Configuración de STP para controlar la topología

Guillermo Pérez Trabado *©2016-2024*

Diseño de Infraestructuras de Redes

Depto. de Arquitectura de Computadores - Universidad de Málaga

Escenario

En este documento vamos a aprender a configurar el Spanning Tree Protocol (STP) de los switches para controlar la forma en que se genera el **árbol de expansión mínima** del grafo formado por los switches de la red de un site.

Antes de empezar recordaremos qué es el Spanning Tree de una red de switches Ethernet y cómo funciona el protocolo STP:

Spanning Tree

El enrutamiento usado por los switches Ethernet tiene dos propiedades que tenemos que destacar:

- Cuando un switch no conoce la ruta a un destino usa la **inundación de la red** emitiendo copias de un paquete por todos sus enlaces de salida (broadcast). Si ningún switch conoce aún la ruta al destino, todos usarán la inundación, por lo que llegarán copias del paquete a todos los switches de la red y a todos los puertos de los mismos. Además, cada switch recuerda el enlace por el que se recibe un paquete en cada momento como la ruta más corta al puerto donde está conectada la MAC de origen de dicho paquete.
- Ethernet no incluye ningún tipo de mecanismo de control de bucles, por lo que los switches asumen que la topología de la red **nunca contiene un ciclo**. Si el grafo de la red contiene un ciclo, tendrá consecuencias desastrosas ya que los paquetes que entren en el ciclo no desaparecerán nunca de la red.

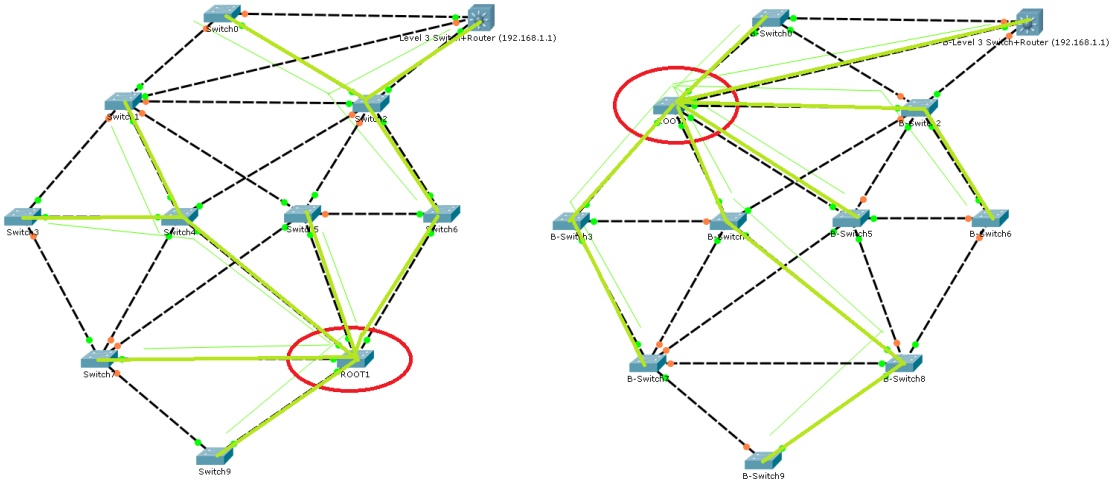
Es decir, el encaminamiento por broadcast y el aprendizaje asumen que la red es un árbol (grafo acíclico) y nunca un grafo con ciclos. Por tanto, antes de permitir el enrutamiento, los switches deben garantizar que la topología de la red es un árbol a base de **podar los enlaces sobrantes** que generarían ciclos.

Minimum Spanning Tree

El cálculo de los enlaces a desactivar no es trivial ya que se pretende que el árbol generado sea óptimo desde el punto de vista del encaminamiento. El árbol buscado se denomina **minimum spanning tree (MST)**, que es el subgrafo acíclico con el menor coste global de caminos posible. Para un mismo grafo, es posible calcular más de un MST con el mismo coste global. En la figura siguiente puedes ver dos MST diferentes (en verde) calculados para el mismo grafo. Observa que:

- Cada uno de ellos tiene la raíz del MST en un nodo distinto del grafo original.

- El MST resultante es un árbol que, en cada nivel, prioriza la amplitud frente a la profundidad. Eso quiere decir que, en caso de existir dos caminos entre un nodo y la raíz, se prioriza el más corto ya que esto minimiza la distancia entre cualquier par de nodos medida en *número de saltos* (**hops**).



