Содержание

[ВВЕДЕНИЕ 3](#_Toc405043740)

[1 ИЗБЫТОЧНОСТЬ В СЕТЯХ С КОММУТАЦИЕЙ 5](#_Toc405043741)

[1.1 Широковещательные штормы 5](#_Toc405043742)

[1.2 Множественная передача фреймов 6](#_Toc405043743)

[1.3 Неустойчивые базы данных MAC-адресов 8](#_Toc405043744)

[2 Обзор протокола связующего дерева 9](#_Toc405043749)

[2.1 Работа протокола связующего дерева 1](#_Toc405043750)0

[2.2 Использование связующего дерева для создания свободной от петель топологии сети 11](#_Toc405043751)

[3 РАСШИРЕННЫЕ ФУНКЦИИ ПРОТОКОЛА STP 14](#_Toc405043749)

[3.1 Выбор корневого моста 1](#_Toc405043750)5

[3.2 Последовательность состояний порта в протоколе связующего дерева 1](#_Toc405043750)7

[3.3 Таймеры протокола STP 1](#_Toc405043750)9

[ЗАКЛЮЧЕНИЕ 21](#_Toc405043752)

# ВВЕДЕНИЕ

Наличие избыточности в сети является важным фактором. Избыточность делает сеть устойчивой к возможным сбоям. Избыточность топологии защищает сеть и обес­печивает ее функционирование в случае сбоя в отдельном канале, на порте или в сете­вом устройстве. Однако топологии сетей, построенных на использовании коммутато­ров и мостов, обладающие избыточностью, подвержены широковещательным штор­мам (broadcast storms), множественным передачам фреймов и неустойчивости баз данных, в которых находятся адреса управления доступом к среде передачи (Media Ac­cess Control — MAC), называемые также МAC-адресами.

В настоящее время частные компании и государственные организации в своей дея­тельности все более полагаются на компьютерные сети. Доступ к файловым серверам, базам данных, к сети Internet, к интранет- и экстранет-сетям является критически важным для успешной работы компаний. В случае неработоспособности сети резко падает производительность работы компании, а потребители выражают неудовлетво­ренность. Поэтому компании стремятся добиться круглосуточной работоспособности сети на протяжении всей рабочей недели.

Достижение 100% работоспособности сети, вероятно, невозможно, однако обеспе­чение работоспособности сети на 99,999% (такой уровень надежности называется “пятью девятками”) — цель, которую ставят перед собой компании и организации. Это значение можно представить в среднем как один день простоя сети за 30 лет, один час простоя за 4000 дней или 5,25 минут простоя за год. Эта оценка применима и к другим важным службам, таким как финансовые операции, больницы, работа провай­деров служб Internet и т.д. Цена простоев оказывается весьма высокой. Многим другим предприятиям для уменьшения расходов приходится жертвовать надежностью сети. Достижение уровня надежности 99,999% требует исключительной надежности от сети. Эта надежность достигается за счет использования надежного оборудования и путем проектирования сети, устойчивой к сбоям и ошибкам. Для этого сеть проек­тируется таким образом, чтобы в ней быстро происходила реконвергенция и при пе­редаче данных обходился участок со сбоем.

Устойчивость к сбоям достигается за счет создания избыточности в сети. Избы­точность означает наличие в сети средств, которые превышают требования обычной ситуации при работе сети. Как избыточность помогает обеспечить надежность?

Предположим, что единственным способом добраться до работы является ваш личный автомобиль. Если в нем произошла поломка, которая сделала его неработо­способным, то добраться до работы станет невозможным до тех пор, пока автомобиль не будет отремонтирован.

Если в автомобиле происходит поломка, которая делает его неработоспособным, один раз в десять дней, то процент использования автомобиля составляет 90%. Воз­можность поехать на работу при этом обеспечивается в 9 днях из 10. Соответствен­но, надежность составляет 90%.

Покупка еще одного автомобиля улучшает ситуацию, однако для того, чтобы в обыч­ной ситуации просто добраться до работы, двух автомобилей не требуется. Один из них является лишним, однако он доступен в том случае, когда имеются проблемы с первым автомобилем. Это обеспечивает дополнительную надежность.

Целью создания избыточности в сети является предотвращение простоя в сети в тех случаях, когда в отдельной точке сети происходит сбой. Для повышения на­дежности все сети должны обладать избыточностью. Примером сети с избыточно­стью может служить сеть автомобильных дорог. Если одна трасса закрыта для ремон­та, то имеются другие маршруты к требуемому месту.

