Устранение неполадок протокола STP на коммутаторах Catalyst с ПО Cisco IOS System

Содержание

Введение

В данном документе приведены инструкции по использованию ПО Cisco IOS® для устранения неполадок, связанных с протоколом связующего дерева (STP; Spanning-Tree Protocol). Существуют особые команды, которые применимы только к коммутаторам Catalyst 6500/6000, однако большинство принципов можно использовать в любых коммутаторах Cisco Catalyst с ПО Cisco IOS.

Существует три самые распространенные проблемы протокола STP:

- закольцовывание
- чрезмерный поток пакетов вследствие высокой скорости изменений топологии (TC; Topology Changes) STP
- проблемы, связанные с временем схождения

Поскольку в мостовом соединении отсутствует механизм отслеживания, передается ли определенный пакет несколько раз (например, время жизни (TTL) IP-пакета используется для сбрасывания трафика, циркулирующего в сети слишком долго), между устройствами, расположенными в одном домене 2 уровня, может существовать только один путь.

Назначением STP является преобразование избыточной физической топологии в древовидную топологию при помощи блокирования избыточных портов, основанном на STP-алгоритме. Закольцовывание (например, петля STP) возникает в случае отсутствия блокированных портов в топологии с резервированием и циклической передачи трафика в течение неопределенного времени.

Закольцовывание вызывает перегрузку каналов с низкой пропускной способностью по всей цепи, и если все каналы имеют одинаковую пропускную способность, то будут все перегружены. Перегрузка повлечет за собой потерю пакетов и приведет к выходу из строя сети в затронутом домене 2 уровня.

При лавинной маршрутизации результаты будут столь же ощутимы. Некоторые медленные каналы могут быть перегружены лавинным трафиком, что значительно понизит работоспособность пользователей и устройств или приведет к разрыву соединения.

Предварительные условия

Требования

Cisco рекомендует ознакомиться с содержанием следующих разделов:

- Типы связующих деревьев и их настройка. Дополнительные сведения см. в документе Настройка STP и IEEE 802.1s MST.
- Функции связующих деревьев и их настройка. Дополнительные сведения см. в документе Настройка функций STP.

Используемые компоненты

Сведения, содержащиеся в данном документе, касаются следующих версий программного и аппаратного обеспечения.

- Catalyst 6500 с модулем управления Supervisor Engine 2
- Программное обеспечение Cisco IOS версии 12.1(13)E

Данные для документа были получены в специально созданных лабораторных условиях. Все устройства, используемые в этом документе, были запущены в исходной (заданной по умолчанию) конфигурации. Если ваша сеть работает в реальных условиях, убедитесь, что вы понимаете потенциальное воздействие каждой команды.

Условные обозначения

Дополнительную информацию об используемых в документе обозначениях см. в документе Условные обозначения, используемые в технической документации Cisco.

Причины сбоя STP

Протокол STP накладывает некоторые требования на рабочую среду. Требования, относящиеся к этому документу:

- Каждый канал между двумя мостами – двунаправленный. Например, если A непосредственно подключено к B, A будет принимать пакеты от B, а B будет принимать пакеты от A до тех пор, пока между ними установлено соединение.
- Каждый мост, поддерживающий протокол STP, имеет возможность постоянно принимать, обрабатывать и отправлять блоки данных протокола моста (BPDU; Bridge Protocol Data Units) STP, так же известные как STP-пакеты.

Не смотря на то, что приведенные выше требования являются логическими и очевидными, существуют ситуации, когда они не удовлетворяются. Большинство этих ситуаций включает в себя аппаратные проблемы, однако, ошибки программного обеспечения могут также приводить к отказам STP. Отказы оборудования, неверная конфигурация или плохая укладка кабеля являются причиной большинства ошибок протокола STP, в то время как ошибки ПО несильно влияют на работу протокола. Сбои в работе протокола STP могут произойти из-за лишних дополнительных подключений между коммутаторами. Такие подключения могут вывести из строя VLAN. Для решения этой проблемы необходимо удалить все нежелательные подключения между коммутаторами.

Когда одно из этих требований не выполняется, один или более мостов не смогут получать и обрабатывать пакеты BPDU. Это означает, что мост (или мосты) не сможет обнаружить топологию сети. Без данных о правильной топологии коммутатор не сможет блокировать петли. Таким образом, лавинный трафик будет циркулировать в циклической топологии, снижая пропускную способность, что приведет к выходу сети из строя.

Причины, по которым коммутаторы не принимают BPDU-пакеты: плохой приемопередатчик или конвертеры интерфейса Gigabit (GBIC), неисправный кабель, а также аппаратные ошибки порта, линейной платы или управляющего модуля. Самой распространенной причиной сбоя протокола STP является однонаправленное соединение между мостами. В таком случае, один мост отправляет BPDU-пакеты, но нисходящий их не получает. Обработка протоколом STP также может быть прервана из-за перегрузки ЦП (более 99 %), потому что коммутатор не сможет обрабатывать полученные BPDU-пакеты. Повреждение BPDU-пакетов во время передачи от одного моста другому также может привести к ненормальному поведению STP.

Независимо от наличия закольцовывания, когда нет заблокированных портов, существуют ситуации, при которых передача определенных пакетов через блокирующие порты выполняется неправильно. В большинстве случаев это вызвано программным ошибками. Такое поведение может вызвать "медленные петли". Это значит, что несколько пакетов зациклены, но основная часть трафика все еще передается по сети, так как каналы, скорее всего не перегружены.

В следующих разделах представлены инструкции по устранению большинства основных проблем, связанных с протоколом STP.

Устранение неполадок, связанных с закольцовыванием

Происхождение (причины) и эффекты закольцовывания весьма разные. Из-за большого количества причин, вызывающих проблемы в работе протокола STP, в данном документе приведены только общие инструкции по устранению неполадок, связанных с закольцовыванием.

Для устранения неполадок необходима следующая информация:

- Фактическая схема топологии с детальным описанием всех коммутаторов и мостов
- Их соответствующие (взаимосвязанные) номера портов
- Подробные сведения о конфигурации STP (указание корневого и резервного корневого коммутатора, каналы с настройками стоимости и приоритета не по умолчанию, а также размещение блокирующих портов)

Обычно для устранения неполадок необходимы следующие шаги (в зависимости от ситуации, некоторые шаги могут не понадобиться):

1. Идентификация петли.

