

Entender e configurar o backbone rapidamente em switches Catalyst

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[BPDU's e como compará-los](#)

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

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

[Detectar Falhas de Link Indireto](#)

[Reagir a Falhas de Link Indireto](#)

[O PDU de Root Link Query](#)

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

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

[Configuração para CatOS](#)

[Configuração do Cisco IOS](#)

[Informações Relacionadas](#)

Introduction

Este documento 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®.

Prerequisites

Requirements

Não existem requisitos específicos para este documento.

Componentes Utilizados

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

- Switches Catalyst 2950 Series que executam o Cisco IOS Software Release 12.1(6)EA2 e posteriores
- Switches Catalyst 3550 Series que executam o Cisco IOS Software Release 12.1(4)EA1 e posterior

- Switches Catalyst 4000 Series que executam CatOS 5.1(1a) e posteriores
- Switches Catalyst 4500/4000 Series que executam o Cisco IOS Software Release 12.1(8a)EW e posterior
- Switches Catalyst 5500/5000 Series que executam CatOS Versão 4.1(1) e posterior
- Switches Catalyst 6500/6000 Series que executam CatOS Versão 5.1(1)CSX e posterior
- Switches Catalyst 6500/6000 Series que executam o Cisco IOS Software Release 12.0-7XE e posteriores

BPDUs e como compará-los

As BPDUs (Bridge Protocol Data Units, Unidades de Dados de Protocolo de Bridge) podem ser rigorosamente classificadas pelos campos que transportam. Entre esses campos estão o ID da bridge raiz, o custo do caminho até a raiz e o ID da bridge do remetente. Uma BPDU é considerada melhor que outra BPDU pelas seguintes razões:

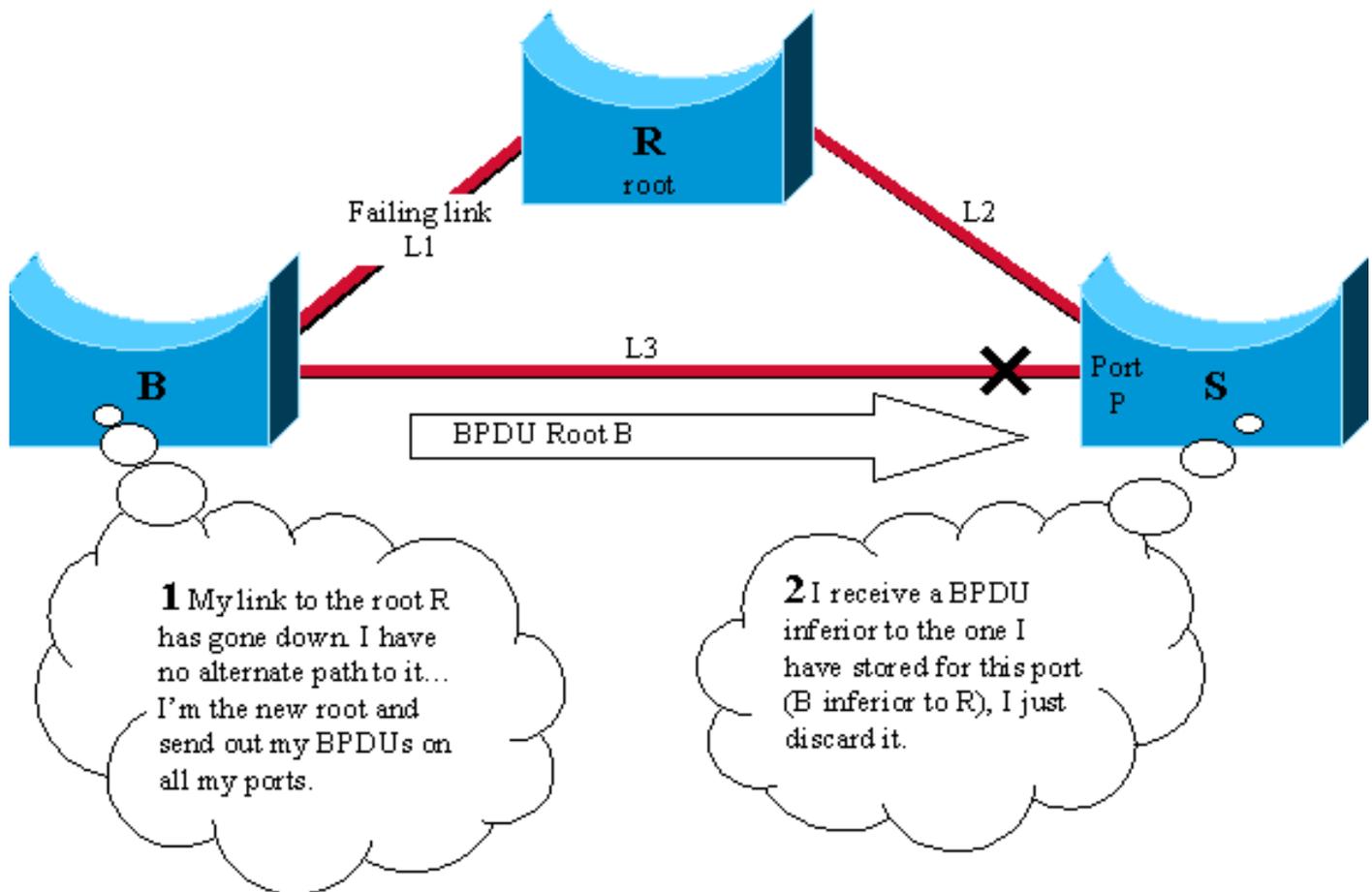
- Quando uma BPDU transporta um ID de bridge raiz melhor do que outro. Quanto menor o valor, melhor.
- Quando os valores de identificação do Root Bridge são iguais, então o BPDU com o custo de caminho inferior para a raiz é melhor.
- Quando os valores do ID da bridge raiz são iguais e os custos para a raiz são os mesmos, o BPDU com o melhor ID da bridge do remetente é melhor. Quanto menor o valor, melhor.

