

Compreenda e configurar o Backbone Fast em Catalyst Switches

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[BPDUs e como compará-los](#)

[Como o STP se recupera de uma falha indireta de enlace](#)

[Melhorias de Backbone Fast no STP padrão](#)

[Detecte falhas indireta do link](#)

[Reaja às falhas indireta do link](#)

[O PDU de Root Link Query](#)

[Cenário de exemplo com o recurso Backbone Fast ativado](#)

[Configurar o Backbone Fast para Cactos e Cisco IOS](#)

[Configuração para CatOS](#)

[Configuração para o Cisco IOS](#)

[Informações Relacionadas](#)

Introdução

Este original descreve como configurar o Backbone Fast. O Backbone Fast é uma característica exclusiva da Cisco que, quando habilitada em todos os switches de uma rede de bridge, pode salvar um switch em até 20 segundos (max_age) ao se recuperar de uma falha indireta do link. Após uma revisão rápida de alguns princípios do Spanning Tree Protocol (STP), você poderá ver o cenário de falha exato ao qual o Backbone Fast se aplica e como configurá-lo para switches Catalyst que executam nos softwares CatOS e Cisco IOS®.

Pré-requisitos

Requisitos

Não existem requisitos específicos para este documento.

[Componentes Utilizados](#)

As informações neste documento são baseadas nestas versões de software e hardware:

- Catalyst 2950 Series Switch que executam o Cisco IOS Software Release 12.1(6)EA2 e mais tarde
- Catalyst 3550 Series Switch que executam o Cisco IOS Software Release 12.1(4)EA1 e mais tarde

- Catalyst 4000 Series Switch que executam Cactos 5.1(1a) e mais atrasado
- Switches do 4500/4000 Series do catalizador que executa o Cisco IOS Software Release 12.1(8a)EW e Mais Recente
- Series Switch do Catalyst 5500/5000 que executam o 4.1(1) da versão cactos e mais tarde
- Catalyst 6500/6000 series switch que executa a versão cactos 5.1(1)CSX e mais tarde
- Catalyst 6500/6000 series switch que executa o Cisco IOS Software Release 12.0-7XE e Mais Recente

BPDU e como compará-los

O bridge protocol data units (BPDU) pode restritamente ser classificado pelos campos que levam. Entre estes campos são o ID de bridge raiz, os custos de caminho à raiz, e o ID de Bridge de remetente. Um BPDU é considerado melhor do que um outro BPDU por estas razões:

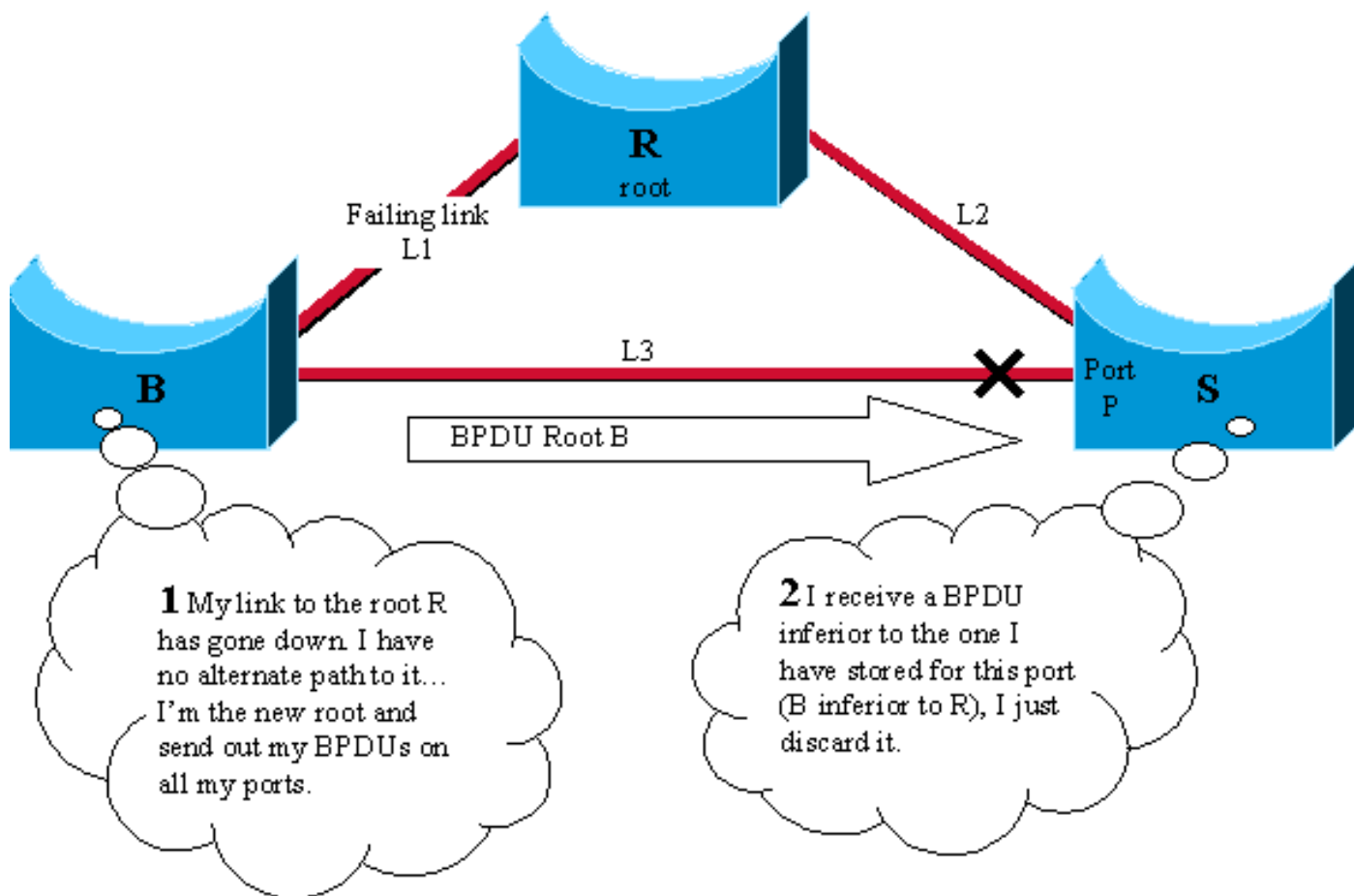
- Quando um BPDU levar um ID de bridge raiz melhor do que outro. Mais baixo o valor, melhor.
- Quando os valores de identificação do Root Bridge são iguais, então o BPDU com o custo de caminho inferior para a raiz é melhor.
- Quando os valores de ID de Root Bridge são igual e os custos à raiz são os mesmos, a seguir o BPDU com o ID de Bridge de remetente melhor é melhor. Mais baixo o valor, melhor.

