

## **П/з №9. Семейство протоколов STP.**

В сетях с коммутацией имеется специфическая проблема обеспечения надежности: стандартный протокол моста работает корректно только в древовидной топологии, где каждая пара узлов имеет только один путь соединения. Однако для обеспечения надежности сети необходимы дополнительные маршруты между узлами, которые могут использоваться при отказе основного пути. Простым решением этой проблемы является создание сети с запасными маршрутами, ручное определение связной древовидной топологии и ручное блокирование всех портов, не входящих в эту топологию. В случае отказа сети этот процесс должен повторяться вручную. Однако такой подход обеспечивает невысокую надежность сети, так как время простоя будет минимальным: необходимо обнаружить отказ, определить его местоположение, найти новую рабочую топологию сети и настроить ее.

Для автоматического выполнения перечисленных действий, то есть нахождения и конфигурирования активной древовидной топологии, мониторинга состояния ее связей и перехода к новой древовидной топологии при обнаружении отказа связи в коммутируемых локальных сетях используются алгоритм покрывающего дерева (Spanning Tree Algorithm, STA) и реализующий его протокол покрывающего дерева (**Spanning Tree Protocol, STP**).

В 1983 году был разработан алгоритм покрывающего дерева, который получил признание от IEEE и был включен в спецификацию 802.1D вместе с алгоритмом прозрачного моста. Протокол STP является упрощенной версией протокола маршрутизации, где кадры направляются по активному маршруту независимо от их адреса назначения, в отличие от протоколов маршрутизации, где активный маршрут выбирается для каждого адреса индивидуально. В настоящее время протокол STP широко применяется на коммутаторах, являющихся основными устройствами в локальных сетях. Версия протокола RSTP (Rapid STP) была разработана для ускорения процесса поиска новой топологии и занимает всего несколько секунд.

Семейство протокола Spanning Tree Protocol (STP) включает в себя несколько различных протоколов, разработанных для предотвращения петель в сетях Ethernet. Основными протоколами в этом семействе являются:

1. Spanning Tree Protocol (STP) – это оригинальный протокол, разработанный компанией Cisco Systems. Он используется для определения и блокирования избыточных путей в сети Ethernet, чтобы избежать возникновения петель.

2. Rapid Spanning Tree Protocol (RSTP). Данный протокол был разработан для улучшения скорости сходимости сети после изменения топологии. RSTP работает быстрее, чем оригинальный STP, благодаря использованию дополнительных состояний портов.

3. Multiple Spanning Tree Protocol (MSTP). Этот протокол позволяет группировать VLAN в различные экземпляры STP, что обеспечивает более эффективное использование ресурсов сети и повышает производительность.

Каждый из этих протоколов имеет свои особенности и предназначен для определенных сценариев использования. Далее мы рассмотрим каждый протокол отдельно.

Другими словами, STP— канальный протокол, основной задачей которого является устранение петель в топологии произвольной сети Ethernet, в которой есть один или более сетевых мостов, связанных избыточными соединениями. Протокол автоматически блокирует соединения, которые в данный момент для полной связности коммутаторов являются избыточными.

Петля коммутации (Bridging loop, Switching loop) — состояние в сети, при котором происходит бесконечная пересылка фреймов между коммутаторами, подключенными в один и тот же сегмент сети. Из определения становится ясно, что возникновение петли создает большие проблемы — ведет к перегрузке свитчей и неработоспособности данного сегмента сети. На рисунке 1 показан пример топологии, при которой произойдет образование петли при отсутствии защитных механизмов.

