Spanning Tree Protocol

Area de Ingeniería Telemática
http://www.tlm.unavarra.es

Grado en Ingeniería en Tecnologías de Telecomunicación, 3º

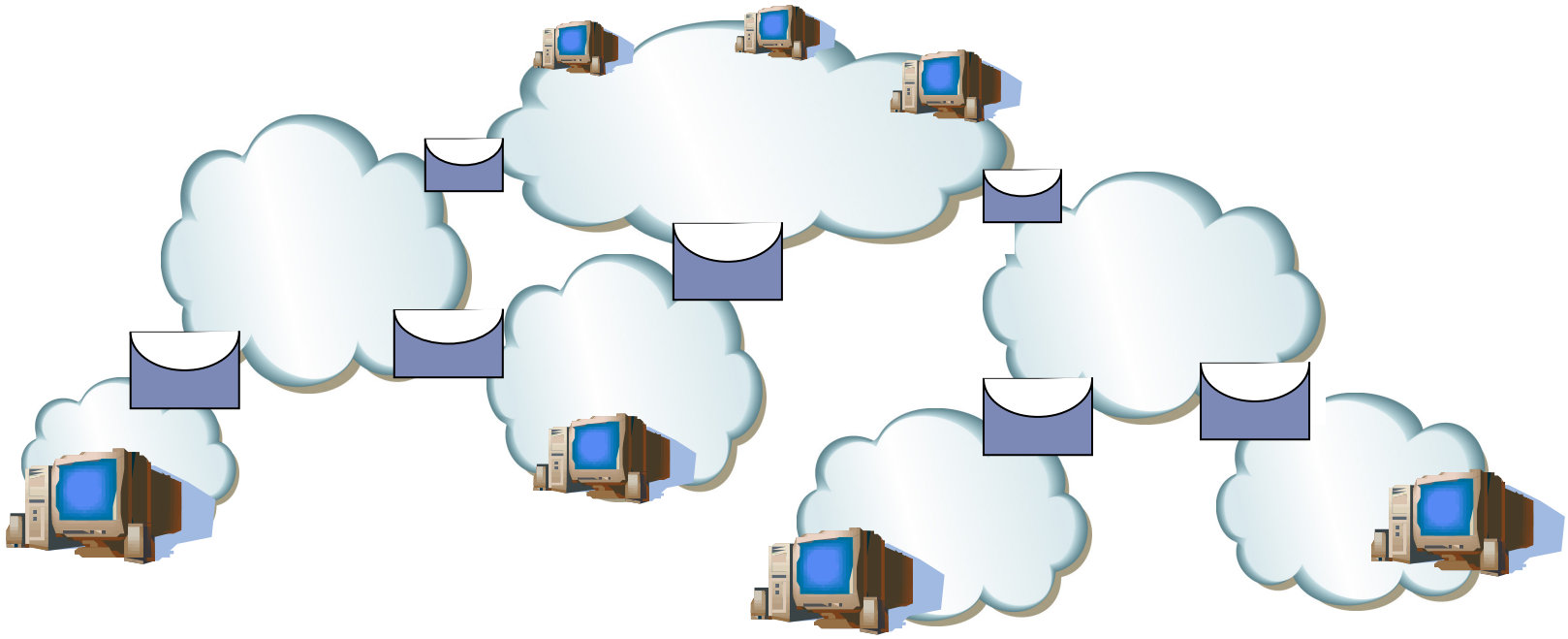
Temario

0. Introducción
1. **Tecnologías LAN**
   - Tecnologías Ethernet
   - Conmutación Ethernet
   - VLANs
   - **Spanning Tree Protocol**
   - Otros mecanismos en LANs Ethernet
   - WiFi
   - Diseño de redes campus
2. Tecnologías WAN
3. Redes de acceso

Bucles en LANs con puentes

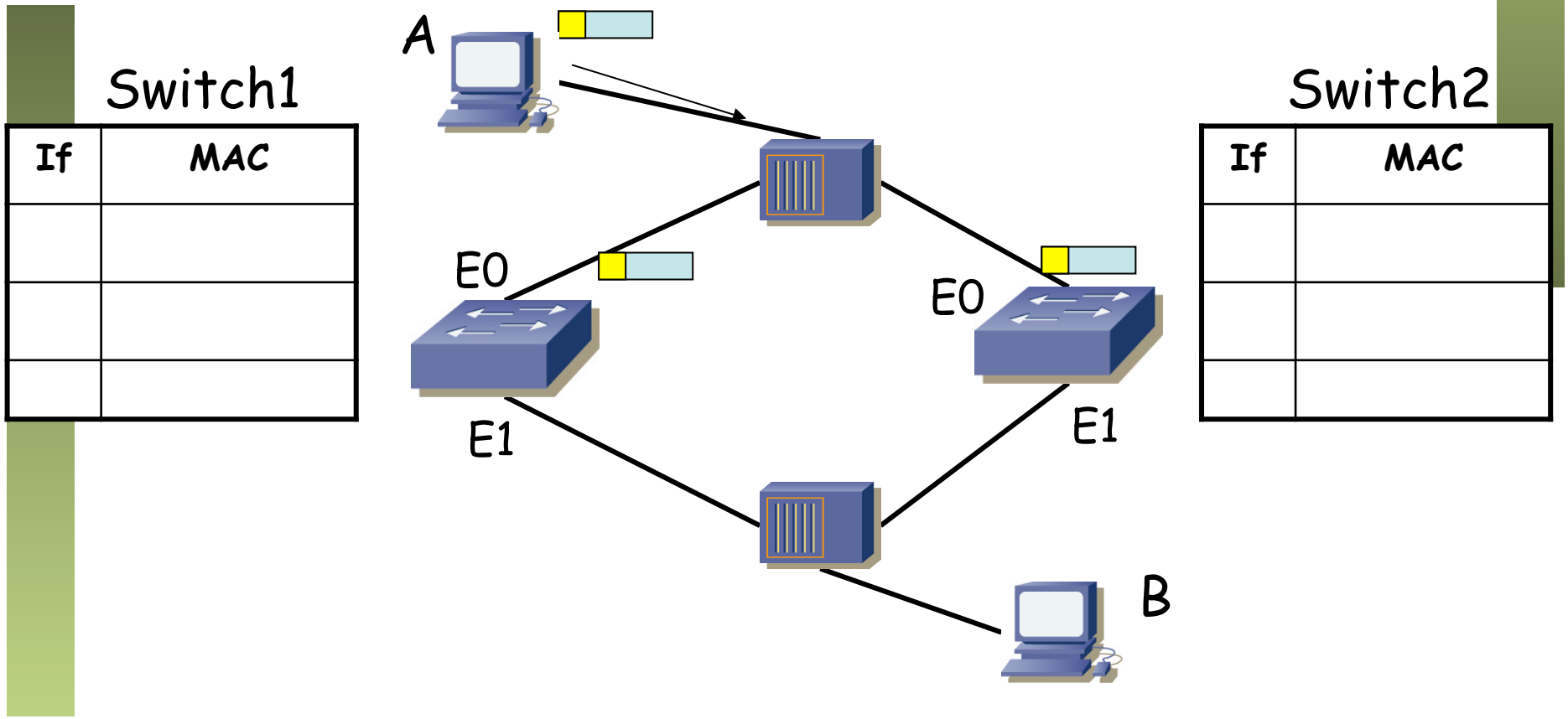
Caminos alternativos

- Ofrecerían la posibilidad de:
  - Balanceo de carga
  - Reconfiguración ante fallos
- Requiere tomar decisiones de encaminamiento



Caminos alternativos

- El host A envía una trama al host B

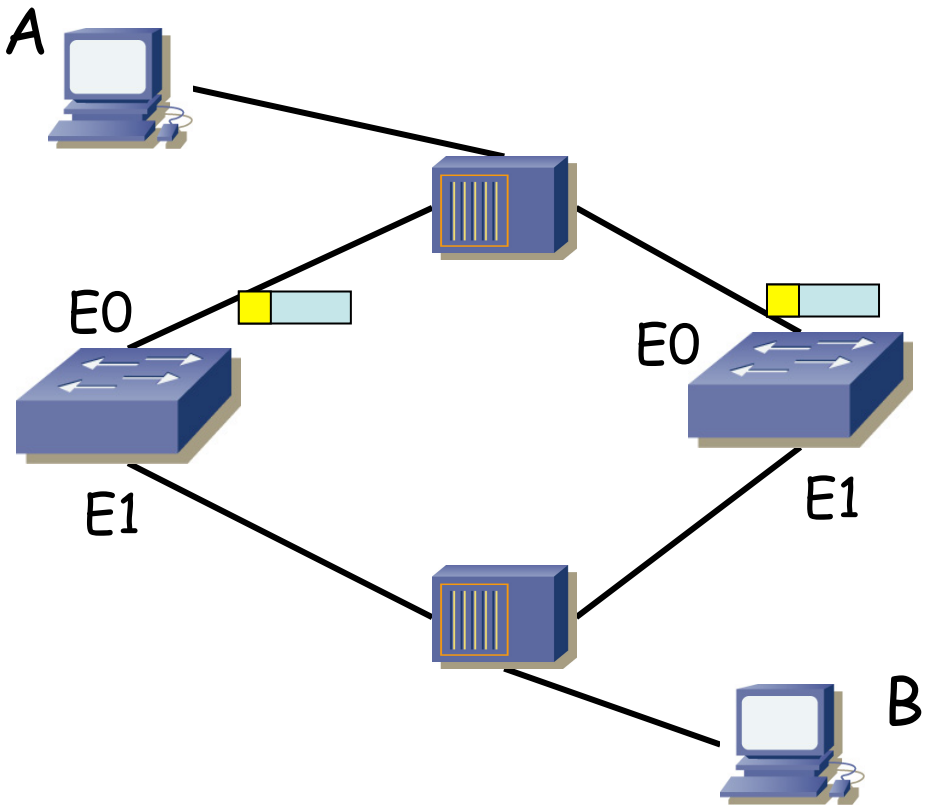


Caminos alternativos

- Switch1 y Switch2 aprenden la localización del host A

Switch1

If	MAC
E0	A

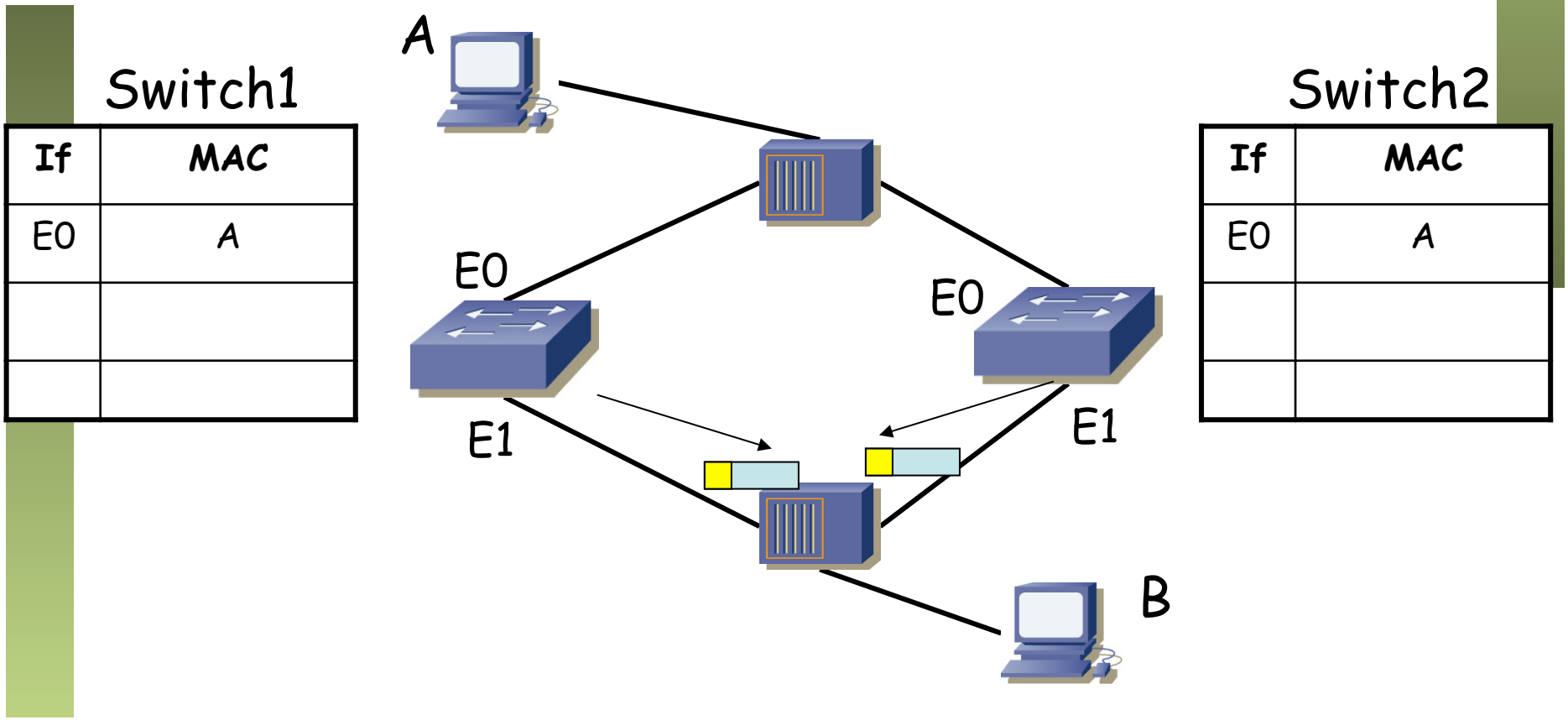


Switch2

If	MAC
E0	A

Caminos alternativos

- Los conmutadores no conocen al destino
- Reenvían por todos los puertos menos por donde recibieron

