

# Práctica 6: 802.1Q en routers Cisco. Routing y Bridging

## 1- Objetivos

En esta práctica veremos el empleo de 802.1Q en interfaces de routers Cisco, y cómo mezclar conmutación en capa 2 y 3 en routers Cisco y PCs con GNU/Linux.

## 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
- Configuración de puentes en Linux

## 3- 802.1Q en router Cisco

Sobre el interfaz físico Ethernet de un router Cisco se pueden configurar interfaces “virtuales”, en concreto nos interesan en esta práctica los “subinterfases”. Indica la documentación de Cisco:

*“A subinterface is a mechanism that allows a single physical interface to support multiple logical interfaces or networks. That is, several logical interfaces or networks can be associated with a single hardware interface.”*

Queremos que un interfaz del router pueda enviar tramas ethernet con encapsulado 802.1Q y etiquetas de diferentes VLAN. Si el puerto del conmutador al que está conectado está en “trunk” y acepta tramas de todas esas VLANs entonces podremos conseguir que efectivamente el router tenga un interfaz lógico en cada VLAN.

En el laboratorio conecte un interfaz ethernet del router2 a un puerto del switch1. En PacketTracer puede emplear un 2901 para hacer de router. Active ese interfaz (no shutdown) en el router y configure el puerto del switch para que esté en trunk. Cree un par de VLANs en el switch (por ejemplo la VLAN 2 y la 3).

Vamos a crear un subinterfaz en el router para cada VLAN. El subinterfaz representará al interfaz del router en esa LAN. Los subinterfases se indentifican por un número. Para recordar mejor qué subinterfaz corresponde a cada VLAN los identificaremos con el número de la VLAN (aunque no es obligatorio). Para entrar en el modo de configuración del subinterfaz 2 deberá usar (según cuál sea el interfaz que esté empleando):

```
Router(config)# interface GigabitEthernet0/0/0.2
```

Ahora, en el modo de configuración de ese subinterfaz indique que desea emplear encapsulado 802.1Q y la VLAN 2 (comando encapsulation).

Asigne dirección IP a este subinterfaz en una LAN, por ejemplo en la LAN 192.168.1.0/24 (que llamaremos LAN A).

Repita los pasos con el subinterfaz 3, asignándolo a la VLAN 3 y con dirección IP en la LAN B (que debe ser independiente de la LAN A)

Puede ver ahora ambos interfaces por ejemplo con:

```
Router# show ip interface brief
```

Y vea también las entradas en la tabla de rutas.

Para probar la configuración colocaremos un PC en cada una de esas VLANs.

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 LAN A. Configure la ruta por defecto de este PC con siguiente salto la IP del router en la LAN 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. Configure la ruta por defecto de este PC con siguiente salto la IP del router en la LAN B.

Pruebe a hacer ping y traceroute entre ambos PCs y verifique el correcto funcionamiento de la comunicación y que efectivamente los paquetes están siendo enrutados por router.

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

#### 4- PC Linux como puente con VLANs y router IP

Hemos visto un switch capa 2/3 que era un switch capa 2 con soporte de VLANs donde creábamos interfaces capa 3 en cada VLAN, entre los que el equipo era capaz de encaminar.

Vamos a construir algo parecido con un PC Linux (en IMUNES, con suficientes PCs se podría hacer en el laboratorio). Ya hemos visto cómo construir un puente con un PC, así como un puente con soporte de VLANs (en estos ejemplos hemos creado puentes para cada VLAN). Lo que haremos ahora es configurar esos interfaces capa 3 en esas VLANs en el equipo que queremos que haga conmutación en capa 2 y también en capa 3.

La figura 1 muestra la topología física a construir. Los pc1 y pc5 son hosts normales en la VLAN 2. Los pc7 y pc6 son host normales en la VLAN 3. Los pc2, pc3 y pc4 van a actuar como puentes, con soporte de VLANs, empleando trunking 802.1Q en los enlaces que los interconectan entre si. De esa forma obtenemos una topología funcional similar a la figura 2.

