

Spanning Tree Protocol (STP)

Primera Parte



Antes de Imprimir este documento
considere si es necesario
Ayudemos al Ambiente !!!!

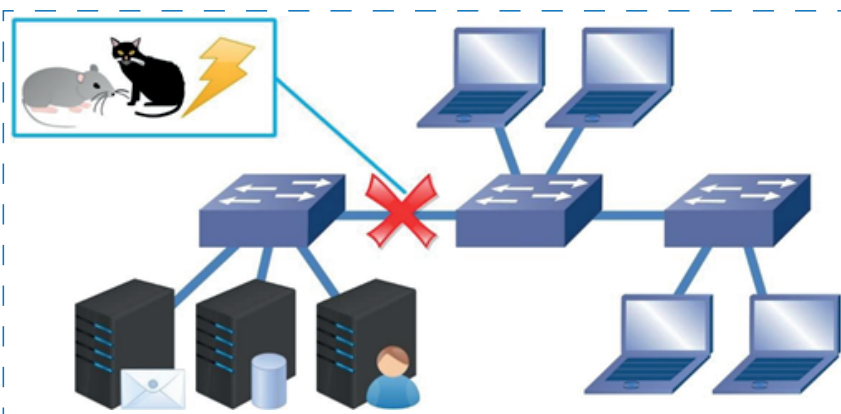
Universidad San Carlos de Guatemala
—— **DANILO ESCOBAR** ——

STP

► Spanning Tree Protocol (STP)

Ya sea de manera planeada o no, las redes de comunicaciones siempre tienden a crecer. Muchas veces ante la urgente necesidad de satisfacer nuevos requerimientos dicho crecimiento se realiza de manera desordenada, conectando los nuevos dispositivos a los ya existentes en cascada conforme estos se hacen necesarios.

Este tipo de disposición hace a la red más propensa a fallar al hacerla menos resiliente contra desperfectos mecánicos, eléctricos y factores externos.



Una falla en una red pobremente diseñada puede limitar o eliminar completamente la conectividad.

Esta es la razón por la que dentro de un buen diseño de red siempre debe contemplarse el crecimiento de la misma e incluir también cierto grado de redundancia, la cual permitirá reducir el tiempo inoperativo y mitigar las pérdidas económicas derivadas.

No obstante los evidentes beneficios, la implementación de dicha redundancia tiene su costo tanto económica como operacionalmente, siendo un caso especialmente problemático cuando se utiliza con *switches* tradicionales capaces de funcionar solamente a nivel de capa 2.

Considérese el siguiente escenario donde dos *switches* han sido conectados entre sí utilizando enlaces redundantes y la computadora A trata de enviar un mensaje a la computadora B.

Para averiguar la dirección física de B, la computadora A enviará un ARP *Request* como un *broadcast* hacia el resto de la red. La trama llegará al *switch*, quien examinará la dirección MAC de origen de la misma para luego agregar una entrada en su memoria y situar al dispositivo A en el puerto por donde está ingresó, antes de reenviar dicha trama por todas las demás interfaces.

STP



El switch aprende la dirección MAC de la computadora A y reenvía el *broadcast* por todos sus otros puertos.

La transmisión llega al *switch* inferior el cual sigue el mismo proceso y reenvía las tramas a través de todos los puertos exceptuando aquel que la recibió originalmente. De haber una sola conexión entre ambos *switches* dicha transmisión sólo alcanzaría a la computadora B, sin embargo, al existir un enlace redundante dicha información es retransmitida nuevamente al *switch* superior quien ahora creerá que la computadora A esta conectada en algún punto del dispositivo inferior.