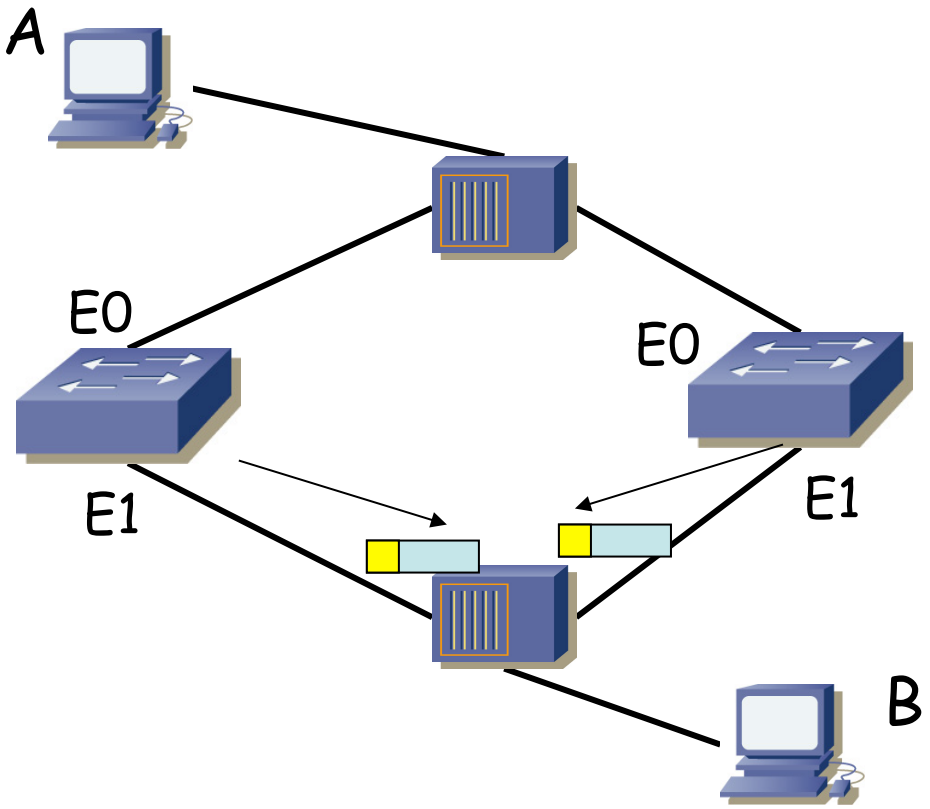


Caminos alternativos

- Host B recibe la trama
- Switch2 recibe la trama que envió Switch1
- Switch1 recibe la trama que envió Switch2

Switch1

If	MAC
E0	A

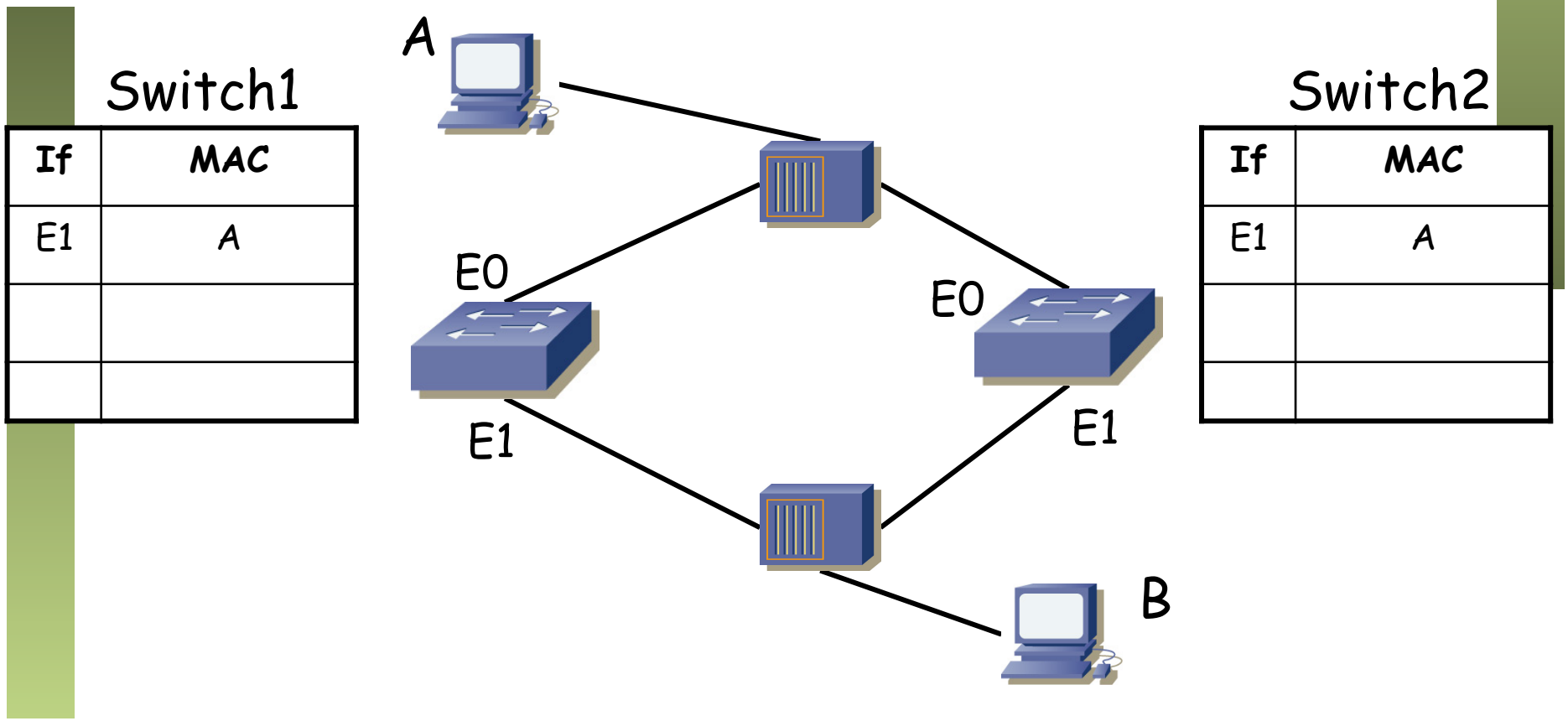


Switch2

If	MAC
E0	A

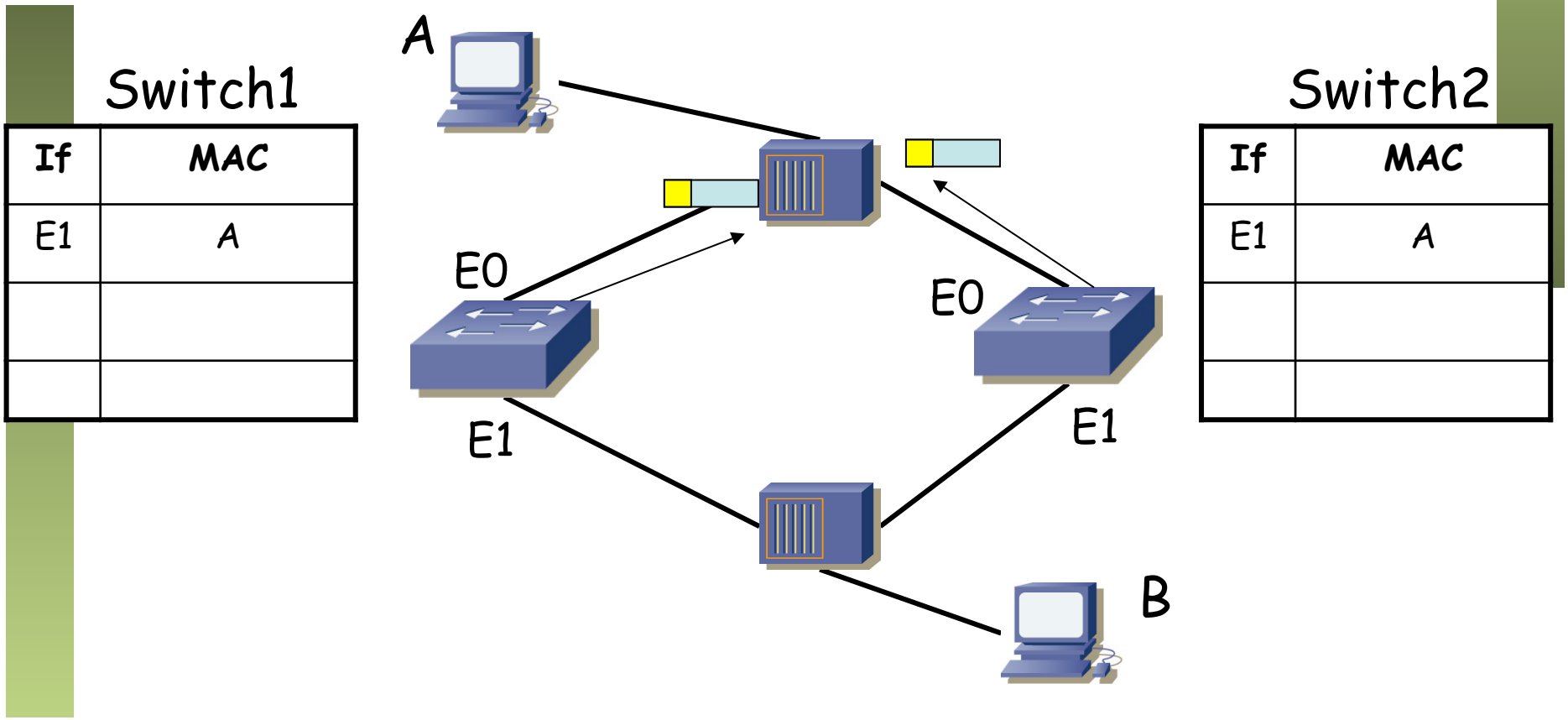
Caminos alternativos

- Aprenden una nueva ubicación del host A



Caminos alternativos

- Aprenden una nueva ubicación del host A
- Y reenvían por todos los puertos menos por donde recibieron la trama

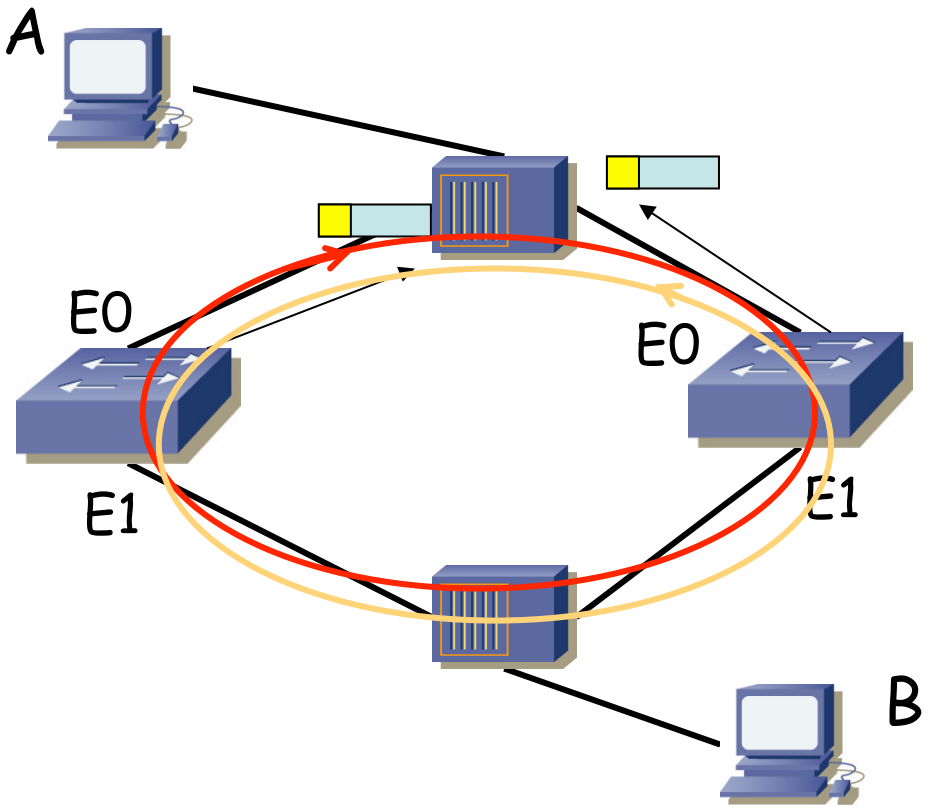


Caminos alternativos

- Y se repite…
- No hay TTL en la trama Ethernet
- Además todos los hosts la deberían procesar

