

CCIE Enterprise Infrastructure: STP, Rapid STP

### Сегодня в выпуске:

- STP
  - Выбор корневого коммутатора
  - Выбор корневого порта
  - Основные таймеры
- RSTP

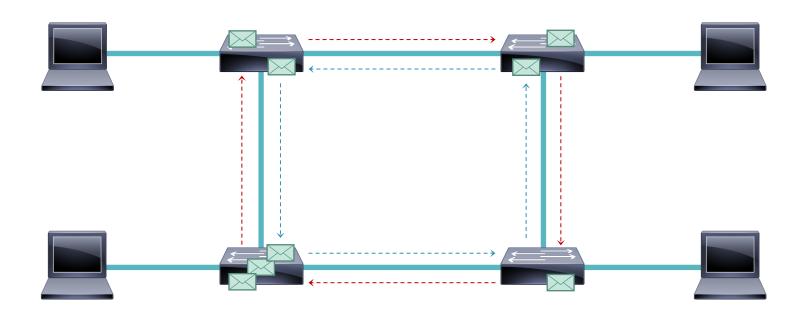




**Spanning Tree Protocol** 

### Проблемы в сетях с избыточностью

- Шторм из широковещательных кадров
- Нестабильность базы МАС-адресов
- Множественная повторная доставка кадра





### **Spanning Tree Protocol**

- Семейство протоколов, блокирующих коммутацию кадров на портах, формирующих петлю в Ethernet
- Используется «алгоритм остовного дерева»
- Разработан Радьей Перлман в 1985 г.
- IEEE включила STP в 802.1D-1990
- Существуют другие стандартные версии STP
  - 802.1w Rapid Spanning Tree, впоследствии объединен с 802.1D-2004
  - 802.1s Multiple Spanning Trees, впоследствии объединен с 802.1Q-2003
- Кроме того, существуют и нестандартные реализации
  - PVST/PVRST проприетарное расширение Cisco для работы с VLAN



### Основные принципы STP

- Выбирается один корневой коммутатор
- Выбирается один корневой порт для каждого коммутатора
  - За исключение корневого коммутатора
- Выбирается один выделенный (Designated) порт для каждого сегмента



### Выбор корневого коммутатора

- Корневым коммутатором становится коммутатор с наименьшим Bridge ID
- В начале процесса каждый коммутатор считает себя root bridge
  - И рассылает BPDU с указанием своего Bridge ID в качестве Root Bridge ID
- При получении от соседа BPDU, Root Bridge ID в которой указан меньше текущего Root Bridge ID, коммутатор начинает считать корневым тот Bridge ID, который указан в полученной BPDU
  - И начинает рассылать новые BPDU уже с указанием нового Root Bridge ID



### Bridge ID

- Bridge ID состоит из
  - Bridge Priority
    - От 0 до 61440
      - По умолчанию 32768
      - Может принимать значения 0, 4096, 8192 и т.д.
  - System ID Extension
    - 0 4095
      - По сути, номер VLAN
  - MAC address



### Стоимость до корневого коммутатора

- Каждый коммутатор рассылает BPDU с указанием Bridge ID корневого коммутатора и стоимость пути до него (Root Path Cost, RPC)
- У корневого коммутатора RPC по определению равна нулю
- При получении BPDU коммутатор прибавляет к указанному в поле Root Path Cost значению стоимость порта, на котором получена BPDU
  - Корневым портом становится тот, на котором сумма получилась наименьшей
  - Полученная на корневом порту сумма становится RPC для коммутатора



### Корневой порт

- Корневой порт тот, на котором получена самая «выгодная» BPDU
  - Также она «лучше» (superior) той, которую в порт могли бы отправить мы
  - При подсчете RPC стоимость порта прибавляется к заявленной в BPDU
- Корневой порт разблокируется после пересчета топологии
- На каждом коммутаторе (кроме корневого) ровно один root port



### Альтернативный порт

- Alternate port «запасной корневой порт»
  - BPDU на alternate port «хуже» (inferior) по сравнению с BPDU на корневом порту (после прибавления стоимостей портов)
  - Однако она «лучше» (superior) той, которую в порт могли бы отправить мы
- Alternate port остается заблокированным после пересчета топологии
- Таких портов на коммутаторе может быть несколько



### Выделенный порт

- Designated port порт, на котором отправляемая нами BPDU оказалась выгоднее по сравнению с теми, которые мы получаем
  - А на самом деле не получаем нет смысла отправлять inferior BPDU
  - Поэтому DP единственные, в которые BPDU отправляются
- На корневом коммутаторе все порты designated
  - Порт разблокируется после окончания пересчета топологии
    - В начале пересчета топологии все порты designated (но заблокированы)



### Запасной порт

- Backup port «запасной designated port»
  - Принимается superior BPDU, у которой Sender Bridge ID совпадает с нашим
  - Коммутатор в сегмент смотрит более чем одним интерфейсом, один из которых выбран designated, а остальные backup
- Backup port остается заблокированным после пересчета топологии