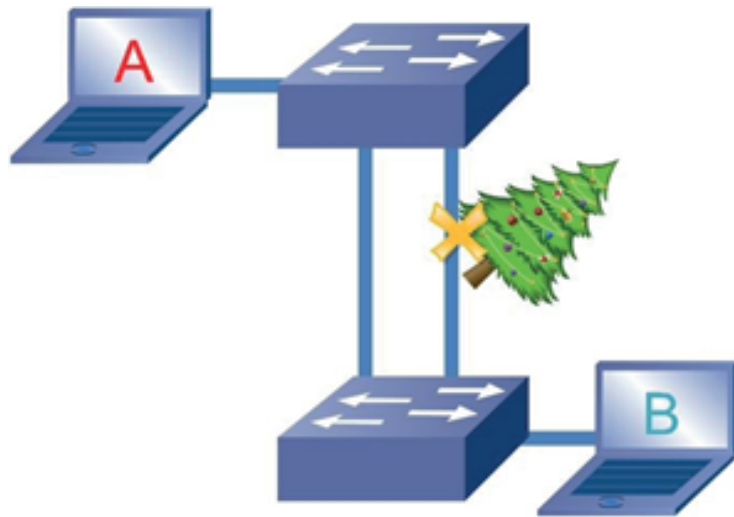


El enlace redundante hace posible que el *broadcast* original retorne al dispositivo en donde se originó.

La inestabilidad en la base de datos de direcciones MAC provocará cada vez más retransmisiones y generará más *broadcast*, el cual al deber ser procesado en *software* (Por lo que carga al CPU) y ser dirigido hacia todos los dispositivos llevará a la red a un paro general una vez cierto límite sea sobrepasado, condición que se conoce como una tormenta de *broadcast*.

Para mitigar dicho problema se utiliza un protocolo capaz de detectar y bloquear enlaces redundantes, evitando la formación de bucles a nivel lógico al mismo tiempo que se mantiene la redundancia física en la red y que es conocido como *Spanning Tree Protocol* (STP).

STP



Spanning Tree desactiva a nivel lógico los enlaces redundantes para prevenir bucles.

STP fue creado en 1985 por Radia Perlman con el propósito de que los dispositivos de capa 2 pudieran detectar y bloquear enlaces redundantes y luego reactivarlos en caso de una falla. El estándar abierto fue publicado en 1990 con el nombre de 802.1D y presentaba algunas diferencias con respecto al original.

Con el transcurrir de los años se han hecho varias enmiendas y se han creado nuevas implementaciones a partir de la primera versión (Ej.: 802.1s, 802.1w), todas las cuales son contenidas hoy en día en el estándar 802.1Q.

Antes de proseguir es importante remarcar que STP es un protocolo antiguo que ha mantenido mucho de su terminología original, por esta razón muchas definiciones y configuraciones hacen referencia al dispositivo antecesor al *switch*: El *bridge*. De esta manera un bucle entre switches es referido como un *bridging loop* y no como un *switching loop* (término más intuitivo). No obstante, durante el resto de la discusión de este protocolo se favorecerá el término *switch* (cuando sea posible) por motivos pedagógicos.

STP

► Bridge protocol data units (BPDUs)

Al contrario de los protocolos de estado de enlace (*Link state*) en donde cada uno de los dispositivos conoce a todos los demás presentes en la topología, los *switches* ejecutando *Spanning Tree* trabajan de manera independiente, ajenos a los demás dispositivos y su colocación dentro de la red.

Así pues, para realizar las elecciones necesarias y detectar cambios en la topología, los *switches* envían y reciben a través de sus interfaces tramas especiales llamadas *Bridge protocol data units* (BPDUs), las cuales son enviadas a un grupo de *multicast* al que solo pertenecen los dispositivos ejecutando STP.

Las BPDUs incluyen mucha información y pueden ser clasificadas acorde a su propósito como:

- BPDUs de configuración (Configuration BPDU): Que incluyen toda la información necesaria para realizar los cálculos requeridos por Spanning Tree (Información del switch, timers, etc.), y que son generadas únicamente por el switch raíz para luego ser propagados al resto de dispositivos.
- BPDUs de notificación de cambio de topología (Topology Change Notification – TCN BPDU): Utilizada para manejar los cambios que ocurren dentro de la topología. Se generan cuando una interfaz que pertenece al proceso de STP cambia de estado, para luego propagarse de dispositivo a dispositivo hasta llegar al switch raíz el cual indicará a los demás que deben renovar su base de datos de direcciones MAC en un tiempo más corto de lo usual, para que estos puedan ajustarse al cambio.