Switch1

If	MAC
E1	A

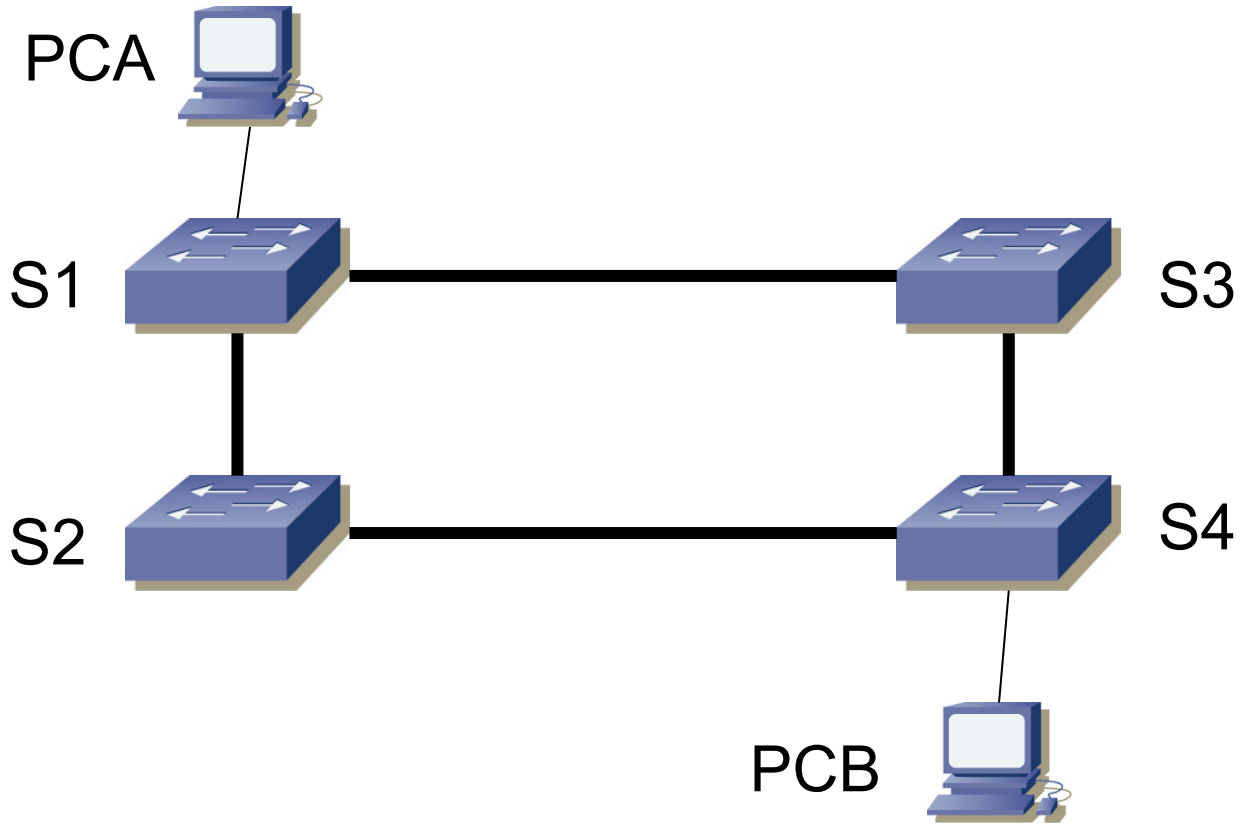


Switch2

If	MAC
E1	A

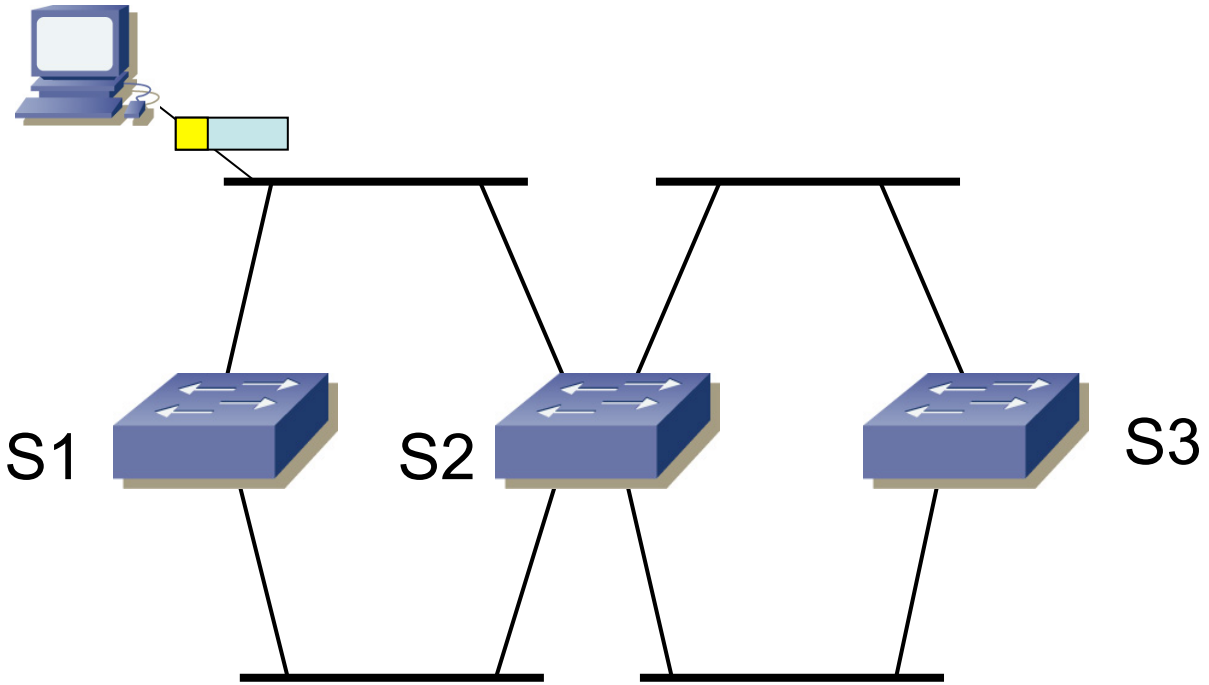
Sin hubs también sucede

- Envía una trama a la MAC de broadcast desde uno de estos PCs...



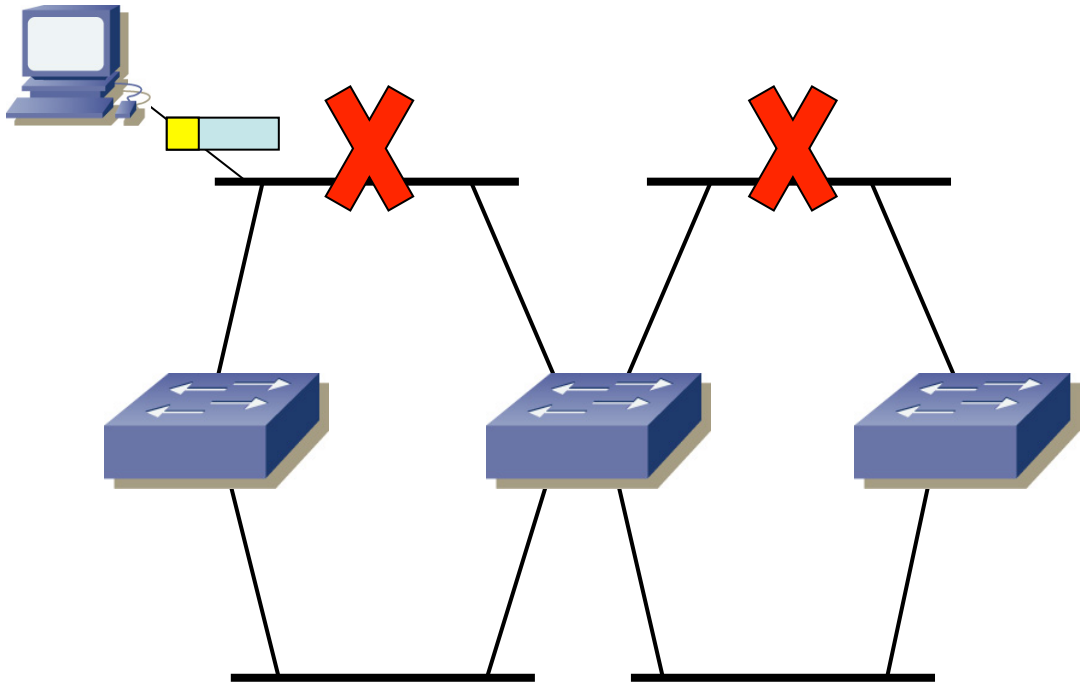
Ejemplo

- Y en este caso se multiplican al pasar por S2
- ¿Habéis oído el término *broadcast storm?*
- Creedme, no queréis una en una red que gestionéis



Soluciones

- ¡ Emplear una topología sin ciclos !



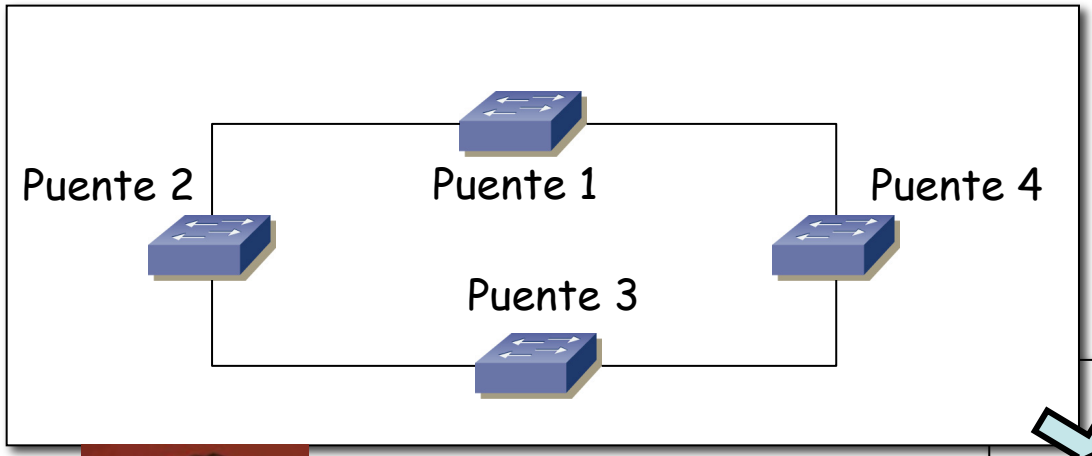
Soluciones

- ¡ Emplear una topología sin ciclos !
- Vale, resuelve el problema pero no conseguimos la protección
- Yo no he dicho que fuera la mejor solución
- ¿ Otras alternativas ? (¿Ideas?)

Spanning-Tree Protocol

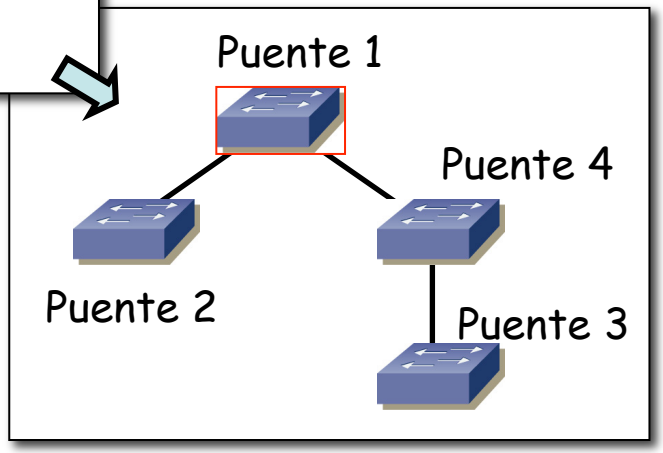
Spanning-Tree Protocol (STP)

- Calcula una topología libre de ciclos
- A partir del grafo de la topología crea un árbol
- ¿Cómo? Seleccionan un puente como "raíz" y desde él calculan un árbol
- Desactivan los enlaces que no están en el árbol



