

# Sieci komputerowe

Tadeusz Kobus, Maciej Kokociński  
Instytut Informatyki, Politechnika Poznańska

# Routing statyczny w Linuksie

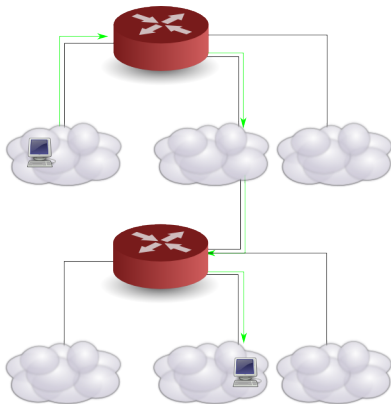
# Routing

Internet składa się z mniejszych, połączonych ze sobą sieci komputerowych.

Sieci łączone są przy pomocy **routerów**.

Urządzenie (komputer/router) korzysta z **tablicy routingu** zawierającej:

- adres sieci docelowej (destination),
- maskę podsieci (mask),
- bramę (gateway).



# Tablica routingu

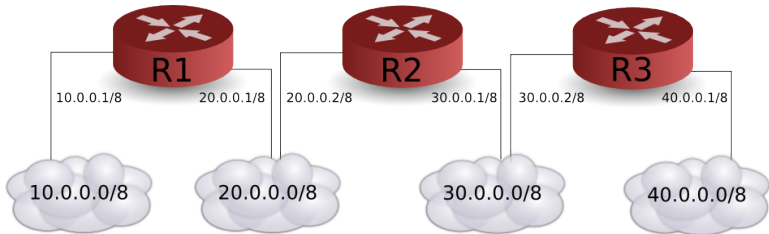
Sieć	Maska	Brama
192.168.1.0	255.255.255.0	10.0.0.1
192.168.2.0	255.255.255.0	10.0.0.2
192.168.3.0	255.255.255.0	10.0.0.3
192.168.4.0	255.255.255.0	10.0.0.4
10.0.0.0	255.0.0.0	-

- Pakiet o adresie docelowym 192.168.3.42 jest kierowany do sieci 192.168.3.0/24 przez bramę 10.0.0.3.
- Pakiet o adresie docelowym z sieci, w której jest urządzenie (w tabelicy brak określonej bramy), np. 10.143.11.85, jest kierowany do tej sieci.
- Każdy kolejny router na ścieżce pakietu niezależnie podejmuje decyzję o tym, dokąd dalej przesłać pakiet.
- Przy wyznaczaniu trasy adres źródłowy pakietu nie jest brany pod uwagę.

# Typy routingu

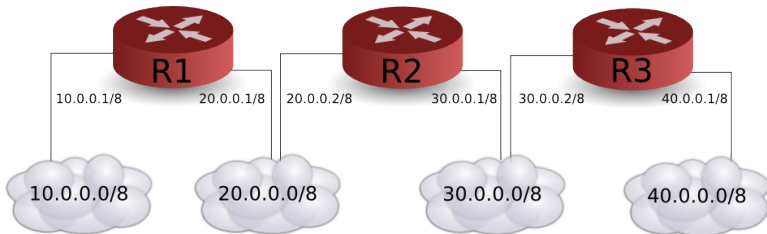
- Routing statyczny (static routing) – zawartość tablicy routingu zdefiniowana przez administratora (na stałe).
- Routing dynamiczny (dynamic routing) – tablica routingu wyznaczona jest przez protokół routingu (np. RIP, OSPF, IS-IS, IGRP) w czasie działania urządzenia.
- Routing sprzętowy – realizowany przez hardware.

