Stp, Vlan, Port bonding

Nazareno Necchi, Agustín luzzini, Lautaro Teta Musa

5to Informática IPS

1. Introduccion teórica

1.1. Switch

Un switch es un dispositivo digital lógico de interconexión de equipos que opera en la capa de enlace de datos del modelo OSI. Su función es interconectar dos o más host de manera similar a los puentes de red, pasando datos de un segmento a otro de acuerdo con la dirección MAC de destino de las tramas en la red y eliminando la conexión una vez finalizada ésta.

1.2. STP

El protocolo permite a los dispositivos de interconexión activar o desactivar automáticamente los enlaces de conexión, de forma que se garantice la eliminación de bucles.
Cuando existen bucles en la topología de red, los dispositivos de interconexión de nivel de enlace de datos reenvían indefinidamente las tramas broadcast y multicast, creando así un bucle infinito que consume tanto el ancho de banda de la red como CPU de los dispositivos de enrutamiento.
STP calcula una única ruta libre de bucles entre los dispositivos de la red pero manteniendo los enlaces redundantes desactivados como reserva.
El protocolo establece identificadores por puente y elige el Root Bridge. Este establecerá el camino de menor coste para todas las redes; cada puerto tiene un parámetro configurable: el Span path cost. Después, entre todos los puentes que conectan un segmento de red, se elige un puente designado, el de menor coste (en el caso que haya el mismo coste en dos puentes, se elige el que tenga el menor identificador "dirección MAC"), para transmitir las tramas hacia la raíz. En este puente designado, el puerto que conecta con el segmento, es el puerto designado y el que ofrece un camino de menor coste hacia la raíz, el puerto raíz. Todos los demás puertos y caminos son bloqueados, esto es en un estado ya estacionario de funcionamiento.

1.3. Vlan

Una VLAN, acrónimo de virtual LAN (red de área local virtual), es un método para crear redes lógicas independientes dentro de una misma red física.
Una VLAN consiste en dos o más redes de computadoras que se comportan como si estuviesen conectados al mismo conmutador, aunque se encuentren físicamente conectados a diferentes segmentos de una red de área local.

1.4. Port bonding

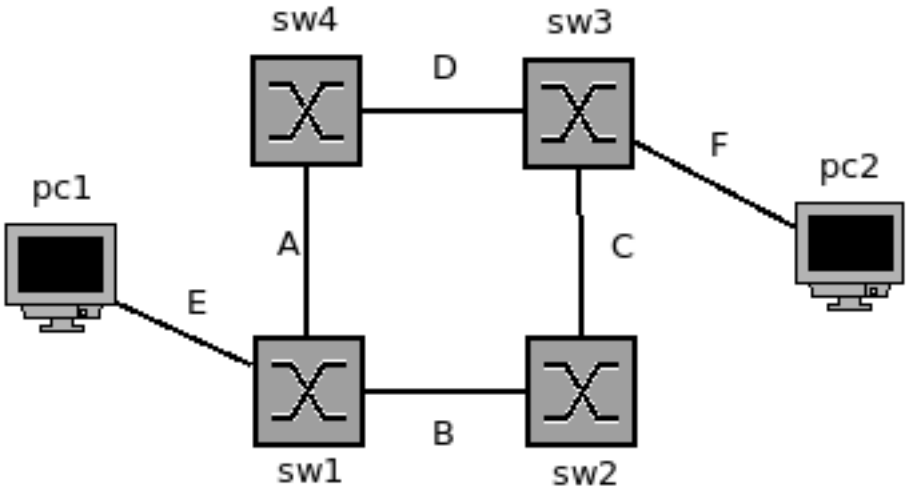
Port bonding resuelve dos problemas con las conexiones Ethernet: limitaciones de ancho de banda y falta de resiliencia.

