UD3. STP

Índice

Tormentas de broadcast

Una tormenta de difusión, en redes informáticas, es una situación que puede darse cuando se transmiten tramas de difusión o broadcast en la red, cada una de las cuales requiere que el nodo receptor responda reenviando su mensaje. La posible consecuencia es un aumento exponencial del tráfico de red, que conduce a una saturación completa de los recursos de red disponibles o, en cualquier caso, a una disminución drástica del rendimiento. Por tanto, es evidente que debe evitarse, por todos los medios posibles, que se puedan producir las tormentas de difusión.

A veces al interconectar conmutadores se producen bucles, es decir hay más de un camino posible entre dos redes. Estos bucles pueden hacerse por error o porque se quiere disponer de varios caminos para tener mayor fiabilidad y tolerancia a fallos. Además las tablas MAC se cambian continuamente. Esto se debe a dos características de los puentes transparentes

Proceden por inundación cuando la dirección de destino no está en su tabla de direcciones

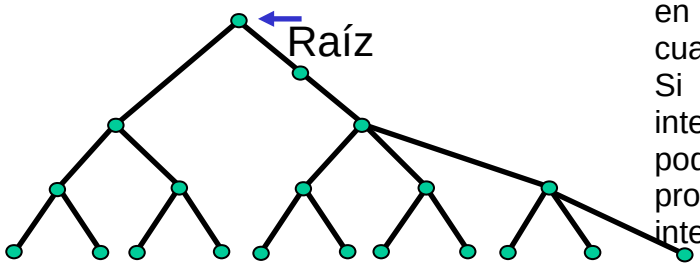
Cuando reenvían una trama la copia es indistinguible del original. No existe ningún campo (p. ej. un contador de saltos) que permita diferenciar las sucesivas copias.

Existen dos posibles estrategias:

Se prohíbe taxativamente la creación de redes con bucles

Se habilita algún mecanismo, por software, que permita a los conmutadores detectar la presencia de bucles para que desactiven las interfaces necesarias para que no haya bucles

Spanning Tree



Un Spanning Tree, o árbol de expansión, es un grafo en el que hay uno y solo un camino posible entre cualquier par de nodos (un árbol sin bucles).
Si podemos pintar una red de conmutadores interconectados como un spanning tree, entonces podemos asegurar que no hay bucles. El objetivo del protocolo Spanning Tree es desactivar lógicamente interfaces para conseguir siempre un spanning tree.

STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/ Transitional
Blocking	NO/YES	NO	NO	Stable
Listening	YES/YES	NO	NO	Transitional
Learning	YES/YES	NO	YES	Transitional
Forwarding	YES/YES	YES	YES	Stable
Disabled	NO/NO	NO	NO	Stable

Funcionamiento de STP

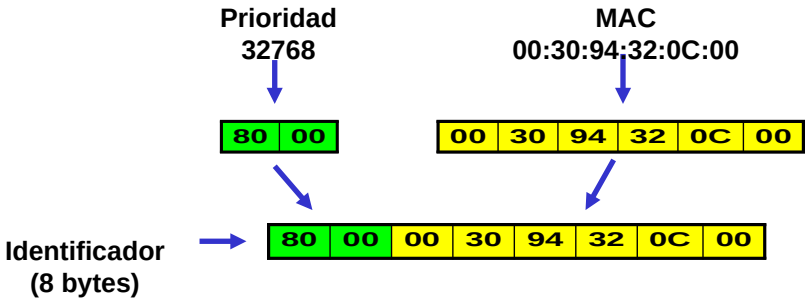
Los conmutadores intercambian regularmente información sobre la topología de la red. Los mensajes que utilizan se denominan BPDUs (Bridge Protocol Data Units).

Las BPDUs emplean un Ethertype propio y se envían a una dirección multicast reservada, la 01-80-C2-00-00-00. Así se asegura que se identifican fácilmente y que los conmutadores sin ST los propagarán de forma transparente.

Cada conmutador dispone de un identificador único (ID) que crea a partir de una dirección MAC globalmente única que le ha asignado el fabricante. Además cada puerto del conmutador recibe un identificador y tiene asociado un costo.

El ID se construye a partir de una prioridad (configurable) y de la dirección MAC canónica del conmutador (fija). La prioridad puede valer entre 0 y 65535. Por defecto es 32768

STP Timer	Purpose	Duration
Hello	How often the root bridge sends hello BPDUs	2sec
Forward delay	How long the switch will stay in the Listening and Learning states (each state is 15 seconds = total 30 seconds)	15sec
Max Age	How long an interface will wait to change the STP topology after ceasing to receive Hello BDPUs. The timer is reset every time a BPDU is received.	20sec (10* hello)



Velocidad	Costo
4 Mb/s	250
10 Mb/s	100
16 Mb/s	62
45 Mb/s	39
100 Mb/s	19
155 Mb/s	14
200 Mb/s	12
622 Mb/s	6
1 Gb/s	4
2 Gb/s	3
10 Gb/s	2

Si se usa siempre la prioridad por defecto el conmutador con la MAC más baja es elegido raíz. Si a un conmutador le ponemos prioridad 32767 y dejamos el valor por defecto en el resto ese será seguro el de ID más bajo, y por tanto será elegido raíz. Los BPDU se mandan a la MAC 01:00:0c:cc:cc:cd

Al principio los switches empiezan eligiéndose a sí mismos como raíz y comunicando lo que saben por todos los puertos. Cuando un switch ve a otro que tiene una prioridad mejor deja de proclamarse a sí mismo como raíz y anunciará la nueva raíz en pasos siguientes.

Cada conmutador calcula el grafo de la red y observa si existe algún bucle; en ese caso se van desactivando interfaces siguiendo unas reglas claras hasta cortar todos los bucles y construir un spanning tree. Los conmutadores eligen como raíz del árbol a aquel que tiene el ID más bajo. Todos eligen al mismo.

