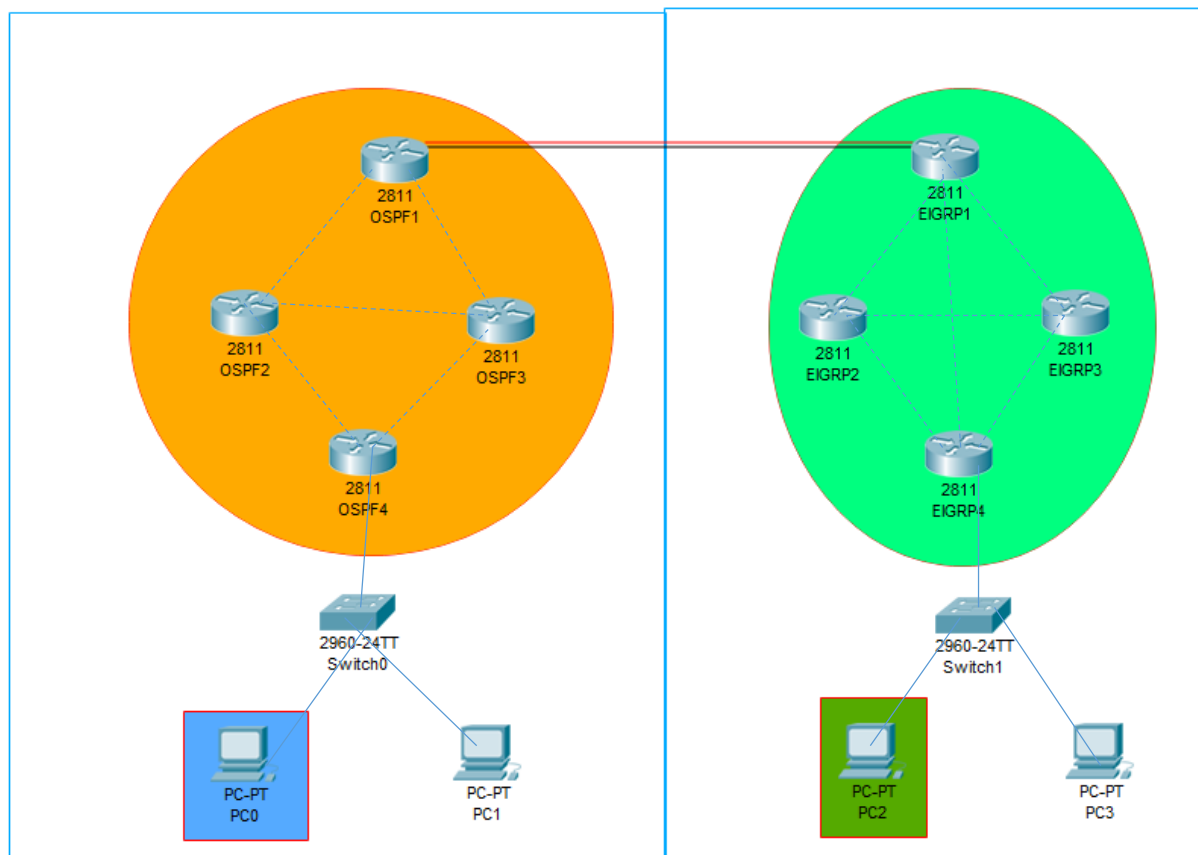


Предварительные настройки

Итак, топология. Наверное, у вас будет другая, в таком случае надеюсь, что этот мануал всё же будет вам полезен:



Оранжевая часть – там, где между роутерами должна быть OSPF-маршрутизация.

Зеленая – EIGRP.

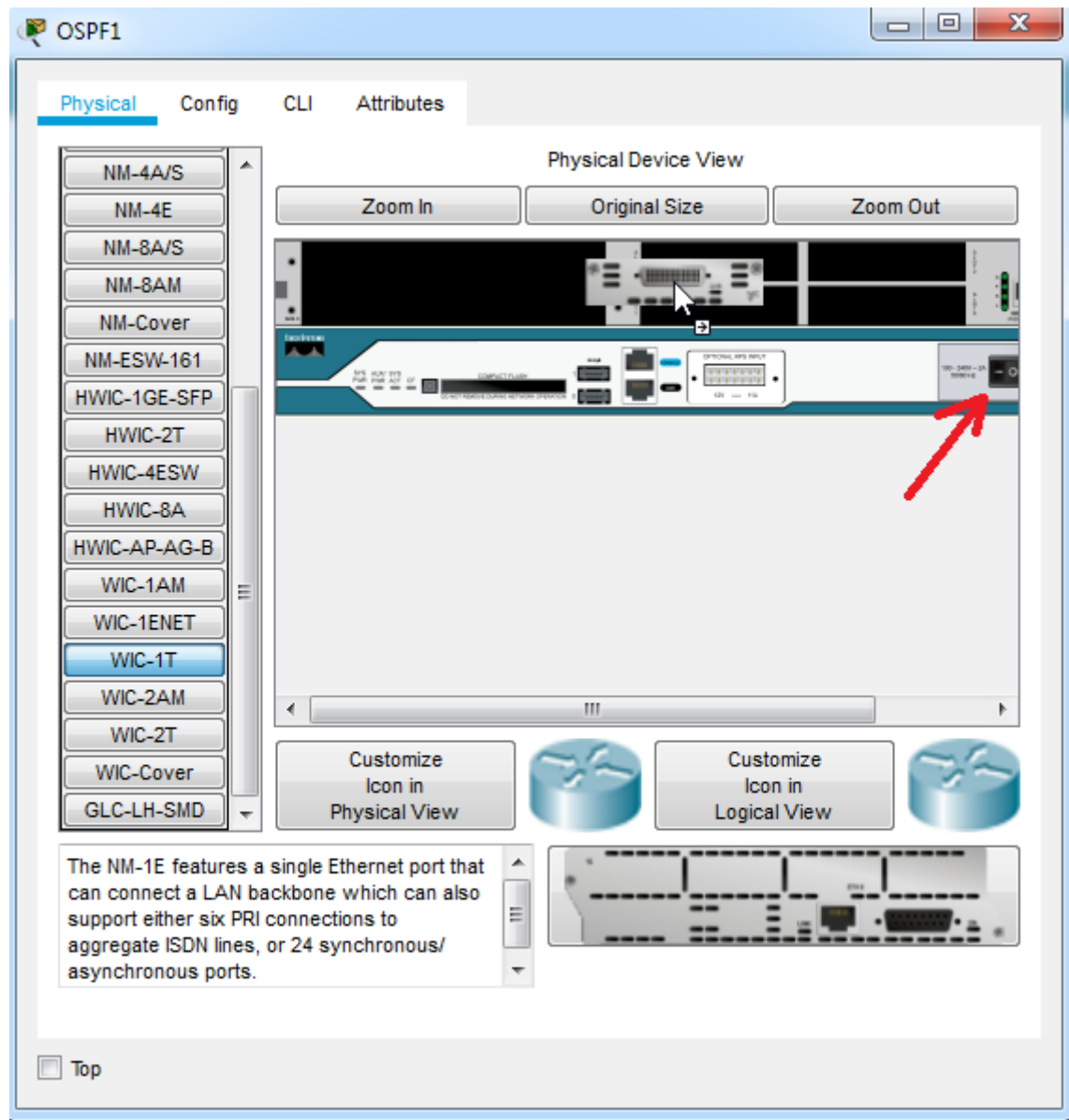
Голубой компьютер в левой нижней части и зеленый в правой – находятся в отдельных виланах.

Оглавление

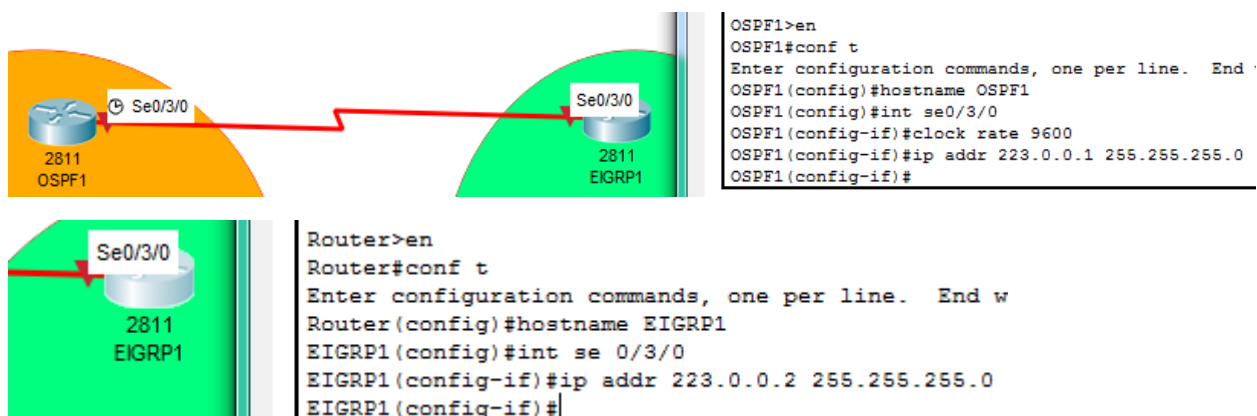
Предварительные настройки.....	1
Настройка OSPF:	14
Настройка EIGRP:.....	17
Настройка статической маршрутизации между EIGRP и OSPF:.....	20
Дополнительные вопросы и задания:.....	22

Между верхними роутерами проходит serial-кабель. Для начала добавим на верхних роутерах модуль WIC-1T, для этого тыкнем по ним и перейдём на вкладочку Physical. Здесь обязательно предварительно отключить роутеры, поскольку вставлять модуль под напряжением («на горячее») строго запрещается, красной стрелкой показано, куда нужно тыкнуть, чтобы отключить.

После этого вставляем наш модуль.



Настройки верхних роутеров:



Несколько пояснений по командам:

[hostname] – нужна просто, чтобы поменять имя роутера в командной строке. Это нужно было мне при создании мануала, чтобы лучше показать материал, но в принципе не обязательно его менять в вашей работе.