STP

Estados de *Spanning Tree*

Los puertos que participan de *Spanning Tree* pueden estar en uno de los siguientes estados:

- **Bloqueando (*Blocking*):** En este estado el puerto no es capaz de enviar o recibir información ni de aprender direcciones MAC. Es el estado inicial de todos los puertos (a manera de evitar la formación de bucles cuando el dispositivo inicia) y aquel al que regresan aquellos que deben estar bloqueados para quebrar bucles en los enlaces redundantes. En esta fase los puertos no pueden enviar BPDUs por lo que se limitan a procesar aquellas que reciben.
Si el puerto se está inicializando y es considerado como candidato para ser utilizado para el envío de información o si se necesita utilizar dicho puerto a causa de una falla en la red, pasará al siguiente estado, pudiendo llegar a demorarse hasta 20 segundos según el caso.
- **Escuchando (*Listening*):** Los puertos que son considerados por el *switch* como candidatos para empezar a enviar información pasan a este estado, durante el cual todavía no es posible enviar o recibir información ni aprender direcciones MAC. La diferencia entre esta fase y la anterior radica en que el puerto además de recibir y procesar también puede enviar sus propias BPDUs, con lo que pasa a participar activamente del proceso y decisiones tomadas por STP.

Es también en esta etapa donde se decide si un puerto regresará a estar bloqueado o si será utilizado para transmitir datos, pasando en este último caso al siguiente estado después de 15 segundos.

- **Aprendiendo (*Learning*):** Este es el último estado antes de empezar a transmitir. Se siguen recibiendo y enviando BPDUs, además el puerto comienza a aprender direcciones MAC, siendo 15 segundos el tiempo otorgado para este proceso.
- **Transmitiendo (*Forwarding*):** En este estado el puerto es completamente operacional siendo capaz de recibir, enviar y procesar tanto información como BPDUs y agregar entradas en la base de datos de direcciones MAC.
Finalmente hay un último estado que no forma parte directamente del proceso de *Spanning Tree* pero que debe considerarse:
- **Deshabilitado (*Disabled*):** Estado en el cual el puerto ha sido apagado por un administrador o deshabilitado por algún protocolo.

STP

Roles en *Spanning Tree*

Después de que la red haya convergido y de que los puertos hayan pasado por uno o varios de los estados presentados anteriormente (y que los mismos no se encuentren deshabilitados), estos pasarán a operar en uno de los siguientes roles.

- Puerto raíz (*root port*): Es aquel que posee la mejor ruta hacia el *switch* raíz, formando parte de los enlaces activos en la topología por lo que siempre estará transmitiendo (*Forwarding*). Este rol no existe en el *switch* raíz (Lo cual a menudo es fuente de confusión) estando el mismo reservado a los demás dispositivos ejecutando STP, los cuales pueden tener un solo puerto cumpliendo esta función. Este es también el puerto utilizado para comunicarle al *switch* raíz cuando ha ocurrido un cambio en la topología.

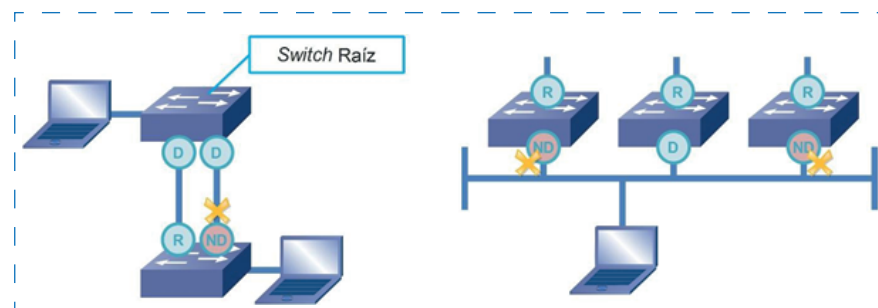
- Puerto designado (*designated port*): Los puertos en este rol siempre se encuentran transmitiendo (*Forwarding*) y son los únicos capaces de enviar BPDUs de configuración, motivo por el cual se encuentran presentes en todos los segmentos de la topología STP, dentro de cada uno los cuales debe existir necesariamente un único puerto que cumpla esta función. Este diseño permite que *Spanning Tree* pueda detectar bucles inclusive en enlaces conectados a un segmento compartido.

Todos los puertos del *switch* raíz son puertos designados.

- Puerto no Designado (*Non-Designated Port*): Este puerto forma parte de un enlace redundante por lo que no transmite información y se limita a escuchar las BPDUs provenientes de algún puerto designado (*Blocking*).

