

Spanning Tree Protocol

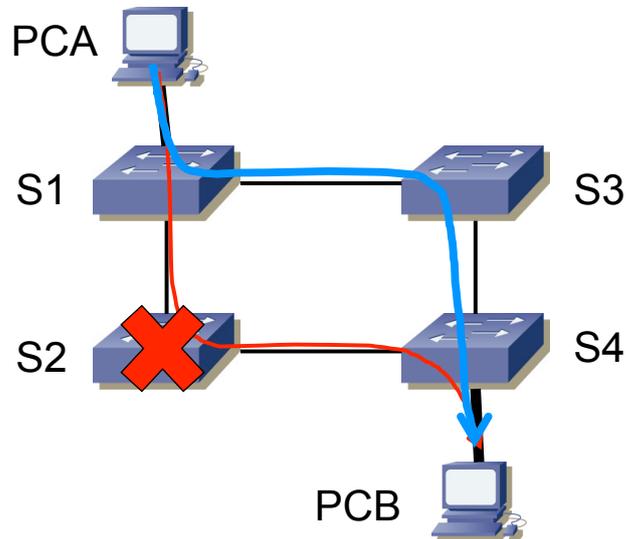
Area de Ingeniería Telemática
<http://www.tlm.unavarra.es>

Grado en Ingeniería en Tecnologías de
Telecomunicación, 3º

Bucles en LANs con puentes

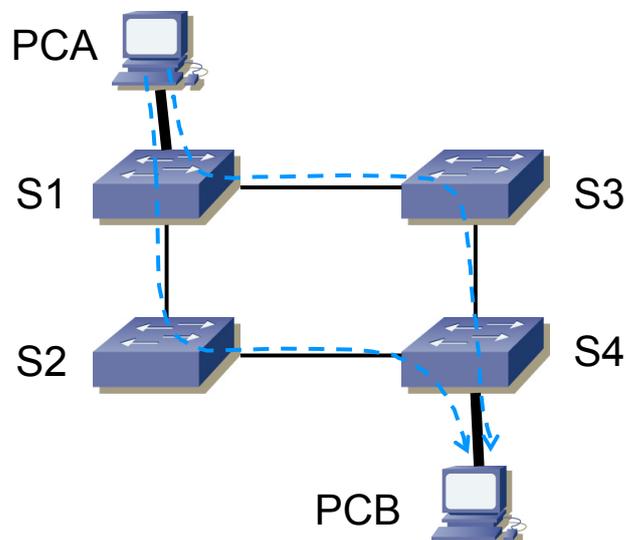
Caminos alternativos

- Ofrecerían la posibilidad de:
 - Reconfiguración ante fallos (...)
 - (...)



