Поймите и настройте Backbone Fast на коммутаторах Catalyst

Содержание

Введение

Этот документ описывает, как настроить Backbone Fast. Backbone Fast — это собственная функция Cisco, которая, будучи включенной на всех коммутаторах сети с мостовыми подключениями, может ускорить восстановление коммутатора после сбоя обходного канала на 20 секунд (max_age). После краткого знакомства с основами протокола STP (Spanning-Tree Protocol) вы сможете проанализировать точный сценарий сбоя, при котором применяется функция Backbone Fast, а также узнаете, как настраивать ее для коммутаторов Catalyst, работающих под управлением CatOS и программного обеспечения Cisco IOS®.

Предварительные условия

Требования

Для этого документа отсутствуют особые требования.

Используемые компоненты

Сведения, содержащиеся в данном документе, касаются следующих версий программного обеспечения и оборудования:

- Коммутаторы Catalyst серии 2950, работающие под управлением программного обеспечения Cisco IOS версии 12.1(6)EA2 и выше
- Коммутаторы Catalyst серии 3550, работающие под управлением программного

обеспечения Cisco IOS версии 12.1(4)EA1 и выше
- Коммутаторы Catalyst серии 4000, работающие под управлением CatOS версии 5.1(1a) и выше
- Коммутаторы Catalyst серий 4500/4000, работающие под управлением программного обеспечения Cisco IOS версии 12.1(8a)EW и выше
- Коммутаторы Catalyst серий 5500/5000, работающие под управлением CatOS версии 4.1(1) и выше
- Коммутаторы Catalyst серий 6500/6000, работающие под управлением CatOS версии 5.1(1)CSX и выше
- Коммутаторы Catalyst серий 6500/6000, работающие под управлением программного обеспечения Cisco IOS версии 12.0-7XE и выше

BPDU и их сравнение

Блоки данных протокола моста (Bridge Protocol Data Unit — BPDU) можно строго классифицировать по содержащимся в них полям. Эти поля включают идентификатор корневого моста, стоимость пути к корню и идентификатор передающего моста. Блок BPDU считается лучше, чем другой BPDU, в следующих случаях:

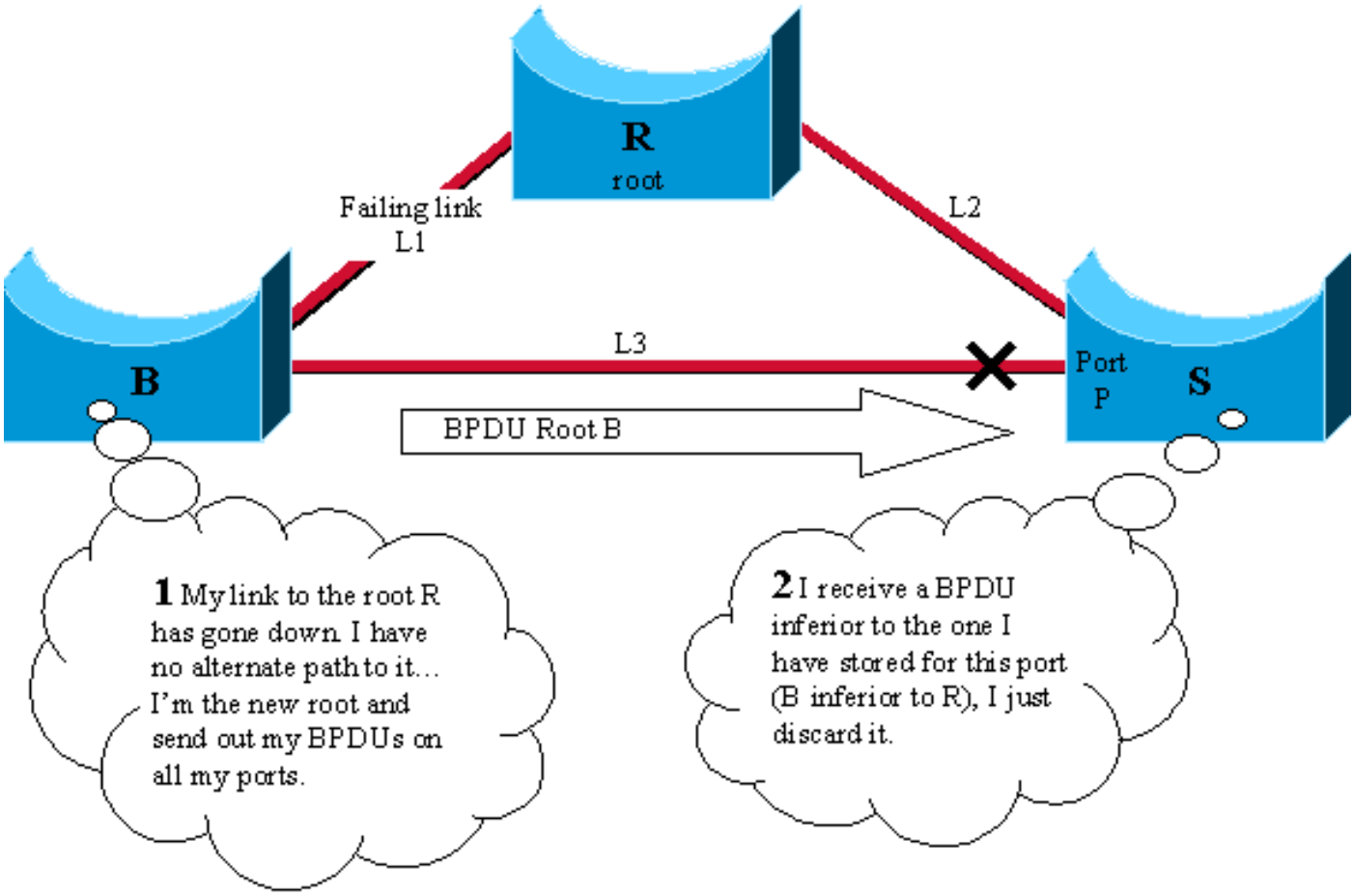
- Когда одна BPDU содержит более подходящий идентификатор корневого моста, чем другой (чем меньше значение, тем лучше).
- Если значения идентификаторов корневого моста одинаковы, то блок BPDU с наименьшей стоимостью пути к маршруту лучше.
- Когда значения идентификаторов корневых мостов и стоимости маршрутов до корня одинаковы предпочтительней BPDU с лучшим идентификатором передающего моста (чем ниже значение, тем лучше).