Há outras variáveis que então podem atuar como um tiebreaker. Contudo, o melhor um BPDU, melhor o acesso ao melhor bridge-raiz.

Uma ponte que receba um BPDU em uma porta melhor do que essa ele manda, põe esta porta no modo de bloqueio a menos que for sua porta de raiz. Isso significa que no segmento conectado a essa porta, existe outra ligação designada. Uma ponte armazena o valor do BPDU em uma porta enviada pelo bridge designada atual.

Como o STP se recupera de uma falha indireta de enlace

Isto ilustra como o STP se comporta quando tem que voltar a calcular após uma falha indireta do link, isto é, quando uma ponte tem que mudar o estado de algumas de suas portas devido a uma falha em um link que não lhe esteja anexado diretamente.



Considere este diagrama, que envolve três Switches R, B, e S inteiramente em uma topologia em malha. Supõe que R é o bridge-raiz e B é o root bridge de backup. S obstrui sua porta P e B é o bridge designada para o link L3.

1. Se o link L1 vai para baixo, o switch B detecta imediatamente a falha e supõe-na que é a raiz. Começa enviar BPDU a S e reivindica ser a raiz nova.
2. Quando S recebe esse novo BPDU de B, ele percebe que é inferior àquele que ele tinha armazenado para a porta P e o ignora.
3. Depois que o temporizador do `max_age` expira (20 segundos à revelia), o BPDU armazenado em S para a porta P envelhece para fora. A porta vai imediatamente à escuta e S começa enviar seu melhor BPDU ao B.
4. Assim que B receber o BPDU de S, para de enviar seu BPDU.
5. A porta P passa para o estado de encaminhamento através dos estados de escuta e de aprendizagem. Esse procedimento requer duas vezes o valor de `fw_delay`; ou seja, 30 segundos adicionais. A conectividade direta é restaurada então.

Tomou o valor do `max_age` (20 segundos) mais duas vezes o valor `fw_delay` (segundos 2×15) para recuperar desta falha indireta do link. Corresponde a 50 segundos com os parâmetros padrão. A característica do Backbone Fast propõe salvar o `max_age` (20 segundos). A fim fazer isto, envelhece para fora imediatamente depois que a porta recebe BPDU inferiores.

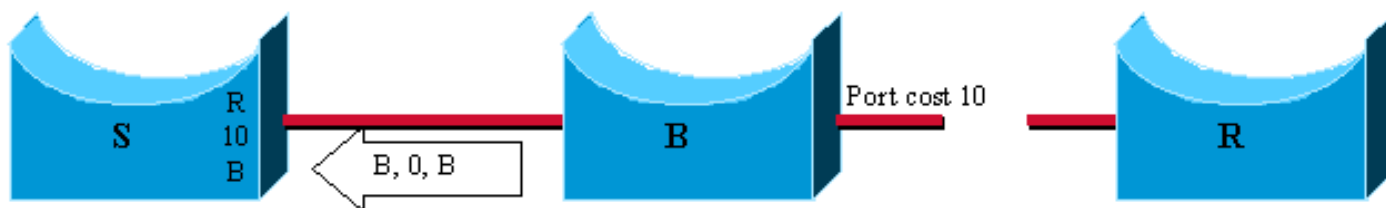
Melhorias de Backbone Fast no STP padrão

Com o exemplo anterior, o STP invalida a informação que se torna errada devido a uma falha indireta do link. A fim fazer isto, espera passivamente o `max_age`. A fim obter livrou deste atraso do `max_age`, Backbone Fast introduz dois realces:

- A habilidade de detectar uma falha indireta de link, o mais breve possível. Isto é conseguido seguindo os BPDU inferiores que um bridge designada envia quando experimenta uma falha de link direto.
- Um mecanismo que permita uma verificação imediata da verificação se a informação bpdv armazenada em uma porta é ainda válida. Isto é executado com uma unidade de dados de protocolo (PDU) nova e o Root Link Query, referidos neste original como o RLQ PDU.

