

## 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.

Como el switch s1 esta apagado los pcs que se encuentras fuera de la red donde se hace el ping no recibirán nada, mientras que la captura de pc2 que si está en la red de red de p1 capturará de un mensaje de arp preguntando la dirección y el arp respuesta sobre la dirección y los 6 mensajes del ping (3 de ida y 3 de vuelta), más tarde pc2 envía solicitud de arp (no va dirigida al broadcast, va directamente a pc1 porque pc2 ya sabe el origen de pc1 porque le ha preguntado antes)(estos mensajes ocurren en los 5 segundos después del ping) sobre pc2 y pc1 responde. (pc5 y pc3, 0 paquetes capturados)

2-

```
pc2
#####
--- Netkit phase 2 initialization terminated ---

pc2 login: root (automatic login)
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
12:28:18.731236 arp who-has pc2 tell pc1
12:28:18.732430 arp reply pc2 is-at 00:07:e9:00:00:02 (oui Unknown)
12:28:18.732500 IP pc1 > pc2: ICMP echo request, id 23554, seq 1, length 64
12:28:18.732625 IP pc2 > pc1: ICMP echo reply, id 23554, seq 1, length 64
12:28:19.720918 IP pc1 > pc2: ICMP echo request, id 23554, seq 2, length 64
12:28:19.720959 IP pc2 > pc1: ICMP echo reply, id 23554, seq 2, length 64
12:28:20.721751 IP pc1 > pc2: ICMP echo request, id 23554, seq 3, length 64
12:28:20.721797 IP pc2 > pc1: ICMP echo reply, id 23554, seq 3, length 64
12:28:23.719957 arp who-has pc1 tell pc2
12:28:23.720407 arp reply pc1 is-at 00:07:e9:00:00:01 (oui Unknown)

10 packets captured
10 packets received by filter
0 packets dropped by kernel
pc2:~#
```

```
pc3
pc3 login: root (automatic login)
pc3:~# tcpdump -i eth
tcpdump: SIOCGIFHWADDR: No such device
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

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

```
pc5
pc5 login: root (automatic login)
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

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

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

```
pc1:~# ping -c 1 11.29.0.4
PING 11.29.0.4 (11.29.0.4) 56(84) bytes of data.
From 11.29.0.1 icmp_seq=1 Destination Host Unreachable
--- 11.29.0.4 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
pc1:~#
```

El ping no consigue llegar debido a que s1 está off.

## 1.2. Comunicación entre máquinas con s1 arrancado

**2. Piensa en qué paquetes se capturarán ahora en pc2, pc3 y en pc5 repitiendo el mismo ping.**

Captura realizada en pc1.

Recibe el arp de pregunta realizado por el pc1 que llega a él a través del hub 1. Ya que la solicitud de arp va al broadcast y los reciben todos los pcs. Y el pc2 realiza su respuesta. Y luego los 6 paquetes del ping (3 ida y 3 vuelta).

```
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
12:57:02.687645 IP pc1 > pc2: ICMP echo request, id 53762, seq 1, length 64
12:57:02.698760 arp who-has pc1 tell pc2
12:57:02.698830 arp reply pc1 is-at 00:07:e9:00:00:01 (oui Unknown)
12:57:02.698834 IP pc2 > pc1: ICMP echo reply, id 53762, seq 1, length 64
12:57:03.686823 IP pc1 > pc2: ICMP echo request, id 53762, seq 2, length 64
12:57:03.686866 IP pc2 > pc1: ICMP echo reply, id 53762, seq 2, length 64
12:57:04.685702 IP pc1 > pc2: ICMP echo request, id 53762, seq 3, length 64
12:57:04.685715 IP pc2 > pc1: ICMP echo reply, id 53762, seq 3, length 64

8 packets captured
8 packets received by filter
0 packets dropped by kernel
pc2:~#
```

Captura realizada en pc3.

Recibe el arp de pregunta realizado por el pc1 que llega a él a través del hub 1, s1, hub 2. Ya que la solicitud de arp va al broadcast y los reciben todos los pcs. Y el pc3 realiza su respuesta.

```
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
12:57:02.687714 IP pc1 > pc2: ICMP echo request, id 53762, seq 1, length 64
12:57:02.698865 arp who-has pc1 tell pc2

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

Captura realizada en pc5.

Recibe el arp de pregunta realizado por el pc1 que llega a él a través del hub 1, s1, hub 3. Ya que la solicitud de arp va al broadcast y los reciben todos los pcs. Y el pc5 realiza su respuesta.

```

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
12:57:02.687703 IP pc1 > pc2: ICMP echo request, id 53762, seq 1, length 64
12:57:02.698865 arp who-has pc1 tell pc2

