

Ethernet. Bucles LAN

Álvaro González Sotillo

4 de marzo de 2021

Índice

| | |
|--|----|
| 1. Introducción | 1 |
| 2. Bucles LAN | 1 |
| 3. 802.1D: Spanning Tree | 2 |
| 4. IEEE 802.3ad: Link Aggregation | 13 |
| 5. Redundancia a nivel de red | 16 |
| 6. Referencias | 17 |

1. Introducción

Bucles lan

- Generalmente, una LAN se configura de forma jerárquica
 - Óptimo según el cableado estructurado
 - Fácil de entender
- A veces conviene una topología en forma de malla
 - Mayor tolerancia a fallos
 - A veces, mayor velocidad entre switches

2. Bucles LAN

- Problema
 - Una trama podría estar *viajando* por la red por siempre (**Switching loop**)
- Solución 1: **STP**
 - Ciertos enlaces de la red se deshabilitan, para volver a una topología jerárquica
 - Estos enlaces se habilitan de nuevo cuando se modifican las condiciones

-
- Solución 2: **Link Aggregation**
 - Se modifica el comportamiento de los switches
 - Utilizan varias conexiones como una sola, con más velocidad

3. 802.1D: **Spanning Tree**

- Es un protocolo de red para conseguir un árbol (red jerárquica) a base de deshabilitar enlaces de una malla
 - El administrador puede definir una **raíz** (centro de estrella)
 - Los switches envían paquetes entre sí para calcular sus distancias a esa raíz
 - El puerto más **cercano** de cada switch se convierte en el puerto raíz

3.1. Nomenclatura

- **Root Port (RP)**: es el puerto por el que se llega más “barato” a la raíz
 - Coste del enlace: un enlace lento es más costoso. Esta es la medida para decidir entre enlaces.
 $\text{Coste} = 10000/\text{Mbs}$
 - nos quedamos con el más “barato”, que es el más rápido
- **Designated Port (DP)**: es el puerto conectado al RP de algún otro switch
- **Bumped Port (BP)**: Un puerto enchufado a otro switch, que no es RP ni DP

3.2. Costes de un enlace

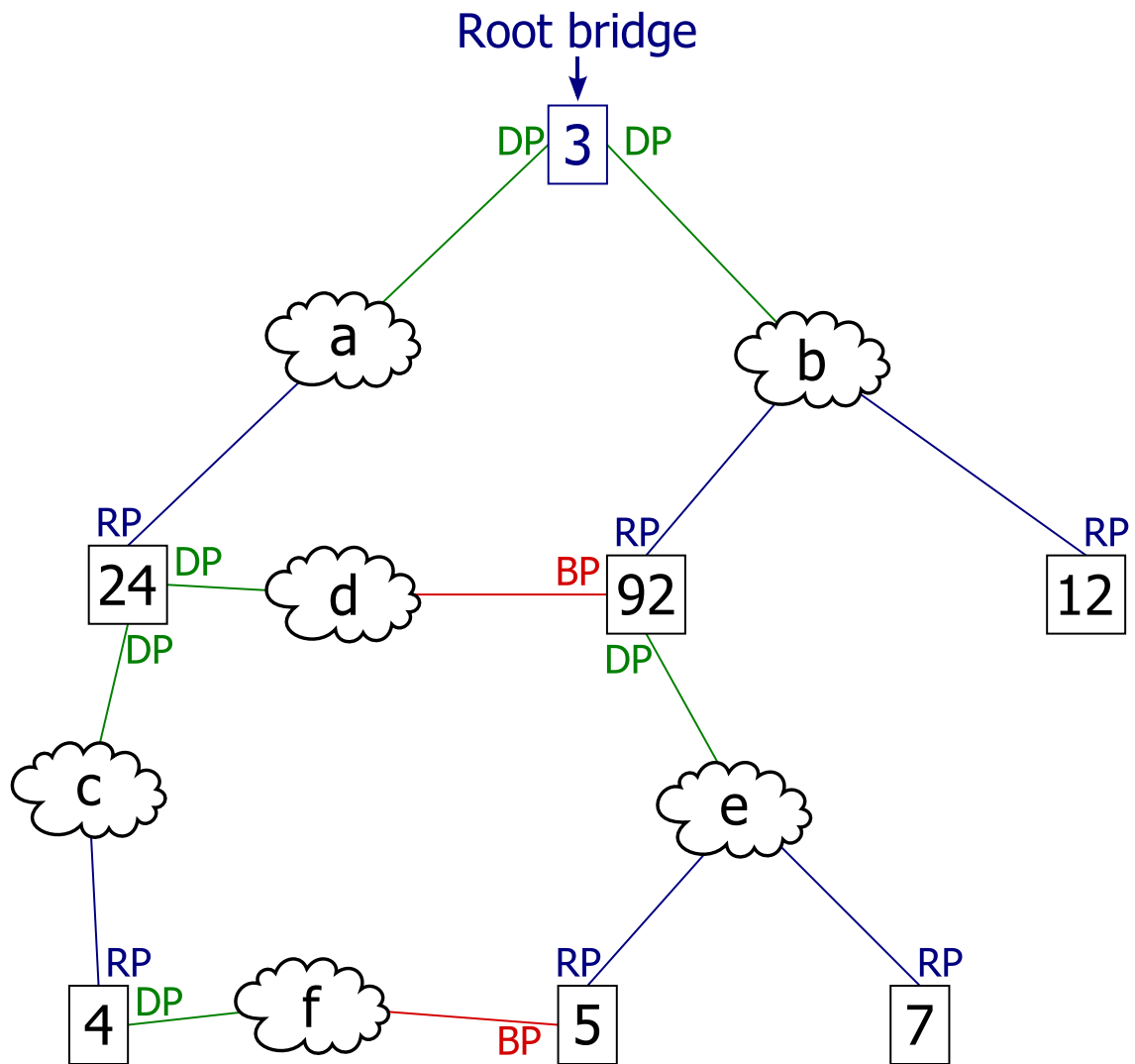
- Definidos por un estándar
- Nosotros los aproximaremos por $10000/\text{velocidad}$

