Лекция 3

Добрый день, дорогие студенты! Я вас рад приветствовать на 3 лекции по компьютерным сетям.

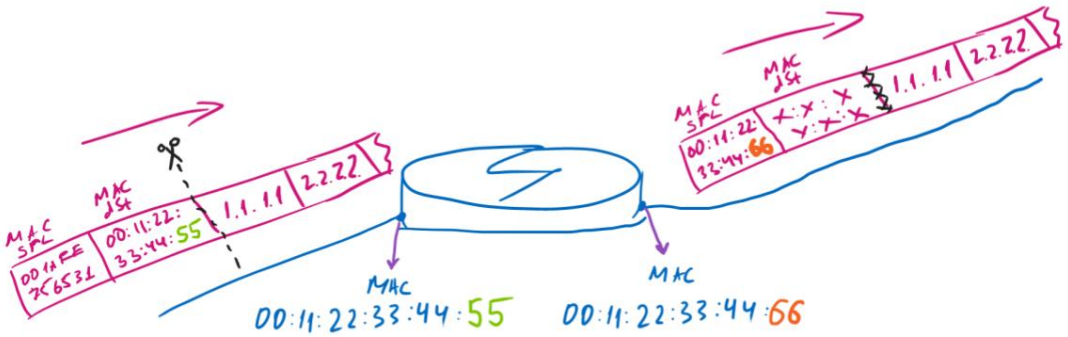
Итак, давайте вспомним, чем завершилось у нас прошлое занятие? На нем мы узнали что такое IP сети, а также мы познакомились с основными принципами маршрутизации трафика между роутерами. И я напоминаю, что основной принцип заключается в том, что когда пакет прилетает на роутер, он смотрит в нём IP Destination адрес, и на основе своей специальной таблицы, которая называется таблица маршрутизации, принимает решение куда направить пакет дальше. И собственно про это давайте более подробно поговорим.

На прошлом уроке мы делали оговорку, что у нас такая таблица была заполнена заранее. И сегодня мы всё-таки теперь поговорим про то как она заполняется. Заполнить эту таблицу можно либо руками сетевого инженера - тогда у нас будет статическая маршрутизация, либо автоматически специальным протоколом - тогда у нас будет динамическая маршрутизация. Также мы сегодня поговорим о протоколе OSPF - динамический протокол маршрутизации. И алгоритме поиска кратчайших путей между сетями - алгоритм Дейкстры. А в конце разберем технологию разделения сетей на уровне L2 - VLAN.

И для начала мы разберем как таблица маршрутизации заполняется руками.

Взаимодействие L2 и L3

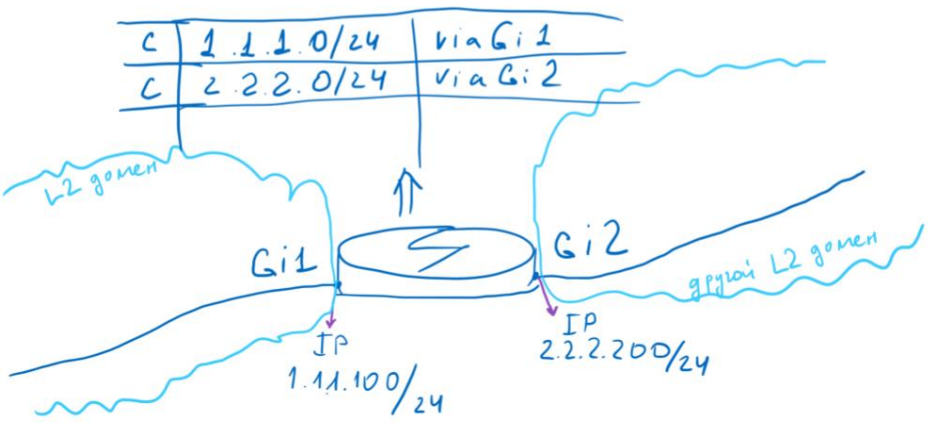
Важный факт №1



Я напомню важный факт , что пакет который приходит на сетевой интерфейс роутера, приходит в определённом L2 бродкаст домене, и в нем у нас Source MAC адрес компа, который создал этот пакет, а Destination MAC Address - это MAC адрес интерфейса роутера. Роутер должен отрезать этот L2 заголовок, так как он предназначается ему, и при пересылке пакета, повесить новый, в котором Source MAC адрес будет MAC Address интерфейса **с которого** должен вылететь от пакет.

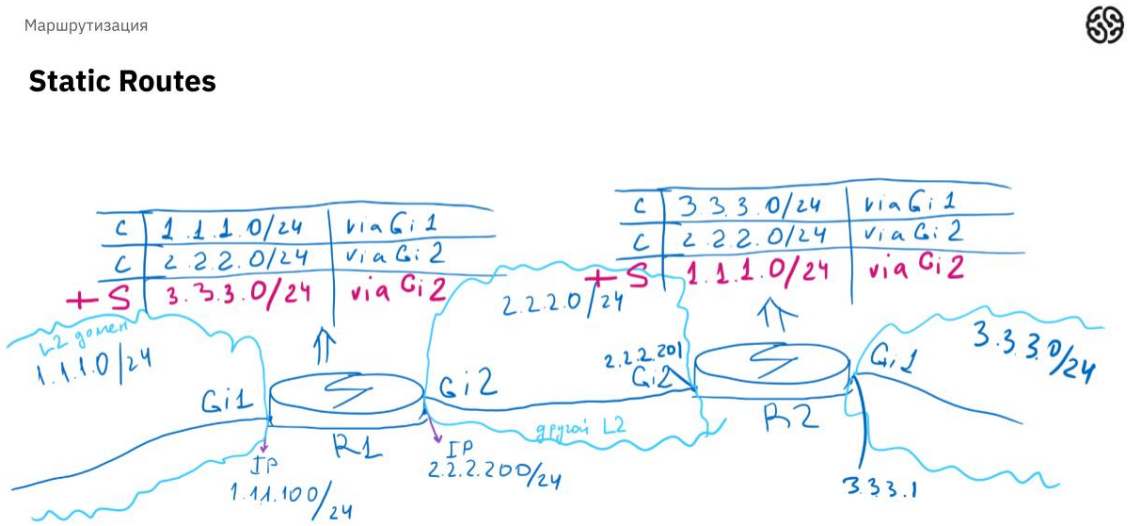
Взаимодействие L2 и L3

Connected Routes



Теперь вспомним, что на сетевых интерфейсах роутера настроены IP адреса, которые принадлежат какой-либо IP сети. Значит информацию об

этих сетях роутер уже может занести в свою таблицу маршрутизации. Это так называемые Connected сети - непосредственно присоединенные сети. На большинстве роутеров в таблице маршрутизации эти сети помечаются буквой “C”. Важно отметить, что на роутере два интерфейса не могут лежать в одной IP сети, иначе теряется смысл маршрутизации, роутер на такую настройку выдаст ошибку.



Трафик между Connected сетями может ходить без проблем, т.к. у нас они всегда лежат в таблице маршрутизации нам известно в какой интерфейс отправить пакет. Но если мы возьмем уже два роутера, то чтобы трафик из сети 1.1.1.0/24 дошел до сети 3.3.3.0/24, на каждом роутере должна быть информация о сети 3.3.3.0/24, т. к. dst IP в пакете будет IP адрес из сети B. На первом роутере этой сети нет (там будут только Connected сети A и C), поэтому мы должны прописать сеть B вручную, на втором роутере сеть 3.3.3.0/24 и так Connected.

Маршруты которые добавляются сетевым инженером на роутер, называются Static routes - статическими маршрутами, и помечаются они буквой “S”.

