

Тема № 4. СТАТИЧНА МАРШРУТИЗАЦИЯ

1. Какво представлява маршрутизирането в Internet, в кой слой на OSI модела се извършва и защо е необходимо?
2. Видове маршрутизиране – статично и динамично, протоколи за динамично маршрутизиране, автономни системи, маршрутизираща таблица. Как се определя маршрута с мрежовите маски;
3. Статично маршрутизиране – на основата на работна станция с Centos: ifconfig; ip add & route add; traceroute, netstat, ss, mtr

МАРШРУТИЗИРАНЕ – процесът на придвижване на един пакет с информация от един физически сегмент на мрежата до друг физически сегмент се нарича маршрутизиране (routing). Маршрутизирането управлява процеса на изпращане на логически адресираните (от IP протокола) пакети с информация от техния източник до получателя им с помощта на междинни устройства, наричани маршрутизатори (routers).

Маршрутизирането се извършва в третия слой на модела OSI, от протокола IP, който се занимава с логическото адресиране в мрежата. Може да се каже, че маршрутизатора е интелигентно устройство за разлика от комутатора. Маршрутите може да бъдат задавани административно (статична маршрутизация) или да бъдат изчислени с помощта на алгоритми (динамична маршрутизация), работещи с информация за топологията и състоянието на мрежата, получена от протоколите за маршрутизация.

Маршрутизацията в компютърните мрежи се изпълнява от специални устройства, наричани маршрутизатори. При мрежи със сравнително проста топология, маршрутизацията може да бъде изпълнявана и от компютри с общо предназначение, под управлението на стандартни операционни системи (UNIX, GNU/Linux, даже и Windows),

Маршрутизиращата таблица (routing table or RIB – Routing Information base) представлява структура от данни в табличен вид, в която се съхранява информацията, необходима за маршрутизирането на пакетите в мрежата (Internet). Всеки пакет съдържа информация за неговия източник (source address) и за неговата цел (destination address), а самата маршрутна таблица съдържа следната информация:

1. **Назначение** – IP адресът на следващото местоназначение (следващия hop). Това е (next hop) IP адресът, до който ще се изпрати пакетът;
2. **Метрика** – определя разстоянието до всеки маршрут, така че да може да се избере най-ефективния такъв;
3. **Маршрути** – съдържа както директно свързани подмрежи, така и индиректно свързани подмрежи (такива, които не са директно свързани към възела, но до които може да се достигне през няколко hop-a);
4. **Интерфейс** – изходният мрежови интерфейс, който трябва да се използва за препращане на пакета към крайното му местоназначение.

Три основни етапа на маршрутизирането:

1. **Намиране на MAC адреса** – използва се когато възелът изпраща дейтаграма. Намирането на този адрес е необходимо за капсулирането на IP дейтаграмата. Ако някой възел иска да изпрати дейтаграма, той трябва да капсулира MAC адреса на получаващото мрежово устройство във фрейма. Този адрес може да бъде получен с помощта на таблица, поддържаща съответствието на MAC

адресите и съответстващите им IP адреси в мрежата. Обикновено се използва протокола arp (Address Resolution Protocol) за изграждане на такава таблица. Тя може да се види със следната команда:

```
# arp -a
```

```
? (169.254.248.177) at 2c:44:fd:29:5d:a0 on en0 [ethernet]
```

```
? (172.18.0.1) at 0:2:44:8e:f8:e0 on en0 ifscope [ethernet]
```

2. **Определяне на междумрежовите шлюзове (gateway)** – необходим е, защото Internet е съставен от голям брой отделни мрежи, които се свързват помежду си с помощта на шлюзове, които имат физическа и логическа връзка с повече от една мрежа.
3. **Определяне на цифровия адрес** на получателя от неговия символен адрес. IP адресите в техния цифров вид, даже и в десетично точковата нотация са трудни за запомняне, затова се използват символни адреси (url). Услугата DNS извършва това превеждане – на цифровите адреси в символни и обратно.

Всеки маршрутизатор или възел от мрежата поддържа маршрутна таблица за възможните маршрути според адреса на получателя на пакета. В тази таблица може да бъде включена и информация за съответния резултат за всеки от маршрутите, така че ако има няколко маршрута до една и съща мрежа, да може да бъде избран най-добрият маршрут (най-добър не означава винаги най-къс или най-бърз). Този резултат се определя от метриките, определени от маршрутизатора или от маршрутизиращия протокол.