De esta manera, los enlaces activos están compuestos por un puerto raíz y uno designado mientras que los enlaces inactivos están formados por un puerto designado y uno que no lo está a manera de romper el bucle. Dentro de cada segmento siempre debe existir un único puerto designado; este diseño permite a *Spanning Tree* tratar con enlaces redundantes conectados al mismo segmento (lo que no es común en estos días).

Estos roles no son mostrados en las salidas del Cisco IOS.



Un puerto puede tomar el rol de un puerto designado (D), un puerto raíz (R) o el de un puerto no designado (ND).

- a) Dos switches, dos segmentos de red.
- b) Tres switches, un segmento compartido

STP

Elección del *switch* raíz y el rol de cada puerto

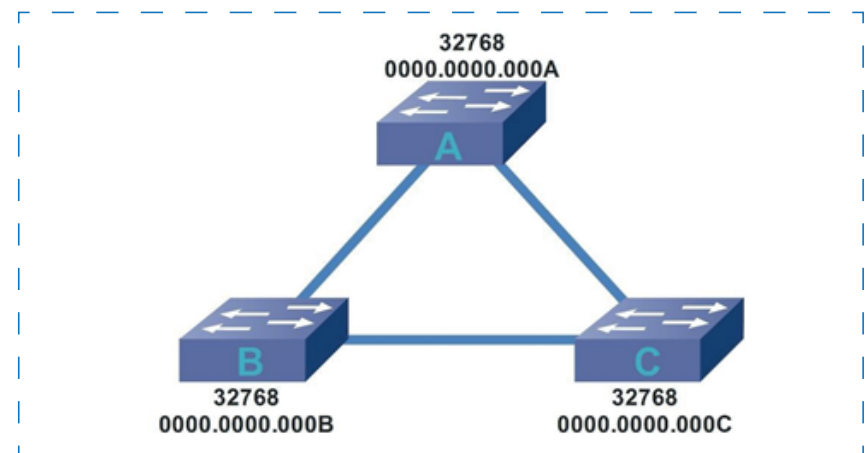
De acuerdo a lo expuesto anteriormente, para llevar a la red desde la inicialización de la misma hasta lograr una topología lógica libre de bucles, *Spanning Tree* necesita elegir un *switch* raíz, determinar el puerto raíz en cada dispositivo y luego elegir los puertos designados dentro de cada segmento.

Lo criterios utilizados por STP para tomar estas decisiones tienen en común que siempre prefieren los valores más bajos de sus respectivos parámetros, como se explica posteriormente.

Para elegir el *switch* raíz se utiliza como parámetro el identificador del *switch* (*bridge ID*), un valor compuesto por la combinación de un campo conocido como la prioridad (Un valor numérico con un tamaño original de 2 bytes) y la dirección MAC del dispositivo, siendo electo como raíz aquel con el *bridge ID* más bajo.

Por recomendación del estándar (802.1D) todo *switch* comienza con una prioridad por defecto de 32768, aunque este valor puede ser modificado para influir en la elección. Entre más baja la prioridad, más probabilidades tiene un dispositivo de ser elegido como raíz, utilizándose la dirección MAC solamente como medio de desempate (En caso todos los *switches* tengan la misma prioridad).

Para ilustrar el proceso de elección se introduce la siguiente topología donde todos los dispositivos siguen configurados con la prioridad por defecto,



Elección del switch raíz.

STP

Los *switches* intercambian y comparan información utilizando BPDUs, tramas dentro de las cuales cada dispositivo coloca el identificador del *switch* que estos reconocen como raíz de la topología así como su propio *bridge ID* para que los demás sepan de donde viene dicha comunicación.

BPDU		
Root Bridge ID	32768 0000.0000.000A	/ ← Identificador del switch raíz.
Sender Bridge ID	32768 0000.0000.000B	/ ← Identificador del switch que envía el BPDU.
.....	

Incluidos dentro de los campos del BPDU se encuentra el bridge ID del dispositivo que lo originó, así como la dirección que este reconoce como perteneciente al switch raíz.

