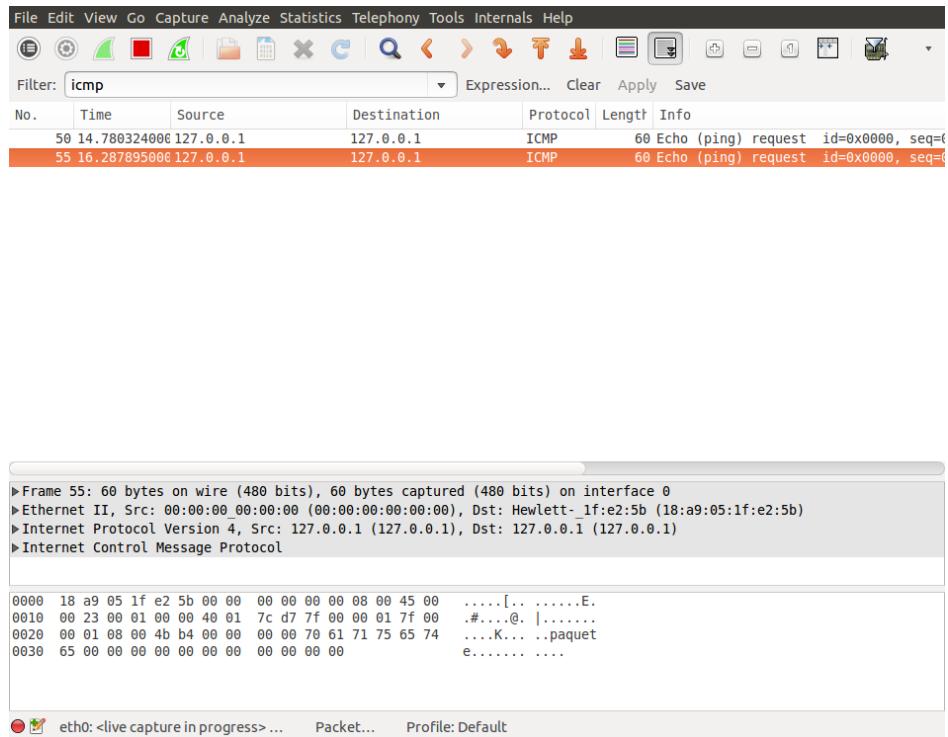
parámetros adicionales de las capas superiores a excepción de la ultima capa que la usamos para identificar fácilmente nuestro paquete.

```

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()
>>> raw=Raw()
>>> raw.load='paquete'
>>> paquete=enlace/ip/icmp/raw
>>> sendp(paquete)
.
Sent 1 packets.
>>> █

```

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



2.4.2. Hub

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

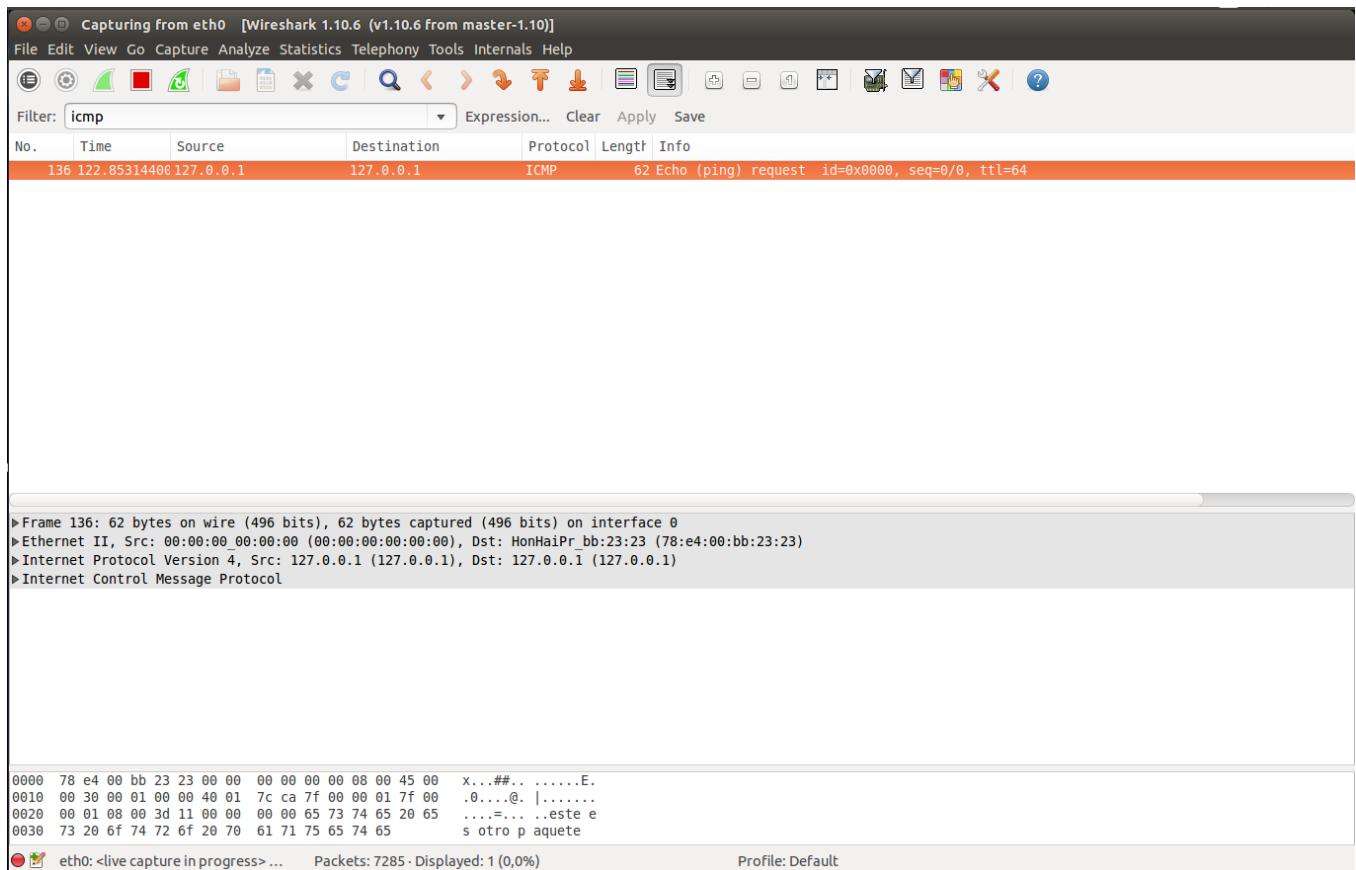
2.5.1. Switch

En esta ocasión se utilizó 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'. 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.

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

```
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.
>>> █
```



2.5.2. Hub

3. Conclusión