[int se 0/3/0] – для тех, кто к четвертой УТП только вышел из запоя или вылез из танка :D Так мы заходим на интерфейс для его настройки. Чтобы выйти можно написать [exit].

[clock rate] – эта штука по сути настраивает пропускную способность. Serial-кабели тем и прекрасны, что можно настроить какую хотите скорость передачи данных. Подробнее читайте тут: <http://ciscotips.ru/clockrate> (очень советую заглянуть из-за важных команд, там написанных). Я лишь обращаю ваше внимание, что подключение выполняется при помощи вот такого кабеля:



Какой из роутеров вы подключите первым будет DCE, на нём вы и выполняете команду clock rate.

[ip address] – это мы выдаём айпишники. В моём примере я решил использовать 223.0.0.0 сеть для связи между роутерами. У левого роутера айпишник 223.0.0.1, у правого 223.0.0.2. Также вы можете использовать линки, что это такое читайте в «Сетях для самых маленьких»:

<https://linkmeup.ru/blog/14.html> - ctrl+F и поиск по слову "линк" - о линковых сетях и бесклассовой IP-адресации

Но в данном примере для простоты будем использовать сети с маской 255.255.255.0

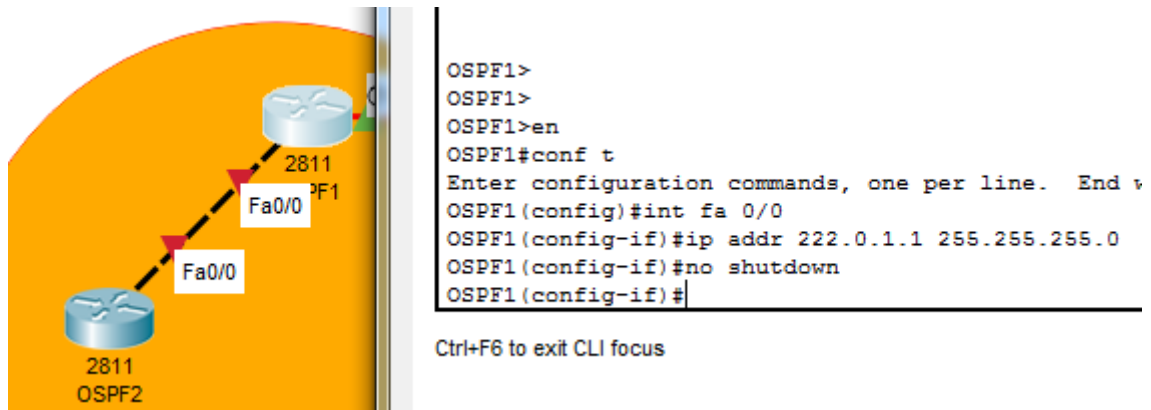
Не забудьте включить [se 0/3/0] интерфейсы на обоих роутерах командой [no shutdown]

Перейдём к настройке сегментов между роутерами.

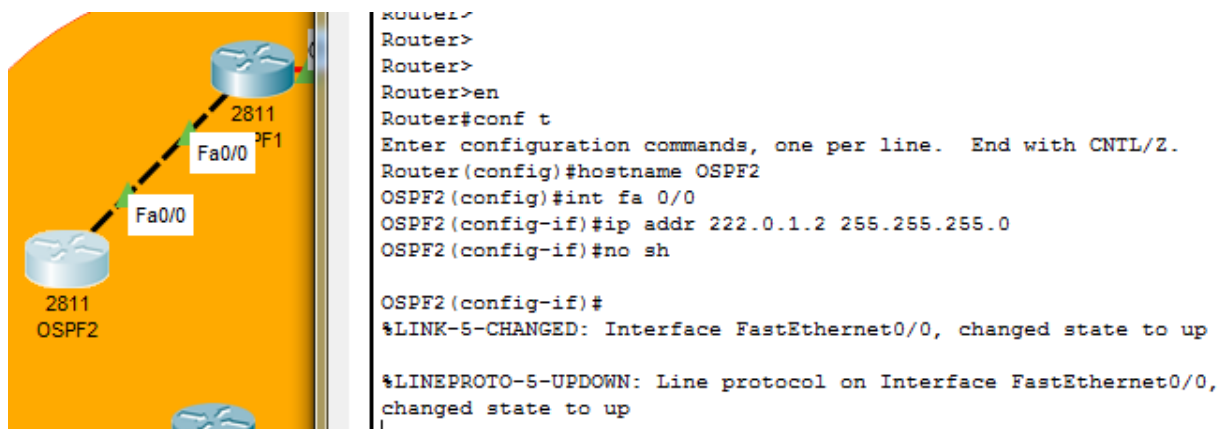
Сегмент сети – участок сети, устройства которого делят общую полосу пропускания.

В качестве сети между роутерами использую 222.0.1.0

Верхний из них будет иметь в этом сегменте IP 222.0.1.1



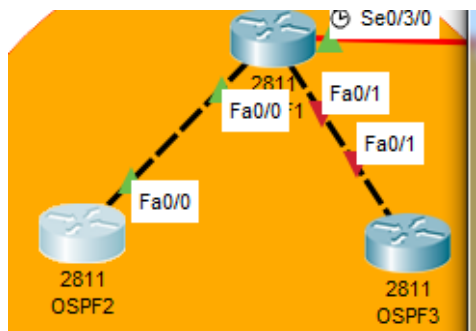
А нижний соответственно 222.0.1.2



Аналогично будут настраиваться и остальные сети.

Тут я вспомнил, что для полного построения топологии нам необходимо, чтобы на средних и нижнем роутерах было по три интерфейса FastEthernet.

На уже включенном и настроенном OSPF2 произвожу такие действия:

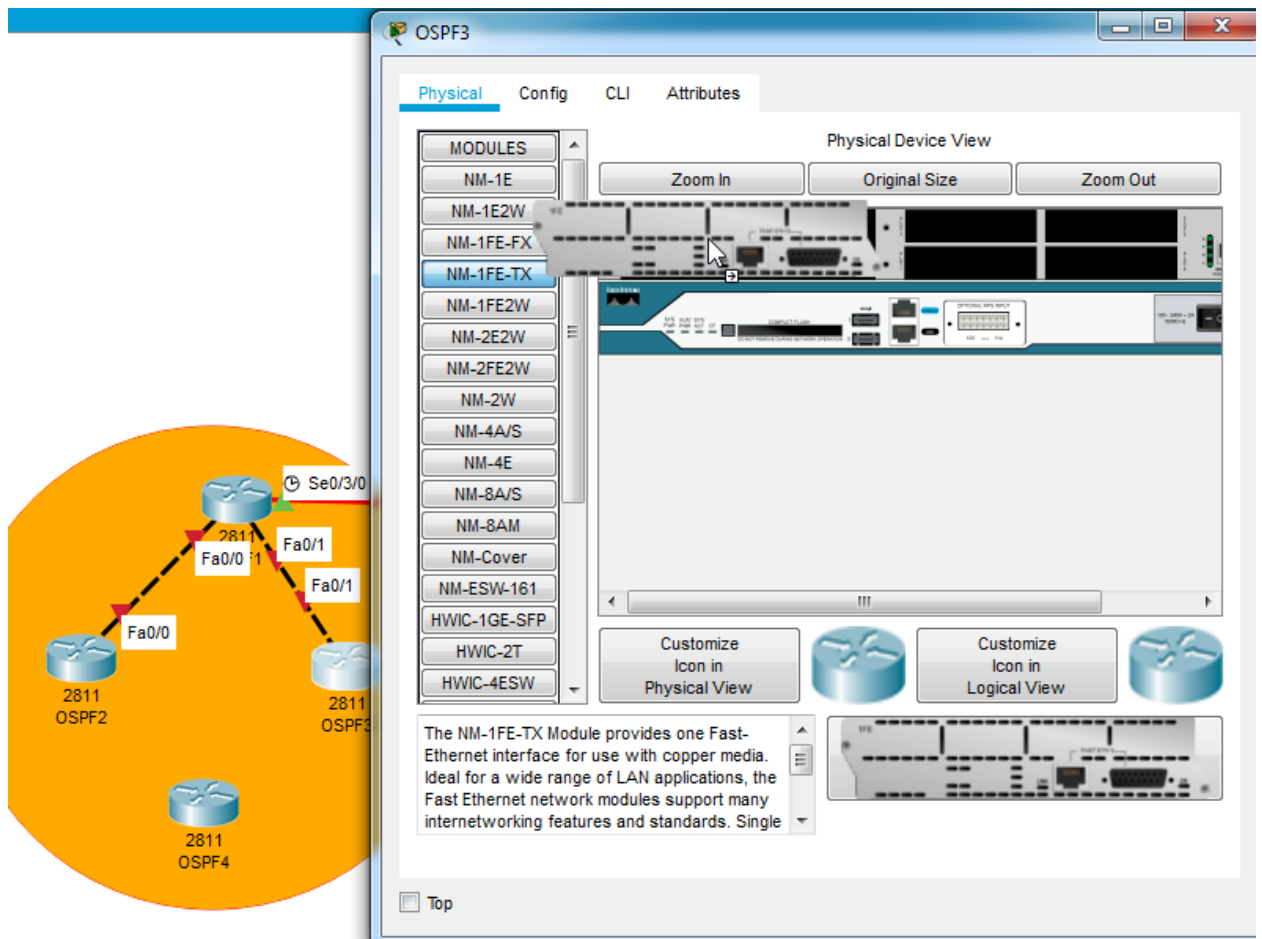


```
OSPF2 (config-if)#exit
OSPF2 (config)#exit
OSPF2#
%SYS-5-CONFIG_I: Configured from console by console

OSPF2#copy run
OSPF2#copy running-config st
OSPF2#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
OSPF2#
```

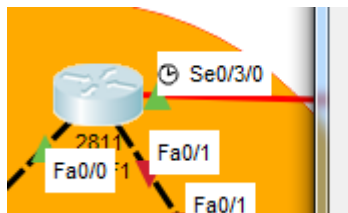
[copy running-config startup-config] – все настройки роутера, как и свитча, хранятся в файлах. Running-config отвечает за текущие настройки. А startup-config – за настройки, которые автоматически восстановятся после перезагрузки устройства. Командой copy мы копируем текущие настройки в файл с «несгораемыми» настройками. Теперь при внезапном отключении мы не потеряем все плоды наших трудов. Хорошо бы так делать на всех устройствах, лишним точно не будет.

