

Основы компьютерных сетей. Тема №7. Протокол связующего дерева: STP / Хабр

 habr.com/ru/articles/321132

Денис

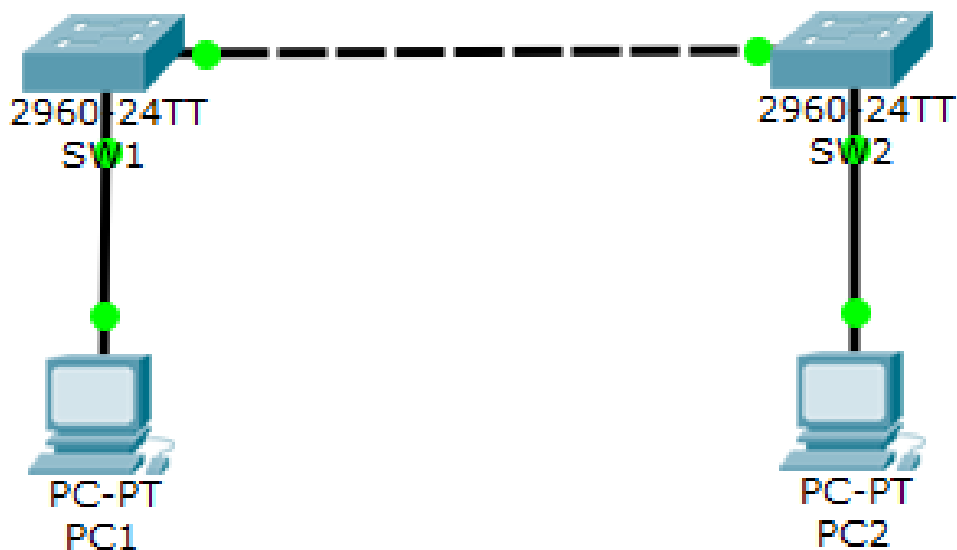
17 апреля 2017 г.



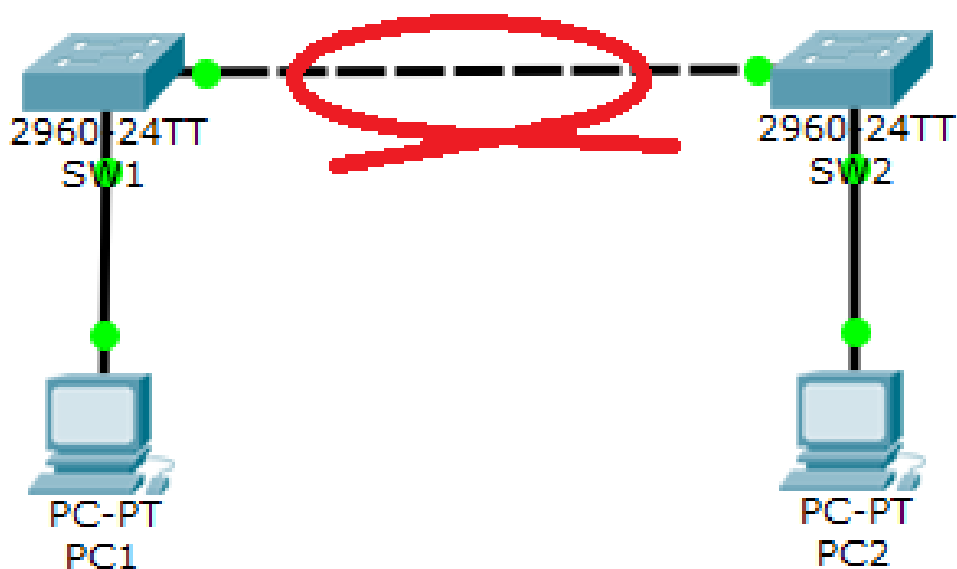
Приветствую на очередной статье по основам компьютерных сетей. Сегодня затронем еще одно семейство протоколов в мире коммутации. И сегодня мы поговорим о протоколах связующего дерева или STP. Узнаем, как это дерево строиться, как можно им управлять, что такое петли, как с ними бороться. Тема интересная, поэтому приглашаю ознакомиться подробнее.

Содержание

Долго думал с чего начать. По-хорошему начинать надо с теории. Но смысл разбирать протокол, когда еще не сталкивался с проблемой, которую этот протокол может решить. Поэтому решил начать с небольшой практики и показать, обо что можно сразу споткнуться. Далее разобраться с этой проблемой и показать, что делать дальше. Соберу самую обычную схему.



Есть 2 компьютера и 2 коммутатора, подключенных друг к другу. Адрес у PC1- 192.168.1.2, а у PC2-192.168.1.3. Компьютеры общаются друг с другом, что-то друг другу отправляют. Но мы замечаем уязвимое место.



Если произойдет обрыв кабеля, то участники останутся без связи. И самая первая мысль, которая приходит в голову — это воткнуть еще один кабель. Но первая мысль не всегда бывает верна. На картинках это не показать, поэтому я покажу это в виде анимации.

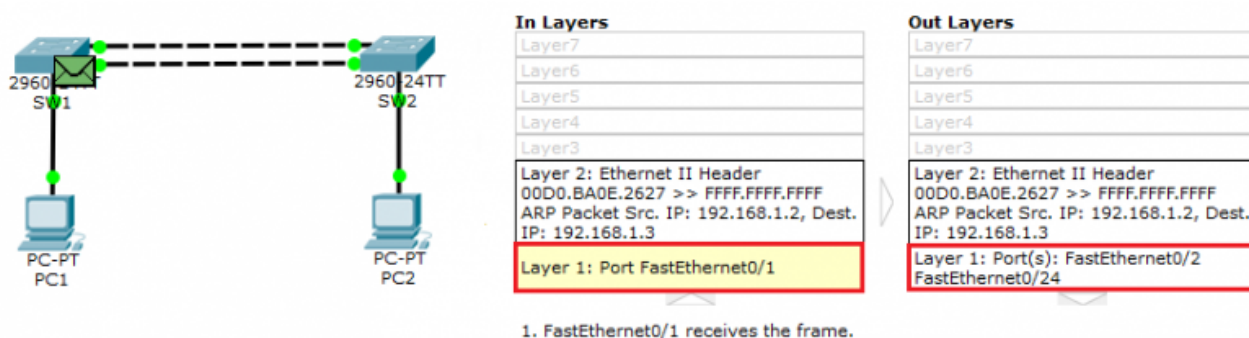
Коммутационная петля

Думаю заметили, как странно и синхронно замигали линки. Это явление зовут

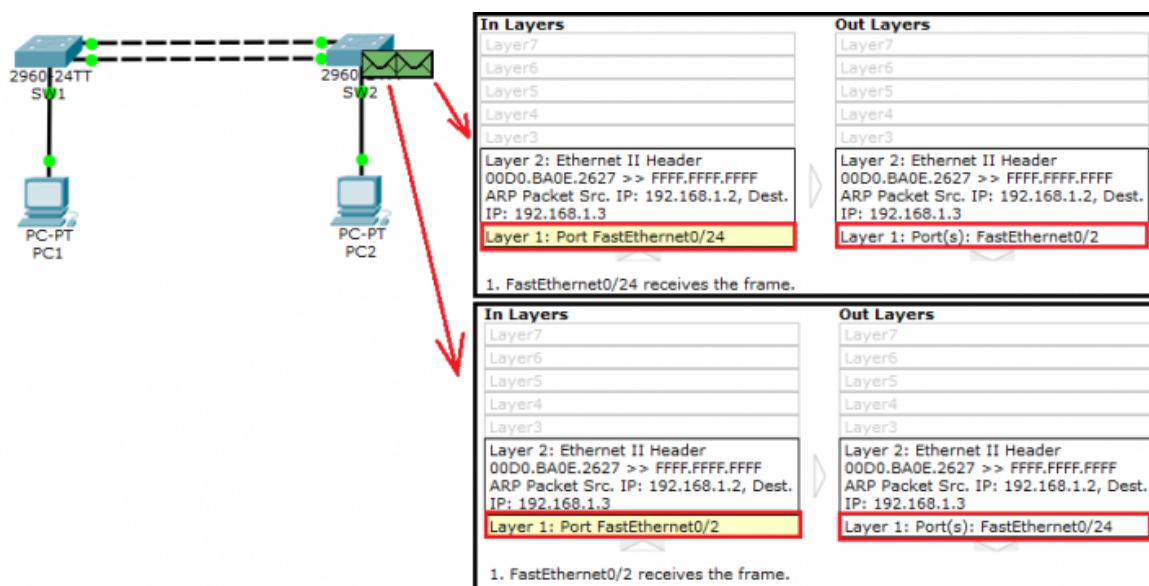
петлей. Чтобы подробнее с ней ознакомиться, необходимо перейти в режим симуляции. Открывайте спойлер ниже и любуйтесь.