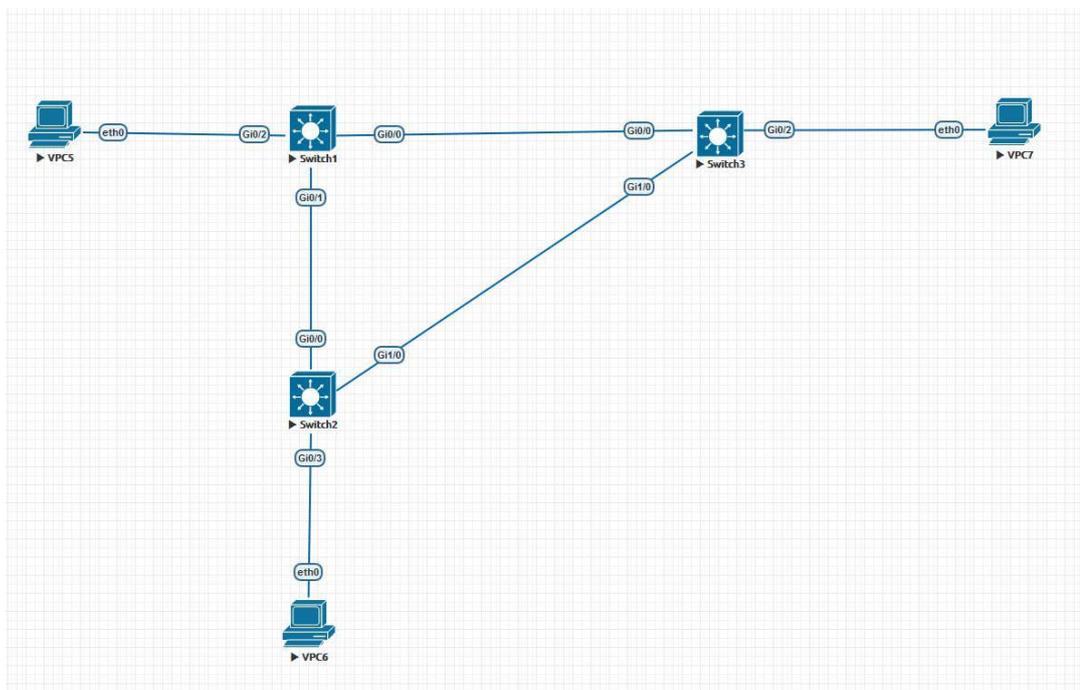


Рисунок 1 – Пример топологии

Рассмотрим случаи, когда могут возникнуть петли.

Какой-либо из хостов посылает бродкаст фрейм:

- 1) К примеру, *VPC5* отправляет пакет с бродкастовым адресом назначения.
- 2) *Switch1* приняв данный пакет, должен отправить его через все порты, кроме порта, с которого пришел данный пакет. Пакет отправится через порты *Gi0/0*, *Gi1/0*.
- 3) Коммутаторы *Switch2*, *Switch3* приняв данный пакет также должны будут его разослать пакет. Таким образом *Switch2*, получивший пакет от *Switch1* отправит его *Switch3*, а *Switch3* отправит его *Switch2*.
- 4) Далее, *Switch2* получив пакет от *Switch3*, отправит его *Switch1*, а *Switch3* получив пакет от *Switch2*, отправит его также *Switch1*. Тем самым, мы приходим к шагу 1 и он будет продолжаться бесконечно. Также все усугубляется тем, что на 4 шаге *Switch1* будет иметь уже два экземпляра фрейма, так как получит их и от *Switch2*, и от *Switch3*.

Шаги 1) — 4) будут повторяться бесконечно и на коммутаторах это происходит в долю секунды. Также образование петли приводит к тому, что постоянно будет меняться таблица MAC-адресов на коммутаторах и MAC-адрес отправителя *VPC5* будет постоянно приписываться то к интерфейсу *Gi0/0*, то *Gi1/0*

или же Gi0/2 (если в этот момент VPC5 будет отправлять другие пакеты). Такой цикл приведет к некорректной работе сети и всех коммутаторов.

Также может возникнуть ситуация, когда петля может образоваться и без отправки бродкаст фрейма.

К примеру, VPC5 отправляет фрейм с MAC-адресом назначения.