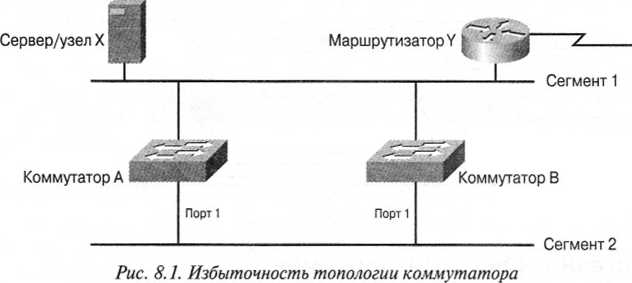
Предположим, что у крупного города есть небольшой пригород, который отделен от центральной части города рекой. Если через эту реку существует только один мост, то, соответственно, имеется лишь один маршрут к центру города. При такой топологии избыточность отсутствует.

Если на мосту возникла “пробка” или он поврежден в результате дорожного про­исшествия, то поездка в город через этот мост становится невозможной. Построение второго моста через эту реку создает избыточность в такой автомобильной сети. В этом случае пригород не оказывается отрезанным от центра города в том случае, если по одному из мостов проехать нельзя.

# 1 ИЗБЫТОЧНОСТЬ В СЕТЯХ С КОММУТАЦИЕЙ

Сети, в которых имеется избыточность маршрутов и устройств, позволяют сети поддерживать работоспособное состояние в течение более длительного времени. Как показано на рис. 8.1, топологии с избыточностью устойчивы по отношению к отдельным точкам сбоев. Если какой-либо маршрут или устройство выходят из строя, то избыточный маршрут или устройство могут принять на себя выполнение их функций.

Если коммутатор А выходит из строя, то потоки данных по прежнему могут передаваться из сегмента 2 в сегмент 1 и далее в канал маршрутизатора через коммутатор В.



Коммутаторы изучают *МАС-адреса* устройств, подсоединенных к их портам для того, чтобы соответствующим образом передать данные получателю. До тех пор, пока не стали известны все МАС-адреса устройств, одним из решений является отправка фреймов с неизвестным адресом получателя методом лавинной рассылки (to flood frames), т.е. всем устройствам, кроме того, от которого эти фреймы поступили. Широковеща­тельные и многоадресные сообщения также рассылаются методом лавинной рассылки.

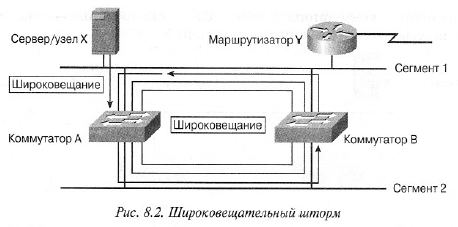
В сети, использующей коммутацию, и обладающей избыточной топологией, про­исходят широковещательные штормы (broadcast storms), многократное копирование фреймов и возникают проблемы, связанные с неустойчивостью таблиц МАС-адресов.

## Широковещательные штормы

Широковещательные и многоадресатные сообщения являются причиной потенци­альных проблем в сети с коммутацией. Коммутаторы обращаются с многоадресатными сообщениями как с широковещательными. Широковещательные и многоадресатные сообщения рассылаются методом лавинной рассылки. В этом случае фрейм рассыла­ется со всех портов коммутатора, за исключением того, на котором он был получен.

Как показано на рис. 8.2, если узел X посылает широковещательное сообщение, такое, например, как запрос протокола преобразования адресов (Address Resolution Protocol — ARP) относительно адреса 2-го уровня для маршрутизатора, то коммута­тор А рассылает это сообщение со всех своих портов. Поскольку коммутатор В сам находится в этом же сегменте, то и он также получает все широковещательные со­общения. Коммутатор В получает все широковещательные сообщения, рассылаемые коммутатором А, а коммутатор А, в свою очередь, получает все широковещательные сообщения, рассылаемые коммутатором В. Коммутатор А получает эти новые ши­роковещательные сообщения и вновь широковещательно рассылает их, как и ком­мутатор В. Пример такой ситуации приведен на рис. 8.2.

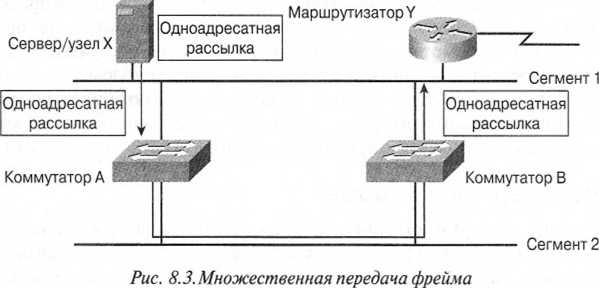
Таким образом, коммутаторы непрерывно рассылают одни и те же широковеща­тельные фреймы. Такое явление называется широковещательным штормом и оно продолжается до тех пор, пока один из коммутаторов не будет отсоединен от сети. При этом коммутаторы и конечные устройства настолько заняты обработкой широ­ковещательных сообщений, что передача данных пользователей становится малове­роятной. Передача полезных данных по сети при этом либо происходит крайне мед­ленно, либо вообще не функционирует.



