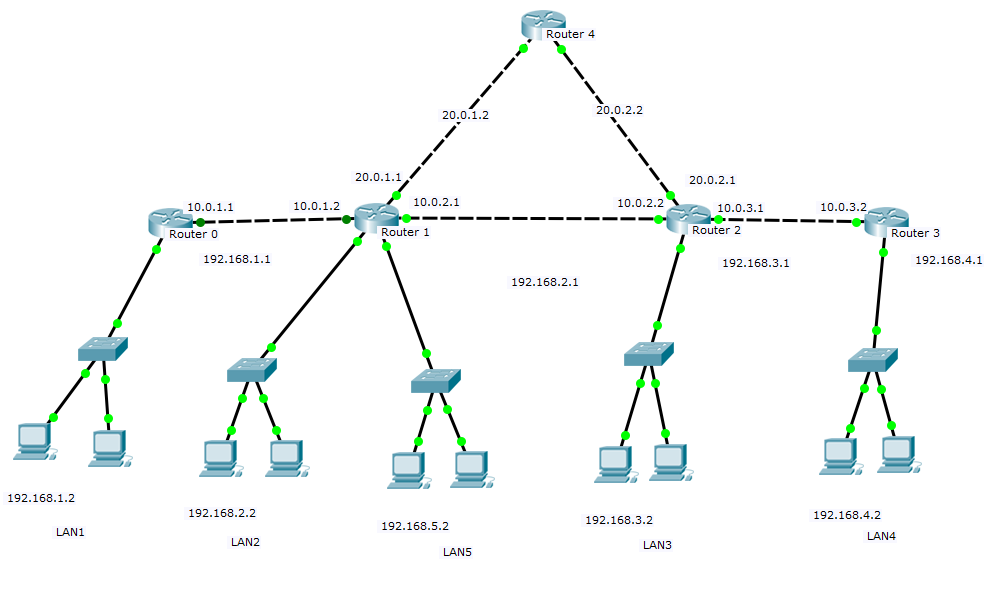
# 

# Лабораторная работа №9. Сообщения Query и Reply. Суммирование. NBMA сети. Аутентификация по ключевым цепочкам

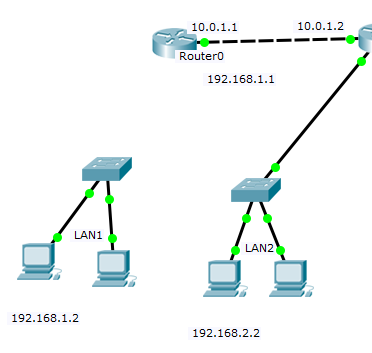
За основу возьмём нашу сеть с прошлой лабораторной работы.



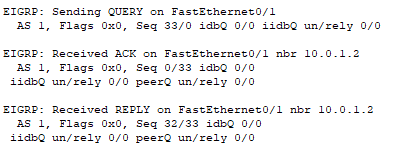
1. Если роутер теряет соединение с какой-либо сетью, он начинает рассылать **Query** пакеты на соседние Роутеры, те спрашивают остальные Роутеры, которых он знает. Если какой-то Роутер знает, как добраться до этой сети, то он отправляет **Reply** пакет. В итоге перестраивается таблица маршрутизации.
2. Также существует такое состояние как **Stuck-in-Active**, когда **Reply** пакеты по какой-то причине теряются, в итоге роутер не сможет добраться до нужной сети, хотя маршрут есть, но Reply ответ потерялся. Для того чтобы такого не происходило, есть еще одна группа пакетов **SIA-Query** и **SIA-Reply**. Наш Роутер ждёт 2мин. 55сек. Reply ответ, если он его не получает, он отправляет **SIA-Query** пакет соседнему роутеру и спрашивает у него “WTF!? где ответ”, а тот ему ответит, “я тут не причём, это вот тот роутер дальше не ответил”, и отправляет **SIA-Query** роутеру который дальше “WTF!? где ответ, тебя там роутер спрашивает за мной”, а тот ему отвечает **SIA-Reply** и говорит,“да я всё отправлял вот еще раз отправляю” в итоге он доходит до нашего роутера. Если даже после **SIA-Query** ответа не следует, то Роутер понимает, что до потерянной сети уже точно никак не добраться и выкидывает её из таблицы маршрутизации.
3. Давайте посмотрим на эти пакеты с помощью команды # **debug eigrp packe** на Роутере0, пока мы ничего не видим т.к. сеть не теряется



