

12. Въведение в маршрутизацията.

Маршрутни алгоритми.

Софтуер за маршрутизация
Quagga Routing Suite с отворен
код. Инструментариум iproute2.

Статична маршрутизация.

Маршрутни алгоритми

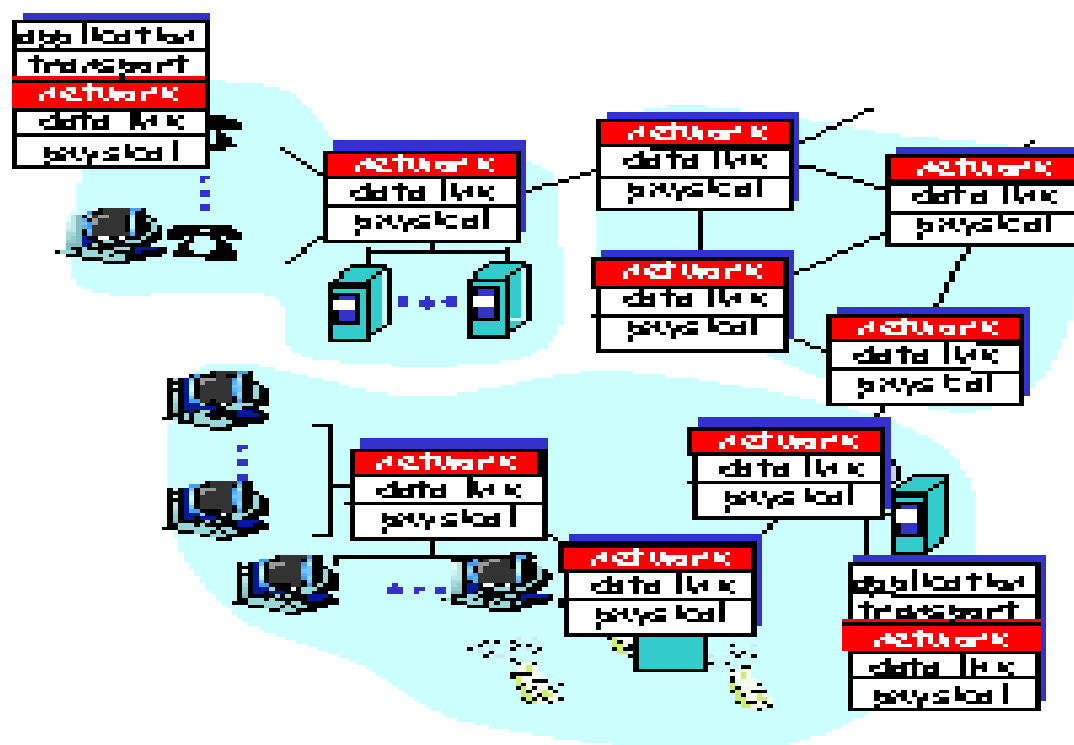
Основната функция на мрежовото ниво е да маршрутизира пакетите от източника към получателя - през няколко хопа.

Маршрутен алгоритъм е част от софтуера на мрежовото ниво, която определя по коя от изходните линии да се изпрати пристигнал пакет. За целта всеки маршрутизатор притежава **маршрутна таблица**.

Ако мрежата е с **пакетна комутация** (дейтаграми), решението трябва да се взима наново за всеки пристигнал пакет, тъй като оптималният маршрут може да се е променил.

Ако се използва **виртуален канал**, решенията по маршрутизацията се взимат при създаването му.

Функции на мрежовия слой



Маршрутизиращи протоколи

Маршрутизиращите протоколи трябва да отговарят на множество изисквания.

Да са достатъчно прости и лесни за конфигуриране и да осигуряват надеждна и стабилна работа на мрежата.

Да реагират своевременно на отпадане на маршрутизатори или връзки между тях.

Да бъдат в състояние да открият алтернативни пътища за доставяне на пакетите, ако такива съществуват.