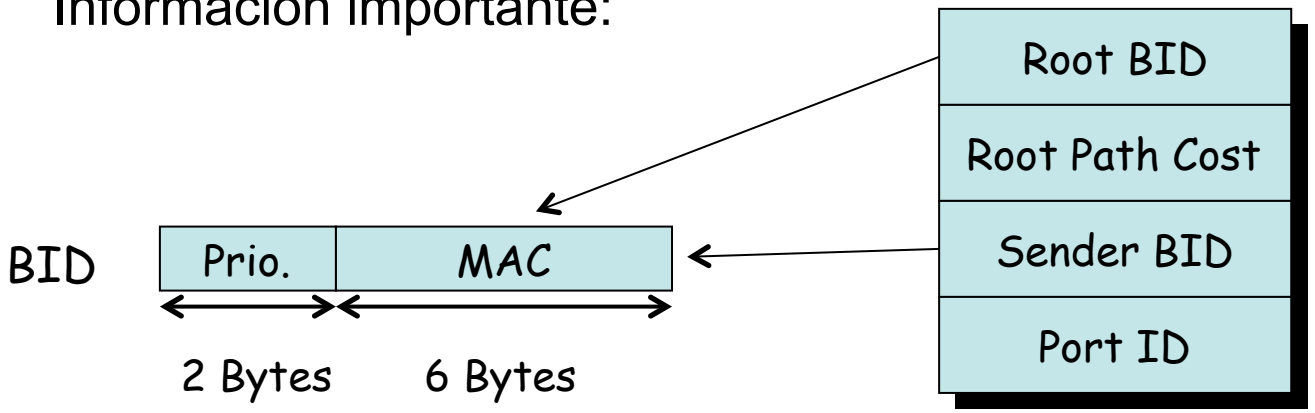
IEEE 802.1D

Radia Perlman



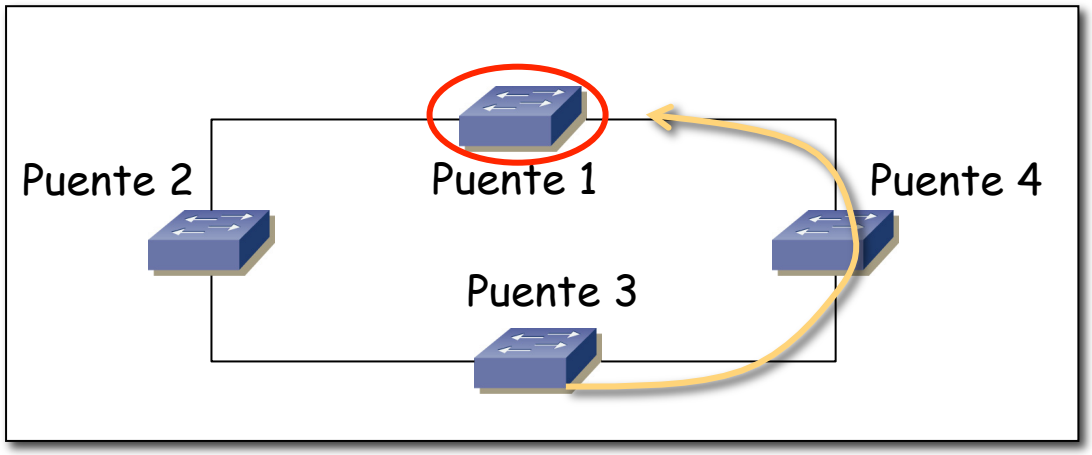
STP: BPDUs

- Bridge Protocol Data Units
- Enviadas periódicamente por los puentes, por todos sus puertos
- Destino 01:80:C2:00:00:00 (Bridge Group Address)
- No son reenviadas
- BID = Bridge ID
- Información importante:



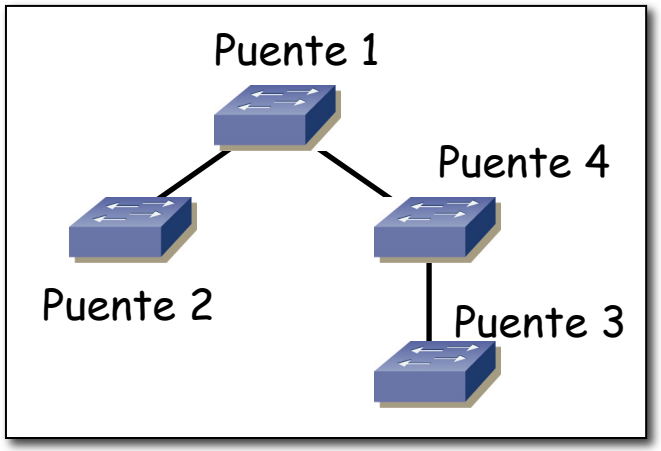
STP: BPDUs

- Se pueden "comparar" entre si y decidir si una BPDU recibida por un puerto es "mejor" que otra
- "Mejor" en el sentido de "mejor" camino a la raíz
- Para ello cada enlace tendrá un coste y se suman
- El coste de un enlace suele depender de su velocidad



Dos detalles importantes

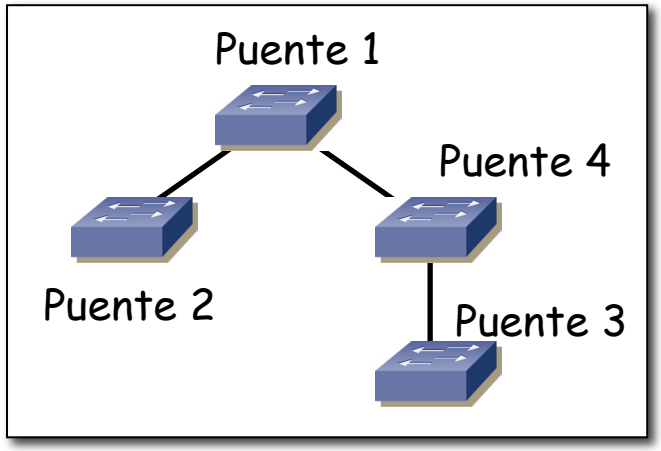
- Los conmutadores siguen enviando BPDUs tras calcular árbol
  - Un conmutador no tiene forma de saber que se ha alcanzado una topología estable
  - Seguir enviando y aceptando BPDUs permite recalcular el árbol ante fallos
- El plano de datos del conmutador no cambia
  - Es decir, los conmutadores siguen reenviando los paquetes de usuario en función de sus tablas
  - Y siguen aprendiendo igual
  - Simplemente algunos puertos los tienen desactivados
  - Para decidir por dónde va una trama necesitamos saber qué puertos están desactivados pero por lo demás sigue la misma lógica de siempre



STP/RSTP: Mecanismos

Mecanismos

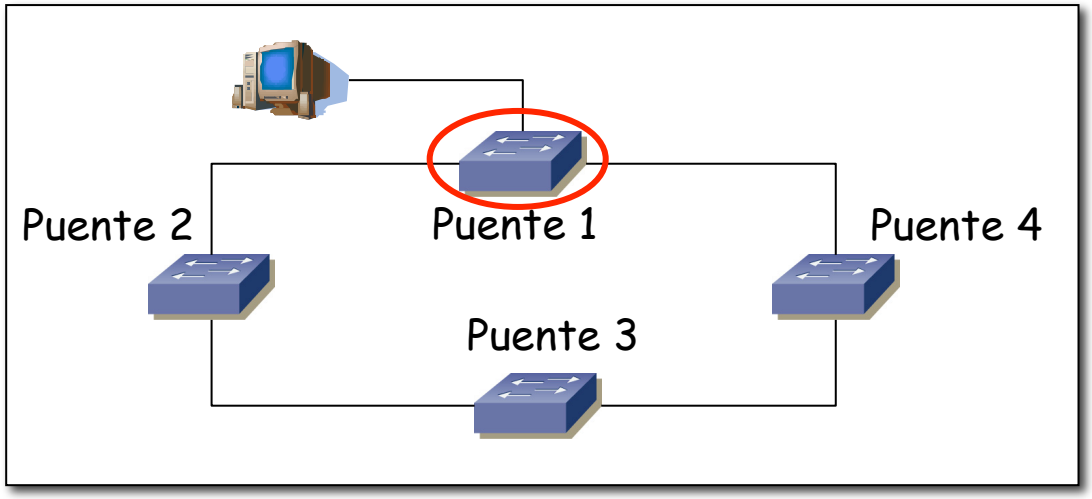
- Vamos a intentar entender cómo se calcula el árbol
- No nos interesa el “transitorio” sino solamente el estado final

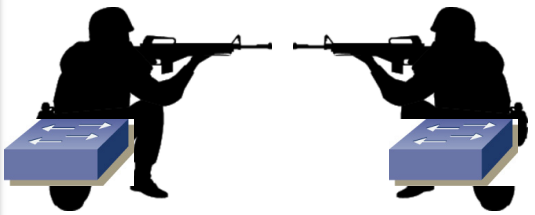


STP: Root Bridge

Selección de un *Root Bridge* (Root War !!!)

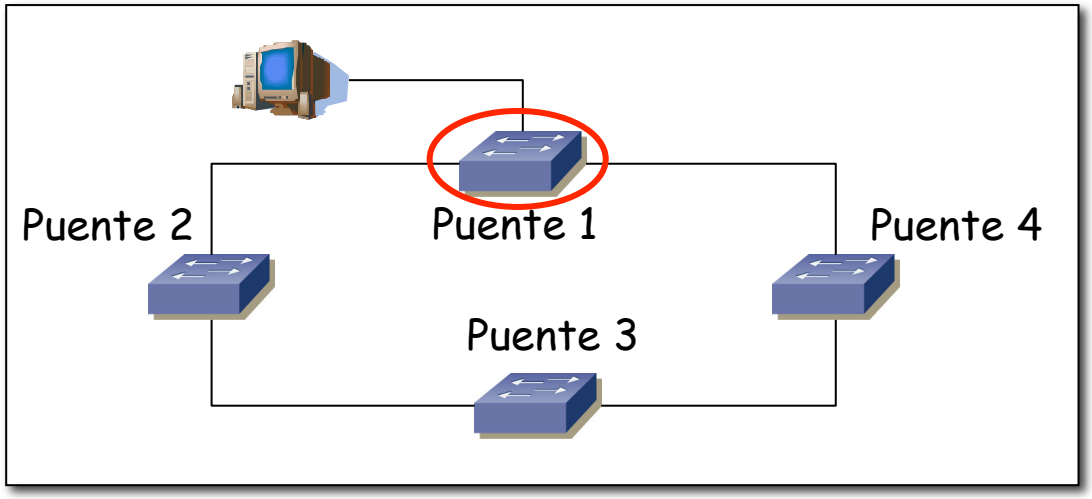
- Raíz para el árbol
- No es un primer paso sino que para cualquier BPDU que se recibe se decide si anuncia mejor root
- A partir de un valor de prioridad y una MAC del puente
  - Vienen en las BPDU
  - Puente de prioridad más baja (def. 0x8000)
  - MAC más baja en caso de empate

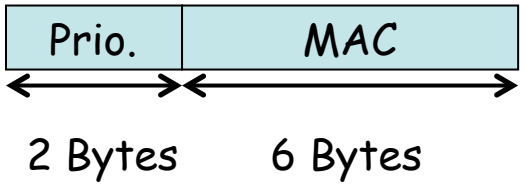




STP: Root Bridge

- Por defecto todos los puentes la misma prioridad
- Gana el de dirección MAC más baja
- Primeros 3 bytes de la MAC son el OUI
- ¡ Luego el ganador depende del fabricante !
- Cuidado pues puede ser el conmutador más lento
- Selección manual con el campo de prioridad





STP: Path Cost

- Asociado a cada LAN; según su velocidad
- Originalmente *1000 / Velocidad*(Mbps)
- 802.1D-2004 :

Table 17-3—Port Path Cost values

Link Speed	Recommended value	Recommended range	Range
<=100 Kb/s	200 000 000*	20 000 000–200 000 000	1–200 000 000
1 Mb/s	20 000 000[a]	2 000 000–200 000 000	1–200 000 000
10 Mb/s	2 000 000[a]	200 000–20 000 000	1–200 000 000
100 Mb/s	200 000[a]	20 000–2 000 000	1–200 000 000
1 Gb/s	20 000	2 000–200 000	1–200 000 000
10 Gb/s	2 000	200–20 000	1–200 000 000
100 Gb/s	200	20–2 000	1–200 000 000
1 Tb/s	20	2–200	1–200 000 000
10 Tb/s	2	1–20	1–200 000 000

*Bridges conformant to IEEE Std 802.1D, 1998 Edition, i.e., that support only 16-bit values for Path Cost, should use 65 535 as the Path Cost for these link speeds when used in conjunction with Bridges that support 32-bit Path Cost values.

- Nos podemos encontrar otros valores:
  - Cisco “short path cost”: 10Mbps (100), 100Mbps (19), 1Gbps (4), 10Gbps (2)
- Se va agregando en un camino creando el *Root Path Cost*
- En un switch se calcula a partir de los anuncios recibidos en cada puerto, sumando el coste del puerto por el que han llegado

STP: Port States

- STP definía 5 estados posibles para un puerto: *disabled, listening, learning, blocking, forwarding*
- Estos estados mezclaban por un lado si el puerto reenviaba o no tramas y por otro el papel que jugaba el puerto en el árbol
- RSTP separa *port states* de *port roles*
  - Los *estados* definen si se reenvían las tramas, si se aprenden direcciones MAC
  - Los *roles* definen el papel que juega el puerto en el árbol

RSTP

Rapid Spanning-Tree Protocol

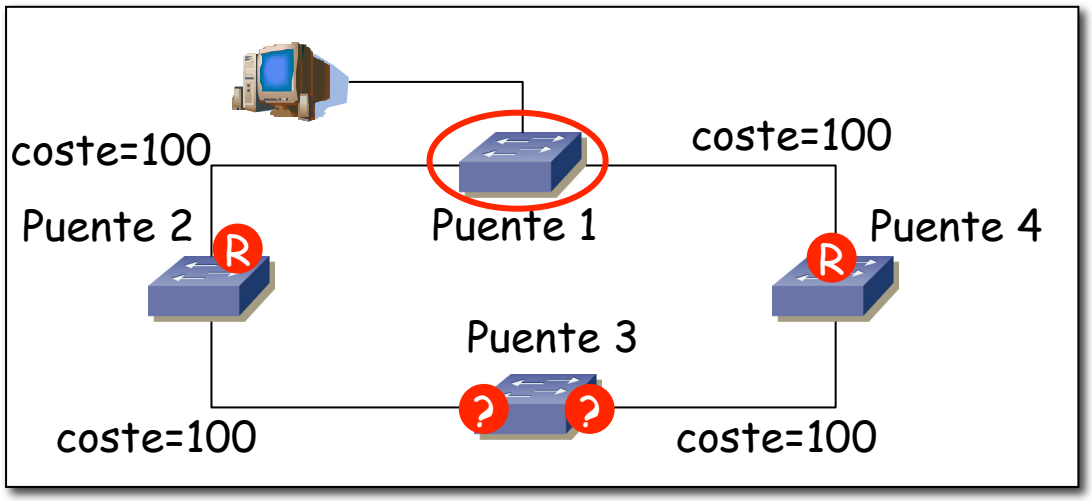
- IEEE 802.1w
- STP obsoleto
- RSTP en 802.1D-2004
- Tiempos de convergencia de 2-3 segs
- Tres **estados** posibles para un puerto:
  - *Discarding*: ni envía ni acepta paquetes de usuario
  - *Learning*: no envía ni acepta paquetes de usuario pero aprende MACs
  - *Forwarding*: funcionamiento normal
- No vamos a detallar el diagrama de estados con sus transiciones
- Sí vamos a detallar el significado de los *roles* pues ayudan a entender cómo se calcula el árbol

