

Comprensión y configuración de Backbone Fast en switches Catalyst

Contenido

[Introducción](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[BPDU y cómo compararlos](#)

[¿Cómo se recupera STP de una falla de link indirecto?](#)

[Mejoras en Backbone \(tronco principal\) rápidas en STP estándar](#)

[Detectar fallas de link indirecto](#)

[Reaccionar a fallas de link indirecto](#)

[PDU de consulta de link raíz](#)

[Ejemplo de situación con la función Backbone Fast habilitada](#)

[Configuración de Backbone Fast para CatOS y Cisco IOS](#)

[Configuración para CatOS](#)

[Configuración para Cisco IOS](#)

[Información Relacionada](#)

Introducción

Este documento describe cómo configurar la estructura básica rápidamente. Backbone Fast es una función propietaria de Cisco que, una vez habilitada en todos los switches de una red de Bridge, puede salvar un switch hasta 20 segundos (max_age, intervalo máximo) cuando se recupera de una falla de link indirecto. Después de una revisión rápida de algunos fundamentos de STP (Spanning-Tree Protocol), podrá ver el escenario de falla exacto al que se aplica la función Backbone Fast y cómo configurarlo para los switches Catalyst que ejecutan el software CatOS y Cisco IOS®.

Prerequisites

Requirements

No hay requisitos específicos para este documento.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Catalyst 2950 Series Switches que ejecutan Cisco IOS Software Release 12.1(6)EA2 y posteriores

- Catalyst 3550 Series Switches que ejecutan Cisco IOS Software Release 12.1(4)EA1 y posteriores
- Switches Catalyst serie 4000 que ejecutan CatOS 5.1(1a) y posteriores
- Catalyst 4500/4000 Series Switches que ejecutan Cisco IOS Software Release 12.1(8a)EW y posteriores
- Catalyst 5500/5000 Series Switches que ejecutan CatOS versión 4.1(1) y posteriores
- Switches Catalyst 6500/6000 Series que ejecutan CatOS versión 5.1(1)CSX y posteriores
- Catalyst 6500/6000 Series Switches que ejecutan Cisco IOS Software Release 12.0-7XE y posteriores

BPDU y cómo compararlos

Las unidades de datos de protocolo de puente (BPDU) se pueden clasificar estrictamente según los campos que llevan. Entre estos campos se encuentran el ID del puente raíz, el costo de trayectoria a la raíz y el ID del puente del remitente. Una BPDU se considera mejor que otra BPDU por estas razones:

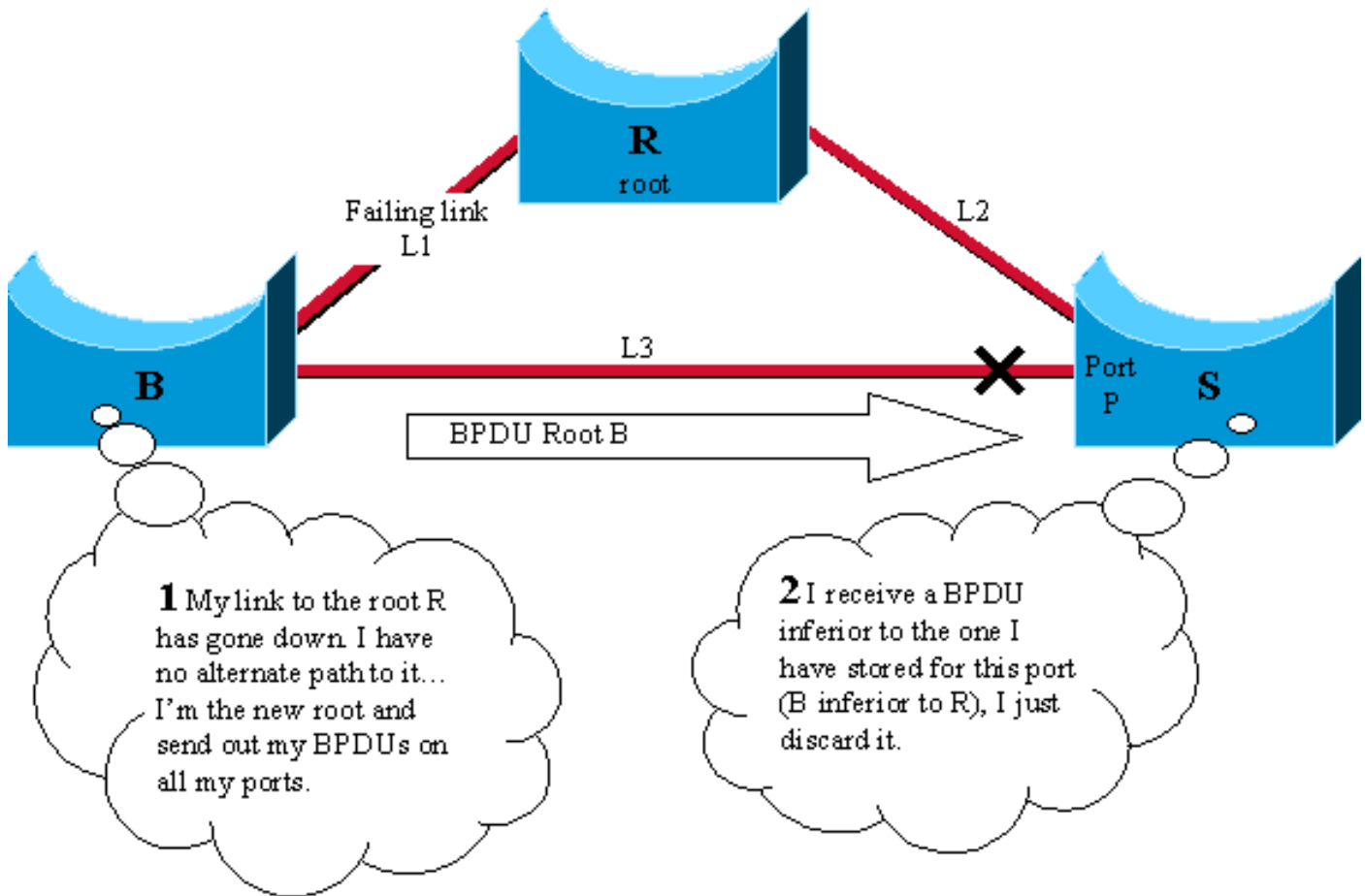
- Cuando una BPDU lleva un ID de bridge raíz mejor que otra. Cuanto más bajo sea el valor, mejor.
- Cuando los valores de ID del puente raíz sean iguales, entonces es mejor el BPDU con el costo de trayecto más bajo a la raíz.
- Cuando los valores de ID de puente raíz son iguales y los costos para la raíz son los mismos, entonces la BPDU con el mejor ID de puente de remitente es mejor. Cuanto más bajo sea el valor, mejor.