Возможна ситуация, что MAC-адрес назначения отсутствует в таблице MAC-адресов коммутаторов. В данном случае, коммутатор будет пересылать пакет через все порты, кроме порта с которого получил данный фрейм. И получаем такую же ситуацию, как и с бродкаст фреймом.

Рассмотрим, как работает протокол STP.

В качестве примера приведена сеть, изображенная на рисунке 2.

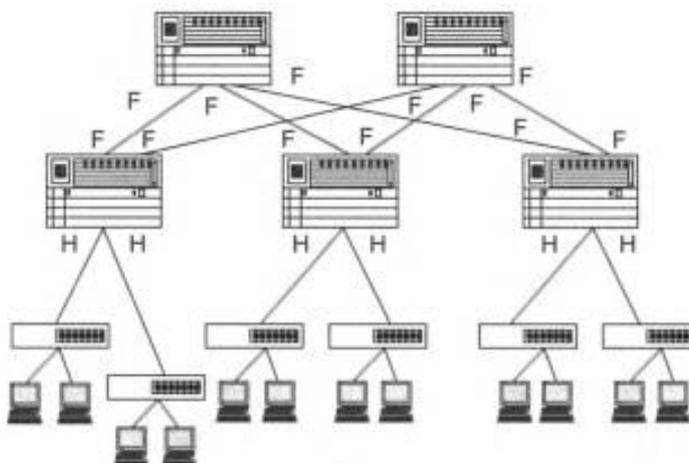


Рисунок 2 – Пример сети

Протокол STP формализует сеть (рисунок 2) в виде графа (рисунок 3), вершинами которого являются коммутаторы и сегменты сети.

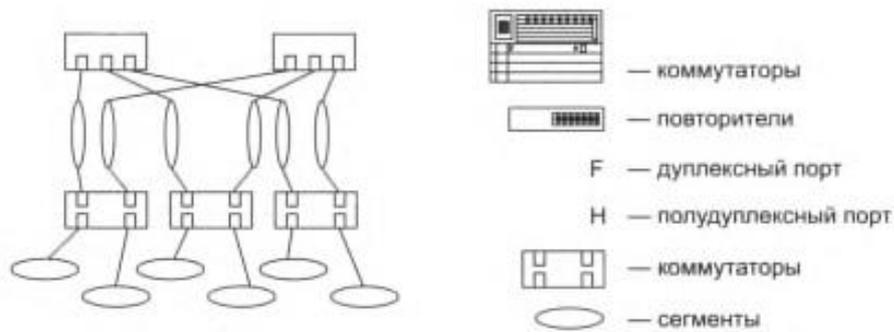


Рисунок 3 – Формализованное представление сети в соответствии с алгоритмом STA

Сегмент — это связная часть сети, не содержащая коммутаторов (маршрутизаторов). Сегмент может быть разделяемым (во времена создания алгоритма STA это был единственный тип сегмента) и включать устройства физического уровня — повторители/концентраторы, существование которых коммутатор, будучи устройством канального уровня, «не замечает».

Сегмент может представлять собой и двухточечный канал. В коммутируемых локальных сетях, применяемых сегодня, это единственный тип сегмента. Протокол покрывающего дерева обеспечивает построение древовидной топологии связей с единственным путем минимальной длины от каждого коммутатора и от каждого сегмента до некоторого выделенного корневого коммутатора — корня дерева. Единственность пути гарантирует отсутствие петель, а минимальность расстояния — рациональность маршрутов следования трафика от периферии сети к ее магистрали, роль которой исполняет корневой коммутатор. В качестве расстояния в STA используется метрика — традиционная для протоколов маршрутизации величина, обратно пропорциональная пропускной способности сегмента.

Для обмена информацией между собой свичи используют специальные пакеты, так называемые BPDU (Bridge Protocol Data Units). BPDU бывают двух видов:

1. Configuration BPDU (CBPDU) - эти пакеты используются для обмена информацией о топологии сети и настройках портов между коммутаторами.