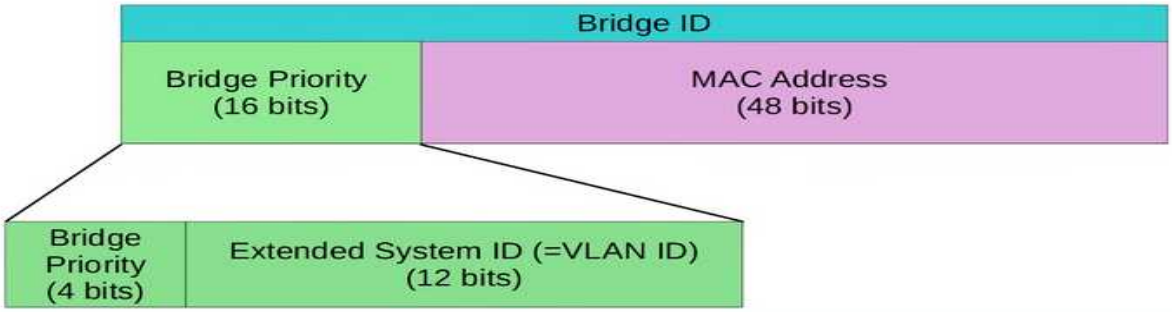
Cada conmutador envía BPDUs por sus interfaces indicando su ID, el ID del conmutador raíz y el costo de llegar a él; los mensajes se van propagando por toda la red; cada conmutador al reenviar los mensajes de otros les suma el costo de la interfaz por la que los emite.

Puerto raíz: es un puerto que indica que es el mejor camino para llegar a la raíz.
Puerto designado: es un puerto no raíz que es el mejor del segmento para llegar a la raíz.
Puerto bloqueado: en un segmento es un puerto que no se usa.

Ver el estado actual de STP podemos ejecutar este comando en modo administrador:

```
Switch#show spanning-tree
Switch#show spanning-tree detail
```

Calculo de BID (Bridge ID)



Los switches CISCO usan una versión de STP llamada PVST+ (Per-VLAN Spanning Tree + dot1q) que corre una instancia separada por cada VLAN, así cada VLAN puede tener distintas interfaces en estado forwarding/blocking

La prioridad por defecto es 32768 que hace un BID por defecto para la VLAN 1 de 32769

La prioridad solo puede ser cambiada en unidades de 4096.

Los valores válidos serán entonces 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, 61440

Bridge Priority				Extended System ID (VLAN ID)											
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bridge Priority				Extended System ID (VLAN ID)											
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1

= 28673 (16384 + 8192 + 4096 + 1)

Activar STP (si no está ya activada)

```
Switch (config)# spanning-tree vlan <numero>
Switch (config)# spanning-tree mode rapid-pvst
```

Desactivar STP

```
Switch (config)# no spanning-tree vlan <numero>
```

Poner un Switch concreto como raiz poniendo su prioridad como 24576 y 28672. Si recibe una BPDU con un id menor se configura con 4096 menos que lo que reciba para seguir siendo el root

```
switch (config)# spanning-tree vlan <numero> root primary[secondary]
```

Cambiar la prioridad de un switch:

```
Switch> enable
Switch# configure terminal
Switch(config)# spanning-tree vlan <numero> priority <numero>
Switch(config)# spanning-tree vlan <numero>,<numero> priority <numero>
```

Con las BPDUs recibidas cada conmutador calcula por que puerto puede llegar él al raíz al mínimo costo. Ese es su puerto raíz. En caso de empate elige el puerto de ID más bajo.

Cada LAN tiene un puerto designado, que es aquel por el que esa LAN accede al conmutador raíz al mínimo costo.

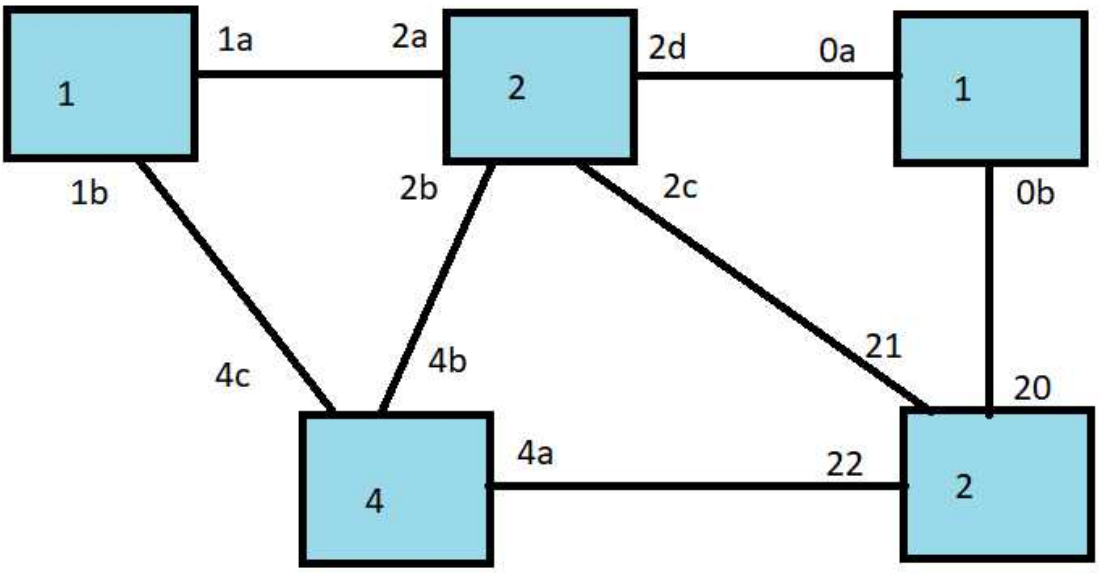
Los puertos que no son ni raíz ni designados son puertos bloqueados. Esos puertos son innecesarios para la comunicación y si se les deja funcionar provocan bucles

Así, el proceso es más o menos este:

- El switch raíz pone todos sus puertos a **designado**. Es lógico, el switch raíz es el mejor y sus puertos son siempre los mejores de cualquier segmento.
- El resto tienen que ir puerto por puerto examinando lo que han enviado y lo que han recibido. Este proceso es a su vez más complicado y lo desglosamos aparte.