Маршрутизиращи протоколи

Две други цели на маршрутизиращите протоколи си противоречат (на пръв поглед):

- **минимизиране на времето за закъснение** (по-малък престой на пакетите в междинните възли);
- **максимизиране на общия поток** предполага буферите в маршрутизаторите да работят на максимален капацитет.

Освен това максимизирането на общия поток може да влезе в противоречие с изискването мрежовите ресурси да могат да се използват от всички потребители в мрежата.

Маршрутизиращи алгоритми

Маршрутизиращите алгоритми са два вида - **неадаптивни** и **адаптивни**.

При **неадаптивните** маршрутизацията не се извършва на базата на текущата топология на мрежата.

Маршрутите между всеки два възела в мрежата се изчисляват предварително и **се записват ръчно от мрежовите администратори**, след което влизат в маршрутните таблици.

При промяна на топологията на мрежата (например при отпадане на възел или на връзка), администраторите ръчно трябва да променят маршрутите.

Това прави неадаптивните алгоритми приложими само в малки мрежи, при които рядко настъпват промени.

Неадаптивните алгоритми се наричат още **статични**.

Попълване на маршрутна таблица. Пример.

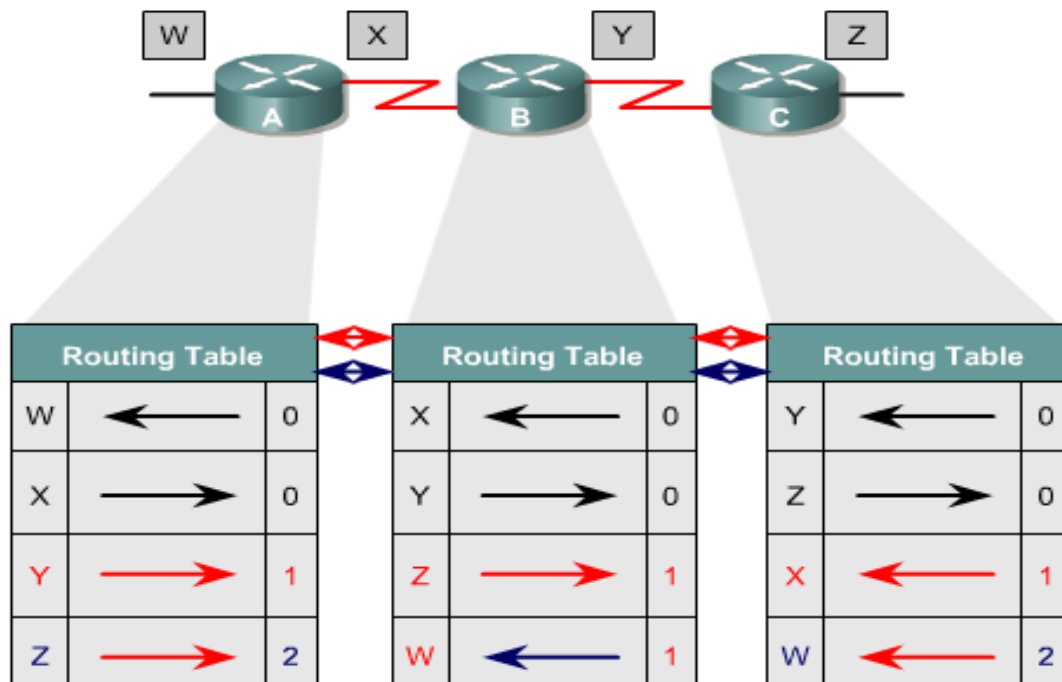
Мрежата представяме като **граф** - върховете са възлите в мрежата, а дъгите са комуникационните линии.

Метриката в графа се определя на базата на разстоянието до крайната точка (**хопове**), време-закъснението за минаване на един пакет,

надеждност на линията, цена, брой хопове и др. Възможно е да се комбинират една или повече от изброените характеристики.

Попълване на маршрутна таблица. Хопове.

Метриката се определя от това през колко маршрутизатора (хопа) ще мине пакета до крайната точка (мрежа).