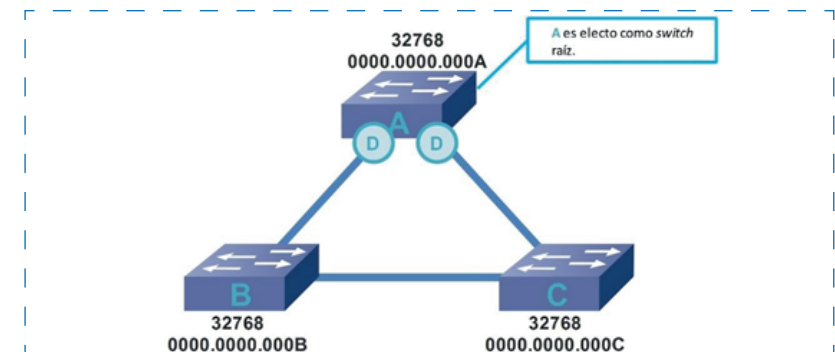
Cuando los *switches* se inician cada uno de ellos se considera a sí mismo el *switch* raíz de la topología y lo comienza a anunciar de esta manera a los dispositivos vecinos al utilizar su propia dirección MAC tanto en el campo que indica la dirección de origen de la comunicación como para el campo reservado al raíz.

Si durante el intercambio de información el *switch* recibe un BPDU de un dispositivo con un *bridge ID* menor al suyo, este lo

reconoce como el nuevo *switch* raíz de la topología y actualiza la información enviada a los vecinos. Este proceso se repite hasta que todos los participantes reconocen a un único *switch* raíz.

En el caso presentado anteriormente, al tener la misma prioridad el *bridge ID* más bajo será determinado por la dirección MAC más baja, razón por la cual el *switch* A será el *switch* raíz de la topología.

Una vez finalizada la elección, cada dispositivo debe determinar el rol que será asignado a cada uno de sus puertos con excepción del *switch* raíz, ya que todos sus puertos serán puertos designados.



Switch "A" es electo como switch raíz y como resultado todos sus puertos son puertos designados (D).

STP

Para elegir los roles de los puertos se utilizan varios parámetros comparados en secuencia, deteniéndose en el primero de ellos que no resulte en un empate y siendo siempre preferidos los valores más bajos.

El orden en que se comparan dichos parámetros se presenta a continuación:

1. **Costo de la ruta hacia el switch raíz** (*Root Path Cost*). Mismo que se explica en breve.
2. **Identificador del switch que envía el BPDU** (*Sender Bridge ID*). Explicado anteriormente.
3. **Identificador del puerto que envía el BPDU** (*Sender Port ID*). Compuesto por la prioridad del puerto (Un valor numérico de 4 bits) y el número de la interface. Al modificar la prioridad de un puerto podemos influir en la elección del puerto raíz una vez llegado el proceso a esta instancia. Este campo también está incluido en el BPDU.
4. **Identificador del puerto que recibe el BPDU** (*Receiver Port ID*). Compuesto de la misma forma que el campo anterior, con la excepción de que este es local al dispositivo por lo que no existe un campo correspondiente en las BPDU.

El *Root Path Cost* es un campo incluido dentro de las BPDU cuyo valor (llamado costo) se ve incrementado cada vez que entra

por una interface y que tiene como objetivo determinar las mejores rutas hacia el switch raíz.

El costo relacionado con cada una de las interfaces es llamado costo de la ruta (*Path Cost*) y es determinado por la velocidad de cada una de estas, aunque su valor puede modificarse dentro de cada interfaz para influir en el rol que le será dado. Al no existir valores recomendados para los costos que debían asignarse a las interfaces, Cisco utilizó la siguiente fórmula en sus primeras implementaciones de STP.

$$\text{Costo} = \frac{1 \text{ Gigabit/s}}{\text{ancho de banda}}$$

Fórmula original empleada por Cisco para determinar el costo de cada interfaz en base a su velocidad.

