****

**­**

**Министр науки и высшего образования Российской̆**

**Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

Факультет информационных технологий и программирования

Рабочий протокол и отчет по лабораторной работе №5

Маршрутизация в IP сетях

**Выполнил(а) студент группы** № **M33091**

Сидорцов Владимир  
Цыденов Алексей  
Мирзабеков Ренат  
Максимов Лев

**Подпись:**

**Проверил:**

Санкт-Петербург

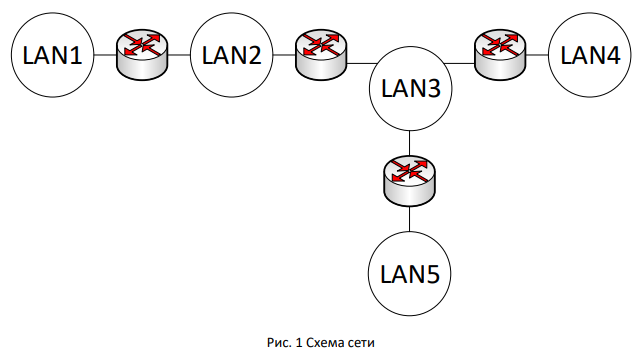
2023г.

**Цель работы.**

Получить представление о работе IP маршрутизатора; получить опыт в составлении таблиц

маршрутизации и работе протоколов внутренней и внешней маршрутизации.

**Описание работы**

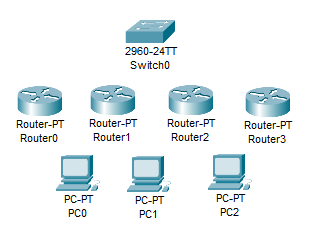


На приведенной схеме представлена составная локальная сеть. Отдельные локальные сети соединены маршрутизаторами. В сетях 1, 4 и 5 находится по одному компьютеру.

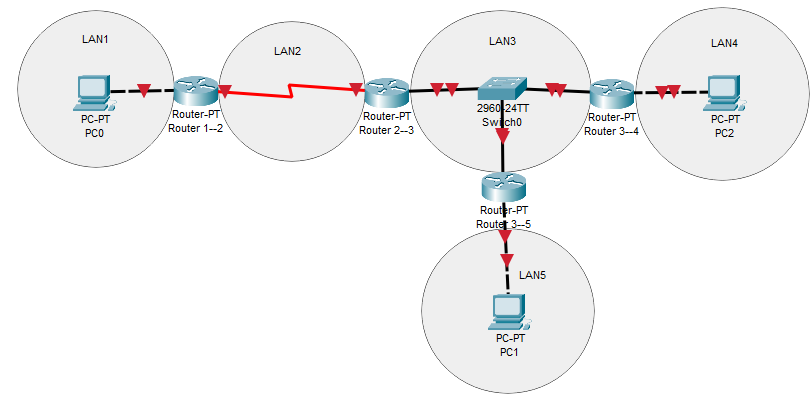
Необходимо реализовать схему, приведенную на рисунке 1, смоделировав ее в программе Packet Tracer, обратив внимание, что сеть №3 — это одна локальная сеть, в которой маршрутизаторы соединяются через коммутатор.

Решение

Создадим начальное оборудование:



И построим сеть, указав логические сети:



configure terminal

interface FastEthernet0/0

ip address 192.168.1.1 255.255.255.0

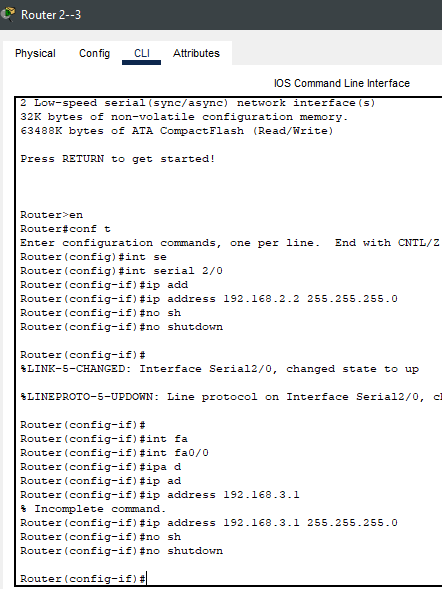
no shutdown

interface Serial2/0

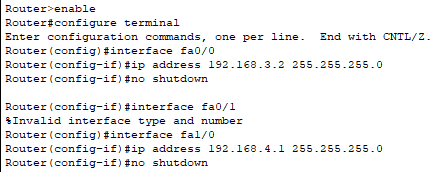
ip address 192.168.2.1 255.255.255.0

no shutdown

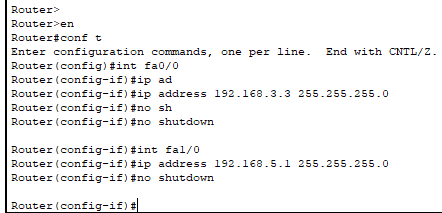
Настроим маршрутизатор меж 2 и 3 сетями:



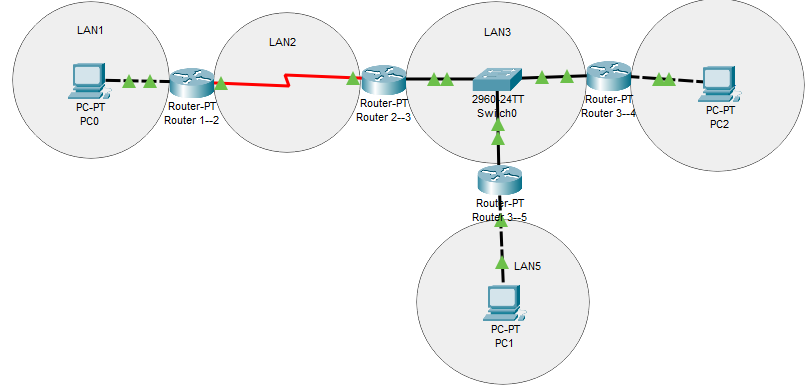
Роутер между 3 и 4 сетью (красиво, чтобы был понятен порядок команд):



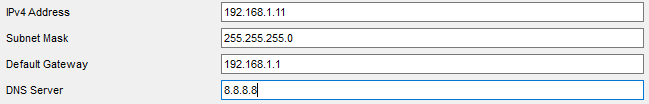
И между 4 и 5 сетью:

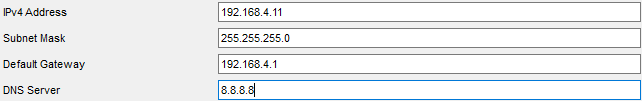


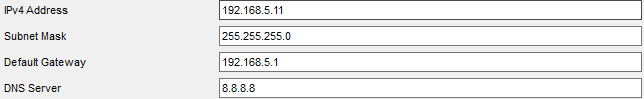
Видим, что теперь все соединения настроены и готовы:



Чтобы протестировать их, назначим статические адреса компьютерам:

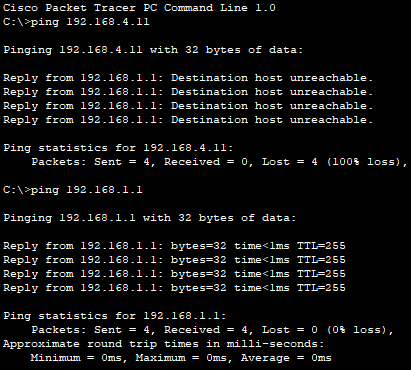






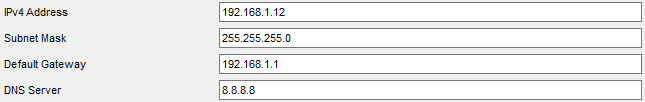
Адреса компьютеров специально выбрали с 11, чтобы по разряду можно было не путаться компьютер ли это или техническая инфраструктура: 1-10 – это роутеры, 11+ - номер компьютера.