Hay otras variables que luego pueden actuar como desempate. Sin embargo, cuanto mejor sea una BPDU, mejor será el acceso al mejor puente raíz.

Un bridge que recibe una BPDU en un puerto mejor que el que envía, pone este puerto en modo de bloqueo a menos que sea su puerto raíz. Esto significa que en el segmento conectado a este puerto existe otro puente que constituye un puente designado. Un bridge almacena el valor de la BPDU en un puerto enviado por el bridge designado actual.

¿Cómo se recupera STP de una falla de link indirecto?

Esto ilustra cómo el STP se comporta cuando tiene que recalcularse después de una falla de link indirecto, es decir, cuando un bridge tiene que cambiar el estado de algunos de sus puertos debido a una falla en un link que no está directamente conectado a él.



Considere este diagrama, que implica tres switches R, B y S en una topología de malla completa. Suponga que R es el root bridge y B es el root bridge de respaldo. S bloquea su puerto P y B es el puente designado para el link L3.

1. Si el link L1 se desactiva, el switch B detecta inmediatamente la falla y asume que es la raíz. Comienza a enviar BPDU a S y afirma ser la nueva raíz.
2. Cuando S recibe esta nueva BPDU desde B, se da cuenta de que es inferior a la ya almacenada para el puerto P y la ignora.
3. Después de que caduque el temporizador `max_age` (20 segundos de forma predeterminada), la BPDU almacenada en S para el puerto P se desactualiza. El puerto pasa inmediatamente a escuchar y S comienza a enviar su mejor BPDU a B.
4. Tan pronto como B recibe la BPDU de S, deja de enviar su BPDU.
5. El puerto P pasa al estado de reenvío a través de los estados de escuchar y aprender. Esto requiere el doble del valor `fw_delay`, un tiempo adicional de 30 segundos. A continuación, se restablece la conectividad completa.

Se tardó el valor `max_age` (20 segundos) más el doble del valor `fw_delay` (2 x 15 segundos) en recuperarse de esta falla de link indirecto. Esto es 50 segundos con los parámetros predeterminados. La función Backbone Fast propone guardar `max_age` (20 segundos). Para hacer esto, se desactualiza inmediatamente después de que el puerto reciba BPDU inferiores.

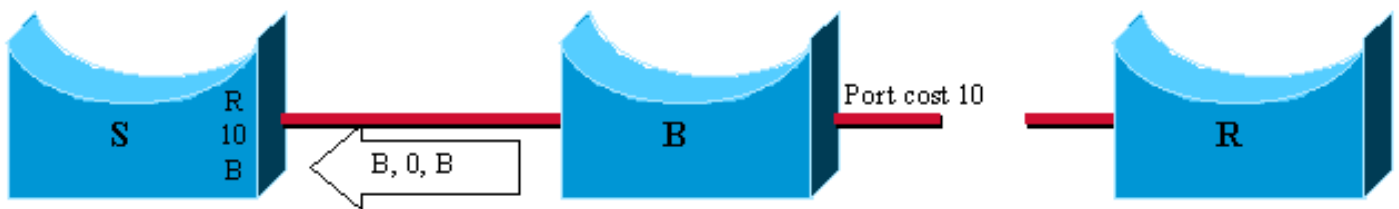
Mejoras en Backbone (tronco principal) rápidas en STP estándar

Con el ejemplo anterior, el STP invalida la información que se vuelve incorrecta debido a una falla de link indirecto. Para hacerlo, espera pasivamente a `max_age`. Para deshacerse de este retardo `max_age`, backbone fast introduce dos mejoras:

- La capacidad de detectar una falla de link indirecto lo antes posible. Esto se logra mediante el seguimiento de las BPDU inferiores que envía un bridge designado cuando experimenta una falla de link directo.
- Mecanismo que permite verificar inmediatamente si la información BPDU almacenada en un puerto sigue siendo válida. Esto se implementa con una nueva unidad de datos de protocolo (PDU) y la consulta de link raíz, a la que se hace referencia en este documento como la PDU de RLQ.

Detectar fallas de link indirecto

Si se recibe una BPDU inferior en un puerto desde nuestro bridge designado, este bridge ha perdido la raíz y comienza a anunciar una raíz con un ID de bridge más alto, una raíz peor que la nuestra.



In this case, B lost the root and sends a BPDU with root id B, path cost 0 and bridge id B. It is inferior to the one that S had stored, because R is a better root than B.