## Множественная передача фреймов

В сети с избыточной топологией конечное устройство может получить несколько копий одного и того же фрейма (рис. 8.3).

Большинство протоколов устроены таким образом, что они игнорируют множест­венные передачи или выдерживают их без существенного нарушения работы сети. В це­лом протоколы, использующие механизм последовательной нумерации, предполагают, что множественная передача закончена, и нумерация в последовательности вновь начи­нается с нуля.

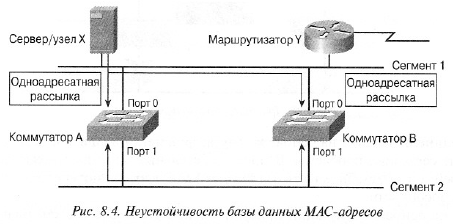
Другие протоколы пытаются передать фреймы, дублирующие друг друга, соответствующему протоколу более высокого уровня, часто с непредсказуемыми резуль­татами. Для того чтобы понять, как возникает множественная передача, рассмотрим си­туацию, показанную на рис 8.3, в которой происходят описанные ниже действия.

* Когда узел X посылает одноадресатный фрейм маршрутизатору Y, одна его ко­пия принимается через непосредственное соединение Ethernet в сегменте 1, в то время как коммутатор А также получает его копию и помещает ее в свой буфер.
* Если коммутатор А просматривает поле адреса получателя в этом фрейме и не обнаруживает соответствующей позиции для маршрутизатора Y в своей таб­лице МАС-адресов, то он лавинным образом рассылает этот фрейм со всех своих портов, за исключением порта, на который этот фрейм поступил.
* Если коммутатор В получает копию этого фрейма через коммутатор А в сег­менте 2 и также не находит соответствующей записи в своей таблице МАС- адресов, то он так же пересылает копию этого фрейма в сегмент 1.
* Маршрутизатор Y получает вторую копию того же самого фрейма.

Решением, позволяющим избежать появления петель и решить проблему множе­ственной передачи одних и тех же фреймов, является логический разрыв петли и предотвращение рассылки фреймов с одного из четырех интерфейсов при нор­мальной работе сети.

## Неустойчивость базы данных МАС-адресов

В сети, использующей коммутацию и обладающей избыточностью, коммутаторы могут получить неправильную информацию. Возможна ситуация, когда коммутато­ру поступает информация о том, что порту соответствует некоторый МАС-адрес, что на самом деле не соответствует действительности. Пример неустойчивости базы данных МАС-адресов показан на рис. 8.4.



В этом примере коммутатор В устанавливает соответствие между МАС-адресом узла X и портом 0, который подсоединяется к сегменту 1 при поступлении первого фрейма. Несколько позже, когда копия этого фрейма, переданная через коммутатор А, поступает на порт 1 коммутатора В, этот коммутатор должен удалить первую за­пись и создать новую, в которой будет ошибочно указано, что МАС-адресу станции X соответствует порт 1, подсоединенный к сегменту 2.

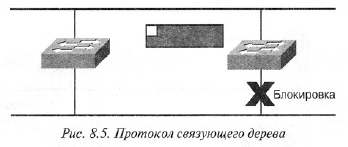
В зависимости от внутренней архитектуры рассматриваемого коммутатора по­следний может успевать или не успевать за быстрыми изменениями базы данных МАС-адресов. В данном случае решением, позволяющим решить проблему, вы­званную тем, что коммутатор не успевает за быстрыми изменениями базы данных МАС-адресов, является логический разрыв петли и предотвращение рассылки фреймов с одного из четырех интерфейсов при нормальной работе сети.

**2** **ОБЗОР ПРОТОКОЛА СВЯЗУЮЩЕГО ДЕРЕВА**

*Протокол связующего дерева (Spanning Tree Protocol* — *STP)* является протоколом управления каналом 2-го уровня, который используется для поддержки такого состоя­ния сети, в котором в ней отсутствуют петли. Первоначальная версия протокола STP была разработана корпорацией Digital Equipment (Digital). Впоследствии комитет IEEE 802 модернизировал алгоритм связующего дерева и опубликовал его в виде спе­цификации IEEE 802. Id. Алгоритмы корпорации Digital и комитета IEEE отличаются друг от друга и несовместимы. Коммутаторы Cisco, такие как Catalyst 1900 и 2950, ис­пользуют версию протокола STP спецификации IEEE 802.Id.