RSTP: Port Roles

RSTP: Port Roles

Root Port

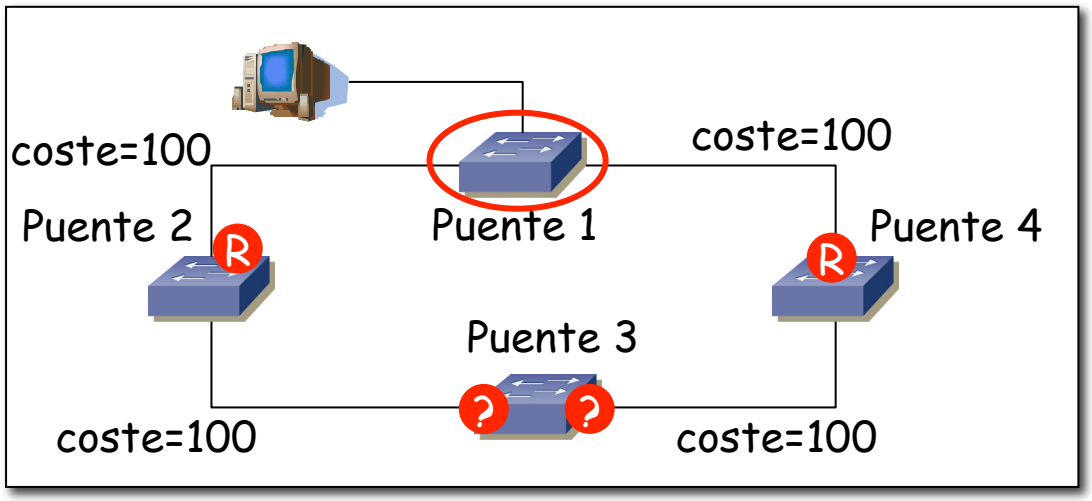
- Uno en cada puente salvo en el puente raíz
- El puente raíz es el único sin un puerto raíz
- Es el puerto de un conmutador que tiene el menor *Root Path Cost+Port Cost* (menor coste hasta la raíz)
- En este ejemplo supongamos que todos los puertos tienen el mismo coste
- Está claro cuál es *root port* en los puentes 2 y 4 pero ¿y en 3?



RSTP: Port Roles

¿Empates?

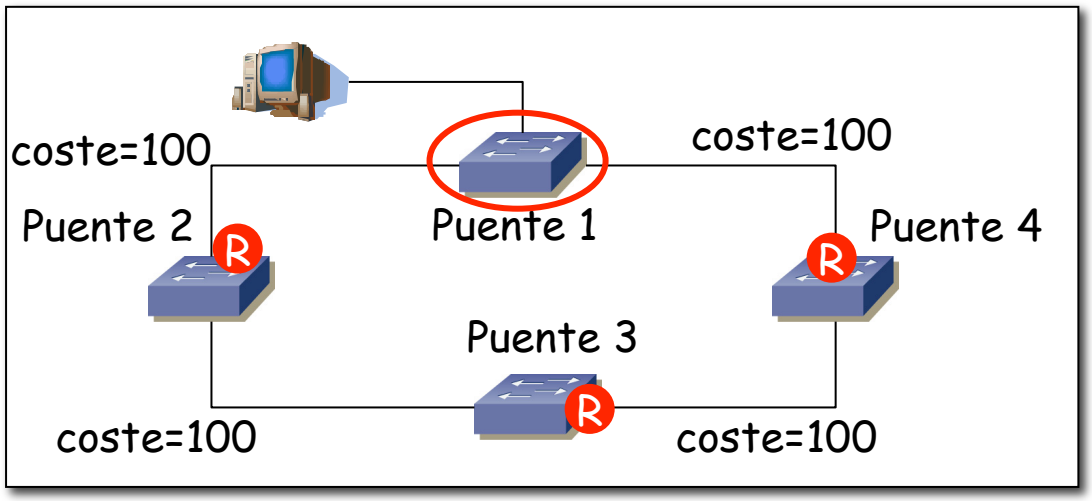
- En el puente 3 los costes que recibe de puente 2 y de puente 4 hasta la raíz son el mismo = 100
- A cada uno le suma el coste del puerto por el que lo ha recibido y empatan (le sale 200 en ambos)
- Entonces se comparan los BID de los puentes que hacen el anuncio
- El anuncio que venga del BID menor gana (…)



RSTP: Port Roles

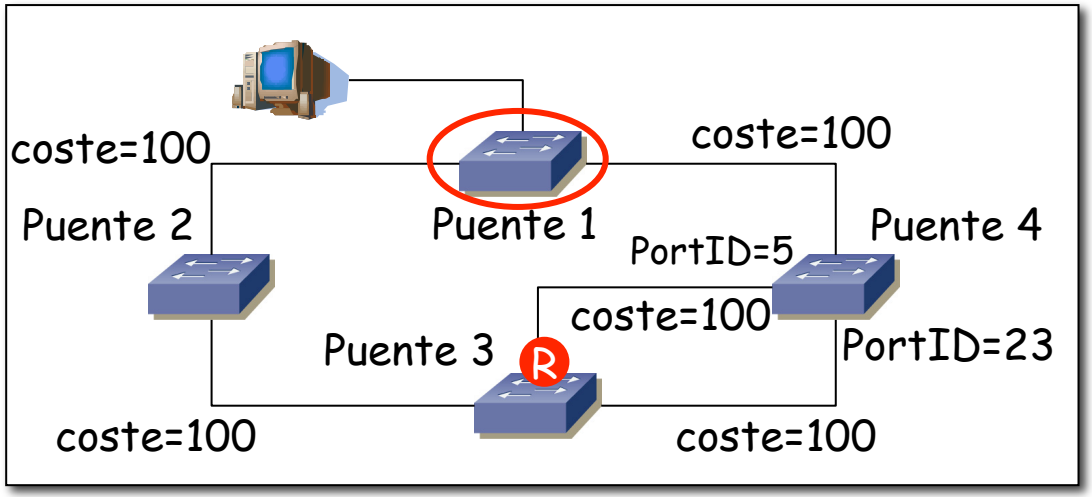
Desempate

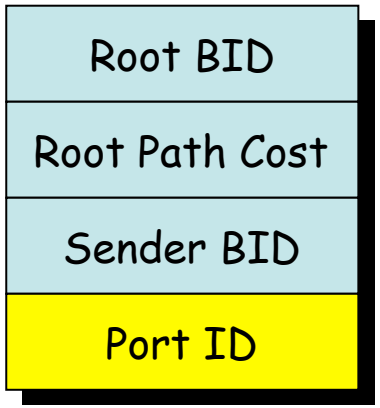
- Suponiendo que BID del puente 4 < BID del puente 2 sería puerto raíz de puente 3 el que va al puente 4



RSTP: Port Roles

- Hemos dicho que cuando hay que elegir un camino a la raíz se toma el menor coste agregado
- Si hay empate el menor BID
- ¿Hay más posibilidades de empate?
- En este caso empatarían los puertos a puente 4 (suponiendo que el BID del puente 4 es menor que el del puente 2)
- ¿Desempate? Menor identificador de puerto

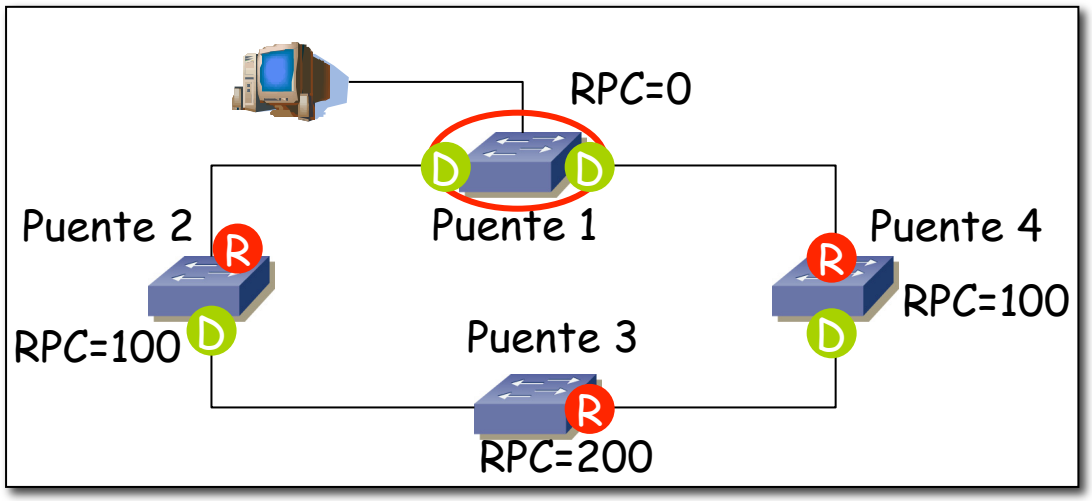




RSTP: Port Roles

Designated Port

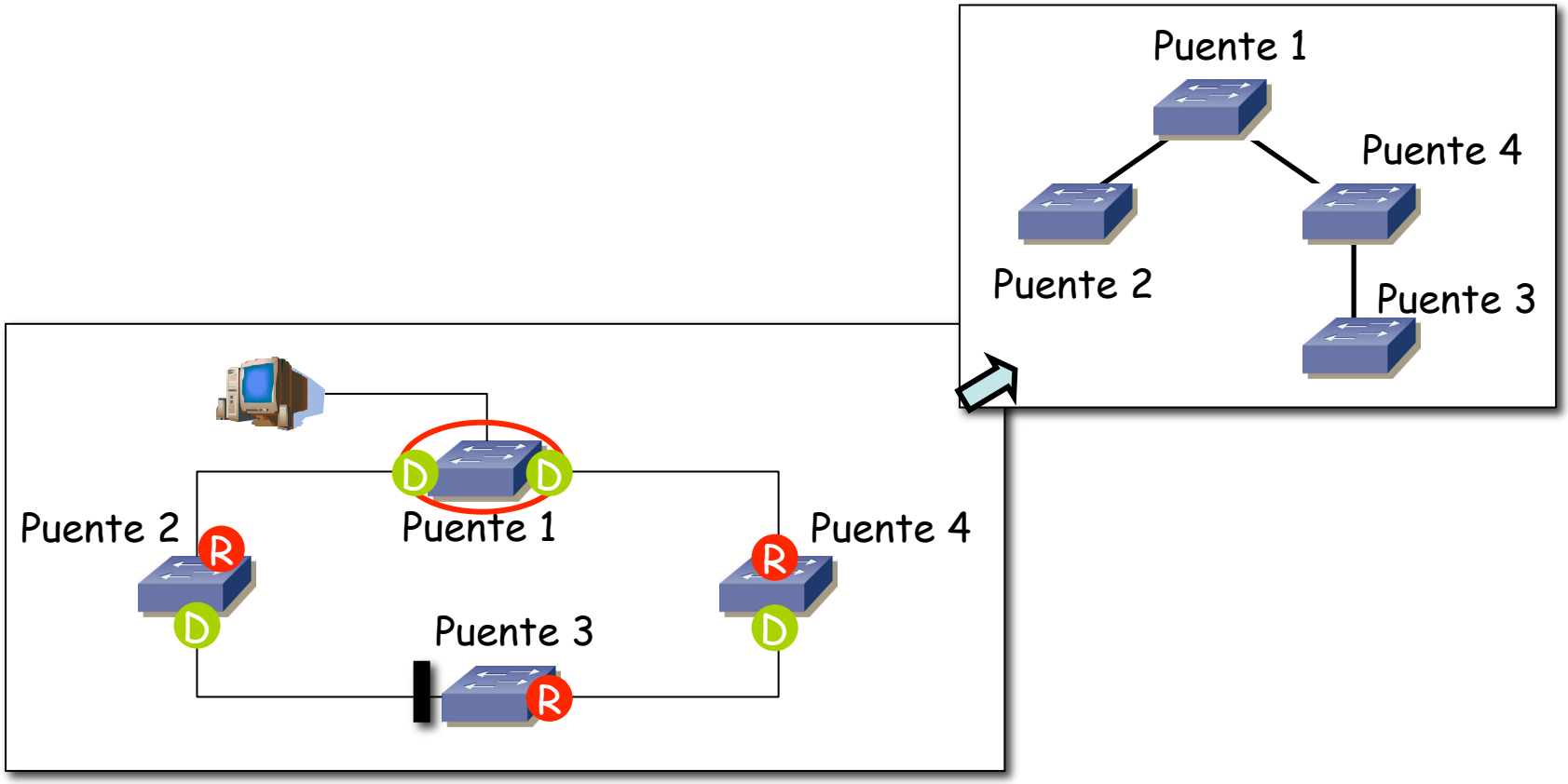
- En un segmento de LAN habrá varios puertos de conmutador
- El puerto del conmutador de esos con menor *Root Path Cost +Port Cost* es puerto designado para la LAN
- Uno por segmento de LAN
- Si hay empate por costes se desempata por el BID