Коммутационная петля в режиме симуляции

Объясню подробнее. Итак, PC1 решает отправить пакет ICMP компьютеру PC2. Как правило, перед началом отправки, нужно узнать его MAC-адрес, и он пускает в ход ARP. Вспоминаем, как работают коммутаторы с ARP. Они отправляют его на все порты, кроме исходящего. Что происходит у нас.



Коммутатор, согласно своей логике, отправляет ARP на оба порта (fa0/2 и fa0/24). Но не отправляет его на fa0/1.

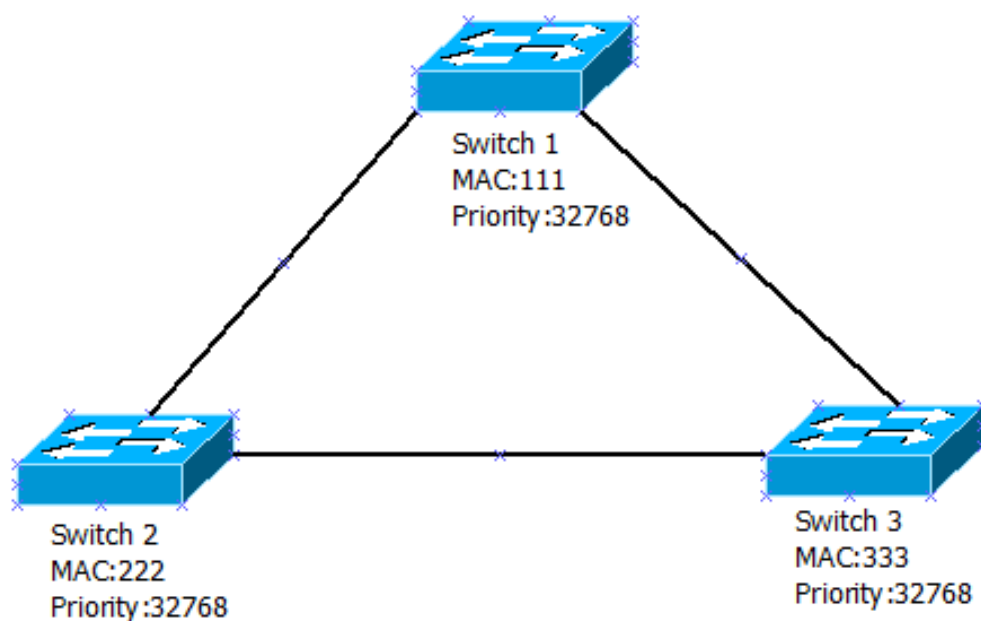


SW2 поступит точно также. Тот ARP, который он получил с порта fa0/24, он отправит на активный порт fa0/2. А второй ARP, полученный с порта fa0/2, отправит на fa0/24. Казалось бы, что мы уже получали с 24-ого порта ARP. Но тут нюанс. Мы получали ARP с другого порта и отдельным ARP сообщением. Поэтому для коммутатора — это 2 разных кадра и обрабатываются они независимо друг от друга. Ну а дальше по аналогии. SW2 отправит один из ARP-ов обратно на SW1, а тот, в свою очередь, обратно SW2. И гулять он будет так до бесконечности, пока не будет выдернут кабель или пока коммутатор не «захлебнется» кадрами и перестанет отвечать. Это

и есть петля. Соответственно, чем больше коммутаторов, тем больше кадров будут создано, что приведет к быстрому отказу сети. Поэтому повышая избыточность соединений, мы повышаем вероятность получения петель. Кому интересно посмотреть на это мерцание у себя на компьютере, качайте [отсюда](#).

Поняли ведущие умы, что это плохо и с этим нужно бороться. Задачу эту возложили на плечи выдающегося инженера Радия Перлман (Radia Joy Perlman) в 1985 году. В чем суть ее технологии. У вас есть N-ое количество коммутаторов, соединенных друг с другом. И перед тем, как передавать пользовательские данные, они ведут переговоры между собой на право стать корневым коммутатором или «root switch». Остальные коммутаторы оставляют включенными только те интерфейсы, которые ведут к корневому коммутатору, а остальные отключают. Тем самым, к каждому коммутатору можно попасть только по одному пути. Разберем этот процесс более подробно.

У нас есть 3 коммутатора, соединенных друг с другом.



Я условно дал им имена и MAC-адреса. На каждом коммутаторе включен классический протокол STP (о других поговорим чуть ниже). Как я говорил выше, им нужно определить, кто из них станет корневым коммутатором. Для этого они начинают обмениваться BPDU-кадрами. Посмотрим, что этот кадр из себя представляет.



Если плохо видно, можно кликнуть по ней и откроется оригинал изображения (открывайте изображение нажатием колеса мышки, либо правой кнопкой по «Открыть ссылку в новой вкладке», чтобы не закрывать саму статью).

Очень много непонятных полей. Ознакомимся с ними и приведем всю эту кашу в порядок.

1. **Идентификатор протокола (Protocol Identifier)** — 2-х байтовое поле, которое всегда равно нулю.
2. **Версия STP протокола (Protocol Version Identifier)** — поле размером в 1 байт, значение которого, всегда равно «0».
3. **Тип BPDU (BPDU type)** — 1 байт, которые принимает значение «0», если это конфигурационный BPDU (CBPDU), или «1», если это TCN BPDU.
 - **CBPDU (Configuration Bridge Protocol Data Unit)** — кадр, используемый для вычисления связующего дерева. То есть, когда значение = 0.
 - **TCNBPDU (Topology Change Notification Bridge Protocol Data Unit)** — кадр, используемый, чтобы уведомить других о изменениях в топологии. То есть, когда значение = 1. Проще говоря, если коммутатор видит, что произошло, какое-то изменение в топологии (линк отвалился, «умер» сосед и т.д.), он пускает BPDU со значением «1» в поле BPDU Type. А дальше работают кадры со значением «0», чтобы заново перестроить дерево.
4. **Флаги (Flags)** — в этом поле используются только 1 байт. Эти флаги используются при изменении топологии (бит «1») и при подтверждении топологии (бит «8»).
5. **Идентификатор корневого моста (Root Identifier)** — в этом поле содержится информация о корневом коммутаторе, а именно его приоритет и MAC-адрес.

6. **Расстояние до корневого моста (Root Path Cost)** — здесь содержится суммарная стоимость до корневого коммутатора. Приведу табличку, где указана стоимость.