Há outras variáveis que podem atuar como um desempate. No entanto, quanto melhor uma BPDU, melhor será o acesso à melhor bridge raiz.

Uma bridge que recebe uma BPDU em uma porta melhor do que a que envia, coloca essa porta no modo de bloqueio, a menos que seja sua porta raiz. Isso significa que no segmento conectado a essa porta, existe outra ligação designada. Uma bridge armazena o valor da BPDU em uma porta enviada pela bridge designada atual.

Como o STP se recupera de uma falha indireta de enlace

Isso ilustra como o STP se comporta quando precisa recalculer após uma falha indireta de link, ou seja, quando uma bridge precisa alterar o status de algumas de suas portas devido a uma falha em um link que não está diretamente conectado a ela.



Considere este diagrama, que envolve três switches R, B e S em uma topologia totalmente em malha. Suponha que R é a bridge raiz e B é a bridge raiz de backup. S bloqueia sua porta P e B é a bridge designada para o link L3.

1. Se o link L1 cair, o switch B detectará imediatamente a falha e assumirá que é a raiz. Ele começa a enviar BPDUs para S e alega ser a nova raiz.
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. Após o temporizador `max_age` expirar (20 segundos por padrão), a BPDU armazenada em S para a porta P expira. A porta vai imediatamente para a escuta e S começa a enviar sua melhor BPDU para B.
4. Assim que B recebe a BPDU de S, ele pára de enviar sua 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 completa é restaurada.

Foi necessário o valor `max_age` (20 segundos) mais o dobro do valor `fw_delay` (2x15 segundos) para se recuperar dessa falha indireta de link. Corresponde a 50 segundos com os parâmetros padrão. O recurso `backbone fast` propõe salvar `max_age` (20 segundos). Para fazer isso, ele expira imediatamente após a porta receber BPDUs inferiores.

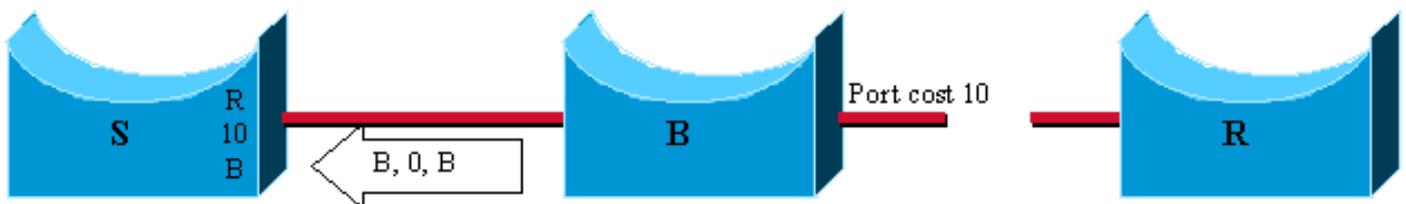
Melhorias de Backbone Fast no STP padrão

Com o exemplo anterior, o STP invalida informações que se tornam erradas devido a uma falha indireta de link. Para fazer isso, ele espera passivamente por `max_age`. Para se livrar desse atraso `max_age`, o `backbone fast` introduz dois aprimoramentos:

- A habilidade de detectar uma falha indireta de link, o mais breve possível. Isso é obtido por meio do rastreamento das BPDUs inferiores que uma bridge designada envia quando experimenta uma falha direta de link.
- Um mecanismo que permite verificar imediatamente se as informações de BPDUs armazenadas em uma porta ainda são válidas. Isso é implementado com uma nova unidade de dados de protocolo (PDU) e a consulta de enlace de raiz, chamada neste documento de PDU de RLQ.

Detectar Falhas de Link Indireto

Se um BPDUs inferior for recebido em uma porta de nossa bridge designada, então essa bridge perdeu a raiz e começa a anunciar uma raiz com um ID de bridge mais alto, uma raiz pior que a nossa.