Вот теперь можно выключить OSPF2 и вставить в него наш модуль с FastEthernet. Аналогично поступим с OSPF3 – если уже успели на нём настроить сети, то сохранились, как выше.

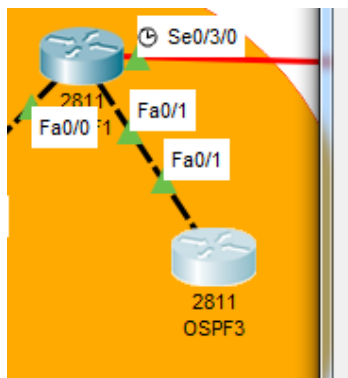


Включаемся и продолжаем настройку.

Дальше всё в принципе видно на скриншотах.



```
OSPF1(config-if)#
OSPF1(config-if)#
OSPF1(config-if)#int fa 0/1
OSPF1(config-if)#ip addr 222.0.2.1 255.255.255.0
OSPF1(config-if)#no sh
```



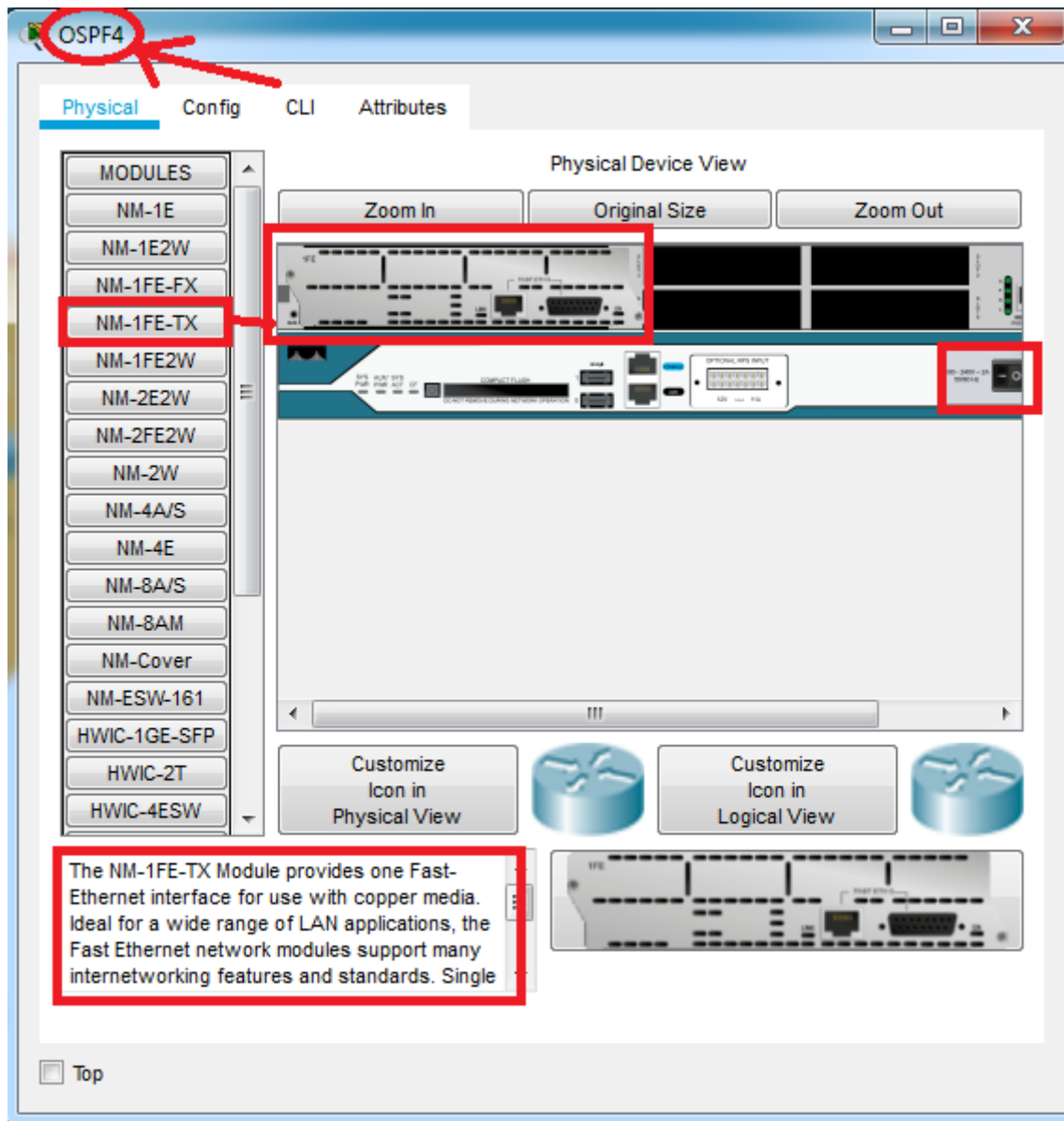
```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname OSPF3
OSPF3(config)#int fa 0/1
OSPF3(config-if)#ip addr 222.0.2.2 255.255.255.0
OSPF3(config-if)#no sh

OSPF3(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
changed state to up
```

```
Router#copy run st
Destination filename [startup-config]?
Building configuration...
[OK]
```

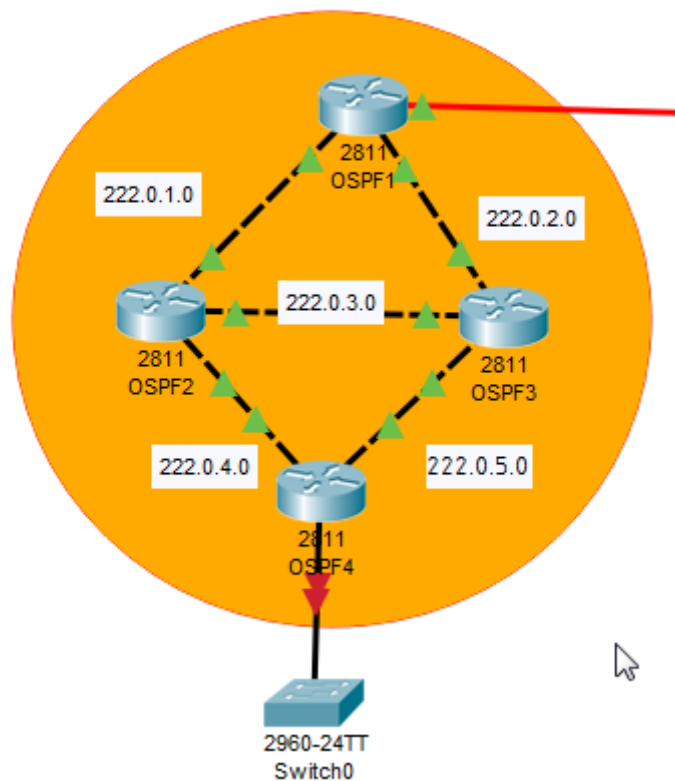
В нижний роутер тоже вставляем NM-1FE-TX.



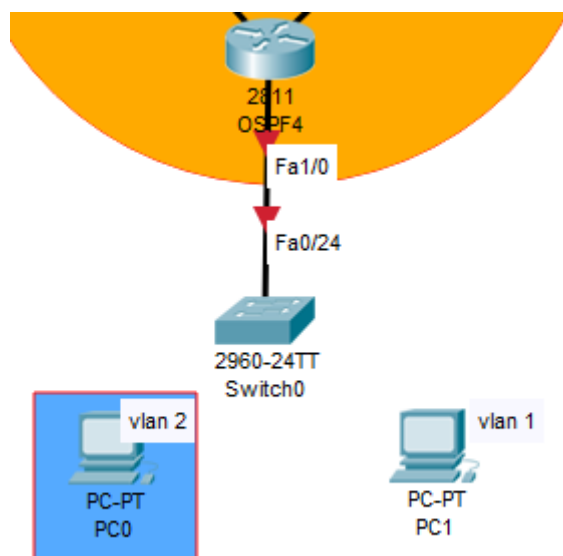
Смотря на самый первый рисунок в документе, настройте аналогично вышеизложенным примерам сеточки.

Снизу можете посмотреть какие конкретно, но можете настроить и свои.

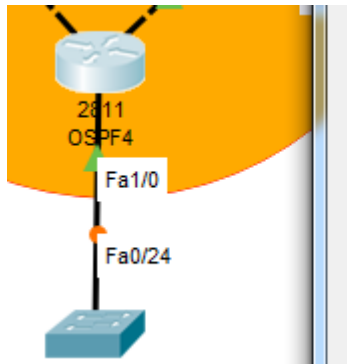
Роутеру OSPF4, предварительно сохранив конфигурацию, тоже добавьте модуль NM-1FE-TX.



Теперь осталось настроить то, что снизу.



Заходим на OSPF4 и делаем так:



```
Router(config)#int fa 1/0.1
Router(config-subif)#enc dot1q 1
Router(config-subif)#ip addr 192.168.1.1 255.255.255.0
Router(config-subif)#no sh
Router(config-subif)#exit
Router(config)#int fa 1/0.2
Router(config-subif)#enc dot1q 2
Router(config-subif)#ip addr 192.168.2.1 255.255.255.0
Router(config-subif)#no sh
Router(config-subif)#exit
Router(config)#int fa 1/0
Router(config-if)#no sh
```

[int fa 1/0.1] – заходим на первый сабинтерфейс fa 1/0

[encapsulation dot1q 1] – настройка приёма тэгированного трафика с первого vlan'а.

