

habr.com

Принцип работы протокола STP

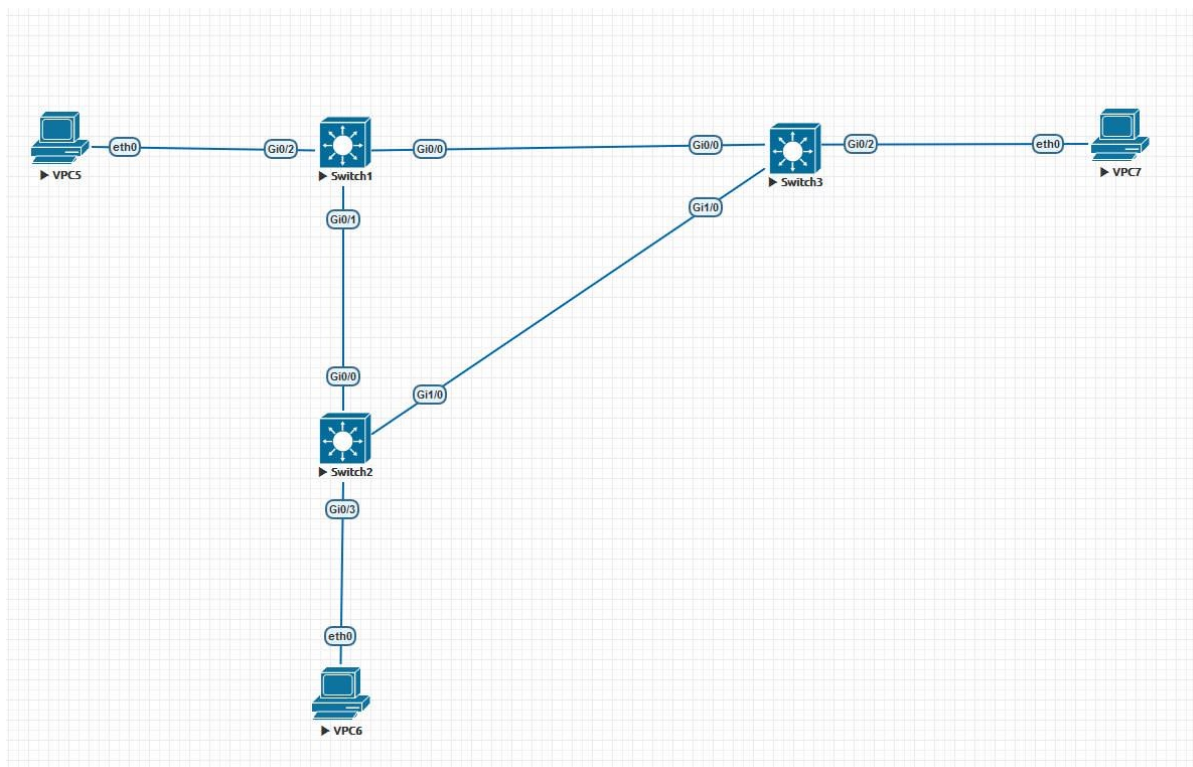
18-26 минут

Причина создания STP

Причиной создания протокола STP стало возникновение петель на коммутаторах. Что такое петля? Определение петли звучит так:

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. Как возникает петля? На картинке ниже приведена топология, при которой будет возникать петля при отсутствии каких-либо защитных механизмов:



Возникновение петли при следующих условиях:

1. 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 она 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.

