

PROTOKOŁY DYNAMICZNEGO ROUTINGU IP – RIP i OSPF PODSTAWY DZIAŁANIA ZAGADNIENIA PROJEKTOWE I WDROŻENIOWE

Łukasz Bromirski

**lukasz@bromirski.net
lbromirski@cisco.com**

Parę uwag na początek

- **Slajdy będą dostępne na mojej stronie prywatnej i na stronie konferencji**

<http://lukasz.bromirski.net>

- **W sieci jest bardzo dużo informacji**

Polecane URLe i książki na końcu tej prezentacji

- **Jutrzejsza sesja dotyczyć będzie BGP w tym samym ujęciu oraz nowemu projektowi, który BGP wykorzystuje**
- **Please, do ask questions**

Agenda

- **Powtórka z rozrywki: podstawy routingu IP**
- **Zebra/Quagga, XORP i OpenOSPFd**
- **Protokół RIP**
- **Protokół OSPF**
- **Protokoły routingu dynamicznego a**
 - NAT**
 - tunelowanie (GRE i IP-w-IP)**
 - filtrowanie ruchu**
- **Q & A**

POWTÓRKA Z ROZRYWKI: ROUTING IP



Routing IP

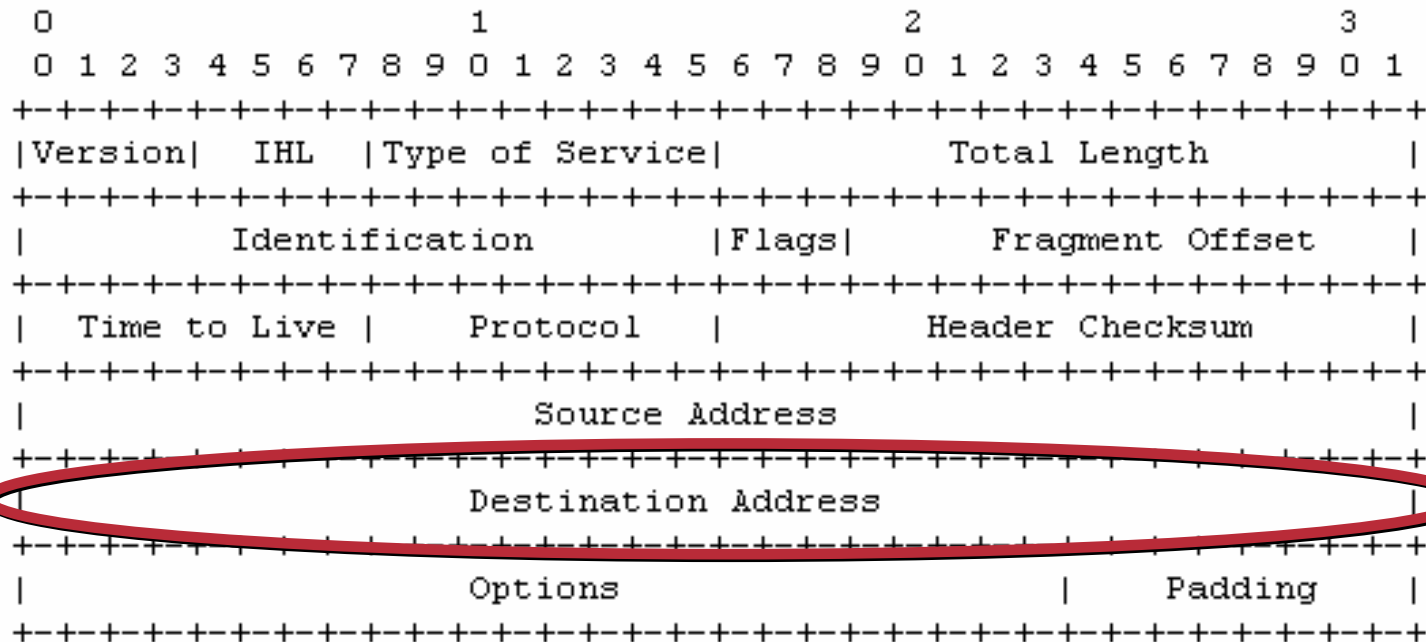
O czym mówimy?

- Routing IP to decyzja wykonana na podstawie adresu **docelowego** pakietu IP
- Kernel podejmuje tą decyzję na podstawie tablicy **FIB** – Forwarding Information Base
- Aplikacje zapewniające routing dynamiczny utrzymują zwykle swoją tablicę – **RIB** – Routing Information Base – z której najlepsze wpisy eksportowane są do FIB
- Narzędzia systemowe wpływają na FIB
- Narzędzia aplikacji wpływają na RIB właściwy dla pakietu

Routing IP

Czym zajmuje się router?

- Router otrzymuje datagramy IP w postaci:



RFC 791, <http://www.ietf.org/rfc/rfc0791.txt?number=791>

Routing IP

24x7x365...

- Router w dużym uproszczeniu cały czas wykonuje następującą pętlę:

odbiera pakiet

jeśli nie TTL=1, adres docelowy=adres mojego interfejsu lub [...]

sprawdź, na jaki interfejs wskazuje w tablicy routingu wpis dla adresu docelowego z pakietu

jeśli wpis zawiera inny adres, rozwiąż go na prawidłowy adres następnej bramy

zmniejsz TTL o 1

wstaw pakiet do bufora wyjściowego interfejsu, który wybrałeś po znalezieniu w tablicy routingu najdokładniejszego wpisu

odbiera pakiet

...