Адаптивни алгоритми. Скорост на сходимость (конвергенция)

При адаптивните алгоритми маршрутните таблици се променят динамично, за да отразяват промени в топологията и натовареността на трафика.

Важна характеристика на адаптивния алгоритъм е неговата **скорост на сходимость**: времето, което е необходимо да се преизчислят маршрутните таблици на всички маршрутизатори в мрежата при промяна в топологията или трафика. (конвергенция)

Адаптивни алгоритми. Принцип за оптималност

Оптималните пътища между всеки два възела в мрежата се изчисляват по някои от алгоритмите за намиране на **най-къс път в граф** (след като е въведена метрика в графа, представлящ мрежата).

Всички тези алгоритми се базират на **принципа за оптималност**:

всяка част от оптимален път е също оптимален път между съответните два върха.

Като следствие от този принцип, **оптималните пътища от един връх към всички останали образуват дърво (sink tree)**.

Алгоритъм на Дейкстра

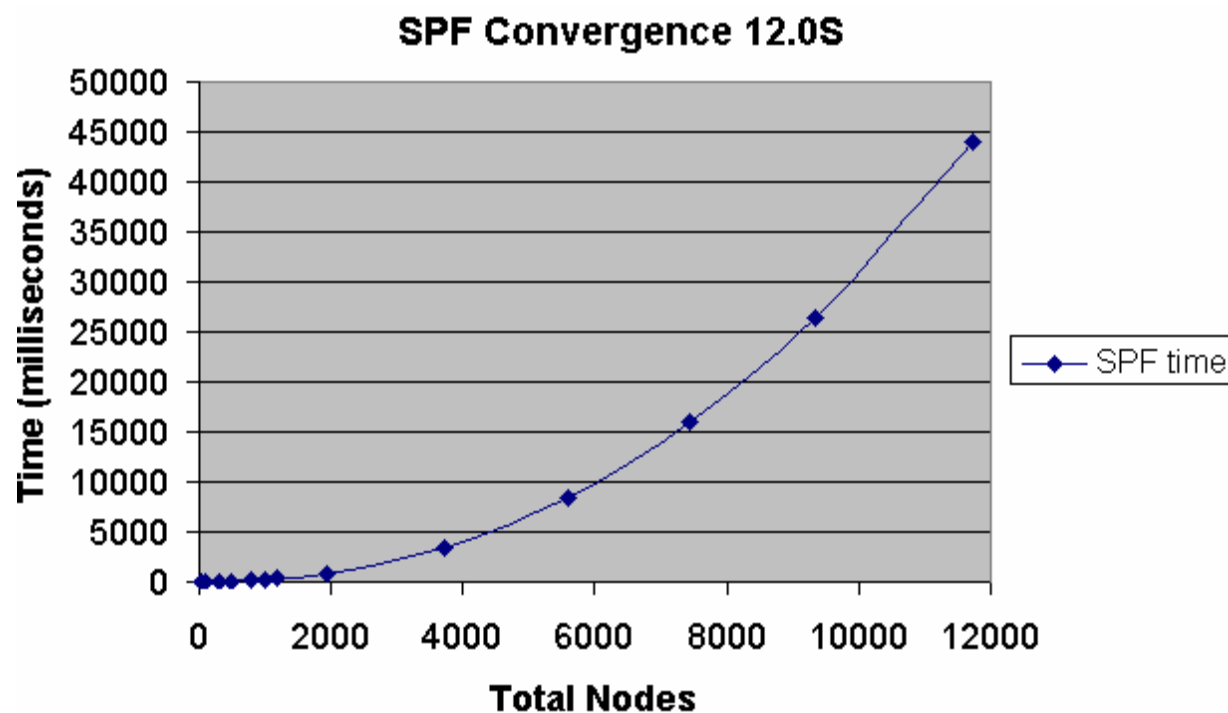
Алгоритъмът на Дейкстра е алгоритъм за намиране на най-къс път в граф от даден връх до всички останали върхове.

Важно е да се отбележи, че при алгоритъма на Дейкстра теглата на ребрата трябва да са положителни.

