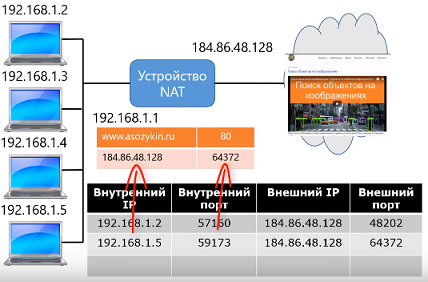
***Протоколы, интерфейсы и сервисы*** сокет - единственный интерфейс транспортного уровня в стеке протколов TCP/IP. Через этот интерфейс можно получить доступ ксервисам двух типов:надёжной и ненадёжной доставке данных.

***NAT Network Address Translation*** - трансляция сетевых адресов. Чаще всего используется, чтобы засенить айпи адрес внутренней(частной) сети в айпи адреса внешней сети (Интернет). Создано с целью преодоления нехватки адресов IPv4. Статический NAT -  отображение один к одному - нужно иметь столько же внешних адресов, сколько и компьютеров. Такое бывает редко. Чаще всего это используется для соединения сетей двух организаций, где также используются частные адреса и возможны конфликты адресов. Динамический NAT - отображение внутренних адресов на группу внешних адресов(меньшую), они используются попеременно по надобности. Чаще всего же используется вариант : отображение внутренних адресов на один внешний адрес. Преобразование реализуеься с пом. т.н. NAT таблицы, к-рая нах-тся внутри устройства NAT. Использует комбинацию IP-адрес + порт. На картинке ниже внешний порт сгенерирован случайно. Когда приходит ответ, производится поиск комбинации внешнего айпи и порта и туда шлётся инфа(перед этим у пакета происходит соответственная замена адресов.



Преимущества: преодоление нехватки IPv4, легко развернуть и использовать, структура сети скрывается от внешнего мира. Недостатки: нарушается фундаментальный принцип построения айпи сетей: каждый компьютер может соединиться с любым другим; нет возможности подключиться к компьютерам во внутренней сети из внешнего мира; некоторые прикладные протоколы работают неправлильно(FTP), т.к. были созданы давно с расчётом на возможность подсоединения к любым другим компьютерам. FTP нужно два соединения - для соединения и для передачи данных. Соединение для передачи данных обычно устанавливается с сервером. Если клиент нах-тся за устр-вом NAT. то сервер FTP не может установить с ним соединение для передачи данных; также плохо работают протоколы, не устанавливающие соединения, т.к. не могут сохранить запись в таблице NAT на всё время жизни соединения; нет единого стандарта NAT, а есть много разных вариантов, поддерживающихся разными производителями оборудования и ПО.    Решение проблем: статическое отображение(требуется несколько внешних айпишников); статическое отображение портов: порт 80 - внутренний адрес ВЕБ-сервера и порт 80. порт 25 - внутренний адреспочтового сервера и порт 25, порт 21 - внутренний адрес FTP-сервера и порт 21; Технология NAT Traversal: позволяет устанавливать соединение с компьютерами во внутренней сети, RFC и другие варианты, Используется VoIP приложениями(Skype). NAT реализуется маршрутизаторами, межсетевыми экранами и другими сетевыми устройствами.

***Межсетевые экраны; Прикладной уровень; Система доменных имён DNS; Протокол DNS; Типы записей DNS; Протокол HTTP; постояннон соединение в HTTP; Кэширование в HTTP; Электронная почта; Протокол SMTP; Протокол POP3; Протокол IMAP; Протокол FTP; Заключение.***

Чтобы перезагрузить роутер cisco: сначала exit, чтобы выйти из режима Router(config-if)#, потом exit, чтобы выйти из режима Router(config)#, потом router#reload. То же самое со свичом, только тоже в режим с хештегом войти: enable. На роутере войти в режим Router(config)# - команда config. Отключить поиск в системе DNS: (config)# no ip domain-lookup. Посмотреть инфу по сетевой карте и прочим: ipconfig, ipconfig/all . на роутере: > show interfaces g0/1; > show arp;

Ст 81 Шерстнева: Если адреса подсетей:172.16.0.0, 172.16.1.0, 172.16.2.0,172.16.3.0. маски подсетей все 255.255.255.0, то хранится ли где-нибудь маска всей сети 255.255.252.0? И как понять адреса последних ТП в подсетях: 172.16.0.140, 172.16.1.133, 172.16.2.119, 172.16.3.126? Как они получены? Там надо было создать 518 точек вроде...

