Práctica 4: Agregación de enlaces y monitorización en switches Cisco. Bridging y 802.1Q en GNU/Linux

1- Objetivos

En esta práctica veremos cómo emplear 802.1Q en PCs Linux. Veremos cómo agregar varios interfaces Ethernet en uno solo lógico y finalmente alteraremos el comportamiento de un conmutador para que nos permita ver todo el tráfico que reenvía por una VLAN.

2- Conocimientos previos

- Configuración IP básica de PCs con Linux y de routers Cisco
- VLANs y 802.1Q
- Configuración básica y de VLANs en conmutadores Cisco
- 802.3ad

3- Etherchannel

Etherchannel es la forma que tiene Cisco de llamar a la agregación de interfaces Ethernet. La gestión de esta agregación se puede basar en el protocolo estándar LACP (802.3ad) o en el protocolo propietario de Cisco PAgP.

Cuando se crea un EtherChannel se crea un interfaz lógico. A partir de ahí se pueden asignar manualmente interfaces físicos al interfaz lógico del Etherchannel.

En este apartado vamos a crear un canal entre dos conmutadores que agregue dos puertos de cada uno.

En primer lugar lleve a cabo la configuración de la figura 1, donde los dos enlaces entre los conmutadores (por ejemplo los puertos 23 y 24 en ambos) estarán configurados como trunks y spanning-tree habrá bloqueado automáticamente uno de ellos. Coloque los tres PCs en la misma LAN y genere tráfico simultáneamente desde los PCs B y C hacia el PC A. Compruebe por qué enlace entre los switches circula el tráfico.

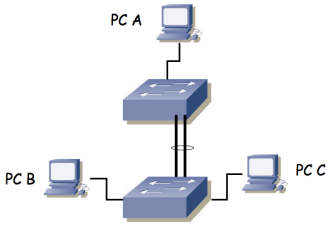

Figura 1.- Escenario con interfaz Etherchannel

Ahora crearemos la agrupación de los enlaces de trunk. En cada switch debemos colocar los puertos de interconexión dentro del mismo grupo, para lo cual, dentro de la configuración de cada uno de esos interfaces deberíamos hacer:

```
Switch(config-if)# channel-group 1 mode active
```

Suponiendo que queremos añadir el interfaz al grupo 1 y queremos que emplee LACP (802.3ad en lugar de la solución propietaria de Cisco, ¿qué ventajas puede tener emplear la solución estándar?)

Dado que vamos a repetir el mismo comando en varios interfaces, existe otra forma más cómoda de hacerlo. Podemos aplicar los mismos comandos a varios interfaces. Para ello debemos entrar en modo configuración de varios al mismo tiempo. Por ejemplo, en este caso se podría hacer del siguiente modo:

```
Switch(config)# interface range fa0/23 -24
Switch(config-if-range)# channel-group 1 mode active
```

El resultado es el mismo, es una simple cuestión de comodidad y rapidez.

Por supuesto, debemos repetir los comandos de configuración en el segundo conmutador.

Podemos ver ahora información sobre el estado de la agrupación con:

```
Switch> show interfaces etherchannel
```

Analice la información que puede obtener.

También podemos ver el nuevo interfaz lógico (`Port-channel1`) en el resultado de comandos como por ejemplo `show interfaces summary`.

Estudie el estado actual de los puertos en el spanning-tree.

Repita el envío de tráfico simultáneamente de PC B y C a PC A.

Punto de control: Muestre esta última configuración a su profesor de prácticas.

¿Qué enlace emplean las tramas? En el laboratorio puede probar a transferir un fichero grande para ver la actividad en las luces o en los contadores que llevan los conmutadores para cada interfaz. En PacketTracer puede intercalar un Sniffer para ver por dónde circulan los paquetes o simplemente emplear el modo Simulation.

Y si genera tráfico de PC A a PC B y de PC A a PC C simultáneamente ¿qué camino siguen?

Las respuestas a las cuestiones anteriores dependen de cómo estén haciendo los conmutadores el reparto de la carga entre los enlaces del etherchannel. Cisco permite dos modos de reparto de carga que se controlan con el comando:

```
Switch(config)# port-channel load-balance {dst-mac|src-mac}
```

Estudie las diferentes opciones y pruébelas. En PacketTracer tendrá más opciones, tal vez pudiendo hacer balanceo en base a más campos de los paquetes que solamente las direcciones MAC.