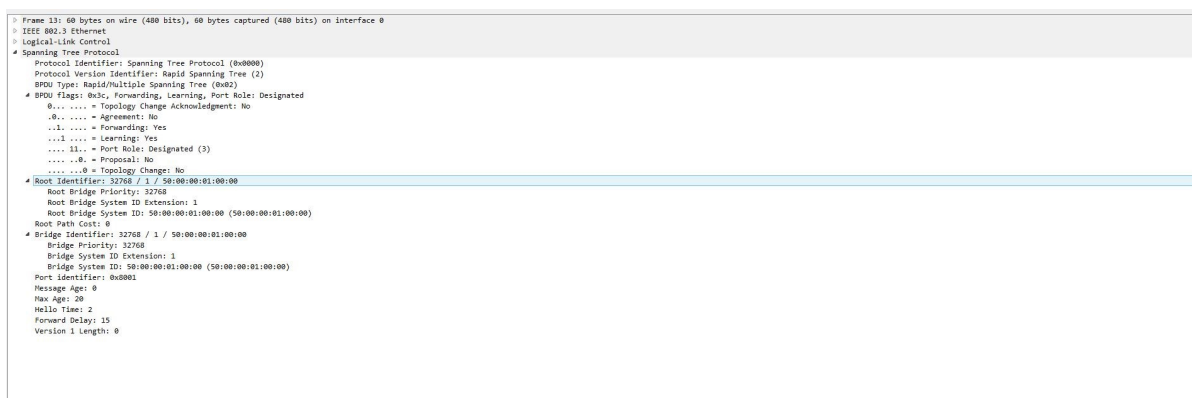
Шаги 1) — 4) будут повторяться бесконечно и на коммутаторах это происходит в долю секунды. Также образование петли приводит к тому, что постоянно будет меняться таблица мак-адресов на коммутаторах и мак-адрес отправителя VPC5 будет постоянно приписываться то к интерфейсу Gi0/0, то Gi1/0 или же Gi0/2(если в этот момент VPC5 будет отправлять другие пакеты). Такой цикл приведет к некорректной работе сети и всех коммутаторов. А отправка бродкастовых пакетов для хостов — это обычное дело, в качестве примера протокол ARP.

2. Также петля может образоваться и без отправки бродкаст фрейма.

1. К примеру, VPC5 отправляет фрейм с уникальным мак-адресом назначения.
2. Возможна ситуация, что мак-адрес назначения отсутствует в таблице мак-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.
3. Ниже мы будем рассматривать протокол STP на коммутаторах Cisco. На них используется STP отдельно для каждого vlan-а, протокол PVST+. У нас всего один vlan, поэтому смысл от этого не меняется.

Основы STP

Принцип работы данного протокола построен на том, что все избыточные каналы между коммутаторами логически блокируются и трафик через них не передается. Для построения топологии без избыточных каналов строится дерево (математический граф). Чтобы построить такое дерево вначале необходимо определить корень дерева, из которого и будет строиться граф. Поэтому первым шагом протокола STP является определение корневого коммутатора (Root Switch). Для определения Root Switch-а, коммутаторы обмениваются сообщениями BPDU. В общем, протокол STP использует два типа сообщений: BPDU — содержит информацию о коммутаторах и TCN — уведомляет о изменении топологии. Рассмотрим BPDU более детально. Про TCN более подробно поговорим ниже. При включении STP на коммутаторах, коммутаторы начинают рассылать BPDU сообщения. В данных сообщениях содержится следующая информация:



Фрейм BPDU имеет следующие поля:

- Идентификатор версии протокола STA (2 байта). Коммутаторы должны поддерживать одну и ту же версию протокола STA
- Версия протокола STP (1 байт)
- Тип BPDU (1 байт). Существует 2 типа BPDU — конфигурационный и уведомление о реконфигурации

- Флаги (1 байт)
- Идентификатор корневого коммутатора (8 байт)
- Стоимость маршрута до корневого свича (Root Path Cost)
- Идентификатор отправителя (Bridge ID) (8 байт)
- Идентификатор порта, из которого отправлен данный пакет (Port ID) (2 байта)
- Время жизни сообщения (2 байта). Измеряется в единицах по 0,5 с, служит для выявления устаревших сообщений
- Максимальное время жизни сообщения (2 байта). Если кадр BPDU имеет время жизни, превышающее максимальное, то кадр игнорируется коммутаторами
- Интервал hello (2 байт), интервал через который посылаются пакеты BPDU
- Задержка смены состояний (2 байта). Минимальное время перехода коммутатора в активное состояние

Основные поля, которые требуют особого внимания следующие:

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

Для определения корневого коммутатора используется идентификатор коммутатора — Bridge ID. Bridge ID это число длиной 8 байт, которое состоит из Bridge Priority (приоритет, от 0

до 65535, по умолчанию 32768) и MAC-адреса устройства. Корневым коммутатором выбирается коммутатор с самым низким приоритетом, если приоритеты равны, то сравниваются MAC-адреса (посимвольно, тот который меньше, тот побеждает).

