

# статическая и динамическая на примере RIP, OSPF и EIGRP / Хабр

 [habr.com/ru/articles/335090](https://habr.com/ru/articles/335090)

Денис

21 октября 2019 г.



Всем привет! Спустя продолжительное время возвращаемся к циклу статей. Долгое время мы разбирали мир коммутации и узнали о нем много интересного. Теперь пришло время подняться чуть повыше и взглянуть на сторону маршрутизации. В данной статье поговорим о том, зачем нужна маршрутизация, разберем отличие статической от динамической маршрутизации, виды протоколов и их отличие. Тема очень интересная, поэтому приглашаю всех-всех к прочтению.

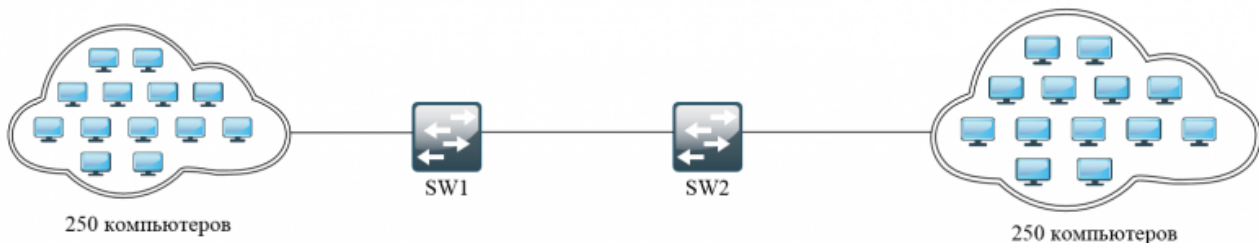
## Содержание

В предыдущих статьях мы разбирали отличия сетевых устройств. А именно, чем коммутатор отличается от маршрутизатора (можно почитать [здесь](#) и [здесь](#)). То есть коммутатор в классическом понимании — это устройство, которое получает Ethernet-кадры на одном интерфейсе и передает эти кадры на другие интерфейсы, базируясь на заголовках и своей таблицы коммутации. Работает коммутатор канальном уровне.

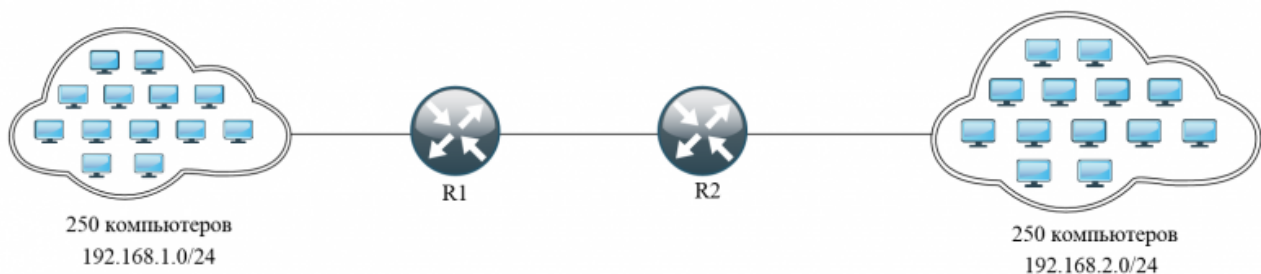
Маршрутизаторы работают аналогично. Только оперируют IP-пакетами. И работают на сетевом уровне. Хочу заметить, что есть коммутаторы и маршрутизаторы,

которые работают и на более высоких уровнях, но мы сейчас говорим о классических устройствах.

Встает вопрос. Почему мы не можем просто коммутировать весь трафик? И зачем требуются IP-адреса и маршрутизация. Ведь что MAC-адреса, что IP-адреса уникальны у каждого сетевого устройства (ПК, телефон, сервер и т.д.). Сейчас ответу более развернуто.

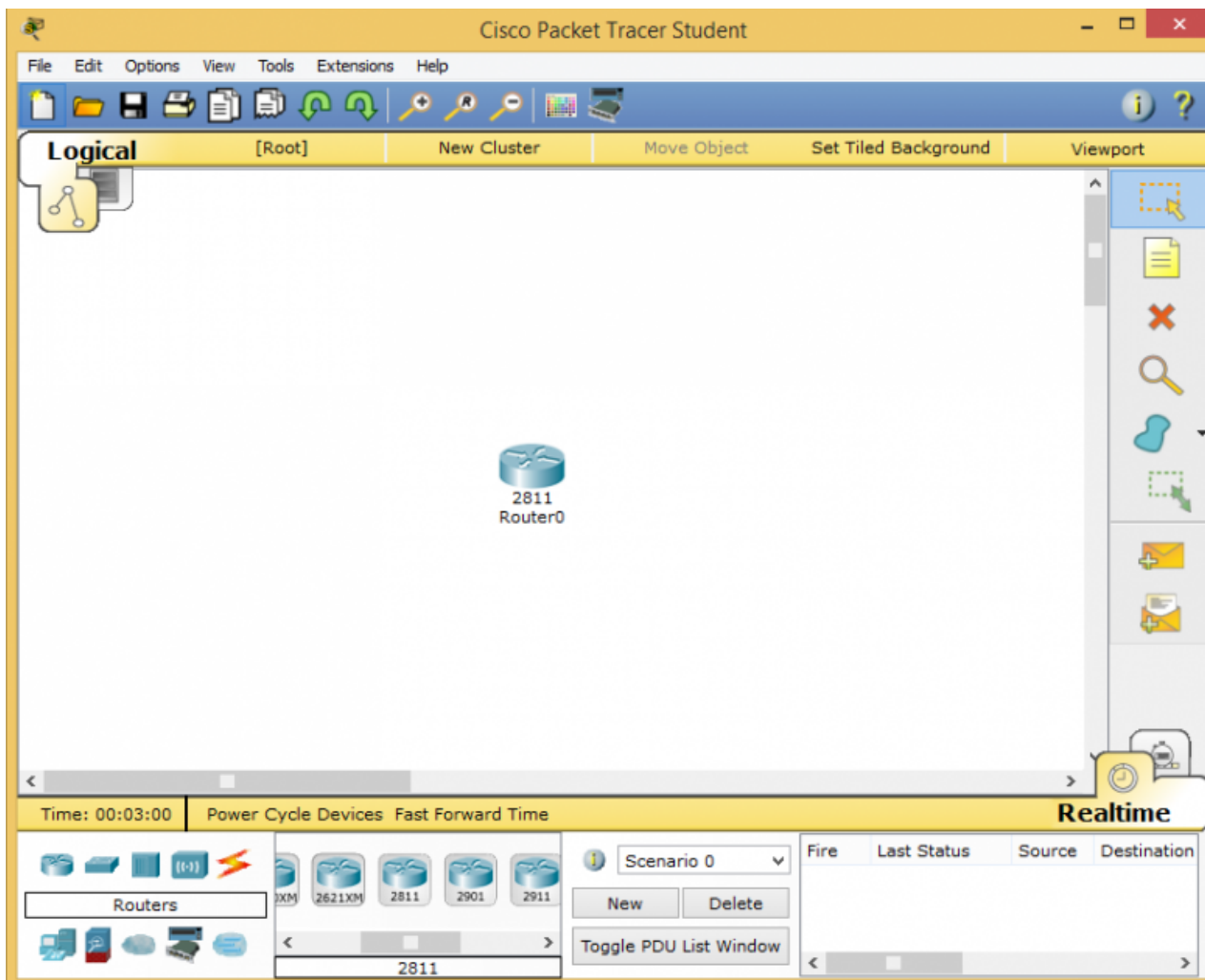


На рисунке представлены 2 коммутатора, к которым подключено по 250 пользователей. Соответственно, чтобы обеспечить связность между всеми участниками, коммутаторы должны знать MAC-адреса всех участников сети. То есть таблица каждого коммутатора будет содержать 500 записей. Это уже не мало. А если представить, что таким образом будет работать Интернет, в котором миллиарды устройств? Следовательно нужно искать выход. Проблема коммутации заключается в том, что она плохо масштабируется. И тяжело соблюдать иерархию. Теперь посмотрим на эту ситуацию с точки зрения маршрутизации.



Здесь вводится понятие IP-адресации. Слева сеть 192.168.1.0/24 соединенная с левым маршрутизатором (R1), а справа сеть 192.168.2.0/24 соединенная с правым маршрутизатором (R2), соответственно. R1 знает, что добраться до сети 192.168.2.0 можно через соседа R2 и наоборот R2 знает, что добраться до сети 192.168.1.0 можно через соседа R1. Тем самым 500 записей в таблице коммутации заменяются одной в таблице маршрутизации. Во-первых это удобно, а во-вторых экономит ресурсы. Вдобавок к этому, можно соблюдать иерархичность, при построении. Теперь поговорим о том, как таблица маршрутизации заполняется. Как только маршрутизатор включается «с коробки», он создает таблицу маршрутизации. Но самостоятельно он туда может записать только информацию о сетях, с которыми он связан напрямую (connected).

Покажу на примере в CPT:

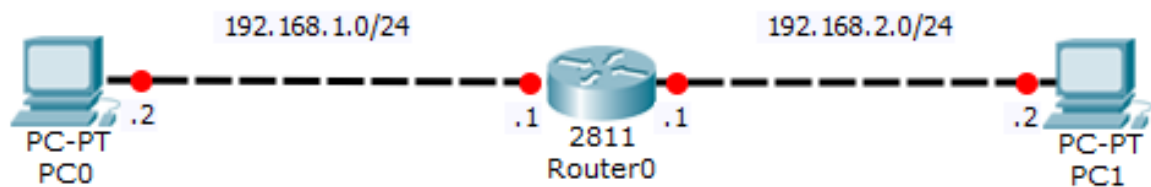


Добавляю маршрутизатор с пустой конфигурацией. Дожидаюсь загрузки и смотрю таблицу маршрутизации:

```
Router#showiproute
Codes:C-connected, S-static, I-IGRP, R-RIP, M-mobile, B-BGP
D-EIGRP, EX-EIGRPexternal, O-OSPF, IA-OSPFinterarea
N1-OSPFNSSAexternaltype1, N2-OSPFNSSAexternaltype2
E1-OSPFexternaltype1, E2-OSPFexternaltype2, E-EGP
i-IS-IS, L1-IS-ISlevel-1, L2-IS-ISlevel-2, ia-IS-ISinterarea
*-candidatedefault, U-per-userstaticroute, o-ODR
P-periodicdownloadedstaticroute
```

```
Gatewayoflastresortisnotset
```

Сейчас таблица есть, но она пустая из-за того, что не подключен ни один из интерфейсов и не заданы IP-адреса. Соберем схему.



Зададим IP-адреса на интерфейсах маршрутизатора:

```
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastEthernet0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

```
Router(config-if)#exit
Router(config)#interface fastEthernet0/1
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown
```

```
Router(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(config-if)#end
```

И посмотрим, что изменилось в таблице маршрутизации:

```
Router#showiproute
Codes:C-connected, S-static, I-IGRP, R-RIP, M-mobile, B-BGP
D-EIGRP, EX-EIGRPexternal, O-OSPF, IA-OSPFinterarea
N1-OSPFNSSAexternaltype1, N2-OSPFNSSAexternaltype2
E1-OSPFexternaltype1, E2-OSPFexternaltype2, E-EGP
i-IS-IS, L1-IS-ISlevel-1, L2-IS-ISlevel-2, ia-IS-ISinterarea
*-candidatedefault, U-per-userstaticroute, o-ODR
P-periodicdownloadedstaticroute
```

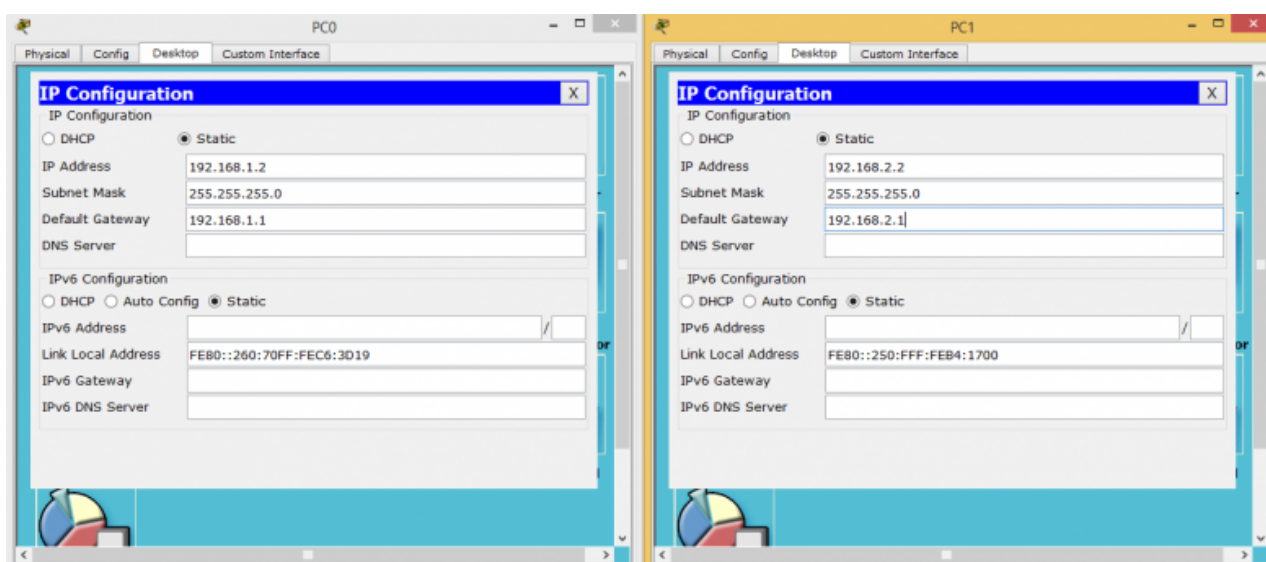
Gatewayoflastresortisnotset

```

C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/1
```

В таблице появились 2 записи. Маршрутизатор автоматически добавил подсети, в которых находятся его интерфейсы. Сверху есть коды, показывающие каким образом маршрут был добавлен.

Настроим обе рабочие станции и проверим связность:



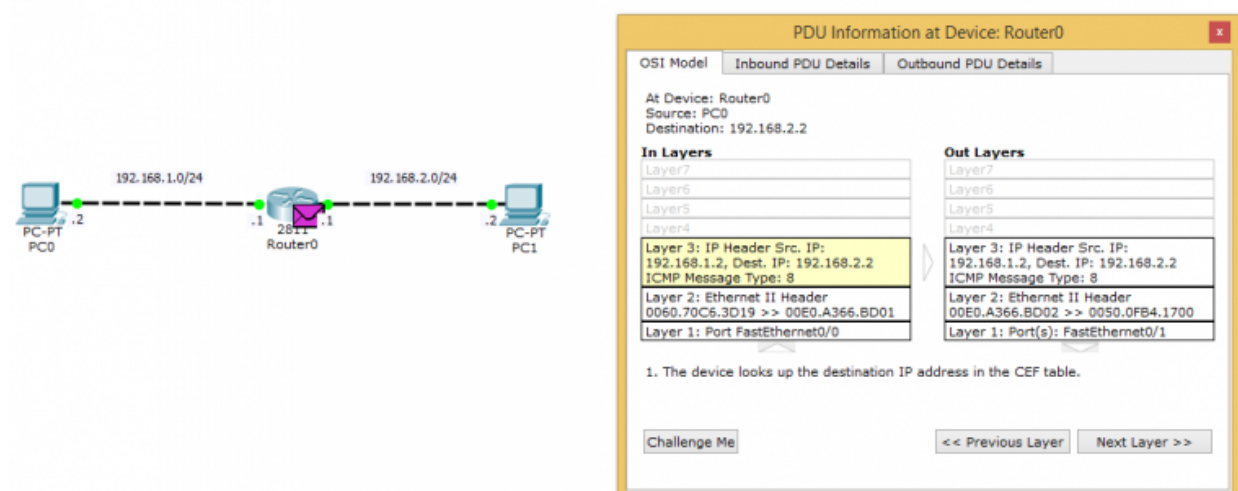
```
PacketTracerPCCommandLine1.0
PC>ping192.168.2.2
```

Pinging192.168.2.2with32bytesofdata:

```
Replyfrom192.168.2.2:bytes=32time=0msTTL=127
Replyfrom192.168.2.2:bytes=32time=0msTTL=127
Replyfrom192.168.2.2:bytes=32time=0msTTL=127
Replyfrom192.168.2.2:bytes=32time=1msTTL=127
```

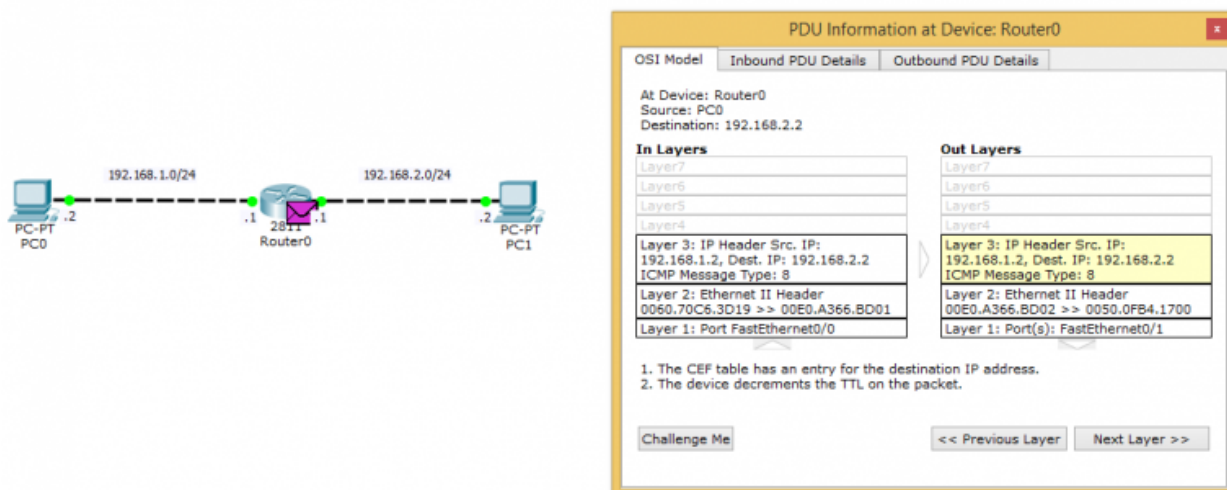
Ping statistics for 192.168.2.2:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 1ms, Average = 0ms

Теперь детально рассмотрим, что происходит с пакетом, когда он попадает на маршрутизатор.



Пакет приходит. Маршрутизатор сразу читает IP-адрес назначения в заголовке и сверяет его со своей таблицей.





Находит совпадение, изменяет TTL и отправляет на нужный интерфейс. Соответственно, когда ответный пакет придет от PC1, он проделает аналогичную операцию.

То есть отличие в том, что маршрутизатор принимает решение исходя из своей таблицы маршрутизации, а коммутатор из таблицы коммутации. Единственное, что важно запомнить: и у коммутатора, и у маршрутизатора есть ARP-таблица.

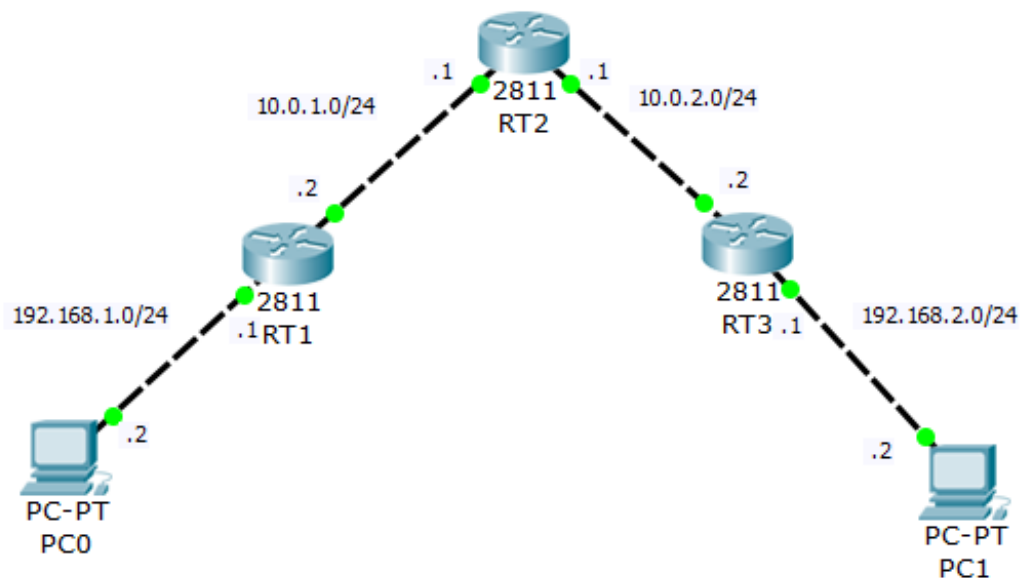
Несмотря на то, что маршрутизатор работает с 3 уровнем по модели OSI и читает заголовки IP-пакетов, он не может игнорировать работу стека и обязан работать на канальном и физическом уровне. В свою ARP-таблицу он записывает соотношения MAC-адреса к IP-адресу и с какого интерфейса к нему можно добраться. Причем ARP-таблица у каждого сетевого устройства своя. Пишу команду **show arp** на маршрутизаторе:

```
Router#show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	192.168.1.1	-	0060.5C16.3B01	ARPA	FastEthernet0/0
Internet	192.168.1.2	6	00E0.F73D.E561	ARPA	FastEthernet0/0
Internet	192.168.2.1	-	0060.5C16.3B02	ARPA	FastEthernet0/1
Internet	192.168.2.2	7	0002.179D.455A	ARPA	FastEthernet0/1

Как только PC0 отправил ICMP до PC1 и пакет дошел до маршрутизатора, он увидел в заголовках IP-пакета адрес отправителя (PC0) и его MAC-адрес. Он добавляет его в ARP-таблицу. Следующее, что он видит — это IP-адрес получателя. Он не знает, куда отправлять пакет, так как в его ARP-таблице нет записи. Но видит, что адрес получателя из той же сети, что и один из его интерфейсов. Тогда он запускает ARP с этого интерфейса, чтобы получить MAC-адрес запрашиваемого хоста. Как только приходит ответ, он заносит информацию в ARP-таблицу. Это базовый пример того, как работает маршрутизация. Прикладываю [ссылку](#) на скачивание.

Усложним немного схему.



На ней представлены 2 рабочие станции и 3 маршрутизатора. Не буду заострять внимание на том, как прописать IP-адрес на интерфейс, а лишь покажу итоговую конфигурацию:

**RT1 (раскрыть)**

**RT2 (раскрыть)**

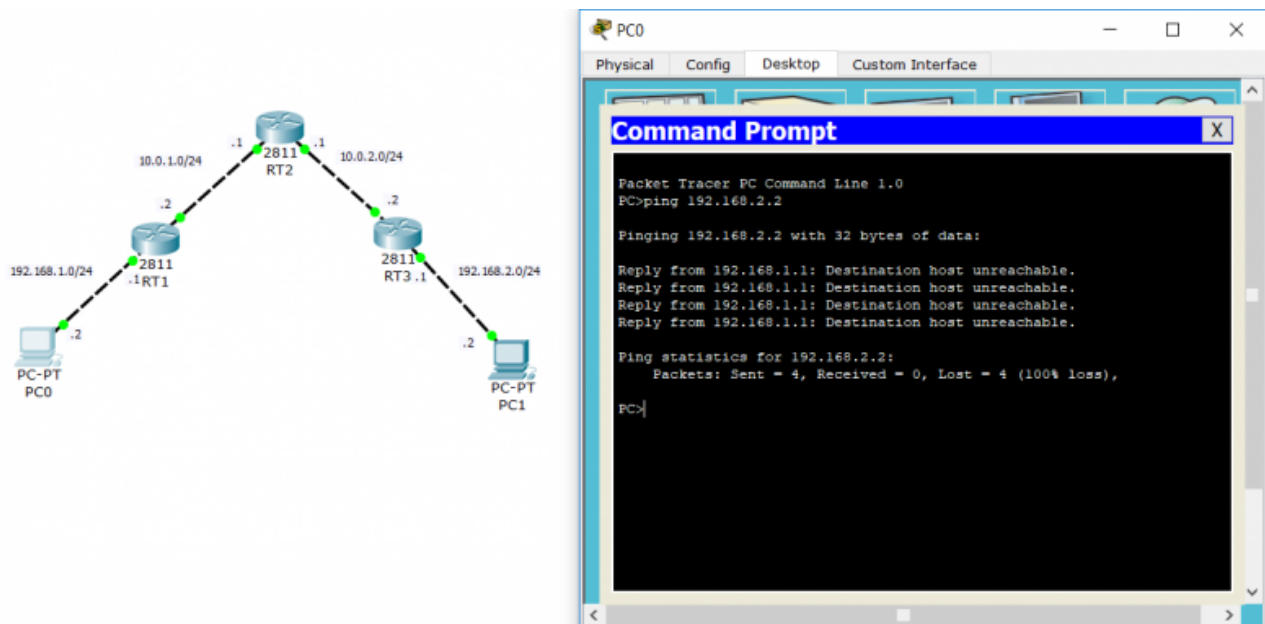
**RT3 (раскрыть)**

**PC0 (раскрыть)**

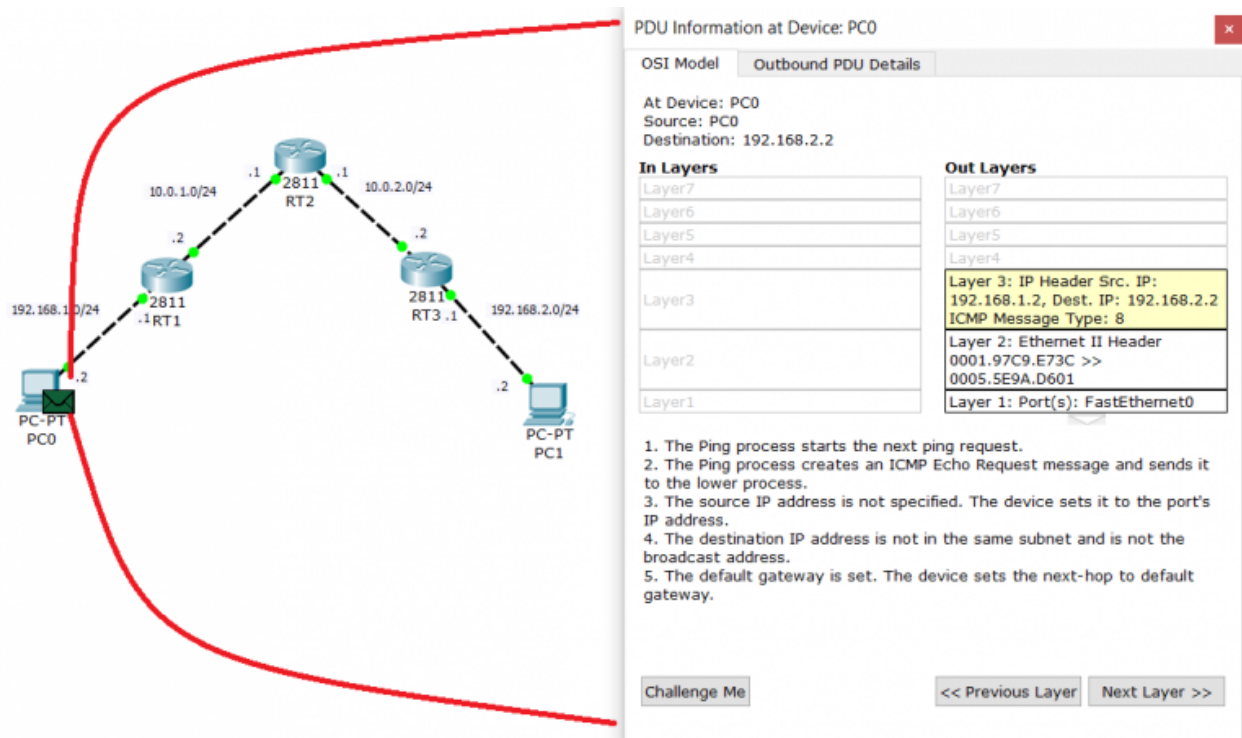
**PC1 (раскрыть)**

Все устройства сконфигурированы. Теперь проверим связность между PC0 и PC1:

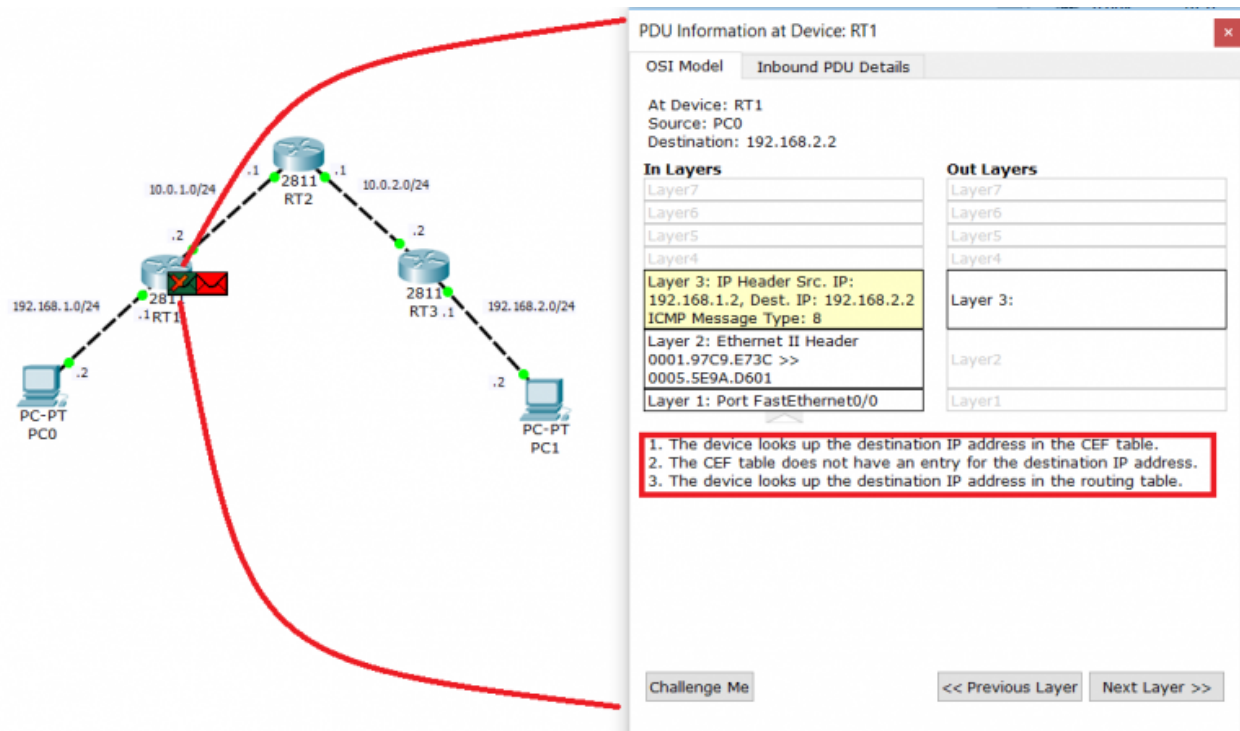




В консоли PC0 вылезает сообщение о недоступности узла. Но ведь все адреса прописаны и добраться можно. В чем же проблема? Переходим в режим симуляции и копаем глубже:



PC0 формирует ICMP-сообщение. Смотрит на IP-адрес назначения и понимает, что получатель находится в другой сети. Соответственно передать надо своему основному шлюзу, а дальше пускай сам разбирается.



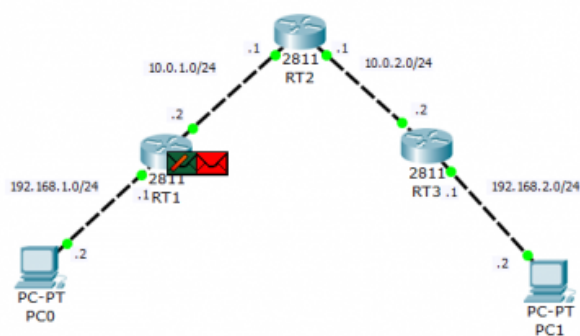
Пакет доходит до RT1. Смотрит в Destination IP и сравнивает со своей таблицей маршрутизации.

```
RT1#showiproute
Codes:C-connected, S-static, I-IGRP, R-RIP, M-mobile, B-BGP
D-EIGRP, EX-EIGRPexternal, O-OSPF, IA-OSPFinterarea
N1-OSPFNSSAexternaltype1, N2-OSPFNSSAexternaltype2
E1-OSPFexternaltype1, E2-OSPFexternaltype2, E-EGP
i-IS-IS, L1-IS-ISlevel-1, L2-IS-ISlevel-2, ia-IS-ISinterarea
*-candidatedefault, U-per-userstaticroute, o-ODR
P-periodicdownloadedstaticroute
```

```
Gatewayoflastresortisnotset
```

```
10.0.0.0/24 is subnetted, 1 subnets
C    10.0.1.0 is directly connected, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
```

И вуаля. Совпадений нет. А значит RT1 понятия не имеет, что делать с этим пакетом.



**PDU Information at Device: RT1**

OSI Model    Outbound PDU Details

At Device: RT1  
Source: RT1  
Destination: PC0

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer3	Layer3: IP Header Src. IP: 192.168.1.1, Dest. IP: 192.168.1.2 ICMP Message Type: 3 ICMP Message Type: 8
Layer2	Layer2: Ethernet II Header 0005.5E9A.D601 >> 0001.97C9.E73C
Layer1	Layer1: Port(s): FastEthernet0/0

- The device sends back an ICMP Host Unreachable message.
- The device looks up the destination IP address in the CEF table.
- The CEF table has an entry for the destination IP address.

Challenge Me    << Previous Layer    Next Layer >>

Но так просто отбросить его не может, так как надо уведомить того, кто это послал. Он формирует ответный ICMP с сообщением «Host Unreachable».

**PDU Information at Device: PC0**

OSI Model    Inbound PDU Details

At Device: PC0  
Source: RT1  
Destination: PC0

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer5
Layer3	Layer4
Layer2	Layer3
Layer1	Layer2

Layer 3: IP Header Src. IP: 192.168.1.1, Dest. IP: 192.168.1.2  
ICMP Message Type: 3 ICMP Message Type: 8  
Layer 2: Ethernet II Header 0005.5E9A.D601 >> 0001.97C9.E73C  
Layer 1: Port FastEthernet0

1. FastEthernet0 receives the frame.

**PC0**

Physical    Config    Desktop    Custom Interface

**Command Prompt**

```

Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.1.1: Destination host unreachable.
  
```

Как только пакет доходит до PC0, в консоли высвечивается сообщение «Reply from

192.168.1.1: Destination host unreachable.». То есть RT1 (192.168.1.1) говорит о том, что запрашиваемый хост недоступен.

Выход из ситуации следующий: нужно «сказать» сетевому устройству, как добраться до конкретной подсети. Причем это можно сделать вручную или настроить все сетевые устройства так, чтобы они переговаривались между собой. Вот на этом этапе маршрутизация делится на 2 категории:

- Статическая маршрутизация
- Динамическая маршрутизация

Начнем со статической. В качестве примера возьмем схему выше и добьемся связности между PC0 и PC1. Так как первые проблемы с маршрутизацией начались у RT1, то перейдем к его настройке:

```
RT1#conf t
RT1(config)#ip route 192.168.2.0 255.255.255.0 10.0.1.1
```

Маршрут прописывается командой **ip route**. Синтаксис прост: «*подсеть*» «*маска*» «*адрес следующего устройства*».