В маршрутизиращата таблица се поддържа списък на най-добрите маршрути до различни мрежи. За определяне на най-добрия маршрут се използват мерни единици, наречени **метрики**. Те представляват оценка или стойност на някакъв даден параметър на мрежовата връзка. Най-често използвани в маршрутизиращите протоколи мерни единици са:

1. **брой преходи (hop count)** – най-разпространена мерна единица. Измерва броя на маршрутизаторите, през които преминават пакетите от мрежата източник до мрежата получател;
2. **закъснение (delay)** – измерва времето, необходимо да придвижване на пакет от мрежата източник до мрежата получател. Върху закъснението влияние оказват следните фактори: пропускателната способност на мрежата, броя заявки, обслужвани от всеки маршрутизатор, мрежовите задръствания, разстоянието между двете мрежи;
3. **пропускателната способност (bandwidth) на мрежата** – измерва наличния капацитет на мрежовата връзка. Връзка 100 mbps е за предпочитане пред връзка 64 kbps;
4. **надеждност (reliability)** – оценява надеждността на мрежовите връзки. Някои връзки излизат от строя по-често отколкото други --> предпочитат се по-надеждните връзки. Друг параметър на надеждността е времето за възстановяване на пропаднала връзка;
5. **цена на връзката (communication cost)** – понякога доставянето на пакета за най-малко време може да не е основна цел. Цел може да бъде минимизирането на цената на мрежовия транспорт.

Тези мерни единици се ползват от различните маршрутизиращи протоколи. Някои протоколи използват комбинация от мерни единици, като задават различни нива на значимост за всяка използвана мерна единица за определяне на най-оптималния маршрут.

АДМИНИСТРАТИВНА ДИСТАНЦИЯ – използва се от маршрутизаторите, когато са налични повече от един маршрут до дадена мрежа, научени от различни маршрутизиращи протоколи, използващи различни метрики. Следователно, административната дистанция дефинира надеждността на самия протокол. Всеки маршрутизиращ протокол притежава административна дистанция. Колкото е по-малка нейната стойност, толкова по-предпочитан е даден маршрут. Статичните маршрути по подразбиране имат административна дистанция 1. Обикновено административната дистанция е първият критерий, който използват маршрутизаторите в опита си да намерят най-добрият от два или повече маршрута до дадена мрежа.

АВТОНОМНИ СИСТЕМИ – група от IP мрежи и маршрутизатори под административния контрол на една или няколко организации и придържащи се към единни и ясно дефинирани правила за маршрутизация в Internet – [RFC 1930 Guidelines for creation, selection and registration of an autonomous system \(AS\)](#).

Обикновено, но не винаги, организациите, които контролират автономните системи са доставчиците на Internet (ISP).

Всяка автономна система притежава **уникален номер**, който се дава от организацията [IANA](#).

Три категории AS:

1. **multihomed AS** – автономни системи, имащи връзка към повече от една автономна система (респективно доставчик на Internet свързаност). По този начин при отпадане на един от доставчиците, автономната система остава свързана към internet;
2. **stub AS** – автономната система е свързана само към една друга автономна система, респективно доставчик на internet;
3. **transit AS** – осигуряват връзка между автономните системи, свързани към тях. Обикновено доставчиците на Internet са автономни системи от този вид.

За да получите номер на автономна система (за Европа – от организацията [RIPE](#)), тя трябва да бъде multihomed автономна система. Например, автономната система на софийския университет има два излаза към други автономни системи и на теория може да бъде класифицирана и като транзитна, но тя няма право да транзитира трафик, а и прилаганата в нея политика е политика на stub автономна система. В този смисъл, разликата между multihomed и transit автономните системи е по-скоро политическа, отколкото техническа.

В рамките на една автономна система се използват така наречените вътрешни маршрутизиращи протоколи ([Interior Gateway Protocol – IGP](#)). Те се разделят на два вида – дистанционно-векторни ([distance vector](#)) и протоколи със следене на връзката ([link-state](#)).

За маршрутизиране между автономните системи се използват външни (екстериорни) протоколи. Такъв протокол е [BGP – Border Gateway Protocol](#).

Съществуват два основни начина за конфигуриране на един маршрутизатор и въвеждане на необходимата информация в маршрутизиращата таблица – статичен и дина-

мичен. С динамичната маршрутизация ще се занимаем в следващото занятие, а днес ще видим как се конфигурира статично маршрутизатор на основата на работна станция под управлението на linux.