STP

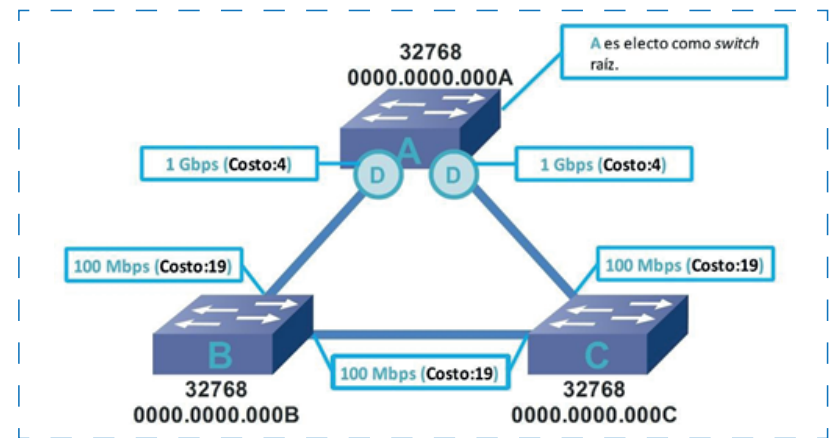
Sin embargo, debido al gran aumento en la velocidad de las interfaces y al hecho en que la fórmula presentada no funcionaba adecuadamente para enlaces más rápidos de 1 *Gigabit/s*, se hizo una revisión del estándar (802.1D-1998) utilizando un escala no lineal, con otra fórmula no especificada en el mismo y que recomienda la utilización de los siguientes valores.

802.1D-1998		
Velocidad del puerto		Costo
<i>Ethernet</i>	10 Mbps	100
<i>FastEthernet</i>	100 Mbps	19
<i>GigabitEthernet</i>	1 Gbps	4
<i>10-GigabitEthernet</i>	10 Gbps	2

Costos recomendados por la revisión del estándar original (802.1D-1998).

Cuando un BPDU es generado por el *switch* raíz este inicia con un *Root Path Cost* de cero. Cuando dicho BPDU sea recibido por la interfaz de otro *switch* este incrementará dicho campo utilizando el costo asociado a dicha interfaz (*Path Cost*), antes de reenviarlo hacia otros dispositivos quienes luego utilizarán el valor total del mismo para determinar las rutas con menor costo y así elegir sus respectivos puertos raíces.

A manera de ejemplo se vuelve a introducir la topología presentada anteriormente, con las velocidades asociadas a sus respectivos puertos.

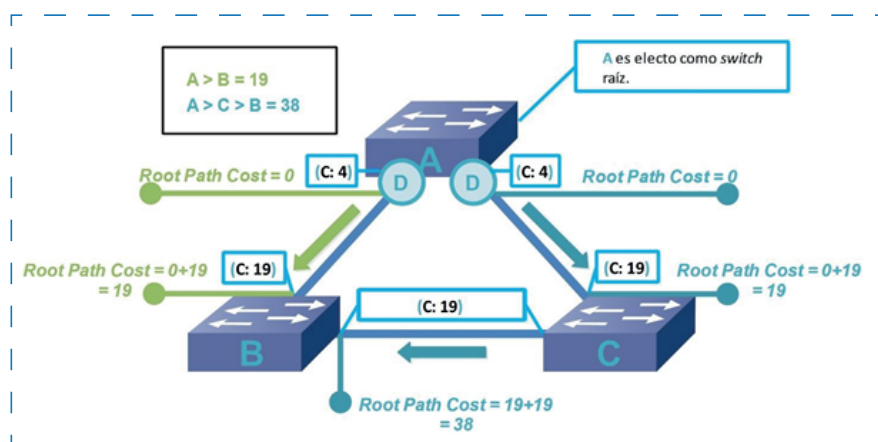


Elección de los puertos raíces. Interfaces y costos asociados.



STP

En este caso, el *switch* raíz generará un BPDUs con un *root path cost* inicial de cero, mismo que se incrementará con el costo asociado a cada interfaz por donde acceda, como se muestra a continuación.



Cálculo del root path cost.

El *switch* B recibirá dos BPDUs provenientes del *switch* raíz, una a través de su interfaz directamente conectada con un costo de 19 y otra que viene desde el *switch* C con un costo de 38 y seleccionará como puerto raíz aquella interface por donde ingreso el BPDUs con la mejor ruta.

Nótese que el costo está asociado con la velocidad de cada interface y no con cualquier velocidad negociada en el segmento.