[ip address 192.168.1.1 255.255.255.0] – установление айпи-адреса для первого сабинтерфейса.

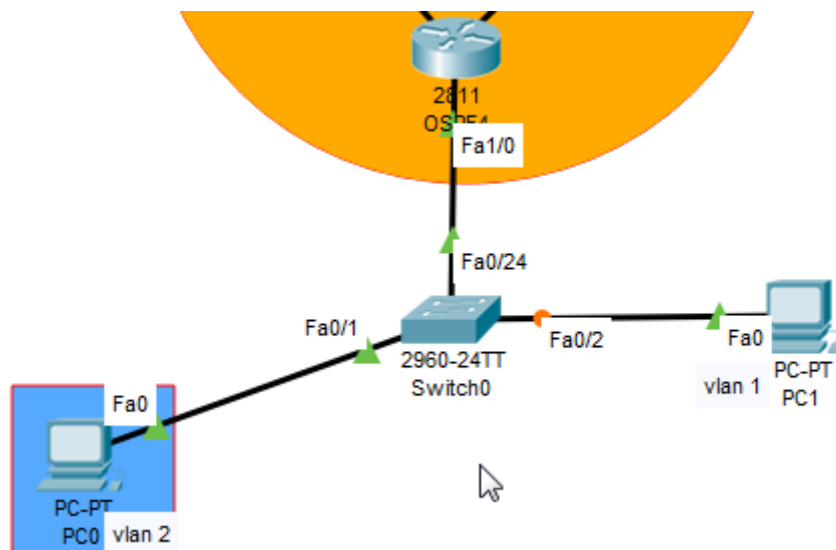
int fa 1/0.2 – аналогично.

Подробнее про это:

<https://linkmeup.ru/blog/14.html> - «Сети для самых маленьких», InterVLAN Routing.

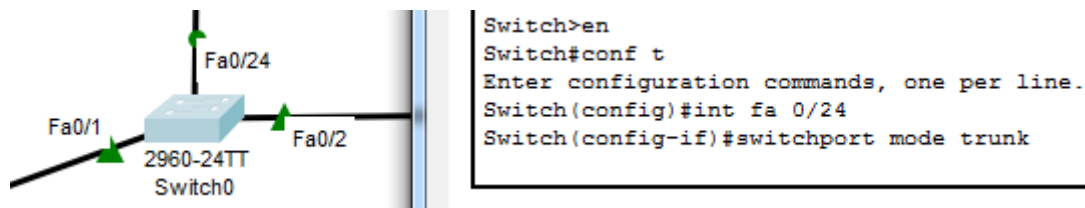
(1-я лабораторная).

А далее настройка Switch0:

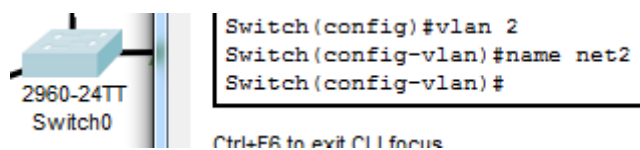


Fa0/2 свитча0 должен принимать трафик от PC1 и помечать его меткой первого вилана. Далее он передаёт его на fa0/24 по trunk'у, который этот тэг вилана сохраняет. На OSPF4 роутер видит, например, ip назначения 192.168.2.2 (это PC0). Он помечает пакет меткой второго вилана и передаёт его на fa1/0.2. Далее наш пакет опять по trunk'у приходит на свитч. Свитч снимает метку второго вилана и отдаёт пакетик PC0 по fa0/1.

Пришло время настроить свитч.

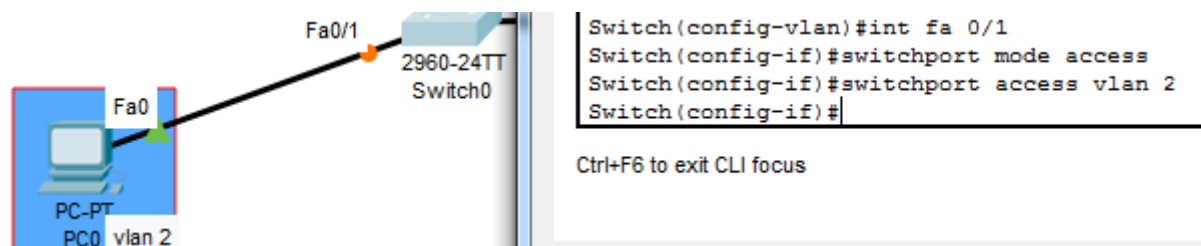


[switchport mode trunk] – многие могут спросить, почему без указания конкретных виланов далее. Дело в том, что так мы обозначаем разрешение всем пакетам любых виланов проходить через данный порт без снятия тега.



[vlan 2] – создание на свитче соответствующего вилана. Во втором утп вы наверное разобрались с VTP и передачей виланов.

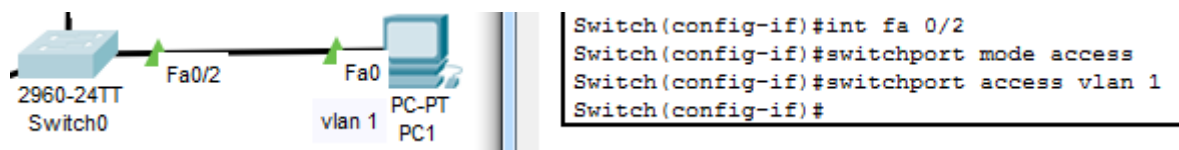
[name net2] – даём вилану имя.



[int fa 0/1]

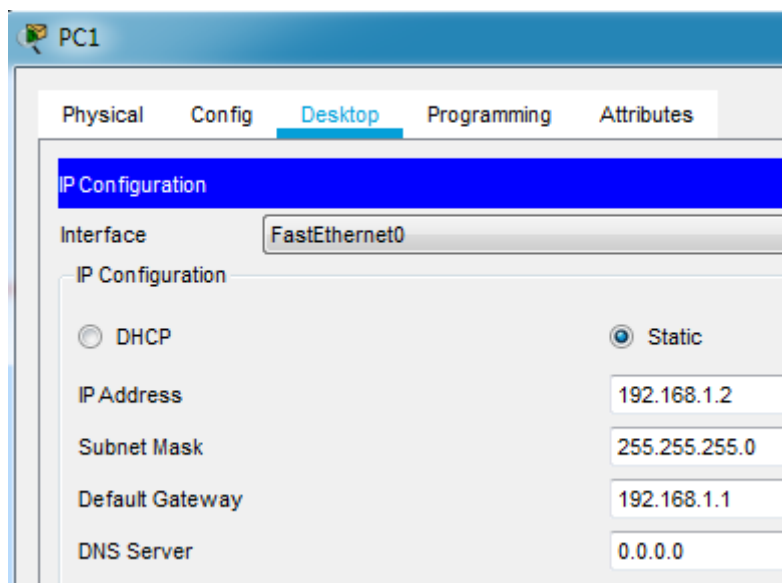
[switchport mode access] – переводим порт в режим установления метки какого-то вилана.

[switchport access vlan 2] – говорим, какой именно меткой пометить.

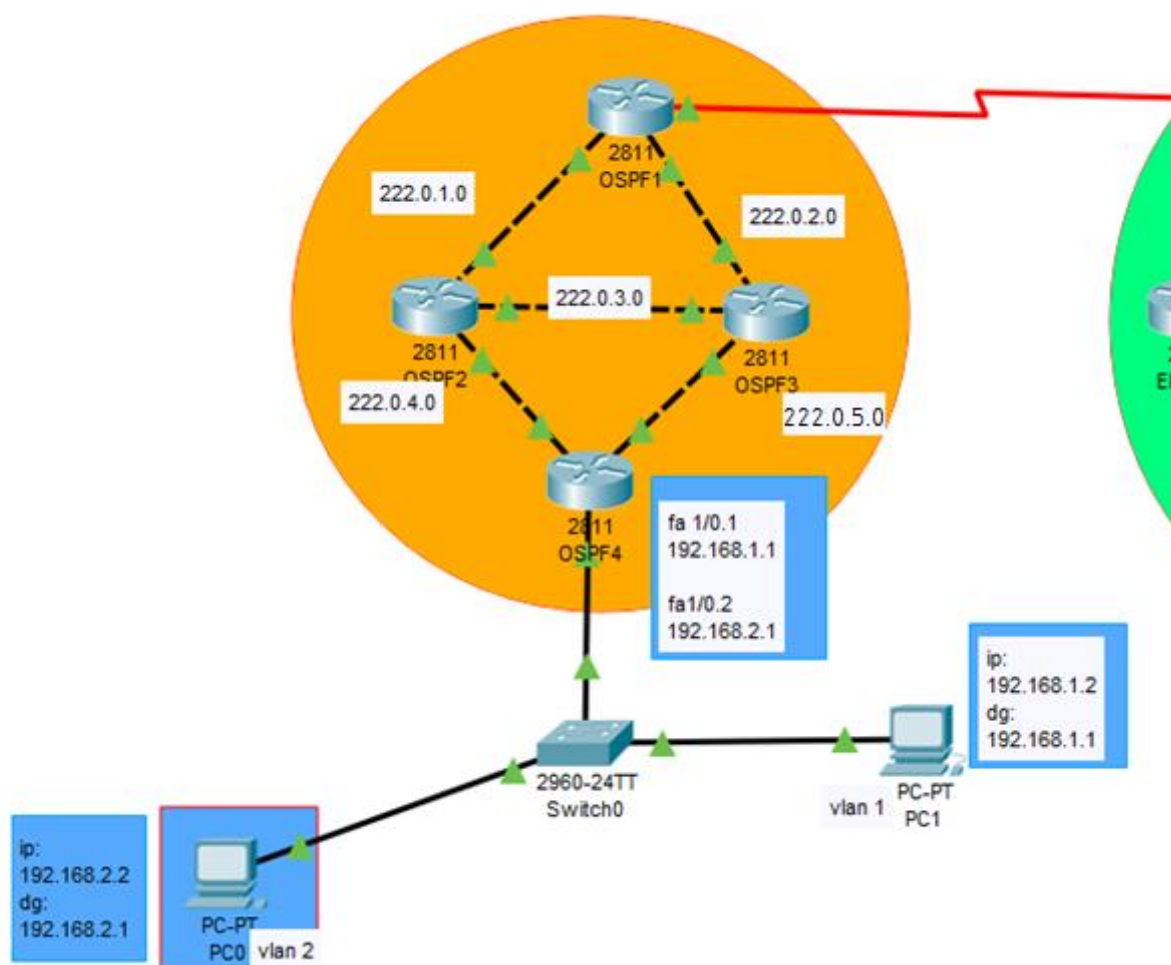


Аналогично.