Резултатът от алгоритъма е дърво на оптималните пътища от дадения връх до всички останали.

The Shortest Path Tree (SPF).

Сходимость (конвергенция)



Вътрешни и външни протоколи

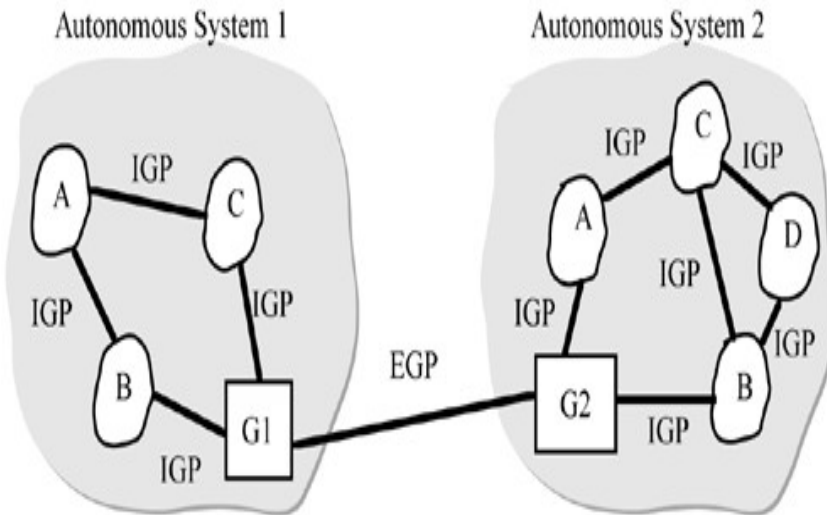
IGP vs EGP

IGP:

- **DV**: RIP, (E)IGRP
- **LS**: OSPF, IS-IS

EGP:

BGP 4/4+



Quagga



Quagga е **open source** софтуерен пакет за маршрутизация.

Поддържа: RIPv1, RIPv2, RIPvng, OSPFv2, OSPFv3, BGP-4 и BGP-4+ (т.е **IPv4** и **IPv6**)

Quagga рутер



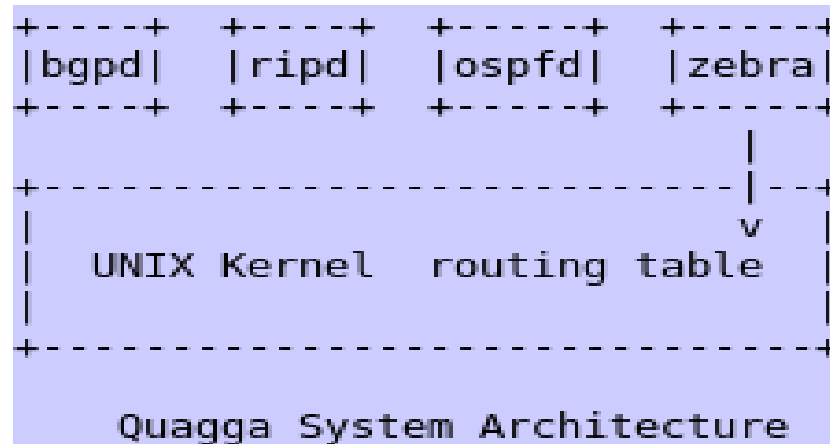
Quagga рутер

Компютър с Quagga си е рутер със **Cisco CLI**.

Обменя информация за маршрутите с помощта на маршрутни протоколи.

Quagga я използва, за да обновява таблицата с маршрутите в ядрото.

Quagga. Архитектура.



Мултипроцесна архитектура.

(все още няма **multi-thread**)

Всеки демон - `.conf` файл и терминал.

Статичен маршрут – `zebra.conf`

BGP - `bgpd.conf`

Ядро на Linux рутер

```
less /etc/sysctl.conf
```

```
...
```

```
# Controls IP packet forwarding
```

```
net.ipv4.ip_forward = 1
```

```
...
```

```
net.ipv6.route.max_size = 15000000
```

```
net.ipv4.route.max_size = 15000000
```

XORP



XORP рутер с интерфейс на **Juniper**.