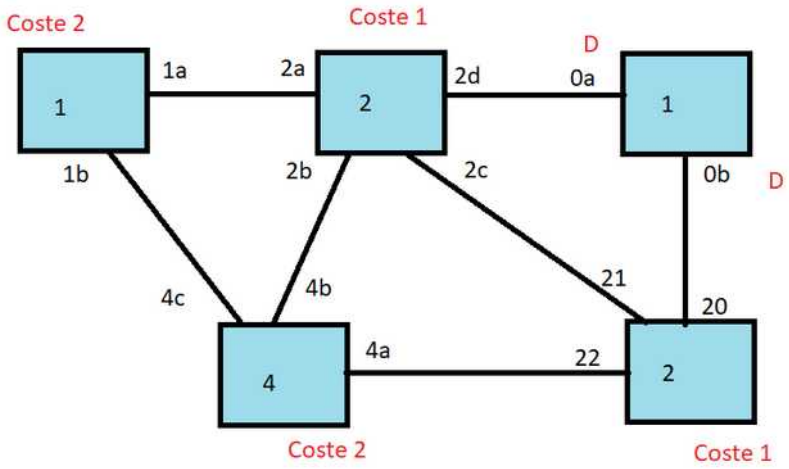
1. Si un puerto tiene un coste mejor que el otro puerto del segmento se pone a «designado».
2. Si un puerto tiene igual coste que el otro puerto del segmento se examinan las prioridades y se elige el puerto que lleve al switch con la mejor prioridad.
3. Los puertos que queden son puertos que se ponen a «bloqueado». Han perdido frente a sus competidores, ya sea por coste o por MAC.

En la figura siguiente se observa una red de switches. Podemos ver en su interior las prioridades que se les han dado y en los cables podemos ver la MAC de cada interfaz. Observando esto, ¿en qué estado quedarán los distintos puertos de los distintos switches?



Switch 1 (derecha), raíz

En primer lugar debemos saber quien actuará como raíz. Despues del proceso de elecciones ocurrirá que el switch 1 de la derecha ganará el proceso, ya que aunque tiene la misma prioridad que otro, el switch 1 de la derecha tiene la menor MAC (tiene la 0a). Esto significa que el proceso empieza declarando al switch 1 de la derecha el switch raíz y poniendo todos sus puertos a «designado». Todos los switches toman nota del coste que les supone llegar a la raíz.



A continuación iremos viendo los distintos switches y las decisiones que toman

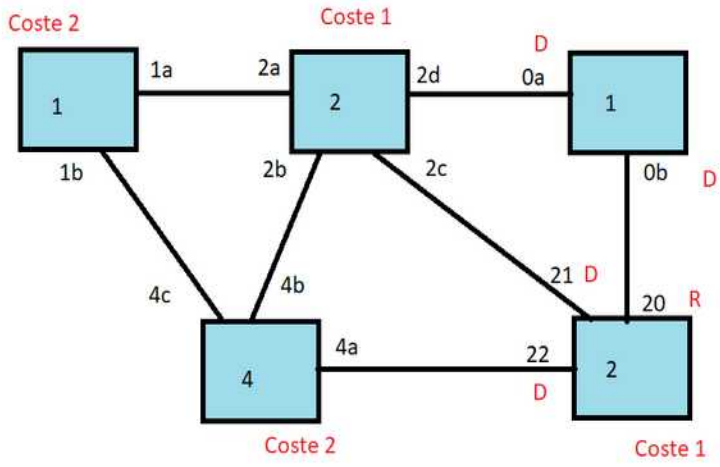
Switch 2 (abajo)

Ocurre esto:

Todos los switches deben empezar indicando su puerto raíz. En este caso, su mejor puerto para llegar a la raíz es el 20

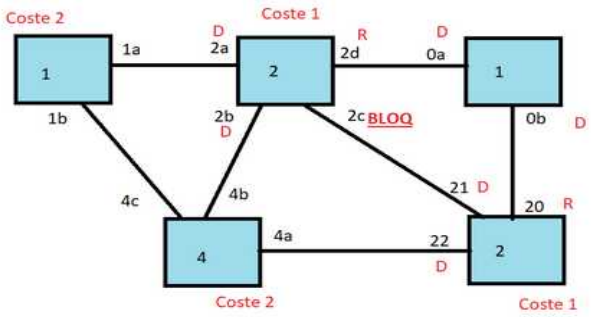
Examinamos el puerto 21. Este switch tiene un coste 1 y el switch de «enfrente» también, pero nuestra mac 21 es mejor que la del vecino (que es 2c), así que ponemos este puerto a «designado».

Examinamos el puerto 22. Nuestro coste es mejor que el del vecino de enfrente, así que nuestro puerto «gana» y se pone a «designado».



Switch 2 (centro-arriba)

El puerto 2d es el mejor. Se declara puerto raíz.



El puerto 2c se compara con el vecino de enfrente. El vecino y nosotros tenemos el mismo coste (que es 1) pero nuestra MAC es peor. Perdemos y bloqueamos este puerto con la MAC 2c.

El puerto 2b se compara con el vecino de enfrente. El vecino (switch 4) tiene un coste peor, así que él pierde y declaramos este puerto 2b como «designado».

El puerto 2a se compara con el vecino de enfrente. El vecino (switch 1, izquierda) tiene un coste peor. Él pierde y declaramos nuestro puerto 2a como «designado».

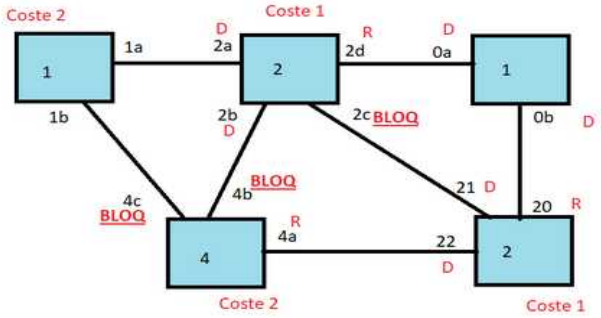
Switch 4

Examina sus propios puertos:

Su puerto 4a es el mejor, se declara raíz.