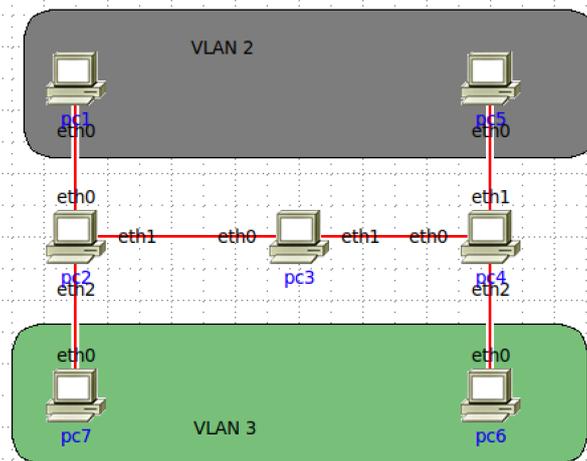


Figura 1 - Topología con PCs en IMUNES

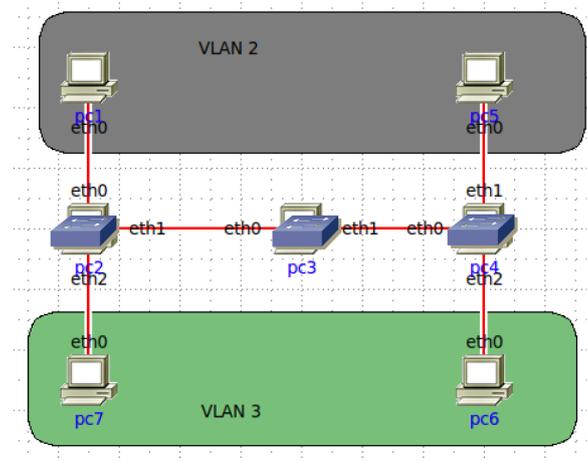


Figura 2 - Topología con PCs en IMUNES (los intermedios actúan como puentes con VLANs)

Lleve a cabo esta configuración (se basa en lo visto en prácticas anteriores).

Las dos subredes se encuentran aisladas. No hay forma de comunicar ahora mismo los hosts de una subred con la otra. Necesitaríamos un router intermedio, que tuviera un interfaz en cada VLAN.

Vamos a implementar este router dándole capacidades capa 3 al pc3 (figura 3). Los interfaces que hemos creado para hacer referencia a los puentes (con el comando brctl) son el equivalente a los interfaces “VlanX” en Cisco IOS. Podemos asignarles dirección IP. Asigne a cada interfaz de bridge de pc3 una dirección IP en la subred correspondiente y configúrelos en los PCs como los siguientes saltos en las rutas por defecto. Pruebe ahora la comunicación entre subredes y vea cómo pc3 actúa para ese tráfico como un router, mientras que para el tráfico interno a cada VLAN y que pase por él sigue actuando en capa 2 (figura 4). En pc3 podrá ver tanto una tabla de rutas IP como las bases de datos de filtrado del puente de cada VLAN.

