

# Verständnis und Konfiguration von Backbone Fast auf Catalyst Switches

## Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[BPDU's und Vergleich](#)

[Wie STP nach einem Indirect Link Failure \(Indirekter Verbindungsausfall\) wiederherstellt](#)

[Fast Enhancements im Backbone auf Standard-STP](#)

[Erkennung von Indirect Link Failure](#)

[Reaktion auf Fehler bei indirekten Verbindungen](#)

[PDU für die Root-Link-Abfrage](#)

[Beispielszenario mit aktivierter Backbone Fast-Funktion](#)

[Konfigurieren von Backbone Fast für CatOS und Cisco IOS](#)

[Konfiguration für CatOS](#)

[Konfiguration für Cisco IOS](#)

[Zugehörige Informationen](#)

## Einführung

In diesem Dokument wird beschrieben, wie Backbone schnell konfiguriert werden kann. Backbone Fast ist eine proprietäre Funktion von Cisco, mit der ein Switch nach Aktivierung auf allen Switches eines Bridge-Netzwerks bis zu 20 Sekunden (max\_age) sparen kann, wenn er nach einem indirekten Verbindungsausfall wiederhergestellt wird. Nach einer kurzen Übersicht über einige STP-Grundlagen (Spanning Tree Protocol) können Sie das genaue Fehlerszenario sehen, auf das sich das Backbone schnell bezieht und wie es für Catalyst Switches konfiguriert wird, die sowohl CatOS- als auch Cisco IOS<sup>®</sup> Software ausführen.

## Voraussetzungen

### Anforderungen

Für dieses Dokument bestehen keine speziellen Anforderungen.

### Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Catalyst Switches der Serie 2950 mit Cisco IOS Software Release 12.1(6)EA2 und höher
- Catalyst Switches der Serie 3550 mit Cisco IOS Software Release 12.1(4)EA1 und höher

- Catalyst Switches der Serie 4000 mit CatOS 5.1(1a) und höher
- Catalyst Switches der Serien 4500/4000 mit Cisco IOS Software Release 12.1(8a)EW und höher
- Catalyst Switches der Serien 5500/5000 mit CatOS Version 4.1(1) und höher
- Catalyst Switches der Serien 6500/6000 mit CatOS Version 5.1(1)CSX und höher
- Catalyst Switches der Serien 6500/6000 mit Cisco IOS Software Release 12.0-7XE und höher

## **BPDU und Vergleich**

Bridge Protocol Data Units (BPDUs) können durch die von ihnen transportierten Felder streng klassifiziert werden. Zu diesen Feldern gehören die Root-Bridge-ID, die Pfadkosten zum Root und die Sender-Bridge-ID. Eine BPDU wird aus den folgenden Gründen als besser angesehen als eine andere BPDU:

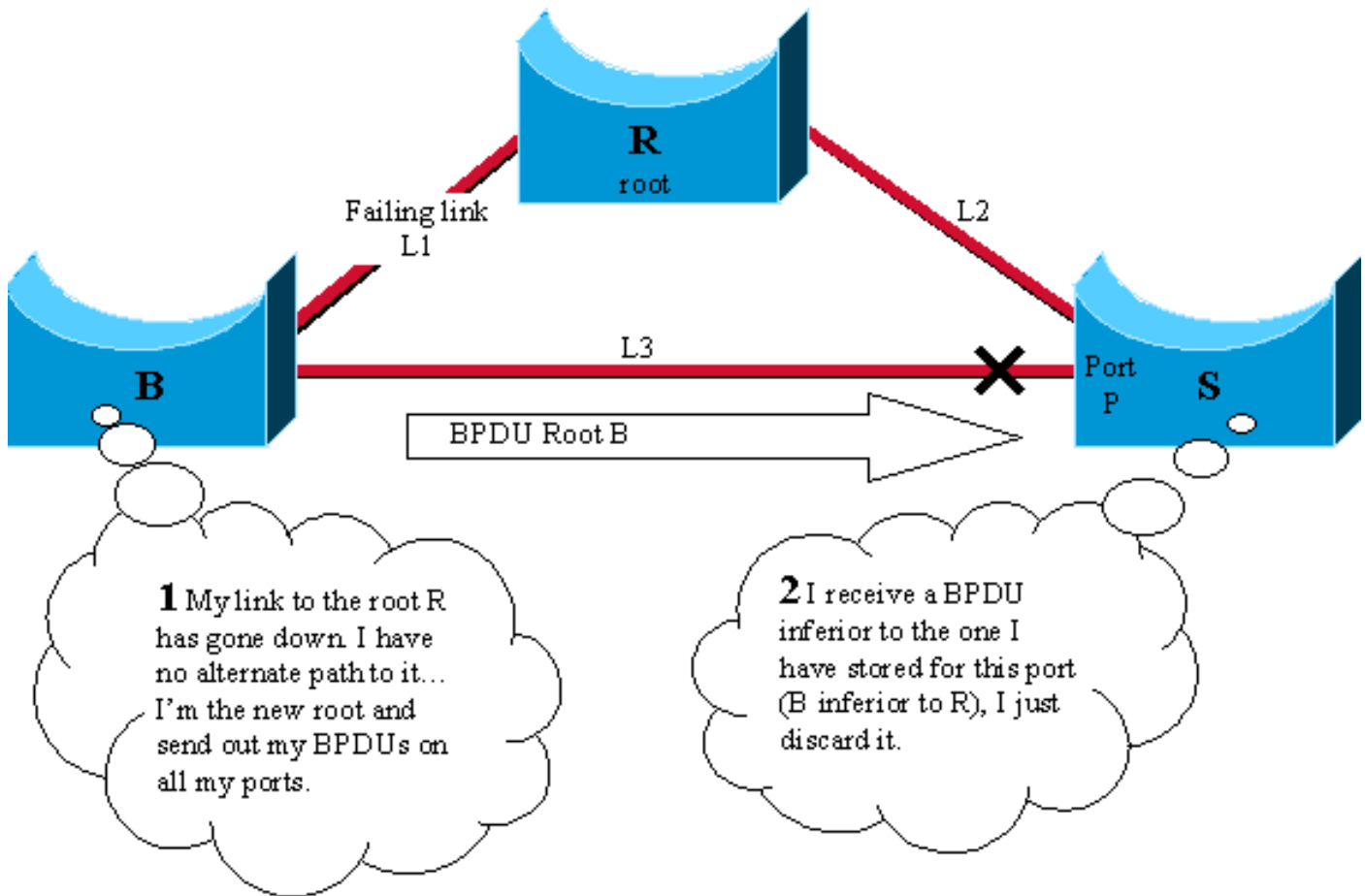
- Wenn eine BPDU über eine bessere Root Bridge-ID verfügt als eine andere. Je niedriger der Wert, desto besser.
- Wenn die Root-Bridge-ID-Werte gleich sind, ist die BPDU mit den niedrigsten Pfadkosten zum Root besser.
- Wenn die Root-Bridge-ID-Werte gleich sind und die Kosten für das Root gleich sind, ist die BPDU mit der besseren Sender-Bridge-ID besser. Je niedriger der Wert, desto besser.