Вот вывод информации о Bridge ID с коммутатора Switch1 из первой картинки. Priority — 32769 (по умолчанию 32768 + Vlan Id), MAC-адреса — Address 5000.0001.0000:

```
Switch#show span
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     5000.0001.0000
             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     5000.0001.0000
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec
```

Представим картину, коммутаторы только включились и теперь начинают строить топологию без петель. Как только коммутаторы загрузились, они приступают к рассылке BPDU, где информируют всех, что они являются корнем дерева. В BPDU в качестве Root Bridge ID, коммутаторы указывают собственный Bridge ID. Например, Switch1 отправляет BPDU коммутатору Switch3, а Switch3 отправляет к Switch1. BPDU от Switch1 к Switch3:

```
Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Configuration (0x00)
  BPDU flags: 0x00
    0... .. = Topology Change Acknowledgment: No
    ....0 = Topology Change: No
  Root Identifier: 32768 / 1 / 50:00:00:01:00:00
    Root Bridge Priority: 32768
    Root Bridge System ID Extension: 1
    Root Bridge System ID: 50:00:00:01:00:00 (50:00:00:01:00:00)
```

```
Root Path Cost: 0
  Bridge Identifier: 32768 / 1 / 50:00:00:01:00:00
    Bridge Priority: 32768
    Bridge System ID Extension: 1
    Bridge System ID: 50:00:00:01:00:00 (50:00:00:01:00:00)
  Port identifier: 0x8001
  Message Age: 0
  Max Age: 20
  Hello Time: 2
  Forward Delay: 15
```

BPDU от Switch3 к Switch1:

```
Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Configuration (0x00)
  BPDU flags: 0x00
    0... .... = Topology Change Acknowledgment: No
    .... ...0 = Topology Change: No
  Root Identifier: 32768 / 1 / 50:00:00:03:00:00
    Root Bridge Priority: 32768
    Root Bridge System ID Extension: 1
    Root Bridge System ID: 50:00:00:03:00:00 (50:00:00:03:00:00)
  Root Path Cost: 0
  Bridge Identifier: 32768 / 1 / 50:00:00:03:00:00
    Bridge Priority: 32768
    Bridge System ID Extension: 1
    Bridge System ID: 50:00:00:03:00:00 (50:00:00:03:00:00)
  Port identifier: 0x8001
  Message Age: 0
  Max Age: 20
  Hello Time: 2
  Forward Delay: 15
```

Как видим из Root Identifier, оба коммутатора друг другу сообщают, что именно он является Root коммутатором.

Выбор корневого коммутатора

Пока топология STP не построена, обычный трафик не передается из-за специальных состояний портов, о которых будет сказано ниже. Итак, Switch3 получает BPDU от Switch1 и изучает данное сообщение. Switch3 смотрит в поле Root Bridge ID и видит, что там указан другой Root Bridge ID, чем в

том сообщении, которое отправил сам Switch3. Он сравнивает Root Bridge ID в данном сообщении со своим Root Bridge ID и видит, что хоть Priority одинаковые, но MAC-адрес данного коммутатора (Switch1) лучше (меньше), чем у него. Поэтому Switch3 принимает Root Bridge ID от Switch1 и перестает отправлять свои BPDU, а только слушает BPDU от Switch1. Порт, на котором был получен наилучший BPDU становится Root Port-ом. Switch1 также получив BPDU от Switch3, проводит сравнение, но в этом случае поведение Switch1 не меняется, так как полученный BPDU содержит худший Root Bridge ID, чем у Switch1. Таким образом, между Switch1 и Switch3 был определен корневой коммутатор. По аналогичной схеме происходит выбор корневого коммутатора между Switch1 и Switch2. Порты Gi0/0 на Switch2 и Switch3 становятся Root Port — порт, который ведет к корневому коммутатору. Через данный порт коммутаторы Switch2 и Switch3 принимают BPDU от Root Bridge. Теперь разберемся, что произойдет с каналом между Switch2 и Switch3.