El comportamiento habitual con respecto a las especificaciones del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) es simplemente ignorar cualquier BPDU inferior. Backbone Fast los utiliza porque tan pronto como se recibe uno, es seguro que se produjo un error en la trayectoria hacia la raíz y que debe desconectar al menos un puerto.

Nota: Una falla de link indirecto puede ocurrir sin ninguna generación de BPDU inferior en la red. Simplemente agregue un hub en el diagrama anterior:



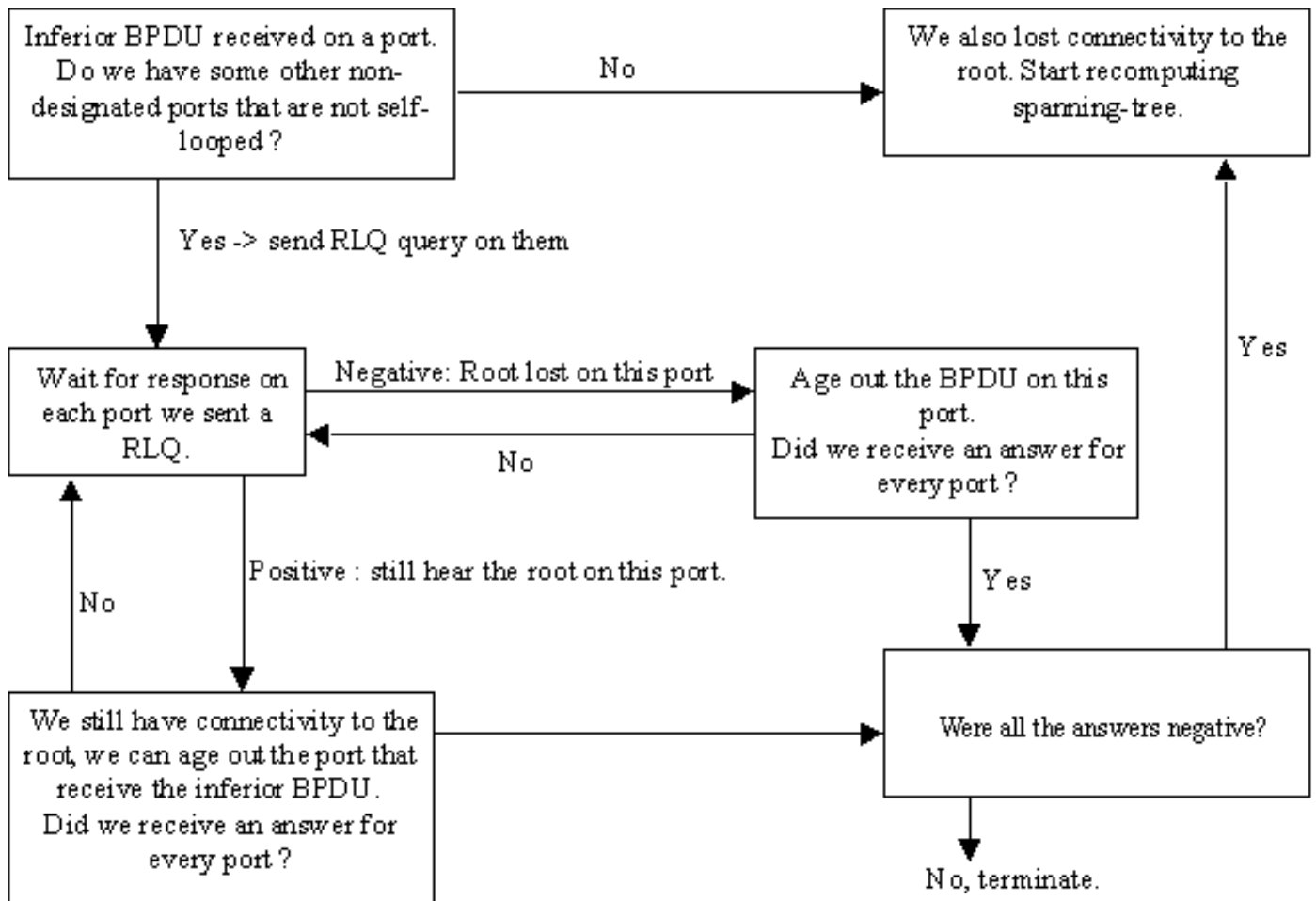
La falla de link ocurre entre el bridge raíz R y el hub. B no detecta que el link se desactiva y espera max_age antes de que diga ser la nueva raíz. Recuerde que el mecanismo sólo funciona si un puente detecta una falla de link directo.

Sólo realice el seguimiento de BPDU inferiores enviadas por el puente designado. Dado que ésta es la BPDU que está almacenada en el puerto. Si, por ejemplo, un puente recién insertado comienza a enviar BPDU inferior, no inicia la función backbone fast.

Reaccionar a fallas de link indirecto

Cuando se detecta una BPDU inferior en un puerto no designado, se activa la segunda fase de la estructura básica rápida. En lugar de esperar pasivamente a que max_age se agote el tiempo de

espera de los puertos que pueden verse afectados por la falla, se introduce una forma proactiva de probarlos inmediatamente mediante la PDU de RLQ. RLQ se utiliza para lograr un tipo de ping para la raíz en un puerto no designado y permite rápidamente confirmar si la BPDU almacenada en un puerto aún es válida o necesita ser descartada.



Al recibir una BPDU inferior desde un puente designado, envíe una PDU de RLQ en todos los puertos no designados, excepto el puerto donde recibió la BPDU inferior y los puertos de loop intrínseco. Esto es para verificar que aún se oye de la raíz en los puertos donde se utiliza para recibir BPDU. El puerto en el que recibió la BPDU inferior se excluye porque ya sabe que sufrió de una falla, los puertos auto-loop y designados no son útiles, ya que no conducen a la raíz.