In this case, B lost the root and sends a BPDUs 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 habitual em relação às especificações do Institute of Electrical and Electronics Engineers (IEEE) é simplesmente ignorar qualquer BPDUs inferior. O Backbone Fast os usa porque assim que um é recebido, é certo que ocorreu uma falha no caminho para a raiz e que você deve envelhecer em pelo menos uma porta.

Note: Uma falha indireta de link pode ocorrer sem qualquer geração de BPDUs inferior na rede. Simplesmente adicione um hub ao diagrama anterior:



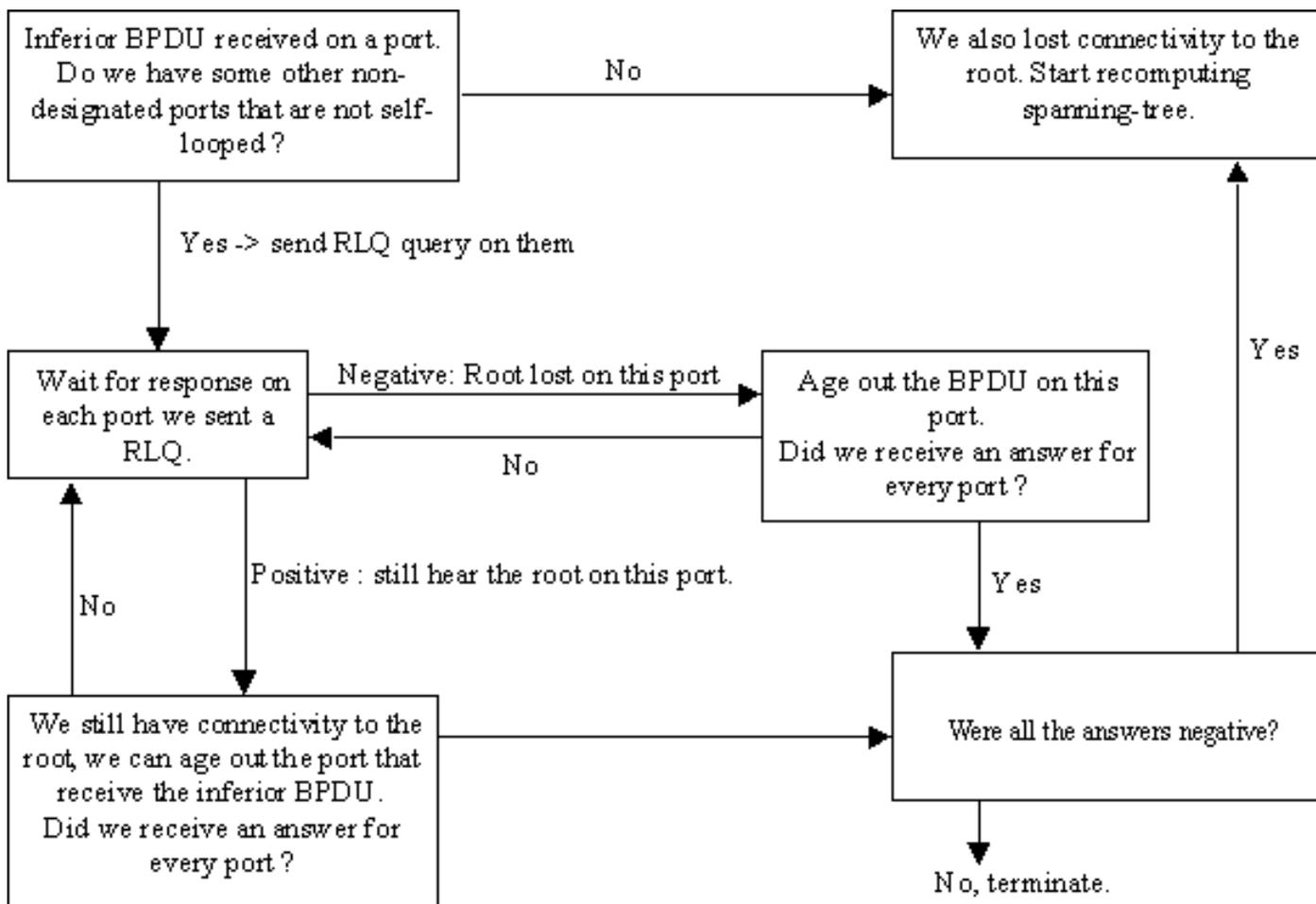
A falha de link ocorre entre a bridge raiz R e o hub. B não detecta que o link fica inativo e aguarda max_age antes que ele afirme ser a nova raiz. Lembre-se de que o mecanismo só funciona se uma bridge detectar uma falha de link direto.

Mantenha o controle somente de BPDUs inferiores enviados pela ponte designada. Um vez que esse é o BPDUs armazenado na porta. Se, por exemplo, uma bridge recém-inserida começa a enviar BPDUs inferior, ela não inicia o recurso backbone fast.