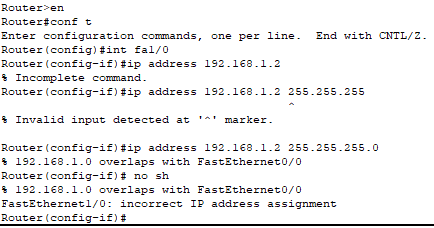
Попытаемся соединиться из 1 сети в 4:



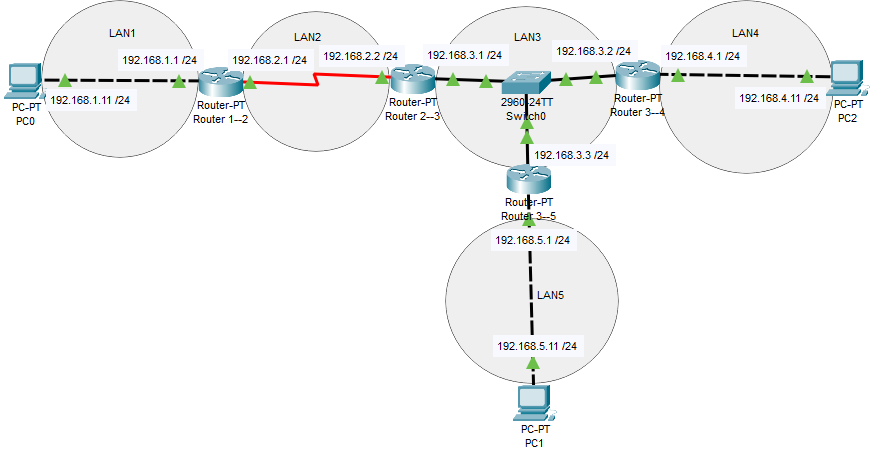
Видим, что мы не можем получить доступ к компьютеру в 4 сети. Действительно, наш маршрутизатор в нашей сети говорит нам, что этот хост не достигаем. Другими словами, наш маршрутизатор не знает как и куда именно перенаправить наши пакеты, чтобы задача была выполнена.

Но мы можем получить доступ к самому роутеру, ведь мы настроили на нем адрес, находящийся в нашей сети.  
Теперь попробуем добавить еще 1 комп в 1 сеть и проверим доступность до него:



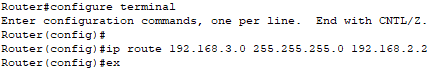


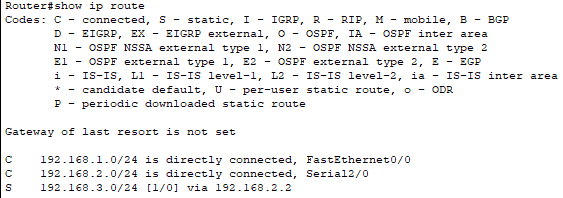
И здесь мы совершили ошибку, думая, что компьютеры из одной lan можно подключить на разные порты роутера, как к коммутатору: мы нарушили само определение маршрутизатора. Роутер создан для того, чтобы соединять **разные** локальные сети в одну, поэтому так просто нам видимость не проверить.



Для того, чтобы получить возможность коммуникации меж различными сетями, настроим статические таблицы маршрутизации:

На роутере 1—2:





Здесь мы буквально говорим, что, мол, ты (роутер), видишь напрямую только 1 и 2 сеть, но если надо будет посылать что-то в 3-ю сеть (с маской), то ты направь их на 192.168.2.2, там будет (должен быть) другой маршрутизатор, ну он там сам дальше разберется.  
Повторим маршруты для каждой сети:

Router(config)#ip route 192.168.4.0 255.255.255.0 192.168.2.2

Router(config)#ip route 192.168.5.0 255.255.255.0 192.168.2.2

Router 2—3:

Router(config)#ip route 192.168.1.0 255.255.255.0 192.168.2.1

Router(config)#ip route 192.168.4.0 255.255.255.0 192.168.3.2

Router(config)#ip route 192.168.5.0 255.255.255.0 192.168.3.3

Router 3—4:

Router(config)#ip route 192.168.1.0 255.255.255.0 192.168.3.1

Router(config)#ip route 192.168.2.0 255.255.255.0 192.168.3.1

Router(config)#ip route 192.168.5.0 255.255.255.0 192.168.3.3

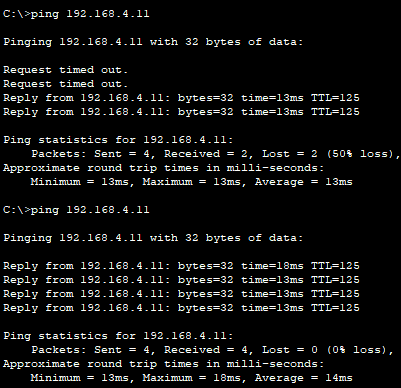
Router 3—5:

Router(config)#ip route 192.168.1.0 255.255.255.0 192.168.3.1

Router(config)#ip route 192.168.2.0 255.255.255.0 192.168.3.1

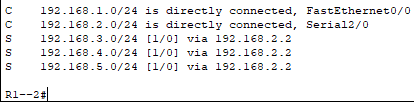
Router(config)#ip route 192.168.4.0 255.255.255.0 192.168.3.2

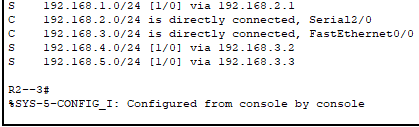
Проверим работает ли наша взаимосвязь теперь:

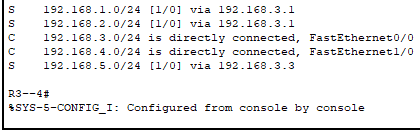


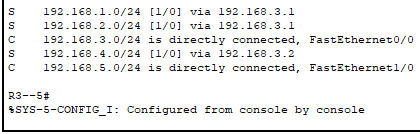
Да, видим, что работает, но почему в первый раз мы потеряли 2 пакета? Дело в том, что мы настроили ip адреса, но в таблице маршрутизации фигурируют MAC адреса, чтобы дальше передать данные по канальному уровню. В первый раз наши маршрутизаторы посылали ARP запросы, чтобы понять, а кто именно скрывается под ip адресом (какой же нам нужен MAC адрес). Пока они его отправляли, пока им приходил ответ и обновлялись таблицы, пока пакеты посылались, и процедура повторялась на следующем роутере, время ожидания на компе-источнике проходило. Во второй же раз таблицы уже были заполнены маршрутами.

