

## Лабораторная работа 6 Динамическая маршрутизация на протоколах RIP и EIGRP

**Маршрутизация** - процесс определения в сети наилучшего пути, по которому пакет может достигнуть адресата. *Динамическая маршрутизация* может быть осуществлена с использованием одного и более протоколов (*RIP v2*, *OSPF* и др.).

### Новый термин

**Динамическая маршрутизация** — вид маршрутизации, при котором таблица маршрутизации заполняется и обновляется автоматически при помощи одного или нескольких протоколов маршрутизации (*RIP*, *OSPF*, *EIGRP*, *BGP*).

Каждый *протокол маршрутизации* использует свою систему оценки маршрутов (*метрику*). *Маршрут* к сетям назначения строится на основе таких критериев как

- количество ретрансляционных переходов
- пропускная способность канала связи
- задержки передачи данных
- и др.

Маршрутизаторы обмениваются друг с другом информацией о маршрутах с помощью служебных пакетов по протоколу *UDP*. Такой обмен информации увеличивает наличие дополнительного трафика в сети и нагрузку на эту *сеть*. Возможна также ситуация, при которой таблицы маршрутизации на роутерах не успевают согласоваться между собой, что может повлечь появление ошибочных маршрутов и потерю данных.

Протоколы маршрутизации делятся на три типа:

- Дистанционно векторные протоколы (*RIP*)
- Протоколы с отслеживанием состояния каналов (*OSPF*)
- Смешанные протоколы (*EIGRP*)
- И др.

## Протокол RIP

*RIP* — протокол дистанционно-векторной маршрутизации, использующий для нахождения оптимального пути *алгоритм* Беллмана-Форда. *Алгоритм* маршрутизации *RIP* — один из самых простых протоколов маршрутизации. Каждые 30 секунд он передает в *сеть* свою таблицу маршрутизации. Основное отличие протоколов в том, что *RIPv2* (в отличие от *RIPv1*) может работать по мультикасту, то есть, рассылаясь на мультикаст *адрес*. Максимальное количество "хопов" (шагов до места назначения), разрешенное в *RIP1*, равно 16 (*метрика* 16). Ограничение в 16 хопов не дает применять *RIP* в больших сетях, поэтому протокол наиболее распространен в небольших компьютерных сетях. Вторая версия протокола — протокол *RIP2* была разработана в 1994 году и является улучшенной версией первого. В этом протоколе повышена *безопасность* за счет введения дополнительной маршрутной информации. Принцип дистанционно-векторного протокола: каждый *маршрутизатор*, использующий протокол *RIP* периодически широковещательно рассылает своим соседям специальный пакет-*вектор*, содержащий расстояния (измеряются в метрике) от данного маршрутизатора до всех известных ему сетей. *Маршрутизатор* получивший такой *вектор*, наращивает компоненты вектора на величину расстояния от себя до данного соседа и дополняет *вектор* информацией об известных непосредственно ему самому сетях или сетях, о которых ему сообщили другие маршрутизаторы. Дополненный *вектор маршрутизатор* рассылает всем своим соседям. *Маршрутизатор* выбирает из нескольких альтернативных маршрутов *маршрут* с наименьшим значением метрики, а *маршрутизатор*, передавший информацию о таком маршруте помечается как следующий (*next hop*). Протокол непригоден для работы в больших сетях, так как засоряет *сеть* интенсивным трафиком, а узлы сети оперируют только векторами-расстояний, не имея точной информации о состоянии каналов и топологии сети. Сегодня даже в небольших сетях протокол вытесняется превосходящими его по возможностям протоколами *EIGRP* и *OSPF*.

## Практическая работа 6-1. Настройка протокола RIP версии 2 для сети из шести устройств

Наша задача – настроить маршрутизацию на схеме, представленной на

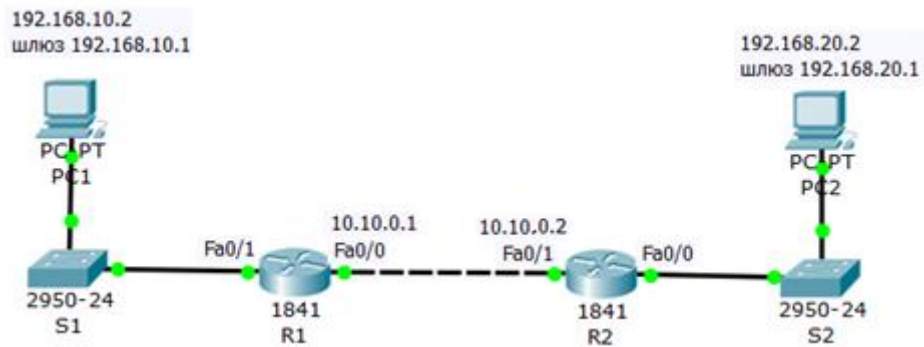


Рис. 6.1. Схема сети

#### Примечание

При настройке сети не забывайте включать порты.

### Настройка протокола RIP на маршрутизаторе R1

Войдите в конфигурации в консоль роутера и выполните следующие настройки

```
R1
Physical Config CLI
IOS Command Line Interface

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#192.168.10.1
Router(config-router)#network 192.168.10.1
Router(config-router)#network 10.10.0.1
Router(config-router)#end
Router#
```

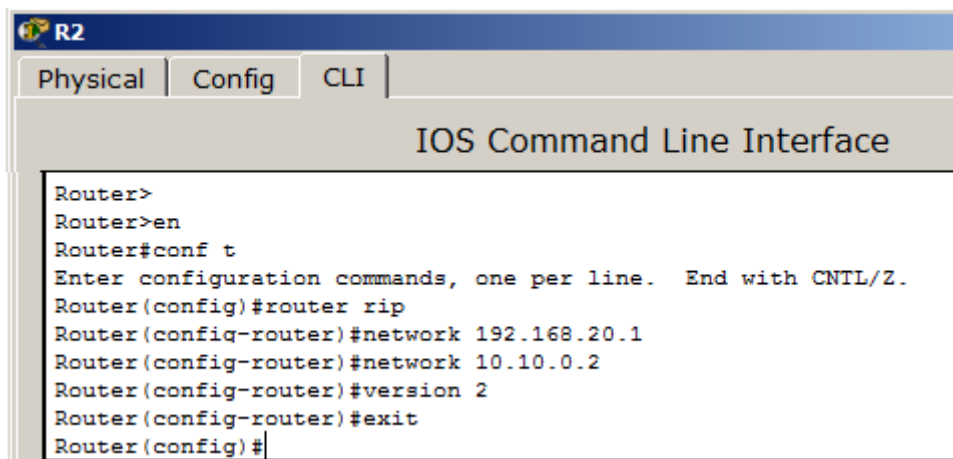
Рис. 6.2. Настройка протокола RIPv2 на маршрутизаторе Router1