Блокирование избыточных каналов

Как мы видим из топологии, канал между Switch2 и Switch3 должен быть заблокирован для предотвращения образования петель. Как STP справляется с этим?

После того, как выбран Root Bridge, Switch2 и Switch3 перестают отправлять BPDU через Root Port-ы, но BPDU, полученные от Root Bridge, они пересылают через все свои остальные активные порты, при этом изменив в данных BPDU только следующие поля:

- Идентификатор отправителя (Bridge ID) — заменяется на свой идентификатор.
- Идентификатор порта, из которого отправлен данный пакет (Port ID) — изменяется на идентификатор порта, с которого будет отправлен BPDU.
- Стоимость маршрута до корневого свича (Root Path Cost) — вычисляется стоимость маршрута относительно самого коммутатора.

Таким образом, Switch2 получает следующий BPDU от Switch3:

```

> Frame 7: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> IEEE 802.3 Ethernet
> Logical-Link Control
  Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
    > BPDU flags: 0x00
    Root Identifier: 32768 / 1 / 50:00:00:01:00:00
      Root Bridge Priority: 32768
      Root Bridge System ID Extension: 1
      Root Bridge System ID: 50:00:00:01:00:00 (50:00:00:01:00:00)
      Root Path Cost: 4
    Bridge Identifier: 32768 / 1 / 50:00:00:02:00:00
      Bridge Priority: 32768
      Bridge System ID Extension: 1
      Bridge System ID: 50:00:00:02:00:00 (50:00:00:02:00:00)
      Port identifier: 0x8005
      Message Age: 1
      Max Age: 20
      Hello Time: 2
      Forward Delay: 15
```

A Switch3 от Switch2 получает такой BPDU:

```

> Frame 6: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> IEEE 802.3 Ethernet
> Logical-Link Control
  Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
    > BPDU flags: 0x00
    Root Identifier: 32768 / 1 / 50:00:00:01:00:00
      Root Bridge Priority: 32768
      Root Bridge System ID Extension: 1
      Root Bridge System ID: 50:00:00:01:00:00 (50:00:00:01:00:00)
      Root Path Cost: 4
    Bridge Identifier: 32768 / 1 / 50:00:00:03:00:00
      Bridge Priority: 32768
      Bridge System ID Extension: 1
      Bridge System ID: 50:00:00:03:00:00 (50:00:00:03:00:00)
```

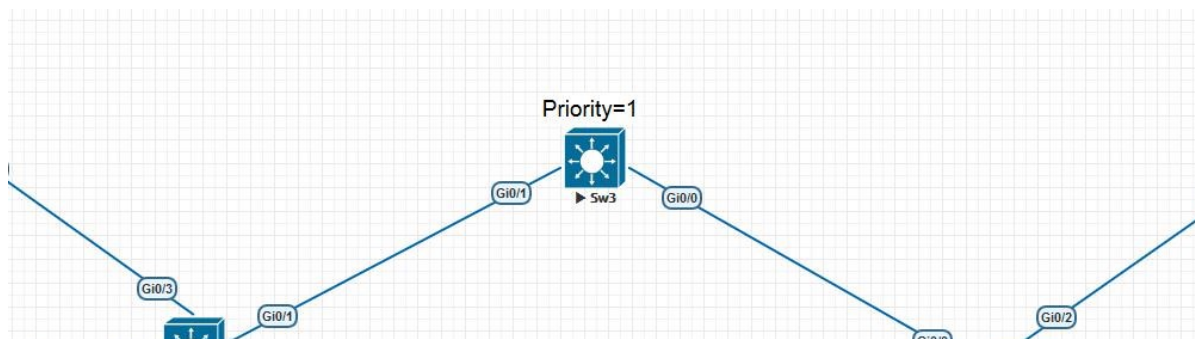
```
Port identifier: 0x8005  
Message Age: 1  
Max Age: 20  
Hello Time: 2  
Forward Delay: 15
```