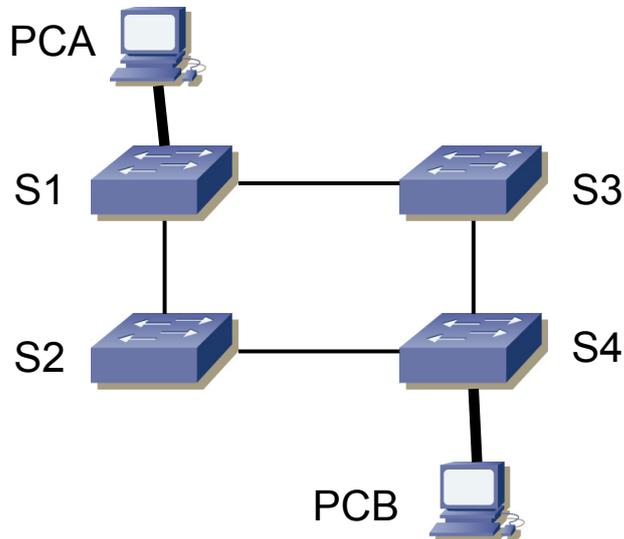
Caminos alternativos

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



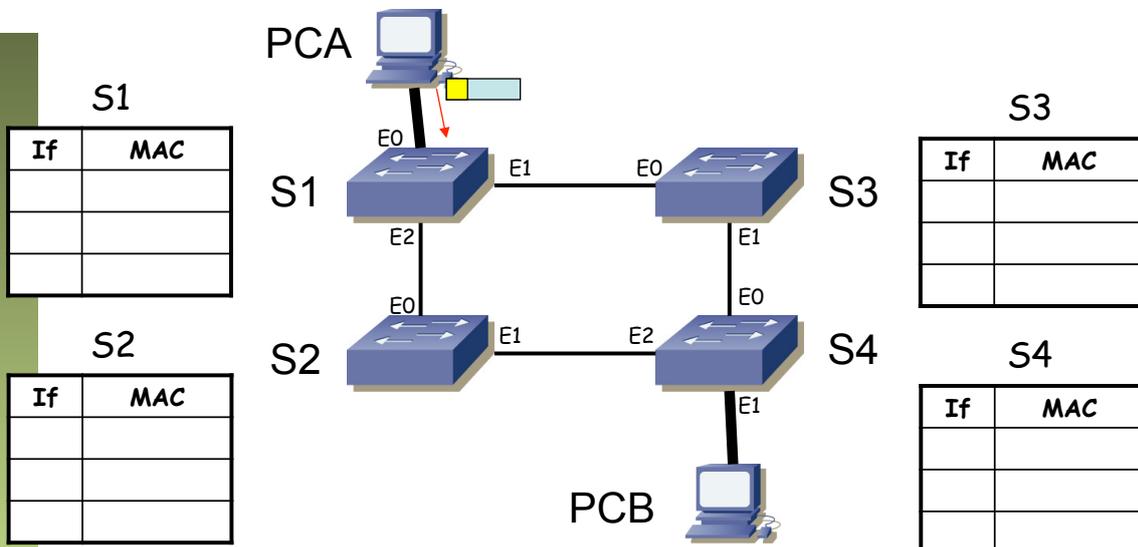
Pero...

- Ejemplo:
 - PCA envía una trama al PCB y los switches no han aprendido la MAC de PCB
 - O PCA envía una trama a broadcast
 - En ambos casos habrá inundación



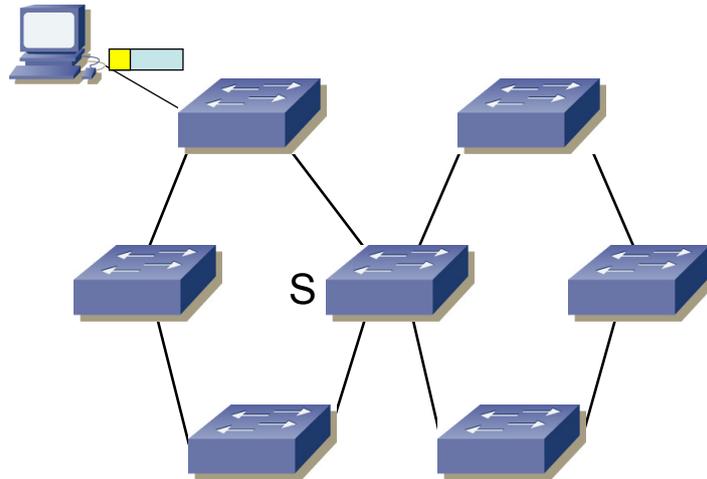
Pero...

- Ejemplo:
 - PCA envía una trama al PCB y los switches no han aprendido la MAC de PCB
 - O PCA envía una trama a broadcast
 - En ambos casos habrá inundación



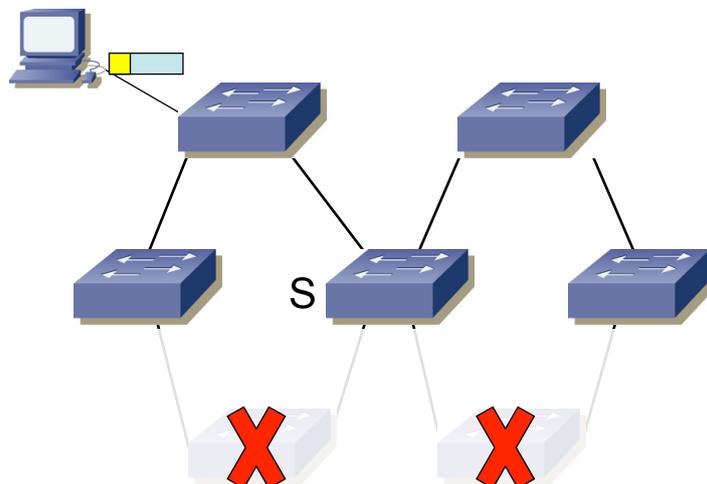
Ejemplo

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



Soluciones

- ¡ Emplear una topología sin ciclos !



