

Universidad Nacional Autónoma de Nicaragua
UNAN - León
Facultad de Ciencias y Tecnologías



Practica 2

Componente:

o **Redes de Computadoras**

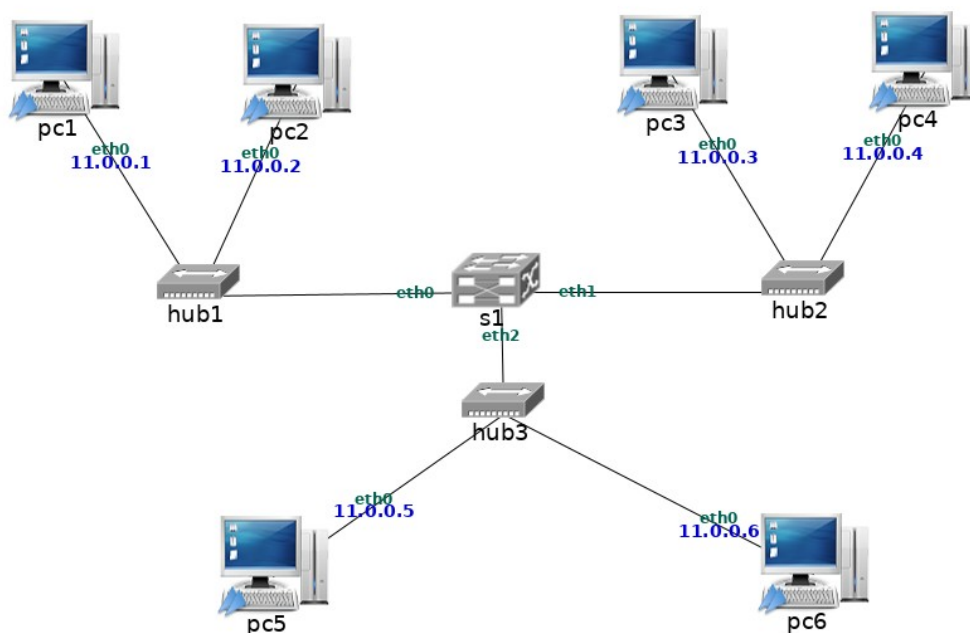
Integrante:

➤ **Bismarck Antonio Berrios Lopez**

1. Funcionamiento de hubs y switch

En el fichero lab-hub-switch.tgz está definida una red como la de la figura 1. Descomprime el fichero (con `tar -xvzf lab-hub-switch.tgz`), arranca NetGUI y abre el escenario. No arranques aún s1.

Arranca el resto de las máquinas de una en una.



Deja por ahora sin arrancar el switch s1. Cada uno de los hubs estará aislado de los demás. Por lo tanto sólo habrá conectividad entre los ordenadores que están conectados al mismo hub. Las tramas Ethernet no pueden salir del hub en el que aparecen.

NOTA: Las capturas a realizar en este apartado no es necesario redirigirlas a un fichero para estudiarlas con wireshark. Basta con ver la salida de `tcpdump` directamente en el terminal de cada máquina virtual, escribiendo: **`tcpdump -i eth0`**.

1.1. Comunicación entre máquinas con s1 apagado

1. Piensa en qué paquetes se capturarán en pc2, pc3 y en pc5 si se hace un ping desde pc1 a pc2.

2. Lanza `tcpdump` en pc2, pc3 y en pc5. A continuación ejecuta la siguiente orden en pc1 para hacer un ping a pc2:

```
pc1:~# ping -c 3 11.0.0.2
```

(-c 3 hace que el ping sólo envíe 3 paquetes ICMP)

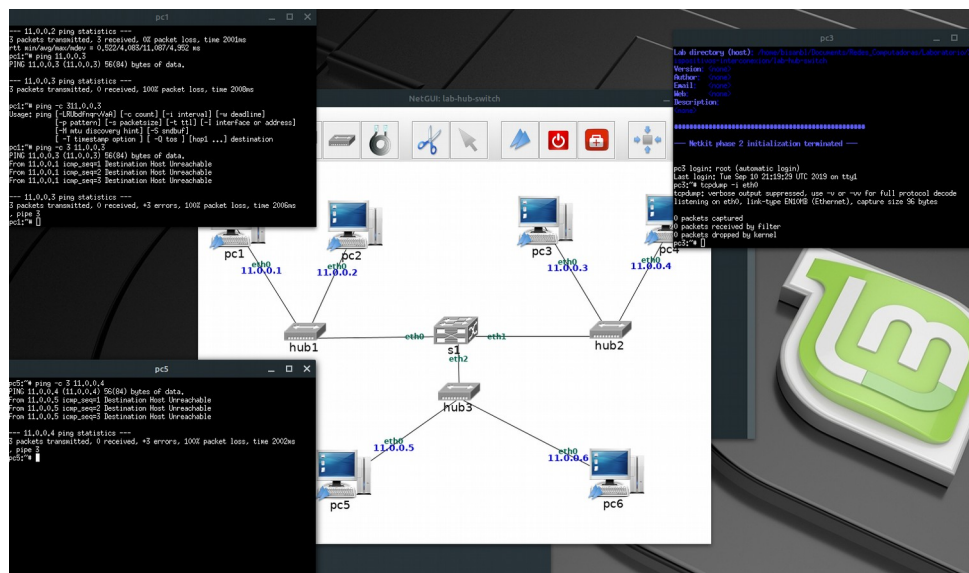
Observa el tráfico capturado en pc2, pc3 y pc5 y comprueba si ha ocurrido lo que pensabas.

The image displays four terminal windows from a network simulation. The top-left window (pc5) shows a directory listing and a 'Netkit phase 2 initialization terminated' message. The top-right window (pc2) shows a directory listing, a 'Netkit phase 2 initialization terminated' message, and a series of ICMP echo requests and replies between 11.0.0.1 and 11.0.0.2. The bottom-left window (pc1) shows a directory listing, a 'Netkit phase 2 initialization terminated' message, and a series of ICMP echo requests and replies between 11.0.0.2 and 11.0.0.1. The bottom-right window (pc3) shows a directory listing, a 'Netkit phase 2 initialization terminated' message, and a series of ICMP echo requests and replies between 11.0.0.1 and 11.0.0.2.

Ha ocurrido lo esperado ya que no existe comunicación debido al switch apagado, las únicas que recibe la trama es pc2.

3. Comprueba que no existe conectividad (es decir, que no puede hacerse ping) entre máquinas que estén en diferentes hubs.

No existe conectividad.



1.2.

Comunicación entre máquinas con s1 arrancado

1. Arranca el switch s1.

2. Piensa en qué paquetes se capturarán ahora en pc2, pc3 y en p5 repitiendo el mismo ping

```
pc1:~# arp -a
pc1:~#
```

```
s1
Adding eth1 interface
Adding eth2 interface
Switch s1 is configured
>>> End of s1 specific startup script.

=====

Lab directory (host): /home/bisarb1/Documents/Redes_Computadoras/Laboratorio/2.1.1
(specificativos-interconexion/lab-hub-switch
Version: <none>
Author: <none>
Email: <none>
Web: <none>
Description:
<none>

=====

--- Netkit phase 2 initialization terminated ---

s1 login: root (automatic login)
s1:~#
```

3. Comprueba la caché de ARP en pc1. Si aún está en ella la dirección Ethernet de pc2 borra esa entrada de la caché de ARP.

