



UNIVERSIDAD DE GRANADA

GRADO INGENIERÍA INFORMÁTICA (2017 – 2018)

FUNDAMENTOS DE REDES

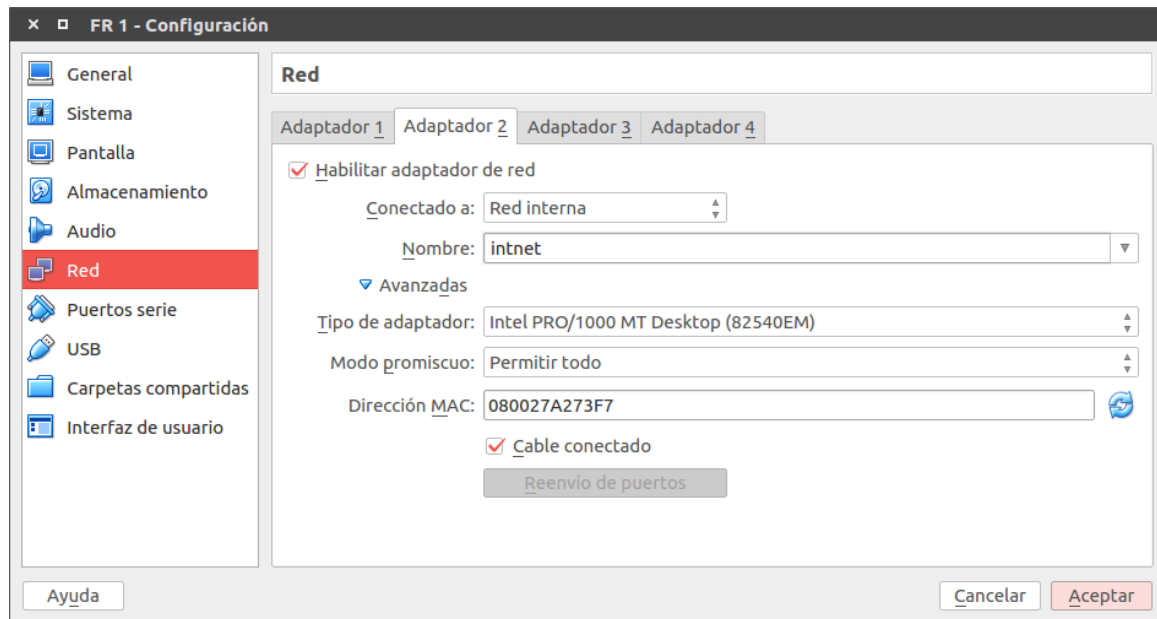
Seminario 1: Wireshark + ping

Trabajo realizado por Antonio Miguel Morillo Chica

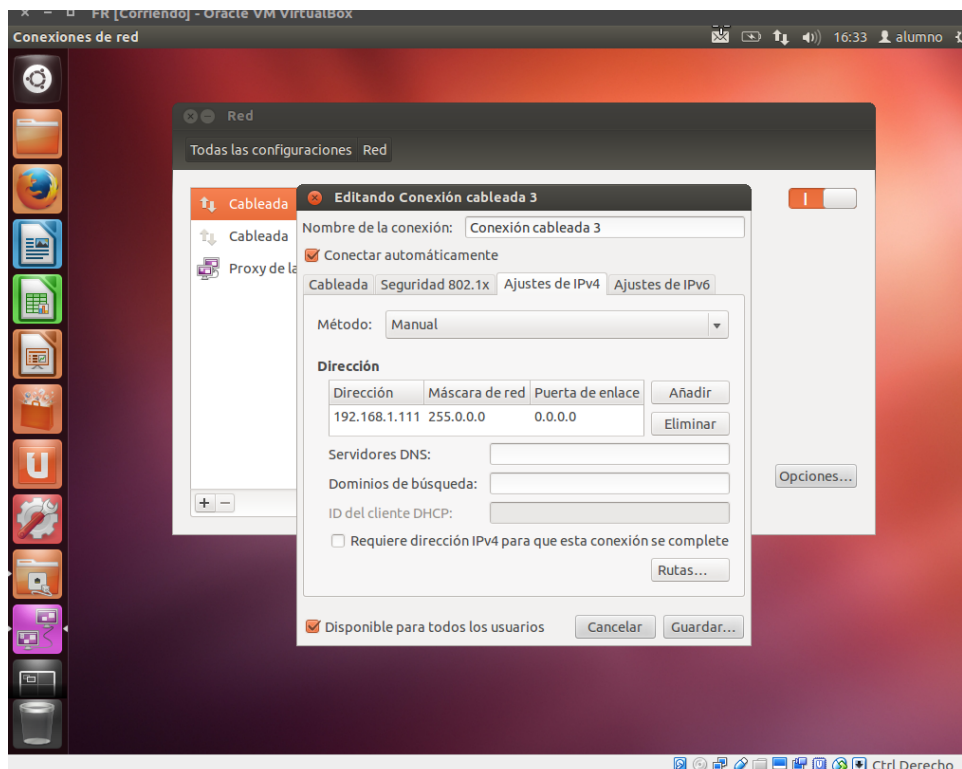
1. Configuración de red.