После можно набрать команду **show ip route** и посмотреть таблицу маршрутизации:

```
RT1#showiproute
Codes:C-connected, S-static, I-IGRP, R-RIP, M-mobile, B-BGP
D-EIGRP, EX-EIGRPexternal, O-OSPF, IA-OSPFinterarea
N1-OSPFNSSAexternaltype1, N2-OSPFNSSAexternaltype2
E1-OSPFexternaltype1, E2-OSPFexternaltype2, E-EGP
i-IS-IS, L1-IS-ISlevel-1, L2-IS-ISlevel-2, ia-IS-ISinterarea
*-candidatedefault, U-per-userstaticroute, o-ODR
P-periodicdownloadedstaticroute
```

```
Gatewayoflastresortisnotset
```

```
          10.0.0.0/24 is subnetted, 1 subnets
C          10.0.1.0 is directly connected, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S    192.168.2.0/24 [1/0] via 10.0.1.1
```

Появился статический маршрут (о чем свидетельствует код **S** слева). Здесь много различных параметров и о них я расскажу чуть позже. Сейчас задача прописать маршруты на всех устройствах. Перехожу к RT2:

```

RT2(config)#iproute192.168.1.0255.255.255.010.0.1.2
RT2(config)#iproute192.168.2.0255.255.255.010.0.2.2
RT2#showiproute
Codes:C-connected, S-static, I-IGRP, R-RIP, M-mobile, B-BGP
D-EIGRP, EX-EIGRPexternal, O-OSPF, IA-OSPFinterarea
N1-OSPFNSSAexternaltype1, N2-OSPFNSSAexternaltype2
E1-OSPFexternaltype1, E2-OSPFexternaltype2, E-EGP
i-IS-IS, L1-IS-ISlevel-1, L2-IS-ISlevel-2, ia-IS-ISinterarea
*-candidatedefault, U-per-userstaticroute, o-ODR
P-periodicdownloadedstaticroute

```

Gatewayoflastresortisnotset

```

          10.0.0.0/24 is subnetted, 2 subnets
C       10.0.1.0 is directly connected, FastEthernet0/0
C       10.0.2.0 is directly connected, FastEthernet0/1
S       192.168.1.0/24 [1/0] via 10.0.1.2
S       192.168.2.0/24 [1/0] via 10.0.2.2

```

Обратите внимание, что маршрут прописан не только в 192.168.2.0/24, но и 192.168.1.0/24. Без обратного маршрута полноценной связности не будет. Остался RT3:

```

RT3(config)#iproute192.168.1.0255.255.255.010.0.2.1
RT3(config)#end
RT3#showiproute
Codes:C-connected, S-static, I-IGRP, R-RIP, M-mobile, B-BGP
D-EIGRP, EX-EIGRPexternal, O-OSPF, IA-OSPFinterarea
N1-OSPFNSSAexternaltype1, N2-OSPFNSSAexternaltype2
E1-OSPFexternaltype1, E2-OSPFexternaltype2, E-EGP
i-IS-IS, L1-IS-ISlevel-1, L2-IS-ISlevel-2, ia-IS-ISinterarea
*-candidatedefault, U-per-userstaticroute, o-ODR
P-periodicdownloadedstaticroute

```

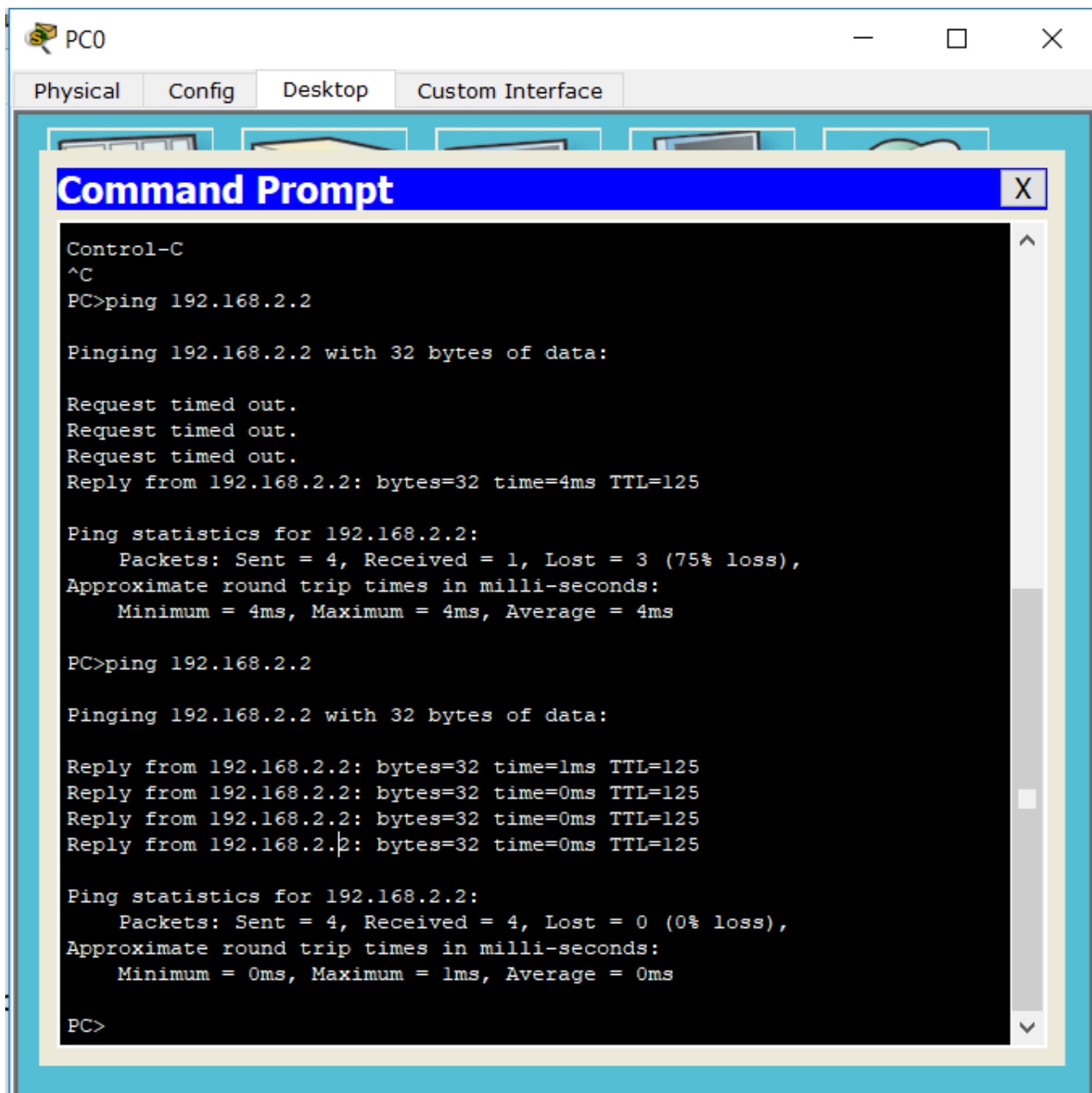
Gatewayoflastresortisnotset

```

          10.0.0.0/24 is subnetted, 1 subnets
C       10.0.2.0 is directly connected, FastEthernet0/1
S       192.168.1.0/24 [1/0] via 10.0.2.1
C       192.168.2.0/24 is directly connected, FastEthernet0/0

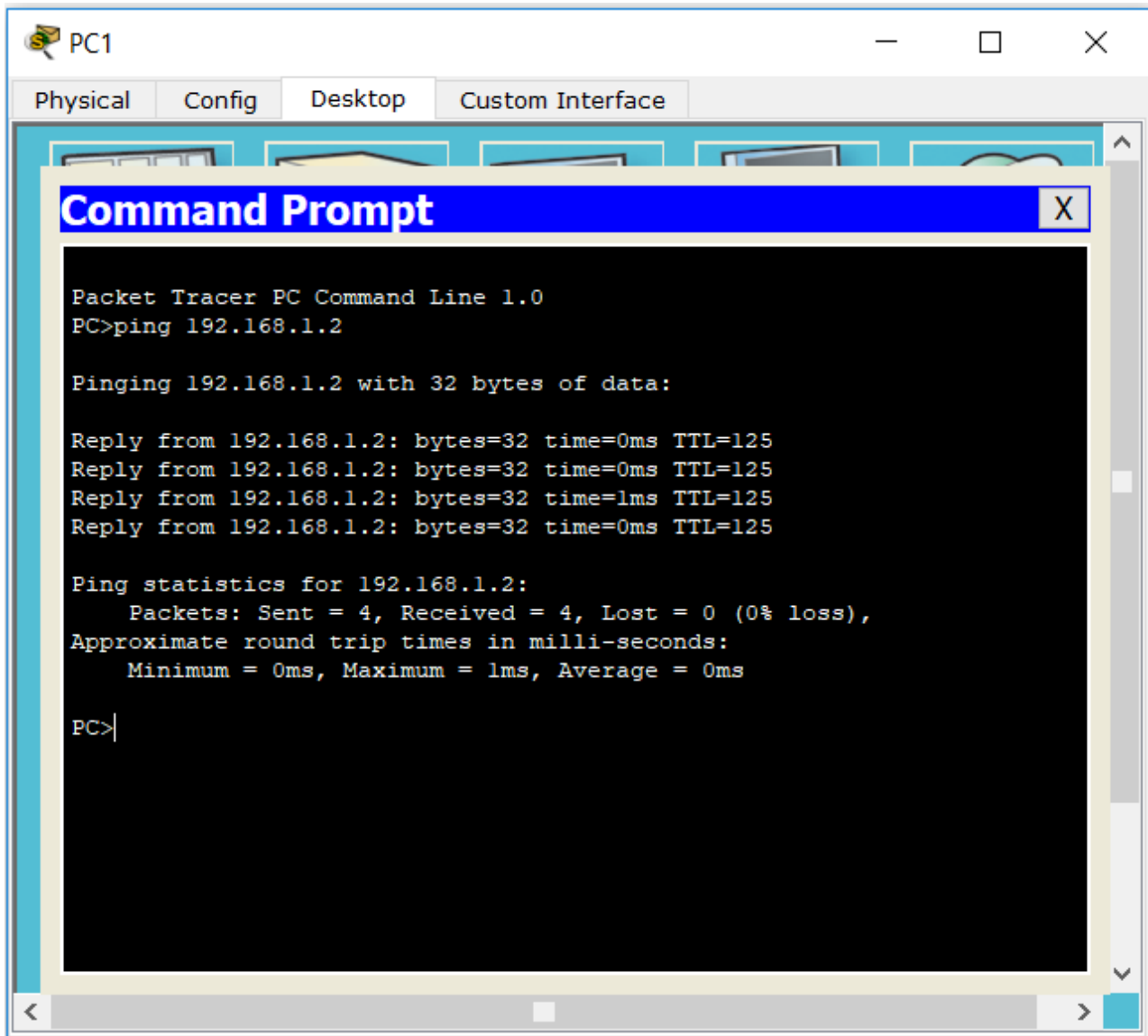
```

Маршруты на всех устройствах прописаны, а значит PC0 сможет достигаться до PC1 и наоборот PC1 до PC0. Проверим:



Обратите внимание на то, что первые 3 запроса потерялись по тайм-ауту (не Unreachable). Это так СРТ эмулирует работу ARP. По сути эти 3 потерянных пакета — это следствие того, что каждый маршрутизатор по пути запускал ARP-запрос до своего соседа. В итоге после всех работ PC0 успешно пингует PC1. Проверим обратную связь:





И с этой стороны все прекрасно.

[Ссылка](#) на скачивание.

Теперь на примере таблицы R3 объясню, что она из себя представляет:

```
RT3#showiproute
Codes:C-connected, S-static, I-IGRP, R-RIP, M-mobile, B-BGP
D-EIGRP, EX-EIGRPexternal, O-OSPF, IA-OSPFinterarea
N1-OSPFNSSAexternaltype1, N2-OSPFNSSAexternaltype2
E1-OSPFexternaltype1, E2-OSPFexternaltype2, E-EGP
i-IS-IS, L1-IS-ISlevel-1, L2-IS-ISlevel-2, ia-IS-ISinterarea
*-candidatedefault, U-per-userstaticroute, o-ODR
P-periodicdownloadedstaticroute
```

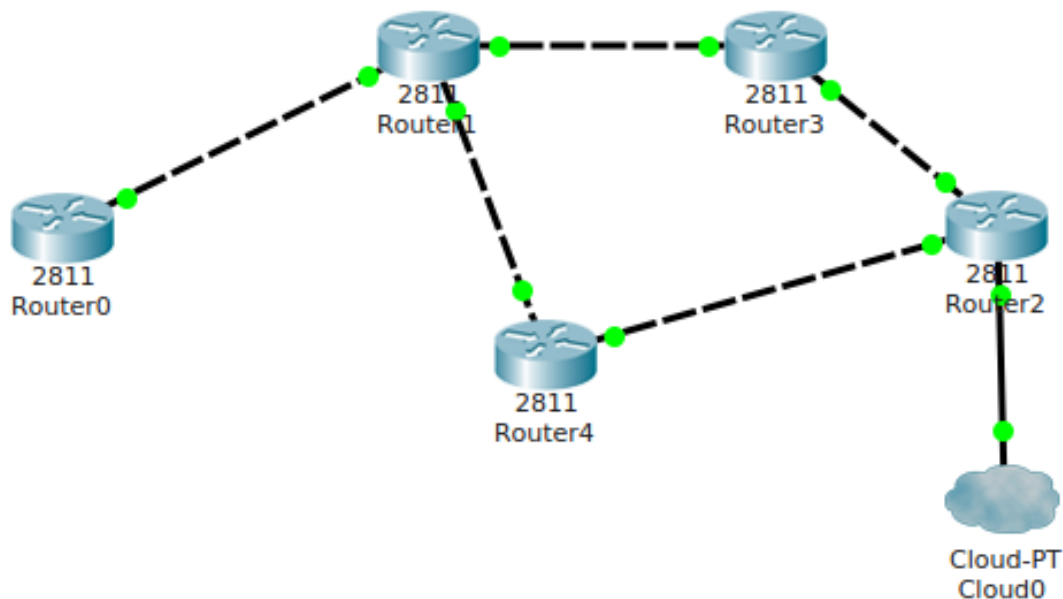
```
Gatewayoflastresortisnotset
```

```
10.0.0.0/24 is subnetted, 1 subnets
C    10.0.2.0 is directly connected, FastEthernet0/1
S    192.168.1.0/24 [1/0] via 10.0.2.1
C    192.168.2.0/24 is directly connected, FastEthernet0/0
```

Коды (они же легенды) показывают, каким методом данный маршрут попал в таблицу. Их тут много и заострять внимание на все нет смысла (так как ныне не используются). Остановимся на двух — **C**(connected) и **S**(static).

Как только мы прописываем IP-адрес и активируем интерфейс, подсеть, к которой он принадлежит, автоматически попадает в таблицу маршрутизации. Поэтому справа от этой строки подписано **directly connected** и интерфейс, привязанный к этой подсети. Тоже самое с подсетью 192.168.2.0/24. А вот со статически заданным адресом чуть по другому. Подсеть 192.168.1.0/24 не напрямую подсоединена к текущему маршрутизатору, а доступна через 10.0.2.1. А вот этот next-hop уже принадлежит к 10.0.2.0/24 (которая напрямую доступна). Таким образом можно добраться до удаленной подсети, через знакомую сеть. Это может показаться немного запутанным, но именно так работает логика маршрутизатора. Тут еще можно заметить, что в строчке со статическим маршрутом присутствует запись [1/0]. Я чуть позже объясню что это, когда будет разбираться динамическая маршрутизация. Просто на фоне ее эти цифры сразу обретут смысл. А сейчас важно просто запомнить, что первое число — это административная дистанция, а второе — метрика.

Теперь перейдем к разделу динамической маршрутизации. Начну сразу с картинки:



И сразу вопрос: В чем сложность этой схемы? На самом деле ни в чем, до того момента, пока не придется это все настраивать. Сейчас мы умеем настраивать статическую маршрутизацию. И за n-ое количество времени поднимем сеть и она будет работать. А теперь несколько но:

- На одном из маршрутизаторов появилась новая подсеть. Это значит, что нужно на всех маршрутизаторах вручную прописать маршрут до нее.
- Допустим мы из Router0 ходили до Cloud0 по цепочке **0 -> 1 -> 3 -> 2 -> Cloud0**. Теперь внезапно сгорел/умер/украли Router3. Соответственно не было запасного пути и доступ до Cloud0 закрыт. Сеть стоит и компания не может работать. Тут придется подрываться и переписать цепочку по **0 -> 1 -> 4 -> 2 -> Cloud0**. То есть нет никакого резерва. Если сеть падает, то без админа ничего не решить. Сеть не может сама перестроиться.
- Ну и еще один аргумент, почему строить сеть исключительно на статических маршрутах — зло и не практично. Это, конечно, масштабируемость. Практически любая компания рано или поздно растет, расширяется и сетевых узлов становится все больше. А значит, в конечном итоге, сеть со статическими маршрутами начнет превращаться в ад для сетевого инженера.

Вот на помощь как раз приходит динамическая маршрутизация. Она оперирует двумя очень созвучными понятиями, но совершенно разными по смыслу:

1. **Routing protocols (протоколы маршрутизации)** — это как раз те протоколы, о которых чуть ниже поговорим. При помощи этих протоколов, роутеры обмениваются маршрутной информацией и строят топологию.
2. **Routed protocols (маршрутизируемые протоколы)** — это как раз те протоколы, которые мы маршрутизируем. В данном случае — это IPv4, IPv6.

Протоколы динамической маршрутизации делятся на 2 категории:

- IGP (interior gateway protocols) — внутренние протоколы маршрутизации (RIP, OSPF, EIGRP). Гости этого выпуска.
- EGP (external gateway protocols) — внешние протоколы маршрутизации (на сегодня BGP).

Отличий в них много, но самые главные — IGP запускается внутри одной автономной системы (считайте компании), а EGP запускается между автономными системами (то есть это маршрутизация в Интернете. При помощи него автономные системы связываются между собой). Сейчас представитель EGP остался один — это BGP. Я не буду долго на нем останавливаться, так как он выходит за рамки CCNA. Да и по нему лучше делать отдельную статью, чтобы не смешивать и так довольно емкий материал.

Теперь про IGP. Это прозвучит смешно, но и они делятся на несколько категорий:

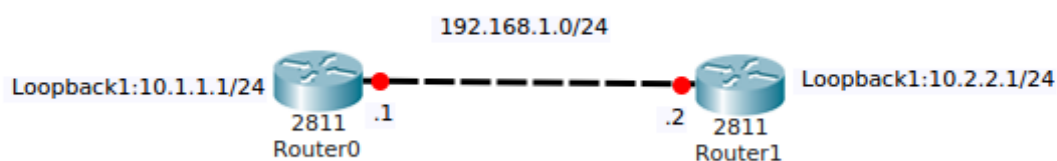
- Distance-Vector (дистанционно-векторные)
- Hybrid or Advanced Distance Vector (гибридные или продвинутые дистанционно-векторные)
- Link-State (протокол состояния канала)

Начну с **дистанционно-векторного**. Он, на мой взгляд, самый простой для понимания.

