

Práctica 1

Dispositivos

Iván Moreno Martín

GITT

1. Funcionamiento de HUB y Switch

1.1 Comunicación entre máquinas con S1 apagado

1.1 .1

Al hacer ping entre Pc1 y Pc2 solo vamos a recibir tramas en Pc2 porque solo habrá intercambio de paquetes entre máquinas de la misma red ya que S1 está desconectado y esto impide la conectividad entre otras redes.

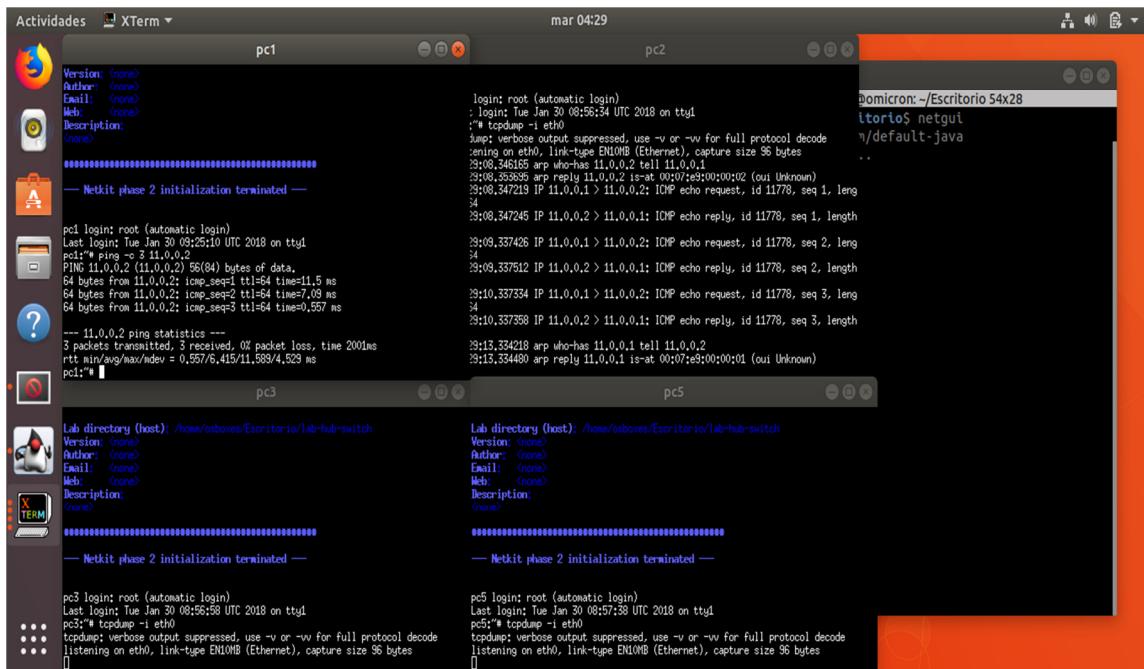
Los paquetes que vamos a encontrar en Pc2 serán una trama con dirección IP destino a Broadcast con la caché de ARP vacía (al inicio).

1.1.2

A continuación hacemos el ping de Pc1 a Pc2 y realizamos la captura con el comando en pc2 en pc3 y pc5:

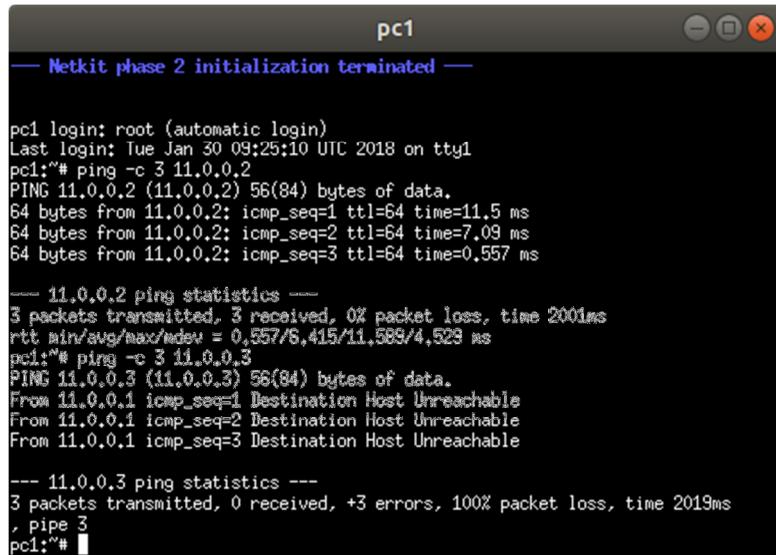
```
$tcpdump -i eth0
```

Como observamos en la captura de pantalla ejercicio1-1.png efectivamente el único que recibe los paquetes es pc2 mientras el resto de Pc's están aún a la escucha.



1.1.3

Comprobamos que no hay conectividad con otras haciendo por ejemplo ping de pc1 pc3 y como vemos en la captura de pantalla de pc1 todos los paquetes se han perdido, ocurriría exactamente lo mismo con pc5



```
pc1
— Netkit phase 2 initialization terminated —

pc1 login: root (automatic login)
Last login: Tue Jan 30 09:25:10 UTC 2018 on tty1
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=11.5 ms
64 bytes from 11.0.0.2: icmp_seq=2 ttl=64 time=7.09 ms
64 bytes from 11.0.0.2: icmp_seq=3 ttl=64 time=0.557 ms

--- 11.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.557/6.415/11.588/4.529 ms
pc1:# ping -c 3 11.0.0.3
PING 11.0.0.3 (11.0.0.3) 56(84) bytes of data.
From 11.0.0.1 icmp_seq=1 Destination Host Unreachable
From 11.0.0.1 icmp_seq=2 Destination Host Unreachable
From 11.0.0.1 icmp_seq=3 Destination Host Unreachable

--- 11.0.0.3 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2019ms
, pipe 3
pc1:#
```

1.2 Comunicación entre máquinas con S1 arrancado

1.2.2

Al repetir el mismo ping y realizando las capturas en las máquinas de pc2 pc3 y pc5 obtendríamos el siguiente camino.

1. Hacemos ping -c 11.0.0.33
2. La trama con dirección broadcast llega a pc2 y al switch
3. El switch comprueba la dirección IP origen y la guarda en su tabla y comprueba la IP destino como no la tiene guardada entonces copia la trama recibida por todas sus bocas menos por aquella por la que le ha llegado
4. las tramas viajan al resto de la red pero serán descartadas por el resto de máquinas por no ir dirigidas a ellas.
5. al mismo tiempo la respuesta del ping viajará al mismo tiempo directamente a pc1 y al switch
6. este último aprende una nueva dirección de memoria la de pc2, y envía la trama a la dirección que tiene de pc1

```

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=0.6 ms
64 bytes from 11.0.0.2: icmp_seq=2 ttl=64 time=0.311 ms
64 bytes from 11.0.0.2: icmp_seq=3 ttl=64 time=0.312 ms

--- 11.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.311/0.755/10.642/4.369 ms
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=0.6 ms
64 bytes from 11.0.0.2: icmp_seq=2 ttl=64 time=0.311 ms
64 bytes from 11.0.0.2: icmp_seq=3 ttl=64 time=0.312 ms

--- 11.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.311/0.755/10.642/4.369 ms
pc1:~# [REDACTED]

pc2:~# tcptrace -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
08:41:25.151438 (P 11.0.0.1 > 11.0.0.2: ICMP echo request, id 45570, seq 1, length 64
08:41:25.193493 (P 11.0.0.2 > 11.0.0.1: ICMP echo reply, id 45570, seq 1, length 64
08:41:26.158291 (P 11.0.0.1 > 11.0.0.2: ICMP echo request, id 45570, seq 2, length 64
08:41:26.158439 (P 11.0.0.2 > 11.0.0.1: ICMP echo reply, id 45570, seq 2, length 64
08:41:27.157375 (P 11.0.0.1 > 11.0.0.2: ICMP echo request, id 45570, seq 3, length 64
00:41:27.157419 (P 11.0.0.2 > 11.0.0.1: ICMP echo reply, id 45570, seq 3, length 64
08:41:30.142038 arp whc-has 11.0.0.1 tel 11.0.0.2
08:41:30.143115 arp reply 11.0.0.1 is-at 00:07:E9:00:00:01 (ou: Unknown)
[REDACTED]

pc3:~# Author: <none>
pc3:~# Email: <none>
pc3:~# Web: <none>
pc3:~# Description: <none>
[REDACTED]
Netkit phase 2 initialization terminated

pc5:~# Author: <none>
pc5:~# Email: <none>
pc5:~# Web: <none>
pc5:~# Description: <none>
[REDACTED]
Netkit phase 2 initialization terminated

```

1.2.3

Comprobamos la cache de ARP con

```
arp -a
```

Si esta entonces

```
arp -d 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=12.2 ms
64 bytes from 11.0.0.2: icmp_seq=2 ttl=64 time=2.80 ms
64 bytes from 11.0.0.2: icmp_seq=3 ttl=64 time=4.42 ms

--- 11.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 2.802/6.506/12.291/4.144 ms
pc1:~# arp -a
? (11.0.0.2) at 00:07:E9:00:00:02 [ether] on eth0
pc1:~# arp -d 11.0.0.2
pc1:~# arp -a
-bash: arp-a: command not found
pc1:~# arp -a
? (11.0.0.2) at <incomplete> on eth0
pc1:~#