Също е **open source** платформа за IPv4 и IPv6 маршрутизация.

Поддържа OSPF, RIP, BGP, OLSR, VRRP (Virtual Router Redundancy Protocol), PIM, IGMP (Multicast).

iproute2

Iproute2 – сбор от средства за контрол на TCP/IP мрежи и трафик в Linux.

Пример: добавя адрес 10.0.0.1 с префикс 24 (255.255.255.0) и стандартен broadcast към интерфейс eth0

```
[root@XXX]#ip addr add 10.0.0.1/24 brd +  
dev eth0
```

```
[root@XXX]#ifdown eth0
```

```
[root@XXX]#ifup eth0
```

iproute2

Показване на състоянието:

```
[root@XXX]# ip address show dev eth0
```

или

```
[root@XXX]# ip a [ls eth0]
```

Изтриване на адрес:

```
[root@XXX]# ip addr del 10.0.0.1/24 dev  
eth0
```

Статична маршрутизация

Ръчното добавяне на маршрути в конфигурацията на маршрутизатор се нарича **статична маршрутизация (static routing)**. Подходящо е за малки мрежи. Маршрутите се описват чрез фиксирани пътища (**статични маршрути**), които се въвеждат в маршрутизатора от мрежовия администратор.

Статична маршрутизация

Така чрез статични маршрути (**static routes**) може да се опише цялата мрежа.

Този подход не е отказоустойчив. В случай на промени или прекъсвания на връзки и/или устройства не се получава автоматично пренасочване на трафика.

Статична маршрутизация

В някои случаи статичните маршрути са даже за предпочитане, даже влияят положително на производителността.

Това са мрежите с един единствен изход (**stub networks**) и маршрутите по подразбиране (**default routes**).