Detecte falhas indireta do link

Se um BPDU inferior é recebido em uma porta de nosso bridge designada, a seguir esta ponte perdeu a raiz e começa a anunciar uma raiz com um ID de bridge mais alto, uma raiz mais ruim do que os nossos.



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.

O comportamento comum com respeito ao instituto de especificações elétricas e dos engenheiros eletrônicos (IEEE) é ignorar simplesmente todos os BPDU inferiores. O Backbone Fast usa-os porque assim que um for recebido, está absolutamente certo que uma falha ocorreu no trajeto à raiz e que você deve envelhecer para fora pelo menos uma porta.

Note: Uma falha indireta do link pode acontecer sem nenhuma geração do BPDU inferior na rede. Simplesmente adicione um hub ao diagrama anterior:



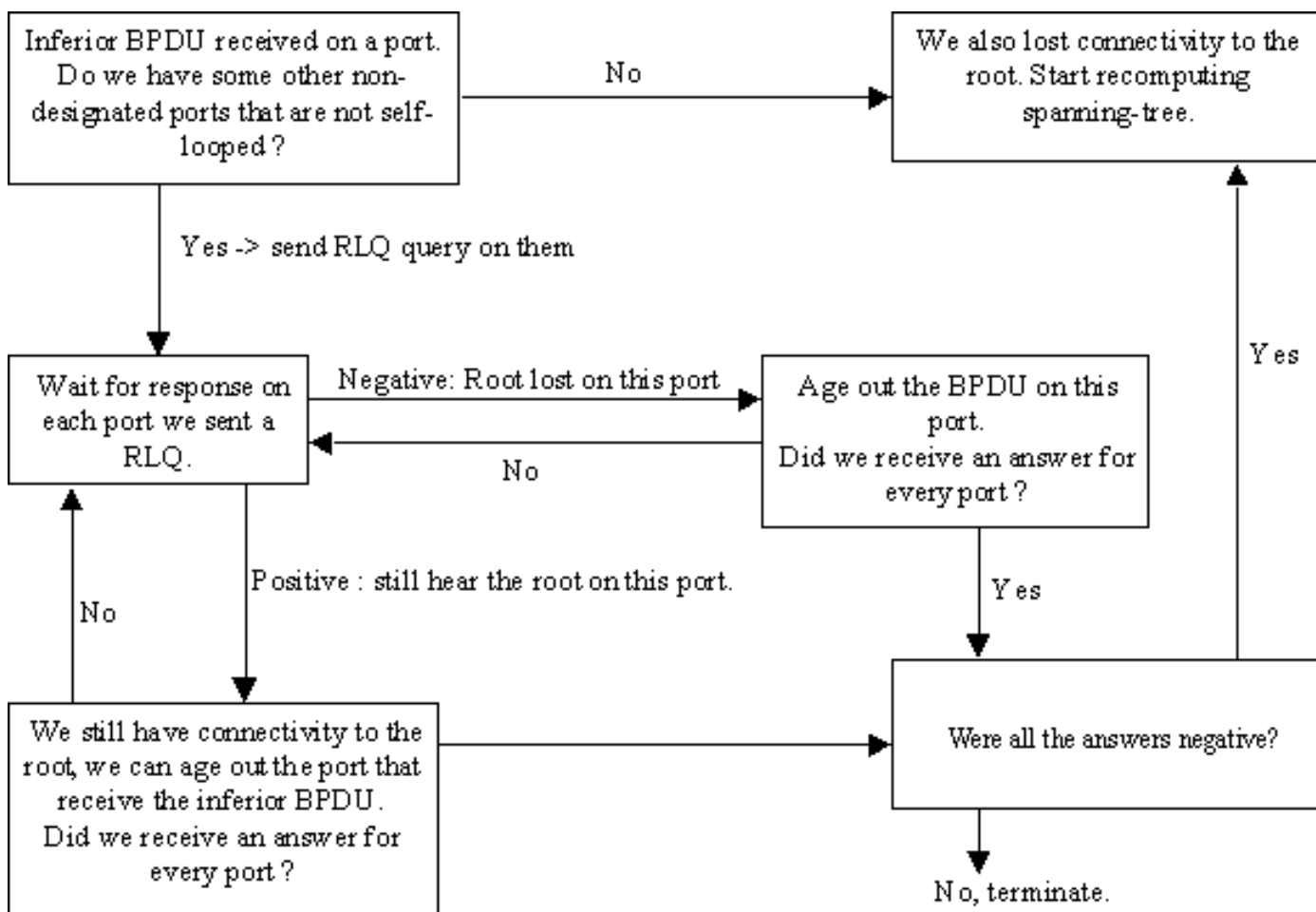
A falha do link ocorre entre o bridge-raiz R e o hub. B não detecta que o link vai para baixo e espera o max_age antes que reivindique ser a raiz nova. Recorde que o mecanismo trabalha somente se uma ponte detecta uma falha de link direto.

Mantenha o controle somente de BPDUs inferiores enviados pela ponte designada. Um vez que esse é o BPDU armazenado na porta. Se, por exemplo, uma ponte recentemente introduzida começa enviar o BPDU inferior, não começa a característica do Backbone Fast.

Reaja às falhas indireta do link

Quando um BPDU inferior é detectado em uma porta não designada, a segunda fase de Backbone Fast está provocada. Em vez do max_age passivamente de espera para envelhecer para fora as portas que podem ser afetadas pela falha, uma maneira proativa testá-las é

introduzida imediatamente por meio do RLQ PDU. O RQL é usado para atingir um tipo de ping da raiz em uma porta não designada e permitiu confirmar rapidamente se o BPDU armazenado em uma porta ainda é válido ou precisa ser descartado.