```

1.2.4

Volvemos a realizar el ping y comprobamos que efectivamente la solicitud de ping llega a pc3 y a pc5 porque el switch ha realizado copias de la trama por todas sus bocas menos por la que le ha llegado

1.2.5

- La solicitud ARP llega a pc3 y a pc5 porque el switch copia la trama por todas sus bocas excepto por aquella por la que ha llegado la trama
- No les llega la respuesta de ICMP porque el switch ya conoce la dirección de pc1
- porque pc2 conoce el origen de la trama ICMP que pregunta por su IP

1.2.6

Pc1

eth0

```
inet addr:11.0.0.1 Bcast:11.0.0.255 mask:255.255.255.0
```

Pc2

eth0

```
inet addr:11.0.0.2 Bcast:11.0.0.255 mask:255.255.255.0
```

Pc3

eth0

```
inet addr:11.0.0.3 Bcast:11.0.0.255 mask:255.255.255.0
```

Pc4

eth0

```
inet addr: 11.0.0.4 Bcast:111.0.0.255 mask 255.255.255.0
```

Pc5

eth0

```
inet addr: 11.0.0.5 Bcast:11.0.0.255 mask 255.255.255.0
```

Pc6

eth0

inet addr:11.0.0.6 Bcast:11.0.0.255 mask 255.255.255.0

1.2.7

Como podemos observar en la imagen las direcciones aprendidas se corresponden con Pc1 y Pc2

```
s1:~# brctl showmacs s1
port no mac addr      is local?    ageing timer
  1  00:07:e9:00:00:01  no          189.66
  1  00:07:e9:00:00:02  no          189.66
  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:~#
```

Pasados 300 ms volvemos a comprobar las macs y vemos que el switch las olvida

```
s1:~# brctl showmacs s1
port no mac addr      is local?    ageing timer
  1  00:07:e9:00:00:01  no          189.66
  1  00:07:e9:00:00:02  no          189.66
  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:~# brctl showmacs s1
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:~#
```

1.2.8

Una forma de hacerlos es hacer ping de pc1 a todas las máquinas y mirar la tabla de direcciones mac aprendidas en el switch S1

port no	mac addr	is local?	ageing timer
1	00:07:e9:00:00:01	no	21.67
1	00:07:e9:00:00:02	no	46.04
2	00:07:e9:00:00:03	no	40.72
2	00:07:e9:00:00:04	no	35.04
3	00:07:e9:00:00:05	no	28.68
3	00:07:e9:00:00:06	no	21.67
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

2. Redes conectadas a través de switch y router

2.1 COMUNICACIÓN ENTRE PC1 Y PC2

2.1.1

Como vemos en la imagen cuando hacemos ping de pc1 a pc2 existe conectividad entre ambos vemos que esa trama también le llega a pc3, pero no le llega a pc5

```
pc1 login: root (automatic login)
pc1# ping -c 3 13.0.0.20
PING 13.0.0.20 (13.0.0.20) 56(84) bytes of data.
64 bytes from 13.0.0.20: icmp_seq=1 ttl=64 time=0.5 ms
64 bytes from 13.0.0.20: icmp_seq=2 ttl=64 time=1.23 ms
64 bytes from 13.0.0.20: icmp_seq=3 ttl=64 time=1.23 ms
--- 13.0.0.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 1.225/4.342/10.568/4.402 ms
pc1# 

pc2 login: root (automatic login)
pc2# tcptrace -i eth0
tcptrace: command not found
pc2# tcptdump -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
12:34:56.838506 arp who-has pc1 tell pc1
12:34:56.841041 arp who-has pc2 tell pc1
12:34:56.838548 IP pc1 > pc2: ICMP echo request, id 27650, seq 1, length 64
12:34:57.830097 IP pc1 > pc2: ICMP echo request, id 27650, seq 2, length 64
12:34:57.830137 IP pc2 > pc1: ICMP echo reply, id 27650, seq 2, length 64
12:34:58.829005 IP pc1 > pc2: ICMP echo request, id 27650, seq 3, length 64
12:34:59.829045 IP pc2 > pc1: ICMP echo reply, id 27650, seq 3, length 64
12:35:01.834047 arp who-has pc1 tell pc2
12:35:01.834047 arp reply pc1 is-at e6:31:1e:0a:2a:59 (oui Unknown)
pc2# 

pc3 login: root (automatic login)
pc3# tcptdump -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
12:34:56.841064 arp reply pc2 is-at f2:4f:99:b0:ac:89 (oui Unknown)
12:34:56.838568 IP pc1 > pc2: ICMP echo request, id 27650, seq 1, length 64
12:34:57.830097 IP pc1 > pc2: ICMP echo request, id 27650, seq 2, length 64
12:34:57.830202 IP pc2 > pc1: ICMP echo reply, id 27650, seq 2, length 64
12:34:58.828996 IP pc1 > pc2: ICMP echo request, id 27650, seq 3, length 64
12:35:01.834591 arp who-has pc1 tell pc3
12:35:01.834591 arp reply pc1 is-at e6:31:1e:0a:2a:59 (oui Unknown)
pc3# 

pc5 login: root (automatic login)
pc5# tcptdump -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
pc5# 
```

2.1.2

Como podemos observar en el Switch ha aprendido dos direcciones Mac que no son las suyas

En eth0 aprende la dirección Ethernet de pc1

En eth1 aprende la dirección de pc2

```
s1
Email: <none>
Web: <none>
Description:
<none>

=====
— Netkit phase 2 initialization terminated —

s1 login: root (automatic login)
s1:~# brctl showmacs s1
port no mac addr          is local?    ageing timer
  1  22:0a:5b:8a:28:f2    yes          0.00
  2  9a:2d:fb:8:af:14:ef  yes          0.00
  1  e6:31:1e:0a:2a:59    no           262.19
  2  f2:4f:39:fb:0:ac:88  no           262.19
s1:~# brctl showmacs s1
port no mac addr          is local?    ageing timer
  1  22:0a:5b:8a:28:f2    yes          0.00
  2  9a:2d:fb:8:af:14:ef  yes          0.00
  1  e6:31:1e:0a:2a:59    no           17.99
  2  f2:4f:39:fb:0:ac:88  no           17.99
s1:~#
```

podemos comprobarlo con:

brctl showmacs s1

Y después en las maquinas vemos con ifconfig.

2.1.3

En principio llegarán tramas a pc3 por estar en la misma subred de pc2 aunque la descarta por no ir dirigida a él. Comprobamos haciendo capturas en pc4 y en pc5 para ver si llegan tramas. Como vemos en la imagen solo llegan tramas a pc2 pc3

```
pc1
TX packets:16 errors:0 dropped:0 overrun:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1000 (1000.0 B) 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 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:4 errors:0 dropped:0 overrun:0 frame:0
TX packets:4 errors:0 dropped:0 overrun:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:200 (200.0 B) TX bytes:200 (200.0 B)

pc1:"# ping -c 5 127.0.0.20
PING 127.0.0.20(127.0.0.20) 56(64) bytes of data.
64 bytes from 13.0.0.20: icmp_seq=1 ttl=64 time=0.105 ms
64 bytes from 13.0.0.20: icmp_seq=2 ttl=64 time=0.341 ms
64 bytes from 13.0.0.20: icmp_seq=3 ttl=64 time=0.730 ms

--- 13.0.0.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 0.730/4.058/10.555/4.575 ms
pc1:"# 

pc2
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:4 errors:0 dropped:0 overrun:0 frame:0
TX packets:4 errors:0 dropped:0 overrun:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:200 (200.0 B) TX bytes:200 (200.0 B)

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
12:57:43.220228 IP pc1 > pc2: ICMP echo request, id 63234, seq 1, length 64
12:57:43.220228 IP pc1 > pc2: ICMP echo reply, id 63234, seq 1, length 64
12:57:43.219483 IP pc1 > pc2: ICMP echo request, id 63234, seq 1, length 64
12:57:43.219483 IP pc2 > pc1: ICMP echo reply, id 63234, seq 1, length 64
12:57:43.219501 IP pc2 > pc1: ICMP echo reply, id 63234, seq 1, length 64
12:57:44.211759 IP pc1 > pc2: ICMP echo request, id 63234, seq 2, length 64
12:57:44.211759 IP pc2 > pc1: ICMP echo reply, id 63234, seq 2, length 64
12:57:44.211798 IP pc1 > pc2: ICMP echo request, id 63234, seq 2, length 64
12:57:44.211798 IP pc2 > pc1: ICMP echo reply, id 63234, seq 2, length 64
12:57:45.216509 IP pc1 > pc2: ICMP echo request, id 63234, seq 3, length 64
12:57:45.216509 IP pc2 > pc1: ICMP echo reply, id 63234, seq 3, length 64
12:57:46.213627 IP pc1 > pc2: ICMP echo request, id 63234, seq 3, length 64
12:57:46.213627 IP pc2 > pc1: ICMP echo reply, id 63234, seq 3, length 64
12:57:46.214083 arp who-has pc1 tell pc2
12:57:46.214083 arp reply pc1 is-at 6:31:1e:0:a:2a:59 (oui Unknown)
pc2:"# 

pc3
12:57:45.7332002 IP pc1 > pc2: ICMP echo request, id 27650, seq 2, length 64
12:57:45.7332005 IP pc1 > pc2: ICMP echo reply, id 27650, seq 2, length 64
12:57:45.822996 IP pc1 > pc2: ICMP echo request, id 27650, seq 3, length 64
12:57:45.923159 IP pc1 > pc2: ICMP echo reply, id 27650, seq 3, length 64
12:35:01.0134391 arp who-has pc1 tell pc2
12:35:01.0134391 arp reply pc1 is-at 6:31:1e:0:a:2a:59 (oui Unknown)
pc3:"# 

pc4
10 packets captured
10 packets received by filter
0 packets dropped by kernel
pc4:"# 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
12:57:43.220103 IP pc1 > pc2: ICMP echo request, id 63234, seq 1, length 64
12:57:43.219489 IP pc1 > pc2: ICMP echo request, id 63234, seq 1, length 64
12:57:43.219511 IP pc2 > pc1: ICMP echo reply, id 63234, seq 1, length 64
12:57:44.211754 IP pc1 > pc2: ICMP echo request, id 63234, seq 2, length 64
12:57:44.212065 IP pc2 > pc1: ICMP echo reply, id 63234, seq 2, length 64
12:57:44.212068 IP pc1 > pc2: ICMP echo request, id 63234, seq 3, length 64
12:57:44.212068 IP pc2 > pc1: ICMP echo reply, id 63234, seq 3, length 64
12:57:45.216698 IP pc1 > pc2: ICMP echo request, id 63234, seq 3, length 64
12:57:45.216698 IP pc2 > pc1: ICMP echo reply, id 63234, seq 3, length 64
12:57:46.213282 arp who-has pc1 tell pc2
12:57:46.214075 arp reply pc1 is-at 6:31:1e:0:a:2a:59 (oui Unknown)
pc4:"# 

pc5
Version: <none>
Author: <none>
Email: <none>
Web: <none>
Description: <none>
***** 
— Netkit phase 2 initialization terminated —
***** 

ic5$5 login: root (automatic login)
ic5$5:"# 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

0 packets captured
0 packets received by filter
0 packets dropped by kernel
ic5$5:"# 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
ic5$5:"# 

Lab directory (host): /home/alumnos/ivmoreno/Escritorio/lab-switch-router
Version: <none>
Author: <none>
Email: <none>
Web: <none>
Description: <none>
***** 
— Netkit phase 2 initialization terminated —
***** 
```