Examina su puerto 4b. El vecino de enfrente tiene un coste 1 y nosotros 2. Perdemos y declaramos este puerto 4b como «bloqueado».

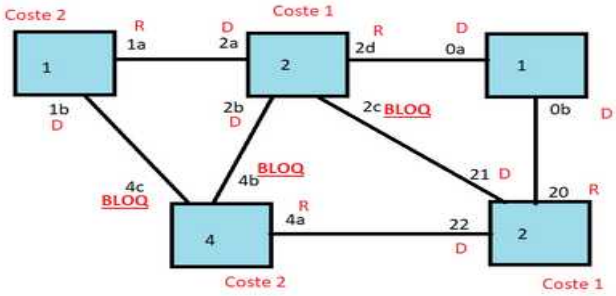
Examina su puerto 4c. El vecino tiene un coste 2 y nosotros también. Nuestra MAC 4c es peor que la suya (que es 1b), así que perdemos y declaramos este puerto como «bloqueado».



Switch 1 (izquierda)

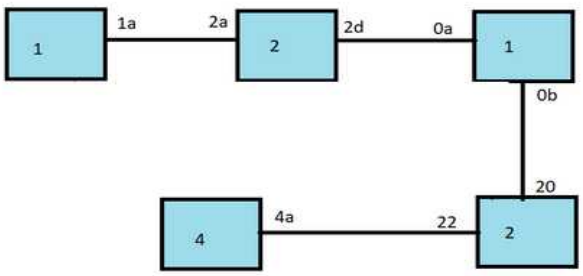
Su puerto 1a es el mejor para llegar a la raíz, así que se declara puerto raíz.

Su puerto 1b se compara con el vecino de enfrente. El vecino coste 2 y nosotros también, pero nuestra MAC es menor. Nuestro puerto 1b gana a su MAC 4c así que él pierde y nosotros ponemos nuestro puerto 1b a «designado».



Estado final

Si asumimos que los puertos bloqueados anulan el cable al que pertenecen observamos que la topología ha cambiado y queda algo como esto:



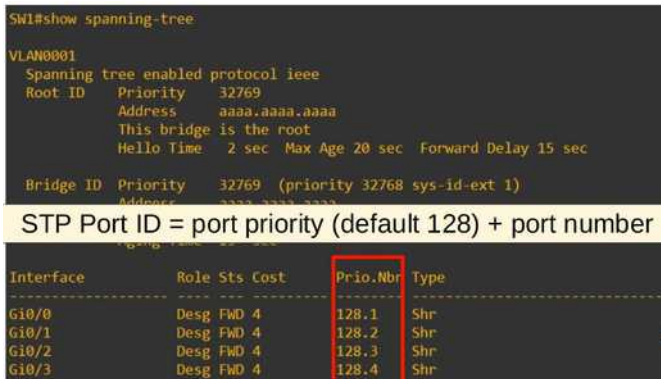
Pasos para averiguar la topología Spanning Tree

1. Asignar costos a todas las interfaces
2. Elegir el conmutador raíz (el de **BID** más bajo)
3. Elegir el puerto raíz de los demás conmutadores (el que les lleva al menor **costo** al puente raíz). En caso de empate elegir el puerto que lleva al switch con **BID** más bajo, en caso de empate elegimos el puerto que lleva al puerto con **PortID** más bajo del vecino. Este ID de puerto se calcula con una

   prioridad de puerto + número de puerto

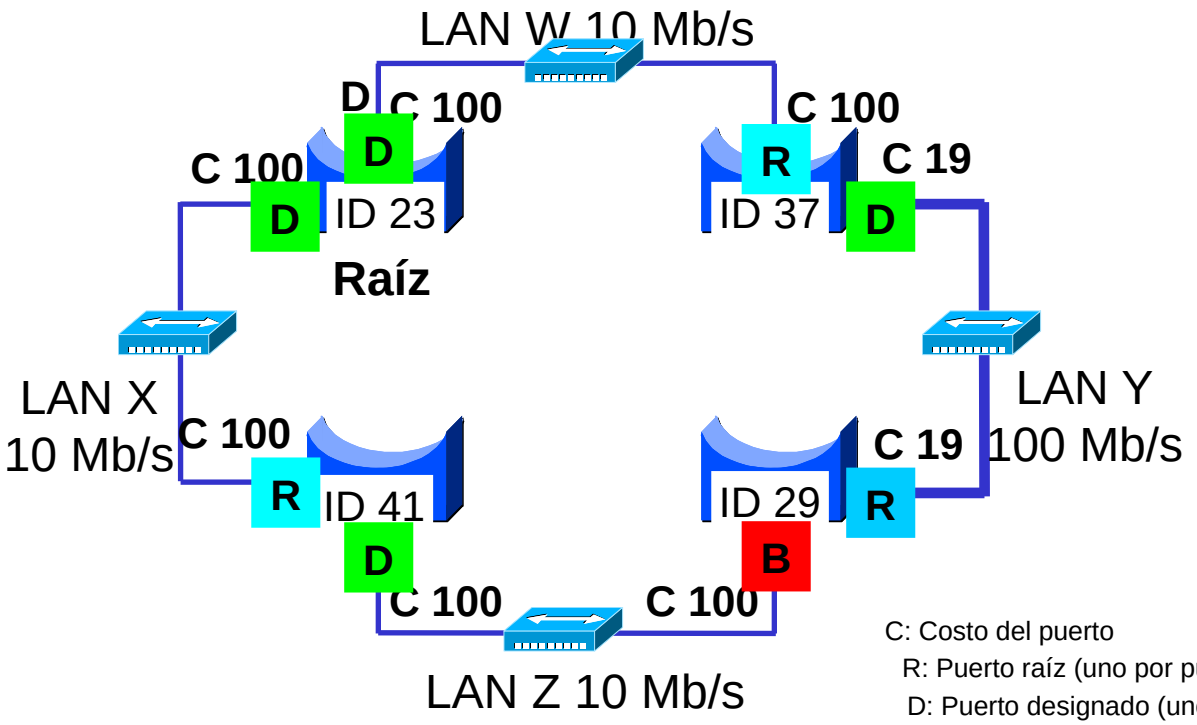
   Cambiar el coste de un puerto

```
      switch (config-if)# spanning-tree [vlan <numero>] cost <numero>
Cambiar la prioridad de un puerto:
      Switch(config-if)# spanning-tree [vlan <numero>] port-priority <numero>
```