***Настройка ПК в GNS3:*** ip 10.255.1.2 255.255.255.0 - назначил айпи и маску;

***Настройка ПК в Cisco Packet Tracer:*** тупо щёлкаешь по интерфейсам и прописываешь. **почему-то только Laptop1 & PC1 пингуются... Это походу потому что у них маски 255.255.0.0 да. Поэтому. Надо всё исправить с 255.255.255.0 на 255.255.0.0. В Циско очень удобно проверять всё в режиме последовательных шагов. Например выяснил наконец-то почему серевра не пинговались: на тот порт свича просто у сервера не было прописано айпишника, а роутер у себя в таблице назначения серверов не видит... И прописать их я туда не могу... Говорит, что так перенаправить нельзя, т.к. это порт этого же роутера... Назначил второму порту сервера айпишник. Пришлось там указать  10.1.200.1 255.255.255.0 иначе не позволяло из-за оверлапа. Но арп запрос всё равно был отвергнут сервером, т.к. у посылающего(10.1.1.2 255.255.0.0) другая сеть, а в разных сетях(10.1.200.0 и 10.1.0.0 - разные сети) доступ для них д.б., видимо, только через роутер.**

***Настройка роутера в Cisco Packet Tracer:*** copy run start сохраняет конфигурацию в файл;

***Настройка роутера в GNS3:*** перейти из # в (config)# : conf t; перейти в (config-if)# : int fa0/0 (походу конфигурация конкретно интрефейса 0/0); Moscow(config-if)#ip address 192.168.1.1 255.255.255.0 - назначили айпишник и маску; end - выход из конфига к #; wr mem - сохранение изменеий(айпи и маска)? ; show run - покажет сохранённые айпишники, маски и т.п.; список команд на хосте : ?;  ip ? - покажет команды для айпи; ip 192.168.1.2/24 192.168.1.1 - настройка айпи хоста и роутера(gateway) одновременно? да. просто прописали гейтвей для этого хоста; arp - показывает арп-таблицу на хосте и сколько ей осталось жить(обычно 110 с лишним секунд); на роутере: ***ip dhcp ?*** - покажет команды для dhcp; вообще для начала динамической настройки на роутере установили статический айпишник... ip dhcp pool DHCP - перешли в режим R1(dhcp-config)#(из R1(config)); network 192.168.2.0 255.255.255.0 раздаём сетку 192.168.2.0 255.255.255.0; default-router 192.168.2.1 - дефолтный маршрут; ip dhcp - на хосте по идее должно получить автоматичесий айпишник, но у меня не находит dhcp server; подключил другой комп, на нём сработало, эффект: DDORA IP 192.168.2.2/24 GW 192.168.2.1; после этого можно пропинговать первый комп: PC3> ping 192.168.1.2

192.168.1.2 icmp\_seq=1 timeout

192.168.1.2 icmp\_seq=2 timeout

84 bytes from 192.168.1.2 icmp\_seq=3 ttl=63 time=21.195 ms

84 bytes from 192.168.1.2 icmp\_seq=4 ttl=63 time=15.082 ms

84 bytes from 192.168.1.2 icmp\_seq=5 ttl=63 time=17.157 ms

trace 192.168.1.2 - трейс до первого компьютера. Но автор говорит - "отлично", хотя в результате выдано:

trace to 192.168.1.2, 8 hops max, press Ctrl+C to stop

1   192.168.2.1   10.137 ms  9.237 ms  9.050 ms

2   \*192.168.1.2   19.196 ms (ICMP type:3, code:3, Destination port unreachable)

save - сохранить конфигурацию на ПК; copy running-config startup-config - сохранение конфигурации в роутере(вроде как то же, что wr mem); сохраняются проекты пока плохо, поэтому отключаю nvidia experience, dragon хрень и т.п; Если комп загружается быстрее роутера, то по DHCP он не успевает получить айпишник, поэтому надо ещё раз вводить ip dhcp;

В GNS3 отсутствует возможность эмулировать коммутаторы... но есть встроенный в gns коммутатор(чисто программный, без командной строки циско, коммутатор 2-го уровня без настроек). Во многих случаях его фуекционала достаточно. А также есть другие пути: рассмотрим позже.