   При появлении закольцовывания в сети появляются следующие признаки:

   - Потеря связи к, от и через затронутые области
   - Повышенная нагрузка на ЦП маршрутизаторов, подключенных к затронутым сегментам сети или VLAN, что может привести к возникновению различных симптомов, например, переброскам соседа по протоколу маршрутизации или переброскам активного маршрутизатора по протоколу маршрутизации в режиме "горячего" резерва (HSRP).
   - Высокий коэффициент использования канала (часто 100%)
   - Высокая нагрузка на объединительную плату (по сравнению с обычной нагрузкой)
   - Сообщения системного журнала, которые указывают на зацикливание пакета в сети (например, сообщения дублированного IP адреса HSRP)
   - Сообщение системного журнала, которые указывает на постоянное изменение адресов или появление сообщений о переброске MAC-адресов
   - Возросшее количество отброшенных исходящих пакетов во многих интерфейсах

   Примечание. По отдельности, каждый из этих признаков может указывать на разные проблемы (или проявляться при их отсутствии). Однако, когда одновременно наблюдается несколько таких признаков, скорее всего в сети образовалась петля передачи данных.

   Примечание. Самый быстрый способ убедиться в этом – проверить загруженность трафиком объединительной платы коммутатора:

```
cat# show catalyst6000 traffic-meter

 traffic meter =   13%  Never cleared
         peak =   14%  reached at 12:08:57 CET Fri Oct 4 2002
```

   Примечание. В настоящее время данная команда не поддерживается коммутатором Catalyst 4000 с ПО Cisco IOS.

Если текущий уровень трафика существенно превышает норму или базовый уровень не известен, необходимо проверить, был ли пиковый уровень достигнут недавно и близок ли он к текущему уровню трафика. Например, пиковый уровень трафика составляет 15% и был достигнут 2 минуты назад, а уровень текущего трафика – 14%. Это означает, что коммутатор работает в необычно высокой загрузке.

Если уровень нагрузки по потоку сообщений является нормальным, возможно, это значит, что либо петля отсутствует, либо устройство не затронуто этой петлей. Однако оно может находиться в "медленной петле".

2. Обнаружение топологии (области действия) петли.

Если причиной выхода из строя сети является закольцовывание, то в первую очередь необходимо остановить зацикливание и восстановить работу сети. Чтобы остановить зацикливание, необходимо выяснить, какие порты находятся в петле: найти порты с наибольшей загрузкой канала (количеством пакетов в секунду). Команда **show interface** ПО Cisco IOS отображает загрузку каждого интерфейса.

Для отображения только имени интерфейса и сведений о загрузке (для быстрого анализа) можно использовать фильтр вывода регулярных выражений ПО Cisco IOS. Введите команду **show interface | include line|\/sec** для отображения количества пакетов, передаваемых в секунду, и имени интерфейса:

```
cat# show interface | include line|\/sec

GigabitEthernet2/1 is up, line protocol is down
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/2 is up, line protocol is down
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/3 is up, line protocol is up
 5 minute input rate 99765230 bits/sec, 24912 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/4 is up, line protocol is up
 5 minute input rate 1000 bits/sec, 27 packets/sec
  5 minute output rate 101002134 bits/sec, 25043 packets/sec
GigabitEthernet2/5 is administratively down, line protocol is down
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/6 is administratively down, line protocol is down
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/7 is up, line protocol is down
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
GigabitEthernet2/8 is up, line protocol is up
 5 minute input rate 2000 bits/sec, 41 packets/sec
 5 minute output rate 99552940 bits/sec, 24892 packets/sec
```

Особое внимание необходимо уделить интерфейсам с наибольшей загрузкой канала. В данном примере интерфейсы g2/3, g2/4 и g2/8 возможно попали в петлю.

3. Остановка зацикливания.

Чтобы остановить зацикливание, необходимо отключить или отсоединить соответствующие порты. Очень важно не только остановить зацикливание, а также найти и устранить причину его возникновения. Остановить зацикливание достаточно несложно.

Примечание. Для того, чтобы устранить причину проблемы, необходимо не отключать все порты одновременно, а отключать их поочередно. Как правило, целесообразнее выключать порты в точке агрегирования, которая входит в петлю, такой как распределяющий или основной коммутатор. Отключение сразу всех портов, включение и подключение их один за другим может не сработать, поскольку цикл будет остановлен и не начнется сразу после подключения виновного порта. Таким образом, будет сложно определить, на каком порту произошла ошибка.

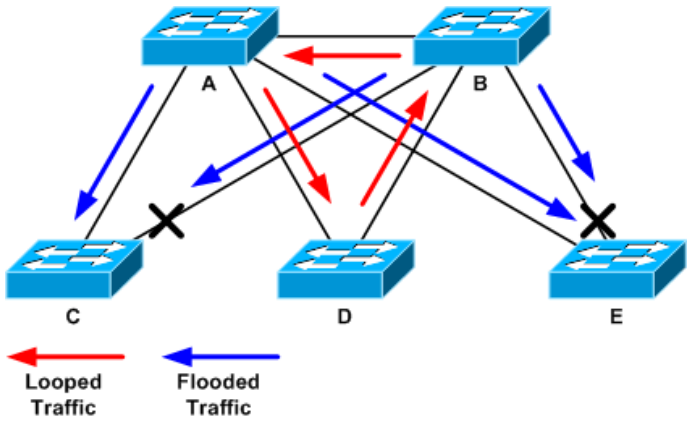
Примечание. Рекомендуется собрать все необходимые сведения до перезагрузки коммутатора в целях остановки зацикливания. Иначе будет сложно выяснить исходную причину проблемы.

После отключения каждого порта необходимо проверить, восстановлен ли нормальный уровень загрузки объединительной платы.

Примечание. Необходимо помнить, что обычно некоторые порты не создают петлю, а создают лавинный трафик из прибывающего по петле. Отключение таких перегружающих портов лишь немного снизит загрузку объединительной платы, но не остановит зацикливание.

В представленном ниже примере топологии петля установилась между коммутаторами A, B и D, поэтому каналы AB, AD и BD

поддерживают ее. Если отключить любой из этих каналов, можно остановить зацикливание. Каналы AC, AE, BC и BE лишь распространяют потоки трафика, поступающего с петли.



После отключения поддерживающего петлю порта загрузка объединительной платы снизится до нормального значения. Важно заметить, отключение какого порта приведет к нормализации уровня загрузки объединительной платы (и других портов).

На данном этапе зацикливание остановлено и улучшена работа сети. Но поскольку причина возникновения петли не устранена, неполадки могут возникнуть вновь.

4. Поиск и устранение причины возникновения петли.

После остановки зацикливания необходимо выяснить причину его возникновения. Обычно это самая трудная часть процесса, так как причины могут быть разными. Также сложно формализовать точную процедуру, которая подойдет для любого случая. Тем не менее, есть несколько основных инструкций:

- Исследовать схему топологии, чтобы найти дублируемый маршрут. Он включает в себя поддерживающий петлю порт, определенный на предыдущем шаге, и возвращается на тот же коммутатор (по пути пакетов, попавших в петлю). В предыдущем примере топологии, такой путь – AD-DB-BA.

- Для всех коммутаторов, расположенных на дублированном маршруте, выполняется проверка на наличие следующих проблем:

1. Знает ли коммутатор правильный корень STP?