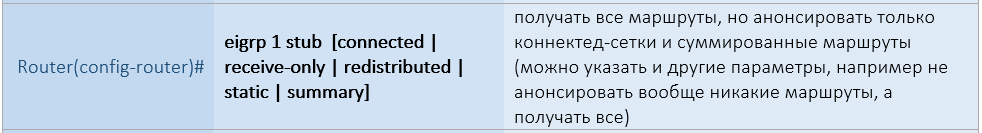
1. Давайте отключим нашу LAN 1 и посмотрим что будет.



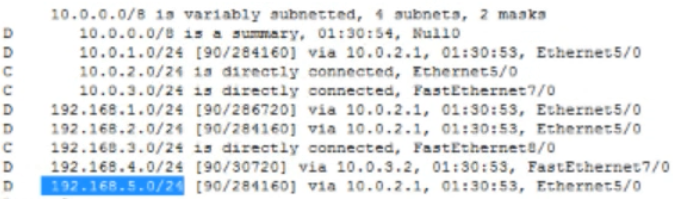
1. В итоге мы увидим в дебаге **QUERY** и ответ **REPLY** что добраться до этой сети нельзя через другие роутеры.



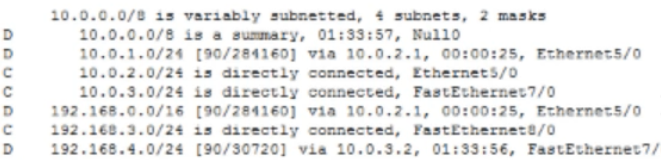
1. Есть такое понятие как **Stub Router**, если мы конфигурируем роутер в данной концепции, то роутер не будет спрашивать у других при потере сети и другие у него тоже не будут спрашивать. Это может понадобится, если у нас есть канал до какой-то сети, но мы не хотим его предоставлять другим Роутерам, другим сетям. Тогда мы можем сконфигурировать роутер как **Stub.** В CPT мы этого сделать не сможем, но на реальном оборудовании это задается командой # **egrp 1 stub**. Еще один пример, у нас есть 2 дата центра, к которым мы подключены, если прямая связь между датацентрами пропадёт, то весь трафик может пойти через наш офис, что скорее всего положит всю нашу сеть. Что не есть хорошо, поэтому **Stub Routing** это важно.
2. У **stub** есть несколько параметров, **connected** – значит, что Роутер будет рассказывать только о тех сетях, которые подключены к нему, о сетях, о которых он узнал от других роутеров, он не расскажет. **receive-only** - запретить рассказывать о любых сетях, будет только получать эти сети. **redistributed** - разрешается рассказывать о маршрутах полученных по EIGRP. **static** - разрешено рассказывать о маршрутах добавленных вручную. **summary** - посмотрим далее. Здесь мы говорим только о **QUERY** и **SIA-Query** запросах, в обычных **Update** пакетах на Роутер будет всё рассказывать.



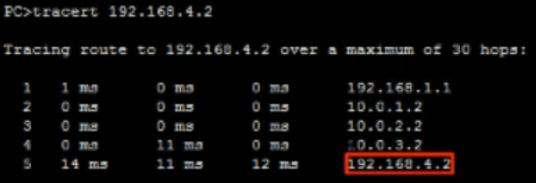
1. Давайте теперь поговорим о **суммаризации маршрутов**, давайте посмотрим какие маршруты есть на Роутер2, наберём **# show ip route** мы увидим маршруты до всех сетей. Давайте попробуем объединить LAN 1, 2 и 3 (192.168.1.0, 192.168.2.0, 192.168.3.0) в одну большую сеть, чтобы Роутер2 получал один маршрут до них, а не 3.



1. Для того, чтобы это сделать, перейдем на Роутер1 и перейдем на интерфейс который соединяется с Роутер3. Введем команду (config-if)# **ip summary-address eigrp 1 192.168.0.0 255.255.0.0** мы указываем сеть которую хотим просуммировать, такая маска задана не случайно. Суммаризация выполнена.
2. Давайте снова проверим таблицу маршрутизации на Роутер3, мы увидим такой результат, пропали сети 192.168.1.0, 192.168.2.0, 192.168.3.0 пропали и появилась сеть 192.168.0.0**/16**.