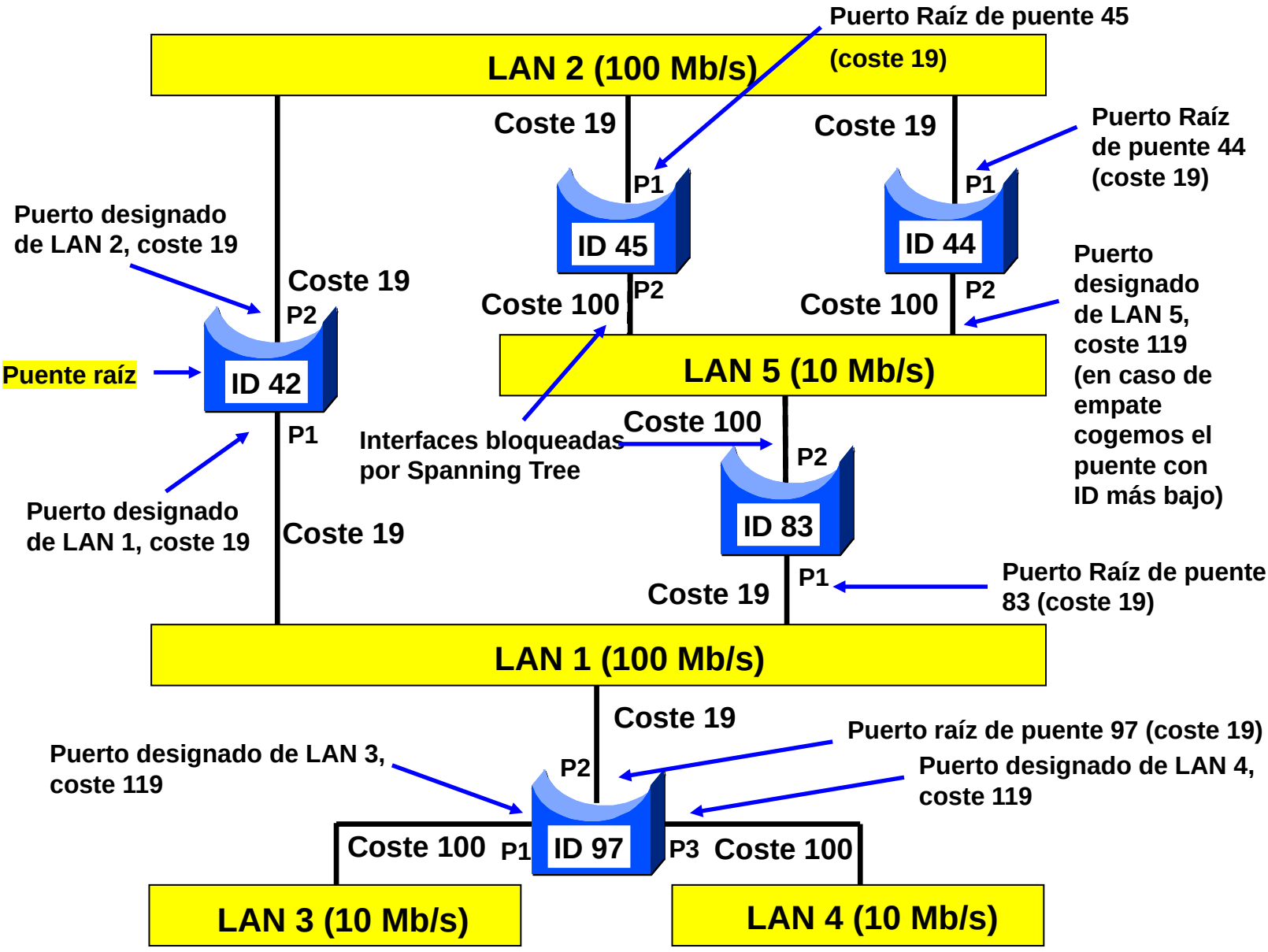
4. Elegir el puerto designado para cada enlace (el que le lleva al menor **costo** al puente raíz). Para ello elegimos la interfaz en el switch que tenga el menor coste al raíz. En caso de empate elegir el conmutador con **BID** más bajo.
5. Los puertos que no han sido elegidos como raíz ni como designados deben bloquearse

ejemplo:



C: Costo del puerto
R: Puerto raíz (uno por puente)
D: Puerto designado (uno por LAN)
B: Puerto bloqueado