En primer lugar lo que he hecho ha sido usar la maquina virtual (.ova) aportada por el profesor. Posteriormente se clona restableciendo siempre la dirección MAC.

Antes de iniciarlas hay que añadirles un nuevo adaptador que será de tipo red interna, como se ve en la imagen.



Posteriormente iniciamos las máquinas, en mi caso FE y FR 1, y establecemos una IP estática para poder hacer un ping fácilmente a la misma dirección siempre. Vamos a ajustes, red y buscamos la conexión 3 ya que la 1 es lo y la 2 la NAT. He seleccionado en ajustes IPv4 como manual y en añadir he añadido la siguiente IP, se puede ver en la imagen:



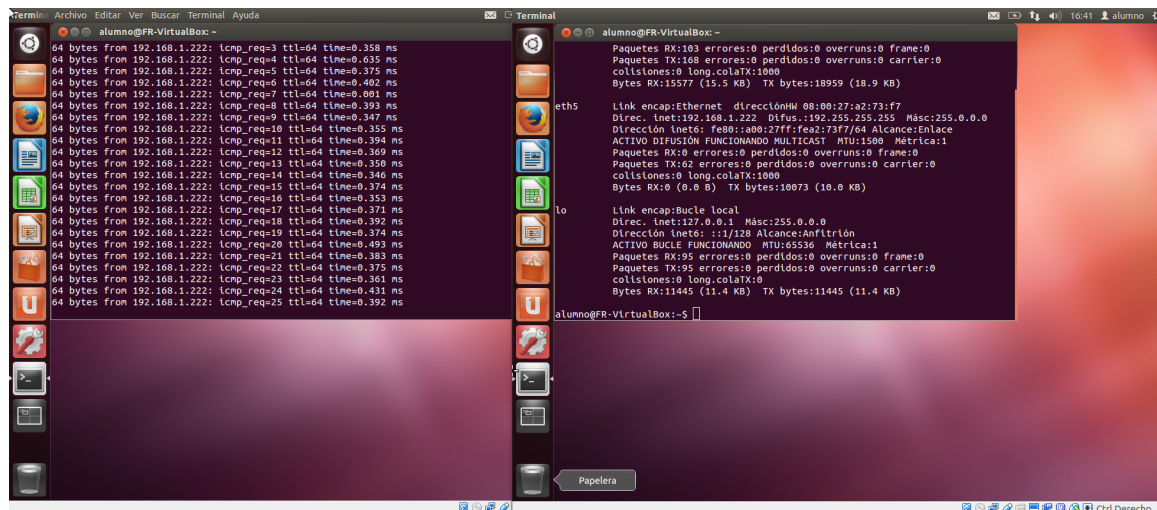
Tras esto solo hay que reiniciar el servicio de red con: `$ sudo /etc/init.d/network-manager restart`.

2. Ping entre máquinas.

Una vez configuradas las máquinas la primera tendrá en mi caso la dirección 192,168,1,111 y la máquina segunda la: 192,168,1,222. Que podremos comprobar con el comando: `$ ifconfig`.