En la topología de la figura 1 ¿cuál sería la configuración de reparto de carga más rentable para aprovechar ambos enlaces?

4- SPAN

Como recordará, si tenemos varios PCs conectados en un hub o en varios que formen un solo dominio de colisión, cualquiera de ellos puede ver el tráfico que generan todos los demás. Esta característica es muy útil en muchas tareas de depuración de problemas. Sin embargo, en el caso de tener conmutadores el dominio de colisión se limita a un puerto por lo que desde un PC conectado a un puerto del conmutador no podremos ver los paquetes que se intercambian otras máquinas (salvo que vayan dirigidos a la MAC de broadcast o a alguna de multicast).

Algunos conmutadores soportan la funcionalidad SPAN (*Switched Port Analyzer*) que permite que las tramas que se envían/reciben por uno o varios puertos o VLANs se copien en otro puerto al que se conectaría el analizador de tráfico.

Con los conmutadores Cisco del laboratorio o PacketTracer podemos configurar un puerto de SPAN por el que se replique el tráfico de otros puertos del conmutador. Esto se hace con el comando `monitor` en modo configuración.

En el laboratorio conecte 3 PCs a uno de sus conmutadores Cisco. Establezca una comunicación entre dos de ellos y compruebe que desde el tercero no puede capturar esos paquetes. Ahora active en el conmutador que clone en el puerto del tercer PC los paquetes enviados y recibidos por uno de los puertos de los otros PCs. En Cisco PacketTracer haga la misma configuración pero en lugar de poner un tercer PC para recibir el tráfico del puerto de SPAN conecte ahí un Sniffer.

5- Bridging en GNU/Linux

Igual que hemos configurado un PC con kernel Linux para que actúe como un router IP podemos configurarlo para que reenvíe tramas Ethernet como haría un puente. Emplearemos para ello las

`bridge-utils`. En el entorno de simulación usaremos IMUNES, el cual nos permite lanzar mini-máquinas virtuales (en realidad contenedores) para emular cada PC.
Cree en IMUNES la topología de la Figura 2.

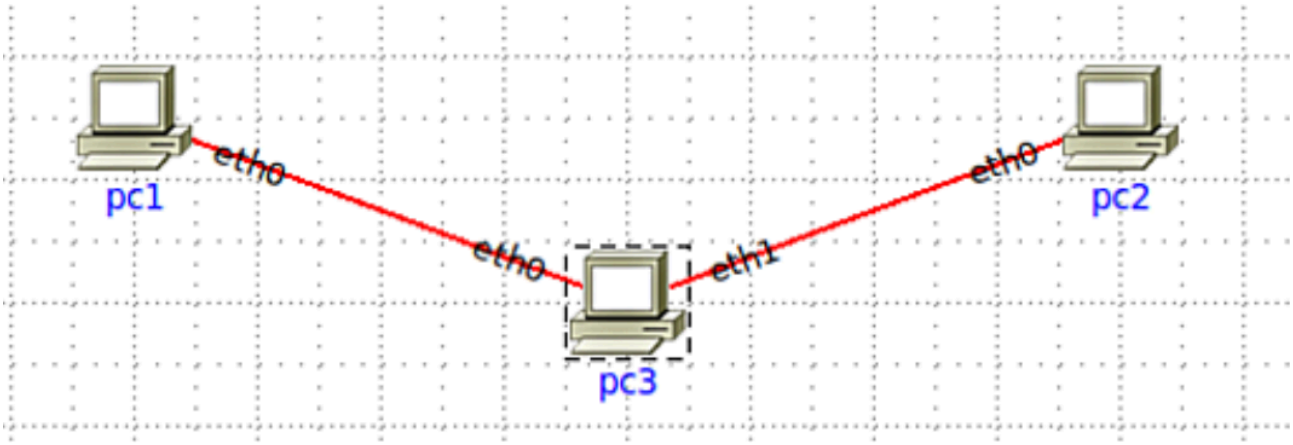

Figura 2.- Topología básica para *bridging* en IMUNES

Desactive la autoconfiguración de direcciones en los PCs y lance el experimento. Configure pc1 y pc2 en la misma subred IP (no necesitan ruta por defecto, no hay routers). En pc3 no necesita configurar direcciones IP ya que lo que queremos es conseguir que actúe como un puente entre esas dos máquinas.