Routing IP

Budowa FIB

- Tablica routingu (FIB) zawiera wpisy pochodzące z wielu źródeł, ale w znormalizowanej postaci
- W FreeBSD pola obecne dla każdego wpisu to między innymi:

destination – sieć lub host docelowy

gateway – przez jakie next-hop IP osiągaln(a/y)

flags – dodatkowe atrybuty trasy

use – ile razy użyto trasy

netif – przez który interfejs pakiet zostanie wysłany

Routing IP

Budowa FIB

- **Zawartość FIB:**

```
$ netstat -nrf inet
```

```
Internet:
```

Destination	Gateway	Flags	Use	Netif
default	62.111.150.245	UGS	98373	fxp2
1	127.0.0.1	UG1B	1036	lo0
2	127.0.0.1	UG1B	602	lo0
3	62.111.128.61	UG1	2470	fxp1
4	62.111.128.61	UG1	297	fxp1
4.17.225/24	62.111.128.61	UG1	0	fxp1
[...]				

Routing IP

Droga trasy do FIB

- **Protokół routingu dynamicznego poznaje trasę**
- **Protokół routingu dynamicznego w wyniku działania swoich algorytmów ustala, że jest to trasa najlepsza i umieszcza ją (je) w RIB**
- **Demon odpowiedzialny za interakcję z kernelem, eksportuje najlepsze trasy z RIB (mogą pochodzić z różnych protokołów) do systemowego FIB**

QUAGGA/ZEBRA

a

XORP

a

OpenOSPFd



Quagga/Zebra

- Quagga posiada budowę modułarną
- Proces **zebra** odpowiada za interakcje wszystkich pozostałych z kernelem (FIB) i zarządzanie RIB
- Osobne procesy odpowiedzialne za protokoły routingu
 - ripd (v1/v2), ripngd (v3 dla IPv6)
 - ospfd (v2), ospf6d (v3 dla IPv6)
 - bgpd (v4+)
 - is-is*
- Dostępne narzędzie vtysh do zarządzania „wszystkim naraz”
- Quagga jest przygotowana do przechowywania wielu takich samych tras w RIB:

```
configure [...] --enable-multipath=X
```

XORP

- XORP również posiada budowę modułarną
- Router manager (**rtrmng**r) nadzoruje pracę grupy procesów
- Dwie osobne ścieżki:
 - unicast: BGP4+, RIP, OSPF*, IS-IS
 - multicast: PIM-SM, IGMPv1/v2 (v3*), MLD
- Wydzielony RIB dla wszystkich protokołów
- Wydzielona FEA, pozwalająca uniezależnić się od systemu/dostępnych interfejsów
- Dostępna powłoka **xorpsh** do zarządzania

OpenOSPFd

- **Nowy projekt zespołu OpenBSD**
Henning Brauer, Claudio Jeker & Esen Norby
- **Projekt w trakcie dopracowywania**
problemy z działaniem SPF
częstkowe wsparcie dla tras External
problemy z redystrybucją/wstrzyknięciem trasy default
- **Podobnie jak OpenBGPd:**
ospfd – demon odpowiedzialny za protokół
ospfctl – narzędzie do kontroli
/etc/ospfd.conf - konfiguracja

PROTOKÓŁ ROUTINGU RIP



Protokół routingu RIP

Routing Information Protocol

- **Routery wymieniają się swoimi tablicami routingu co określone odstępy czasu**
 - RIP standardowo co 30 sekund (z małymi różnicami)
- **Metryką trasy w protokole RIP jest ilość hopów, jaką musi pokonać pakiet, by dotrzeć do sieci/hosta**
 - 1-14
 - 15 = trasa nieosiągalna
- **„Głośny” protokół**
 - co 30 sekund wymieniamy pełne tablice
- **Tylko RIPv2 przesyła maskę dla trasy**

Protokół routingu RIP

Routing Information Protocol

```
ripd# show ip rip status
```

```
Routing Protocol is "rip"
```

```
  Sending updates every 30 seconds with +/-50%, next due in 6 seconds
```

```
  Timeout after 180 seconds, garbage collect after 120 seconds
```

```
  Outgoing update filter list for all interface is not set
```

```
  Incoming update filter list for all interface is not set
```

```
  Default redistribution metric is 1
```

```
  Redistributing:
```

```
  Default version control: send version 2, receive version 2
```

Interface	Send	Recv	Key-chain
fxp0	2	2	
lo0	2	2	

```
  Routing for Networks:
```

```
    172.16.91.0/24
```

```
    192.168.0.0/24
```

```
  Routing Information Sources:
```

Gateway	BadPackets	BadRoutes	Distance	Last Update
---------	------------	-----------	----------	-------------

