



# Маршрутизация в информационных сетях





**Маршрутизация** (*Routing*) — процесс определения маршрута следования информации в сетях связи.

Маршрутизация означает передвижение информации от источника к пункту назначения через объединенную сеть. При этом, как правило, на пути встречается по крайней мере один узел.

# Компоненты маршрутизации

## • Определение маршрута

Определение маршрута может базироваться на различных показателях (величинах, результирующих из алгоритмических вычислений по отдельной переменной - например, длина маршрута) или комбинациях показателей. Программные реализации алгоритмов маршрутизации высчитывают показатели маршрута для определения оптимальных маршрутов к пункту назначения.

Для облегчения процесса определения маршрута, алгоритмы маршрутизации инициализируют и поддерживают таблицы маршрутизации, в которых содержится маршрутная информация. Маршрутная информация изменяется в зависимости от используемого алгоритма маршрутизации.

## • Транспортировка пакетов через объединенную сеть (Коммутация)

Алгоритмы коммутации сравнительно просты и в основном одинаковы для большинства протоколов маршрутизации. В большинстве случаев главная вычислительная машина определяет необходимость отправки пакета в другую главную вычислительную машину. Получив определенным способом адрес роутера, главная вычислительная машина-источник отправляет пакет, адресованный специально в **физический адрес роутера** (уровень MAC), однако с адресом протокола (сетевой уровень) компьютера пункта назначения.

После проверки адреса протокола пункта назначения пакета роутер определяет, знает он или нет, как передать этот пакет к следующему роутеру. Во втором случае (когда роутер не знает, как переслать пакет) пакет, как правило, игнорируется. В первом случае роутер отсылает пакет к следующей роутеру путем замены физического адреса пункта назначения на физический адрес следующего роутера и последующей передачи пакета.

# Алгоритмы маршрутизации

## Цели разработки алгоритмов маршрутизации

- **Оптимальность**

- Оптимальность, вероятно, является самой общей целью разработки. Она характеризует способность алгоритма маршрутизации выбирать "наилучший" маршрут. Наилучший маршрут зависит от показателей и от "веса" этих показателей, используемых при проведении расчета. Например, алгоритм маршрутизации мог бы использовать несколько пересылок с определенной задержкой, но при расчете "вес" задержки может быть им оценен как очень значительный. Естественно, что протоколы маршрутизации должны строго определять свои алгоритмы расчета показателей.

- **Простота и низкие производительные затраты**

- Алгоритмы маршрутизации разрабатываются как можно более простыми. Другими словами, алгоритм маршрутизации должен эффективно обеспечивать свои функциональные возможности, с минимальными затратами программного обеспечения и коэффициентом использования.

- **Живучесть и стабильность**

- Алгоритмы маршрутизации должны обладать живучестью. Другими словами, они должны четко функционировать в случае неординарных или непредвиденных обстоятельств, таких как отказы аппаратуры, условия высокой нагрузки и некорректные реализации.

- **Быстрая сходимост**

- Алгоритмы маршрутизации должны быстро сходиться. Сходимость - это процесс соглашения между всеми роутерами по оптимальным маршрутам. Когда какое-нибудь событие в сети приводит к тому, что маршруты или отвергаются, или становятся доступными, роутеры рассылают сообщения об обновлении маршрутизации. Алгоритмы маршрутизации, которые сходятся медленно, могут привести к образованию петель маршрутизации или выходам из строя сети.

- **Гибкость**

- Алгоритмы маршрутизации должны быть также гибкими. Другими словами, алгоритмы маршрутизации должны быстро и точно адаптироваться к разнообразным обстоятельствам в сети. Например, предположим, что сегмент сети отвергнут. Многие алгоритмы маршрутизации, после того как они узнают об этой проблеме, быстро выбирают следующий наилучший путь для всех маршрутов, которые обычно используют этот сегмент. Алгоритмы маршрутизации могут быть запрограммированы таким образом, чтобы они могли адаптироваться к изменениям полосы пропускания сети, размеров очереди к роутеру, величины задержки сети и других переменных.

# Типы алгоритмов

- **Статические алгоритмы.** Распределение *статических* таблиц маршрутизации устанавливается администратором сети до начала маршрутизации. Оно не меняется, если только администратор сети не изменит его. Алгоритмы, использующие статические маршруты, просты для разработки и хорошо работают в окружениях, где трафик сети относительно предсказуем, а схема сети относительно проста.
- **Динамические алгоритмы** маршрутизации подстраиваются к изменяющимся обстоятельствам сети в масштабе реального времени. Они выполняют это путем анализа поступающих сообщений об обновлении маршрутизации. Если в сообщении указывается, что имело место изменение сети, программы маршрутизации пересчитывают маршруты и рассылают новые сообщения о корректировке маршрутизации.
- **Одномаршрутные или многомаршрутные алгоритмы.** Некоторые сложные протоколы маршрутизации обеспечивают множество маршрутов к одному и тому же пункту назначения. Преимущества многомаршрутных алгоритмов очевидны - они могут обеспечить значительно большую пропускную способность и надежность.