# Таймеры

- Message Age
  - Срок, который BPDU уже «прожила»
- Max Age
  - Срок, до которого BPDU может «дожить»
- Hello time
  - Задается на корневом коммутаторе, применяе
  - Частота отправки BPDU на интерфейсах
- Forward Delay
  - Время, в течение которого информация в BPDU передастся от любого коммутатора в топологии

	Поле	Размер
Заголовок	Protocol Identifier	2 байта
	Protocol Version Identifier	1 байт
	BPDU Type	1 байт
	Flags	1 байт
Priority Vector		
		2 байта
Гаймеры	Message Age	2 байта
	Max Age	2 байта
	Hello Time	2 байта
/	Forward Delay	2 байта



### Таймеры по-умолчанию

- Hello time 2 секунды
- Forward Delay 15 секунд
  - Таймер в 802.1D применяется дважды для состояний Listening и Learning
- Max Age 20 секунд
  - Время, в течение которого коммутатор «помнит» полученные BPDU
- Таймеры рассчитаны из рекомендуемых значений:
  - Maximum bridge diameter 7 коммутаторов
  - Maximum bridge transit delay 1 секунда
  - Maximum BPDU transmission delay 1 секунда
  - Maximum Message Age increment overestimate 1 секунда



# Состояния Listening/Learning

- В состоянии Listening порт ожидает завершения пересчета топологии
  - Из-за возможного наличия петли коммутация не осуществляется
  - Пополнение таблицы МАС-адресов не производится, т.к. кадры от узлов могут приходить с тех интерфейсов, которые позже не разблокируются
  - Порт отправляет или принимает только BPDU протокола STP (исключение могут составлять некоммутируемые кадры протоколов LLDP, CDP, VTP и т.п.)
- В состоянии Learning порт готов коммутировать кадры
  - Коммутация не осуществляется, т.к. возможно возникновение петли
  - Происходит наполнение таблицы МАС-адресов актуальной информацией
- С настройками по умолчанию при пересчете топологии STP блокирует коммутацию на порту на 30 секунд



## Состояния Forwarding/Blocking

- Стабильные состояния
- Forwarding
  - Осуществляется коммутация
  - Осуществляется нормальное пополнение таблицы МАС-адресов
  - Отправляются/принимаются BPDU и кадры служебных протоколов
- Blocking
  - Коммутация не производится, полученные кадры сбрасываются
  - Пополнение таблицы МАС-адресов не производится
  - BPDU принимаются (но не отправляются по определению)
  - Кадры некоммутируемых протоколов (LLDP, VTP, CDP и т.п.) отправляются и принимаются нормально

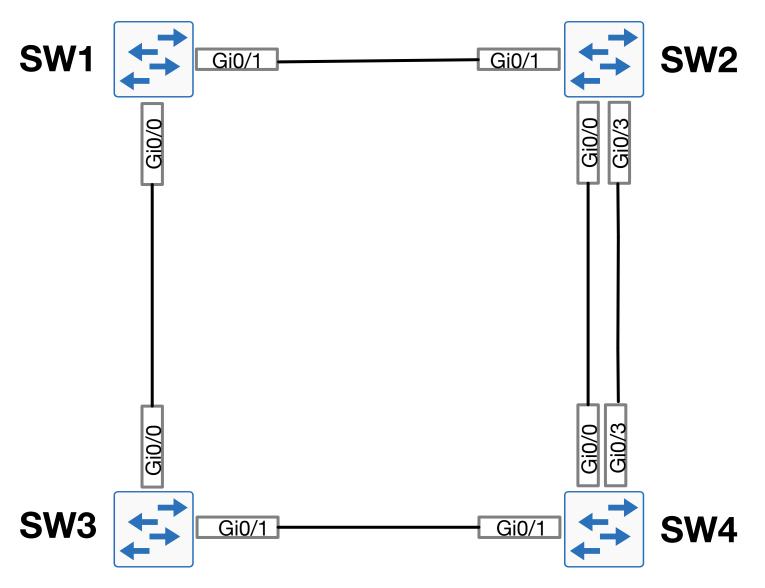


### Смена топологии

- Что ее вызывает?
  - Получение Topology Change Notification BPDU на Designated-порту
  - Получение роли Root Bridge
  - Переход порта из состояний Forwarding или Learning в Blocking
  - Переход порта из состояния Learning в Forwarding, если на коммутаторе имеются Designated-порты
    - Смена Root-порта на транзитном коммутаторе
    - Активация нового Designated-порта
- Что это по факту?
  - Очистка таблиц МАС-адресов на всех коммутаторах



# Разберем топологию?







Rapid STP

### Rapid Spanning Tree

- Использует тот же алгоритм расчета дерева, что и STP
  - По тому же принципу выбираются корневой коммутатор и forwardingпорты
  - Обратно совместим с «медленным» STP
- Значительно уменьшает как время пересчета топологии, так и время восстановления работоспособности сети после отказа
  - В основном за счет отказа от поддержки shared топологий
- Особенно влияют на ускорение:
  - Появление сообщений «запрос-ответ», синхронизирующих состояние двух коммутаторов на P2P-канале не за Hello time, а за единицы миллисекунд
  - Появление Edge port, на котором коммутация выполняется сразу после включения (но возможно кратковременное формирование петли)
    - Смена состояния Edge-порта на Forwarding не вызывает смену топологии