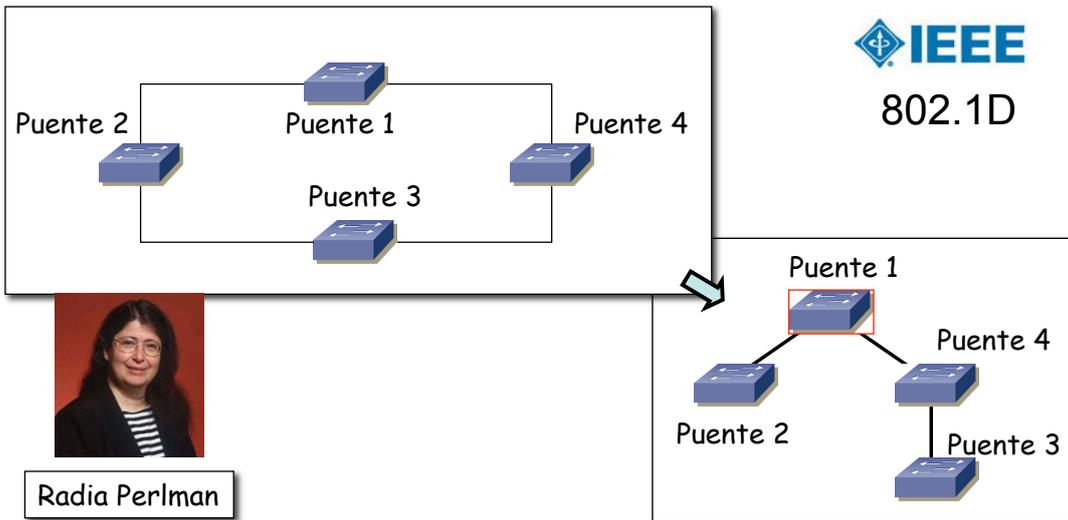
Soluciones

- ¡ 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 (STP)

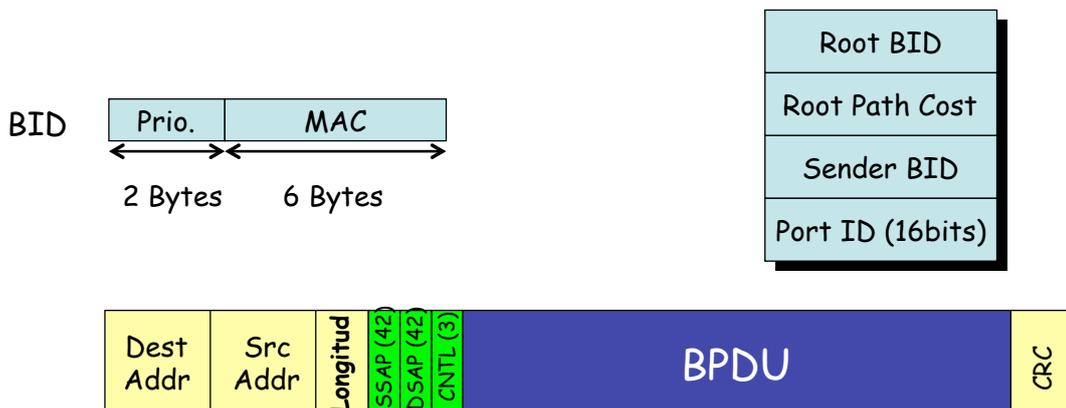
- Calcula una topología libre de ciclos
- A partir del grafo de la topología crea un árbol
- ¿Cómo? Seleccionan un puente como “raíz” y desde él calculan un árbol
- Desactivan los enlaces que no están en el árbol



Radia Perlman

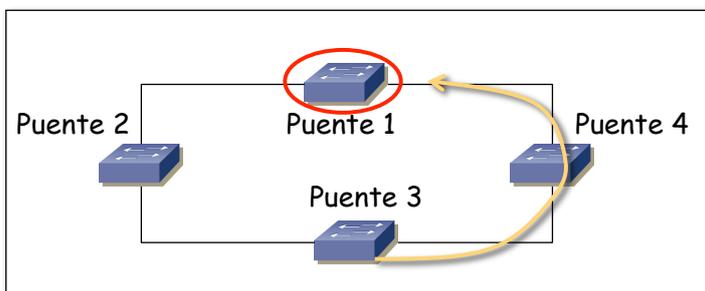
STP: BPDUs

- 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 = { Prioridad, MAC }
- Prioridad (default: 32768) en múltiplos de 4096