Если что, как-то так настраивается dg (default gateway) для компьютера:



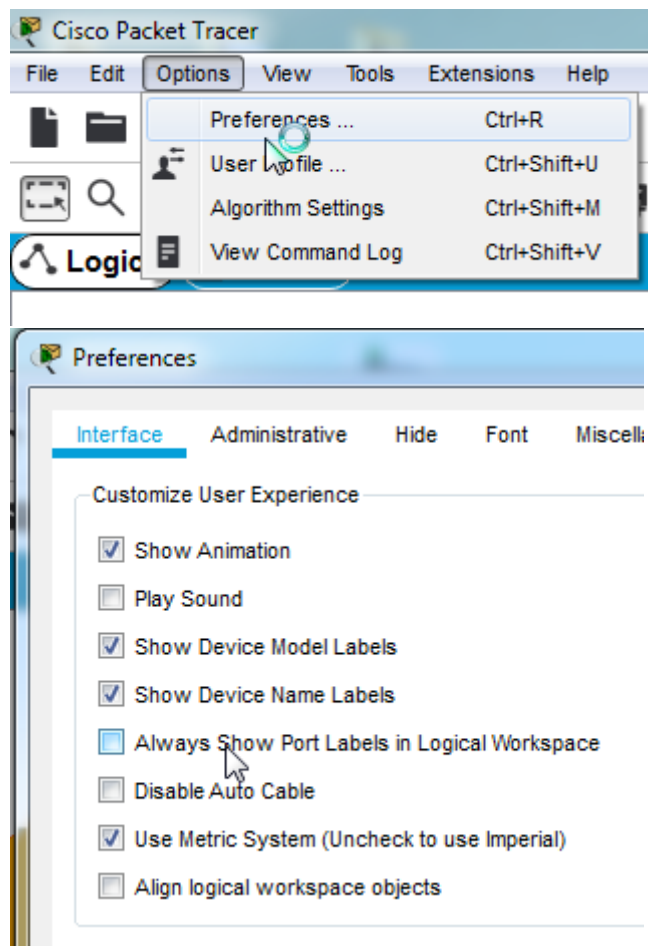
В общем, вот что у нас в итоге получилось:



Я больше не буду утруждать вас столь подробными объяснениями. Правую часть сетки настройте сами. Если что, ниже привожу свои настройки всех сетей топологии. Займёмся после всего этого динамической маршрутизацией! Наконец-то! Ура!

Кстати.

Вот как можно спрятать или вернуть порты.



Штука полезная. Вернёмся к топологии. В правой части нужно всем роутерам добавить модуль с FastEthernet. А на нижний роутер даже с двумя портами. Он называется NM-2FE2W. Не забудьте сохранить настройки верхнего, [copy run st] (Команды можно сокращать, если что. Я тут уже неоднократно этим пользовался). Если что-то забыли, листайте вверх.

Если вы дали айпишник не тому интерфейсу, зайдите на него и введите [no ip addr]. Слово «no» вообще позволяет отменить действие любой команды, главное указать точно, какой.

Для проверки пользуйтесь [sh run] (show running-config)

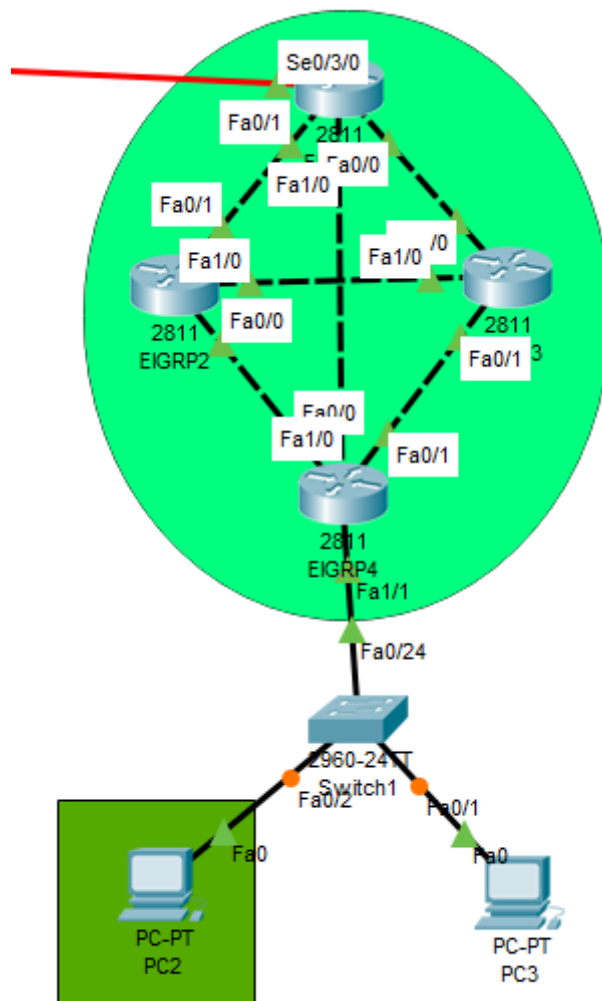
Некоторые скриншоты с места событий, чтобы вы если что сверились:

```
EIGRP1(config)#int fa 1/0
EIGRP1(config-if)#ip addr 221.0.2.1 255.255.255.0
EIGRP1(config-if)#no sh
EIGRP1(config-if)#int fa 0/0
EIGRP1(config-if)#ip addr 221.0.3.1 255.255.255.0
EIGRP1(config-if)#no sh
EIGRP4(config-if)#int fa 1/1.1
EIGRP4(config-subif)#enc dot1q 1
EIGRP4(config-subif)#ip addr 192.168.3.1 255.255.255.0
EIGRP4(config-subif)#int fa 1/1.2
EIGRP4(config-subif)#enc dot1q 2
EIGRP4(config-subif)#ip addr 192.168.4.1 255.255.255.0
EIGRP4(config-subif)#int fa 1/1
EIGRP4(config-if)#no sh
```

```
Switch(config)#int fa 0/24
Switch(config-if)#switchport mode trunk
```

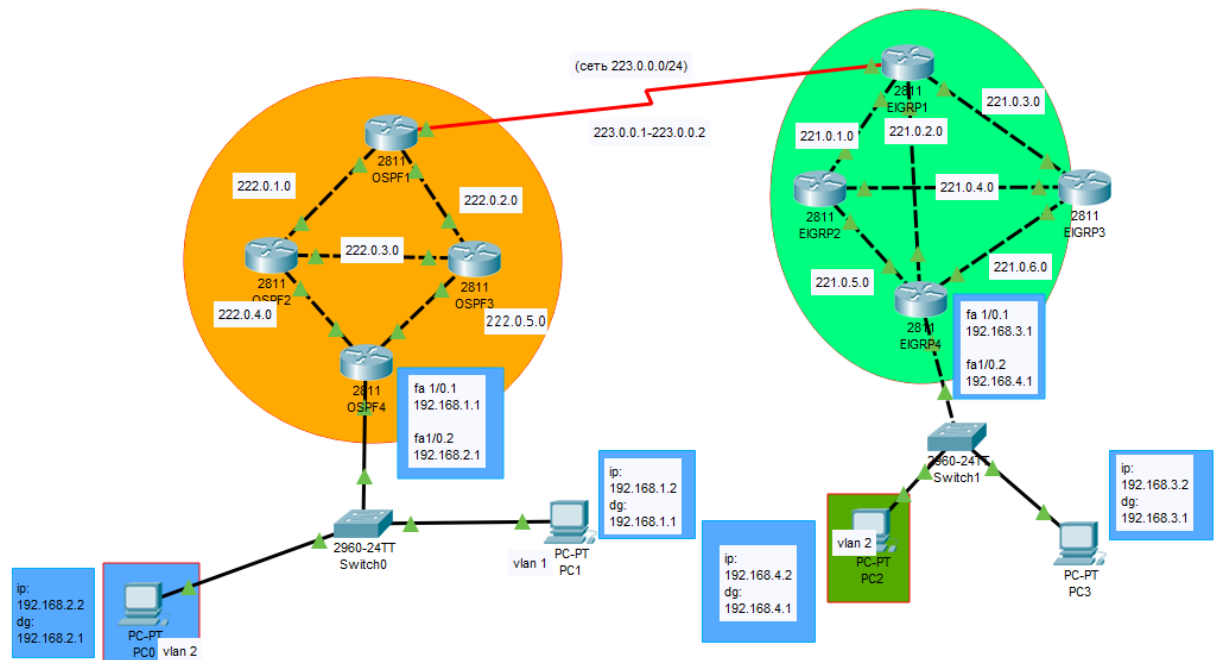
```
Switch(config-if)#int fa 0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
% Access VLAN does not exist. Creating vlan 2
Switch(config-if)#int fa 0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 1
Switch(config-if)#exit
```

```
Switch(config)#vlan 2
Switch(config-vlan)#name net2
```



Советую соединять между собой одинаково называемые порты, т.к. это очень удобно. Чтобы когда они накладываются, понять, где какой, просто смотришь на порт соседа.

Вот что в итоге получилось:



Настройка OSPF:

Вот что в принципе нужно усвоить из теории, чтобы сделать лабу:

- Что такое метрика и чем она отличается от административного расстояния.