- **Внутридоменные или междоменные алгоритмы**

Некоторые алгоритмы маршрутизации действуют только в пределах доменов; другие - как в пределах доменов, так и между ними. Природа этих двух типов алгоритмов различная. Поэтому понятно, что оптимальный алгоритм внутридоменной маршрутизации не обязательно будет оптимальным алгоритмом междоменной маршрутизации.

# Типы алгоритмов

Некоторые алгоритмы маршрутизации оперируют в плоском пространстве, в то время как другие используют иерархии маршрутизации. В **одноуровневой системе маршрутизации** все роутеры равны по отношению друг к другу. В иерархической системе маршрутизации некоторые роутеры формируют то, что составляет основу маршрутизации. Пакеты из небазовых роутеров перемещаются к базовым роутерам и пропускаются через них до тех пор, пока не достигнут общей области пункта назначения. Начиная с этого момента, они перемещаются от последнего базового роутера через один или несколько небазовых роутеров до конечного пункта назначения.

Основным преимуществом **иерархической маршрутизации** является то, что она имитирует организацию большинства компаний и следовательно, очень хорошо поддерживает их схемы трафика. Большая часть сетевой связи имеет место в пределах групп небольших компаний (доменов). Внутридоменным роутерам необходимо знать только о других роутерах в пределах своего домена, поэтому их алгоритмы маршрутизации могут быть упрощенными. Соответственно может быть уменьшен и трафик обновления маршрутизации, зависящий от используемого алгоритма маршрутизации.

# Типы алгоритмов

## *Алгоритмы состояния канала или вектора расстояния*

Алгоритмы **состояния канала** направляют потоки маршрутной информации во все узлы объединенной сети. Однако каждый роутер посылает только ту часть маршрутной таблицы, которая описывает состояние его собственных каналов.

**Преимущества:** быстрая сходимость, не склонны к образованию петель маршрутизации.

Алгоритмы **вектора расстояния** (известные также как алгоритмы Бэлмана-Форда) требуют от каждого роутера посылки всей или части своей маршрутной таблицы, но только своим соседям. Алгоритмы состояния каналов фактически направляют небольшие корректировки по всем направлениям, в то время как алгоритмы вектора расстояний отсылают более крупные корректировки только в соседние роутеры.

**Преимущества:** просты, менее ресурсоемкие.

Вследствие этого, реализация и поддержка алгоритмов состояния канала может быть более дорогостоящей. Несмотря на их различия, оба типа алгоритмов хорошо функционируют при самых различных обстоятельствах.

# Показатели алгоритмов (метрики)

## Длина маршрута

Длина маршрута является наиболее общим показателем маршрутизации. Некоторые протоколы маршрутизации позволяют администраторам сети назначать произвольные цены на каждый канал сети. В этом случае длиной тракта является сумма расходов, связанных с каждым каналом, который был traversирован. Часто используют "количество переходов", т.е. показатель, характеризующий число проходов, которые пакет должен совершить на пути от источника до пункта назначения через роутеры.

## Надежность

Надежность, в контексте алгоритмов маршрутизации, относится к надежности каждого канала сети (обычно описываемой в терминах соотношения бит/ошибка).

## Задержка

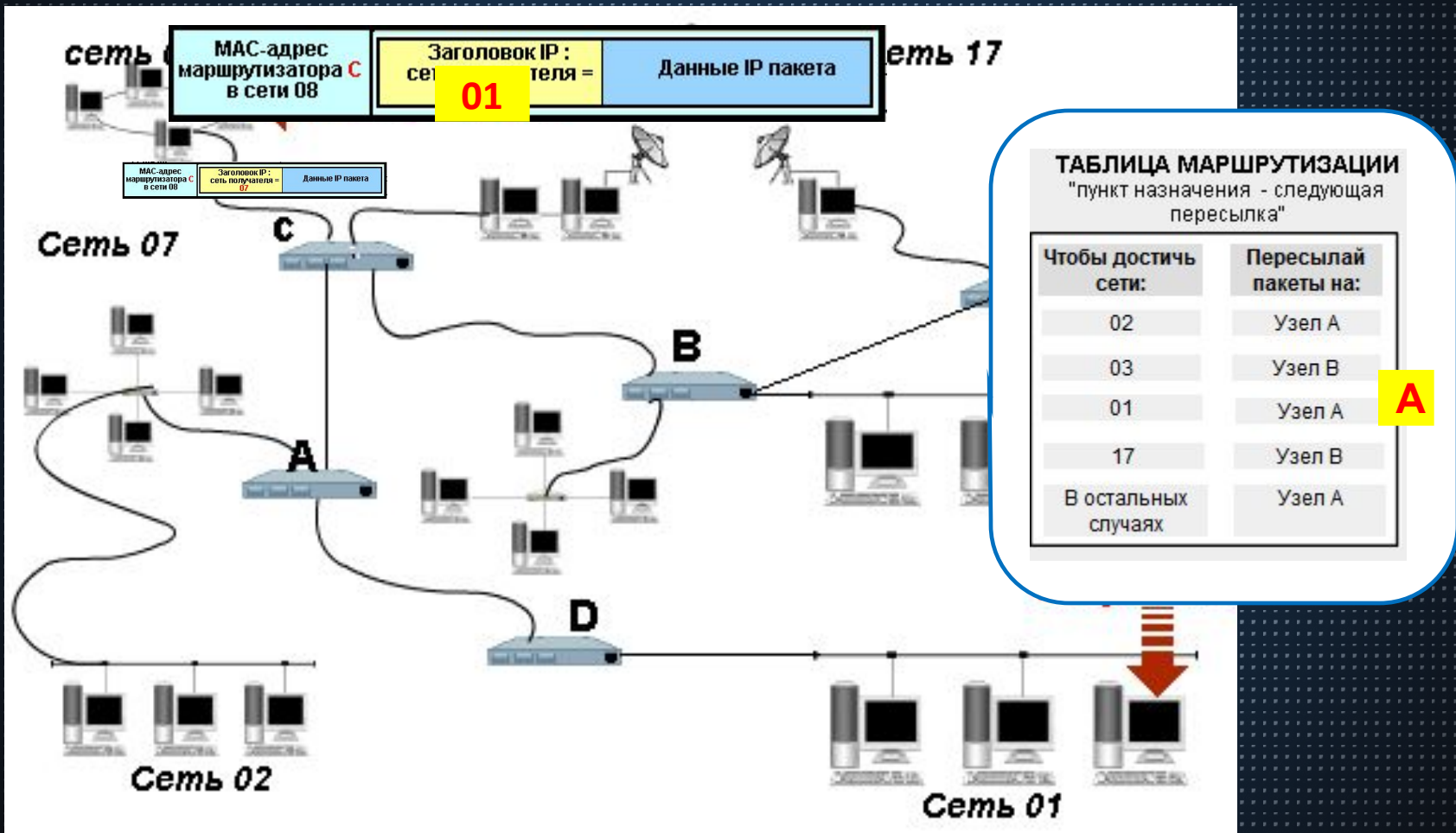
Под задержкой маршрутизации обычно понимают отрезок времени, необходимый для передвижения пакета от источника до пункта назначения через объединенную сеть.