1. Spanning Tree Protocol

El MST de la red es construido entre todos los switches del grado mediante el protocolo STP. En este apartado no vamos a detallar cómo funciona, pero sí cómo se puede controlar su funcionamiento de forma práctica para optimizar nuestro diseño de red.

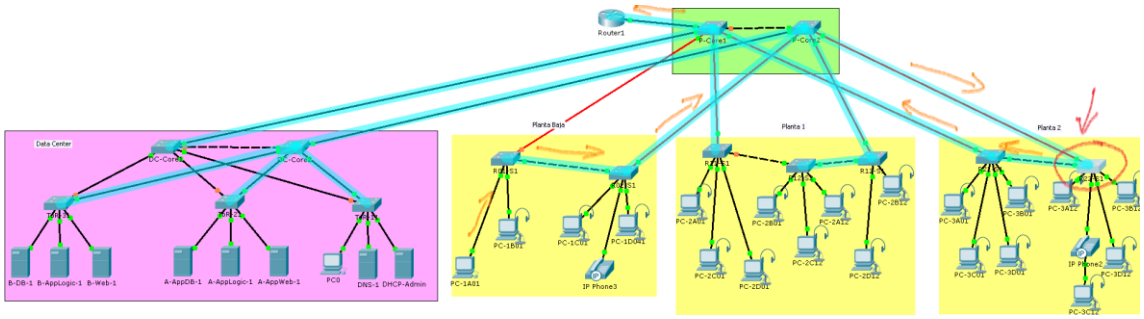
Elección de la raíz del MST en una red

Uno de los principales detalles a saber de STP es que los switches tienen que comenzar eligiendo qué switch va a ser la raíz del MST. STP define una prioridad para cada switch de forma determinista. El switch con el valor más pequeño es elegido como **root** de un grupo de switches. STP usa en cada switch un número de 64 bits como prioridad compuesta de dos campos:

- Un campo de prioridad de 16 bits. Su valor por defecto es 32769 en todos los switches si no se configura.
- Un campo de 48 bits cuyo valor es la dirección MAC del switch.

Eso quiere decir que, si no se configura STP, el switch con la MAC más pequeña (más antigua ya que se asignan por orden de fabricación) será siempre elegido como root. Por un lado, eso nos proporciona una negociación automática de STP que funciona sin configuración alguna y que siempre elige el mismo switch como root de forma estable. Pero por otro, puede depararnos algunas sorpresas desagradables en nuestras configuraciones ya que el switch raíz puede no el que esperamos (un core switch).

El caso del grafo del ejemplo anterior no es un representativo de la estructura de red de un edificio con dos niveles de repartidores y un data center. En un edificio típico, la red es un árbol con doble raíz y caminos redundantes entre ambas raíces y los switches de cada armario, que además tienen un ciclo entre ellos. El siguiente esquema muestra el MST generado para esta estructura. Sin embargo, no se ha configurado la prioridad de los switches y, por tanto, el **STP root** elegido es aquel con la MAC más baja (marcado por un círculo rojo en la figura). El MST de la red construido por STP puede verse dibujado en azul:



Se puede observar que los enlaces con los LED en naranja han sido desactivados por STP para eliminar los ciclos del grafo original formado por los enlaces redundantes. El administrador, de forma intuitiva considera que la raíz de la red debería ser *Core-1* ya que ahí está conectado el router LAN que interconecta todas las VLANs entre sí. Todos los terminales deberían estar a la mínima distancia posible de dicho router.

Sin embargo, nos puede sorprender lo ineficiente de la topología negociada (en azul), ya que el MST calculado se ha optimizado para reducir la distancia de todos los nodos al **switch raíz** (que no es Core-1 como podríamos creer al ver el mapa de la red). La ruta marcada por las flechas requiere **7 hops** (saltos Ethernet) para llegar al router cuando podría llegarse en solo **3 hops**. El problema es que el router no está conectado a la raíz, sino a una hoja del árbol, aunque creamos lo contrario.