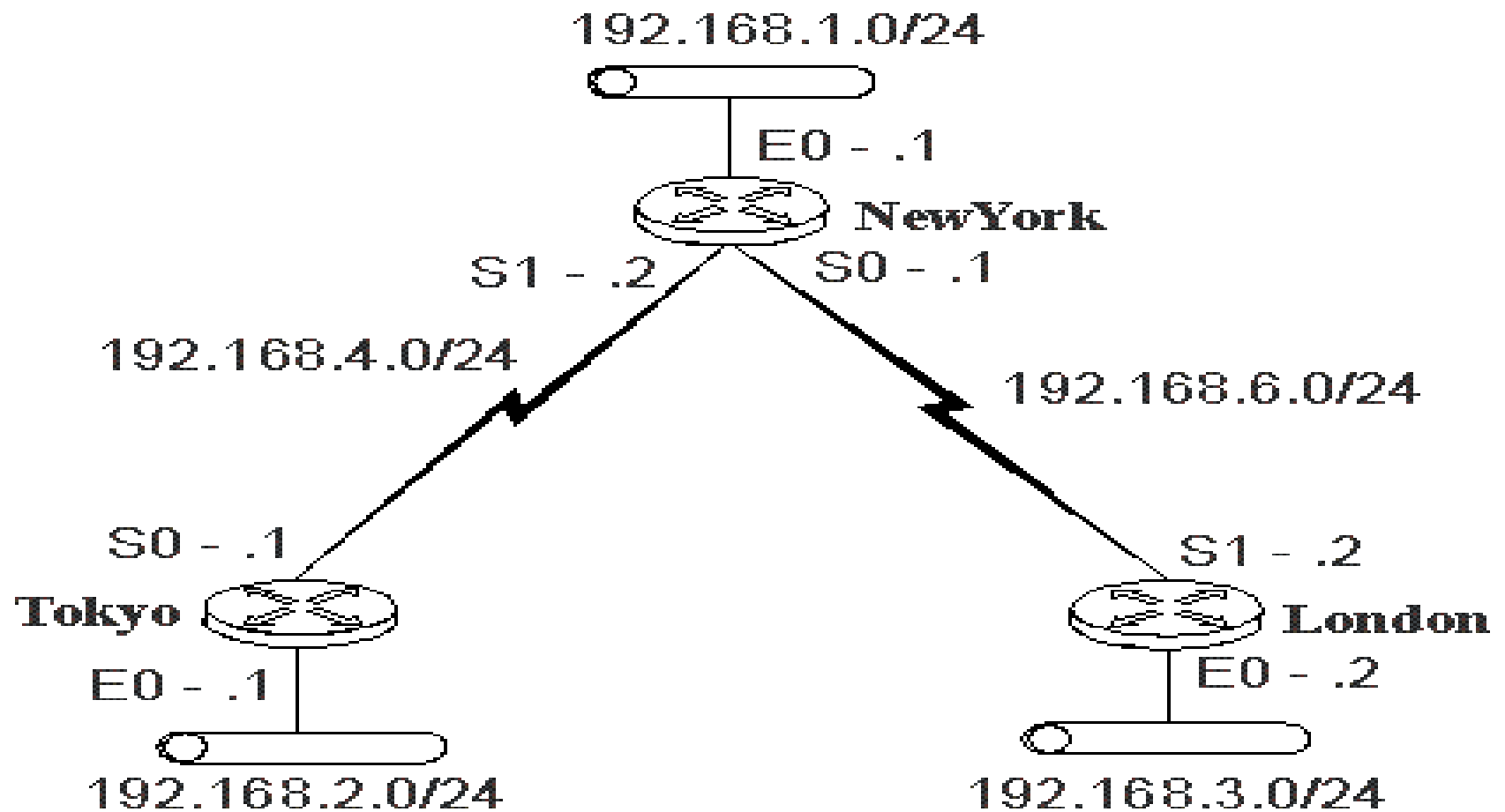
Stub Networks

Статичните маршрути най-добре се вписват в **Stub** мрежите.

stub network е като “задънена улица” – само с един изход към външния свят.

На фигурата по-долу двете клонови локални мрежи - Tokyo и London са с по един изход към главната квартира.

Stub Networks



Stub Networks

Не е необходимо да пускаме динамичен протокол за маршрутизация по WAN линиите между NewYork и клоновете.

Статични маршрути отвеждат трафика до “клоновете” LAN-ве.

Докато клоновите маршрутизатори е необходимо да “знаят” само дали пакета е насочен към мрежа извън техните локални.

Т.е да го “докарат” до NewYork, който има по-богата маршрутна таблица.

Конфигурации

По-долу са дадени статичните маршрути в NewYork и Tokyo.

Редове 10 и 11 са статичните маршрути в NewYork към съответните “клонови LAN-ве”.

Ред 19 е по подразбиране (default route) в Tokyo (маршрут до мрежа 0.0.0.0), който сочи обратно пътя към NewYork.

Конфигурацията на London е подобна на Tokyo.

Конфигурации: static routes

NewYork Configuration

...

10) ip route 192.168.2.0 255.255.255.0
192.168.4.1

11) ip route 192.168.3.0 255.255.255.0
192.168.6.2

!!! В конфигурацията освен мрежата и маската е зададен и IP адреса следващия възел по пътя (**Next Hop**)

Конфигурации: default route

Tokyo Configuration

...

```
19) ip route 0.0.0.0 0.0.0.0 192.168.4.2
```

Частен случай: интерфейсни маршрути

Статичен маршрут, на който е зададен само **изходящия интерфейс** на рутера, се нарича още **интерфейсен**.

В този случай все едно, че мрежата на крайната точка е директно закачена за интерфейс на рутер.

Интерфейсни маршрути

```
NewYork#conf t
```

```
NewYork(config)#no ip route 192.168.2.0  
255.255.255.0 192.168.4.1
```

```
NewYork(config)#no ip route 192.168.3.0  
255.255.255.0 192.168.6.2
```

```
NewYork(config)#ip route 192.168.2.0  
255.255.255.0 s 1
```

```
NewYork(config)#ip route 192.168.3.0  
255.255.255.0 s 0
```


Интерфейсни маршрути

NewYork#sh ip route

...