СТАТИЧНО МАРШРУТИЗИРАНЕ

1. ръчно конфигуриране на всички пътища в мрежата. Подходящо е за малки мрежи със сравнително постоянна топология;
2. при промяна в мрежата трябва да се направи ръчно преконфигуриране. Ако не се направи, ще имаме некоректно маршрутизиране. При отпадане на маршрутизатор в мрежата, маршрутизиращата таблица трябва да се преконфигурира така, че отпадналият сегмент да бъде заобиколен. В големи мрежи, статично маршрутизиране се прилага за повишаване на надеждността – ако отпадне динамично научения маршрут, тогава зададения статично се използва като резервен.

Статичните маршрути могат да бъдат постоянни във времето или да се променят на определено време (по разписание).

ДИНАМИЧНО МАРШРУТИЗИРАНЕ

Използва маршрутизиращи протоколи за автоматично построяване на маршрутизиращата таблица. При възникване на промяна в топологията на свързване поради отпадане на един или няколко маршрута, маршрутизаторите обновяват своята маршрутна таблица и намират алтернативен път за доставяне на пакетите до тяхното местоназначение. Едни от най-често използваните алгоритми за динамично маршрутизиране са [алгоритъма на Белман-Форд](#) и [алгоритъма на Дийкстра](#).

Маршрутизиращата таблица се попълва от следните три източника:

1. програмното осигуряване на TCP/IP стека. При инициализацията на маршрутизатора, то автоматично попълва няколко записа в маршрутизиращата таблица, като по този начин се създава така наречената [минимална маршрутизираща таблица](#).

В нея се намират следните записи:

- записите на непосредствено свързаните мрежи и маршрутизатори;
- записите за специалните адреси loopback, multicast и broadcast;
- маршрутите за възел на:
 - локалния интерфейс localhost;
 - на локалната подмрежа;
 - broadcast адреса на локалната мрежа;
 - вътрешния маршрут;
 - multicast адрес;
 - глобалния broadcast адрес;
 - маршрута по подразбиране (default).

В някои маршрутизиращи таблици, тези записи на особени адреси може да липсват.

Съдържанието на маршрутизиращата таблица може да бъде показано с някоя от командите:

```
# ip route show
# route
```

Статичната маршрутизация се извършва с командите:

```
# ip route add
# route add
```

Записите имат дървовидна структура, тоест едни записи могат да бъдат включени в други записи. Не е възможно да има частично пресичане. Границите на адресното пространство, в което действат тези записи се изравняват с неговите размери. Маршрутната таблица се ползва само в случаите, когато трябва да се определи как да бъдат доставени пакетите.

За целите на маршрутизацията се използва само част от информацията от заглавната част (IP header) на IP пакета – destination address на пакета.

Специални адреси:

0.0.0.0 – адреса на възела, генерирал този пакет; обръщение към себе си;

255.255.255.255 – пакет с такъв адрес се изпраща до всички възли в мрежата;

Забранено е да се изпраща пакет там, от където е пристигнал.

СТАТИЧНА МАРШРУТИЗАЦИЯ ПОД LINUX

За да може една работна станция под linux да се използва за маршрутизатор, тя трябва да има поне два физически мрежови интерфейса (eth0 и eth1). Двата интерфейса трябва да бъдат 1. правилно конфигурирани, 2. съответните маршрути да са добавени в маршрутизиращата таблица и 3. да бъде разрешено маршрутизирането.

конфигуриране на мрежовите интерфейси

```
# ifconfig eth0 10.10.23.12 netmask 255.255.255.0 broadcast 10.10.23.255
# ifconfig eth1 10.10.24.12 netmask 255.255.255.0 broadcast 10.10.24.255
# ip addr add 10.10.23.12/24 dev eth0
# ip addr add 10.10.24.12/24 dev eth0
```

добавяне на двете мрежи в маршрутизиращата таблица

```
# route add -net 192.168.11.0
# route add -net 172.18.0.0
# ip route add 192.168.11.0/24 dev eth0
# ip route add 172.18.0.0/24 dev eth 1
```

добавяне на маршрут по подразбиране

```
# route add default gw 192.168.11.1
# ip route add default via 92.168.1.1
```

проверка дали е разрешено маршрутизирането между двата мрежови интерфейса

```
# cat /proc/sys/net/ipv4/ip_forward
```

Ако горната команда върне резултат **1**, значи маршрутизирането е разрешено. А ако върне **0**, значи то не е разрешено и трябва да се изпълни следната команда:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

или

```
vi /usr/lib/sysctl.d/00-system.conf (??)
```

Проверка на маршрутизирането

команда **ifconfig**:

USAGE:

```
ifconfig [-a] [-i] [-v] [-s] <interface> [[<AF>] <address>]
[add <address>[/<prefixlen>]]
[del <address>[/<prefixlen>]]
[[-]broadcast <address>] [[-]pointopoint <address>]]
[netmask <address>] [dstaddr <address>] [tunnel <address>]
[outfill <NN>] [keepalive <NN>]
[hw <HW> <address>] [metric <NN>] [mtu <NN>]
[[-]trailers] [[-]arp] [[-]allmulti]
[mem_start <NN>] [io_addr <NN>] [irq <NN>] [media <type>]
[txqueuelen <NN>]
[[-]dynamic]
[up|down] ...
```

команда **traceroute** – използва се за проследяване на маршрута, през който се преминава за достигане до даден IP адрес.

USAGE:

```
traceroute [-dFINrvx] [-g gateway] [-i iface] [-f first_ttl]
[-m max_ttl] [-p port] [-q nqueries] [-s src_addr] [-t tos]
[-w waittime] [-z pausesecs] host [packetlen]
```

netstat (route) – netstat е полезен инструмент за проверка на конфигурацията и работата на мрежата. Всъщност това са няколко инструмента събрани заедно. Когато стартираме netstat с флага **-r**, ще бъде отпечатана таблицата с маршрути в ядрото по начин, аналогичен на командата route

```
[root@sakurajima sysctl.d]# netstat -r
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS Window	irrtt	Iface
default	192.168.11.1	0.0.0.0	UG	0 0	0	wlp3s0
192.168.11.0	0.0.0.0	255.255.255.0	U	0 0	0	wlp3s0

```
[root@sakurajima sysctl.d]# netstat -r -n
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS Window	irrtt	Iface
0.0.0.0	192.168.11.1	0.0.0.0	UG	0 0	0	wlp3s0
192.168.11.0	0.0.0.0	255.255.255.0	U	0 0	0	wlp3s0

```
[root@sakurajima sysctl.d]# route
```

Kernel IP routing table

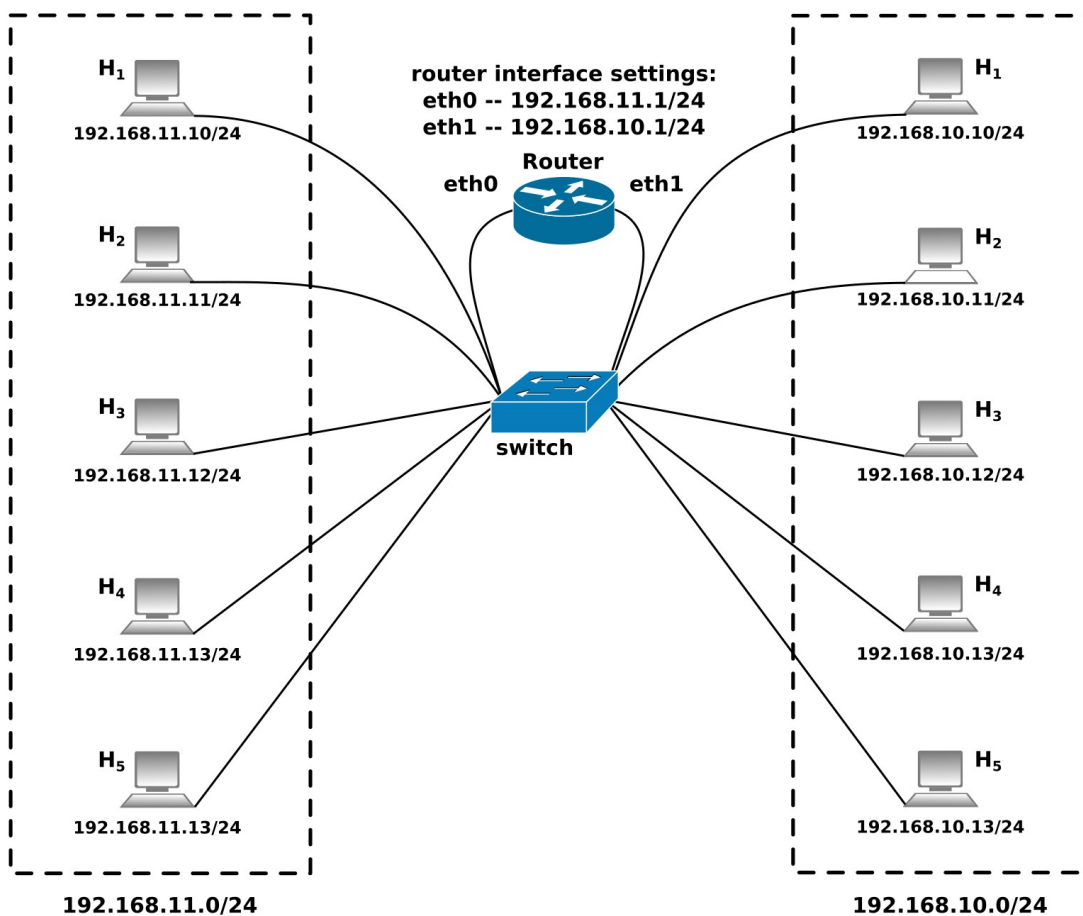
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	192.168.11.1	0.0.0.0	UG	1024	0	0	wlp3s0
192.168.11.0	0.0.0.0	255.255.255.0	U	0	0	0	wlp3s0