# Routing – przykład (1)



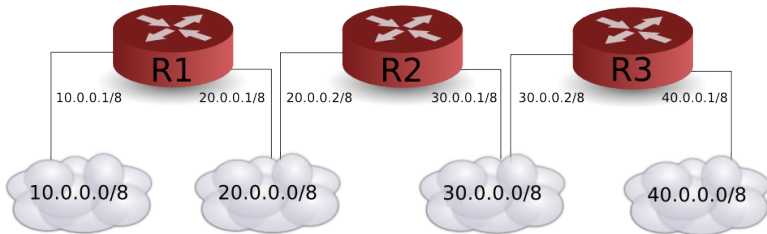
	Sieć	Maska	Brama
R1	10.0.0.0	255.0.0.0	-
	20.0.0.0	255.0.0.0	-
	30.0.0.0	255.0.0.0	20.0.0.2
	40.0.0.0	255.0.0.0	20.0.0.2

## Routing – przykład (2)



	Sieć	Maska	Brama
R2	10.0.0.0	255.0.0.0	20.0.0.1
	20.0.0.0	255.0.0.0	-
	30.0.0.0	255.0.0.0	-
	40.0.0.0	255.0.0.0	30.0.0.2

## Routing – przykład (3)



	Sieć	Maska	Brama
R3	10.0.0.0	255.0.0.0	30.0.0.1
	20.0.0.0	255.0.0.0	30.0.0.1
	30.0.0.0	255.0.0.0	-
	40.0.0.0	255.0.0.0	-



# Konfiguracja routingu statycznego (1)

```
# ip route show
default via 150.254.44.1 dev wlp3s0 proto static metric 600
150.254.44.0/23 dev wlp3s0 proto kernel scope link src 150.254.45.101
                                                    metric 600

# ip addr add 192.168.1.101/24 dev wlp3s0
# ip route
default via 150.254.44.1 dev wlp3s0 proto static metric 600
150.254.44.0/23 dev wlp3s0 proto kernel scope link src 150.254.45.101
                                                    metric 600
192.168.1.0/24 dev wlp3s0 proto kernel scope link src 192.168.1.101
# ip route add 172.16.0.0/16 via 192.168.1.111
# ip route
default via 150.254.44.1 dev wlp3s0 proto static metric 600
150.254.44.0/23 dev wlp3s0 proto kernel scope link src 150.254.45.101
                                                    metric 600
172.16.0.0/16 via 192.168.1.111 dev wlp3s0
192.168.1.0/24 dev wlp3s0 proto kernel scope link src 192.168.1.101
                                                    metric 600

# ip route add default via 192.168.1.1
# ip route del 172.16.0.0/16
```

## Konfiguracja routingu statycznego (2)

```
# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref  Use  Iface
0.0.0.0          150.254.44.1    0.0.0.0         UG    600   0    0    wlp3s0
150.254.44.0    0.0.0.0         255.255.254.0   U     600   0    0    wlp3s0
# ifconfig wlp3s0:1 192.168.1.101 netmask 255.255.255.0
# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref  Use  Iface
0.0.0.0          150.254.44.1    0.0.0.0         UG    600   0    0    wlp3s0
150.254.44.0    0.0.0.0         255.255.254.0   U     600   0    0    wlp3s0
192.168.1.0     0.0.0.0         255.255.255.0   U     0     0    0    wlp3s0
# route add -net 172.16.0.0 netmask 255.255.0.0 gw 192.168.1.111
# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref  Use  Iface
0.0.0.0          150.254.44.1    0.0.0.0         UG    600   0    0    wlp3s0
150.254.44.0    0.0.0.0         255.255.254.0   U     600   0    0    wlp3s0
172.16.0.0      192.168.1.111  255.255.0.0     UG    0     0    0    wlp3s0
192.168.1.0     0.0.0.0         255.255.255.0   U     0     0    0    wlp3s0
# route add default gw 192.168.1.1
# route del -net 172.16.0.0 netmask 255.255.0.0
```

# Procedura wyboru trasy (1)

1. Trasa do sieci, w której znajduje się adres docelowy o najdłuższej (najbardziej restrykcyjnej) masce.

```
172.21.0.0/16 via 172.16.0.16 dev eth2
172.21.0.0/24 via 172.16.0.24 dev eth2
172.21.0.0/28 via 172.16.0.28 dev eth2
```