4. Lanza tcpdump en pc2, pc3 y en pc5. A continuación vuelve a hacer en pc1 el ping a pc2:

pc1:~# ping -c 3 11.0.0.2

```
pc1:~# ping -c 3 11.0.0.2
PING 11.0.0.2 (11.0.0.2) 56(84) bytes of data.
64 bytes from 11.0.0.2: icmp_seq=1 ttl=64 time=11.2 ms
64 bytes from 11.0.0.2: icmp_seq=2 ttl=64 time=0.643 ms
64 bytes from 11.0.0.2: icmp_seq=3 ttl=64 time=0.494 ms

--- 11.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.494/4.124/11.235/5.028 ms
pc1:~#
```

```
pc2:~# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
22:30:16.352585 arp who-has 11.0.0.2 tell 11.0.0.1
22:30:16.352846 IP 11.0.0.1 > 11.0.0.2: ICMP echo request, id 52483, seq 1, length 64
22:30:16.352895 IP 11.0.0.2 > 11.0.0.1: ICMP echo reply, id 52483, seq 1, length 64
22:30:17.344076 IP 11.0.0.1 > 11.0.0.2: ICMP echo request, id 52483, seq 2, length 64
22:30:17.344125 IP 11.0.0.2 > 11.0.0.1: ICMP echo reply, id 52483, seq 2, length 64
22:30:18.342884 IP 11.0.0.1 > 11.0.0.2: ICMP echo request, id 52483, seq 3, length 64
22:30:18.342933 IP 11.0.0.2 > 11.0.0.1: ICMP echo reply, id 52483, seq 3, length 64
22:30:21.354280 arp who-has 11.0.0.1 tell 11.0.0.2
22:30:21.354738 arp reply 11.0.0.1 is-at 00:07:e9:00:00:01 (oui Unknown)
```

```
pc3:~# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
22:30:16.352811 arp who-has 11.0.0.2 tell 11.0.0.1

1 packets captured
1 packets received by filter
0 packets dropped by kernel
pc3:~#
```

```
pc5:~# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
22:30:16.352868 arp who-has 11.0.0.2 tell 11.0.0.1

1 packets captured
1 packets received by filter
0 packets dropped by kernel
pc5:~#
```

Observa el tráfico capturado en pc2, pc3 y pc5 y comprueba si ha ocurrido lo que pensabas.

Esperaba que ocurriera esto debido a que como el switch esta funcionando el paquete arp lo reciben todas las computadoras de los distintos hubs. Sin embargo es el único paquete que recibe.

5. Responde a estas preguntas:

¿Por qué llega a pc3 y a pc5 la solicitud de ARP enviada por pc1?

Porque ARP es enviado a todas las computadoras de la red a través de la dirección de difusión/Broadcast.

¿Por qué NO llega a pc3 y a pc5 la respuesta de ARP enviada por pc2?

Debido a que esta respuesta ya va dirigida a pc1 porque conoce explícitamente cómo llegar a ella.

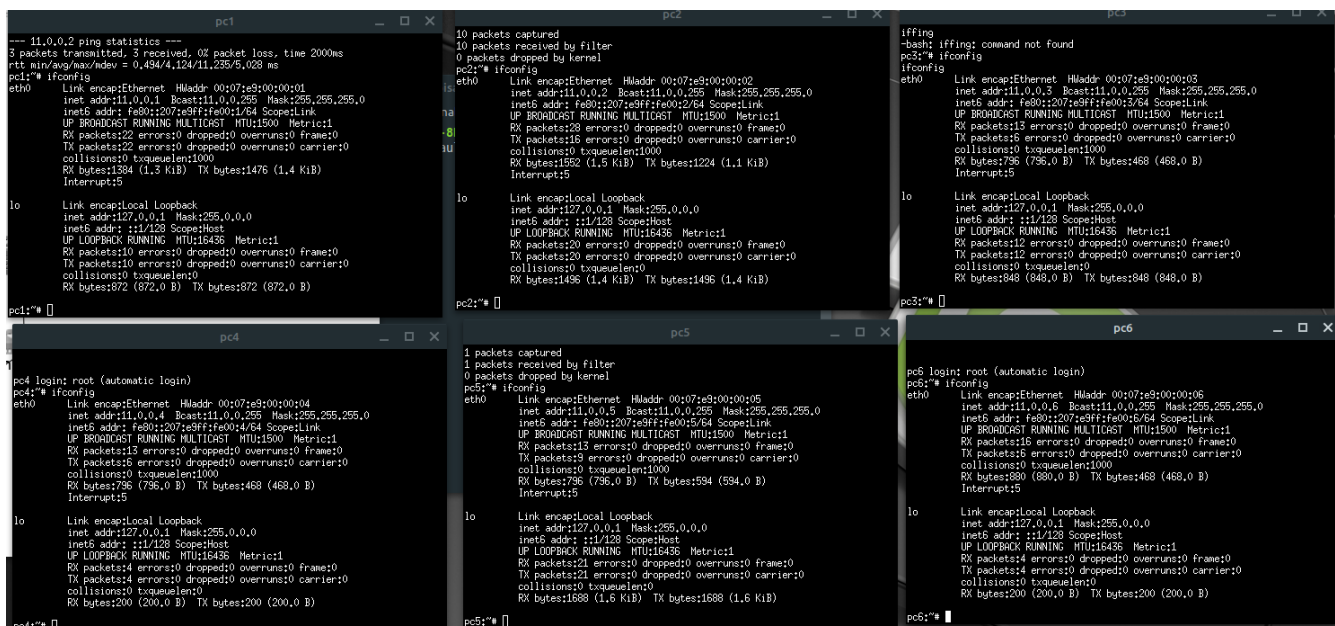
¿Por qué NO llega a pc3 y a pc5 el ICMP echo request enviado por pc1?

Porque al pc1 y pc2 haberse encontrado mediante el paquete arp anterior ya pueden comunicarse explícitamente usando solo sus direcciones MAC y el switch ya aprendió a quien le pertenece cual ip.

¿Por qué NO llega a pc3 y a pc5 el ICMP echo reply enviado por pc2?

Por las mismas razones que en la respuesta anterior.