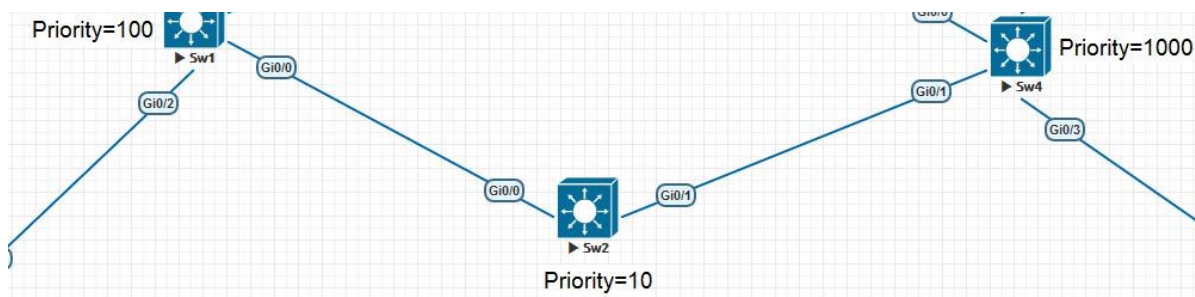
После обмена такими BPDU, Switch2 и Switch3 понимают, что топология избыточна. Почему коммутаторы понимают, что топология избыточна? И Switch2, и Switch3 в своих BPDU сообщают об одном и том же Root Bridge. Это означает, что к Root Bridge, относительно Switch3, существует два пути — через Switch1 и Switch2, а это и есть та самая избыточность против которой мы боремся. Также и для Switch2 два пути — через Switch1 и Switch3. Чтоб избавиться от этой избыточности необходимо заблокировать канал между Switch3 и Switch2. Как это происходит?

Выбор на каком коммутаторе заблокировать порт происходит по следующей схеме:

- Меньшего Root Path Cost.
- Меньшего Bridge ID.
- Меньшего Port ID.

В данной схеме Root Path Cost играет более важную роль, чем Bridge ID. Раньше я думал, что данный выбор происходит аналогично выбору Root коммутатора и был удивлен, что, например, в такой топологии будет блокироваться не порт на коммутаторе с самым плохим приоритетом:





Здесь как оказалось заблокируется порт Gi 0/1 на коммутаторе Sw2. В данном голосовании определяющим становится Root Path Cost. Вернемся к нашей топологии. Так как путь до Root Bridge одинаковый, то в данном выборе побеждает Switch2, так как его priority равны, сравниваются Bridge ID. У Switch2 — 50:00:00:02:00:00, у Switch3 — 50:00:00:03:00:00. У Switch2 MAC-адрес лучше (меньше). После того, как выбор сделан, Switch3 перестает переслать какие-либо пакеты через данный порт — Gi1/0, в том числе и BPDU, а только слушает BPDU от Switch2. Данное состояние порта в STP называется Blocking(BLK). Порт Gi1/0 на Switch2 работает в штатном режиме и пересылает различные пакеты при необходимости, но Switch3 их сразу отбрасывает, слушая только BPDU. Таким образом, на данном примере мы построили топологию без избыточных каналов. Единственный избыточный канал между Switch2 и Switch3 был заблокирован при помощи перевода порта Gi1/0 на Switch3 в специальное состояние блокирования — BLK. Теперь более детально разберем механизмы STP.

Состояния портов

Мы говорили выше, что, например, порт Gi1/0 на Switch3 переходит в специальное состояние блокирования — Blocking. В STP существуют следующие состояния портов:

Blocking — блокирование. В данном состоянии через порт не

передаются никакие фреймы. Используются для избежания избыточности топологии.

Listening — прослушивание. Как мы говорили выше, что до того, пока еще не выбран корневой коммутатор, порты находятся в специальном состоянии, где передаются только BPDU, фреймы с данными не передаются и не принимаются в этом случае. Состояние Listening не переходит в следующее даже, если Root Bridge определен. Данное состояние порта длится в течении Forward delay timer, который, по умолчанию, равен 15. Почему всегда надо ждать 15 секунд? Это вызвано осторожностью протокола STP, чтоб случайно не был выбран некорректный Root Bridge. По истечению данного периода, порт переходит в следующее состояние — Learning.

Learning — обучение. В данном состоянии порт слушает и отправляет BPDU, но информацию с данными не отправляет. Отличие данного состояния от Listening в том, что фреймы с данными, который приходят на порт изучаются и информация о MAC-адресах заносится в таблицу MAC-адресов коммутатора. Переход в следующее состояние также занимает Forward delay timer.