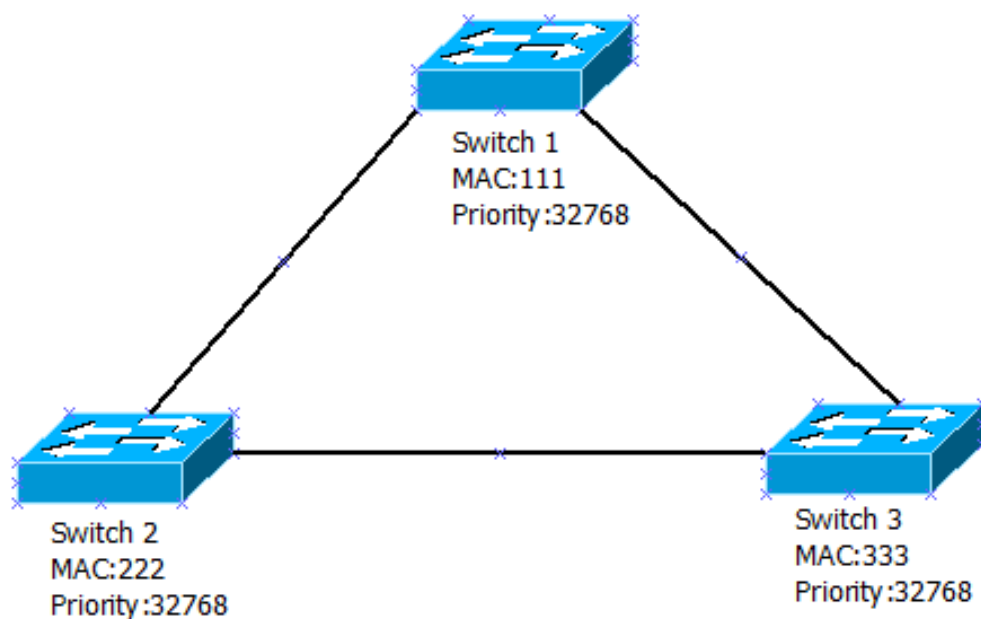
Скорость канала	Стоимость
10 Гбит/с	2
1 Гбит/с	4
100 Мбит/с	19
10 Мбит/с	100

Эта таблица измененная. Раньше стоимость считалась по другому, но с приходом новых стандартов скорости, предыдущая спецификация утратила свою силу.

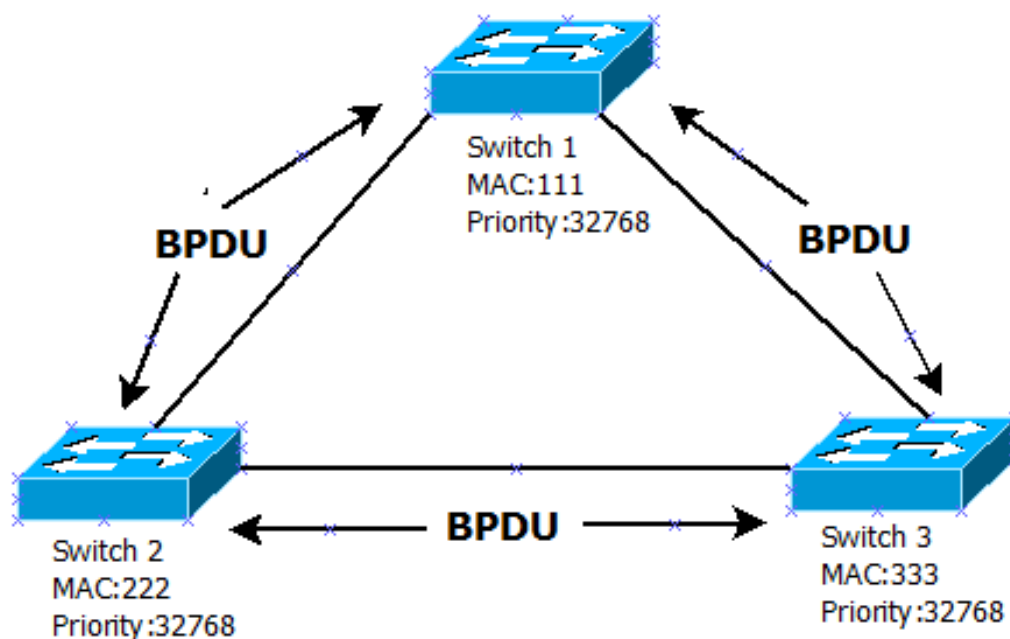
7. **Идентификатор моста (Bridge Identifier)** — сюда коммутатор-отправитель записывает свои данные (приоритет + MAC-адрес).
8. **Идентификатор порта (Port Identifier)** — сюда коммутатор-отправитель записывает идентификатор порта (то есть тот, с которого этот BPDU выйдет).
9. **Время жизни сообщения (Message Age)** — здесь содержится временной интервал (в секундах). Он нужен для того, чтобы распознать устаревшие кадры и отбросить. Его формирует корневой коммутатор и устанавливает в первоначальное значение «0». Далее каждый последующий коммутатор увеличивает это значение на время задержки. Как только это значение превысит максимальное пороговое значение, оно будет отброшено.
10. **Максимальное время жизни сообщения (Max Age)** — это поле отвечает, как раз, за максимальное время жизни. Превысив его, коммутатор отбрасывает кадр.
11. **Время приветствия (Hello Time)** — Временной интервал, через который коммутатор посылает BPDU кадры. По-умолчанию — это 2 секунды.
12. **Задержка смены состояний (Forward Delay)** — временной интервал, указывающий сколько секунд порт коммутатора будет находиться в состоянии **прослушивания и обучения**.

Это стандартный набор BPDU в STP. В зависимости от версии, поля могут называться иначе, но принцип работы у них един. Сам кадр большой и с ходу может не сразу все уложиться в голове. Это нормальное явление. Знать все поля наизусть не обязательно. Главное, что нужно твердо понимать в рамках CCNA — это поля 5, 6, 7 и 8. Поэтому переходим к разбору работы протокола STP.

Во многих изданиях «цисковских» и сторонних, работу STP показывают на примере 3 коммутаторов, соединенных между собой. Не буду отходить от традиции и сделаю аналогично.



Условно я дал им имена и MAC-адреса, чтобы не засорять голову длинной адресацией. Двигаемся дальше.



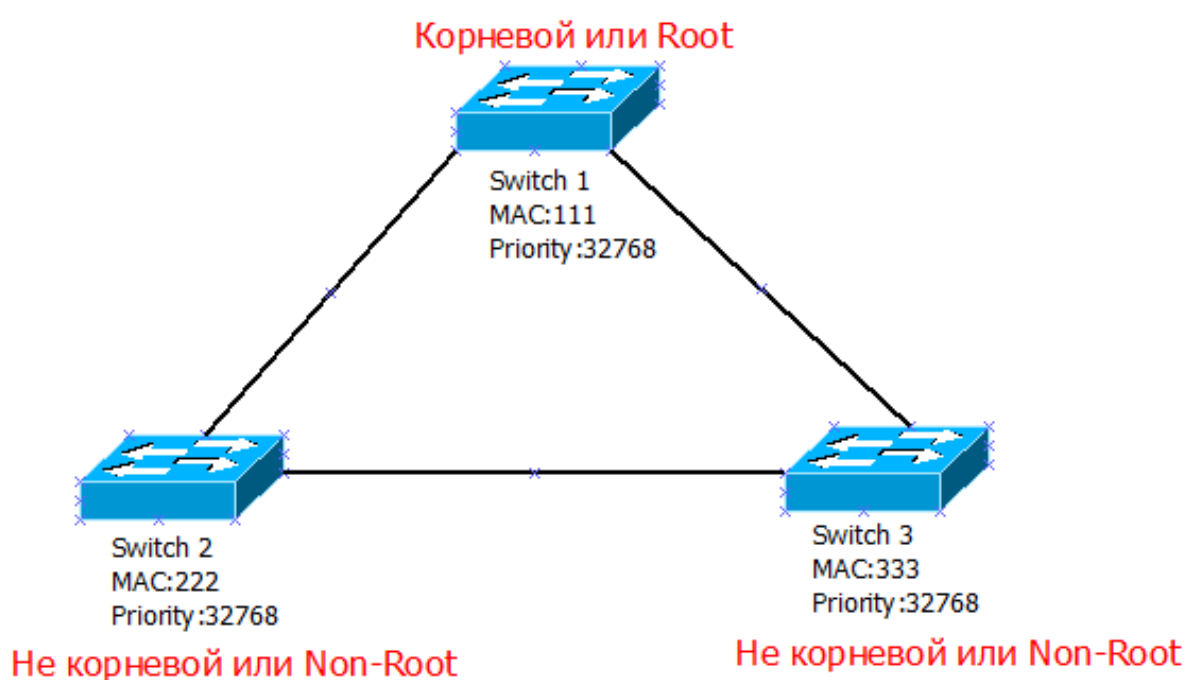
И так как на коммутаторах работает протокол STP, им нужно выбрать того, кто будет главным в топологии или **корневым (root)**. Для этого, они начинают обмениваться BPDU-кадрами. Вот тут как раз важны поля 5, 6 и 7. Я специально хочу остановиться на них. Изначально коммутаторы в поле 5 (Идентификатор корневого моста или Root Identifier) начинают записывать свой «приоритет + MAC-адрес». Если вручную ничего не менять, то приоритет равен 32678. Дальше коммутатор, который получит этот кадр от соседа, будет сравнивать свой «Root Identifier» с вновь прибывшим. Если он увидит, что у соседа этот Root ID ниже, то с этого момента он

будет ретранслировать его BPDU. В результате в сети останется только один коммутатор, который будет генерировать BPDU.