Назначение протокола STP состоит в поддержке такого состояния сети, в кото­ром в ней отсутствуют петли. Для того, чтобы сеть была свободна от петель, при ее обнаружении мост или коммутатор автоматически осуществляет логическую блоки­ровку одного или нескольких избыточных портов.

Протокол STP постоянно проверяет сеть на предмет появления нового канала или сбоя в уже существующих; в этом случае коммутатор или мост получает инфор­мацию о таком событии. Пример работы протокола STP показан на рис. 8.5. В слу­чае изменения топологии сети коммутаторы и мосты, на которых функционирует протокол STP, автоматически реконфигурируют свои порты для того, чтобы предот­вратить потерю связи в каналах и возникновение петель.



В объединенных сетях на основе коммутаторов петли на физическом уровне мо­гут вызвать серьезные проблемы. Широковещательные штормы, множественная пе­редача фреймов и нестабильность баз данных МАС-адресов могут сделать такие сети неработоспособными.

Сетям на основе коммутации присущи такие преимущества, как малый размер коллизионных доменов, микросегментация и дуплексный режим работы. Проще го­воря, преимуществом таких сетей является высокая производительность. Протокол STP используется в сетях с коммутацией для создания логической топологии, сво­бодной от петель, из физической топологии, в которой петли присутствуют. Каналы, порты и коммутаторы, которые не являются элементами активной, свободной от петель топологии не участвуют в пересылке фреймов данных

Если какой-либо элемент активной топологии выходит из строя, то необходимо определить новую свободную от петель топологию. Вычислить новую свободную от петель топологию или выполнить конвергенцию сети необходимо как можно быст­рее для того, чтобы свести к минимуму время, в течение которого конечные станции не имеют доступа к сетевым ресурсам.

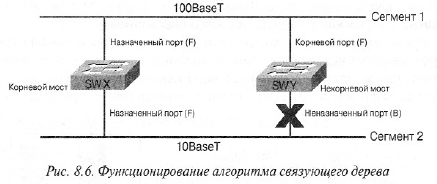
Для современных сетей версия протокола STP, определенная в стандарте IEEE 802.Id, осуществляет конвергенцию к новой топологии слишком медлен­но. Для преодоления этих ограничений был разработан новый стандарт — IEEE 802. lw или RSTP. В настоящей главе рассматривается как протокол STP, так и протокол RSTP.

## Работа протокола связующего дерева

После того, как работа сети стабилизирована и в ней произошла конвергенция, каждая сеть имеет одно связующее дерево. В результате этого для каждой сети ис­пользующей коммутацию, будут выполнены следующие условия:

* В каждой сети существует один корневой мост;
* У каждого моста, который не является корневым, имеется один корневой порт;
* В каждом сегменте имеется один назначенный порт;
* Порты, которые не являются назначенными, не используются.

Для пересылки (forwarding — F) используются только корневые и назначенные порты. Порты, которые не являются назначенными, отбрасывают поступающие на них фреймы. Такие порты называются блокирующими (blocking — В) или отбрасы­вающими. Корневым портом моста называется порт, ближайший к корневому мос­ту. Для каждого некорневого моста должен быть выбран один корневой порт. На рис. 8.6 показана работа алгоритма связующего дерева.



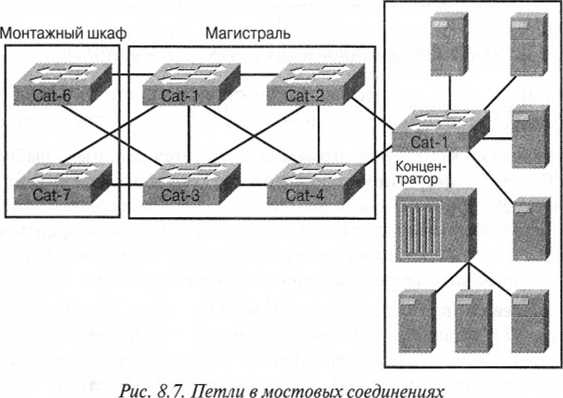
Сначала протокол STP осуществляет конвергенцию сети к свободной от петель логической топологии, путем выполнения описанных ниже трех этапов.