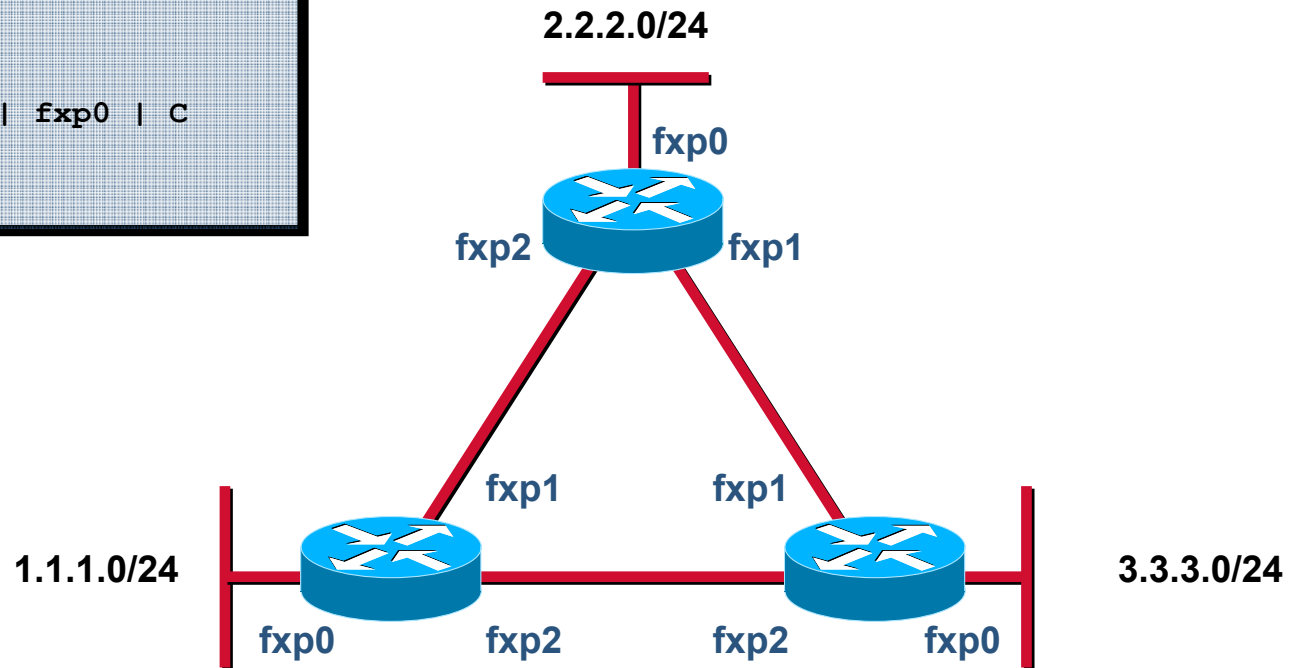
```
  Distance: (default is 120)
```

Protokół routingu RIP

Jak działa?

Router B:

2.2.2.0/24 | fxp0 | C



Router A:

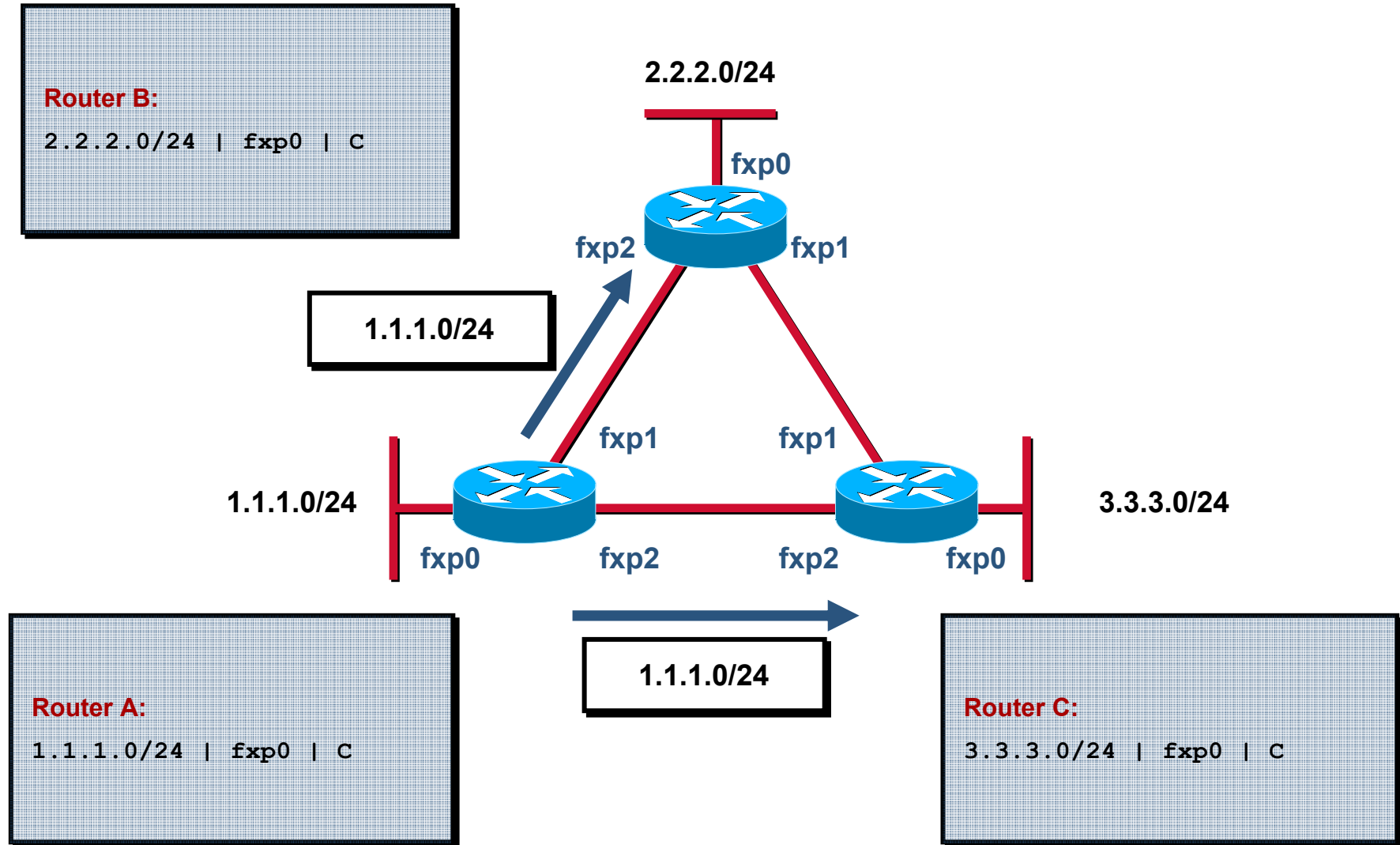
1.1.1.0/24 | fxp0 | C

Router C:

3.3.3.0/24 | fxp0 | C

Protokół routingu RIP

Jak działa?

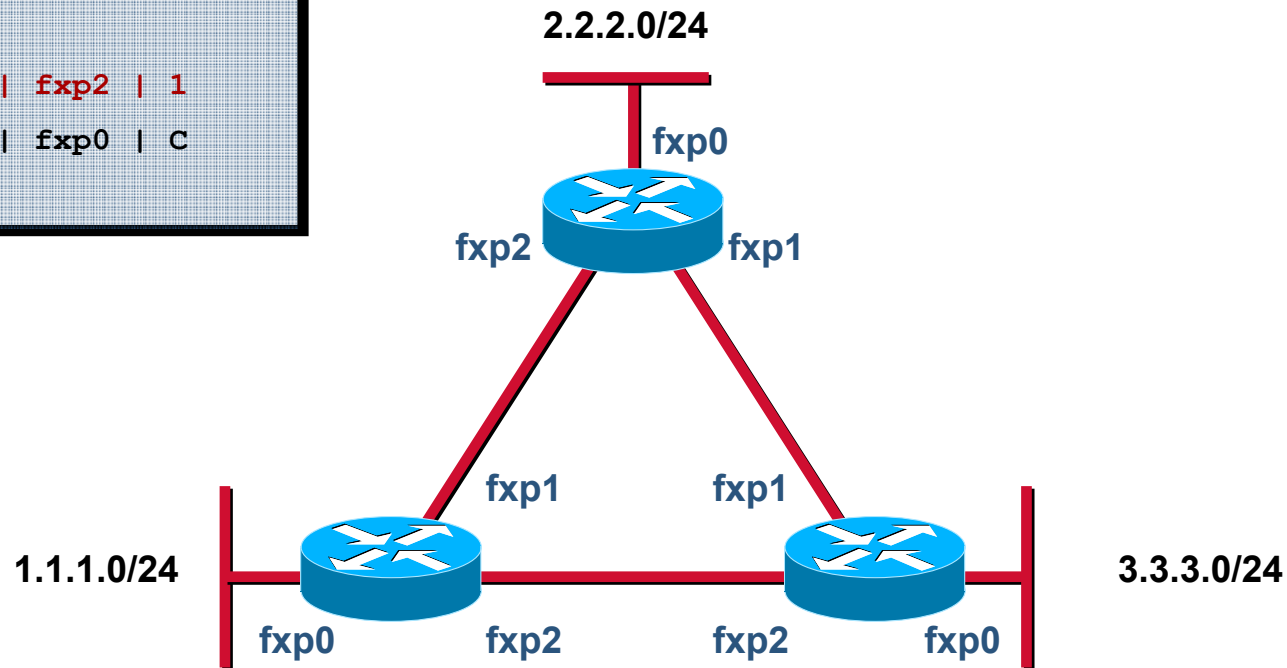


Protokół routingu RIP

Jak działa?

Router B:

```
1.1.1.0/24 | fxp2 | 1  
2.2.2.0/24 | fxp0 | C
```



Router A:

```
1.1.1.0/24 | fxp0 | C
```

Router C:

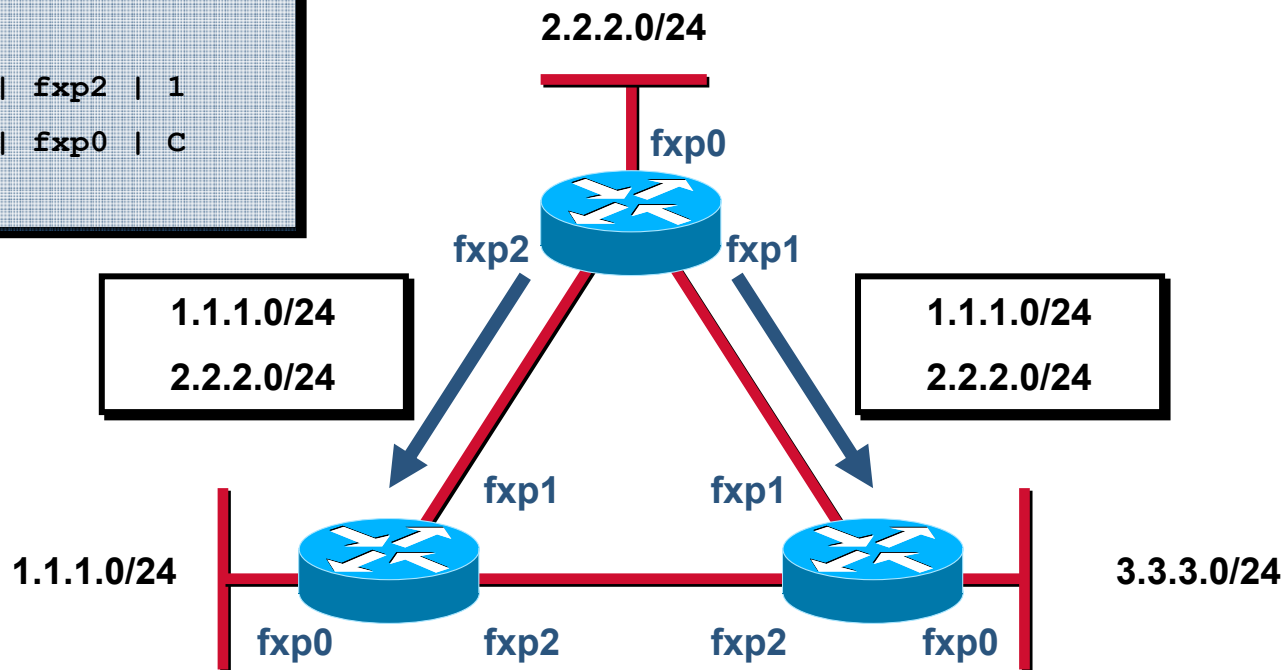
```
1.1.1.0/24 | fxp2 | 1  
3.3.3.0/24 | fxp0 | C
```

Protokół routingu RIP

Jak działa?

Router B:

```
1.1.1.0/24 | fxp2 | 1  
2.2.2.0/24 | fxp0 | C
```



Router A:

```
1.1.1.0/24 | fxp0 | C
```

Router C:

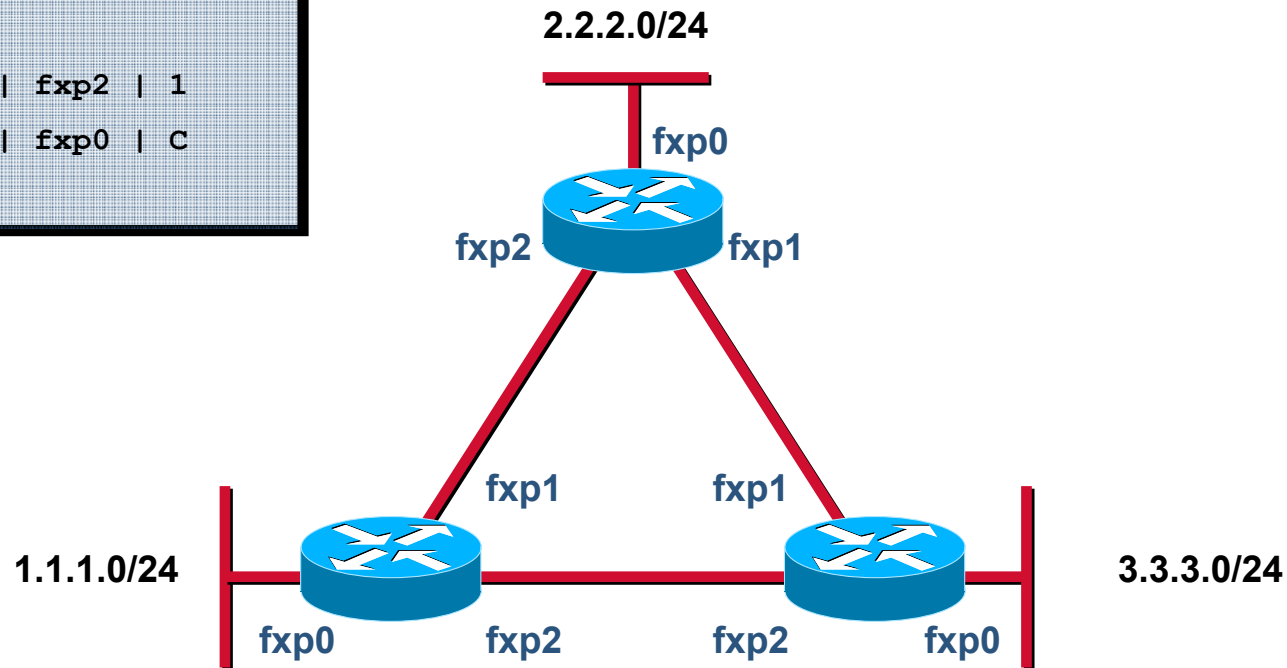
```
1.1.1.0/24 | fxp2 | 1  
3.3.3.0/24 | fxp0 | C
```

Protokół routingu RIP

Jak działa?