6. Comprueba las direcciones Ethernet que tiene cada interfaz de cada máquina de la figura (usando ifconfig), y apúntalas.



```
pc1:~# ifconfig
eth0: Link encap:Ethernet HWaddr 00:07:e9:00:00:01
       inet addr:111.0.0.1 Bcast:111.0.0.255 Mask:255.255.255,0
       inet6 addr: fe80::207:e9ff:fe00:124 ScopeLink
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:22 errors:0 dropped:0 overruns:0 frame:0
       TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:1384 (1.3 KiB) TX bytes:1476 (1.4 KiB)
       Interrupt:5

lo: Link encap:Local Loopback
     inet addr:127.0.0.1 Mask:255.0.0.0
     inet6 addr: ::1/128 ScopeHost
     UP LOOPBACK RUNNING MTU:65536 Metric:1
     RX packets:10 errors:0 dropped:0 overruns:0 frame:0
     TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:0
     RX bytes:872 (872.0 B) TX bytes:872 (872.0 B)

pc1:~#

pc2:~# ifconfig
eth0: Link encap:Ethernet HWaddr 00:07:e9:00:00:02
       inet addr:111.0.0.2 Bcast:111.0.0.255 Mask:255.255.255,0
       inet6 addr: fe80::207:e9ff:fe00:124 ScopeLink
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:20 errors:0 dropped:0 overruns:0 frame:0
       TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:1592 (1.5 KiB) TX bytes:1224 (1.1 KiB)
       Interrupt:5

lo: Link encap:Local Loopback
     inet addr:127.0.0.1 Mask:255.0.0.0
     inet6 addr: ::1/128 ScopeHost
     UP LOOPBACK RUNNING MTU:65536 Metric:1
     RX packets:20 errors:0 dropped:0 overruns:0 frame:0
     TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:0
     RX bytes:1496 (1.4 KiB) TX bytes:1496 (1.4 KiB)

pc2:~#

pc3:~# ifconfig
eth0: Link encap:Ethernet HWaddr 00:07:e9:00:00:03
       inet addr:111.0.0.3 Bcast:111.0.0.255 Mask:255.255.255,0
       inet6 addr: fe80::207:e9ff:fe00:124 ScopeLink
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:13 errors:0 dropped:0 overruns:0 frame:0
       TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:736 (736.0 B) TX bytes:468 (468.0 B)
       Interrupt:5

lo: Link encap:Local Loopback
     inet addr:127.0.0.1 Mask:255.0.0.0
     inet6 addr: ::1/128 ScopeHost
     UP LOOPBACK RUNNING MTU:65536 Metric:1
     RX packets:12 errors:0 dropped:0 overruns:0 frame:0
     TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:0
     RX bytes:848 (848.0 B) TX bytes:848 (848.0 B)

pc3:~#

pc4:~# ifconfig
eth0: Link encap:Ethernet HWaddr 00:07:e9:00:00:04
       inet addr:111.0.0.4 Bcast:111.0.0.255 Mask:255.255.255,0
       inet6 addr: fe80::207:e9ff:fe00:124 ScopeLink
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:13 errors:0 dropped:0 overruns:0 frame:0
       TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:736 (736.0 B) TX bytes:468 (468.0 B)
       Interrupt:5

lo: Link encap:Local Loopback
     inet addr:127.0.0.1 Mask:255.0.0.0
     inet6 addr: ::1/128 ScopeHost
     UP LOOPBACK RUNNING MTU:65536 Metric:1
     RX packets:4 errors:0 dropped:0 overruns:0 frame:0
     TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:0
     RX bytes:200 (200.0 B) TX bytes:200 (200.0 B)

pc4:~#

pc5:~# ifconfig
eth0: Link encap:Ethernet HWaddr 00:07:e9:00:00:05
       inet addr:111.0.0.5 Bcast:111.0.0.255 Mask:255.255.255,0
       inet6 addr: fe80::207:e9ff:fe00:124 ScopeLink
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:13 errors:0 dropped:0 overruns:0 frame:0
       TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:736 (736.0 B) TX bytes:594 (594.0 B)
       Interrupt:5

lo: Link encap:Local Loopback
     inet addr:127.0.0.1 Mask:255.0.0.0
     inet6 addr: ::1/128 ScopeHost
     UP LOOPBACK RUNNING MTU:65536 Metric:1
     RX packets:21 errors:0 dropped:0 overruns:0 frame:0
     TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:0
     RX bytes:1688 (1.6 KiB) TX bytes:1688 (1.6 KiB)

pc5:~#

pc6:~# ifconfig
eth0: Link encap:Ethernet HWaddr 00:07:e9:00:00:06
       inet addr:111.0.0.6 Bcast:111.0.0.255 Mask:255.255.255,0
       inet6 addr: fe80::207:e9ff:fe00:124 ScopeLink
       UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
       RX packets:16 errors:0 dropped:0 overruns:0 frame:0
       TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:880 (880.0 B) TX bytes:468 (468.0 B)
       Interrupt:5

lo: Link encap:Local Loopback
     inet addr:127.0.0.1 Mask:255.0.0.0
     inet6 addr: ::1/128 ScopeHost
     UP LOOPBACK RUNNING MTU:65536 Metric:1
     RX packets:4 errors:0 dropped:0 overruns:0 frame:0
     TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:0
     RX bytes:200 (200.0 B) TX bytes:200 (200.0 B)

pc6:~#
```

7. Mira la tabla de direcciones aprendidas por el switch s1 utilizando la orden `brctl showmacs s1`. Puede utilizarla junto con la orden `watch` para observar periódicamente los cambios en las direcciones aprendidas:

`s1:~# watch brctl showmacs s1`

(watch repite cada 2 segundos la ejecución de la orden que se le pasa como parámetro)

Identifica las máquinas a las que pertenece cada dirección Ethernet y explica su presencia en la tabla de direcciones aprendidas de s1.

Tras 300 segundos comprobarás que el switch olvida las direcciones aprendidas (mira cómo va creciendo el valor de la columna ageing timer, contador de envejecimiento, en la salida de la orden). Comprueba también cómo el ageing timer de una dirección Ethernet se reinicializa cada vez que el switch ve una nueva trama con esa dirección Ethernet.

```
s1
Every 2.0s: brctl showmacs s1
Tue Sep 10 22:44:47 2019

port no mac addr          is local?    ageing timer
 1    00:07:e9:00:ff:f0      yes           0.00
 2    00:07:e9:00:ff:f1      yes           0.00
 3    00:07:e9:00:ff:f2      yes           0.00
```

```
s1
Every 2.0s: brctl showmacs s1
Tue Sep 10 23:23:24 2019

port no mac addr          is local?    ageing timer
 1    00:07:e9:00:00:01      no            141.34
 1    00:07:e9:00:00:02      no            116.08
 2    00:07:e9:00:00:03      no            100.55
 2    00:07:e9:00:00:04      no            141.34
 3    00:07:e9:00:00:05      no            116.08
 3    00:07:e9:00:00:06      no            100.55
 1    00:07:e9:00:ff:f0      yes           0.00
 2    00:07:e9:00:ff:f1      yes           0.00
 3    00:07:e9:00:ff:f2      yes           0.00
```

8. Comprueba que ahora sí existe conectividad entre todas las máquinas de la figura utilizando la orden ping.

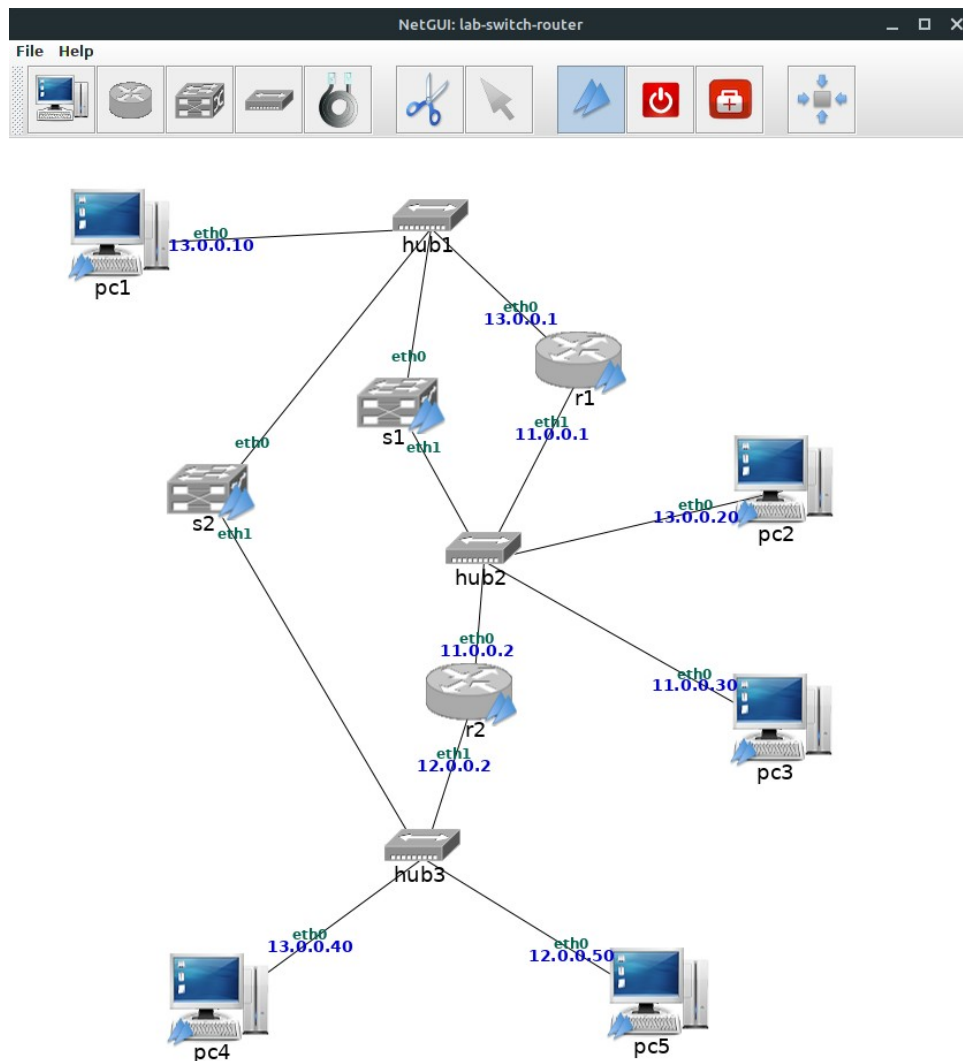
```
pc1:~# nano /etc/net
netkit-filesysen-version networks
network/
pc1:~# nano /etc/network/interfaces
pc1:~# ping 11.0.0.4
PING 11.0.0.4 (11.0.0.4) 56(84) bytes of data.
64 bytes from 11.0.0.4: icmp_seq=1 ttl=64 time=11.7 ms
64 bytes from 11.0.0.4: icmp_seq=2 ttl=64 time=0.841 ms
64 bytes from 11.0.0.4: icmp_seq=3 ttl=64 time=0.734 ms
--- 11.0.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1990ms
rtt min/avg/max/ndev = 0.734/4.436/11.733/0.159 ms
pc1:~#

pc5:~# TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:736 (736.0 B) TX bytes:594 (594.0 B)
Interrupt:5
pc5:~# lo
Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:10 errors:0 dropped:0 overruns:0 frame:0
TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:872 (872.0 B) TX bytes:872 (872.0 B)
pc5:~# ping 11.0.0.2
PING 11.0.0.2 (11.0.0.2) 56(84) bytes of data.
64 bytes from 11.0.0.2: icmp_seq=1 ttl=64 time=1.5 ms
64 bytes from 11.0.0.2: icmp_seq=2 ttl=64 time=1.06 ms
64 bytes from 11.0.0.2: icmp_seq=3 ttl=64 time=1.05 ms
--- 11.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/ndev = 1.051/4.952/11.533/4.940 ms
pc5:~#

pc6:~# TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:880 (880.0 B) TX bytes:468 (468.0 B)
Interrupt:5
pc6:~# lo
Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:14 errors:0 dropped:0 overruns:0 frame:0
TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:200 (200.0 B) TX bytes:200 (200.0 B)
pc6:~# ping 11.0.0.3
PING 11.0.0.3 (11.0.0.3) 56(84) bytes of data.
64 bytes from 11.0.0.3: icmp_seq=1 ttl=64 time=10.7 ms
64 bytes from 11.0.0.3: icmp_seq=2 ttl=64 time=0.878 ms
64 bytes from 11.0.0.3: icmp_seq=3 ttl=64 time=0.671 ms
--- 11.0.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/ndev = 0.671/4.106/10.770/4.712 ms
pc6:~#
```