Существуют и другие переменные, которые могут выполнять функции арбитра, однако чем лучше BPDU, тем лучшим будет доступ к наилучшему корневому мосту.

Мост, порт которого принимает BPDU лучше отправленного, устанавливает этот порт в режим блокировки, если это не его корневой порт. Это означает, что на сегменте, подключенном к данному порту, существует другой мост, который является назначенным мостом. Мост сохраняет значение BPDU на порту, отправленное текущим назначенным мостом.

Восстановление STP после сбоя обходного канала

Ниже показано поведение STP, когда приходится произвести пересчет параметров после сбоя обходного канала, то есть, когда мост должен изменить состояние некоторых из своих портов из-за отказа на канале, который не соединен с ним напрямую.



Рассмотрим эту схему, которая включает три коммутатора R, B и S в полносвязанной топологии. Предположим, что R — корневой мост, а B — резервный корневой мост. S блокирует порт P, а B является назначенным мостом для канала L3.

1. Если канал L1 выходит из строя, коммутатор B сразу обнаруживает сбой и предполагает, что он связан с корнем. Он начинает отправлять блоки BPDU коммутатору S и объявляет себя новым корнем.
2. Когда S получает новый блок BPDU от B, он определяет, что этот BPDU является подчиненным по отношению к сохраненному блоку BPDU для порта P, и игнорирует его.
3. По истечении срока таймера max_age (по умолчанию 20 секунд) блок BPDU, сохраненный на коммутаторе S для порта P, устаревает. Порт сразу переходит в состояние прослушивания, и S начинает отправлять свой блок BPDU с лучшими параметрами коммутатору B.
4. Как только B получает блок BPDU от S, он прекращает отправку своих BPDU.
5. Порт P переходит в состояние пересылки через промежуточные состояния прослушивания и обучения. Это занимает период, равный удвоенному значению fw_delay, — дополнительные 30 секунд. Затем соединение полностью восстанавливается.

Восстановление после сбоя обходного канала занимает период, равный сумме значения max_age (20 секунд) и удвоенного значения fw_delay (2x15 секунд). При использовании параметров по умолчанию это составляет 50 секунд. Функция Backbone Fast позволяет сэкономить max_age (20 секунд). Чтобы реализовать эту возможность, BPDU устаревает сразу после того, как порт получил подчиненные блоки BPDU.

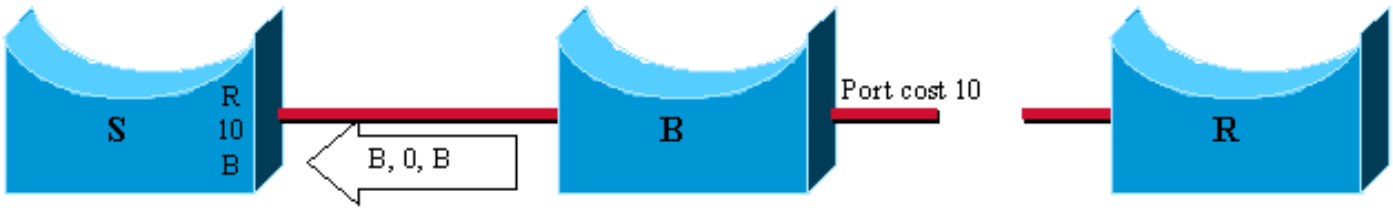
Усовершенствования стандартного протокола STP с помощью функции Backbone Fast

В предыдущем примере STP признает недействительными сведения, которые стали неверными из-за отказа обходного канала. Для этого он пассивно ожидает в течение периода max_age. Чтобы исключить эту задержку max_age, функция Backbone Fast предусматривает два усовершенствования:

- Возможность как можно раньше обнаруживать сбой обходного канала. Это реализуется путем отслеживания подчиненных блоков BPDU, которые отправляет назначенный мост при возникновении сбоя прямого канала.
- Механизм, позволяющий сразу проверять, по-прежнему ли действительна информация BPDU, которая хранится на порту. Для этого вводится новый блок данных протокола (PDU) и запрос корневого канала (что обозначается в данном документе как RLQ PDU).