   Все коммутаторы в сети L2 должны достичь согласования при помощи общего корня STP. Отображение мостами разных идентификаторов корня STP определенной VLAN или экземпляра STP является явным признаком проблемы. Для отображения идентификатора корневого моста заданной VLAN необходимо выполнить команду **show spanning-tree vlan *vlan-id***:

```
cat# show spanning-tree vlan 333

MST03
 Spanning tree enabled protocol mstp
 Root ID     Priority    32771
             Address     0050.14bb.6000
             Cost        20000
             Port        136 (GigabitEthernet3/8)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

 Bridge ID   Priority    32771  (priority 32768 sys-id-ext 3)
             Address     00d0.003f.8800
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- --------------------------
Gi3/8            Root FWD 20000     128.136  P2p
Po1              Desg FWD 20000     128.833  P2p
```

   Номер виртуальной локальной сети можно найти из порта, так как порты, попавшие в петлю, были определены на предыдущих этапах. Если данные порты являются магистральными, как правило, отображаются все VLAN магистрали. Если это не тот случай (например, если петля возникла в одной VLAN), тогда можно использовать команду **show interfaces | include L2|line|broadcast** (только на модулях управления Supervisor Engine 2 и более поздних версиях коммутаторов Catalyst 6500/6000, так как Supervisor 1 не поддерживает отображение статистики коммутации для отдельной VLAN). Необходимо проверять только интерфейсы VLAN. Наибольшая вероятность появления петли существует в VLAN с наибольшим количеством коммутируемых пакетов:

```
cat# show int | include L2|line|broadcast

Vlan1 is up, line protocol is up
 L2 Switched: ucast: 653704527 pkt, 124614363025 bytes - mcast:
        23036247 pkt, 1748707536 bytes
    Received 23201637 broadcasts, 0 runts, 0 giants, 0 throttles
Vlan10 is up, line protocol is up
 L2 Switched: ucast: 2510912 pkt, 137067402 bytes - mcast:
        41608705 pkt, 1931758378 bytes
    Received 1321246 broadcasts, 0 runts, 0 giants, 0 throttles
Vlan11 is up, line protocol is up
 L2 Switched: ucast: 73125 pkt, 2242976 bytes - mcast:
        3191097 pkt, 173652249 bytes
    Received 1440503 broadcasts, 0 runts, 0 giants, 0 throttles
Vlan100 is up, line protocol is up
 L2 Switched: ucast: 458110 pkt, 21858256 bytes - mcast:
        64534391 pkt, 2977052824 bytes
    Received 1176671 broadcasts, 0 runts, 0 giants, 0 throttles
Vlan101 is up, line protocol is up
 L2 Switched: ucast: 70649 pkt, 2124024 bytes - mcast:
        2175964 pkt, 108413700 bytes
    Received 1104890 broadcasts, 0 runts, 0 giants, 0 throttles
```

В примере, приведенном выше, VLAN 1 отвечает за наибольшее число широковещательных передач и коммутируемого трафика L2.

2. Правильно ли определен корневой порт?

   Корневой порт должен иметь наименьшую нагрузку на корневой мост (иногда один путь короче с точки зрения переходов, но длиннее из-за нагрузки, поскольку на низкоскоростных портах нагрузка выше).

   Для определения корневого порта в заданной VLAN необходимо использовать команду **show spanning-tree vlan *vlan***:

```
cat# show spanning-tree vlan 333

MST03
 Spanning tree enabled protocol mstp
 Root ID    Priority    32771
            Address     0050.14bb.6000
            Cost        20000
            Port        136 (GigabitEthernet3/8)
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

 Bridge ID  Priority    32771  (priority 32768 sys-id-ext 3)
            Address     00d0.003f.8800
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role Sts Cost      Prio.Nbr Status
---------------- ---- --- --------- -------- ------------------------
Gi3/8            Root FWD 20000     128.136  P2p
Po1              Desg FWD 20000     128.833  P2p
```

3. Регулярно ли корневой порт и предположительно блокирующие порты принимают пакеты BPDU?

   Пакеты BPDU отправляются корневым мостом через каждый `hello` интервал (по умолчанию 2 секунды). Некорневые мосты принимают, обрабатывают, изменяют и передают полученные с корневого моста пакеты BPDU.

   Чтобы просмотреть, принимает ли мост пакеты BPDU, необходимо выполнить команду **show spanning-tree interface *interface* detail**:

```
cat# show spanning-tree interface g3/2 detail

Port 130 (GigabitEthernet3/2) of MST00 is backup blocking
  Port path cost 20000, Port priority 128, Port Identifier 128.130.
  Designated root has priority 0, address 0007.4f1c.e847
  Designated bridge has priority 32768, address 00d0.003f.8800
  Designated port id is 128.129, designated path cost 2000019
  Timers: message age 4, forward delay 0, hold 0
  Number of transitions to forwarding state: 0
  Link type is point-to-point by default, Internal
  Loop guard is enabled by default on the port
  BPDU: sent 3, received 53
```

```
cat# show spanning-tree interface g3/2 detail

Port 130 (GigabitEthernet3/2) of MST00 is backup blocking
  Port path cost 20000, Port priority 128, Port Identifier 128.130.
  Designated root has priority 0, address 0007.4f1c.e847
  Designated bridge has priority 32768, address 00d0.003f.8800
  Designated port id is 128.129, designated path cost 2000019
  Timers: message age 5, forward delay 0, hold 0
  Number of transitions to forwarding state: 0
  Link type is point-to-point by default, Internal
  Loop guard is enabled by default on the port
  BPDU: sent 3, received 54
```

Примечание. Между двумя результатами команды был получен один пакет BPDU (счетчик поменял значение с 53 до 54).

Отображенные счетчики в действительности являются счетчиками, которые процесс STP обслуживает самостоятельно. Это означает, что в случае возрастания счетчиков приема, пакет BPDU был получен не только физическим портом, но и процессом STP.

Если значение BPDU-счетчика `received` порта, который считается резервным, не увеличивается, необходимо проверить, принимает ли порт многоадресные рассылки вообще (отправить пакеты BPDU как многоадресную рассылку). Необходимо выполнить команду **show interface *interface* counters**:

```
cat# show interface g3/2 counters

Port            InOctets   InUcastPkts   InMcastPkts   InBcastPkts
Gi3/2           14873036             2         89387             0

Port           OutOctets  OutUcastPkts  OutMcastPkts  OutBcastPkts
Gi3/2          114365997         83776        732086            19

cat# show interface g3/2 counters

Port            InOctets   InUcastPkts   InMcastPkts   InBcastPkts
Gi3/2           14873677             2         89391             0

Port           OutOctets  OutUcastPkts  OutMcastPkts  OutBcastPkts
Gi3/2          114366106         83776        732087            19
```

(Краткое описание ролей портов STP см. в подразделе Краткое описание ролей портов протокола STP раздела Оптимизация протокола связующего дерева с помощью функций Loop Guard (контроль петли) и BPDU Skew Detection (обнаружение искажения протокольной информационной единицы моста).)