Метрика – расстояние от точки сети, в которой в данный момент находится пакет, до сети назначения. Вычисляется метрика на основе различных параметров (число промежуточных маршрутизаторов, пропускная способность, надежность канала, задержка и т.п.) в зависимости от используемого протокола маршрутизации.

Административное расстояние (Administrative Distance, AD) – целое число от 0 до 255, показывающее приоритет источника маршрута. Маршрут с более низким AD считается более надежным. Таким образом, административное расстояние – степень доверия к источнику, предоставившему информацию о маршруте. Примером ситуации, в которой административное расстояние играет существенную роль, является ситуация, когда информация об одной и той же сети назначения предоставляется двумя и более протоколами маршрутизации (будет выбран маршрут, предоставленный протоколом с наименьшим административным расстоянием)

- Простая математическая дробь:

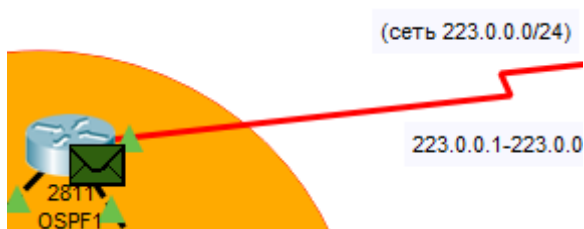
$$\text{Cost} = 10^8 / \text{link bandwidth}$$

Из которой видно, что чем выше пропускная способность (bandwidth), тем ниже метрика. (OSPF)

- И как это дело настраивать

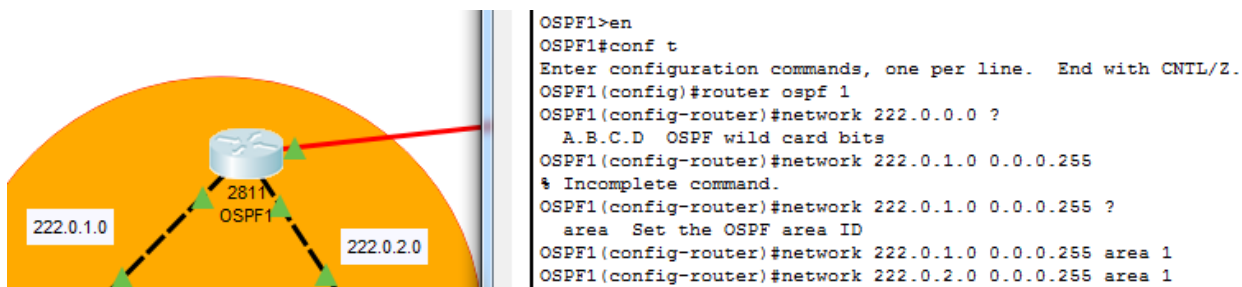


Всё! Вперёд и с песней, проходимся по всем роутерам и запускаем процесс ospf для наших сеточек между роутерами.



У верхнего роутера не вносим в процесс сеть 223.0.0.0, так как по заданию у нас там статическая маршрутизация.

Первая настройка:



Самое время объяснить, что такое wildcard маска.

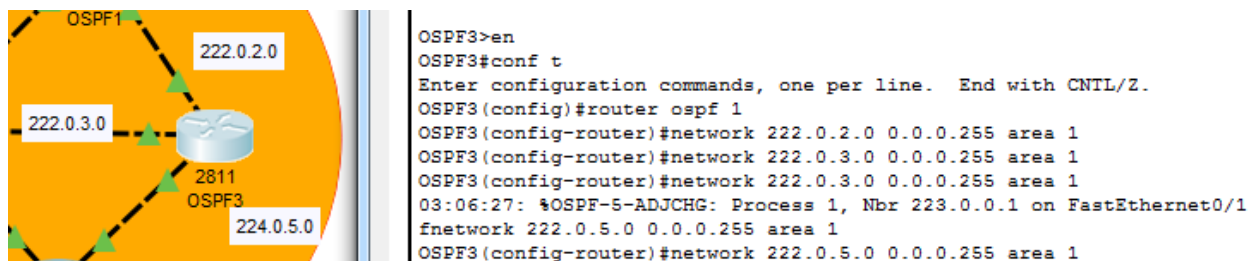
В видео курса дана обтекаемая формулировка, что в ней могут даже чередоваться нули и единицы. Почему – читайте тут: <https://habr.com/ru/post/131712/>

Мы же будем ~~стрелять из пушки по воробьям~~ пользоваться ей как обычной маской.

Была у нас маска 255.255.255.0, обратная ей 0.0.0.255.

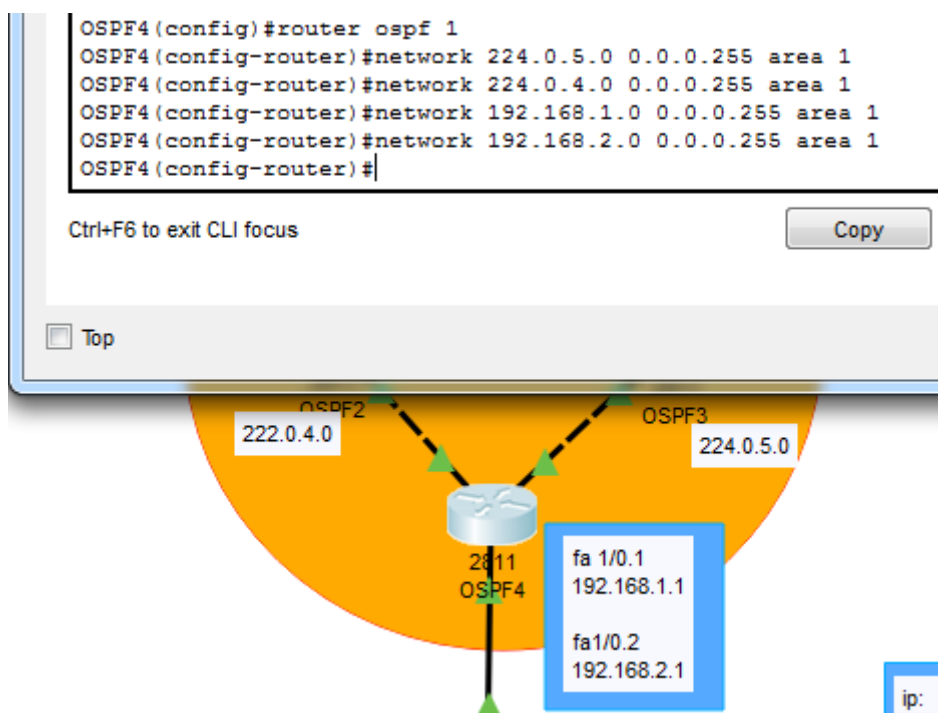
И в OSPF при настройке как раз используется обратная.

Продолжим настройку:

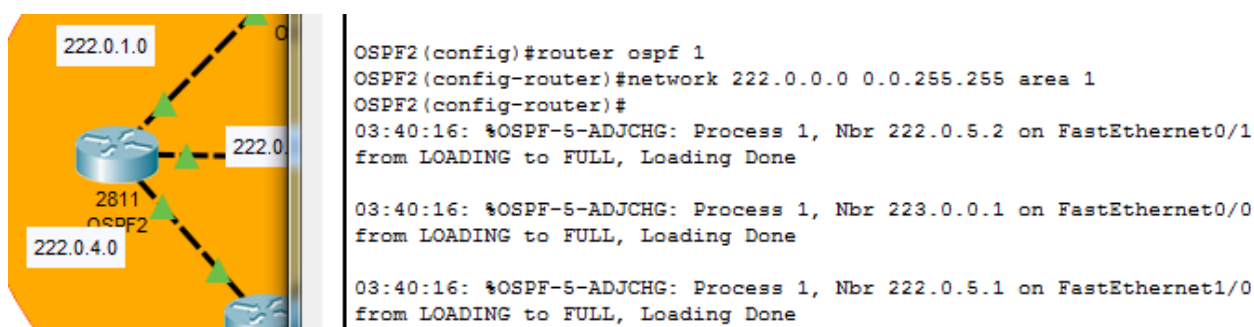


Здесь у меня не совсем прямые руки, разумеется, достаточно один раз прописать нужную команду.

На нижнем роутере обязательно добавляем в процесс OSPF нижние сеточки, идущие до компьютеров. Таким образом все роутеры выше нижнего узнают об этих сеточках и построят маршруты в них. (Ошибка: вместо 224 в сети используйте 222).



Давайте на последнем оставшемся роутере немного поэкспериментируем. Вообще-то мы могли сразу писать вот так:



То есть такая запись wildcard означает, что 2 октета справа могут меняться как они хотят.

Благодаря этому мы запустили процесс на всех сеточках, которые начинаются с 222.0 и смогли достучаться до роутеров-соседей. Подробнее про тот вывод в командной строке, который мы наблюдаем, да и вообще про OSPF смотрите в «Сетях для самых маленьких»:

<https://www.youtube.com/watch?v=8zxdW6ag8Us&t=5s>

OSPF настроен, можете попробовать попинговать роутеры в OSPF с компьютеров. Только не забудьте перейти в Realtime из симуляции :D Многие люди забывали это и не сдавали утп.

Доказано на опыте, что если сразу всё настроить правильно, всё заработает. Это утп приходилось переделывать не раз и не два, но поверьте, оно того стоит.

Если всё ещё ничего не работает, посмотрите как сделано в «Сетях для самых маленьких».

Loopback-интерфейсы и Router-ID действительно важны, я и сам этого не понимал, пока у меня при написании этих строк внезапно не отказался работать OSPF :D

Если что, вас всегда спасёт `[no router ospf 1]`. Дальше просто пишете всё заново.

В `[show ip route]` должны быть маршруты от OSPF, обозначенные буквой O. Если вы наблюдаете только маршруты C (connected), настройте OSPF заново.

Настройка EIGRP:

- В теории EIGRP достаточно сложен и многогранен. Но на практике в этой лабе мы должны знать, что метрика будет высчитываться как:

$$Metric = B + D$$

Где:

- пропускная способность (B)
- задержка (D)

То есть просто суммируются два показателя, которые вы можете посмотреть для любого интерфейса командой `[show interface]`

```
OSPF1#sh int se0/3/0
Serial0/3/0 is up, line protocol is up (connected)
Hardware is HD64570
Internet address is 223.0.0.1/24
MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
```

BW – пропускная способность, DLY – задержка.