Para crear puentes en el kernel Linux, así como para gestionarlos, podemos emplear el comando `brctl`. Cree un puente en pc3 con:

```
# brctl addbr <nombreDelPuente>
```

A continuación añada los dos interfaces a ese puente con la opción "addif" de brctl (si ejecuta brctl sin opciones verá las que tiene y cómo se usan). Finalmente debe activar el puente con un:

```
# ifconfig <nombreDelPuente> up
```

Desde ese momento debería funcionar la comunicación entre pc1 y pc2 y se inspecciona las tramas que envían y reciben verá que no han sido modificadas por pc3.

Por supuesto, desde el momento en que tenemos puentes tenemos un problema ante posibles bucles. En la topología de la Figura 2 es imposible un bucle pero podríamos crearlo con más puentes. Prueba a crear un bucle con una topología similar a la de la Figura 3.

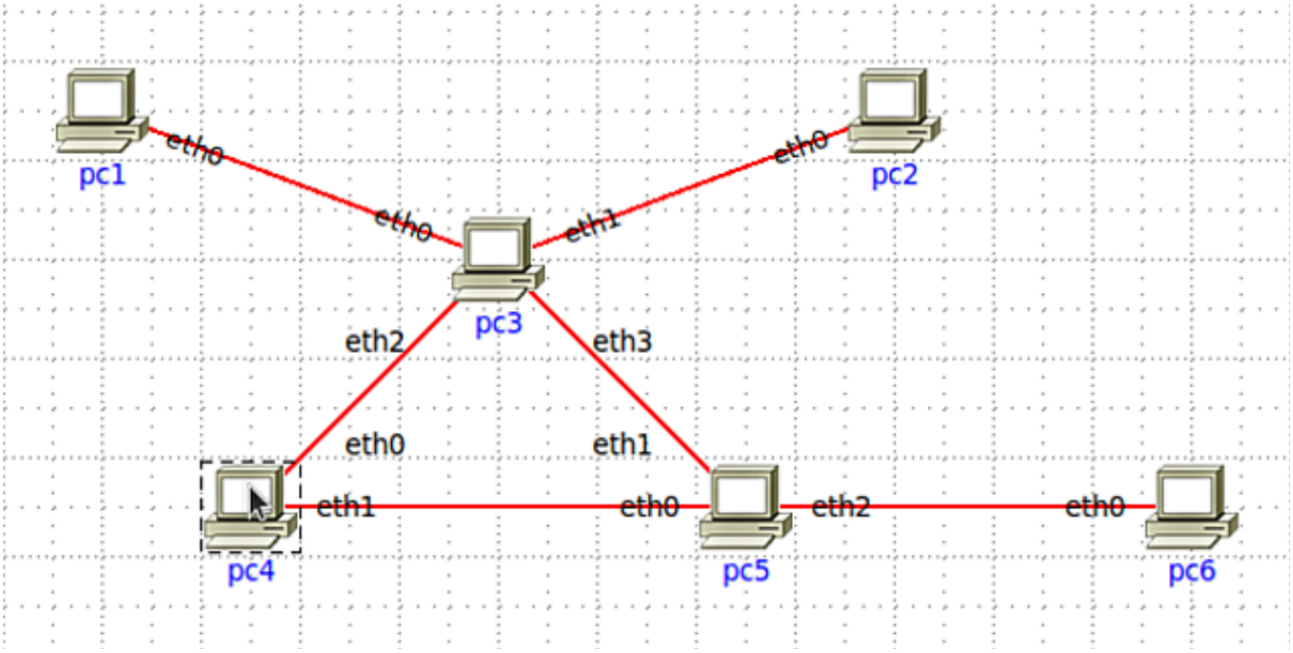

Figura 3.- pc1, pc2 y pc6 son hosts en la misma LAN. Pc3, pc4 y pc5 actúan como puentes entre todos sus interfaces

Punto de control: Muestre un escenario similar a la figura 3. Mantenga un interfaz del bucle desconectado hasta que quiera mostrar el efecto del bucle.