Es gibt weitere Variablen, die dann als Bindeglied fungieren können. Je besser eine BPDU, desto besser ist der Zugriff auf die beste Root Bridge.

Eine Bridge, die eine BPDU auf einem Port empfängt, der besser ist als die, die sie sendet, setzt diesen Port in den Blockierungsmodus, es sei denn, es handelt sich um den Root-Port. Das bedeutet, dass auf dem mit diesem Port verbundenen Segment eine andere Bridge als designierte Bridge vorhanden ist. Eine Bridge speichert den BPDU-Wert auf einem Port, der von der aktuell designierten Bridge gesendet wird.

## **Wie STP nach einem Indirect Link Failure (Indirekter Verbindungsausfall) wiederherstellt**

Dies veranschaulicht das Verhalten von STP, wenn es nach einem indirekten Verbindungsausfall neu berechnet werden muss, d. h. wenn eine Bridge den Status einiger ihrer Ports ändern muss, weil eine Verbindung ausfällt, die nicht direkt mit ihr verbunden ist.



Betrachten Sie dieses Diagramm, das drei Switches R, B und S in einer vollständig vernetzten Topologie umfasst. Angenommen, R ist die Root Bridge und B die Backup Root Bridge. S blockiert den Port P und B ist die designierte Bridge für die Verbindung L3.

1. Wenn die Verbindung L1 ausfällt, erkennt Switch B den Ausfall sofort und geht davon aus, dass es sich um den Root handelt. Es beginnt, BPDUs an S zu senden und behauptet, der neue Root zu sein.
2. Wenn S diese neue BPDU von B empfängt, stellt es fest, dass sie niedriger ist als die, die sie für Port P gespeichert hat, und ignoriert sie.
3. Nachdem der Timer `max_age` abgelaufen ist (standardmäßig 20 Sekunden), wird die auf S gespeicherte BPDU für Port P deaktiviert. Der Port geht sofort zum Zuhören und S beginnt, seine bessere BPDU an B zu senden.
4. Sobald B die BPDU von S empfängt, beendet es das Senden seiner BPDU.
5. Port P wechselt durch Zuhören und Lernen in den Weiterleitungsstatus. Dies erfordert das Doppelte des `fw_delay`-Werts, also weitere 30 Sekunden. Anschließend wird die vollständige Verbindung wiederhergestellt.

Es dauerte 20 Sekunden, bis der Wert für `max_age` (`max_age`) plus das Doppelte des Werts `fw_delay` ( $2 \times 15$  Sekunden) für die Wiederherstellung nach diesem indirekten Verbindungsausfall ermittelt wurde. Dies sind 50 Sekunden mit den Standardparametern. Die Backbone Fast-Funktion schlägt vor, `max_age` (20 Sekunden) zu speichern. Dazu werden unmittelbar nach dem Empfang unterlegener BPDUs durch den Port Warnmeldungen ausgegeben.

## Fast Enhancements im Backbone auf Standard-STP

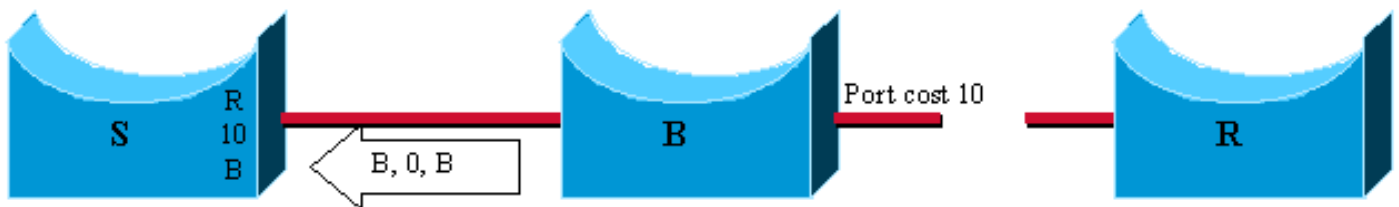
Im vorherigen Beispiel werden Informationen, die aufgrund eines indirekten Verbindungsausfalls

falsch sind, durch STP ungültig gemacht. Dazu wartet sie passiv auf max\_age. Um diese Verzögerung bei max\_age zu vermeiden, bietet Backbone schnell zwei Verbesserungen:

- Die Möglichkeit, einen indirekten Verbindungsausfall so schnell wie möglich zu erkennen. Dies wird durch die Verfolgung der untergeordneten BPDUs erreicht, die eine designierte Bridge sendet, wenn eine direkte Verbindung ausfällt.
- Ein Mechanismus, der eine sofortige Überprüfung der Gültigkeit der auf einem Port gespeicherten BDU-Informationen ermöglicht. Diese wird mit einer neuen Protokolldateneinheit (PDU) und der Root Link Query (Root-Verbindungsabfrage) implementiert, die in diesem Dokument als RLQ PDU bezeichnet wird.