- Настраиваться будет так:

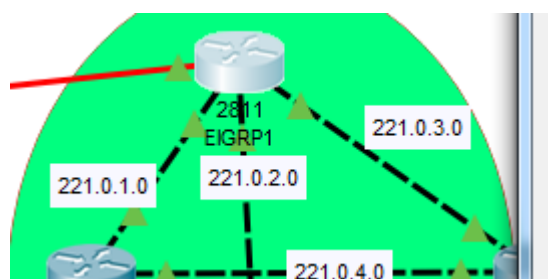
EIGRP

конфигурирование

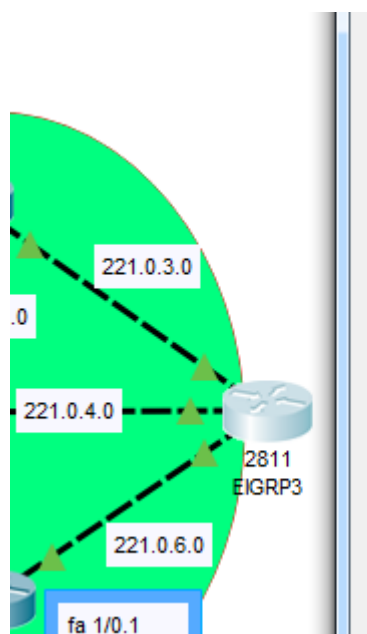
R1(config-router)#network <network> [wildcard mask]

Здесь уже можно указывать wildcard, можно не указывать.

Поехали!



```
R1(config)#
EIGRP1(config)#
EIGRP1(config)#
EIGRP1(config)#
EIGRP1(config)#
EIGRP1(config)#router eigrp 1
EIGRP1(config-router)#network 221.0.1.0
EIGRP1(config-router)#network 221.0.3.0
EIGRP1(config-router)#no auto
EIGRP1(config-router)#network 221.0.2.0
```

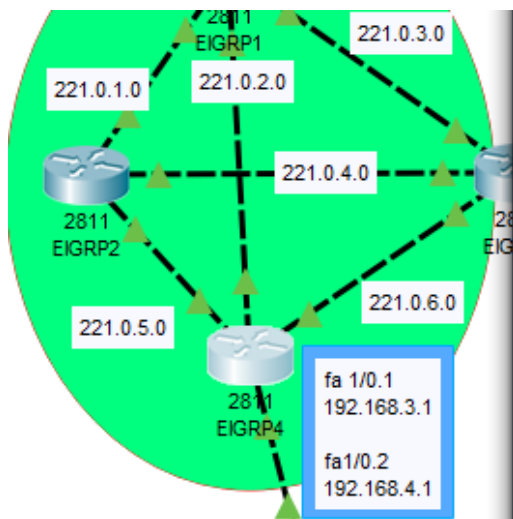


```
EIGRP3>en
EIGRP3#conf t
Enter configuration commands, one per line. End w
EIGRP3(config)#router eigrp 1
EIGRP3(config-router)#network 221.0.3.0
EIGRP3(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 221.0.3.1
is up: new adjacency

EIGRP3(config-router)#network 221.0.4.0
EIGRP3(config-router)#network 221.0.6.0
EIGRP3(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 221.0.6.2
is up: new adjacency

EIGRP3(config-router)#no auto
EIGRP3(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 221.0.3.1
resync: summary configured

%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 221.0.6.2
resync: summary configured
```

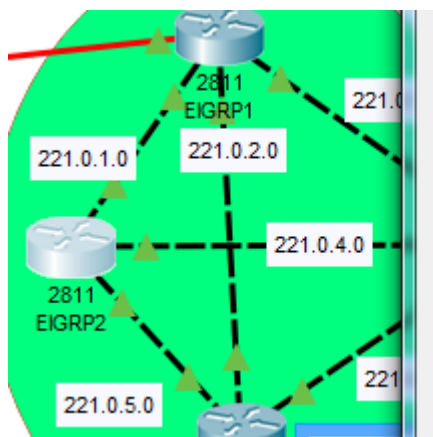


```
EIGRP4(config-router)#router eigrp 1
EIGRP4(config-router)#network 221.0.5.0
EIGRP4(config-router)#network 221.0.6.0
EIGRP4(config-router)#network 221.0.2.0
EIGRP4(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 221.0.2.1
is up: new adjacency

EIGRP4(config-router)#no auto
EIGRP4(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 221.0.6.1
resync: summary configured

%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 221.0.2.1
resync: summary configured

EIGRP4(config-router)#network 192.168.3.1
EIGRP4(config-router)#network 192.168.4.1
EIGRP4(config-router)#
```



```
EIGRP2(config)#router eigrp 1
EIGRP2(config-router)#network 221.0.1.0
EIGRP2(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 221.0.1.1
is up: new adjacency

EIGRP2(config-router)#network 221.0.5.0
EIGRP2(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 221.0.5.2
is up: new adjacency

EIGRP2(config-router)#network 221.0.4.0
EIGRP2(config-router)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 221.0.4.2
is up: new adjacency
```

```
EIGRP2(config)#router eigrp 1
EIGRP2(config-router)#no auto
```

Обязательное уточнение:

Когда настраиваете EIGRP, на всех пишете [no auto-summary].

Почему – хорошо написано в «Сетях для самых маленьких»

<https://linkmeup.ru/blog/33.html> (ctrl+f автоматическое суммирование)

Вот в общем-то и всё. Теперь можно попинговать какой-нибудь роутер с компов.

Если будут проблемы, советы те же –

[sh ip route] – должны быть маршруты с буквой D

[no router eigrp 1] – удалить процесс маршрутизации, чтобы запустить заново.

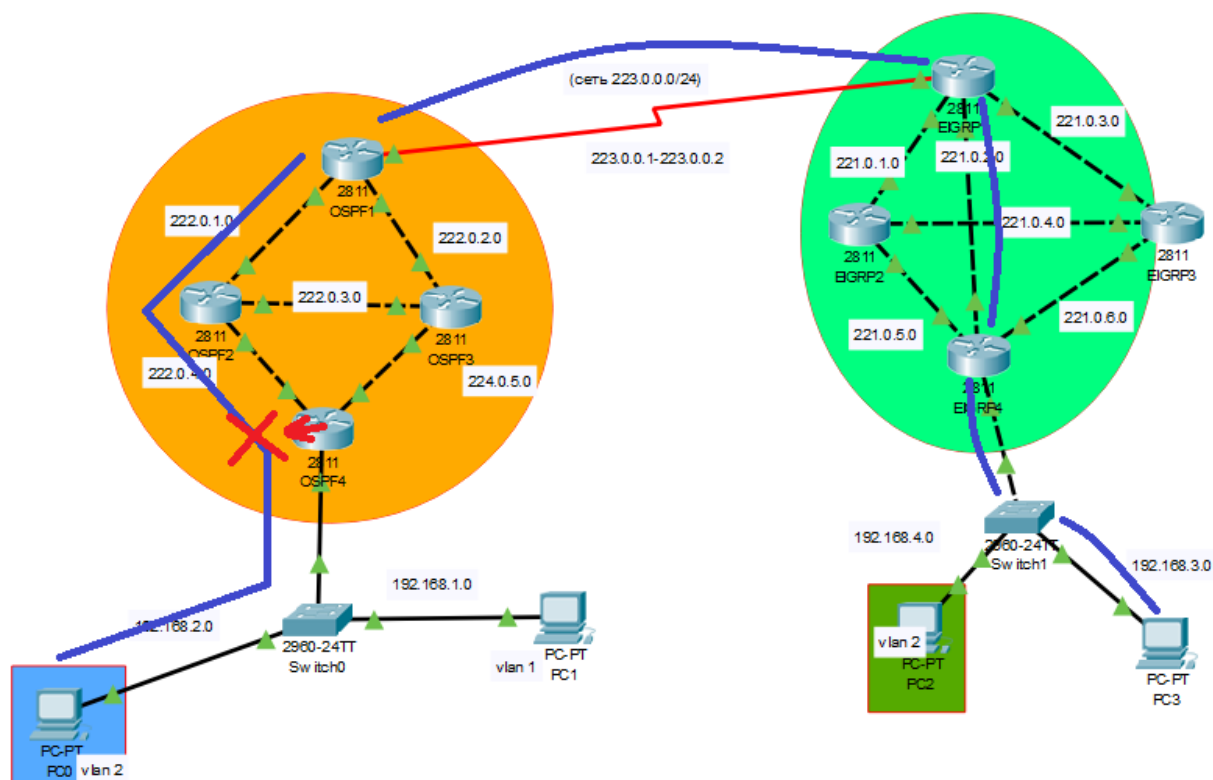
Кстати, мельком заметил тут отличную команду на [http://xgu.ru/wiki/OSPF в Cisco](http://xgu.ru/wiki/OSPF_в_Cisco)

Перезапустить процесс OSPF можно командой:

```
dyn3# clear ip ospf process
```

Настройка статической маршрутизации между EIGRP и OSPF:

Представим ситуацию. Пакетику нужно пройти с левого нижнего компьютера до правого нижнего.



После красного крестика пакетик никуда не пойдёт, потому что роутер OSPF4 понятия не имеет о том, где расположена сеть 192.168.3.0 и кому дальше передавать пакетик для её достижения.

При этом в правой половине всем роутерам EIGRP прекрасно известно, что такое сеть 192.168.3.0

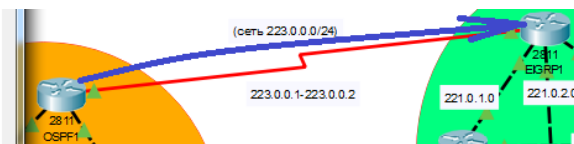
Эта проблема решится, если и в левой половине все роутеры узнают об этой сети.