2.2 Comunicación de Pc1 a Pc3

2.2.1

Al hacer ping de pc1 a pc3 vemos que nos sale un mensaje de máquina inalcanzable. Además comprobamos que la trama no pasa por pc2

2.2.2

En este caso podemos solucionarlo configurando r1 metiendo Gateway en pc1 y en pc3 y con eso se solucionará el problema de conectividad entre las dos máquinas.

2.2.3

Para este caso vamos a configurar (en nuestro caso con mcedit):

Pc1 -> gw 13.0.0.15

Pc3 -> gw 11.0.0.5

En r1 tenemos que configurar dos direcciones en eth0 y en eth1

inet eth0

iface eth0 inet static

address 13.0.0.5

network 13.0.0.0

netmask 255.255.255.0

broadcast 13.0.0.255

inet eth1

iface eth1 inet static

address 11.0.0.5

network 11.0.0.0

netmask 255.255.255.0

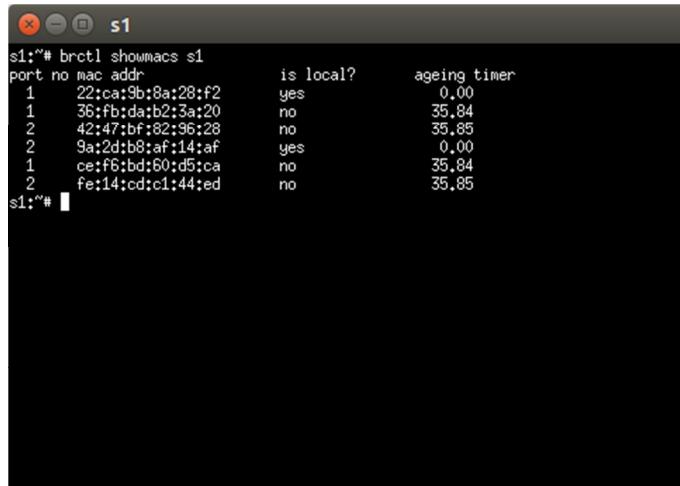
broadcast 11.0.0.255

2.2.4

Como vemos en la captura que hemos realizado pc3 recibe solicitud ARP preguntando por su dirección IP en la siguiente línea manda que respuesta a who-as <dirIP> tell <dirIP pc3>

2.2.5

Vemos en la captura que aprende 4 direcciones



```
s1:~# brctl showmacs s1
port no mac addr      is local?    ageing timer
 1  22:ca:9b:8a:28:f2  yes          0.00
 1  36:fb:da:b2:3a:20  no           35.84
 2  42:47:bf:82:36:28  no           35.85
 2  9a:2d:b8:af:14:af  yes          0.00
 1  ce:f6:bd:60:d5:ca  no           35.84
 2  fe:14:cd:c1:44:ed  no           35.85
s1:~#
```

2.3 COMUNICACION ENTRE PC2 Y PC4

2.3.1

Como podemos ver al hacer ping de pc2 a pc4 nos dice que host unreachable

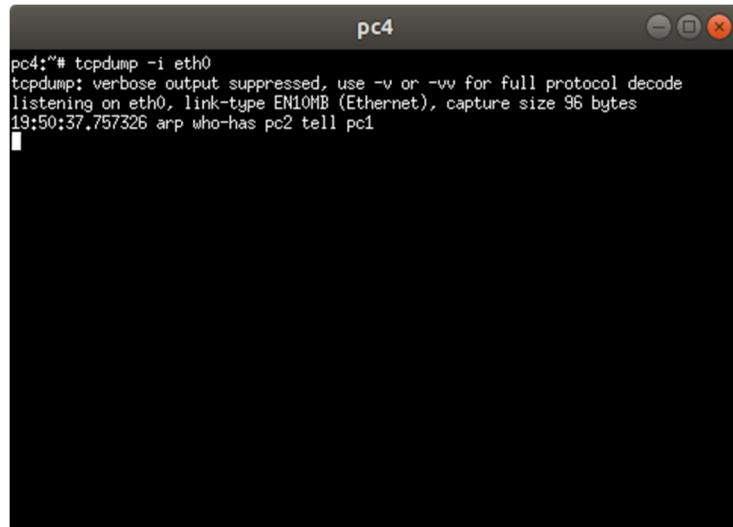
2.3.2

En este caso solo nos hace falta arrancar s2 no nos hace falta una configuración adicional en otras máquinas

2.3.3

Comprobamos por un lado que la cache de ARP en los pcs están vacías con el comando arp -a y en los switches comprobamos sus tablas con **brctl showmacs <nombre_swith>** y vemos que ambos casos sus esta todo vacío.

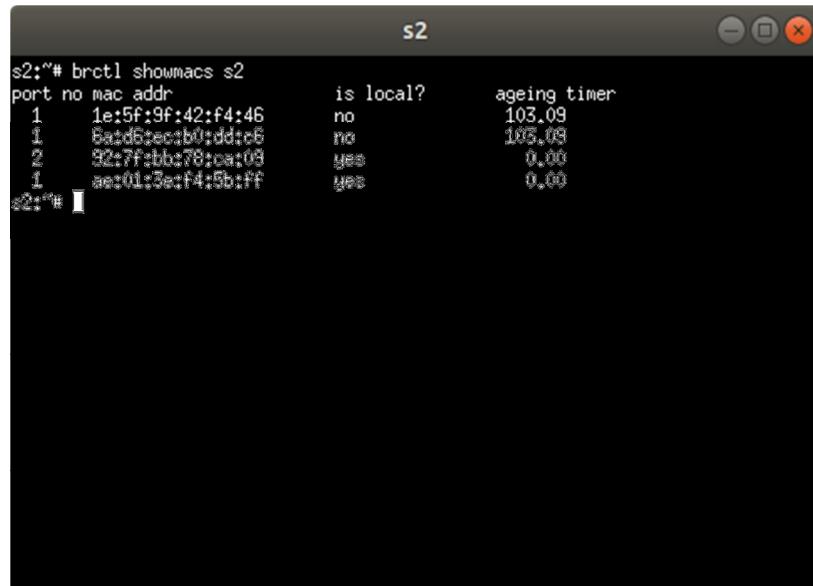
A continuación realizamos el ping y esperamos que llegue una trama de solicitud de ARP que pc4 desecha por no ir dirigido a él, y podemos verlo en la siguiente imagen



```
pc4:~# 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
19:50:37.757326 arp who-has pc2 tell pc1
```

2.3.4

Comprobamos que direcciones aprende S2 cuando pc1 hace un ping a pc2 y como se ve a continuación s2 aprende las direcciones de pc1 y pc2, por el puerto eth0.



```
s2:~# brctl showmacs s2
port no mac addr          is local?    ageing timer
  1  1e:5f:9f:42:f4:46    no           103.09
  1  6add6:ec:b0:dd:c6   no           105.09
  2  32:7f:bb:78:ca:08   yes          0.00
  1  ac:01:3e:f4:5b:ff   yes          0.00
s2:~#
```