Название ему такое дали не с проста. **Дистанция** показывает расстояние до точки назначения. Дальностью оперирует такой показатель, как метрика (о чем я упоминал выше). **Вектор** показывает направление до точки назначения. Это может быть выходной интерфейс, IP-адрес соседа.

Мне этот протокол напоминает дорожный указатель. То есть по какому направлению идти и какое расстояние до точки назначения.

Теперь покажу на практике, как он работает и по ходу детально разберем.



Чтобы не загромождать статью однообразными настройками, я заранее сконфигурировал устройства. А именно прописал IP-адреса и включил интерфейсы. Оставляю под спойлерами настройки:

**Router0:**

**Router1:**

Единственное, что может показаться новым — это Loopback интерфейсы. Он практически не отличается от других интерфейсов, за исключением того, что не представлен физически и к нему ничего нельзя воткнуть. Он программно создан внутри самого устройства. Такой интерфейс есть и на многих ОС, как Windows и Linux-подобных. На примере он используется для того, чтобы не рисовать множество маршрутизаторов со своими подсетями.

Сейчас таблицы маршрутизации выглядят следующим образом:

Router0:

```
Router0#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
10.0.0.0/24 is subnetted, 1 subnets
C      10.1.1.0 is directly connected, Loopback1
C      192.168.1.0/24 is directly connected, FastEthernet0/0
```

Router1:

```
Router1#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
10.0.0.0/24 is subnetted, 1 subnets
C      10.2.2.0 is directly connected, Loopback1
C      192.168.1.0/24 is directly connected, FastEthernet0/0
```

То есть у каждого в таблице маршрут общий с соседом (192.168.1.0/24) и недоступный другому соседу (10.1.1.0 и 10.2.2.0 соответственно).

Теперь для связности 2 маршрутизатора должны обмениваться своими маршрутными информациями. И вот тут поможет протокол RIP.

Переключаю РТ в режим симуляции и перехожу к настройкам:

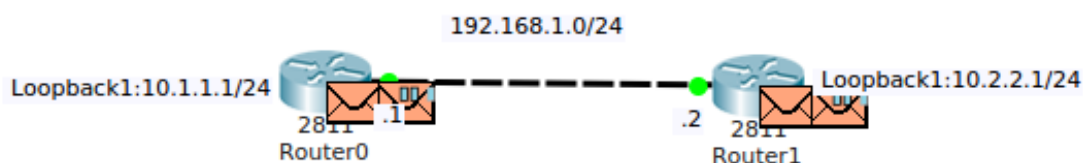
Router0:

```
Router0#conf t -- переход в режим глобальной конфигурации
Enter configuration commands, one per line. End with CNTL/Z.
Router0(config)#router rip -- переход к настройке протокола
Router0(config-router)#version 2 -- включается протокол 2-ой версии
Router0(config-router)#no auto-summary -- отключается автоматическое суммирование
Router0(config-router)#network 10.1.1.0 -- активируется RIP на интерфейсе из
данной подсети
Router0(config-router)#network 192.168.1.0
```

Router1:

```
Router1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router1(config)#router rip
Router1(config-router)#version 2
Router1(config-router)#no auto-summary
Router1(config-router)#network 10.2.2.0
Router1(config-router)#network 192.168.1.0
```

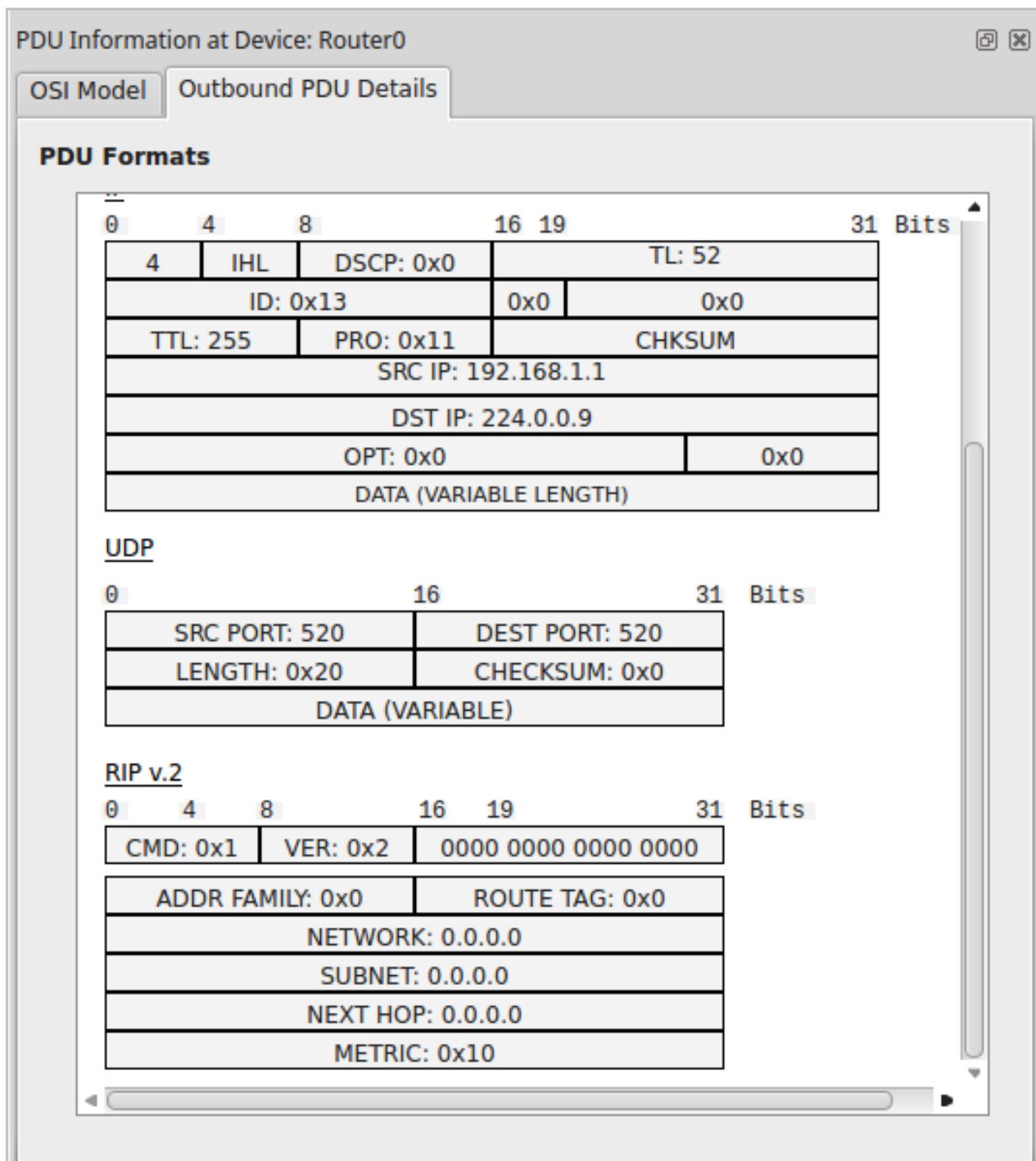
Сразу оговорюсь, что протокол RIP (также как EIGRP и OSPF) не анонсирует подсети таким образом. Он включает протокол на данном интерфейсе. То есть нельзя анонсировать то, что устройство не знает. И замечу, что включена вторая версия протокола и отключено автосуммирование. Изначально RIP был придуман для сетей с классовой адресацией. Поэтому суммирование он выполняет по тем же правилам, что не корректно в применении к бесклассовой. После перехода на бесклассовую адресацию, нужно было изменить работу протокола RIP. И вот во второй версии помимо подсети, передается еще и маска.



На схеме сразу же оба маршрутизатора что-то сгенерировали:

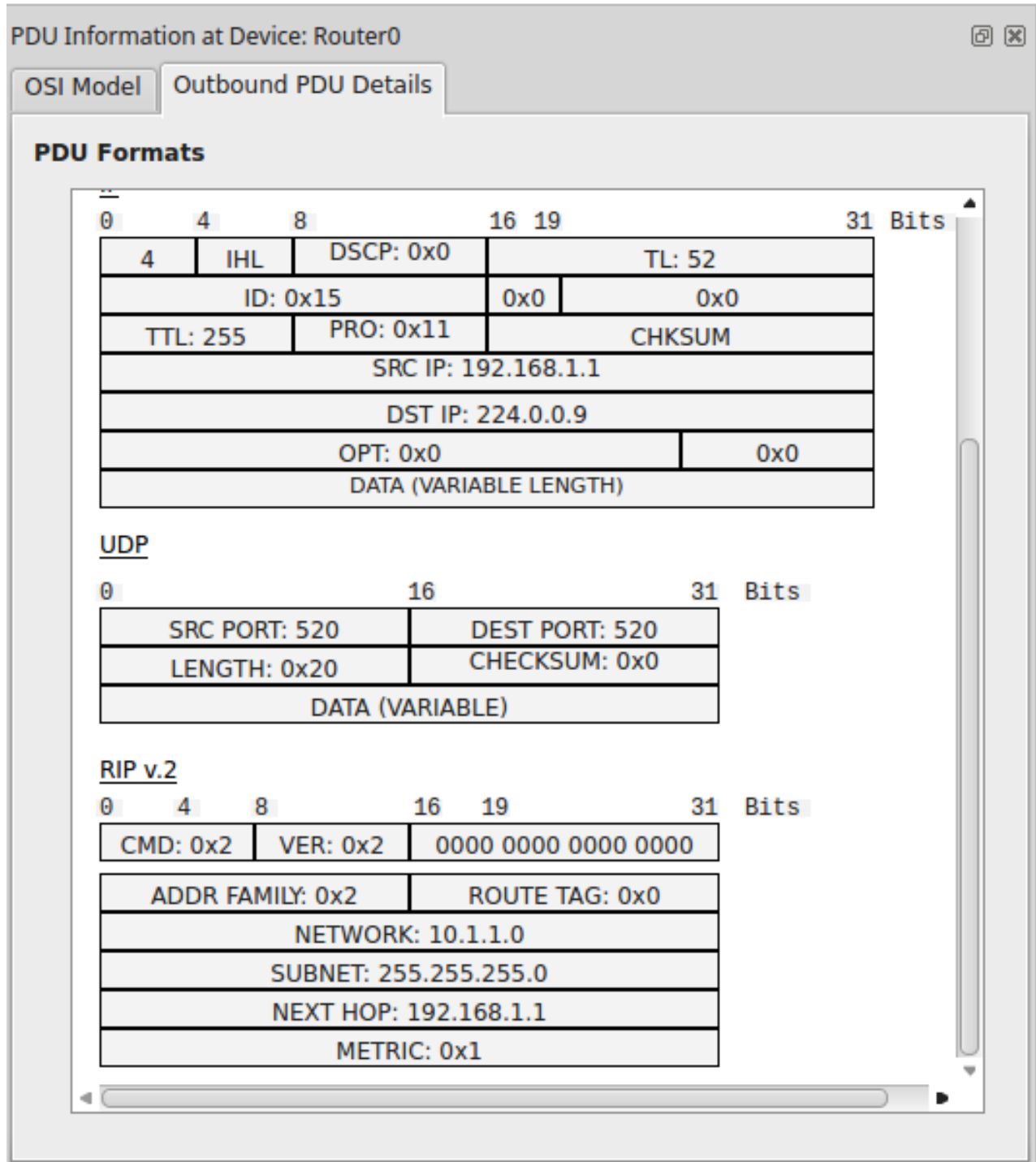
Первый пакет:





Это первый пакет, который генерирует роутер, при включении RIP. Тут важный аспект, что ничего не анонсируется и метрика = 16. (0x10 в шестнадцатичном значение = 16 в десятичном).

Второй пакет:



А вот этот пакет уже несет полезную информацию.

- 1) **ADDR FAMILY: 0x2** — означает IP протокол. В большинстве случаев это поле не меняется.
- 2) **NETWORK: 10.1.1.0** — подсеть, которая анонсируется.
- 3) **SUBNET: 255.255.255.0** — маска
- 4) **NEXT HOP: 192.168.1.1** — следующий узел для достижимости анонсированной подсети.
- 5) **METRIC: 0x1** — стоимость пути (в данном случае 1).

С обратной стороны придет точно такой же анонс (только будет соответствующая подсеть, nexthop).

В итоге после получения анонсов, таблицы у обоих роутеров будут выглядеть следующим образом:

Router0:

```
Router0#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
      10.0.0.0/24 is subnetted, 2 subnets
C       10.1.1.0 is directly connected, Loopback1
R       10.2.2.0 [120/1] via 192.168.1.2, 00:00:03, FastEthernet0/0
C      192.168.1.0/24 is directly connected, FastEthernet0/0
```

Router1:

```
Router1#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
      10.0.0.0/24 is subnetted, 2 subnets
R       10.1.1.0 [120/1] via 192.168.1.1, 00:00:16, FastEthernet0/0
C       10.2.2.0 is directly connected, Loopback1
C      192.168.1.0/24 is directly connected, FastEthernet0/0
```

В таблице появилась пометка с кодом **R**. То есть получен по протоколу RIP.

Если пускать пинги:

Router0:

```
Router0#ping 10.2.2.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms
```

Router1:

```
Router1#ping 10.1.1.1
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

Анонсируемые подсети достижимы. Еще важный аспект, при работе с протоколами маршрутизации — это просмотр сформированной базы. Таблица маршрутизации — это конечный итог, куда заносится маршрут. Посмотреть базу можно командой *show ip rip database*:

Router0:

```
Router0#show ip rip database
```

```
10.1.1.0/24    auto-summary
```

```
10.1.1.0/24    directly connected, Loopback1
```

```
10.2.2.0/24    auto-summary
```

```
10.2.2.0/24
```

```
    [1] via 192.168.1.2, 00:00:03, FastEthernet0/0
```

```
192.168.1.0/24    auto-summary
```

```
192.168.1.0/24    directly connected, FastEthernet0/0
```

Router1:

```
Router1#show ip rip database
```

```
10.1.1.0/24    auto-summary
```

```
10.1.1.0/24
```

```
    [1] via 192.168.1.1, 00:00:13, FastEthernet0/0
```

```
10.2.2.0/24    auto-summary
```

```
10.2.2.0/24    directly connected, Loopback1
```

```
192.168.1.0/24    auto-summary
```

```
192.168.1.0/24    directly connected, FastEthernet0/0
```

Эта команда полезна, когда маршруты никак не заносятся в таблицу, при этом вроде как RIP включен и настроено все верно. Если маршрута нет в базе, значит он никак не попадет в таблицу и тут надо копать глубже. У циски, к счастью, есть хороший инструмент для дебага, который позволяет практически моментально понять, что происходит. В СРТ он урезан и многое не показать, но на реальных железах, он прекрасен.

Например:

```
Router0#debug ?
aaa          AAA Authentication, Authorization and Accounting
crypto       Cryptographic subsystem
custom-queue Custom output queueing
eigrp        EIGRP Protocol information
ephone       ethernet phone skinny protocol
frame-relay  Frame Relay
ip           IP information
ipv6         IPv6 information
ntp          NTP information
ppp          PPP (Point to Point Protocol) information
```

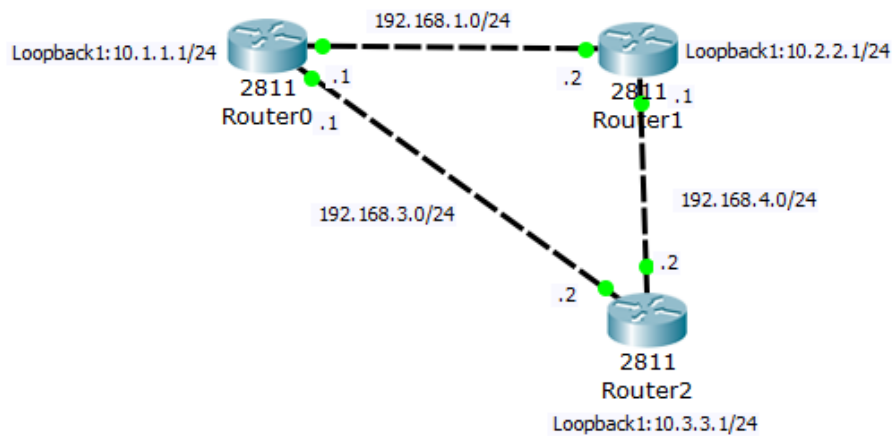
Посмотрим, что происходит в RIP:

```
Router0#debug ip rip
RIP protocol debugging is on
Router0#RIP: sending v2 update to 224.0.0.9 via Loopback1 (10.1.1.1)
RIP: build update entries
    10.2.2.0/24 via 0.0.0.0, metric 2, tag 0
    192.168.1.0/24 via 0.0.0.0, metric 1, tag 0
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (192.168.1.1)
RIP: build update entries
    10.1.1.0/24 via 0.0.0.0, metric 1, tag 0
RIP: received v2 update from 192.168.1.2 on FastEthernet0/0
    10.2.2.0/24 via 0.0.0.0 in 1 hops
```

Сейчас все хорошо. Видно, что приходят/уходят апдейты и записи обновляются. Из-за того, что дебажный инструмент обширен, лучше явно указывать что нужно ловить (как представлено выше). Иначе можно достаточно хорошо пригрузить устройство. Важно помнить про команду *undebg all*. Она отключает весь дебаг на устройстве.

[Ссылка](#) на скачивание лабы. Можете добавить еще один маршрутизатор к существующей схеме и связать их через RIP.

Теперь усложним схему и посмотрим в чем преимущество динамической маршрутизации.



Добавился Router2, который соединен с ранее созданными маршрутизаторами и анонсирует подсеть 10.3.3.0/24.

Настраиваются аналогично предыдущему примеру. Поэтому покажу только конфигурации:

#### Router0

#### Router1

#### Router2

Итого на Router0 мы имеем следующую таблицу маршрутизации:



```
Router0# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

Gateway of last resort is not set

```
10.0.0.0/24 is subnetted, 3 subnets
C      10.1.1.0 is directly connected, Loopback1
R      10.2.2.0 [120/1] via 192.168.1.2, 00:00:05, FastEthernet0/0
R      10.3.3.0 [120/1] via 192.168.3.2, 00:00:14, FastEthernet0/1
C      192.168.1.0/24 is directly connected, FastEthernet0/0
C      192.168.3.0/24 is directly connected, FastEthernet0/1
R      192.168.4.0/24 [120/1] via 192.168.1.2, 00:00:05, FastEthernet0/0
                        [120/1] via 192.168.3.2, 00:00:14, FastEthernet0/1
```

Из новых маршрутов — это 10.3.3.0/24, который доступен через 192.168.3.2 (т.е. Router2). И второй маршрут — это 192.168.4.0/24, который доступен через 192.168.1.2 (т.е. Router1) и 192.168.3.2 (т.е. Router2).