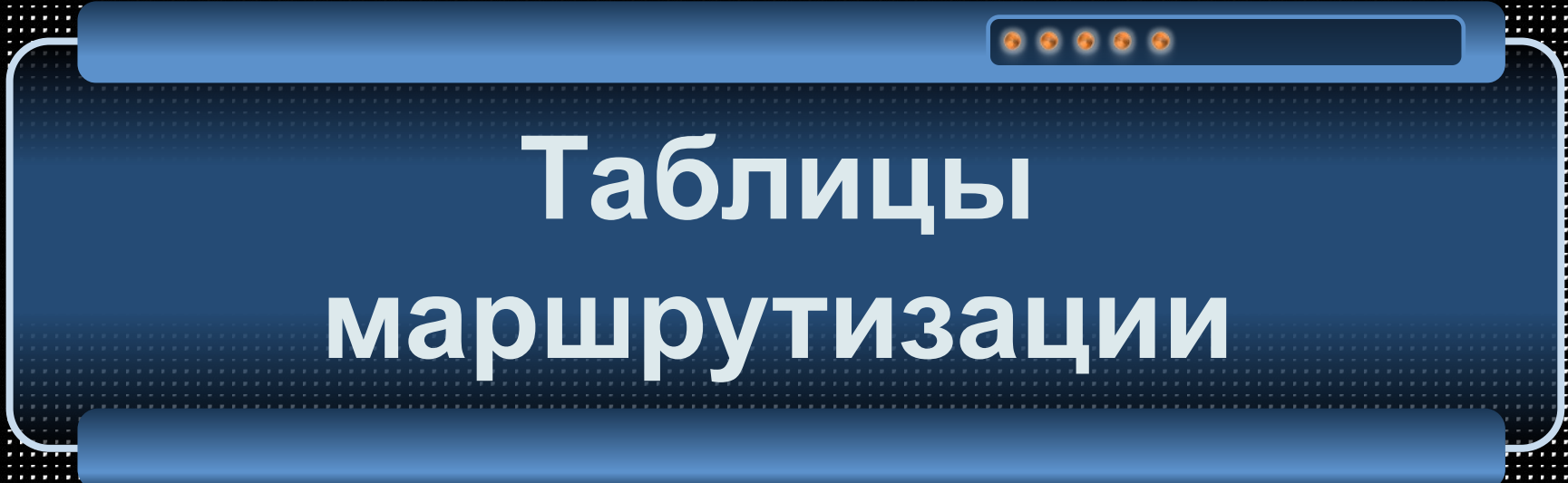
## Полоса пропускания

Полоса пропускания относится к имеющейся мощности трафика какого-либо канала.



# Принцип работы





# Таблицы маршрутизации

# Структура таблицы маршрутизации Windows

- **адрес сети** или **узла назначения**, либо указание, что маршрут является *маршрутом по умолчанию*
- **маску сети назначения** (маска /32 (255.255.255.255) позволяет указать единственный узел сети)
- **шлюз**, обозначающий адрес маршрутизатора в сети, на который необходимо отправить пакет, следующий до указанного адреса назначения
- **интерфейс**
- **метрику** — числовой показатель, задающий предпочтительность маршрута. Чем меньше число, тем более предпочтителен маршрут.