Optimización de la topología de red

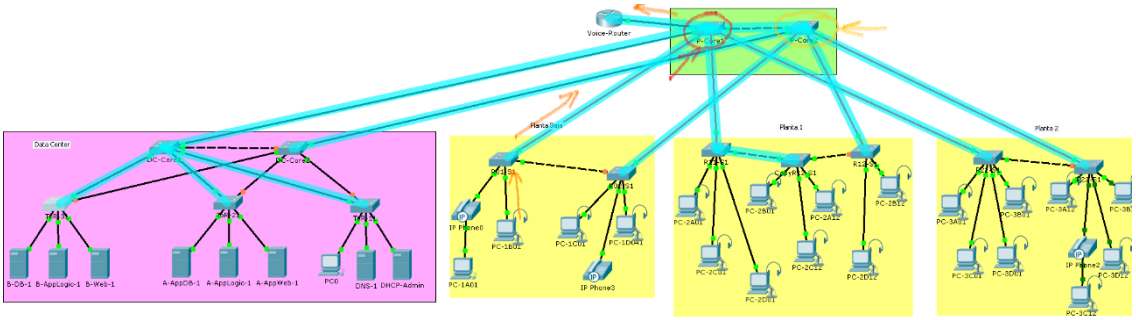
Por tanto, uno de los aspectos más importantes a configurar en una red Ethernet es la prioridad de los core switches, para forzar su elección como raíces del MST. Para ello usaremos el comando de configuración siguiente **solo en el switch que deseamos como primario**:

```
Core1(config)#spanning-tree vlan 1-1023 root primary
```

En caso de avería del core primario tenemos que garantizar que le reemplaza el core secundario. Por tanto, es necesario configurar este segundo switch con una prioridad ligeramente más baja para que se convierta en el nuevo root. Este comando **solo se ha de ejecutar en el secundario**:

```
Core2(config)#spanning-tree vlan 1-1023 root secondary
```

En la figura siguiente puede verse que ahora el **STP root** es *Core-1* y que el MST generado está optimizado para que todos los terminales estén a la mínima distancia de dicho nodo, y por tanto también del router LAN. Ahora vemos que el camino para llegar desde la mayor parte de terminales hasta el router es de 3 hops y el camino más largo es de 4 hops. Una ventaja de elegir bien el root es que el RTT es uniforme dentro de la red para todas las comunicaciones cliente-servidor (PC ⇔ Servidor del Data Center y (PC ⇔ Servidor externo a la red).



2. Per VLAN STP

Hay que detallar que en realidad los switches de Cisco usan por defecto una versión mejorada de STP llamada PVSTP (per VLAN STP), que negocia un árbol MST distinto (y un root distinto) de forma independiente para cada VLAN definida. Normalmente el core switch principal debería ser la raíz del MST de todas las VLANs, por lo que el comando especifica la prioridad para todas las VLANs posibles a la vez. Sin embargo, en algunas situaciones puede ser interesante balancear el tráfico de distintas VLANs para que fluya por distintos enlaces troncales entre un repartidor secundario y el primario. Esto se consigue definiendo un **STP root** diferente para cada grupo de VLANs. Por ejemplo:

```
Core1(config)#spanning-tree vlan 10-15,100,102-199 root primary
Core1(config)#spanning-tree vlan 16-29,101,200-210 root secondary
```

y

```
Core2(config)#spanning-tree vlan 10-15,100,102-199 root secondary
Core2(config)#spanning-tree vlan 16-29,101,200-210 root primary
```

Con esta configuración, por ejemplo, el switch intermedio del segundo repartidor enviaría el tráfico de las VLANs 10 a 15,100 y 102 a 199 por el trunk hacia el switch de su izquierda y el tráfico de las VLANs 16 a 29, 101 y 200 a 210 por el enlace hacia el switch de su derecha, aprovechando la capacidad de ambos enlaces.

3. Breve descripción del algoritmo usado por STP

STP implementa un algoritmo distribuido en el que un grupo de switches resuelven un problema de teoría de grafos (cálculo del MST). Cada switch envía por todos sus **enlaces en modo trunk** (switches vecinos) PDUs con la información sobre el **root STP** que conoce hasta el momento. Mediante el intercambio iterativo, cada switch obtiene información de la topología de la red a través de sus vecinos. De esa forma, el conocimiento sobre la topología del grafo se va distribuyendo a todos los nodos hasta que todos tienen la misma información (convergencia de la red). El algoritmo STP tiene dos partes diferenciadas:

- Elegir al switch raíz (**root**) del MST.
- Construir el MST.