Обнаружьте сбои обходного соединения

Если подчиненный BPDU получен на порту от нашего назначенного моста, то этот мост потерял root и начинает объявлять root с более высоким идентификатором моста, худший root, чем наш.



1 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.

Стандартное поведение, согласно спецификациям Института инженеров по электротехнике и электронике (IEEE), — просто проигнорировать все подчиненные BPDU. Функция Backbone Fast использует их, поскольку при получении такого BPDU становится ясно, что на пути к корню произошел сбой и информация хотя бы на одном порту должна быть признана устаревшей.

> **Примечание:** Сбой обходного соединения может произойти без любой генерации подчиненного BPDU в сети. Просто добавим концентратор на предыдущую схему:

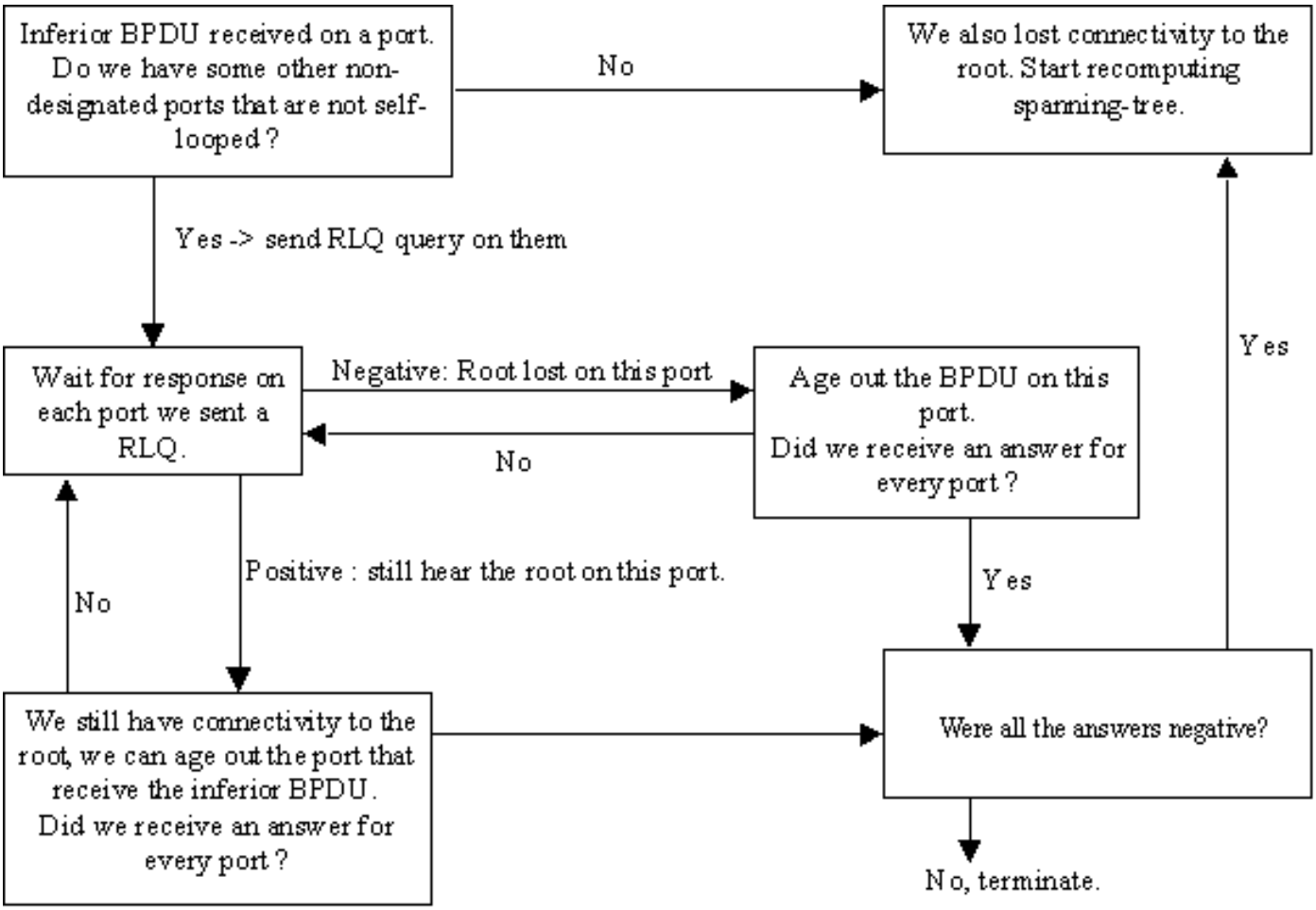
Сбой связи происходит между корневым мостом R и концентратором. B не обнаружил, что канал вышел из строя, и ждет в течение периода max_age, прежде чем объявить себя новым корнем. Помните, что этот механизм работает, только если мост обнаруживает сбой

прямого канала.

Отслеживаются только подчиненные BPDU, отправленные назначенным мостом. Поскольку это BPDU, которые хранятся на порту. Если, например, новый созданный мост начнет отправлять подчиненные BPDU, функция Backbone Fast активирована не будет.