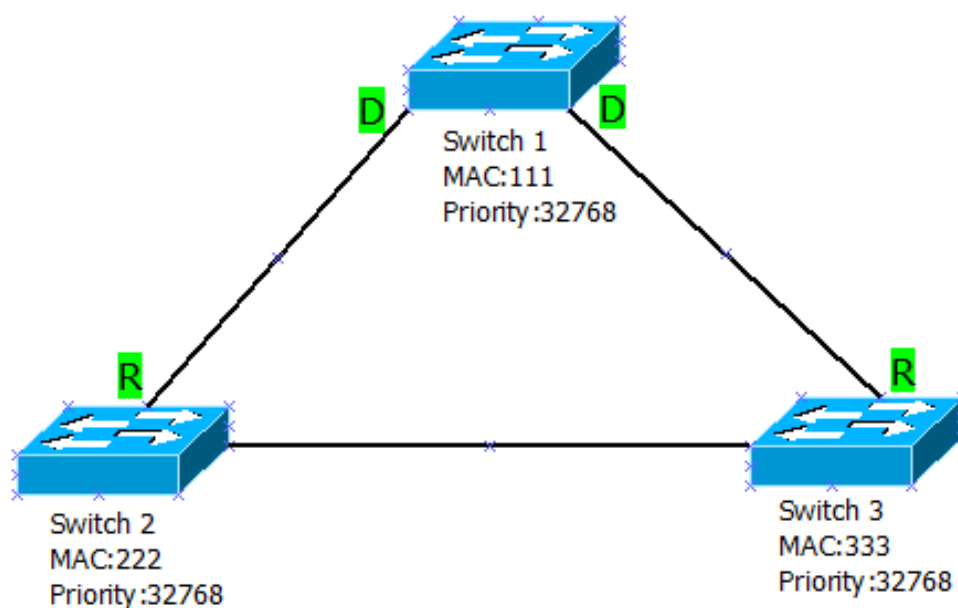
В поле 6 «Root Path Cost» коммутатор запишет стоимость пути. При создании BPDU, корневого коммутатора записывает туда 0, так как это он и есть. А вот следующие коммутаторы уже начинают суммировать стоимость по таблице, представленной выше.

Ну и в поле 7 «Bridge Identifier» записывается связка «приоритет + MAC-адрес» самого коммутатора. То есть, если в «Root Identifier» всегда записывается связка корневого коммутатора, то в это поле, он всегда записывает свою. То есть при ретрансляции BPDU от соседа к соседу, коммутаторы сюда дописывают свой Bridge ID.

Скажу пару слов о связке «приоритет + MAC-адрес». Они ни в коем случае не суммируются. Знак "+" я вставил в том контексте, что они всегда работают вместе. Сначала коммутаторы, при проведении выборов, смотрят на **приоритет**. И если приоритеты равны (а по-умолчанию они равны), то начинает опираться на MAC-адреса. И тот, у кого MAC-адрес меньше, становится главным, корневым или root. Называйте как вам удобно. Вот приоритет нужен как раз для того, чтобы административно влиять на выбор корневого коммутатора. Представьте ситуацию, что у вас есть 2 коммутатора. Один из них новый и производительный, а второй старый, древний и в скором времени пойдет под списание. И тут выясняется, что у старого коммутатора MAC-адрес меньше, чем у нового коммутатора, а значит, при равных приоритетах, выигрывать всегда будет старый коммутатор. Вот для решения такой спорной задачи и нужен приоритет. Причем, когда вы меняете приоритет, он обязан быть кратным 4096 (то есть 32768, 28672, 24576 и так далее). Возвращаемся к схеме.

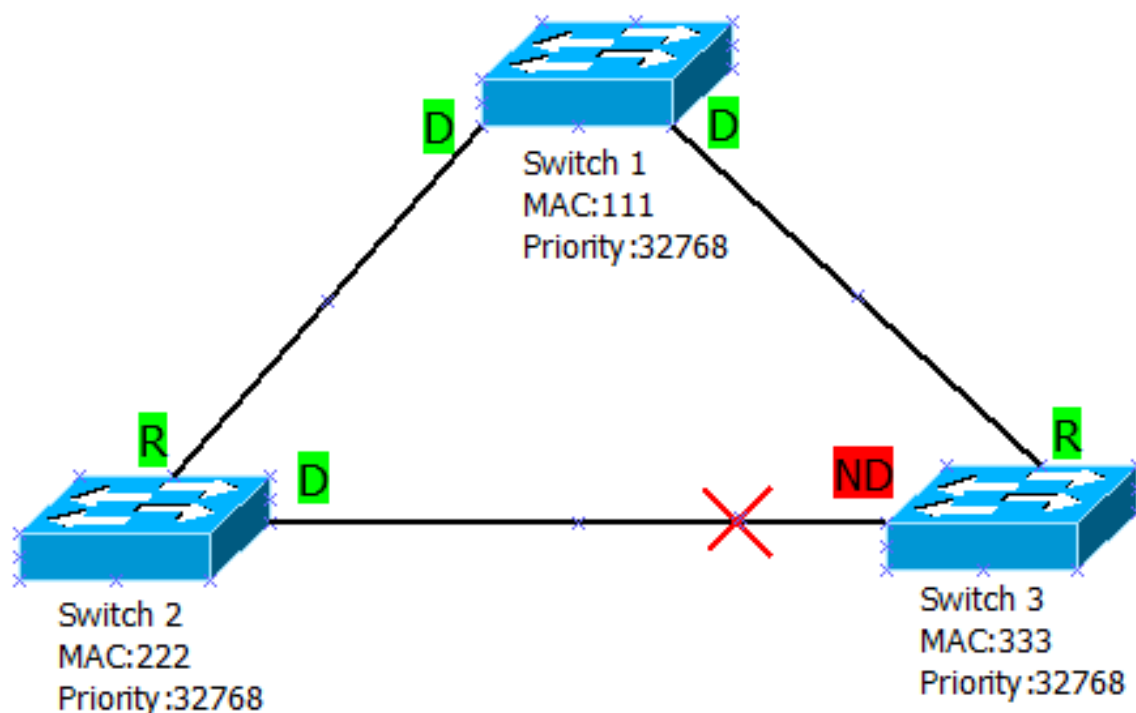


Ну и так как приоритеты у трех коммутаторов одинаковые, то выборы они начинают по MAC-адресам. Наименьший MAC-адрес у Switch 1 => он становится корневым.



Раз Switch 1 становится корневым, то он сразу переводит все свои интерфейсы в режим «Designated». То есть это порт, который имеет самый короткий путь до корневого коммутатора (в данном случае до самого себя).

Дальше Switch 2 и Switch 3 должны решить для себя, какой порт будет корневым. То есть тот порт, который имеет наименьшую стоимость пути до корневого коммутатора. Здесь все очевидно. Если вдруг получится, что стоимость по нескольким портам будет одинаковая, то он выберет порт с наименьшим порядковым номером или именем. Например, из портов fa0/1, fa0/2 и fa0/3, будет выбран fa0/1.



Root-порты определены, но что делать с линком между Switch 2 и Switch 3, ведь он может создать петлю?! Для ее предотвращения они договариваются, кто из них отключит свой порт.