Router B:

```
1.1.1.0/24 | fxp2 | 1  
2.2.2.0/24 | fxp0 | C
```



Router A:

```
1.1.1.0/24 | fxp0 | C  
2.2.2.0/24 | fxp1 | 1
```

Router C:

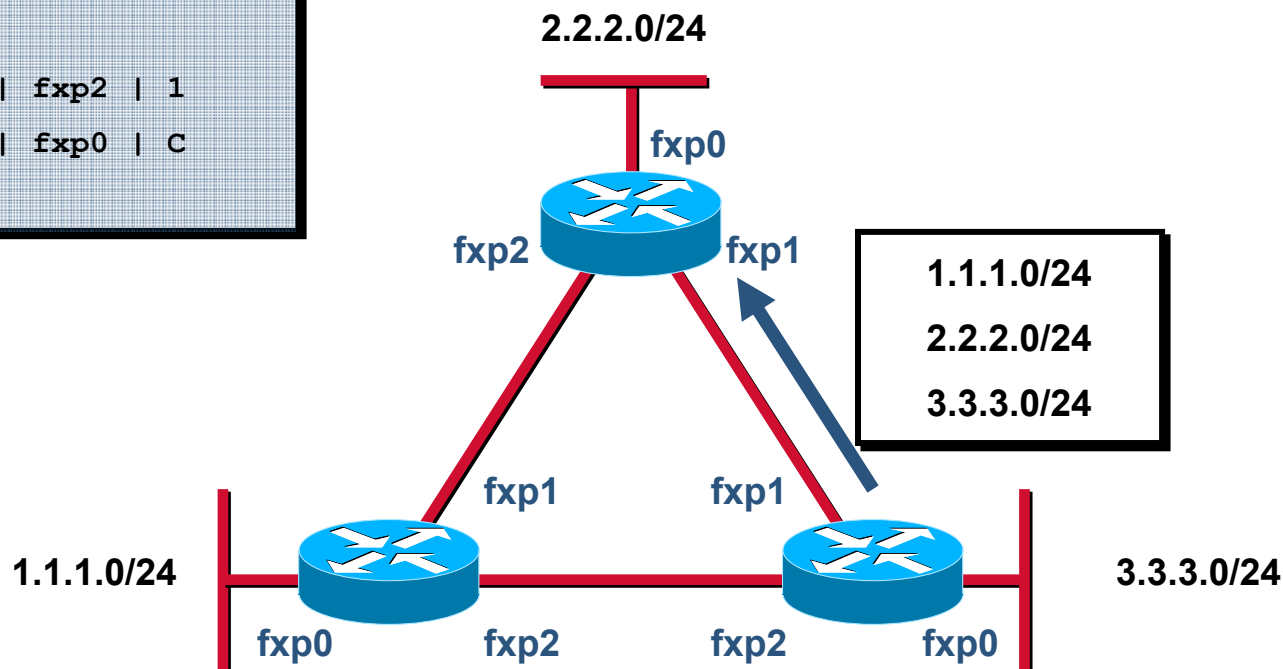
```
1.1.1.0/24 | fxp2 | 1  
2.2.2.0/24 | fxp1 | 1  
3.3.3.0/24 | fxp0 | C
```

Protokół routingu RIP

Jak działa?

Router B:

```
1.1.1.0/24 | fxp2 | 1  
2.2.2.0/24 | fxp0 | C
```



Router A:

```
1.1.1.0/24 | fxp0 | C  
2.2.2.0/24 | fxp1 | 1
```

Router C:

```
1.1.1.0/24 | fxp2 | 1  
2.2.2.0/24 | fxp1 | 1  
3.3.3.0/24 | fxp0 | C
```

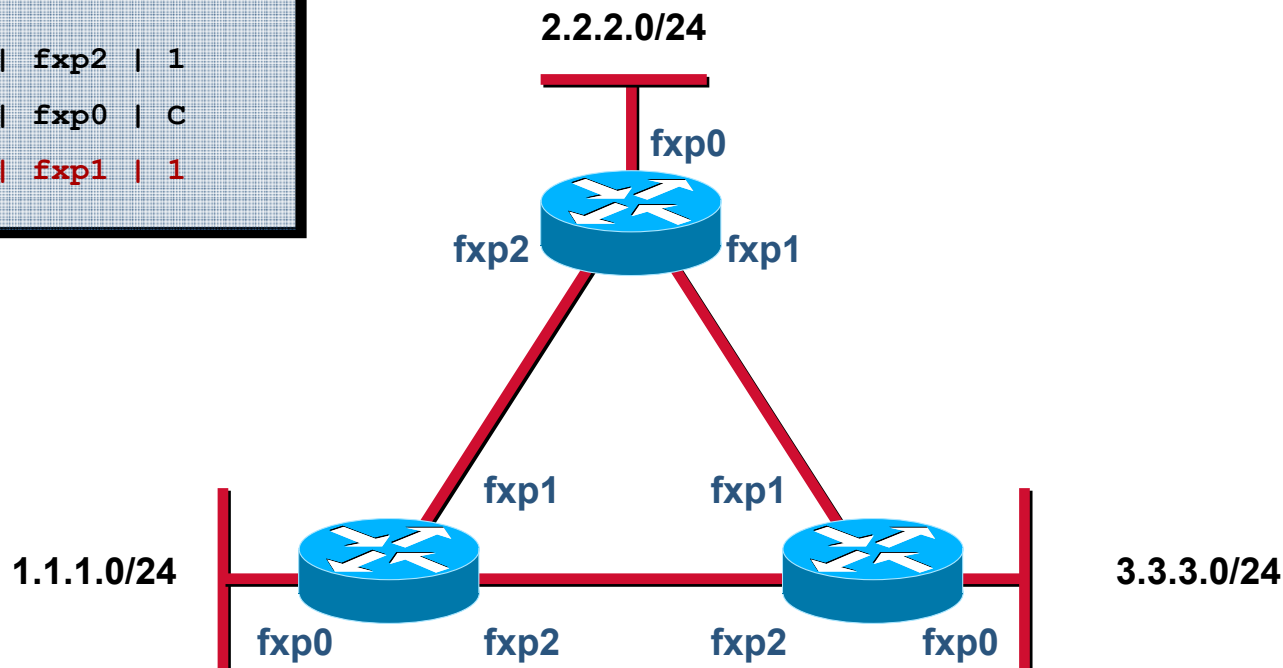
```
1.1.1.0/24  
2.2.2.0/24  
3.3.3.0/24
```

Protokół routingu RIP

Jak działa?