# Заполнение таблиц маршрутизации

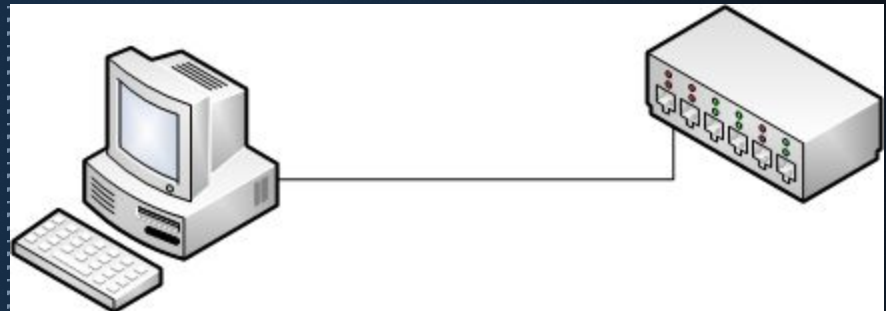
## Источники записей в таблице маршрутизации

- Записи о непосредственно подключенных сетях и маршрутизаторах по умолчанию,
- Адреса особого назначения,
- Записи, введенные принудительно администратором сети,
- Записи, полученные в результате работы протоколов маршрутизации.

# Типы записей в таблице маршрутизации:

- маршрут до компьютера,
- маршрут до сети,
- маршрут по умолчанию,
- маршрут на собственный обратный адрес,
- маршрут для широковещательных пакетов,
- маршрут для многоадресной рассылки.

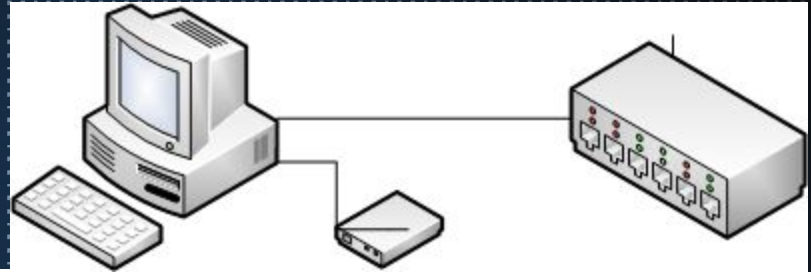
# Пример таблицы маршрутизации



Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	192.168.0.1	192.168.0.4	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.0.0	192.168.0.4	192.168.0.4	2
192.168.0.4	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.0.255	255.255.255.255	192.168.0.4	192.168.0.4	1
224.0.0.0	224.0.0.0	192.168.0.4	192.168.0.4	1
255.255.255.255	255.255.255.255	192.168.0.4	192.168.0.4	1

Сетевая карта с IP-адресом  
192.168.0.4.

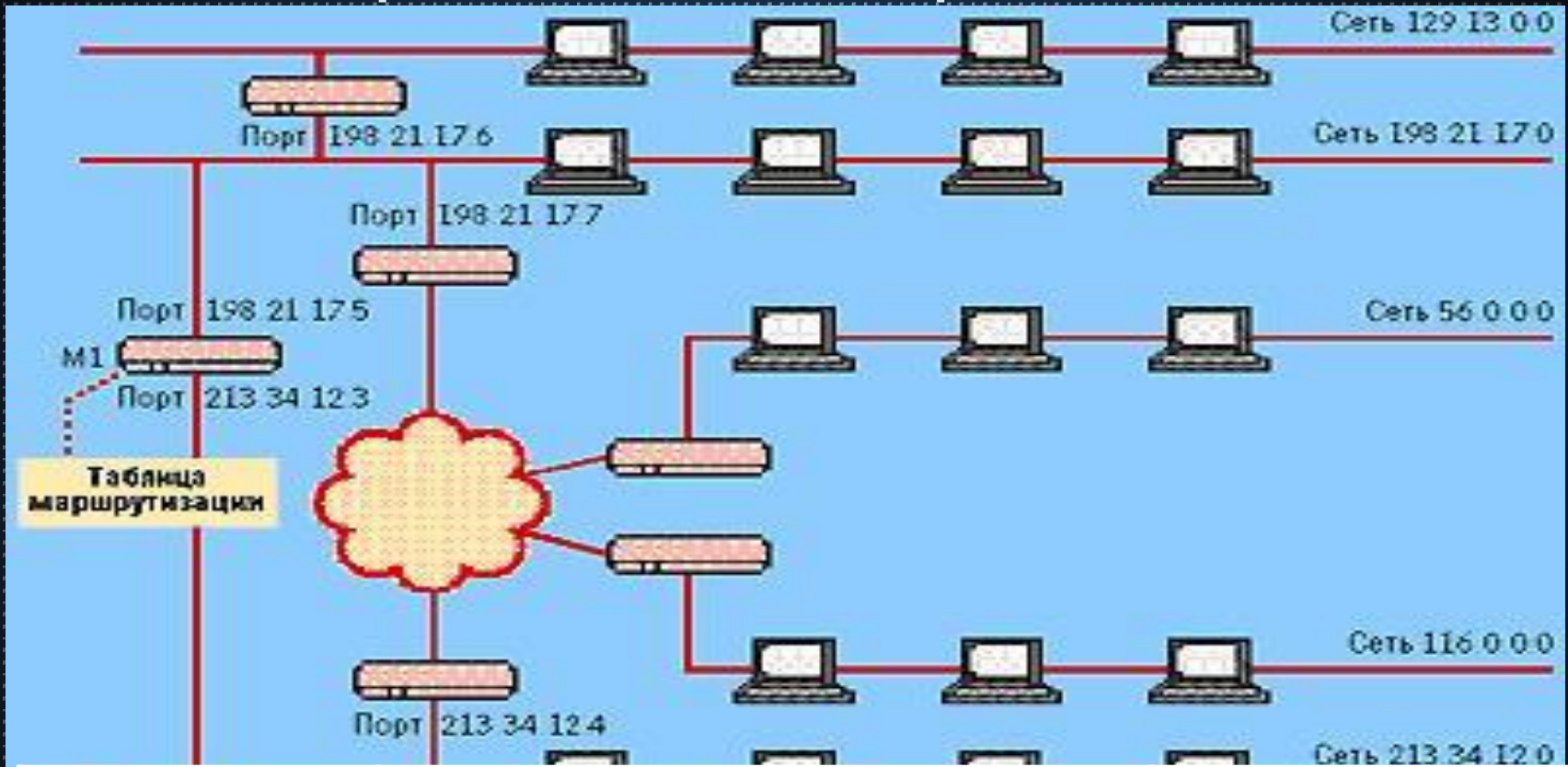
# Пример 2.