Реагируйте на сбои обходного соединения

Когда подчиненный BPDU обнаружен на неназначенном порту, инициируется второй этап функции Backbone Fast. Вместо того, чтобы пассивно ждать в течение периода max_age, прежде чем признать устаревшей информацию на портах, на которые мог повлиять сбой, механизм RLQ PDU позволяет сразу проверить эти порты. RLQ используется для проведения своего рода эхо-запроса корня на неназначенном порту и позволяет быстро подтвердить, является ли BPDU, который хранится на порту, по-прежнему действительным или его нужно отбросить.



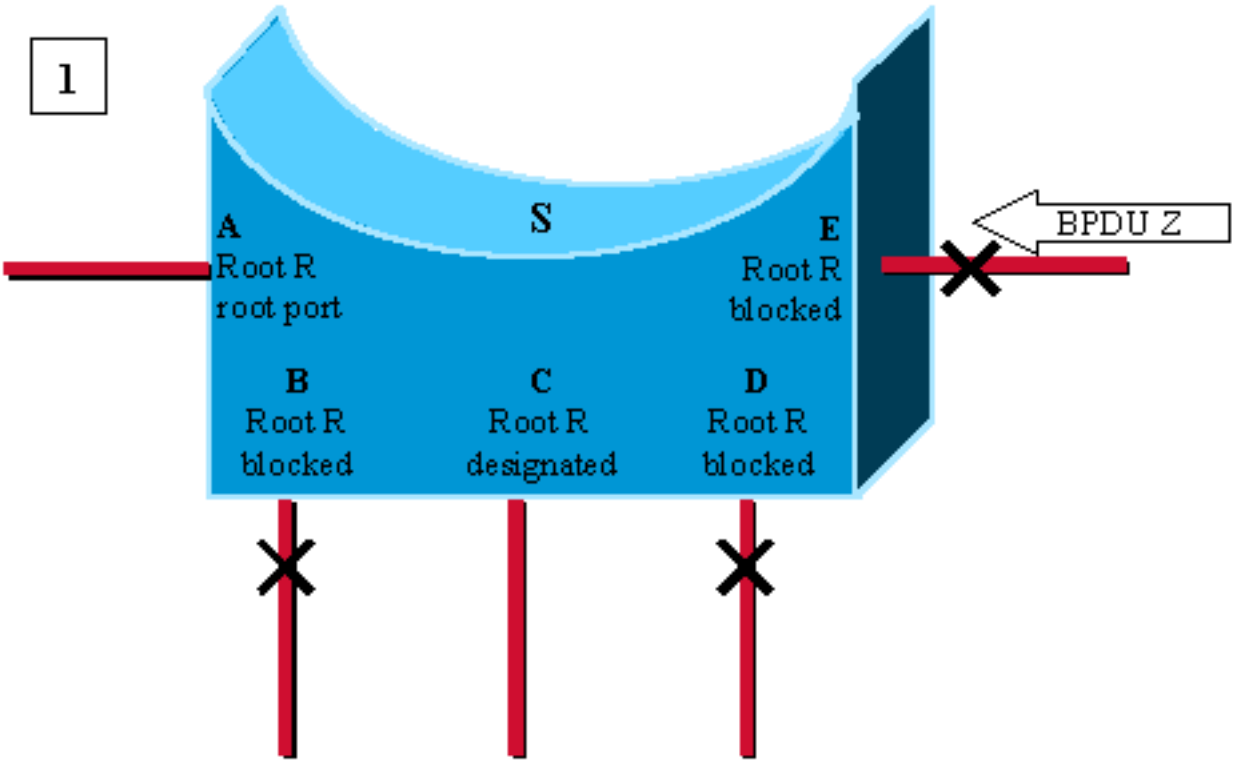
После получения подчиненного BPDU от назначенного моста отправьте запрос RLQ PDU на все неназначенные порты, кроме порта, на котором получен подчиненный BPDU, и портов с зацикливанием на себя. Это позволит проверить, по-прежнему ли поступают данные с корня на портах, которые использовались для получения BPDU. Порт, на который поступил подчиненный BPDU, исключен, поскольку вы уже знаете о его неисправности; порты с зацикливанием на себя и назначенные порты непригодны, так как они не ведут к корню.

Если в ответ на запрос RLQ на порту получен отрицательный ответ, то порт потерял соединение с корнем и можно признать его BPDU устаревшим. Более того, если все прочие неназначенные порты уже получили отрицательный ответ, то мост в целом потерял корень и может начинать расчеты параметров STP с нуля.