***Настройка VLAN:*** правой мышью на свич, configure, double click on needed port, VLAN numericUp to choose the needed VLAN number, add, Apply, Ok. Ping the nodes to check if they're not reacheable from different virtual nets. These switches don't support STP! You should avoid loops. Чтобы проложить путь из роутера к роутеру в другой сети нужно прописать путь в таблице маршрутизации: ip route 10.255.7.0(~~удалённый роутер~~ удалённая сеть) 255.255.255.0 10.255.6.1(соседний роутер, через который нужно идти) всё это в режиме conf t. Здесь маска применяется к первому адресу(назначения). Причём получается первым адресом указывать только всю сеть, иначе показывает ошибку "Inconsistent address mask" or  route... Похоже надо маску подбирать... Или просто в каждом роутере записи делать только для сети. Сначала не пинговалось, но это похоже потому, что надо было и в обратныую сторону(с роутера назначения) указать путь, чтобы подтверждение пинга могло вернуться. **show ip route** - показать таблицу маршрутизации роутера; Чтобы начать отслеживать трафик wireshark-ом нужно проавой кнопкой кликнуть по интерфейсу на топологии и выбрать "Start capture" . Как получить доступ к консоли конфигурирования маршрутизатора CISCO (продемонстрируйте). В GNS для начала маршрутизатор нужно загрузить: edit -> prefrences -> dynamips ios routers -> new -> выбираем файл образа устройства. Например c3725-adventerprisek9-mz.124-15.T14.bin, устанавливаем его. Ставим галочку, разрешающую связь с сетевой картой устройства: Virtual box template -> edit -> network -> Allow GNS to use any configured VirtualBox adapter. Перетаскиваем устройство в проект, включаем, правой кнопкой мыши – Console. WinServer2016 password: winServ1; Вместо Ctrl + Alt + Del в виртуал боксе надо жать правый Ctrl + Del;

R1# configure terminal

R1(config)# interface FastEthernet 0/0

R1(config-if)# ip address dhcp

R1(config-if)# no shutdown

R1(config-if)# end

R1#

Чтобы  настроить  **PPP - CHAP** нужно установить:  ip route 172.16.0.0(сеть назначения) 255.255.0.0 s2/0(порт на текущем роутере, через который есть доступ к промежуточной сети). Потом: username rdop(консольное имя роутера, с которым связываемся) password cisco;  int ser0/3/0; encapsulation ppp; ppp authentication chap; ip addr 1.1.1.1 255.255.255.0(опционально); no shutdown(опционально); и так на обоих роутерах. Важно, что имя брать из консольки.

Маршрутизатор воткнутый в роутер во второй лабе никак не контактирует с этим роутером получается... просто обеспечивает связь между теми, кто подключается к этим портам.

***Windows Server 2016:*** route print - просмотр таблицы маршрутизации; ipconfig - попроще; route /p(делает изменение persistent, т.е. постоянным,оно сохранится даже после перезагрузки сервера) 10.255.2.0(сеть, доступ к которой хочу открыть) mask 255.255.255.0 10.255.1.1(gateway, router, etc); WinServer2016 password: winServ1;

***Ubuntu 18 IP config*** ip addr - показать айпишник и интерфейсы; nmcli connection add con-name stepanInt(имя соединения) type ethernet ifname enp0s3(имя LAN карты(номер два в списке после ввода route print)) ipv4.addresses 10.255.1.2/24 ipv4.method manual connection.autoconnect yes    (  nmcli connection add con-name stepanInt(имя соединения) type ethernet ifname enp0s3(имя LAN карты(номер два в списке после ввода route print)) ipv4.addresses 10.255.1.2/24 ipv4.method manual connection.autoconnect yes  );

***Делаю конфигурацию "роутер на палочке"(это как коммутатор L3 получается распознает айпишники)***. Распределил сеть из 4-х компов по 2-м VLAN. К пятому порту коммутатора подключил роутер. Этот порт должен работать в режиме TRUNK. Конфигурирую свич: два раза по пятому порту, чтобы напротив Port появилось 5;  напротив Type выставляю dot1q (trunk); Номер ВЛАН-а не трогаю, оставляю по умолчанию 1; apply, Ok; стартую роутер; поднимаю физ. интерфейс fa0/0 роутера: conf t, int fa0/0, no shutdown, exit; int fa0/0.2(создал саб-интерфейс для второго ВЛАН-а)(выглядит: R1(config-subif)#); encapsulation dot1Q 2 - указал, что это второй влан; ip address 192.168.2.1 255.255.255.0 - повесил айпи адрес; no shutdown; exit. Т.о. на роутер вешаю айпи-адрес, являющийся **шлюзом** для наших компьютеров(в подсети 2); конфигугрирую для ВЛАН 3: fa0/0.3; encapsulation dot1Q 3; ip address 192.168.3.1 255.255.255.0; no shutdown; end; wr mem; ещё раз выставляю dot1q для пятого порта(надо так, чтобы вместо access было dot1q(в таблице); проверяю шлюз по умолчанию ПК1: 192.168.2.1; пингуется; Теперь пингуем соседнюю виртуальную подсеть: ping 192.168.3.2  и она пингуется, т.к. мы эти сети объединили роутером. Все свичи должны быть соединены транками.