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

```

### Apartado 3 y 4:

Con el comando arp –a miramos si la cache de pc1 está vacía, en este caso si así que no tenemos que borrar nada.

Captura en pc2

Pc2 capturará de un mensaje de arp preguntado la dirección y el arp respuesta sobre la dirección y los 6 mensajes del ping (3 de ida y 3 de vuelta), más tarde pc2 envía solicitud de arp (no va dirigida al broadcast, va directamente a pc1 porque pc2 ya sabe el origen de pc1 porque le ha preguntado antes) (estos mensajes ocurren en los 5 segundos después del ping) sobre pc2 y pc1 responde. (pc5 y pc3, 0 paquetes capturados).

No.	Time	Source	Destination	Protocol	Length	Info
1 0.000000		Intel_00:00:01	Broadcast	ARP	42	who has 11.29.0.2? Tell 11.29.0.1
2 0.000048		Intel_00:00:02	Intel_00:00:01	ARP	42	11.29.0.2 is at 00:07:e9:00:00:02
3 0.000030		11.29.0.1	11.29.0.2	ICMP	98	Echo (ping) request id=0xd402, seq=1/256, ttl=64 (reply in 4)
4 0.000041		11.29.0.2	11.29.0.1	ICMP	98	Echo (ping) reply id=0xd402, seq=1/256, ttl=64 (request in 3)
5 0.990298		11.29.0.1	11.29.0.2	ICMP	98	Echo (ping) request id=0xd402, seq=2/512, ttl=64 (reply in 6)
6 0.990344		11.29.0.2	11.29.0.1	ICMP	98	Echo (ping) reply id=0xd402, seq=2/512, ttl=64 (request in 5)
7 1.990648		11.29.0.1	11.29.0.2	ICMP	98	Echo (ping) request id=0xd402, seq=3/768, ttl=64 (reply in 8)
8 1.990694		11.29.0.2	11.29.0.1	ICMP	98	Echo (ping) reply id=0xd402, seq=3/768, ttl=64 (request in 7)
9 4.994186		Intel_00:00:02	Intel_00:00:01	ARP	42	Who has 11.29.0.1? Tell 11.29.0.2
10 4.994652		Intel_00:00:01	Intel_00:00:02	ARP	42	11.29.0.1 is at 00:07:e9:00:00:01

Captura en pc3.

No.	Time	Source	Destination	Protocol	Length	Info
1 0.000000		Intel_00:00:01	Broadcast	ARP	42	Who has 11.29.0.2? Tell 11.29.0.1

Captura en pc5.

No.	Time	Source	Destination	Protocol	Length	Info
1 0.000000		Intel_00:00:01	Broadcast	ARP	42	Who has 11.29.0.2? Tell 11.29.0.1

### 5. Responde a estas preguntas:

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

Porque la dirección de envío del arp es broadcast.

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

Porque solo llega al pc que ha preguntado.

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

Porque el ping va dirigido al pc2 y los demás pcs no intervienen con los mensajes ICMP, solo podrían intervenir con arp pero no es necesario.

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

Porque el ping va dirigido al pc2 y los demás pcs no intervienen con los mensajes ICMP, solo podrían intervenir con arp pero no es necesario.

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

Pc1

00:07:e9:00:00:01

Pc2

00:07:e9:00:00:02

Pc3

00:07:e9:00:00:03

Pc4

00:07:e9:00:00:04

Pc5

00:07:e9:00:00:05

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

port	no	mac	addr	is local?	ageing	timer
1	00:07:e9:00:00:01		00:07:e9:00:00:01	no	72.94	
1	00:07:e9:00:00:02		00:07:e9:00:00:02	no	72.94	
2	00:07:e9:00:00:03		00:07:e9:00:00:03	no	156.99	
2	00:07:e9:00:00:04		00:07:e9:00:00:04	no	157.90	
3	00:07:e9:00:00:05		00:07:e9:00:00:05	no	155.55	
3	00:07:e9:00:00:06		00:07:e9:00:00:06	no	154.98	
1	00:07:e9:00:ff:f0		00:07:e9:00:ff:f0	yes	0.00	
2	00:07:e9:00:ff:f1		00:07:e9:00:ff:f1	yes	0.00	
3	00:07:e9:00:ff:f2		00:07:e9:00:ff:f2	yes	0.00	

La tabla de direcciones aprendidas tienes 3 entradas locales, correspondientes con las entradas de los hub. Y las direcciones de los pcs porque al enviar el arp request necesita saber las direcciones.

#### **8. Lanza tcpdump en pc2, en pc3 y en pc5. Y ejecuta en pc1 el ping a pc6: pc1.**

Captura en pc2

hub-switch-04.cap							
No.	Time	Source	Destination	Protocol	Length	Info	
1	0.000000	Intel_00:00:01	Broadcast	ARP	42	Who has 11.29.0.6? Tell 11.29.0.1	
2	0.000570	Intel_00:00:06	Intel_00:00:01	ARP	42	11.29.0.6 is at 00:07:e9:00:00:06	
3	0.000757	11.29.0.1	11.29.0.6	ICMP	98	Echo (ping) request id=0x7602, seq=1/256	
4	0.001292	11.29.0.6	11.29.0.1	ICMP	98	Echo (ping) reply id=0x7602, seq=1/256	
5	0.988677	11.29.0.1	11.29.0.6	ICMP	98	Echo (ping) request id=0x7602, seq=2/512	
6	0.989253	11.29.0.6	11.29.0.1	ICMP	98	Echo (ping) reply id=0x7602, seq=2/512	
7	1.994330	11.29.0.1	11.29.0.6	ICMP	98	Echo (ping) request id=0x7602, seq=3/768	
8	1.995065	11.29.0.6	11.29.0.1	ICMP	98	Echo (ping) reply id=0x7602, seq=3/768	
9	4.999335	Intel_00:00:06	Intel_00:00:01	ARP	42	Who has 11.29.0.1? Tell 11.29.0.6	
10	4.999431	Intel_00:00:01	Intel_00:00:06	ARP	42	11.29.0.1 is at 00:07:e9:00:00:01	

Captura en pc3

hub-switch-05.cap							
No.	Time	Source	Destination	Protocol	Length	Info	
1	0.000000	Intel_00:00:01	Broadcast	ARP	42	Who has 11.29.0.6? Tell 11.29.0.1	

Captura en pc5

hub-switch-06.cap							
No.	Time	Source	Destination	Protocol	Length	Info	
1	0.000000	Intel_00:00:01	Broadcast	ARP	42	Who has 11.29.0.6? Tell 11.29.0.1	
2	0.000229	Intel_00:00:06	Intel_00:00:01	ARP	42	11.29.0.6 is at 00:07:e9:00:00:06	
3	0.000489	11.29.0.1	11.29.0.6	ICMP	98	Echo (ping) request id=0x7602, seq=1/256, ttl=64 (reply in 4)	
4	0.000717	11.29.0.6	11.29.0.1	ICMP	98	Echo (ping) reply id=0x7602, seq=1/256, ttl=64 (request in 3)	
5	0.988520	11.29.0.1	11.29.0.6	ICMP	98	Echo (ping) request id=0x7602, seq=2/512, ttl=64 (reply in 6)	
6	0.988663	11.29.0.6	11.29.0.1	ICMP	98	Echo (ping) reply id=0x7602, seq=2/512, ttl=64 (request in 5)	
7	1.994202	11.29.0.1	11.29.0.6	ICMP	98	Echo (ping) request id=0x7602, seq=3/768, ttl=64 (reply in 8)	
8	1.994343	11.29.0.6	11.29.0.1	ICMP	98	Echo (ping) reply id=0x7602, seq=3/768, ttl=64 (request in 7)	
9	4.998637	Intel_00:00:06	Intel_00:00:01	ARP	42	Who has 11.29.0.1? Tell 11.29.0.6	
10	4.999229	Intel_00:00:01	Intel_00:00:06	ARP	42	11.29.0.1 is at 00:07:e9:00:00:01	

Si ha ocurrido lo que pensaba. Ya que todos reciben el arp pregunta porque va a la dirección broadcast. El pc2 recibe todos los mensajes ya que desde pc1 van al hub y este envía todo, al igual que pc5. Una vez que se realiza el arp request , se emite la arp reply de pc6 y el s1 ya sabe dónde está, una vez que todos los mensajes llegan a s1, van al hub 3 y por eso pc5 también recibe todos los mensajes. Como s1 sabía dónde estaba pc6 no envía los mensajes por el hub2 y por eso pc2 solo recibe el arp request.

## 9. Responde a estas preguntas:

**¿Por qué NO llegan a pc3 y a pc4 los mensajes ICMP echo request e ICMP echo reply?**

Porque una vez que se emite el arp reply, s1 ya sabe dónde se sitúa pc6 y no tiene que enviarlo por todos lados. Es decir, va al hub3 y no al hub2.

**¿Por qué SÍ llegan a pc2 y a pc5 todos los mensajes enviados por pc1 y pc6?**

Cuando pc1 envía los mensajes, el hub1 lo envía a todo lo que está en su red, por eso llegan a pc2. Y lo mismo ocurre con pc5, el s1 envía el mensaje hacia pc6, pero hay un hub al que llega primero, y este envía a todos lados menos por donde ha llegado, en este caso pc5.

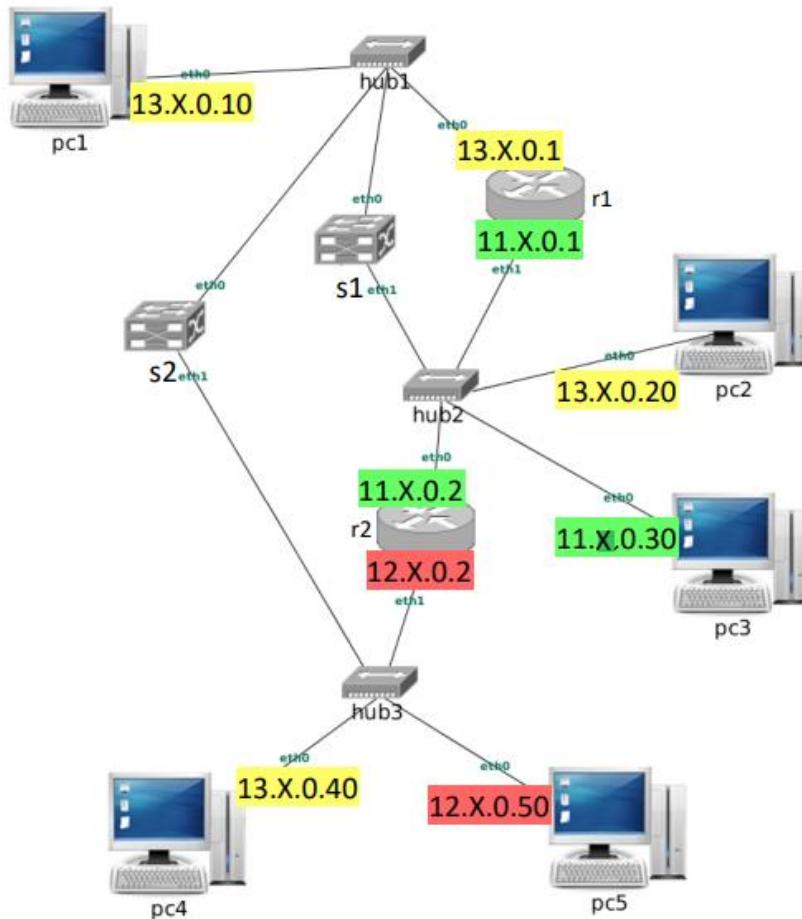
**¿Crees qué si cambiamos todos los hubs de la figura por switches, los mensajes capturados anteriormente serían los mismos?**

No serían los mismos. La captura realizada en pc3 si sería igual. Y las realizadas en pc2 y pc5 serían igual que la pc3. Es decir, en las capturas solo habría el mensaje arp request.

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

Como todas las maquinas están conectadas por s1, es posible hacer ping entre todas.

## 2. Redes conectadas a través de switch y router



### Aparatado 1:

Para realizar un ping de pc2-pc4. Recorrerá pc2-hub2-r1-hub1-s2-pc4.

Ya que estos dispositivos tienen la misma red que pc2-pc4.

### Aparatado 2: (Revisar)

Primero necesitaremos un arp request de pc2 que recibirán todos y su correspondiente arp reply que sale de pc4. Una vez se envíe todos los mensajes se realizará otro arp reply de pc4 dirección pc2, y el arp reply de este.

Solicitudes:

- Solicitud de pc2-pc4 la capturas en cualquier interfaz, pc o router.
- Solicitud de pc4-pc2 capturas en cualquier interfaz, pc o router.

Respuestas: Podrás capturar en las interfaces que la ruta tenga configurada para la vuelta.

**3. Para ver todo el tráfico generado deberás lanzar un tcpdump por cada hub de la figura. Justifica la respuesta.**

La ruta que sigue el ping de pc2-pc4 pasa por los dos hubs, por lo que si lanzas una captura en estas interfaces obtendrás toda la información. Ya que el hub envía todo lo que recibe por sus interfaces y además en un hub no se puede realizar un tcpdump.

**Aparatado 4 y 5:**

**Direcciones aprendidas s1**

Aprende las direcciones de pc2 y pc4

**Direcciones aprendidas s2**

Aprende las de pc2-pc4 y todos sus vecinos (pc1,pc5)

**6. ¿Crees que habrá llegado alguno de los mensajes ICMP echo request a pc1, pc3 o pc5? Justifica la respuesta.**

Llegan todos los mensajes a los pc2 mencionados, debido a la ruta y sobre todo a los hubs.

## **2.2. Comunicación entre pc1 y pc3.**

Para realizar un ping de pc1 a pc3 primero debe pasar por r1 ya que, si vas mirando las configuraciones con el comando route, pc1 te manda al r1 por defecto y r1 ya tiene la red de pc3 como vecina.

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

Primero necesitaremos un arp request de pc1 que recibirán todos y su correspondiente arp reply que sale de pc3. Una vez se envíe todos los mensajes se realizará otro arp reply de pc4 dirección pc2, y el arp reply de este.

Solicitudes:

- Solicitud de pc1-pc3 la capturas en cualquier interfaz, pc o router.
- Solicitud de pc3-pc1 capturas en cualquier interfaz, pc o router.

Respuestas: Podrás capturar en las interfaces que la ruta tenga configurada para la vuelta.

**4. Indica qué direcciones Ethernet habrán aprendido s1 y s2 después de ejecutar el ping y explica qué mensajes han generado dicho aprendizaje.**

**Direcciones aprendidas s2**

Aprende las direcciones de pc1 y pc3

**Direcciones aprendidas s1**

Aprende las de pc2-pc4 y todos sus vecinos (pc1,pc5,r1,r2)

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

Sí, han llegado a los 3 pcs, debido a la ruta y que de por medio hay hubs que comparten toda la información.

## **2.3. Comunicación entre pc2 y pc5**

Para hacer un ping de pc2-pc5 y mirando las rutas de encaminamiento con el comando route, vemos que va de pc2-r1-s2-pc5. ([Saber pq sigue este recorrido](#))

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

Primero necesitaremos un arp request de pc2 que recibirán todos y su correspondiente arp reply que sale de pc5. Una vez se envíe todos los mensajes se realizará otro arp reply de pc5 dirección pc2, y el arp reply de este.

Solicitudes:

- Solicitud de pc2-pc5 la capturas en cualquier interfaz, pc o router.
- Solicitud de pc5-pc2 capturas en cualquier interfaz, pc o router.

Respuestas: Podrás capturar en las interfaces que la ruta tenga configurada para la vuelta.

**4. Indica qué direcciones Ethernet habrán aprendido s1 y s2 después de ejecutar el ping y explica qué mensajes han generado dicho aprendizaje.**

#### Direcciones aprendidas s2

Aprende las direcciones de pc2 y pc5

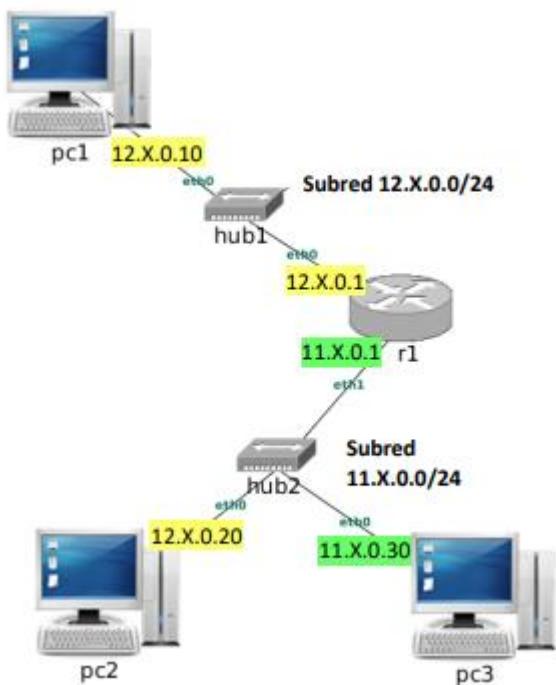
#### Direcciones aprendidas s3

Aprende las de pc2-pc5 y todos sus vecinos.

**5. ¿Crees que habrá llegado alguno de los mensajes ICMP echo request a pc1, pc3 o pc4?**

Sí, primero llega a pc3, ya que se lo envía el hub2. Y siguiendo el trayecto pasa por r1-hub1 y este se lo envía pc1, y más tarde pasa por s2-hub3 y este se lo envía a pc4 y pc5.

### 3. Proxy ARP



**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.+**

#### Camino de pc1-pc2.

1: Sirve para activar arp prox en la interfaz eth0