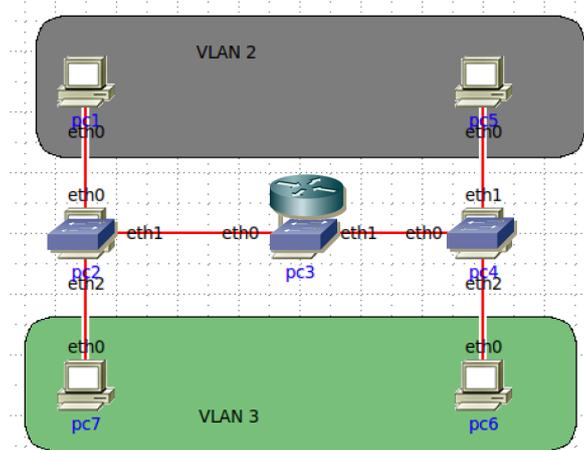


Figura 3 - pc3 tiene también funcionalidad capa 3

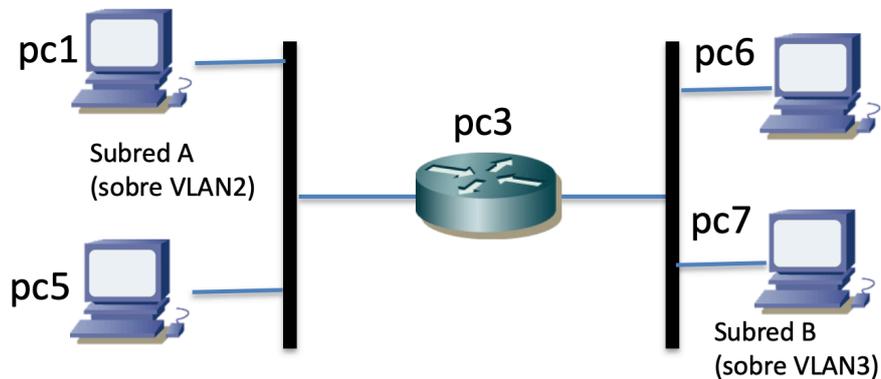


Figura 4 - Topología en capa 3

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

## 5- Router Cisco como puente [OPCIONAL]

### Solo en el laboratorio (funcionalidad no soportada en Packet Tracer)

Cisco IOS permite que un router actúe también como un puente entre varios interfaces. Es decir, se puede hacer que varios interfaces del router actúen como interfaces enrutados con normalidad mientras que otros se unan para formar un puente. Sería equivalente a lo que se representa en la figura 5. En esta figura tenemos un router con 8 interfaces. Entre 3 de ellos está enrutando IP. Otros 3 están siendo puenteados y los dos restantes forman un puente diferente. Ni las tramas ethernet ni los paquetes IP pueden pasar de un dominio a otro. Es decir, un paquete IP en uno de los dominios de broadcast no será reenviado por el otro dominio ni salir por uno de los interfaces enrutados.

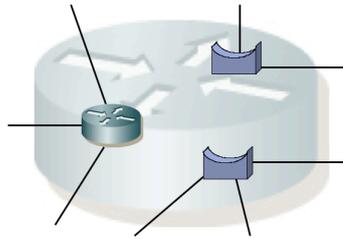


Figura 5 - Router Cisco actuando simultáneamente como puente

Lo que vamos a hacer en este apartado es simplemente configurar los 2 interfaces ethernet de un router (por ejemplo router2) como un puente. Tradicionalmente, Cisco IOS permite que haya unos interfaces enrutados y otros puenteados mediante lo que llama CRB (*Concurrent Routing and Bridging*) o mediante IRB (*Integrated Routing and Bridging*). En los routers ISR4221 del laboratorio se está empleando Cisco IOS XE. Esta versión permite implementar lo descrito mediante las funcionalidades asociadas a EVCs (Ethernet Virtual Connections).

Conecte un PC a cada interfaz Ethernet del router2. Asigne direcciones IP de la misma subred a los interfaces del PC. Verifique que los interfaces del router están activados pero no configure direcciones en ellos. Compruebe que no hay comunicación entre los PCs (por ejemplo no puede hacer ping de uno a otro, el ARP que envía uno no lo recibe el otro, etc).

Vamos a configurar los dos interfaces Ethernet de router2 en el mismo dominio de broadcast o lo que Cisco llama un “*Bridge Domain*”. De esa forma el router estará actuando como un puente (no cambiaremos nada en los PCs). Para ello debe entrar en la configuración de cada interfaz del router y crear un interfaz lógico creando lo que se llama un “Ethernet Flow Point (EFP) service instance”. Este interfaz lógico nos permitirá crear la asociación entre el interfaz físico y el *bridge domain*. Al crear esta *service instance* debe asignarle un número que debe ser único *dentro de ese interfaz*; esto es así porque se pueden crear múltiples *service instances* en el mismo interfaz y este número sirve para diferenciarlas (no necesitan ser diferentes de las creadas en otro interfaz).

Una vez dentro de la configuración de la *service instance* debemos indicarle qué tráfico del interfaz queremos que se asigne a la misma. En este caso asignaremos todo el tráfico que venga sin encapsulado 802.1Q (el tráfico *untagged*). Finalmente, crearemos la asociación entre la *service instance* y un *bridge domain*.

Todo esto se ve en los siguientes comandos:

```
interface GigabitEthernet0/0/0
  service instance 1 ethernet
  encapsulation untagged
  bridge-domain 1
```

Repita estos comandos para el segundo interfaz Ethernet. Es importante que el valor numérico indicado en el comando *bridge-domain* sea el mismo, dado que es lo que hará que ambos interfaces asocien las tramas sin encapsulado 802.1Q al mismo puente. El valor numérico de identificador de la *service instance* podría ser diferente.

Compruebe que ahora sí hay comunicación entre los PCs, siendo puenteadas las tramas que envían.

## **6- Router Cisco como puente con interfaz virtual enrutado [OPCIONAL]**

### **Solo en el laboratorio (funcionalidad no soportada en Packet Tracer)**

IOS permite enrutar paquetes IP entre interfaces puenteados e interfaces enrutados. Para ello crea un interfaz lógico asignado al *bridge domain*. Este interfaz se llama un BDI (*Bridge Domain Interface*) Ese interfaz lógico actúa como un interfaz capa 3 del router, conectado a dicho puente como se representa en la figura 6. El icono de conmutador y de router representan esas funcionalidades de capa 2 y capa 3 que está llevando a cabo de forma simultanea el router. Externamente, para los equipos conectados no hay forma de reconocer que el router no es en realidad dos equipos.