Ao receber um BPDU inferior de uma ponte designada, envie um PDU de RLQ para todas as portas não designadas, exceto para a porta onde você recebeu o BPDU inferior e as portas com auto-loop. Este é a fim certificar-se de você ainda se ouça da raiz em portas onde você é usado a receber BPDU. A porta onde você recebeu o BPDU inferior é excluída porque você está já ciente que sofreu de uma falha, loop automático e portas designadas não é útil, porque não conduzem à raiz.

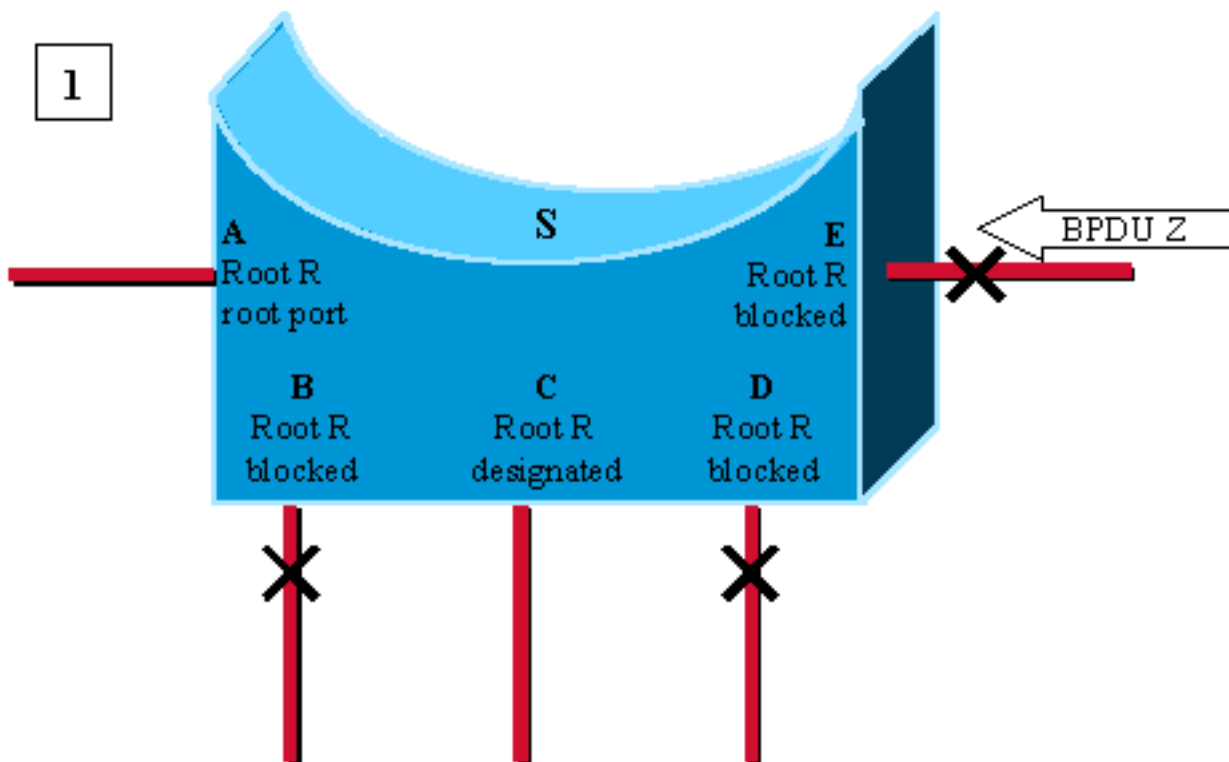
No recibo de uma resposta de RLQ em uma porta, se a resposta é negativa, a porta perdeu a conexão à raiz e você pode envelhecer para fora seu BPDU. Além disso, se todas portas não designadas restantes já receberam uma resposta negativa, a ponte inteira perde a raiz e pode começar o cálculo de STP a partir do zero.

Se a resposta confirma você pode ainda alcançar o bridge-raiz através desta porta, você pode imediatamente envelhecer para fora a porta em que nós recebemos inicialmente o BPDU inferior.

Neste exemplo, move A, B, D, e E é portas não designadas para o interruptor S. A é a porta de raiz e o outro está obstruindo. Quando E recebe um BPDU inferior (1), o backbone fast é ativado para acelerar o novo cálculo de STP.