```
r1:~# echo 1 > /proc/sys/net/ipv4/conf/eth0/proxy_arp
```

2: Permite que la interfaz eth0 conteste sobre el pc indicado (pc2) cuando se pregunta por él.

```
r1:~# arp -i eth0 -Ds 12.29.0.20 eth0 netmask 255.255.255.255
```

**3:** Añade la ruta para llegar a pc2

**Camino de pc2-pc1.**

```
r1:~# route add -host 12.29.0.20 dev eth1
```

**4:** Sirve para activar arp prox en la interfaz eth1

```
r1:~# echo 1 > /proc/sys/net/ipv4/conf/eth1/proxy_arp
```

**5:** Permite que la interfaz eth1 conteste sobre el pc indicado (pc1) cuando se pregunta por él

```
r1:~# arp -i eth1 -Ds 11.0.0.2 eth1 netmask 255.255.255.255
```

**2. Con las cachés de ARP vacías, realiza una captura en la interfaz r1(eth0) (proxyARP-01.cap) y en pc3 (proxyARP-02.cap) y ejecuta un ping desde pc1 a la dirección IP de r1(eth0), enviando sólo 3 paquetes, y después un ping desde pc1 a pc2, enviando sólo 3 paquetes. Interrumpe la captura y explícalas solicitudes de ARP que ves en el tráfico capturado en ambos ficheros.**

**Captura en r1 (eth0).**

proxyARP-01.cap						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	c6:26:00:10:96:2b	Broadcast	ARP	42	Who has 12.29.0.1? Tell 12.29.0.10
2	0.000112	1a:f3:52:99:29:2d	c6:26:00:10:96:2b	ARP	42	12.29.0.1 is at 1a:f3:52:99:29:2d
3	0.000207	12.29.0.10	12.29.0.1	ICMP	98	Echo (ping) request id=0x5b02, seq=1/256, ttl=64 (reply in 4)
4	0.000300	12.29.0.1	12.29.0.10	ICMP	98	Echo (ping) reply id=0x5b02, seq=1/256, ttl=64 (request in 3)
5	0.988220	12.29.0.10	12.29.0.1	ICMP	98	Echo (ping) request id=0x5b02, seq=2/512, ttl=64 (reply in 6)
6	0.988274	12.29.0.1	12.29.0.10	ICMP	98	Echo (ping) reply id=0x5b02, seq=2/512, ttl=64 (request in 5)
7	1.990097	12.29.0.10	12.29.0.1	ICMP	98	Echo (ping) request id=0x5b02, seq=3/768, ttl=64 (reply in 8)
8	1.990152	12.29.0.1	12.29.0.10	ICMP	98	Echo (ping) reply id=0x5b02, seq=3/768, ttl=64 (request in 7)
9	4.997945	1a:f3:52:99:29:2d	c6:26:00:10:96:2b	ARP	42	Who has 12.29.0.10? Tell 12.29.0.1
10	4.998465	c6:26:00:10:96:2b	1a:f3:52:99:29:2d	ARP	42	12.29.0.10 is at c6:26:00:10:96:2b
11	12.673158	c6:26:00:10:96:2b	Broadcast	ARP	42	Who has 12.29.0.20? Tell 12.29.0.10
12	12.906808	1a:f3:52:99:29:2d	c6:26:00:10:96:2b	ARP	42	12.29.0.20 is at 1a:f3:52:99:29:2d
13	12.907301	12.29.0.10	12.29.0.20	ICMP	98	Echo (ping) request id=0x5d02, seq=1/256, ttl=64 (reply in 14)
14	13.409751	12.29.0.20	12.29.0.10	ICMP	98	Echo (ping) reply id=0x5d02, seq=1/256, ttl=63 (request in 13)
15	13.665160	12.29.0.10	12.29.0.20	ICMP	98	Echo (ping) request id=0x5d02, seq=2/512, ttl=64 (reply in 16)
16	13.665774	12.29.0.20	12.29.0.10	ICMP	98	Echo (ping) reply id=0x5d02, seq=2/512, ttl=63 (request in 15)
17	14.666408	12.29.0.10	12.29.0.20	ICMP	98	Echo (ping) request id=0x5d02, seq=3/768, ttl=64 (reply in 18)
18	14.666950	12.29.0.20	12.29.0.10	ICMP	98	Echo (ping) reply id=0x5d02, seq=3/768, ttl=63 (request in 17)

Los dos primeros arps son para el broadcast y son para localizar el destino (r1 eth0), y la contestación de r1. Los dos siguientes son del router preguntando sobre pc1 (no es al broadcast ya que sabe dónde se sitúa pc1) y su contestación. Y los dos últimos son de pc1 a la dirección

Y los últimos dos arps, el primero es request al broadcast de pc1 sobre pc2 y el reply de pc2.

**Captura en pc3 (eth0).**

proxyARP-02.cap						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	02:e8:8a:03:81:92	Broadcast	ARP	42	Who has 12.29.0.20? Tell 11.29.0.1
2	0.000200	d6:c9:64:9f:f0:13	02:e8:8a:03:81:92	ARP	42	12.29.0.20 is at d6:c9:64:9f:f0:13
3	0.000092	12.29.0.10	12.29.0.20	ICMP	98	Echo (ping) request id=0x5d02, seq=1/256, ttl=63 (reply in 6)
4	0.000368	d6:c9:64:9f:f0:13	Broadcast	ARP	42	Who has 12.29.0.10? Tell 12.29.0.20
5	0.490222	02:e8:8a:03:81:92	d6:c9:64:9f:f0:13	ARP	42	12.29.0.10 is at 02:e8:8a:03:81:92
6	0.490496	12.29.0.20	12.29.0.10	ICMP	98	Echo (ping) reply id=0x5d02, seq=1/256, ttl=64 (request in 3)
7	0.746245	12.29.0.10	12.29.0.20	ICMP	98	Echo (ping) request id=0x5d02, seq=2/512, ttl=63 (reply in 8)
8	0.746502	12.29.0.20	12.29.0.10	ICMP	98	Echo (ping) reply id=0x5d02, seq=2/512, ttl=64 (request in 7)
9	1.747452	12.29.0.10	12.29.0.20	ICMP	98	Echo (ping) request id=0x5d02, seq=3/768, ttl=63 (reply in 10)
10	1.747688	12.29.0.20	12.29.0.10	ICMP	98	Echo (ping) reply id=0x5d02, seq=3/768, ttl=64 (request in 9)

Los dos mensajes de arp corresponden con el r1 preguntando a la dirección broadcast donde se sitúa el pc2 y su correspondiente reply de pc2.

Y luego encontramos otros dos en los que pc2 pregunta a broadcast sobre la dirección de pc1. Para saber la forma de enviar un mensaje si en algún momento esto sería necesario.

### 3. A partir de la captura y de las direcciones IP de r1, ¿cómo puedes saber que r1 está realizando proxy ARP?

En la primera captura, hemos capturado tanto los mensajes que iban a r1, como a pc2. Mientras que en la segunda hemos capturado solo los que llegan a pc2.

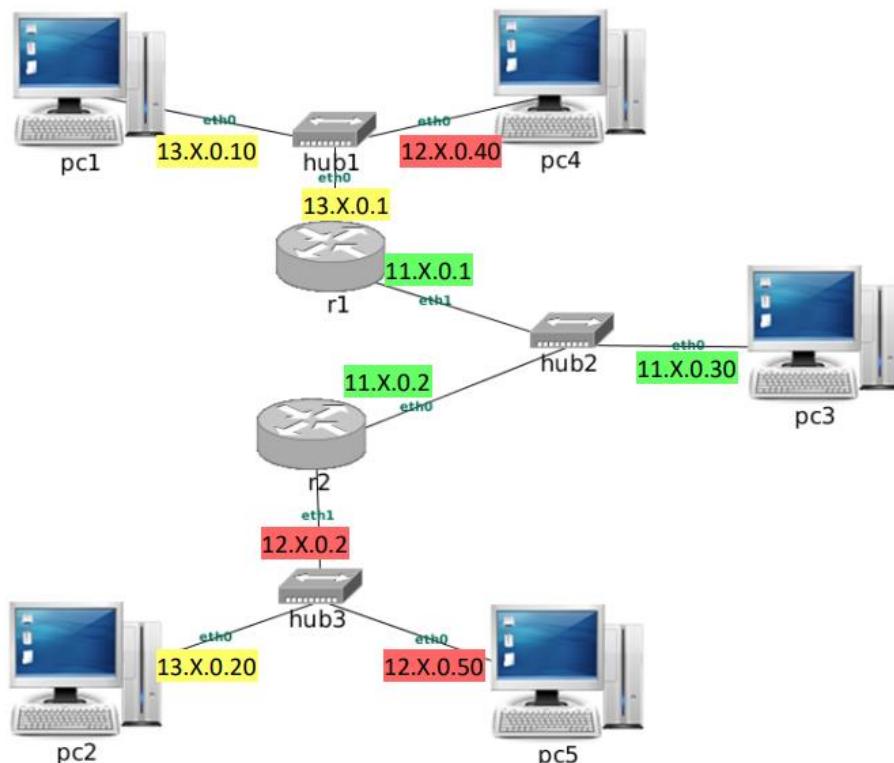
Si te fijas en los mensajes de arp, en la primera captura el, mensaje 10 responde sobre donde esta 12.29.0.10, y si te vas a la segunda captura y ves el mensaje 5 en el que también responde sobre donde esta .10, vemos que las direcciones ethernet son diferentes, eso nos indica que se ha realizado proxy arp.

### 4. Si se ha borrado la caché de ARP de pc1 vuelve a ejecutar los 2 ping anteriores y consulta la caché de ARP de pc1, indica qué observas, explicando a qué máquina/s pertenece la información almacenada.

La dirección que se ha quedado almacenada en la cache de arp corresponde con la dirección ethernet de r1 (eth0)