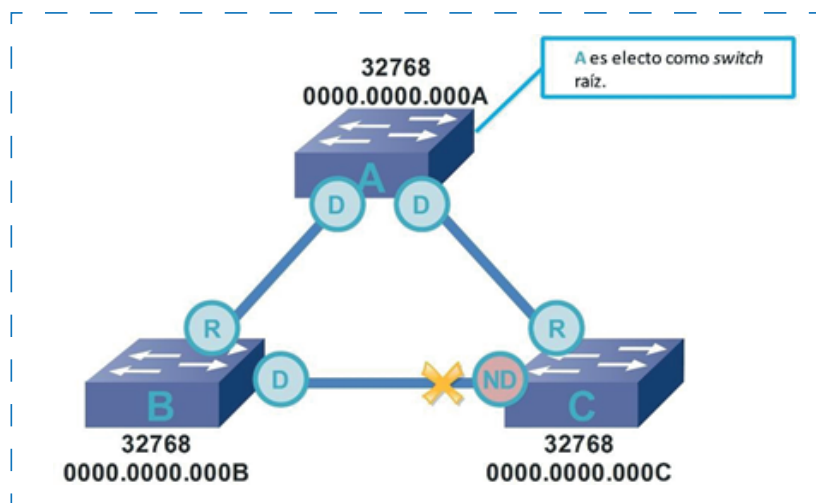
Después de hacer un análisis similar con el *switch* C, se establecen los puertos raíces y se determina que el enlace redundante es el que conecta dicho dispositivo con el *switch* B.

Una vez seleccionados los puertos raíces, se procede a seleccionar los puertos designados, uno por cada segmento.

En el caso del enlace redundante tanto *switch* B como *switch* C reciben BPDUs con el mismo costo hacia el *switch* raíz por lo que se utiliza el siguiente parámetro en la lista: el identificador del *switch* que envía el BPDUs (*Sender Bridge ID*).

Como B posee un *bridge ID* más bajo que el *switch* C sus puertos serán preferidos sobre los de este último si se llega a esta instancia, por lo que el puerto asociado con el primero ocupará la función de puerto designado, mientras que el extremo asociado con C será un puerto no designado.

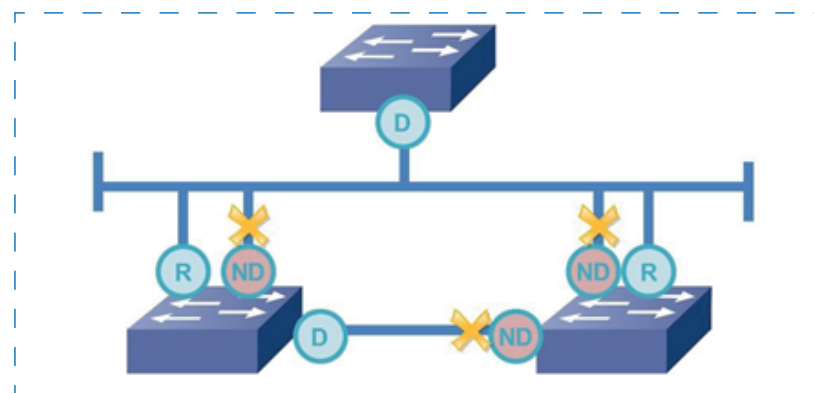
STP



Convergencia de Spanning Tree. Se muestran los puertos raíces (R), los puertos designados (D) y el no designado (ND) para romper el bucle a nivel lógico.

Los dos últimos parámetros de la lista (*Sender/Receiver port ID*) no son utilizados en el ejemplo anterior, lo que no quiere decir que no existan situaciones en donde deban utilizarse.

A manera de ejemplo, se presenta el siguiente caso (Donde deben utilizarse todos los parámetros presentados) el cual evidencia la versatilidad de STP,



Configurado correctamente Spanning Tree puede cumplir su propósito aun en casos extremos.



STP

PortFast

Una de las debilidades de *Spanning Tree* consiste en que este protocolo no ofrece una opción a través de la cual un puerto pueda hacer una distinción directa entre dispositivos, en otras palabras, STP funcionará de la misma manera independientemente del dispositivo que sea conectado.

Este comportamiento, inofensivo en un principio, empezó a generar problemas una vez las redes empezaron a hacerse más grandes y los elementos que las componían más modernos, llegando a ser un caso especialmente difícil el de los dispositivos finales conectados a la red a través de puertos sujetos a STP.

