Въведение в маршрутизацията.

Маршрутни алгоритми.

Софтуер за маршрутизация с отворен код. Инструментариум iproute2.

Статична маршрутизация.

Маршрутизацията: от графите към мрежите

Както казахме още в началото, теорията на компютърните мрежи – топология, **маршрутизация** – се базира на теорията на **графите**.

Маршрут в граф G=(X, U) се нарича една крайна последователност от ребра: s=(x0, x1)(x1, x2),…, (xl-1, xl), където x0 и xl – са началният и крайният връх съответно. Броят на ребрата в маршрута е неговата дължина.

От теория на графите

Маршрут, в който няма повтаряне на ребра, се нарича верига. Ако всички върхове в маршрута са различни, нарича се проста верига. Верига, в която началнят и крайният връх съвпадат (т.е x0=xl), се нарича цикъл (**loop**).

Свързан граф без цикли е дърво. Началният връх е корен, от който излизат ребра, наречени клони. Всеки два върха в дървото са свързани с една единствена верига. Във всеки свързан граф G е възможно да се отдели някакво дърво T.

Мерен граф

Теглови или мерен граф (Weighted graph) е граф, чиито ребра имат тежести (стойнсти), които обикновено са цели положителни числа. Тежестта на дадено ребро може да е:

- Стойност или дистанция = количеството усилие, необходимо за пропътуване от едно място до друго;

- Капацитет = максималното количество на поток, който може да бъде транспортиран от едно място до друго.

Маршрутизацията в компютърните мрежи

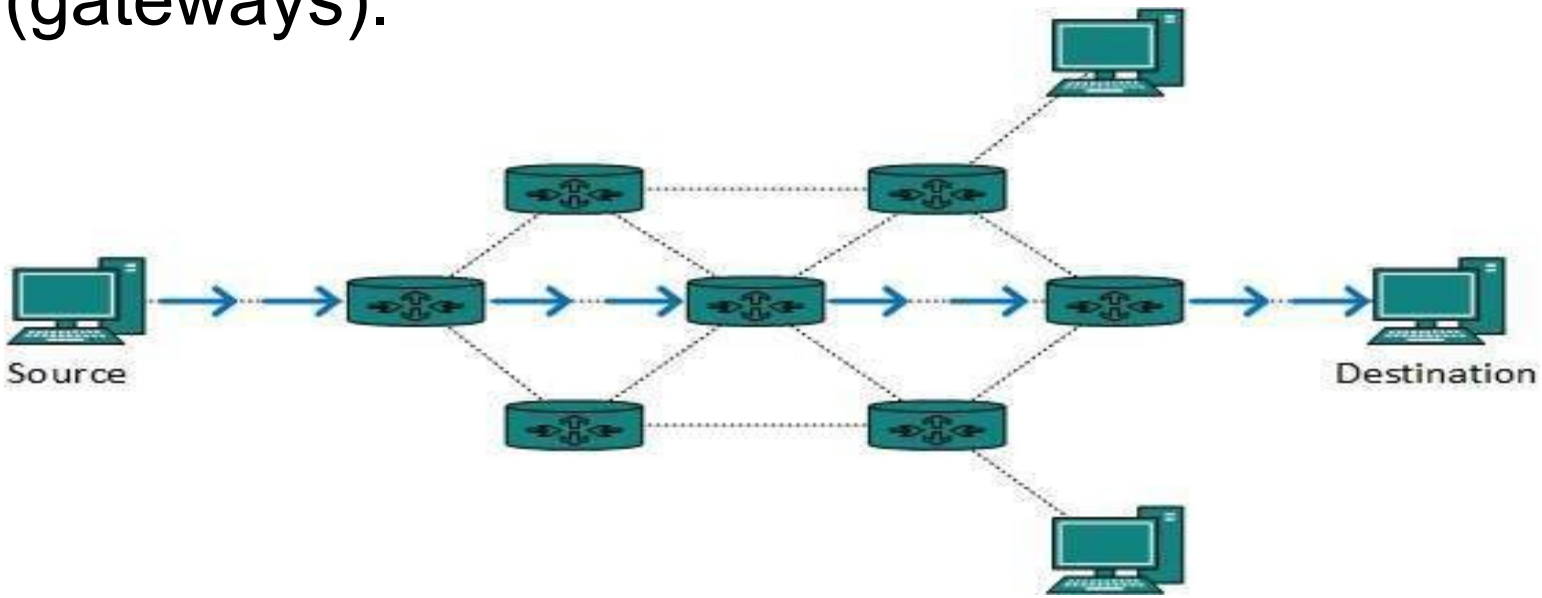
Маршрутизацията е **процес на избор на път без зацикляния** за трафика вътре в една мрежа или през множество от мрежи - компютърни, телефонни, електрически и др.

При IP мрежите с пакетна комутация маршрутизацията е процес на вземане на решение за определяне на посоката на мрежовите пакети от възела-сорс до възела-дестинация през междинни възли.