## Erkennung von Indirect Link Failure

Wenn eine unterlegene BDU von unserer designierten Bridge auf einem Port empfangen wird, hat diese Bridge den Root verloren und beginnt, einen Root mit einer höheren Bridge-ID anzukündigen, einer schlechteren Root als unsere.



In this case, B lost the root and sends a BDU 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.

Das übliche Verhalten in Bezug auf die Spezifikationen des Institute of Electrical and Electronics Engineers (IEEE) besteht darin, unterlegene BPDUs einfach zu ignorieren. Backbone verwendet sie schnell, da es sicher ist, dass ein Fehler auf dem Pfad zum Root aufgetreten ist, sobald ein solcher empfangen wird, und dass mindestens ein Port verworfen werden muss.

**Hinweis:** Ein indirekter Verbindungsausfall kann ohne untergeordnete BDU-Generierung im Netzwerk auftreten. Fügen Sie einfach im vorherigen Diagramm einen Hub hinzu:

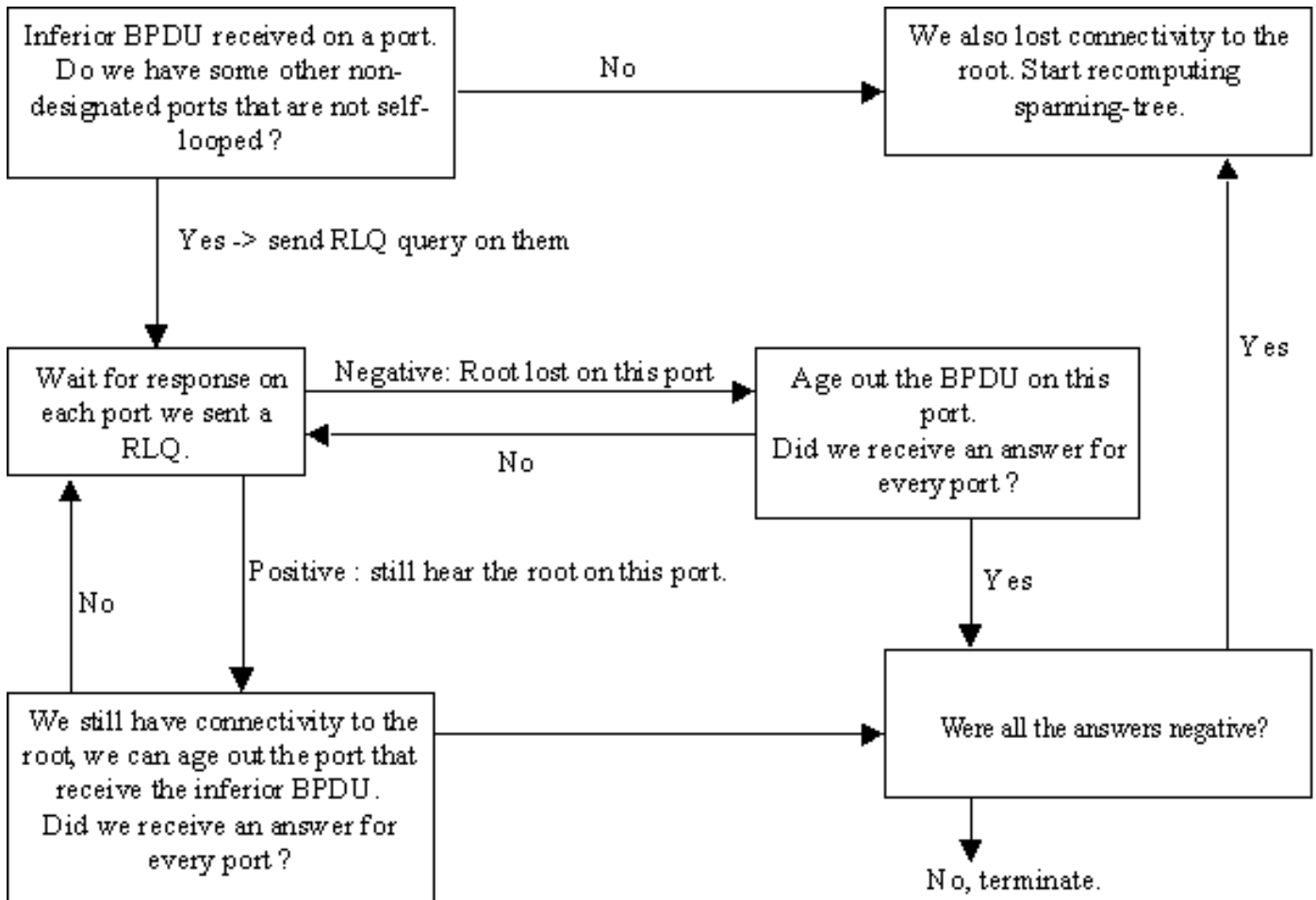


Der Verbindungsfehler tritt zwischen der Root-Bridge R und dem Hub auf. B erkennt nicht, dass die Verbindung ausfällt und wartet auf max\_age, bevor sie behauptet, der neue Root zu sein. Beachten Sie, dass der Mechanismus nur funktioniert, wenn eine Bridge einen direkten Verbindungsausfall erkennt.