2.3.5

Los paquetes que pasan por Pc1 serán primero solicitudes de ARP preguntando por Pc4 a continuación Pc1 no contesta a dicha solicitud por no ir dirigida a él, sin embargo va a recibir tanto los paquete de pc2 y pc4. Esto se debe a que cuando la trama viaja por la red primero sale de pc2 al switch y al router, y las respuestas de pc4 pasaran por el switch s2 y pasaran por el router y el switch. Como pc1 se encuentra detrás de un hub que esta entre el router r1 el switch s1 y el switch s1 y s2 le llegaran todos los paquetes dirigidos a pc2 y viceversa.

pc1:"# 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

20:11:08.840420 arp who-has pc4 tell pc2
20:11:08.841769 arp reply pc4 is-at 4a:97:2f:93:6b:a6 (oui Unknown)
20:11:08.842163 IP pc2 > pc4: ICMP echo request, id 12803, seq 1, length 64
20:11:08.842426 IP pc4 > pc2: ICMP echo reply, id 12803, seq 1, length 64
20:11:09.829231 IP pc2 > pc4: ICMP echo request, id 12803, seq 2, length 64
20:11:09.832496 IP pc4 > pc2: ICMP echo reply, id 12803, seq 2, length 64
20:11:10.832726 IP pc2 > pc4: ICMP echo request, id 12803, seq 3, length 64
20:11:10.832733 IP pc4 > pc2: ICMP echo reply, id 12803, seq 3, length 64
20:11:13.836935 arp who-has pc2 tell pc4
20:11:13.838374 arp reply pc2 is-at 6a:d6:ec:b0:dd:c6 (oui Unknown)

2.3.6

Como vemos en la imagen inferior S1 por un lado aprende las direcciones de pc4 por eth0 y la de pc2 por eth1. En el caso de S2 vemos que por eth0 aprende pc2 y por eth1 aprende pc4

pc1:"# 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

20:11:08.840420 arp who-has pc4 tell pc2
20:11:08.841769 arp reply pc4 is-at 4a:97:2f:93:6b:a6 (oui Unknown)
20:11:08.842163 IP pc2 > pc4: ICMP echo request, id 12803, seq 1, length 64
20:11:08.842426 IP pc4 > pc2: ICMP echo reply, id 12803, seq 1, length 64
20:11:09.829231 IP pc2 > pc4: ICMP echo request, id 12803, seq 2, length 64
20:11:09.832496 IP pc4 > pc2: ICMP echo reply, id 12803, seq 2, length 64
20:11:10.832728 IP pc2 > pc4: ICMP echo request, id 12803, seq 3, length 64
20:11:10.832733 IP pc4 > pc2: ICMP echo reply, id 12803, seq 3, length 64
20:11:13.836935 arp who-has pc2 tell pc4
20:11:13.838374 arp reply pc2 is-at 6a:d6:ec:b0:dd:c6 (oui Unknown)

pc2:"# ping -c 3 13.0.0.40

PING 13.0.0.40 (13.0.0.40) 56(84) bytes of data.
64 bytes from 13.0.0.40: icmp_seq=1 ttl=64 time=12.9 ms
64 bytes from 13.0.0.40: icmp_seq=2 ttl=64 time=5.63 ms
64 bytes from 13.0.0.40: icmp_seq=3 ttl=64 time=1.37 ms

--- 13.0.0.40 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 199ms
rtt min/avg/max/mdev = 1.375/6.653/12.952/4.781 ms

s1:"# brctl showmacs s1

port no	mac addr	is local?	ageing timer
1	26:85:a1:fc:1e:54	yes	0.00
1	4a:97:2f:93:6b:a6	no	5.61
2	6a:d6:ec:b0:dd:c6	no	5.61
2	92:cc:22:16:ff:77	yes	0.00

s1:"#

s2:"# brctl showmacs s2

port no	mac addr	is local?	ageing timer
2	4a:97:2f:93:6b:a6	no	8.31
1	6a:d6:ec:b0:dd:c6	no	8.31
2	92:7fb:78:c0:09	yes	0.00
1	ae:01:3e:f4:5b:ff	yes	0.00

s2:"#

2.4 COMUNICACION ENTRE PC5 Y PC2

2.4.1

Comprobamos que al hacer un ping de pc5 a pc2 nos salta el mensaje de máquina inalcanzable

2.4.2

Para que pueda funcionar correctamente un ping de pc5 a pc2 añadimos gateways a las máquinas y configuramos las interfaces de r2

2.4.3

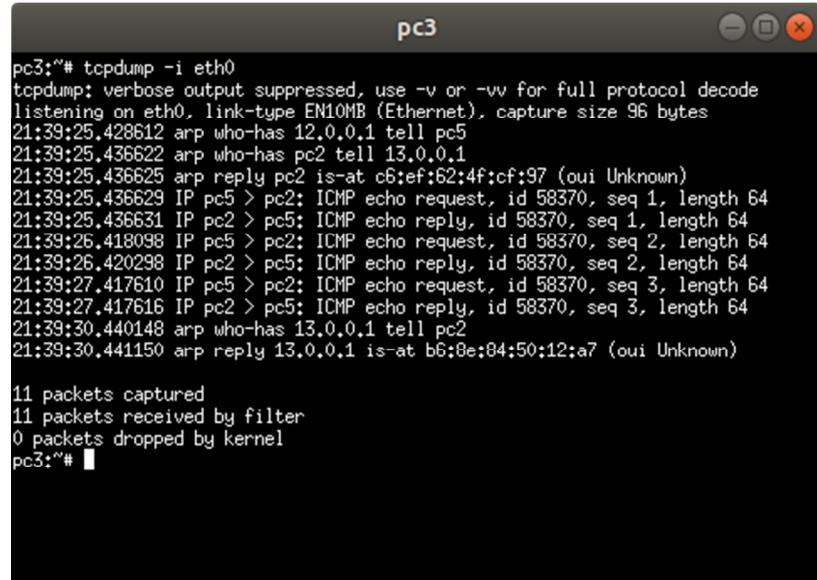
Para poder realizar un ping de pc5 a pc2 vamos a añadir a pc5 un gateway a 12.0.0.1, y a pc2 un gateway a 13.0.0.1 y por otro lado vamos a configurar r2 de la siguiente manera:

```
auto eth0  
  
iface eth0 inet static  
  
    address 13.0.0.1  
  
    network 13.0.0.0  
  
    netmask 255.255.255.0  
  
    broadcast 13.0.0.255
```

```
auto eth1  
  
iface eth1 inet static  
  
    address 12.0.0.1  
  
    network 12.0.0.0  
  
    netmask 255.255.255.0  
  
    broadcast 12.0.0.255
```

2.4.4

Realizamos una captura en pc3 para saber que paquetes pasaran por él, a primera vista por estar en la misma subred que pc2 le llegan todos los paquetes aunque los descartará por no ir dirigidos a él.



```
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
21:39:25.428612 arp who-has 12.0.0.1 tell pc5
21:39:25.436622 arp who-has pc2 tell 13.0.0.1
21:39:25.436625 arp reply pc2 is-at c6:ef:62:4f:cf:97 (oui Unknown)
21:39:25.436629 IP pc5 > pc2: ICMP echo request, id 58370, seq 1, length 64
21:39:25.436631 IP pc2 > pc5: ICMP echo reply, id 58370, seq 1, length 64
21:39:26.418098 IP pc5 > pc2: ICMP echo request, id 58370, seq 2, length 64
21:39:26.420298 IP pc2 > pc5: ICMP echo reply, id 58370, seq 2, length 64
21:39:27.417610 IP pc5 > pc2: ICMP echo request, id 58370, seq 3, length 64
21:39:27.417616 IP pc2 > pc5: ICMP echo reply, id 58370, seq 3, length 64
21:39:30.440148 arp who-has 13.0.0.1 tell pc2
21:39:30.441150 arp reply 13.0.0.1 is-at b6:8e:84:50:12:a7 (oui Unknown)

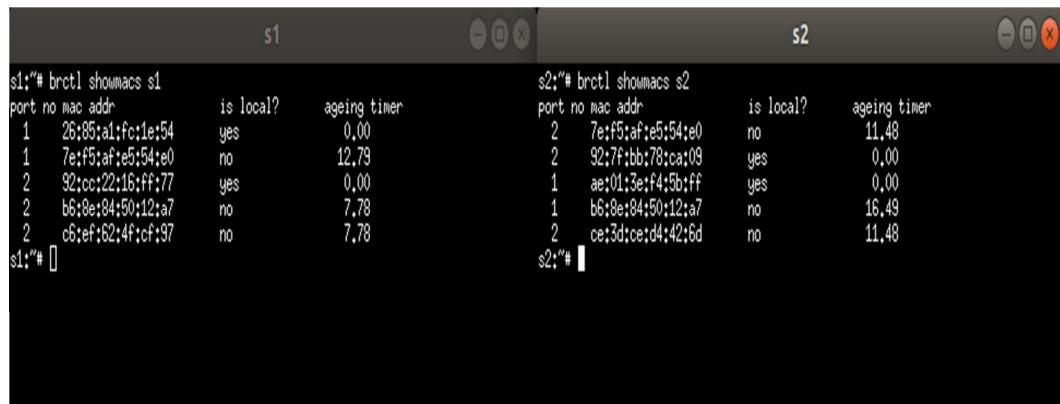
11 packets captured
11 packets received by filter
0 packets dropped by kernel
pc3:~#
```

Como observamos en la captura efectivamente se ven todos los paquetes intercambiados entre pc5 y pc2 ya que tanto pc3 como pc3 están conectados al mismo hub.