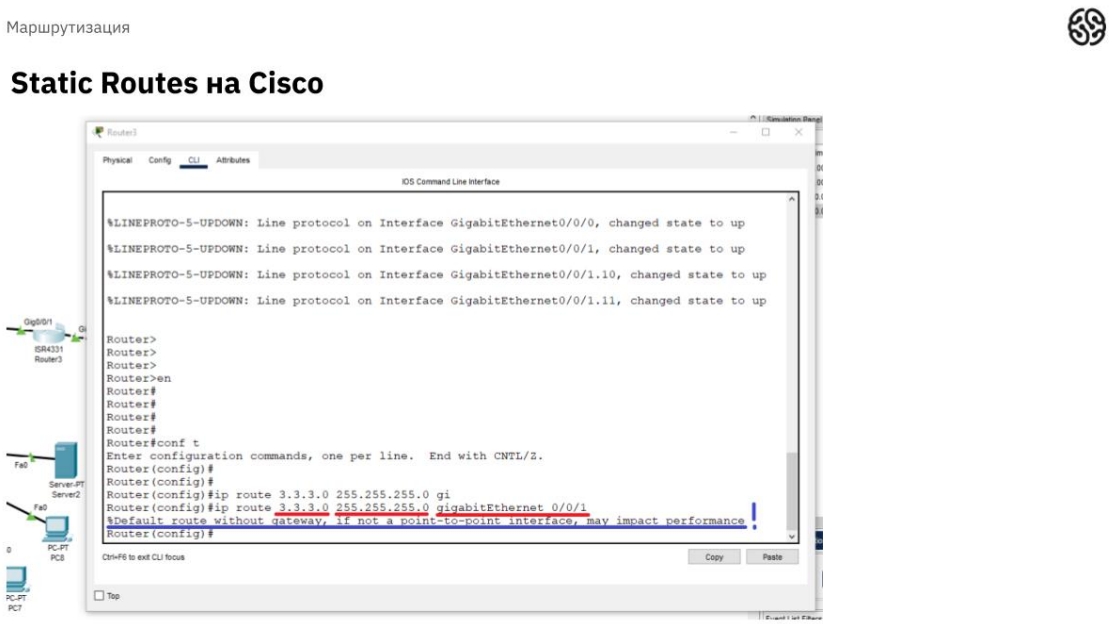
Если мы усложним схему и добавим сколь угодно много роутеров в нашу схему, принцип не изменится, чтобы пакет дошел до сети B, на каждом

роутере должен быть маршрут в эту сеть B, и неважно какой маршрут Connected или Static или какой-либо ещё.

Когда ответный трафик пойдет из сети 3.3.3.0/24 в сеть 1.1.1.0/24, то в пакетах такого трафика в поле IP dst будет уже IP адрес из сети 1.1.1.0/24. Тогда все роутеры на пути следования пакета будут искать в таблице маршрут в сеть 1.1.1.0/24.

Итак, чтобы добавить статический маршрут мы должны указать:

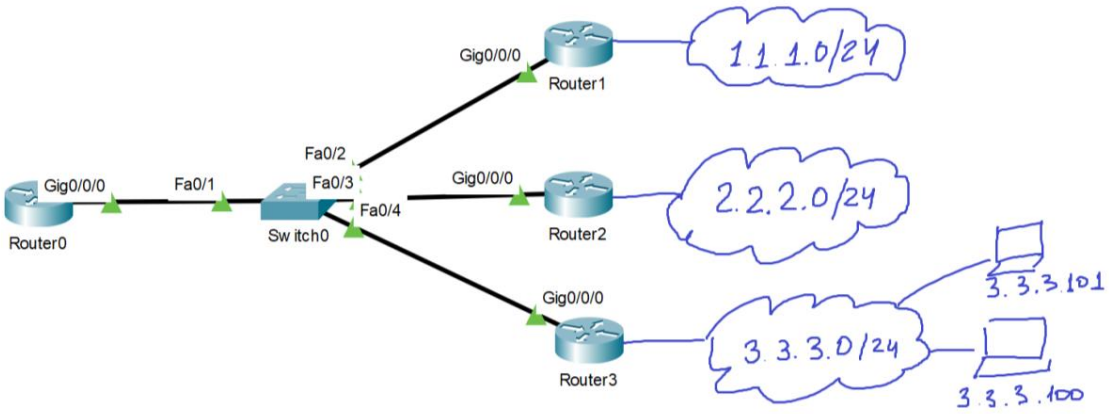
- IP адрес сети, куда отправляются пакеты,
- маску этой сети,
- интерфейс, куда отправлять пакеты.



Вот пример как прописывается маршрут на роутерах Cisco.

Но посмотрите внимательнее, при прописывании маршрута таким образом, Cisco Router выдает на следующее: “Default route without gateway, if not a point-to-point interface, may impact performance”. Что буквально означает предупреждение, что если мы прописываем такой маршрут на интерфейс, на котором у нас может быть несколько роутеров, то это может повлиять на производительность.

Static Routes на Cisco



И действительно, посмотрите на такую схему, если мы на Router0 пропишем маршрут в сеть 3.3.3.0/24 на интерфейс Gi0/0/0, то по факту на какой из трех следующих роутеров ему надо отправить пакет например с dst IP 3.3.3.100? Тут роутер поставит пакет на паузу и сделает ARP запрос, в котором спросит “Who has IP 3.3.3.100?”, который дойдет до всех роутеров в L2 броадкаст домене свича. Эти роутеры обрабатывают ARP запрос немного по-другому, чем конечные хосты с прошлого урока:

- Router1 и Router2 убеждаются что IP адрес из ARP-запроса отсутствует в их таблицах маршрутизации и просто дропнут запрос.
- А Router3 убедится, что IP адрес из ARP-запроса в таблице маршрутизации - есть. И отправит ARP ответ, что IP 3.3.3.100 находится за MAC адресом своего интерфейса Gi0/0/0.

Все это время изначальный пакет к 3.3.3.100 будет висеть на Router0. Более того, такая процедура будет выполняться для любого другого пакета, в котором будет IP dst адрес будет для любого другого хоста из сети 3.3.3.0/24, например для 3.3.3.101.

Static Routes на Cisco

Согласитесь, это не очень эффективно, поэтому роутер нас об этом и предупреждает. Best practice является маршрут, прописанный следующим образом:

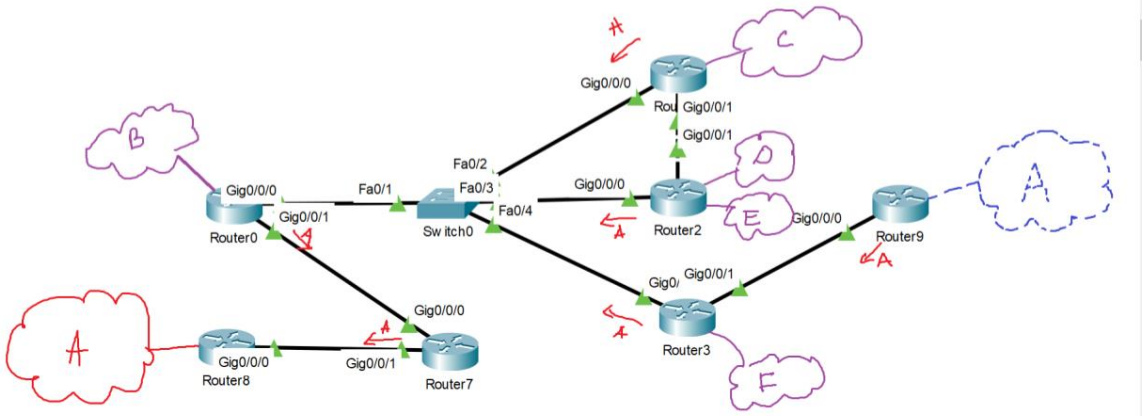
- IP адрес сети, куда отправляются пакеты,
- маску этой сети,
- next hop - IP адрес интерфейса следующего роутера, куда отправлять пакеты.

В этом случае мы сразу точно говорим за каким следующим маршрутизатором - next hop, лежит такая-то сеть с такой-то маской и роутеру не придется искать каким MAC адресом надо снабдить пакет и какому роутеру надо отправить этот пакет дальше. Роутеру надо узнать только MAC адрес next-hop’а и всё. Прописывая маршрут таким образом, также читабельнее становится конфиг роутера, другой сетевой инженер без проблем поймет куда и через какой роутер ведет тот или иной маршрут, в отличии от неявного прописывания за интерфейс. Теперь и в будущем мы будем использовать именно такую запись маршрута.

Небольшое резюме: чтобы у нас пакеты ходили между сетями A и B, (обычно говорят: “Чтобы между сетями A и B была связность”) на всех роутерах между этими сетями должны быть маршруты в таблице маршрутизации и в сеть A и в сеть B. Любые другие сети там необязательны, т.к. роутеры смотрят **только** на поле IP destination в пакете.