Если не принято ни одного пакета BPDU, необходимо проверить наличие ошибок порта. Для этого необходимо выполнить команду **show interface *interface* counters errors**:

```
cat# show interface g4/3 counters errors

Port     Align-Err    FCS-Err   Xmit-Err    Rcv-Err UnderSize OutDiscards
Gi4/3            0          0          0          0         0           0

Port  Single-Col Multi-Col  Late-Col Excess-Col Carri-Sen   Runts    Giants
Gi4/3          0         0         0          0         0        0         0
```

Возможно, что BPDU принимаются физическим портом, но все же не доходят до процесса STP. Если команды, используемые в предыдущих примерах, отображают, что порт принимает многоадресные рассылки и количество ошибок не увеличивается, необходимо проверить, не отбрасываются ли пакеты BPDU на уровне процесса STP. Необходимо выполнить команду **remote command switch test spanning-tree process-stats** на коммутаторе Catalyst 6500:

```
cat# remote command switch test spanning-tree process-stats

------------------TX STATS------------------
transmission rate/sec     = 2
paks transmitted          = 5011226
paks transmitted (opt)    = 0
opt chunk alloc failures  = 0
max opt chunk allocated   = 0
------------------RX STATS------------------
 receive rate/sec          = 1
paks received at stp isr  = 3947627
paks queued at stp isr    = 3947627
```

```
 paks dropped at stp isr   = 0
drop rate/sec             = 0
paks dequeued at stp proc = 3947627
paks waiting in queue     = 0
queue depth               = 7(max) 12288(total)
--------------PROCESSING STATS---------------
queue wait time (in ms)   = 0(avg) 540(max)
processing time (in ms)   = 0(avg) 4(max)
proc switch count         = 100
add vlan ports            = 20
time since last clearing  = 2087269 sec
```

Данная команда отображает статистику процесса STP. Важно убедиться, что значение счетчика сбросов не увеличивается и число полученных пакетов возрастает.

Если количество полученных пакетов не увеличивается, а физический порт принимает многоадресные рассылки, необходимо проверить, принимаются ли пакеты внутренним интерфейсом коммутатора (интерфейс ЦП). Необходимо выполнить команду **remote command switch show ibc | i rx_input** на коммутаторе Catalyst 6500/6000:

```
cat# remote command switch show ibc | i rx_input

rx_inputs=5626468, rx_cumbytes=859971138

cat# remote command switch show ibc | i rx_input

rx_inputs=5626471, rx_cumbytes=859971539
```

В данном примере между результатами выполнения команды внутренний порт принял 23 пакета.

Примечание. Эти 23 пакета являются не только пакетами BPDU, это глобальный счетчик всех пакетов, принятых внутренним портом.

Если признаки того, что на локальном коммутаторе или порте происходит удаление пакетов BPDU, отсутствуют, необходимо проверить, что коммутатор на другом конце канала отправляет BPDU.

4. Регулярно ли отправляются пакеты BPDU на назначенные некорневые порты?

   Если порт, в соответствии со своей ролью, отправляет пакеты BPDU, но соседнее устройство их не получает, необходимо проверить, действительно ли пакеты были отправлены. Для этого необходимо выполнить команду **show spanning-tree interface *interface* detail**:

```
cat# show spanning-tree interface g3/1 detail

Port 129 (GigabitEthernet3/1) of MST00 is designated forwarding
  Port path cost 20000, Port priority 128, Port Identifier 128.129.
  Designated root has priority 0, address 0007.4f1c.e847
  Designated bridge has priority 32768, address 00d0.003f.8800
  Designated port id is 128.129, designated path cost 2000019
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 0
  Link type is point-to-point by default, Internal
  Loop guard is enabled by default on the port
  BPDU: sent 1774, received 1

cat# show spanning-tree interface g3/1 detail

Port 129 (GigabitEthernet3/1) of MST00 is designated forwarding
  Port path cost 20000, Port priority 128, Port Identifier 128.129.
  Designated root has priority 0, address 0007.4f1c.e847
  Designated bridge has priority 32768, address 00d0.003f.8800
  Designated port id is 128.129, designated path cost 2000019
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 0
  Link type is point-to-point by default, Internal
  Loop guard is enabled by default on the port
  BPDU: sent 1776, received 1
```

В данном примере между двумя результатами выполнения команды были переданы 2 пакета BPDU.

Примечание. Процесс STP использует счетчик `BPDU: sent`. То есть счетчик указывает, что пакеты BPDU, были отправлены на физический порт для передачи. Необходимо проверить, увеличивается ли значение счетчика порта по передаче многоадресных пакетов. Необходимо выполнить команду **show interface *interface* counters**. Это поможет

определить, передаются ли пакеты BPDU:

```
cat# show interface g3/1 counters

Port            InOctets   InUcastPkts   InMcastPkts   InBcastPkts
Gi3/1          127985312         83776        812319            19

Port           OutOctets  OutUcastPkts  OutMcastPkts  OutBcastPkts
Gi3/1          131825915          3442        872342           386

cat# show interface g3/1 counters

Port            InOctets   InUcastPkts   InMcastPkts   InBcastPkts
Gi3/1          127985312         83776        812319            19

Port           OutOctets  OutUcastPkts  OutMcastPkts  OutBcastPkts
Gi3/1          131826447          3442        872346           386
```

Выполнение данных шагов необходимо для поиска коммутатора или канала, которые не принимают, не отправляют или не обрабатывают пакеты BPDU.

Маловероятно, но возможно, что протокол STP рассчитывает правильное состояние порта, но из-за проблемы уровня управления, не смог установить это состояние на устройстве передачи. Существует возможность появления петли, если предполагаемые блокирующий порт не заблокирован на аппаратном уровне. Если есть подозрение на существование в сети данной проблемы, необходимо связаться со службой Технической поддержки Cisco для получения дополнительной помощи.

5. Восстановление резервирования.

   Как только канал или устройство, образующие петлю, найдены, их необходимо отключить от сети или исправить проблему (например, заменить оптоволокно или GBIC). Резервные каналы, отключенные на шаге 3, необходимо восстановить.

   Очень важно производить с каналом или устройством, послужившим причиной возникновения петли, как можно меньше действий, потому что большинство приводящих к этому условий очень кратковременны, периодичны и нестабильны. Это значит, что если в течение или после устранения неисправности условие было сброшено, таким образом может пройти некоторое время прежде чем оно возникнет вновь. Но существует вероятность, что условие не возникнет. Необходимо предпринять все меры для сохранения условия таким образом, чтобы в будущем его можно было исследовать в службе Технической поддержки Cisco. Необходимо собрать все необходимые сведения об условии прежде, чем перезапустить коммутаторы. Если условие больше не существует, то определить главную причину возникновения петли часто оказывается невозможным. Очень важно обнаружить устройство или канал, порождающие петлю. Однако необходимо удостовериться, что какая-либо другая неисправность того же типа не послужит причиной повторного возникновения петли. Дополнительные сведения см. в разделе Защита сети от закольцовывания.