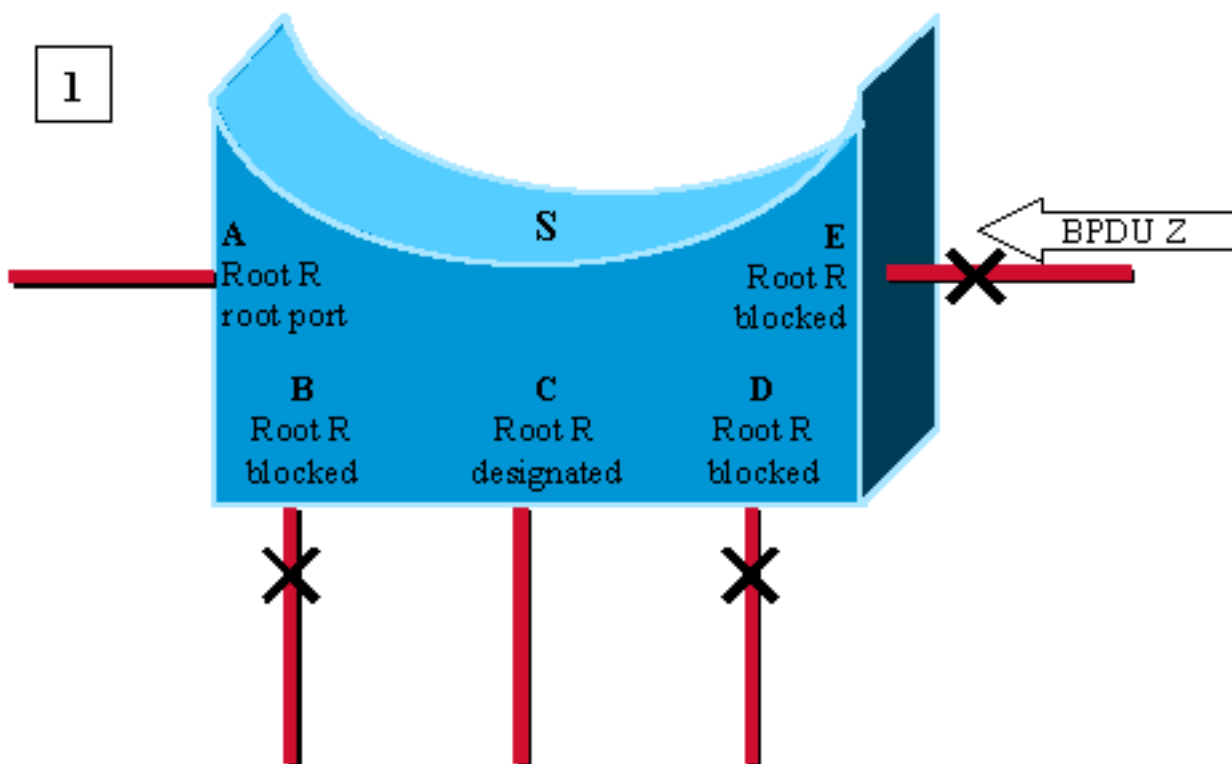
Al recibir una respuesta RLQ en un puerto, si la respuesta es negativa, el puerto perdió la conexión con la raíz y puede desconectar su BPDU. Además, si todos los otros puertos no designados ya recibieron una respuesta negativa, el puente entero pierde la raíz y puede iniciar el cálculo STP desde el principio.

Si la respuesta confirma que aún puede acceder al bridge raíz a través de este puerto, puede desconectar inmediatamente el puerto en el que inicialmente recibimos la BPDU inferior.

En este ejemplo, los puertos A, B, D y E son puertos no designados para el switch S. A es el puerto raíz y los otros están bloqueando. Cuando E recibe un BPDU (1) inferior, la estructura básica rápida comienza a acelerar el cálculo STP.

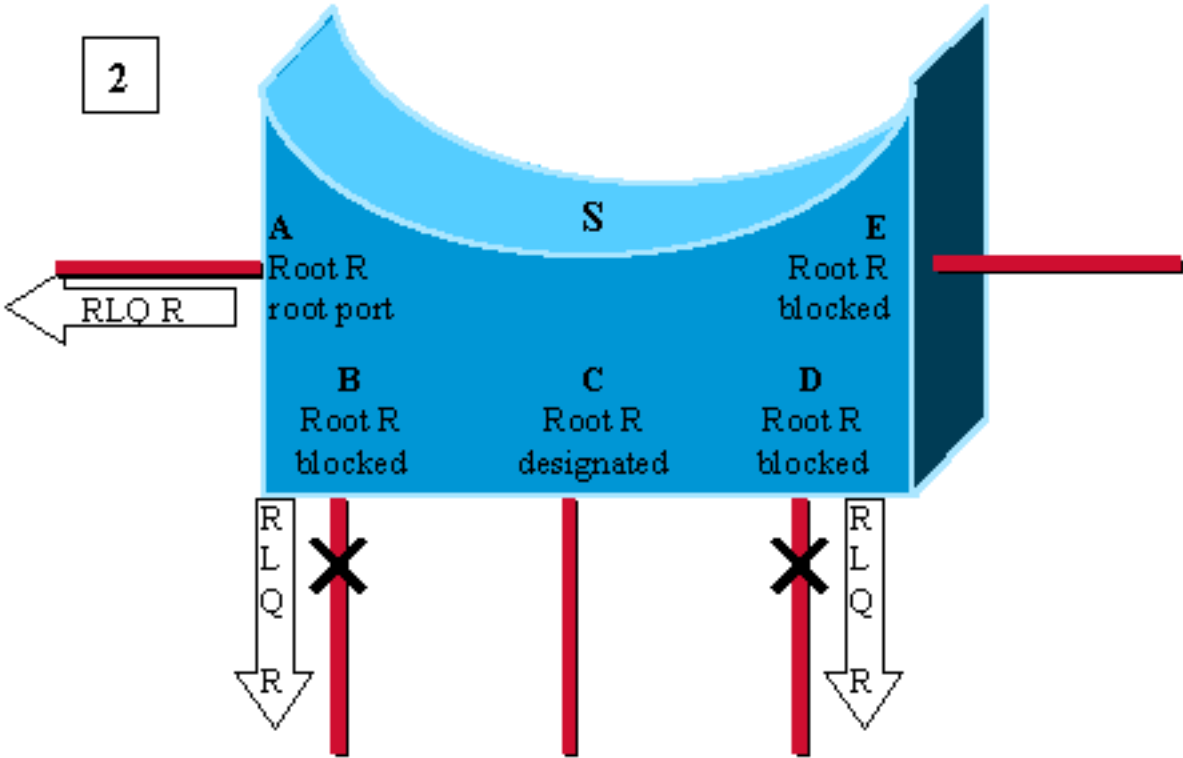
Envíe una solicitud RLQ, que busca la raíz R en todos los puertos no designados excepto E (2). Las respuestas especifican a qué raíz se puede acceder a través de estos puertos. La respuesta RLQ que recibe D especifica que D perdió su trayectoria hacia la raíz R. Antigüedad de su BPDU inmediatamente (3). Los Puertos A y B reciben confirmación de que aún cuentan con un trayecto