Network Address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	203.96.10.254	03.96.10.51	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.0.0	255.255.0.0	192.168.0.4	192.168.0.4	2
192.168.0.4	255.255.255.255	127.0.0.1	127.0.0.1	1
192.168.0.255	255.255.255.255	192.168.0.4	192.168.0.4	1
203.96.10.0	255.255.255.0	203.96.10.51	203.96.10.51	1
203.96.10.51	255.255.255.255	127.0.0.1	127.0.0.1	1
203.96.10.255	255.255.255.255	203.96.10.51	203.96.10.51	1
224.0.0.0	224.0.0.0	203.96.10.51	203.96.10.5	1
224.0.0.0	224.0.0.0	192.168.0.4	192.168.0.4	1
255.255.255.255	255.255.255.255	192.168.0.4	192.168.0.4	1

Сетевая карта с IP-адресом 192.168.0.4  
Модем с IP-адресом 203.96.10.51

# ТАБЛИЦА МАРШРУТИЗАЦИИ В СЕТЯХ IP



Network address	Netmask	Gateway Address	Interface	Metric
224.0.0.0	224.0.0.0	198.21.17.6	198.21.17.6	1
224.0.0.0	224.0.0.0	213.34.12.3	213.34.12.3	1
213.34.12.0	255.255.255.0	213.34.12.3	213.34.12.3	1



# Таблица маршрутизации Windows

Network address	Netmask	Gateway Address	Interface	Metric
0.0.0.0	0.0.0.0	198.21.17.7	198.21.17.5	1
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
56.0.0.0	255.0.0.0	213.34.12.4	213.34.12.3	15
116.0.0.0	255.0.0.0	213.34.12.4	213.34.12.3	13
129.13.0.0	255.255.0.0	198.21.17.6	198.21.17.5	2
198.21.17.0	255.255.255.0	198.21.17.5	198.21.17.5	1
198.21.17.5	255.255.255.255	127.0.0.1	127.0.0.1	1
198.21.17.255	255.255.255.255	198.21.17.5	198.21.17.5	1
213.34.12.0	255.255.255.0	213.34.12.3	213.34.12.3	1
213.34.12.3	255.255.255.255	127.0.0.1	127.0.0.1	1
213.34.12.255	255.255.255.255	213.34.12.3	213.34.12.3	1
224.0.0.0	224.0.0.0	198.21.17.6	198.21.17.6	1
224.0.0.0	224.0.0.0	213.34.12.3	213.34.12.3	1
255.255.255.255	255.255.255.255	198.21.17.6	198.21.17.6	1

# Настройка таблиц маршрутизации

## route

Обработка таблиц сетевых маршрутов.

**ROUTE [-f] [-p] [-4|-6] command [destination]**

**[MASK netmask] [gateway] [METRIC metric] [IF interface]**

-f Очистка таблиц маршрутов от записей всех шлюзов.

-p При использовании с командой ADD задает сохранение маршрута

при перезагрузке системы. По умолчанию маршруты не сохраняются

при перезагрузке.

-4 Обязательное использование протокола IPv4.

-6 Обязательное использование протокола IPv6.

command Одна из следующих команд:

PRINT Печать маршрута

ADD Добавление маршрута

DELETE Удаление маршрута

CHANGE Изменение существующего маршрута

**destination** Адресуемый узел.

**netmask** Значение маски подсети для записи данного маршрута (по умолчанию 255.255.255.255)

**gateway** Шлюз.

**interface** Номер интерфейса для указанного маршрута.

# Примеры

Для команд PRINT и DELETE можно указать узел и шлюз с помощью подстановочных знаков или опустить параметр «шлюз».

Знак '\*' соответствует любой строке, а '?' - одному знаку.

