

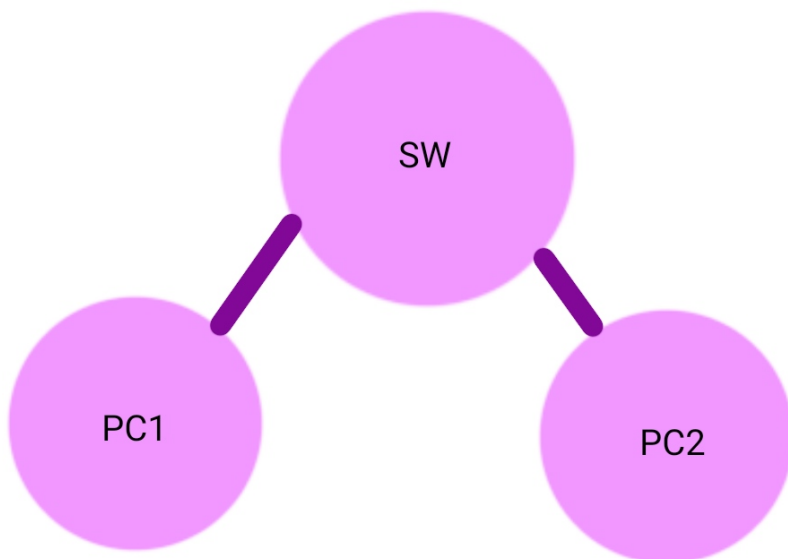
# TRABAJO PRACTICO

Maite Lukasiunas, Celeste Papalato, Marco Martin

May 2019

# 1 SWITCH

Un SWITCH es un equipo de interconexión utilizado para conectar equipos de red, formando las redes de área local LAN. El SWITCH es el elemento central de la topología en la red.



*Topología:*  
PC1[0]="A"  
PC2[0]="B"  
SWITCH[0]="A"  
SWITCH[1]="B"

## En PC1:

Activamos la placa de red:  
ifconfig eth0 192.168.0.1 up

## En PC2:

Activamos la placa de red:  
ifconfig eth0 192.168.0.2 up

## En SWITCH:

Activamos las placas de red:

```
ifconfig eth0 up  
ifconfig eth1 up
```

Creamos un puente entre las placas de ethernet:

```
brctl add br0
```

Añadimos la primer placa:

```
brctl addif br0 eth0
```

Añadimos la segunda placa:

```
brctl addif br0 eth1
```

Activamos el puente:

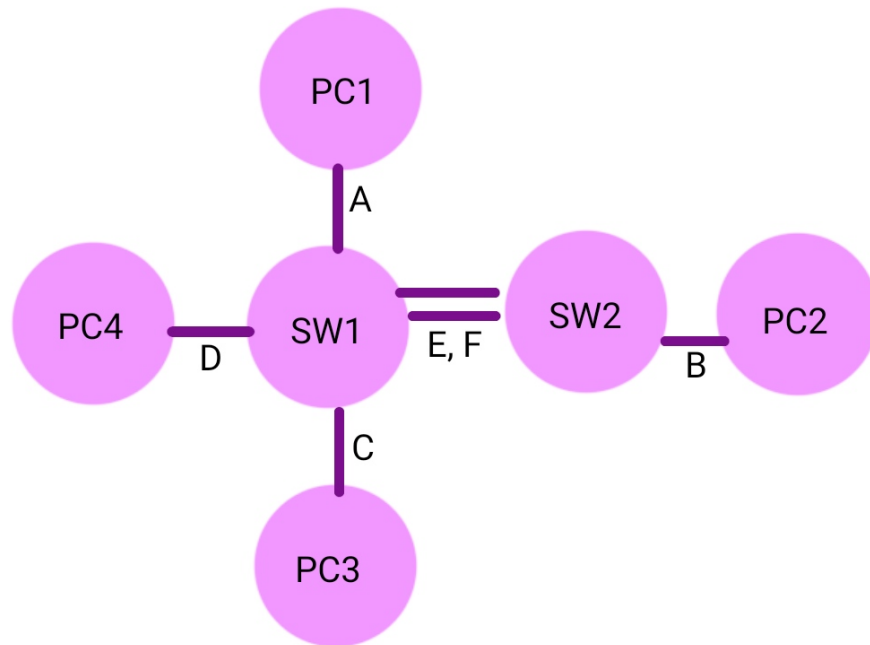
```
ifconfig br0 up
```

Ahora el switch podrá enviar los paquetes de las 2 PC debido a que conectamos las placas de ethernet con un *bridge*.