Таблицы маршрутизации:









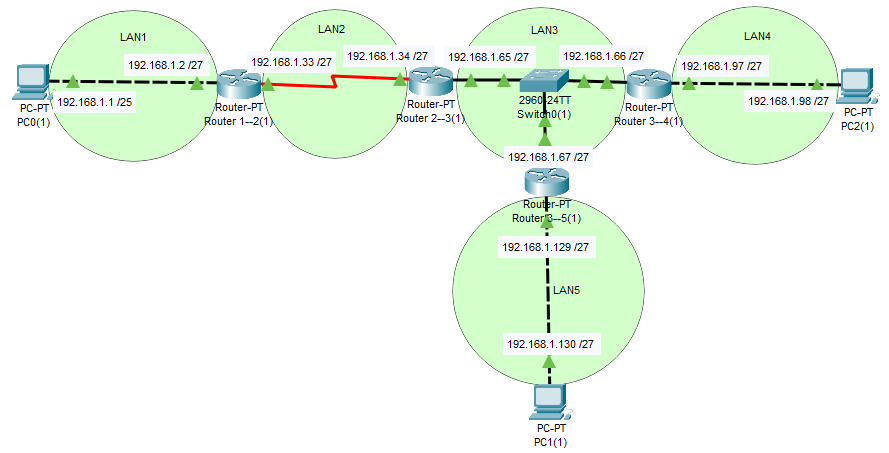
Часть 3

Нам нужно поделить сеть 192.168.1.0/24 на 5 подсетей, количество адресов в каждой будет 256/5 = 51 адрес, но мы не можем выделить количество-не-степеней-двойки, поэтому выделим по 32 адреса.

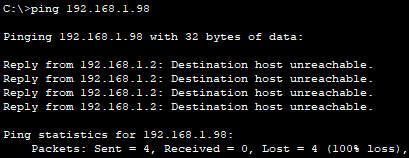
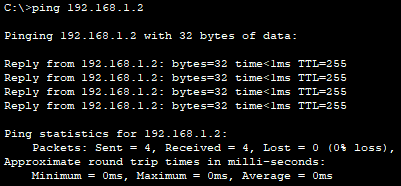
Если мы поделим сеть 256 на подсети по 32 адреса, у нас может вместиться максимум 8 подсетей (256/32=8). То есть нам нужно 8 разных сетей, надо их как-то различать, а делаем мы это только с помощью маски, то есть нам нужно **3 бита**, чтобы различить все 8 сетей:

1100 0000 . 1010 1000 . 0000 0001 . 000 00000 (192.168.1.0/27)   
1100 0000 . 1010 1000 . 0000 0001 . 001 00000 (192.168.1.32/27)  
1100 0000 . 1010 1000 . 0000 0001 . 010 00000 (192.168.1.64/27)  
1100 0000 . 1010 1000 . 0000 0001 . 011 00000 (192.168.1.96/27)  
1100 0000 . 1010 1000 . 0000 0001 . 100 00000 (192.168.1.128/27)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Информация по подсетям | | | | |
| Двоичный адрес сети | Адрес сети | Первый возможный адрес хоста | Последний возможный адрес хоста | Широковещательный адес |
| 1100 0000 . 1010 1000 . 0000 0001 . 000 00000 **/27** | 192.168.1.0 | 192.168.1.1 | 192.168.1.30 | 192.168.1.31 |
| 1100 0000 . 1010 1000 . 0000 0001 . 001 00000 **/27** | 192.168.1.32 | 192.168.1.33 | 192.168.1.62 | 192.168.1.63 |
| 1100 0000 . 1010 1000 . 0000 0001 . 010 00000 **/27** | 192.168.1.64 | 192.168.1.65 | 192.168.1.94 | 192.168.1.95 |
| 1100 0000 . 1010 1000 . 0000 0001 . 011 00000 **/27** | 192.168.1.96 | 192.168.1.97 | 192.168.1.126 | 192.168.1.127 |
| 1100 0000 . 1010 1000 . 0000 0001 . 100 00000 **/27** | 192.168.1.128 | 192.168.1.129 | 192.168.1.158 | 192.168.1.159 |

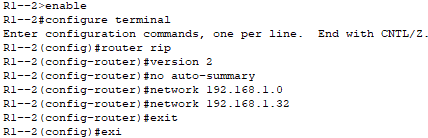


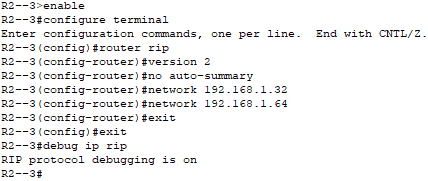
На первом компьютере до его роутера мы можем достучаться, а до компьютера в 4 сети – нет:



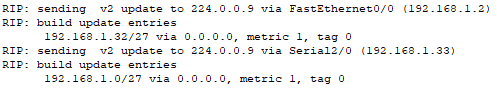
Настало время настроить работу протокола RIP2, чтобы наши таблицы маршрутизации обновлялись автоматически!

enable  
configure terminal  
router rip  
version 2  
no auto-summary  
network 192.168.1.64  
network 192.168.1.128  
passive-interface FastEthernet1/0  
exit  
exit  
debug ip rip



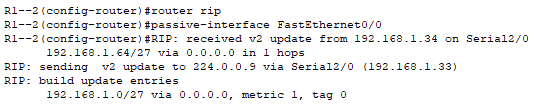


Сразу же видим следующую особенность:



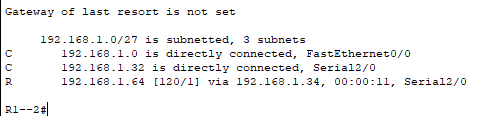
Наш маршрутизатор отправляет свою таблицу на оба порта, а на другом конце FastEthernet0/0 (192.168.1.2) находится компьютер. Нам бы не хотелось нагружать клиента или предоставлять техническую информацию во избежание вреда.

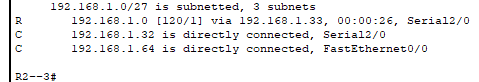
Поэтому объявим порты с компьютерами пассивными (на них не будут посылаться таблицы):



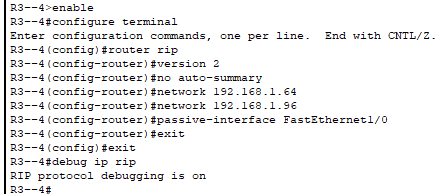
Как мы видим, больше через порт FastEthernet0/0 маршрутизатор информацию не передает.