Führen Sie nur eine Übersicht über untergeordnete BPDUs, die von der festgelegten Bridge gesendet wurden. Da dies die BDU ist, die auf dem Port gespeichert wird. Wenn beispielsweise eine neu eingelegte Bridge anfängt, unterlegene BPDUs zu senden, startet sie die Backbone Fast Feature nicht.

## Reaktion auf Fehler bei indirekten Verbindungen

Wenn eine untergeordnete BPDU auf einem nicht designierten Port erkannt wird, wird die zweite Phase des Backbone-Fast-Triggers ausgelöst. Statt passiv auf `max_age` zu warten, um Ports auszuschalten, die vom Ausfall betroffen sein können, wird mithilfe der RLQ PDU eine proaktive Möglichkeit eingeführt, diese sofort zu testen. Die RLQ wird verwendet, um eine Art von Ping für den Root an einem nicht designierten Port zu erreichen, und es wird ermöglicht, schnell zu bestätigen, ob die auf einem Port gespeicherte BPDU noch gültig ist oder verworfen werden muss.



Senden Sie nach Erhalt einer untergeordneten BPDU von einer designierten Bridge eine RLQ PDU an alle nicht designierten Ports außer dem Port, an den Sie die unterlegene BPDU und selbstgedrehte Ports erhalten haben. Dadurch wird überprüft, ob Sie immer noch vom Root der Ports hören, an denen Sie BPDUs empfangen. Der Port, an den Sie die untergeordnete BPDU erhalten haben, ist ausgeschlossen, da Sie bereits wissen, dass ein Ausfall aufgetreten ist, sich selbst Schleifen durchsetzende und designierte Ports nicht nützlich sind, da sie nicht zum Root führen.

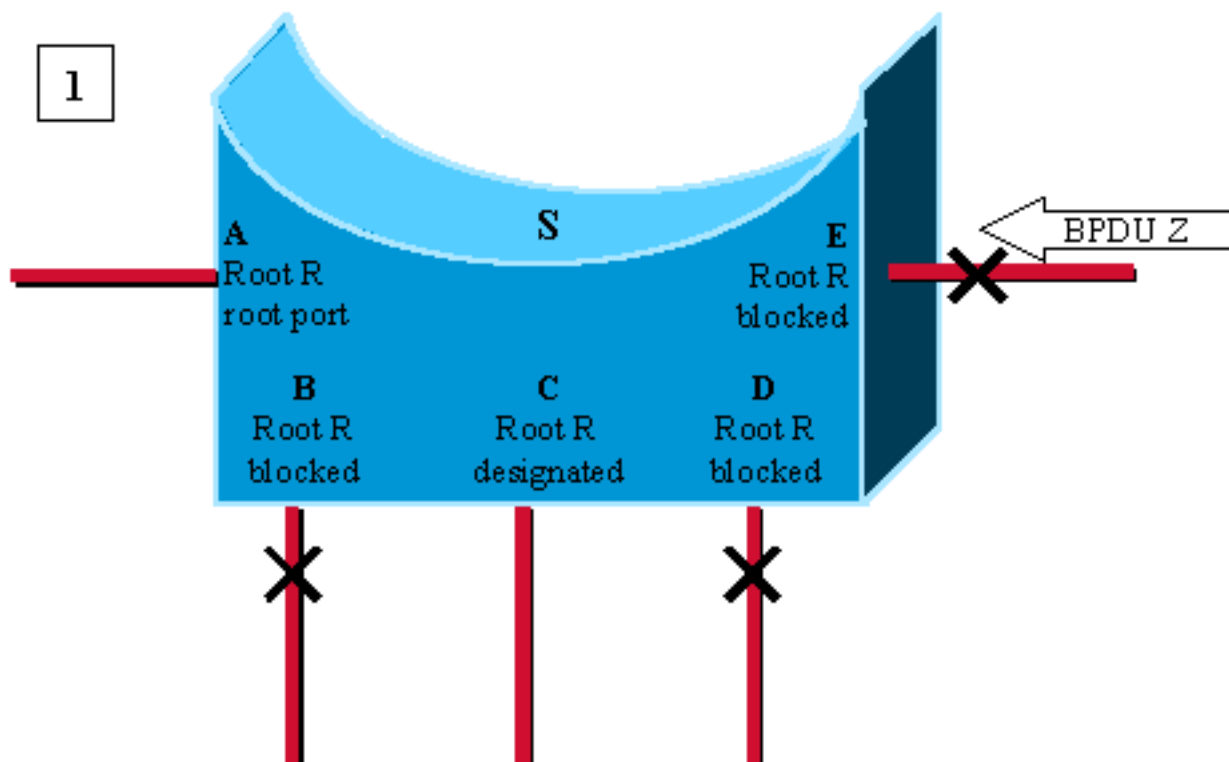
Wenn eine RLQ-Antwort an einem Port eingeht, hat der Port bei einer negativen Antwort die Verbindung zum Root verloren, und Sie können die BPDU löschen. Wenn darüber hinaus alle anderen nicht designierten Ports bereits eine negative Antwort erhalten haben, verliert die gesamte Bridge die Root und kann die STP-Berechnung von Grund auf neu starten.

Wenn die Antwort bestätigt, dass Sie weiterhin über diesen Port auf die Root Bridge zugreifen können, können Sie sofort den Port ausweisen, an dem die untergeordnete BPDU ursprünglich empfangen wurde.