Los pasos para la elección del **root** son:

- Cada switch envía a todos sus vecinos una PDU (trama) con su identidad (prioridad) y también la del switch con prioridad más baja que conozca (que inicialmente es él mismo).
- Si recibe una PDU de un vecino con prioridad más baja, lo elige como **root** y en la siguiente iteración vuelve a enviar la PDU con información de root actualizada.
- Al cabo de un número de iteraciones (el diámetro del grafo), todos los switches de la red conocen al switch con la prioridad más baja.

Los pasos para construir el MST son:

- Cada switch envía a sus vecinos una PDU que indica su distancia al **root STP.** Es decir, la longitud del camino más corto conocido medida en **hops** (enlaces).

- Si un switch decide reemplazar su **root** previo por otro con menos prioridad, adopta como camino más corto al **root** el enlace por el que ha recibido la PDU y aumenta la distancia en 1.
- Si el switch ya conocía este mismo **root** anteriormente, elige como camino el más corto de los dos ahora conocidos. Sin embargo, ambos caminos quedan como **candidatos para llegar al root**.
- Una vez pasado un tiempo sin que se descubran nuevos caminos, cada switch termina **activando el forwarding** en el enlace candidato con el camino más corto y **desactivando el forwarding** (pero no apagando) del resto de enlaces candidatos. De esta manera, se desactivan solo los enlaces mínimos necesarios para eliminar los ciclos.
- Por los enlaces desactivados no se reenvían tramas ethernet de datos, pero se siguen enviando periódicamente tramas DTP, VTP y STP para mantener el estado de todos esos protocolos.

Alta Disponibilidad (High Availability)

Los switches continúan mandando tramas STP de forma periódica por todos sus enlaces. Si un switch se avería, se desactiva un puerto o un medio de transmisión es desenchufado o cortado, el switch vecino deja de recibir tramas STP de su vecino. En consecuencia:

- Al cabo de un tiempo sin recibir actualizaciones borrará la información STP de dicho enlace.
- Si este enlace era su camino hacia el **la raíz STP**, lo bloqueará y elegirá como camino al **root** el camino más corto que conozca usando **otro enlace candidato**. Además, en la próxima PDU enviará la información actualizada a sus vecinos, que podrían decidir elegir otro camino conocido más corto al root.

Todo eso implica que, si un switch o un enlace dejan de funcionar, de forma automática se genera un cambio en la topología del árbol que se va propagando en cascada por todo el grafo hasta que vuelve a converger. Lo mismo ocurre cuando se vuelve a conectar un enlace o switch incluso cuando se añade un nuevo switch o una nueva conexión redundante.

Esto permite mantener el servicio (HA) sin necesidad de reconfigurar nada. Además, tanto Ethernet como IP están basados en datagramas, por lo que el control de retransmisión TCP se encargará de repetir las tramas perdidas y de continuar con la transferencia. La única crítica es que **el tiempo de reacción de STP** por su diseño puede ser suficientemente alto como para que los **temporizadores de una conexión TCP** desistan definitivamente y consideren la **conexión cerrada**.

4. Rapid per VLAN Spanning Tree Protocol (RPVST)

El problema del STP es que el intercambio de información tiene lugar solo cada vez que un temporizador lo indica (por defecto cada 15s). El número de intercambios necesario para que la información se propague depende del **diámetro del grafo** (*diámetro*: la *distancia* máxima entre cualquier par de nodos del grafo). El resultado es que el **tiempo de convergencia de STP** es bastante grande ya que se mide en múltiplos de periodos de 15s según el diámetro de la red.

Además, cuando un switch se reinicia, se bloquea el **forwarding** en todos sus enlaces hasta que STP no ha llegado al estado de convergencia. Esto hace que el encendido de una red de cierto

diámetro (por ejemplo, el backbone de un campus universitario) requiera periodos incluso de 10 minutos o más sin intercambio de tráfico.

Aceleración de la convergencia (Rapid PVSTP)

Para acelerar el tiempo de convergencia de STP se desarrolló una variante conocida como Rapid STP (y también Rapid PVSTP) que mejora la implementación del algoritmo convergiendo mucho más rápido. La diferencia es que un switch con R-STP manda **PDUs de actualización** a todos sus vecinos **inmediatamente** cuando detecta un **cambio en la topología**, cosa que ocurre en dos casos:

- Cuando detecta que un enlace en modo trunk pierde la conexión con el vecino.
- Cuando recibe una PDU de otro vecino notificando un cambio de topología.