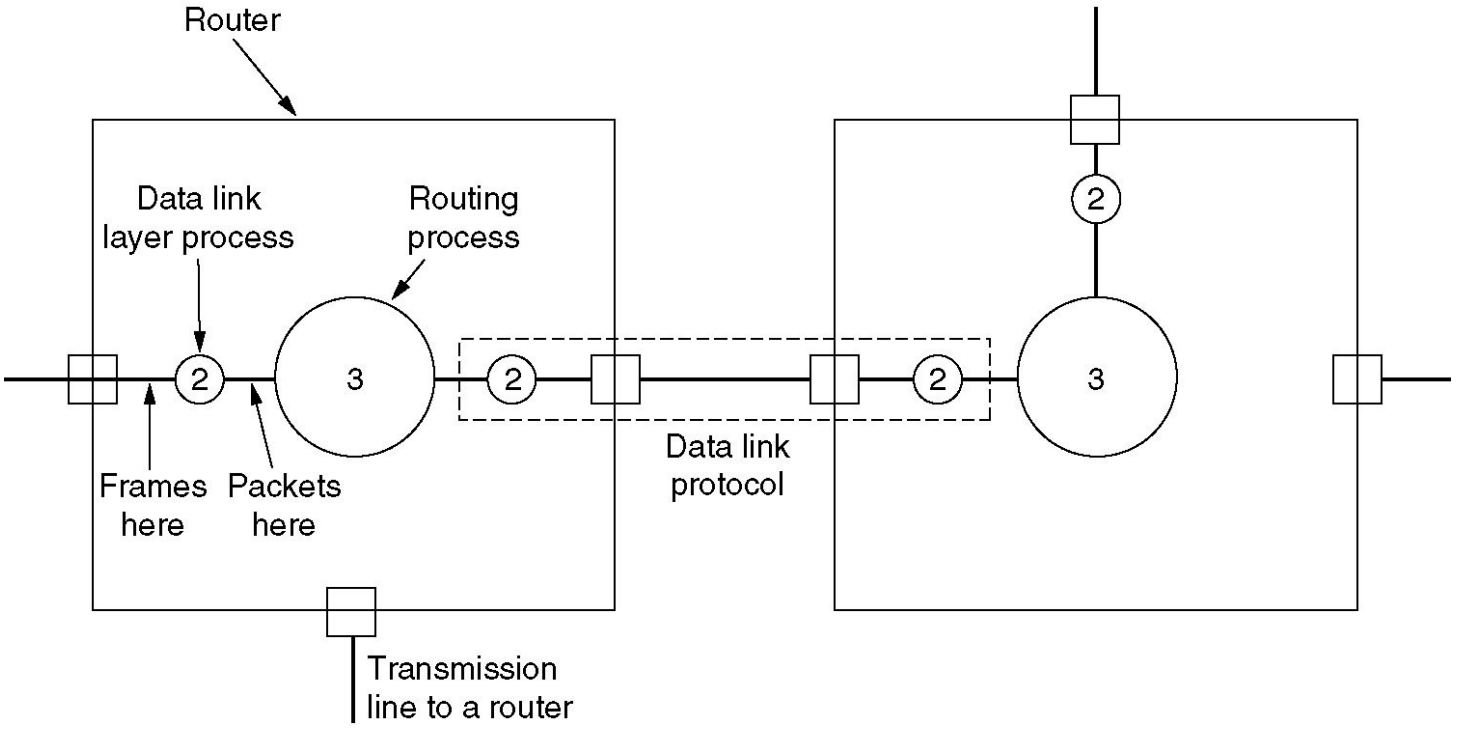
От възел на възел се прилагат различни техники на пренос в зависимост от особеностите на конкретната комуникационна линия.

Маршрутизацията в компютърните мрежи

Тези междинни възли са мрежов хардуер като маршрутизатори (рутери), L3 суичове, компютри с качен софтуер за маршрутизация или обикновени шлюзове (gateways).



Маршрутизация и техники на пренос



Маршрутни алгоритми и маршрутизатори

Маршрутен алгоритъм е част от софтуера на мрежовото ниво, който се поддържа в маршрутизаторите (които са си компютри) и определя по коя от изходните линии да се изпрати пристигнал пакет. За целта всеки маршрутизатор притежава **маршрутна таблица**.

Ако мрежата е с пакетна комутация, 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.

Таблица с маршрути

Таблицата с маршрутите съдържа записи, които включват:

- Код на протокола – C, S, B, R...;
- адрес на IP мрежата – дестинация;
- [Админ. Дистанция / Метрика] ;
- Next Hop IP address;
- изходен интерфейс.

Пример на таблица с маршрути

```
# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 10.10.10.0/28 is directly connected, eth1.704
C>* 10.10.128.0/24 is directly connected, eth1.190
C>* 10.20.109.0/24 is directly connected, eth1.996
...
S>* 0.0.0.0/0 [1/0] is directly connected, Null0
S>* 62.44.96.132/30 [1/0] via 62.44.109.35, eth1.109
S>* 62.44.97.192/27 [1/0] via 62.44.97.238, eth1.6
S>* 62.44.107.128/25 [1/0] via 62.44.96.182, eth1.801
...
B>* 1.0.0.0/24 [200/0] via 62.44.127.21, eth0
B>* 1.0.4.0/22 [200/0] via 62.44.127.21, eth0
B>* 1.0.4.0/24 [200/0] via 62.44.127.21, eth0
```

Избор на най-добрия път

Рутерите правят избор на най-добрия път, прилагайки метрика на маршрутите.