6- 802.1Q en un PC

A continuación vamos a ver cómo emplear 802.1Q en un PC con Linux. Haremos que un PC tenga interfaces lógicos en varias VLANs con un solo interfaz físico e incluso enrute entre ellas.

Versión en el Laboratorio

Conecte el interfaz 0 del PC A al switch1 y configure ese puerto del switch en modo acceso y en la VLAN 2. Configure la IP del interfaz del PC dentro de la subred A. Conecte el interfaz 0 del PC B al switch1 y configure ese puerto del switch en modo acceso y en la VLAN 3. Configure la IP del interfaz del PC dentro de la LAN B.

Conecte ahora el PC C a un puerto del switch1. Configure ese puerto del switch en trunk. El resultado del conexionado físico se ve en la figura 2.

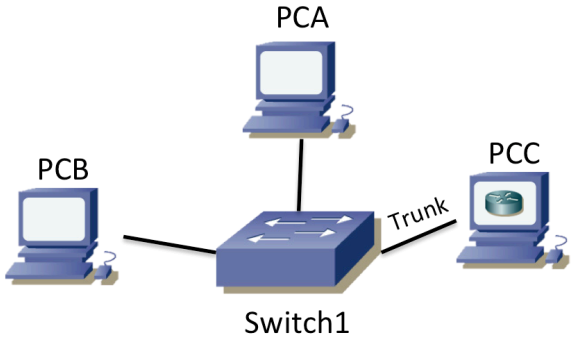

Figura 4. – Topología física

Para crear los interfaces lógicos en el PC primero debemos asegurarnos de que el interfaz físico está activo, por ejemplo:

```
$ sudo ifconfig eth0 up
```

Ahora puede emplear el comando `vconfig` para crear cada interfaz lógico asociado a una VLAN. Por ejemplo, para crear uno asociado a la VLAN 2 (empleando encapsulado 802.1Q) sería:

```
$ sudo vconfig add eth0 2
```

Nota: puede que salga el siguiente mensaje de aviso. Esto no quiere decir que no se haya creado la interfaz lógico, para eso hay que comprobarlo con ***`ifconfig`***.

```
Could not open /proc/net/vlan/config. Maybe you need to load the 802.1q
module, or maybe you are not using PROCS
```

A partir de ese momento debería tener un interfaz llamado `eth0.2`

Si le asigna dirección IP al interfaz consigue por el camino que se active. Si no le asigna dirección pero quiere que esté activo recuerde hacerle “up”.

Si en algún momento quiere borrar un subinterfaz de estos deberá emplear el mismo comando vconfig con la opción ‘`rem`’

Cree de esta forma el subinterfaz en el PC C para la VLAN 2 y el de la VLAN 3. Asigne dirección IP a cada uno de ellos en la subred que les corresponde. Compruebe que puede hacer ping desde PC C a PC A y a PC B. Puede ver las cabeceras de las tramas con wireshark, tanto en PC C como en A y B.

Configure ruta por defecto en PC A y PC B con siguiente salto la dirección IP de PC C en el interfaz en la misma subred que el PC. Active el reenvío de paquetes IP en PC C. El resultado en capa 3 se

ve en la figura 3. Pruebe ahora a hacer ping entre PC A y PC B y vea las tramas en los 4 interfaces. Analice cómo cambia el encapsulado Ethernet y la cabecera IP.

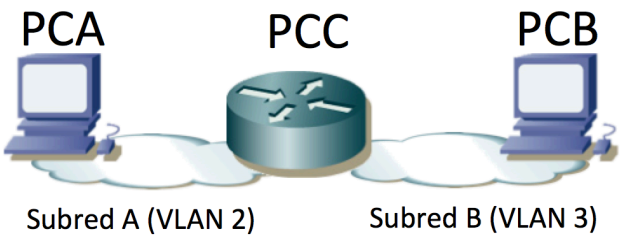

Figura 5. – Topología capa 3 una vez completada la configuración

Versión en IMUNES

Como en PackerTracer no disponemos de equipos Linux emplearemos IMUNES, con la limitación de que ahí no podemos usar conmutadores Cisco, así que tendremos que emplear puentes construidos con otros PCs Linux.