Reagir a Falhas de Link Indireto

Quando uma BPDUs inferior é detectada em uma porta não designada, a segunda fase do backbone fast é acionada. Em vez de esperar passivamente max_age para encerrar as portas

que podem ser afetadas pela falha, uma maneira proativa de testá-las imediatamente é apresentada por meio da PDU do RLQ. 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. Isso é feito para verificar se você ainda ouve da raiz nas portas em que está usado para receber BPDUs. A porta em que você recebeu o BPDU inferior é excluída porque você já sabe que ele sofreu de uma falha, as portas autocarregadas e designadas não são úteis, pois não levam à raiz.

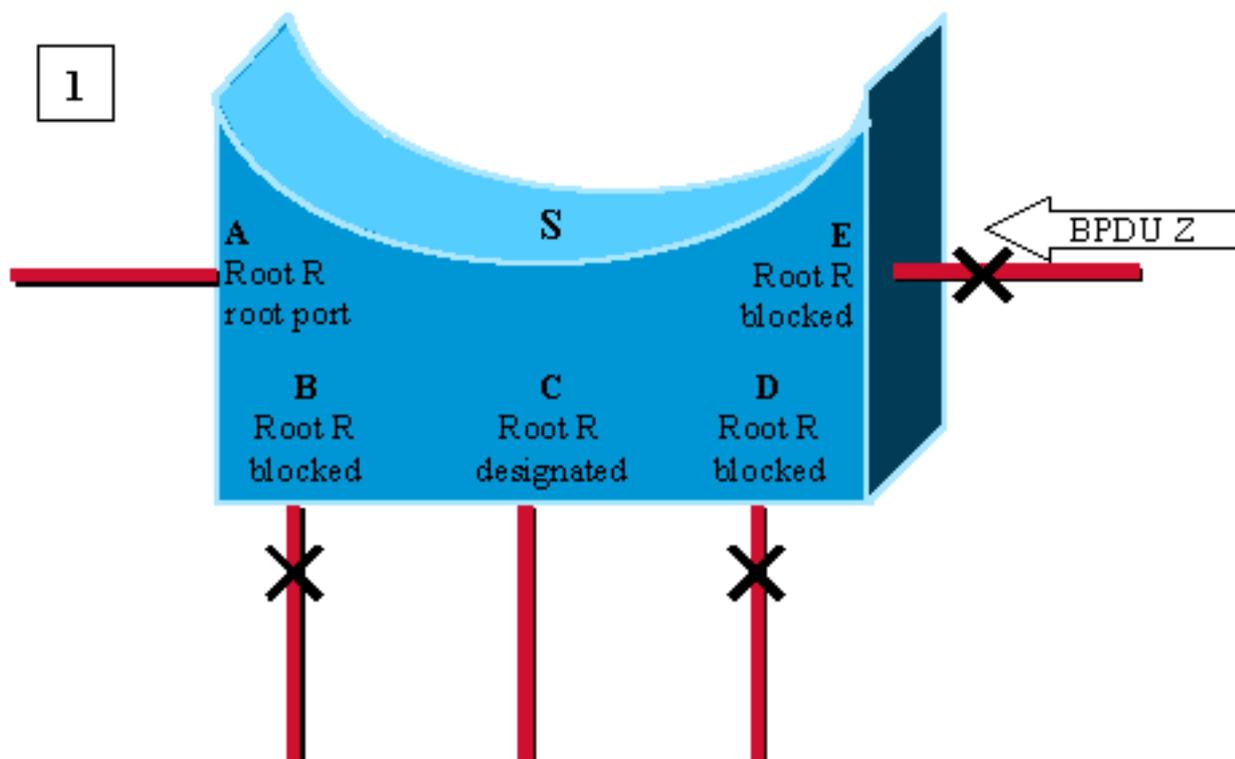
Ao receber uma resposta de RLQ em uma porta, se a resposta for negativa, a porta perdeu a conexão com a raiz e você pode encerrar sua BPDU. Além disso, se todas as outras portas não designadas já receberam uma resposta negativa, a bridge inteira perde a raiz e pode iniciar o cálculo do STP do zero.

Se a resposta confirmar que você ainda pode acessar a bridge raiz por meio dessa porta, você pode encerrar imediatamente a porta na qual recebemos inicialmente o BPDU inferior.

Neste exemplo, as portas A, B, D e E são portas não designadas para o switch S. A é a porta raiz e as outras estão bloqueando. Quando E recebe um BPDU inferior (1), o backbone fast é ativado para acelerar o novo cálculo de STP.