***2-й способ использования коммутатора в GNS3(некое подобие cisco 3560 или 3750)*** схема роутер + EtherSwitch module = L3 Switch. Т.о. получаем 16 доп. портов, к-рые можем настраивать как порты L2 или L3 коммутатора. Всё просто. Плюсы метода: хороший L2 функционал, наличие L3 функционала, малое потребеление системных ресурсов. Минусы: не сохраняется конфигурация, не совместим с встроенным свичем GNS3, нестабильность. Не факт, что мне это надо. Пока пауза.

~~поставил локалхост 5190... много чего ещё вероятно произошло... но теперь открываеи сохранённые проекты.~~

***Технология открытия файлов проектов в GNS3; подключения виртуал бокса к сети:*** после запуска дождатьсяпоявления окна создания нового проекта, там выбрать recent и нужный проект. Загрузится. Чтобы соединить пустой контейнер со свичем, надо правой кнопкой на контейнере, configure, network, allow GNS to use any configured VirtualBox adapter.

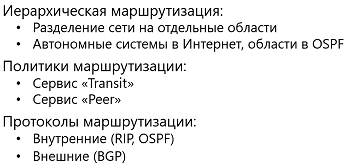
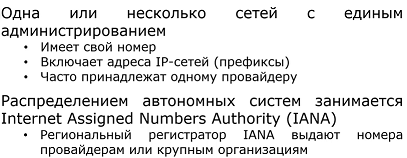
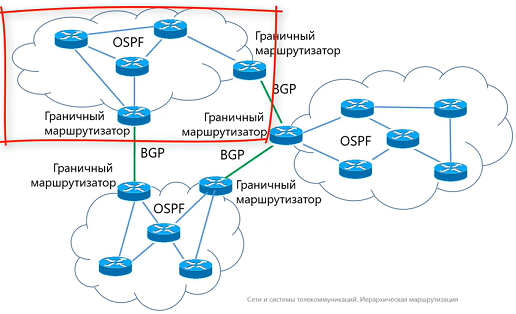
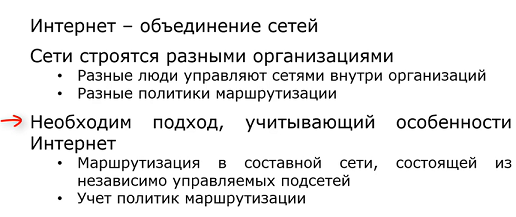
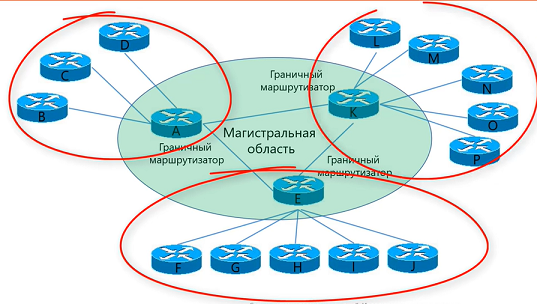
***Коммутаторы третьего уровня модели OSI(L3):*** Ip-маршрутизация; агрегирование коммутаторов уровня доступа; используются в качестве коммутаторов уровня распределения; высокая производительность, низкая стоимость по сравнению с маршрутизаторами. Ни в коем случае не заменяют маршрутизаторы. Работают для маршрутизации трафика внутри сети, но не наружу.