| Data rate | STP Cost (802.1D-1998) | RSTP Cost (802.1W-2001) |
|------------|------------------------|-------------------------|
| 4 Mbit/s | 250 | 5000000 |
| 10 Mbit/s | 100 | 2000000 |
| 16 Mbit/s | 62 | 1250000 |
| 100 Mbit/s | 19 | 200000 |
| 1 Gbit/s | 4 | 20000 |
| 2 Gbit/s | 3 | 10000 |
| 10 Gbit/s | 2 | 2000 |

3.3. Bridge Protocol Data Units

- Cada switch viene de fábrica con un ID para ser root
 - Aunque el administrador lo puede cambiar
- Se envían tramas BPDU para comunicarse
 - Qué switch tiene el menor ID
 - Qué costes tiene cada uno de sus puertos para llegar a la raíz
- Generalmente, se envían cada 2 segundos

3.4. Ejemplo Wikipedia



3.5. Ejemplo 2

3.6. Ejemplo 3

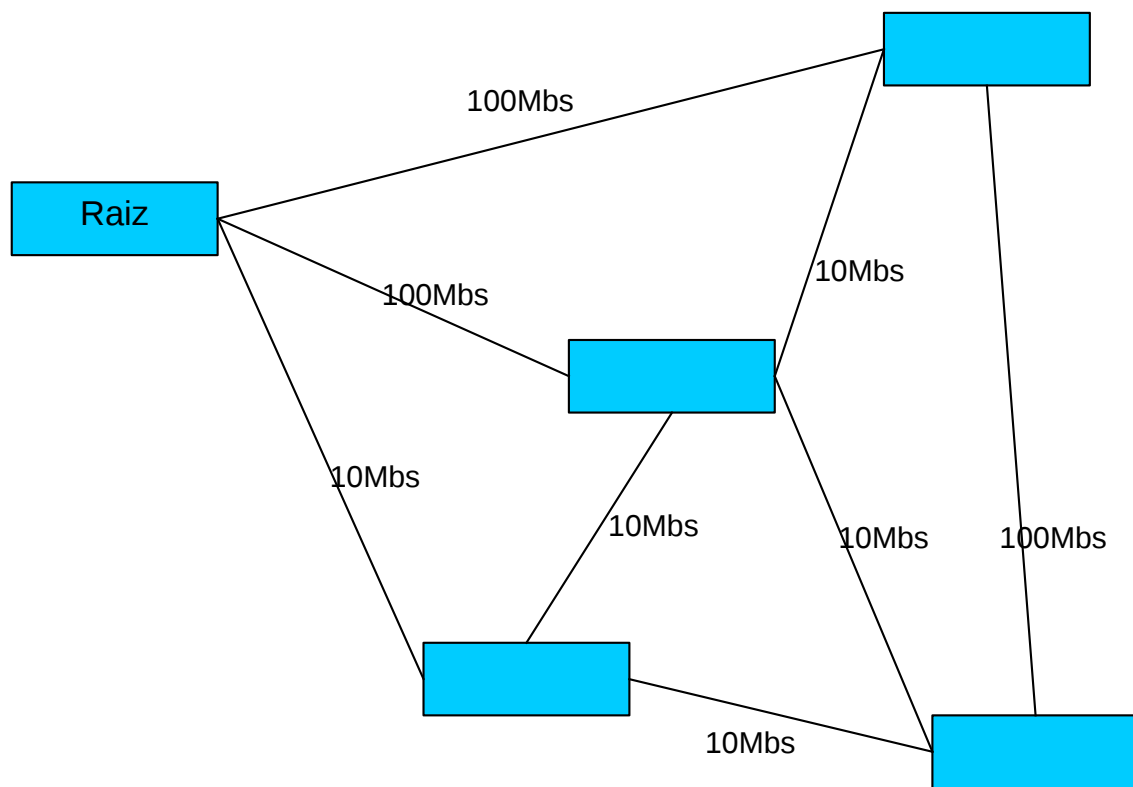


Figura 1: Red inicial