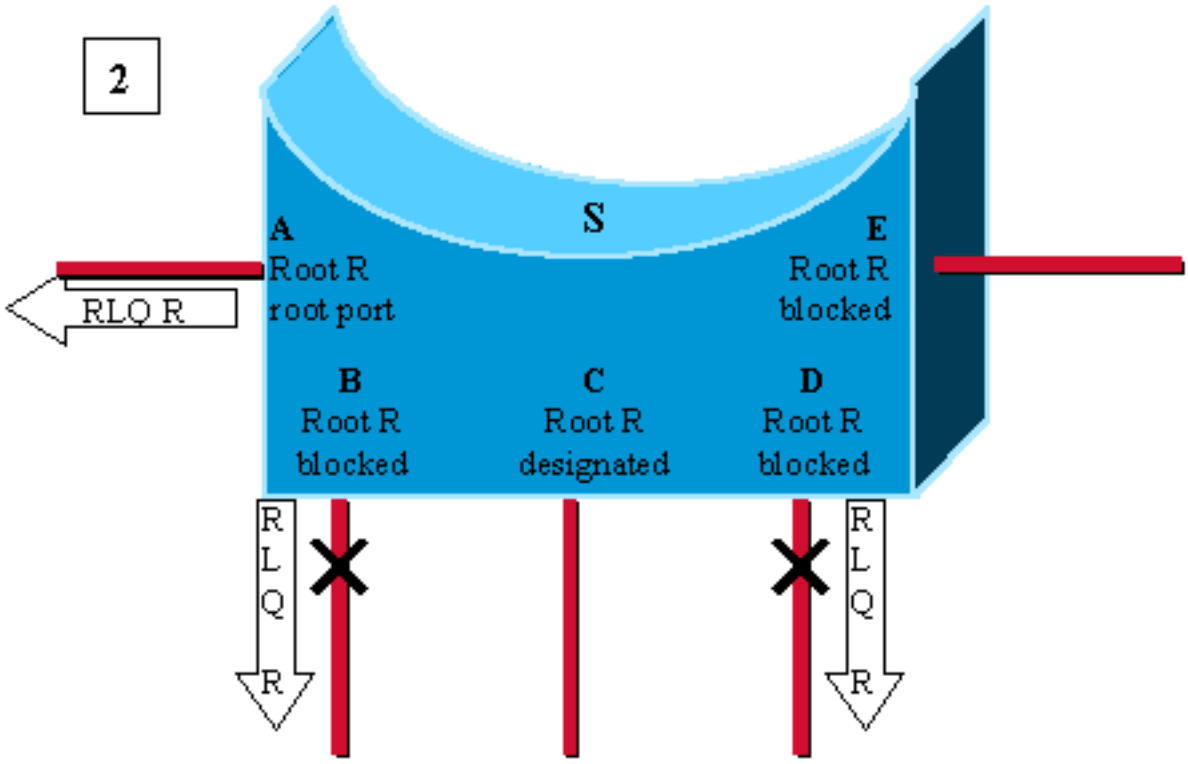
Если ответ подтверждает наличие доступа к корневому мосту через данный порт, можно сразу признать устаревшими данные на порту, на который изначально поступил подчиненный BPDU.

В приведенном примере порты A, B, D и E являются неназначенными портами для коммутатора S. A — это корневой порт, а остальные — блокирующие. Когда порт E получает подчиненный BPDU (1), активизируется функция Backbone Fast, ускоряющая пересчет параметров STP.

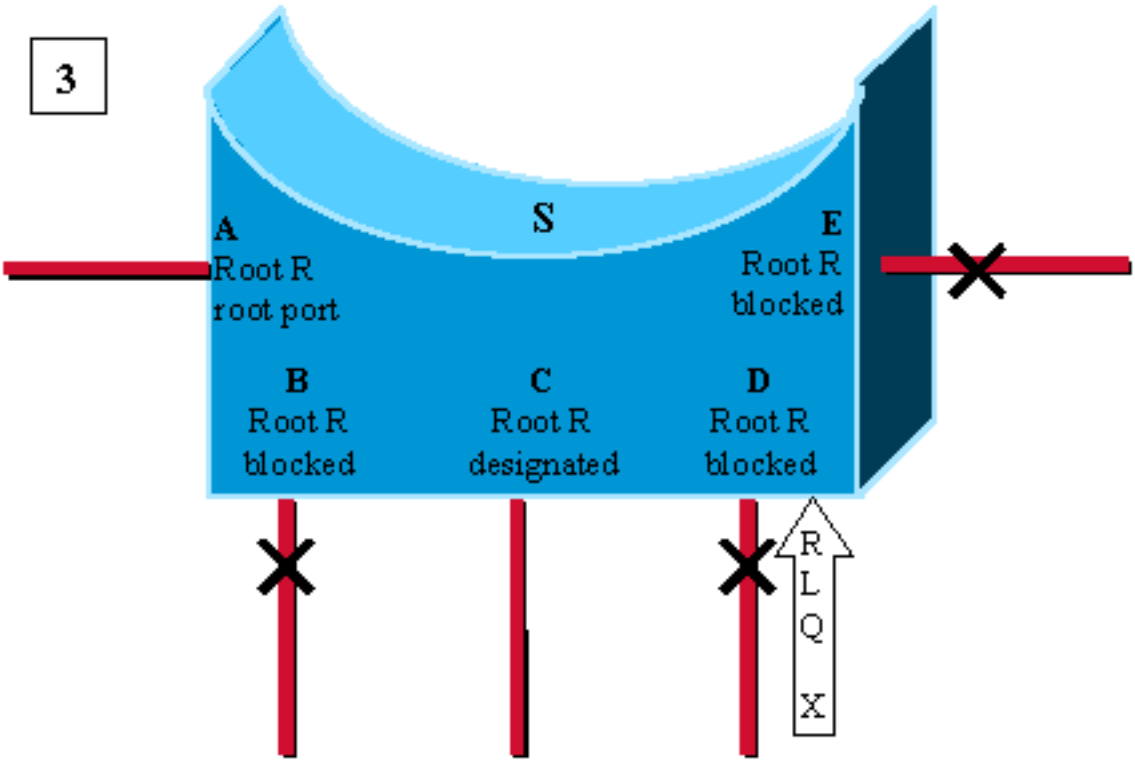
Отправим RLQ-запрос для поиска корня R на всех неназначенных портах, кроме E (2). В ответе будет указан корень, доступный через эти порты. В ответе RLQ, который получает D, сообщается, что D потерял свой путь к корню R. Сразу признаем его BPDU устаревшим (3). Порты A и B получают подтверждение о наличии пути к R (4). Поскольку коммутатор S по-прежнему соединен с корнем, немедленно признаем данные на порту E устаревшими и будем продолжать в соответствии с обычными правилами STP (5).