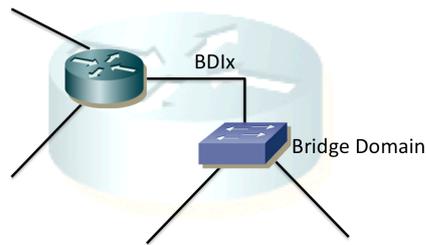


Figura 6.- Router Cisco con 3 interfaces enrutadas, de los cuales uno es un BDI

Puede crear el interfaz lógico mediante:

```
Router(config)# interface bdi 1
```

Dentro del modo configuración de este interfaz virtual podremos asignarle dirección IP teniendo en cuenta que como se representa en la figura 6, ese es el interfaz del router en esa LAN.

Vamos a probar una configuración de este estilo creando la topología de capa 3 que se ve en la figura 7.

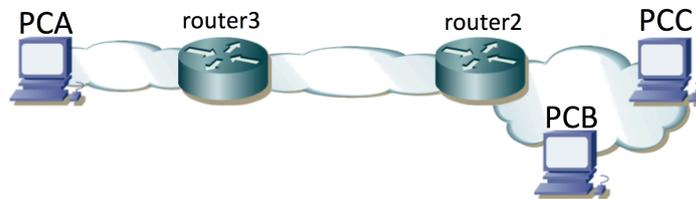


Figura 7 – Topología de red IP objetivo

Como se puede ver, en la figura 7 los equipos router2, PCB y PCC se encuentran en la misma subred IP. Por ello deben tener un interfaz en el mismo dominio de broadcast. Podríamos crear ese dominio empleando un conmutador Ethernet al cual se conectarán los tres equipos pero lo vamos a hacer con la funcionalidad descrita en esta práctica. La topología física a crear es la que se muestra en la figura 8.

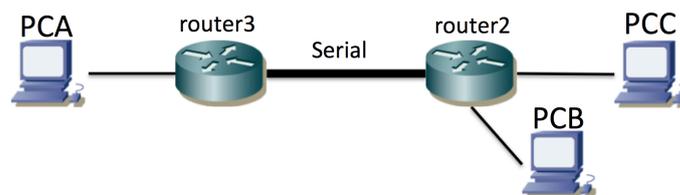


Figura 8 – Topología física

Con este conexionado físico PCB y PCC estarían conectados a diferentes interfaces enrutadas de router2, por lo que no estarían en el mismo dominio de broadcast y no podrían comunicarse con una configuración con direcciones IP en la misma subred (un ARP que enviara uno no llegaría al otro). Debemos convencer a router2 de que puentee esos dos interfaces creando un *bridge domain*. El resultado será algo parecido a lo que se muestra en la figura 9, donde hemos representado de nuevo el *bridge domain* con el icono de un conmutador. Los iconos de conmutador y de router son funciones que lleva a cabo el mismo equipo router2 y que hemos separado para entenderlas mejor. El interfaz BDI1 es el interfaz enrutado que tiene el equipo en el dominio de broadcast al que se asigna el tráfico sin tag de los dos interfaces Ethernet del mismo.

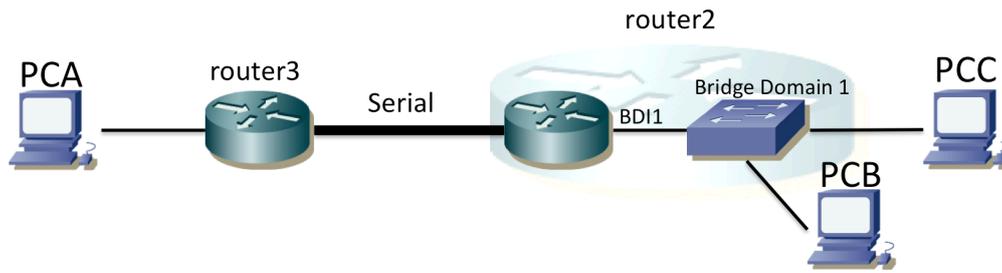


Figura 9 – Topología incluyendo elementos virtuales

Averigüe la dirección MAC que está empleando el router en ese BDI.

Configure las tablas de rutas de PCs y routers para tener conectividad IP entre todos los equipos. Compruebe que la comunicación entre PCB y PCC es puenteada mientras que con PCA es enrutada, siguiendo los saltos que se ven en la topología.