Повечето алгоритми използват само един маршрут в даден момент, но има и възможност за прилагане на множество алтернативни пътища в един и същи момент, напр. с цел балансиране на трафика.

Метрика

Метриката се изчислява от алгоритмите за маршрутизация въз основа на:

скорост на линията в bit/s (bandwidth), закъснение (delay), hop count - брой на междинните възли, цена на пътя (path cost), натоварване, MTU, надеждност и цена на комуникацията (лева/бит).

В таблицата с маршрутите се записват само най-добрите маршрути, а топологичната база от данни (link-state database - **LSDB**) съхранява всички възможни маршрути.

Повече от един маршрут

При повече от един маршрут до дадена дестинация в таблицата се записва този с:

- най-широка мрежова маска (Prefix-Length), т.е най-конкретният маршрут;
- с по-малка метрика/цена;
- с по-малка административна дистанция, т.е - по-надеждния протокол за машрутизация.

Според Cisco Systems протоколът OSPF, базиран на алгоритъма на Дейкстра, е за предпочитане пред RIP, базиран на дистантния вектор (Bellman-Ford).

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.

Видове маршрутизиращи алгоритми

Маршрутизиращите алгоритми са два вида:
неадаптивни - **статични** и адаптивни - **динамични**.

При неадаптивните 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.

Динамична маршрутизация

Динамичната маршрутизация решава проблема чрез автоматично създаване на таблици с маршрути, базирани на информация, която се обменя между рутерите в зависимост от протокола за марщрутизация, който използват в момента.

Мрежата работи почти автономно, избягвайки откази и/или прекъсвания на връзката в определени места.

Скорост на сходимост (конвергенция)

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

По тази причина важна характеристика на адаптивния алгоритъм е неговата **скорост на сходимост (конвергенция)**:

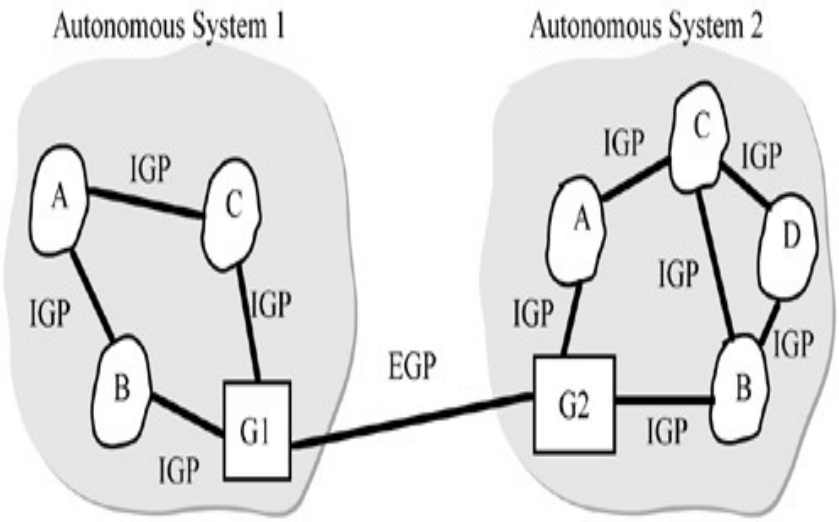
Състоянието, при което всички рутери в една мрежа имат **една и съща информация за топологията ѝ**. Тази информация се събира чрез размяна на ъпдейти между тях, дефинирани от конкретния протокол за маршрутизация.

Защо е важна скоростта на сходимост

Скоростта е времето, което е необходимо, за да се преизчислят маршрутните таблици на всички рутери в мрежата при промяна в топологията, така че да се получи конвергенция.

Мрежата не може да бъде функционална, ако не настъпи конвергенция.

Видове протоколи според алгоритмите за маршрутизация и периметър на действие



IGP (interior gateway protocol):

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

EGP (exterior gateway protocol):

Path vector: BGP 4/4+

Протоколи с дистантен вектор

Протоколите с дистантен вектор (Distance-vector - DV), прилагат алгоритмите на Bellman–Ford и Ford–Fulkerson.

Обработват вектори (масиви) от дистанции до другите възли в мрежата.

Това е първият алгоритъм, приложен в ARPANET.

Най-добрият маршрут се определя по броя на хоповете (протокол RIP), но някои прилагат по-сложна метрика (EIGRP).

Протоколи със следене на състоянието на връзката

Този алгоритъм предполага всеки възел (рутер) да си създаде топологична карта на мрежата (**LSDB**).