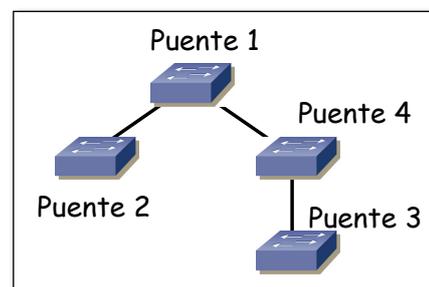
STP: BPDUs

- Se pueden “comparar” entre sí 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

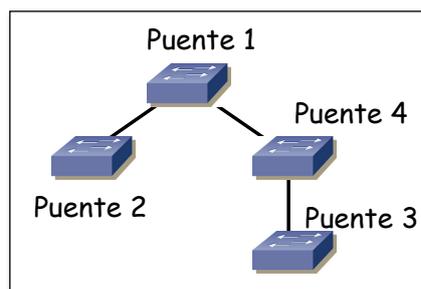
- Los conmutadores siguen enviando BPDUs tras calcular árbol
 - Un conmutador no tiene forma de saber que se ha alcanzado una topología estable
 - Seguir enviando y aceptando BPDUs permite recalcular el árbol ante fallos
- El plano de datos del conmutador no cambia
 - Es decir, los conmutadores siguen 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

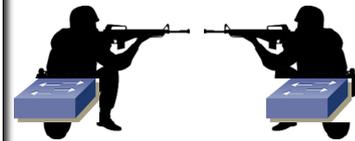
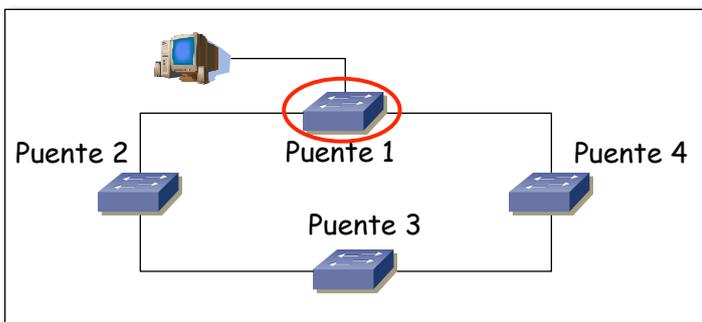
- Vamos a intentar entender cómo se calcula el árbol
- No nos interesa el “transitorio” sino solamente el estado final
- El mecanismo es básicamente *distance-vector* (Bellman-Ford distribuido)



STP: Root Bridge

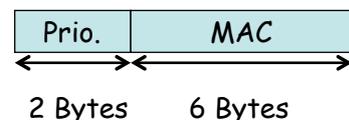
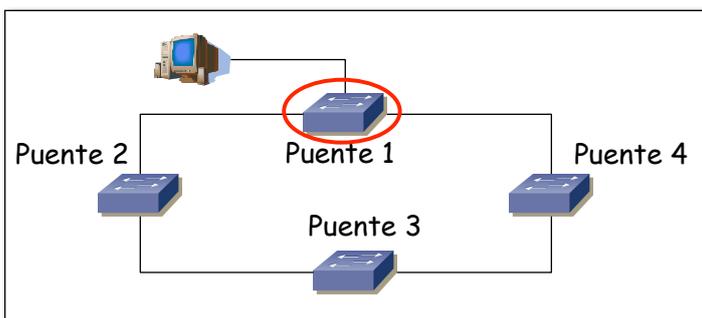
Selección de un *Root Bridge* (Root War !!!)

- 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 = 32768)
 - 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 o administrativamente
- 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
- 802.1D-1998:

Table 8-5—Path Cost Parameter Values

Parameter	Link Speed	Recommended value	Recommended range	Range
Path Cost	4 Mb/s	250	100–1000	1–65 535
Path Cost	10 Mb/s	100	50–600	1–65 535
Path Cost	16 Mb/s	62	40–400	1–65 535
Path Cost	100 Mb/s	19	10–60	1–65 535
Path Cost	1 Gb/s	4	3–10	1–65 535
Path Cost	10 Gb/s	2	1–5	1–65 535

- ¿Y si tuviéramos un enlace de 2x10Gbps? ¿Coste 1?
- ¿Y si tuviéramos un enlace de 4x10Gbps?