***Virtual Box:*** Ctrl(right) + F - вход/выход из полноэкранного режима с возвращением кнопок меню;

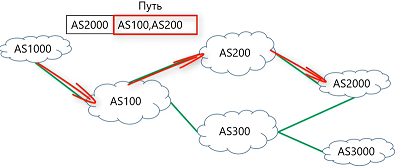
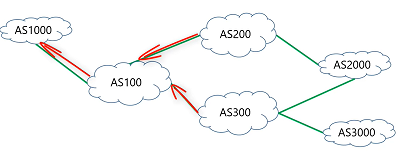
***Протокол RIP:*** (Routing Information Protocol) - самый первый проткол маршрутизации в сетях айпи(1982).  Дистанционно-векторный проткол(в противоположность OSPF например, который сначала изучает всю сеть и только после этого выполняет расчёт кратчайших путей) . Здесь для метрики измеряется только число промежуточных маршрутизаторов. Широковещательные сообщения о конфигурации вектора расстояний. Протколы маршрутизации используются для того, чтобы маршрутизаторы могли в автоматическом режиме, без участия адинистраторов сети составить карту сети и таблицы маршрутизации. Используется алгоритм **Форда-Беллмана**(простой перебор вершин итеративно до тех пор, пока расстояния не перестанут уменьшаться или в худшем случае V-1 раз). Передача данных через протокол UDP, порт 520. В основе реализации дистанционно-векторных протколов лежит вектор расстояния(distance vector). Этот одномерный массив содержит адреса сети и расстояния до них. Метрика очень простая - число промежуточных маршрутизаторов. В протколе RIP есть ограничение на максимальную длину - 16. Если длина маршрута равна 16, то это эквивалентно бесконечному расстоянию в RIP(такая сеть считается недостижимой). Это сделано для предотвращения образования петель в крупных сетях. Резервные маршруты не добавляются в таблицу. Обмен сообщениями с векторами расстояний каждые 30 секунд. Если от маршрутизатора нет сообщений 180 секунд, то он считается отказавшим.  Возможна ситуация бесконечного счёта, когда связь до одной сети теряется и соседний маршрутизатор это учитывает, но от ещё одного соседнего, которому он обеспечивал связь с этой утерянной сетью поступает сообщение, что он всё еще имеет эту связь(равную 1). Наш герой увеличивает это значение на 1  (до 2) и прописывает к себе в таблицу этот маршрут. В итоге маршрутизаторы шлют друг другу эту инфу и постоянно инкрементируют расстояние до утерянной сети. В итоге счёт дойдёт до 16 и сеть станет считаться недоступной. Но до этого все пакеты, направлявшиеся в утерянную сеть также будут гулять между нашими двумя маршрутизаторами по эдакой петле. Для избежания этого в IP уже реализовано TTL. А в RIP используется: 1) расщепление горизонта(Split Horizon): не отправлять информацию о сети на тот интерфейс, через который эта информация получена. 2) Отравление маршрута(Route Poisoning): При обнаружении проблемы с маршрутом отправка сообщения о его недоступности(расстояние равно 16). 3) Holddown: таймер на изменение информации о недоступном маршруте(будет некоторое время игнорировать сообщения от соседей, что до пропавшей сети всё ещё есть связь). На практике этот протокол почти не используется. Его аналог - EIGRP от Cisco.



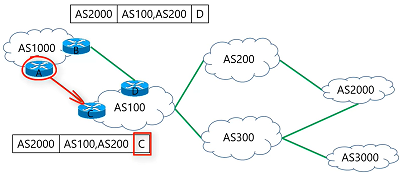
***Иерархическая маршрутизация*** Ни OSPF, ни RIP не подходят для крупных сетей и Интернета из-за ограничений масштабируемости(весь трафик будет уходить на перестройку огромных таблиц маршрутизации, не хватит мощностей для расчета OSPF). Решение в иерархической маршрутизации: Разделение сети на отдельные области; внутри областей и между ними протоколы маршрутизации работают по-разному.Примеры: 1) Иерархическая маршрутизация а OSPF: Делим сеть на три области, объединённые магистральной областью. Расчёт Дейкстрой происходит для каждой области отдельно. . 2) мрашрутизация в Интернет. Для внешней маршрутизации используется протокол BGP



***Протокол BGP***  (Border Gateway Protocol - протокол граничного маршрутизатора(шлюза)) BGPv4 - де-факто стандарт протокола внешней маршрутизации. Версия BGPv4 описана в документе RFC 4271, 2006. Дистанционно-векторный алгоритм(вектор пути (в противопоставление вектору расстояний в RIP)). Для передачи данных протокол BGP использует проткол TCP, порт 179. Вычисление маршрутов на уровне автономных систем. Одна или несколько сетей с единым администрированием. Автономные системы:

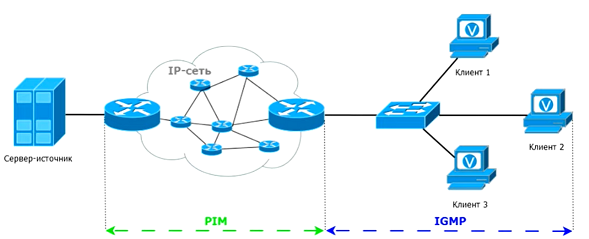
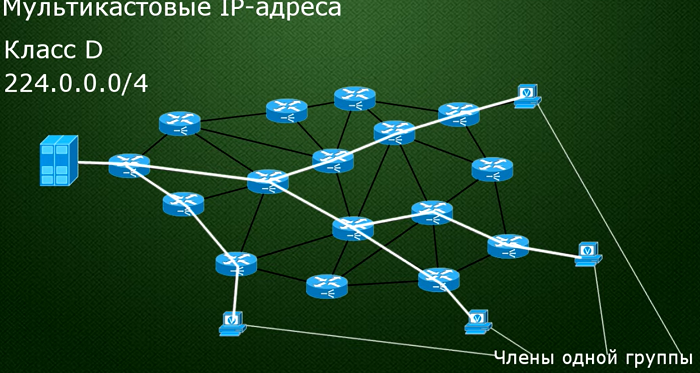
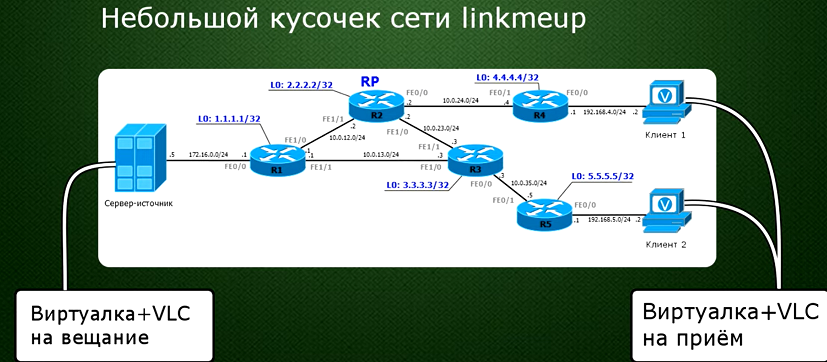