Соответствие шаблону поддерживает только команда PRINT.

```
>route PRINT
```

```
> route PRINT -4
```

```
> route PRINT -6
```

```
> route PRINT 157*      .... Печать только узлов, начинающихся со  
157
```

```
> route ADD 157.0.0.0 MASK 255.0.0.0 157.55.80.1 METRIC 3 IF 2  
           узел^           ^маска      ^шлюз           метрика^
```

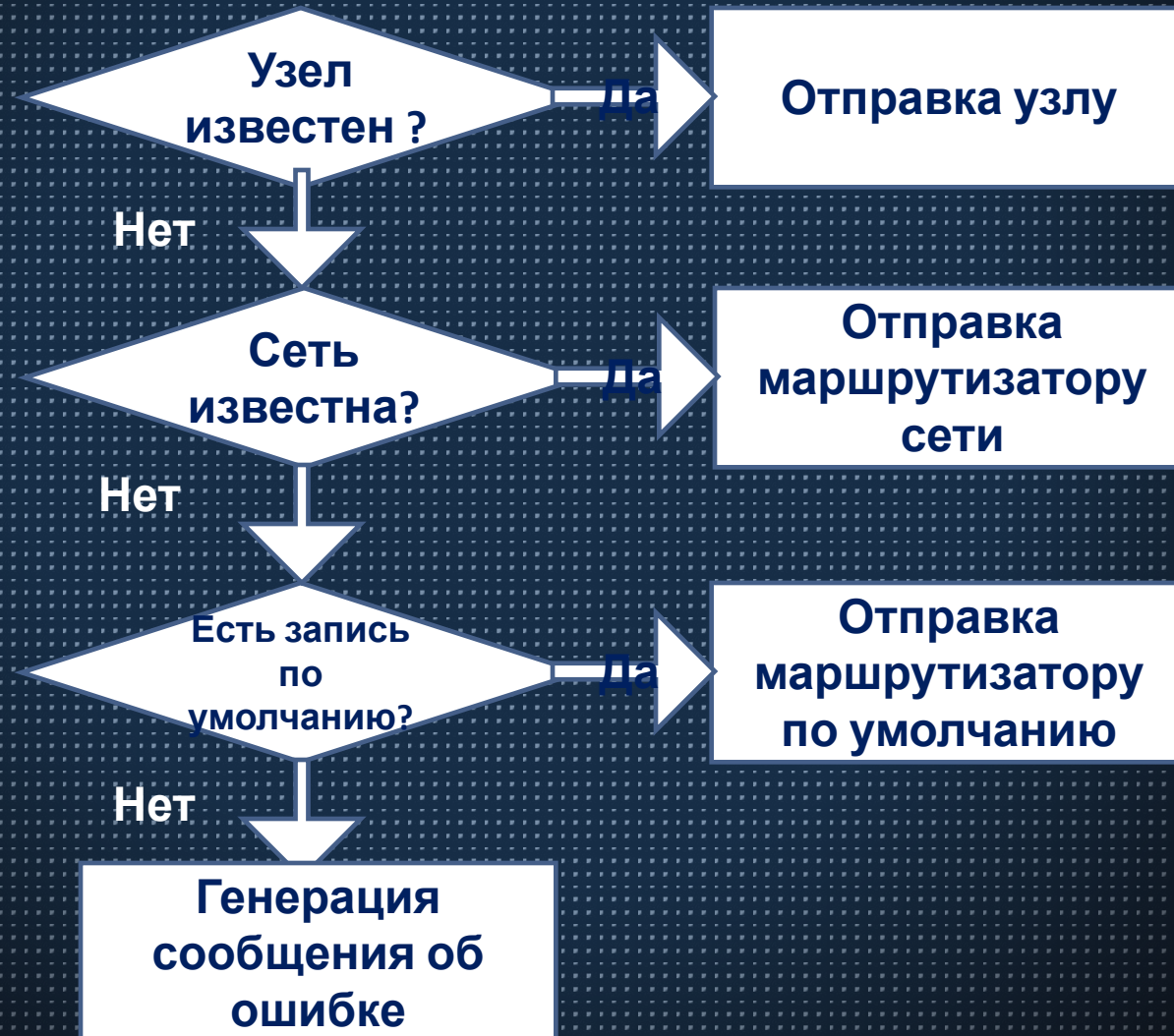
интерфейс^

```
> route CHANGE 157.0.0.0 MASK 255.0.0.0 157.55.80.5 METRIC 2  
IF 2
```

```
> route DELETE 157.0.0.0
```

```
> route DELETE 3ffe::/32
```

# Алгоритм работы маршрутизации



# Таблица маршрутизации Unix-маршрутизатора

Destination	Gateway	Flags	Refcnt	Use	Interface
127.0.0.0	127.0.0.1	UH	1	154	lo0
Default	198.21.17.7	UG	5	43270	le0
198.21.17.0	198.21.17.5	U	35	246876	le0
213.34.12.0	213.34.12.3	U	44	132435	le1
129.13.0.0	198.21.17.6	UG	6	16450	le0
56.0.0.0	213.34.12.4	UG	12	5764	le1
116.0.0.0	213.34.12.4	UG	21	23544	le1

Flags:

**U** - показывает, что маршрут активен и работоспособен.

**H** — признак специфического маршрута к определенному хосту. Маршрут ко всей сети, к которой принадлежит данный хост, может отличаться от данного маршрута;