Ejemplo de red con bucles

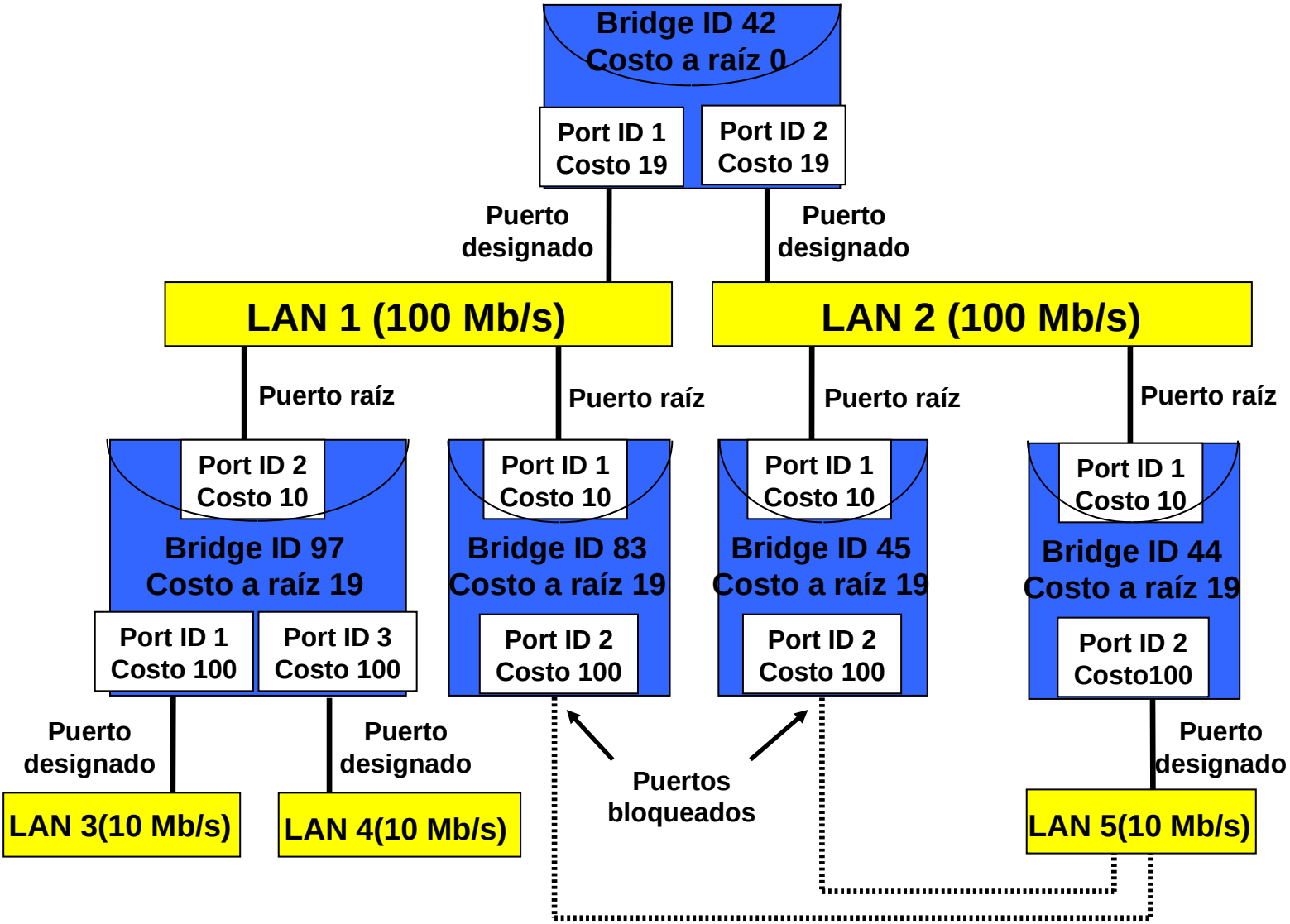


Dada una topología de red el conmutador raíz es siempre el mismo, independientemente del orden como se enciendan los equipos o como se conecten los cables

Si se utiliza la prioridad por defecto el conmutador raíz es el de la MAC más baja, que puede ser cualquier conmutador, probablemente uno periférico o poco importante. Si el conmutador raíz se apaga los demás han de elegir de entre ellos un nuevo raíz y recalcular el árbol, esto consume CPU y puede provocar inestabilidad si se tarda en llegar a la convergencia.

La prioridad permite controlar la selección del conmutador raíz asegurando que esa función recaiga por ejemplo en uno que esté siempre encendido, evitando así problemas de convergencia.

Spanning Tree de la red anterior

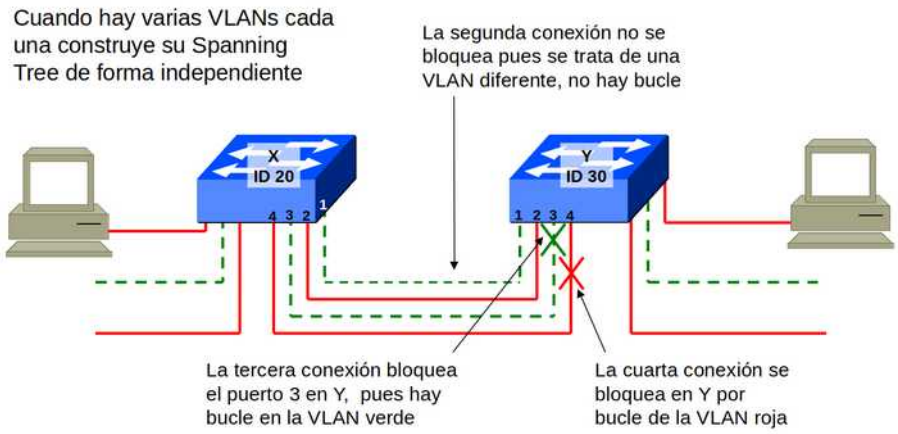


Vlans y STP. PVSTP+

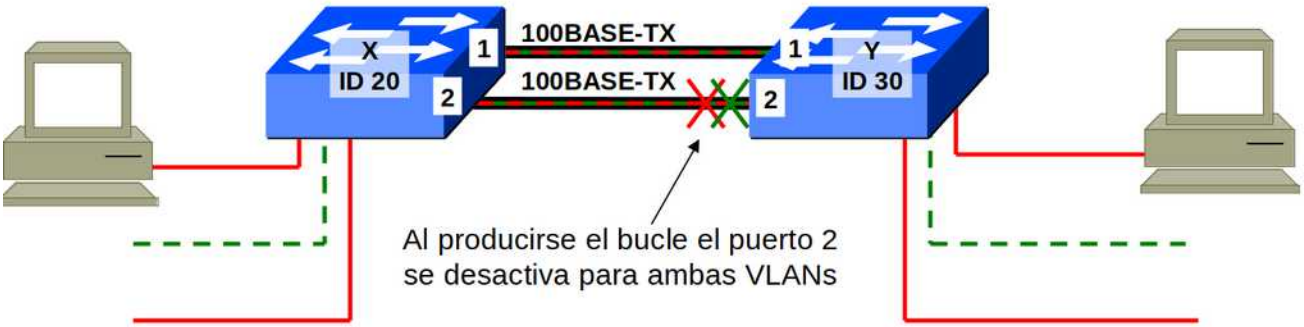
En principio cuando hay VLANs configuradas en un conmutador este ejecuta una instancia independiente de Spanning Tree para cada VLAN, esto es propietario de CISCO