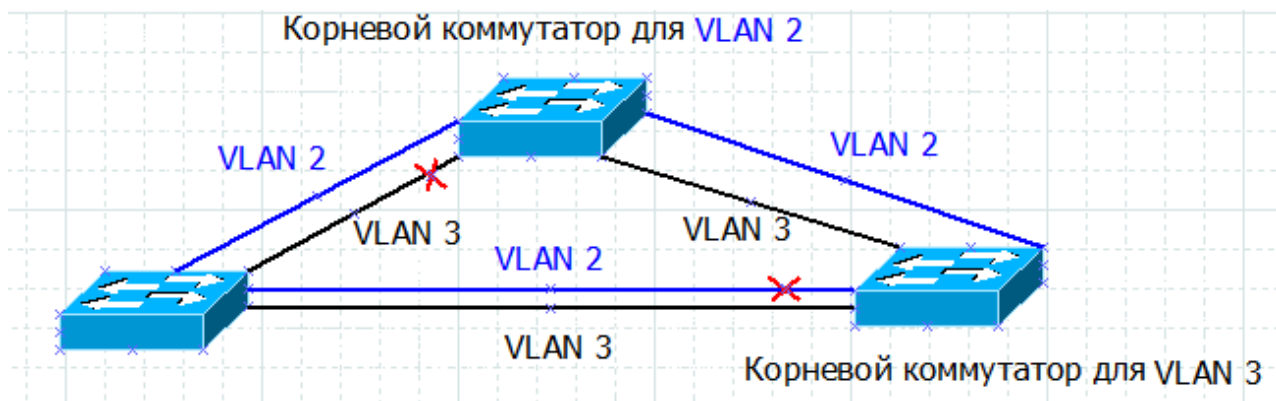
Договариваться они будут также по Bridge ID. Приоритеты равны, поэтому смотрим по MAC-адресам. У Switch 2 MAC-адрес меньше, поэтому он переводит порт в режим «Designated», а Switch 3 в режим «Non-Designated». «Non-Designated» — такой режим, при котором порту запрещено передавать какие-либо данные, но разрешено слушать, что происходит в сети. То есть, если отвалится какой-то линк, он может включиться и полноценно работать.

Помимо ролей, у портов есть состояния, которые они должны пройти в обязательном порядке. Объясню на примере построенной топологии. Вот у нас построено выше дерево STP. Петель нет и все замечательно. Один из портов коммутатора Switch 3 находится в состоянии **Blocking**. Вот он слушает BPDU и никого не трогает. Но если вдруг отвалится где-то линк или произойдет изменение топологии, он сразу переходит в состояние **Listening** или **Прослушивание**. В этом состоянии он отправляет, слушает только BPDU кадры и обрабатывает полученную информацию. Если он видит, что у соседей параметры хуже, чем у него, то по истечении 15 секунд, переходит в следующее состояние **Learning** или **Обучение**. Эта фаза длится также 15 секунд. В «Learning» порт делает практически все тоже самое, что и в предыдущем состоянии, за исключением того, что теперь строит таблицу коммутации на основании полученных кадров. Если по истечении 15 секунд, он не получит BPDU с параметрами лучше, чем у него, то перейдет в последнее состояние **Forwarding** или **Продвижение**. Это такое финальное и

полноправное состояние. Он обменивается не только служебной информацией, но и пользовательскими данными. То есть переход из состояния **Listening** в **Forwarding** длится 30 секунд.

Есть еще состояние **Disable** или **Отключен**, когда вручную отключаете порт, но я не считаю, что это состояние STP. В этом состоянии передаваться ничего не будет. Это, грубо говоря, физическое отключение порта.

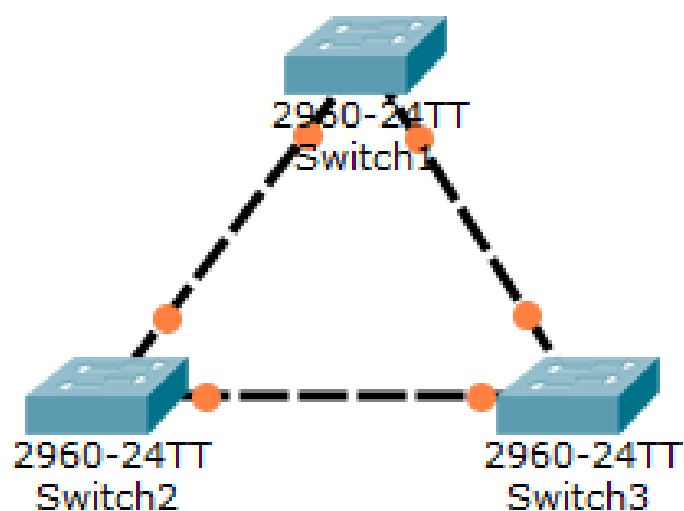
Вышепоказанный пример — это работа классического протокола STP, который еще называют **CST (Classic Spanning Tree)**. Одним из его минусов — это то, что он строит одно единственное дерево для всей топологии. А учитывая, что появились VLAN-ы, то нужно было модифицировать этот протокол под них. Cisco, как пионер, выпустила протокол **PVST (Per-VLAN Spanning Tree)**. Он позволял строить отдельное дерево для каждого VLAN. Единственное, что он работал с ISL (проприетарный цисковский протокол, работающий с тегированными кадрами), который применялся только на устройствах данного производителя. Но с появлением открытого протокола 802.1q, они быстренько модернизировали **PVST** и дали ему имя **PVST+**. Работает он также, как и его предшественник, но с 802.1q. Нарисую схему и объясню более подробно.



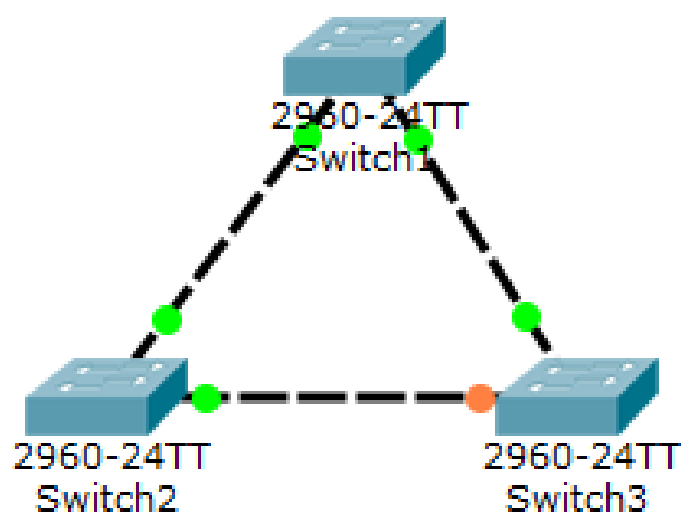
Вот, к примеру, у меня есть 2 VLAN-а. И для каждого VLAN-а, протокол PVST+ строит отдельное дерево. В принципе — это его отличие от CST. Выборы и переходы проходят аналогично и с тем же интервалом по времени. К сожалению, или к счастью, современные Cisco-коммутаторы уже не поддерживают CST.

Поэтому попрактикуемся на PVST+. Тем более, что, при работе сети в одном VLAN-е (который является VLAN-ом по-умолчанию), он мало чем будет отличаться от классического STP.

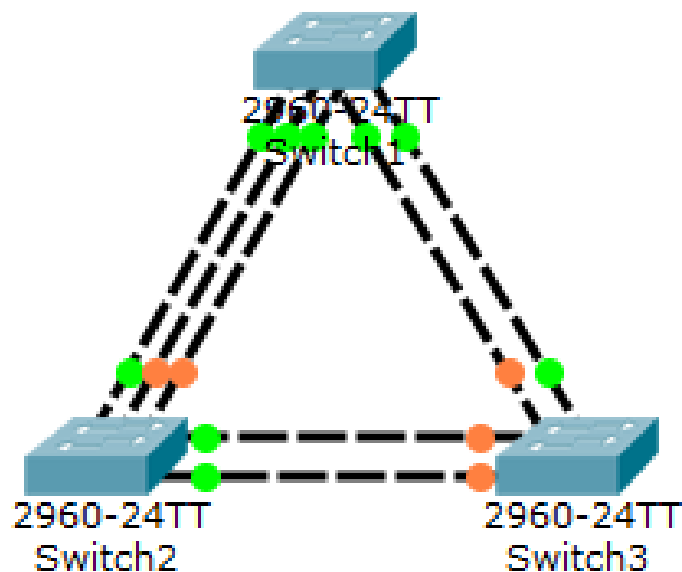
Я уже быстренько собрал лабораторку из 3-х коммутаторов и сейчас все наглядно покажу.



И вот как только коммутаторы прошли все стадии, образуется STP-дерево.