Устранение неполадок лавинной маршрутизации, вызванных изменениями топологии

Механизм изменения топологии (TC) предназначен для исправления таблицы передачи L2 после изменения топологии передачи. Это необходимо для предотвращения разрыва соединения, поскольку после изменения топологии некоторые MAC-адреса, ранее доступные через определенные порты, могут стать недоступными через другие порты. TC сокращает время устаревания таблицы передачи на всех коммутаторах VLAN, в которых происходит TC, таким образом, что не переназначенный адрес устаревает и происходит лавина, чтобы позволить пакетам наверняка достичь MAC-адреса назначения.

TC происходит благодаря изменению состояния STP порта из или в состояние `forwarding` протокола STP. Даже если определенный MAC-адрес назначения устарел, лавинная маршрутизация не должна продолжаться долго. Адрес будет переназначен первым пакетом, принятым с хоста, MAC-адрес которого устарел. Проблема может возникнуть в случае многократного частого изменения топологии. Таблицы пересылки коммутаторов постоянно будут быстро устаревать, поэтому и лавинные потоки будут практически постоянными.

Примечание. Благодаря Rapid STP и Multiple STP (IEEE 802.1w и IEEE 802.1s), изменение топологии происходит благодаря изменению состояния порта на `forwarding`, а также изменению роли с `designated` на `root`. Rapid STP мгновенно очищает таблицу пересылки данных 2 уровня, в отличие от 802.1d, что сокращает время старения. Немедленная очистка таблицы пересылки быстрее восстанавливает возможность соединения, но вызывает большее переполнение.

TC должно быть редким событием в хорошо настроенной сети. Когда канал порта коммутатора включается или выключается, через определенное время происходит TC, как только состояние STP порта меняется на `forwarding` или наоборот.. Постоянные переназначения порта могут вызвать повторяющиеся TC и лавинную маршрутизацию.

Порты с включенной функцией STP portfast позволяют избежать изменения топологии во время перехода в состояние `forwarding` и обратно. Рекомендуется использовать конфигурацию portfast на всех портах оконечных устройств (принтерах, ПК и серверах), чтобы ограничить количество TC. Дополнительные сведения об изменениях топологии см. в разделе Общие сведения об изменении топологии протокола STP.

Если в сети происходят повторяющиеся TC, необходимо определить их источник и принять меры по их устранению, чтобы свести лавинные передачи к минимуму.

С 802.1d информация STP о событии TC распространяется через мосты с помощью уведомления об изменении топологии (TCN), являющимся специальным типом BPDU. По портам, принимающим пакеты TCN BPDU, можно найти устройство, порождающие изменение топологии.

Проверка того, что лавинная маршрутизация вызвана изменением топологии STP.

Обычно лавинная маршрутизация определяется по признаку замедления работы, отбрасыванию пакетов в каналах, которые не должны бы быть перегруженными, и по отображению анализатором пакетов большого количества одноадресных пакетов с одинаковым местом назначения, не находящимся в локальном сегменте.

Дополнительные сведения об одноадресной лавинной передаче см. в разделе Односторонняя лавинная маршрутизация в коммутируемых кампусных сетях.

На коммутаторах Catalyst 6500/6000 с ПО Cisco IOS можно проверить счетчик модуля переадресации (только на модуле Supervisor 2), чтобы приблизительно подсчитать количество лавинных потоков. Для этого необходимо выполнить команду **remote command switch show earl statistics | i MISS_DA|ST_FR**:

```
cat# remote command switch show earl statistics | i MISS_DA|ST_FR

        ST_MISS_DA      =       18                      530308834
        ST_FRMS         =       97                      969084354

cat# remote command switch show earl statistics | i MISS_DA|ST_FR

        ST_MISS_DA      =       4                       530308838
        ST_FRMS         =       23                      969084377
```

В приведенном выше примере в первом столбце отображается изменение с момента последнего выполнения данной команды, а во втором столбце отображаются суммарное значение с момента последней перезагрузки. В первой строке отображено количество лавинно перенаправленных кадров, а во второй – количество обработанных. Если обе величины близки по значению или первая величина увеличивается с высокой скоростью, возможно, коммутатор осуществляет лавинную маршрутизацию трафика. Это нужно использовать в совокупности с другими способами определения лавинной передачи, так как счетчики не подробные). Есть один счетчик на коммутатор, но не на порт или VLAN. Наличие некоторого количества лавинных пакетов вполне обычно, так как коммутатор всегда будет выполнять лавинную пересылку, если MAC-адреса назначения нет в таблице пересылки. Например, это будет происходить, когда коммутатор получит пакет с адресом направления, который ему еще не знаком.

Определение источника изменений топологии

Если известен номер VLAN, в которой возникает лавинная передача, необходимо проверить счетчики STP, чтобы проконтролировать, остается ли значение TC высоким или постоянно увеличивается. Для этого необходимо выполнить команду **show spanning-tree vlan *vlan-id* detail** (в данном примере используется VLAN 1):

```
cat# show spanning-tree vlan 1 detail

VLAN0001 is executing the ieee compatible Spanning Tree protocol
 Bridge Identifier has priority 32768, sysid 1, address 0007.0e8f.04c0
 Configured hello time 2, max age 20, forward delay 15
 Current root has priority 0, address 0007.4f1c.e847
 Root port is 65 (GigabitEthernet2/1), cost of root path is 119
 Topology change flag not set, detected flag not set
 Number of topology changes 1 last change occurred 00:00:35 ago
         from GigabitEthernet1/1
 Times:  hold 1, topology change 35, notification 2
```

```
        hello 2, max age 20, forward delay 15
Timers: hello 0, topology change 0, notification 0, aging 300
```

Если номер VLAN неизвестен, можно использовать анализатор пакетов или проверить счетчики TC для всех VLAN.

Действия по предотвращению чрезмерно частой смены топологии (TC).

Чтобы отслеживать, возрастает ли количество изменений топологии, необходимо контролировать счетчик `number of topology changes`. Затем необходимо перейти к мосту, подключенному к порту, на котором было зафиксировано последнее изменение топологии (в предыдущем примере это порт GigabitEthernet1/1), и проверить, откуда оно было принято. Этот процесс необходимо повторять, пока не будет найден порт конечной станции с отключенным STP portfast или постоянно переключающийся канал, который необходимо исправить. Процедуру необходимо повторить, если изменения топологии поступают и из других источников. Если канал принадлежит конечному узлу, необходимо настроить функцию portfast, чтобы предотвратить появление TC.