2. Redes conectadas a través de switch y router

En el fichero lab-switch-router.tgz está definida una red como la que aparece en la figura 2. Descomprime el fichero, lanza NetGUI y abre el escenario. Arranca todas las máquinas: pcs, routers y switches.



2.1. Comunicación entre pc2 y pc4

Con las cachés de ARP vacías y las tablas de direcciones aprendidas de los switches vacías se desea realizar un ping de pc2 a pc4:

1. Observa la configuración que hay en el escenario para que pc2 y pc4 puedan intercambiar tráfico. ¿Cuál de los siguientes caminos crees que seguirán los mensajes ICMP echo request desde pc2 a pc4?

pc2 → s1 → s2 → pc4

pc2 → r1 → s2 → pc4

pc2 → r2 → pc4

Justifica la respuesta.

Tomara el primero camino Pc2 → s1 → s2 → pc4. Debido a que r1 y r2 tienen ips en redes diferentes a pc2. Esta no puede llegar a ellos sin pasar primero por un router.

2. Indica cuántas solicitudes y respuestas de ARP serían necesarias para que dicho ping funcionase. Explica en qué pcs/routers/switches y su interfaz eth concreta se podrían capturar:

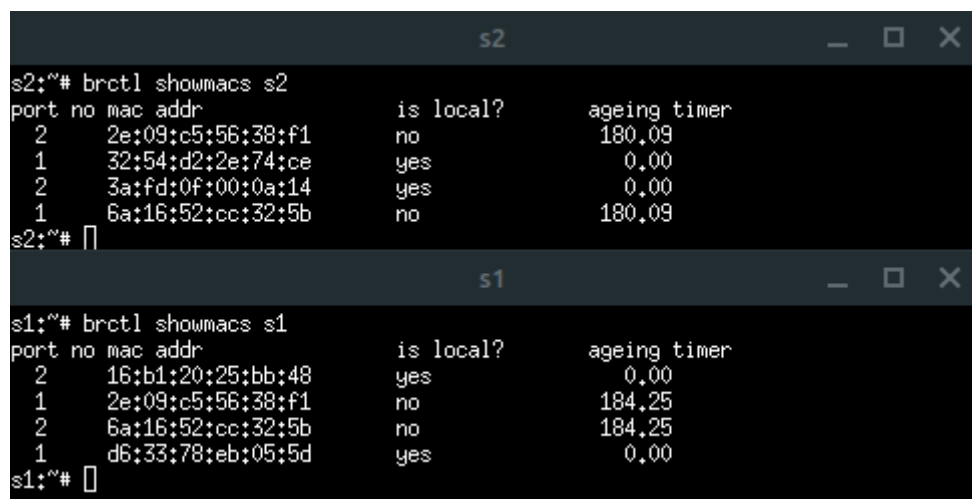
- solicitud/es de ARP.
- respuesta/s de ARP.

Los mensajes de solicitud y respuesta de ARP pueden verse desde cualquier interfaz en todos los pcs/routers/switches ya que todos estan conectados a hubs en ultima instancia las solicitudes de broadcast le llegaran a todos sin distincion.

Lanza tcpdump en las máquinas que necesites para ayudarte a comprobar tus suposiciones 1 . No hace falta redirigir las capturas a un fichero, puedes hacerlas escribiendo: tcpdump -i <interfaz>.

Tcpdump mostró que estaba en lo cierto.

3. Indica qué direcciones Ethernet habrán aprendido s1 y s2 después de ejecutar el ping. Compruébalo. La de pc2 y pc4.



```
s2:~# brctl showmacs s2
port no mac addr          is local?  ageing timer
 2    2e:09:c5:56:38:f1    no         180.09
 1    32:54:d2:2e:74:ce    yes         0.00
 2    3a:fd:0f:00:0a:14    yes         0.00
 1    6a:16:52:cc:32:5b    no         180.09
s2:~#

s1:~# brctl showmacs s1
port no mac addr          is local?  ageing timer
 2    16:b1:20:25:bb:48    yes         0.00
 1    2e:09:c5:56:38:f1    no         184.25
 2    6a:16:52:cc:32:5b    no         184.25
 1    d6:33:78:eb:05:5d    yes         0.00
s1:~#
```

4. ¿Crees que habrá llegado alguno de los mensajes ICMP echo request a pc1, pc3 o pc5? Justifica la respuesta. Si porque de la misma forma que con los ARP al estar conectados a un Hub todos reciben todos los mensajes de forma indistinta.

2.2. Comunicación entre pc1 y pc3

Con las cachés de ARP vacías y las tablas de direcciones aprendidas de los switches vacías se desea realizar un ping de pc1 a pc3:

1. Observa la configuración que hay en el escenario para que pc1 y pc3 puedan intercambiar tráfico. ¿Cuál de los siguientes caminos crees que seguirán los mensajes ICMP echo request desde pc1 a pc3?

- pc1 → r1 → pc3
- pc1 → s1 → pc3
- pc1 → s2 → r2 → pc3

Justifica la respuesta.

El primero de pc1 a r1 y de este a pc3. La razón es porque al estar en redes diferentes. Tiene que pasar por el gw que en el caso de pc1 debe de ser r1 al ser el unico router existente en la red.