2.4.5

A través del comando **brctl showmacs** vemos que en el caso de s1 aprende las direcciones de pc5 por eth0, por eth1 aprende la de pc2 y r2, y en eth0 la de pc5.

En el caso de s2 por eth1 aprende las direcciones de pc5 y r2 y por eth0 aprende pc2.



Port	Mac Address	Local?	Ageing Timer
1	26:85:a1:fc:1e:54	yes	0.00
1	7e:f5:af:e5:54:e0	no	12.79
2	92:cc:22:16:f7:77	yes	0.00
2	b6:8e:84:50:12:a7	no	7.78
2	c6:ef:62:4f:cf:97	no	7.78

Port	Mac Address	Local?	Ageing Timer
2	7e:f5:af:e5:54:e0	no	11.48
2	92:7f:bb:78:ca:09	yes	0.00
1	ae:01:3e:f4:5b:ff	yes	0.00
1	b6:8e:84:50:12:a7	no	16.49
2	ce:3d:ce:d4:42:6d	no	11.48

2.4.6

No responde porque el que responde es el router r2, por eso la siguiente petición de ARP está dirigida a la dirección 13.0.0.1.

3.PROXI ARP

3.1

Para activar proxy arp haremos lo siguiente en r1:

Pc1 --> Pc2

- 1) echo 1> /proc/sys/net/ipv4/conf/eth0/proxy_arp
- 2) arp -i eth0 -Ds 12.0.0.20 eth0 netmask 255.255.255.0
- 3) route add -host 12.0.0.20 dev eth1

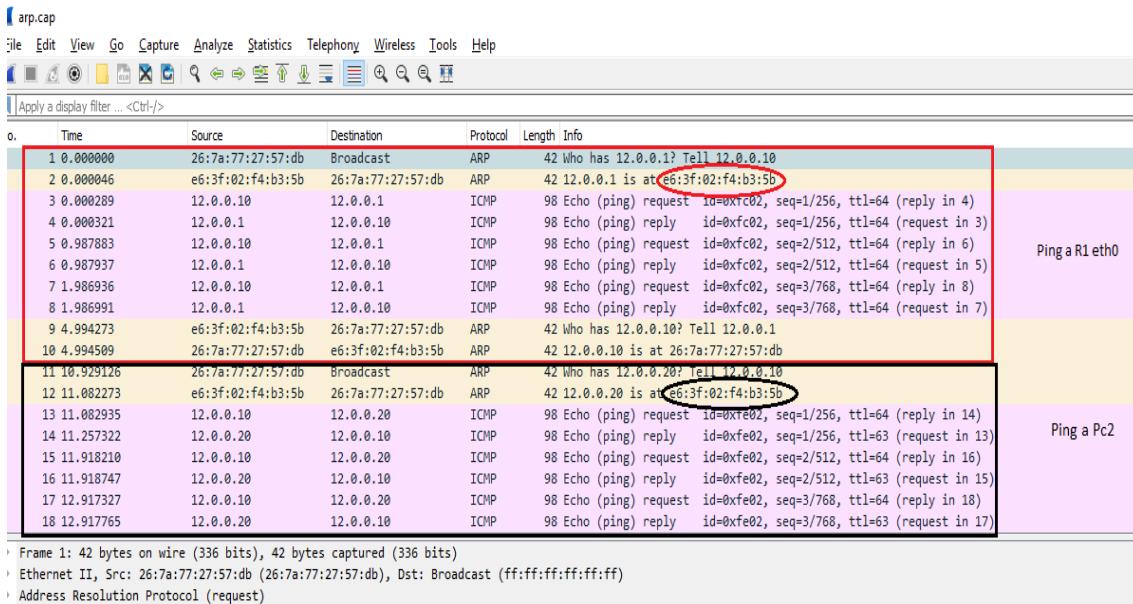
Pc2 --> Pc1

- 1) echo 1> /proc/sys/net/ipv4/conf/eth1/proxy_arp
- 2) arp -i eth1 -Ds 12.0.0.10 eth1 netmask 255.255.255.0

Nota : para este último caso no es necesario hacer route add -host.

En nuestro caso hemos configurado las interfaces del router r1 para que pueda haber intercambio entre Pc1 <-> Pc2, y hemos agregado una ruta específica para Pc2 en la interfaz eth1

3.2



Como podemos observar en la imagen superior podemos saber que R1 está haciendo proxy ARP porque contesta con la misma dirección Ethernet cuando se hace ping a la interfaz eth0 de r1 o bien a pc2

3.3

Podemos ver con el comando **arp -a** que las dos IP's guardadas en la caché de ARP de Pc1 tienen la misma dirección Ethernet.

4. IP Aliasing

4.1

Configuramos en Pc2 → gateway 13.0.0.2

Configuramos en R2 → ifconfig eth1:0 13.0.0.2 netmask 255.255.255.0

Por ultimo añadimos en pc3 una ruta a la red con

```
up route add -host 13.0.0.20 gw 11.0.0.2
```

4.2

Como podemos ver en la siguiente figura podría parecer proxy ARP porque contesta con la misma dirección Ethernet a dos peticiones ARP realizadas a IP's distintas, pero se trata de IP's distintas en una misma interfaz, por lo que se deduce que estamos haciendo IPAliasing.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	be:09:b4:27:9f:64	Broadcast	ARP	42	Who has 13.0.0.2? Tell 13.0.0.20
2	0.000254	32:c4:72:e2:45:ff	be:09:b4:27:9f:64	ARP	42	13.0.0.2 is at 32:c4:72:e2:45:ff
3	0.000068	13.0.0.20	11.0.0.30	ICMP	98	Echo (ping) request id=0xe102, seq=1/256, ttl=64 (reply in 4)
4	0.001953	11.0.0.30	13.0.0.20	ICMP	98	Echo (ping) reply id=0xe102, seq=1/256, ttl=63 (request in 3)
5	0.992743	13.0.0.20	11.0.0.30	ICMP	98	Echo (ping) request id=0xe102, seq=2/512, ttl=64 (reply in 6)
6	0.993113	11.0.0.30	13.0.0.20	ICMP	98	Echo (ping) reply id=0xe102, seq=2/512, ttl=63 (request in 5)
7	1.991453	13.0.0.20	11.0.0.30	ICMP	98	Echo (ping) request id=0xe102, seq=3/768, ttl=64 (reply in 8)
8	1.991622	11.0.0.30	13.0.0.20	ICMP	98	Echo (ping) reply id=0xe102, seq=3/768, ttl=63 (request in 7)
9	4.996747	32:c4:72:e2:45:ff	be:09:b4:27:9f:64	ARP	42	Who has 13.0.0.20? Tell 13.0.0.2
10	4.996959	be:09:b4:27:9f:64	32:c4:72:e2:45:ff	ARP	42	13.0.0.20 is at be:09:b4:27:9f:64
11	8.928453	fe:38:fd:b7:0b:69	Broadcast	ARP	42	Who has 12.0.0.2? Tell 12.0.0.50
12	8.928462	32:c4:72:e2:45:ff	fe:38:fd:b7:0b:69	ARP	42	12.0.0.2 is at 32:c4:72:e2:45:ff
13	8.928507	12.0.0.50	12.0.0.2	ICMP	98	Echo (ping) request id=0xed02, seq=1/256, ttl=64 (reply in 14)
14	8.928525	12.0.0.2	12.0.0.50	ICMP	98	Echo (ping) reply id=0xed02, seq=1/256, ttl=64 (request in 13)
15	9.921235	12.0.0.50	12.0.0.2	ICMP	98	Echo (ping) request id=0xed02, seq=2/512, ttl=64 (reply in 16)
16	9.921276	12.0.0.2	12.0.0.50	ICMP	98	Echo (ping) reply id=0xed02, seq=2/512, ttl=64 (request in 15)
17	10.920034	12.0.0.50	12.0.0.2	ICMP	98	Echo (ping) request id=0xed02, seq=3/768, ttl=64 (reply in 18)
18	10.920048	12.0.0.2	12.0.0.50	ICMP	98	Echo (ping) reply id=0xed02, seq=3/768, ttl=64 (request in 17)

4.3

Comprobamos haciendo ping de pc1 a pc5 que no hay retorno de ping pero si llega a pc5. Para poder hacer ping a pc5 realizamos las siguientes modificaciones

```
r2 --> route add -host 13.0.0.10 gw 11.0.0.1
```

```
r1 --> ifconfig eth0:0 12.0.0.1 netmask 255.255.255.0
```

```
route add -host 12.0.0.50 gw 11.0.0.2
```

4.4

Comprobamos si con la configuración anterior podemos hacer ping de pc1 a pc4 y no hay conexión para ello realizamos la siguiente modificación:

```
pc4 --> up route add -host 13.0.0.10 gw 12.0.0.1
```