Todos los parámetros característicos de Spanning Tree (prioridad, costo, etc.) se configuran independientemente para cada VLAN

Si se hace un bucle entre puertos asignados a diferentes VLANs no se bloqueará ningún puerto ya que en la topología de Spanning Tree no hay ningún bucle



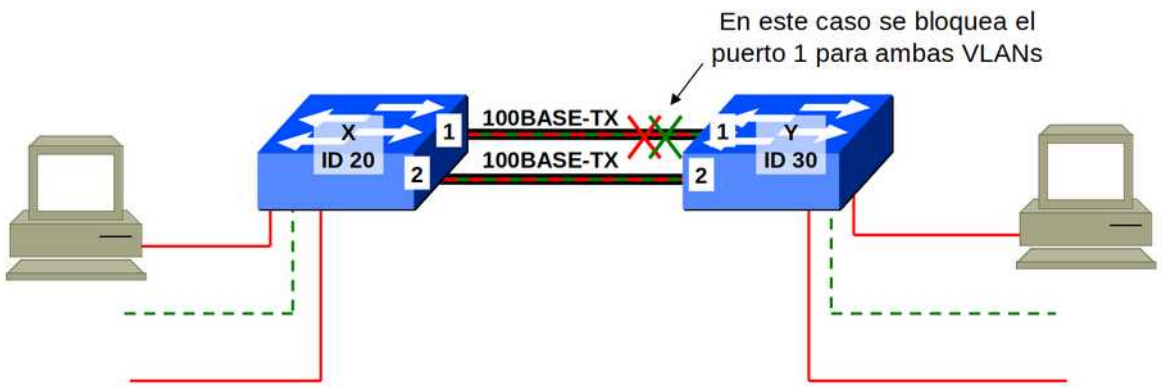
STP con enlaces Trunk



Dado un mismo costo y prioridad se elige como raíz el puerto de número menor, y por tanto se bloquea el de número mayor. La prioridad por defecto es 128.

VLAN	Puerto	Costo	Prioridad
Roja	1	19	128
	2	19	128
Verde	1	19	128
	2	19	128

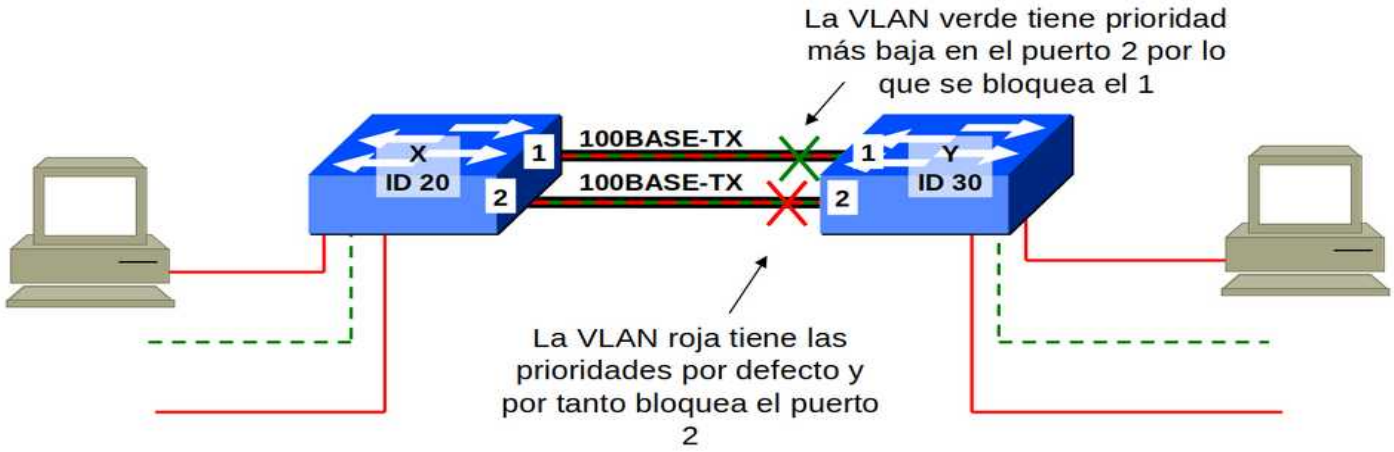
Si modificamos la prioridad, modificamos el puerto que se bloqueará



Modificando la prioridad se puede alterar la elección del spanning tree. Si se le da una prioridad menor al puerto 2 se le sitúa por delante del 1 y se le elige como puerto raíz, bloqueando entonces el 1.

VLAN	Puerto	Costo	Prioridad
Roja	1	19	128
	2	19	**127**
Verde	1	19	128
	2	19	**127**

Balanceo de carga en enlaces Trunk



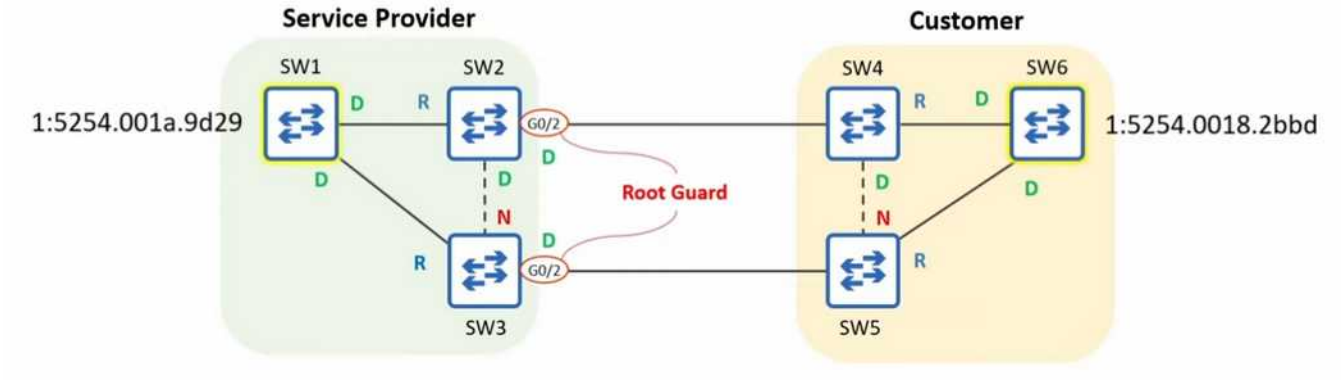
Si modificamos la prioridad en una VLAN y a la otra le dejamos los valores por defecto el puerto bloqueado será diferente en cada VLAN

El resultado es que la VLAN roja usa el enlace 1-1 y la verde el 2-2. Se consigue balancear tráfico entre ambos enlaces.