1. **Выбор корневого моста.** В протоколе STP имеется механизм выбора корневого моста. В одной конкретной сети в качестве корневого может выступать только один мост. У корневого моста все порты являются назначенными. Назначен­ные порты в обычном режиме осуществляют пересылку фреймов. Когда порт находится в таком состоянии, он может получать и отправлять потоки данных. В данном примере в качестве корневого моста выбирается коммутатор X.
2. **Выбор корневого порта на некорневых мостах.** Для каждого некорневого моста протокол STP устанавливает один корневой порт. В качестве корневого порта выбирается маршрут от некорневого моста до корневого моста с наименьшей оценкой. Корневые порты в обычном состоянии осуществляют пересылку дан­ных. Оценка маршрута в алгоритме связующего дерева представляет собой ак­кумулированную оценку, которая вычисляется на основе ширины полосы про­пускания. Например, от коммутатора Y маршрутом к корневому мосту с наи­меньшей оценкой является маршрут через канал 100 BaseT Fast Ethernet.
3. **Выбор назначенного порта в каждом сегменте сети.** В каждом сегменте протокол STP выбирает один назначенный порт. В качестве назначенного порта выбира­ется порт моста, который имеет наименьшую оценку маршрута к корневому мосту. Назначенные порты в обычном состоянии осуществляют пересылку дан­ных для данного сегмента. В данном примере назначенный порт для обоих сег­ментов находится на корневом мосту, поскольку корневой мост непосредствен­но подсоединен к обоим сегментам. Порт 10BASE-T Ethernet на коммутаторе Y не является назначенным портом, поскольку для каждого сегмента имеется только один назначенный порт. Порты, которые не являются назначенными, в обычной ситуации находятся в состоянии блокировки для осуществления ло­гического разбиения топологии, содержащей петли. Когда порт находится в со­стоянии блокировки, он не пересылает данные, однако может их получать.

## Использование связующего дерева для создания свободной от петель топологии сети

Сетевые топологии с избыточностью создаются для того, чтобы обеспечить функционирование сети в случае сбоев в отдельных точках сети. В таких случаях пользователи менее болезненно воспринимают нарушения своей работы, поскольку сеть продолжает функционировать. Любое нарушение работы сети, вызываемое сбо­ем, необходимо сделать как можно более кратковременным.

Избыточность повышает надежность сети. В сетях, построенных на использовании коммутаторов или мостов, наличие избыточных каналов между коммутаторами и мос­тами помогает преодолеть последствия сбоя на каком-либо из каналов. Эти соедине­ния создают в сети физические петли. Пример такой ситуации приведен на рис. 8.7.



Петли в мостовых соединениях создаются таким образом, чтобы в случае сбоя в каком-либо из каналов его функции по пересылке данных мог выполнить другой канал. Коммутаторы функционируют на 2-м уровне эталонной модели OSI и реше­ния о пересылке принимаются на этом уровне. Следствием этого является недопус­тимость существования петель в сетях с коммутацией.

В случае, когда получатель еще не известен, коммутаторы выполняют лавинную рас­сылку фреймов со всех своих портов. Широковещательные и многоадресатные сообще­ния рассылаются со всех портов, кроме того, на который эти сообщения поступают (лавинная рассылка). Такие фреймы могут попасть в петлю. В заголовке 2-го уровня от­сутствует поле *времени существования (time to live* — *TTL).* Если фрейм посылается в сеть, топология 2-го уровня которой содержит петли, то он может попасть в такую петлю и двигаться по ней бесконечно, в результате чего напрасно затрачивается полоса пропус­кания и сеть становится неработоспособной.

На 3-м уровне модели OSI имеется поле времени существования пакета. При движении пакета по петле его значение последовательно уменьшается и когда оно становится равным нулю, пакет отбрасывается. Однако в этом случае возникает ди­лемма. Физическая топология, в которой имеются петли через коммутаторы или мосты, необходима для обеспечения надежности сети, однако сеть с коммутацией не должна иметь петель. Решением этой дилеммы является допустимость физических петель, но создание свободной от петель логической топологии.

Создаваемая при этом свободная от петель логическая топология называется де­ревом. Такая топология, которая является звездообразной или расширенной звездо­образной логической топологией, представляет собой связующее дерево сети. Такая топология является связующим деревом, поскольку по этому дереву достижимы все устройства сети. Алгоритм, используемый для создания свободной от петель логиче­ской топологии, называется алгоритмом связующего дерева. Для конвергенции всей сети такому алгоритму требуется относительно большое время.

**3 РАСШИРЕННЫЕ ФУНКЦИИ ПРОТОКОЛА STP**

Протокол STP выбирает корневой узел, называемый корневым мостом, а затем строит топологию сети, в которой к каждому сетевому узлу существует только один маршрут. Образующееся при этом дерево исходит из корневого моста. Избыточные каналы, которые не являются частью дерева кратчайших маршрутов, блокируются. Создание свободной от петель топологии становится возможным именно из-за того, что некоторые маршруты блокируются. Фреймы данных, поступающие в заблокиро­ванный канал, отбрасываются.