Forwarding — пересылка. Это обычное состояние порта, в котором отправляются и пакеты BPDU, и фреймы с обычными данными. Таким образом, если мы пройдемся по схеме, когда коммутаторы только загрузились, то получается следующая схема:

1. Коммутатор переводит все свои подключенные порты в состояние Listening и начинает отправлять BPDU, где объявляет себя корневым коммутатором. В этот период времени, либо

коммутатор остается корневым, если не получил лучший BPDU, либо выбирает корневой коммутатор. Это длится 15 секунд.

2. После переходит в состояние Learning и изучает MAC-адреса. 15 секунд.
3. Определяет какие порты перевести в состояние Forwarding, а какие в Blocking.

Роли портов

Помимо состояний портов, также в STP нужны определить портам их роли. Это делается для того, чтоб на каком порте долженждаться BPDU от корневого коммутатора, а через какие порты передавать копии BPDU, полученных от корневого коммутатора. Роли портов следующие:

Root Port — корневой порт коммутатора. При выборе корневого коммутатора также и определяется корневой порт. Это порт через который подключен корневой коммутатор. Например, в нашей топологии порты Gi0/0 на Switch2 и Switch3 являются корневыми портами. Через данные порты Switch2 и Switch3 не отправляют BPDU, а только слушают их от Root Bridge.

Возникает вопрос — как выбирается корневой порт? Почему не выбран порт Gi1/0? Через него ведь тоже можно иметь связь с коммутатором? Для определения корневого порта в STP используется метрика, которая указывает в поле BPDU — Root Path Cost (стоимость маршрута до корневого свича). Данная стоимость определяется по скорости канала.

Switch1 в своих BPDU в поле Root Path Cost ставит 0, так как

сам является Root Bridge. А вот, когда Switch2, когда отправляет BPDU к Switch3, то изменяет данное поле. Он ставит Root Path Cost равным стоимости канала между собой и Switch1. На картинке BPDU от Switch2 и Switch3 можно увидеть, что в данном поле Root Path Cost равен 4, так как канал между Switch1 и Switch2 равен 1 Gbps. Если количество коммутаторов будет больше, то каждый следующий коммутатор будет суммировать стоимость Root Path Cost. Таблица Root Path Cost.

Designated Port — назначенный порт сегмента. Для каждого сегмента сети должен быть порт, который отвечает за подключение данного сегмента к сети. Условно говоря, под сегментом сети может подразумеваться кабель, который осуществляет подключение данного сегмента. Например, порты Gi0/2 на Switch1, Switch3 подключают отдельные сегменты сети, к которым ведет только данный кабель. Также, например, порты на Root Bridge не могут быть заблокированы и все являются назначенными портами сегмента. После данного пояснения можно дать более строгое определение для назначенных портов:

Designated Port (назначенный) — некорневой порт моста между сегментами сети, принимающий трафик из соответствующего сегмента. В каждом сегменте сети может быть только один назначенный порт. У корневого коммутатора все порты — назначенные.

Также важно заметить, что порт Gi1/0 на Switch2 также является назначенным, несмотря на то, что данный канал связи заблокированным на Switch3. Условно говоря, Switch2 не имеет информации о том, что на другом конце порт заблокирован.

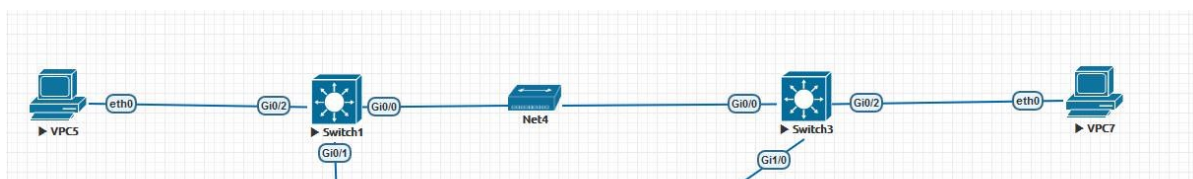
Nondesignated Port — неназначенный порт сегмента. Non-