VLAN	Puerto	Costo	Prioridad
Roja	1	10	128
	2	10	128
Verde	1	10	128
	2	10	**127**

Root Guard

Podemos asegurar que un switch concreto es el raíz poniendo su prioridad a cero. Pero que sucede si nuestra LAN es conectada a otra? Podemos perder nuestro switch raiz en favor de otro de la LAN ajena. Podemos entonces habilitar Root Guard en los puertos unidos a los switches de la otra LAN para que se bloqueen si reciben BPDUs anunciando root switches con un BID mejor



Podemos habilitarlo con
SW2(config-if)# spanning-tree guard root

```
SW2(config)# interface g0/2
SW2(config-if)# spanning-tree guard root
*Sep 21 08:38:56.314: %SPANTREE-2-ROOTGUARD_CONFIG_CHANGE: Root guard enabled on port GigabitEthernet0/2.
*Sep 21 08:38:56.321: %SPANTREE-2-ROOTGUARD_BLOCK: Root guard blocking port GigabitEthernet0/2 on VLAN0001.
SW2(config-if)# do show spanning-tree
!output omitted
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Root FWD 4         128.1    P2p
Gi0/1               Desg FWD 4         128.2    P2p
Gi0/2               Desg BKN*4         128.3    P2p *ROOT_Inc
```

BKN = Broken
ROOT_Inc = Root Inconsistent

Para volver el puerto al estado normal debe solucionarse el problema que lo creó, es decir, el Service Provider debe decirle al customer que eleve la prioridad de su switch raiz para que vuelva a funcionar todo. Una vez hecho esto, como cada 20 segundos caducan los BPDU's, el puerto volvería a la normalidad y el STP convergería teniendo en cuenta los nuevos switches.

Loop Guard

Protege en contra de los enlaces unidireccionales, por ejemplo si se fastidia uno de los dos cables de fibra óptica
Si un puerto deja de recibir BPDU's se pone en estado bloqueado, se desbloqueará cuando se solucione el problema que lo causó, es decir, cambiemos el cable de fibra por ejemplo.
Para habilitarlo podemos hacerlo por puerto o en global
SW(config-if)# **spanning-tree guard loop**
o bien
SW(config)# **spanning-tree loopguard default**
y luego podemos deshabilitarlo en los que queramos con
SW(config-if)# **spanning-tree guard none**

```
%SPANTREE-2-LOOPGUARD_BLOCK: Loop guard blocking port GigabitEthernet0/1 on VLAN0001.

SW3(config-if)# do show spanning-tree

!output omitted

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Root FWD 4         128.1    P2p
Gi0/1               Desg BKN*4         128.2    P2p *LOOP_Inc
```

RootGuard y LoopGuard no pueden estar a la vez configuradas en el mismo puerto. Si configuramos una y luego la otra, la primera se deshabilita.

RSTP+ .Rapid Spanning Tree

En 1998 se estandarizó el Rapid Spanning Tree (RST, IEEE 802.1w) una variante del protocolo original que reduce el tiempo de convergencia a unos 6 seg. Actualmente el STP tradicional está declarado obsoleto.

Speed	STP Cost	RSTP Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gbps	4	20,000
10 Gbps	2	2000
100 Gbps	X	200
1 Tbps	X	20

Se basa en un nuevo intercambio de paquetes entre switches que produce que los puertos pasen al estado de forwarding más rapidamente.

Se han actualizado los costes para acomodarse a las nuevas velocidades de los enlaces. En packet tracer siguen las antiguas.

También se ha cambiado el sistema de estado de puertos, uniendo el bloqueado y apagado en uno (discarding, pero en show spanning-tree aparecerá como block, y elmininando el estado de listening

STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/ Transitional
Discarding	NO/YES	NO	NO	Stable
Learning	YES/YES	NO	YES	Transitional
Forwarding	YES/YES	YES	YES	Stable

Entre otras mejoras en RST los conmutadores mantienen información sobre la segunda ruta de menor costo al raíz, con lo que la conmutación a la nueva topología en caso de fallo de la actual es mucho más rápida. También en caso de cambios se vacía la tabla MAC. Admás ahora los puertos pueden ser root, designated, alternate y hay un nuevo rol llamado backup que se usa si hubiese un hub en el medio de dos switches que actua como backup de un puerto designado, ya que solo puede haber uno por dominio de colisión. RSTP es compatible con STP, si los mezclamos utilizarán STP clásico.

Para habilitarlo usamos

SW(config)# **spanning-tree mode rapid-pvst**

Edge

Los puertos Edge están conectados a hosts y por tanto no se pueden crear bucles con ellos. Pueden ir directamente al estado forwarding sin proceso de negociación. Funciona igual que portfast.

Se configura un puerto como edge habilitando portfast en él

SW(config-if)# **spanning-tree portfast**

Point to point

Estos están conectados directamente a otros switches, no hubs en el medio, por lo que funcionarán en full-duplex. No necesitamos configurarlos directamente pues debería detectarse pero aún así puede configurarse explícitamente

SW(config-if)# **spanning-tree link-type point-to-point**

MSTP

Multiple Spanning Tree es un protocolo estandarizado que todavía no tiene equivalencia propia en CISCO por lo que corre en el standard original. Permite correr diferentes instancias de RSTP por grupos de VLAN, lo cual ahorra mucho trabajo si hay muchas VLAN y tenemos que configurar switches raiz para cada una de ellas.