Вот в тех случаях, когда маршруты от разных устройств до одной подсети приходят с одинаковой метрикой, оба заносятся в таблицу. Такой случай называют **балансировкой** или **ECMP (Equal-cost multi-path routing)**.

Если пройти по нему через traceroute:

```
Router0#traceroute 192.168.4.1
Type escape sequence to abort.
Tracing the route to 192.168.4.1

 1  192.168.1.2      1 msec    0 msec    0 msec
Router0#traceroute 192.168.4.1
Type escape sequence to abort.
Tracing the route to 192.168.4.1

 1  192.168.3.2      1 msec    0 msec    0 msec
```

То есть меняется next-hop по очереди. Сама тема балансировки заслуживает отдельного внимания, т.к. у балансировки есть несколько стратегий по выбору оптимального пути. Случай, когда балансировка работает по очереди, как в нашем случае — называют **Round-Robin**.

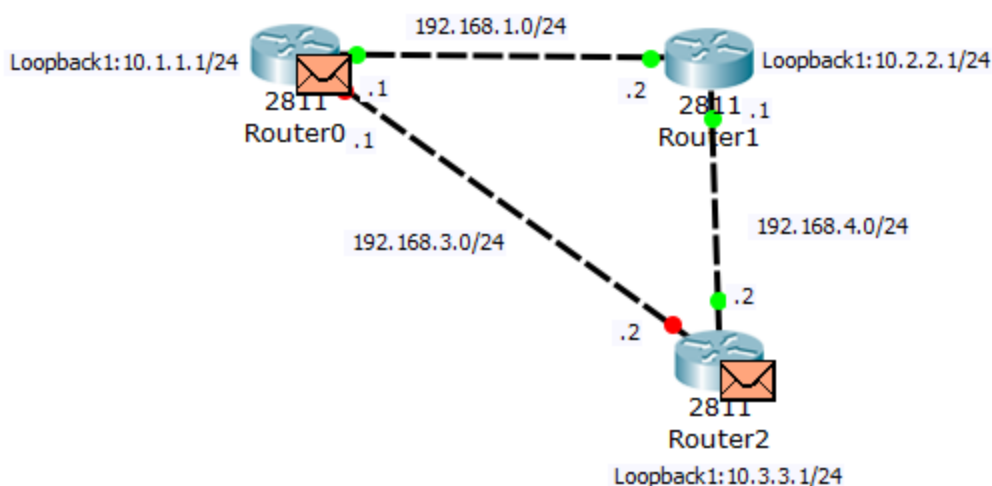
Посмотрим базу RIP на Router0:

```

Router0#show ip rip database
10.1.1.0/24    auto-summary
10.1.1.0/24    directly connected, Loopback1
10.2.2.0/24    auto-summary
10.2.2.0/24
    [1] via 192.168.1.2, 00:00:01, FastEthernet0/0
10.3.3.0/24    auto-summary
10.3.3.0/24
    [1] via 192.168.3.2, 00:00:23, FastEthernet0/1
192.168.1.0/24    auto-summary
192.168.1.0/24    directly connected, FastEthernet0/0
192.168.3.0/24    auto-summary
192.168.3.0/24    directly connected, FastEthernet0/1
192.168.4.0/24    auto-summary
192.168.4.0/24
    [1] via 192.168.1.2, 00:00:01, FastEthernet0/0    [1] via 192.168.3.2,
00:00:23, FastEthernet0/1

```

То есть нет никакого запасного маршрута, на случай выхода из строя 192.168.3.2. Теперь переключаю в режим симуляции и смотрю, что произойдет, если отключить на Router0 интерфейс fa0/1:



Видим, что отключился линк на Router0 и Router2. И сразу оба устройства генерируют сообщения:

Router0:

# PDU Information at Device: Router0



OSI Model

Outbound PDU Details

## PDU Formats

### UDP

0	16	31	Bits
SRC PORT: 520		DEST PORT: 520	
LENGTH: 0x48		CHECKSUM: 0x0	
DATA (VARIABLE)			

### RIP v.2

0	4	8	16	19	31	Bits
CMD: 0x2		VER: 0x2		0000 0000 0000 0000		
ADDR FAMILY: 0x2			ROUTE TAG: 0x0			
NETWORK: 10.3.3.0						
SUBNET: 255.255.255.0						
NEXT HOP: 0.0.0.0						
METRIC: 0x10						
ADDR FAMILY: 0x2			ROUTE TAG: 0x0			
NETWORK: 192.168.3.0						
SUBNET: 255.255.255.0						
NEXT HOP: 0.0.0.0						
METRIC: 0x10						
ADDR FAMILY: 0x2			ROUTE TAG: 0x0			
NETWORK: 192.168.4.0						
SUBNET: 255.255.255.0						
NEXT HOP: 0.0.0.0						
METRIC: 0x10						

Router1:

## PDU Formats

UDP

0	16	31	Bits
SRC PORT: 520		DEST PORT: 520	
LENGTH: 0x48		CHECKSUM: 0x0	
DATA (VARIABLE)			

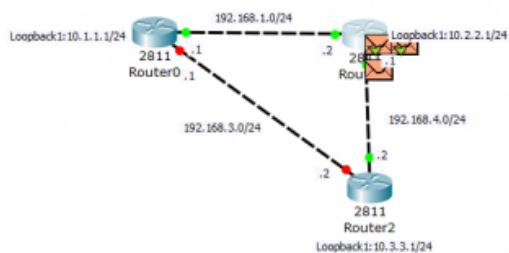
RIP v.2

0	4	8	16	19	31	Bits
CMD: 0x2		VER: 0x2		0000 0000 0000 0000		
ADDR FAMILY: 0x2			ROUTE TAG: 0x0			
NETWORK: 10.1.1.0						
SUBNET: 255.255.255.0						
NEXT HOP: 0.0.0.0						
METRIC: 0x10						
ADDR FAMILY: 0x2			ROUTE TAG: 0x0			
NETWORK: 192.168.1.0						
SUBNET: 255.255.255.0						
NEXT HOP: 0.0.0.0						
METRIC: 0x10						
ADDR FAMILY: 0x2			ROUTE TAG: 0x0			
NETWORK: 192.168.3.0						
SUBNET: 255.255.255.0						
NEXT HOP: 0.0.0.0						
METRIC: 0x10						

Сразу сообщают, что данные маршруты теперь недостижимы. Делают они это, при помощи метрики, которая становится равной 16. Исторически так сложилось, что протокол RIP был рассчитан на работу с 15 транзитными участками. В то время никто не подразумевал, что сеть может быть настолько большой:-). Называется этот механизм **Poison Reverse**.

Таким образом сосед, получивший такой апдейт должен удалить этот маршрут из таблицы.

Вот, что происходит на Router1:



```

Router1>
Router1>
Router1>en
Router1#enable
Router1#sh
Router1#show ip ro
Router1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
       inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 3 subnets
R    10.1.1.0 [120/1] via 192.168.1.1, 00:00:07, FastEthernet0/0
C    10.2.2.0 is directly connected, Loopback1
R    10.3.3.0 [120/1] via 192.168.4.2, 00:00:00, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
R    192.168.1.0/24 is possibly down, routing via 192.168.4.2,
FastEthernet0/1
C    192.168.4.0/24 is directly connected, FastEthernet0/1
Router1#
  
```

И самое интересное, что после этого Router1 отправит Router0 следующее:

## PDU Information at Device: Router0

OSI Model

Inbound PDU Details

### PDU Formats

#### IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 52		
ID: 0x349			0x0	0x0	
TTL: 255		PRO: 0x11	CHKSUM		
SRC IP: 192.168.1.2					
DST IP: 224.0.0.9					
OPT: 0x0				0x0	
DATA (VARIABLE LENGTH)					

#### UDP

0	16	31 Bits
SRC PORT: 520		DEST PORT: 520
LENGTH: 0x20		CHECKSUM: 0x0
DATA (VARIABLE)		

#### RIP v.2

0	4	8	16	19	31	Bits
CMD: 0x2		VER: 0x2		0000 0000 0000 0000		
ADDR FAMILY: 0x2				ROUTE TAG: 0x0		
NETWORK: 192.168.3.0						
SUBNET: 255.255.255.0						
NEXT HOP: 0.0.0.0						
METRIC: 0x10						

То есть я больше не знаю о 192.168.3.0/24.

На данный момент таблица на Router0 выглядит следующим образом:



```
Router0#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

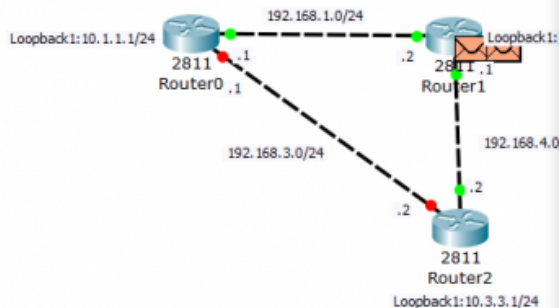
Gateway of last resort is not set

10.0.0.0/24 is subnetted, 2 subnets

```
C      10.1.1.0 is directly connected, Loopback1
R      10.2.2.0 [120/1] via 192.168.1.2, 00:00:29, FastEthernet0/0
C     192.168.1.0/24 is directly connected, FastEthernet0/0
R     192.168.4.0/24 [120/1] via 192.168.1.2, 00:00:29, FastEthernet0/0
```

То есть знает о своих подсетях и тех, что анонсировал Router1.

Двигаемся дальше:



PDU Information at Device: Router1

OSI Model    Outbound PDU Details

PDU Formats

RIP v.2

048161931Bits

CMD: 0x2

VER: 0x2

0000 0000 0000 0000

ADDR FAMILY: 0x2

ROUTE TAG: 0x0

NETWORK: 10.2.2.0

SUBNET: 255.255.255.0

NEXT HOP: 192.168.1.2

METRIC: 0x1

ADDR FAMILY: 0x2

ROUTE TAG: 0x0

NETWORK: 10.3.3.0

SUBNET: 255.255.255.0

NEXT HOP: 192.168.1.2

METRIC: 0x2

ADDR FAMILY: 0x2

ROUTE TAG: 0x0

NETWORK: 192.168.3.0

SUBNET: 255.255.255.0

NEXT HOP: 192.168.1.2

METRIC: 0x10

ADDR FAMILY: 0x2

ROUTE TAG: 0x0

NETWORK: 192.168.4.0

SUBNET: 255.255.255.0

NEXT HOP: 192.168.1.2

METRIC: 0x1

Видим, что Router1 генерирует пакет с кучей подсетей и отправляет соседям. В том числе там подсеть 10.4.4.0.

И в таблице Router0 теперь:

```
Router0#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

Gateway of last resort is not set

```
10.0.0.0/24 is subnetted, 3 subnets
C      10.1.1.0 is directly connected, Loopback1
R      10.2.2.0 [120/1] via 192.168.1.2, 00:00:00, FastEthernet0/0
R      10.3.3.0 [120/2] via 192.168.1.2, 00:00:00, FastEthernet0/0
C      192.168.1.0/24 is directly connected, FastEthernet0/0
R      192.168.4.0/24 [120/1] via 192.168.1.2, 00:00:00, FastEthernet0/0
```

Замечу, что в таблице она записана с метрикой 2. Потому что данный маршрут направлен не напрямую от соседа, породившего его, а через транзитный маршрутизатор, который добавил 1.

Проверим доступность:

```
Router0#ping 10.3.3.1
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms
```

```
Router0#traceroute 10.3.3.1
Type escape sequence to abort.
Tracing the route to 10.3.3.1
```

```
 1  192.168.1.2      0 msec    0 msec    0 msec
 2  192.168.4.2      2 msec    0 msec    0 msec
```

Пинги проходят, а через traceroute видим, что пакет сначала попадает на Router1, а дальше маршрутизируется на Router2.

То есть видно очевидное преимущество динамического протокола маршрутизации над статическими. При падении линка и наличии резервного пути, топология сама перестроилась. На сегодняшний день мало кто использует данный протокол. И на это есть множество причин. Одна из них — это количество транзитных маршрутов. Вдобавок ко всему — это время сходимости. По умолчанию все маршрутизаторы отправляют друг другу апдейты каждые 30 секунд. Если обновление не приходит в течении 180 секунд, маршрут помечается, как Invalid. А как время простоя доходит до 240 секунд, он удаляется. Конечно таймеры можно подкрутить. Но проблема еще в том, что в большой сети, при наличии проблемы где-нибудь по середине, апдейт с одного конца до другого может просто-напросто не дойти. Хотя он доступен. Есть

еще одна проблема. RIP хранит только лучший маршрут. Поэтому когда отключился линк, маршрут пропал и резервного пути не было. А значит, пока никто из соседей не проанонсирует подсеть, она будет недоступной. Это очень ощутимо для сетей, в которых простой стоит дорого. В связи с этим были придуманы протоколы, у которых время сходимости выше и есть резервные пути. О них и поговорим. Хочу также отметить, что RIP — протокол не плохой (уж явно лучше, чем использование только статических маршрутов в растущей сети). Поэтому изучение лучше начать с него. Таким образом концепция динамической маршрутизации уляжется лучше. Да что тут говорить, если Cisco сначала убрала RIP из своих экзаменов, а теперь снова включила.

[Ссылка](#) на скачивание.

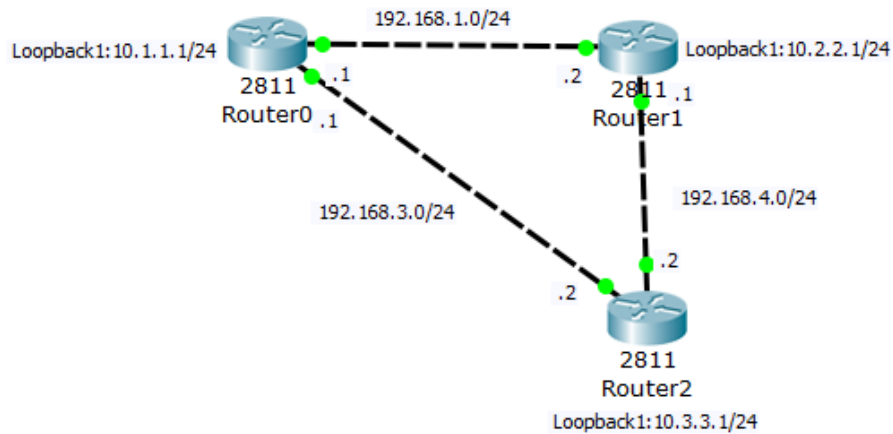
Теперь перейдем к **EIGRP**. Если RIP уже давно является открытым протоколом, то EIGRP был проприетарным и работал только на устройствах Cisco. Но в 2016 году Cisco решила все же открыть его, оставив авторство за собой. [Ссылка](#) на RFC7868. Cisco называет его гибридным (имея в виду, что он взял что-то от Distance-Vector, а что-то от Link-State). В отличие от RIP он работает более «умно». В том плане, что у него есть резервные маршруты и он «хранит некую топологию сети» (хотя это верно очень частично).

Опирирует он 3-мя таблицами:

- 1) **EIGRP Neighbor Table**: Здесь представлены все напрямую соединенные соседи (то есть кто Next-Хор и с какого интерфейса к нему добраться).
- 2) **EIGRP Topology Table**: Здесь представлены все изученные маршруты от соседей (с точкой назначения и метрикой)
- 3) **Global Routing Table**: Общая для всех таблица и сюда попадают лучшие маршруты из предыдущей таблицы.

Соберем топологию и запустим на ней EIGRP. Попутно буду рассказывать, что происходит, чтобы совместить минимум теории с максимумом практики.

Топологию возьмем ту же, что и с RIP. На ней настроены все IP-адреса, подняты интерфейсы, но не запущен протокол маршрутизации.



**Router0:**

**Router1**

**Router2**

Сейчас в маршрутных таблицах роутеров только **Connected** подсети.  
Переходим в настройки EIGRP.

Router0:

```

router eigrp 1 - номер автономной системы (должен совпадать на всех устройствах)
network 10.1.1.0 0.0.0.255
network 192.168.1.0 0.0.0.255
network 192.168.3.0 0.0.0.255
no auto-summary
  
```

Router1:

```

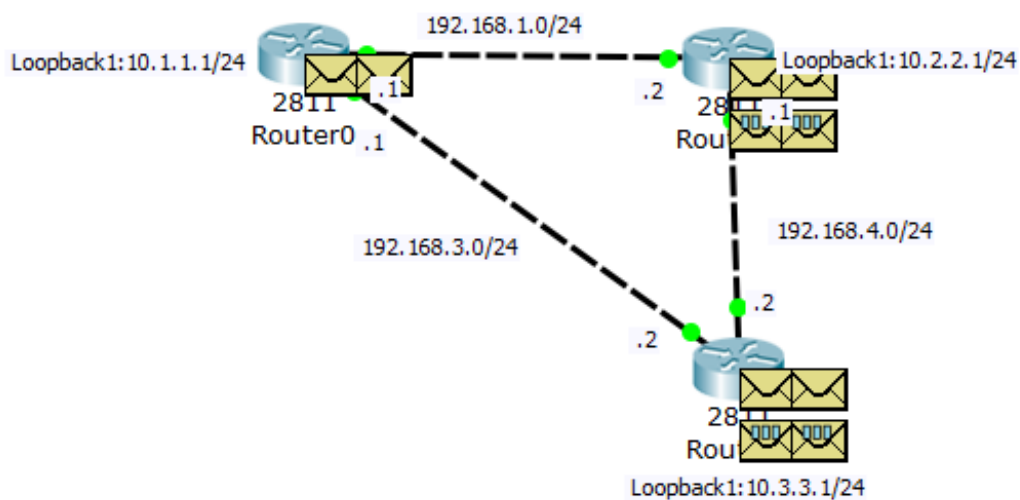
router eigrp 1 - номер автономной системы (должен совпадать на всех устройствах)
network 10.2.2.0 0.0.0.255
network 192.168.1.0 0.0.0.255
network 192.168.4.0 0.0.0.255
no auto-summary
  
```

Router2:

```
router eigrp 1 - номер автономной системы (должен совпадать на всех устройствах)
network 10.3.3.0 0.0.0.255
network 192.168.3.0 0.0.0.255
network 192.168.4.0 0.0.0.255
no auto-summary
```

Как описал выше, при включении EIGRP, ему присваивается номер AS. И он должен совпадать на всех соседях. В настройках анонса сети теперь добавляется wildcard маска. Если не вдаваться в подробности — это обратная запись маски (т.е. 0.0.0.255 — это 255.255.255.0). И отключение автосуммирования (наследие классовых сетей).