Con respecto al primer problema: los requisitos de ancho de banda no se escalan linealmente.Históricamente, los anchos de banda de Ethernet han aumentado diez veces cada generación: 10 megabit/s , 100 Mbit/s , 1000 Mbit/s , 10,000 Mbit/s . Si uno comenzara a toparse con los techos de ancho de banda, entonces la única opción era pasar a la siguiente generación, lo que podría tener un costo prohibitivo. Una solución alternativa, introducida por muchos de los fabricantes de redes a principios de la década de 1990, es combinar dos enlaces Ethernet físicos en un enlace lógico a través del enlace de canales. La mayoría de estas soluciones requieren una configuración manual y un equipo idéntico en ambos lados de la agregación.
El segundo problema involucra los tres puntos únicos de falla en una conexión típica puerto-cable-puerto. En la configuración habitual de computadora a conmutador o en una configuración de conmutador a conmutador, el cable o cualquiera de los puertos a los que está conectado puede fallar. Se pueden hacer múltiples conexiones físicas, pero muchos de los protocolos de nivel superior no fueron diseñados para fallar completamente sin problemas.

2. Laboratorios

2.1. STP

2.1.1. Topología



Además vamos a añadir un Sniffer conectado a todos los dominios de colisión para observar el tráfico de la red. Como podemos observar si no se implementa STP se creará un bucle entre los conmutadores sw1, sw2 , sw3 y sw4.

2.1.2. Configuración

Lo primero que vamos a configurar es el lab.conf para establecer las conexiones físicas

```
pc1[0]=E
pc2[0]=F

sw1[0]=A
sw1[1]=B
sw1[2]=E

sw2[0]=B
sw2[1]=C

sw3[0]=C
sw3[1]=F
sw3[2]=D

sw4[0]=D
sw4[1]=A

sniffer[0]=A
sniffer[1]=B
sniffer[2]=C
sniffer[3]=D
```

ahora vamos a adsignarles IP a cada PC

```
PC1$ ifconfig eth0 192.168.0.1/24
PC2$ ifconfig eth0 192.168.0.2/24
```

Por último vamos a configurar los switch procurando activar el STP

```
brctl addbr br0             creamos un bridge
brctl addif br0 eth0        le a adimos la interfaz al bridge
brctl addif br0 eth1
brctl addif br0 eth2

ifconfig eth0 up            activamos la interfaz
ifconfig eth1 up
ifconfig eth2 up
brctl stp br0 on            activamos el protocolo STP en br0
ifconfig br0 up             activamos el bridge
```

Configuramos los demás switch de forma similiar.

2.1.3. Testeo

Para comprobar que el protocolo haya funcionado correctamente debemos verificar que alguna conexión se marque como blocking. Para esto utilizaremos el comando brctl showstp br0 que nos permitirá saber el estado de cada puerto.Como en nuestro caso todos los caminos tienen el mismo coste y la prioridad de los switch es la misma, el root switch y el puerto que se bloquea serán elegidos aleatoriamente

ejecutando el comando mencionado deberíamos obtener una salida como esta:

```
sw4 login: root (automatic login)
Last login: Sat May 18 16:22:45 UTC 2019 on tty1
sw4:~# brctl showstp br0
br0
 bridge id              8000.56fee46f0f8b
 designated root        8000.0e7ff33c14fb
 root port                 1                    path cost              100
 max age                  20.00                 bridge max age          20.00
 hello time                2.00                 bridge hello time        2.00
 forward delay            15.00                 bridge forward delay    15.00
 ageing time             300.00
 hello timer               0.00                 tcn timer                0.00
 topology change timer     0.00                 gc timer                 8.44
 flags


eth0 (1)
 port id                8001                   state                forwarding
 designated root        8000.0e7ff33c14fb      path cost              100
 designated bridge      8000.0e7ff33c14fb      message age timer       18.90
 designated port        8003                   forward delay timer      0.00
 designated cost           0                   hold timer               0.00
 flags

eth1 (2)
 port id                8002                   state                forwarding
 designated root        8000.0e7ff33c14fb      path cost              100
 designated bridge      8000.56fee46f0f8b      message age timer        0.00
 designated port        8002                   forward delay timer      0.00
 designated cost         100                   hold timer               0.44
 flags
```