в них таблицы строятся точно так же, как в RIP. Только уже по целым автономным системам. Отличие в том, что в информации о маршруте передаётся 3 части: 1) автономная система - одна или несколько айпи сетей с масками(иногда их называют префиксами). 2) вектор пути - список промежуточных автономных ситем. Петель не возникает, т.к. если в маршруте сеть обнаружит саму себя в списке,тто поймёт, что этот маршрут использовать не нужно. 3) следующий маршрутизатор. Обычно автономная система - это напрмиер интернет -провайдер.  В системе для скорости может быть несколько маршрутизаторов и соответственно несколько путей. При этом проткол BGP никак не указывает как рассчитывать куда слать пакеты на A или на B. Указываются только внешние маршруты(C & D).



Автономные системы могут предоставлять сервис Transit, если они передают данные через себя в другие сети. Если же не передают, то они называются тупиковыми. Здесь это AS1000, AS3000 и Созыкин говорит, что AS2000 то же... хотя там есть связи... Дело в том, что организация могла придерживаться такой политики и сделать сеть тупиковой. Наша так и делает и сообщает AS200 & AS300, что здесь путь только к AS2000.

***Протокол IGMP:*** (Internet Group Management Protocol) -



***Протокол OSPF:*** (Open Shortest Path First) - современный протокол маршрутизации, который чаще всего используется на практике в сетях IP. Для расчета используется алгоритм Дейкстры. Протокол с учётом состояния канала(link-state protocol). Передача данных через IP, код протокола 89. Особенности: децентрализованный глобальный алгоритм; расчёт стоимости путей после получения полной информации о сети. Сейчас используется версия OSPFv2 - 1998 года, RFC 2328; OSPFv3 - 2008 года, RFC 5340 - для работы с сетями IPv6. Стоимость маршрута зависит не только от числа маршрутизаторов, но и от их скорости и от любых других параметров, которые посчитает нужным учесть сисадмин.

Этап работы:

1 изучение топоогии сети: маршрутизаторы изучают подключённые сети и ближайших соседей; информация о топологии распространяется по всей сети с помощью лавинной рассылки(flooding).

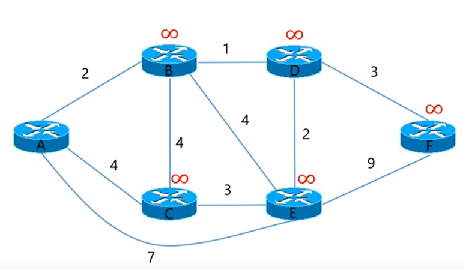
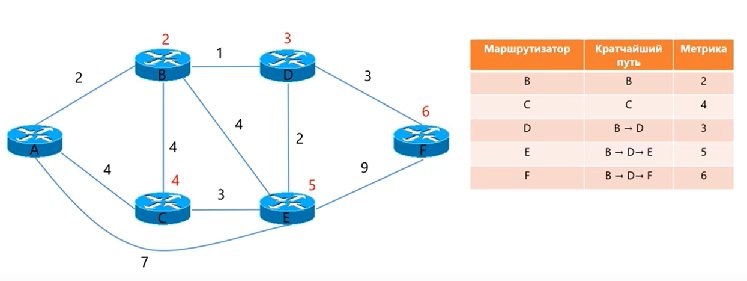
2 (на этом этампе все маршрутизаторы имеют одинаковую информацию о конфигурации сети) Расчёт стоимости маршрутов в сети: выполняется после того, как будет известан полная конфигурация сети; каждый маршрутизатор выполняет расчёт самостоятельно. (**Дейкстра**)

3 Обновление информации о конфигурации сети: маршрутизаторы проверяют доступность соседей; Рассылка информации об изменении конфигурации сети. (если сеть поменялась со временем)

Метрика маршрута в OSPF: каждое соединение между маршрутизаторами имеет стоимость; стоимость задаётся администратором сети; предпочтительны маршрути с низкой стоимостью.