In diesem Beispiel sind die Ports A, B, D und E nicht designierte Ports für Switch S. A ist der Root-

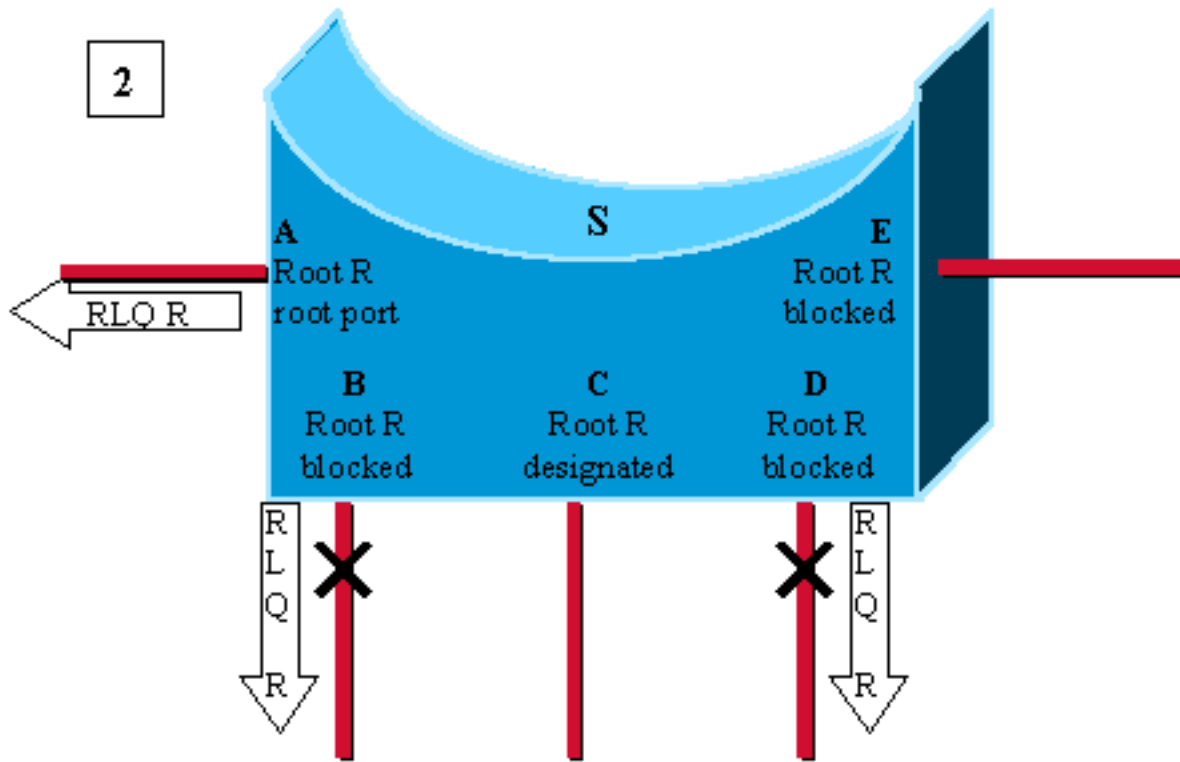
Port, die anderen blockieren. Wenn E eine unterlegene BPDUs (1) erhält, wird die Neuberechnung des STP durch Backbone schnell beschleunigt.

Senden Sie eine RLQ-Anfrage, die nach Root R an allen nicht designierten Ports außer E (2) sucht. Die Antworten geben an, auf welchen Root über diese Ports zugegriffen werden kann. Die RLQ-Antwort, die D erhält, gibt an, dass D seinen Pfad zum Root-R verloren hat. BPDUs sofort veralten (3). Die Ports A und B erhalten die Bestätigung, dass sie noch einen Pfad zu R (4) haben. Da der Switch S immer noch über eine Verbindung zum Root verfügt, wird der Port E sofort abgemeldet und die regulären STP-Regeln werden eingehalten (5).



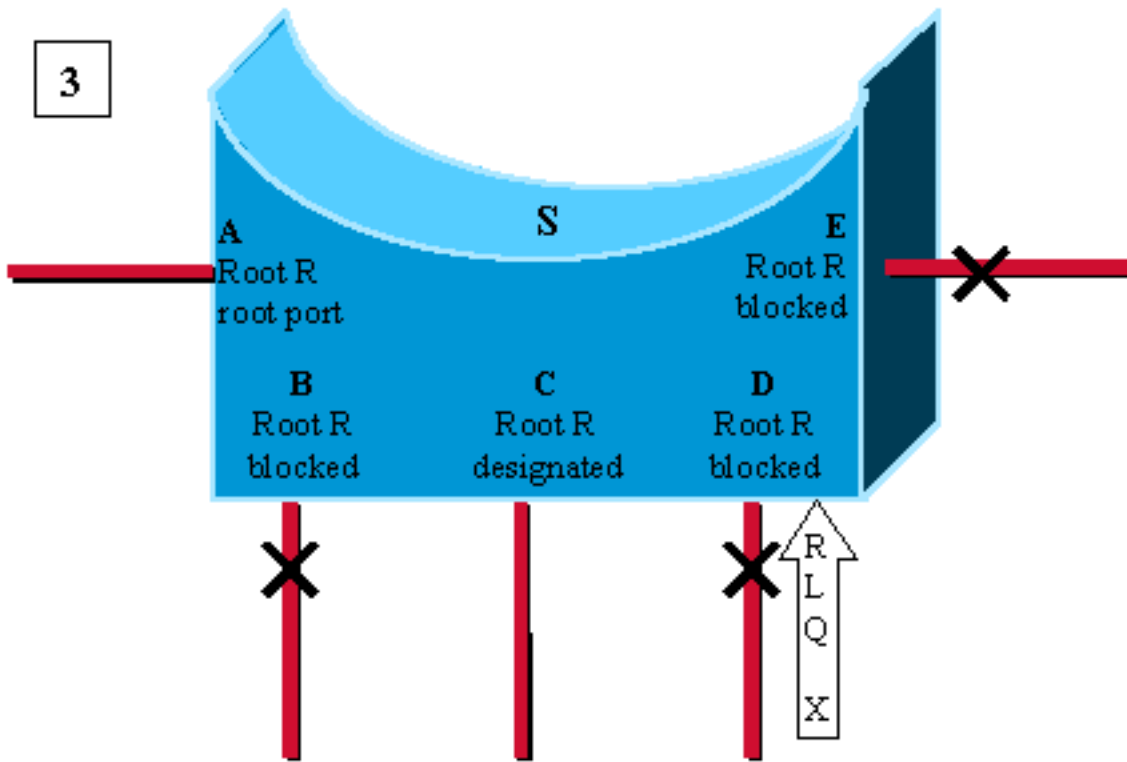
Port E receives an inferior BPDUs, advertising root Z instead of root R stored on the different ports.