Маршрутизация

Static Routes

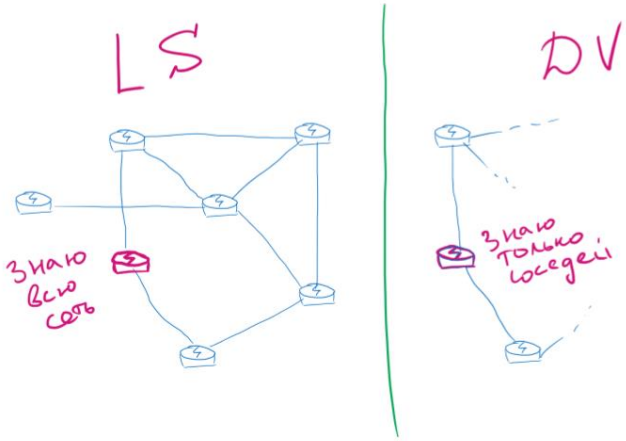


Обычно в корпоративных сетях для большей гибкости и доступности ресурсов, делается связность между всеми сетями. Посмотрите на слайд, допустим у нас относительно небольшая сеть, но чтобы сделать связность между всеми сетями надо прописать большое количество статики. Это неудобно и не гибко. Во-первых, тут есть человеческий фактор, сетевой инженер может просто ошибиться. Во-вторых, представьте что стоит задача сеть А перенести за другой роутер на другом конце сети, для этого надо переписать всю статику для этой сети на всех роутерах. Согласитесь, не очень удобно.

Напрашивается вопрос, а нет ли автоматизированного решения для заполнения таблиц маршрутизации? И ответ - есть. Это динамические протоколы маршрутизации.

Маршрутизация

Dynamic Routing



Динамические протоколы маршрутизации бывают двух видов - Link State и Distance Vector.
Link State - это протоколы, которые основываясь на состоянии каналов связи, собирают на каждом роутере полную карту сети, и уже на основе этой карты сети выстраивается таблица маршрутизации.
В Distance Vector протоколах же полной карты сети на каждом роутере нет, есть только информация о конкретной сети и ближайших роутерах-соседях, через которые эта сеть достижима. На основании различных характеристик выбирается самый оптимальный ближайший next-hop в эту конкретную сеть, который и заносится в таблицу маршрутизации.

Link State хороши тем, что они реагируют на каждое изменение отдельного линка в сети. Порвался где-то кабель в сети? Карта сети и таблицы маршрутизации перестроятся. Добавился новый линк в новую сеть? Информация о ней расползется по роутерам и таблицы маршрутизации обновятся. Но минус в том, что для этого должна быть соответствующая вычислительная мощность, и чем больше у нас сетей и узлов, тем больше информации надо хранить и обрабатывать, причем зависимость эта возрастает логарифмически. Такие протоколы хороши в ограниченных корпоративных сетях, представьте что мы использовали бы такой протокол для маршрутизации трафика в Интернете, где карта сети постоянно меняется, а количество роутеров исчисляется сотнями тысяч.

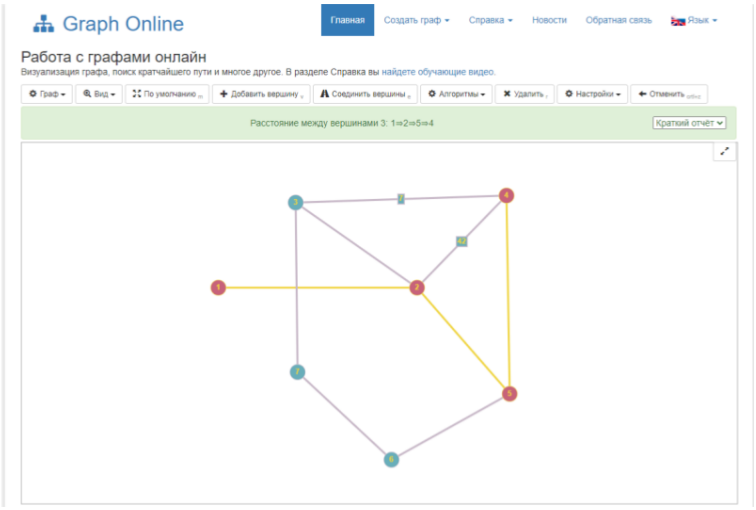
Distance Vector протоколы этих недостатков лишены, на роутере только информация о ближайших next-hop’ах и тех маршрутах, о которых они рассказывают. Много вычислений тут проводить не надо. Но и к изменениям на сети эти протоколы не чувствительны. Такие протоколы хороши в больших сетях, таких как интернет. И основной связующий протокол в интернете, BGP, является Distance Vector.

Есть два популярных LS (Link State) протокола маршрутизации:

- OSPF - Open Shortest Path First - самый популярный, чаще встречается в корпоративных сетях, его мы и рассмотрим сегодня на уроке, и с ним поработаем на семинаре.
- ISIS - Intermediate System to Intermediate System - менее популярный, чаще используется во внутренних провайдерских сетях, но тоже может часто встречаться.

OSPF и IS-IS открытые протоколы, т.е. их может реализовывать любой вендор (производитель) любого оборудования.
Главная задача LS протокола - создать консистентную - то есть согласованную и непротиворечивую базу данных о карте сети на каждом роутере. Ведь роутеры не люди, они не могут по картинке, как мы с вами, сообразить где кратчайший маршрут а где нет. Вся “карта” сети хранится на роутерах в виде базы данных - таблиц, которые в себе хранят граф сети. Под графом сети - понимается математическое представление карты сети.

Граф



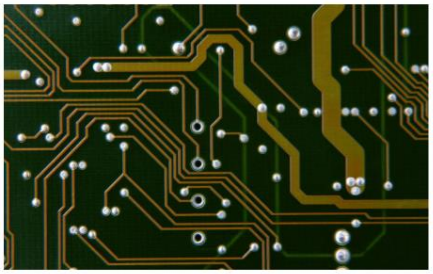
В математике граф (тут можно продемонстрировать на http://graphonline.ru/?graph=pCUPpVSJRGuwDHHS) - это множество объектов, которые называются вершинами графа, и множество ребер - линий, соединяющих какие-либо из этих вершины. При этом не важно как эти линии проведены, важен именно сам факт соединения.

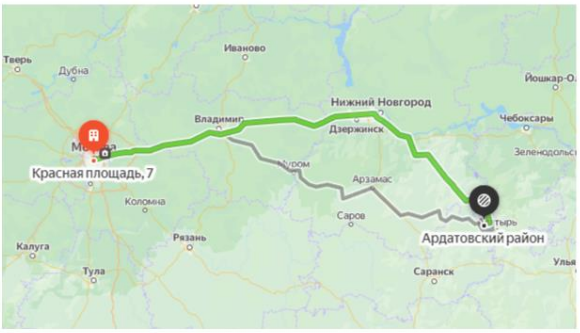
Ребро может иметь вес - некоторое значение, описывающее приоритет этого ребра. В сетях вес описывает приоритет соединения, например если линк с пропускной способностью 1Gbit/s, то он будет более приоритетным, чем линк с пропускной способностью 100Mbit/s.

Граф очень хорошо описывает модель компьютерной сети, ведь мы можем представить что вершины графа - это роутеры и сети, а физические линии связи - это ребра. В математике есть целый раздел посвященный графам - теория графов. Учитывая, что математики проделали огромную работу по графам и в том числе разобрали алгоритмы, как эффективно искать кратчайшие пути от любой вершины графа к любой другой вершине графа - грех не воспользоваться этой моделью, ведь нам в сетях для построения маршрутизации как раз это и нужно - знать из какой сети, в какую сеть, ведет самый кратчайший и выгодный маршрут. В теории графов кратчайший маршрут ищется с помощью алгоритма, который изобрел нидерландский учёный Дейкстра.

Маршрутизация

И тут графы





Изначально он придумал этот алгоритм для поиска кратчайших путей на электрических платах между элементами электрических цепей для экономии меди в дорожках, но в дальнейшем его алгоритм был усовершенствован и стал использоваться и в других отраслях - например в навигации (привет маршрутам в Яндекс Картах) и в наших компьютерных сетях.