2. Indica cuántas solicitudes y respuestas de ARP serian necesarias para que dicho ping funcionase. Explica en qué pcs/routers/switches y su interfaz eth concreta se podrían capturar:

- solicitud/es de ARP.
- respuesta/s de ARP.

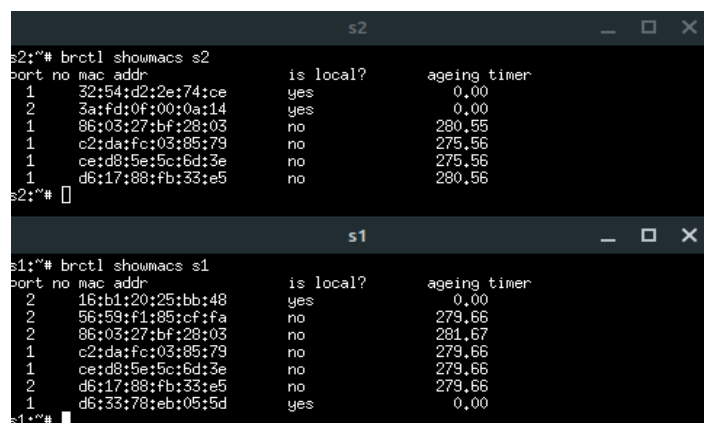
1 mensaje ARP por parte de PC1 buscando a PC3 y un mensaje de respuesta de PC3 a PC1.

Adicional 1 mensaje ARP desde PC3 preguntando por PC1.

Por la configuración de la red se puede capturar el trafico en todas las interfaces de todos los pcs/routers/switches

Lanza tcpdump en las máquinas que necesites para ayudarte a comprobar tus suposiciones. No hace falta redirigir las capturas a un fichero, puedes hacerlas escribiendo: tcpdump -i <interfaz>.

Mi suposición de las interfaces de captura era correcta pero PC3 no pregunta por PC1 si no por r2. 3. Indica qué direcciones Ethernet habrán aprendido s1 y s2 después de ejecutar el ping. Compruébalo. Las de pc1 y pc3



```
s2:~# brctl showmacs s2
port no mac addr is local? ageing timer
1 32:54:d2:2e:74:ce yes 0.00
2 3a:fd:0f:00:0a:14 yes 0.00
1 86:03:27:bf:28:03 no 280.55
1 c2:da:fc:03:85:79 no 275.56
1 ce:d8:5e:5c:6d:3e no 275.56
1 d6:17:88:fb:33:e5 no 280.56
s2:~#

s1:~# brctl showmacs s1
port no mac addr is local? ageing timer
2 16:b1:20:25:bb:48 yes 0.00
2 56:59:f1:85:cf:fa no 279.66
2 86:03:27:bf:28:03 no 281.67
1 c2:da:fc:03:85:79 no 279.66
1 ce:d8:5e:5c:6d:3e no 279.66
2 d6:17:88:fb:33:e5 no 279.66
1 d6:33:78:eb:05:5d yes 0.00
s1:~#
```

4. ¿Crees que habrá llegado alguno de los mensajes ICMP echo request a pc2, pc4 o pc5? Justifica la respuesta. A PC2 si pero a PC4 y PC5 no. debido a que S2 no les enviaria los mensajes.

2.3. Comunicación entre pc2 y pc5

Con las cachés de ARP vacías y las tablas de direcciones aprendidas de los switches vacías se desea realizar un ping de pc2 a pc5:

1. Observa la configuración que hay en el escenario para que pc2 y pc5 puedan intercambiar tráfico. ¿Cuál de los siguientes caminos crees que seguirán los mensajes ICMP echo request desde pc2 a pc5?

- pc2 → r2 → pc5
- pc2 → r1 → s2 → pc5
- pc2 → r2 → s1 → s2 → pc5
- pc2 → r2 → r1 → s2 → pc5
- pc2 → s1 → r1 → r2 → pc5

Justifica la respuesta.

El camino a utilizar sera pc2 → s1 → r1 → r2 → pc5 porque pc2 no conoce la red de pc5 y por lo tanto le enviara el mensaje a su gw r1. Y r1 se lo enviara a r2 que es quien conoce la red de pc5 y posteriormente r2 se lo entregara a pc5.

2. Indica cuántas solicitudes y respuestas de ARP serían necesarias para que dicho ping funcionase. Explica en qué pcs/routers/switches y su interfaz eth concreta se podrían capturar:

- solicitud/es de ARP.
- respuesta/s de ARP.

3 solicitudes y 3 respuestas.

Tipo	Dispositivo	Interfaz
Solicitud	Pc2 → r1	Eth0(r1 y pc2)
Solicitud	R1 → Pc2	Eth1(r1) y Eth0 (r2)
Solicitud	R2 → broadcast 12.0.0.0	Eth1(r2)
Respuesta	Pc5 → r2	Eth0 (Pc5) y Eth1(r2)
Respuesta	R2 → r1	Eth0(r2) y eth1(r1)
Respuesta	R1 → Pc2	Eth0(r1) y eth0(pc2)

Lanza tcpdump en las máquinas que necesites para ayudarte a comprobar tus suposiciones. No hace falta redirigir las capturas a un fichero, puedes hacerlas escribiendo: tcpdump -i <interfaz>.

3. Indica qué direcciones Ethernet habrán aprendido s1 y s2 después de ejecutar el ping. Compruébalo. Las de pc5

```
s2:~# brctl showmacs s2
port no mac addr          is local?    ageing timer
 2  2a:32:fe:9b:af:a8      no           131.73
 1  32:54:d2:2e:74:ce      yes          0.00
 2  3a:fd:0f:00:0a:14      yes          0.00
 1  6a:16:52:cc:32:5b      no           131.72
 2  7e:a2:34:57:b0:d5      no           131.73
 1  c2:da:fc:03:85:79      no           131.72
 1  d6:17:88:fb:33:e5      no           136.73
s2:~#

s1:~# brctl showmacs s1
port no mac addr          is local?    ageing timer
 2  16:b1:20:25:bb:48      yes          0.00
 1  d6:33:78:eb:05:5d      yes          0.00
s1:~# brctl showmacs s1
port no mac addr          is local?    ageing timer
 2  16:b1:20:25:bb:48      yes          0.00
 2  56:59:f1:85:cf:fa      no           2.91
 2  6a:16:52:cc:32:5b      no           2.91
 1  7e:a2:34:57:b0:d5      no           3.88
 1  c2:da:fc:03:85:79      no           2.91
 2  d6:17:88:fb:33:e5      no           2.91
 1  d6:33:78:eb:05:5d      yes          0.00
```

S2 aprendió la mac de r2 eth1 y Pc5 eth0

S1 aprendió r2 eth0, pc5 eth0 y r1 eth1

4. ¿Crees que habrá llegado alguno de los mensajes ICMP echo request a pc1, pc3 o pc4? Justifica la respuesta.

Si. A todos ellos les llegaron todos los mensajes ICMP . Por la presencia de los hubs en la red que le replican los mensajes a todos los dispositivos conectados sin importar si el mensaje es para ellos.

3. Proxy ARP

En el fichero lab-proxyARP.tgz está definida una red como la que aparece en la figura 3. Descomprime el fichero, lanza NetGUI y abre el escenario. Arranca las máquinas de una en una. Características del escenario:

- Los pcs y el router r1 están configurados con las direcciones IP que se muestran en la figura.
- En el fichero /etc/hosts de cada pc están los nombres y direcciones IP de pc1, pc2 y pc3, por lo que puedes referirte a ellos por su nombre además de por su IP en las órdenes que utilices.