C 192.168.4.0/24 is directly connected, Serial1

C 192.168.6.0/24 is directly connected, Serial0

C 192.168.1.0/24 is directly connected, Ethernet0

S 192.168.2.0/24 is directly connected, Serial1

S 192.168.3.0/24 is directly connected, Serial0

Default Route

Маршрут по подрабиране (**default route**) се използва, когато в таблицата няма друг маршрут до целта.

Това е маршрут до мрежата 0.0.0.0. Изходът от командата **show ip route** показва на първия ред на маршрутната таблица:

Gateway of last resort is 192.168.4.1 to network 0.0.0.0

Пакет, който не намери съвпадение в маршрутната таблица, поема към **“gateway of last resort”**.

Това е рутер с по-подробна информация за маршрутите.

Ако няма default route и адреса на получателя не бъде открит в таблицата, пакетът се изхвърля и на IP адреса на източника се въща ICMP съобщение: **‘Destination or Network Unreachable’**.

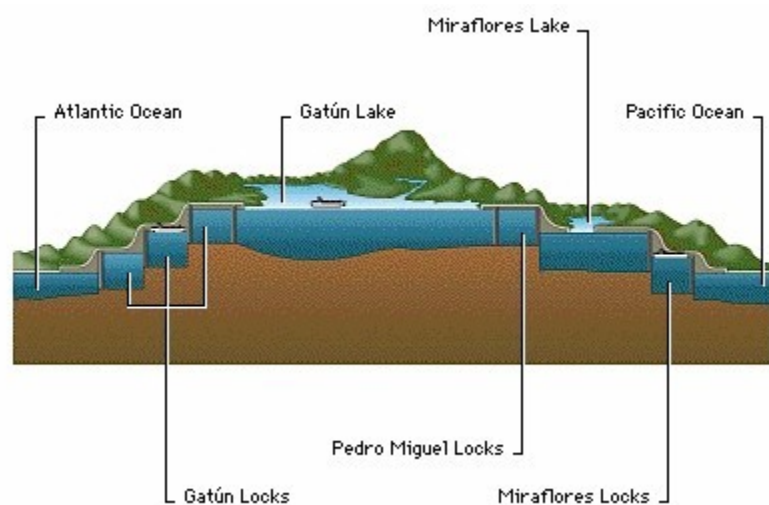
Терминът Gateway

‘Gateway’ е по-стария термин за маршрутизатор (рутер). През него можеше да се изпратят пакети към мрежа с различна преносна среда и канални протоколи.

В днешно време показва IP адреса на устройството, от което се излиза от локалната мрежа към “ВЪНШНИЯ СВЯТ”. И обикновено е с два интерфейса – вътрешен и външен.

Докато маршрутизаторът има повече от два интерфейса и изпълнява по-сложни функции по маршрутизацията.

Шлюзове: Панамский канал



Default Route. Конфигуриране.

```
Tokyo(config)#ip route 0.0.0.0 0.0.0.0 192.168.4.2
```

...

```
Tokyo#sh ip route
```

...

```
Gateway of last resort is 192.168.4.2 to network  
0.0.0.0
```

```
C 192.168.4.0/24 is directly connected, Serial0
```

```
C 192.168.2.0/24 is directly connected, Ethernet0
```

```
S* 0.0.0.0/0 [1/0] via 192.168.4.2
```

Default Route в IPv6

Спомнете си:

<https://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml>

Сегментът за "unicast" IPv6 мрежовите алокации се описва с префикса на мрежа **2000::/3 (Global Unicast)**.

Default Route в IPv6

Следователно:

Маршрутът по подразбиране за "unicast" следва да се анонсира като 2000::/3:

```
ipv6 route 2000::/3 lo
```

Защо изходящ интерфейс е
`loopback`?

default равносилно ли е на 2000::/3

```
[stefan@shuttle ~]$ ip -6 route show default  
default via 2001:67c:20d0:10::5 dev eth0  
proto static metric 1024 mtu 1500 advmss  
1440 hoplimit 4294967295
```

```
[stefan@shuttle ~]$ ip -6 route get ::/0  
need at least destination address
```


default равносилно ли е на 2000::/3

НЕ "default" се разбира цялото IPv6 пространство, което все още не е алокирано:

<https://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml>

Защо обявяваме 2000::/3 за маршрут по подразбиране"?