Observamos que el sw3 ha sido establecido como root ya que su bridge-id y el designated-root son iguales

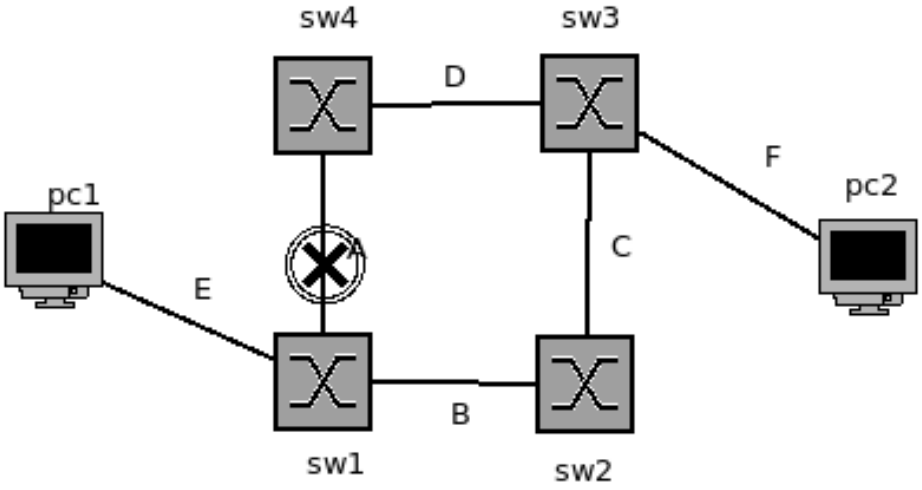
```
sw3 login: root (automatic login)
Last login: Sat May 18 16:22:38 UTC 2019 on tty1
sw3:~# brctl showstp br0
br0
 bridge id              8000.0e7ff33c14fb
 designated root        8000.0e7ff33c14fb
 root port                 0                    path cost                0
 max age                  20.00                 bridge max age          20.00
 hello time                2.00                 bridge hello time        2.00
 forward delay            15.00                 bridge forward delay    15.00
 ageing time             300.00
 hello timer               1.51                 tcn timer                0.00
 topology change timer     0.00                 gc timer                 9.51
 flags
```

La interfaz eth1 del sw1 ha sido bloqueada por el STP como podemos ver en la salida del showstp.

SW1$:

```
eth0 (1)
 port id               8001                    state                blocking
 designated root       8000.0e7ff33c14fb       path cost              100
 designated bridge     8000.56fee46f0f8b       message age timer      18.53
 designated port       8002                    forward delay timer     0.00
 designated cost        100                    hold timer              0.00
 flags
```

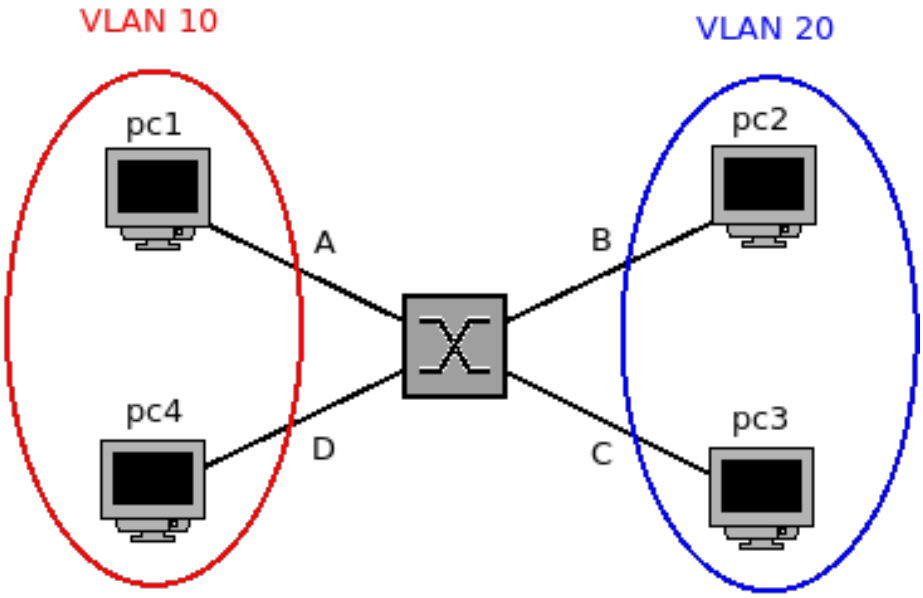
En este caso la topología de la red luego de implementar el STP quedaría de la siguiente manera:



2.2. Vlan

2.2.1. Topología

Vamos a realizar un laboratorio con la siguiente topología



2.2.2. Configuración

configuramos el lab.conf de la siguiente manera:

```
sw[0]=A
sw[1]=B
sw[2]=C
sw[3]=D

pc1[0]=A
pc2[0]=B
pc3[0]=C
pc4[0]=D
```

Luego configuramos el switch activando todas sus interfaces y creando un bridge.

```
ifconfig eth0 up             activamos la interfaz
ifconfig eth1 up
ifconfig eth2 up
ifconfig eth3 up

brctl addbr br0             creamos un bridge
brctl addif br0 eth0         le a adimos la interfaz al bridge
brctl addif br0 eth1
brctl addif br0 eth2
brctl addif br0 eth3
ifconfig br0 up             activamos el bridge
```

Ahora los que nos queda es configurar las PC y sus puertos para que pertenezcan a las Vlan determinadas. Para ello utilizaremos el comando vconfig add que nos permite crear una sub interfaz de red anexada a una Vlan por un ID.

Aplicaremos la siguiente configuración en los .startup de cada PC cambiando la V por el número de la Vlan a la que deseamos que pertenezca y la X por 1, 2, 3 o 4 según corresponda.

```
ifconfig eth0 up
vconfig add eth0 V
ifconfig eth0.V 192.168.0.X up
```

Una vez hecho esto ya estarán configuradas nuestra red Vlan.

2.2.3. Testeo

Para testear que las Vlans estén correctamente configuradas debemos asegurarnos que solo se puedan trasmitir paquetes entre PCs que pertenezcan a la misma Vlan. Para esto vamos a realizar un ping entre PC1 y PC2 y luego entre PC1 y PC4. Si todo funciona adecuadamente deberíamos poder realizar el primer ping normalmente pero el segundo debería dar error.

```
PC1$ ping 192.168.0.2
```

```
pc1 login: root (automatic login)
Last login: Sun May 19 19:51:36 UTC 2019 on tty0
pc1:~# ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=7.07 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.492 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.414 ms
64 bytes from 192.168.0.2: icmp_seq=4 ttl=64 time=0.535 ms
^C
--- 192.168.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 0.414/2.128/7.074/2.856 ms
pc1:~#
```