STP: Path Cost

- Estos valores recomendados cambian en 802.1t
- 802.1D-2004 (desde 802.1t-2001):
 - Recomendado $2 \times 10^7 / \text{Link_Speed_en_Mbps}$
 - Podemos acumular al menos 20 saltos del peor coste sin exceder el máximo de un contador de 32bits

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.

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 y si se aprenden direcciones MAC
 - Los *roles* definen el papel que juega el puerto en el árbol

RSTP

Rapid Spanning-Tree Protocol

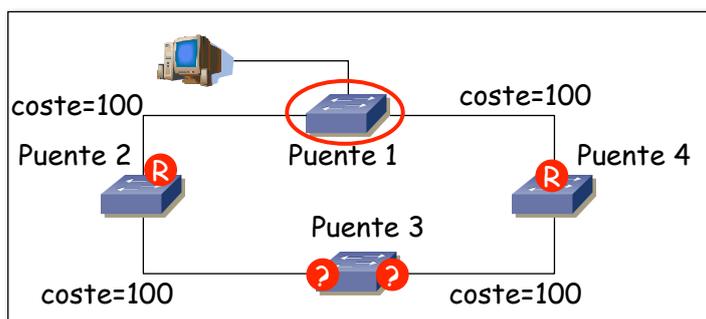
- STP obsoleto y retirado del estándar
- RSTP es IEEE 802.1w
- RSTP es el STP que aparece en 802.1D-2004
- Tiempos de convergencia de 2-3 segs (aunque según la topología puede llegar a 30s y cuentas a infinito)
- Tres **estados** posibles para un puerto:
 - *Discarding*: ni envía ni acepta paquetes de 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 ni cómo se adapta a cambios
- 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

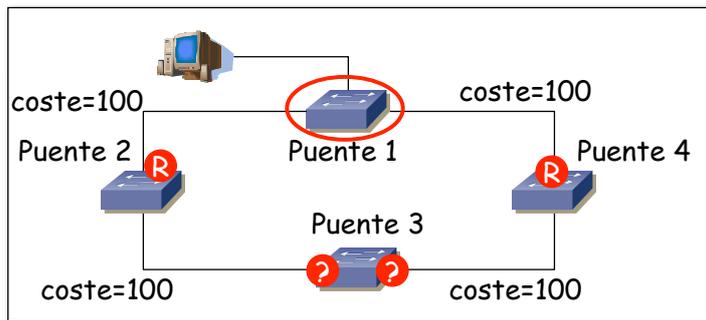
- Uno en cada puente salvo en el puente raíz
- El puente raíz es el único sin un puerto raíz
- Es el puerto de un conmutador que tiene el menor *Root Path Cost+Port Cost* (menor coste hasta la raíz)
- En este ejemplo supongamos que todos los puertos tienen el mismo coste
- Está claro cuál es *root port*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



RSTP: Port Roles

¿Empates?

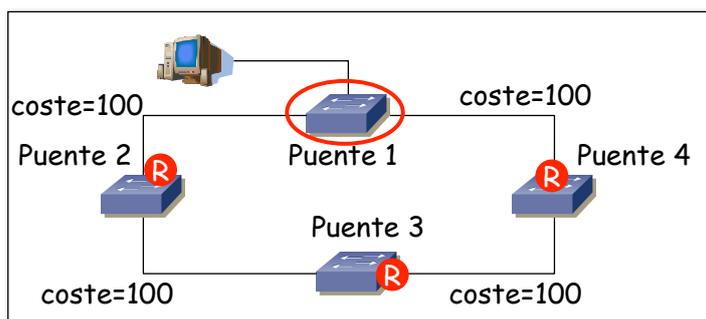
- En el puente 3 los costes que recibe de puente 2 y de puente 4 hasta la raíz son el mismo = 100
- A cada uno le suma el coste del puerto por el que lo ha recibido y empatan (le sale 200 en ambos)
- Entonces se comparan los BID de los puentes que hacen el anuncio
- El anuncio que venga del BID menor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