Изолира се паразитния трафик, който може да се причини от грешки и злонамерен код.

Създаване на статичен маршрут с IPROUTE2

Добавяне на маршрут до 10.0.0.0/24 през gw 193.233.7.65

```
ip route add 10.0.0.0/24 via 193.233.7.65
```

Променяме го да минава през виртуален интерфейс dummy0

```
ip ro chg 10.0.0.0/24 via 193.233.7.65 dev dummy0
```

Изтриване на маршрут

```
ip route del 10.0.0.0/24 via 193.233.7.65
```

Създаване на IPv6 маршрут:

```
ip -6 route add fe80::20e:2eff:fed1:ab15/64  
dev dummy0
```

Изтриване:

```
ip -6 route del fe80::20e:2eff:fed1:ab15/64  
dev dummy0
```

Показване на маршрут

```
[root@shuttle ~]# ip route
```

```
62.44.109.0/26 dev eth0 proto
```

```
kernel scope link src
```

```
62.44.109.11
```

```
169.254.0.0/16 dev eth0 scope
```

```
link
```

```
default via 62.44.109.5 dev eth0
```

ip -6 route

```
[root@shuttle ~]# ip -6 route
```

```
...
```

```
fe80::/64 dev eth0 metric 256  
expires 21207257sec mtu 1500  
advms 1440 hoplimit 4294967295  
default via 2001:67c:20d0:10::5  
dev eth0 metric 1 expires  
21207261sec mtu 1500 advms 1440  
hoplimit 4294967295
```

telnet localhost zebra

```
zebra@rec-gw> en
```

```
zebra@rec-gw# sh run
```

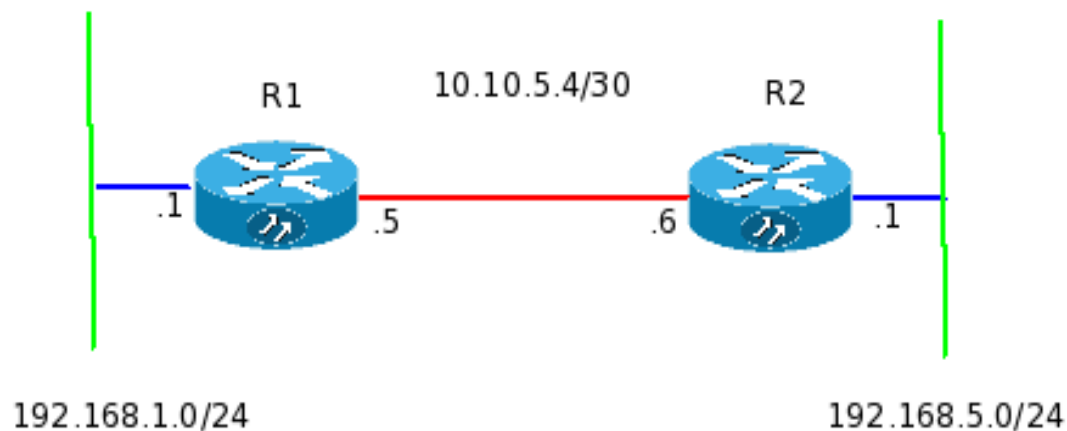
```
...
```

```
ip route 62.44.116.0/24 62.44.110.14
```

```
ip route 62.44.117.0/25 62.44.110.9
```

```
ipv6 route 2001:67c:20d1:116::/64  
2001:67c:20d1:110::14
```

Задача



- На R1 създайте статичен маршрут до 192.168.5.0/24.
- Защо `ping 192.168.5.1` работи, а
- `ping -I 192.168.1.1 192.168.5.1` не работи?