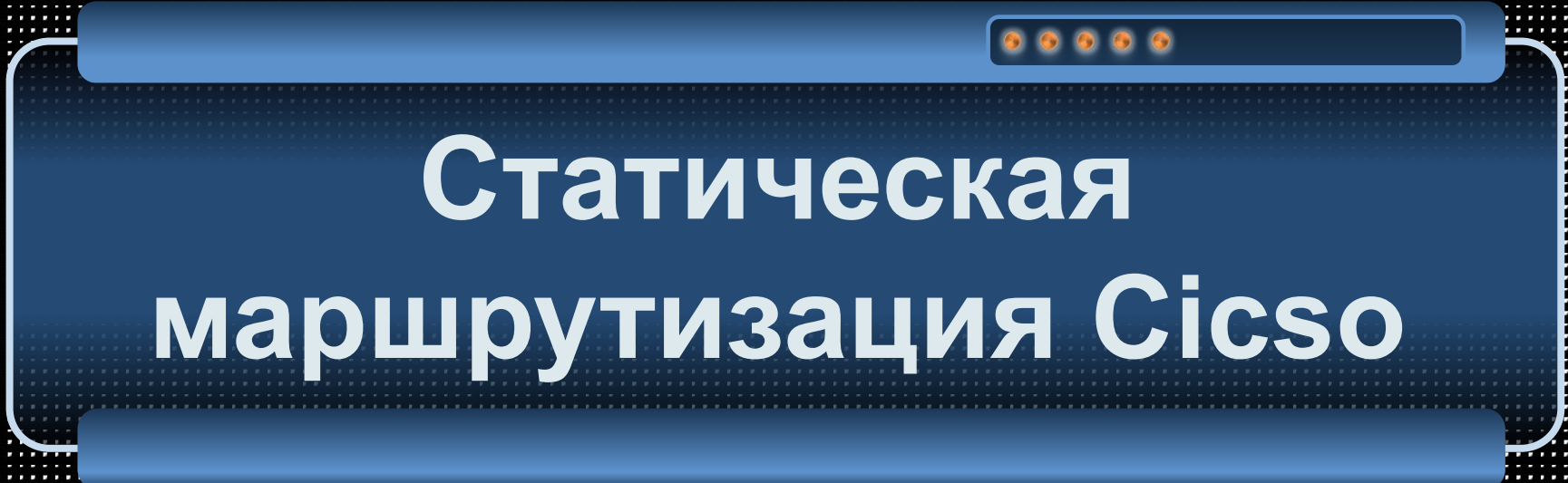
**G** - означает, что маршрут пакета проходит через промежуточный маршрутизатор.

**D** - означает, что маршрут получен из сообщения Redirect протокола ICMP. Такой признак может присутствовать только в таблице маршрутизации конечного узла.

Справочные поля:

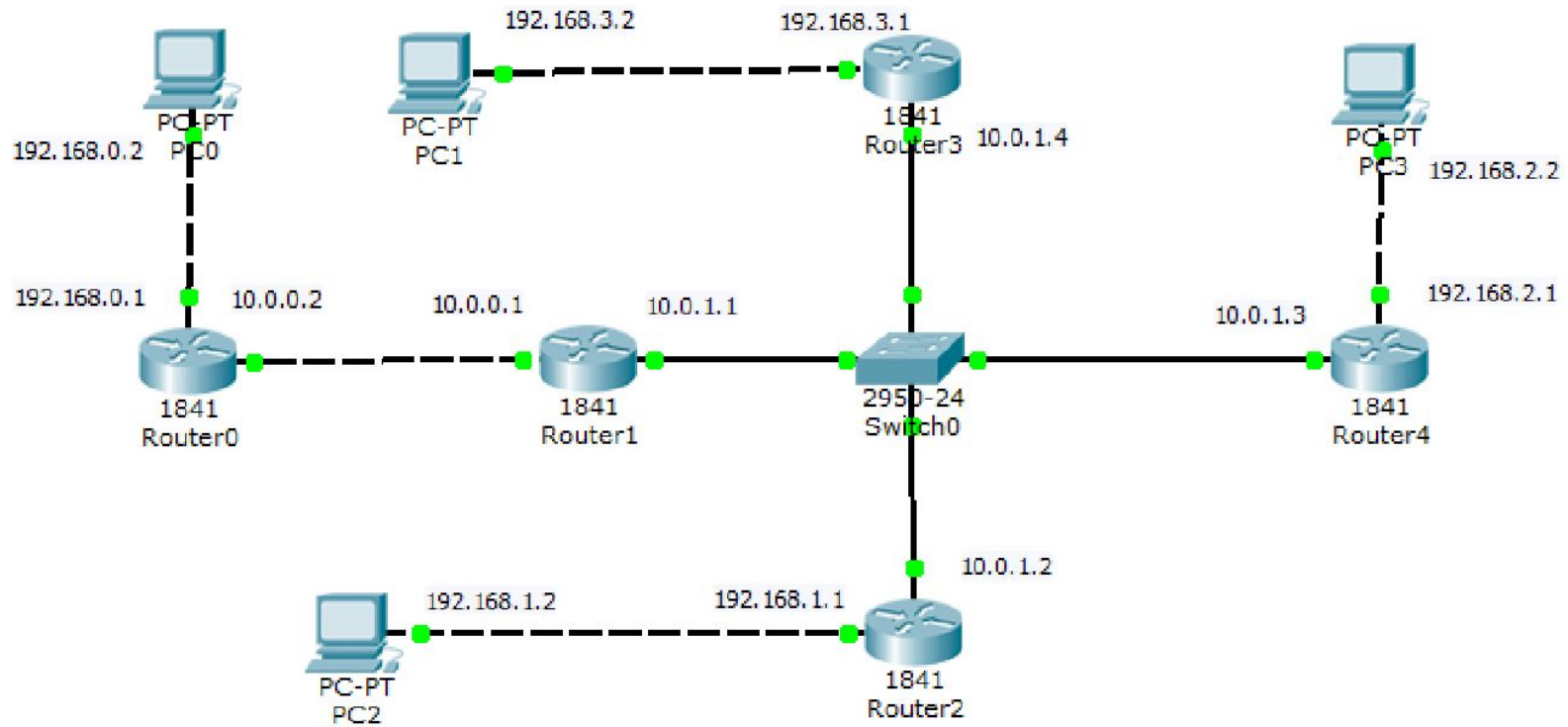
**Refcnt** - показывает, сколько раз на данный маршрут ссылались при продвижении пакетов.