Можно подытожить и сказать, что работа LS алгоритмов разделена на 2 этапа:

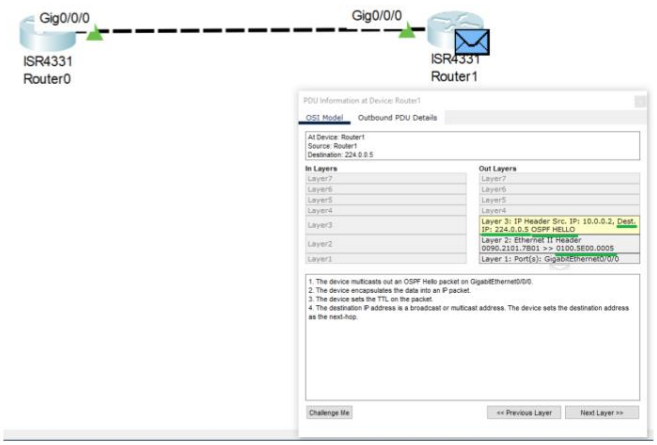
1. Роутеры обмениваются информацией, постепенно выстраивая одинаковую “карту сети” у себя.
2. После этого каждый роутер вычисляет кратчайший путь от себя до каждой IP-сети с помощью алгоритма Дейкстры и формирует таблицу маршрутизации.

Давайте обзорно рассмотрим как на роутерах работает OSPF.

OSPF по дизайну задумывался изначально как простой в настройке протокол, где вся сложность скрыта от администратора. По сути надо просто включить OSPF процесс на роутере и указать сети, участвующие в OSPF процессе, т.е. те сети о которых роутер будет рассказывать своим соседям, а также те сети через которые он будет искать соседей.

OSPF

Hello packet



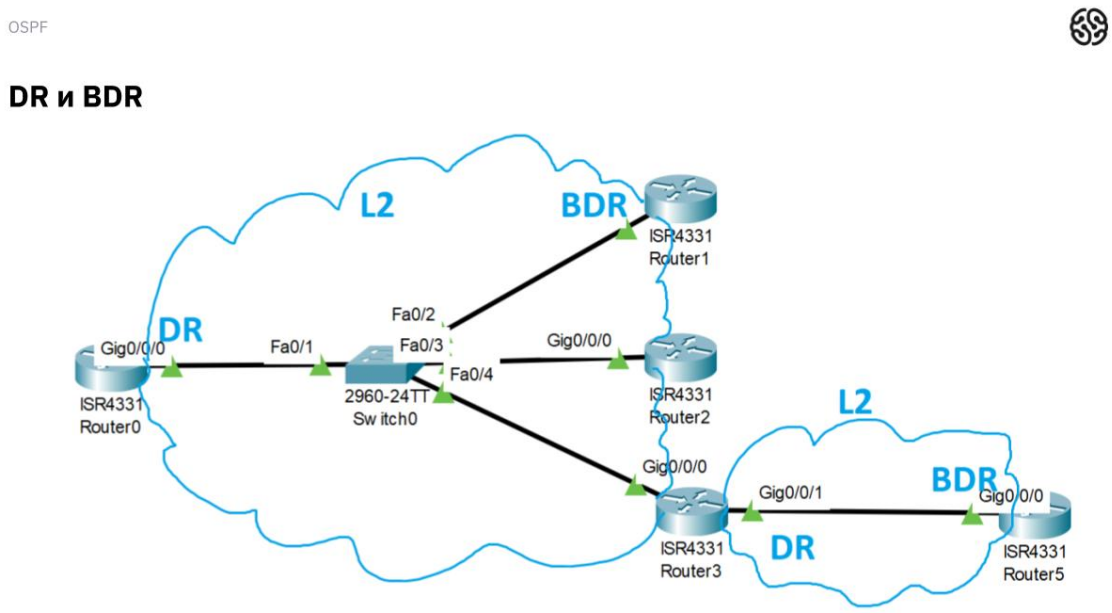
Давайте на примере схемы на слайде, рассмотрим, как отрабатывает OSPF.

1. После того как мы включим OSPF и задействуем сеть какого-либо интерфейса, Роутер 1 начинает рассылать специальный hello-пакет с этого интерфейса раз в 10 секунд.

В этом hello-пакете он сообщает свой уникальный 32-битный номер - Router ID и перечисляет своих соседей, которых пока нет. Также в этом пакете специальный IP dst 224.0.0.5 и специальный MAC dst 01:00:5E:00:00:05. Это специальные, зарезервированные стандартом, адреса, попадая на свитч, этот пакет рассылается во все порты, почти как броадкаст, т.к. все свитчи по умолчанию распознают их. Это помогает роутеру достучаться до своих соседних роутеров, на которых тоже должен быть включен OSPF. Time to Live в этом пакете равен единице, что означает, что за пределы одного L2 бродкаст домена этот пакет не выйдет. Таким образом hello пакет будет доставлен всем своим ближайшим роутерам в радиусе своего броадкаст домена, и, если интерфейсы этих роутеров включены в OSPF, то этот hello-пакет будет обработан. Так и работает обнаружение соседей.

2. Роутер 0, получив hello пакет, понимает, что у него есть сосед Роутер 1 и добавляет его в свой список соседей. Далее, Роутер 0 посылает свой Hello пакет со списком своих соседей, в котором уже есть Роутер 1.

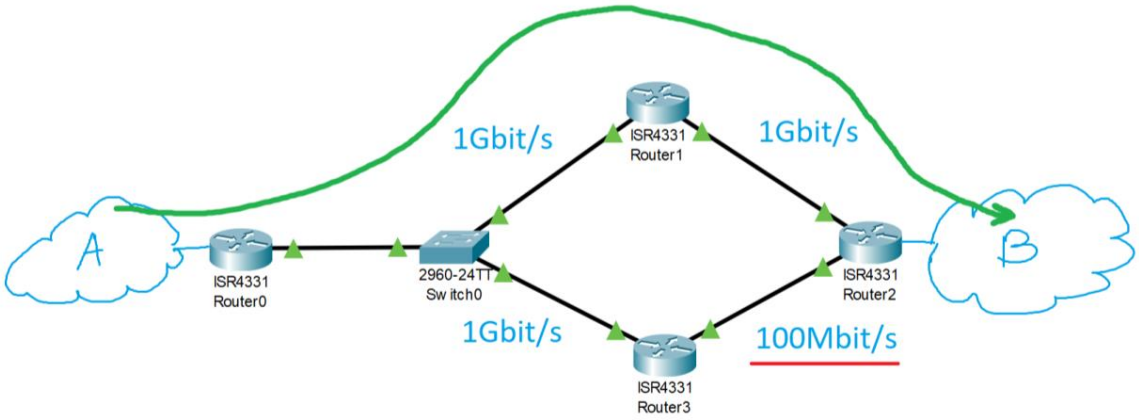
3. Роутер 1, получив hello пакет и обнаружив себя в соседях, понимает что здесь уже можно устанавливать OSPF соседство - OSPF Adjacency.



4. Далее, если в одном L2 бродкаст домене больше двух роутеров, то это означает, что у нас несколько роутеров имеют выход в одну и ту же сеть. В этой L2 сети у нас выбирается специальный роутер или Designated router (DR). Делается это для оптимизации работы алгоритма Дейкстры, так как сам алгоритм чувствителен к количеству узлов в графе, т.к. чем больше связей в графе, тем больше вычислительной мощности требуется для вычисления кратчайшего пути. Поэтому, чтобы сократить количество связностей (соседства), между этими роутерами выбирается один и с ним все остальные роутеры устанавливают соседство. На случай его поломки, также выбирается Backup DR, с ним тоже устанавливается соседство.