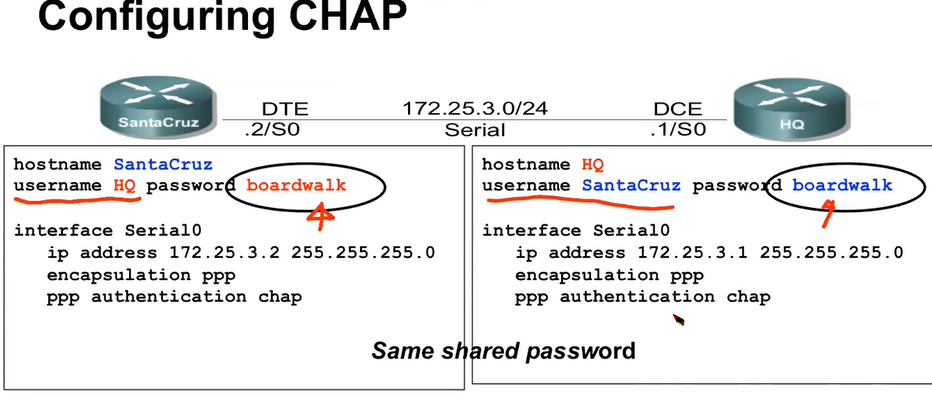
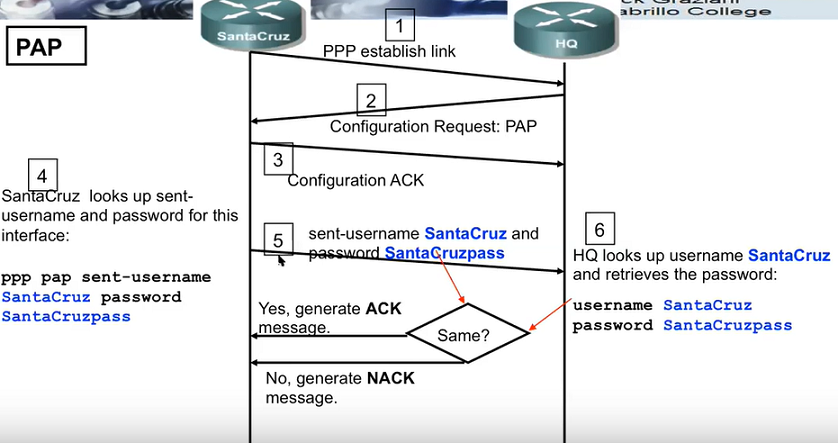
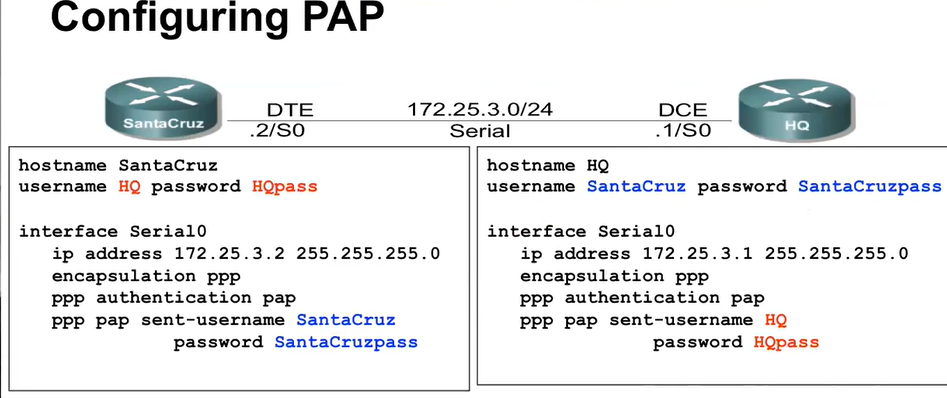
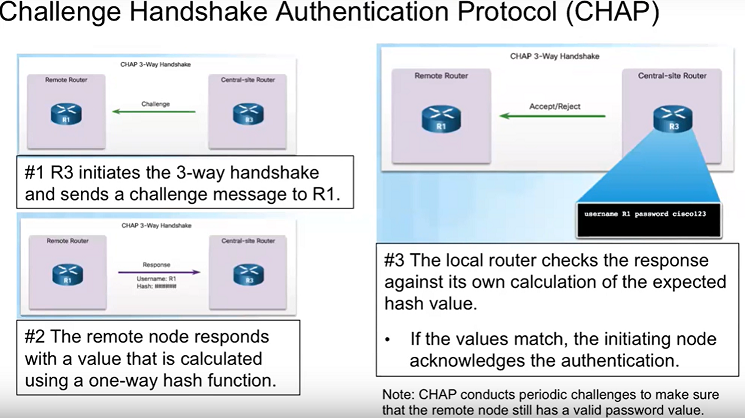
Факторы, влияющие на стоимость: пропускная способность канала; Загрузка канала(на практике не делают); Особенноти архитектуры сети и предпочтительные пути для трафика. В общем продолжение здесь:   <https://www.youtube.com/watch?v=rDPF5LZiyOw> Маршрутизатор рассылает своим соседям: Объявление о состоянии канала(link state advertisement): все каналы маршрутизатора, состояние каналов, доступные сети. На основе этих объявлений маршрутизаторы ведут базу данных состояния каналов(link state database). Такая база ведётся каждым маршрутизатором независимо от других. В идеале все эти базы данных д.б. синхронизированы между собой. псоле такой синхронизации нужно сделать расчет стоимости путей. Но на практике это сложно. Поэтому синхронизируются только соседние.