```
pc1:~# arp -a
pc2 (12.29.0.20) at 1A:F3:52:99:29:2D [ether] on eth0
? (12.29.0.1) at 1A:F3:52:99:29:2D [ether] on eth0
pc1:~#
```

### 4. IP aliasing



**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. Indica por qué has configurado esas direcciones IP adicionales y en qué interfaces.**

Como queremos realizar el ping desde pc2 a pc3 primero debemos introducirle la red 13.X.0.2 al router (eth1) para que pueda recibir mensajes de pc2. Como ya tiene la red 11 no es necesaria introducirla.

Además, al ser un ping debemos crear la ruta de ida y de vuelta. Al ya tener conexión pc2-r2 y r2-pc3, falta conectar pc2-pc3 y viceversa.

Ruta 13 añadida en router:

**Ifconfig eth1:0 13.29.0.2 netmask 255.255.255.0**

Conexión pc2-pc3:

**Route add -net 11.29.0.0 netmask 255.255.255.0 gw 13.29.0.2**

Conexión pc3-pc2:

**Route add -net 13.29.0.0 netmask 255.255.255.0 gw 11.29.0.2**

Para que r2 alcance pc1, en r2 introducimos:

**Route add -host 13.29.0.10 gw 11.29.0.1**

**2. Realiza una captura en r2(eth1) (ipAliasing-01.cap) y ejecuta un ping desde pc2 a pc3 enviando 3 paquetes y después ejecuta un ping desde pc5 a la dirección IP de r2(eth1). Interrumpe la captura y explica las solicitudes de ARP que observas.**

ipAliasing-01.cap							
Time	Source	Destination	Protocol	Length	Info		
<b>Aplique un filtro de visualización ... &lt;Ctrl-/&gt;</b>							
1 0.000000	26:72:e2:01:2c:63	Broadcast	ARP	42	Who has 13.29.0.2? Tell 13.29.0.20		
2 0.000101	16:39:d4:3f:bc:09	26:72:e2:01:2c:63	ARP	42	13.29.0.2 is at 16:39:d4:3f:bc:09		
3 0.000077	13.29.0.20	11.29.0.30	ICMP	98	Echo (ping) request id=0x2c02, seq=1/256, ttl=64 (reply in 4)		
4 0.010575	11.29.0.30	13.29.0.20	ICMP	98	Echo (ping) reply id=0x2c02, seq=1/256, ttl=63 (request in 3)		
5 0.988777	13.29.0.20	11.29.0.30	ICMP	98	Echo (ping) request id=0x2c02, seq=2/512, ttl=64 (reply in 6)		
6 0.989028	11.29.0.30	13.29.0.20	ICMP	98	Echo (ping) reply id=0x2c02, seq=2/512, ttl=63 (request in 5)		
7 1.987783	13.29.0.20	11.29.0.30	ICMP	98	Echo (ping) request id=0x2c02, seq=3/768, ttl=64 (reply in 8)		
8 1.987944	11.29.0.30	13.29.0.20	ICMP	98	Echo (ping) reply id=0x2c02, seq=3/768, ttl=63 (request in 7)		
9 5.011038	16:39:d4:3f:bc:09	26:72:e2:01:2c:63	ARP	42	Who has 13.29.0.20? Tell 13.29.0.2		
10 5.011287	26:72:e2:01:2c:63	16:39:d4:3f:bc:09	ARP	42	13.29.0.20 is at 26:72:e2:01:2c:63		
11 27.585016	ee:ac:2f:a8:dd:28	Broadcast	ARP	42	Who has 12.29.0.2? Tell 12.29.0.50		
12 27.585027	16:39:d4:3f:bc:09	ee:ac:2f:a8:dd:28	ARP	42	12.29.0.2 is at 16:39:d4:3f:bc:09		
13 27.585207	12.29.0.50	12.29.0.2	ICMP	98	Echo (ping) request id=0x2c02, seq=1/256, ttl=64 (reply in 14)		
14 27.585222	12.29.0.2	12.29.0.50	ICMP	98	Echo (ping) reply id=0x2c02, seq=1/256, ttl=64 (request in 13)		
15 32.586908	16:39:d4:3f:bc:09	ee:ac:2f:a8:dd:28	ARP	42	Who has 12.29.0.50? Tell 12.29.0.2		
16 32.587113	ee:ac:2f:a8:dd:28	16:39:d4:3f:bc:09	ARP	42	12.29.0.50 is at ee:ac:2f:a8:dd:28		

El primer arp es de pc2 preguntando por la dirección introducida manualmente a r2.

El segundo arp es la contestación del router sobre la dirección pedida.

Y los paquetes 9 y 10 son lo mismo, pero viceversa, para que r2 sepa donde se encuentra pc2.

Como pc2 y pc3 saben sus direcciones ya que las hemos introducido manualmente, por eso no se produce arps.

Y luego ocurre lo mismo entre r2-pc5. Pc5 pregunta por r2 y este responde sobre su dirección.

Y los paquetes 15 y 16 son lo mismo, pero viceversa, para que r2 sepa donde se encuentra pc5.

### 3. ¿Se puede saber sólo mirando el fichero de captura que en r2 no se ha configurado proxy ARP?

Si, porque las direcciones ethernet en este caso no varían.

### 4. 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.

La configuración si permite a pc1-pc5 comunicarse mutuamente. Esto es porque r2 ya tiene la red de pc5 (12.29.0.0) y como hemos introducido en r2 una ruta directa a pc1 a través de r1. Y para la vuelta del ping el mensaje sale de pc1-r1 y este tiene como ruta por defecto 11.29.0.2 perteneciente a r2, y como este es vecino de pc5, se permite la llegada correcta del ping.

### 5. 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. 6. Realiza una captura en r1(eth0) (ipAliasing-02.cap) para ver qué paquetes se intercambian cuando pc4 hace ping a pc1. Explica los resultados en la memoria.

Para conseguir que pc4 efectúe un ping hacia pc2 necesitaremos las siguientes rutas:

Primero en r1:

```
Ifconfig eth0:0 12.29.0.1 netmask 255.255.255.0
```

```
Route add -host 12.29.0.50 gw 11.29.0.2
```

En pc4:

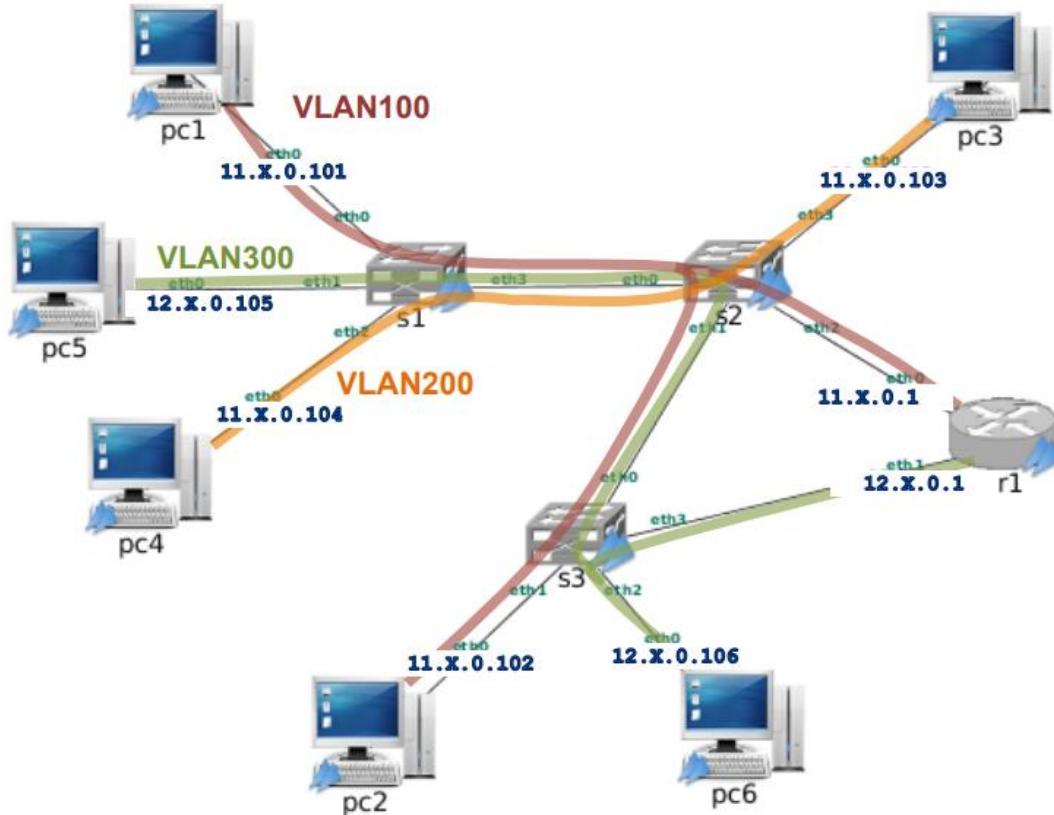
```
Route add -host 13.29.0.10 gw 12.29.0.1
```

En pc1:

```
Route add -host 12.29.0.40 gw 11.29.0.2
```

ipAliasing-02.cap						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	16:b5:f0:cf:1e:f2	Broadcast	ARP	42	Who has 12.29.0.1? Tell 12.29.0.40
2	0.000080	92:c6:d2:be:c2:34	16:b5:f0:cf:1e:f2	ARP	42	12.29.0.1 is at 92:c6:d2:be:c2:34
3	0.000115	12.29.0.40	13.29.0.10	ICMP	98	Echo (ping) request id=0xd02, seq=1/256, ttl=64 (no response fo
4	0.010221	92:c6:d2:be:c2:34	Broadcast	ARP	42	Who has 13.29.0.10? Tell 13.29.0.1
5	0.010423	4e:42:fc:f6:a0:f4	92:c6:d2:be:c2:34	ARP	42	13.29.0.10 is at 4e:42:fc:f6:a0:f4
6	0.010429	12.29.0.40	13.29.0.10	ICMP	98	Echo (ping) request id=0xd02, seq=1/256, ttl=63 (reply in 7)
7	0.010620	13.29.0.10	12.29.0.40	ICMP	98	Echo (ping) reply id=0xd02, seq=1/256, ttl=64 (request in 6)
8	0.010634	13.29.0.1	13.29.0.10	ICMP	126	Redirect (Redirect for host)
9	0.010640	13.29.0.10	12.29.0.40	ICMP	98	Echo (ping) reply id=0xd02, seq=1/256, ttl=63
10	5.007691	92:c6:d2:be:c2:34	16:b5:f0:cf:1e:f2	ARP	42	Who has 12.29.0.40? Tell 12.29.0.1
11	5.007944	16:b5:f0:cf:1e:f2	92:c6:d2:be:c2:34	ARP	42	12.29.0.40 is at 16:b5:f0:cf:1e:f2
12	5.018032	4e:42:fc:f6:a0:f4	92:c6:d2:be:c2:34	ARP	42	Who has 13.29.0.1? Tell 13.29.0.10
13	5.018044	92:c6:d2:be:c2:34	4e:42:fc:f6:a0:f4	ARP	42	13.29.0.1 is at 92:c6:d2:be:c2:34

## 5. Vlans



Todos los pcs se puede comunicar entre ellos.

Respecto a donde se podría encontrar el mensaje de arp pregunta de pc1 sobre pc2, podríamos en cualquier red ya que es un mensaje dirección broadcast y llega a todos.

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

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

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

Con el comando **./vlan-s1.sh** vemos en que redes se ha añadido una Vlan nuevo. Por ejemplo, en s2 pone:

Added Vlan with VID = 100 to if eth0, e igual con eth1. Es decir, en eth1 vemos en que redes se ha añadido. Además, si luego hacemos **brctl show** vemos esta información como ethX.100, y en las que no se ha añadido nada como ethX.

Ejecutando **brctl show**, vemos y confirmamos todas las interfaces por donde esta dicha Vlan.

En s1:

Podemos ver como tiene configurada una red Vs100 que atraviesa por eth0 y eth3

```
s1 login: root (automatic login)
Last login: Thu Feb  9 17:33:16 UTC 2023 on tty1
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           8000.1af42106ec9b    no            eth0
                                         eth3.100
s1:~#
```

En s2:

Podemos ver como tiene configurada una red Vs100 que atraviesa por eth0, eth1 y eth2

```
s2 login: root (automatic login)
Last login: Thu Feb  9 17:17:13 UTC 2023 on tty1
s2:~# ./vlan-s1.sh
-bash: ./vlan-s1.sh: No such file or directory
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           8000.22d290ea8767    no            eth0.100
                                         eth1.100
                                         eth2
s2:~#
```

En s3:

Podemos ver como tiene configurada una red Vs100 que atraviesa por eth0 y eth1

```
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
-bash: brctl: command not found
s3:~# brctl show
bridge name      bridge id          STP enabled    interfaces
vs100           8000.0afb8f134951    no            eth0.100
                                         eth1
s3:~# brctl show
```

**3. Haz un dibujo de cada switch que muestre las interfaces que intervienen en la VLAN100, indicando si estas interfaces llevan o no etiqueta VLAN.**

S1:

Vlan 100: entra por eth0 y sale por eth3.100

S2:

Vlan 100: entra por eth0.100 y sale por eth2.100 y eth1.100

S3:

Vlan 100: entra por eth0.100 y sale por eth1.

#### 4. Indica qué máquinas se pueden comunicar entre ellas.

Una vez activadas las Vlan con los comandos del principio ya no podemos establecer comunicación de todas las maquinas con todas. Por lo que las maquinas que se pueden comunicar son las que comparten Vlan.

Vlan100:

Pc1, s1, s2, r1(eth0), s3 y pc2.

Vlan200:

Pc4, s1, s2, pc3.

Vlan300:

Pc5, s1, s2, s3, pc6 y r1 (eth1).

**5. Suponiendo que la caché de ARP de pc1 está vacía, indica donde se puede capturar una solicitud de ARP que la máquina pc1 envía preguntando por la dirección Ethernet de la máquina pc2.**

**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).**

Se podrá capturar en cualquier interfaz que este en la red de la Vlan perteneciente a estos pcs. Por ejemplo, en s1 (eth0) si se capturara, pero en s1 (eth1/2) no se capturaría.

Es decir, se podrá en pc1, s1 (eth0 y eth3), s2 (eth0, eth1 y eth2), r1 (eth0), s3(eth0 y eth1) y pc2.

**6. 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.**

Primero el pc1 realiza un arp broadcast para saber la dirección de pc2 y este responde directamente hacia pc1. Aunque la dirección sea broadcast, se emite por TODAS LAS INTERFACES DE VLAN100.

Y cuando pc1 recibe al arp reply emite los mensajes ICMP hacia pc2. Saliendo de pc1, pasando por s1, s2 y s3.

**7. Asegúrate de que la caché de ARP de pc1 está vacía, bórrala si es necesario. Arranca tcpdump en las siguientes interfaces: pc1(eth0) (vlan-01.cap), s1(eth3) (vlan-02.cap), s2(eth2) (vlan-03.cap), s3(eth0) (vlan-04.cap) y pc2(eth0) (vlan-05.cap), guardando esta vez el tráfico capturado en un fichero. Realiza un ping desde pc1 a pc2.**

vian-01.cap						
No.	Time	Source	Destination	Protocol	Length	Info
1 0.000000		PcsCompu_00:00:01	Broadcast	ARP	42	Who has 11.29.0.102? Tell 11.29.0.101
2 0.003597		PcsCompu_00:00:02	PcsCompu_00:00:01	ARP	42	11.29.0.102 is at 08:00:27:00:00:02
3 0.003612		11.29.0.101	11.29.0.102	ICMP	98	Echo (ping) request id=0x4102, seq=1/256, ttl=64 (reply in 4)
4 0.004980		11.29.0.102	11.29.0.101	ICMP	98	Echo (ping) reply id=0x4102, seq=1/256, ttl=64 (request in 3)
5 4.995227		PcsCompu_00:00:02	PcsCompu_00:00:01	ARP	42	Who has 11.29.0.101? Tell 11.29.0.102
6 4.995253		PcsCompu_00:00:01	PcsCompu_00:00:02	ARP	42	11.29.0.101 is at 08:00:27:00:00:01

vlan-02.cap						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:00:01	Broadcast	ARP	46	Who has 11.29.0.102? Tell 11.29.0.101
2	0.002947	PcsCompu_00:00:02	PcsCompu_00:00:01	ARP	46	11.29.0.102 is at 08:00:27:00:00:02
3	0.003318	11.29.0.101	11.29.0.102	ICMP	102	Echo (ping) request id=0x4102, seq=1/256, ttl=64 (reply in 4)
4	0.004345	11.29.0.102	11.29.0.101	ICMP	102	Echo (ping) reply id=0x4102, seq=1/256, ttl=64 (request in 3)
5	4.994496	PcsCompu_00:00:02	PcsCompu_00:00:01	ARP	46	Who has 11.29.0.101? Tell 11.29.0.102
6	4.995023	PcsCompu_00:00:01	PcsCompu_00:00:02	ARP	46	11.29.0.101 is at 08:00:27:00:00:01

vlan-03.cap						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:00:01	Broadcast	ARP	42	Who has 11.29.0.102? Tell 11.29.0.101

vlan-04.cap						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:00:01	Broadcast	ARP	46	Who has 11.29.0.102? Tell 11.29.0.101
2	0.000322	PcsCompu_00:00:02	PcsCompu_00:00:01	ARP	46	11.29.0.102 is at 08:00:27:00:00:02
3	0.001496	11.29.0.101	11.29.0.102	ICMP	102	Echo (ping) request id=0x4102, seq=1/256, ttl=64 (reply in 4)
4	0.001888	11.29.0.102	11.29.0.101	ICMP	102	Echo (ping) reply id=0x4102, seq=1/256, ttl=64 (request in 3)
5	4.991715	PcsCompu_00:00:02	PcsCompu_00:00:01	ARP	46	Who has 11.29.0.101? Tell 11.29.0.102
6	4.993312	PcsCompu_00:00:01	PcsCompu_00:00:02	ARP	46	11.29.0.101 is at 08:00:27:00:00:01

vlan-05.cap						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:00:01	Broadcast	ARP	42	Who has 11.29.0.102? Tell 11.29.0.101
2	0.000216	PcsCompu_00:00:02	PcsCompu_00:00:01	ARP	42	11.29.0.102 is at 08:00:27:00:00:02
3	0.001437	11.29.0.101	11.29.0.102	ICMP	98	Echo (ping) request id=0x4102, seq=1/256, ttl=64 (reply in 4)
4	0.001492	11.29.0.102	11.29.0.101	ICMP	98	Echo (ping) reply id=0x4102, seq=1/256, ttl=64 (request in 3)
5	4.991139	PcsCompu_00:00:02	PcsCompu_00:00:01	ARP	42	Who has 11.29.0.101? Tell 11.29.0.102
6	4.993315	PcsCompu_00:00:01	PcsCompu_00:00:02	ARP	42	11.29.0.101 is at 08:00:27:00:00:01

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