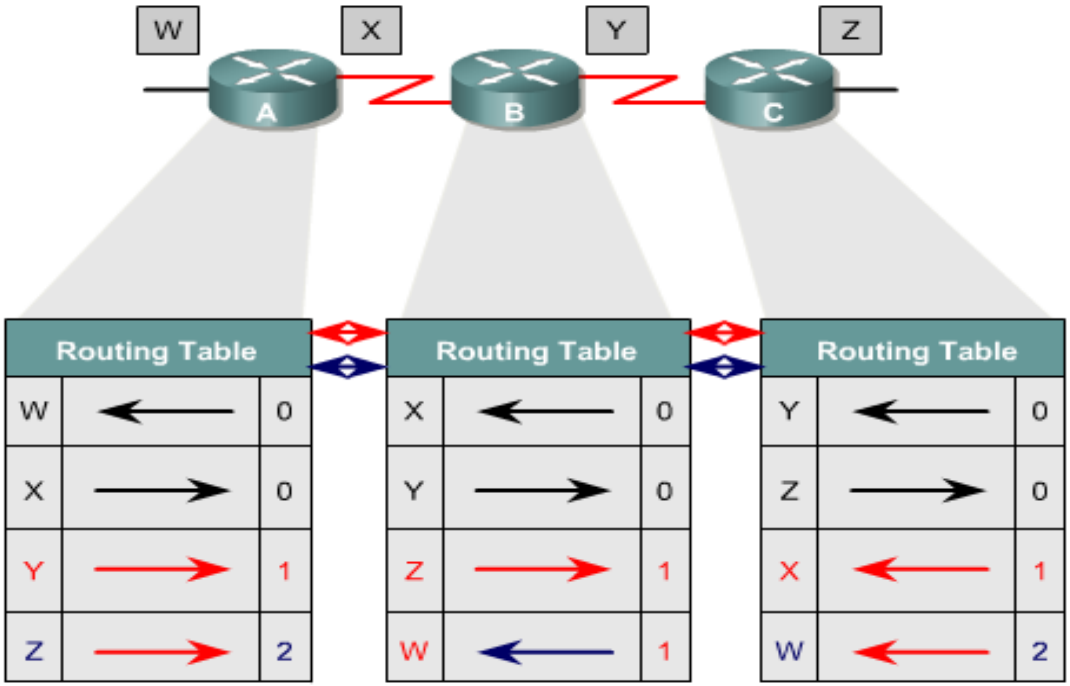
За целта “наводнява” мрежата с информация за преките си връзки към други възли.

С помощта на LSDB възелът определя пътищата с най-ниска стойност от него до всеки друг възел, прилагайки **алгоритъма на Dijkstra**.

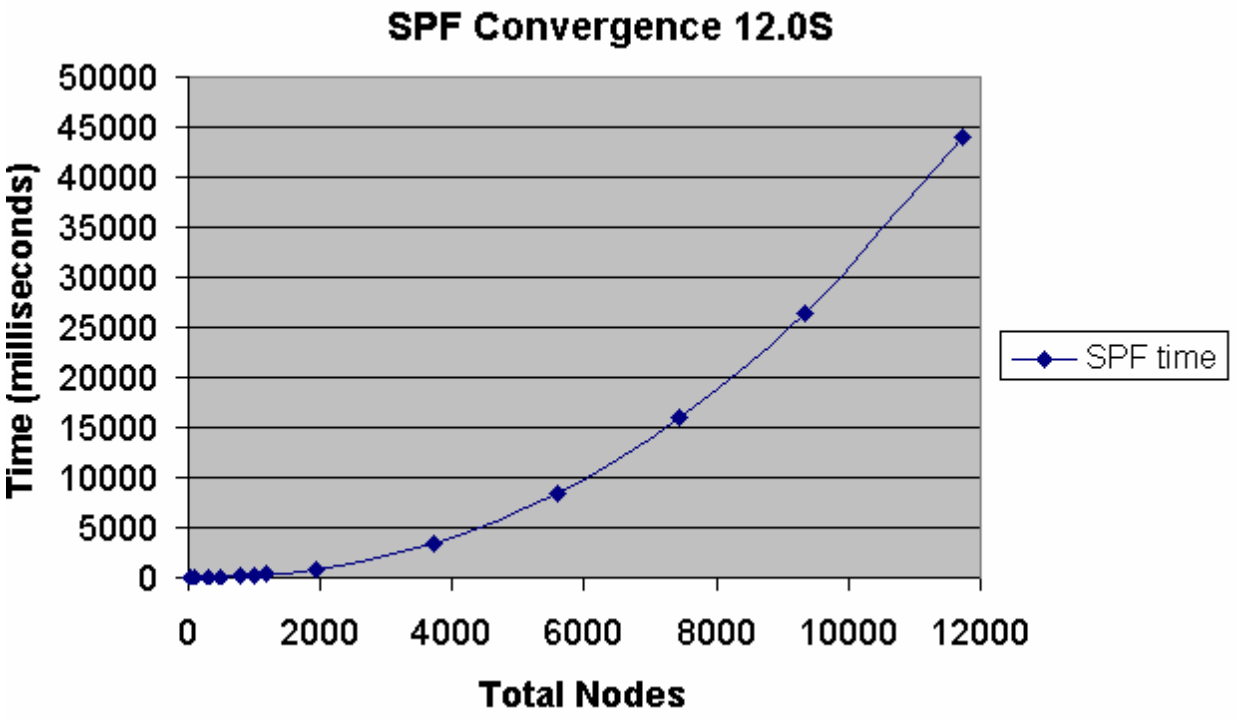
Така всеки възел си изгражда дърво на най-късите пътища (**SPF Tree**), по което се създава таблицата с маршрутите.