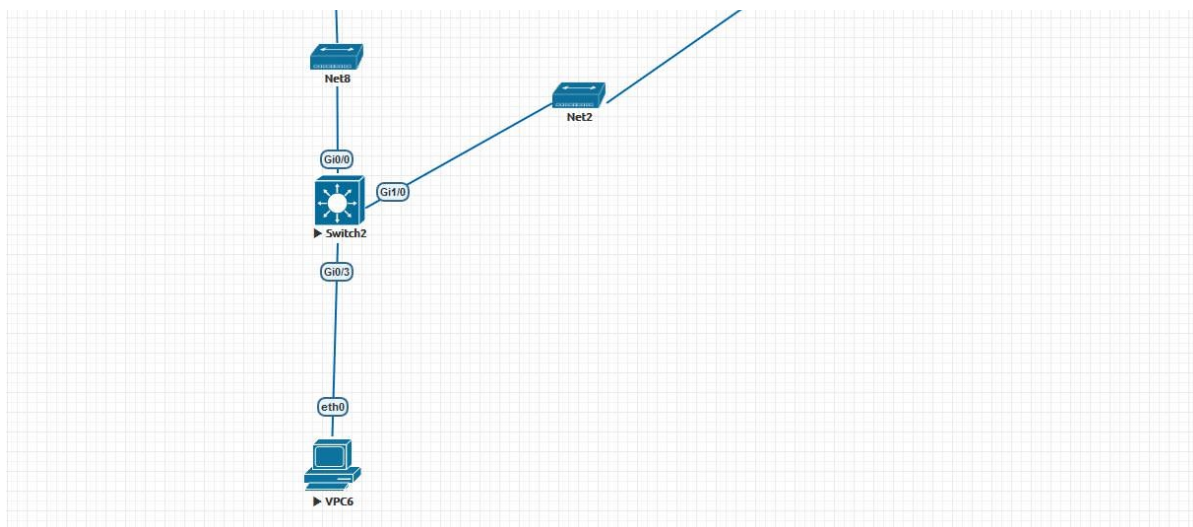
designated Port (неназначенный) — порт, не являющийся корневым, или назначенным. Передача фреймов данных через такой порт запрещена. В нашем примере, порт Gi1/0 является неназначенным.

Disabled Port — порт который находится в выключенном состоянии.

Таймеры и сходимость протокола STP

После того, как STP завершил построение топологии без петель, остается вопрос — Как определять изменения в сети и как реагировать на них? Сообщения BPDU при помощи которых работает STP, рассылаются Root Bridge каждые 2 секунды, по умолчанию. Данный таймер называется Hello Timer. Остальные коммутаторы получив через свой root port данное сообщение пересылают его дальше через все назначенные порты. Выше сказано более подробно какие изменения происходят с BPDU при пересылки его коммутаторов. Если в течении времени, определенным таймером Max Age (по умолчанию — 20 секунд), коммутатор не получил ни одного BPDU от корневого коммутатора, то данное событие трактуется как потеря связи с Root Bridge. Для того, чтобы более корректно описать сходимость протокола необходимо изменить нашу топологию и поставить между коммутаторами хабы. Мы добавили хабы, чтоб при выходе из строя одного из коммутаторов или выхода из строя линка, другие коммутаторы не определяли это по падению линка, а использовали таймеры:





Перед тем, как начать также важно рассказать подробнее о другом типе сообщения STP — TCN. TCN рассылается коммутаторами в случае изменения топологии — как только на каком-либо коммутаторе изменилась топология, например, изменилось состояние интерфейса. TCN отправляется коммутатором только через Root Port. Как только корневой коммутатор получит TCN, он сразу меняет параметр времени хранения MAC-адресов в таблице с 300 секунд до 15 (для чего это делается будет сказано ниже) и в следующем BPDU, Root Switch проставляет флаг — TCA (Topology Change Acknowledgement), который отправляется коммутатору отправившем TCN для уведомления о том, что TCN был получен. Как только TCN достигает Root Bridge, то он рассылает специальный BPDU, который содержит TCN флаг по всем остальным интерфейсам к другим коммутаторам. На картинке показана структура TCN:

```
▸ Frame 25: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▸ IEEE 802.3 Ethernet
▸ Logical-Link Control
▸ Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Topology Change Notification (0x80)
```


TCN был включен в STP, чтоб некорневые коммутаторы могли уведомлять об изменении в сети. Обычными BPDU они этого делать не могут, так как некорневые коммутаторы не отправляют BPDU. Как можно заметить структура TCN не несет в себе никакой информации о том, что именно и где изменилось, а просто сообщает что где-то что-то изменилось. Теперь перейдем к рассмотрению вопроса о сходимости STP.