Para hacer el ping por ejemplo, desde la máquina primera a la segunda lo que haremos será desde la primera terminal: `$ ping 192,168,1,111`, como podemos ver funciona:



```
alumno@FR-VirtualBox:~$ ping 192.168.1.222
64 bytes from 192.168.1.222: icmp_req=3 ttl=64 time=0.350 ms
64 bytes from 192.168.1.222: icmp_req=4 ttl=64 time=0.635 ms
64 bytes from 192.168.1.222: icmp_req=5 ttl=64 time=0.375 ms
64 bytes from 192.168.1.222: icmp_req=6 ttl=64 time=0.402 ms
64 bytes from 192.168.1.222: icmp_req=7 ttl=64 time=0.091 ms
64 bytes from 192.168.1.222: icmp_req=8 ttl=64 time=0.393 ms
64 bytes from 192.168.1.222: icmp_req=9 ttl=64 time=0.347 ms
64 bytes from 192.168.1.222: icmp_req=10 ttl=64 time=0.355 ms
64 bytes from 192.168.1.222: icmp_req=11 ttl=64 time=0.394 ms
64 bytes from 192.168.1.222: icmp_req=12 ttl=64 time=0.369 ms
64 bytes from 192.168.1.222: icmp_req=13 ttl=64 time=0.350 ms
64 bytes from 192.168.1.222: icmp_req=14 ttl=64 time=0.346 ms
64 bytes from 192.168.1.222: icmp_req=15 ttl=64 time=0.374 ms
64 bytes from 192.168.1.222: icmp_req=16 ttl=64 time=0.353 ms
64 bytes from 192.168.1.222: icmp_req=17 ttl=64 time=0.371 ms
64 bytes from 192.168.1.222: icmp_req=18 ttl=64 time=0.392 ms
64 bytes from 192.168.1.222: icmp_req=19 ttl=64 time=0.374 ms
64 bytes from 192.168.1.222: icmp_req=20 ttl=64 time=0.493 ms
64 bytes from 192.168.1.222: icmp_req=21 ttl=64 time=0.383 ms
64 bytes from 192.168.1.222: icmp_req=22 ttl=64 time=0.375 ms
64 bytes from 192.168.1.222: icmp_req=23 ttl=64 time=0.361 ms
64 bytes from 192.168.1.222: icmp_req=24 ttl=64 time=0.431 ms
64 bytes from 192.168.1.222: icmp_req=25 ttl=64 time=0.392 ms

alumno@FR-VirtualBox:~$ ifconfig eth5
Paquetes RX:103 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:168 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colaTX:1000
Bytes RX:15377 (15.3 KB) TX bytes:18959 (18.9 KB)

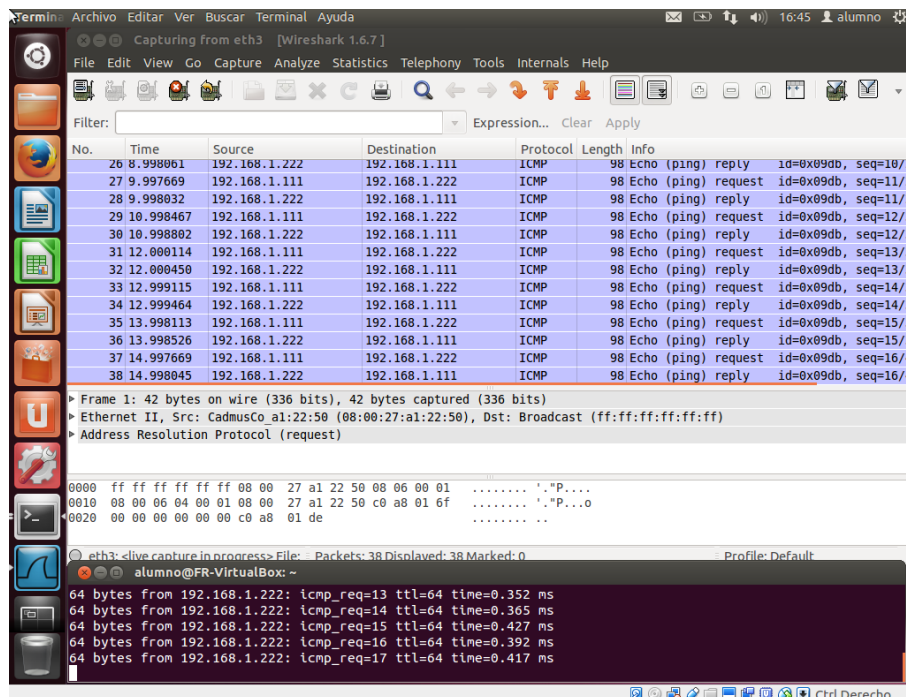
Link encap:Ethernet direcciónHW 08:00:27:a2:73:f7
Direc. inet:192.168.1.222 Difus.:192.255.255.255 Masc:255.0.0.0
Dirección inet6: fe80:a00:27ff:fe27:73f7:64 Alcance:Enlace
ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Metrica:1
Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:62 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colaTX:1000
Bytes RX:0 (0.0 B) TX bytes:10073 (10.0 KB)

alumno@FR-VirtualBox:~$ ifconfig lo
Link encap:Bucle local
Direc. inet:127.0.0.1 Masc:255.0.0.0
Dirección inet6: ::1/128 Alcance:Anfitrión
ACTIVO BUCLE FUNCIONANDO MTU:65536 Metrica:1
Paquetes RX:95 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:95 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colaTX:0
Bytes RX:11445 (11.4 KB) TX bytes:11445 (11.4 KB)

alumno@FR-VirtualBox:~$
```