#### Примечание

**Router(config)#router rip** (Вход в режим конфигурирования протокола RIP). **Router(config-router)#network 192.168.10.1** (Подключение клиентской сети к роутеру со стороны коммутатора S1). **Router(config-router)#network 192.168.20.1** (Подключение второй сети, то есть сети между роутерами). **Router(config-router)#version 2** (Задание использования второй версии протокол RIP).

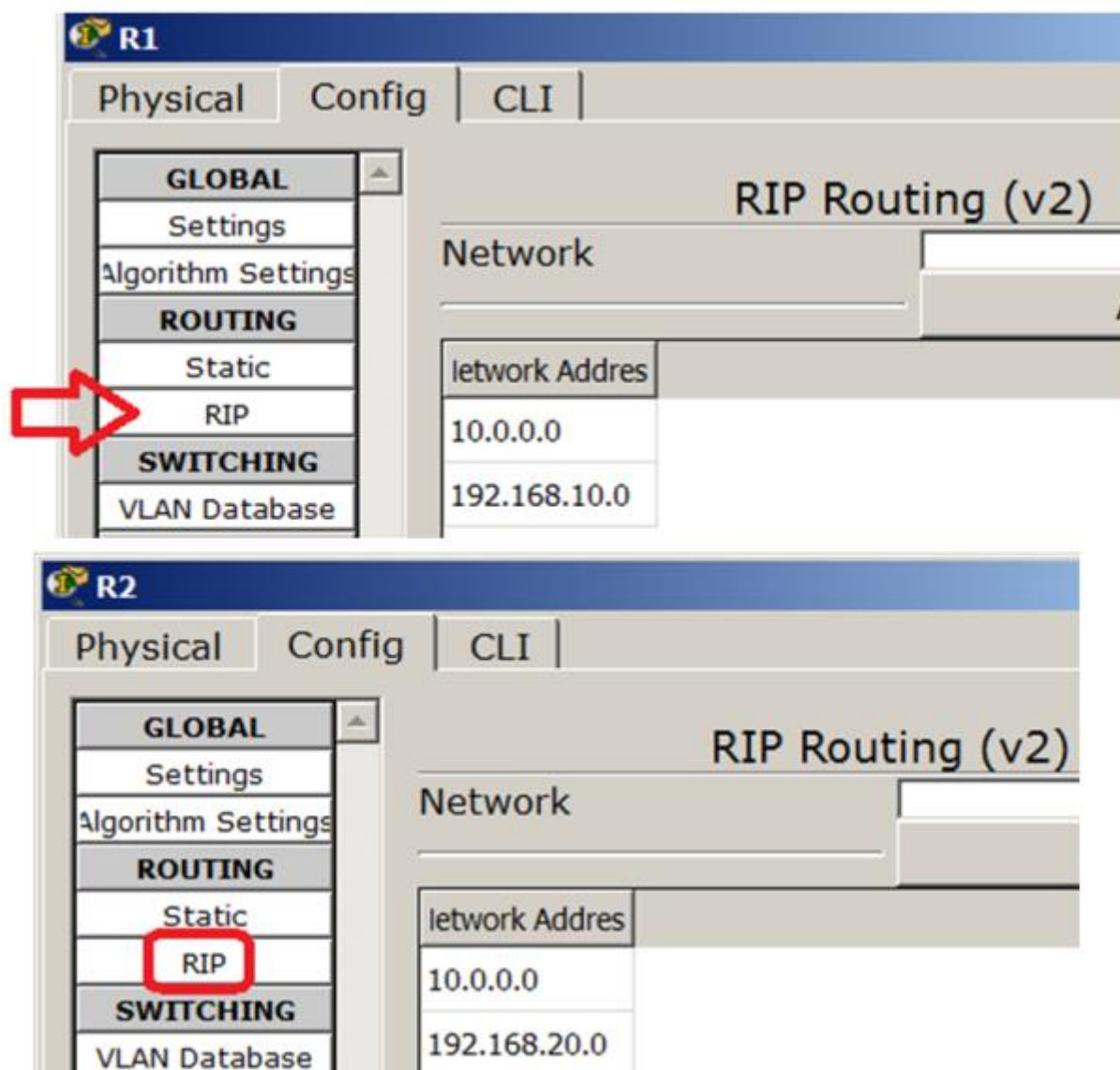
### Настройка протокола RIP на маршрутизаторе R2

Войдите в конфигурации роутера 2 и выполните следующие настройки



**Рис. 6.3.** Настройка протокола RIPv2 на маршрутизаторе R2  
**Проверяем настройки коммутаторов и протокола RIP**

Давайте посмотрим настройки протокола RIPv2 на маршрутизаторах R1 и R2



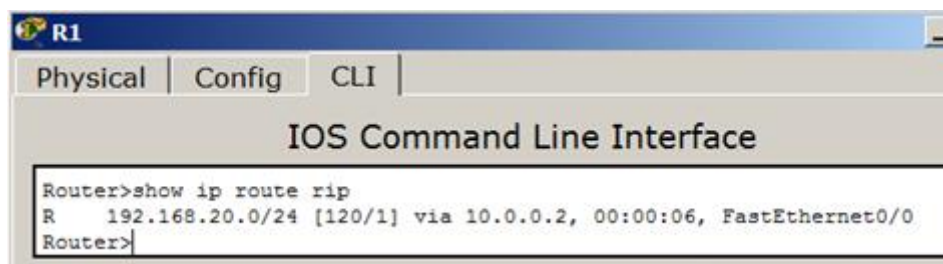
**Рис. 6.4.** Настройки маршрутизаторов R1 и R2

Чтобы убедиться в том, что маршрутизаторы действительно правильно сконфигурированы и работают корректно, просмотрите таблицу RIP роутеров, используя команду: **Router#show ip route rip**

```
Router>show ip route rip
R    192.168.10.0/24 [120/1] via 10.10.0.1, 00:00:12, FastEthernet0/1
Router>
```

**Рис. 6.6.** Таблица маршрутизации R1

Данная таблица показывает, что к сети 192.168.10.0 есть только один маршрут: через R1(сеть 10.10.0.1).