```
s1:~# brctl show s1
bridge name      bridge id          STP enabled      interfaces
vs100           8000.ba0d80ac9666    no              eth0
                                         eth3.100
```

```
s2:~# brctl show s2
bridge name      bridge id          STP enabled      interfaces
vs100           8000.4a5c220bc51f    no              eth0.100
                                         eth1.100
                                         eth2
```

```
s3:~# brctl show s3
bridge name      bridge id          STP enabled      interfaces
vs100           8000.4a660fa82fd0    no              eth0.100
                                         eth1
```

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

### a) ¿Qué switch introduce dicha etiqueta?

El s1 ya que es el primero por el que pasa el mensaje de dicha red Vlan.

**b) ¿Qué switch elimina dicha etiqueta?**

El s3 ya que es el último por el que pasa el mensaje de dicha red Vlan.

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

Observando que los paquetes enviados tienen un apartado denominado

Virtual Lan.

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

Porque en esa interfaz eth2 solo pasa el arp request con dirección broadcast, el arp reply y los paquetes no es capaz de capturarlos esa interfaz, ya que no son el camino de ida o de vuelta escogido.

**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 que tiene dos unidades más de bits (pc1 = 335, r1 = 368), debido a que la captura realizada en s1 ya tiene implementada la parte de Virtual Lan, en la captura de p1 no.

**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 lo mismo que la respuesta anterior. Ahora son 784 frente a 816.

**10. 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) (vlan-06.cap) y otra en s1(eth3) (vlan-07.cap). Explica los resultados.**

Como anteriormente hemos activado las redes Vlan aunque pc1 y pc4 se encuentren en la misma subred no podrán comunicarse porque no se encuentran en la misma Vlan.

En las capturas se puede ver como el pc1 envía un arp broadcast y como no recibe contestación vuelve a intentarlo con otros dos, pero no es posible.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:00:01	Broadcast	ARP	42	Who has 11.29.0.104? Tell 11.29.0.101
2	0.997052	PcsCompu_00:00:01	Broadcast	ARP	42	Who has 11.29.0.104? Tell 11.29.0.101
3	1.994450	PcsCompu_00:00:01	Broadcast	ARP	42	Who has 11.29.0.104? Tell 11.29.0.101

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:00:01	Broadcast	ARP	46	Who has 11.29.0.104? Tell 11.29.0.101
2	0.996963	PcsCompu_00:00:01	Broadcast	ARP	46	Who has 11.29.0.104? Tell 11.29.0.101
3	1.994302	PcsCompu_00:00:01	Broadcast	ARP	46	Who has 11.29.0.104? Tell 11.29.0.101

## 5.2. Configuración de VLAN200

**1. Haz un dibujo de cada switch que muestre las interfaces que intervienen en la VLAN200, indicando si estas interfaces llevan o no etiqueta VLAN.**



```
s2:~# brctl show
bridge name      bridge id
vs200           8000.6aa725a80865
                           STP enabled
                           no
                           interfaces
                           eth0.200
                           eth3
s2:~#
```