СВРРDU помогает в определении корневого моста, вычислении кратчайшего пути до корневого моста и определении состояния портов.

2. Topology Change Notification BPDU (TCN BPDU) - эти пакеты используются для уведомления других коммутаторов о изменениях в топологии сети, таких как добавление или удаление портов. TCN BPDU помогает ускорить процесс обновления топологии сети и предотвращает возможные петли в сети.

Конфигурационные BPDU содержат несколько полей, остановимся на самых важных:

- Идентификатор отправителя (Bridge ID)
- Идентификатор корневого коммутатора (Root Bridge ID)
- Идентификатор порта, из которого отправлен данный пакет (Port ID)
- Стоимость маршрута до корневого коммутатора (Root Path Cost)

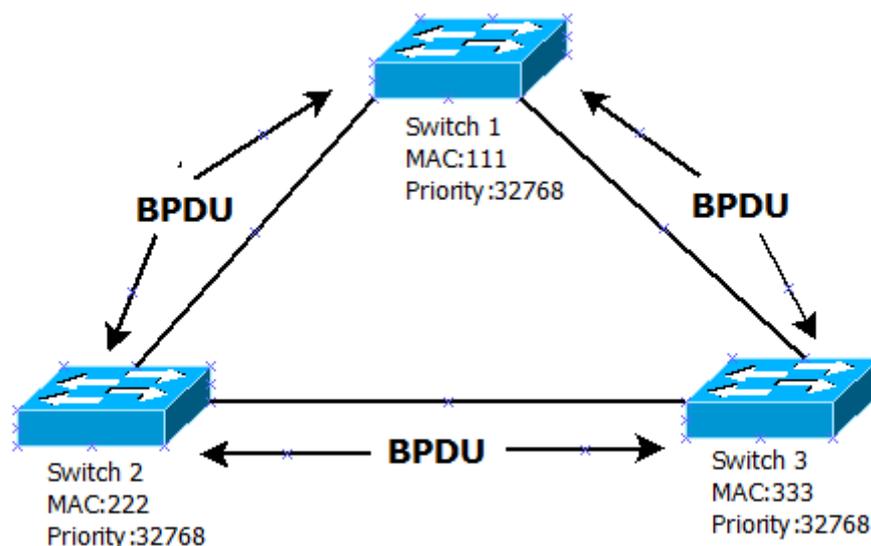


Рисунок 4 – Обмен BPDU

Так как устройства не знают и не хотят знать своих соседей, никаких отношений (смежности/соседства) они друг с другом не устанавливают. Они шлют BPDU из всех работающих портов на мультикастовый ethernet-адрес 01-80-c2-00-00-00 (по умолчанию каждые 2 секунды), который прослушивают все порты коммутатора с включенным STP.

Рассмотрим, как формируется топология без петель.

Сначала выбирается так называемый корневой мост/свич (root bridge). Это устройство, которое STP считает точкой отсчета, центром сети; все дерево STP

сходится к нему. Выбор базируется на таком понятии, как идентификатор свича (Bridge ID). Bridge ID это число длиной 8 байт, которое состоит из Bridge Priority (приоритет, от 0 до 65535, по умолчанию 32768+номер VLAN или инстанс MSTP, в зависимости от реализации протокола), и MAC-адреса устройства. В начале выборов каждый коммутатор считает себя корневым, о чем и заявляет всем остальным с помощью BPDU, в котором представляет свой идентификатор как ID корневого свича. При этом, если он получает BPDU с меньшим Bridge ID, он начинает анонсировать полученный Bridge ID в качестве корневого. В итоге, корневым оказывается тот коммутатор, чей Bridge ID меньше всех.