Примечание. В реализации STP ПО Cisco IOS значение счетчика TC увеличивается только при получении TCN BPDU с порта VLAN. При обычной конфигурации BPDU с установленным флагом TC значение счетчика TC не увеличивается. Это означает, что если TC предположительно является причиной лавинной маршрутизации, то лучше всего начать отслеживать источники TC с моста корневого STP в VLAN. Таким образом можно получить наиболее точные данные относительно источника и количества TC.

Устранение неполадок, связанных с временем схождения

Бывают ситуации, когда действительное функционирование STP не соответствует ожидаемому. Существуют две наиболее распространенные причины:

- Схождение или расхождение STP занимает больше времени, чем планировалось.
- Результирующая топология отличается от ожидаемой.

В большинстве случаев к этому приводят следующие причины:

- Несоответствие между реальной и документированной топологиями
- Неверная конфигурация, например, несовместимая конфигурация таймеров STP, чрезмерно большой диаметр STP или неверная конфигурация portfast
- Перегрузка ЦП коммутатора в процессе схождения или расхождения
- Программная ошибка

Как уже было сказано, из-за большого количества причин, вызывающих проблемы в работе протокола STP, в данном документе приведены только общие инструкции по устранению неполадок.

Чтобы определить причину слишком долгого времени схождения, необходимо проследить последовательность событий STP, чтобы выяснить, что произошло и в каком порядке. Так как реализация STP в Cisco IOS не предусматривает отдельной регистрации (кроме как конкретных событий, таких как несогласованность порта), можно использовать возможности отладки Cisco IOS STP, чтобы получить представление о том, что происходит.

В Catalyst 6500/6000 под управлением ПО Cisco IOS обработка STP осуществляется в процессоре коммутатора (SP) (или в модуле Supervisor), поэтому в SP функции отладки должны быть включены. Для мостовых групп ПО Cisco IOS обработка выполняется в процессоре маршрутизатора (RP), поэтому необходимо активировать отладку на RP (MSFC).

Команды отладки STP

Многие команды **debug** STP предназначены для разработчиков. Эти команды не предоставляют никаких выходных данных, понятных

при отсутствии детальных знаний о реализации STP в ПО Cisco IOS. Некоторые команды отладки предоставляют выходные данные, которые можно сразу прочитать, например, изменение состояния порта и роли, события (например, TC), а также содержимое переданных и полученных пакетов BPDU. В данном разделе приводится только краткое описание наиболее часто используемых команд отладки.

Примечание. Необходимо минимизировать необходимость использования команд отладки **debug**. Если не требуется отладка в реальном времени, выходные данные лучше записать в журнале, а не выводить на консоль. Слишком много команд отладки могут привести к перегрузке ЦП и сбоям в работе коммутатора. Чтобы перенаправить выходные данные отладки в журнал вместо вывода на консоль или в сеанс Telnet, необходимо выполнить команды **logging console informational** и **no logging monitor** в режиме глобальной конфигурации.

Чтобы просмотреть журнал общих событий, необходимо выполнить команду **debug spanning-tree event** для связующего дерева VLAN (PVST) и Rapid-PVST. Это первая отладка, которая дает представление о том, что происходит с STP.

В режиме Multiple Spanning-Tree (MST) команда **debug spanning-tree event** не работает. Однако, команда **debug spanning-tree mstp roles** позволяет контролировать изменение роли порта.

Чтобы проследить изменения состояния STP порта, необходимо выполнить команду **debug spanning-tree switch state** вместе с командой **debug pm vp**:

```
cat-sp# debug spanning-tree switch state

Spanning Tree Port state changes debugging is on

cat-sp# debug pm vp

Virtual port events debugging is on
Nov 19 14:03:37: SP:     pm_vp 3/1(333): during state forwarding, got event 4(remove)
Nov 19 14:03:37: SP: @@@ pm_vp 3/1(333):
 forwarding -> notforwarding

port 3/1 (was forwarding) goes down in vlan 333

Nov 19 14:03:37: SP: *** vp_fwdchange: single: notfwd: 3/1(333)
Nov 19 14:03:37: SP: @@@ pm_vp 3/1(333): notforwarding -> present
Nov 19 14:03:37: SP: *** vp_linkchange: single: down: 3/1(333)
Nov 19 14:03:37: SP: @@@ pm_vp 3/1(333): present -> not_present
Nov 19 14:03:37: SP: *** vp_statechange: single: remove: 3/1(333)
Nov 19 14:03:37: SP:     pm_vp 3/2(333): during state notforwarding,
 got event 4(remove)
Nov 19 14:03:37: SP: @@@ pm_vp 3/2(333): notforwarding -> present
Nov 19 14:03:37: SP: *** vp_linkchange: single: down: 3/2(333)

Port 3/2 (was not forwarding) in vlan 333 goes down

Nov 19 14:03:37: SP: @@@ pm_vp 3/2(333): present -> not_present
Nov 19 14:03:37: SP: *** vp_statechange: single: remove: 3/2(333)

Nov 19 14:03:53: SP:     pm_vp 3/1(333): during state not_present,
 got event 0(add)
Nov 19 14:03:53: SP: @@@ pm_vp 3/1(333): not_present -> present
Nov 19 14:03:53: SP: *** vp_statechange: single: added: 3/1(333)
Nov 19 14:03:53: SP:     pm_vp 3/1(333): during state present,
 got event 8(linkup)
Nov 19 14:03:53: SP: @@@ pm_vp 3/1(333): present ->
 notforwarding
Nov 19 14:03:53: SP: STP SW: Gi3/1 new blocking req for 0 vlans
Nov 19 14:03:53: SP: *** vp_linkchange: single: up: 3/1(333)

Port 3/1 link goes up and blocking in vlan 333

Nov 19 14:03:53: SP:     pm_vp 3/2(333): during state not_present,
 got event 0(add)
Nov 19 14:03:53: SP: @@@ pm_vp 3/2(333): not_present -> present
Nov 19 14:03:53: SP: *** vp_statechange: single: added: 3/2(333)
Nov 19 14:03:53: SP:     pm_vp 3/2(333): during state present,
 got event 8(linkup)
Nov 19 14:03:53: SP: @@@ pm_vp 3/2(333): present ->
 notforwarding
Nov 19 14:03:53: SP: STP SW: Gi3/2 new blocking req for 0 vlans
Nov 19 14:03:53: SP: *** vp_linkchange: single: up: 3/2(333)

Port 3/2 goes up and blocking in vlan 333

Nov 19 14:04:08: SP: STP SW: Gi3/1 new learning req for 1 vlans
```

```
Nov 19 14:04:23: SP: STP SW: Gi3/1 new forwarding req for 0 vlans
Nov 19 14:04:23: SP: STP SW: Gi3/1 new forwarding req for 1 vlans
Nov 19 14:04:23: SP:     pm_vp 3/1(333): during state notforwarding,
 got event 14(forward_notnotify)
Nov 19 14:04:23: SP: @@@ pm_vp 3/1(333): notforwarding ->
 forwarding
Nov 19 14:04:23: SP: *** vp_list_fwdchange: forward: 3/1(333)

Port 3/1 goes via learning to forwarding in vlan 333
```