В итоге видим следующую картину:



Посмотрим, что сгенерировал Router0:

## PDU Formats

0x800		0x0
-------	--	-----

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 20		
ID: 0x40d			0x0	0x0	
TTL: 255		PRO: 0x58	CHKSUM		
SRC IP: 192.168.3.1					
DST IP: 224.0.0.10					
OPT: 0x0				0x0	
DATA (VARIABLE LENGTH)					

EIGRP

0	4	8	16	19	31 Bits
VER: 0x2		OPC: 0x5		CHECKSUM: 0x0	
FLAGS: 0x0					
SEQ. NUM: 1					
ACKNUM: 0					
AUTONOMOUS SN: 1					
TYPE: 0x1			LENGTH: 0xc		
K1: 0x1		K2: 0x0		K3: 0x1	K4: 0x0
K5: 0x0		RES: 0x0		HOLD TIME: 15000	
TYPE: 0x4			LENGTH: 0x8		
EIGRP VER: 0x102			IOS VER: 0xc02		

Видим кучу полей и попробуем разобраться, что в них. Мы помним, что RIP был не самым надежным вариантом. Он не понимал какой номер пакета, не было механизма отслеживания, подтверждения и прочего. Да и плюс нижестоящий протокол был UDP, который тоже не имеет механизма надежности. EIGRP вообще работает сразу поверх IP (не используя механизмы транспортного уровня). Поэтому все механизмы по отслеживанию ложатся на его поля.

Из важного: появились флаги, SEQ. NUM (номер отправляемого пакета), ACK.NUM (подтверждение на принятый пакет), номер автономной системы (заданный при создании), и параметры K. Вот тут остановлюсь. В RIP метрика считалась тривиально. Пакет пришел, добавляю единицу и передаю дальше. В EIGRP метрика считается исходя из K значений:

- 1) **K1** — bandwidth (или пропускная способность)
- 2) **K2** — load (загруженность)
- 3) **K3** — delay (задержка)
- 4) **K4** — reliability (надежность)
- 5) **K5** — MTU (Maximum Transmission Unit).

Но как правило, при расчете используются только K1 и K3.

Формула таким образом выглядит:

$$\text{Metric} = (\text{K1} * \text{bandwidth}) + [(\text{K2} * \text{bandwidth}) / (256 - \text{load})] + (\text{K3} * \text{delay})$$

.

Запоминать ее наизусть не надо. Просто важно понимать, как происходит расчет метрики.

Вот, что происходит, когда пакет доходит до Router0:

```
Router0(config)#
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.1.2 (FastEthernet0/0) is up: new adjacency

%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.3.2 (FastEthernet0/1) is up: new adjacency
```

К сожалению CPT наглухо тормозит от количества пакетов, поэтому покажу, что происходит в непосредственно таблицах Router0 (в остальных будет аналогично. Поэтому покажу на одном). А после подробно покажу процесс установления соседства в режиме дебага между двумя маршрутизаторами:

### 1) Neighbor Table:

```
Router0#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
```

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RT0	Q Cnt	Seq Num
0	192.168.1.2	Fa0/0	11	00:00:41	40	1000	0	42
1	192.168.3.2	Fa0/1	10	00:00:41	40	1000	0	38

Из важного. Здесь показан сосед, интерфейс (за которым он находится), hold (таймер, по истечении которого, произойдет разрыв соседства. При получении пакета от соседа, он повышается), uptime (как долго живет соседство), SRTT (время между отправкой и подтверждением), RTO (интервал между отправкой) и номер пакета.

### 2) Router0#show ip eigrp topology

## IP-EIGRP Topology Table for AS 1/ID(10.1.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,  
r - Reply status

```
P 10.1.1.0/24, 1 successors, FD is 128256
    via Connected, Loopback1
P 10.2.2.0/24, 1 successors, FD is 156160
    via 192.168.1.2 (156160/128256), FastEthernet0/0
P 10.3.3.0/24, 1 successors, FD is 156160
    via 192.168.3.2 (156160/128256), FastEthernet0/1
P 192.168.1.0/24, 1 successors, FD is 28160
    via Connected, FastEthernet0/0
P 192.168.3.0/24, 1 successors, FD is 28160
    via Connected, FastEthernet0/1
P 192.168.4.0/24, 2 successors, FD is 30720
    via 192.168.3.2 (30720/28160), FastEthernet0/1
    via 192.168.1.2 (30720/28160), FastEthernet0/0
```

Тут все просто. Если все хорошо с полученным маршрутом, то он становится Passive. О других полях и их значениях расскажу чуть позже. Сейчас достаточно того, что в данной таблице все хорошо. Из нового — вводится понятие Successor. Successor-ом выбирается тот, у кого наименьшая стоимость до конкретной подсети. Сейчас на каждый маршрут по одному Successor-у и только на маршрут 192.168.4.0 их два. Причем они оба выбраны Successor-ами из за одинаковой метрики (следовательно будет работать балансировка). Теперь обращаю внимание на странные числа у каждого Successor-а.

EIGRP при расчете метрики оперирует 2-мя понятиями: **Advertised Distance** и **Feasible Distance**. Оба рассчитываются той страшной формулой:

1) **Advertised Distance** — это анонс стоимости от соседа. То есть сколько стоит от него (соседа) и до точки назначения.

2) **Feasible Distance** — это стоимость от самого роутера до точки назначения. То есть — это Advertised Distance + стоимость линка до соседа.

Возьмем для примера запись от маршрута 10.2.2.0:

```
P 10.2.2.0/24, 1 successors, FD is 156160
    via 192.168.1.2 (156160/128256), FastEthernet0/0
```

Число **128256** — это **Advertised Distance**, а **156160** — это **Feasible Distance**.

Соответственно, чем меньше Feasible Distance, тем выгоднее маршрут и такой сосед объявляется Successor-ом. После записи о количестве successors, всегда пишется какая FD была выбрана.

На текущий момент он работает приблизительно также, как и RIP. Только почему то метрика стала сложнее и добавилось больше таблиц. Но вот у EIGRP есть несколько фокусов в кармане. Один из них — это **Feasible Successor** (не путать с Feasible Distance). Это как раз тот самый резервный путь на случай отказа Successor. Сейчас у нас нет резервного пути (например до маршрута 10.2.2.0). Если



падает 192.168.1.2, этот маршрут теряется до момента, пока о нем не расскажет другой сосед. Но мы прекрасно знаем, что о нем может рассказать Router2 (пусть и с худшей метрикой). Но EIGRP все же основан на неких правилах, что не позволяет ему так сделать. А правило заключается в следующем:

Advertised distance of feasible successor < Feasible distance of successor

.

То есть стоимость анонсируемая от Feasible Successor (потенциально backup-роутера) должна быть меньше, чем Feasible Distance Successor (то есть полная стоимость через основного).

Звучит тяжело, но если проще. Взять тот же маршрут 10.2.2.0. Через него FD = 156160. Значит AD от Feasible Successor должна принять любое число меньше 156160. Причем не важно сколько стоит линк от текущего роутера до соседа (хоть 1000000). Главное, чтобы backup-сосед анонсировал с меньшей метрикой, чем successor. Это правило используется для предотвращения петель.

Чтобы понять, как это работает, внесем изменения в топологию.

Сейчас на Router0 таблица топологии выглядит следующим образом:

```
Router0#show ip eigrp topology
```

```
IP-EIGRP Topology Table for AS 1/ID(10.1.1.1)
```

```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status
```

```
P 10.1.1.0/24, 1 successors, FD is 128256
    via Connected, Loopback1
P 10.2.2.0/24, 1 successors, FD is 156160
    via 192.168.1.2 (156160/128256), FastEthernet0/0
P 10.3.3.0/24, 1 successors, FD is 156160
    via 192.168.3.2 (156160/128256), FastEthernet0/1
P 192.168.1.0/24, 1 successors, FD is 28160
    via Connected, FastEthernet0/0
P 192.168.3.0/24, 1 successors, FD is 28160
    via Connected, FastEthernet0/1
P 192.168.4.0/24, 2 successors, FD is 30720
    via 192.168.1.2 (30720/28160), FastEthernet0/0
    via 192.168.3.2 (30720/28160), FastEthernet0/1
```

Маршрут до 10.2.2.0/24 доступен через 192.168.1.2, что верно, так как Router1 его породил и так добраться быстрее всего. Поэтому Router2 не сможет проанонсировать лучше, так как его AD будет всегда выше.

Теперь переведем скорость интерфейсов между Router0 и Router1 на 10Мбит/с. Таким образом ухудшим канал, и внесем изменения в пересчет топологии.

Router0:

```
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed 10
```

Router1:

```
interface FastEthernet0/0
 ip address 192.168.1.2 255.255.255.0
 duplex auto
 speed 10
```

Таким образом на Router0:

```
Router0#show ip eigrp topology
IP-EIGRP Topology Table for AS 1/ID(10.1.1.1)
```

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,  
r - Reply status

```
P 10.1.1.0/24, 1 successors, FD is 128256
    via Connected, Loopback1
P 10.2.2.0/24, 1 successors, FD is 158720
    via 192.168.3.2 (158720/156160), FastEthernet0/1
    via 192.168.1.2 (179200/128256), FastEthernet0/0
P 10.3.3.0/24, 1 successors, FD is 156160
    via 192.168.3.2 (156160/128256), FastEthernet0/1
P 192.168.1.0/24, 1 successors, FD is 51200
    via Connected, FastEthernet0/0
P 192.168.3.0/24, 1 successors, FD is 28160
    via Connected, FastEthernet0/1
P 192.168.4.0/24, 1 successors, FD is 30720
    via 192.168.3.2 (30720/28160), FastEthernet0/1
    via 192.168.1.2 (53760/28160), FastEthernet0/0
```

Видим, что до 10.2.2.0 теперь 2 пути, но Successor выбирается тот, у кого FD выгоднее. А выгоднее, через 192.168.3.2 (то есть Router2), так как у него скорость интерфейсов 100Мбит/с, хоть и преодолеть придется 2 хопа. А теперь обратим внимание, почему попали 2 записи в этот маршрут.

```
    via 192.168.3.2 (158720/156160), FastEthernet0/1
    via 192.168.1.2 (179200/128256), FastEthernet0/0
```

А потому что AD у 192.168.1.2 лучше, чем FD у 192.168.3.2 ( $128256 < 158720$ ). И в таблицу маршрутизации попадет маршрут через выбранного Successor-а, то есть 192.168.3.2:

```
Router0#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

Gateway of last resort is not set

```
      10.0.0.0/24 is subnetted, 3 subnets
C       10.1.1.0 is directly connected, Loopback1
D       10.2.2.0 [90/158720] via 192.168.3.2, 00:14:49, FastEthernet0/1
D       10.3.3.0 [90/156160] via 192.168.3.2, 00:59:42, FastEthernet0/1
C     192.168.1.0/24 is directly connected, FastEthernet0/0
C     192.168.3.0/24 is directly connected, FastEthernet0/1
D     192.168.4.0/24 [90/30720] via 192.168.3.2, 00:59:42, FastEthernet0/1
```

Для теста отказоустойчивости, запустим пинг на 1000 пакетов и в этот момент  
поотключаем основной канал через 192.168.3.2:

```
Router0#ping
Protocol [ip]:
Target IP address: 10.2.2.1
Repeat count [5]: 1000
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 1000, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to
down

%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.3.2 (FastEthernet0/1) is down:
interface down
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to
up

%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.3.2 (FastEthernet0/1) is up: new
adjacency
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (403/403), round-trip min/avg/max = 0/0/4 ms
```

Как видно, линк падал, но пакеты не прекращали ходить. Тем самым  
резервирование отработывало. Это одна из фишек EIGRP.  
Вторая фишка — это неэквивалентная балансировка. Как помним, обычная

балансировка работает, если 2 маршрута приходят с абсолютно одинаковой метрикой. EIGRP же умеет балансировать маршрутами с разной метрикой. Проверим на существующей топологии. На Router0 имеем следующее:

### Topology Table:

```
Router0#show ip eigrp topology
IP-EIGRP Topology Table for AS 1/ID(10.1.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 10.1.1.0/24, 1 successors, FD is 128256
    via Connected, Loopback1
P 10.2.2.0/24, 1 successors, FD is 158720
    via 192.168.3.2 (158720/156160), FastEthernet0/1
    via 192.168.1.2 (179200/128256), FastEthernet0/0
P 10.3.3.0/24, 1 successors, FD is 156160
    via 192.168.3.2 (156160/128256), FastEthernet0/1
P 192.168.1.0/24, 1 successors, FD is 51200
    via Connected, FastEthernet0/0
P 192.168.3.0/24, 1 successors, FD is 28160
    via Connected, FastEthernet0/1
P 192.168.4.0/24, 1 successors, FD is 30720
    via 192.168.3.2 (30720/28160), FastEthernet0/1
    via 192.168.1.2 (53760/28160), FastEthernet0/0
```

### Route Table:

```
Router0#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 3 subnets
C    10.1.1.0 is directly connected, Loopback1
D    10.2.2.0 [90/158720] via 192.168.3.2, 00:02:57, FastEthernet0/1
D    10.3.3.0 [90/156160] via 192.168.3.2, 00:04:45, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.3.0/24 is directly connected, FastEthernet0/1
D    192.168.4.0/24 [90/30720] via 192.168.3.2, 00:04:45, FastEthernet0/1
```

Gateway of last resort is not set

```
10.0.0.0/24 is subnetted, 3 subnets
C    10.1.1.0 is directly connected, Loopback1
D    10.2.2.0 [90/158720] via 192.168.3.2, 00:02:57, FastEthernet0/1
D    10.3.3.0 [90/156160] via 192.168.3.2, 00:04:45, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.3.0/24 is directly connected, FastEthernet0/1
D    192.168.4.0/24 [90/30720] via 192.168.3.2, 00:04:45, FastEthernet0/1
```

То есть сейчас мы имеем два маршрута до 10.2.2.0/24, но используем всего один (наилучший, исходя из метрики). Чтобы правило заработало, нужно изменить **множитель метрики** (или с англ. **variance**).

Правило его работы следующее:

FD Feasible Successor < FD Successor

. Иначе говоря стоимость полного пути запасного маршрута должна быть «искусственно» меньше основного.

Сейчас ситуация следующая:

```
P 10.2.2.0/24, 1 successors, FD is 158720
    via 192.168.3.2 (158720/156160), FastEthernet0/1
    via 192.168.1.2 (179200/128256), FastEthernet0/0
```

Значит нужно метрику 158720 умножить настолько, чтобы она стала больше 179200. Умножать можно только на целое число, поэтому выберем 2.

```
Router0(config)#router eigrp 1
Router0(config-router)#variance 2
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.1.2 (FastEthernet0/0) is up: new adjacency
```

```
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.3.2 (FastEthernet0/1) is up: new adjacency
```

В итоге имеем:

```
Router0#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

Gateway of last resort is not set

```
10.0.0.0/24 is subnetted, 3 subnets
C      10.1.1.0 is directly connected, Loopback1
D      10.2.2.0 [90/158720] via 192.168.3.2, 00:02:31, FastEthernet0/1
        [90/179200] via 192.168.1.2, 00:02:31, FastEthernet0/0
D      10.3.3.0 [90/156160] via 192.168.3.2, 00:02:31, FastEthernet0/1
C      192.168.1.0/24 is directly connected, FastEthernet0/0
C      192.168.3.0/24 is directly connected, FastEthernet0/1
D      192.168.4.0/24 [90/30720] via 192.168.3.2, 00:02:31, FastEthernet0/1
        [90/53760] via 192.168.1.2, 00:02:31, FastEthernet0/0
```

Оба маршрута попали в таблицу маршрутизации. Теперь проверим, что балансировка действительно работает:

```
Router0#traceroute 10.2.2.1
Type escape sequence to abort.
Tracing the route to 10.2.2.1
```

```
 1  192.168.3.2      0 msec    0 msec    0 msec
```

```
Router0#traceroute 10.2.2.1
Type escape sequence to abort.
Tracing the route to 10.2.2.1
```

```
 1  192.168.1.2      0 msec    0 msec    0 msec
```

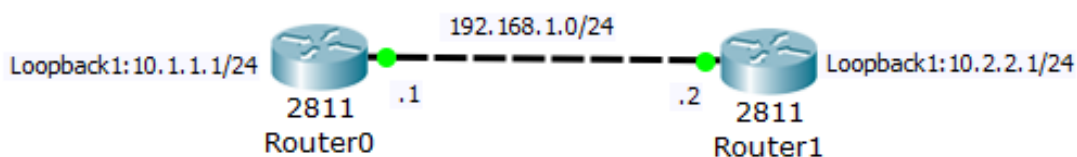
Балансировка работает.

Ссылка на собранную EIGRP топологию.

И ссылка на топологию с измененной скоростью и балансировкой. Если EIGRP не совсем уложился в голове (а это нормальное явление, если изучаете его впервые), то лучше самому собрать топологию, ориентируясь на статью.

Теперь рассмотрим, как происходит соседство в режиме дебага. Если вы дошли сюда с самой первой статьи и принцип хождения пакетов понятен, то лучше уже учиться со включенным дебагом. В рабочих условиях не будет такого инструмента, чтобы красиво смотреть на пакеты и придется пользоваться другими методами. К счастью, если это циска — то решение с дебагом отличное. Единственное — важно включать не все режимы, а только необходимые. Можно, конечно, отзеркалировать порт и просниффать через wireshark. Но не всегда есть физический доступ к железу.

Итак, топология:



Я просто удалил Router2, отключил интерфейсы, которые были соединены с ним и удалил анонсы маршрутов из EIGRP.