Dichos dispositivos (computadoras, impresoras, etc.) están destinados a satisfacer las necesidades de los usuarios por lo que entran y salen de la red a medida que son necesarios. Si los puertos asociados a los mismos forman parte del proceso STP, cada vez que un host cambie de estado (Ej.: Una computadora es apagada o encendida), se generará una notificación de cambio de topología (TCN BPDU) lo que provocará que todos los *switches* renueven su base de datos de direcciones MAC en un tiempo más corto de lo usual (A pesar de no existir un cambio significativo) provocando inestabilidad.

Además, a pesar que la conexión con un solo *host* no puede introducir un bucle en la topología, este debe esperar de 30 a 50 segundos para incorporarse a la red, retraso que no es aceptable hoy en día.

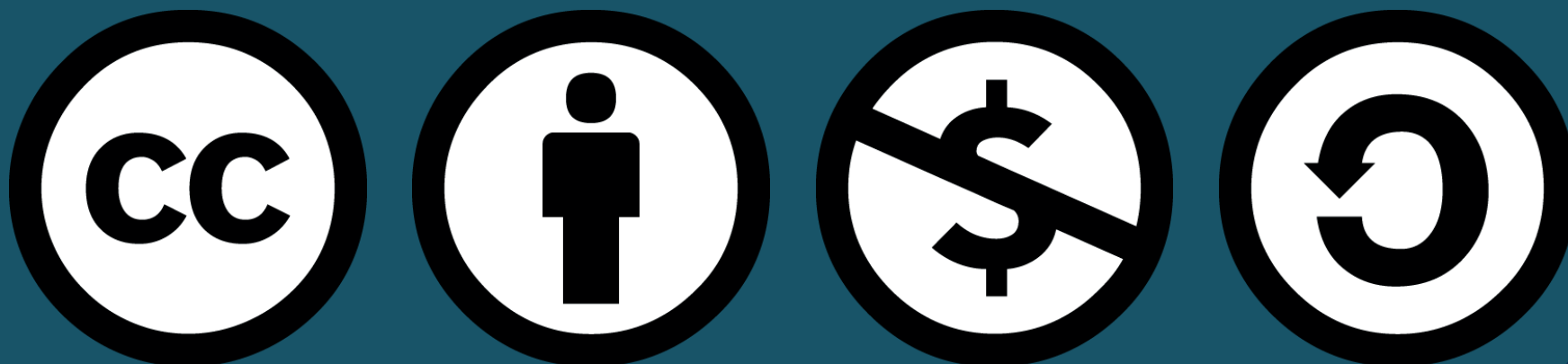
En vista de estos problemas, desactivar *Spanning Tree* en los puertos destinados para estos dispositivos puede parecer tentador, sin embargo, debido a la gran probabilidad de que los usuarios introduzcan bucles de manera accidental en la topología STP jamás debe desactivarse.

Ante esta situación, Cisco creó una mejora propietaria llamada *Portfast* para mitigar estos problemas.

Un puerto configurado con Portfast, es un puerto que sigue ejecutando STP (sigue enviando BPDUs) con la diferencia que un cambio en su estado no genera un TCN (Por lo que es más estable) y que su estado inicial es transmitiendo (*Forwarding*) en vez de bloqueando (*Blocking*).

Esto permite una red más estable y que los dispositivos finales se incorporen a la red inmediatamente, con la desventaja de la introducción de la posibilidad de la formación de bucles temporales en caso otro tipo de equipo como *hubs*, *switches*, *Access points*, etc., sean conectados a puertos configurados con dicha característica, debido a que cuando estos se inician pasan a transmitir información inmediatamente.

Los bucles aludidos son de carácter temporal, debido a que al recibir un BPDU los puertos configurados con *Portfast* deshabilitan esta característica.



Diseño y edición:

María Esther Pineda
Carolina Villatoro

Descargo de Responsabilidad

El autor y los colaboradores de este trabajo han hecho su mejor esfuerzo en la preparación del mismo para asegurar que su contenido sea lo más exacto posible, sin embargo, no se hacen responsables por el uso de la información en este documento así como de errores u omisiones que pudieran resultar en pérdida de cualquier tipo.

La información está proporcionada “como está” para ser utilizada bajo “su propia cuenta y riesgo”.