Конвергенция при DV протокол

Мрежата се състои от три маршрутизатора. Конвергенция настъпва на 3-ти пас. Времето на сходимост е дълго – 7 минути при 15 възела.



Време на сходимост при LS протоколи – сравнително ниско при мрежи с под 1000 възела



Вектор на пътищата за външна маршрутизация

Както видяхме при DV и LS времето за сходимост расте бързо с растенето на броя на възлите. Расте и необходимостта от ресурси – CPU, RAM.

Затова те се прилагат за вътрешна маршрутизация, в рамките на автономна система, където броят на възлите е ограничен и предвидим.

В глобалната мрежа се прилага вектора на пътищата (Path-vector routing).

Вектор на пътищата за външна маршрутизация

Path-vector е подобен е на DV. Един или повече възли в автономната система (АС) играе роля на говорител (speaker).

speaker създава таблицата с маршрутите и я рекламира към speaker-и в съседни АС.

Само speaker в дадена АС може да комуникира със себеподобните си от други АС.

Рекламира пътища през АС, но не и метрика.

FRRouting

FRR (**Free Range Routing** - https://frrouting.org/) е софтуер за IP маршрутизация, написан на **C**, за **nix** (**Unix-like**) **оперционни системи** – подобни на "добрата стара" Unix. Такива са Linux, FreeBSD и Mac OS X (ядрото й Darwin е базирано на BSD). Включва демони за BGP, IS-IS, OSPF, RIP и др.

Корените на FRR са в **Quagga** (www.quagga.net), също open source пакет за маршрутизация, работещ под Linux.

Стартира се на универсални компютри.

FRR/Quagga рутер

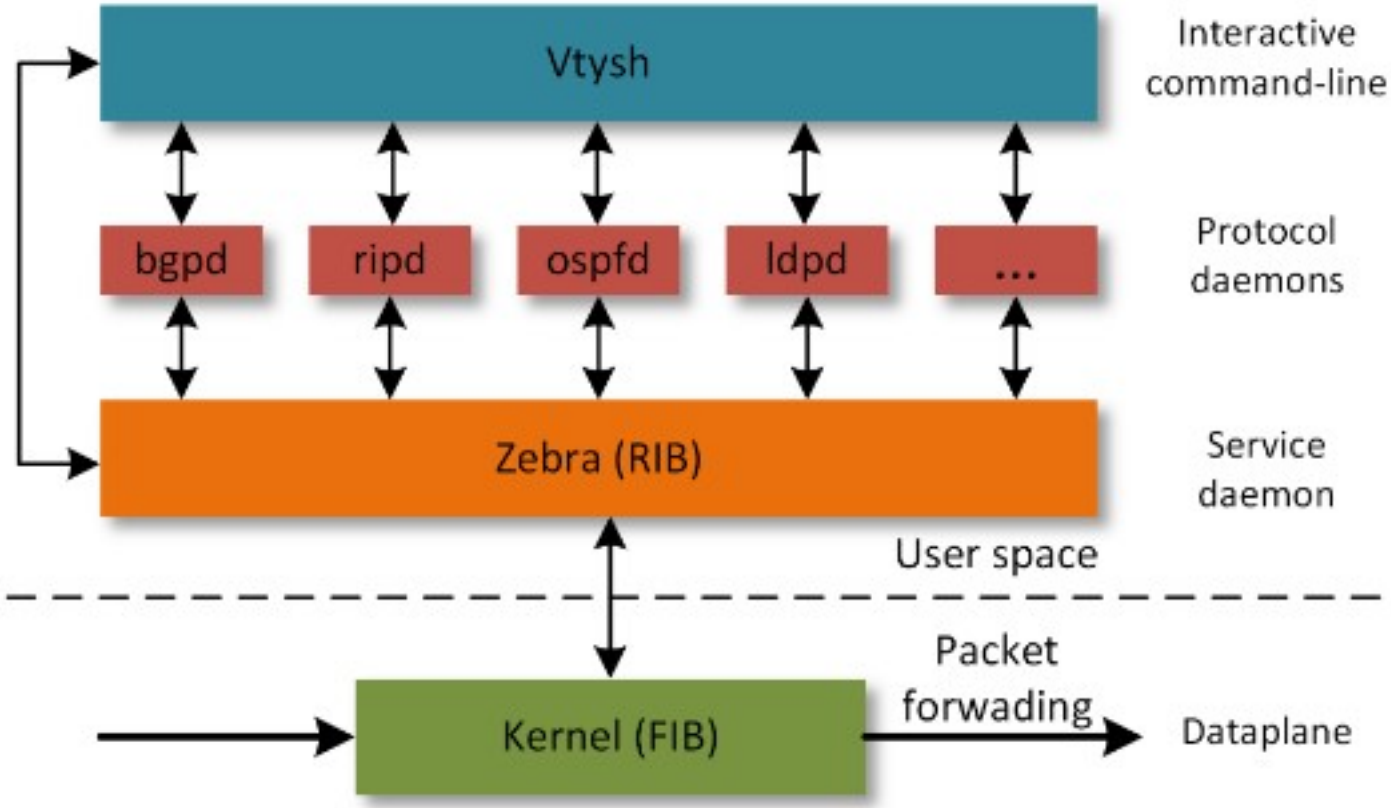
Компютър с FRR/Quagga си е рутер със **Cisco CLI**.