Протокол STP требует чтобы сетевые устройства обменивались сообщениями для обнаружения петель в мостовых соединениях. Каналы, которые вызывают появле­ние петли, переводятся в заблокированное состояние. Сообщения, которые посы­лают коммутаторы для того, чтобы можно было создать свободную от петель логиче­скую топологию, называются *модулями данных мостового протокола (bridge protocol data units* — *BPDU).* Эти модули BPDU продолжают приниматься на заблокирован­ных портах, что обеспечивает возможность построения нового связующего дерева в случае, если происходит сбой в устройстве или на активном маршруте.

Модули BPDU содержат достаточное количество информации для того чтобы коммутаторы могли выполнить описанные ниже действия.

* Выбрать один коммутатор который будет выполнять функции корня связую­щего дерева;
* Вычислить кратчайший путь от себя до корневого коммутатора;
* Для каждого сегмента LAN-сети назначить один из коммутаторов ближайшим к корневому коммутатору. Этот мост называется назначенным коммутатором. Назначенный коммутатор обрабатывает все данные, пересылаемые из этой LAN-сети корневому мосту.
* Каждый некорневой коммутатор выбирает один из своих портов в качестве корневого порта. Этот интерфейс обеспечивает наилучший маршрут к корне­вому коммутатору.
* Выбрать порты, которые являются частью связующего дерева (назначенные порты). Порты, которые не являются назначенными, блокируются.

Когда связующее дерево создает свободную от петель логическую топологию, оно всегда использует описанную ниже последовательность принятия решений, состоя­щую из четырех этапов.

1. Выбор наименьшего идентификатора (ID) корневого моста (bridge ID — BID);
2. Вычисление наименьшей оценки маршрута к корневому мосту.
3. Выбор наименьшего идентификатора (ID) моста-отправителя (bridge ID — BID);
4. Выбор наименьшего идентификатора (ID) порта.

Мост использует эту состоящую из четырех этапов последовательность принятия решений для того, чтобы сохранить копию “наилучшего ” модуля BPDU на каждом порте. Когда мост выполняет эту оценку маршрутов, он рассматривает все модули BPDU, которые были получены на данном порте, а также BPDU, которые были бы посланы на этот порт. При поступлении каждого модуля BPDU для него выполняет­ся описанная выше четырехэтапная проверка с целью выяснить, не является ли он более привлекательным (т.е. имеет меньшее значение оценки), чем существующий BPDU, сохраненный для этого порта. Если новый BPDU (или локально сгенериро­ванный BPDU) более привлекательный, то он заменяет прежнее значение.

Кроме того, этот процесс “сохранения наилучшего BPDU” управляет отправкой модулей BPDU. Когда мост в первый раз становится активным, все его порты посы­лают модули BPDU каждые 2 секунды (если используются значения таймеров по умолчанию). Однако если порт получает от другого моста BPDU, который более привлекателен, чем BPDU, который он сам отправляет, то этот локальный порт пре­кращает рассылку BPDU. Если от соседнего устройства в течение определенного времени (по умолчанию 20 секунд) не поступает более привлекательного модуля BPDU, то локальный порт возобновляет рассылку.

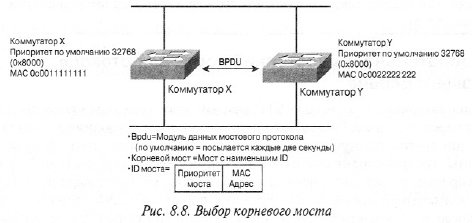
## Выбор корневого моста

При использовании протокола STP в качестве корневого выбирается мост с наи­меньшим значением идентификатора ID. Параметр BID включает в себя приоритет и МАС-адрес моста.

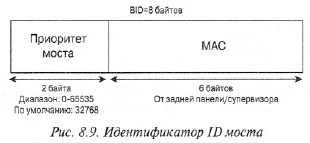
Коммутаторы и мосты, выполняющие алгоритм связующего дерева, регулярно обмениваются сообщениями о конфигурации с другими коммутаторами и мостами (по умолчанию каждые две секунды) с помощью сообщений BPDU. Частью инфор­мации, включаемой в BPDU является идентификатор BID.

Протокол STP требует, чтобы каждому мосту был назначен уникальный идентифи­катор (BID). Обычно этот идентификатор BID состоит из значения приоритета (двабайта) и МАС-адреса моста (шесть байтов).