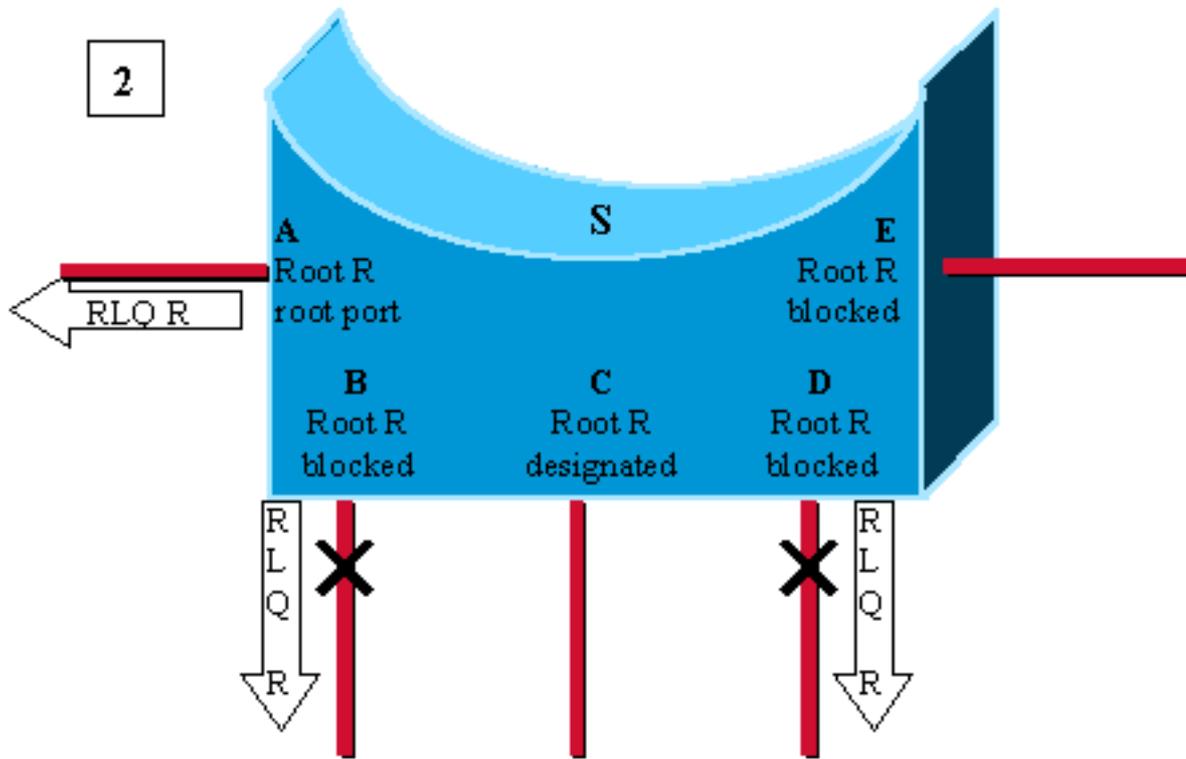
Envie uma solicitação RLQ, que procura a raiz R em todas as portas não designadas, exceto E (2). As respostas especificam qual raiz pode ser acessada por meio dessas portas. A resposta RLQ que D recebe especifica que D perdeu seu caminho para a raiz R. Encerre sua BPDU

imediatamente (3). As portas A e B recebem confirmação de que ainda têm um caminho para R (4). Assim, enquanto o Switch S ainda tiver conectividade com a raiz, atinja a porta E e continue com as regras de STP regulares (5).



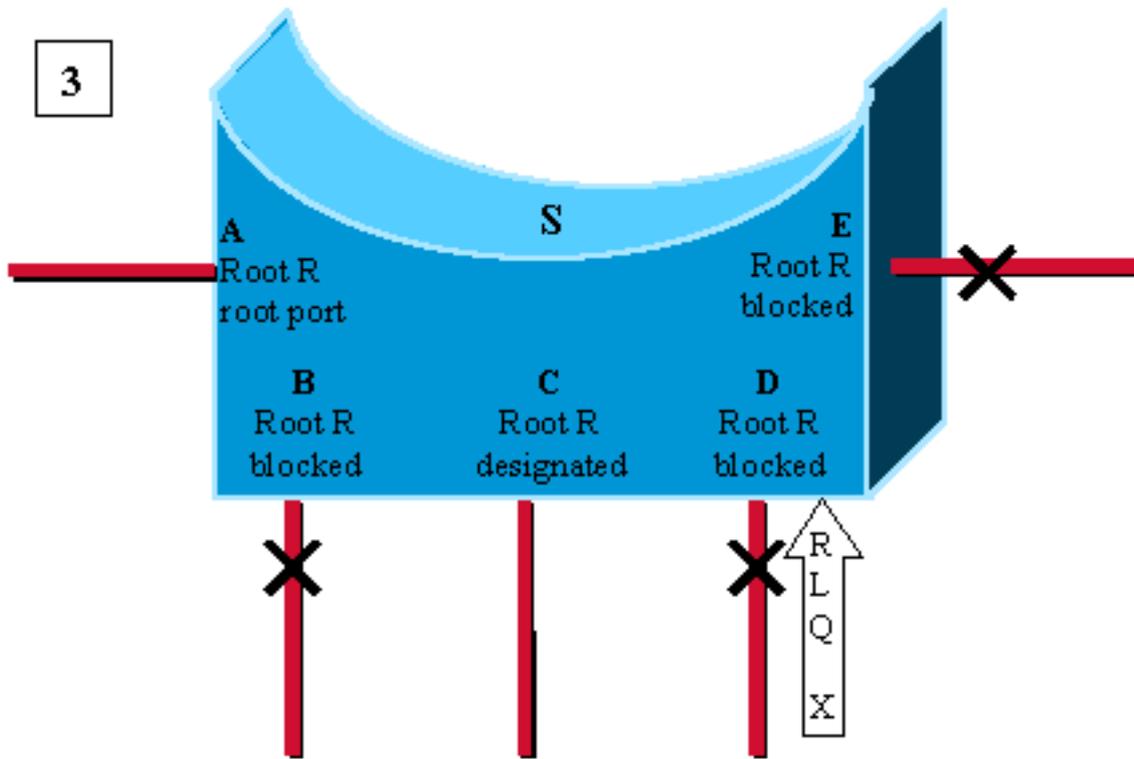
Port E receives an inferior BPDU, advertising root Z instead of root R stored on the different ports.