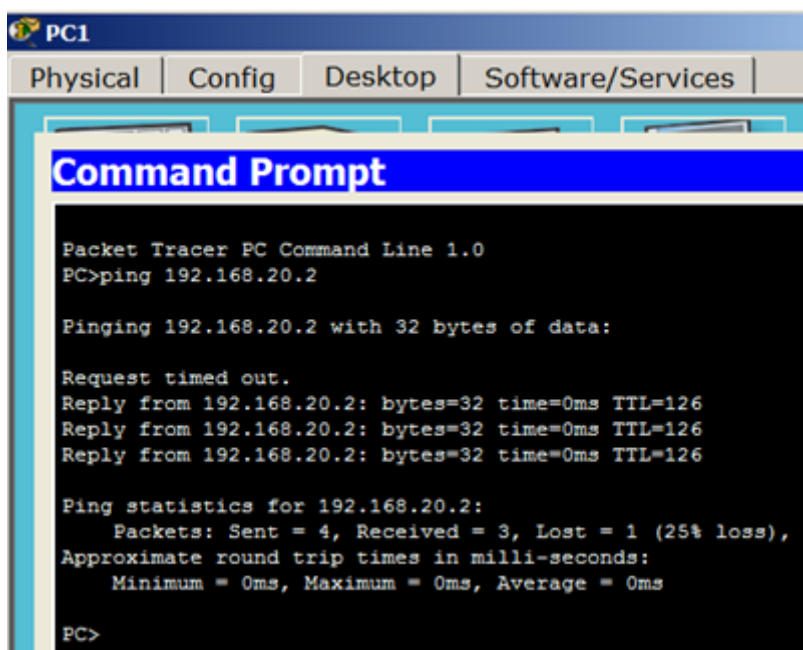
```
R1
Physical Config CLI
IOS Command Line Interface
Router>show ip route rip
R    192.168.20.0/24 [120/1] via 10.0.0.2, 00:00:06, FastEthernet0/0
Router>
```

**Рис. 6.6.** Таблицы маршрутизации R2

Данная таблица показывает, что к сети 192.168.20.0 есть только один маршрут: через R2 (сеть 10.10.0.2).

## Проверка связи между PC1 и PC2

Проверим, что маршрутизация производится верно



```
PC1
Physical Config Desktop Software/Services
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

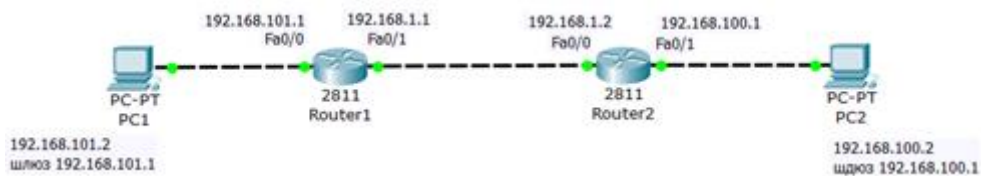
Request timed out.
Reply from 192.168.20.2: bytes=32 time=0ms TTL=126
Reply from 192.168.20.2: bytes=32 time=0ms TTL=126
Reply from 192.168.20.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>
```

**Рис. 6.7.** Пинг с PC1 на PC2

## Практическая работа 6-2. Конфигурирование протокола RIP версии 2 для сети из четырех устройств

На представлена *сеть*, на примере которой мы сконфигурируем *протокол маршрутизации RIP v2*.



**Рис. 6.6.** Сеть для конфигурации протоколов маршрутизации

Сначала сконфигурируем R1

```

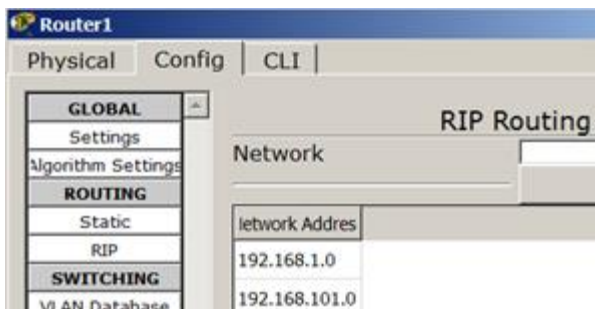
Router1
Physical Config CLI
IOS Command Line Interface

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 192.168.101.1
Router(config-router)#network 192.168.1.0
Router(config-router)#exit
Router(config)#

```

**Рис. 6.9.** Настройка RIP на R1

Смотрим результат на вкладке **Config**



**Рис. 6.10.** Окно R1, вкладка Config

Конфигурируем R2

```

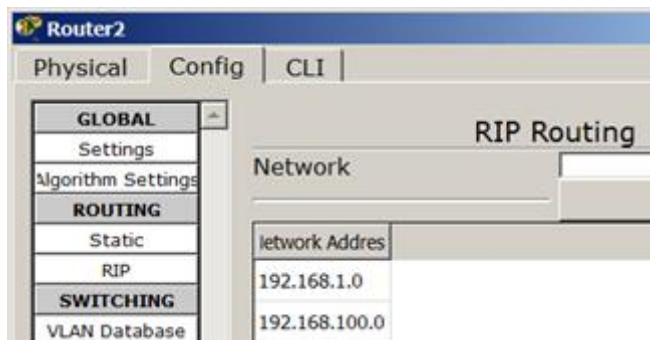
Router2
Physical Config CLI
IOS Command Line Interface

Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 192.168.100.1
Router(config-router)#network 192.168.1.0
Router(config-router)#exit
Router(config)#