OSPF

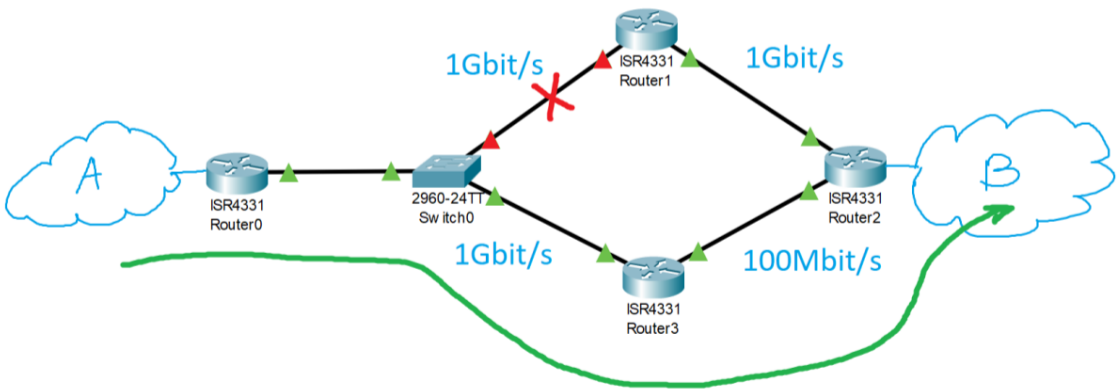
OSPF выбор оптимального пути



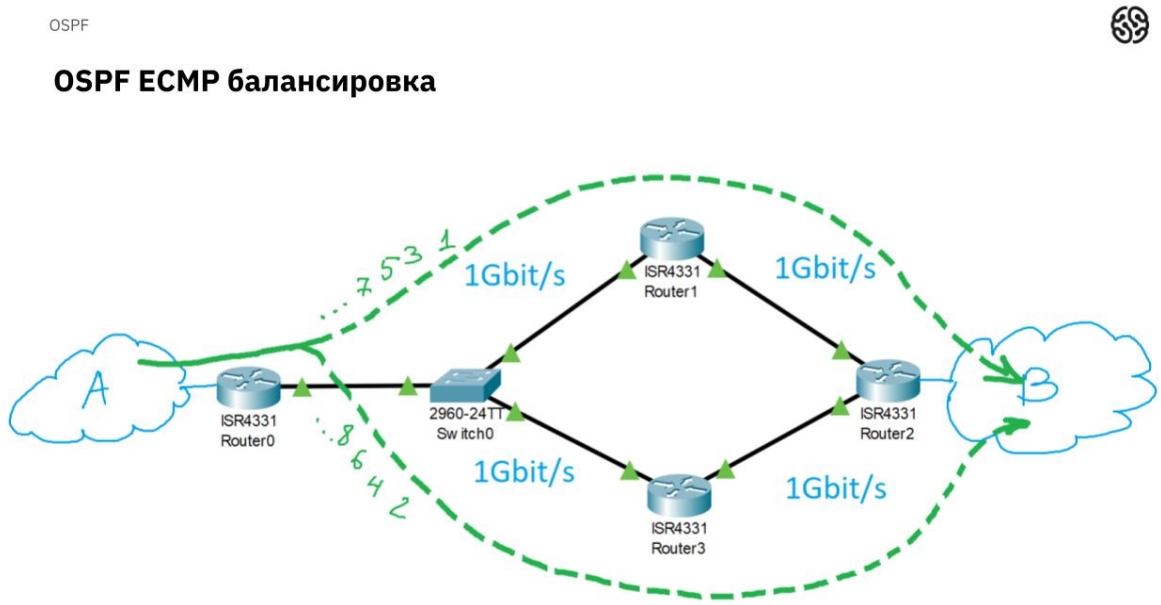
5. После чего идет рассылка специальных сообщений, по которым на каждом роутере собирается наш граф сети. Как только граф сети собран, они переходят в состояние “FULL VIEW”. Это означает, что можно запускать алгоритм Дейкстры и строить таблицу маршрутизации. Маршруты полученные таким образом, помечаются буквой “O”. Маршруты строятся с учётом стоимости пути, т.е. если у нас в одну сеть ведут два пути, и на первом пути все линки 1 Gbit/s, а на втором пути есть участок 100 Mbit/s, то выберется первый путь, как более оптимальный.

OSPF

OSPF резервирование



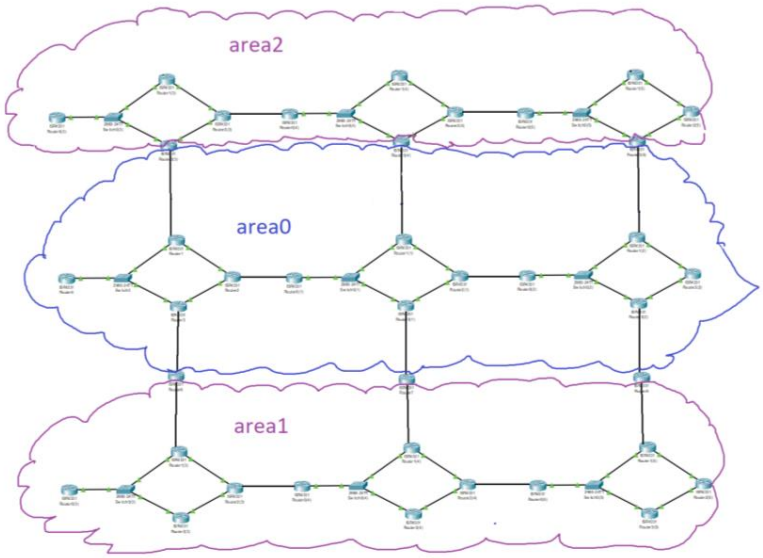
При обрыве какого-либо линка, роутер, на котором интерфейс упал, рассылает специальное сообщение, которое обновит наш граф. Роутеры пересчитают таблицу маршрутизации, и если есть резервный канал связи, то пакеты пойдут через него.



Также стоит отметить, что если у маршрутизатора будет два одинаковых по стоимости пути, то по умолчанию включится балансировка: один пакет пойдет по первому пути, второй по второму, третий по первому, четвертый по второму, и т. д. Такая балансировка называется ECMP - Equal Cost Multiple Path.

OSPF

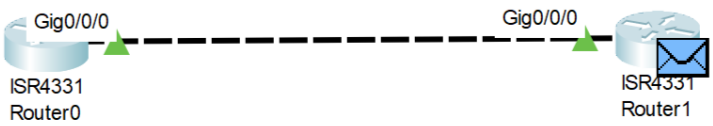
OSPF Areas



Я упомянул, что алгоритм Дейкстры чувствителен к количеству узлов и связей, и чем больше узлов в сети, тем больше необходимо вычислительных ресурсов для его работы, при этом зависимость логарифмическая. Чтобы упростить работу для этого алгоритма, большие сети разбиваются на зоны (area). В каждой зоне отрабатывает свой OSPF процесс, а затем маршрутная информация между зонами распространяется с помощью специальных роутеров - Area Border Router. Главной зоной является area 0, все остальные должны быть соединены только с ней.

OSPF

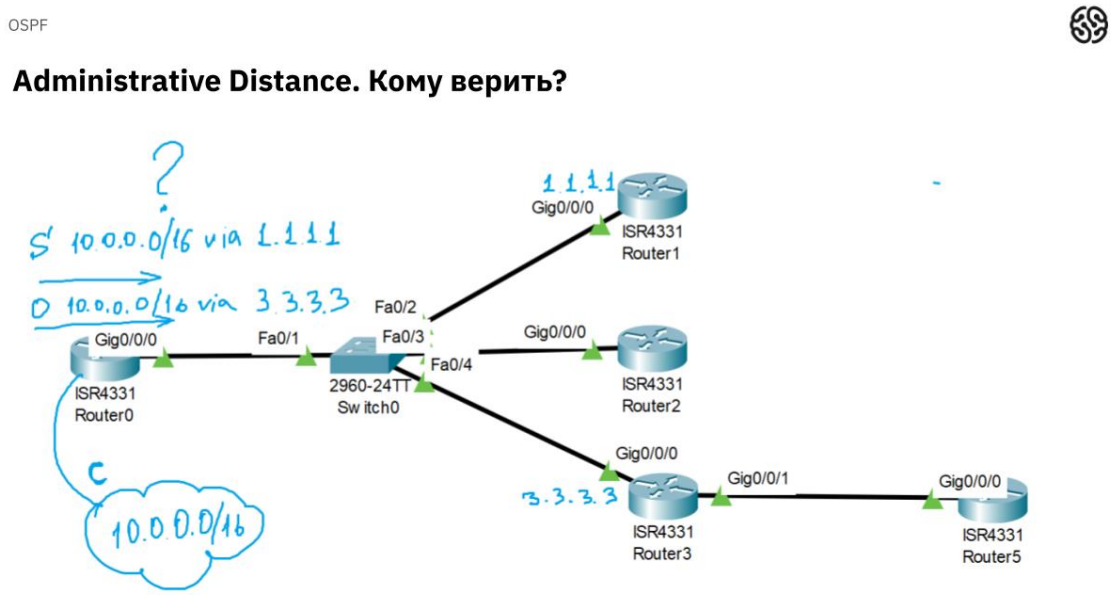
OSPF. Условия связности (соседства).