a R (4). Por consiguiente, como el switch S aún tiene conectividad con la raíz, hace que el puerto E se venza inmediatamente y continúa con las reglas STP habituales (5).



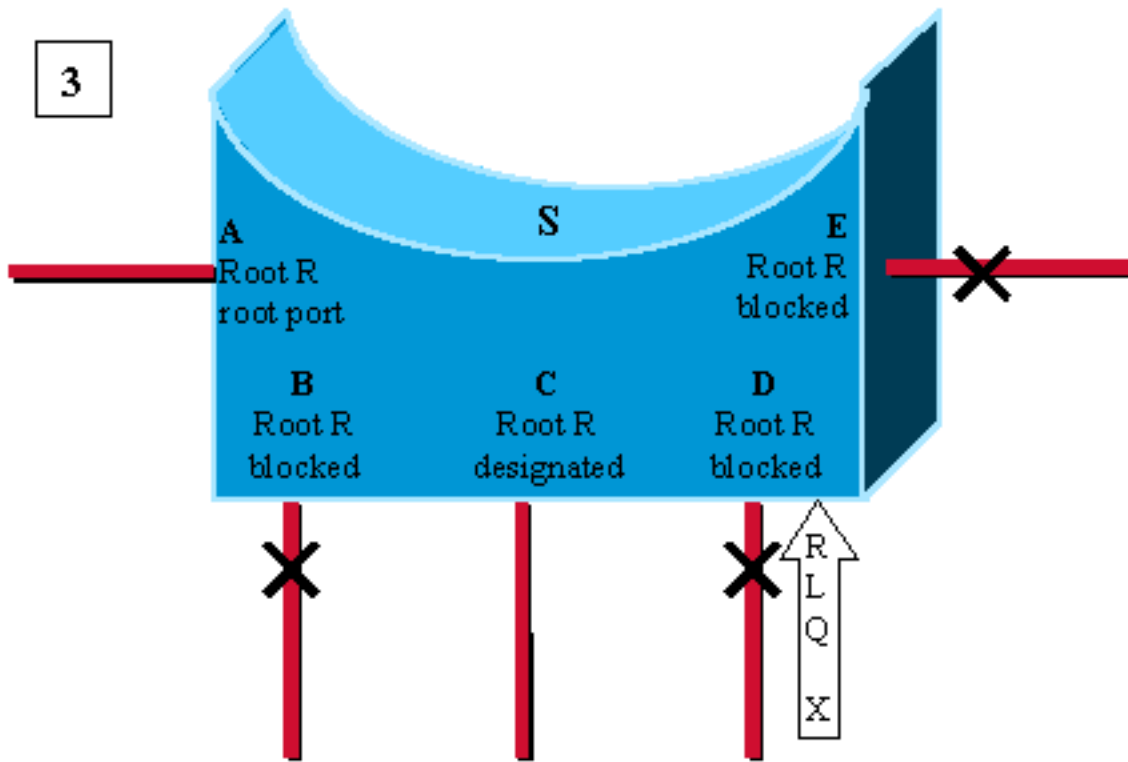
Port E receives an inferior BPDU, advertising root Z instead of root R stored on the different ports.