El resultado es que el cambio de topología provocado por el apagado de un enlace se propague en pocos milisegundos a todos los switches de la red. Con este tiempo de reacción, la pérdida de datagramas Ethernet es mínima y además puede ser recuperada por TCP de forma normal, por lo que un cambio en la topología apenas es percibido por los usuarios.

Configuración de Rapid PVSTP

Para usar este protocolo, hemos de cambiar el modo de funcionamiento de STP en **todos los switches** con el comando:

```
(config)#spanning-tree mode rapid-pvst
```

5. Activación de PortFast

Otra aceleración posible de la convergencia es evitar que un switch de acceso intente negociar STP o RSTP en los puertos donde se conectan terminales o servidores. A priori, un switch no sabe si en un enlace hay un switch averiado o un equipo que no implementa STP (PC o servidor). Por defecto los switches intentan usar STP en cada puerto por precaución en prevención de que puedan encontrarse con otro switch conectado a un puerto de acceso y que pueda provocar un bucle.

Una ventaja inmediata de usar PortFast con los PCs de las oficinas es que acelera la configuración mediante DHCP:

- Sin usar PortFast, suele ocurrir que cuando el driver del SO activa el puerto Ethernet, el switch no activa el forwarding hasta pasados unos 30s esperando la negociación STP que nunca ocurre. Los primeros reintentos del cliente DHCP se van perdiendo y éste aumenta el timer entre reintentos pasado un cierto tiempo, siempre antes de que el switch active el **forwarding**. Cuando por fin se activa, pueden pasar incluso algunos minutos hasta el próximo reintento del cliente DHCP. La configuración de la red parece eternizarse en cada arranque.
- Al usar PortFast, el switch activa el **forwarding** del puerto en cuanto detecta la activación del puerto del PC. Cuando el cliente DHCP del SO envíe un Request durante el boot del sistema, el puerto ya se encuentra activado y la respuesta llega inmediatamente.

Configuración de PortFast

El siguiente comando deshabilita por defecto la negociación STP en aquellos puertos que están configurados en modo **access**:

```
(config)#spanning-tree portfast default
```

PELIGRO: Este último comando solo debe usarse en los puertos con terminales (servidores o PCs) en los **switches de acceso**. Si usamos PortFast en un switch y conectamos a un puerto en modo acceso un switch que genere un ciclo en la red o simplemente se unen con un cable dos puertos en modo access, el ciclo no será detectado y el tráfico dará vueltas sin parar por el mismo. Activar PortFast es un riesgo si no controlamos lo que se conecta a un puerto.

6. Monitorización de STP

Aunque son raros los problemas de configuración de STP, no sería extraño que queramos examinar el switch raíz de cada VLAN y otros detalles en caso de sospechar que hay problemas de rendimiento que podrían ser debidos a la topología MST.

Los comandos de configuración muestran el estado desglosado para cada VLAN definida en el switch. Algunos de los datos más interesantes son las direcciones MAC del Root ID (root switch de una VLAN) y del Bridge ID (descripción del switch local). Si son idénticas, como en el primer ejemplo, estamos viendo el **root** de una VLAN. Además, el propio comando indica en su salida que estamos en el **root switch**:

```
P-Core1#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    24577
              Address     0030.F2C4.20D9
              This bridge is the root
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    24577  (priority 24576 sys-id-ext 1)
              Address     0030.F2C4.20D9
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi1/1            Desg FWD 4         128.2    Shr
Gi8/1            Desg FWD 4         128.9    P2p
Gi9/1            Desg FWD 4         128.10   P2p
Gi0/1            Desg FWD 4         128.1    Shr
Gi2/1            Desg FWD 4         128.3    Shr
```

Si son diferentes, como en este otro ejemplo, estamos en un nodo distinto al root:

```
R1-S1#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    24577
              Address     0030.F2C4.20D9
              Cost        4
              Port        1(GigabitEthernet0/1)
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     0002.1687.E375
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  20

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi0/1            Root FWD 4         128.1    Shr
Gi2/1            Desg FWD 19        128.3    P2p
```

```
Gi3/1            Desg FWD 19        128.4    P2p
Gi9/1            Desg FWD 4         128.10   P2p
```