1. Одинаковый Hello-interval
2. Одинаковый Dead-interval
3. Одна IP сеть на встречных интерфейсах
4. Одна Area на встречных интерфейсах
5. Одинаковый MTU на встречных интерфейсах
6. Уникальный Router ID

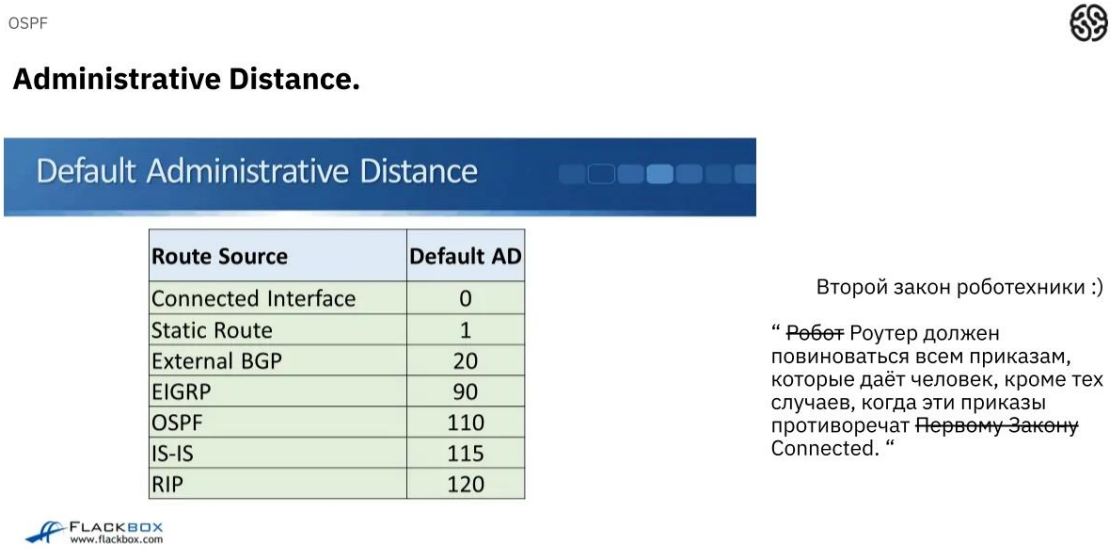
Учитывая алгоритм работы OSPF, для установления связности OSPF между роутерами, должно выполняться несколько требований:

1. Одинаковый Hello-interval, по умолчанию на роутерах он равен 10 секундам. По нему роутеры отслеживают, что роутер все еще доступен и живой. А если он будет разным, то один роутер для другого будет считаться недоступным, и наоборот, что привело бы к некорректной работе OSPF, поэтому связности не будет.
2. Т.к. на сети вообще могут теряться пакеты, то случайная потеря одного Hello пакета привела бы к перестроению карты сети и перезапуску алгоритма Дейкстры. Поэтому есть такое понятие как Dead-interval - количество допустимых потерянных Hello пакетов, после которых сосед считается точно недоступным. Он тоже должен быть одинаков на роутерах, иначе связности не будет.
3. Интерфейсы, подключенные друг к другу, должны быть в одной IP сети.
4. Номера зон интерфейсов должны совпадать.
5. Должны совпадать MTU на встречных интерфейсах.
6. У каждого роутера должен быть уникальный Router-ID.



Итак, получается у нас могут быть маршруты Connected, Static, OSPF. И предположим, что у нас есть сеть 10.0.0.0/16, которая на роутере присутствует как Connected. Допустим, что эту сеть также прописал администратор (по ошибке) за каким-нибудь роутером 1.1.1.1. И

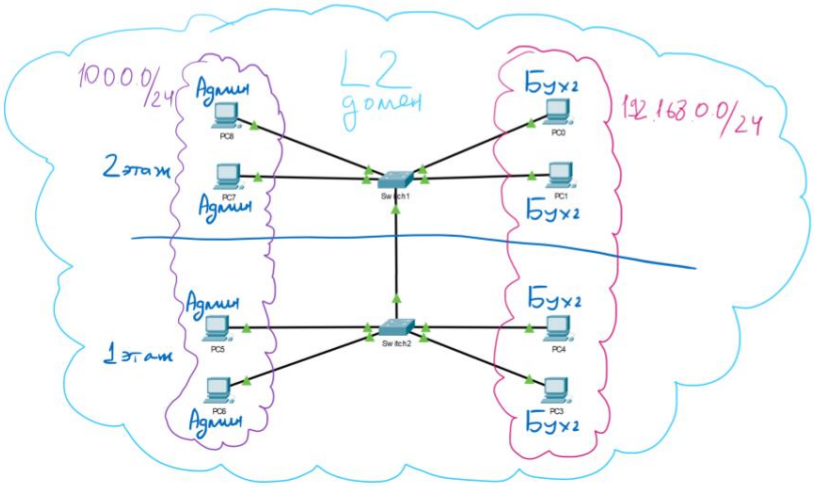
допустим, что нам эту же сеть анонсирует другой роутер 2.2.2.2 по OSPF. Какой маршрут попадет в таблицу маршрутизации?



Тут на сцену выходит Administrative Distance - это внутренняя метрика на роутере, которая отвечает за приоритет маршрутов, полученных от разных протоколов, если маршруты в одну и ту же сеть. Чем она меньше, тем приоритетнее маршрут. У Connected она вообще равна нулю, что логично, ведь это маршрут полученный из настройки на интерфейсе. У Static она равна 1, ведь администратор знает что делает, и если делает, то неспроста, значит это будет самый важный маршрут. Почти все как во втором законе робототехники Азимова: “Робот должен повиноваться всем приказам, которые даёт человек, кроме тех случаев, когда эти приказы противоречат ~~Первому Закону~~ Connected”. У маршрутов полученных по OSPF это значение на Cisco роутерах (и на большинстве других производителей) равно 110. Так что в нашем случае маршрут Connected попадет в таблицу маршрутизации, остальные нет.

Также важно запомнить еще одно правило, что чем более точен маршрут(чем больше “1” в маске), тем он приоритетнее, независимо от Administrative Distance. Т.е. если у нас есть маршрут полученный по OSPF 10.0.0.0/24 via 3.3.3.3 и есть Static 10.0.0.0/16 via 1.1.1.1,то пакеты, в которых dst IP из диапазона 10.0.0.0-10.0.0.255, пойдут на 3.3.3.3, а пакеты в который dst IP из диапазона 10.0.1.0-10.0.255.255 пойдут на 1.1.1.1.

VLAN. Один L2 домен на всех - плохо.

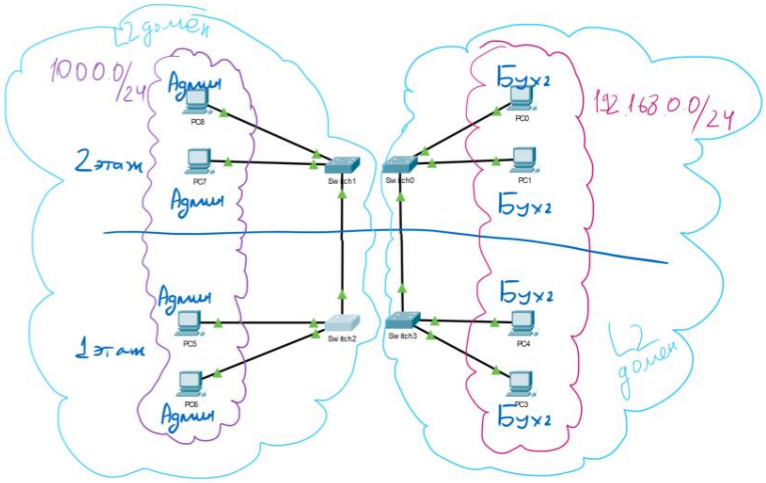


Теперь, когда мы вооружены знаниями про L2 и L3, давайте поговорим еще про одну технологию, которая позволяет разделять L2 броадкаст домены и делать в каждом отдельном L2 домене отдельную IP сеть. Ведь до этого мы жили в парадигме, когда L2 домен был один. И представьте, что в нашей сети есть связка свитчей (например по одному на каждый этаж нашего офиса), на каждом этаже сидят бухгалтеры и сисадмины. У бухгалтеров сеть 192.168.0.0/24, у сисадминов сеть 10.0.0.0/24. Если у нас будет две IP-сети в одном L2-домене, то это не очень безопасно:

1. При каждом запросе ARP при поиске соседей у нас все участники сети будут видеть эти запросы. Все будут знать кто кого ищет.
2. Предположим, что в IP-сети администраторов размещены важные сервера, доступ на которые возможен только из их IP-сети 10.0.0.0/24. В таком случае, если компьютер бухгалтеров будет взломан - то хакеру достаточно будет просто взять IP адрес из сети админов и у него будет доступ ко всем этим защищенным серверам, т.к. все по тому же протоколу ARP он без проблем обнаружит эти сервера.