2



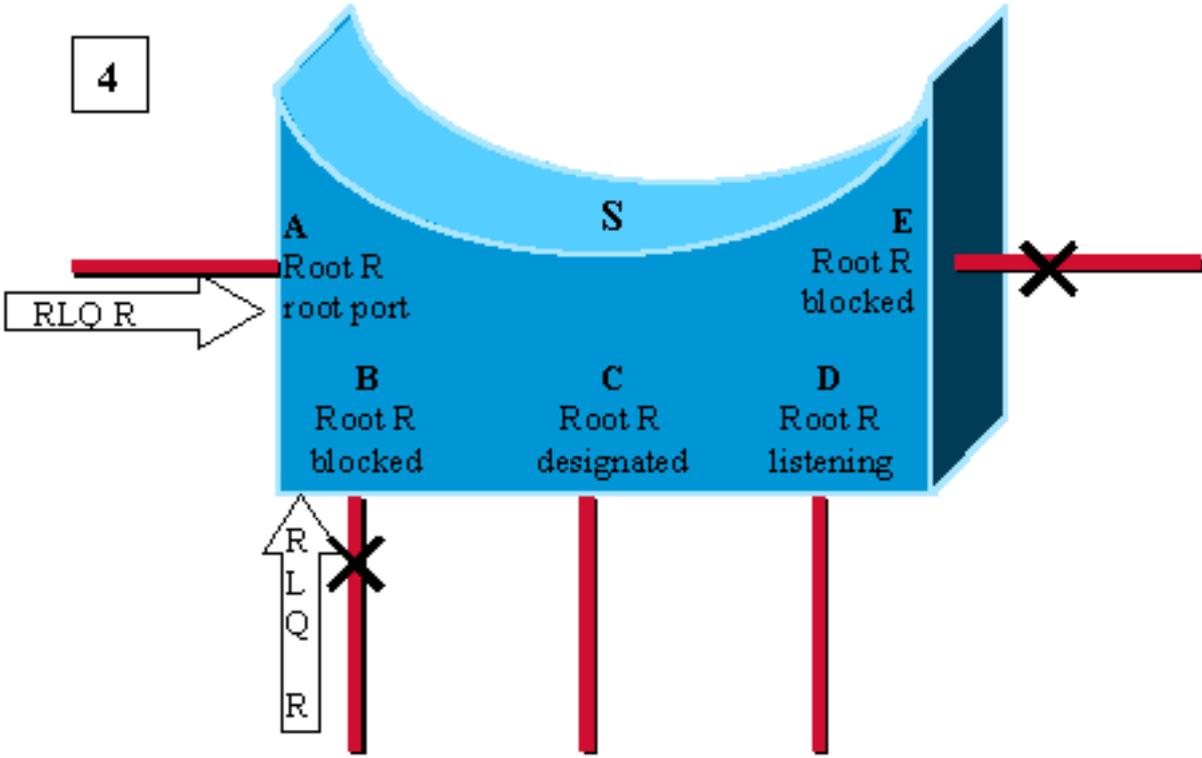
Switch S needs to recheck all its other non-designated ports. It sends out a RLQ request for root R on ports A,B and D.