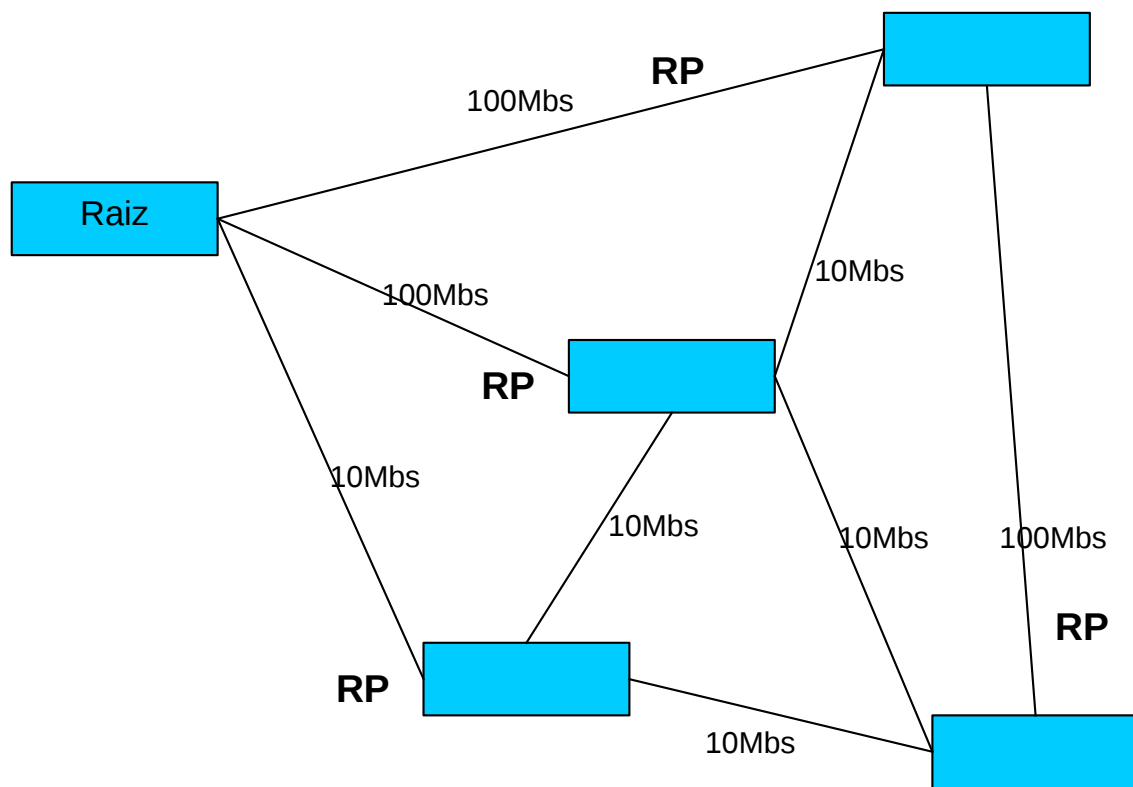


Figura 2: Los RP son los puertos más cercanos a la raíz

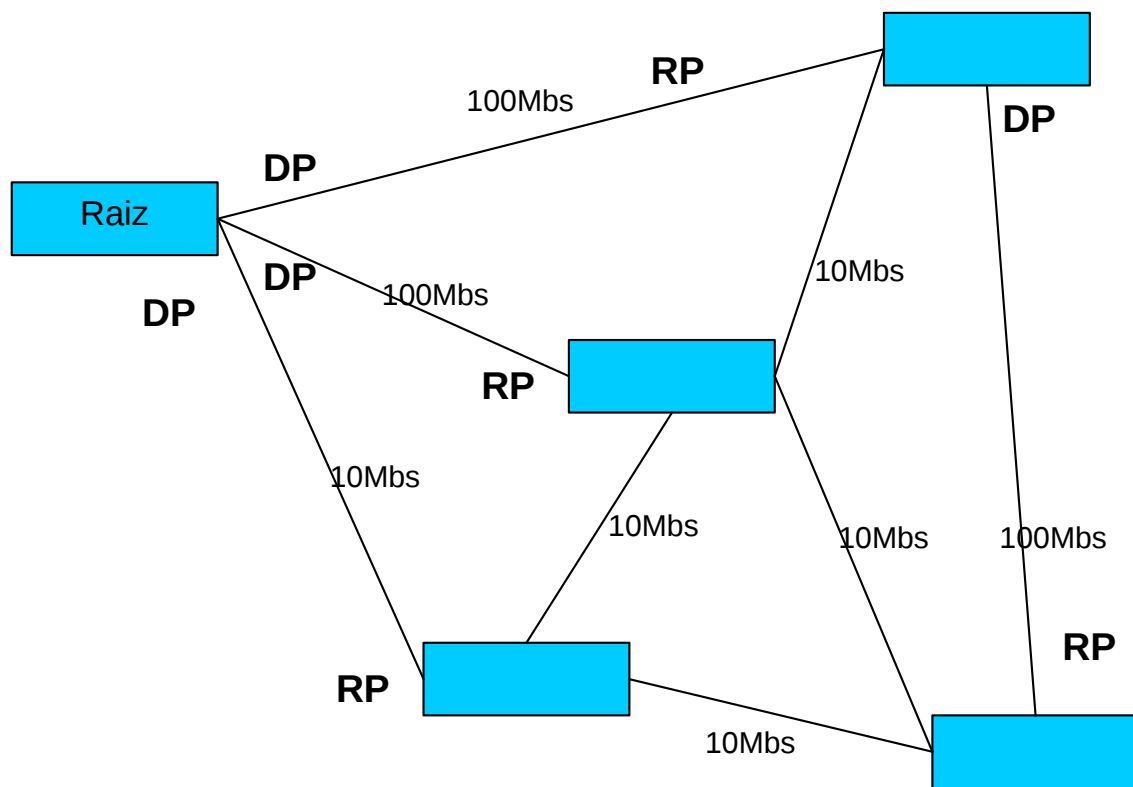


Figura 3: Los DP son los que se conectan a algún RP

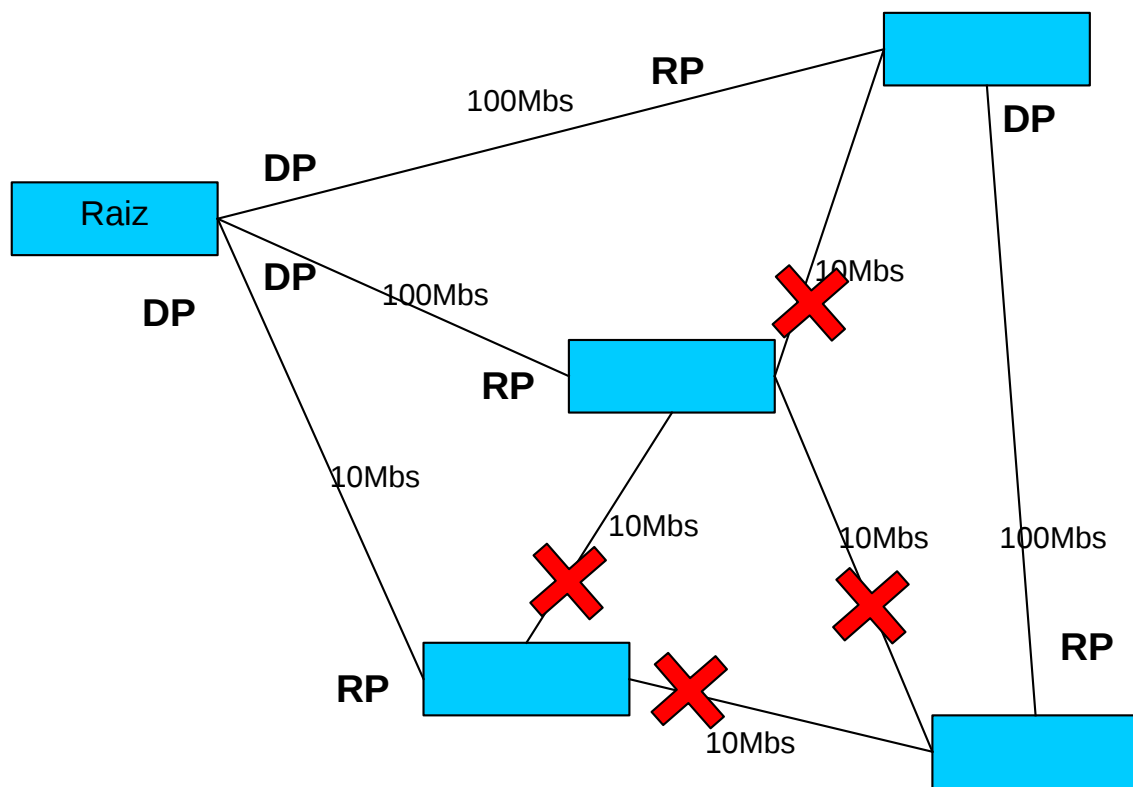