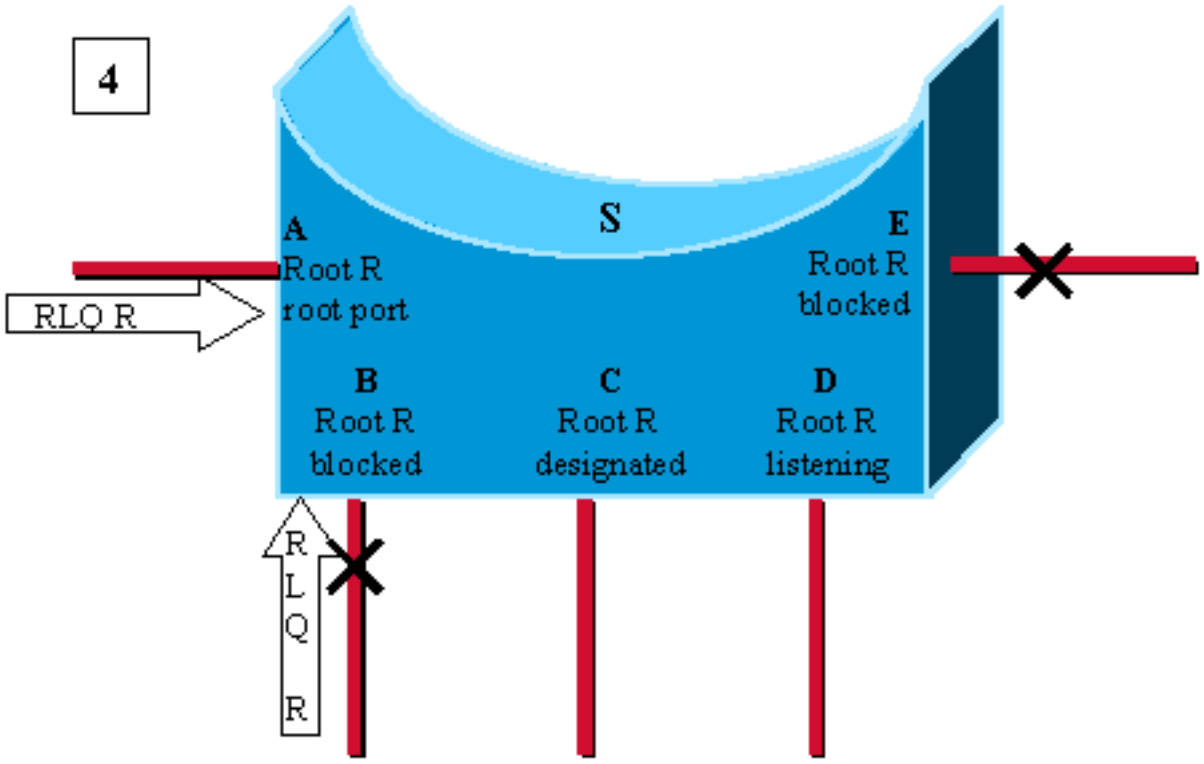
Port E receives an inferior BPDU, advertising root Z instead of root R stored on the different ports.



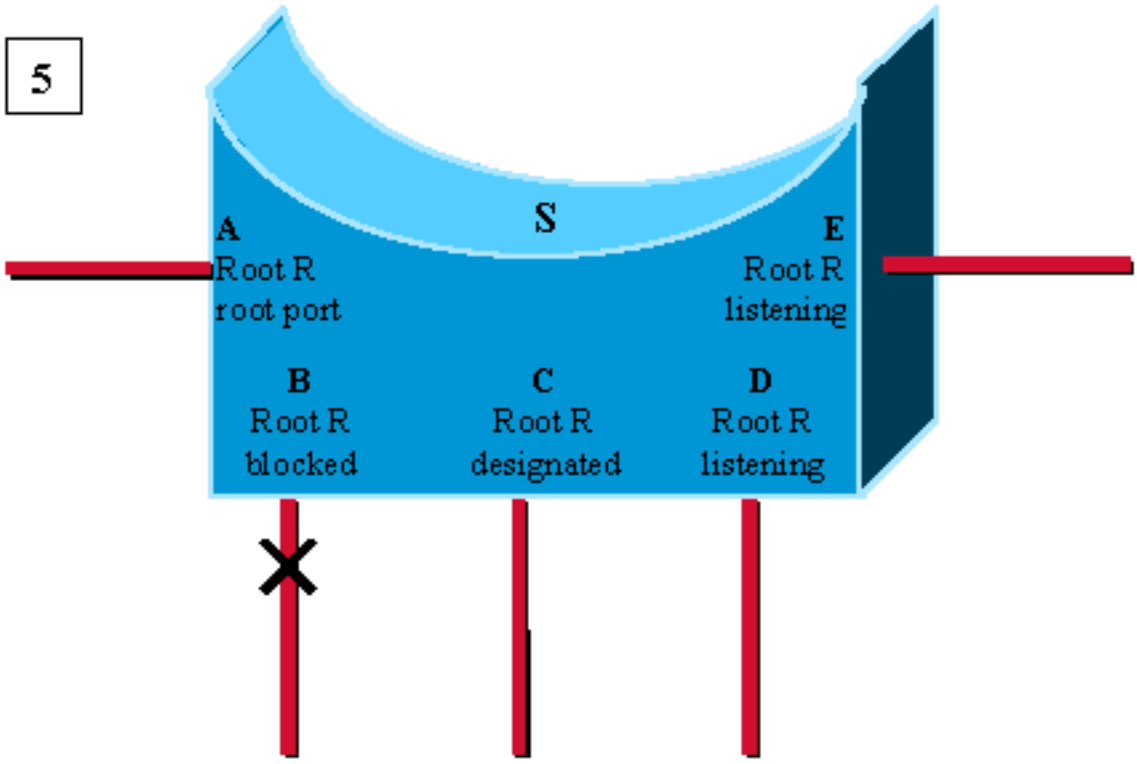
Switch S needs to recheck all its other non-designated ports. It sends out a RLQ request for root R on ports A,B and D.