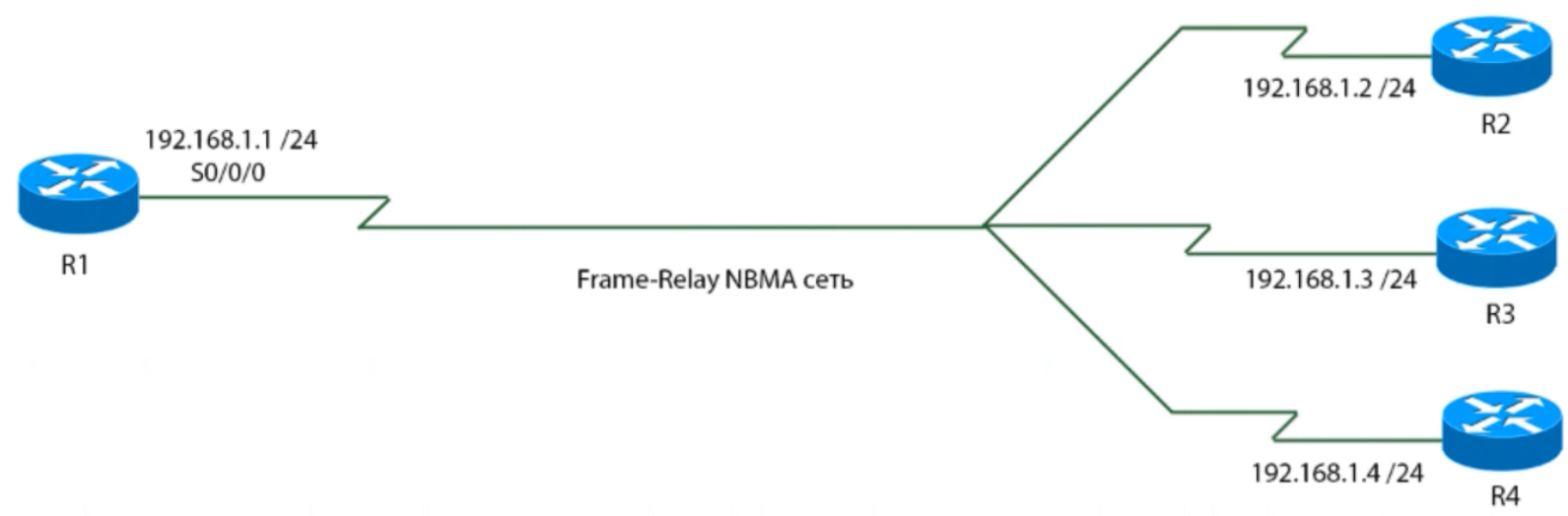


1. Теперь роутер не видит наши сети, но пакеты будут доходить, т.к. мы создали одну большую сеть с новой маской. Проверьте, работает или нет сеть, отправьте обычный пинг.
2. Зачем это нужно? Представьте, что у нас не 3, а 255 подсетей, согласитесь, что 1 запись в таблице маршрутизации лучше, чем 255. Поэтому суммаризацию нужно применять там, где это возможно. При этом есть важная деталь, у нас есть просуммированный 192.168.0.0**/16** и два конкретных 192.168.3.0/24, 192.168.4.0/24. Причём суммированный по маске включает наши конкретные, но Роутер умный и он отдаёт приоритет сначала конкретным адресам, а потом уже просуммированным, так что пакеты не теряются.
3. Давайте создадим петлю маршрутизации, в которую может попасть пакет при неправильной настройке. Давайте пропишем статический маршрут на Роутер0 (config)# **ip route 192.168.8.0 255.255.255.0 10.0.1.2** - мы говорим что до сети 192.168.8.0 можно добраться через 10.0.1.2 (это Роутер1) - но такой сети конечно же нет.
4. А Роутер1 зададим (config)# **ip route 192.168.8.0 255.255.255.0 10.0.1.1** . Теперь он думает, что до 192.168.8.0 можно добраться через 10.0.1.1 (это Роутер0) в итоге роутеры будут гонять пакеты между собой, это называется петля маршрутизации.
5. Отправим ping 192.168.8.15 с LAN 1 и посмотрим в режиме симуляции как будет гулять ICMP пакет между роутерами. Но мы так же помним, что есть параметр TTL (время жизни пакета) который равен 128, каждый раз, когда пакет проходит через роутер, происходит -1 от TTL. В итоге пакет будет гулять по роутерам, пока LLT не будет = 0 и пакет будет удален.
6. Давайте посмотрим, как работает команда **traceroute**. Введем команду **tracert 192.168.4.2** на ПК0 в Command Prompt. В итоге мы увидим адреса всех роутеров, которые встретились на пути к **192.168.4.2.**