### Роли портов

После того, как коммутаторы получили и сравнили ID и выбрали root bridge, каждый из остальных коммутаторов должен найти один, и только один порт, который будет вести к корневому свичу. Такой порт называется **корневым портом (Root port)**. Чтобы понять, какой порт лучше использовать, каждый некорневой свич определяет стоимость маршрута от каждого своего порта до корневого. Эта стоимость определяется суммой стоимостей всех линков, которые нужно пройти кадру, чтобы дойти до корневого свича. В свою очередь, стоимость линка определяется просто – по его скорости (чем выше скорость, тем меньше стоимость). Процесс определения стоимости маршрута связан с полем BPDU “Root Path Cost” и происходит так:

- 1) Корневой свич посылает BPDU с полем Root Path Cost, равным нулю
- 2) Ближайший свич смотрит на скорость своего порта, куда BPDU пришел, и добавляет стоимость согласно таблице

Скорость порта	Стоимость STP (802.1d)
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2

- 3) Далее этот коммутатор посылает BPDU нижестоящим коммутаторам, но уже с новым значением Root Path Cost, и далее по цепочке вниз

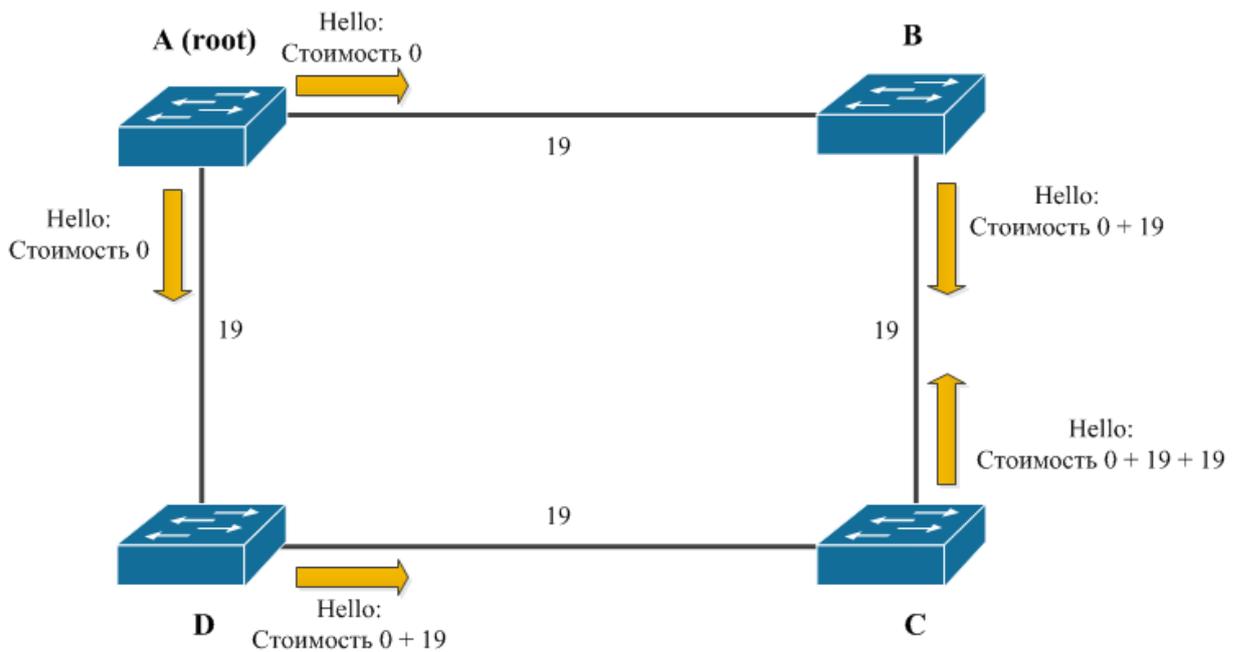


Рисунок 5 – Пример сети. Отправка стоимости

*Если имеют место одинаковые стоимости — корневым выбирается меньший порт.*

- 4) Далее выбираются **назначенные (Designated)** порты. Из каждого конкретного сегмента сети должен существовать только один путь по направлению к корневому свичу, иначе возникнет петля. В данном случае имеем в виду физический сегмент, в современных сетях без концентраторов это, грубо говоря, просто провод. Назначенным портом выбирается тот, который имеет лучшую стоимость в данном сегменте. У корневого коммутатора все порты — назначенные.
- 5) И после того, как выбраны корневые и назначенные порты, оставшиеся блокируются, таким образом разрывая петлю.