**Use** - отражает количество пакетов, переданных по данному маршруту.



# Статическая маршрутизация Cisco

# Пример сети



# Настройка маршрутизации Router0

The screenshot shows the configuration interface for Router0, specifically the Static Routes section. The interface is divided into three tabs: Physical, Config, and CLI. The Config tab is active, and the Static Routes configuration is displayed. The configuration includes a table for Static Routes with columns for Network, Mask, and Next Hop. The values are 0.0.0.0, 0.0.0.0, and 10.0.0.2 respectively. There is an Add button below the table. Below the table, there is a text box containing the network address 0.0.0.0/0 via 10.0.0.2 and a Remove button. At the bottom, there is a section for Equivalent IOS Commands, which is circled in red. The commands are: Enter configuration commands, one per line. End with CNTL/Z. Router(config)#no ip route 0.0.0.0 0.0.0.0 10.0.0.2 Router(config)#ip route 0.0.0.0 0.0.0.0 10.0.0.2 Router(config)#

Router0

Physical Config CLI

**GLOBAL**

- Settings
- Algorithm Settings

**ROUTING**

- Static
- RIP

**SWITCHING**

- VLAN Database

**INTERFACE**

- FastEthernet0/0
- FastEthernet0/1

**Static Routes**

Network	0.0.0.0
Mask	0.0.0.0
Next Hop	10.0.0.2

Add

Network Address  
0.0.0.0/0 via 10.0.0.2

Remove

**Equivalent IOS Commands**

```
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#no ip route 0.0.0.0 0.0.0.0 10.0.0.2  
Router(config)#ip route 0.0.0.0 0.0.0.0 10.0.0.2  
Router(config)#
```



# Просмотр таблицы маршрутизации Router0

Router# en

Router# 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 10.0.0.2 to network 0.0.0.0

10.0.0.0/24 is subnetted, 1 subnets

**C** 10.0.0.0 is directly connected, FastEthernet0/0

**C** 192.168.0.0/24 is directly connected, FastEthernet0/1

**S\*** 0.0.0.0/0 [1/0] via 10.0.0.2

Router#

# Настройка маршрутизации Router1

The screenshot shows the configuration interface for Router1, specifically the Static Routes section. The interface is divided into three tabs: Physical, Config, and CLI. The Config tab is active, showing a tree view on the left with categories: GLOBAL, ROUTING, SWITCHING, and INTERFACE. The ROUTING category is expanded, showing Static, RIP, and other options. The main area displays the Static Routes configuration for the network 192.168.3.0/24, with a mask of 255.255.255.0 and a next hop of 10.0.1.4. Below this, a list of network addresses is shown, including 0.0.0.0/0 via 10.0.1.2 and several 192.168.x.x/24 networks via 10.0.1.x. A Remove button is visible at the bottom right of the list. At the bottom of the interface, the Equivalent IOS Commands section is highlighted with a red oval, showing the following commands:

```
Router(config)#ip route 192.168.1.0 255.255.255.0 10.0.1.2
Router(config)#ip route 192.168.2.0 255.255.255.0 10.0.1.3
Router(config)#ip route 192.168.3.0 255.255.255.0 10.0.1.4
Router(config)#
```

# Просмотр таблицы маршрутизации Router0

Router#en

Router#show ip route

Codes: C - connected, S - static,

\* - candidate default, U - per-user static route, o - ODR

P - periodic downloaded static route

Gateway of last resort is 10.0.1.2 to network 0.0.0.0

10.0.0.0/24 is subnetted, 2 subnets

**C 10.0.0.0 is directly connected, FastEthernet0/0**

**C 10.0.1.0 is directly connected, FastEthernet0/1**

**S 192.168.0.0/24 [1/0] via 10.0.0.1**

**S 192.168.1.0/24 [1/0] via 10.0.1.2**

**S 192.168.2.0/24 [1/0] via 10.0.1.3**

**S 192.168.3.0/24 [1/0] via 10.0.1.4**

**S\* 0.0.0.0/0 [1/0] via 10.0.1.2**

Router#