```

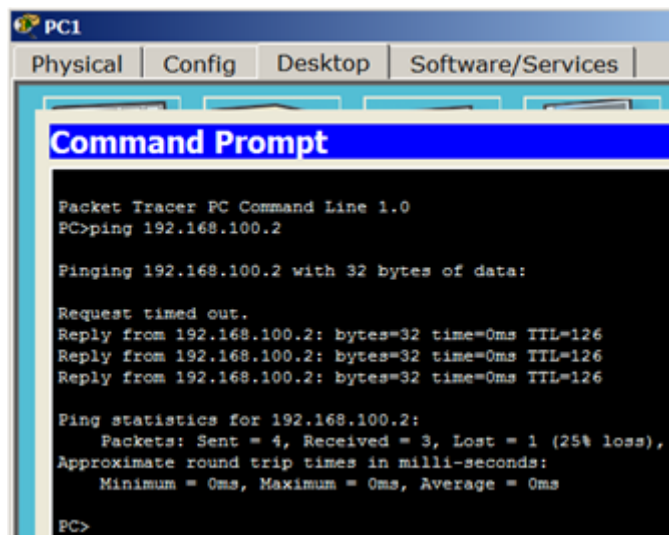
**Рис. 6.11.** Настройка RIP на R2

Наблюдаем результат



**Рис. 6.12.** Окно R2, вкладка Config

Проверяем доступность ПК из разных сетей



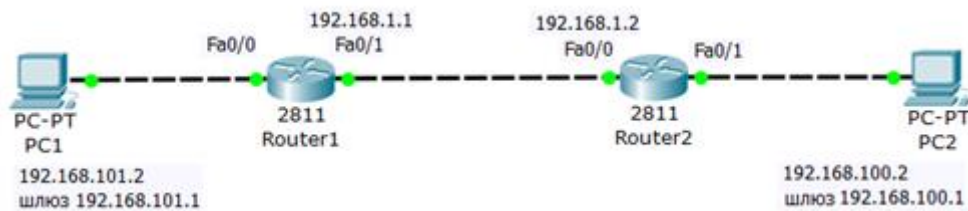
**Рис. 6.13.** Результат маршрутизации по протоколу RIP

## Протокол маршрутизации EIGRP

Протокол *EIGRP* более прост в реализации и менее требователен к вычислительным ресурсам маршрутизатора, чем протокол *OSPF*. Также *EIGRP* имеет более продвинутый алгоритм вычисления метрики. В формуле вычисления метрики есть возможность учитывать загруженность и надежность интерфейсов на пути пакета. Недостатком протокола *EIGRP* является его ограниченность в его использовании только на оборудовании компании Cisco.

## Практическая работа 6-3. Конфигурирование протокола EIGRP

Схема сети изображена на



**Рис. 6.14.** Схема для конфигурации протокола EIGRP

Настройка протокола *EIGRP* очень похожа на настройку протокола *RIP*.

## Программирование R1

Конфигурируем R1

```

Router1
Physical Config CLI
IOS Command Line Interface

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router eigrp 10
Router(config-router)#network 192.168.101.1
Router(config-router)#exit
Router(config)#
  
```

**Рис. 6.16.** Конфигурирование R1

## Программирование R2

Конфигурируем R2

```

Router2
Physical Config CLI
IOS Command Line Interface

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router eigrp 10
Router(config-router)#network 192.168.100.1
Router(config-router)#network 192.168.1.0
Router(config-router)#exit
Router(config)#
  
```

**Рис. 6.16.** Конфигурирование R2

## Проверка работы сети

Проверяем работу маршрутизаторов



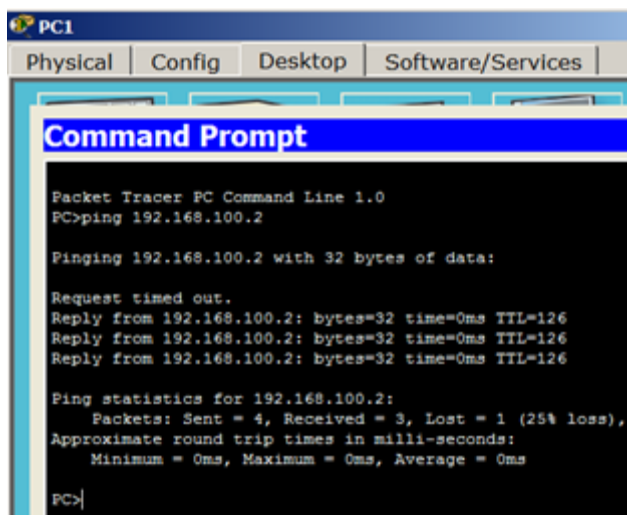


Рис. 6.17. Результат проверки работоспособности сети

## Протокол OSPF

Алгоритм работы протокола динамической маршрутизации *OSPF* основан на использовании всеми маршрутизаторами единой *базы данных*, описывающей, с какими сетями связан каждый *маршрутизатор*. Описывая каждую *связь*, маршрутизаторы связывают с ней метрику – *значение*, характеризующее "качество" канала связи. Это позволяет маршрутизаторам *OSPF* (в отличие от *RIP*, где все каналы равнозначны) учитывать реальную пропускную способность канала и выявлять наилучшие маршруты. Важной особенностью протокола *OSPF* является то, что используется групповая, а не широковещательная рассылка (как в *RIP*), то есть, нагрузка каналов меньше.

*OSPF (Open Shortest Path First)* — протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала link-state (*LSA*). Основан на алгоритме для поиска кратчайшего пути. Отслеживание состояния канала требует отправки объявлений о состоянии канала (*LSA*) на активные интерфейсы всех доступных маршрутизаторов зоны. В этих объявлениях содержится описание всех каналов маршрутизатора и *стоимость* каждого канала. *LSA* сообщения отправляются, только если произошли какие-либо изменения в сети, но раз в 30 минут *LSA* сообщения отправляются в принудительном порядке. Протокол реализует *деление* автономной системы на зоны (areas). Использование зон позволяет снизить нагрузку на *сеть* и процессоры маршрутизаторов и уменьшить размер таблиц маршрутизации.