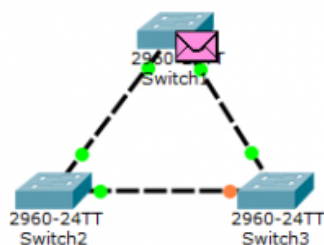


Я думаю вы заметили, что один из портов коммутатора Switch3, горит оранжевым. Это означает, что данный порт находится в состоянии **Blocking**. Не путайте с **Disabled**. То есть он не касается пользовательского трафика, но слушает, что происходит в сети. И не важно сколько мы воткнем кабелей. В топологии всегда будет отработывать STP и закрывать петли.



Собственно, что и показано на рисунке.

Теперь покажу, что происходит с коммутаторами, когда дерево уже построено. По логике STP, корневой коммутатор должен отправлять Hello-кадр «подчиненным» коммутаторам с интервалом времени в 2 секунды.



Ethernet 802.3

0	4	7	8	14	19	Bytes
PREAMBLE: 1010 1010		S F D	DEST ADDR: 0180.C200.0000	SRC ADDR: 0060.70B9.0901		
LENGTH / TYPE: 0x3		DATA (VARIABLE LENGTH)			FCS: 0x0	

LLC

0	8	16	24 Bits
DSAP:0x42	SSAP:0x42	CONTROL BIT: 3	

STP BPDU

0		4		8		10		16		24		31Bits	
PROTOCOL ID: 0								VERSION: 0				MESSAGE TYPE: 0	
T C	P R	P O R T R O L E	L F A C E										
ROOT ID: 32769 / 0009.7C94.0557													
ROOT PATH COST : 0													
BRIDGE ID: 32769 / 0009.7C94.0557													
PORT ID: 32769								MESSAGE AGE: 0					
MAX AGE: 20								HELLO TIME: 2					
FORWARD DELAY: 15													

Что он из себя представляет, вы видите на картинке выше. Прошу обратить внимание на поля кадра Ethernet 802.3. А именно «Source MAC-Address» и

«Destination MAC-Address». В «Source MAC-Address» он записывает MAC-адрес своего порта (в данном случае FastEthernet 0/1). А в «Destination MAC-Address» мультикастовый адрес «0180.C200.0000», который посылается всем участникам, знающим, что такое STP и работающим с ним. Ну и сам кадр STP BPDU. Тут куча полей. Но заострю внимание на более важных, которые я отметил красным прямоугольником.

- **Root ID** — 32769 / 0009.7C94.0557. Это, как раз, та самая связка «приоритет + MAC-адрес». Но тут интересный момент. Если в классическом STP приоритет по-умолчанию равнялся 32768, то здесь мы видим 32769. В протоколе PVST+ и других, умеющих работать с VLAN, к приоритету добавляется параметр **system ID extension**. Этот параметр содержит в себе номер VLAN-а и по нему коммутатор понимает, к какому процессу STP его отнести. То есть в данном случае у нас VLAN №1 => приоритет = 32768 + 1 = 32769. Если бы мы настраивали для 10-ого VLAN-а, то приоритет равнялся 32778. Ну а после дробной черты, сам MAC-адрес интерфейса.
- **Root Path Cost** — стоимость пути. Мы изучаем кадр, когда он выходит из корневого коммутатора => там стоит 0.
- **Bridge ID** — идентификатор коммутатора, который ретранслирует данный BPDU. В данный момент он такой же, как и Root ID.
- **Port ID** — идентификатор порта. Такой же, как и приоритет — 32769.
- **Message Age** — интервал времени. Так как BPDU «свежий», то там стоит 0.
- **Max Age** — максимальное время жизни — 20 секунд.
- **Hello Time** — интервал послышки приветствия — 2 секунды.
- **Forward Delay** — указывает сколько секунд находиться в одной фазе (прослушивания или обучения) — 15 секунд.

В принципе здесь ничего нового и все это мы разбирали выше. Я это показал для того, чтобы вы понимали, для чего я так долго грузил сухим текстом.

Мы уже знаем, кто является корневым коммутатором и какой порт заблокирован для устранения петли. Но на экзамене и в повседневной жизни мы будем оперировать командами, при помощи которых можно будет узнать, кто в сегменте является корневым, у кого заблокирован порт и прочую информацию. Начнем с коммутатора **Switch1** и с самой важной команды **«show spanning-tree»**. Ее важно запомнить.

```
Switch1#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0009.7C94.0557
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
             Address     0009.7C94.0557
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Desg	FWD	19	128.1	P2p
Fa0/2	Desg	FWD	19	128.2	P2p

Данная команда выводит информацию о всех процессах STP (то есть за каждый VLAN), в которых участвует коммутатор. В нашем случае всего один VLAN. Теперь поговорим о том, что означают эти письма.

Первое, что бросается в глаза — это блок **Root ID**.

```
Root ID    Priority    32769
           Address     0009.7C94.0557
           This bridge is the root
           Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
```

Он содержит информацию о приоритете, MAC-адресе и таймерах корневого коммутатора. Здесь красуется еще одна важная строчка «This bridge is the root». Она говорит о том, что именно этот коммутатор является корневым за данный VLAN. Поэтому, если вам надо будет найти корневой коммутатор, то ищите эту надпись. На соседнем коммутаторе (не являющимся корневым) этой строчки не будет.

Следующий блок — **Bridge ID**.

```
Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
           Address     0009.7C94.0557
           Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
           Aging Time  20
```

Здесь, соответственно, информация о текущем коммутаторе. На корневом коммутаторе этот блок идентичен вышестоящему.

Ну и ниже располагается таблица.

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Desg	FWD	19	128.1	P2p
Fa0/2	Desg	FWD	19	128.2	P2p

В ней записаны интерфейсы, относящиеся к данному VLAN-у, их роли, статусы и прочее. Остановимся немного на ней.

Так как это корневой коммутатор, то порты автоматически переводятся в роль «Designated».

Статус «Forwarding» говорит о том, что порты прошли все стадии и сейчас находятся в активном режиме (пересылка).

Дальше идет стоимость, и она равна 19. FastEthernet работает на скорости 100 Мбит/с и для этой скорости стоимость равна 19 (выше приведена табличка).

Следом идет колонка Prio.Nbr или Priority Number. Это приоритет порта. По умолчанию этот параметр равен 128, а после точки записывается порядковый номер порта. Соответственно для Fa0/1 — это 128.1, а для Fa0/2 — 128.2.

Тип «p2p» говорит о том, что порт коммутатора работает в режиме «full-duplex». Это означает, что порт может одновременно вести и передачу, и прием.

Если же там будет указан «shared», то это будет означать, что порт работает в режиме «half-duplex». То есть он либо передает, либо получает (не одновременно).

Перейдем к следующему коммутатору **Switch2**. Аналогично введу команду «show spanning-tree» и посмотрю, что он покажет.

```
Switch2#show spanning-tree
```

```
VLAN0001
```

```
Spanning tree enabled protocol ieee
```

```
Root ID    Priority    32769
Address     0009.7C94.0557
Cost        19
Port        1(FastEthernet0/1)
Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
```

```
Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
Address     00D0.9776.B743
Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
Aging Time  20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	19	128.1	P2p
Fa0/2	Desg	FWD	19	128.2	P2p

Обратите внимание на блок **Root ID**.

```
Spanning tree enabled protocol ieee
```

Root ID	Priority	32769
	Address	0009.7C94.0557
	Cost	38
	Port	1(FastEthernet0/1)
	Hello Time	2 sec Max Age 20 sec Forward Delay 15 sec

Как говорилось ранее, здесь содержится информация о корневом коммутаторе. Но здесь уже нет надписи «This bridge is the root», так как этот коммутатор не корневой. Но есть другая запись Port. В ней указан порт, ведущий на корневой коммутатор, и это FastEthernet0/1. Выше есть строчка Cost и она равна 19. Не путайте эту строчку Cost с такой же строчкой в таблице интерфейсов ниже. Если в таблице интерфейсов стоимость указана за конкретный порт, то здесь записывается суммарная стоимость до корневого коммутатора. Например, если за коммутатором Switch2 будет еще один коммутатор с интерфейсом FastEthernet, то его стоимость будет выше.

```
Spanning tree enabled protocol ieee
```

Root ID	Priority	32769
	Address	0009.7C94.0557
	Cost	38
	Port	1(FastEthernet0/1)
	Hello Time	2 sec Max Age 20 sec Forward Delay 15 sec

То есть он сложит стоимость своего интерфейса со стоимостью интерфейса соседа. Двигаемся дальше и натываемся на блок **Bridge ID**. Сюда он записывает информацию о себе. Можете заметить, что MAC-адреса отличаются. Далее идут таймеры. Это важный показатель и старайтесь про него не забывать. Лучше его не менять. Но, если все-таки появилась нужда это сделать, то меняйте и на соседних коммутаторах. Иначе это может привести к серьезным ошибкам и займет не мало времени на устранение.