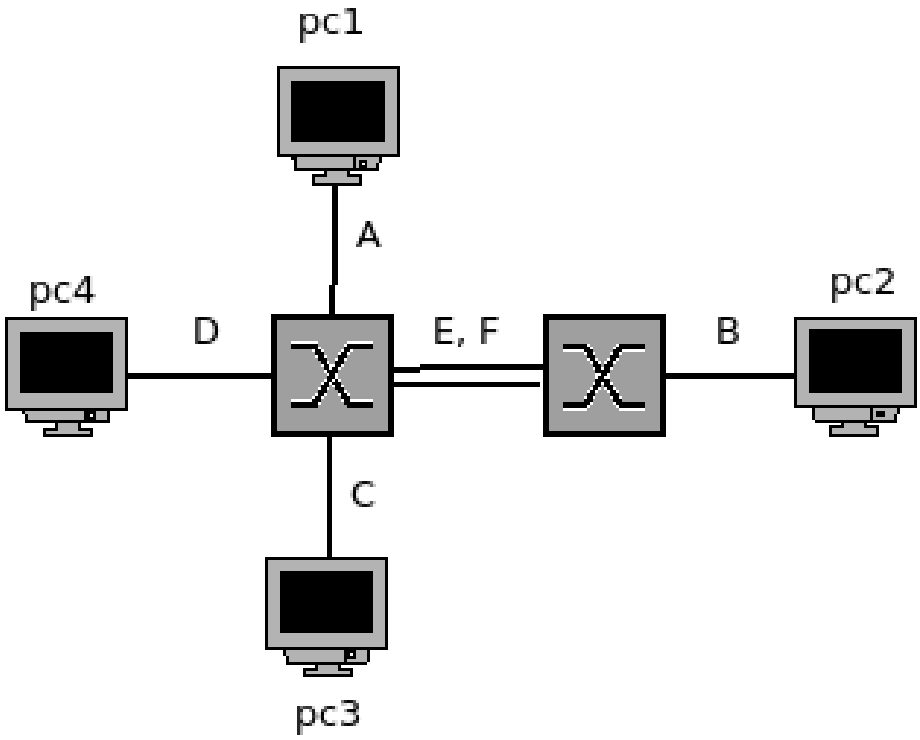
```
PC1$ ping 192.168.0.4
```

```
pc1:~# ping 192.168.0.4
PING 192.168.0.4 (192.168.0.4) 56(84) bytes of data.
From 192.168.0.1 icmp_seq=1 Destination Host Unreachable
From 192.168.0.1 icmp_seq=2 Destination Host Unreachable
From 192.168.0.1 icmp_seq=3 Destination Host Unreachable
^C
--- 192.168.0.4 ping statistics ---
5 packets transmitted, 0 received, +3 errors, 100% packet loss, time 4021ms
, pipe 3
pc1:~#
```

Como vemos PC1 puede comunicarse con PC2 pero no con PC4, ya que se encuentra en una Vlan diferente. Esto significa que hemos configurado Vlan correctamente.

2.3. Port bonding

2.3.1. Topología



2.3.2. Configuración

Lo primero que vamos a hacer a establecer todas las conexiones editando lab.conf:

```
pc1[0]=A
pc2[0]=B
pc3[0]=C
pc4[0]=D

sw1[0]=A
sw1[1]=C
sw1[2]=D
sw1[3]=E
sw1[4]=F

sw2[0]=B
sw2[1]=E
sw2[2]=F
```

Para el sw1 vamos a realizar la siguiente configuración

```
modprobe bonding
brctl addbr br0 up       a adimos un bridge

echo balance-rr > /sys/class/net/bond0/bonding/mode
#seleccionamos el modo del port bonding
echo +eth3 > /sys/class/net/bond0/bonding/slaves
#asignamos eth3 como parte del bond
echo +eth4 > /sys/class/net/bond0/bonding/slaves
#asignamos eth4 como parte del bond

brctl addif br0 bond0        #a adimos bond0 al bridge
brctl addif br0 eth0         #       "       "
brctl addif br0 eth1         #       "       "
brctl addif br0 eth2         #       "       "

ifconfig eth0 up             #actvamos eth0
ifconfig eth1 up             #   "     eth1
ifconfig eth2 up             #   "     eth2
ifconfig eth3 up             #   "     eth3
ifconfig eth4 up             #   "     eth4
ifconfig br0 up              #   "     br0
ifconfig bond0 up            #   "     bond0
```

El sw2 se configura de forma similar pero con menos interfaces:

```
modprobe bonding
brctl addbr br0

echo balance-rr > /sys/class/net/bond0/bonding/mode
echo +eth1 > /sys/class/net/bond0/bonding/slaves
echo +eth2 > /sys/class/net/bond0/bonding/slaves

brctl addif br0 bond0
brctl addif br0 eth0

ifconfig eth0 up
ifconfig eth1 up
ifconfig eth2 up
ifconfig br0 up
ifconfig bond0 up
```

Por último simplemente le asignamos IP a las distintas pcs

```
ifconfig eth0 up
ifconfig eth0.V 192.168.0.X up
```