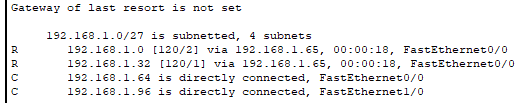
А соседнюю сеть мы видим:



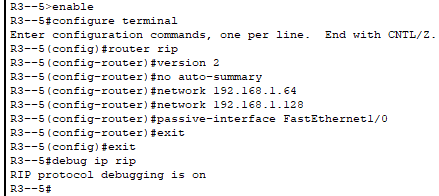


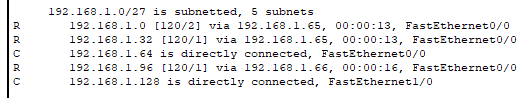
Настроим роутер, соединяющий 3 и 4 сети:



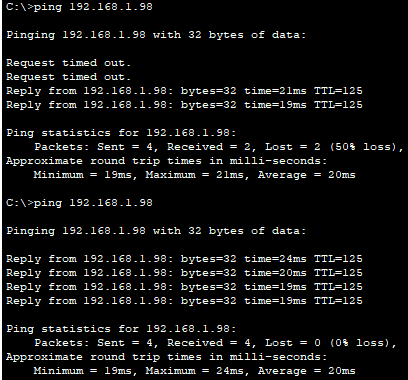


Настроим роутер, соединяющий 3 и 5 сети:

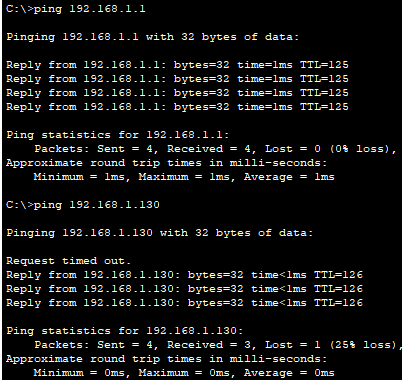




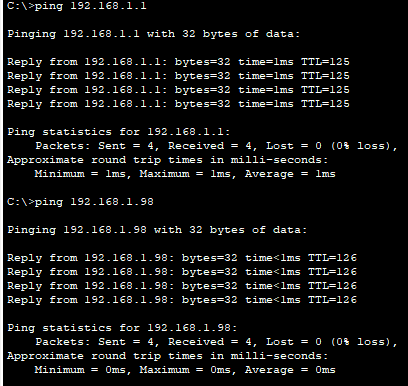
Попробуем проверить соединение между компьютерами в 1 и 4 сети, видим такую же историю:



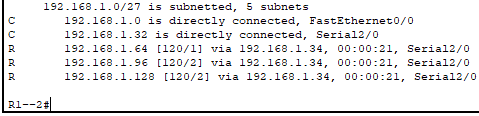
Причем на 4 компьютере мы имеем более интересную ситуацию:

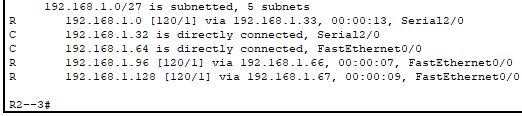


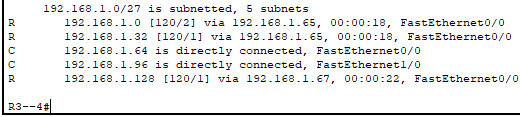
До 1 компьютера мы доходим без всяких проблем из-за уже обученных маршрутизаторов, но вот до компа в 5 сети мы дошли быстрее, чем 1 до нас. Это связано с банально более коротким путем.

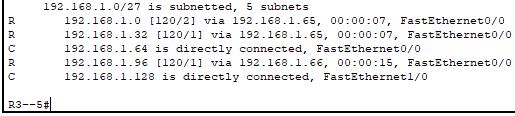
А при соединении от 5 аппарата к первому вообще не было проблем: 

Таблицы маршрутизации:



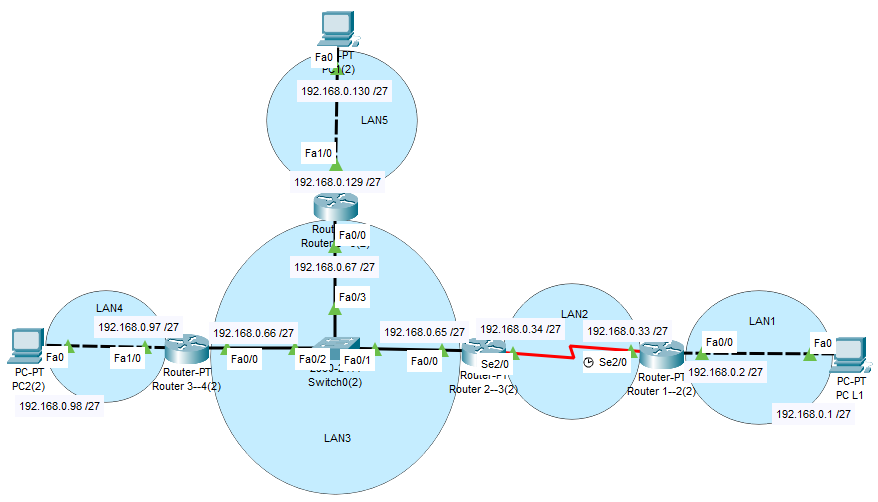




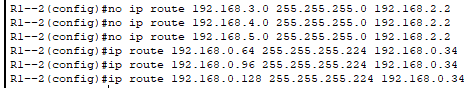


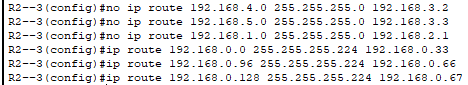
Часть 4

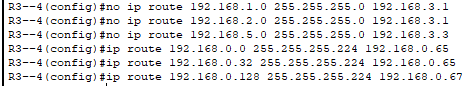
Заметив, что в первой части работы нужно было брать подсети из 192.168.0.0/24, мы скопировали сеть из первой части и разбили по тому же принципу, что и во второй части, изменив сеть:

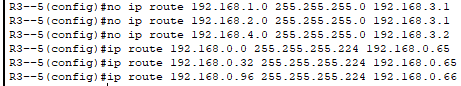


Затем мы перенастроили статическую маршрутизацию на всех роутерах:

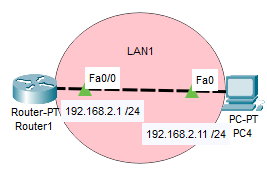
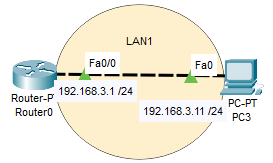