### Описание работы протокола:

Все маршрутизаторы обмениваются специальными Hello-пакетами через все интерфейсы, на которых активирован протокол *OSPF*. Таким образом, определяются маршрутизаторы-соседи, разделяющие общий *канал передачи данных*. В дальнейшем hello-пакеты посылаются с интервалом раз в 30 секунд. Маршрутизаторы пытаются перейти в состояние соседства со своими соседями. Переход в данное состояние определяется типом маршрутизаторов и типом сети, по которой происходит обмен hello-пакетами, по зонному признаку. Пара маршрутизаторов в состоянии соседства синхронизирует между собой базу данных состояния каналов. Каждый *маршрутизатор* посылает объявление о состоянии канала своим соседям, а каждый получивший такое объявление записывает информацию в базу данных состояния каналов и рассылает копию объявления другим своим соседям. При рассылке объявлений по зоне, все маршрутизаторы строят идентичную базу данных состояния каналов. Каждый *маршрутизатор* использует *алгоритм SPF* для вычисления графа (дерева кратчайшего пути) без петель. Каждый *маршрутизатор* строит собственную маршрутизацию, основываясь на построенном дереве кратчайшего пути.

## Прямая и обратная маска

В оборудовании **Cisco** иногда приходится использовать обратную маску, то есть не привычную нам **266.266.266.0** (Subnet mask — прямая *маска*), а **0.0.0.266** (Wildcard mask — обратная *маска*).



Обратная *маска* используется в листах допуска (*access list*) и при описании сетей в протоколе **OSPF**. Прямая *маска* используется во всех остальных случаях. Отличие масок заключается также в том, что прямая *маска* оперирует сетями, а обратная — хостами. С помощью обратной маски вы можете, например, выделить во всех подсетях хосты с конкретным адресом и разрешить им *доступ* в *Интернет*. Так, как чаще всего в локальных сетях используют адреса типа 192.168.1.0 с маской 255.255.255.0, то самая распространенная Wildcard mask (шаблонная *маска* или обратная *маска*, или инверсная *маска*) - маска 0.0.0.255.

### Новый термин

Шаблонная маска (wildcard mask) — маска, указывающая на количество хостов сети. Является дополнением для маски подсети. Вычисляется по формуле для каждого из октетов маски подсети как 255-маска\_подсети. Например, для сети 192.168.1.0 и маской подсети 255.255.255.242 шаблонная маска будет выглядеть как 0.0.0.13. Шаблонная маска используется в настройке некоторых протоколов маршрутизации, а также является удобным параметром ограничений в списках доступа.

### Расчёт Wildcard mask

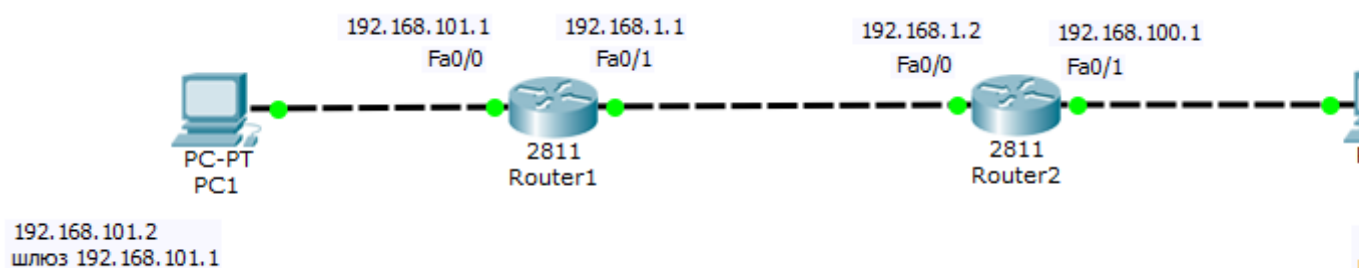
Существует *связь*, между обратной и *прямой* маской: в сумме эти маски по каждому разряду должны составлять 255. Пусть наша *сеть* 192.168.32.0 /26. Рассчитает wildcard mask: *префикс* /26 это 255.255.255.240 или 11111111.11111111.11111111.11110000. Для wildcard mask нам нужны только нули, то есть, 11110000 переводим в десятичное число и считаем: 128/64/32/16/8/4/2/1 это будет 6+4+2+1=13, т.е. наша wildcard mask будет равна 0.0.0.13.

### Самостоятельно

Дана прямая *маска* **255.255.255.246**. Выполните расчет и докажите, что обратная равна **0.0.0.7**.