2



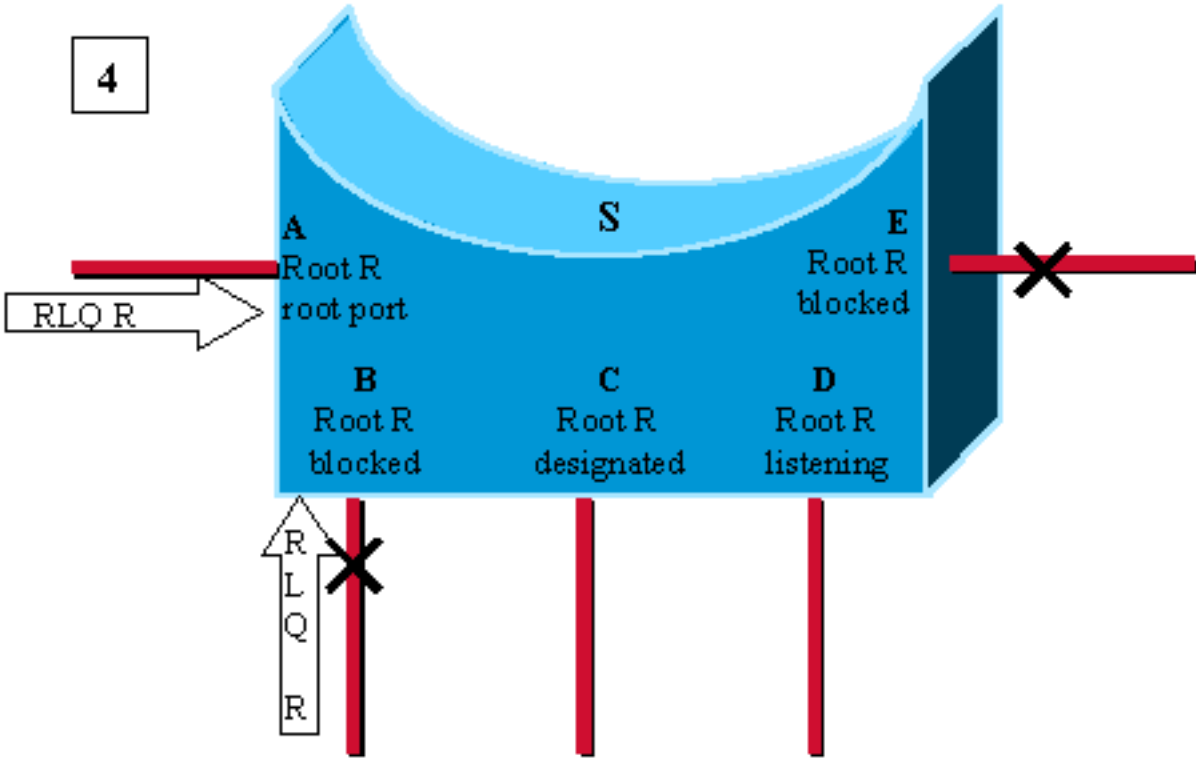
Switch S needs to recheck all its other non-designated ports. It sends out a RLQ request for root R on ports A,B and D.

3



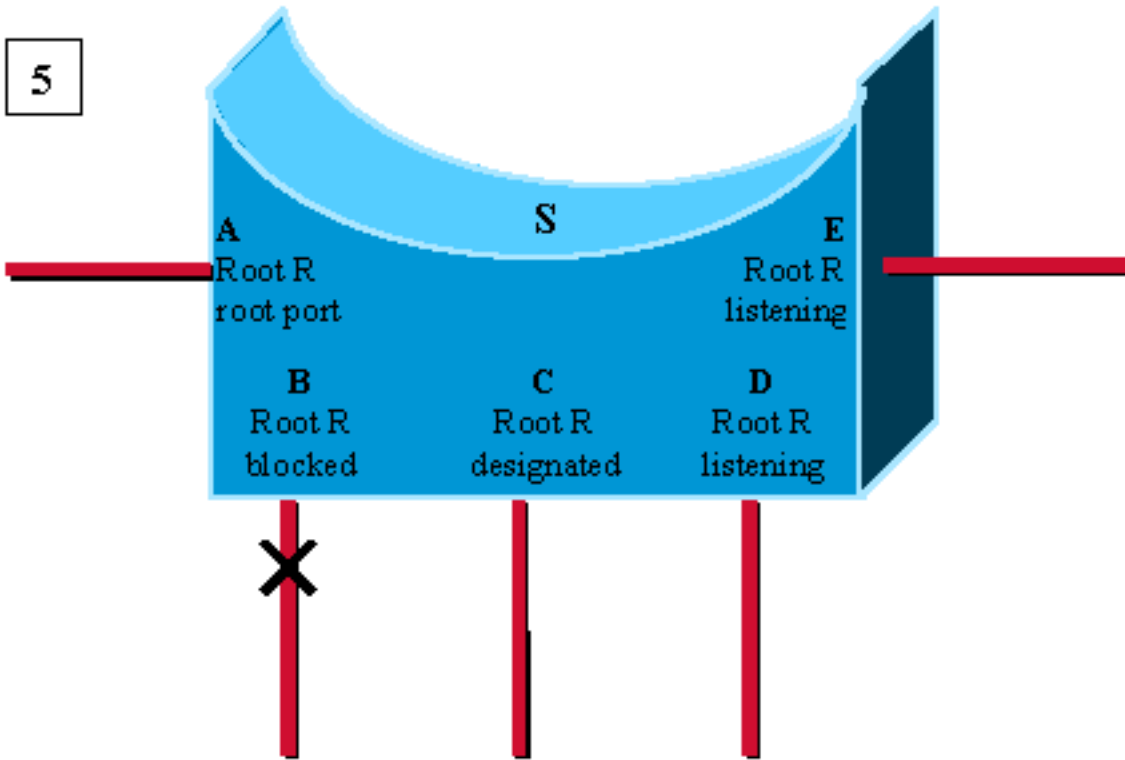
Port D is the first to receive and RLQ response from bridge X claiming to be the root. It is a negative response: D has lost connectivity to the root R. We age out immediately the BPDU on port D and go to listening. As we don't know if we still have connectivity to the root R, we don't age out port E yet.

4



Here, A and B receive a RLQ response confirming R as being the root. As switch S still has connectivity to the root, we can age out immediately the BPDUs stored on port E.

5



Port E transitions to listening, without waiting for `max_age`. Usual spanning-tree rules then apply to determine whether E and D will eventually go to blocking or forwarding.