Обменя информация за маршрутите с помощта на маршрутни протоколи.

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

Интерфейсът от команден ред – CLI се нарича 'vty'.

FRR. Архитектура.



FRR. Архитектура.

В архитектурата на FRR всеки протокол за маршрутизация се се реализира в собствен процес (демон).

Всички тези демони “говорят” на демон-посредник (**zebra**), който отговаря за координиране на решенията за маршрутизация и насочване на данни през т.нар. Dataplane.

Ядро на Linux рутер и отражението в конфигурацията на демона zebra

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
```

…

zebra.conf:

```
…
ip forwarding
ipv6 forwarding
...
```

Още един open source проект BIRD

Целта на университетския проект (Charles University Prague) **BIRD** (bird.network.cz) е разработване на демон за IP маршрутизация за Linux, FreeBSD и др. UNIX-подобни системи. Поддържа множество маршрутни таблици, IPv4 и IPv6 RIP, OSPF, BGP и статична маршрутизация, CLI (`birdc' клиент).

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.

Този подход не е отказоустойчив. При промени или аварии нямаме автоматично пренасочване на трафика.

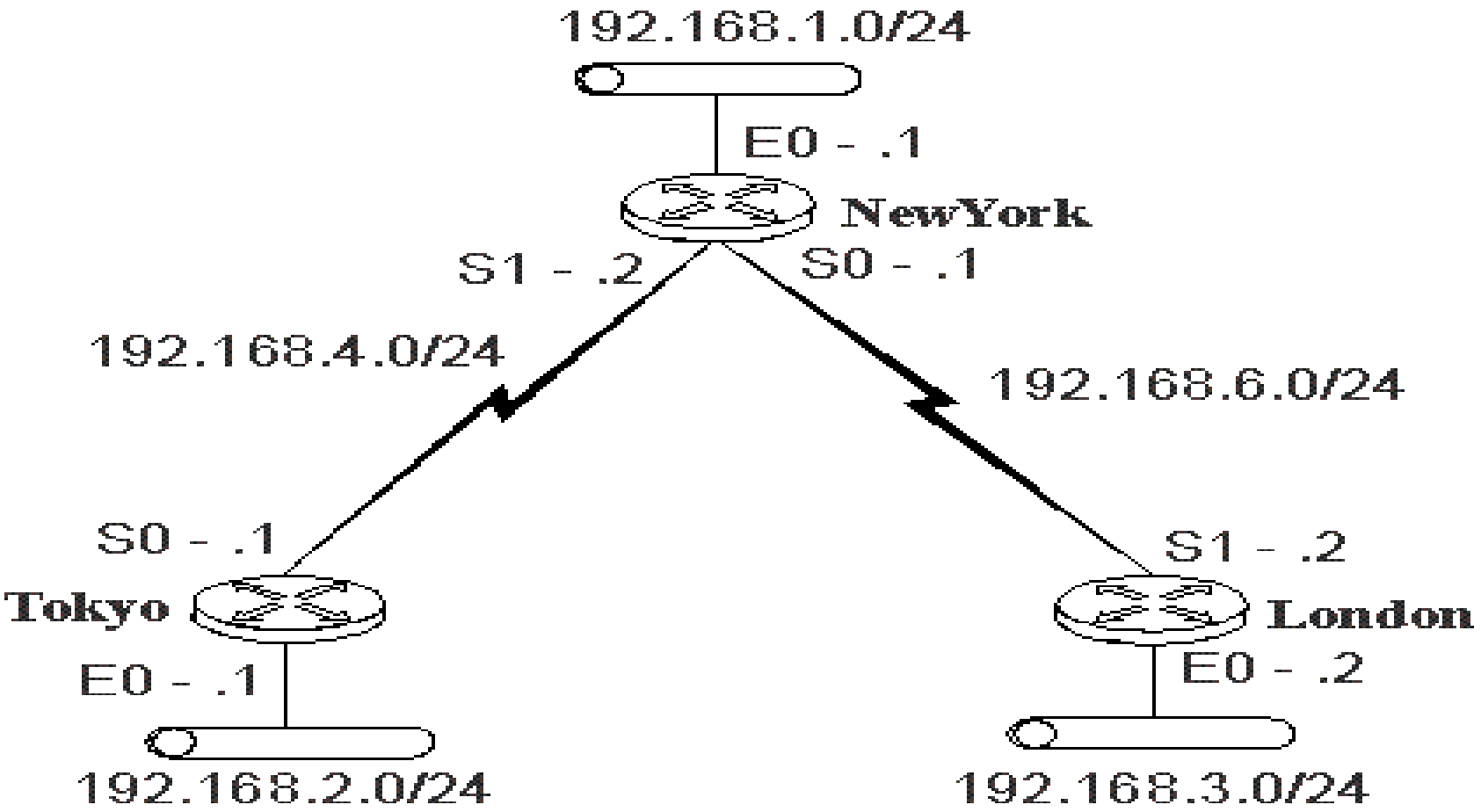
Stub Networks

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

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

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

Static & Stub Networks



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