### Практическая работа 6-2-1. Пример конфигурирования протокола OSPF для 4-х устройств

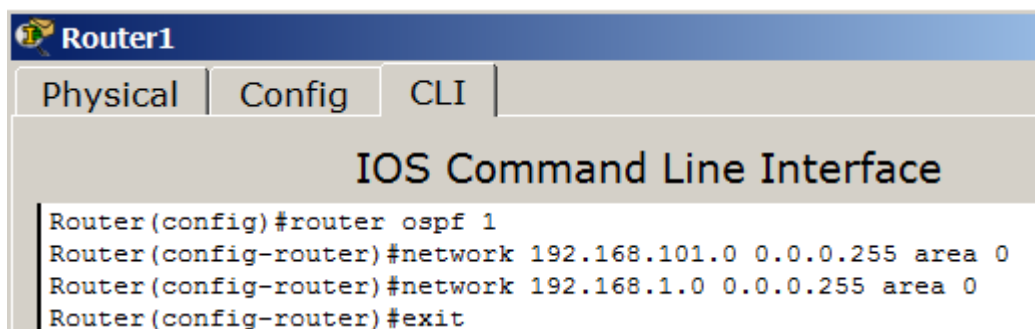
Соберите схему, изображенную на



**Рис. 6.16.** Схема для конфигурации протокола OSPF

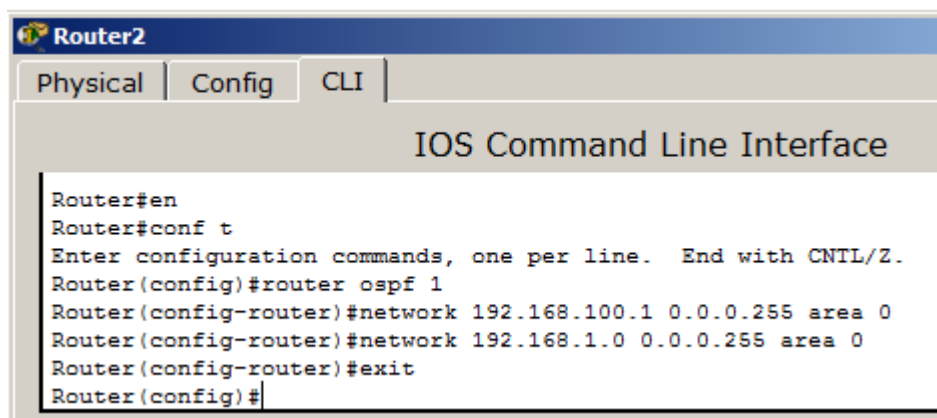
### Настройка роутеров

Выполним конфигурирование R1



**Рис. 6.19.** Настройка R1

Теперь выполним настройки R2



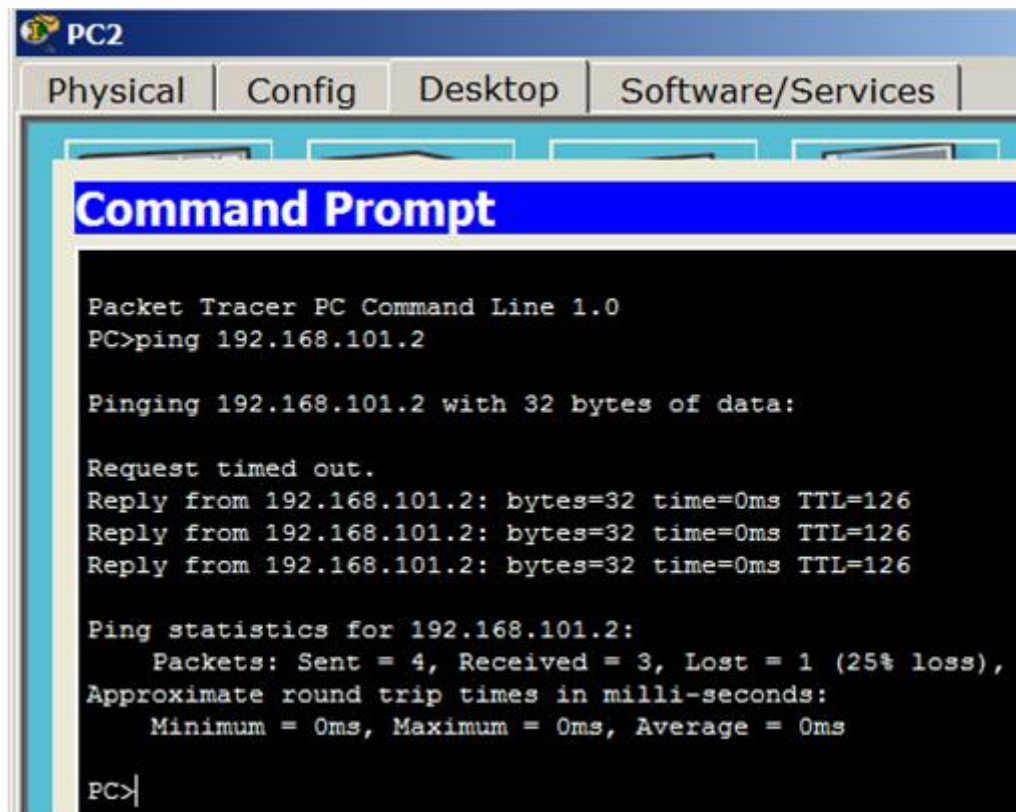
**Рис. 6.20.** Настройка R2

#### **Совет**

Если вам потребуется в СРТ сбросить настройки роутера, то следует выключить его тумблер питания, а затем снова включить.