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.



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 BPDU stored on port E.



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.

Если коммутатор получил только ответы, где был указан корень, отличный от R, то корень считается потерянным и сразу запускается пересчет параметров STP с нуля. Следует отметить, что такой случай также имеет место, когда единственный неназначенный порт (без зацикливания на себя) на мосту является корневым и на этот порт поступает подчиненный BPDU.

PDU запроса корневого канала

Существуют две формы RLQ — запросы RLQ и ответы RLQ.

Запрос RLQ отправляется на порт, на котором вы обычно получаете BPDU, с целью проверить, существует ли еще подключение к корню через этот порт. Укажите в запросе, какой мост является корневым, и в ответе RLQ будет указан корневой мост, доступный через этот порт. Если эти два корневых моста совпадают, соединение существует, в противном случае оно потеряно.

Мост, на который поступает запрос RLQ, сразу отвечает, если ему достоверно известно о потере соединения с корнем, указанным в запросе RLQ, поскольку корневой мост отличен от запрашиваемого, и если этот мост является корневым.

Иначе этот мост пересылает запрос корню через свой корневой порт.

Ответы RLQ передаются на назначенные порты. Отправитель запроса RLQ указывает свой

идентификатор моста в PDU. Это гарантирует, что при получении ответа на собственный запрос он не будет рассылать ответ на все назначенные порты.

Структура пакетов RLQ PDU аналогична обычным блокам BPDU STP. Единственное различие заключается в использовании двух особых SNAP-адресов Cisco: одного для запроса и одного для ответа.

Это стандартный формат BPDU:

DA SA Длина DSAP SSAP CNTL SNAP PDU

Поле PDU:

Идентификатор протокола	Version	Тип сообщения	Флаги	Идентификатор корня	Стоимость корневого пути
Идентификатор отправителя	Идентификатор порта	Возраст сообщения	Max age	Время приветствия	Forward delay

Тип сообщения, используемый в PDU, также отличается от стандартного BPDU.

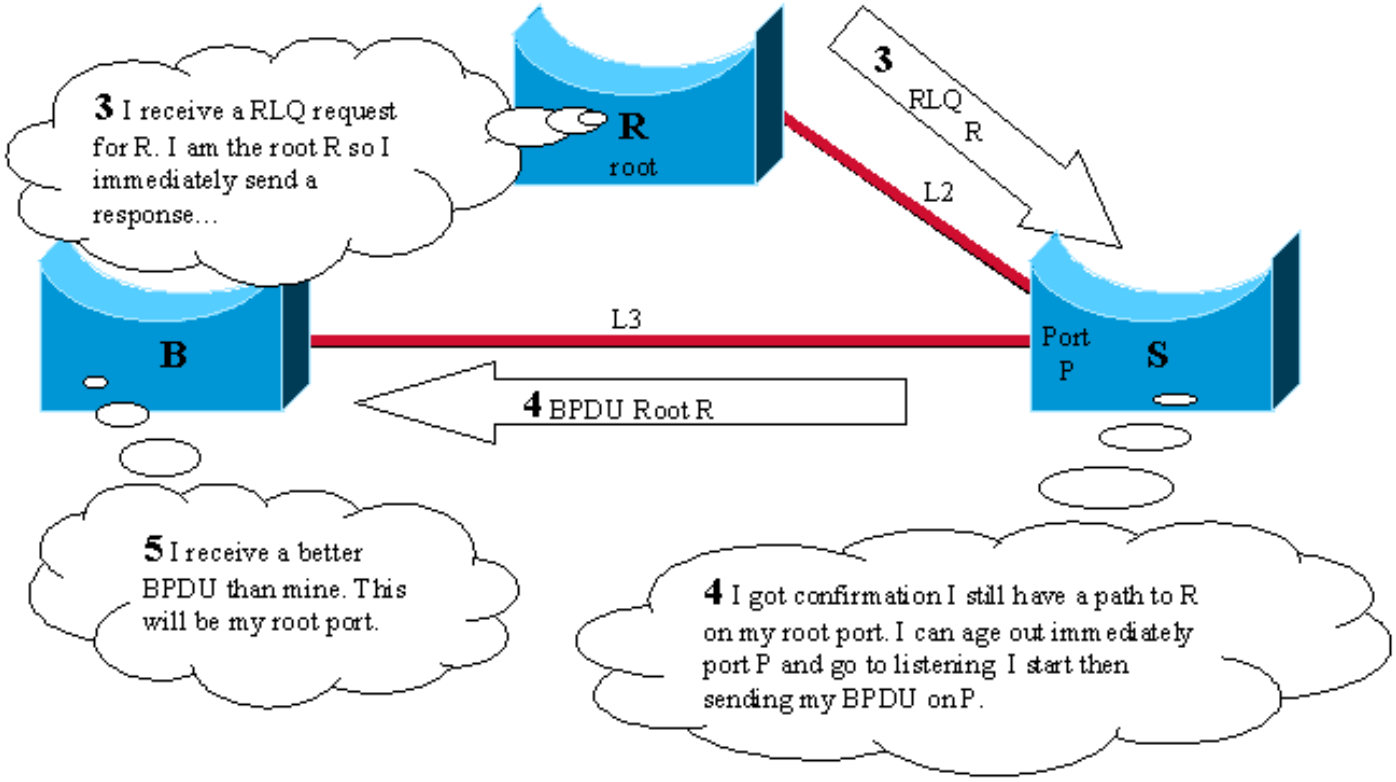
Используются только поля идентификатора корня и идентификатора передающего моста.