Таблица интерфейсов отличается от корневого коммутатора тем, что роль FastEthernet0/1 не «Designated», а «Root». То есть этот порт ведет к корневому коммутатору.

Остался последний коммутатор **Switch3**

```
Switch3#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
            Address     0009.7C94.0557
            Cost        19
            Port        1(FastEthernet0/1)
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
            Address     00D0.BA84.7C38
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time  20

Interface                Role Sts Cost      Prio.Nbr Type
-----
Fa0/1                    Root FWD 19        128.1    P2p
Fa0/2                    Altn BLK 19        128.2    P2p
```

Здесь конфигурация аналогичная, за исключением порта FastEthernet0/2.

Fa0/2 **Altn BLK 19** **128.2** **P2p**

Он в роли **Alternate**. То есть, в качестве запасного. А статус **Blocking** говорит о том, что порт заблокирован, дабы «оборвать» петлю. Вот принцип работы классического STP. Прикладываю [ссылку](#) на скачивание данной лабораторки.

Но данный вид уже не очень актуален, так как вы не встретите серьезную организацию, у которой всего один VLAN. Соответственно, наша задача подружить STP с VLAN.

Поэтому создаем VLAN-ы на каждом коммутаторе. Можно, конечно, включить VTP и они автоматически синхронизируются, но я не сторонник этого протокола. Поэтому в блокноте подготовил шаблон команд, которые вставляю на каждый коммутатор.

Список команд

И теперь проверю, что получилось на выходе командой «show spanning-tree».


```
Switch1#show spanning-tree
```

```
VLAN0001
```

```
Spanning tree enabled protocol ieee
```

```
Root ID      Priority    32769
Address      0009.7C94.0557
This bridge is the root
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
```

```
Bridge ID   Priority    32769 (priority 32768 sys-id-ext 1)
Address      0009.7C94.0557
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
Aging Time   20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Desg	FWD	19	128.1	P2p
Fa0/2	Desg	FWD	19	128.2	P2p

```
VLAN0002
```

```
Spanning tree enabled protocol ieee
```

```
Root ID      Priority    32770
Address      0009.7C94.0557
This bridge is the root
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
```

```
Bridge ID   Priority    32770 (priority 32768 sys-id-ext 2)
Address      0009.7C94.0557
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
Aging Time   20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Desg	FWD	19	128.1	P2p
Fa0/2	Desg	FWD	19	128.2	P2p

```
VLAN0003
```

```
Spanning tree enabled protocol ieee
```

```
Root ID      Priority    32771
Address      0009.7C94.0557
This bridge is the root
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
```

```
Bridge ID   Priority    32771 (priority 32768 sys-id-ext 3)
Address      0009.7C94.0557
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
Aging Time   20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Desg	FWD	19	128.1	P2p
Fa0/2	Desg	FWD	19	128.2	P2p

Получилось длинное полотно текста, в котором описан процесс STP для каждого

VLAN-а. Если внимательно посмотреть, то можно увидеть, что Switch1 является корневым для каждого VLAN-а. Но не всегда так бывает нужно.

Сейчас объясню. Например, у нас есть Switch3, который блокирует порт для устранения петли. Давайте взглянем на его обновленную конфигурацию.

Switch3#show spanning-tree

VLAN0001

Spanning tree enabled protocol ieee

Root ID Priority 32769
 Address 0009.7C94.0557
 Cost 19
 Port 1(FastEthernet0/1)
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
 Address 00D0.BA84.7C38
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 20

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	19	128.1	P2p
Fa0/2	Altn	BLK	19	128.2	P2p

VLAN0002

Spanning tree enabled protocol ieee

Root ID Priority 32770
 Address 0009.7C94.0557
 Cost 19
 Port 1(FastEthernet0/1)
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32770 (priority 32768 sys-id-ext 2)
 Address 00D0.BA84.7C38
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 20

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	19	128.1	P2p
Fa0/2	Altn	BLK	19	128.2	P2p

VLAN0003

Spanning tree enabled protocol ieee

Root ID Priority 32771
 Address 0009.7C94.0557
 Cost 19
 Port 1(FastEthernet0/1)
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32771 (priority 32768 sys-id-ext 3)
 Address 00D0.BA84.7C38
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 20

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	19	128.1	P2p
Fa0/2	Altn	BLK	19	128.2	P2p

Видим, что он блокирует интерфейс FastEthernet0/2 во всех 3-х VLAN-ах. И вот возникла ситуация, что нужно сделать Switch3 корневым коммутатором для VLAN 3. Как описывалось ранее, на помощь придет игра с приоритетом. Сейчас он равен 32771 (32786 + 3). Мне надо его уменьшить. Сделать это можно несколькими способами. Первый способ — это задать приоритет вручную. Захожу на Switch 3 и пишу:

```
Switch3(config)#spanning-tree vlan 3 priority 30000
% Bridge Priority must be in increments of 4096.
% Allowed values are:
   0      4096  8192  12288  16384  20480  24576  28672
 32768  36864  40960  45056  49152  53248  57344  61440
```

Я решил задать приоритет 30000, так как он меньше 32768. Да, обратите внимание, что мы меняем именно приоритет без sys-id-ext. Но после ввода, выходит сообщение, что нужно ввести число кратное 4096. И ниже предлагает допустимый приоритет. Можно ввести одно из предложенных значений и приоритет изменится.

Но я покажу другой способ изменения приоритета.

```
Switch3(config)# spanning-tree vlan 3 root primary
```

При вводе этой команды, коммутатор смотрит, какой Bridge ID был у корневого коммутатора и меняет его на меньшее значение. Только отнимает он не 4096, а 8192. То есть делает меньше на 2 порядка. Я введу эту команду и посмотрю, что изменится.

Switch3#show spanning-tree

VLAN0001

Spanning tree enabled protocol ieee

Root ID Priority 32769
 Address 0009.7C94.0557
 Cost 19
 Port 1(FastEthernet0/1)
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
 Address 00D0.BA84.7C38
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 20

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	19	128.1	P2p
Fa0/2	Altn	BLK	19	128.2	P2p

VLAN0002

Spanning tree enabled protocol ieee

Root ID Priority 32770
 Address 0009.7C94.0557
 Cost 19
 Port 1(FastEthernet0/1)
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32770 (priority 32768 sys-id-ext 2)
 Address 00D0.BA84.7C38
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 20

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Root	FWD	19	128.1	P2p
Fa0/2	Altn	BLK	19	128.2	P2p

VLAN0003

Spanning tree enabled protocol ieee

Root ID Priority 24579
 Address 00D0.BA84.7C38
 This bridge is the root
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 24579 (priority 24576 sys-id-ext 3)
 Address 00D0.BA84.7C38
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 20

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/1	Desg	FWD	19	128.1	P2p
Fa0/2	Desg	FWD	19	128.2	P2p

И вижу, что секция VLAN 3 изменилась. Теперь там приоритет 24579 (24576 + 3) и

красуется строчка «This bridge is the root», указывающая, что данный коммутатор теперь корневой для VLAN 3. Оба порта в роли «Designated» и статусе «Forward» (что верно для корневого коммутатора). Но две верхних секции с VLAN-ами остались без изменения и для них FastEthernet 0/2 останется по-прежнему заблокированным.

Теперь посмотрим, как отреагировал Switch 1 на то, что у него забрали корону.