Теперь включаю дебаг на Router0 и наблюдаю:

```

Router0#debug eigrp fsm
EIGRP FSM Events/Actions debugging is on
DUAL: rcvupdate: 192.168.1.0/24 via Connected metric 28160/0 -- connected маршрут.
AD=0 (так как ему он пришел не от соседа. А вот его цена интерфейса 28160.

DUAL: Find FS for dest: 192.168.1.0/24. FD is 4294967295, RD is 4294967295

DUAL: RT installed 192.168.1.0/24 via 0.0.0.0 -- маршрут 192.168.1.0/24 заносится
в таблицу, как connected (то есть на себя).
DUAL: Send update about 192.168.1.0/24. Reason: metric chg -- отправляет
измененную метрику.

DUAL: Send update about 192.168.1.0/24. Reason: new if -- отправляет информацию,
что появился новый интерфейс

DUAL: rcvupdate: 10.1.1.0/24 via Connected metric 128256/0 -- та же история с
Loopback

DUAL: Find FS for dest: 10.1.1.0/24. FD is 128256, RD is 0

DUAL: Send update about 10.1.1.0/24. Reason: new if

%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.1.2 (FastEthernet0/0) is up: new
adjacency

DUAL: rcvupdate: 10.2.2.0/24 via 192.168.1.2 metric 156160/128256 - получает
маршрут с AD и накладывает свою метрику.

DUAL: Find FS for dest: 10.2.2.0/24. FD is 4294967295, RD is 4294967295

DUAL: RT installed 10.2.2.0/24 via 192.168.1.2 -- устанавливает маршрут
10.2.2.0/24 через соседа 192.168.1.2
DUAL: Send update about 10.2.2.0/24. Reason: metric chg

DUAL: rcvupdate: 10.1.1.0/24 via 192.168.1.2 metric 4294967295/4294967295

DUAL: Find FS for dest: 10.1.1.0/24. FD is 128256, RD is 0

```

И еще, что стоит упомянуть — это типы EIGRP сообщений. Их 5:

- 1) **Hello** — эти пакеты отправляются на мультикастовый адрес 224.0.0.10 ближайшим соседям. Подтверждения в ответ не требуют. Нужны только для идентификации и своего рода keeralive механизмом.
- 2) **Update** — содержат маршрутную информацию. Как только обнаруживаются соседи, маршрутизатор сразу отправляет им данный пакет. После чего соседи заполняют таблицу EIGRP топологии. Может отправляться по мультикастовому адресу или юникастовому. Эти пакеты требуют ответа.
- 3) **Query** — пакет запроса потерянного маршрута. То есть когда маршрутизатор теряет запись об этом маршруте и не имеет запасного пути к нему. Может отправляться одному через unicast или группе соседей через multicast.
- 4) **Reply** — ответ на **Query-запрос**. Данный пакет всегда отправляется на unicast-

адрес (то есть тому, кто его запросил). Требуется подтверждения.

5) **ACK** — используется для подтверждения **Update**, **Query** и **Reply** пакетов. Всегда отправляется на unicast-адрес.

Помните топологию EIGRP с множеством кодов? Так вот эти коды и отображают состояние и отправляемое сообщение на каждый из маршрутов. Вот так в принципе работает EIGRP.

Переходим к последнему протоколу — это **OSPF** (англ. Open Shortest Path First). Относится он к группе **link state** или **протокол состояния канала**. Если RIP с EIGRP работали более-менее похоже, то OSPF работает совершенно по другому. Если дистанционно-векторные протоколы сравнивались с дорожными указателями, то протоколы состояния канала можно сравнить с дорожным навигатором. В этом как раз и отличие. OSPF сначала строит карту сети, а потом выбирает лучший путь. Да, таким образом он более ресурсозатратный протокол, нежели его коллеги, но на текущий момент это не столь критично, как было лет 25-30 назад.

Итак. Почему **Link-State**:

- 1) **Link** — интерфейс маршрутизатора.
- 2) **State** — его состояние и как он подключен к соседям.

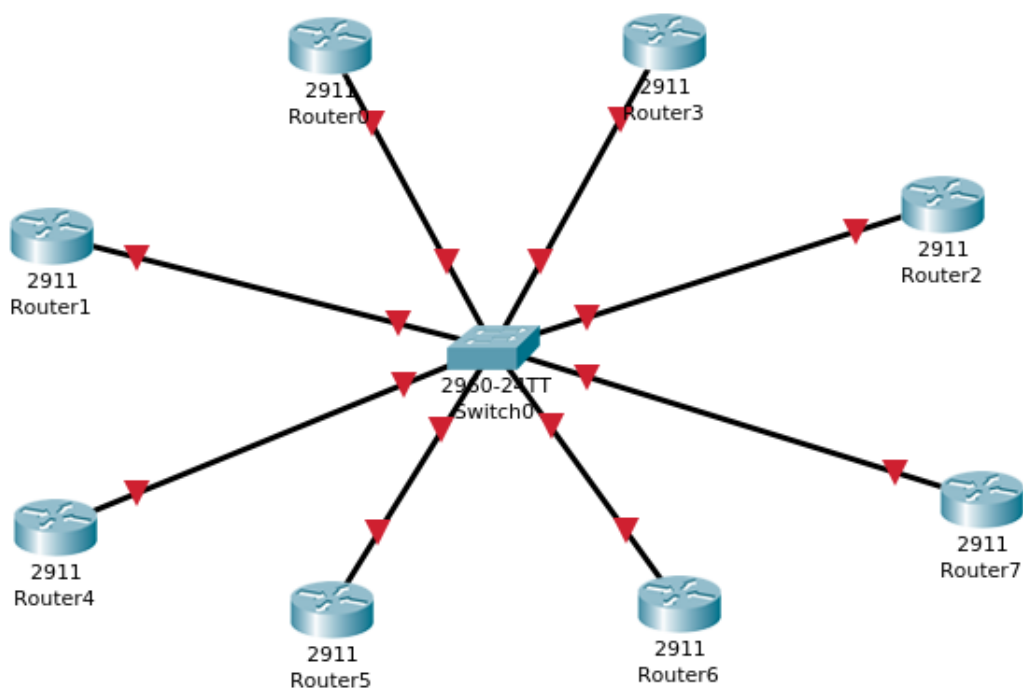
Опиерирует они:

- 1) **LSA** (от англ. link-state advertisements) — это как раз таки объявления, которыми они обмениваются между собой. Ниже их разберем.
- 2) **LSDB** (от англ. link-state database) — как раз эти LSA формируют базу. Или ту самую карту сети.

Тут встает вопрос. А хорошо ли то, что каждый маршрутизатор обменивается своей информацией с каждым соседом?!

Представим топологию:





Что если каждый маршрутизатор будет отсылать маршрут каждому из своих соседей?! Мы получим огромный флуд трафика. При этом один и тот же анонс будет зеркалироваться... Подумали в свое время инженеры и решили, что эффективнее держать одного маршрутизатора, которому все остальные будут отсылать уведомления, а он будет ответственным за весь флуд. Тем самым смысл тот же, только трафика будет меньше. А чтобы не случилось ситуации, когда «главный» умирает и вся сеть останавливается, придумали держать запасного маршрутизатора, который, в случае «смерти» основного, возьмет его обязанности на себя.

Маршрутизатор, который берет роль основного на себя, называется **DR** (от англ. **Designated Router**), а запасной маршрутизатор называется **BDR** (от англ. **Backup Designated Router**).

Такая логика работает автоматически в сетях с множественным доступом, которой и является Ethernet. Если у вас сеть точка-точка (пусть даже Ethernet и соединены друг с другом напрямую), то DR и BDR выбирать не обязательно, так как всего 2 участника (но в Ethernet они все же будут выбраны). Но никто не мешает вам изменить логику OSPF и прописать каждого соседа вручную. Только зачем?)

Так вот после того, как LSDB заполнена, каждый маршрутизатор начинает высчитывать самый выгодный маршрут до каждой подсети. Использует он для этого алгоритм **SPF** (от англ. **Shortest Path First**). Лучший подсчитанный маршрут попадает в таблицу маршрутизации.

Давайте перейдем к практике и по ходу разбираться.

Есть схема:

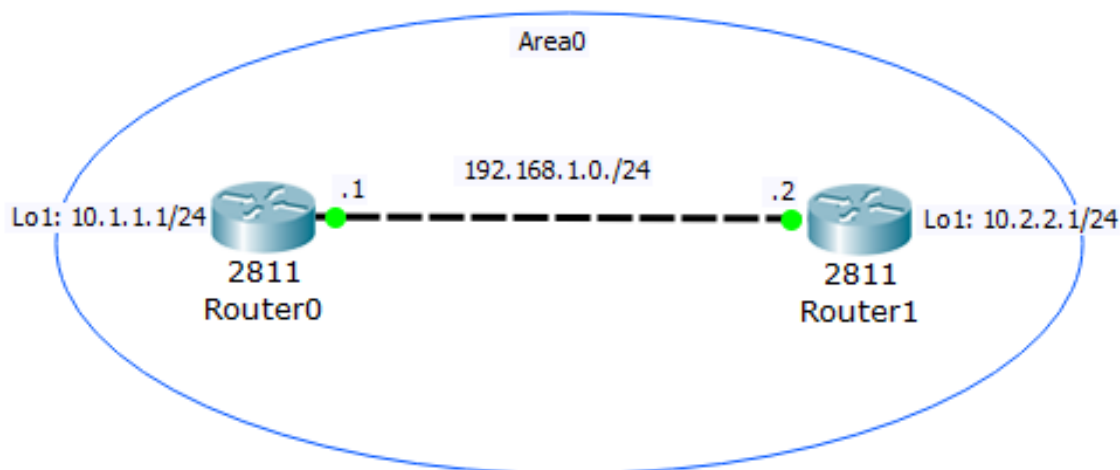


Схема самая простая. Единственное, что новое — это очерчена зона. Я специально ее нарисовал. Дело в том, что OSPF обязательно нужно указывать зону для которой включается протокол. Это сделано для того, чтобы снизить нагрузку в расчетах пути. Как я говорил ранее, протокол появился достаточно давно и для того времени производительность играла большую роль. Сейчас тоже принято делить на зоны. Но сейчас это делается для снижения не нужного трафика.

Зоной по-умолчанию всегда выбирается нулевая. Ее еще называют backbone зоной и не с проста. Если у вас в сети много различных зон, то соединены они должны быть через нулевую. То есть нельзя перейти из 11-ой в 25-ую зону напрямую.

Обязательно нужно пройти через нулевую, а из нулевой проследовать в требуемую. Единственный случай, когда можно пройти из зоны в зоны, миновав нулевую — это использование **Virtual Link**. Почитать о ней можно [здесь](#).

Сейчас у нас 2 маршрутизатора в нулевой зоне. На маршрутизаторах настроены IP-адреса и создан Loopback. Ниже под спойлерами конфиги.

## Router0

## Router1

Теперь включаю OSPF для интерфейсов FastEthernet0/0 и Loopback1 обоих роутеров:

```
router ospf 1
 network 192.168.1.0 0.0.0.255 area 0
 network 10.1.1.0 0.0.0.255 area 0
!
```

```
router ospf 1
 network 10.2.2.0 0.0.0.255 area 0
 network 192.168.1.0 0.0.0.255 area 0
!
```

Конфигурация простая. Указывается подсеть, wildcard маска и номер зоны. После видим сообщения:

На Router0:

```
Router0#
00:56:22: %OSPF-5-ADJCHG: Process 1, Nbr 10.2.2.1 on FastEthernet0/0 from LOADING
to FULL, Loading Done
```

На Router1:

```
Router1(config-router)#
00:56:21: %OSPF-5-ADJCHG: Process 1, Nbr 10.1.1.1 on FastEthernet0/0 from LOADING
to FULL, Loading Done
```

Соседство, судя по сообщению установилось. Но, если обратить внимание, то почему то соседство выбрано между адресами из Loopback интерфейсов. Это на самом деле не адрес, а идентификатор или **Router ID**. Если в самом процессе он явно не указывается, то выбирается автоматически. Если настроены Loopback интерфейсы, то выбирается наибольший IP-адрес из них. Если Loopback не настроены, то выбирается наибольший IP-адрес из обычного физического интерфейса. У нас Loopback был настроен, а значит он и будет выбран RID. Так как процессы на обоих роутерах одинаковые, покажу на примере Router0: Так как соседство установлено, посмотрим список соседей.

```
Router0#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.2.2.1	1	FULL/BDR	00:00:38	192.168.1.2	FastEthernet0/0

Видим 10.2.2.1 (Router1). Статус Full (чуть ниже расскажу и об этом), роль BDR (то есть Router0 выбран DR). Его физический IP-адрес и с какого интерфейса доступен. Теперь посмотрим на базу данных OSPF:

```
Router0#show ip ospf database
      OSPF Router with ID (10.1.1.1) (Process ID 1)
```

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
10.1.1.1	10.1.1.1	259	0x80000004	0x0047fb	2
10.2.2.1	10.2.2.1	259	0x80000004	0x00b586	2

Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
192.168.1.1	10.1.1.1	259	0x80000002	0x00e9ca

Подробное ее содержание изучается в курсе CCNP Route, поэтому расскажу вкратце. Есть несколько типов LSA-сообщений. В нашей схеме используются только Type1 (Router) и Type2(Network). Первое генерится каждым маршрутизатором в

пределах зоны и дальше зоны не уходит. Второй тип генерируется DR-ом и содержит адрес DR и инфу о всех маршрутизаторах в зоне.

Например, так выглядит Type1 с консоли Router0:

```
Router0#show ip ospf database router
```

```
OSPF Router with ID (10.1.1.1) (Process ID 1)
```

```
Router Link States (Area 0)
```

```
LS age: 665
```

```
Options: (No TOS-capability, DC)
```

```
LS Type: Router Links
```

```
Link State ID: 10.1.1.1
```

```
Advertising Router: 10.1.1.1
```

```
LS Seq Number: 80000004
```

```
Checksum: 0x47fb
```

```
Length: 48
```

```
Number of Links: 2
```

```
Link connected to: a Transit Network
```

```
(Link ID) Designated Router address: 192.168.1.1
```

```
(Link Data) Router Interface address: 192.168.1.1
```

```
Number of TOS metrics: 0
```

```
TOS 0 Metrics: 1
```

```
Link connected to: a Stub Network
```

```
(Link ID) Network/subnet number: 10.1.1.1
```

```
(Link Data) Network Mask: 255.255.255.255
```

```
Number of TOS metrics: 0
```

```
TOS 0 Metrics: 1
```

```
LS age: 665
```

```
Options: (No TOS-capability, DC)
```

```
LS Type: Router Links
```

```
Link State ID: 10.2.2.1
```

```
Advertising Router: 10.2.2.1
```

```
LS Seq Number: 80000004
```

```
Checksum: 0xb586
```

```
Length: 48
```

```
Number of Links: 2
```

```
Link connected to: a Stub Network
```

```
(Link ID) Network/subnet number: 10.2.2.1
```

```
(Link Data) Network Mask: 255.255.255.255
```

```
Number of TOS metrics: 0
```

```
TOS 0 Metrics: 1
```

```
Link connected to: a Transit Network
```

```
(Link ID) Designated Router address: 192.168.1.1
```

```
(Link Data) Router Interface address: 192.168.1.2
```

```
Number of TOS metrics: 0
```

```
TOS 0 Metrics: 1
```

То есть LSA каждого маршрутизатора, в которых он сообщает о своих сетях.

А вот так Type2:

```
Router0#show ip ospf database network

        OSPF Router with ID (10.1.1.1) (Process ID 1)

        Net Link States (Area 0)

Routing Bit Set on this LSA
LS age: 686
Options: (No TOS-capability, DC)
LS Type: Network Links
Link State ID: 192.168.1.1 (address of Designated Router)
Advertising Router: 10.1.1.1
LS Seq Number: 80000002
Checksum: 0xe9ca
Length: 32
Network Mask: /24
    Attached Router: 10.2.2.1
    Attached Router: 10.1.1.1
```

То есть как раз адрес DR (кому отправлять свои LSA и список маршрутизаторов в зоне).

И теперь можно посмотреть на таблицу маршрутизации:

```
Router0#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/24 is directly connected, Loopback1
O       10.2.2.1/32 [110/2] via 192.168.1.2, 00:48:02, FastEthernet0/0
C       192.168.1.0/24 is directly connected, FastEthernet0/0
```

Gateway of last resort is not set