```
s3:~# brctl show
bridge name      bridge id
s3              8000.3af8eeb1cac3
                           STP enabled
                           no
                           interfaces
                           eth0
                           eth1
                           eth2
                           eth3
s3:~#
```

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

La etiqueta vlan solo se encuentra en los paquetes de la captura 9, ya que se encuentra en la interfaz eth3 de s1, que es donde se encuentra el vlan. La etiqueta vlan solo se podría haber capturado en s1 (eth3) y no en (eth0).

**6. 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 una captura en pc1 (vlan-12.cap) y otra en pc4 (vlan-13.cap). Explica el resultado.**

El ping no se efectúa y no llega ya que, aunque se encuentran en la misma red no pertenecen a la vlan y no es posible que lleguen.

vlan-12.cap						
No.	Time	Source	Destination	Protocol	Length	Info
1 0.000000		PcsCompu_00:00:01	Broadcast	ARP	42	Who has 11.29.0.104? Tell 11.29.0.101
2 1.000518		PcsCompu_00:00:01	Broadcast	ARP	42	Who has 11.29.0.104? Tell 11.29.0.101
3 2.003885		PcsCompu_00:00:01	Broadcast	ARP	42	Who has 11.29.0.104? Tell 11.29.0.101

Vemos como pc1 intenta establecer conexión con pc4, pero no recibe el arp reply y no se hace el ping. Pc1 envía 3 veces el arp request pero sin resultado.

En la cap vlan-13 no se capturan paquetes.

### 5.3. Configuración de VLAN300

**1. Haz un dibujo de cada switch que muestre las interfaces que intervienen en la VLAN300, indicando si estas interfaces llevan o no etiqueta VLAN.**

S1:

Vlan 300: entra por eth1 y sale por eth3.300

S2:

Vlan 300: entra por eth0.300 y sale por eth3.

S3:

No tiene Vlan200

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

El ping no va a llegar a su destino ya que el pc1 no se encuentra en la misma vlan que el pc6, así que solo se enviarán los arp request que no recibirán si arp reply y no se envía el ping.

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

Primero se realiza el arp request preguntando sobre pc5, como están en la misma vlan si se podrá realizar el ping, así que pc6 recibe el arp reply y se envía el ping, que viajaría por pc6-s3-s2-s2-pc5. Y luego pc5 enviaría un arp request a pc6 y este su correspondiente reply.

## 4. 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).

Aparecerán un arp request broadcast en pc6(eth0), s3(eth2,3,0), s2(eth1,0), s1(eth3,1) y pc5(eth0).

La etiqueta de Vlan300 estaría en las interfaces: s3 (eth0), s2 (eth1,0) y s1 (eth3).

## 5. 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) (vlan-14.cap), s3(eth1) (vlan-15.cap), r1(eth0) (vlan-16.cap), s2(eth0) (vlan-17.cap) y pc1(eth0) (vlan-18.cap), 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.

The figure consists of three vertically stacked screenshots of the Wireshark application interface. Each screenshot shows a table of network traffic captured from a specific interface.

**vlan-14.cap (pc6(eth0)):**

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:06:40	Broadcast	ARP	42	Who has 12.29.0.1? Tell 12.29.0.106
2	0.001054	PcsCompu_00:01:11	PcsCompu_00:06:40	ARP	42	12.29.0.1 is at 08:00:27:00:01:11
3	0.001075	12.29.0.106	11.29.0.101	ICMP	98	Echo (ping) request id=0x4a02, seq=1/256, ttl=64 (reply in 4)
4	0.006607	11.29.0.101	12.29.0.106	ICMP	98	Echo (ping) reply id=0x4a02, seq=1/256, ttl=63 (request in 3)

**vlan-15.cap (s3(eth1)):**

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:00:11	Broadcast	ARP	42	Who has 11.29.0.101? Tell 11.29.0.1

**vlan-16.cap (r1(eth0)):**

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:00:11	Broadcast	ARP	42	Who has 11.29.0.101? Tell 11.29.0.1
2	0.001279	PcsCompu_00:00:01	PcsCompu_00:00:11	ARP	42	11.29.0.101 is at 08:00:27:00:00:01
3	0.001304	12.29.0.106	11.29.0.101	ICMP	98	Echo (ping) request id=0x4a02, seq=1/256, ttl=63 (reply in 4)
4	0.002519	11.29.0.101	12.29.0.106	ICMP	98	Echo (ping) reply id=0x4a02, seq=1/256, ttl=64 (request in 3)
5	5.002669	PcsCompu_00:00:01	PcsCompu_00:00:11	ARP	42	Who has 11.29.0.1? Tell 11.29.0.101
6	5.002703	PcsCompu_00:00:11	PcsCompu_00:00:01	ARP	42	11.29.0.1 is at 08:00:27:00:00:11

vlan-17.cap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:06:40	Broadcast	ARP	46	Who has 12.29.0.1? Tell 12.29.0.106
2	0.003113	PcsCompu_00:00:11	Broadcast	ARP	46	Who has 11.29.0.101? Tell 11.29.0.1
3	0.003967	PcsCompu_00:00:01	PcsCompu_00:00:11	ARP	46	11.29.0.101 is at 08:00:27:00:00:01
4	0.004435	12.29.0.106	11.29.0.101	ICMP	102	Echo (ping) request id=0x4a02, seq=1/256, ttl=63 (reply in 5)
5	0.005249	11.29.0.101	12.29.0.106	ICMP	102	Echo (ping) reply id=0x4a02, seq=1/256, ttl=64 (request in 4)
6	5.005319	PcsCompu_00:00:01	PcsCompu_00:00:11	ARP	46	Who has 11.29.0.1? Tell 11.29.0.101
7	5.005844	PcsCompu_00:00:11	PcsCompu_00:00:01	ARP	46	11.29.0.1 is at 08:00:27:00:00:11

vlan-18.cap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:00:11	Broadcast	ARP	42	Who has 11.29.0.101? Tell 11.29.0.1
2	0.000265	PcsCompu_00:00:01	PcsCompu_00:00:11	ARP	42	11.29.0.101 is at 08:00:27:00:00:01
3	0.001119	12.29.0.106	11.29.0.101	ICMP	98	Echo (ping) request id=0x4a02, seq=1/256, ttl=63 (reply in 4)
4	0.001179	11.29.0.101	12.29.0.106	ICMP	98	Echo (ping) reply id=0x4a02, seq=1/256, ttl=64 (request in 3)
5	5.001031	PcsCompu_00:00:01	PcsCompu_00:00:11	ARP	42	Who has 11.29.0.1? Tell 11.29.0.101
6	5.002602	PcsCompu_00:00:11	PcsCompu_00:00:01	ARP	42	11.29.0.1 is at 08:00:27:00:00:11

Los paquetes capturados con Vlan estarían en las interfaces: s3 (eth0), s2 (eth1,0) y s1 (eth3).

Y efectivamente en la captura Vlan-17 podemos observar que existe dicha etiqueta.

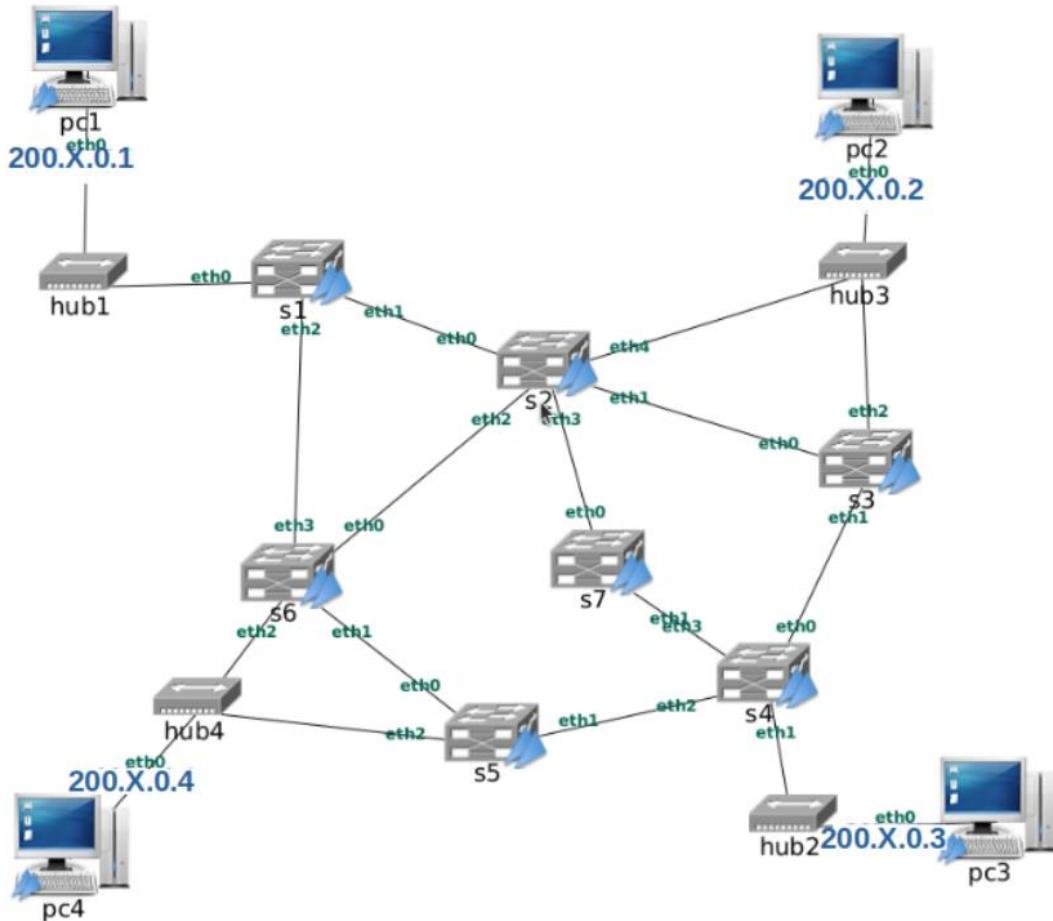
vlan-17.cap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	PcsCompu_00:06:40	Broadcast	ARP	46	Who has 12.29.0.1? Tell 12.29.0.106
2	0.003113	PcsCompu_00:00:11	Broadcast	ARP	46	Who has 11.29.0.101? Tell 11.29.0.1
3	0.003967	PcsCompu_00:00:01	PcsCompu_00:00:11	ARP	46	11.29.0.101 is at 08:00:27:00:00:01
4	0.004435	12.29.0.106	11.29.0.101	ICMP	102	Echo (ping) request id=0x4a02, seq=1/256, ttl=63 (reply in 5)
5	0.005249	11.29.0.101	12.29.0.106	ICMP	102	Echo (ping) reply id=0x4a02, seq=1/256, ttl=64 (request in 4)
6	5.005319	PcsCompu_00:00:01	PcsCompu_00:00:11	ARP	46	Who has 11.29.0.1? Tell 11.29.0.101
7	5.005844	PcsCompu_00:00:11	PcsCompu_00:00:01	ARP	46	11.29.0.1 is at 08:00:27:00:00:11

Frame 1: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)  
 ▾ Ethernet II, Src: PcsCompu\_00:06:40 (08:00:27:00:06:40), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
 ▾ 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 300  
 ...0 ..... = Priority: Best Effort (default) (0)  
 ...0 ..... = DEI: Ineligible  
 .... 0001 0010 1100 = ID: 300  
 Type: ARP (0x0806)  
 ▾ Address Resolution Protocol (request)

En las demás capturas no aparece este protocolo ya que no se han realizado en ninguna de las interfaces que he mencionado anteriormente.

## 6. Spanning Tree Protocol (STP)



**1. Consulta en la información de cada switch cuál es su identificador. Deduce a partir de este identificador la prioridad configurada en cada switch.**

S1: 1000.0007e9290200

S2: 2000.0007e9290200

S3: 3000.0007e9290200

S4: 4000.0007e9290200

S5: 5000.0007e9290200

S6: 6000.0007e9290200

S7: 1000.0007e9290200

El menor será s1 o s7, pero como s1 es menor este es el switch raíz.

**2. Sin consultar la información de STP en cada switch calcula:**

**a) La switch raíz.**

S1, debido al identificador y su menor número con s7.

**b) Para cada switch indica cuál es su puerto raíz.**

S2: Puerto en eth0.

S3: Puerto en eth0. (aunque en brctl showstp ninguna interfaz coincide con el puerto raíz)

S4: Puerto en eth3.

S5: Puerto en eth0. (aunque en brctl showstp ninguna interfaz coincide con el puerto raíz)

S6: Puerto en eth3.

S7: Puerto en eth0.

**c) Para cada tramo de medio compartido (LAN), calcula cuál es switch designado (DS) y su puerto designado (DP).**

S1: Todas sus interfaces son DP.

S2: Todas sus interfaces son DP menos eth0. (raíz)

S3: Todas sus interfaces son DP menos eth2 (blok)

S4: Todas sus interfaces son DP menos eth0,2 (blok) y eth3 rp (raíz)

S5: Todas sus interfaces son DP menos eth2. (blok)

S6: Todas sus interfaces son DP menos eth0. (blok) y eth3 rp (raíz)

S7: Dp en eth1, eth0 es RP (raíz)

**d) Deduce cuáles son los puertos bloqueados.**

S1: No tiene puertos bloqueados.

S2: No tiene puertos bloqueados.

S3: Eth2

S4: Eth0 y eth2

S5: Eth2

S6: Eth0

S7: Dp en eth1, eth0 es RP (raíz)

**e) Realiza un dibujo con el árbol de expansión que tendrán configurados los switches.**

**4. Arranca un tcpdump para capturar el tráfico en s4(eth2) almacenando la captura en el fichero stp-01.cap. Déjala al menos durante 10 segundos e interrumpe la captura. Antes de abrir la captura ¿qué mensajes crees que encontrarás y quién es el switch que los ha generado?**

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Intel_29:05:01	Spanning-tree-(for... STP	52 Conf.	Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
2	1.993788	Intel_29:05:01	Spanning-tree-(for... STP	52 Conf.	Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
3	3.992855	Intel_29:05:01	Spanning-tree-(for... STP	52 Conf.	Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
4	5.993066	Intel_29:05:01	Spanning-tree-(for... STP	52 Conf.	Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
5	7.997573	Intel_29:05:01	Spanning-tree-(for... STP	52 Conf.	Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
6	9.996853	Intel_29:05:01	Spanning-tree-(for... STP	52 Conf.	Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
7	11.998119	Intel_29:05:01	Spanning-tree-(for... STP	52 Conf.	Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
8	13.993708	Intel_29:05:01	Spanning-tree-(for... STP	52 Conf.	Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
9	15.991970	Intel_29:05:01	Spanning-tree-(for... STP	52 Conf.	Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
10	17.990464	Intel_29:05:01	Spanning-tree-(for... STP	52 Conf.	Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	