## 2 PORT BONDING

La función del Port Bonding es resolver los problemas en las limitaciones de ancho de banda y la falta de resiliencia que surgen en las conexiones de Ethernet.

*Topología:*



*Configuración:*

Primero establecemos las conexiones

```

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

```

modprobe bonding
brctl addbr br0 up

```

```
echo balance-rr i/sys/class/net/bond0/bonding/mode (ahora elegimos el modo
del portbonding)
echo +eth3 i/sys/class/net/bond0/bonding/slaves (ahora eth3 es parte del bond)
echo +eth4 i/sys/class/net/bond0/bonding/slaves (ahora eth4 es parte del bond)
brctl addif br0 bond0
```

```
brctl addif br0 eth0
brctl addif br0 eth1
brctl addif br0 eth2
```

```
ifconfig eth0 up
ifconfig eth1 up
ifconfig eth2 up
ifconfig eth3 up
ifconfig eth4 up
ifconfig br0 up
ifconfig bond0 up
```

Hacemos algo muy parecido para el segundo SW

```
modprobe bonding
brctl addbr br0
```

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

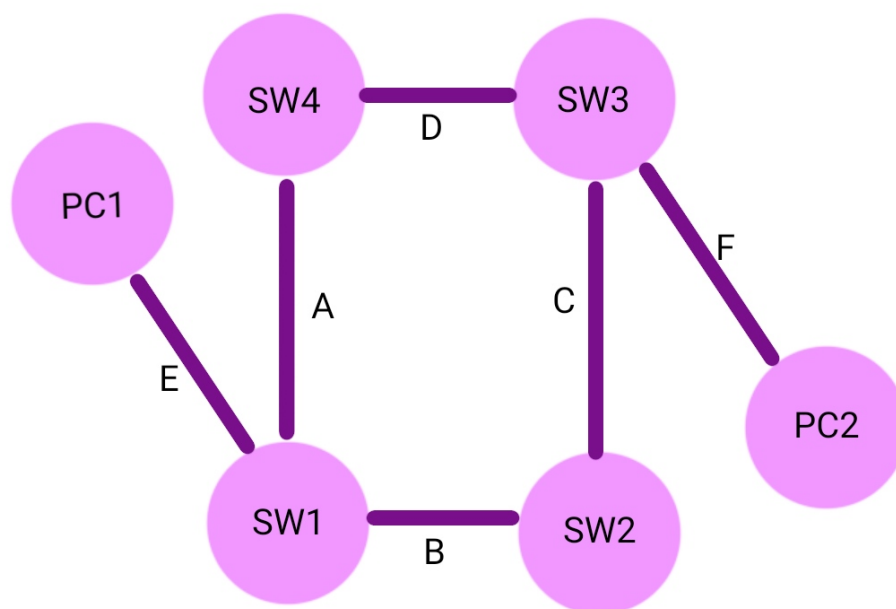
Le asignamos IP a las distintas PCS

```
ifconfig eth0 up
ifconfig eth0.10 192.168.0.1 up
```

### 3 STP

Spanning Tree Protocol es un protocolo encargado de evitar los bucles en topologías de red, para esto se cambia la topología de forma lógica y se mantiene a las estaciones de trabajo informadas de la nueva estructura.

*Topología:*



*Configuración:*

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  
de esta forma logramos las conexiones físicas

PC1 ifconfig eth0 192.168.0.1/24  
PC2 ifconfig eth0 192.168.0.2/24  
aquí se le asigna un IP a cada PC

```
brctl addbr br0
brctl addbr br0 eth0
brctl addbr br0 eth1
brctl addbr br0 eth2
```

```
ifconfig eth0 up
ifconfig eth1 up
ifconfig eth2 up
brctl stp br0 on
ifconfig br0 up
```

por ultimo así es cómo configuramos los switch(y luego lo repetimos con cada uno)

*Comprobación:*

Mediante el comando `brctl showstp br0` vamos a conocer el estado de cada puerto y de esta forma comprobar que el STP funcionó en el caso de que alguna conexión se marque como bloqueada.

Observando el SW1:

```

SW1 login: root (automatic login)
SW1:~# brctl showstp br0
br0
bridge id            8000.3a3700de0b66
designated root       8000.3a3700de0b66
root port            0
max age              20.00
hello time           2.00
forward delay        15.00
ageing time          300.00
hello timer          1.56
topology change timer 0.00
path cost            0
bridge max age       20.00
bridge hello time    2.00
bridge forward delay 15.00
tcn timer            0.00
gc timer             14.56

eth0 (1)
port id              8001
designated root       8000.3a3700de0b66
designated bridge     8000.3a3700de0b66
designated port       8001
designated cost       0
state                forwarding
path cost            100
message age timer    0.00
forward delay timer  0.00
hold timer           0.56

eth1 (2)
port id              8002
designated root       8000.3a3700de0b66
designated bridge     8000.3a3700de0b66
designated port       8002
designated cost       0
state                forwarding
path cost            100
message age timer    0.00
forward delay timer  0.00
hold timer           0.56

eth2 (3)
port id              8003
designated root       8000.3a3700de0b66
designated bridge     8000.3a3700de0b66
designated port       8003
designated cost       0
state                forwarding
path cost            100
message age timer    0.00
forward delay timer  0.00
hold timer           0.56

```

Observando el SW2:



```

SW2 login: root (automatic login)
SW2:~# brctl showstp br0
br0
bridge id            8000.e286ed457194
designated root       8000.3a3700de0b66
root port            1
max age               20.00
hello time            2.00
forward delay         15.00
ageing time           300.00
hello timer           0.00
topology change timer 0.00
flags
path cost            100
bridge max age       20.00
bridge hello time    2.00
bridge forward delay 15.00
tcn timer            0.00
gc timer             5.35

eth0 (1)
port id              8001
designated root       8000.3a3700de0b66
designated bridge     8000.3a3700de0b66
designated port       8002
designated cost        0
flags
state                forwarding
path cost            100
message age timer    19.55
forward delay timer  0.00
hold timer           0.00

eth1 (2)
port id              8002
designated root       8000.3a3700de0b66
designated bridge     8000.e286ed457194
designated port       8002
designated cost       100
flags
state                forwarding
path cost            100
message age timer    0.00
forward delay timer  0.00
hold timer           0.35

```

Observando el SW3:

```

SW3 login; root (automatic login)
SW3:~# brctl showstp br0
br0
bridge id          8000,5eed01fe1f3d
designated root     8000,3a3700de0b66
root port          3
max age            20,00
hello time         2,00
forward delay      15,00
ageing time        300,00
hello timer        0,00
topology change timer 0,00
flags
path cost          200
bridge max age     20,00
bridge hello time  2,00
bridge forward delay 15,00
tcn timer          0,00
gc timer           9,44

eth0 (1)
port id            8001
designated root     8000,3a3700de0b66
designated bridge   8000,e286ed457194
designated port     8002
designated cost     100
flags
state              blocking
path cost          100
message age timer  18,02
forward delay timer 0,00
hold timer         0,00

eth1 (2)
port id            8002
designated root     8000,3a3700de0b66
designated bridge   8000,5eed01fe1f3d
designated port     8002
designated cost     200
flags
state              forwarding
path cost          100
message age timer  0,00
forward delay timer 0,00
hold timer         0,00

eth2 (3)
port id            8003
designated root     8000,3a3700de0b66
designated bridge   8000,c2e76d23ada1
designated port     8001
designated cost     100
flags
state              forwarding
path cost          100
message age timer  18,02
forward delay timer 0,00
hold timer         0,00

```

Observando el SW4:

```

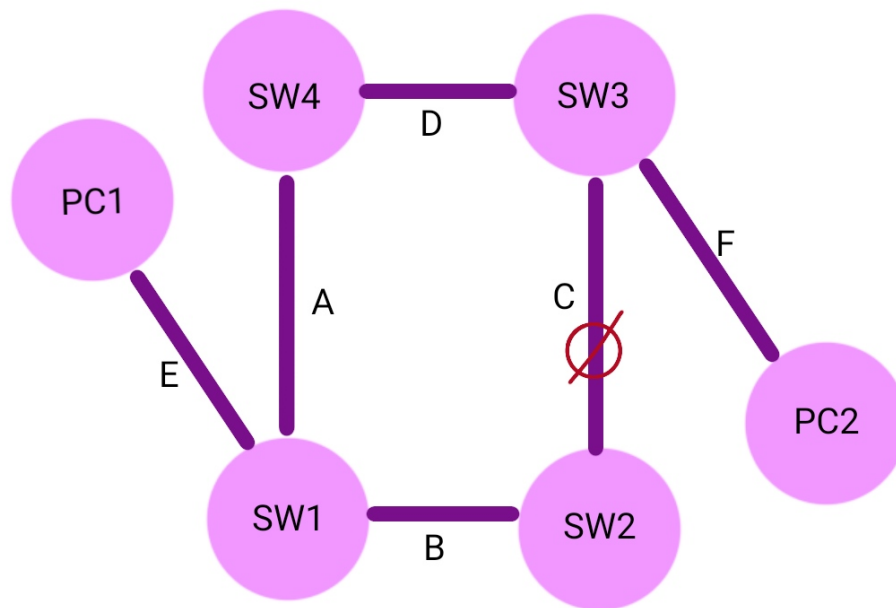
SW4 login: root (automatic login)
SW4:~# brctl showstp br0
br0
bridge id            8000,c2e76d23ada1
designated root       8000,3a3700de0b66
root port            2
max age               20,00
hello time            2,00
forward delay         15,00
ageing time           300,00
hello timer           0,00
topology change timer 0,00
path cost             100
bridge max age        20,00
bridge hello time     2,00
bridge forward delay  15,00
tcn timer             0,00
gc timer              12,44
flags

eth0 (1)
port id              8001
designated root       8000,3a3700de0b66
designated bridge     8000,c2e76d23ada1
designated port       8001
designated cost        100
state                forwarding
path cost             100
message age timer     0,00
forward delay timer   0,00
hold timer            0,44
flags

eth1 (2)
port id              8002
designated root       8000,3a3700de0b66
designated bridge     8000,3a3700de0b66
designated port       8001
designated cost        0
state                forwarding
path cost             100
message age timer     18,71
forward delay timer   0,00
hold timer            0,00
flags

```

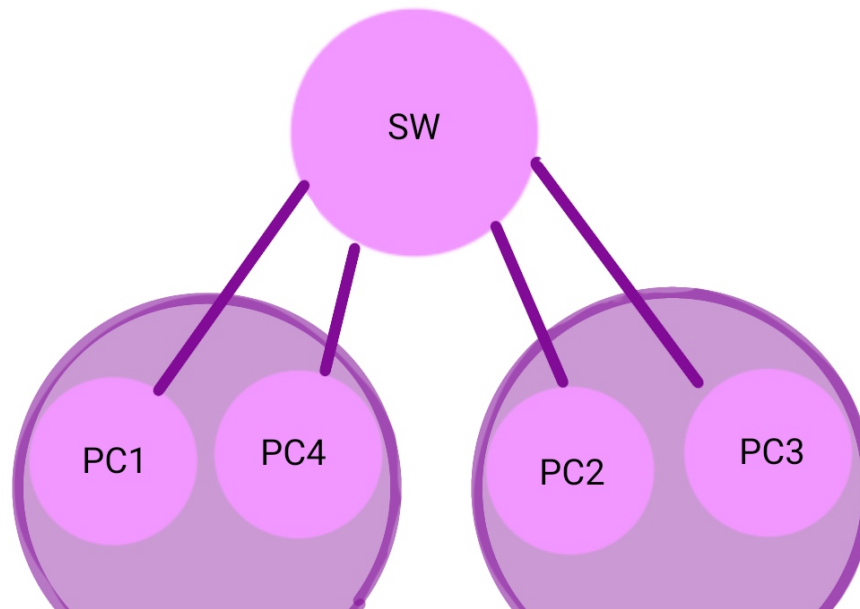
se ve de la siguiente forma:



## 4 VLAN

VLAN es un método para crear redes lógicas independientes de una red física. Puede haber varias VLAN coexistiendo en un mismo computador físico y/o red física. Ayuda a disminuir la difusión y a administrar una red en segmentos lógicos.

Una VLAN consiste en dos o más redes de computadores que se comportan como si estuviesen conectados a un mismo conmutador aunque se encuentren conectadas en la red física.



Topología:  
PC1[0]="A"  
PC2[0]="B"  
PC3[0]="C"  
PC4[0]="D"  
SW1[0]="A"  
SW1[1]="B"  
SW1[2]="C"  
SW1[3]="D"

Pondremos a PC2 y PC3 en la VLAN 100 y a PC1 y PC4 en la VLAN 200.  
Usaremos un SWITCH para la comunicación.

**En PC1:**