Эту особую функцию Cisco нужно настроить на всех коммутаторах в сети, чтобы обрабатывать PDU.

Пример сценария с включенной функцией Backbone Fast

В основе данного сценария используется первый пример, однако на этот раз функция Backbone Fast включена на трех коммутаторах.

1. Первый этап идентичен описанной выше процедуре.
2. Как только S получает от B подчиненный BPDU, он заново подтверждает свои неназначенные порты, не ожидая в течение срока max_age. Он отправляет RLQ-запрос корневого моста R на свой корневой порт.
3. Корневой мост R получает запрос и сразу отправляет RLQ-ответ, в котором указано, что на этой линии существует корень R.
4. К этому моменту S проверил все свои неназначенные порты и по-прежнему соединен с корнем. Затем он сразу признает устаревшей информацию, которая хранится на порту P. P переходит в состояние прослушивания и начинает рассылать BPDU. На этом этапе вы уже сэкономили max_age секунд. Затем применяется стандартный алгоритм связующего дерева (Spanning-Tree Algorithm — STA).
5. B получает от S BPDU с лучшими параметрами (R — лучший корень, чем B) и теперь считает порт, ведущий к L3, своим корневым портом.



Настройте Backbone Fast для CatOS и Cisco IOS

Функция Backbone Fast должна быть включена на всех коммутаторах в сети, поскольку для ее работы требуется механизм запросов и ответов RLQ, чтобы сообщать коммутаторам об устойчивости пути к корню. Протокол RLQ активен, только когда функция Backbone Fast включена на коммутаторе. Кроме того, в сети могут возникнуть проблемы, связанные с лавинной рассылкой RLQ, если функция Backbone Fast включена не на всех коммутаторах. По умолчанию функция Backbone Fast отключена.

Функция Backbone Fast не поддерживается коммутаторами Catalyst 2900XL и 3500XL. В общем случае необходимо включать функцию Backbone Fast, если коммутируемый домен содержит эти коммутаторы помимо других коммутаторов Catalyst. При использовании Backbone Fast в средах строгой топологии с коммутаторами XL можно включать эту функцию там, где коммутатор XL является последним коммутатором на линии и подключен в двух местах только к ядру. Не следует использовать эту функцию, если коммутаторы XL подключены последовательно.

При использовании RSTP или IEEE 802.1w настройка BackboneFast не требуется, поскольку RSTP изначально содержит этот механизм и автоматически включает его. Дополнительные сведения о протоколе RSTP и стандарте IEEE 802.1w см. в статье Пример конфигурации: изменение режима STP с PVST+ на Rapid-PVST.

Конфигурация для CatOS

Для коммутаторов Catalyst серий 4000, 5000 и 6000, работающих под управлением CatOS, используйте следующие команды, чтобы включить функцию Backbone Fast на всех портах и проверить конфигурацию.

```
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)
```

Чтобы вывести статистику функции Backbone Fast, используйте следующие команды:

```
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)
```

Конфигурация для Cisco IOS

Для коммутаторов Catalyst, работающих под управлением программного обеспечения Cisco IOS, используйте следующие команды, чтобы включить функцию Backbone Fast на всех интерфейсах.

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

Чтобы проверить, включена ли функция Backbone Fast, и вывести статистические данные, используйте следующие команды:

```
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#
```

Дополнительные сведения

- Использование функции PortFast и других команд для устранения задержек соединения во время запуска рабочей станции
- Общие понятия и конфигурация функции Cisco Uplink Fast
- Усовершенствование протокола STP с помощью функций Loop Guard и BPDU Skew Detection
- Усиление функции защиты Portfast BPDU для протокола STP

- Пример конфигурации: изменение режима STP с PVST+ на Rapid-PVST
- Техническая поддержка протокола STP
- Поддержка коммутаторов
- Поддержка технологии коммутации локальных сетей
- Cisco Systems – техническая поддержка и документация