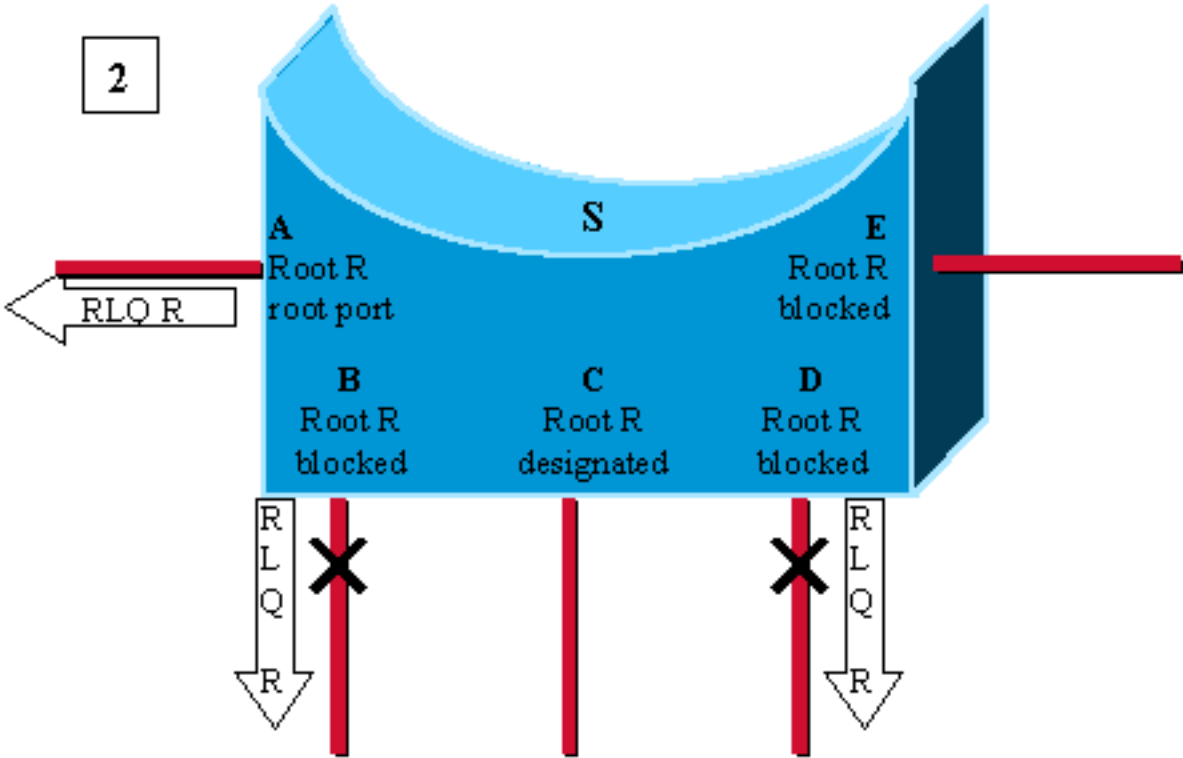
Mande um pedido RLQ, que procure a raiz R em todas as portas não designadas mas em E (2). As respostas especificam que raiz é acessível através destas portas. A resposta RLQ recebida por D especifica que D perdeu seu caminho para o R raiz. Envelheça seu BPDU imediatamente (3). As portas A e B recebem confirmação de que ainda têm um caminho para R (4). Assim, como

o interruptor S ainda tem a Conectividade à raiz, imediatamente a idade para fora move E e continua com regras de STP regulares (5).



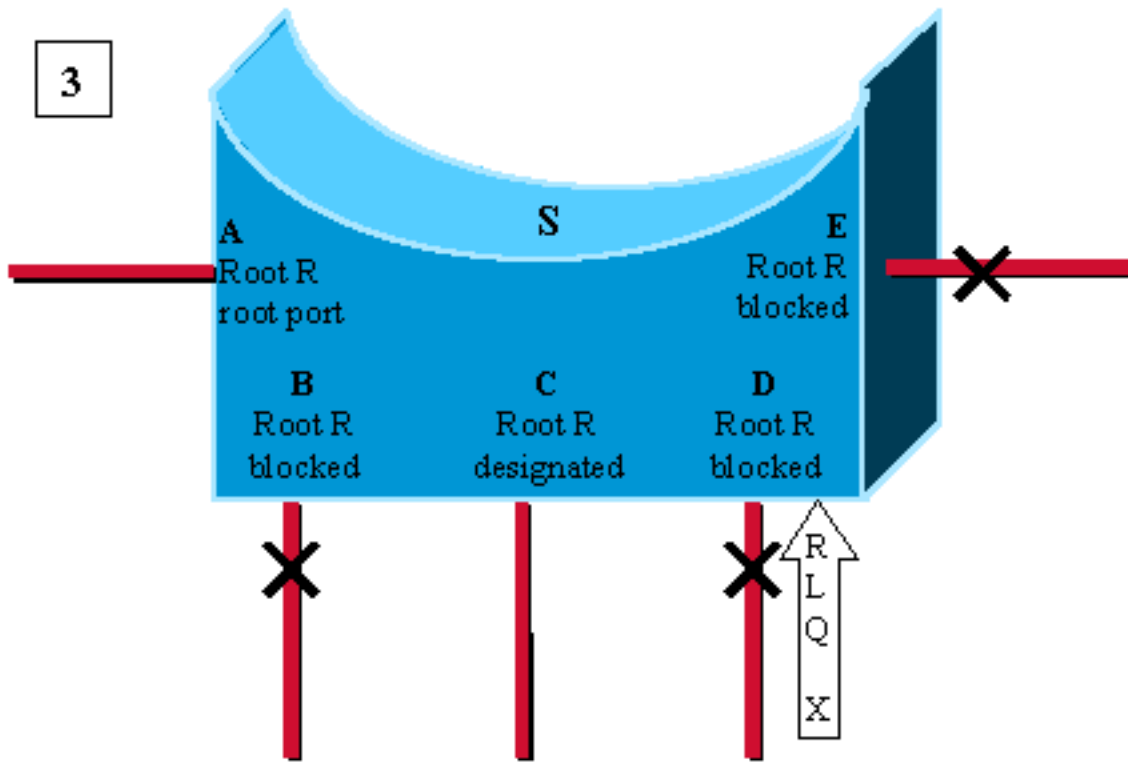
Port E receives an inferior BPDU, advertising root Z instead of root R stored on the different ports.