RSTP: Port Roles

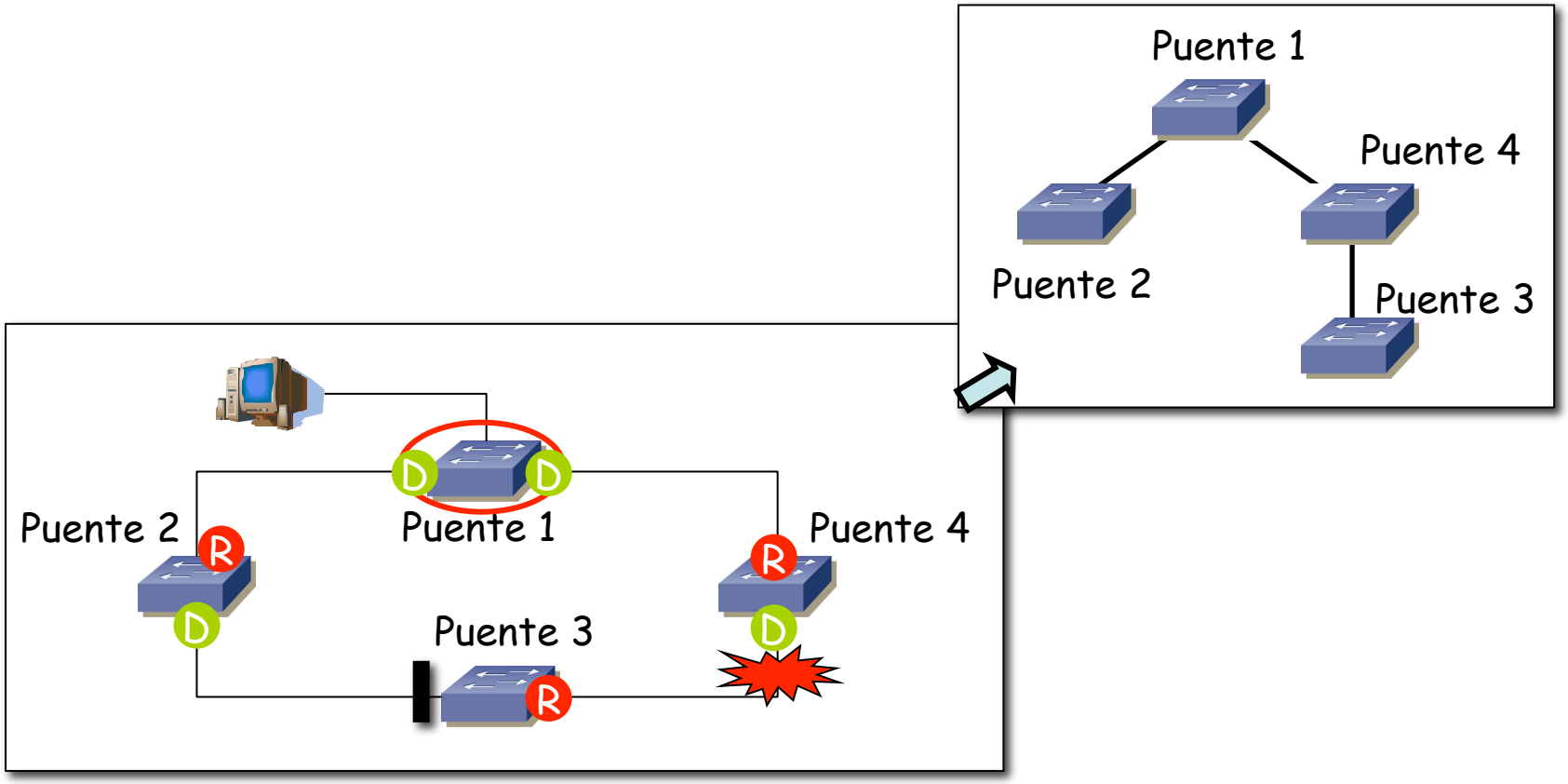
Blocked Port

- No aprenden MACs ni reenvían tramas
- Aceptan BPDUs
- Todos aquellos que ni son *Root* ni *Designated*



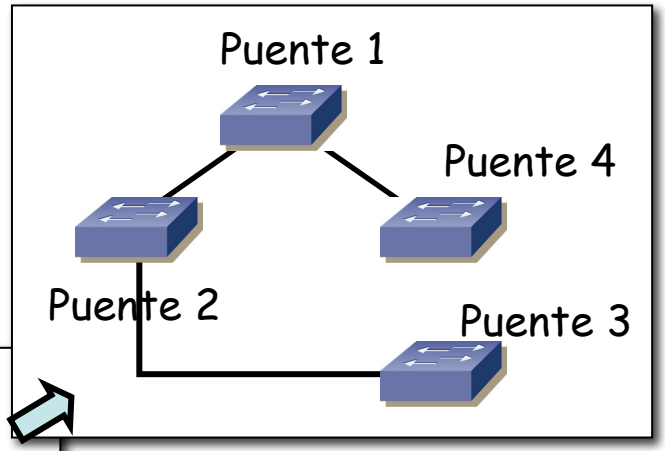
RSTP: cambios en la topología

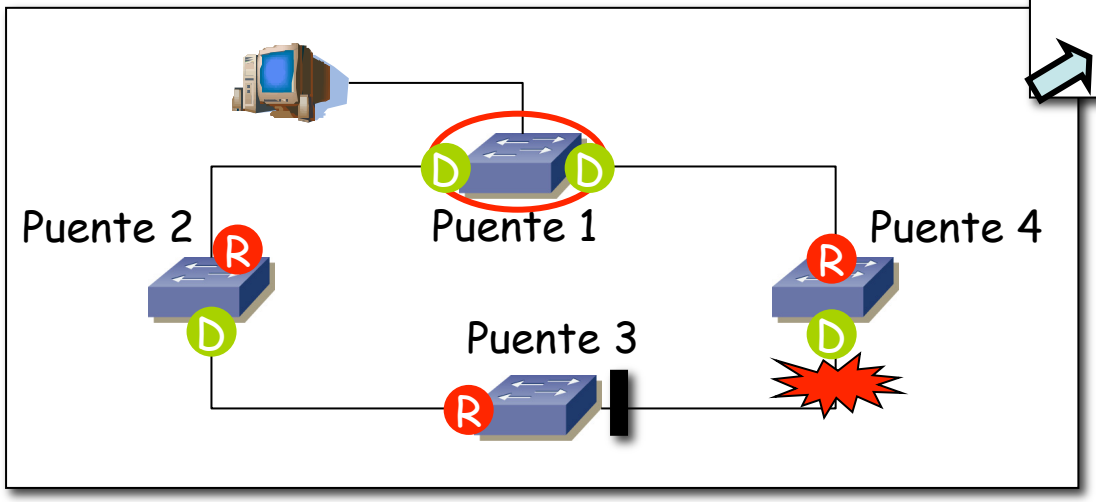
- Ante un fallo se recalcula el árbol (…)



RSTP: cambios en la topología

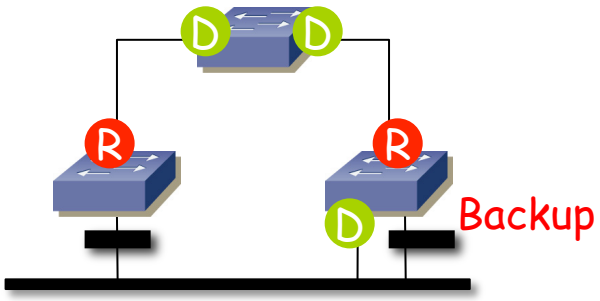
- Ante un fallo se recalcula el árbol (…)
- ¿Cómo?
  - Se dejan de recibir BPDUs donde se produce el fallo
  - Otro camino pasa a ser mejor
  - Se mandan nueva BPDUs
- Tiempo de convergencia:
  - STP 30-60 segs
  - RSTP 2-3 segs





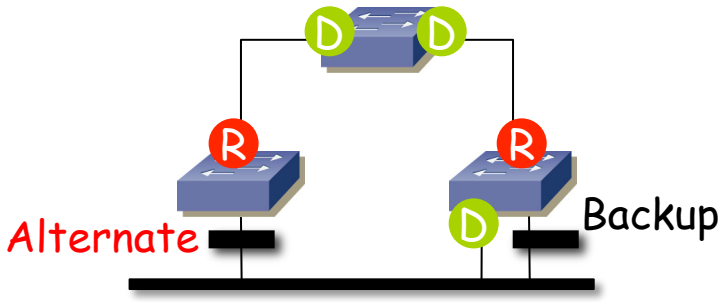
RSTP: *Port Roles*

- *Disabled:* puerto retirado mediante gestión
- *Alternate* y *Backup*:
  - Corresponden a lo que antes eran *blocked port*
  - *Backup* es todo puerto que no es ni *Root* ni *Designated* y el puente es *Designated* para esa LAN
  - *Backup port* da un camino alternativo pero siguiendo el mismo camino que el *Root port*
  - *Backup port* solo existe donde haya 2+ enlaces de un puente a una LAN
  - *Backup* está bloqueado porque se han recibido BPDUs mejores **del mismo switch** en el mismo segmento
  - (...)



RSTP: *Port Roles*

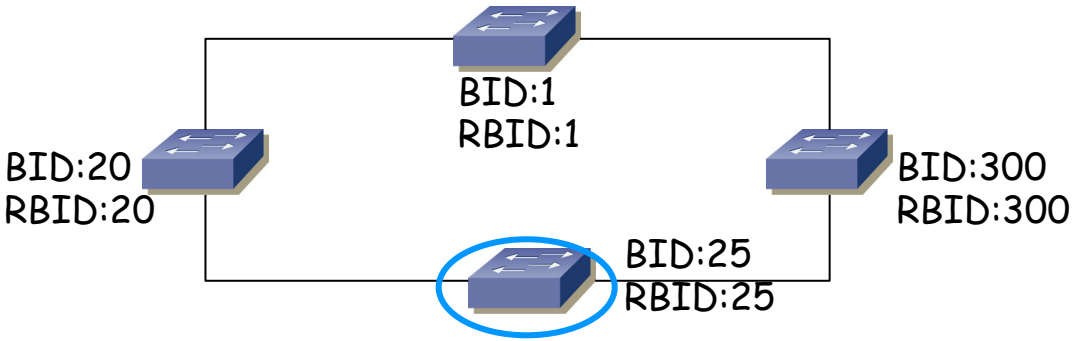
- *Disabled:* puerto retirado mediante gestión
- *Alternate* y *Backup*:
  - Corresponden a lo que antes eran *blocked port*
  - Un *Alternate port* da un camino alternativo hacia el root frente al puerto que se tiene como *Root*
  - *Alternate* está bloqueado porque se han recibido BPDUs mejores (menor coste) de otro switch en el mismo segmento

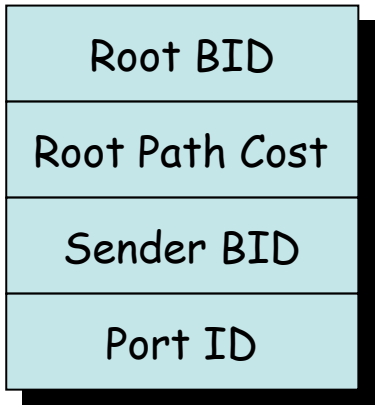


Comparación de costes

Comparación de costes

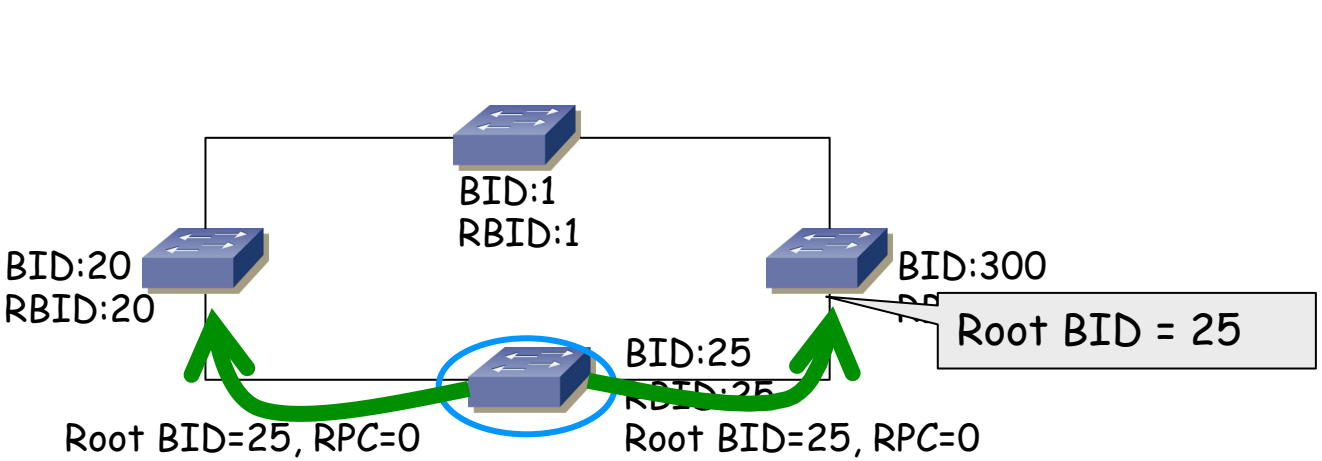
- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - Tiene el mismo valor de Sender BID y de Port ID que los guardados como Designated BID y Port ID (nos lo manda el mismo)
  - o (...)