3



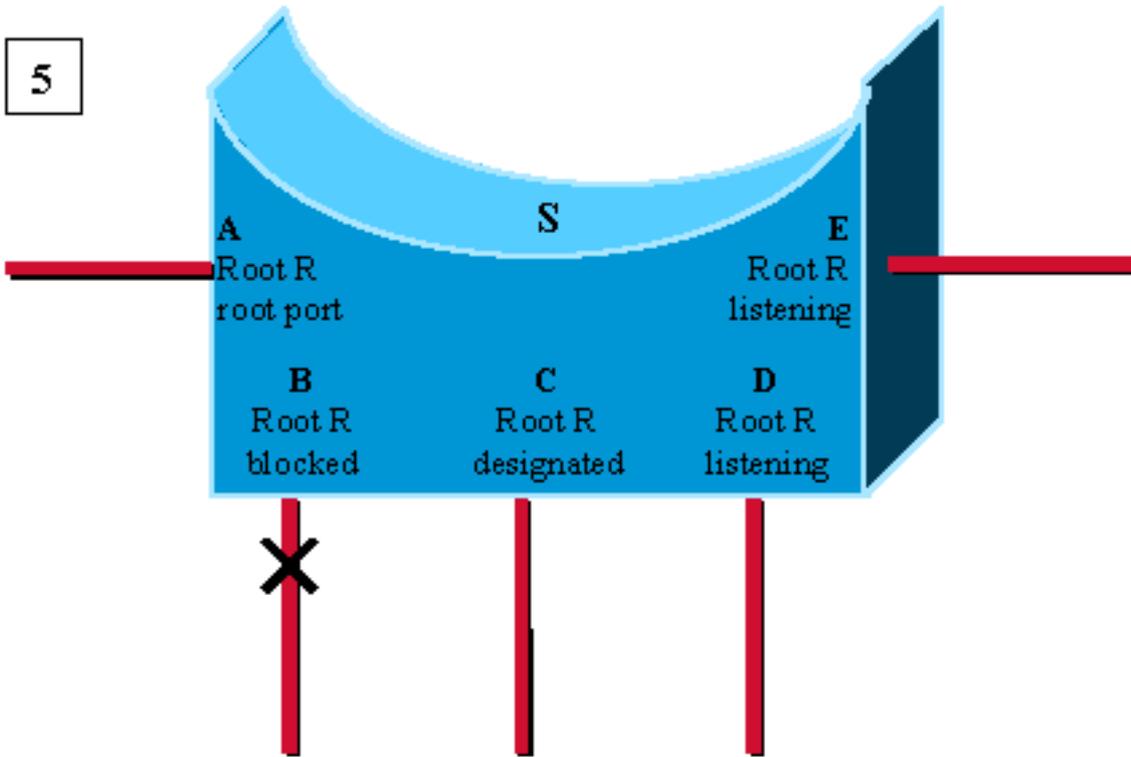
Port D is the first to receive and RLQ response from bridge X claiming to be the root. It is a negative response: D has lost connectivity to the root R. We age out immediately the BPDU on port D and go to listening. As we don't know if we still have connectivity to the root R, we don't age out port E 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 em que o switch recebeu apenas respostas com uma raiz diferente de R, considere a raiz como o cálculo de STP perdido e reiniciado imediatamente. Observe que esse caso também ocorre quando a única porta não designada (e não com autoloop) na bridge é a porta raiz e você recebe um BPDU inferior nessa porta.

O PDU de Root Link Query

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

A solicitação RLQ é enviada em uma porta onde você geralmente recebe BPDUs, para verificar se você ainda tem conectividade com a raiz através dessa porta. Especifique na solicitação qual bridge é sua raiz e a resposta RLQ eventualmente volta com uma bridge raiz que pode ser acessada por meio dessa porta. Se as duas raízes forem iguais, a conectividade ainda estará viva, caso contrário, ela será perdida.

Uma bridge que recebe uma solicitação RLQ atende imediatamente se sabe que perdeu a conexão com a raiz consultada porque tem uma bridge raiz diferente da especificada na consulta RLQ e se é a raiz.

Se esse não for o caso, ele encaminha a consulta para a raiz através de sua porta 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 SNAP específicos da Cisco são usados: um para a solicitação e um para a resposta.

Este é o formato padrão de BPDU:

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 usada na PDU também é diferente da BPDU padrão.

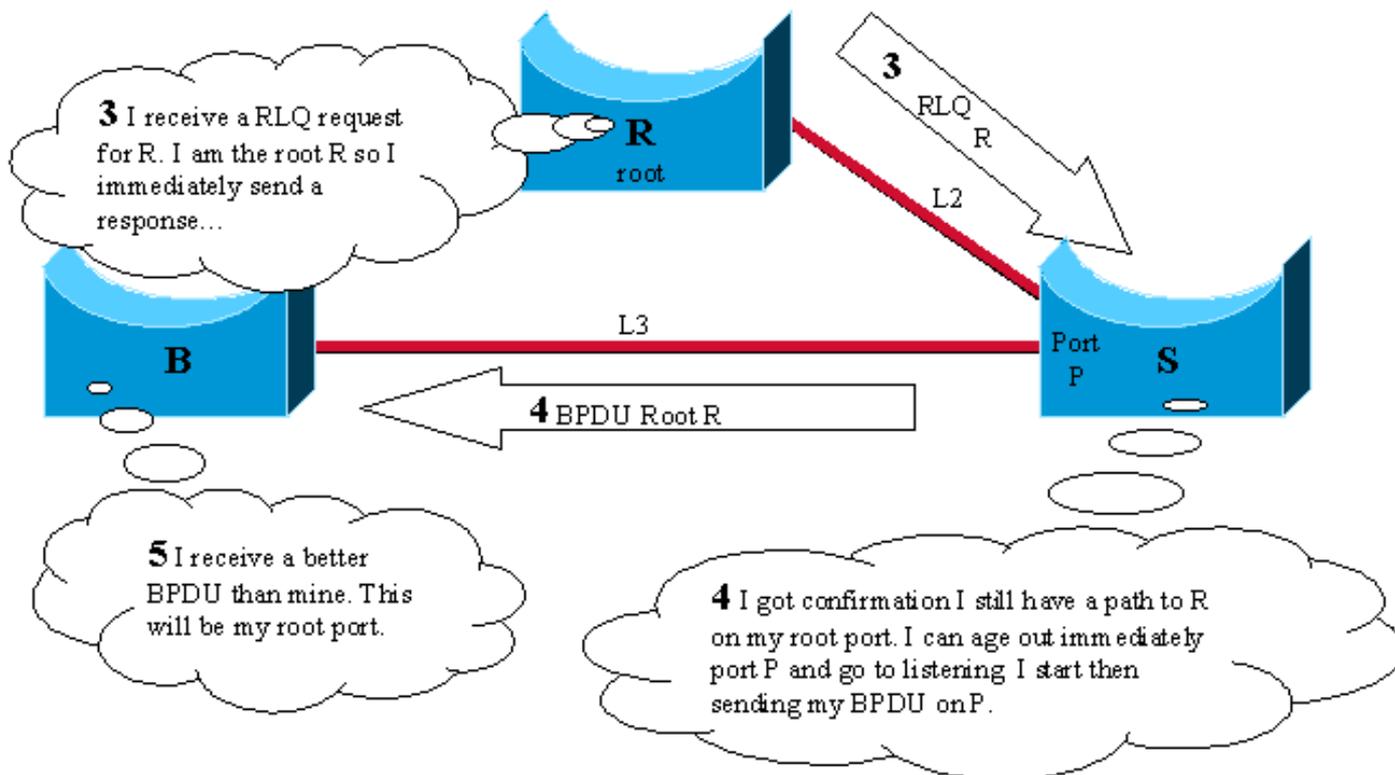
Os únicos campos usados são o ID da raiz e o ID da bridge do remetente.