2



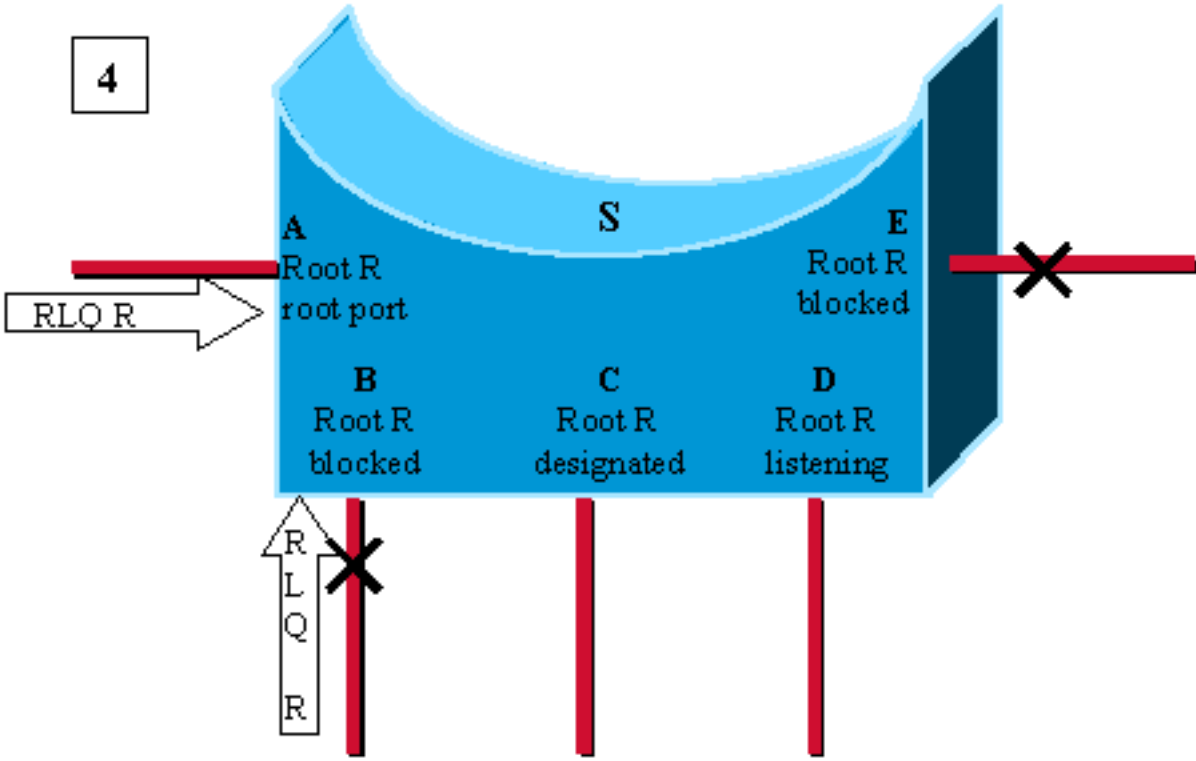
Switch S needs to recheck all its other non-designated ports. It sends out a RLQ request for root R on ports A,B and D.

3



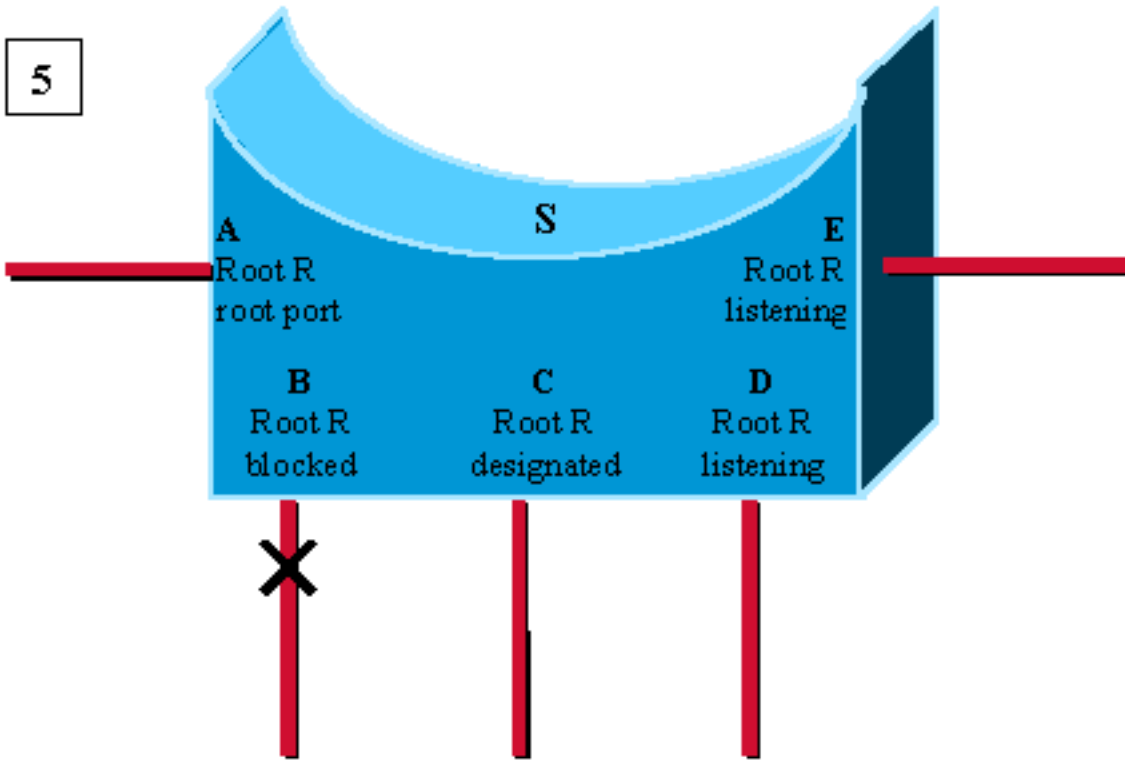
Port D is the first to receive and RLQ response from bridge X claiming to be the root. It is a negative response: D has lost connectivity to the root R. We age out immediately the BPDU on port D and go to listening. As we don't know if we still have connectivity to the root R, we don't age out port E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