RSTP: Port Roles

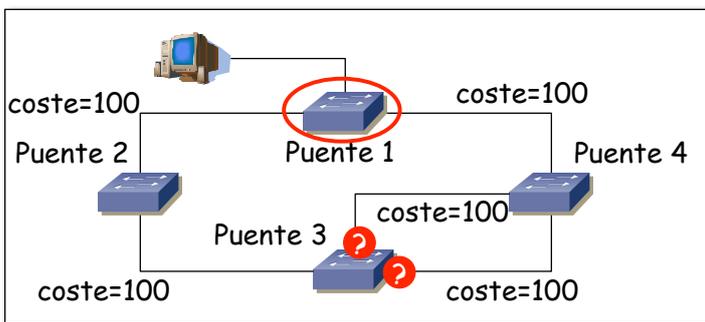
Desempate

- 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

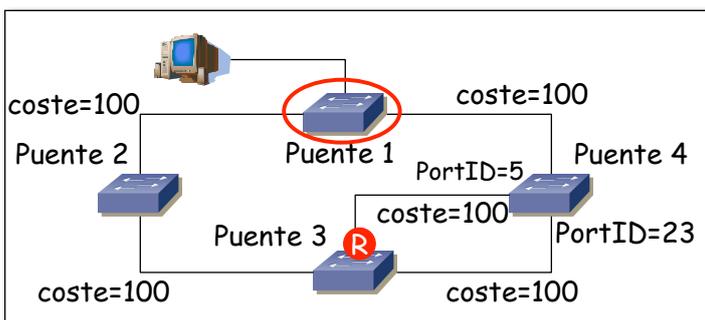
- Hemos dicho que cuando hay que elegir un camino a la raíz se toma el menor coste agregado
- Si hay empate el menor BID
- ¿Hay más posibilidades de empate?
- En este caso empatarían los puertos a puente 4 (suponiendo que el BID del puente 4 es menor que el del puente 2)
- ¿Desempate? (...)



Root BID
Root Path Cost
Sender BID
Port ID

RSTP: Port Roles

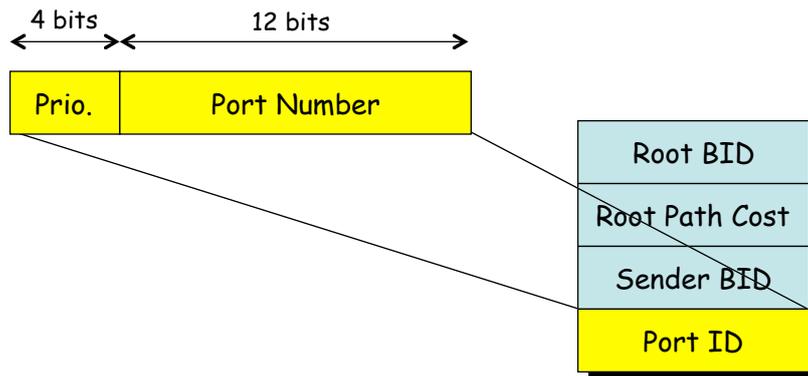
- Hemos dicho que cuando hay que elegir un camino a la raíz se toma el menor coste agregado
- Si hay empate el menor BID
- ¿Hay más posibilidades de empate?
- En este caso empatarían los puertos a puente 4 (suponiendo que el BID del puente 4 es menor que el del puente 2)
- ¿Desempate? Menor identificador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



Root BID
Root Path Cost
Sender BID
Port ID

Port ID

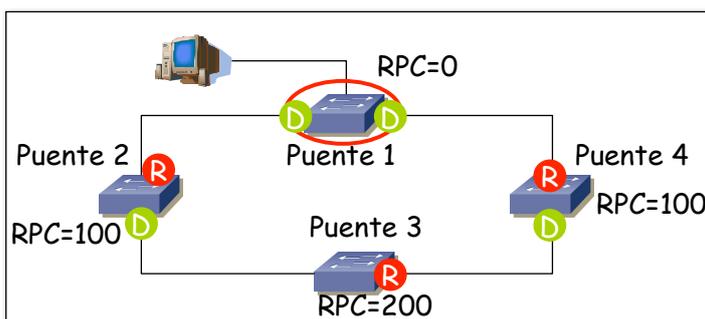
- Es un valor de 16bits
- Tiene una componente de prioridad y un número de puerto
- Es decir, el valor de prioridad, si lo entendemos como el primer byte ignorando los últimos 4 bits, va en múltiplos de 16