4.5

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
18:38:05.084935 IP pc4 > pc1: ICMP echo request, id 4098, seq 1, length 64
18:38:05.087739 IP pc4 > pc1: ICMP echo request, id 4098, seq 1, length 64
18:38:05.086092 IP pc1 > pc4: ICMP echo reply, id 4098, seq 1, length 64
18:38:05.086107 IP 13.0.0.1 > pc1: ICMP redirect pc4 to host pc4, length 92
18:38:05.086158 IP pc1 > pc4: ICMP echo reply, id 4098, seq 1, length 64
18:38:06.088083 IP pc4 > pc1: ICMP echo request, id 4098, seq 2, length 64
18:38:06.088138 IP pc4 > pc1: ICMP echo request, id 4098, seq 2, length 64
18:38:06.088385 IP pc1 > pc4: ICMP echo reply, id 4098, seq 2, length 64
18:38:06.088415 IP 13.0.0.1 > pc1: ICMP redirect pc4 to host pc4, length 92
18:38:06.088472 IP pc1 > pc4: ICMP echo reply, id 4098, seq 2, length 64
18:38:07.087161 IP pc4 > pc1: ICMP echo request, id 4098, seq 3, length 64
18:38:07.087242 IP pc4 > pc1: ICMP echo request, id 4098, seq 3, length 64
18:38:07.089527 IP pc1 > pc4: ICMP echo reply, id 4098, seq 3, length 64
18:38:07.089563 IP 13.0.0.1 > pc1: ICMP redirect pc4 to host pc4, length 92
18:38:07.089580 IP pc1 > pc4: ICMP echo reply, id 4098, seq 3, length 64
18:38:10.071541 arp who-has pc1 tell 13.0.0.1
18:38:10.071653 arp who-has pc4 tell 12.0.0.1
18:38:10.077822 arp reply pc4 is-at ee:61:d2:a9:01:35 (oui Unknown)
18:38:10.077828 arp who-has 13.0.0.1 tell pc1
18:38:10.077859 arp reply 13.0.0.1 is-at 3a:88:7b:15:b0:95 (oui Unknown)
18:38:10.077834 arp reply pc1 is-at 6e:8f:2c:75:17:86 (oui Unknown)
```

5.VLANS

1.

Para ahorrar tiempo y ver la conectividad entre pcs lo que hacemos es desde pc1 hacer ping a los pcs de las distintas redes incluyendo la suya propia. Hemos comprobado que efectivamente tiene conexión con todas las redes.

Hacemos lo mismo con un pc de cada una de las redes y obtenemos como resultado que todas las maquinas tienen conexión con todas las máquinas.

- Pc1 → Conexión con todas las máquinas
- Pc2 → Conexión con todas las máquinas
- Pc3 → Conexión con todas las máquinas
- Pc4 → Conexión con todas las máquinas
- Pc5 → Conexión con todas las máquinas
- Pc6 → Conexión con todas las máquinas

2.

Comprobamos que la cache de ARP de pc1 esta vacía con **arp -a**. A continuación realizamos las capturas en s1 s2 s3 y r1. También haremos captura en pc2 para ver que llega el ping(aunque se ha comprobado en el apartado anterior y hemos visto que hay conexión).

```

s1:# tcpdump -i eth0 -s0
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
14:21:11.805858 arp who-has 11.0.0.102 tell 11.0.0.101
14:21:11.819645 arp reply 11.0.0.102 is-at 42:0d:f3:e1:4b:4b (oui Unknown)
14:21:11.819729 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 24322, seq 1, length 64
14:21:11.826696 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 24322, seq 1, length 64
14:21:12.793937 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 24322, seq 2, length 64
14:21:12.799120 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 24322, seq 2, length 64
14:21:13.792954 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 24322, seq 3, length 64
14:21:13.797165 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 24322, seq 3, length 64
14:21:16.811084 arp who-has 11.0.0.101 tell 11.0.0.102
14:21:16.811521 arp reply 11.0.0.101 is-at 4:a6:60:e9:bd:90 (oui Unknown)

```



```

s2:# tcpdump -i eth0 -s0
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
14:21:11.806934 arp who-has 11.0.0.102 tell 11.0.0.101
14:21:11.818522 arp reply 11.0.0.102 is-at 42:0d:f3:e1:4b:4b (oui Unknown)
14:21:11.822840 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 24322, seq 1, length 64
14:21:11.826124 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 24322, seq 1, length 64
14:21:12.794505 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 24322, seq 2, length 64
14:21:12.798854 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 24322, seq 2, length 64
14:21:13.793551 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 24322, seq 3, length 64
14:21:13.796907 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 24322, seq 3, length 64
14:21:16.805620 arp who-has 11.0.0.101 tell 11.0.0.102
14:21:16.811743 arp reply 11.0.0.101 is-at 4:a6:60:e9:bd:90 (oui Unknown)

```



```

s3:# tcpdump -i eth0 -s0
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
14:21:11.809163 arp who-has 11.0.0.102 tell 11.0.0.101
14:21:11.825253 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 24322, seq 1, length 64
14:21:11.826081 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 24322, seq 1, length 64
14:21:12.797061 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 24322, seq 2, length 64
14:21:12.798514 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 24322, seq 2, length 64
14:21:13.795364 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 24322, seq 3, length 64
14:21:13.796640 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 24322, seq 3, length 64
14:21:16.805516 arp who-has 11.0.0.101 tell 11.0.0.102
14:21:16.811843 arp reply 11.0.0.101 is-at 4:a6:60:e9:bd:90 (oui Unknown)

```

Como vemos en la imagen superior todos reciben las tamañas mandadas por pc1. En el caso de r1 vemos que le llega la solicitud pero la descarta

5.1 Configuración de Vlan100

Primero borramos la configuración de los switch:

```

ifconfig s1 down
brctl del s1
ifconfig s2 down
brctl del s2
ifconfig s3 down
brctl del s3

```

S1

```

vconfig add eth3 100
ifconfig eth3.100 up
brctl addbr vs100
brctl addif vs100 eth0
brctl addif vs100 eth3.100
ifconfig vs100 up

```

S2

```
vconfig add eth0 100
vconfig add eth1 100
ifconfig eth0.100 up
ifconfig eth1.100 up
brctl addbr vs100
brctl addif vs100
eth0.100
brctl addif vs100
eth1.100
brctl addif vs100 eth2
ifconfig vs100 up
```

S3

```
vconfig add eth0 100
ifconfig eth0.100 up
brctl addbr vs100
brctl addif vs100 eth0.100
brctl addif vs100 eth1
ifconfig vs100 up
```

5.1.1

Una vez configurada la VLAN 100 solo se comunicaran máquinas que pertenezcan a ésta, en este caso se comunicaran pc1 con pc2 (aunque también está incluido r1 en esta VLAN).

5.1.2

Se puede encontrar en s1 s2 s3 y r1 que son los dispositivos pertenecientes a esta VLAN. Comprobamos haciendo una captura de tráfico en estos dispositivos.

```

s1:# tcpdump -i eth0 -s0
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
16:20:07.250454 arp who-has 11.0.0.102 tell 11.0.0.101
16:20:07.260312 arp reply 11.0.0.102 is-at 42:0d:f3:e1:4b:4b (oui Unknown)
16:20:07.261674 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 44547, seq 1, length 64
16:20:07.271650 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 44547, seq 1, length 64
16:20:08.240931 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 44547, seq 2, length 64
16:20:08.243571 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 44547, seq 2, length 64
16:20:09.240015 IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 44547, seq 3, length 64
16:20:09.243933 IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 44547, seq 3, length 64
16:20:12.271201 arp who-has 11.0.0.101 tell 11.0.0.102
16:20:12.272042 arp reply 11.0.0.101 is-at 4:a6:60:e9:bd:90 (oui Unknown)
[]

s2:# tcpdump -i eth0 -s0
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
16:20:07.251657 wlan 100, p 0, arp who-has 11.0.0.102 tell 11.0.0.101
16:20:07.257476 wlan 100, p 0, arp reply 11.0.0.102 is-at 42:0d:f3:e1:4b:4b (oui Unknown)
16:20:07.264971 wlan 100, p 0, IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 44547, seq 1, length 64
16:20:07.270930 wlan 100, p 0, IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 44547, seq 1, length 64
16:20:08.241620 wlan 100, p 0, IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 44547, seq 2, length 64
16:20:08.245437 wlan 100, p 0, IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 44547, seq 2, length 64
16:20:09.240147 wlan 100, p 0, IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 44547, seq 3, length 64
16:20:09.245520 wlan 100, p 0, arp reply 11.0.0.102 is-at 42:0d:f3:e1:4b:4b (oui Unknown)
[]

r1:# tcpdump -i eth0 -s0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
16:20:07.252810 wlan 100, p 0, arp who-has 11.0.0.101 tell 11.0.0.101
16:20:07.253799 arp who-has pc2 tell pc1
[]

s3:# ping -c 3 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=2.58 ms
64 bytes from 11.0.0.102: icmp_seq=2 ttl=64 time=1.21 ms
64 bytes from 11.0.0.102: icmp_seq=3 ttl=64 time=7.34 ms

--- 11.0.0.102 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 1.213/3.712/7.343/2.627 ms
pc1:#

pc1:# ping -c 3 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=16.7 ms
64 bytes from 11.0.0.102: icmp_seq=2 ttl=64 time=1.75 ms
64 bytes from 11.0.0.102: icmp_seq=3 ttl=64 time=3.57 ms

--- 11.0.0.102 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.794/7.370/16.783/6.697 ms
pc1:# []
p

r1:# ping -c 3 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=64 time=16.7 ms
64 bytes from 11.0.0.101: icmp_seq=2 ttl=64 time=1.75 ms
64 bytes from 11.0.0.101: icmp_seq=3 ttl=64 time=3.57 ms

--- 11.0.0.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.794/7.370/16.783/6.697 ms
pc1:# []

```

5.1.3

Para comprobar que ocurre cuando se hace ping de pc1 a pc2 hemos realizado una captura en s2 y en pc2. Podemos observar como a s2 le llegan las tramas con la etiqueta de vlan100 dirigidas a pc2.