Опцията “-n” указва на netstat да отпечата адресите като IP номера в десетично-точков формат, вместо да използва символните имена на хостове и мрежи. Втората колона от резултата от netstat показва шлюза, към който сочи маршрута. Ако не се използва шлюз, се извежда звезда. Третата колона показва мрежовата маска за този маршрут. Когато е даден IP адрес, за който се търси подходящ маршрут, ядрото преминава през всеки от записите в таблицата с маршрути, като извършва побитово AND на адреса и маската, преди да го сравни с целта на маршрута. Четвъртата колона отпечата следните флагове, които описват маршрута:

- G – Маршрутът използва шлюз;
- U – Интерфейсът, който ще се използва, е активен (up);
- H – През този маршрут може да се достигне само един хост;
- D – Този маршрут е създаден динамично;
- M – Записът е модифициран от ICMP съобщение за пренасочване;
- ! – Маршрутът е отхвърлящ и дейтаграмите ще бъдат игнорирани.

Задачи:

- Запознаване с командите: tracert (tracert), ifconfig (ipconfig), netstat (route) и тяхното действие.
- Да се проследи маршрутът до отдалечена станция чрез tracert (tracert). Пример: www.google.com, www.microsoft.com, и т.н.
- Да се наблюдават маршрутните таблици на маршрутизираща и локална станция с командата netstat и ключове: “-nr”, “-sp IP”.
- Да се конфигурира Linux станция като **NAT маршрутизатор**, като се използва схемата, дадена на фигура 1. Интерфейсът на маршрутизатора, включен към мрежата на университета, да се конфигурира статично към шлюз (gateway) с адрес 10.10.23.254.



РЕШЕНИЕ

1. конфигуриране на интерфейсите на възлите (H) в мрежата:

```
ip addr add 192.168.11.xxx/24 dev eth0
ip addr add 192.168.10.xxx/24 dev eth0
```

2. конфигуриране на маршрутите:

```
ip route add 192.168.11.0/24 dev eth0
ip route add 192.168.10.0/24 dev eth0
```

3. конфигуриране на маршрут по подразбиране:

```
ip route add default via 192.168.11.1
ip route add default via 192.168.10.1
```

4. конфигуриране на възела, избран за маршрутизатор (R):

```
ip addr add 192.168.11.1/24 dev eth0
ip addr add 192.168.10.1/24 dev eth1
ip addr add 10.10.23.xxx/24 dev eth2
ip route add 192.168.11.0/24 dev eth0
ip route add 192.168.10.0/24 dev eth1
ip route add 10.10.23.0/24 dev eth2 src 10.10.23.xxx
ip route add default via 10.10.23.254 dev eth2
```


Проверяваме дали параметъра в ядрото за препращане на пакети (ip_forward) между двата интерфейса е включен:

```
cat /proc/sys/net/ipv4/ip_forward
```

Ако отговорът на тази команда е 1, значи ip_forward е активен и ядрото ще препраща пакети от един мрежови интерфейс към друг. Ако отговорът е 0, значи параметъра не е активен. За да го направим активен, ще използваме следната команда:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

след което правим повторна проверка на съдържанието на този параметър с командата cat, посочена малко по-нагоре.

Преди да се активира ip_forward (или ако е активиран, след неговото деактивиране) се проследява с командата ping достъпността до възли от собствената мрежа и до възли от другата мрежа:

```
ping -c10 192.168.11.xxx
```

```
ping -c10 192.168.10.xxx
```

При непозволен ip_forward на маршрутизатора, би трябвало да има отговор на ping само от възли от собствената мрежа. При позволен ip_forward възлите от двете логически мрежи вече би трябвало да се виждат един друг.