Чтобы разобраться, почему STP функционирует именно таким образом, необходимо просмотреть принятые и отправленные коммутатором пакеты BPDU:

```
cat-sp# debug spanning-tree bpdu receive

Spanning Tree BPDU Received debugging is on
Nov  6 11:44:27: SP: STP: VLAN1 rx BPDU: config protocol = ieee,
 packet from GigabitEthernet2/1  , linktype IEEE_SPANNING ,
 enctype 2, encsize 17
Nov  6 11:44:27: SP: STP: enc 01 80 C2 00 00 00 00 06 52 5F 0E 50 00 26 42 42 03
Nov  6 11:44:27: SP: STP: Data 00000000000000000074F1CE847000000013804800006525F0E4
 080100100140002000F00
Nov  6 11:44:27: SP: STP: VLAN1 Gi2/1:0000 00 00 00 000000074F1CE847 00000013
 80480006525F0E40 8010 0100 1400 0200 0F00
```

Данная отладка может использоваться в режимах PVST, Rapid-PVST и MST, но она не поддерживает декодирование содержимого пакетов BPDU. Таким образом, этот способ можно использовать для проверки получения пакетов BPDU.

Для просмотра содержимого BPDU-пакета в режимах PVST и Rapid-PVST необходимо выполнить команду **debug spanning-tree switch rx decode** вместе с командой **debug spanning-tree switch rx process**. А для просмотра содержимого BPDU в режиме MST необходимо выполнить команду **debug spanning-tree mstp bpdu-rx**:

```
cat-sp# debug spanning-tree switch rx decode

Spanning Tree Switch Shim decode received packets debugging is on

cat-sp# debug spanning-tree switch rx process

Spanning Tree Switch Shim process receive bpdu debugging is on

Nov  6 12:23:20: SP: STP SW: PROC RX: 0180.c200.0000<-0006.525f.0e50 type/len 0026
Nov  6 12:23:20: SP:     encap SAP linktype ieee-st vlan 1 len 52 on v1 Gi2/1
Nov  6 12:23:20: SP:     42 42 03 SPAN
Nov  6 12:23:20: SP:     CFG P:0000 V:00 T:00 F:00 R:0000 0007.4f1c.e847 00000013
Nov  6 12:23:20: SP:     B:8048 0006.525f.0e40 80.10 A:0100 M:1400 H:0200 F:0F00

Nov  6 12:23:22: SP: STP SW: PROC RX: 0180.c200.0000<-0006.525f.0e50 type/len 0026
Nov  6 12:23:22: SP:     encap SAP linktype ieee-st vlan 1 len 52 on v1 Gi2/1
Nov  6 12:23:22: SP:     42 42 03 SPAN
Nov  6 12:23:22: SP:     CFG P:0000 V:00 T:00 F:00 R:0000 0007.4f1c.e847 00000013
Nov  6 12:23:22: SP:     B:8048 0006.525f.0e40 80.10 A:0100 M:1400 H:0200 F:0F00
```

Для режима MST, подробную расшифровку BPDU можно включить с помощью команды **debug**:

```
cat-sp# debug spanning-tree mstp bpdu-rx

Multiple Spanning Tree Received BPDUs debugging is on
Nov 19 14:37:43: SP: MST:BPDU DUMP [rcvd_bpdu Gi3/2 Repeated]
Nov 19 14:37:43: SP: MST:   Proto:0 Version:3 Type:2 Role: DesgFlags[   F    ]
Nov 19 14:37:43: SP: MST:   Port_id:32897 cost:2000019
Nov 19 14:37:43: SP: MST:   root_id  :0007.4f1c.e847 Prio:0
Nov 19 14:37:43: SP: MST:   br_id    :00d0.003f.8800 Prio:32768
Nov 19 14:37:43: SP: MST:   age:2 max_age:20 hello:2 fwdelay:15
Nov 19 14:37:43: SP: MST:   V3_len:90 PathCost:30000 region:STATIC rev:1
Nov 19 14:37:43: SP: MST:   ist_m_id :0005.74
Nov 19 14:37:43: SP: MST:BPDU DUMP [rcvd_bpdu Gi3/2 Repeated]
Nov 19 14:37:43: SP: MST:   Proto:0 Version:3 Type:2 Role: DesgFlags[   F    ]
Nov 19 14:37:43: SP: MST:   Port_id:32897 cost:2000019
```

```
Nov 19 14:37:43: SP: MST:   root_id  :0007.4f1c.e847 Prio:0
Nov 19 14:37:43: SP: MST:   br_id    :00d0.003f.8800 Prio:32768
Nov 19 14:37:43: SP: MST:   age:2 max_age:20 hello:2 fwdelay:15
Nov 19 14:37:43: SP: MST:   V3_len:90 PathCost:30000 region:STATIC rev:1
Nov 19 14:37:43: SP: MST:   ist_m_id :0005.7428.1440 Prio:32768  Hops:18
 Num Mrec: 1
Nov 19 14:37:43: SP: MST: stci=3  Flags[ F    ] Hop:19 Role:Desg [Repeated]
Nov 19 14:37:43: SP: MST:         br_id:00d0.003f.8800 Prio:32771 Port_id:32897
 Cost:2000028.1440 Prio:32768  Hops:18  Num Mrec: 1
Nov 19 14:37:43: SP: MST: stci=3  Flags[ F    ] Hop:19 Role:Desg [Repeated]
Nov 19 14:37:43: SP: MST:         br_id:00d0.003f.8800 Prio:32771 Port_id:32897
 Cost:20000
```

Примечание. В ПО Cisco IOS 12.1.13E и более поздних версиях предусмотрена поддержка селективной отладки STP. Это означает, что можно производить отладку принятых и переданных пакетов BPDU отдельно для каждого порта или VLAN.

Чтобы ограничить количество выходных данных отладки для каждого интерфейса или VLAN, необходимо выполнить команды **debug condition vlan *номер VLAN*** или **debug condition interface *интерфейс***.

Защита сети от закольцовывания

Чтобы обеспечить возможности STP по обработке определенных типов сбоев, компанией Cisco разработаны несколько функций и усовершенствований, обеспечивающих защиту сети от закольцовывания.

Устранение неисправностей STP помогает выявить причину возникновения проблемы и устранить ее, в то время, как реализация данных усовершенствований – единственный способ обеспечения защиты сети от закольцовывания.

Способы защиты сети от закольцовывания:

1. Активация функции Unidirectional Link Detection (UDLD) (обнаружения однонаправленного соединения) на всех каналах "коммутатор-коммутатор". Дополнительные сведения о функции UDLD см. в разделе Общие сведения и настройка протокола обнаружения однонаправленных соединений.
2. Активация функции Loop Guard (защиты от петель) на всех коммутаторах. Дополнительные сведения о функции защиты от петель см. в разделе Усовершенствование протокола связующего дерева с помощью функций Loop Guard (защита от петель) и BPDU Skew Detection (обнаружение искажения BPDU).