Приоритетом по умолчанию, в соответст­вии со спецификацией IEEE 802.1d, является значение 32768 (1000 0000 0000 0000 в двоичной записи или 0x8000 в шестнадцатеричной), которое является значением из среднего диапазона. На рис. 8.8. оба коммутатора имеют одно и то же значение при­оритета по умолчанию, поэтому корневым станет коммутатор, который имеет меньшее значение МАС-адреса. В данном примере коммутатор X (SW X) является корневым и его ID моста равен 0x8000.



ID моста, показанного на рис. 8.9, состоит из приоритета моста, который по умолчанию равен 32568, и базового МАС-адреса коммутатора.



Когда коммутатор запускается впервые, он считает себя корневым и посылает минимальный BPDU. Он содержит МАС-адрес коммутатора и идентификаторы BID корневого маршрутизатора и отправителя. Все коммутаторы просматривают эти от­сылаемые BID. Когда коммутатор получает BPDU, который имеет меньший корне­вой BID, он заменяет соответствующий BID в модулях BPDU. Все коммутаторы просматривают их и делают вывод, что мост с наименьшим значением BID будет корневым мостом.

Администратор может повлиять на это решение путем установки меньшего значе­ния приоритета коммутатора, чем принимаемое по умолчанию. В этом случае ID моста становится меньше.

## Последовательность состояний порта в протоколе связующего дерева

При использовании протокола STP каждый мост сети при включении питания проходит через состояние блокировки и промежуточное состояние изучения адресов и состояний других коммутаторов. Если порты соответствующим образом сконфи­гурированы, то происходит стабилизация и они переходят в состояние пересылки или блокировки. В состоянии пересылки порты обеспечивают маршрут к корневому мосту с наименьшей оценкой. Два промежуточных состояния имеют место в том случае, когда мост узнает об изменении в сетевой топологии. Во время изменения топологии порт временно находится в состояниях прослушивания и изучения топо­логии сети. На рис. 8.10 показаны состояния порта, на котором работает протокол связующего дерева.



Первоначально все порты моста находятся в состоянии блокировки, в котором они прослушивают модули BPDU. При первой загрузке моста он считает себя кор­невым мостом и переходит в состояние прослушивания. Отсутствие модулей BPDU в течение определенного периода времени (максимальный возраст) может вызвать переход от состояния блокировки к состоянию прослушивания. Когда порт нахо­дится в промежуточном состоянии прослушивания, он может отправлять и получать BPDU для определения текущей топологии. В этом состоянии передача данных пользователей не происходит.

Именно в состоянии прослушивания мост выполняет операции по выбору корневого моста, корневых портов на некорневых мостах и вы­бора назначенных портов для каждого сегмента. Порты, которые остаются назна­ченными или корневыми в течение 15 секунд (задержка пересылки), переходят в со­стояние изучения топологии. Порты, которые не являются назначенными или кор­невыми, возвращаются в состояние блокировки. При переходе порта в состояние изучения топологии он может заполнять свою таблицу МАС-адресов адресами, по­лученными на его портах, однако не пересылает фреймы пользователя. По умолча­нию состояние изучения продолжается 15 секунд (также время, равное задержке пе­ресылки). В это время мост все еще не передает данных пользователя.

Ниже приводятся состояния протокола STP.

* **Состояние блокировки.** Фреймы пользователей не пересылаются, прослуши­ваются модули BPDU.
* **Состояние прослушивания.** Фреймы пользователей не пересылаются, но про­слушиваются.
* **Состояние изучения топологии.** Фреймы пользователей не пересылаются, изу­чаются адреса других устройств.
* **Состояние пересылки.** Пересылаются фреймы пользователей и изучаются ад­реса других устройств.
* **Состояние отключения.** Фреймы пользователей не пересылаются, модули BPDU не прослушиваются.

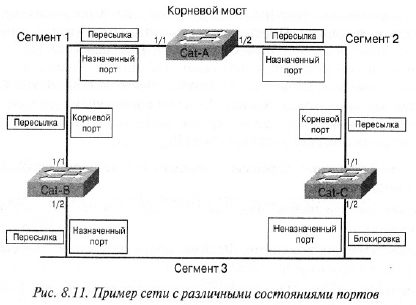
В состоянии изучения топологии уменьшается объем лавинной рассылки, кото­рая требуется в начале пересылки. Если какой-либо порт по-прежнему является на­значенным или корневым портом в заключительной фазе состояния изучения топо­логии, то он переходит в состояние пересылки. Порты, которые не являются назна­ченными или корневыми, возвращаются в состояние блокировки. В состоянии пересылки порт может отправлять и получать данные пользователей.