RSTP: Port Roles

Designated Port

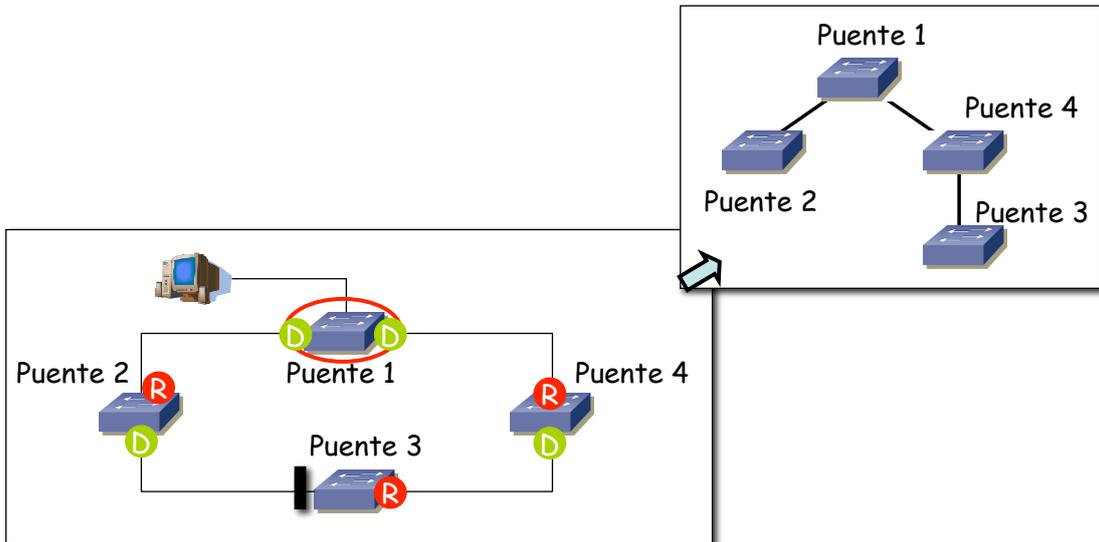
- En un segmento de LAN puede haber varios puertos de conmutador
- El puerto de conmutador en un segmento de LAN con menor *Root Path Cost+Port Cost* es el puerto designado para la LAN
- Uno por segmento de LAN
- Si hay empate por costes se desempata por el BID



RSTP: Port Roles

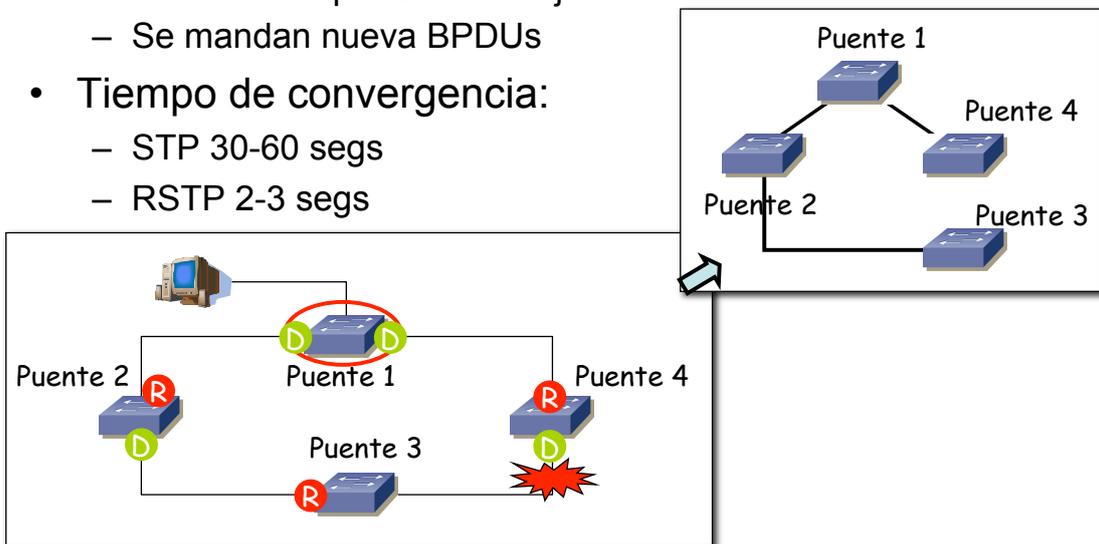
Blocked Port

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



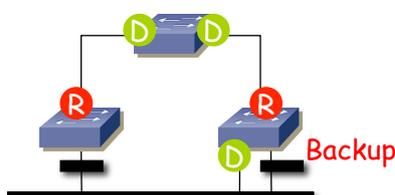
RSTP: cambios en la topología

- Ante un fallo se recalcula el árbol (...)
- ¿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