Router B:

```
1.1.1.0/24 | fxp2 | 1  
2.2.2.0/24 | fxp0 | C  
3.3.3.0/24 | fxp1 | 1
```



Router A:

```
1.1.1.0/24 | fxp0 | C  
2.2.2.0/24 | fxp1 | 1  
3.3.3.0/24 | fxp2 | 1
```

Router C:

```
1.1.1.0/24 | fxp2 | 1  
2.2.2.0/24 | fxp1 | 1  
3.3.3.0/24 | fxp0 | C
```


Protokół routingu RIP

Kiedy zastosować?

- Najlepiej **nie stosować** – są lepsze, efektywniejsze i o większych możliwościach również dostępne na licencjach BSD/GPL/etc.
- Czasami wymaga tego obecność prostej „zamkniętej” bramki w sieci
 - zwykle obsługują tylko RIP
 - uwaga na problemy ze zgodnością
- Do wyboru mamy:
 - routed – w base systemie
 - ripd/ripng – w pakiecie Quagga

Protokół routingu RIP

Scenariusz #1

- Czy w tej sytuacji potrzebujemy w ogóle protokołu routingu?

- RTR-GW:

default – świat

172.16.10.0/24 via 172.16.0.6

172.16.11.0/24 via 172.16.0.2

- RTR-A:

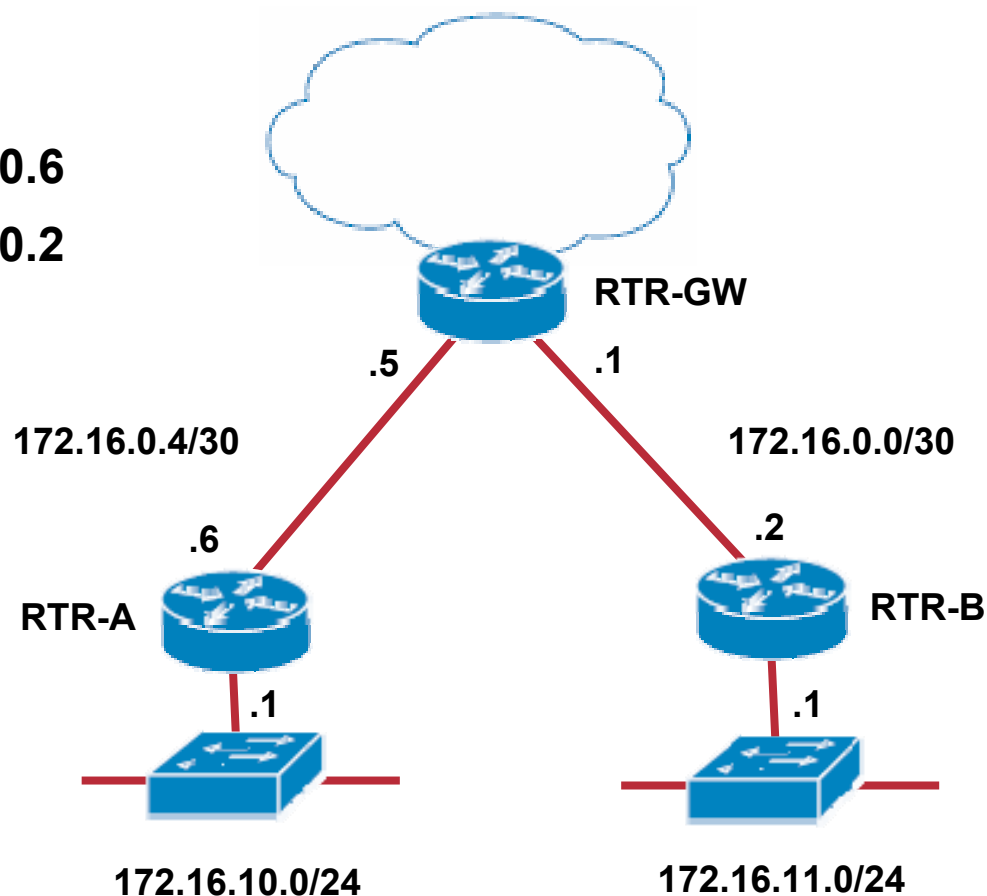
default – 172.16.0.5

172.16.10.0/24 connected

- RTR-B:

default – 172.16.0.1

172.16.11.0/24 connected



Protokół routingu RIP

Scenariusz #2

- Czy w tej sytuacji potrzebujemy w ogóle protokołu routingu?