```

> Frame 6: 52 bytes on wire (416 bits), 52 bytes captured (416 bits)
- IEEE 802.3 Ethernet
  > Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  > Source: Intel_29:05:01 (00:07:e9:29:05:01)
  Length: 38
- Logical-Link Control
  > DSAP: Spanning Tree BPDU (0x42)
  > SSAP: Spanning Tree BPDU (0x42)
  > Control field: U, func=UI (0x03)
- Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Configuration (0x00)
  > BPDU flags: 0x00
  > Root Identifier: 4096 / 0 / 00:07:e9:29:01:00
    Root Bridge Priority: 4096
    Root Bridge System ID Extension: 0
    Root Bridge System ID: Intel_29:01:00 (00:07:e9:29:01:00)
  Root Path Cost: 200
  > Bridge Identifier: 20480 / 0 / 00:07:e9:29:05:00
    Bridge Priority: 20480
    Bridge System ID Extension: 0
    Bridge System ID: Intel_29:05:00 (00:07:e9:29:05:00)
  Port identifier: 0x8002
  Message Age: 0,01953125
  Max Age: 20
  Hello Time: 2
  Forward Delay: 15

```

Los mensajes los ha generado

## 5. Comprueba tus suposiciones y anota la información más relevante que encuentres en los mensajes:

### a) Identificador del switch raíz

### b) Coste al switch raíz

Lo encontramos en root path cost: 200

### c) Identificador del switch que envía el mensaje

Lo encontramos en root identi

fier: 4096 / 0 / 00:07: e9: 29: 01: 00

### d) Número de puerto que está usando el switch que envía el mensaje

8002. Es decir, eth3

## 6. Si ejecutáramos un ping desde pc1 a pc3, explica por dónde le llegarían los mensajes ICMP Echo Request a s4.

Los mensajes salen de pc1 dirección pc3 y pasan por s7, llegando a s4 por eth3 y saliendo hacia px3 por s4 eth1

## 7. ¿Y los mensajes ICMP Echo Reply por donde los recibiría s1?

Los mensajes saldrían de pc3 y llegarían a s1 a través de eth1, es decir el por s2.

Con esto concluimos que el camino de los paquetes del ping pc1-pc3 pasan por s1-s2-s7-s4-pc3, y para los reply pues al revés pc3-s4-s7-s2-s1-pc1.

**8. Inicia una captura en la interfaz de s4 que has calculado en el apartado anterior. Arranca un ping -c 1 en pc1 dirigido a pc3 almacenando el contenido en stp-02.cap. Interrumpe la captura. Para cada uno de los mensajes capturados indica cuál será el camino que han seguido esos mensajes a través de los switches del escenario y cuál de esos mensajes ha provocado el aprendizaje de una dirección Ethernet en los switches de la figura.**

stp-02.cap						
Time	Source	Destination	Protocol	Length	Info	
1 0.000000	Intel_29:07:01	Spanning-tree-(for... STP	STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
2 2.002438	Intel_29:07:01	Spanning-tree-(for... STP	STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
3 4.001007	Intel_29:07:01	Spanning-tree-(for... STP	STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
4 6.004817	Intel_29:07:01	Spanning-tree-(for... STP	STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
5 7.998953	Intel_29:07:01	Spanning-tree-(for... STP	STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
6 10.004739	Intel_29:07:01	Spanning-tree-(for... STP	STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	
7 10.203789	Intel_11:29:11	Broadcast	ARP	42	Who has 200.29.0.3? Tell 200.29.0.1	
8 10.204293	Intel_33:29:33	Intel_11:29:11	ARP	42	200.29.0.3 is at 00:07:e9:33:29:33	
9 10.206161	200.29.0.1	200.29.0.3	ICMP	98	Echo (ping) request id=0x2202, seq=1/256, ttl=64 (reply in 10)	
10 10.206612	200.29.0.3	200.29.0.1	ICMP	98	Echo (ping) reply id=0x2202, seq=1/256, ttl=64 (request in 9)	
11 12.007782	Intel_29:07:01	Spanning-tree-(for... STP	STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00 Cost = 200 Port = 0x8002	

El camino recorrido para la ida ha sido pc1-s1-s2-s7-s4-pc3, mientras que los arp reply han ido justo al revés. Los mensajes que han hecho aprender las direcciones de los pcs han sido los mensajes broadcast.

**9. Apaga el switch s3 y espera al menos 2 minutos a que se haya reconfigurado el nuevo árbol de expansión. Sin consultar la información de STP en cada switch indica:**

**a) Para cada switch indica cuál es su puerto raíz.**

S2: Puerto en eth0.

S3: OFF

S4: Puerto en eth3.

S5: Puerto en eth0. (aunque en brctl showstp ninguna interfaz coincide con el puerto raíz)

S6: Puerto en eth3.

S7: Puerto en eth1.

**b) Para cada tramo de medio compartido (LAN), calcula cuál es switch designado (DS) y su puerto designado (DP).**

S1: Todas sus interfaces son DP.

S2: Todas sus interfaces son DP menos eth0. (Raíz)

S3: OFF

S4: Todas sus interfaces son DP menos eth2 (blok) y eth3 rp (raíz). Eth0 apagado.

S5: Todas sus interfaces son DP menos eth2. (blok)

S6: Todas sus interfaces son DP menos eth0. (blok) y eth3 rp (raíz)

S7: Dp en eth1, eth0 es RP (raíz)

**c) Deduce cuáles son los puertos bloqueados.**

S4: eth2

S5: eth2

S6: eth0

d) Realiza un dibujo con el nuevo árbol de expansión que tendrán configurados los switches.

**10. Comprueba tus suposiciones consultando la información STP en cada switch.**

Todo igual que lo descrito anteriormente.

**11. Si ejecutáramos un ping desde pc1 a pc3, explica por dónde le llegarían los mensajes ICMP Echo Request a s4.**

Llegarían a través de s7 y entrarían a s4 por eth3.

**12. ¿Y los mensajes ICMP Echo Reply por donde los recibiría s1?**

Los mensajes llegarían a través de s2 y entrarían a s1 por eth1.

**13. Inicia una captura en la interfaz de s4 que has calculado en el apartado anterior. Arranca un ping -c 1 en pc1 dirigido a pc3 almacenando el contenido en stp-03.cap. Interrumpe la captura. Para cada uno de los mensajes capturados indica cuál es el camino que han seguido esos mensajes a través de los switches del escenario y cuál de esos mensajes ha provocado el aprendizaje de una dirección Ethernet en los switches de la figura.**

Time	Source	Destination	Protocol	Length	Info
1 0.000000	Intel_29:07:01	Spanning-tree-(for... STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00	Cost = 200 Port = 0x8002
2 1.998067	Intel_29:07:01	Spanning-tree-(for... STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00	Cost = 200 Port = 0x8002
3 3.999406	Intel_29:07:01	Spanning-tree-(for... STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00	Cost = 200 Port = 0x8002
4 5.998612	Intel_29:07:01	Spanning-tree-(for... STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00	Cost = 200 Port = 0x8002
5 7.992870	Intel_29:07:01	Spanning-tree-(for... STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00	Cost = 200 Port = 0x8002
6 10.000454	Intel_29:07:01	Spanning-tree-(for... STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00	Cost = 200 Port = 0x8002
7 11.392545	Intel_11:29:11	Broadcast	ARP	42	Who has 200.29.0.3? Tell 200.29.0.1
8 11.392833	Intel_33:29:33	Intel_11:29:11	ARP	42	200.29.0.3 is at 00:07:e9:33:29:33
9 11.394309	200.29.0.1	200.29.0.3	ICMP	98	Echo (ping) request id=0x2602, seq=1/256, ttl=64 (reply in 10)
10 11.394763	200.29.0.3	200.29.0.1	ICMP	98	Echo (ping) reply id=0x2602, seq=1/256, ttl=64 (request in 9)
11 11.999513	Intel_29:07:01	Spanning-tree-(for... STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00	Cost = 200 Port = 0x8002
12 13.991551	Intel_29:07:01	Spanning-tree-(for... STP	52	Conf. Root = 4096/0/00:07:e9:29:01:00	Cost = 200 Port = 0x8002

El camino recorrido para la ida ha sido pc1-s1-s2-s7-s4-pc3, mientras que los arp reply han ido justo al revés. Los mensajes que han hecho aprender las direcciones de los pcs han sido los mensajes broadcast.