#### **Проверка результата**

Для проверки маршрутизации пропингуем ПК из разных сетей



PC2

Physical | Config | Desktop | Software/Services

### Command Prompt

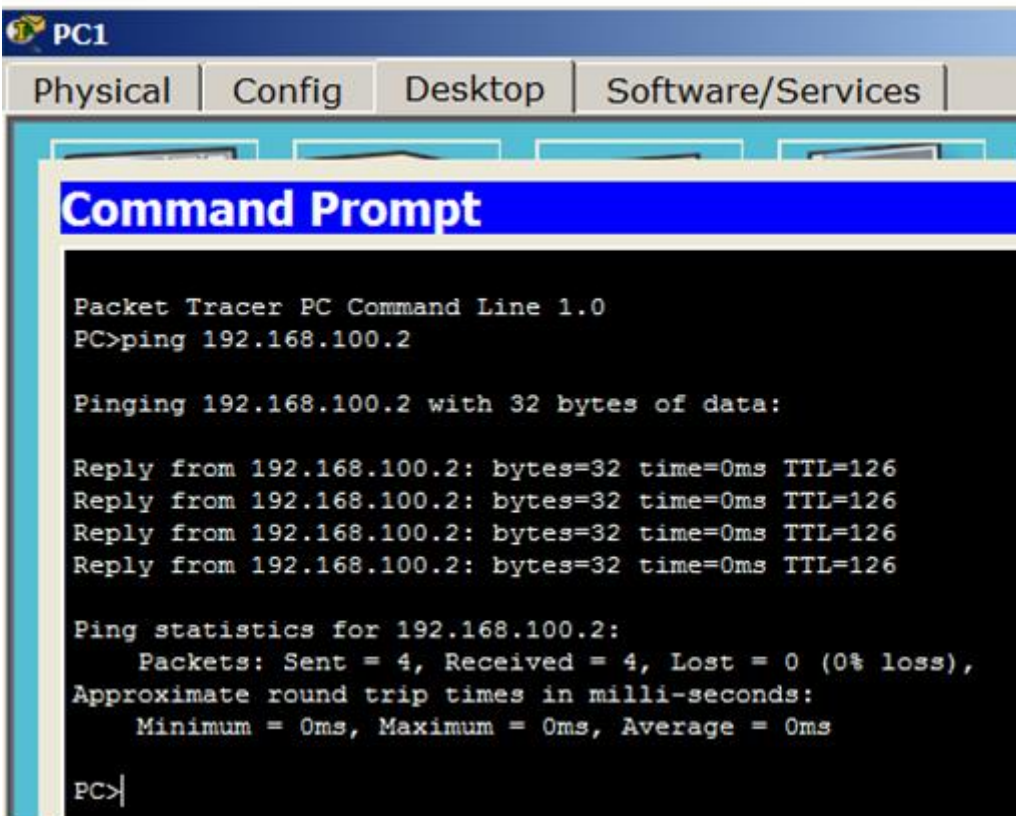
```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.101.2

Pinging 192.168.101.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.101.2: bytes=32 time=0ms TTL=126
Reply from 192.168.101.2: bytes=32 time=0ms TTL=126
Reply from 192.168.101.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.101.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```



PC1

Physical | Config | Desktop | Software/Services

### Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.100.2

Pinging 192.168.100.2 with 32 bytes of data:

Reply from 192.168.100.2: bytes=32 time=0ms TTL=126
Reply from 192.168.100.2: bytes=32 time=0ms TTL=126
Reply from 192.168.100.2: bytes=32 time=0ms TTL=126
Reply from 192.168.100.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.100.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```

Рис. 6.21. Результат проверки работоспособности OSPF

## Практическая работа 6-2-2. Настройка маршрутизации по протоколу OSPF для 6 устройств

Постройте следующую схему

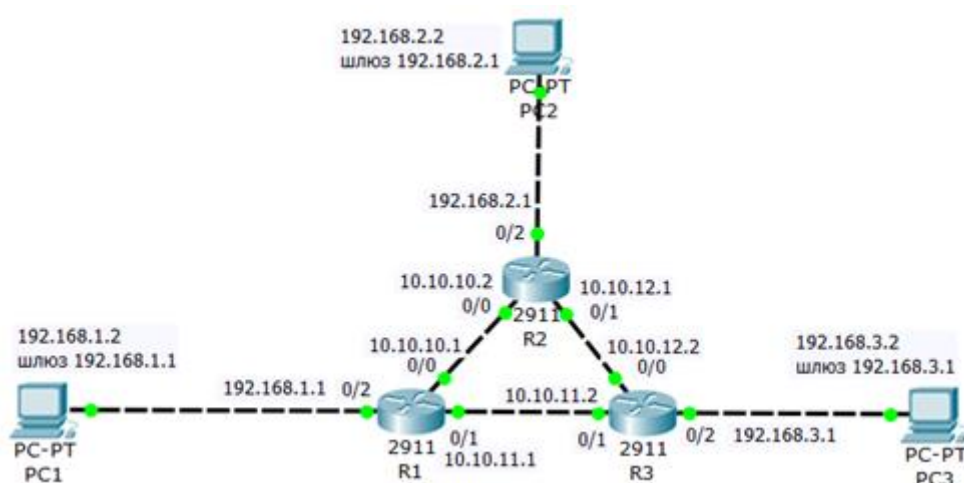


Рис. 6.22. Начальная схема сети для нашей работы

Цель работы – настроить маршрутизацию в данной сети по протоколу *OSPF*.

### Настроим loopback интерфейс на R1

На R1 настроим программный **loopback интерфейс** — алгоритм, который направляет полученный сигнал (или данные) обратно отправителю

#### Примечание

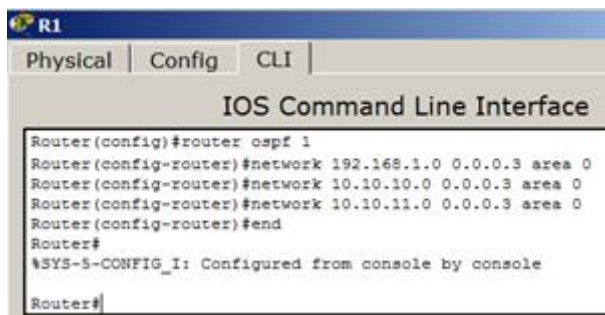
IPv4-адрес, назначенный loopback-интерфейсу, может быть необходим для процессов маршрутизатора, в которых используется IPv4-адрес интерфейса в целях идентификации. Один из таких процессов — алгоритм кратчайшего пути (OSPF). При включении интерфейса loopback для идентификации маршрутизатор будет использовать всегда доступный адрес интерфейса loopback, а не IP-адрес, назначенный физическому порту, работа которого может быть нарушена. На маршрутизаторе можно активировать несколько интерфейсов loopback. IPv4-адрес для каждого интерфейса loopback должен быть уникальным и не должен быть задействован другим интерфейсом.



Рис. 6.23. Настраиваем интерфейс loopback на R1

### Настраиваем протокол OSPF на R1

Включаем OSPF на R1, все маршрутизаторы должны быть в одной зоне **area 0**



**Рис. 6.24.** Включаем протокол OSPF на R1

Подводим курсор мыши к R1 и наблюдаем результат наших настроек

Port	Link	VLAN	IP Address
GigabitEthernet0/0	Up	--	10.10.10.1/30
GigabitEthernet0/1	Up	--	10.10.11.1/30
GigabitEthernet0/2	Up	--	192.168.1.1/24
Loopback0	Up	--	192.168.100.1/32

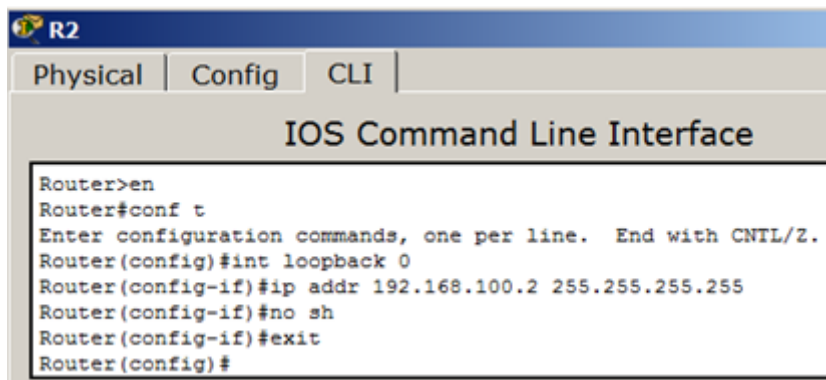
**Рис. 6.26.** Маршрутизатор R1 настроен

#### Примечание

Обратите внимание, что физически порта 192.166.100.1 нет, он существует только логически (программно).

## Настроим loopback интерфейс на R2

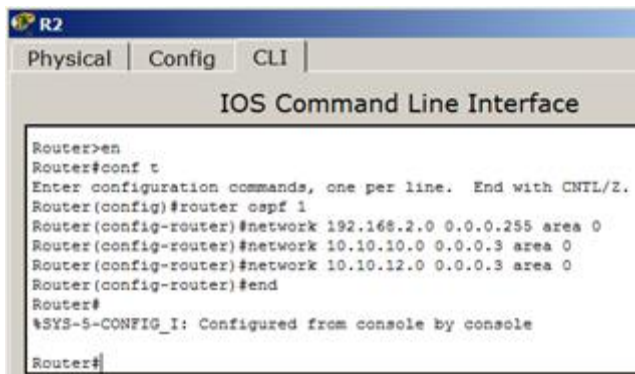
На R2 настроим программный loopback интерфейс по аналогии с R1



**Рис. 6.26.** Настраиваем логический интерфейс loopback на R2

## Настраиваем OSPF на R2

Включаем протокол OSPF на R2, все маршрутизаторы должны быть в одной зоне area 0



**Рис. 6.27.** Включаем протокол OSPF на R2

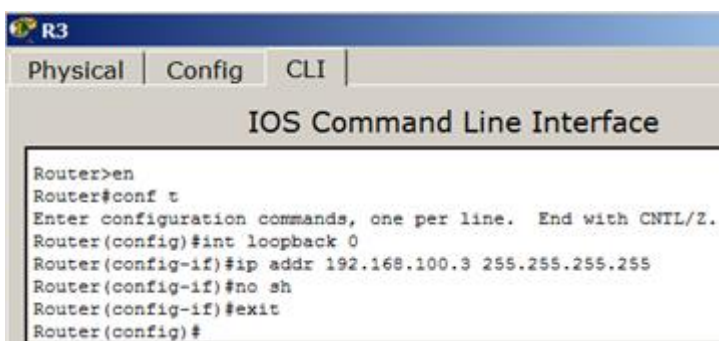
Подводим курсор мыши к R2 и наблюдаем результат наших настроек

Port	Link	VLAN	IP Address
GigabitEthernet0/0	Up	--	10.10.10.2/30
GigabitEthernet0/1	Up	--	10.10.12.1/30
GigabitEthernet0/2	Up	--	192.168.2.1/24
Loopback0	Up	--	192.168.100.2/32

**Рис. 6.26.** Маршрутизатор R2 настроен

## Настраиваем loopback интерфейс на R3

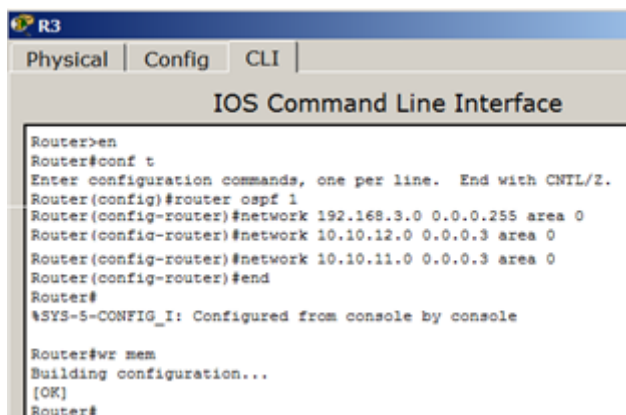
Делаем все аналогично



**Рис. 6.29.** Настраиваем логический интерфейс loopback на R3

## Настраиваем протокол OSPF на R3

Здесь делаем все, как раньше