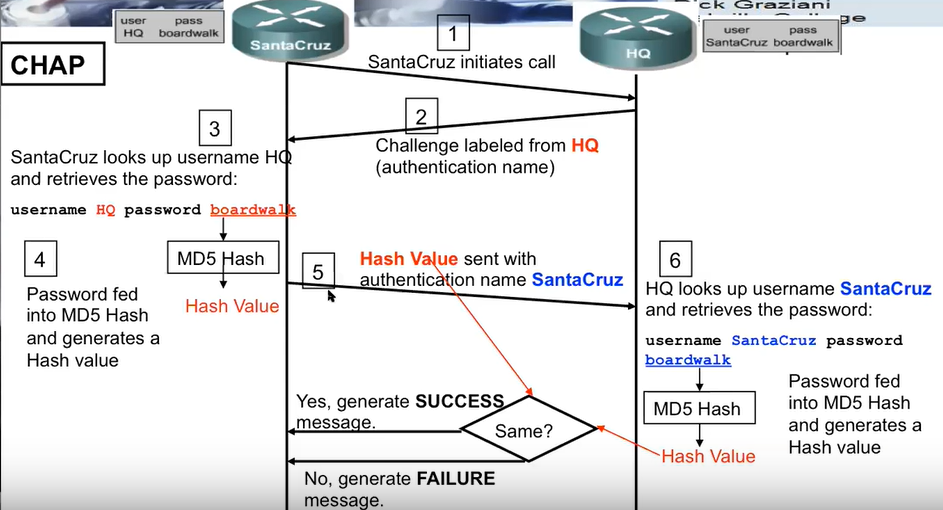
Дейкстрой каждый маршрутизатор(здесь А) считает кратчайшие расстояния до остальных маршрутизаторов, основываясь на стоимости, заданной админом(?). На основе этих данных маршрутизатор строит таблицу:

->

Кстати здесь указан полный путь стрелочками в таблице. Но для таблицы маршрутизауции это, конечно, излишняя информация. Каждые 10 секунд маршрутизаторы шлют своим соседям сообщение "Hello", чтобы проверить их доступность и работоспособность(благодаря этому получаем "быструю сходимость" - быстрый отклик на изменения в конфигурации сети). Потом 40 секунд ждут ответа, тоже "Hello". Если ответ не пришёл, значит произошло изменение конфигурации и соседний маршрутизатор больше не доступен(проблема либо с каналом связи, либо с соседним маршрутизатором). Например вышел из строя маршрутизатор D. Его соседи(B, E, F) не дождались ответа. Лавинной рассылкой эти соседи рассказывают всем остальным. После этого происходит перерасчёт стоимости маршрутов в сети, всё по новой короче. Петли здесь не возникают, т.к. строятся однозначные наиболее быстрые маршруты. В протокле RIP маршрутизаторы используют расстояние, рассчитанное соседними маршрутизаторами, что приводило к проблемам. По сравнению с RIP - более сложный протокол из-за высоких требований к памяти и вычислительной мощности маршрутизаторов.

***PPP(Point to Point Protocol) authentication: CHAP(recommended) & PAP:*** РРР создан на основе проткола HDLC(Higher Level Data Link Control)для организации автоматического согласования параметров канала, авторизации доступа к каналу передачи данных и мультиплексирования  данных от нескольких протоколов сетевого уровня.  В PAP двустороннее рукопожатие: 1) один шлёт имя пользователя и пароль в виде plaintext, 2) второй подтверждает/отвергает соединение.   В CHAP трёхстороннее рукопожатие: 1) центральный роутер запрашивает, 2) удалённый отвечает username-ом & ###(хешем) пароля, 3) центральный сравнивает username и ### со своими и если совпадают, то подтверждает аутентификацию.





***Протокол PIM-SM:***

***IpTV:***

***Способы расчёта сети:***