Figura 4: Los BP son los puertos que no son ni RP ni DP

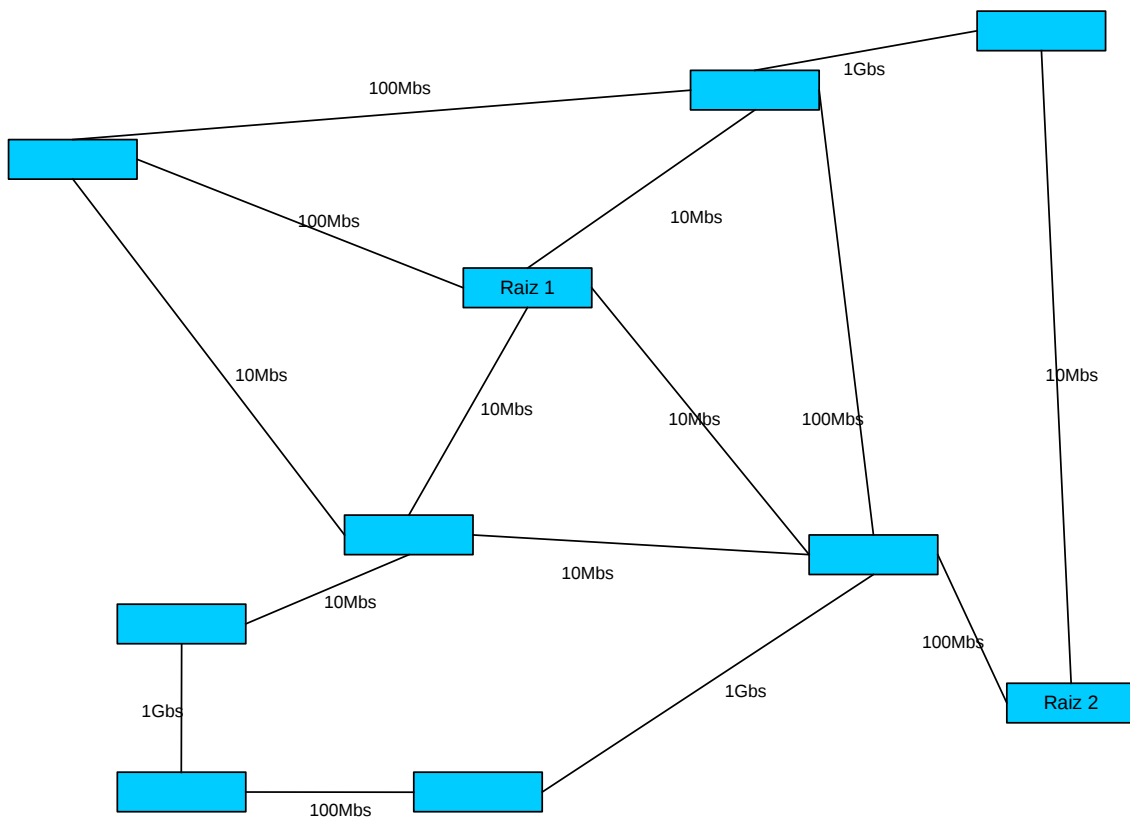


Figura 5: Red inicial

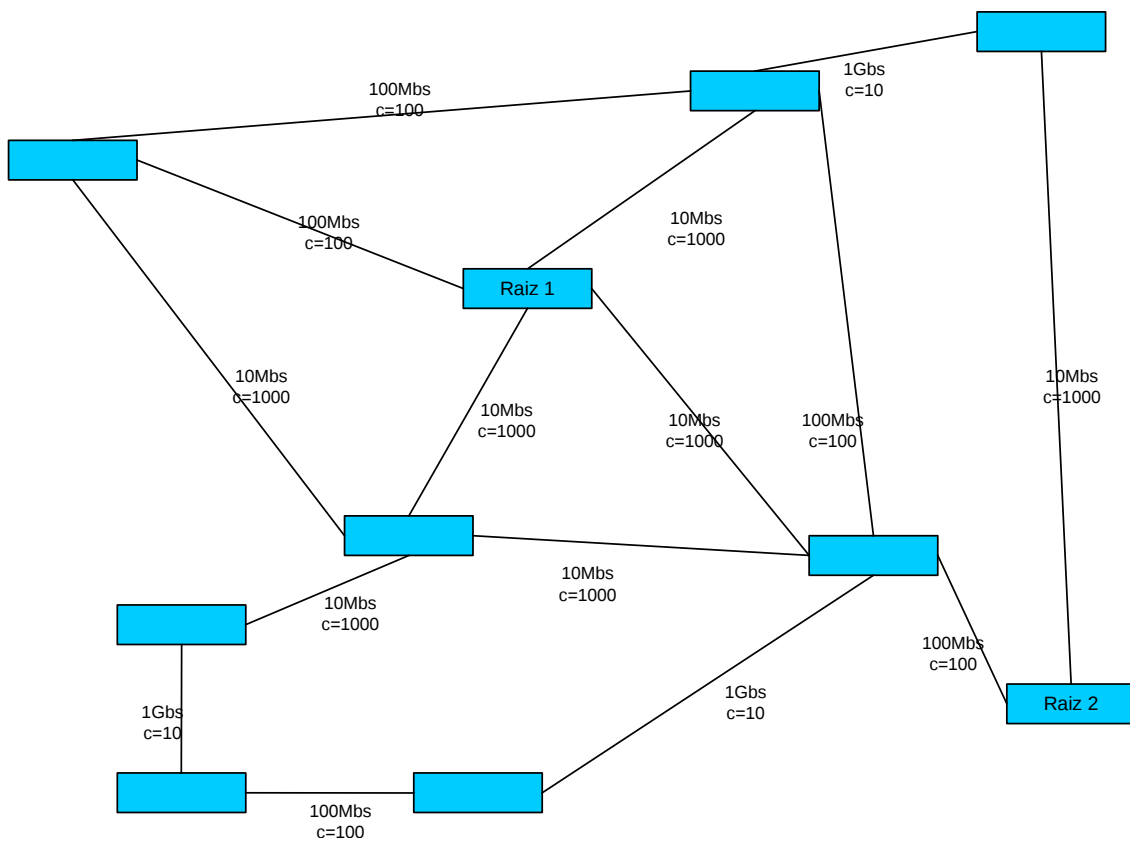


Figura 6: Costes de los enlaces

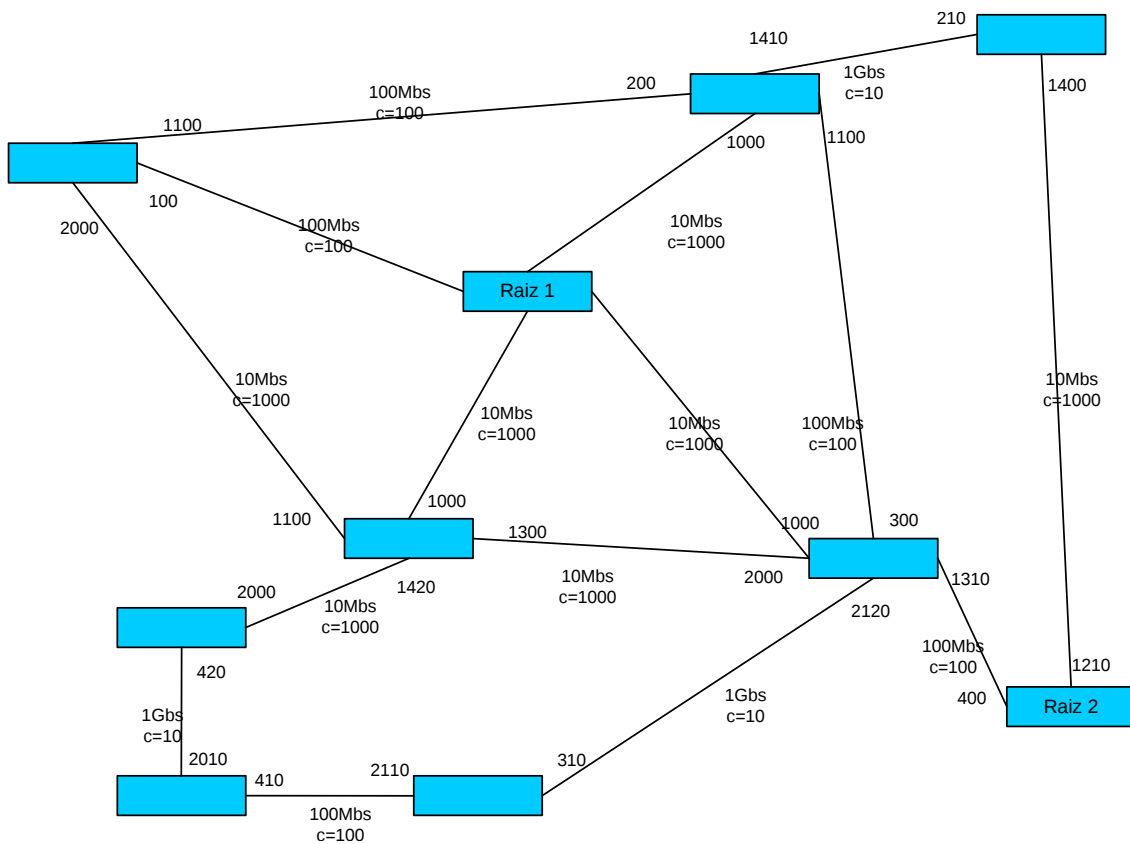
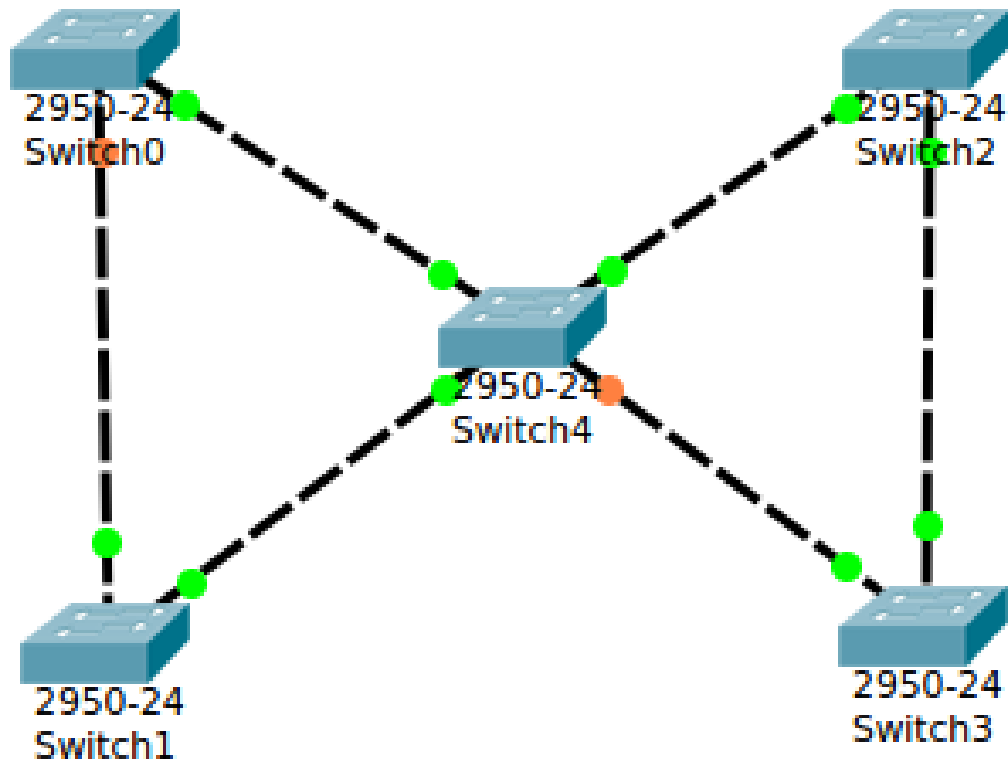