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

Терминът Gateway

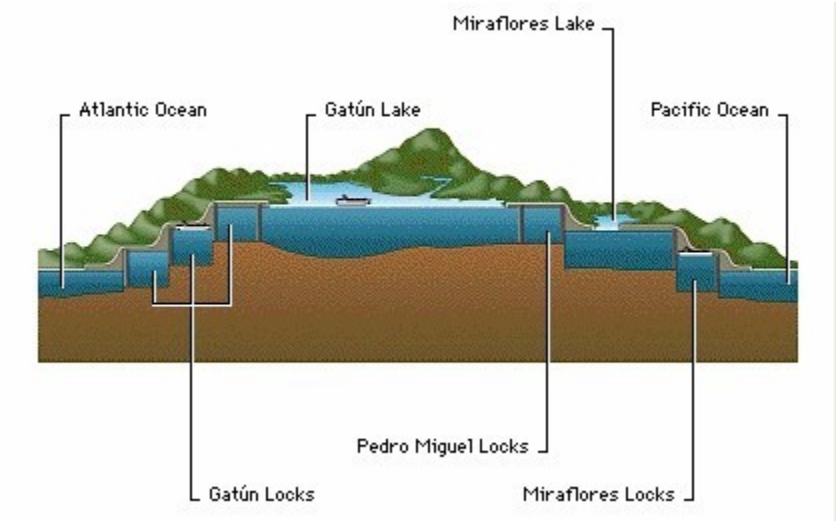
‘Gateway’ (на български шлюз) исторически е съоръжение от речното и канално корабоплаване.

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

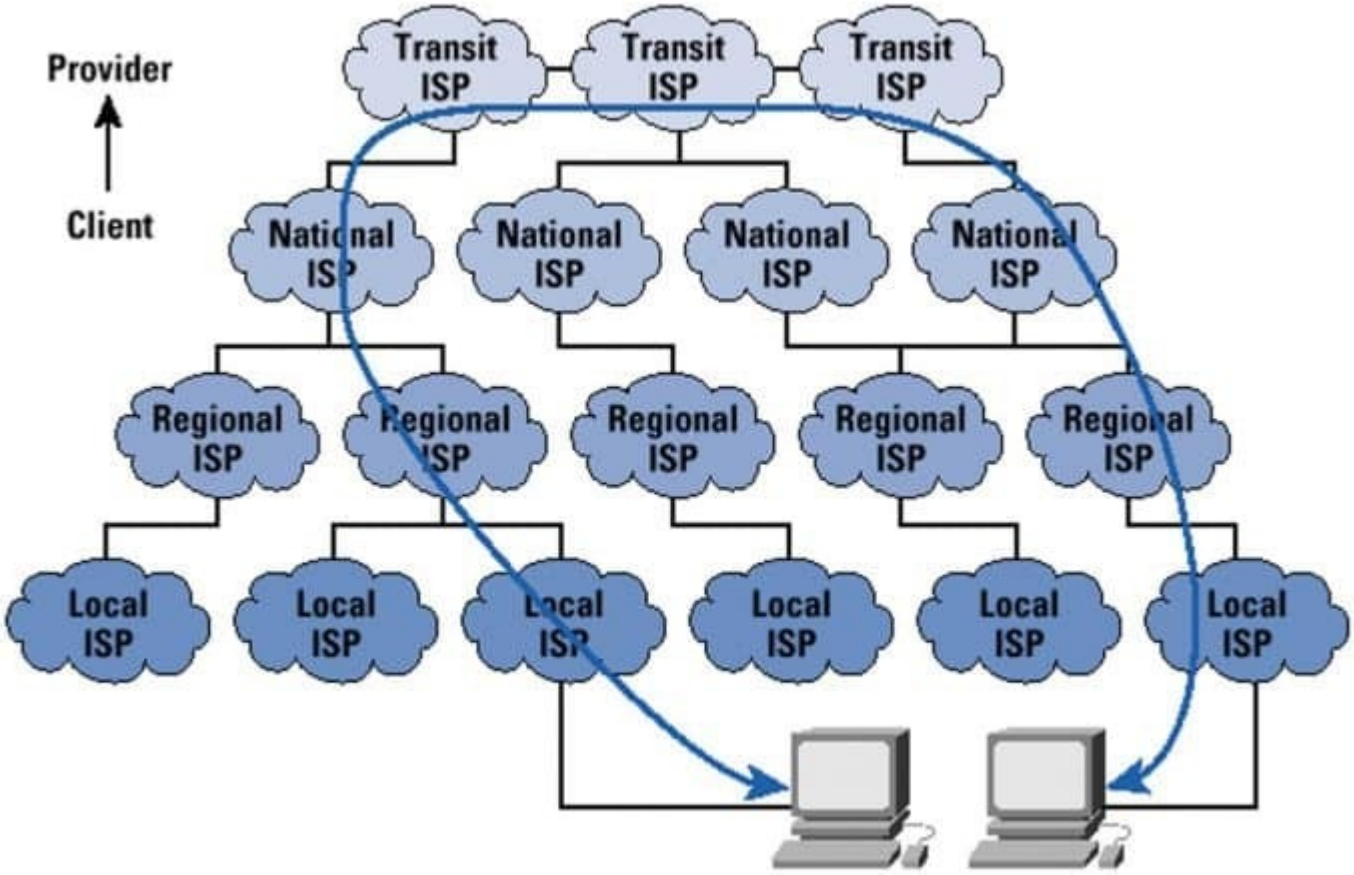
В днешно време показва IP адреса на устройсвото, от което се излиза от локалната мрежа към “външния свят” - от дадено „стъпало“ на Интернет йерархията – на по-горно.

И ма два интерфейса – вътрешен и външен. Докато маршрутизаторът има повече от два интерфейса и изпълнява по-сложни функции по маршрутизацията.

Шлюзове: Панамският канал



Gateway


Provider
Client
Transit ISP
Transit ISP
Transit ISP
National ISP
National ISP
National ISP
National ISP
Regional ISP
Regional ISP
Regional ISP
Regional ISP
Regional ISP
Local ISP
Local ISP
Local ISP
Local ISP
Local ISP
Local ISP

Default Route в IPv6

Спомняте си:

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

В момента се раздават unicast IPv6 адреси единствено и само от префикса **2000::/3 (Global Unicast)**.

Default Route ли е:

ipv6 route 2000::/3 lo

(Защо изходящ интерфейс е loopback?)

default равносилно ли е на 2000::/3

НЕ под "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.

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
[stefan@shuttle ~]$ ip -6 route get ::/128
:: via 2001:67c:20d0:10::5 dev enp1s2 proto
static src fe80::922c:2f0a:c307:a9fa metric
100 pref medium
```