1. Activa proxy ARP en la configuración del router r1 para que las máquinas pc1 y pc2 tengan conectividad IP entre ellas en ambos sentidos. Explica qué modificaciones han sido necesarias y por qué.

```
r1 login: root (automatic login)
Last login: Fri Oct  4 15:05:18 UTC 2019 on tty1
r1:~# echo 1 > /proc/sys/net/ipv4/conf/eth0/proxy_arp
r1:~# arp -i eth0 -Ds 12.0.0.20 eth0 netmask 255.255.255.0
SIOCSARP: Invalid argument
r1:~# arp -i eth0 -Ds 12.0.0.20 eth0 netmask 255.255.255.255
r1:~# route add -host 12.0.0.20 dev eth1
r1:~# echo 1 > /proc/sys/net/ipv4/conf/eth1/proxy_arp
r1:~# arp -i eth1 -Ds 12.0.0.10 eth1 netmask 255.255.255.255
r1:~#
```

Paso 1

Se activo proxy arp en ambas interfaces para que pueda funcionar con los comandos echo.

Paso 2

Se especifica con el comando arp la interfaz para la que se activara arp proxy

cuando se pregunte por una dirección ip en concreto también se le asigna la dirección mac que se especificara que le responde siempre la misma desde la que se recibe la solicitud arp. Además de la máscara la cual especificando 255.255.255.255 se limita a solo un host en concreto.

Paso 3

Se Agrega la ruta por defecto especificando la interfaz para el envío de los paquetes del host en concreto al que se quiere llegar.

Paso 4

Se repite los primeros 2 pasos en la misma interfaz que se especifico en la ruta por defecto. Pero con los valores invertidos.

2. Con las cachés de ARP vacías, realiza una captura en la interfaz r1(eth0) guardando su contenido en un fichero y ejecuta un ping desde pc1 a 12.0.0.1, enviando sólo 3 paquetes, y después un ping desde pc1 a pc2, enviando sólo 3 paquetes. Interrumpe la captura y fíjate en las solicitudes de ARP que ves en el tráfico capturado. A partir de la captura y de las direcciones IP de r1: ¿cómo puedes saber que r1 está realizando proxy ARP?

Debido a que cuando Pc1 pregunta por pc2 la dirección mac que le envía r1 como dirección destino para sus paquetes le pertenece a eth0 de el mismo. Y no a pc2.

3. Si se ha borrado la caché de ARP de pc1 vuelve a ejecutar los 2 pings anteriores y consulta la caché de ARP de pc1, indica qué observas.

```
pc1:~# ping -c3 12.0.0.1
PING 12.0.0.1 (12.0.0.1) 56(84) bytes of data.
64 bytes from 12.0.0.1: icmp_seq=1 ttl=64 time=10.7 ms
64 bytes from 12.0.0.1: icmp_seq=2 ttl=64 time=0.559 ms
64 bytes from 12.0.0.1: icmp_seq=3 ttl=64 time=0.542 ms

--- 12.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.542/3.942/10.727/4.797 ms
pc1:~# ping -c3 12.0.0.20
PING 12.0.0.20 (12.0.0.20) 56(84) bytes of data.
64 bytes from 12.0.0.20: icmp_seq=1 ttl=63 time=873 ms
64 bytes from 12.0.0.20: icmp_seq=2 ttl=63 time=0.736 ms
64 bytes from 12.0.0.20: icmp_seq=3 ttl=63 time=1.09 ms

--- 12.0.0.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.736/291.702/873.276/411.234 ms
pc1:~# arp

```

Address	Hwtype	Hwaddress	Flags	Mask	Iface
pc2	ether	EE:66:EE:7A:1D:65	C		eth0
12.0.0.1	ether	EE:66:EE:7A:1D:65	C		eth0

```
pc1:~#
```

Que las mac address de pc2 y r1 (12.0.0.1) son las mismas.

4. IP aliasing

En el fichero lab-ipAliasing.tgz está definida una red como la que aparece en la figura 4. Descomprime el fichero, lanza NetGUI y abre el escenario. Arranca las máquinas de una en una.

Características del escenario:

- Los pcs y los routers r1 y r2 están configurados con las direcciones IP que se muestran en la figura.
- En el fichero /etc/hosts de cada pc están los nombres y direcciones IP de pc1, pc2, pc3, pc4 y pc5, por lo que puedes referirte a ellos por su nombre además de por su IP en las órdenes que utilices.
- pc1, pc3 y pc5 tienen conectividad IP entre ellos. pc2 y pc4 no tienen conectividad IP, ya que no están conectados a sus respectivas subredes.

1. Asigna direcciones IP adicionales en los routers mediante IP aliasing, y configura las tablas de encaminamiento que sean necesarias para que pc2 pueda hacer ping a pc3, ten en cuenta que desde r2 se debería poder alcanzar también a pc1. Nótese que cuando añades una dirección por IP aliasing a una tabla de encaminamiento se añade automáticamente una entrada para la subred a la que pertenece, entrada que a veces es necesario borrar para que no haya en la misma tabla dos rutas diferentes a la misma subred.

2. Realiza una captura en r2(eth1) guardando su contenido en un fichero y ejecuta un ping desde pc2 a pc3 enviando 3 paquetes y después ejecuta un ping desde pc5 a 12.0.0.2. Interrumpe la captura y observa las solicitudes de ARP. ¿Se puede saber sólo mirando el fichero de captura que en r2 no se ha configurado proxy ARP?

No.

3. Con la configuración que has realizado previamente ¿pueden comunicarse pc1 y pc5? ¿Por qué? Si tu respuesta es negativa, modifica la configuración para que pc5 y pc1 puedan intercambiar tráfico.

Si puede porque r2 y r1 tienen las rutas necesarias para comunicarse usando la red de Pc1.

```
pc5:~# ping -c 3 13.0.0.10
PING 13.0.0.10 (13.0.0.10) 56(84) bytes of data.
64 bytes from 13.0.0.10: icmp_seq=1 ttl=62 time=4.67 ms
64 bytes from 13.0.0.10: icmp_seq=2 ttl=62 time=1.57 ms

--- 13.0.0.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1005ms
rtt min/avg/max/mdev = 1.579/3.124/4.670/1.546 ms
pc5:~#
```

4. Utiliza de nuevo IP aliasing para que pc4 pueda hacer ping a pc1, ten en cuenta que desde r1 se debería poder alcanzar también a pc5.

```
r1
r1:~# ping -c 1 12.0.0.50
PING 12.0.0.50 (12.0.0.50) 56(84) bytes of data.
64 bytes from 12.0.0.50: icmp_seq=1 ttl=63 time=0.693 ms

--- 12.0.0.50 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.693/0.693/0.693/0.000 ms
r1:~#

pc4
pc4:~# ping -c 1 13.0.0.10
PING 13.0.0.10 (13.0.0.10) 56(84) bytes of data.
64 bytes from 13.0.0.10: icmp_seq=1 ttl=63 time=14.6 ms

--- 13.0.0.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 14.650/14.650/14.650/0.000 ms
pc4:~#
```

5. Realiza una captura en r1(eth0) (sin necesidad de guardarlo en un fichero) para ver qué paquetes se intercambian cuando pc4 hace ping a pc1.

```
r1:~# tcpdump -i eth0 -s 0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:02:58.952505 IP pc4 > pc1: ICMP echo request, id 4098, seq 1, length 64
20:02:58.960071 IP pc4 > pc1: ICMP echo request, id 4098, seq 1, length 64
20:02:58.957345 arp who-has 13.0.0.1 tell pc1
20:02:58.957364 arp reply 13.0.0.1 is-at c2:c5:9f:bf:f5:c5 (oui Unknown)
20:02:58.957586 IP pc1 > pc4: ICMP echo reply, id 4098, seq 1, length 64
20:02:58.957615 IP 13.0.0.1 > pc1: ICMP redirect pc4 to host pc4, length 92
20:02:58.957628 IP pc1 > pc4: ICMP echo reply, id 4098, seq 1, length 64
20:03:03.916380 arp who-has 12.0.0.1 tell pc4
20:03:03.916410 arp reply 12.0.0.1 is-at c2:c5:9f:bf:f5:c5 (oui Unknown)
20:03:03.939428 arp who-has pc1 tell 13.0.0.1
20:03:03.939788 arp reply pc1 is-at 0a:c9:e0:9b:e6:67 (oui Unknown)
```

5. VLANs

En el fichero lab-vlan.tgz está definida la topología de una red como la de la figura 5 en la que aún no se han configurado las VLANs. Descomprime el fichero, arranca NetGUI y abre el escenario. Arranca las máquinas de una en una.