```

pc1:# ping -c 3 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=16.7 ms
64 bytes from 11.0.0.102: icmp_seq=2 ttl=64 time=1.75 ms
64 bytes from 11.0.0.102: icmp_seq=3 ttl=64 time=3.57 ms

--- 11.0.0.102 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 1.213/3.712/7.343/2.627 ms
pc1:#

pc1:# ping -c 3 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=16.7 ms
64 bytes from 11.0.0.102: icmp_seq=2 ttl=64 time=1.75 ms
64 bytes from 11.0.0.102: icmp_seq=3 ttl=64 time=3.57 ms

--- 11.0.0.102 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.794/7.370/16.783/6.697 ms
pc1:# []
p

s2:# tcpdump -i eth0 -s0
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
16:42:48.217873 arp who-has pc2 tell pc1
16:42:48.223054 arp reply pc2 is-at 42:0d:f3:e1:4b:4b (oui Unknown)
16:42:48.223054 arp reply pc2 is-at 42:0d:f3:e1:4b:4b (oui Unknown)
16:42:48.222886 IP pc1 > pc2: ICMP echo request, id 13316, seq 1, length 64
16:42:48.222914 IP pc2 > pc1: ICMP echo reply, id 13316, seq 1, length 64
16:42:49.212023 IP pc1 > pc2: ICMP echo request, id 13316, seq 2, length 64
16:42:49.212063 IP pc2 > pc1: ICMP echo reply, id 13316, seq 2, length 64
16:42:50.211682 IP pc1 > pc2: ICMP echo request, id 13316, seq 3, length 64
16:42:50.211719 IP pc2 > pc1: ICMP echo reply, id 13316, seq 3, length 64
16:42:53.216492 arp who-has pc1 tell pc2
16:42:53.223009 arp reply pc1 is-at 4:a6:60:e9:bd:90 (oui Unknown)
[]

s2:# tcpdump -i eth0 -s0
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
16:42:48.221363 wlan 100, p 0, IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 13316, seq 1, length 64
16:42:48.223684 wlan 100, p 0, IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 13316, seq 1, length 64
16:42:49.211191 wlan 100, p 0, IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 13316, seq 2, length 64
16:42:49.212264 wlan 100, p 0, IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 13316, seq 2, length 64
16:42:50.211446 wlan 100, p 0, IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id 13316, seq 3, length 64
16:42:50.213800 wlan 100, p 0, IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 13316, seq 3, length 64
16:42:53.216814 wlan 100, p 0, arp who-has 11.0.0.101 tell 11.0.0.102
16:42:53.219828 wlan 100, p 0, arp reply 11.0.0.101 is-at 4:a6:60:e9:bd:90 (oui Unknown)
[]
```

También podemos hacer una captura o bien en s1 como a s3 por ser frontera y así poder ver qué ocurre con estas etiquetas.

```

s3:~# tcpdump -i eth0 -s0
tcpdump: WARNING: eth0: no IPv4 address assigned
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
16:48:27.552031 vlan 100, p 0, IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id
22276, seq 1, length 64
16:48:27.553671 vlan 100, p 0, IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 2
22276, seq 1, length 64
16:48:28.551261 vlan 100, p 0, IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id
22276, seq 2, length 64
16:48:28.554111 vlan 100, p 0, IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 2
22276, seq 2, length 64
16:48:29.550681 vlan 100, p 0, IP 11.0.0.101 > 11.0.0.102: ICMP echo request, id
22276, seq 3, length 64
16:48:29.551951 vlan 100, p 0, IP 11.0.0.102 > 11.0.0.101: ICMP echo reply, id 2
22276, seq 3, length 64
16:48:32.548081 vlan 100, p 0, arp who-has 11.0.0.101 tell 11.0.0.102
16:48:32.551551 vlan 100, p 0, arp reply 11.0.0.101 is-at 4:a:a6:e9:bd:90 (oui
Unknown)

```

Como vemos este switch al ser frontera de pc2 por un lado le llega la trama con la etiqueta VLAN 100, a continuación este switch se encarga de quitar la etiqueta de VLAN y entrega las tramas a pc2. Por tanto pc2 recibe las tramas sin la etiqueta de VLAN. La respuesta de pc2 se manda a través de s3 quien se encargara de volver a poner la etiqueta de la VLAN.

Este mismo proceso ocurre con el switch s1, cuando le llegan tramas a pc1 s1 le quita las etiquetas y si pc1 envía tramas pone las etiquetas.

5.1.4

Cabe esperar que las etiquetas de VLAN 100 aparezcan en la captura de S1 y S3

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	4:a:a6:60:e9:bd:90	Broadcast	ARP	46	Who has 11.0.0.102? Tell 11.0.0.101
2	0.000741	42:0d:f3:e1:4b:4b	4:a:a6:60:e9:bd:90	ARP	46	11.0.0.102 is at 42:0d:f3:e1:4b:4b
3	0.001005	11.0.0.101	11.0.0.102	ICMP	102	Echo (ping) request id=0xc804, seq=1/256, ttl=64 (reply in 4)
4	0.001570	11.0.0.102	11.0.0.101	ICMP	102	Echo (ping) reply id=0xc804, seq=1/256, ttl=64 (request in 3)
5	0.993919	11.0.0.101	11.0.0.102	ICMP	102	Echo (ping) request id=0xc804, seq=2/512, ttl=64 (reply in 6)
6	0.994729	11.0.0.102	11.0.0.101	ICMP	102	Echo (ping) reply id=0xc804, seq=2/512, ttl=64 (request in 5)
7	1.992937	11.0.0.101	11.0.0.102	ICMP	102	Echo (ping) request id=0xc804, seq=3/768, ttl=64 (reply in 8)
8	1.994208	11.0.0.102	11.0.0.101	ICMP	102	Echo (ping) reply id=0xc804, seq=3/768, ttl=64 (request in 7)
9	4.992118	42:0d:f3:e1:4b:4b	4:a:a6:60:e9:bd:90	ARP	46	Who has 11.0.0.101? Tell 11.0.0.102
10	4.992321	4:a:a6:60:e9:bd:90	42:0d:f3:e1:4b:4b	ARP	46	11.0.0.101 is at 4:a:a6:60:e9:bd:90

> Frame 1: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)
> Ethernet II, Src: 4:a:a6:60:e9:bd:90 (4:a:a6:60:e9:bd:90), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
✓ 802.Q Virtual LAN, PRI: 0, DEI: 0, ID: 100
 00. = Priority: Best Effort (default) (0)
 ..0 = DEI: Ineligible
 0110 0100 = ID: 100
 Type: ARP (0x0806)
> Address Resolution Protocol (request)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	4:a6:60:e9:bd:90	Broadcast	ARP	46	Who has 11.0.0.102? Tell 11.0.0.101
2	0.000213	42:0d:f3:e1:4b:4b	4:a6:60:e9:bd:90	ARP	46	11.0.0.102 is at 42:0d:f3:e1:4b:4b
3	0.000738	11.0.0.101	11.0.0.102	ICMP	102	Echo (ping) request id=0xc804, seq=1/256, ttl=64 (reply in 4)
4	0.001027	11.0.0.102	11.0.0.101	ICMP	102	Echo (ping) reply id=0xc804, seq=1/256, ttl=64 (request in 3)
5	0.993716	11.0.0.101	11.0.0.102	ICMP	102	Echo (ping) request id=0xc804, seq=2/512, ttl=64 (reply in 6)
6	0.994183	11.0.0.102	11.0.0.101	ICMP	102	Echo (ping) reply id=0xc804, seq=2/512, ttl=64 (request in 5)
7	1.992794	11.0.0.101	11.0.0.102	ICMP	102	Echo (ping) request id=0xc804, seq=3/768, ttl=64 (reply in 8)
8	1.993427	11.0.0.102	11.0.0.101	ICMP	102	Echo (ping) reply id=0xc804, seq=3/768, ttl=64 (request in 7)
9	4.991528	42:0d:f3:e1:4b:4b	4:a6:60:e9:bd:90	ARP	46	Who has 11.0.0.101? Tell 11.0.0.102
10	4.992512	4:a6:60:e9:bd:90	42:0d:f3:e1:4b:4b	ARP	46	11.0.0.101 is at 4:a6:60:e9:bd:90