```
Switch1#show spanning-tree
```

```
VLAN0001
```

```
Spanning tree enabled protocol ieee
```

```
Root ID      Priority    32769
Address      0009.7C94.0557
This bridge is the root
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
```

```
Bridge ID    Priority    32769 (priority 32768 sys-id-ext 1)
Address      0009.7C94.0557
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
Aging Time   20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/2	Desg	FWD	19	128.2	P2p
Fa0/1	Desg	FWD	19	128.1	P2p

```
VLAN0002
```

```
Spanning tree enabled protocol ieee
```

```
Root ID      Priority    32770
Address      0009.7C94.0557
This bridge is the root
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
```

```
Bridge ID    Priority    32770 (priority 32768 sys-id-ext 2)
Address      0009.7C94.0557
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
Aging Time   20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/2	Desg	FWD	19	128.2	P2p
Fa0/1	Desg	FWD	19	128.1	P2p

```
VLAN0003
```

```
Spanning tree enabled protocol ieee
```

```
Root ID      Priority    24579
Address      00D0.BA84.7C38
Cost         19
Port         1(FastEthernet0/1)
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
```

```
Bridge ID    Priority    32771 (priority 32768 sys-id-ext 3)
Address      0009.7C94.0557
Hello Time   2 sec    Max Age 20 sec    Forward Delay 15 sec
Aging Time   20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/2	Desg	FWD	19	128.2	P2p
Fa0/1	Root	FWD	19	128.1	P2p

Видим, что отреагировал он спокойно. Switch 1 по-прежнему является корневым для

VLAN 1 и VLAN 2. И лишь для VLAN 3 он изменил свое состояние и состояния портов.

Вот таким образом можно управлять различными процессами STP для каждого из VLAN-ов. Прикладываю [ссылку](#) на скачивание.

Это все конечно хорошо, что коммутатор перед включением порта, всячески все перепроверяет. Но если мы знаем, что за портом коммутатора находится клиентский компьютер, который не создаст петли, то можно сразу перевести порт в режим «Forwarding», не дожидаясь 30 секунд. Для этого есть технология «Portfast».

Зайду на коммутатор Switch2 и продемонстрирую на примере порта FastEthernet 0/3:

```
Switch2(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION
```

```
%Portfast has been configured on FastEthernet0/3 but will only
have effect when the interface is in a non-trunking mode.
```

После ввода, он сразу переводит порт в режим Forwarding, но выводит предупреждение о том, что этот порт должен строго подключаться к одному пользовательскому хосту. Иначе, при подключении коммутаторов и прочих устройств, это может привести к появлению петли. Под спойлером ниже показано, как именно это работает.

Portfast

Как видите, он миновал все стадии и сразу перешел к режиму «Forwarding». Не забывайте про эту технологию, но и пользуйтесь ею с осторожностью, так как окажись там не пользовательский хост, а коммутатор или иное устройство, вы рискуете создать петлю.

Вот основной принцип работы PVST+. Как видите, он мало чем отличается от классического STP или CST.

Я думаю вы заметили какое полотно текста выводит команда «show spanning-tree». И чем больше VLAN-ов, тем больше этот вывод. И если вам нужно будет посмотреть информацию на коммутаторе за 10-ый VLAN, то придется прокручивать весь вывод с самого начала, пока не доберетесь до строчки с нужным VLAN-ом. Для облегчения данной ситуации, есть очень хорошая команда, позволяющая узнать информацию за конкретный VLAN. Это команда «show spanning-tree vlan X». Проверю эту команду.

```
Switch1#show spanning-tree vlan 3
VLAN0003
  Spanning tree enabled protocol ieee
  Root ID    Priority    24579
             Address     00D0.BA84.7C38
             Cost        19
             Port        1(FastEthernet0/1)
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32771 (priority 32768 sys-id-ext 3)
             Address     0009.7C94.0557
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20

Interface                Role Sts Cost        Prio.Nbr Type
-----
Fa0/2                    Desg FWD 19          128.2    P2p
Fa0/1                    Root FWD 19          128.1    P2p
```

И вот он мне по моей команде выводит информацию только за 3-ий VLAN. Очень удобная команда, поэтому берите на заметку.

Есть еще одна интересная команда «show spanning-tree summary».

```
Switch3#show spanning-tree summary
Switch is in pvst mode
Root bridge for: VLAN0003
Extended system ID      is enabled
Portfast Default        is disabled
PortFast BPDU Guard Default is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default       is disabled
EtherChannel misconfig guard is disabled
UplinkFast              is disabled
BackboneFast            is disabled
Configured Pathcost method used is short
```

Name	Blocking	Listening	Learning	Forwarding	STP Active
VLAN0001	1	0	0	1	2
VLAN0002	1	0	0	1	2
VLAN0003	0	0	0	2	2
3 vlans	2	0	0	4	6

Она показывает суммарную и краткую статистику. В каком STP режиме работает коммутатор, для какого VLAN-а он является корневым, какие функции на нем включены. И самое главное, тут есть таблица, содержащая имена VLAN-ов и количество интерфейсов в данном VLAN-е, находящихся в различных состояниях. Это очень полезно, когда надо быстро зайти и посмотреть есть ли на коммутаторе заблокированные порты и для какого VLAN-а они заблокированы.

В принципе из всех команд — эти часто используемые и для уровня CCNA их более, чем достаточно.

На самом деле STP и PVST+ не единственные протоколы предотвращения петель. Есть еще RSTP и MSTP. Если MSTP в программе CCNA практически не упоминается, за исключением того, что он такой есть, то про RSTP говорить открыто и подробно Cisco начала с новой версией программы CCNA 3.0. Поэтому разберу его поподробнее.

Наверное вы заметили, что классический STP, что PVST+ требуют время на сходимость. А именно 30 секунд, при отказе или отключении какого-либо линка. Это конечно не так много, но чем больше сеть, тем больше времени это занимает. И в большой корпоративной среде полная сходимость может занять несколько минут. И вот для разрешения такой ситуации, комитет IEEE выпустил стандарт 802.1w или протокол RSTP.

В чем суть. Если в классическом STP было 4 состояния (Blocking, Listening, Learning, Forwarding), то в RSTP их стало меньше. Всего 3 (Discarding, Listening и Forwarding). То есть коммутатор отбрасывает, изучает или пересылает. Но быстрее он сходится не из-за этого. Быструю сходимость протокол обеспечивает тем, что заранее просчитывает, какой порт включить, если откажет работающий. Тем самым, при отказе порта, он не начинает судорожно изучать топологию и прыгать по различным состояниям, а просто переключается на заранее просчитанный. Очень хорошее дополнение по быстрой сходимости протокола RSTP оставил пользователь под ником [ksg222](#). И за это выражаю ему свою благодарность.

Цитирую:

Быструю сходимость и реакцию на отказы в RSTP обеспечивают:

- генерация BPDU сообщений каждым устройством, независимо от корневого коммутатора;
- механизм proposal/agreement, при активации канала точка-точка, для быстрого перехода портов в состояние «Forwarding»
- механизм использования альтернативного порта, при потери связи через корневой порт (наподобие технологии [UplinkFast](#));
- механизм немедленной реакции на получение BPDU с «худшим» корневым коммутатором от соседа, имеющего designated порты для данного сегмента сети (наподобие технологии [backbonefast](#));
- более оптимальная схема рассылки и обработки сообщений TCN BPDU об изменениях в сети.

Включить протокол RSTP можно командой:

```
Switch2(config)#spanning-tree mode rapid-pvst
```

Я собрал лабораторку и включил на каждом коммутаторе RSTP и проверю, как быстро произойдет перестроение дерева.

Перестроение дерева протоколом RSTP

Как видите, перестроение происходит в считанные секунды. Для тех, кто захочет проверить это на себе, прикладываю [файл](#) с лабораторкой.

Вот и подошла к концу статья о протоколах STP. Теперь мы можем строить процессы STP для каждого VLAN-а, управлять приоритетом и много другого. А для быстроты сходимости можем применять протокол RSTP.

P.S. Постарался осветить все моменты, но из-за того, что статья писалась на протяжении полутора месяцев в различные промежутки времени, то мог что-то упустить. Если есть, что добавить, подчеркнуть или что-то осталось непонятным, смело пишите. И даже, если все понятно, то тоже пишите. Мне будет приятно! Успехов вам в обучении и до встречи в следующей статье.