Los dispositivos de interconexión s1, s2 y s3 están configurados para que funcionen como switches Ethernet.

1. Indica qué máquinas se pueden comunicar entre ellas. Compruébalo realizando ping.

- Pc1, Pc2, Pc3 y Pc4 tiene comunicación entre si
- Pc5 y Pc6 tiene comunicación entre si

```
pc1
pc1:~# ping -c 1 11.0.0.102
PING 11.0.0.102 (11.0.0.102) 56(84) bytes of data.
64 bytes from 11.0.0.102: icmp_seq=1 ttl=64 time=13.9 ms

--- 11.0.0.102 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 13.969/13.969/13.969/0.000 ms
pc1:~# ping -c 1 11.0.0.103
PING 11.0.0.103 (11.0.0.103) 56(84) bytes of data.
64 bytes from 11.0.0.103: icmp_seq=1 ttl=64 time=12.6 ms

--- 11.0.0.103 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 12.668/12.668/12.668/0.000 ms
pc1:~# ping -c 1 11.0.0.104
PING 11.0.0.104 (11.0.0.104) 56(84) bytes of data.
64 bytes from 11.0.0.104: icmp_seq=1 ttl=64 time=12.4 ms

--- 11.0.0.104 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 12.413/12.413/12.413/0.000 ms
pc1:~# █

pc5
pc5:~# ping -c 1 12.0.0.106
PING 12.0.0.106 (12.0.0.106) 56(84) bytes of data.
64 bytes from 12.0.0.106: icmp_seq=1 ttl=64 time=13.2 ms

--- 12.0.0.106 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 13.214/13.214/13.214/0.000 ms
pc5:~# █
```

2. Suponiendo que la caché de ARP de pc1 está vacía, indica dónde se puede capturar un solicitud de ARP que la máquina pc1 envía preguntando por la dirección Ethernet de la máquina pc2. Compruébalo realizando capturas. Para este caso puedes utilizar tcpdump -i <interfaz> -s 0 sin necesidad de guardar la captura en un fichero, de esta forma verás el resultado mostrado en pantalla. (Comprueba antes que en la caché de ARP de pc1 no se encuentra la dirección Ethernet de pc2; si estuviera, bórrala).

Al ser una solicitud de broadcast pueden verlo todas las maquinas en todas las interfaces que posean.

5.1. Configuración de VLAN100

Para facilitar la configuración de las VLANs en cada switch se propone que esta configuración quede almacenada en un fichero de script. Un script es un fichero que contiene comandos que se ejecutarán en el intérprete de comandos, tal y como si los tecleáramos en el terminal.

La configuración de la VLAN100 está escrita en los ficheros vlan-s1.sh, vlan-s2.sh y vlan-s3.sh, que se encuentran en s1, s2 y s3 respectivamente.

1. Estudia estos scripts para entender qué hace cada uno de ellos. Observa que la primera línea `#!/bin/bash` indica el intérprete que va a ejecutar este script, en este caso `bash`. El resto de líneas en el fichero que comienzan por `#` son comentarios. Cada uno de los comandos que se desean ejecutar se escriben en líneas diferentes 2 .

2. Ejecuta los scripts para aplicar la configuración. Debes ejecutar cada uno de esos scripts en su switch, por ejemplo en s1:

```
s1:~# ./vlan-s1.sh
```

Puedes comprobar la configuración que tiene en un switch escribiendo `brctl show`.

```
s1
s1:~# ./vlan-s1.sh
Added VLAN with VID == 100 to IF -:eth3:-
vs100: Dropping NETIF_F_UFO since no NETIF_F_HW_CSUM feature.
s1:~# brctl show
bridge name      bridge id        STP enabled      interfaces
vs100            8000.3a3423e9bbb7  no               eth0
                  eth3,100

s1:~#

s2
s2:~# ./vlan-s2.sh
Added VLAN with VID == 100 to IF -:eth0:-
Added VLAN with VID == 100 to IF -:eth1:-
vs100: Dropping NETIF_F_UFO since no NETIF_F_HW_CSUM feature.
s2:~# brctl show
bridge name      bridge id        STP enabled      interfaces
vs100            8000.16b062751529  no               eth0,100
                  eth1,100
                  eth2

s2:~#

s3
s3:~# ./vlan-s3.sh
Added VLAN with VID == 100 to IF -:eth0:-
vs100: Dropping NETIF_F_UFO since no NETIF_F_HW_CSUM feature.
s3:~# brctl show
bridge name      bridge id        STP enabled      interfaces
vs100            8000.4e3a849b816a  no               eth0,100
                  eth1

s3:~#
```

3. Indica qué máquinas se pueden comunicar entre ellas.

Pc1 y Pc2

4. Suponiendo que la caché de ARP de pc1 está vacía, indica dónde se puede capturar un solicitud de ARP que la máquina pc1 envía preguntando por la dirección Ethernet de la máquina pc2.

En pc1(eth0), r1(eth0), s1(eth0,eth3, eth3.100), s2(eth2,eth0,eth0.100,eth1,eth1.100), s3(eth0,eth0.100,eth1) y pc2(eth0)

Compruébalo realizando las capturas que creas necesarias, sin necesidad de guardar en fichero el tráfico capturado. (Comprueba antes que en la caché de ARP de pc1 no se encuentra la dirección Ethernet de pc2, si estuviera, bórrala).

Ademas de los lugares antes mencionados fue posible capturar el mensaje . en la pc6,pc5, pc4 y pc3.

5. Indica qué ocurre cuando se hace un ping desde pc1 a pc2, teniendo en cuenta que ambas máquinas se encuentran en la misma subred. Compruébalo realizando las capturas necesarias, sin necesidad de guardar en un fichero el tráfico capturado.

Los paquetes icmp se encapsulan dentro de la vlan 100.

6. Asegúrate de que la caché de ARP de pc1 está vacía, bórrala si es necesario. Arranca tcpdump en las siguientes interfaces: pc1(eth0), s1(eth3), s2(eth2), s3(eth0) y pc2(eth0), guardando esta vez el tráfico capturado en un fichero. Realiza un ping desde pc1 a pc2.

7. Interrumpe las capturas. Observa las direcciones Ethernet aprendidas por s1, s2 y s3.

8. Analiza las 5 capturas, indica en qué capturas se observa la etiqueta de VLAN en el tráfico y qué identificador de VLAN contiene.

En S1(eth0) y en s3(eth0) encuentro la etiqueta VLAN y el ID es 100

a) ¿Qué switch introduce dicha etiqueta?

S1 cuando el mensaje proviene de PC1 y s3 cuando es PC2.

b) ¿Qué switch elimina dicha etiqueta?

S1 los elimina cuando se dirigen a PC1 y s3 para PC2.

c) ¿pc1 y pc2 tienen alguna forma de saber si están usando una VLAN para comunicarse?

No.

d) ¿Por qué sólo se ve una trama Ethernet en la captura realizada en la interfaz s2(eth2)?

Porque esta trama van por broadcast.

e) ¿En qué se diferencia la solicitud de ARP que se captura en pc1(eth0) de la misma solicitud que se captura en s1(eth3)?

En el agregado de la etiqueta VLAN. Y por consiguiente la longitud del mensaje.

f) ¿En qué se diferencia el mensaje ICMP Echo request que se captura en pc1(eth0) del mismo mensaje que se captura en s1(eth3)?