Em um caso onde o interruptor receba somente respostas com uma raiz diferente de R, considere a raiz como o cálculo de STP a partir do zero perdido e reiniciado imediatamente. Note que este caso igualmente ocorre quando a única porta não-designado (e não loop automático) na ponte é a porta de raiz e você recebe um BPDU inferior nesta porta.

O PDU de Root Link Query

As duas formas de RLQs são solicitações de RLQ e respostas de RLQ.

O pedido RLQ é mandado em uma porta onde você receba geralmente BPDU, a fim certificar-se de você ainda tenha a Conectividade à raiz através desta porta. Especifique no pedido que a ponte é sua raiz e a resposta de RLQ volta eventualmente com um bridge-raiz que possa ser alcançado através desta porta. Se as duas raízes são as mesmas, a Conectividade está ainda viva, mais, ele está perdida.

Uma ponte que recebesse respostas de um pedido RLQ imediatamente se a conhece perdeu a conexão à raiz perguntada porque tem um bridge-raiz diferente a esse especificado na pergunta RLQ, e se é a raiz.

Se tal não for o caso, então, ele para a frente a pergunta para a raiz através de sua porta de raiz.

As respostas RLQ são inundadas nas portas designadas. O remetente da solicitação de RQL coloca seu ID de ponte no PDU. Isso serve para assegurar que, quando recebe de volta uma

resposta para sua própria consulta, ele não inunda a resposta em suas portas designadas.

A PDU RLQ possui a mesma estrutura de pacotes de uma BPDU STP normal. A única diferença é que dois endereços INSTANTÂNEOS específicos da Cisco diferentes estão usados: um para a solicitação e um para a resposta.

Isto o formato de BPDU padrão:

DA SA Duração DSAP SSAP CNTL SNAP PDU

O campo PDU é:

**Identificador de protocolo Versão Tipo de mensagem Flags ID da raiz Custo do caminho de raiz
ID do remetente ID da porta Idade da mensagem max age hello time retardo de encaminhamento**

O tipo de mensagem usado no PDU é igualmente diferente do BPDU padrão.

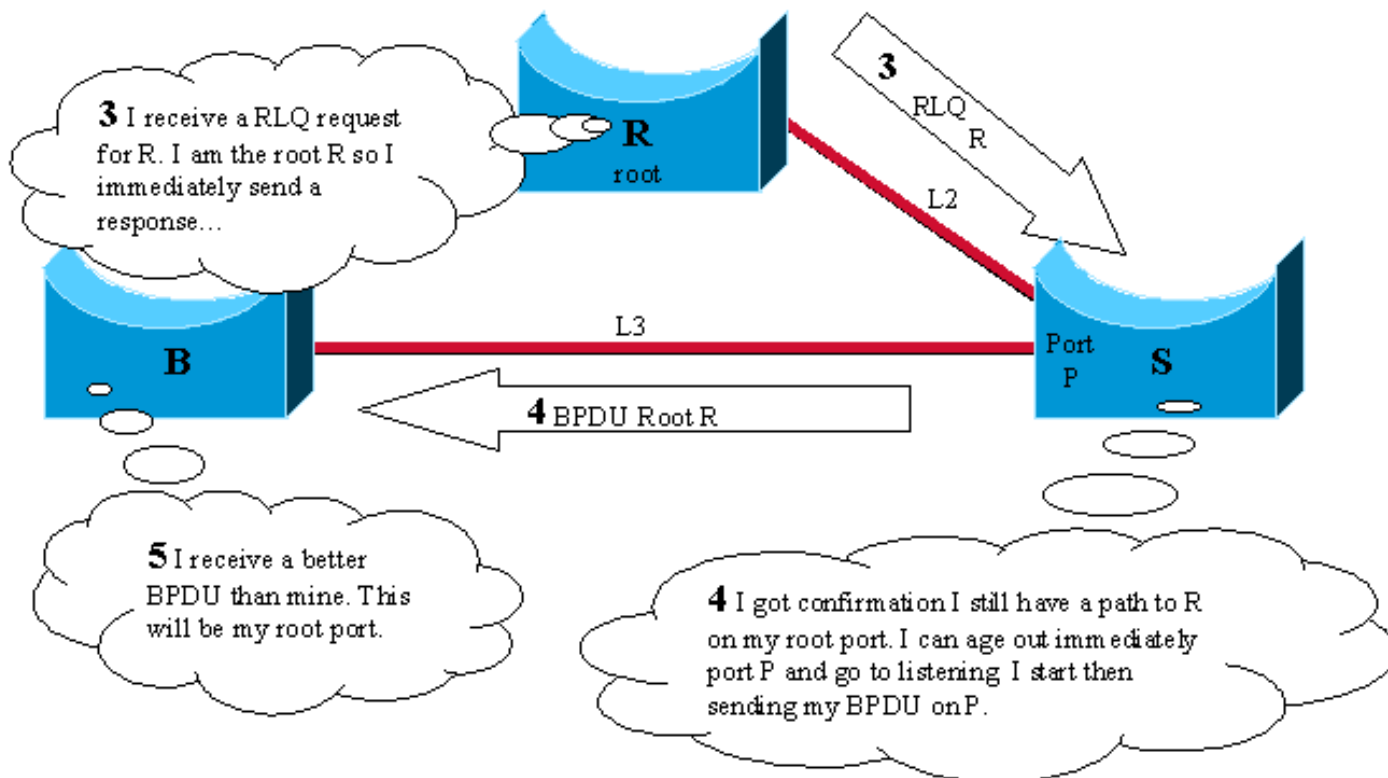
Os únicos campos usados são o ID raiz e o ID de Bridge de remetente.