Esse recurso específico da Cisco precisa ser configurado em todos os switches na rede para processar essas PDUs.

Cenário de exemplo com o recurso Backbone Fast ativado

Esse cenário é baseado no primeiro exemplo, mas, desta vez com o backbone fast habilitado nos três switches.

1. O primeiro estágio é exatamente o mesmo que o explicado anteriormente.
2. Assim que S recebe o BPDU inferior de B, ele começa a reconfirmar suas portas não designadas em vez de esperar max_age. Envia uma consulta RLQ na sua porta raiz para a bridge raiz R.
3. A bridge raiz R recebe a consulta e responde imediatamente com uma resposta RLQ que especifica que ainda há um R raiz nessa direção.
4. O S verificou agora todas as suas portas não designadas e ainda mantém conectividade com a raiz. Em seguida, ele pode envelhecer imediatamente as informações armazenadas na porta P. P para escuta e começa a enviar BPDUs. Nesse estágio, você já salvou max_age segundos, e o STA (Spanning-Tree Algorithm) padrão se aplica então.
5. B recebe o melhor BPDU de S (R melhor raiz que B) e considera agora as portas que levam a L3 como sua porta raiz.



Configurar Backbone Fast para CatOS e Cisco IOS

Quando usado, o backbone fast deve ser ativado em todos os switches da rede porque o backbone fast exige o uso do mecanismo de solicitação e resposta de RLQ para informar aos switches a estabilidade do caminho raiz. O protocolo RLQ está ativo somente quando o backbone fast está ativado em um switch. Além disso, a rede também pode ter problemas com a inundação de RLQ, se o fast de backbone não estiver habilitado em todos os switches. Por padrão, o backbone fast está desabilitado.

O Backbone Fast não é suportado nos switches Catalyst 2900XL e 3500XL. Em geral, você precisa ativar o backbone rapidamente se o domínio do switch contiver esses switches além de outros switches Catalyst compatíveis. Quando você implementa o backbone fast em ambientes com switches XL, sob topologias restritas, você pode ativar o recurso onde o switch XL é o último switch em linha e está conectado apenas ao núcleo em dois lugares. Não implemente esse recurso se a arquitetura dos switches XL estiver em cadeia de margarida.

Você não precisa configurar o backbone rapidamente com RSTP ou IEEE 802.1w porque o mecanismo é incluído nativamente e ativado automaticamente no RSTP. Para obter mais informações sobre RSTP ou IEEE 802.1w, consulte [Exemplo de Configuração de Spanning Tree do PVST+ para o Rapid-PVST Migration](#).

Configuração para CatOS

Para os switches das séries Catalyst 4000, 5000 e 6000 que executam CatOS, use esses comandos para ativar o backbone fast globalmente 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)
```

Para exibir estatísticas rápidas de backbone:

```
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 do Cisco IOS

Para os switches Catalyst executados com o software Cisco IOS, use esses comandos para ativar o backbone fast globalmente para todas as interfaces.

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

Para verificar se o backbone fast está ativado 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](#)
- [Entendendo e configurando o recurso Cisco UplinkFast](#)
- [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](#)
- [Exemplo de configuração para migrar o spanning tree de PVST+ para PVST rápido](#)
- [Protocolo Spanning Tree - Suporte a Tecnologia](#)
- [Switches - Suporte a produtos](#)
- [LAN Switching - Suporte a tecnologia](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)