С Default Route филтрираме паразитен трафик

```
ipv6 route ::/0 lo blackhole
```

Така слагаме "тапа", с която филтрираме само пакети, за които нямаме специфичен път, научен по някой протокол (BGP, OSPF6, статичен и др.). Единствен остава маршрута през /dev/null (blackhole).

За IPv4 е подобно:

```
ip route 0.0.0.0/0 Null0
```

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 IPv4 и създаване на IPv6 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

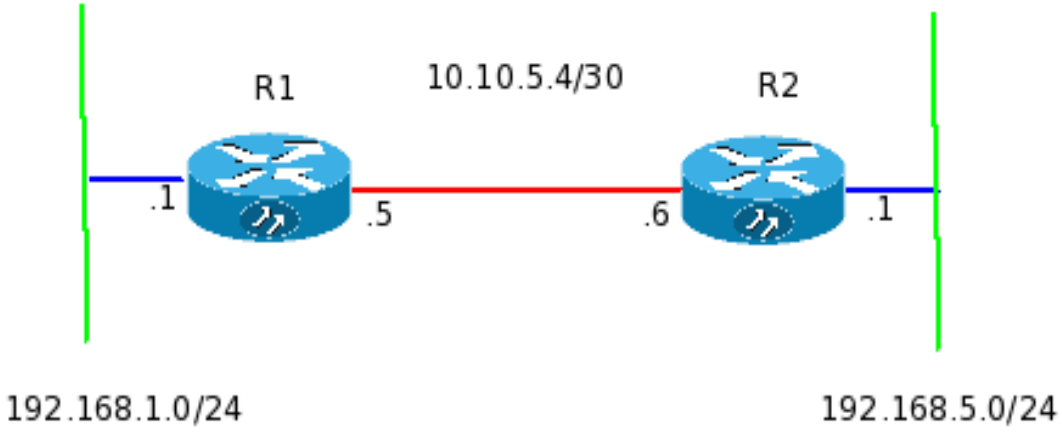
...

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

Статични маршрути във FRR: telnet localhost staticd

```
staticd@border-lozenets# sh run
Current configuration:
!
frr version 7.2.1
frr defaults traditional
!
hostname staticd@border-lozenets
ip route 194.141.252.11/32
194.141.252.21
ipv6 route 2001:4b58:acad:252::11/128
2001:4b58:acad:252::25
```

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?