- 1) Добавить маршрут на левом верхнем роутере OSPF1
- 2) Сделать так, чтобы он рассказал о нём всем роутерам OSPF

Сделать то же самое для сети 192.168.4.0

1) Добавление маршрута

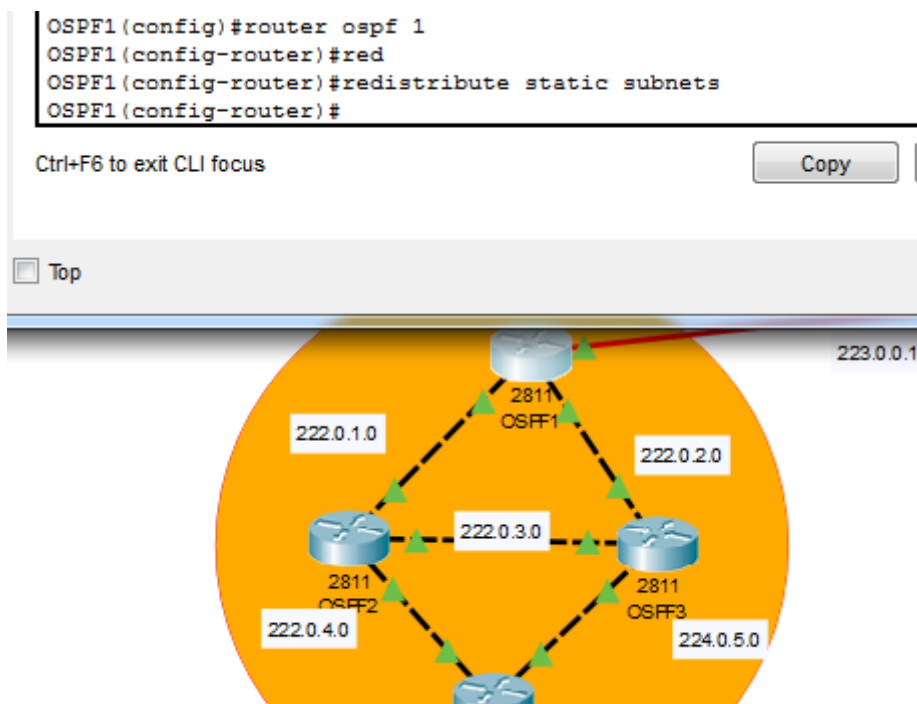
```
OSPF1>en
OSPF1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
OSPF1(config)#ip route 192.168.3.0 255.255.255.0 223.0.0.2
OSPF1(config)#ip route 192.168.4.0 255.255.255.0 223.0.0.2
OSPF1(config)#
```



223.0.0.2 – так называемый ip next hop – айпи адрес интерфейса se0/3/0 роутера EIGRP1, которому нужно передать пакеты, идущие в сети 192.168.3.0 и 4.0.

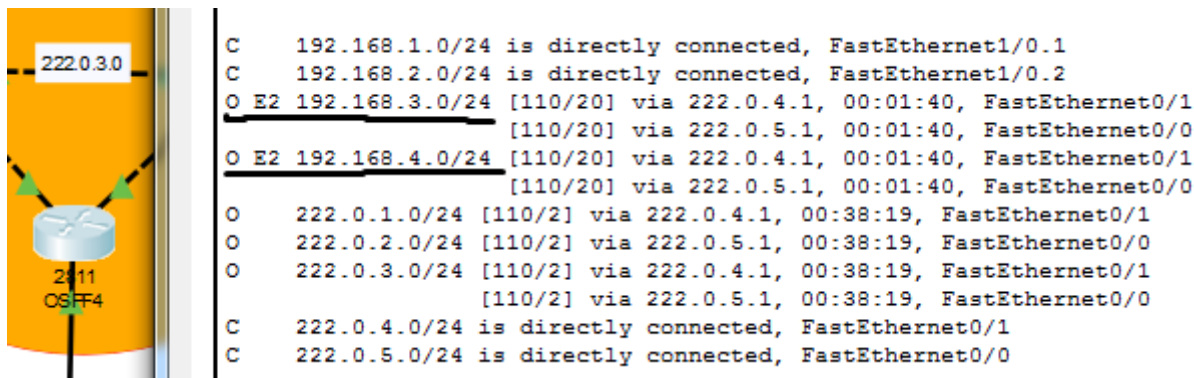
255.255.255.0 – маска этих сетей

2) Рассказать всем о своих маршрутах



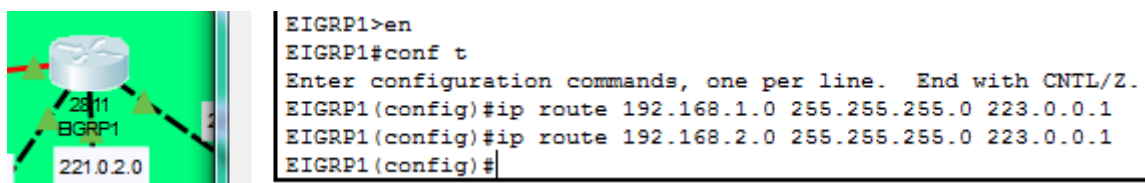
Готово, благодаря этой команде наш OSPF1 передаст информацию о маршрутах другим роутерам OSPF.

Вот что теперь можно увидеть, например, на OSPF4 [sh ip route]:

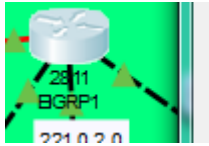


Как вы наверное уже догадались, нужно то же самое настроить для EIGRP.

3) EIGRP. Добавление маршрута



4) EIGRP. Рассказать всем



```
EIGRP1(config)#  
EIGRP1(config)#router eigrp 1  
EIGRP1(config-router)#redistribute static  
EIGRP1(config-router)#
```

Как видите, команды немного отличаются.

В EIGRP: [redistribute static]

В OSPF: [redistribute static subnets]

Это происходит из-за того, что OSPF – классовый протокол, а EIGRP – бесклассовый. В EIGRP мы по умолчанию настраиваем передачу именно подсетей, а в OSPF требуется уточнение, иначе он будет передавать сети по правилам для классов IPv4.

Подробнее про OSPF и EIGRP: <https://linkmeup.ru/blog/33.html>

На этом про настройку этой топологии у меня всё.

Дополнительные вопросы и задания:

1) Балансировка трафика – EIGRP

Если иное не было настроено, метрика маршрута в EIGRP рассчитывается по формуле:

$$\text{Metric} = (B+D) * 256$$

Где B – Bandwidth, пропускная способность, рассчитывается по формуле:

$$\text{Bandwidth} = 10^7 /$$

(минимальная_пропускная_способность_по_дороге_маршрута_в_килобитах)

D – Delay, сумма задержек интерфейсов, из которых *выходит* пакет по дороге маршрута **в десятках микросекунд** (команда [show interface se0/3/0] выводит значение в микросекундах).

Для примера далее рассматривается маршрут от роутера EIGRP4 до сети 223.0.0.0/24 (сеть между роутерами OSPF1 и EIGRP1). Соответствующего маршрута нет в таблице EIGRP4, поскольку на EIGRP1 не включена редистрибуция напрямую подключенных сетей. Последовательность команд для её включения:

```
EIGRP1(config)#router eigrp 1
```

```
EIGRP1(config-router)#redistribute connected metric ?  
<1-4294967295> Bandwidth metric in Kbits per second
```

```
EIGRP1(config-router)#redistribute connected metric 128 ?  
<0-4294967295> EIGRP delay metric, in 10 microsecond units
```

```
EIGRP1(config-router)#redistribute connected metric 128 2000 ?
<0-255> EIGRP reliability metric where 255 is 100% reliable

EIGRP1(config-router)#redistribute connected metric 128 2000 255 ?
<1-255> EIGRP Effective bandwidth metric (Loading) where 255 is 100% loaded

EIGRP1(config-router)#redistribute connected metric 128 2000 255 1 ?
<1-65535> EIGRP MTU of the path

EIGRP1(config-router)#redistribute connected metric 128 2000 255 1 1500
```

Значения пропускной способности, задержки, надежности, загрузки и MTU взяты стандартные.

На EIGRP4 [show ip route] показывает один маршрут до сети 223.0.0.0:

```
D EX 223.0.0.0/24 [170/20514560] via 221.0.2.1, 00:00:10, FastEthernet1/0
```

С метрикой 20514650. И в таблице топологий (``show ip eigrp topology``) этот маршрут тоже только один:

```
P 223.0.0.0/24, 1 successors, FD is 20514560 via Rstatic (20514560/20512000)
```

Так происходит потому, что возможные пути через EIGRP2 и EIGRP3 имеют AD не лучше, чем FD через EIGRP1 у EIGRP4. Подробнее: <https://linkmeup.ru/blog/1195/>

AD – Advertised Distance – Для роутера это метрика маршрута до нужной сети от роутера-соседа, который прислал информацию об этом маршруте

FD – Feasible Distance – Для роутера это метрика маршрута до нужной сети от него самого

Однако очевидно, что существует ещё два возможных пути, через EIGRP2 и EIGRP3, которые могут быть использованы. Задержка на пути через них будет ровно на 100 мкс больше задержки пути через EIGRP1 – поскольку в этих путях ровно на один интерфейс больше.

Таким образом, чтобы в таблице маршрутизации появились эти пути, нужно установить значение задержки на интерфейсе fa1/0 EIGRP4 равное 200 мкс. Последовательность команд (значение указывается в десятках микросекунд):

```
Router(config)#int fa 1/0
Router(config-if)#delay 20
```

Теперь команда [show ip route] показывает три равноценных маршрута до сети 223.0.0.0/24 и пакеты будут идти по трем разным путям.

```
D EX 223.0.0.0/24 [170/20517120] via 221.0.6.1, 00:00:46,
FastEthernet0/1
[170/20517120] via 221.0.5.1, 00:00:46, FastEthernet0/0
[170/20517120] via 221.0.2.1, 00:00:44, FastEthernet1/0
```