3. Usando Wireshark.

Para que funcione bien wireshark hay que levantar el servicio desde la terminal con sudo o siendo administrador. Tras esto seleccionas la interfaz de red que haga referencia a la res interna. Automáticamente podremos capturar paquetes con los que haremos el ping y veremos como se interceptan los paquetes:



```
alumno@FR-VirtualBox:~$ sudo wireshark
Capturing from eth3 [Wireshark 1.6.7]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Length Info
26 8.998061 192.168.1.222 192.168.1.111 ICMP 98 Echo (ping) reply id=0x09db, seq=10/2
27 9.997669 192.168.1.111 192.168.1.222 ICMP 98 Echo (ping) request id=0x09db, seq=11/2
28 9.998032 192.168.1.222 192.168.1.111 ICMP 98 Echo (ping) reply id=0x09db, seq=11/2
29 10.998467 192.168.1.111 192.168.1.222 ICMP 98 Echo (ping) request id=0x09db, seq=12/3
30 10.998802 192.168.1.222 192.168.1.111 ICMP 98 Echo (ping) reply id=0x09db, seq=12/3
31 12.000114 192.168.1.111 192.168.1.222 ICMP 98 Echo (ping) request id=0x09db, seq=13/3
32 12.000450 192.168.1.222 192.168.1.111 ICMP 98 Echo (ping) reply id=0x09db, seq=13/3
33 12.999115 192.168.1.111 192.168.1.222 ICMP 98 Echo (ping) request id=0x09db, seq=14/3
34 12.999464 192.168.1.222 192.168.1.111 ICMP 98 Echo (ping) reply id=0x09db, seq=14/3
35 13.998113 192.168.1.111 192.168.1.222 ICMP 98 Echo (ping) request id=0x09db, seq=15/3
36 13.998526 192.168.1.222 192.168.1.111 ICMP 98 Echo (ping) reply id=0x09db, seq=15/3
37 14.997669 192.168.1.111 192.168.1.222 ICMP 98 Echo (ping) request id=0x09db, seq=16/4
38 14.998045 192.168.1.222 192.168.1.111 ICMP 98 Echo (ping) reply id=0x09db, seq=16/4

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on eth3
Ethernet II, Src: CadmusCo_a1:22:50 (08:00:27:a1:22:50), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)

0000 ff ff ff ff ff 08 00 27 a1 22 50 08 06 00 01 ..... ".P....
0010 08 00 06 04 00 01 08 00 27 a1 22 50 c0 a8 01 6f ..... ".P...o
0020 00 00 00 00 00 00 c0 a8 01 de ..... ..

eth3: alive capture in progress - Filter: Packets: 38 Displayed: 38 Marked: 0
alumno@FR-VirtualBox:~$
```

Como podemos ver por cada request hay un reply es una especie de llamada cliente servidor donde la máquina 1 demanda una llamada a la máquina 2 la cual le responde. El protocolo usado es el ICMP para los reply y request.