En el agregado de la etiqueta VLAN. Y por consiguiente la longitud del mensaje.

9. Indica qué ocurre cuando se hace un ping desde pc1 a pc4, teniendo en cuenta que ambas máquinas se encuentran en la misma subred y conectadas al mismo switch. Compruébalo realizando una captura en pc1(eth0) y otra en s1(eth3), sin necesidad de guardar en un fichero el tráfico capturado

S1 envía el mensaje arp por la vlan 100 al pc4 no pertenecer a la misma no recibe el mensaje nunca

5.2. Configuración de VLAN200

Configura la VLAN200 en los switches que creas necesarios. Para ello edita los ficheros de configuración proporcionados en el apartado anterior y añade la configuración de VLAN 200.

Antes de ejecutar la nueva configuración es necesario borrar la anterior, para ello, reinicia los switches s1 y s2, y a continuación ejecuta sus scripts modificados.

Puedes comprobar la configuración que tiene cada switch escribiendo brctl show.

s1			
s1:~# brctl show			
bridge name	bridge id	STP enabled	interfaces
vs100	8000.8e7dae7e1d95	no	eth0
			eth3,100
vs200	8000.3e77d64351d9	no	eth2
			eth3,200
s1:~#			
s2			
s2:~# brctl show			
bridge name	bridge id	STP enabled	interfaces
vs100	8000.220e835b5be8	no	eth0,100
			eth1,100
			eth2
vs200	8000.aa9bd1ddc696	no	eth0,200
			eth3
s2:~#			

1. Indica qué máquinas se pueden comunicar entre ellas con la configuración de VLAN200.

Pc4 y Pc3

2. Asegúrate de que la caché de ARP de pc4 está vacía, bórrala si es necesario. Arranca tcpdump en las siguientes interfaces: pc4(eth0), s1(eth3), pc3(eth0) y pc1(eth0) guardando esta vez el tráfico capturado en un fichero. Realiza un ping desde pc4 a pc3.

3. Interrumpe las capturas y observa las direcciones Ethernet aprendidas por los switches s1, s2 y s3.

S3 no esta en la captura porque no aprendio ninguna direccion.

s1			
s1:~# brctl showmacs vs200			
port no	mac addr	is local?	ageing timer
2	08:00:27:00:00:03	no	151.76
1	08:00:27:00:00:04	no	151.76
1	3e:77:d6:43:51:d9	yes	0.00
2	ba:57:95:5b:5a:e0	yes	0.00
s1:~#			
s2			
s2:~# brctl showmacs vs200			
port no	mac addr	is local?	ageing timer
2	08:00:27:00:00:03	no	148.61
1	08:00:27:00:00:04	no	148.61
2	aa:9b:d1:dd:c6:96	yes	0.00
1	d2:5e:b1:5d:6b:88	yes	0.00
s2:~#			

4. Analiza las 4 capturas, indica en qué capturas se observa la etiqueta de VLAN en el tráfico y qué identificador de VLAN contiene.

Solamente en la captura de S1 se observa la etiqueta VLAN con el id 200.

5. Indica qué ocurre ahora cuando se hace un ping desde pc1 a pc4, teniendo en cuenta que ambas máquinas se encuentran en la misma subred, conectadas al mismo switch y las interfaces de dicho switch tienen configurada una VLAN. Compruébalo realizando las capturas necesarias, sin necesidad de guardar en un fichero el tráfico capturado.

S1 sigue encapsulando el mensaje ARP por la vlan 100 y es imposible que Pc4 lo reciba.

5.3. Configuración de VLAN300

En este apartado se analiza el comportamiento de 2 VLANs que están conectadas a través de un router. Esta configuración se proporciona en unos scripts que ya se encuentran en el escenario: vlan100y300-s1.sh, vlan100y300-s2.sh y vlan100y300-s3.sh, en s1, s2 y s3 respectivamente.

Antes de ejecutar la nueva configuración es necesario borrar la anterior, para ello, reinicia todos los switches y a continuación ejecuta cada uno de los scripts anteriores.

Puedes comprobar la configuración que tiene en un switch escribiendo brctl show.

s1			
s1:~# brctl show			
bridge name	bridge id	STP enabled	interfaces
vs100	8000.5a40a51f084e	no	eth0 eth3,100
vs300	8000.9afce143657e	no	eth1 eth3,300
s1:~#			
s2			
s2:~# brctl show			
bridge name	bridge id	STP enabled	interfaces
vs100	8000.5a67ff2fee8a	no	eth0,100 eth1,100
vs300	8000.5a67ff2fee8a	no	eth2 eth0,300 eth1,300
s2:~#			
s3			
s3:~# brctl show			
bridge name	bridge id	STP enabled	interfaces
vs100	8000.1a300248f1dc	no	eth0,100 eth1
vs300	8000.1a300248f1dc	no	eth0,300 eth2 eth3
s3:~#			

1. Realiza un ping desde pc6 a pc1. ¿Qué crees que está ocurriendo?

El router esta enviando el trafico a la Vlan 100 por su eth0.

```
pc6:~# ping -c 1 11.0.0.101
PING 11.0.0.101 (11.0.0.101) 56(84) bytes of data.
64 bytes from 11.0.0.101: icmp_seq=1 ttl=63 time=22.9 ms

--- 11.0.0.101 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 22.914/22.914/22.914/0.000 ms
pc6:~#
```

2. Realiza un ping desde pc6 a pc5. ¿Qué crees que está ocurriendo?

```
pc6:~# ping -c 1 12.0.0.105
PING 12.0.0.105 (12.0.0.105) 56(84) bytes of data.
64 bytes from 12.0.0.105: icmp_seq=1 ttl=64 time=13.5 ms

--- 12.0.0.105 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 13.554/13.554/13.554/0.000 ms
pc6:~#
```

El ping se propaga por la vlan 300 entre los switches. Y por ello pueden comunicarse.

3. Suponiendo que la caché de ARP de pc6 está vacía, al realizar un ping de pc6 a pc1, ¿qué solicitudes de ARP hay y en qué interfaces aparecen? ¿Cuáles de ellas tendrán etiqueta VLAN e indica qué etiqueta? Compruébalo realizando las capturas que creas necesarias, sin necesidad de guardar en fichero el tráfico capturado. (Comprueba antes que las cachés de ARP de pc6 y de r1 están vacías, bórralas si es necesario).

Pregunta	Vlan	Interfaces
Pc6 → R1 (pregunta por Pc1)	300	s3(eth0, eth3 sin vlan) s2 (eth1, eth0) s1 (eth3, eth1 sin vlan) r1 (eth1 sin vlan)
R1 → Pc6(pregunta por Pc6)	300	s3(eth0,eth2 sin vlan) s2 (eth1, eth0) s1 (eth3, eth2 sin vlan)
R1 → Pc1 (pregunta por Pc1)	100	r1(eth0 sin vlan) s2 (eth2,eth0, eth1) s3 (eth0, eth1 sin vlan) s1 (eth3, eth0 sin vlan)
Pc1 → R1 (pregunta por R1)	100	S1 (eth3, eth0 sin vlan) s2 (eth0,eth1, eth2 sin vlan) s3(eth0, eth1 sin vlan)

4. Asegúrate de que las cachés de ARP de pc6 y r1 están vacías, bórralas si es necesario. Arranca tcpdump en las siguientes interfaces: pc6(eth0), s3(eth1), r1(eth0), s2(eth0) y pc1(eth0), guardando esta vez el tráfico capturado en un fichero. Realiza un ping desde pc6 a pc1. Supón en qué interfaces aparecerá el tráfico etiquetado y su identificador de VLAN. Comprueba tus suposiciones analizando las capturas, indica en qué capturas se observa la etiqueta de VLAN en el tráfico y qué identificador de VLAN contiene.

Aparecera en S2 con etiqueta vlan 100 y s3 como vlan 300.

Solamente se observa en S2 con vlan 100.