Esta característica específico da Cisco precisa de ser configurada em todo o Switches na rede a fim processar estes PDU.

Cenário de exemplo com o recurso Backbone Fast ativado

Esta encenação é baseada no primeiro exemplo, mas, esta vez com o Backbone Fast permitido nos três Switches.

1. O primeiro estágio é exatamente o mesmo que o explicado anteriormente.
2. Assim que S receber o BPDU inferior de B, começa reconfirmar suas portas não designadas em vez do max_age de espera. Envia uma pergunta RLQ em sua porta de raiz para o bridge-raiz R.
3. O bridge-raiz R recebe a pergunta e responde-a imediatamente com uma resposta de RLQ que especifique lá seja ainda uma raiz R nesse sentido.
4. O S verificou agora todas as suas portas não designadas e ainda mantém conectividade com a raiz. Pode então envelhecer para fora imediatamente a informação armazenada em transições de P.P da porta à escuta e aos começos para enviar BPDU. Nessa fase, você já salvar segundos do max_age, e o algoritmo de Spanning Tree (STA) padrão aplica-se então.
5. B recebe o BPDU melhor de S (a melhor raiz R do que B) e considera agora as portas que conduzem ao L3 como sua porta de raiz.



Configurar o Backbone Fast para Cactos e Cisco IOS

Quando usado, o Backbone Fast deve ser permitido em todos os switches na rede porque o Backbone Fast exige o uso do mecanismo do pedido e da resposta RLQ a fim de informar os switches da estabilidade do caminho de raiz. O protocolo RLQ é ativo somente quando o Backbone Fast é permitido em um interruptor. Além disso, a rede pode igualmente ser executada em edições com inundação RLQ, se o Backbone Fast não é permitido em todos os switches. À revelia, o Backbone Fast é desabilitado.

O Backbone Fast não é apoiado em Catalyst 2900XL e 3500XL switches. Geralmente, você precisa de permitir o Backbone Fast se o domínio do interruptor contém estes switches além de outros Catalyst switches de apoio. Quando você executa o Backbone Fast nos ambientes com XL switch, sob topologias restritas, você pode permitir a característica onde o XL switch é o último interruptor na linha e é conectado somente ao núcleo em dois lugares. Não execute esta característica se a arquitetura dos XL switch está na forma da interligação de equipamentos em cascata.

Você não precisa de configurar o Backbone Fast com RSTP ou IEEE 802.1W porque o mecanismo é nativamente incluído e permitido automaticamente no RSTP. Para obter mais informações sobre o RSTP ou do IEEE 802.1W, refira-se a [medida - árvore do PVST+ ao exemplo de configuração da migração Rápido-PVST](#).

Configuração para CatOS

Para as Catalyst 4000, 5000, e 6000 series switch que executam Cactos, use estes comandos a fim de permitir globalmente o Backbone Fast para todas as portas e verificar a configuração.

```
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)
```

A fim indicar estatísticas do Backbone Fast:

```
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)
```

Configuração para o Cisco IOS

Para Catalyst Switches que executa com Cisco IOS Software, use estes comandos a fim permitir globalmente o Backbone Fast para todas as relações.

```
CAT-IOS# configure terminal
CAT-IOS(config)# spanning-tree backbonefast
CAT-IOS(config)# end
CAT-IOS#
```

A fim verificar que o Backbone Fast está permitido e mostrar estatísticas:

```
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#
```

Informações Relacionadas

- [Utilização de Portfast e outros comandos para reparar retardos de conectividade da inicialização de estação de trabalho](#)
- [Compreendendo e configurando os recursos uplinkfast de Cisco](#)
- [Aperfeiçoamentos do protocolo de extensão de árvore usando os recursos proteção de](#)

circuito e detecção de desvio BPDU

- Aprimoramento do Protetor de BPDU do PortFast de Spanning Tree
- Medida - árvore do PVST+ ao exemplo de configuração da migração Rápido-PVST
- Spanning Tree Protocol - Suporte por tecnologia
- Switches - Sustentação do produto
- Apoio de tecnologia de LAN switching
- Suporte Técnico e Documentação - Cisco Systems