```
R3
Physical | Config | CLI
IOS Command Line Interface

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 192.168.3.0 0.0.0.255 area 0
Router(config-router)#network 10.10.12.0 0.0.0.3 area 0
Router(config-router)#network 10.10.11.0 0.0.0.3 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#wr mem
Building configuration...
[OK]
Router#
```

**Рис. 6.30.** Включаем протокол OSPF на R2

Проверяем результат (

Port	Link	VLAN	IP Address
GigabitEthernet0/0	Up	--	10.10.12.2/30
GigabitEthernet0/1	Up	--	10.10.11.2/30
GigabitEthernet0/2	Up	--	192.168.3.1/24
Loopback0	Up	--	192.168.100.3/32

**Рис. 6.31.** Маршрутизатор R3 настроен

## Проверяем работу сети

Убеждаемся, что роутер R3 видит R2 и R1



```
R3
Physical | Config | CLI
IOS Command Line Interface

Router#sh ip ospf neighbor

Neighbor ID    Pri   State           Dead Time   Address
Interface
192.168.100.2   1     FULL/BDR        00:00:31    10.10.12.1
GigabitEthernet0/0
192.168.100.1   1     FULL/BDR        00:00:31    10.10.11.1
GigabitEthernet0/1
Router#
```

**Рис. 6.32.** Роутер R3 видит своих соседей

Теперь посмотрим таблицу маршрутизации для R3



```
Router#sh ip route
Codes: L - local, C - connected, S - static, 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, 5 subnets, 2 masks
O       10.10.10.0/30 [110/2] via 10.10.11.1, 01:09:30,
GigabitEthernet0/1
        [110/2] via 10.10.12.1, 01:09:30,
GigabitEthernet0/0
C       10.10.11.0/30 is directly connected, GigabitEthernet0/1
L       10.10.11.2/32 is directly connected, GigabitEthernet0/1
C       10.10.12.0/30 is directly connected, GigabitEthernet0/0
L       10.10.12.2/32 is directly connected, GigabitEthernet0/0
O       192.168.1.0/24 [110/2] via 10.10.11.1, 01:09:30, GigabitEthernet0/1
O       192.168.2.0/24 [110/2] via 10.10.12.1, 01:09:30, GigabitEthernet0/0
        192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, GigabitEthernet0/2
L       192.168.3.1/32 is directly connected, GigabitEthernet0/2
--More--
```

Рис. 6.33. Таблица маршрутизации для R3

#### Примечание

В этой таблице запись с буквой "O" говорит о том, что данный маршрут прописан протоколом OSPF. Мы видим, что сеть 192.166.1.0 доступна для R3 через адрес 10.10.11.1 (это порт gig0/1 маршрутизатора R1). Аналогично, сеть 192.166.2.0 доступна для R3 через адрес 10.10.12.1 (это порт gig0/1 маршрутизатора R2).

Теперь проверяем доступность разных сетей

```
Router>ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/0 ms

Router>ping 192.168.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/1 ms

Router>
```

Рис. 6.34. Сети 192.166.1.0 и 192.166.2.0 доступны