OSPF

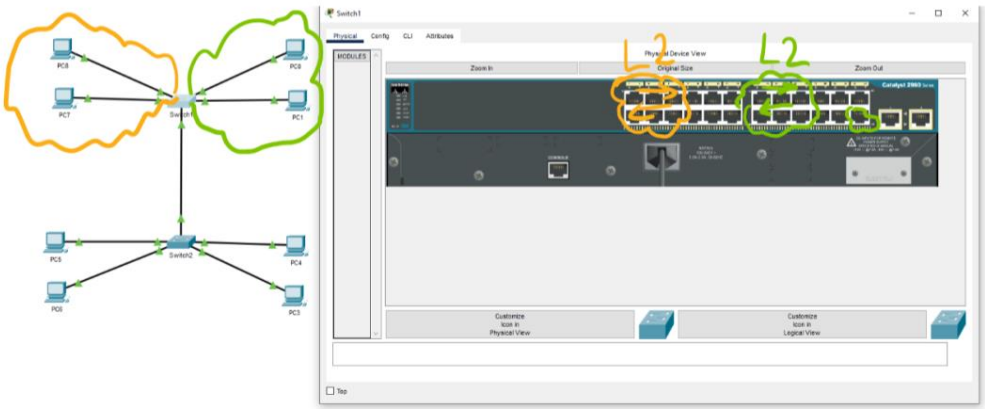
VLAN. Каждому свой L2 домен - хорошо.



С точки зрения хорошего дизайна сети, надо стараться разделять L2 домены, и делать отдельные для бухгалтеров и для админов. Как мы можем это сделать? Ну самое простое решение в лоб - купить отдельные свитчи для бухгалтеров и для админов, тогда L2 домены будут изолированы друг от друга физически. Но в плане стоимости и масштабируемости это решение не очень хорошее. А если у нас не два этажа, а 10? А если у нас не только админы и бухгалтеры, но и программисты, девопсы, HR’ы и т.д.

VLAN

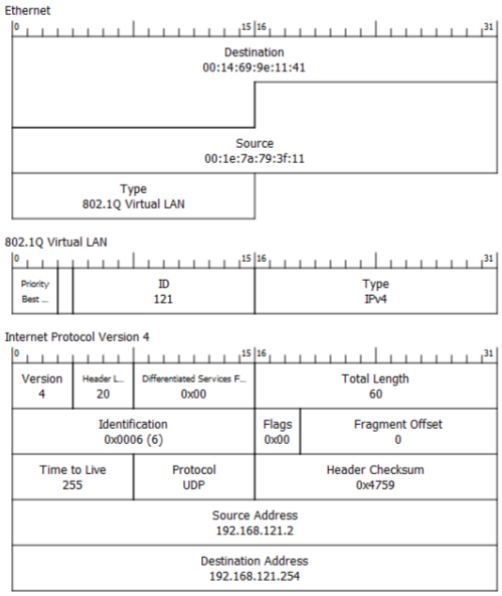
VLAN на одном коммутаторе



Поэтому придумали технологию VLAN - Virtual Local Area Network, которая позволяет из одного коммутатора сделать как-бы несколько независимых коммутаторов с отдельными L2 броадкаст доменами. По сути, мы просто берем на коммутаторе несколько портов, объединяем их и объявляем их одним доменом, со своей таблицей MAC адресов. И берем другие порты, которые также объявляем другим доменом.

VLAN

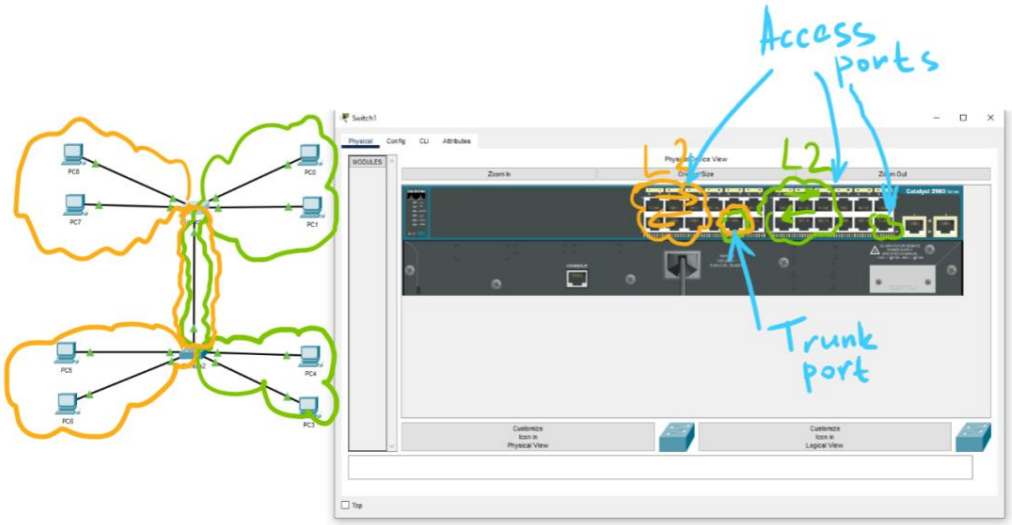
VLAN тег.



Для того, чтобы коммутатор не перепутал домены при обмене трафиком, все пакеты, которые приходят на интерфейс снабжаются специальной вставкой - тегом, который идет между заголовком L2 и L3. Этот тег занимает всего 32 бита, из которых первые 16 бит выделено под VLAN, а другие 16 бит выделено под обозначение следующего протокола.

VLAN

Access и Trunk порты.

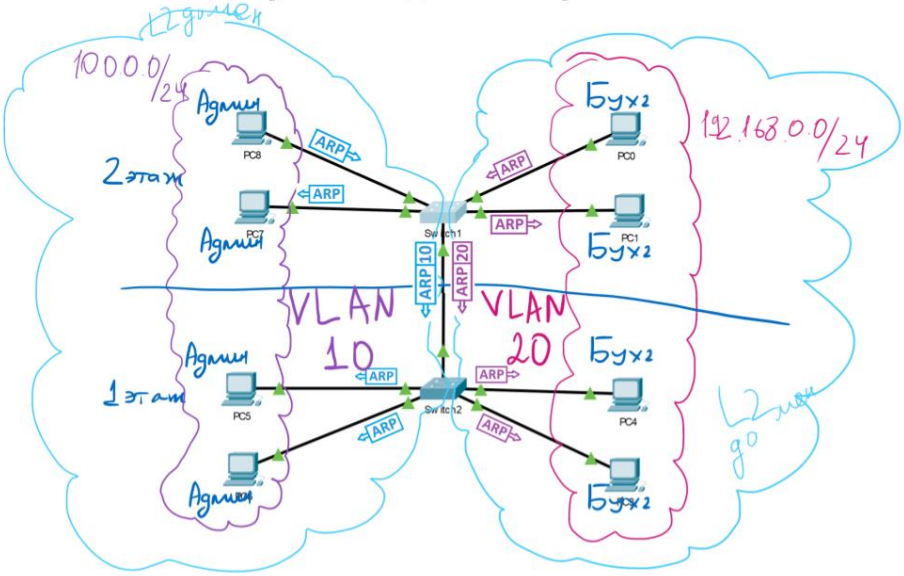


Для того чтобы настроить технологию VLAN, надо первым делом определить те порты, которые будут образовывать отдельный броадкаст домен. На этих портах на весь входящий трафик будет навешиваться тег с определенным номером VLAN, а на весь исходящий - сниматься. Таким образом мы не настраиваем оконечное оборудование, мы настраиваем только сами коммутаторы. Хосты, которые воткнуты в свич ничего не знают о VLAN’ах, которым они принадлежат. Такие порты называются Access Ports. Порт должен настраивается в режиме Access если он соединен с оконечным оборудованием. Один Access порт может принадлежать только одному VLAN. Если у нас один свитч, достаточно настроить только Access порты и станет несколько разграниченных броадкаст доменов.

Но теперь посмотрим на соединение между двумя свитчами - там по одному линку должен ходить трафик нескольких доменов, поэтому каждый пакет надо разбирать какому VLAN он принадлежит, следовательно там должны ходить пакеты только с тегами. Такие порты именуются Trunk Ports. Порт настраивается в режиме Trunk, если он соединен с другим коммутатором например, или с оборудованием, которое разбирает VLAN’ы - роутеры или сервера со специальными сетевыми картами. Один Trunk порт может вмещать в себя несколько VLAN.