### Изменения в BPDU

- Используется поле Flags
- Иногда можно встретить понятие BPDUv2
- BPDU = keepalive
  - каждые 2 секунды

#### Rapid Spanning Tree BPDU



### Порты в RSTP

- Edge: пограничный порт, за которым расположен конечный узел
  - При включении немедленно переходит в Designated Forwarding
  - Меняет тип при получении BPDU
- Point-to-point: порт, "смотрящий" на <u>один</u> транзитный коммутатор
  - Используются механизмы ускорения RSTP proposal/agreement
  - Назначается по умолчанию на полнодуплексных интерфейсах
- Shared: порт, "смотрящий" на <u>несколько</u> коммутаторов
  - Механизмы ускорения отключены
  - Назначаются по умолчанию на полудуплексных интерфейсах



### Механизм А-Р

- Proposal отправляется портом, который хочет стать Designated
- При получении Proposal, коммутатор «выключает» все порты, не являющиеся Edge-портами
- Весь этот процесс теперь не привязан к таймерам

```
Protocol Identifier: Spanning Tree Protocol (0x0000)
Protocol Version Identifier: Rapid Spanning Tree (2)
BPDU Type: Rapid/Multiple Spanning Tree (0x02)
BPDU flags: 0x3c (Forwarding, Learning, Port Role: Designated)
0.... = Topology Change Acknowledgment: No
0... = Agreement: No
0.1 = Forwarding: Yes
0.1 = Port Role: Designated (3)
0.1 = Proposal: No
0.1 = Topology Change: No
```

# Proposal/Agreement

- Механизм ускорения сходимости в RSTP
  - Работает в point-to-point сегменте, где по определению один сосед
  - Designated Bridge в сегменте выбирается не за Forward Delay, а за ~10мс
- Два коммутатора договариваются о будущем Designated Bridge
  - Designated bridge отправляет BPDU с установленным флагом Proposal
  - Нижестоящий коммутатор отвечает inferior BPDU с флагом Agreement



### RSTP может быть медленным

- К порту подключается устройство, поддерживающее только классический STP
- Коммутатор определил соединение как shared
- Подключено устройство, не участвующее в STP, и на порту не задан тип edge



### RSTP Topology Change

- Падение интерфейса не является ТС
- TC = переход non-edge интерфейса в состояние FWD
  - Очистка МАС таблицы
  - Установка ТС бита в ВРDU
- При получении ТС
  - очистка MAC таблицы на всех интерфейсах, за исключение того, через который было получено BPDU
  - распространение TC через Root и Designated интерфейсы





STP Toolkit

# Стабилизация STP топологии

- BPDU Guard
- BPDU Filter
- Rootguard
- Loopguard
- Bridge Assurance
- Portfast

### STP BPDU Guard

- При получении BPDU переводит интерфейс в состояние err-disable
- Включается глобально или на интерфейсе



### **BPDU Filter**

- Фильтрация BPDU сообщений
- Включается глобально или на интерфейс
  - Глобально
    - настройка применяется для всех интерфейсов, на которых работает Portfast
    - при получении BPDU, порт теряет статус Portfast
  - На интерфейсе
    - BPDU не отправляются
    - входящие BPDU игнорируется (отбрасываются)

### **Root Guard**

• При получении Superior BPDU, интерфейс переводится в состояние Root Inconsistent



### Loopguard

- Помогает защититься от однонаправленных неполадок связи (чаще всего обрыв одного оптического волокна)
- Если на интерфейс перестают приходить BPDU, порт переводится в состояние Loop Inconsistence
- Необходимо включать только на BKN интерфейсах

### Bridge Assurance

- С включенным BA, BPDU отправляются через все интерфейсы
- Если на интерфейсе нет входящих BPDU, порт переводится в состояние BA Inconsistence
- Если говорить про IOS, то поддерживается только на Catalyst 6500/6800/4500
  - + на всех коммутаторах Cisco Nexus

### Portfast

- После физического поднятия, интерфейс немедленно переходит в состояние FWD
  - минуются LIS и LRN
- При получении BPDU, роль Edge теряется





Per-VLAN Spanning Tree

### STP + VLAN

- STP блокирует коммутацию на порту без учета информации о VLAN
  - Остовное дерево в 802.1D строится по всей физической топологии, а не по отдельным логическим виртуальным поддоменам (VLAN)
- Есть вероятность блокировки коммутации там, где физически петля существует, но логически ее нет (равно как нет и последствий петли)



# PVST(+)

- PVST проприетарный протокол с поддержкой VLAN
  - PVST поддерживает транки ISL, PVST+ поддерживает также транки 802.1Q, PVRST (Rapid PVST+) добавляет механизмы ускорения RSTP
  - Строит дерево не по физической топологии, а по VLAN
- В протоколах семейства PVST строится одно дерево за каждый VLAN
  - Много VLAN много BPDU в транке
- Для идентификации VLAN используем дополнительное TLV
  - Почему нельзя использовать 802.1Q метку?



# Немного практики





# Networking For everyone