2. Jeśli wybór nie jest jednoznaczny – trasa o najmniejszym koszcie.

```
150.254.44.0/23 dev wlan0 proto kernel scope link
                        src 150.254.45.39 metric 2003
150.254.44.0/23 dev wlan1 proto kernel scope link
                        src 150.254.44.149 metric 1002
```

3. Jeśli wybór nie jest jednoznaczny – pierwsza w tablicy.

```
default via 150.254.44.1 dev wlan0
default via 150.254.130.42 dev eth0
```

## Procedura wyboru trasy (2)

```
# ip route
```

```
default via 150.254.31.1 dev enp0s26u1u2 proto static metric 100
default via 150.254.44.1 dev wlp3s0 proto static metric 600
150.254.6.8 via 150.254.44.1 dev wlp3s0 proto dhcp metric 600
150.254.31.0/25 dev enp0s26u1u2 proto kernel scope link src 150.254.31.15
                                                    metric 100
150.254.44.0/23 dev wlp3s0 proto kernel scope link src 150.254.45.101
                                                    metric 600
```

```
# route -n
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	150.254.31.1	0.0.0.0	UG	100	0	0	enp0s26u1u2
0.0.0.0	150.254.44.1	0.0.0.0	UG	600	0	0	wlp3s0
150.254.6.8	150.254.44.1	255.255.255.255	UGH	600	0	0	wlp3s0
150.254.31.0	0.0.0.0	255.255.255.128	U	100	0	0	enp0s26u1u2
150.254.44.0	0.0.0.0	255.255.254.0	U	600	0	0	wlp3s0

# Forwarding

Ustawienie za pomocą pliku `/proc/sys/net/ipv4/ip_forward`:

- wartość 0 – forwarding wyłączony,
- wartość 1 – forwarding włączony,
- ustawienie na stałe: `/etc/sysctl.conf`.

```
# cat /proc/sys/net/ipv4/ip_forward
0
# echo 1 > /proc/sys/net/ipv4/ip_forward
# cat /proc/sys/net/ipv4/ip_forward
1
# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

# Zadanie 1

1. Połącz komputer z sąsiednimi komputerami, tak by wszystkie komputery w laboratorium były spięte w łańcuszek. Komputery mają być połączone ze sobą bezpośrednio. Użyj w tym celu portów p4p1 i p4p2.
2. Ustal z koleżankami/kolegami spójną numerację komputerów, tak by utworzyć następujące sieci między sąsiednimi komputerami:
  - dla komputerów 1 i 2: 10.0.1.0/24,
  - dla komputerów 2 i 3: 10.0.2.0/24,
  - ...
  - adresy komputera 1: 10.0.1.1/24,
  - adresy komputera N: 10.0.N-1.N/24, 10.0.N.N/24,
3. Skonfiguruj adresy IP na interfejsach p4p1 i p4p2 i sprawdź czy możliwa jest komunikacja z innymi komputerami (np. przy pomocy polecenia ping).
4. Dodaj routing do innych sieci.
5. Co pokazuje traceroute/mtr?

# Komunikaty redirect

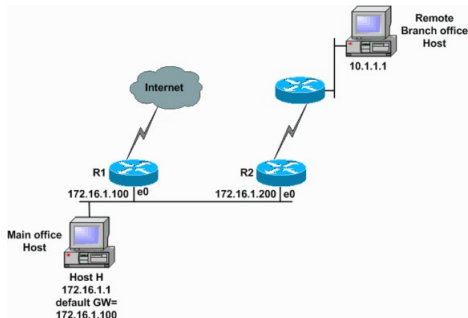
(IP hosta: 172.16.1.1)

```
# ping 10.1.1.1
```

```
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
```

```
From 172.16.1.100: icmp_seq=1 Redirect Host(New nexthop: 172.16.1.200)  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=64 time=82.8 ms
```

```
From 172.16.1.100: icmp_seq=2 Redirect Host(New nexthop: 172.16.1.200)  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=123 ms
```



Router R1 przekazuje pakiet od hosta H przez ten sam interfejs (e0), przez który pakiet przyszedł od H.

Router R1 dodatkowo informuje H, że istnieje prostsze połączenie: H → R2.