```
> Frame 2: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)
> Ethernet II, Src: 42:0d:f3:e1:4b:4b (42:0d:f3:e1:4b:4b), Dst: 4:a6:60:e9:bd:90 (4:a6:60:e9:bd:90)
  ▼ 802.1Q Virtual LAN, PRI: 0, DEI: 0 ID: 100
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    ... 0000 0110 0100 = ID: 100
    Type: ARP (0x0806)
  > Address Resolution Protocol (reply)
```

5.1.5

El switch que introduce la etiqueta a la ida es S1, y la respuesta al ping s3 es el encargado de poner la etiqueta

5.1.6

la etiqueta se elimina en s3 de la solicitud y en s1 se quita la etiqueta a la respuesta

5.1.7

No tienen forma de saber que están usando VLANS para comunicarse porque son los switches los encargados de gestionar el tráfico.

5.1.8

Pues como aparece en la captura de pantalla el ping no llega a pc4 a pesar de estar en la misma subred porque no pertenece a VLAN 100

pc1:~# ping -c 3 11.0.0.104
PING 11.0.0.104 (11.0.0.104) 56(84) bytes of data.
From 11.0.0.101 icmp_seq=1 Destination Host Unreachable
From 11.0.0.101 icmp_seq=2 Destination Host Unreachable
From 11.0.0.101 icmp_seq=3 Destination Host Unreachable

--- 11.0.0.104 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2010ms
, pipe 3
pc1:~#

5.2 Configuracion Vlan 200

S1

```
vconfig add eth3 200  
ifconfig eth3.200 up  
brctl addbr vs200  
brctl addif vs200 eth3.200  
brctl addif vs200 eth2  
ifconfig vs200
```

S2

```
vconfig add eth0 200  
ifconfig eth0.200 up  
brctl addbr vs200  
brctl addif vs200 eth0.200  
brctl addif vs200 eth3  
ifconfig vs200 up
```

5.2.1

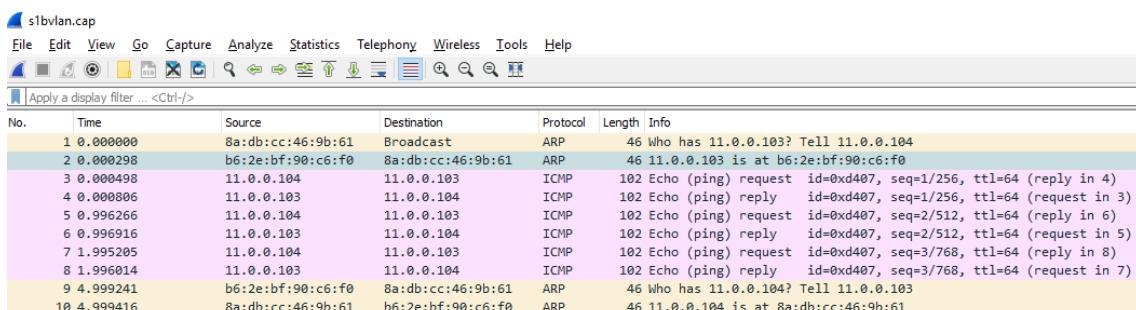
En este caso las máquinas que se pueden comunicar en esta VLAN son pc4 y pc3 a través de s1 y s2.

5.2.3

En este caso al hacer ping de pc4 a pc3 podemos ver que la solicitud de ARP pasará por los switch's s1 y s2. Y la respuesta de ICMP también pasara por pc1 y pc2.

5.2.4

En este caso encontramos las etiquetas en S1 como vemos en la imagen inferior



The Wireshark capture shows the following sequence of frames:

- Frame 1: ARP request from PC4 (11.0.0.104) to Broadcast (11.0.0.103?) asking for MAC address of 11.0.0.103.
- Frame 2: ARP reply from PC3 (11.0.0.103) to PC4 (11.0.0.104).
- Frame 3: ICMP echo request from PC4 (11.0.0.104) to PC3 (11.0.0.103).
- Frame 4: ICMP echo reply from PC3 (11.0.0.103) to PC4 (11.0.0.104).
- Frame 5: ICMP echo request from PC4 (11.0.0.104) to PC3 (11.0.0.103).
- Frame 6: ICMP echo reply from PC3 (11.0.0.103) to PC4 (11.0.0.104).
- Frame 7: ICMP echo request from PC4 (11.0.0.104) to PC3 (11.0.0.103).
- Frame 8: ICMP echo reply from PC3 (11.0.0.103) to PC4 (11.0.0.104).
- Frame 9: ARP request from PC4 (11.0.0.104) to Broadcast (11.0.0.103?) asking for MAC address of 11.0.0.103.
- Frame 10: ARP reply from PC3 (11.0.0.103) to PC4 (11.0.0.104).

Details of the selected frame (Frame 2):

- Frame 2: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)
- Ethernet II, Src: b6:2e:bf:90:c6:f0 (b6:2e:bf:90:c6:f0), Dst: 8a:db:cc:46:9b:61 (8a:db:cc:46:9b:61)
- 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 200 Identificador VLAN 200 S1
- 000. = Priority: Best Effort (default) (0)
-0 = DEI: Ineligible
- 0000 1100 1000 = ID: 200
- Type: ARP (0x0806)
- Address Resolution Protocol (reply)

5.2.5

Como ocurría en el caso anterior tampoco llega el ping de pc1 a pc4 pero en este caso se debe a que ambas máquinas pertenecen a VLANS distintas.

5.2.6

En este caso lo único que podemos ver es la configuración del switch virtual ya que hemos asignado sus direcciones Ethernet a cada una de las VLANS (VLAN 100, VLAN 200)

5.3 Configuracion de vlan 300

S1

```
vconfig add eth3 300
ifconfig eth3.300 up
brctl addbr vs300
brctl addif vs300 eth1
brctl addif vs300 eth3.300
ifconfig vs300 up
```

S2

```
vconfig add eth0 300
vconfig add eth1 300
ifconfig eth0.300 up
ifconfig eth1.300 up
brctl addbr vs300
brctl addif vs300 eth0.300
brctl addif vs300 eth1.300
ifconfig vs300 up
```

S3

```
vconfig add eth0 300
ifconfig eth0.300 up
brctl addbr vs300
brctl addif vs300 eth0.300
brctl addif vs300 eth2
brctl addif vs300 eth3
ifconfig vs300 up
```

5.3.1

Tal y como está configurada VLAN 300 pc5 tiene conexión con pc6 y con el router r1

5.3.2

Como podemos ver al hacer el ping de pc6 a pc1 tenemos conexión esto se debe a que el router es el que está estableciendo esa conexión.

Más en concreto por una razón. Los paquetes que manda pc6 llegan al router, estos paquetes ya no tienen la etiqueta de la VLAN por lo que el router interpreta que van dirigidos a la dirección 11.0.0.101 consulta su tabla y ve que tiene ruta, lo manda por su eth0 a s2. S2 etiquetara los paquetes que le llegan por su eth2 con la VLAN 100 de ahí pasa a s1 que lo manda a pc1. Con la respuesta ocurre lo mismo.

5.3.3

En este caso no hay ningún tipo de problema el ping se realiza correctamente y únicamente viaja hacia pc5

5.3.4

Comprobamos con arp -a que la cache de arp esta vacía en pc6 y procedemos a hacer el ping. Las tramas arp aparecerán

5.3.5

Aparecen las etiquetas de VLAN en el dispositivo s2, como se puede ver en la siguiente captura de pantalla:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	8a:79:4e:55:ed:5d	Broadcast	ARP	46	Who has 12.0.0.1? Tell 12.0.0.106
2	0.011026	b2:52:4f:70:e2:c9	Broadcast	ARP	46	Who has 11.0.0.101? Tell 11.0.0.1
3	0.011632	4a:a6:60:e9:bd:90	b2:52:4f:70:e2:c9	ARP	46	11.0.0.101 is at 4a:a6:60:e9:bd:90
4	0.011868	12.0.0.106	11.0.0.101	ICMP	102	Echo (ping) request id=0x390a, seq=1/256, ttl=63 (reply in 5)
5	0.012175	11.0.0.101	12.0.0.106	ICMP	102	Echo (ping) reply id=0x390a, seq=1/256, ttl=64 (request in 4)
6	0.990855	12.0.0.106	11.0.0.101	ICMP	102	Echo (ping) request id=0x390a, seq=2/512, ttl=63 (reply in 7)
7	0.991806	11.0.0.101	12.0.0.106	ICMP	102	Echo (ping) reply id=0x390a, seq=2/512, ttl=64 (request in 6)
8	1.991303	12.0.0.106	11.0.0.101	ICMP	102	Echo (ping) request id=0x390a, seq=3/768, ttl=63 (reply in 9)
9	1.991906	11.0.0.101	12.0.0.106	ICMP	102	Echo (ping) reply id=0x390a, seq=3/768, ttl=64 (request in 8)
10	5.008022	4a:a6:60:e9:bd:90	b2:52:4f:70:e2:c9	ARP	46	Who has 11.0.0.1? Tell 11.0.0.101
11	5.008685	b2:52:4f:70:e2:c9	4a:a6:60:e9:bd:90	ARP	46	11.0.0.1 is at b2:52:4f:70:e2:c9


```
> Frame 1: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)
> Ethernet II, Src: 8a:79:4e:55:ed:5d (8a:79:4e:55:ed:5d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 300
  000. .... .... .... = Priority: Best Effort (default) (0)
  ...0. .... .... .... = DEI: Ineligible
  .... 0001 0010 1100 = ID: 300
  Type: ARP (0x0806)
> Address Resolution Protocol (request)
```

Práctica también en :

<https://github.com/imorenoma/practica1-st>