Обычно время, требуемое порту для перехода из состояния блокировки в состоя­ние пересылки, составляет от 30 до 50 секунд. Это время таймеров протокола свя­зующего дерева может быть изменено сетевым администратором. Обычно эти пе­риоды таймеров устанавливаются со стандартным значением. Значения по умолча­нию устанавливаются для того, чтобы у сети была возможность собрать правильную информацию о сетевой топологии.

Время, требуемое порту для перехода из состояния прослушивания в состояние изучения или из состояния изучения топологии в состояние пересыпки, называется задержкой пересылки и по умолчанию составляет 15 секунд. Максимальный воз­раст, который по умолчанию равен 20 секундам, равен времени хранения мостом модуля BPDU перед его отбрасыванием. Если порт находится в состоянии блоки­ровки и не получает новых модулей BPDU за время максимального возраста, то он переходит из состояния блокировки в состояние прослушивания.

Если порт коммутатора подсоединен только к станциям конечных пользователей (не имеет соединений с другими коммутаторами или мостами), то на таких портах конечных пользователей следует включить функцию коммутатора Catalyst, называемую *“быстрым портом” (portfast).*

При включенной функции быстрого порта в момент активизации пор­та он автоматически переходит из состояния блокировки в состояние пересылки. Это становится возможным потому, что в этом случае через эти порты не могут возникнуть петли, поскольку к ним не подсоединены другие коммутаторы или мосты. На рис. 8.11 приведен пример сети, порты которой находятся в различных состояниях.



## Таймеры протокола STP

Время приветствия определяет временной интервал между регулярными рассыл­ками модулей конфигурации BPDU. Стандарт 802.1D определяет стандартное зна­чение этого интервала, равное 2 секундам. Однако это значение управляет только модулями конфигурации BPDU, которые генерируются корневым мостом. Осталь­ные мосты распространяют модули BPDU по мере их получения. Иными словами, если в течение 2-20 секунд модули BPDU перестают поступать, то некорневые мос­ты прекращают рассылать регулярные BPDU. Если этот простой длится более 20 се­кунд, что является максимальным значением возраста по умолчанию, то мост делает недействительными сохраненные модули BPDU и начинает искать новый корневой порт. Максимальным возрастом считается время, в течение которого мост хранит BPDU перед их отбрасыванием.

Под задержкой пересылки понимается одно значение, которое определяет оба этих состояния. Первоначально в качестве значения по умолчанию принималось значение 15 секунд, которое определялось предположением, что максимальный раз­мер сети определяется семью переходами, максимальное количество утерянных BPDU равно трем, а интервал приветствия равен двум секундам. Таймер задержки при пересылке также управляет периодом старения адресной таблицы моста после изменения активной сетевой топологии.

Ранее говорилось о том, что каждый порт сохраняет копию наилучшего BPDU. который он получал и просматривал. До тех пор, пока мост каждые 2 секунды полу­чает модули BPDU, получающий их мост поддерживает постоянные копии значений этих модулей. Однако если в устройстве, рассылающем эти модули BPDU с наилуч­шими маршрутами, происходит сбой, то должен заработать механизм, который по­зволит другим мостам выполнить функции этого вышедшего из строя устройства. В таблице 8.3 приведены некоторые значения таймеров протокола STP.

|  |  |  |
| --- | --- | --- |
| Таблица 8.1. Таймеры протокола STP | | |
| Таймер | Назначение | Значение по умолчанию |
| Время приветствия (Hello Time) | Промежуток времени между регулярными рассылками модулей BPDU корневым | 2 секунды |
|  | МОСТОМ |  |
| Задержка пересылки | Продолжительность состояний прослушивания и изучения топологии | 15 секунд |
| Максимальный возраст (Max Age) | Время хранения модулей BPDU | 20 секунд |

В некоторых ситуациях коммутаторы могут обнаружить изменения топологии на непосредственно подсоединенных каналах и сразу перейти в состояние прослуши­вания, не ожидая истечения максимального возраста.

# ЗАКЛЮЧЕНИЕ

В настоящем курсовом проекте были рассмотрены следующие аспекты функционирования протокола STP:

* Избыточность топологии и коммутируемые сети с избыточной топологией, широковещательные штормы, множественная передача фреймов и неста­бильность базы данных МАС-адресов.
* Топологии с избыточностью обеспечивают более надежные и устойчивые к отказам сети.
* Протокол STP и алгоритм связующего дерева обеспечивают свободную от пе­тель логическую топологию в физической топологии, имеющей петли.
* Обсуждены важные для работы протокола STP понятия, такие как функцио­нирование связующего дерева, структура связующего дерева, выбор корневого моста, последовательность состояний портов связующего дерева, выбор на­значенных портов, оценка маршрута, таймеры протокола STP, перерасчет связующего дерева и конвергенция сети.