Посмотрим, что произойдет если мы отключим интерфейс Gi0/1 на Switch1 и посмотрим при помощи каких механизмов перестроится дерево STP. Switch2 перестанет получать BPDU от Switch1 и не будет получать BPDU от Switch3, так как на Switch3 данный порт заблокирован. У Switch2 уйдет 20 секунд (Max Age Timer), чтоб понять потерю связи с Root Bridge. До этого времени, Gi0/0 на Switch2 будет находится в состоянии Forwarding с ролью Root Port. Как только истечет Max Age Timer и Switch2 поймет потерю связи, он будет заново строить дерево STP и как это свойственно STP начнет считать себя Root Bridge. Он отправит новый BPDU, где укажет самого себя в качестве Root Bridge через все активные порты, в том числе и на Switch3. Но таймер Max Age, истекший на Switch2 также истек и на Switch3 для интерфейса Gi1/0. Данный порт уже 20 секунд не получал BPDU и данный порт перейдет в состояние LISTENING и отправит BPDU с указанием в качестве Root Bridge — Switch1. Как только Switch2 примет данный BPDU, он перестанет считать себя Root Bridge и выберет в качестве Root Port — интерфейс Gi1/0. В этот момент Switch2 также отправит TCN через Gi1/0, так как это новый Root Port. Это приведет к тому, что время хранения MAC-адресов на коммутаторах уменьшится с 300 секунд до 15. Но на этом работоспособность сети не

восстановится полностью, необходимо подождать пока порт Gi1/0 на Switch3 пройдет состояние Listening, а затем Learning. Это займет время равное двум периодам Forward delay timer — $15 + 15 = 30$ секунд. Что мы получаем — при потере связи Switch2 ждет пока истечет таймер Max Age = 20 секунд, заново выбирает Root Bridge через другой интерфейс и ждет еще 30 секунд пока ранее заблокированный порт перейдет в состояние Forwarding. Суммарно получаем, что связь между VPC5 и VPC6 прервется на 50 секунд. Как было сказано несколькими предложениями выше при изменении Root Port с Gi0/0 на Gi1/0 на Switch2 был отправлен TCN. Если бы этого не произошло, то все MAC-адреса, изученные через порт Gi 0/0, оставались бы привязаны к Gi0/0. Например, MAC-адрес VPC5 и VPC7 несмотря на то, что STP завершит сходимость через 50 секунд, связь между VPC6 и VPC5, VPC7 не была бы восстановлена, так как все пакеты предназначенные VPC5, VPC7 отправлялись через Gi0/0. Надо было бы ждать не 50 секунд, а 300 секунд пока таблица MAC-адресов перестроится. При помощи TCN, время хранения изменилось с 300 секунд до 15 и пока интерфейс Gi1/0 на Switch3 проходил состояния Listening, а затем Learning и данные о MAC-адресах обновятся.

Также интересен вопрос, что произойдет, если мы заново включим интерфейс Gi0/1 на Switch1? При включение интерфейса Gi0/1, он, как и подобает, перейдет в состояние Listening и начнет рассылать BPDU. Как только Switch2 получит BPDU на порту Gi0/0, то сразу перевыберет свой Root Port, так как тут Cost будет наименьшим и начнет пересылать трафик через интерфейс Gi0/0, но нам необходимо подождать пока интерфейс Gi0/1 пройдет состояния Listening, Learning до

Forwarding. И задержка будет уже не 50 секунд, а 30.

В протоколе STP также продуманы различные технологии для оптимизации и безопасности работы протокола STP. Более подробно в данной статье рассматривать их не буду, материалы по поводу них можно найти в избытке на различных сайтах.