Figura 7: Coste del camino hacia el raíz de cada puerto

3.7. Ejercicio CISCO

```
Switch(config)# spanning-tree vlan 1 priority 1
```

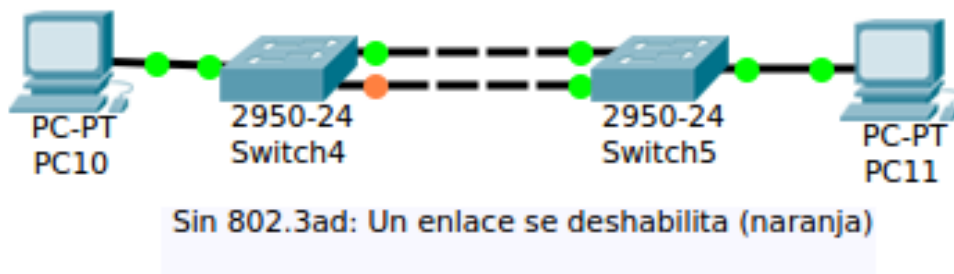
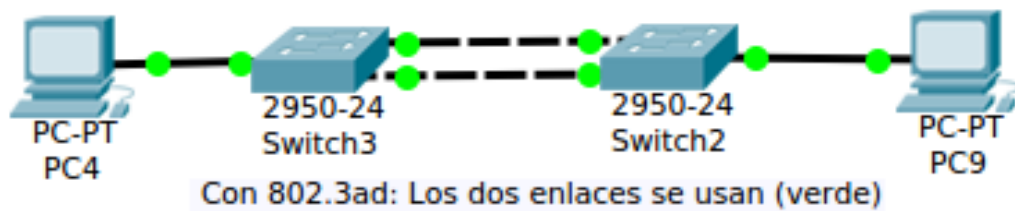
Consigue que switch4 sea el centro de la estrella



Fichero PKT

4. IEEE 802.3ad: Link Aggregation

- Los switches no envían entre sí tramas ethernet estándar
- Agrupan varios enlaces y los utilizan como uno solo
- Sirve para eliminar cuellos de botella entre switches.