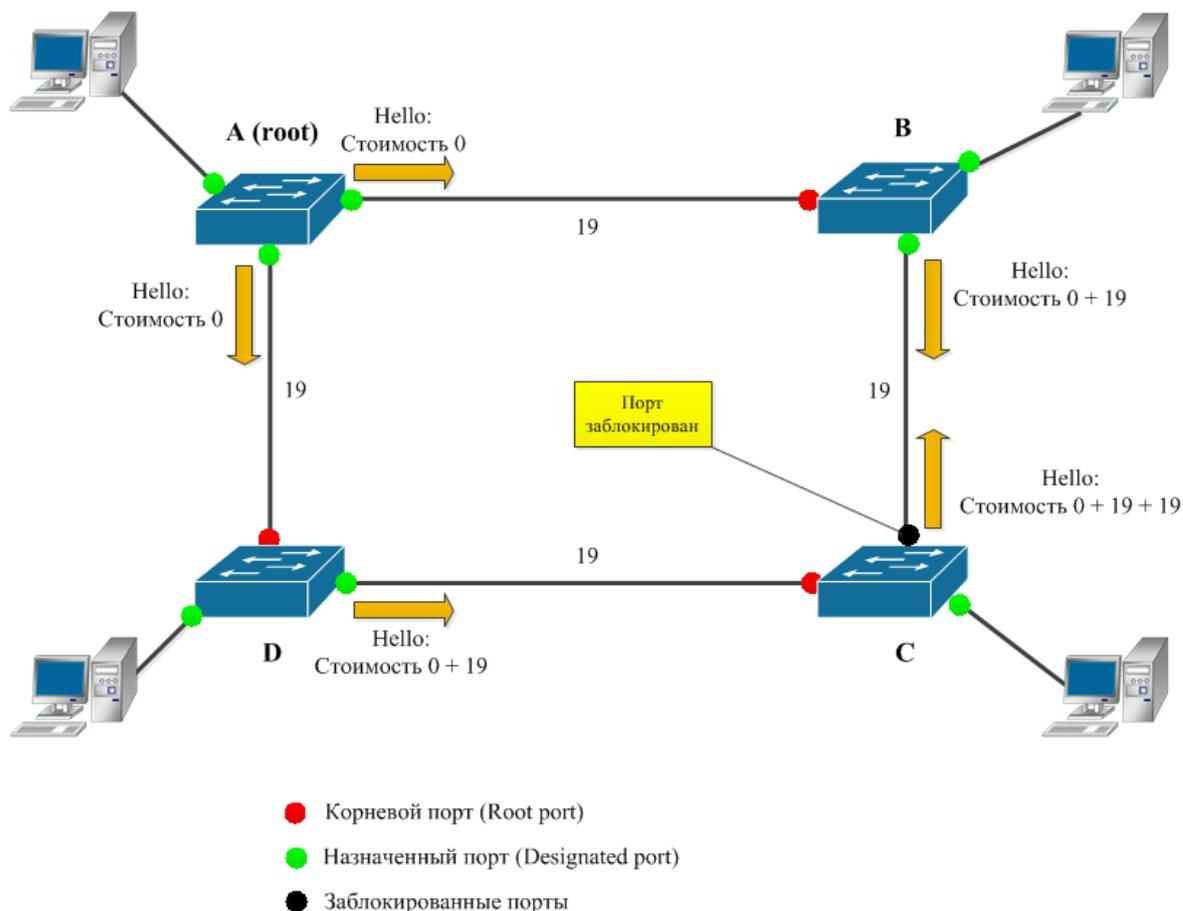


Рисунок 6 – Назначение портов

Чуть раньше мы упомянули состояние блокировки порта, теперь поговорим о том, что это значит, и о других возможных состояниях порта в STP. Итак, в обычном (802.1D) STP существует 5 различных состояний:

**Блокировка (blocking):** заблокированный порт не шлет ничего. Это состояние предназначено, как говорилось выше, для предотвращения петель в сети.