En un caso en el que el switch recibió solamente respuestas con una raíz diferente de R, considere inmediatamente la raíz como cálculo STP perdido y reiniciado desde el principio. Tenga en cuenta que este caso también ocurre cuando el único puerto no designado (y sin loop propio) en el puente es el puerto raíz y usted recibe una BPDU inferior en este puerto.

PDU de consulta de link raíz

Las dos formas de RLQ son solicitudes RLQ y respuestas RLQ.

La solicitud RLQ se envía en un puerto donde usualmente recibe BPDU, para verificar que todavía tiene conectividad con la raíz a través de este puerto. Especifique en la solicitud qué puente es su raíz y la respuesta RLQ finalmente regresa con un puente raíz al que se puede acceder a través de este puerto. Si las dos raíces son iguales, la conectividad sigue activa, de lo contrario, se pierde.

Un bridge que recibe una solicitud RLQ responde inmediatamente si sabe que ha perdido la conexión con la raíz consultada porque tiene un bridge raíz diferente al especificado en la consulta RLQ y si es la raíz.

Si este no es el caso, entonces, reenvía la consulta hacia la raíz a través de su puerto raíz.

Los puertos designados reciben infinitas respuestas RLQ. El emisor de la solicitud RLQ coloca su identificador de puente en la PDU. Esto es para garantizar que, cuando reciba una respuesta a su

consulta, no inunde la respuesta en sus puertos designados.

El RLQ PDU tiene la misma estructura de paquete que un STP BPDU normal. La única diferencia es que se utilizan dos direcciones SNAP diferentes específicas de Cisco: una para la solicitud y otra para la respuesta.

Este es el formato BPDU estándar:

DA SA Longitud DSAP SSAP CNTL SNAP PDU

El campo PDU es:

Identificador de Protocolo	Versión	Tipo de mensaje	Indicadores	ID de raíz	Costo de trayecto raíz
ID del emisor	Identificación del puerto	Edad del mensaje	max age	Tiempo de Hello	demora de recepción

El tipo de mensaje utilizado en la PDU también es diferente de la BPDU estándar.

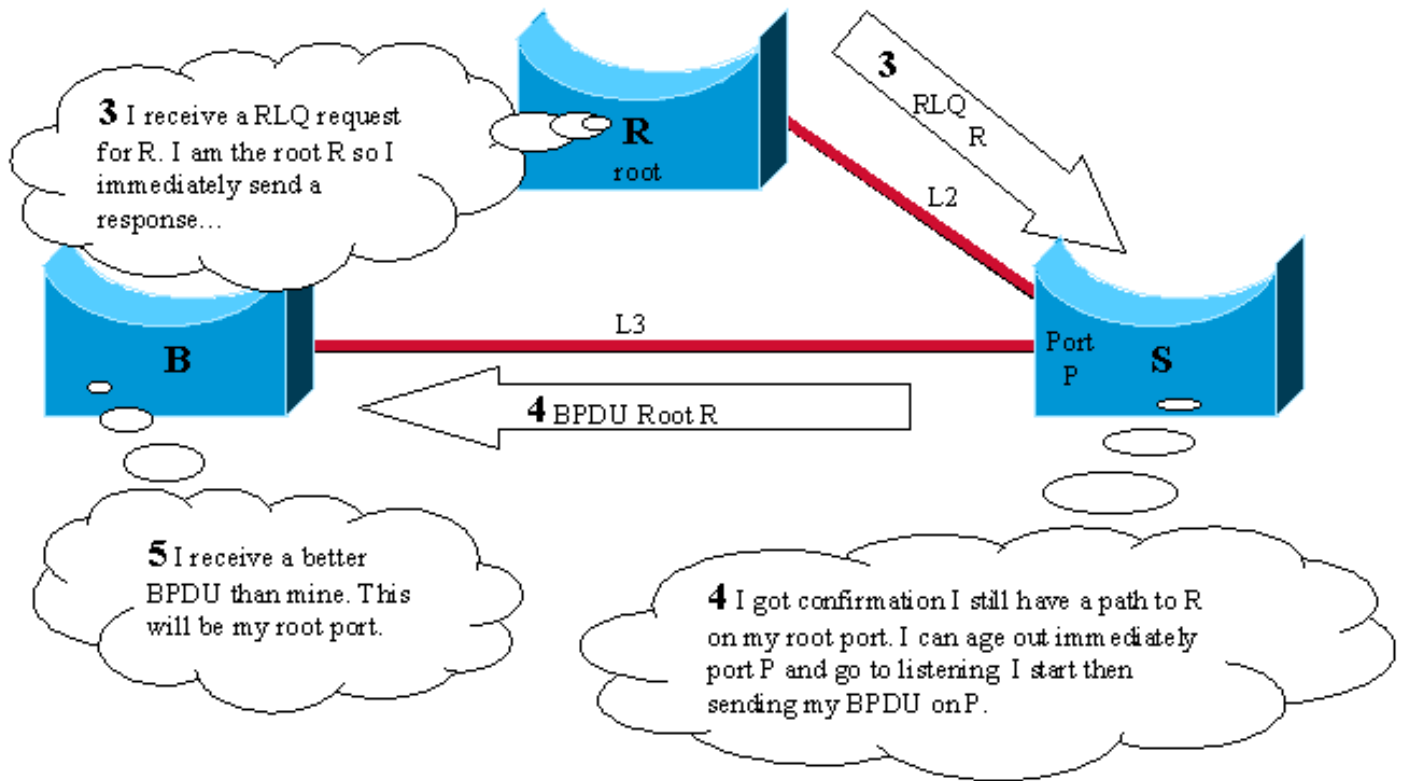
Los únicos campos que se utilizan son el ID de raíz y el ID de puente del remitente.