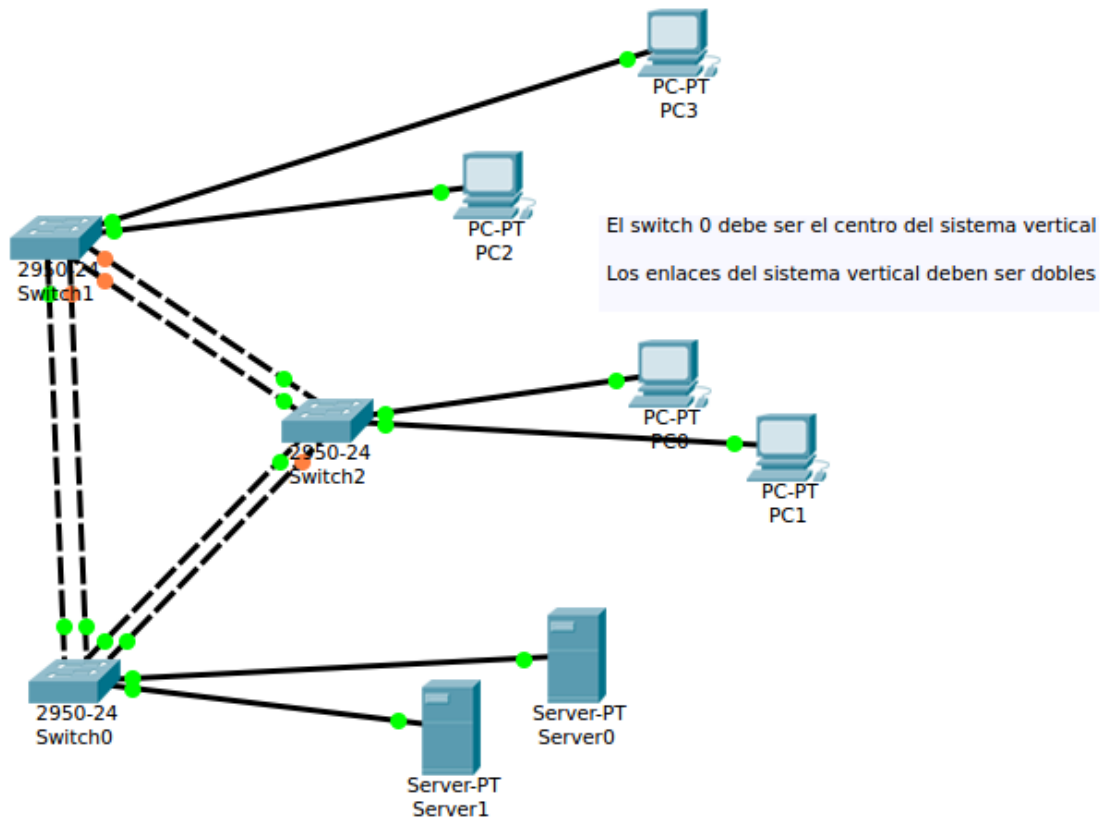


4.1. Configuración en Cisco

- Los puertos que se quieren agregar se añaden al mismo channel-group

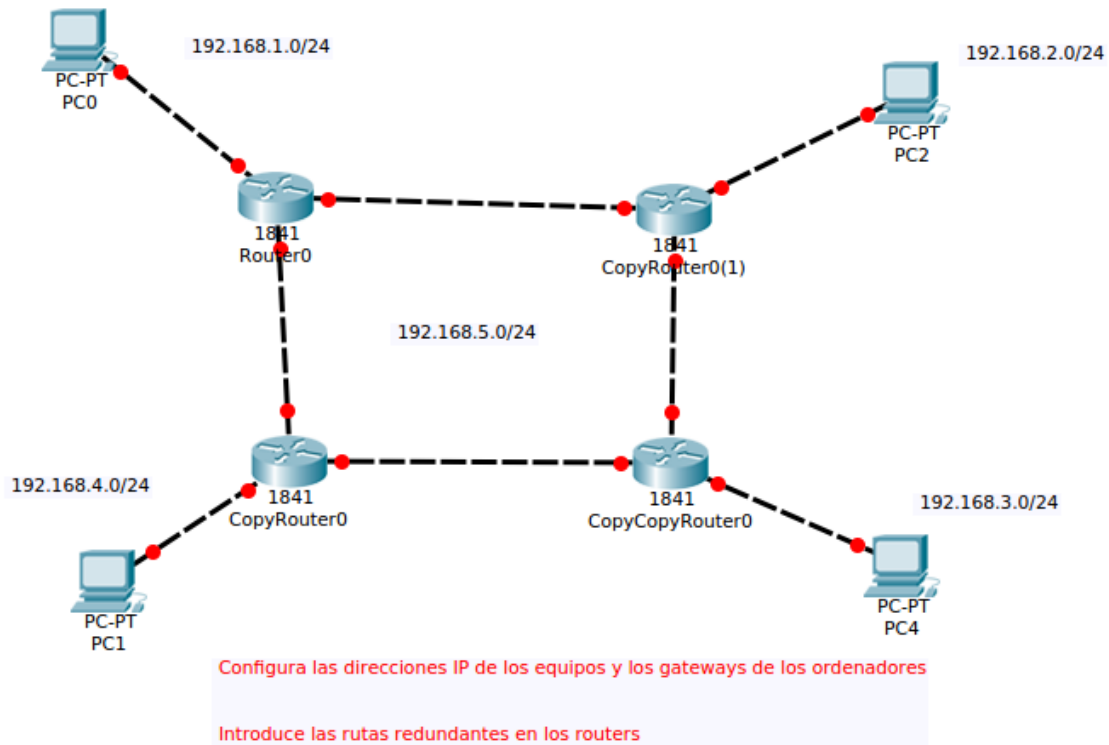
```
Switch(config)# interface fa0/1
Switch(config-if)# channel-group 1 mode active
```

4.2. Ejercicio



Enlace a fichero PKT

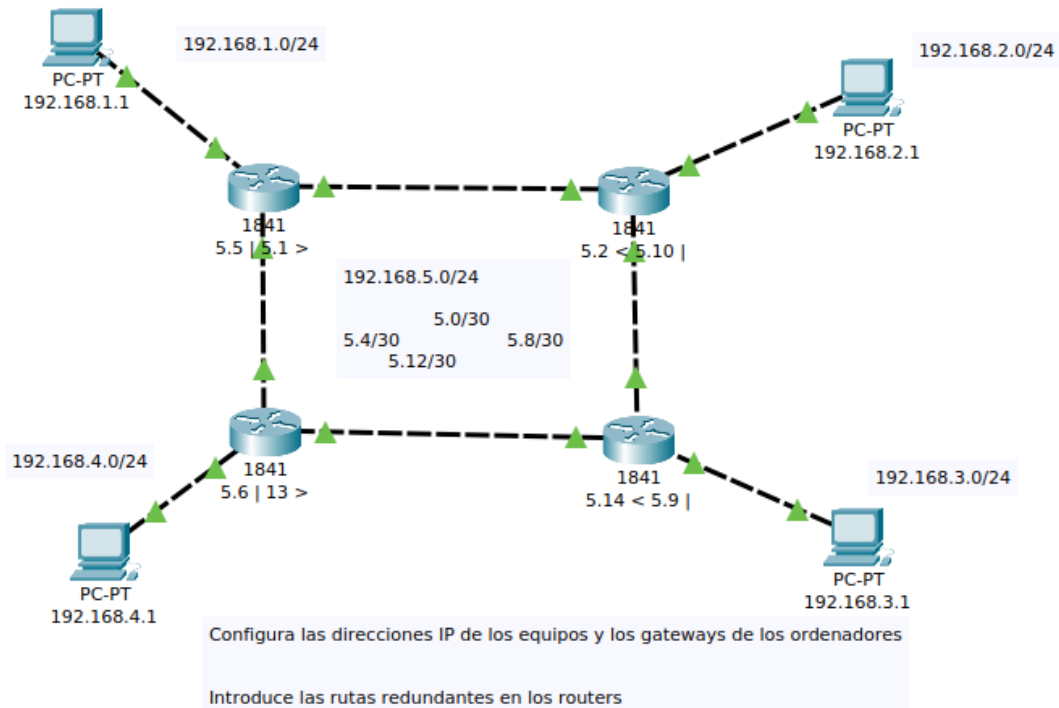
5. Redundancia a nivel de red



[Enlace a fichero PKT](#)

5.1. Problema con enrutamiento estático

- si se rompe un enlace, puede haber Bucles
- solución: algoritmos dinámicos



Enlace a fichero PKT

6. Referencias

- Formatos:
 - [Transparencias](#)
 - [PDF](#)
 - [EPUB](#)
- Creado con:
 - [Emacs](#)
 - [org-re-reveal](#)
 - [Latex](#)
- Alojado en [Github](#)