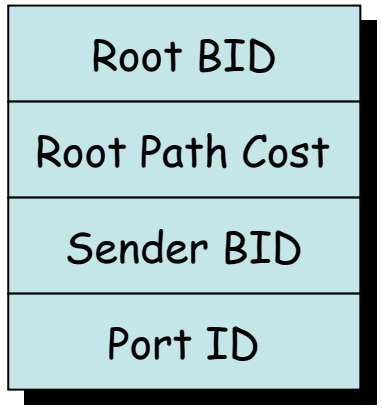




Comparación de costes

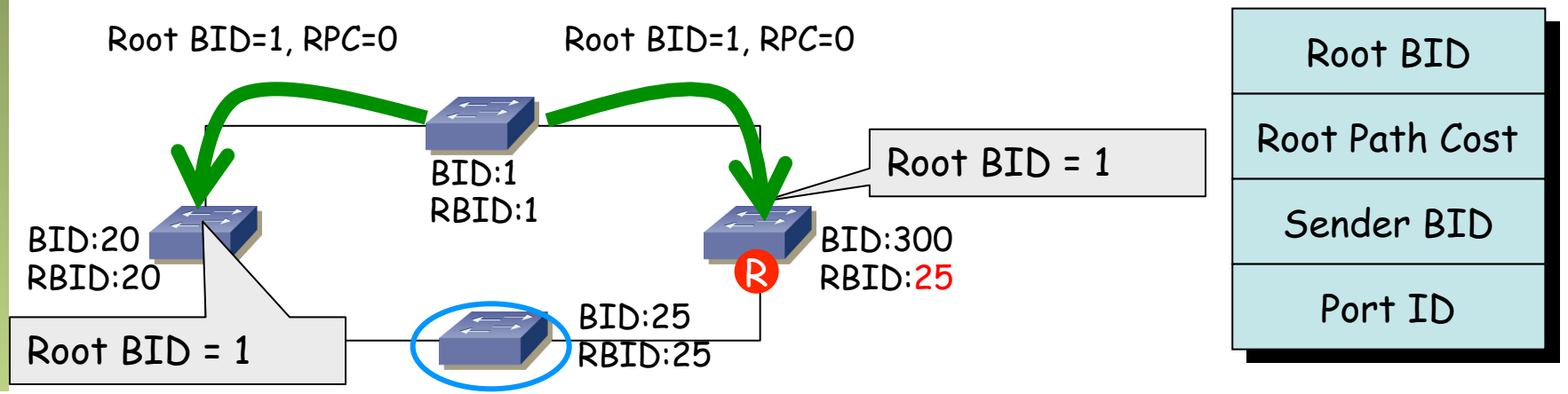
- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - Tiene el mismo valor de Sender BID y de Port ID que los guardados como Designated BID y Port ID (nos lo manda el mismo)
  - o si Root BID en el mensaje < Root BID guardado (cambiamos de puente raíz) (...)





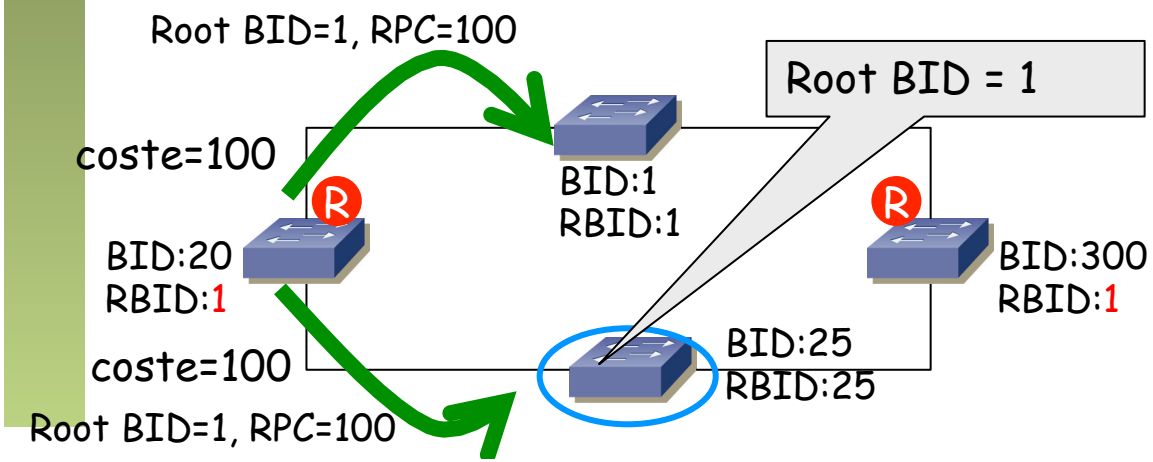
Comparación de costes

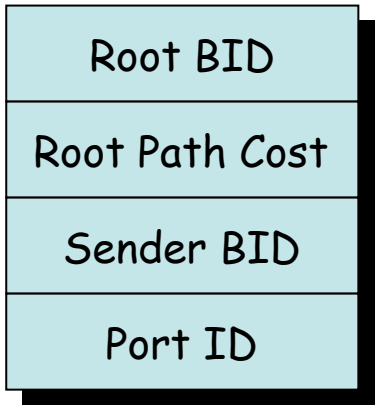
- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - Tiene el mismo valor de Sender BID y de Port ID que los guardados como Designated BID y Port ID (nos lo manda el mismo)
  - o si Root BID en el mensaje < Root BID guardado (cambiamos de puente raíz) (...)



Comparación de costes

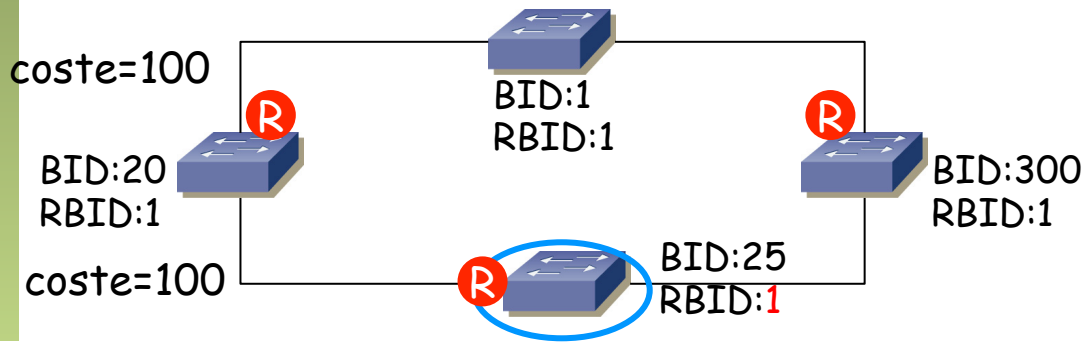
- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - Tiene el mismo valor de Sender BID y de Port ID que los guardados como Designated BID y Port ID (nos lo manda el mismo)
  - o si Root BID en el mensaje < Root BID guardado (cambiamos de puente raíz) (...)

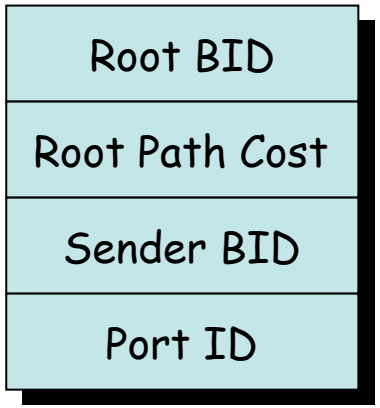




Comparación de costes

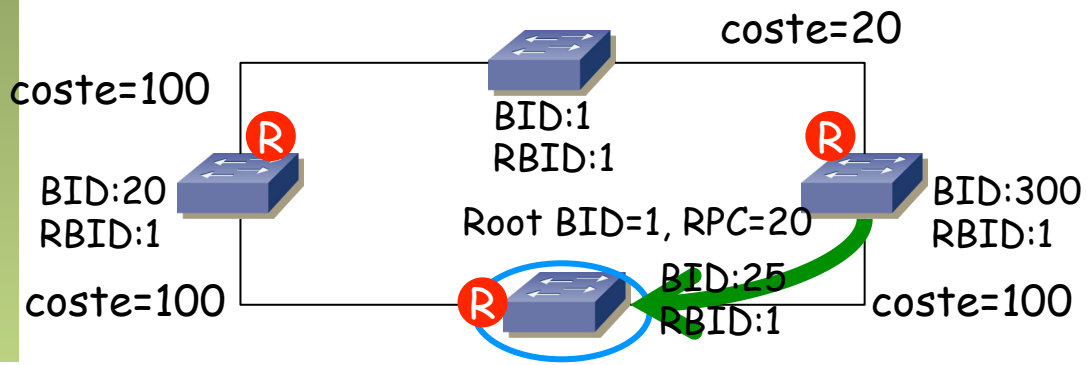
- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - Tiene el mismo valor de Sender BID y de Port ID que los guardados como Designated BID y Port ID (nos lo manda el mismo)
  - o si Root BID en el mensaje < Root BID guardado (cambiamos de puente raíz)
  - o (...)

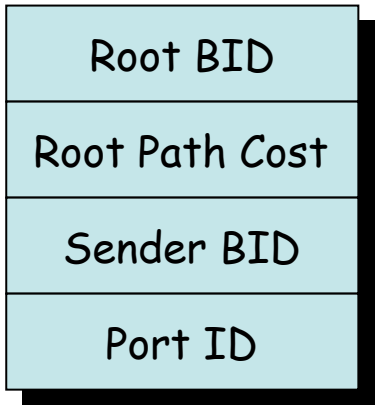




Comparación de costes

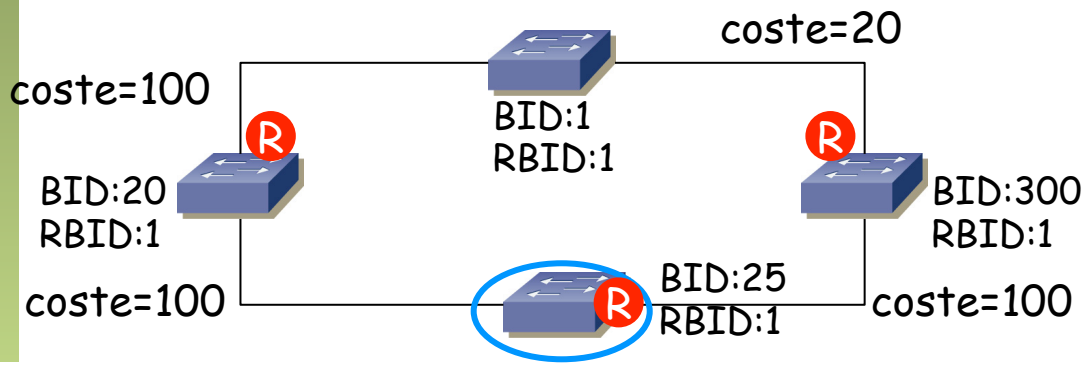
- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - Tiene el mismo valor de Sender BID y de Port ID que los guardados como Designated BID y Port ID (nos lo manda el mismo)
  - o si Root BID en el mensaje < Root BID guardado (cambiamos de puente raíz)
  - o si son iguales y el Root Path Cost (+coste puerto) del mensaje es < que el guardado (nos llega un anuncio con menos coste a la raíz) (...)