1. Что происходит внутри? Происходит тройная отправка ICMP пакета, с TTL=1, следовательно первый Роутер который встретится на пути, уничтожит ICMP пакет и расскажет нам об этом в ответе. Почему отправляется 3 ICMP? Чтобы определить, есть или нет другие Роутеры (например когда включена балансировка, то трафик может отправляться по разным маршрутам, тогда и ответ будет от других Роутеров). Далее отправляется опять 3 ICMP пакета с TTL=2, и данный пакет дойдет до следующего роутера и так далее увеличивается TTL, пока не дойдёт до заданного IP-адреса.
2. Такую же команду вы можете отправить со своего ПК через командную строку, до любого сайта, указав его доменное имя.

## Nonbroadcast Multiple-Access Networks



1. Мы помним что Ethernet это протокол работающий на 2 уровне модели OSI, т.е. на Канальном уровне. Мы знаем что там есть адресация по MAC. На самом деле Ethernet не единственный протокол канального уровня. Давайте рассмотрим протокол канального уровня [Frame-Relay](https://ru.wikipedia.org/wiki/Frame_relay). Данный протокол работает с по Serial Link, их разъём выглядит так:
2. Здесь адресация происходит не по MAC адресам. Какое же различие между Ethernet и Frame-Relay? Frame-Relay это протокол который позволяет построить [NBMA](https://en.wikipedia.org/wiki/Non-broadcast_multiple-access_network) сеть, в этой сети нет broadcast (отправка всем в сети) и multicast (отправка группе в сети). И как следствие в таких сетях не работает система автоматического обнаружения соседей. Т.е. нам нужно и анонсировать сети и вручную прописать соседа с которым мы хотим подружится, т.е. его IP адрес и его интерфейс, также нужно указать его канальный адрес.
3. У этой сети есть куча особенностей, например, наш реальный интерфейс нужно разбивать на виртуальные, чтобы работать с несколькими интерфейсами. Конечно же Frame-Relay уже устарел, и его встретить довольно сложно, но по такой же логике работают некоторые существующие сети.

## Аутентификация по ключевым цепочкам

1. Она нужна для безопасности, сейчас два роутера когда видят друг друга, просто начинают обмениваться маршрутами. Но может случиться так, что к нашей сети подключится злоумышленник со своим роутером, а у нас не сконфигурирован passive-interface и он сможет вбросить в таблицу маршрутизации левые маршруты. Например, он может сказать всем роутерам, что выход в Интернет происходит через него и пустить весь трафик через себя. Для того чтобы этого не случилось можно настроить аутентификацию по ключевым цепочкам.
2. Давайте заглянем в наш [ман](https://drive.google.com/drive/u/0/folders/1CAmrI3KvKim5xwdLWiSv9nMQLmno4S2s) и посмотрим группу команд “настройка аутентификации по ключевым цепочкам”.



1. Первое, что нам нужно сделать, это синхронизировать время на наших роутерах, давайте зайдем на Роутер0 и Роутер1 и посмотрим какое время выставлено. Для этого наберем команду # **show clock**



1. Мы видим что время немного неточное, давайте его настроим, введем команду # **clock set 15:00:00 29 oct 2018** (время выберите своё). Введите данную команду на Роутер0 и Роутер1.
2. Теперь давайте зайдем в режим конфигурирования и введем команду (config)# **key chain SSUCHAIN** мы попадаем в режим конфигурирования
3. Самостоятельно сконфигурируйте аутентификацию по ключевым цепочкам, используя [ман](https://drive.google.com/drive/u/0/folders/1CAmrI3KvKim5xwdLWiSv9nMQLmno4S2s)