Esta función específica de Cisco debe configurarse en todos los switches de la red para procesar estas PDU.

Ejemplo de situación con la función Backbone Fast habilitada

Esta situación se basa en el primer ejemplo, pero, esta vez con la estructura básica rápida habilitada en los tres switches.

1. La primera etapa es igual a la que se explicó anteriormente.
2. Tan pronto como S recibe la BPDU inferior de B, comienza a reconfirmar sus puertos no designados en lugar de esperar max_age. Envía una consulta RLQ en su puerto raíz para root bridge R.
3. El puente raíz R recibe la consulta y responde inmediatamente con una respuesta RLQ que especifica que todavía hay una raíz R en esa dirección.
4. S ya ha verificado todos sus puertos no designados y aún posee conectividad a la raíz. A continuación, puede desconectar inmediatamente la información almacenada en el puerto P. P. y pasar a escuchar y comienza a enviar BPDU. En esa etapa, ya ha guardado los segundos max_age y, a continuación, se aplica el algoritmo estándar de árbol de extensión (STA).
5. B recibe la mejor BPDU de S (R mejor raíz que B) y considera ahora los puertos que llevan a L3 como su puerto raíz.



Configuración de Backbone Fast para CatOS y Cisco IOS

Cuando se utiliza, la estructura básica rápida debe estar habilitada en todos los switches de la red porque la estructura básica rápida requiere el uso del mecanismo de solicitud y respuesta de RLQ para informar a los switches de la estabilidad de la ruta raíz. El protocolo RLQ está activo solamente cuando la estructura básica rápida está habilitada en un switch. Además, la red también puede tropezar con problemas con la inundación de RLQ, si la estructura básica rápida no está habilitada en todos los switches. De forma predeterminada, la estructura básica rápida está desactivada.

Los switches Catalyst 2900XL y 3500XL no admiten Backbone Fast. En general, debe habilitar la estructura básica rápida si el dominio del switch contiene estos switches además de otros switches Catalyst soportados. Cuando implementa la estructura básica rápida en entornos con switches XL, bajo topologías estrictas, puede habilitar la función donde el switch XL es el último switch en línea y está conectado solamente al núcleo en dos lugares. No implemente esta función si la arquitectura de los switches XL está en modo de cadena de margarita.

No necesita configurar la estructura básica rápidamente con RSTP o IEEE 802.1w porque el mecanismo se incluye de forma nativa y se habilita automáticamente en RSTP. Para obtener más información sobre RSTP o IEEE 802.1w, consulte [Ejemplo de Configuración de Migración de Spanning Tree de PVST+ a Rapid-PVST](#).

Configuración para CatOS

Para los Catalyst 4000, 5000 y 6000 Series Switches que ejecutan CatOS, utilice estos comandos para habilitar la estructura básica rápida globalmente para todos los puertos y verificar la configuración.

```
Console> (enable) set spantree backbonefast enable
```

Backbonefast enabled for all VLANs

```
Console> (enable) show spantree backbonefast
```

```
! This command show that the backbonefast feature is enabled. Backbonefast is enabled. Console> (enable)
```

Para mostrar estadísticas rápidas de estructura básica:

```
Console> (enable) show spantree summary
```

Summary of connected spanning tree ports by vlan

Uplinkfast disabled for bridge.

Backbonefast enabled for bridge.

```
Vlan  Blocking Listening Learning Forwarding STP Active
```

```
-----  
1      0          0          0          1          1
```

```
Blocking Listening Learning Forwarding STP Active
```

```
-----  
Total  0          0          0          1          1
```

BackboneFast statistics

```
! The show spantree summary command displays all backbonefast statistics. -----
```

```
- Number of inferior BPDUs received (all VLANs): 0 Number of RLQ req PDUs received (all VLANs):  
0 Number of RLQ res PDUs received (all VLANs): 0 Number of RLQ req PDUs transmitted (all VLANs):  
0 Number of RLQ res PDUs transmitted (all VLANs): 0 Console> (enable)
```

Configuración para Cisco IOS

Para los switches Catalyst que se ejecutan con el software Cisco IOS, utilice estos comandos para habilitar la estructura básica rápida globalmente para todas las interfaces.

```
CAT-IOS# configure terminal
```

```
CAT-IOS(config)# spanning-tree backbonefast
```

```
CAT-IOS(config)# end
```

```
CAT-IOS#
```

Para verificar que la estructura básica rápida está habilitada y para mostrar estadísticas:

```
CAT-IOS# show spanning-tree backbonefast
```

```
BackboneFast          is enabled
```

BackboneFast statistics

```
-----  
Number of transition via backboneFast (all VLANs)           : 0  
Number of inferior BPDUs received (all VLANs)              : 0  
Number of RLQ request PDUs received (all VLANs)            : 0  
Number of RLQ response PDUs received (all VLANs)           : 0  
Number of RLQ request PDUs sent (all VLANs)                 : 0  
Number of RLQ response PDUs sent (all VLANs)                : 0  
CAT-IOS#
```

Información Relacionada

- [Uso de Portfast y otros comandos para solucionar retrasos al iniciar la conectividad de la estación de trabajo](#)
- [Comprensión y configuración de la función Cisco UplinkFast](#)

- [Mejoras del protocolo de árbol de expansión usando las funciones de Loop Guard y BPDU Skew detección de desviación](#)
- [Mejoras de la Protección de Spanning Tree PortFast BPDU](#)
- [Ejemplo de configuración de migración de árbol de expansión de PVST+ a Rapid-PVST](#)
- [Protocolo de árbol de extensión: soporte tecnológico](#)
- [Switches - Asistencia para productos](#)
- [Switching de LAN - Soporte tecnológico](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)