   Функции UDLD и Loop Guard предотвращают причины возникновения закольцовывания. Неисправный канал, вместо порождения петли передачи (или все каналы, в зависимости от аппаратных ошибок) отключается или блокируется.

   Примечание. Хотя эти две функции в некоторой степени повторяют друг друга, у каждой есть уникальные возможности. Таким образом, одновременное использование обеих функций обеспечивает самый высокий уровень защиты. Подробный сравнительный анализ UDLD и Loop Guard см. в разделе Защита от петель и обнаружение однонаправленных каналов.

   Существуют различные мнения о том, какой режим UDLD использовать – агрессивный или обычный. Нужно заметить, что агрессивный UDLD не будет предоставлять более надежную защиту от петель в сравнении с нормальным. Агрессивный режим UDLD обнаруживает зависший порт (когда соединение установлено, но есть не связанное пропадание трафика). Недостатком дополнительных функциональных возможностей является то, что в агрессивном режиме UDLD возможно отключение канала при отсутствии в нем систематических ошибок. Часто путают изменение интервала `hello` UDLD и функцию агрессивного режима UDLD. Это неправильно. Таймеры могут быть изменены в обоих режимах UDLD.

   Примечание. В редких случаях, в агрессивном режиме могут быть отключены все порты восходящего канала, что полностью изолирует коммутатор от сети. Например, это может произойти при высоком уровне загрузки ЦП обоих коммутаторов восходящего потока и использовании агрессивного режима UDLD. Поэтому рекомендуется настроить параметры errordisable-timeout, если коммутатор не имеет внеполосного управления.
3. Включить функцию portfast на всех портах конечных станций.

   Активация функции portfast необходима для ограничения числа TC и, следовательно, лавинной маршрутизации, которая может повлиять на производительность сети. Эта команда используется только на портах конечных станций. В противном случае случайное возникновение петли топологии может стать причиной возникновения петли пакета данных и отключить коммутатор и нарушить работу сети.

   Внимание! Необходимо проявлять осторожность при использовании команды **no spanning-tree portfast**. Эта команда

только отключает все функции portfast для порта. Данная команда неявно активирует функцию portfast, если в режиме глобальной конфигурации задать команду **spanning-tree portfast default** и если порт не является магистральным. Если функция portfast не настроена глобально, команда **no spanning-tree portfast** эквивалентна команде **spanning-tree portfast disable**.

4. Установить на обоих концах каналов EtherChannel режим `desirable` (где поддерживается) и параметр `non-silent`.

   В режиме `Desirable` используется протокол агрегации портов (PAgP), обеспечивающий оперативное согласование между одноранговыми узлами. Это обеспечивает дополнительную степень защиты от возникновения петель, особенно во время изменения конфигурации канала (например, соединение подключается или отключается от канала, а также обнаружение сбоев соединений). Имеется встроенная функция Channel Misconfiguration Guard (средство защиты от неправильной конфигурации канала), включенная по умолчанию и предотвращающая закольцовывание в случае неправильной конфигурации канала и при других условиях. Дополнительные сведения об этой функции см. в разделе Общие сведения об обнаружении несогласованности EtherChannel.

5. Не отключать автосогласование (если поддерживается) в каналах "коммутатор-коммутатор".

   Механизм автосогласования позволяет удаленно передавать сведения об ошибке, что является самым быстрым способом обнаружения ошибки на дальнем конце. Если на дальнем конце обнаружена ошибка, локальная сторона отключает канал, даже если он все еще принимает импульсы. По сравнению с высокоуровневым механизмом обнаружения UDLD, автосогласование является очень быстрым (считанные микросекунды), но отсутствует сквозное покрытие, как в UDLD (например, полный путь передачи данных: CPU/forwarding logic/port1/port2/forwarding logic/CPU versus port1/port2). Для обнаружения неисправностей в агрессивном режиме UDLD доступны похожие функциональные возможности, что при автосогласовании. Если согласование поддерживается с обеих сторон канала связи, нет необходимости включать агрессивный режим UDLD.

6. Необходимо соблюдать осторожность при настройке таймеров STP.

   STP-таймеры зависят друг от друга и от топологии сети. Произвольные изменения таймеров могут нарушить работу STP. Дополнительные сведения о таймерах STP см. в документе Общие сведения и настройка таймеров протокола STP.

7. Если возможны DoS-атаки, необходимо обеспечить безопасность STP-периметра при помощи Root Guard.

   Функции Root Guard и BPDU Guard обеспечивают защиту STP от внешних воздействий. При возможности возникновения подобных атак для защиты сети необходимо использовать функции Root Guard и BPDU Guard. Для получения дополнительной информации о функциях Root Guard и BPDU Guard см. документы:

   - Использование Root Guard для оптимизации работы протокола STP
   - Использование Portfast BPDU Guard для оптимизации работы протокола STP

8. Необходимо включить защиту BPDU на портах, поддерживающих функцию portfast для предотвращения воздействия на STP неавторизованных сетевых устройств (концентраторов, коммутаторов и мостов-маршрутизаторов), подключенных к этим портам.

   Правильно настроенная функция Root Guard предотвращает внешние воздействия на STP. Функция BPDU Guard отключает порты, принимающие любые пакеты BPDU (не только вышестоящие BPDU). Это очень удобно при исследовании подобных ситуаций, так как BPDU Guard оставляет сообщения в системном журнале и отключает порт. Следует отметить, что функции Root Guard и BPDU Guards не защищают от возникновения короткоживущих петель, если два порта, поддерживающих portfast, подключены напрямую или через концентратор.

9. Не использовать управляющую VLAN для пользовательского трафика. Управляющая VLAN состоит из компоновочного блока, а не целой сети.

   Интерфейс управления коммутатором принимает широковещательные пакеты по управляющей VLAN. При возникновении избыточных широковещательных рассылок (например, широковещательный шторм или ошибка приложения) может произойти перегрузка ЦП коммутатора, что приведет к нарушению работы STP.

10. Предсказуемый (жестко запрограммированный) корень STP и расположение резервного корня STP.

    Корень STP и резервный корень STP должны быть настроены таким образом, чтобы в случае возникновения ошибки, схождение происходило предсказуемым образом и для каждого сценария выстраивалась оптимальная топология. Необходимо избегать сохранения значения приоритета STP по умолчанию для предотвращения непредсказуемого выбора корневого коммутатора.

Дополнительные сведения

- **Поддержка продуктов LAN**
- **Поддержка технологии коммутации в сетях LAN**
- **Cisco Systems – техническая поддержка и документация**

Дата генерации PDF файла: Jan 05, 2010

http://www.cisco.com/support/RU/customer/content/10/107677/170.shtml