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.

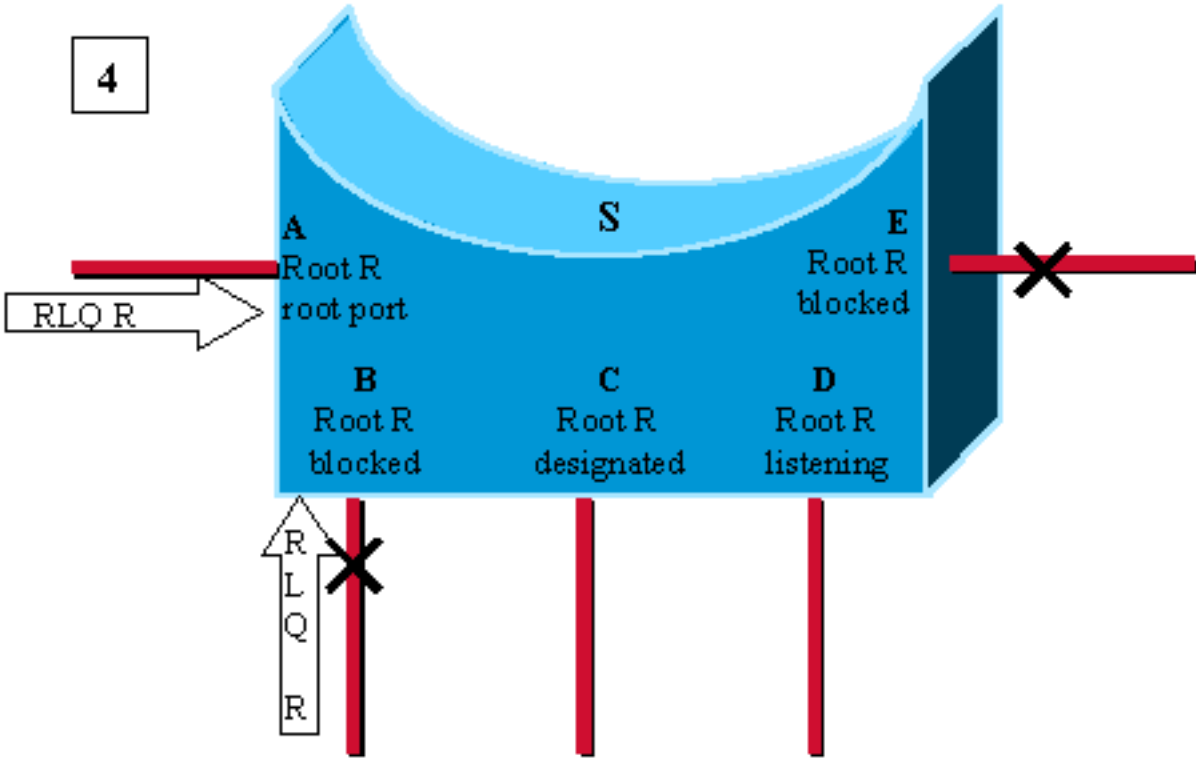
3



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.

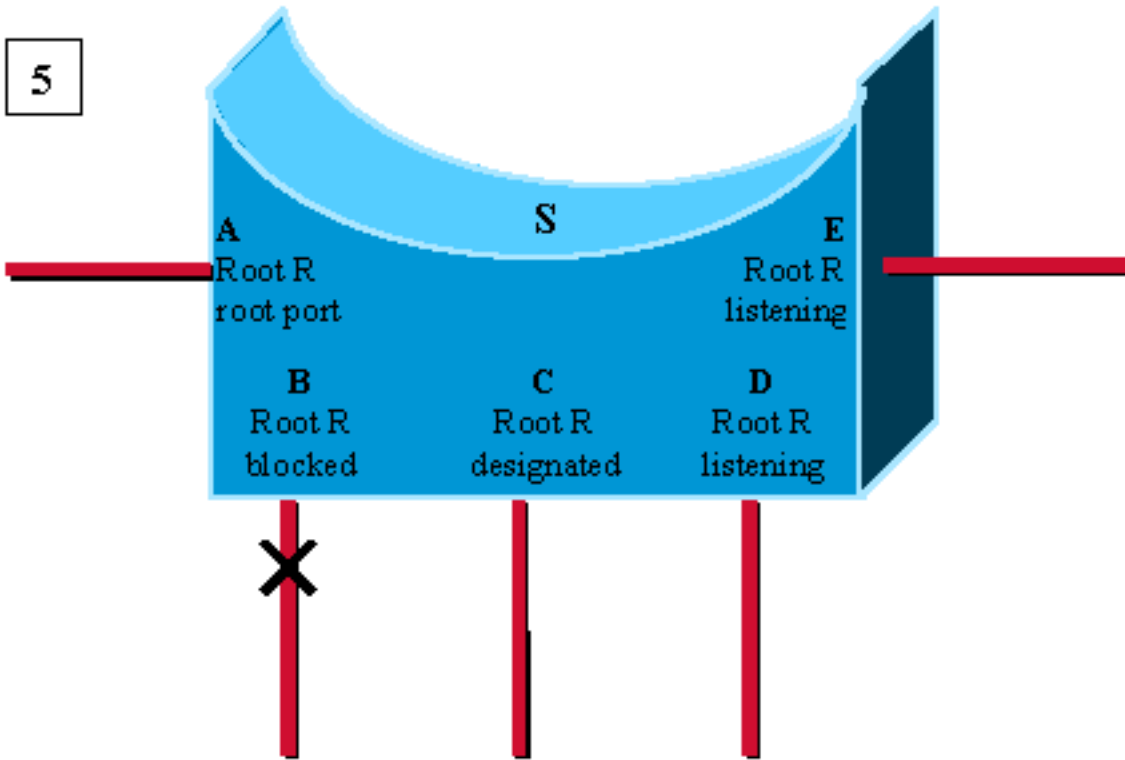


4



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.