Cree la topología física de la figura 6. En esta ocasión pc1 será el equipo que tenga dos subinterfaces, uno en cada VLAN y por lo tanto será quien emplee encapsulado 802.1Q y el que al final de este apartado esté enrutando entre las dos subredes. Así, pc1 deberá tener un subinterfaz en la subred A y otro en la subred B. El pc2 tendrá un interfaz normal en la subred A y el pc5 lo tendrá en la subred B. Los equipos pc4 y pc3 van a actuar como puentes, cursando el tráfico de dos VLANs

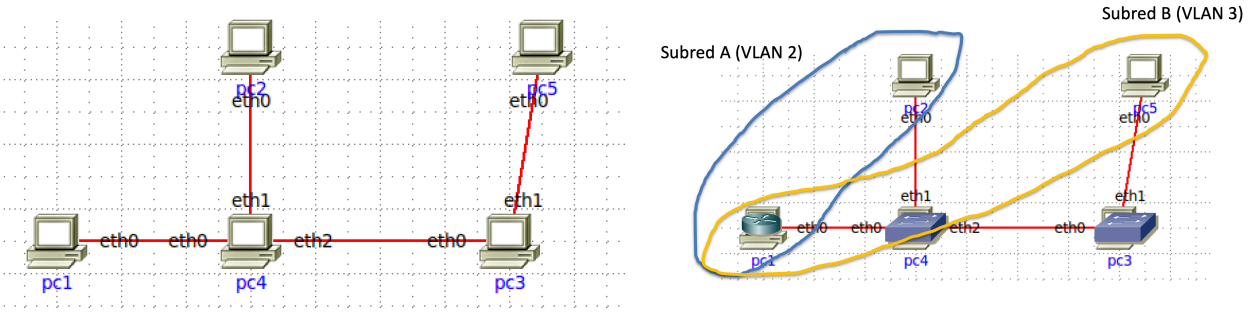

Figura 6. – Topología física en IMUNES

El interfaz de pc2 configúrelo normalmente con una dirección IP de la subred A y lo mismo con el de pc5 en la subred B. De igual forma a como se ha explicado en el apartado "Versión en el Laboratorio" cree dos subinterfaces en pc1, uno de ellos en la VLAN 2 y el otro en la VLAN 3. El primero de ellos deberá tener una dirección IP en la subred A y el segundo en la subred B.

Ahora, para que funcione la comunicación entre pc1 y pc2 por la subred A necesitamos crear un puente en pc4. Cree un puente en pc4. Cree un subinterfaz del interfaz eth0 de pc4 en la VLAN 2 y añádalo a ese puente. Añada al puente también el interfaz de pc4 que va a pc2. Recuerde activar todos esos interfaces (incluido el interfaz que representa al puente). Con esto ya debería funcionar la comunicación en capa 2 entre pc1 y pc2. Si mira las tramas que salen de pc1 y las que llegan a pc2 son idénticas salvo por el encapsulado 802.1Q que tienen al salide de pc1 y que no tienen al entrar en pc2.

Para conseguir la comunicación en capa 2 entre pc1 y pc5 por la subred B hay que hacer llegar el tráfico puenteado de esa VLAN pasando por pc4 y pc3. En pc4 necesitará crear un nuevo puente y añadir a él un subinterfaz de la VLAN 3 de eth0 y otro de eth2. En pc3 tendrá que crear un puente al que añada eth1 y un subinterfaz de eth0 en la VLAN 3.

Con esto ya deberíamos tener comunicación entre pc2 y pc1 por la VLAN 2 y entre pc5 y pc1 por la

VLAN 3. Para lograr el enrutamiento tenemos que configurar las rutas por defecto en pc2 y pc5 apuntando a las direcciones de los correspondientes subinterfaces de pc1.

Punto de control: Muestre esta última configuración a su profesor de prácticas mediante una captura en la que se vea el cambio en el encapsulado al pasar un paquete IP de una VLAN a la otra.