VLAN

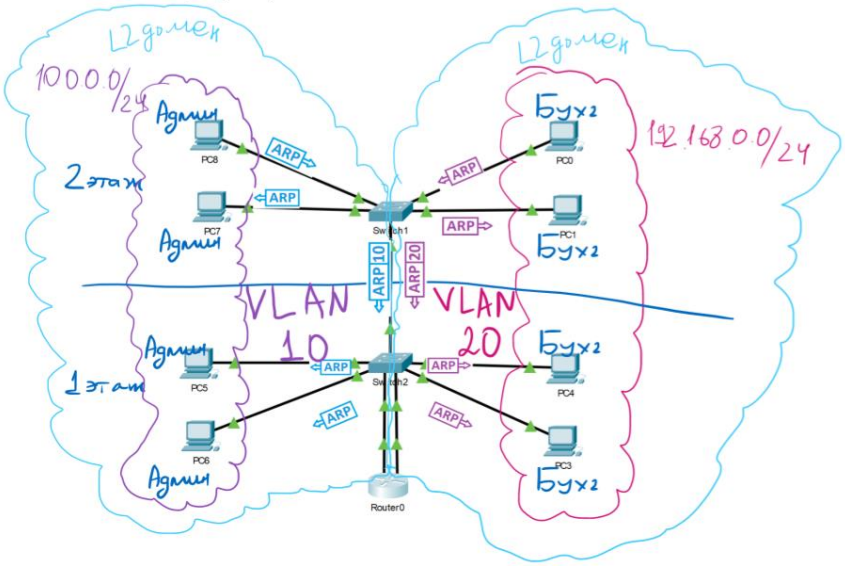
VLAN. Каждому свой L2 домен - хорошо.



Благодаря определению режима портов, мы можем разделять броадкаст домены и изолировать трафик друг от друга. Представим теперь что в нашем начальном примере админы с сетью 10.0.0.0/24 висят во влане 10, а бухгалтеры с сетью 192.168.0.0/24 висят во влане 20. Если бухгалтер создаст броадкаст запрос, то как только он придёт на коммутатор, то он его снабдит меткой с ID 20, и коммутатор направит его уже не во все порты, а только в те Access порты, которые принадлежат влану 20, и только в те Trunk порты, в которых есть влан 20. Второй коммутатор поступит также, т.о. этот броадкаст запрос даже не дойдет до админских компов. И если даже бухгалтерский комп будет взломан, то теперь хакер, даже если будет использовать адрес из сети 10.0.0.0/24, не сможет получить доступ до серверов. Это и называется изоляцией трафика на уровне L2 с помощью VLAN.

VLAN

VLAN на роутере.

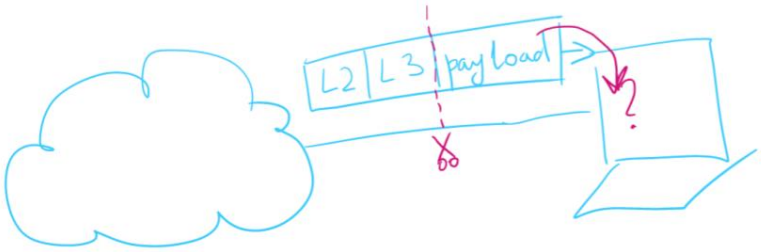


Теперь немного усложним нашу сеть, ведь для выхода из своих сетей и админам и бухгалтерам нужен роутер. По идее мы можем от коммутатора протянуть два линка до роутера. На коммутаторе сделать для этих линков Access порты в соответствующих VLAN’ах, а на роутере настроить два физических порта, один в сети 192.168.0.0/24 а другой в сети 10.0.0.0/24. Но опять таки, а если вланов будет много? Для каждого тянуть отдельную физическую линию не совсем целесообразно. Поэтому роутеры тоже умеют разбирать заголовок VLAN, и, в зависимости от номера VLAN, направлять пакет на специальный подынтерфейс (sub-interface), которых можно много создать на одном физическом интерфейсе. На каждом подынтерфейсе прописывается свой IP адрес для конкретной сети, и они обе становятся Connected.

Итак, подведём итог. Сегодня мы рассмотрели важную тему маршрутизации трафика, как статическую так и динамическую. Рассмотрели их принципы и на семинаре научимся работать с ними обеими, а также настраивать вланы.

На следующем занятии

Кому payload?



Если правильно настроить маршрутизацию на всех роутерах в сети, мы можем получить доступ от одного хоста к любому нужному нам другому хосту в сети. Эти два хоста смогут обмениваться пакетами. Получив такой пакет, хост проанализирует L2 и L3 заголовки, по ним он поймет что пакет предназначен ему и достанет оттуда payload. И что с ним делать дальше? Какому приложению передать на обработку? Вашему веб-браузеру? Клиенту Zoom? Может быть мессенджеру Telegram? Если хост будет пытаться перебирать все возможные варианты, то на это будет много времени уходить в пустую. Эту проблему решает уровень L4, который называется транспортным, его мы и будем рассматривать на следующей лекции. Приходите, будет интересно!

Глоссарий:

Connected Nets - непосредственно настроенные сети на роутере.

Static Routes - “статика” - статический маршрут, который прописывает администратор.

Next Hop - IP адрес интерфейса следующего роутера, куда необходимо отправлять пакеты.

Связность сетей A и B - состояние, когда сеть настроена таким образом, что пакеты могут идти из сеть A в сеть B и наоборот.

Динамические протоколы маршрутизации - специальные протоколы, которые позволяют автоматизировать заполнение таблицы маршрутизации на роутере.

Link State протоколы - вид динамических протоколов маршрутизации, которые собирают всю информацию о топологии сети, и на её основе вырабатывают таблицу маршрутизации.

Distance Vector протоколы - вид динамических протоколов маршрутизации, где роутеры принимают маршруты только от своих ближайших соседей и уже потом заносят самые лучшие маршруты в таблицу маршрутизации.

OSPF - Open Shortest Path First - Link State протокол маршрутизации, основанный на алгоритме Дейкстры, популярен в корпоративных сетях.

ISIS - Intermediate System to Intermediate System - Link State протокол маршрутизации, также основанный на алгоритме Дейкстры, популярен во внутренних провайдерских сетях.

Вендор - производитель оборудования.

Консистентность - согласованность, непротиворечивость.

Граф - математическое понятие, включающее в себя множество вершин и ребер, соединяющих эти вершины.

Алгоритма Дейкстра - алгоритм поиска кратчайшего пути от одной вершины графа до всех других вершин графа.

Hello-пакет - специальный пакет для обнаружения соседей в протоколе OSPF, рассылается постоянно раз в [hello-interval] секунд.

Hello-interval - интервал с которым рассылается hello-пакет (по умолчанию 10 секунд).

Dead interval - сколько hello-interval можно пропустить, пока мы будем считать соседа “мертвым”, т.е. по той или иной причине выбывшим из сети.

Router ID - уникальный 32-битный номер роутера, идентифицирующий его в сети с OSPF. Обычно записывается в формате IP адреса (но не является им!).

OSPF Adjacency - установившееся OSPF соседство, когда роутеры увидели друг друга в соседях в hello пакетах.

Designated Router (DR) - специально назначенный роутер в одной сети, где есть больше двух роутеров, с которым связываются все остальные, для упрощения работы алгоритма Дейкстры.

Backup DR (BDR) - резервный Designated Router, с которым тоже все связываются на случай выхода из строя DR.

Full View - состояние OSPF процесса, когда на роутере собрана вся информация о сети и можно запускать алгоритм Дейкстры.

ECMP - Equal Cost Multiple Path - балансировка пакетов по нескольким маршрутам одинаковой стоимости.

Area - специально выделенная область сети со своим внутренним отдельным OSPF, необходима в больших сетях для упрощения расчета кратчайшего пути. Area 0 - главная area, к ней присоединяются все остальные.

Administrative Distance - специальная метрика, которая решает какой протокол главнее, в случае если есть несколько одинаковых маршрутов от разных протоколов.

VLAN - Virtual Local Area Network - технология разделения броадкаст доменов на коммутаторах.

Access Ports - тип порта, который принадлежит только одному VLAN. На трафик, который приходит на порт, вешается тег - специальная VLAN вставка в заголовок пакета. С трафика, который выходит из порта, тег снимается.

Trunk Ports - тип порта, на котором может быть настроено несколько VLAN. Трафик с и из такого порта ходит с тегами настроенных VLAN. подынтерфейс.