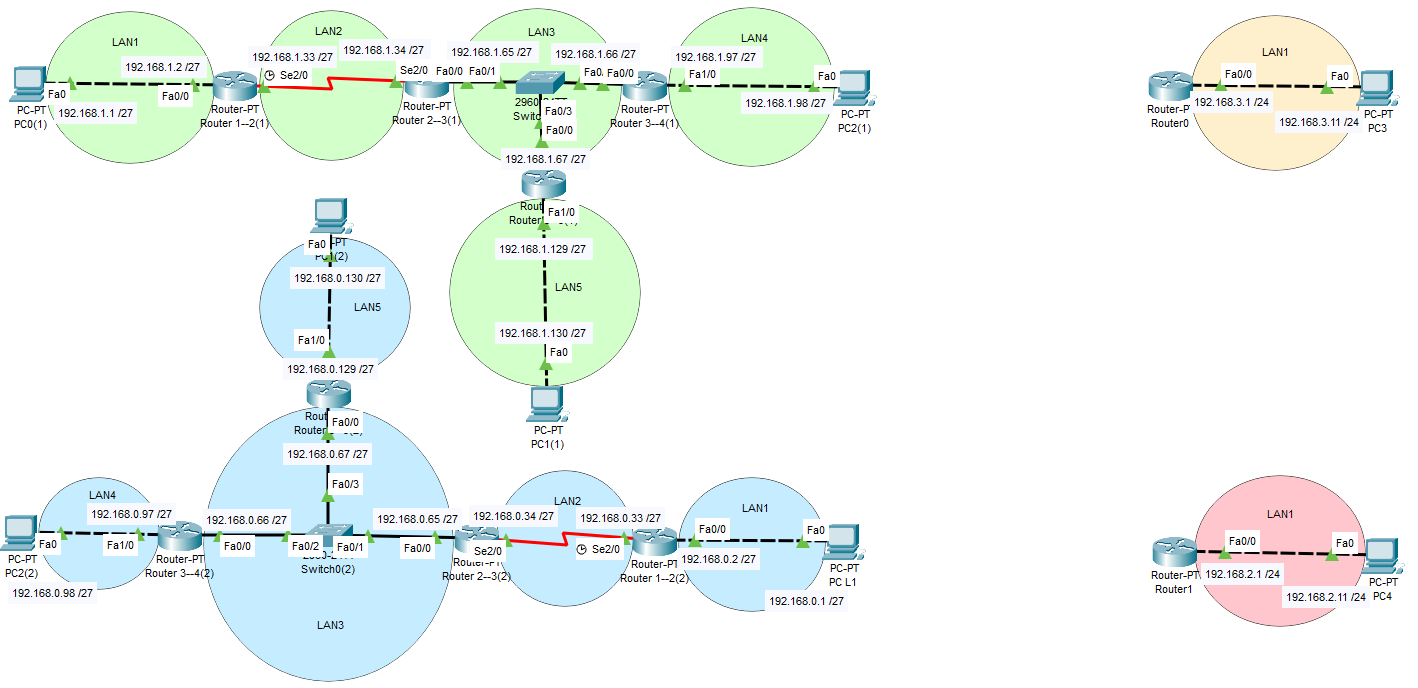




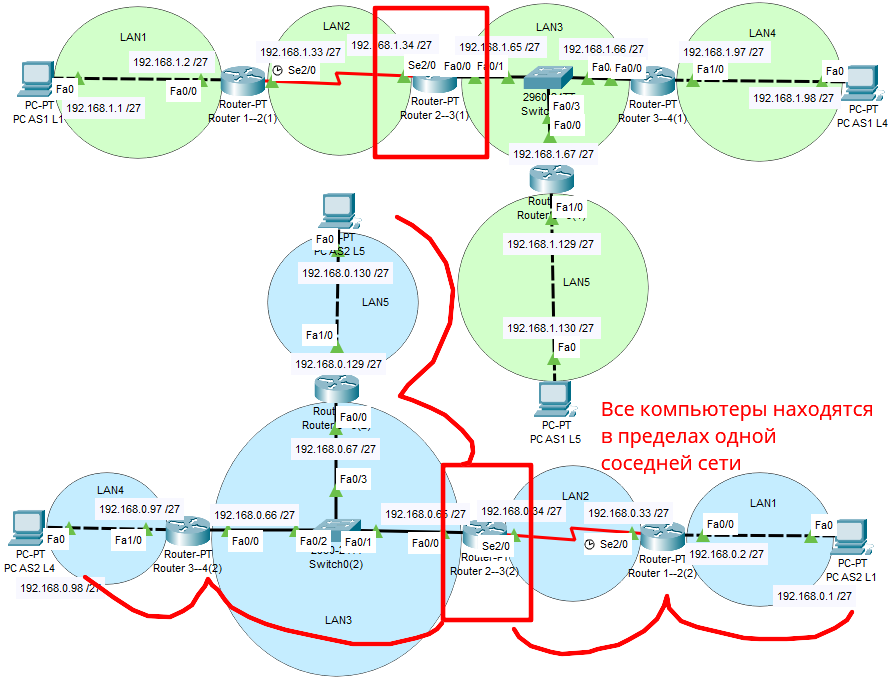
А также создали 2 дополнительные сети:

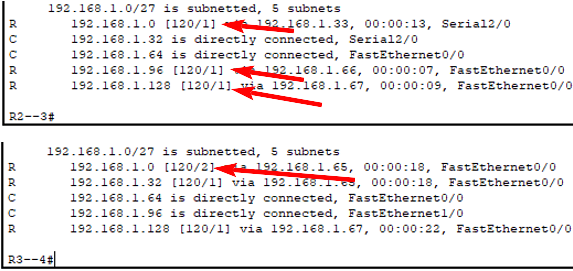
Общая схема сетей:



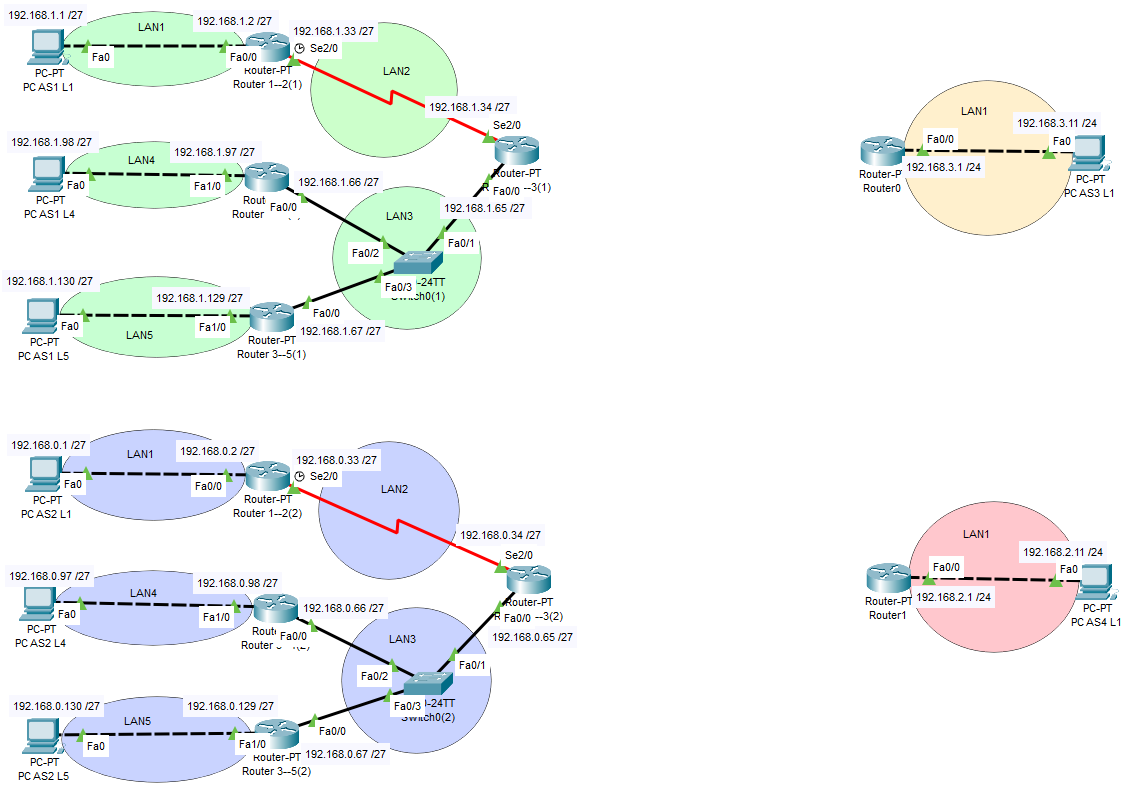
Но заметим заранее, что коммуницирующими маршрутизаторами будут те, у которых пути до компьютеров будут минимальны:



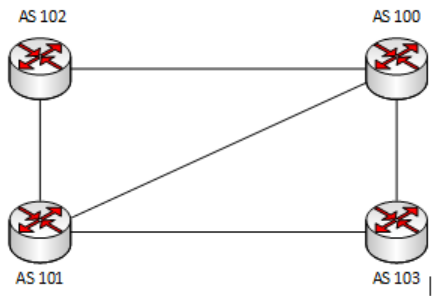
Мы могли наблюдать это раньше при просмотре таблиц маршрутизации:



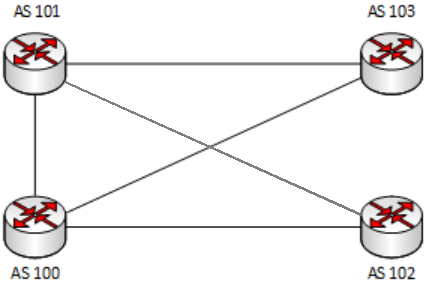
Поэтому переставим сеть в программе, чтобы последующее подключение было более наглядным:

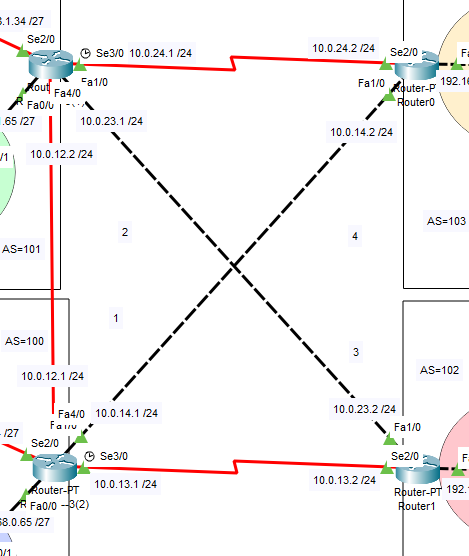


По заданию мы должны соединить наши автономные системы следующим образом:

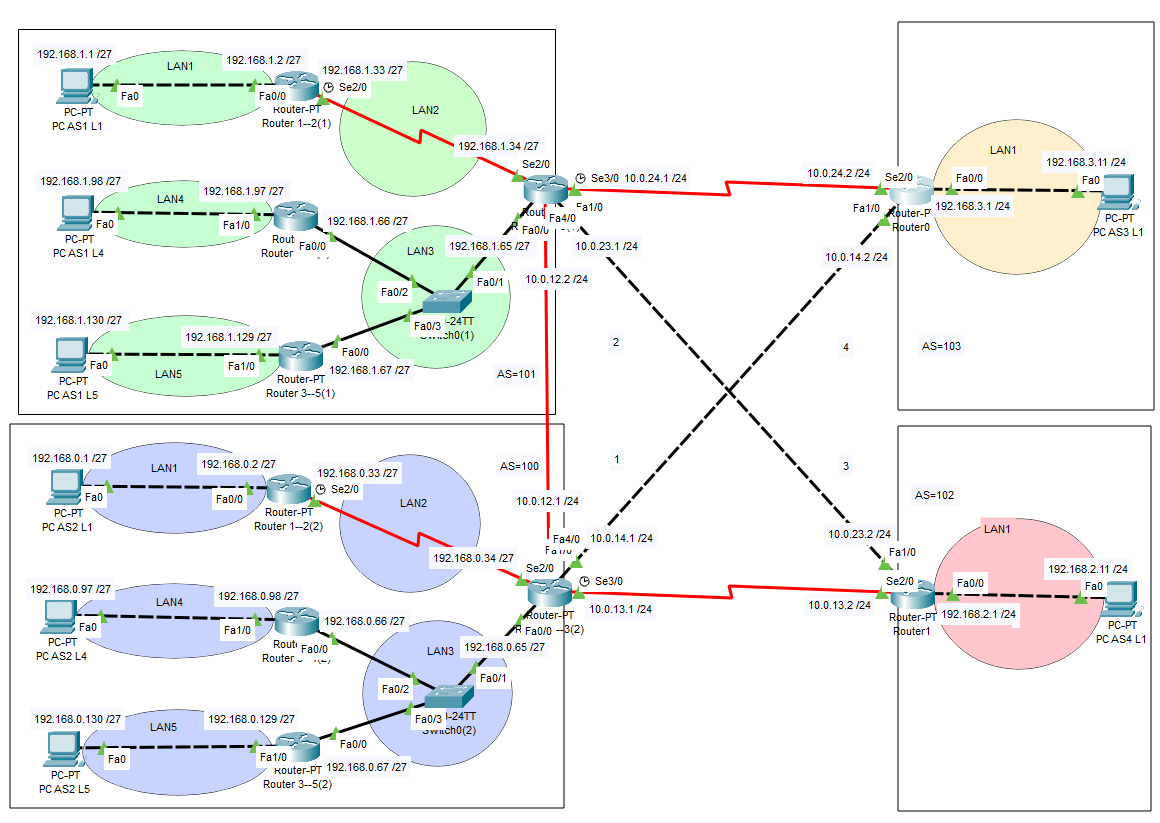


Преобразуем схему, интерпретируя наше расположение сетей:

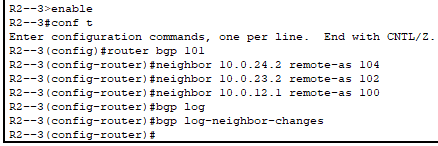


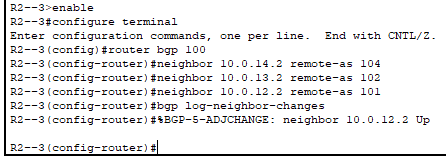


При этом общая схема с указанными областями:

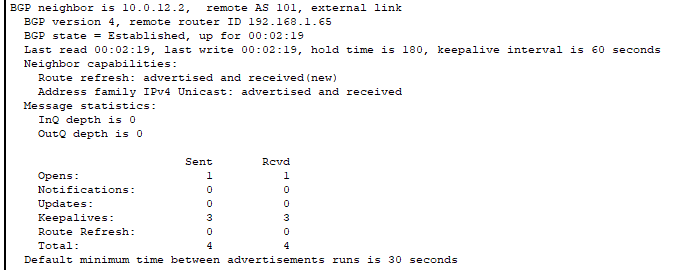


Соединения налажены, приступим к конфигурации протокола eBGP:

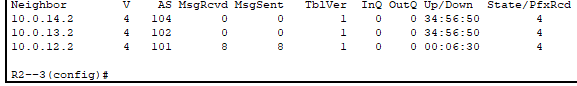




Сразу заметим, что роутер увидел запрос соседа и соединился с ним, посмотрим подробную информацию командой **do show ip bgp neighbors:**

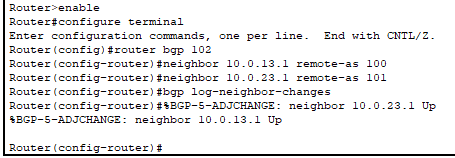


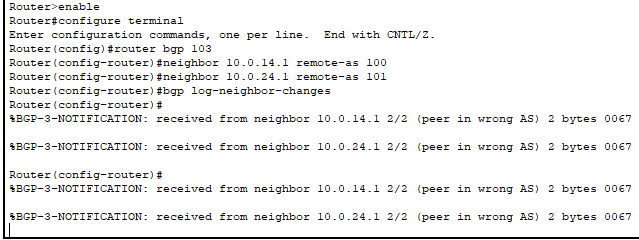
Также посмотрим на состояние соединения командой **do show ip bgp summary**:



Как мы видим, за 6 минут существования соединения они уже 8 раз автоматически послали друг другу уведомления о своём существовании.

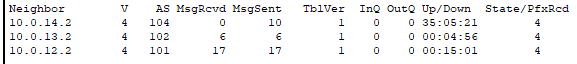
Продолжим конфигурацию:



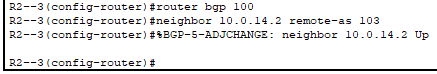


Получаем сообщения, что кто-то, кого мы на самом деле не знаем, посылает нам какие-то сообщения, а точнее, нам приходит пир от неизвестной автономной системы. Значит, что-то не так..

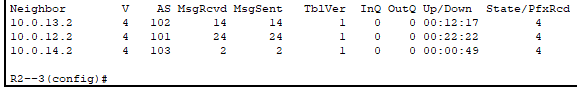
Заметим теперь, что при конфигурации была допущена ошибка:



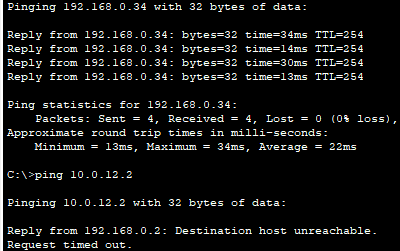
Мы указали номер автономной системы как 104, что не совпадает с условиями. Все-таки при достаточно большом наборе циферок сказывается человеческий фактор :)  
Исправим ситуацию:  

Теперь наш маршрутизатор из автономной системы 100 видит все другие системы:

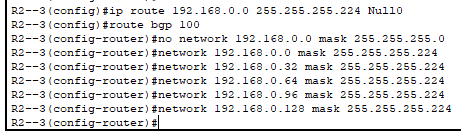


Однако, внутри 100 сети другие маршрутизаторы и хосты не могут получить доступ к другим системам:

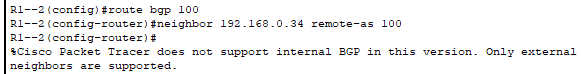


Это связано с тем, что роутеры по-прежнему не знают как попасть за нашу сеть, это знает только крайний роутер. При этом сам протокол BGP является внешним, да еще и взаимным: крайний роутер должен знать что за локальные сети находятся позади него, но сам он узнать это не может. Это работа протоколов внутренней маршрутизации (RIP, OSPF и т.п.).

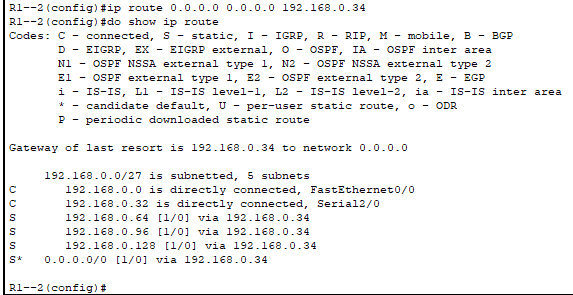
Следовательно, нам нужно указать явно крайнему роутеру в 100-й системе что за сети он представляет:



Если мы попробуем настроить на внутренних роутерах iBGP, чтобы уметь меняться с крайним роутером, то увидим, что данная технология не поддерживается программой:



Поэтому нам нужно настроить шлюза по умолчанию наш крайний роутер, чтобы отправлять все неизвестные в нашей автономной сети пакеты именно туда:

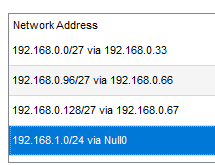


Однако, у нас ничего не пинговалось, мы начали искать проблемы.

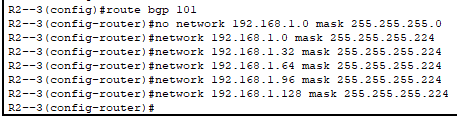
Сначала мы выяснили, что **забыли добавить LAN1** в системах 102 и 103, добавили, теперь компьютеры 192.168.2.11/24 и 192.168.3.11/24 доступны.

Затем мы пытались поставить анонсирование, но ничего не вышло

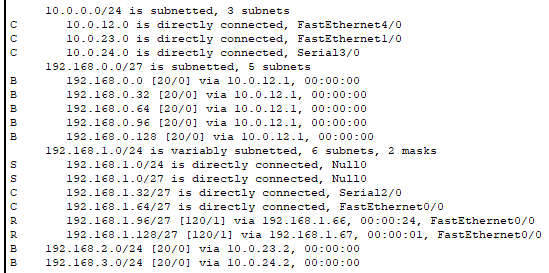
После оказалось, что мы **перепутали сети местами** и написали неправильные адреса… Мы поменяли все как надо, но все равно ничего не работало.

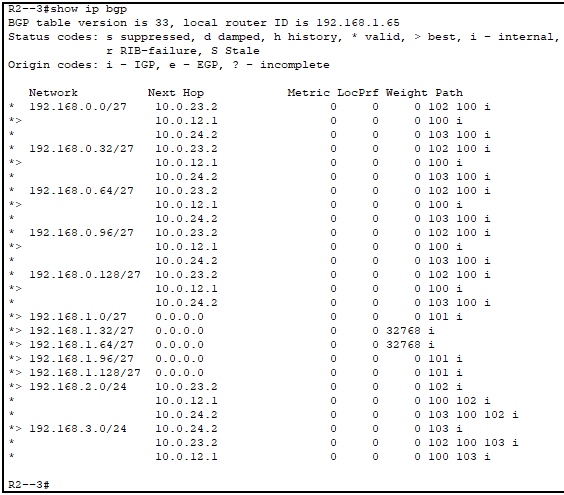


В отчаянии мы решили откатиться с анонсирования общей сети на прямое указание все сетей:



Здесь мы явно указали сети еще и 101-й системе, однако, это можно было заменить командой default-information originate, чтобы роутер сам узнал обо всех сетях посредством автоматической работы RIP2.