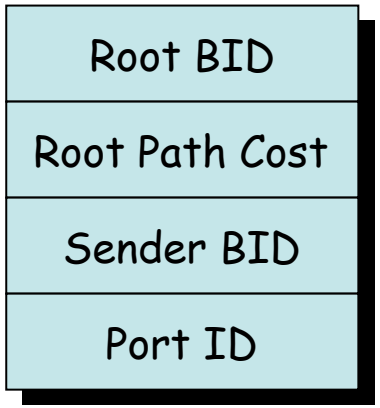




Comparación de costes

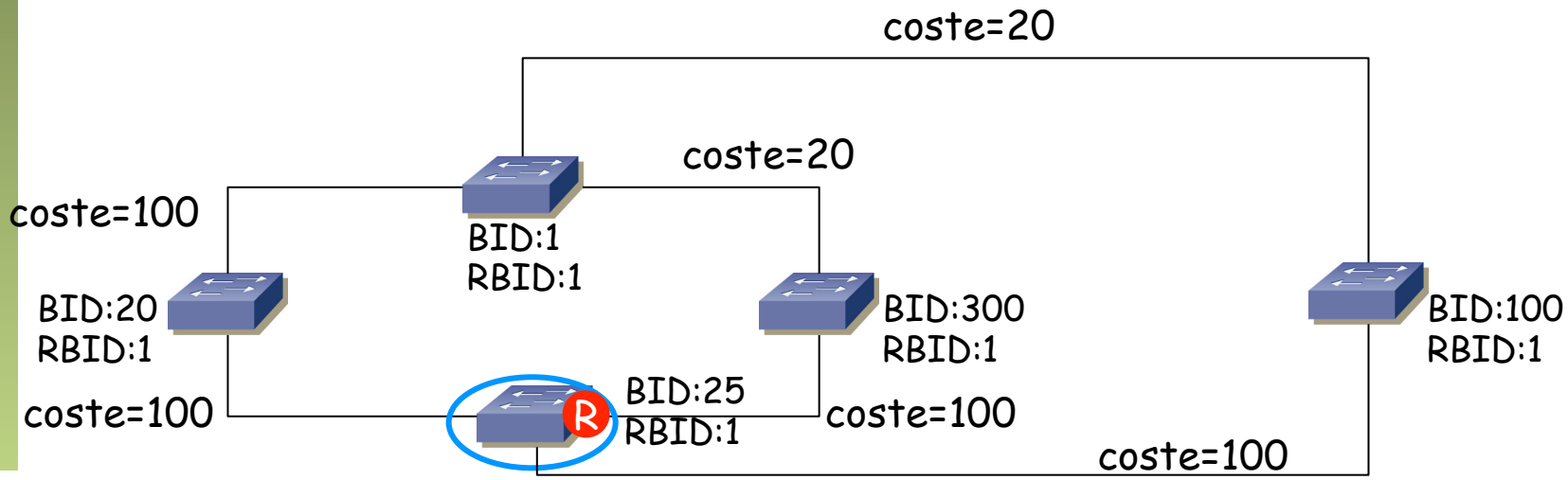
- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - Tiene el mismo valor de Sender BID y de Port ID que los guardados como Designated BID y Port ID (nos lo manda el mismo)
  - o si Root BID en el mensaje < Root BID guardado (cambiamos de puente raíz)
  - o si son iguales y el Root Path Cost (+coste puerto) del mensaje es < que el guardado (nos llega un anuncio con menos coste a la raíz)
  - o (...)





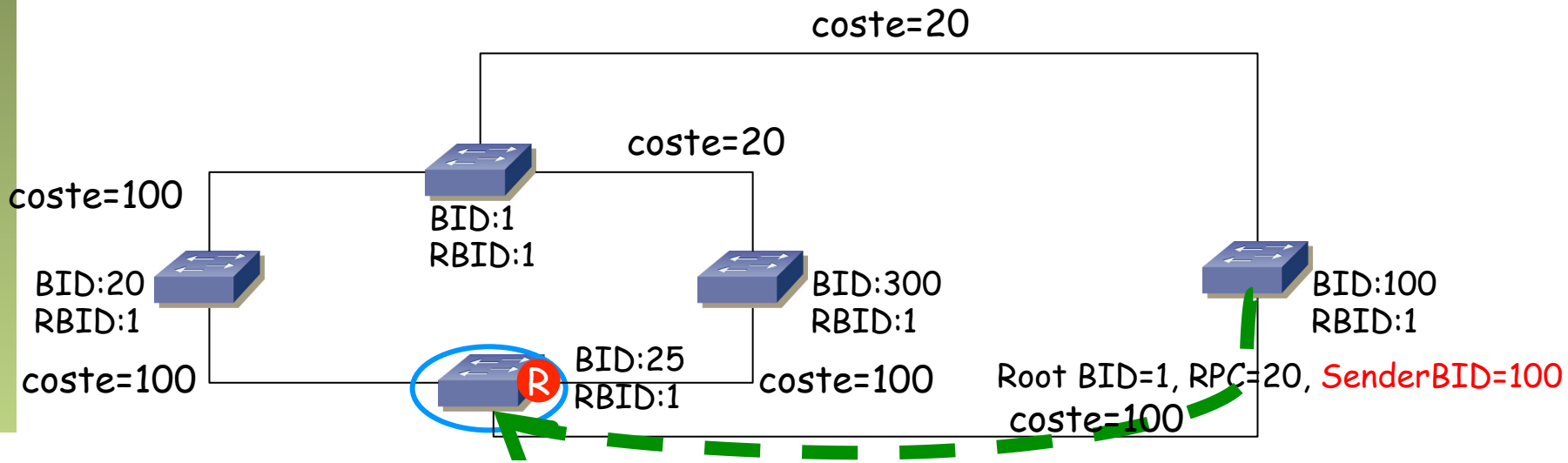
Comparación de costes

- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - [...]
  - o si los dos anteriores son iguales y el BID del puente que lo envía es < que el guardado (llega un anuncio igual pero de otro puente con menor BID)
  - o (...)



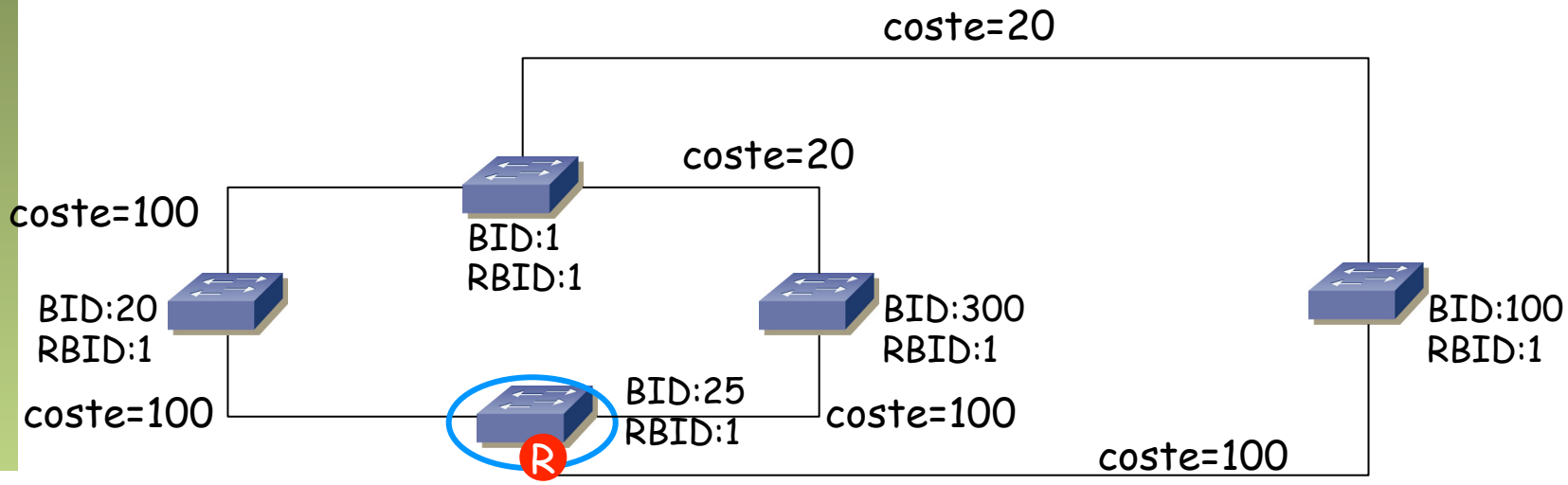
Comparación de costes

- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - [...]
  - o si los dos anteriores son iguales y el BID del puente que lo envía es < que el guardado (llega un anuncio igual pero de otro puente con menor BID)
  - o (...)



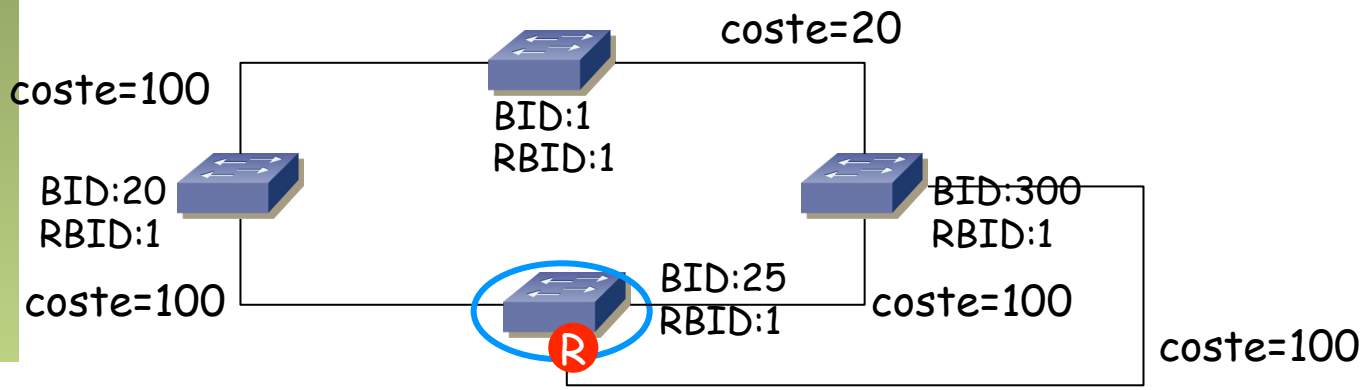
Comparación de costes

- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - [...]
  - o si los dos anteriores son iguales y el BID del puente que lo envía es < que el guardado (llega un anuncio igual pero de otro puente con menor BID)
  - o (...)

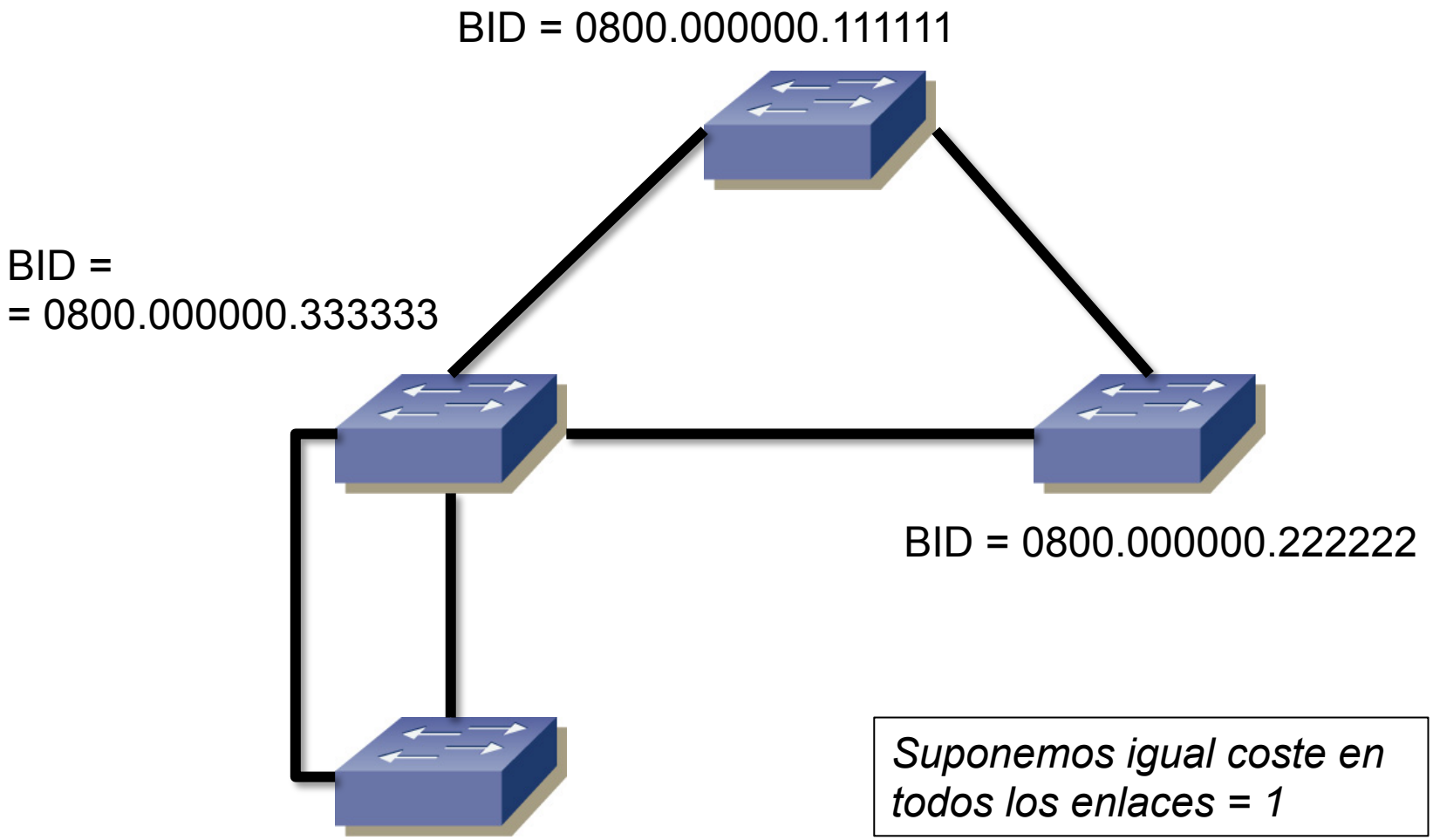


Comparación de costes

- Cada puerto del conmutador guarda un vector con:
  - Root BID, Root Path Cost, Designated BID, Designated Port ID, Bridge Port ID
- Cuando ese puerto recibe un mensaje lo compara con el vector guardado para sustituirlo o no
- Lo sustituye si:
  - [...]
  - o si los tres anteriores son iguales y el port ID en el mensaje es < que el guardado.
- Los envíos del conmutador toman el "mejor" vector (sumando el coste de cada puerto a cada uno)



Próximo día: ¿Roles, árbol?



Suponemos igual coste en todos los enlaces = 1