```
ifconfig eth0 192.168.0.1 up
```

```
//Añadiremos una VLAN al PC  
vconfig add eth0 200
```

Añadiremos una nueva IP al PC en base a esa VLAN  

```
ifconfig eth0.200 200.0.0.1 up
```

**En PC2:**

```
ifconfig eth0 192.168.0.2 up
```

Añadiremos una VLAN al PC

```
vconfig add eth0 100
```

Añadiremos una nueva IP al PC en base a esa VLAN

```
ifconfig eth0.200 100.0.0.1 up
```

**En PC3:**

```
ifconfig eth0 192.168.0.3 up
```

Añadiremos una VLAN al PC

```
vconfig add eth0 100
```

Añadiremos una nueva IP al PC en base a esa VLAN

```
ifconfig eth0.200 100.0.0.2 up
```

**En PC4:**

```
ifconfig eth0 192.168.0.4 up
```

Añadiremos una VLAN al PC

```
vconfig add eth0 200
```

Añadiremos una nueva IP al PC en base a esa VLAN

```
ifconfig eth0.200 200.0.0.2 up
```

**En SW1:**

```
ifconfig eth0 up
```

```
ifconfig eth1 up
```

```
ifconfig eth2 up
```

```
ifconfig eth3 up
```

```
brctl addbr br0
```

```
brctl addif br0 eth0
```

```
brctl addif br0 eth1
```

```
brctl addif br0 eth2
```

```
brctl addif br0 eth3
```

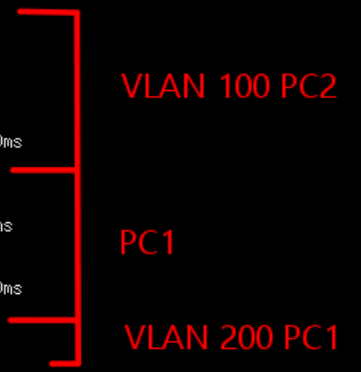
```
ifconfig br0 up
```

Usaremos a PC3 para el ejemplo:

```
PC3 login: root (automatic login)
PC3:~# ping -c1 100.0.0.1
PING 100.0.0.1 (100.0.0.1) 56(84) bytes of data.
64 bytes from 100.0.0.1: icmp_seq=1 ttl=64 time=5.61 ms

--- 100.0.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.614/5.614/5.614/0.000 ms
PC3:~# ping -c1 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.841 ms

--- 192.168.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.841/0.841/0.841/0.000 ms
PC3:~# ping -c1 200.0.0.1
connect: Network is unreachable
PC3:~#
```



A diagram on the right side of the terminal output shows a red bracket grouping three labels: 'VLAN 100 PC2', 'PC1', and 'VLAN 200 PC1'. A red line connects this bracket to the PC3 terminal output, indicating that PC3 is connected to all three VLANs.

Observaremos que PC3 puede comunicarse con PC2, quien se encuentra en su VLAN y no con la VLAN de PC1 y PC4. PC3 aún puede comunicarse con PC1 usando su IP original dado que estamos usando un SWITCH para la comunicación, permitiendo la comunicación a través de la red física.

#### FUENTES:

- [https://es.wikipedia.org/wiki/Spanning\\_tree](https://es.wikipedia.org/wiki/Spanning_tree)
- [http://www.netkit.org/netkit-labs/netkit-labs\\_advanced-topics/netkit-labs\\_spanning-tree/netkit-lab\\_spanning-tree.pdf](http://www.netkit.org/netkit-labs/netkit-labs_advanced-topics/netkit-labs_spanning-tree/netkit-lab_spanning-tree.pdf)
- [http://wiki.netkit.org/index.php/Labs\\_oficial](http://wiki.netkit.org/index.php/Labs_oficial)
- <https://rm-rf.es/configurar-una-vlan-en-linux-con-vconfig/>
- <https://es.wikipedia.org/wiki/VLAN>
- [https://es.wikipedia.org/wiki/Spanning\\_tree](https://es.wikipedia.org/wiki/Spanning_tree)
- [http://www.netkit.org/netkit-labs/netkit-labs\\_advanced-topics/netkit-labs\\_spanning-tree/netkit-lab\\_spanning-tree.pdf](http://www.netkit.org/netkit-labs/netkit-labs_advanced-topics/netkit-labs_spanning-tree/netkit-lab_spanning-tree.pdf)
- [http://wiki.netkit.org/index.php/Labs\\_oficial](http://wiki.netkit.org/index.php/Labs_oficial)