```
    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/24 is directly connected, Loopback1
O       10.2.2.1/32 [110/2] via 192.168.1.2, 00:48:02, FastEthernet0/0
C       192.168.1.0/24 is directly connected, FastEthernet0/0
```

Видим букву О (это значит, что маршрут получен из той же зоны, что и данный маршрутизатор). Можно заметить, что в таблицу записан с маской /32. Это потому что адрес из Loopback интерфейса и обычно такие адреса служат для всяких RID и прочих идентификаторов. Это не подсеть, а значит нет смысла анонсировать с тем же префиксом, что и сам интерфейс. Но такое поведение работает не на всех цисках. Поэтому тут надо быть внимательнее. Рядом видим привычную административную дистанцию (у циски это 110, но можно поменять) и метрику, которая равна 2-ум. Здесь метрика считается проще, чем у EIGRP. Формула:

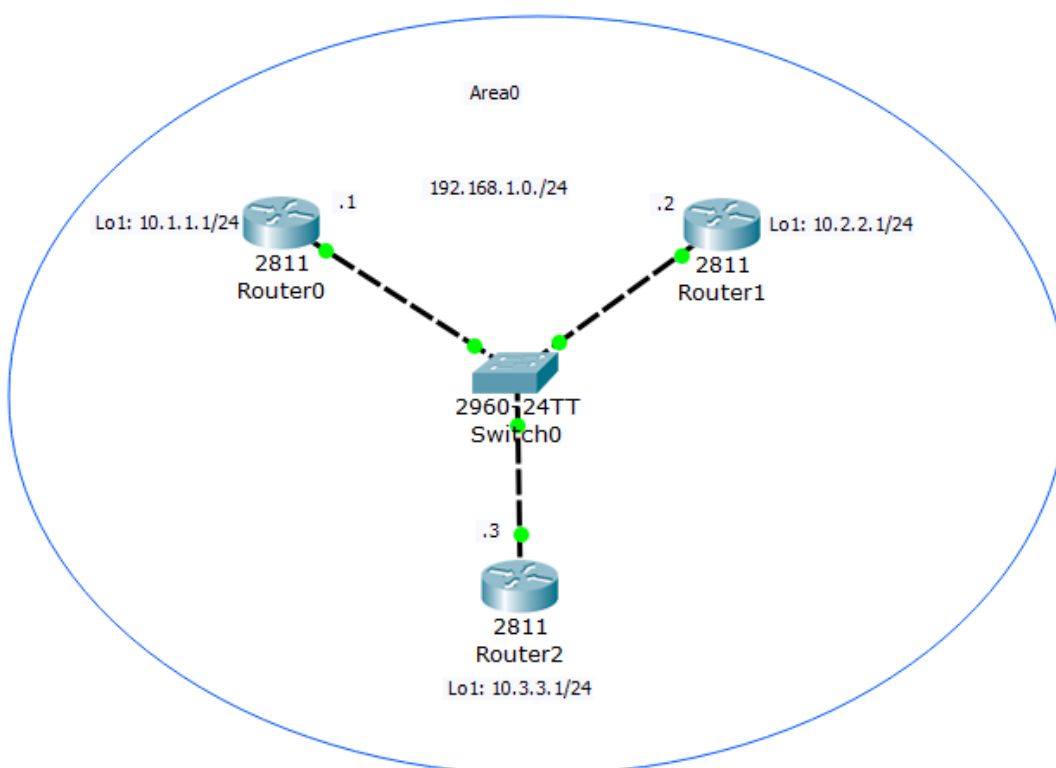
$Cost = Reference\ Bandwidth / Interface\ Bandwidth$

**Reference Bandwidth** — это некое заданное число (здесь по-умолчанию 100). Оно прошито внутри логики и меняется командой *auto-cost reference-bandwidth число* в настройках OSPF процесса.

А вот **Interface Bandwidth** берется ровно такое, какая пропускная способность у интерфейса. На нашем интерфейсе это 100, поэтому метрика = 1. Так как Router1 анонсирует уже с метрикой 1, то накладывая свою стоимость в 1-цу, получаем 2.

OSPF для меня в свое время менялся в сложности понимания. Сначала казалось все легко, включил и все работает. Дальше, когда начинаешь углубляться в структуру LSA и как происходит формирование и расчет, теряешься. А после понимания, он снова становится легким. Его понимание приходит только после практики. Поэтому можете потренироваться на этой топологии. [Ссылка на нее](#). Пару слов по балансировке. Здесь она строго эквивалентная. Нельзя делать, как в EIGRP. Всего в кандидатах может быть до 16 маршрутов, но в таблицу попадут только 4.

Если предыдущая схема понятна, то двигаемся дальше. Добавим еще один маршрутизатор и соединим их, при помощи коммутатора:



Я взял за основу предыдущую, адреса все те же самые, включен OSPF. На Router2

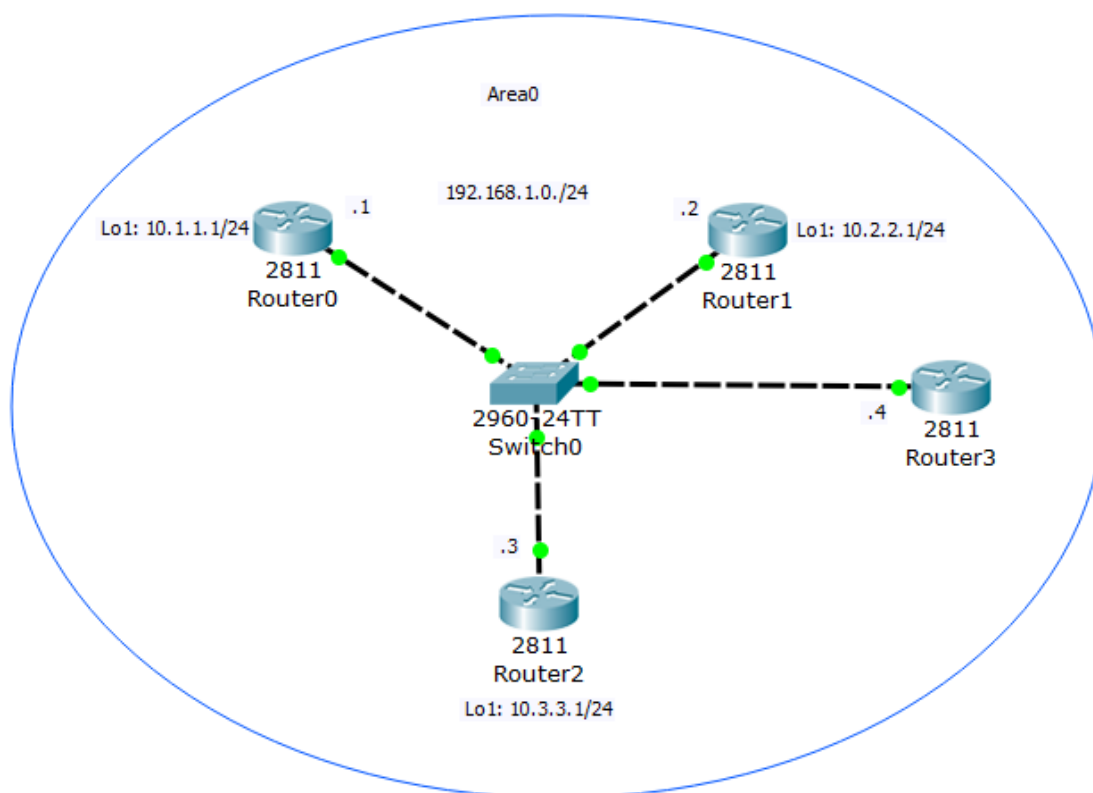
также включен OSPF и настроены адреса согласно схеме. Теперь смотрим, что произошло со стороны того же Router0. Ввожу команду просмотра соседей:

```
Router0#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.2.2.1	1	FULL/BDR	00:00:37	192.168.1.2	FastEthernet0/0
10.3.3.1	1	FULL/DROTHER	00:00:36	192.168.1.3	FastEthernet0/0

И вижу нового соседа, но с пометкой DROTHER. Это значит, что маршрутизатор Router2 (новый) не является DR или BDR. Обратите внимание, что DR (Router0) установил Full соседство со всеми соседями.

Ввожу нового игрока на поле — Router3:



Единственное, что у него настроено — это IP-адрес 192.168.1.4/24 на FastEthernet 0/0 и включен OSPF. Он тут для наглядности.

Со стороны Router0:

```
Router0#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.2.2.1	1	FULL/BDR	00:00:31	192.168.1.2	FastEthernet0/0
10.3.3.1	1	FULL/DROTHER	00:00:31	192.168.1.3	FastEthernet0/0
192.168.1.4	1	FULL/DROTHER	00:00:30	192.168.1.4	FastEthernet0/0

Так как нет адреса на Loopback интерфейсе и не задан вручную RID, выбран адрес с физического интерфейса. А теперь переходим к Router2 и смотрим на его список соседей:

```
Router2#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.2.2.1	1	FULL/BDR	00:00:32	192.168.1.2	FastEthernet0/0
10.1.1.1	1	FULL/DR	00:00:32	192.168.1.1	FastEthernet0/0
192.168.1.4	1	2WAY/DROTHER	00:00:31	192.168.1.4	FastEthernet0/0

Видим, что с ним у него не Full отношения, а 2Way. Почему не Full? На этом остановлюсь и расскажу про процесс установления соседства. В хорошо работающей сети процесс соседства происходит настолько быстро, что все состояния вы не успеете увидеть. Я только опишу их, для общего понимания:

- 1) **Down** — это самый старт, когда маршрутизатор еще не предпринял попытку соседства и ничего в ответ не получает.
- 2) **Init** — маршрутизатор переходит в это состояние после отправки Hello-сообщения, до момента получения ответа.
- 3) **2-WAY** — маршрутизатор переходит в это состояние, если получает ответный Hello и видит внутри него свой RID. Это как раз момент установления соседства. В сетях множественного доступа (типа Ethernet) это состояние конечное между «не DR/BDR» маршрутизаторами. Как раз в этом состоянии осталось соседство между Router2 и Router3.
- 4) **ExStart** — это состояние выбора DR/BDR. Маршрутизатор с наилучшим RID берет на себя эту роль. Он начинает первым процесс обновления LSDB у всех соседей.
- 5) **Exchange** — состояние, в котором маршрутизаторы отправляют друг другу состояние своих LSDB.
- 6) **Loading** — если маршрутизатор видит, что в присланном сообщении есть подсеть, о которой он не знает, он запрашивает информацию о ней. И вот пока запрашиваемая инфа не дойдет до него, он будет висеть в этом состоянии.
- 7) **Full** — конечное состояние. Наступает оно в том случае, когда LSDB между соседями синхронизировано.

Стоит упомянуть, что в OSPF есть таймеры соседства. Нужно для того, чтобы узнать жив ли сосед или пора исключить его. Поэтому каждые 10 секунд маршрутизаторы отсылают друг другу Hello-пакеты, чтобы подтвердить свое существование. Если в течении 40 секунд от соседа ничего не поступало, соседство с ним разрывается.

Посмотреть на таймеры и другие параметры интерфейса, на котором включен OSPF, можно командой *show ip ospf interface*:

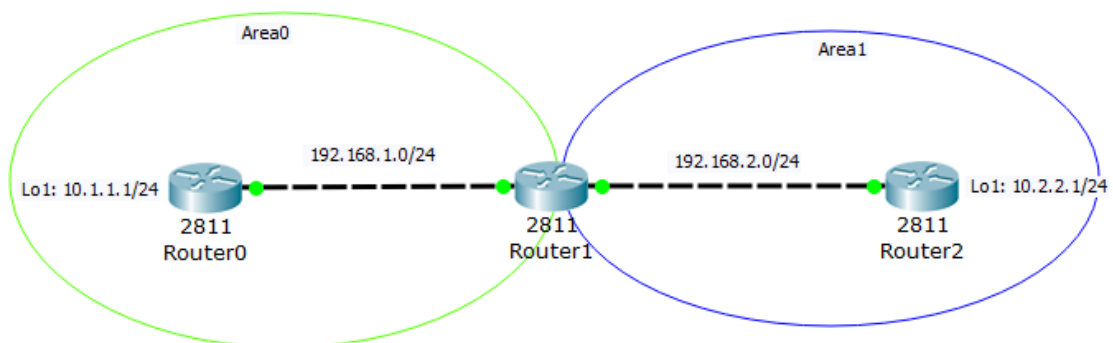


```
Router0#show ip ospf interface
```

```
Loopback1 is up, line protocol is up
  Internet address is 10.1.1.1/24, Area 0
  Process ID 1, Router ID 10.1.1.1, Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host
FastEthernet0/0 is up, line protocol is up
  Internet address is 192.168.1.1/24, Area 0
  Process ID 1, Router ID 10.1.1.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.1.1.1, Interface address 192.168.1.1
  Backup Designated Router (ID) 10.2.2.1, Interface address 192.168.1.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:00
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 3, Adjacent neighbor count is 3
    Adjacent with neighbor 10.2.2.1 (Backup Designated Router)
    Adjacent with neighbor 10.3.3.1
    Adjacent with neighbor 192.168.1.4
  Suppress hello for 0 neighbor(s)
```

Если интересно, как происходит весь процесс установления соседства, откройте топологию по [ссылке](#). Переключитесь в режим симуляции и перезагрузите один из маршрутизаторов. Все сразу особого смысла нет. Скорее быстрее загрузит CPT, нежели получится разобраться.

И последнее, что стоит рассмотреть из раздела OSPF — это Multiarea OSPF (или многозонный OSPF).



Теперь есть 3 маршрутизатора. Router0 находится в нулевой зоне, Router1 в 0-ой и 1-ой зоне и Router2 в 1-ой зоне. Конфигурация проста. Я оставлю ее под спойлерами:

## Router0

## Router1

## Router2

Отличие от предыдущих схем только в том, что для Router1 и Router2 добавляется другой номер зоны, при включении.

Если посмотреть таблицу маршрутизации с Router0:

```
Router0#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/24 is directly connected, Loopback1
O IA    10.2.2.1/32 [110/3] via 192.168.1.1, 00:09:27, FastEthernet0/0
C       192.168.1.0/24 is directly connected, FastEthernet0/0
O IA    192.168.2.0/24 [110/2] via 192.168.1.1, 00:43:49, FastEthernet0/0
```

То добавились маршруты OIA (или OSPF inter area). То есть маршрут из другой зоны. Если посмотреть базу:

```
Router0#show ip ospf database
```

```
OSPF Router with ID (10.1.1.1) (Process ID 1)
```

### Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
10.1.1.1	10.1.1.1	861	0x80000006	0x00c679	2
192.168.2.1	192.168.2.1	861	0x80000006	0x00dbc3	1

### Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
192.168.1.1	192.168.2.1	953	0x80000002	0x009931

### Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
192.168.2.0	192.168.2.1	947	0x80000003	0x00a7dc
10.2.2.1	192.168.2.1	851	0x80000004	0x00bc22

Здесь появился Summary LSA или Type3. Его генерирует маршрутизатор, который находится на границе двух зон. Такой маршрутизатор называют пограничным или ABR (от англ. Area Border Gateway).

Если посмотреть на него поближе:

```
Router0#show ip ospf database summary
```

```
OSPF Router with ID (10.1.1.1) (Process ID 1)
```

```
Summary Net Link States (Area 0)
```

```
LS age: 1146
Options: (No TOS-capability, DC, Upward)
LS Type: Summary Links(Network)
Link State ID: 192.168.2.0 (summary Network Number)
Advertising Router: 192.168.2.1
LS Seq Number: 80000003
Checksum: 0xa7dc
Length: 28
Network Mask: /24
    TOS: 0 Metric: 1
```

```
LS age: 1050
Options: (No TOS-capability, DC, Upward)
LS Type: Summary Links(Network)
Link State ID: 10.2.2.1 (summary Network Number)
Advertising Router: 192.168.2.1
LS Seq Number: 80000004
Checksum: 0xbc22
Length: 28
Network Mask: /32
    TOS: 0 Metric: 2
```

То можно заметить, что анонсирует его 192.168.2.1 (это RID Router1).

Если же посмотреть на таблицу маршрутизации со стороны ABR (т.е. Router1):

```
Router1#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
10.0.0.0/32 is subnetted, 2 subnets
O      10.1.1.1 [110/2] via 192.168.1.2, 00:20:49, FastEthernet0/0
O      10.2.2.1 [110/2] via 192.168.2.2, 00:20:44, FastEthernet0/1
C      192.168.1.0/24 is directly connected, FastEthernet0/0
C      192.168.2.0/24 is directly connected, FastEthernet0/1
```

То для него все маршруты помечены O. Все потому что он находится в обеих зонах

и для него они локальны.

А если посмотреть базу:

```
Router1#show ip ospf database
      OSPF Router with ID (192.168.2.1) (Process ID 1)

      Router Link States (Area 0)

Link ID        ADV Router    Age          Seq#           Checksum Link count
192.168.2.1    192.168.2.1   1326         0x80000006    0x00dbc3  1
10.1.1.1       10.1.1.1      1326         0x80000006    0x00c679  2

      Net Link States (Area 0)

Link ID        ADV Router    Age          Seq#           Checksum
192.168.1.1    192.168.2.1   1417         0x80000002    0x009931

      Summary Net Link States (Area 0)

Link ID        ADV Router    Age          Seq#           Checksum
192.168.2.0    192.168.2.1   1412         0x80000003    0x00a7dc
10.2.2.1       192.168.2.1   1316         0x80000004    0x00bc22

      Router Link States (Area 1)

Link ID        ADV Router    Age          Seq#           Checksum Link count
192.168.2.1    192.168.2.1   1326         0x80000005    0x00f3aa  1
10.2.2.1       10.2.2.1      1326         0x80000005    0x006ccc  2

      Net Link States (Area 1)

Link ID        ADV Router    Age          Seq#           Checksum
192.168.2.1    192.168.2.1   1371         0x80000002    0x0049d0

      Summary Net Link States (Area 1)

Link ID        ADV Router    Age          Seq#           Checksum
192.168.1.0    192.168.2.1   1413         0x80000003    0x00b2d2
10.1.1.1       192.168.2.1   1322         0x80000005    0x00d10e
```

То тут их больше. Все потому, что у него представлены эти LSA на каждую зону, а также он генерирует Type3 в обе стороны. Для самостоятельного ознакомления лабу можно скачать по данной [ссылке](#).

Таким образом OSPF можно делить на зоны. То есть маршрутизатор видит соседей в своей зоне и просчитывает лучший путь сам. А вот межзонные маршруты (Type3) диктует ABR. Поэтому на границу чаще ставят производительные маршрутизаторы. На самом деле EIGRP и OSPF уж очень много всего умеют. И заслуживают отдельных статей. Более подробно они разбираются уже в топиках CCNP. Так что для основ достаточно.

В итоге мы разобрались с маршрутизацией и встает вопрос: что использовать?

Однозначного ответа тут нет. Если у вас вся сеть построена на цисках, то можно выбирать EIGRP. Если у вас сеть мультивендорная, то тут однозначно OSPF. Да, циска вроде как открыла стандарт, но относительно старые железки (не циски) не получают поддержку этого протокола, да и не на всех новых его внедряют. Более того, могу сказать, что даже в сетях построенных исключительно на цисках, выбирают

OSPF. Аргументируя это тем, что OSPF более гибок в настройке, нежели EIGRP. Да и нельзя быть уверенным, что в какой то момент придется ставить сетевое устройство другого вендора. А значит внедрение такого устройства пройдет безболезненно и без перенастройки всей сети.

Подводя итоги, можно сказать, что это самая долгая статья из всех, что я писал. Все потому, что писал я ее больше 2-х лет. Постоянно что-то стопорило ее написание, а когда садился, то не мог сконцентрироваться и написать больше 2-х предложений. Но теперь она написана и можно спокойно выдохнуть. Ее как раз не хватало для основ компьютерных сетей, ведь предыдущие статьи концентрировались в большинстве на L2 уровне. Столь длительное написание привело к тому, что циска уже меняет программу своего экзамена. А значит некоторые темы, которые я хотел далее осветить, уже не актуальны. Поэтому я убери из содержания будущие темы и буду выкладывать статьи, исходя из актуальности.

Спасибо всем, кто ждал статью и интересовался.