Wenn der Switch nur Antworten mit einem anderen Root als R erhalten hat, betrachten Sie den Root als verloren und starten die STP-Berechnung sofort neu. Beachten Sie, dass dies auch der Fall ist, wenn der einzige nicht designierte (und nicht selbstschleifende) Port auf der Bridge der Root-Port ist und Sie an diesem Port eine unterlegene BPDU erhalten.

## PDU für die Root-Link-Abfrage

Die beiden Formen von RLQs sind RLQ-Anfragen und RLQ-Antworten.

Die RLQ-Anfrage wird an einen Port gesendet, an dem Sie normalerweise BPDUs empfangen, um sicherzustellen, dass Sie weiterhin über diesen Port eine Verbindung zum Root haben. Geben Sie in der Anfrage an, welche Bridge Ihr Root ist und die RLQ-Antwort schließlich mit einer Root Bridge zurückgesendet wird, auf die über diesen Port zugegriffen werden kann. Wenn die beiden Wurzeln gleich sind, ist die Konnektivität immer noch lebendig, sonst geht sie verloren.

Eine Bridge, die eine RLQ-Anforderung empfängt, beantwortet sofort, wenn sie weiß, dass sie die Verbindung zum abgefragten Root verloren hat, da sie über eine Root-Bridge verfügt, die von der in der RLQ-Abfrage angegebenen Bridge abweicht, und wenn sie der Root ist.

Ist dies nicht der Fall, leitet er die Abfrage über den Root-Port an den Root weiter.

RLQ-Antworten werden an designierten Ports überflutet. Der Absender der RLQ-Anfrage legt seine Bridge-ID in die PDU. Dadurch soll sichergestellt werden, dass eine Antwort, die eine

Antwort auf eine eigene Anfrage erhält, nicht über die zugewiesenen Ports geflutet wird.

Die RLQ PDU hat dieselbe Paketstruktur wie eine normale STP-BPDU. Der einzige Unterschied besteht darin, dass zwei verschiedene Cisco-spezifische SNAP-Adressen verwendet werden: eine für den Antrag und eine für die Antwort.

Dies ist das standardmäßige BPDU-Format:

**DA SA Länge DSAP SSAP STRG SNAP PDU**

Das PDU-Feld ist:

**Protokollkennung Version Nachrichtentyp Flags Root-ID Root Path-Kosten  
Absender-ID Port-ID Nachrichtenalter Max. Alter Hello-Zeit Vorwärtsverzögerung**

Der in der PDU verwendete Meldungstyp unterscheidet sich ebenfalls von der standardmäßigen BPDU.

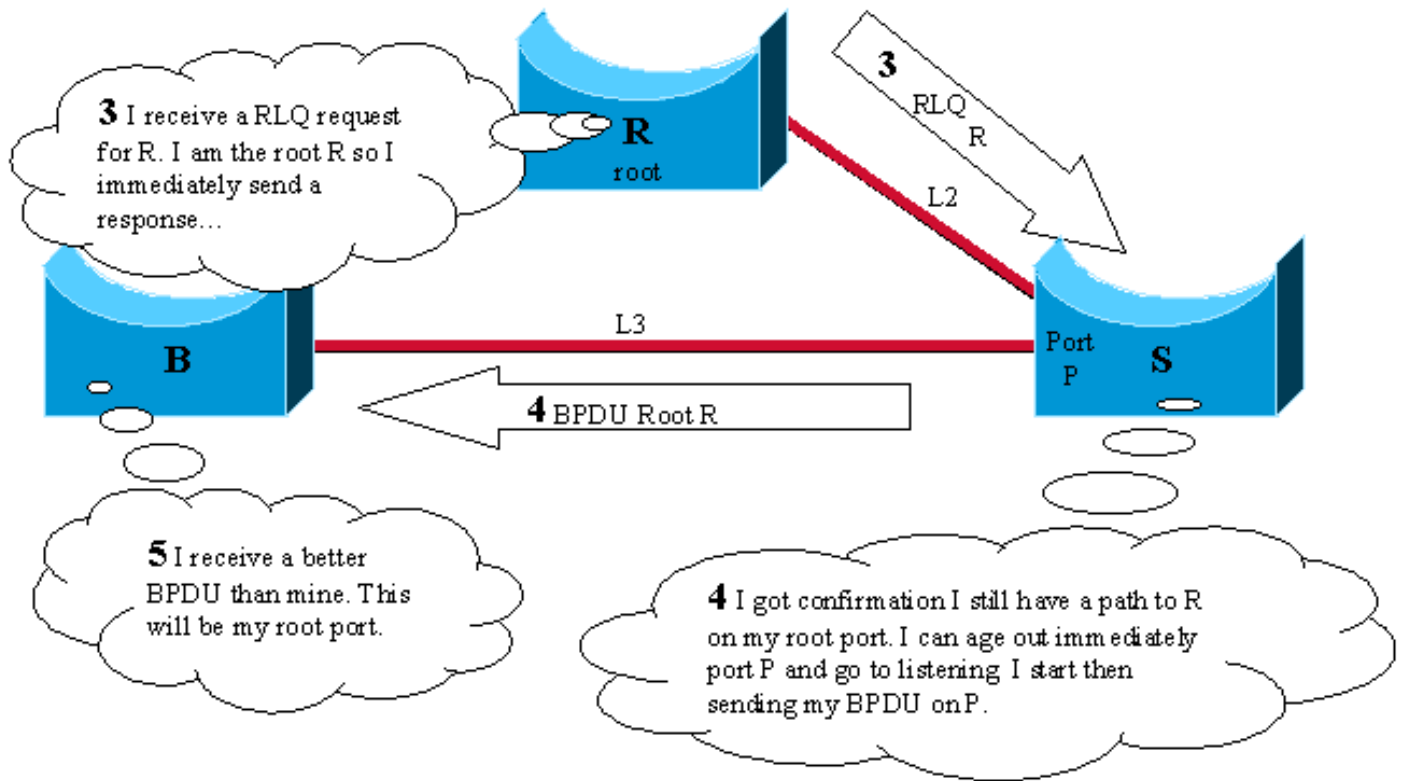
Es werden nur die Root-ID und die Sender Bridge-ID verwendet.