Увы, безуспешно. Мы бились в холодной беспомощной истерике...

Но оказалось, это было из-за того, что мы **забыли поставить на остальные роутеры шлюз** по умолчанию для сливания пакетов.

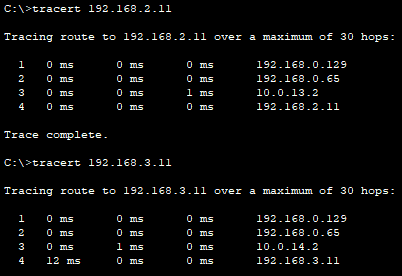
en

conf t

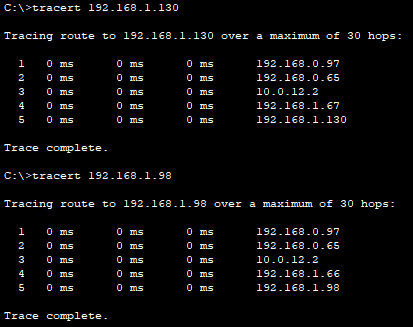
ip route 0.0.0.0 0.0.0.0 192.168.0.65

Наконец задача была решена!

Трассировка пути от компьютера в сети №5 системы 100 до хостов в сетях 102 и 103 соответственно:



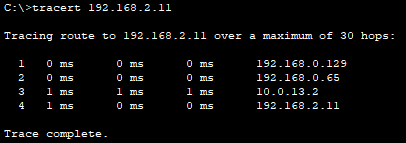
Трассировка пути от компьютера в сети №4 системы 100 до хостов в сетях 5 и 4 системы 101 соответственно:

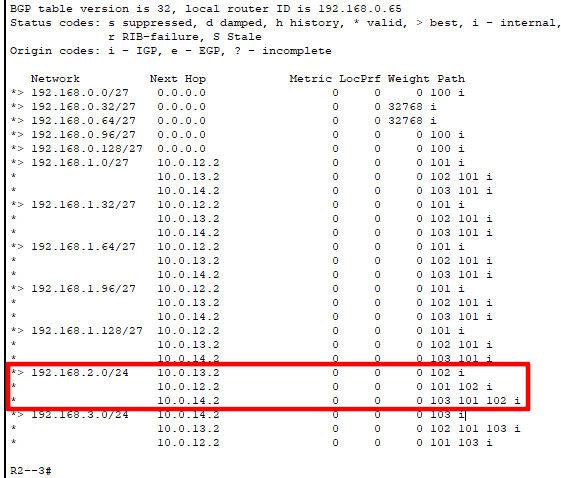


Теперь проверим на эксперименте как протокол eBGP выберет оптимальный путь для доставки пакетов между компьютером в 5-й сети автономной системы 100 и хостом в системе 102:

Как мы можем видеть по самой топологии и выводе команды **show ip bgp**, мы можем попасть туда тремя разными путями:

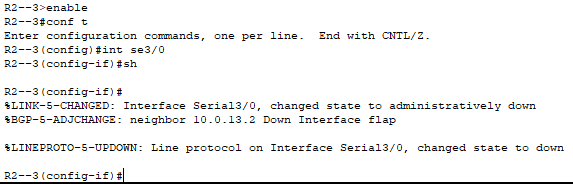
1. Пройти напрямую из системы 100 в систему 102.
2. Обойти через систему 101 и попасть в систему 102.
3. Сходить сначала в систему 103, оттуда (т.к. там нет прямой связи с 102-й) отправиться в систему 101, а уж только затем в систему 102, до нашего компьютера.

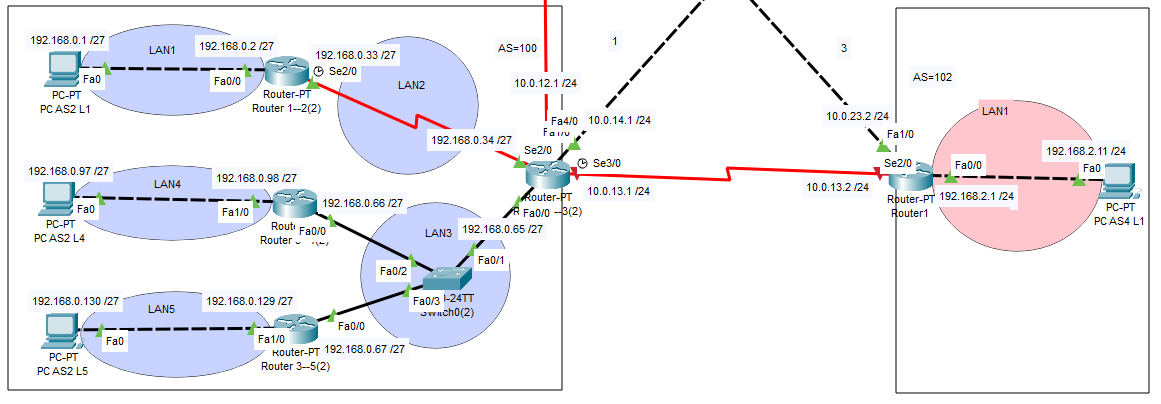


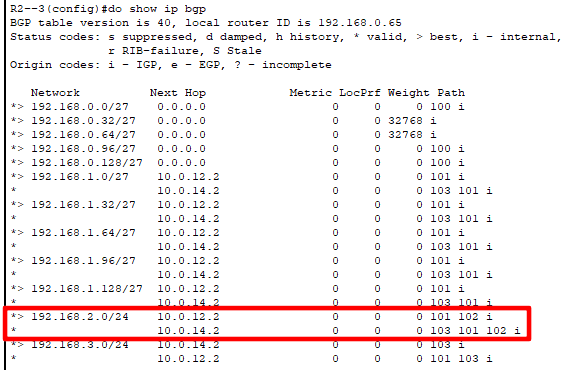


Заметим, что протокол BGP, в отличие от RIP, хранит не только самый оптимальный путь, как раз для таких случаев, который мы имитируем:

Отключим прямой путь с системой 102 и проведем трассировку ещё раз:

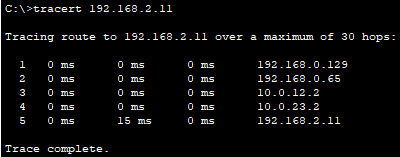






Теперь мы можем наблюдать, что достичь нужной системы мы можем только обходным путём, а добраться до системы 101 у нас возможностей меньше из-за транзитивности уже через 102 систему.

Произведем трассировку:



Действительно, видим, что маршрут был подобран самым на этот момент оптимальным образом.