Блокированный порт, тем не менее, слушает BPDU (чтобы быть в курсе событий, это позволяет ему, когда надо, разблокироваться и начать работать)

**Прослушивание (listening):** порт слушает и начинает сам отправлять BPDU, кадры с данными не отправляет.

**Обучение (learning):** порт слушает и отправляет BPDU, а также вносит изменения в CAM- таблицу, но данные не перенаправляет.

**Пересылка (forwarding):** этот может все: и посылает\принимает BPDU, и с данными оперирует, и участвует в поддержании таблицы MAC-адресов. То есть это обычное состояние рабочего порта.

**Отключен (disabled):** состояние *administratively down*, отключен командой *shutdown*.

Порядок перечисления состояний не случаен: при включении, все порты на устройстве с STP проходят вышеприведенные состояния именно в таком порядке (за исключением disabled-портов).

Реакция сети на изменения в топологии

Теперь представим, что порт Fa 2 коммутатора D отключается (на рисунке 6 видно, что он является корневым)

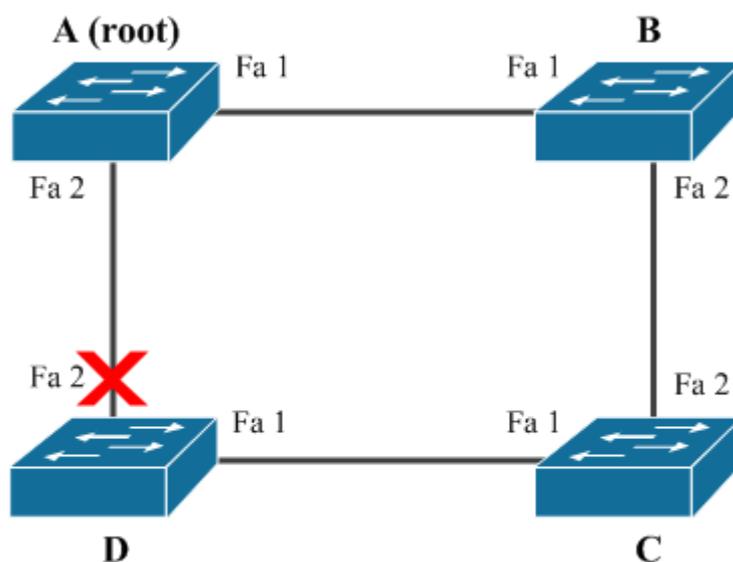


Рисунок 7 – Отключение порта

Коммутатор С перестает получать Hello от корневого коммутатора. Запускается таймер устаревания (Max Age), который равен 20 с. Если в течении работы таймера не поступит Hello от корневого коммутатора, то порт Fa/2 данного коммутатора перейдет в состояние прослушивания. Таблица MAC адресов обнулится. Через 15 с порт перейдет в состояние самообучения. В этом состоянии коммутатор изучает MAC адреса входящих кадров, а также Hello кадры, которые все еще передаются от коммутатора А через коммутатор В.

Так как порт Fa/2 единственный, который подключен к корневному коммутатору, то он переходит в состояние пересылки и становится корневым.

### Контрольные вопросы:

1. Как происходит выбор корневого коммутатора в протоколе STP?

2. Для чего используется протокол STP и каковы последствия для сетевой инфраструктуры при его игнорировании?
3. Что такое BPDU?
4. Для чего нужна отправка Hello кадров?
5. Для чего используется таблица скорости портов в протоколе STP?