Diese Cisco-spezifische Funktion muss auf allen Switches im Netzwerk konfiguriert werden, um diese PDUs zu verarbeiten.

## **Beispielszenario mit aktivierter Backbone Fast-Funktion**

Dieses Szenario basiert auf dem ersten Beispiel, diesmal jedoch mit Backbone Fast auf den drei Switches.

1. Die erste Phase ist genau die gleiche wie zuvor erklärt.
2. Sobald S die untergeordnete BPDU von B empfängt, beginnt es, seine nicht designierten Ports erneut zu bestätigen, anstatt auf max\_age zu warten. Er sendet eine RLQ-Abfrage an seinen Root-Port für Root Bridge R.
3. Root Bridge R empfängt die Abfrage und antwortet sofort mit einer RLQ-Antwort, die angibt, dass noch ein Root-R in diese Richtung vorhanden ist.
4. S hat jetzt alle nicht designierten Ports geprüft und verfügt weiterhin über eine Verbindung zum Root. Anschließend kann er die auf Port P gespeicherten Informationen sofort abrufen und anschließend BPDUs senden. In dieser Phase haben Sie bereits max\_age Sekunden gespeichert, und dann gilt der Spanning-Tree-Algorithmus (STA).
5. B empfängt das bessere BPDU von S (R besser Root als B) und betrachtet nun die Ports, die zu L3 führen, als seinen Root-Port.



## Konfigurieren von Backbone Fast für CatOS und Cisco IOS

Bei Verwendung muss Backbone Fast auf allen Switches im Netzwerk aktiviert werden, da Backbone schnell die Verwendung des RLQ Request and Reply-Mechanismus erfordert, um die Switches über die Stabilität des Root-Pfads zu informieren. Das RLQ-Protokoll ist nur aktiv, wenn Backbone Fast auf einem Switch aktiviert ist. Darüber hinaus kann das Netzwerk auch bei RLQ-Flooding auf Probleme stoßen, wenn Backbone Fast nicht auf allen Switches aktiviert ist. Standardmäßig ist Backbone Fast deaktiviert.

Fast Backbone wird auf Catalyst Switches der Serien 2900XL und 3500XL nicht unterstützt. Im Allgemeinen müssen Sie Backbone schnell aktivieren, wenn die Switch-Domäne diese Switches zusätzlich zu anderen unterstützten Catalyst Switches enthält. Wenn Sie Backbone schnell in Umgebungen mit XL-Switches implementieren, können Sie unter strengen Topologien die Funktion aktivieren, bei der der XL-Switch der letzte Line-Switch ist und nur an zwei Stellen mit dem Core verbunden ist. Implementieren Sie diese Funktion nicht, wenn sich die Architektur der XL-Switches in Reihenschaltung befindet.

Sie müssen den Backbone nicht schnell mit RSTP oder IEEE 802.1w konfigurieren, da der Mechanismus nativ integriert und automatisch im RSTP aktiviert ist. Weitere Informationen zu RSTP oder IEEE 802.1w finden Sie unter [Konfigurationsbeispiel für die Spanning Tree-Migration von PVST+ zu Rapid-PVST](#).

### Konfiguration für CatOS

Verwenden Sie für Catalyst Switches der Serien 4000, 5000 und 6000, die CatOS ausführen, diese Befehle, um Backbone global schnell für alle Ports zu aktivieren und die Konfiguration zu überprüfen.

```
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)
```

So zeigen Sie schnelle Backbone-Statistiken an:

```
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)
```

## Konfiguration für Cisco IOS

Verwenden Sie für Catalyst-Switches, die mit der Cisco IOS-Software ausgeführt werden, diese Befehle, um das Backbone global schnell für alle Schnittstellen zu ermöglichen.

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

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

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

```
CAT-IOS#
```

So überprüfen Sie, ob Backbone Fast aktiviert ist, und zeigen Statistiken an:

```
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#
```

## Zugehörige Informationen

- [Verwenden von PortFast und anderen Befehlen zum Beheben von Verzögerungen bei der Workstation-Startverbindung](#)
- [Verständnis und Konfiguration der Cisco UplinkFast-Funktion](#)

- [Erweiterungen des Spanning Tree Protocol mit Loop Guard und BPDU Skew Detection-Funktionen](#)
- [Spanning Tree PortFast BPDU Guard-Erweiterung](#)
- [Konfigurationsbeispiel für die Migration von PVST+ zu Rapid-PVST mit Spanning Tree](#)
- [Spanning Tree Protocol - Technologieunterstützung](#)
- [Switches - Produktsupport](#)
- [LAN-Switching - Technologieunterstützung](#)
- [Technischer Support und Dokumentation - Cisco Systems](#)