- RTR-GW:

default – świat

172.16.10.0/24 via 172.16.0.6

172.16.11.0/24 via 172.16.0.2

- RTR-A:

default – 172.16.0.5

172.16.10.0/24 connected

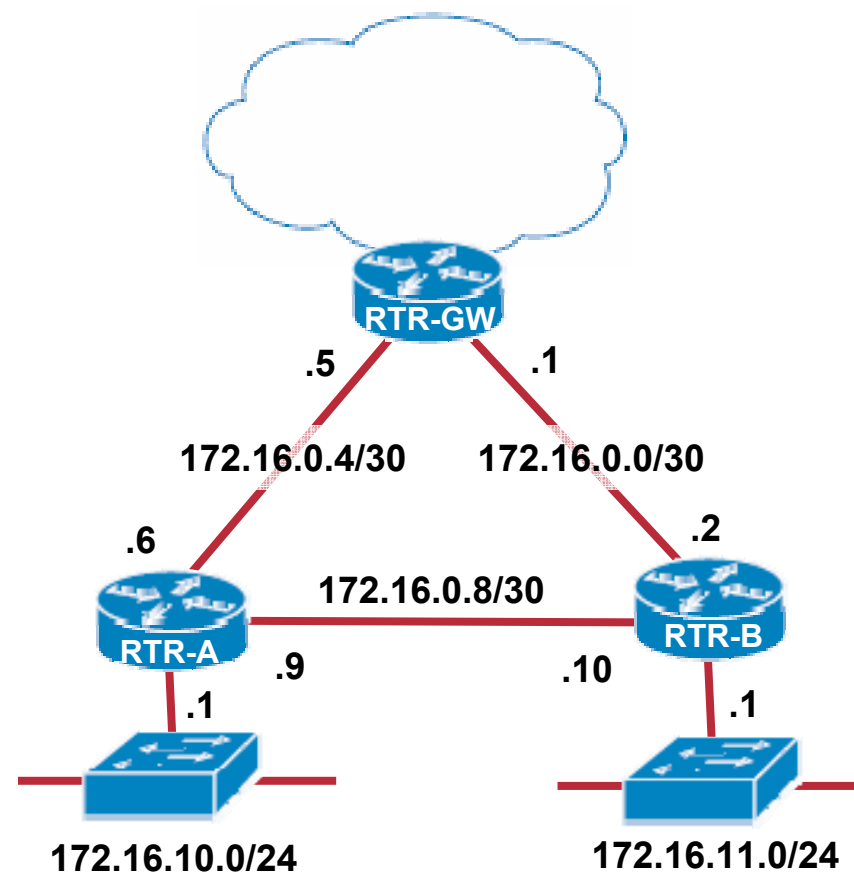
172.16.10.8/30 connected

- RTR-B:

default – 172.16.0.1

172.16.11.0/24 connected

172.16.10.8/30 connected

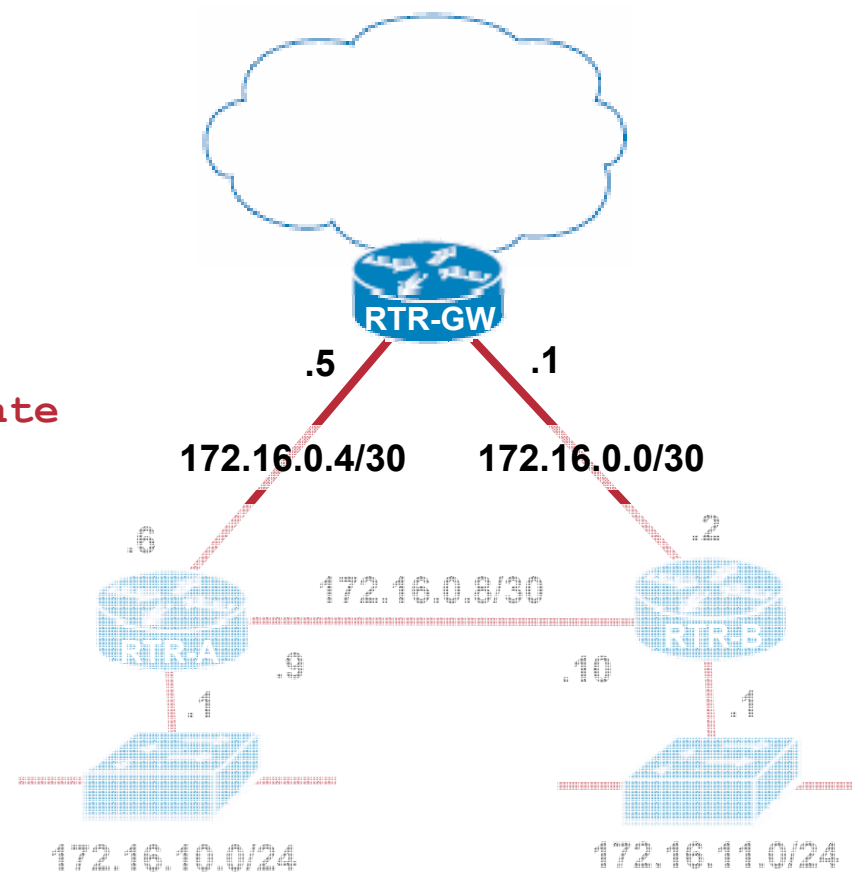


Protokół routingu RIP

Scenariusz #2 - konfiguracja

- Na RTR-GW:
rozgłosić trasę default

```
router rip
  version 2
  default-information originate
  network 172.16.0.0/30
  network 172.16.0.4/30
```



Protokół routingu RIP

Scenariusz #2 - konfiguracja

- Na RTR-A:
rozgłosić widziane sieci

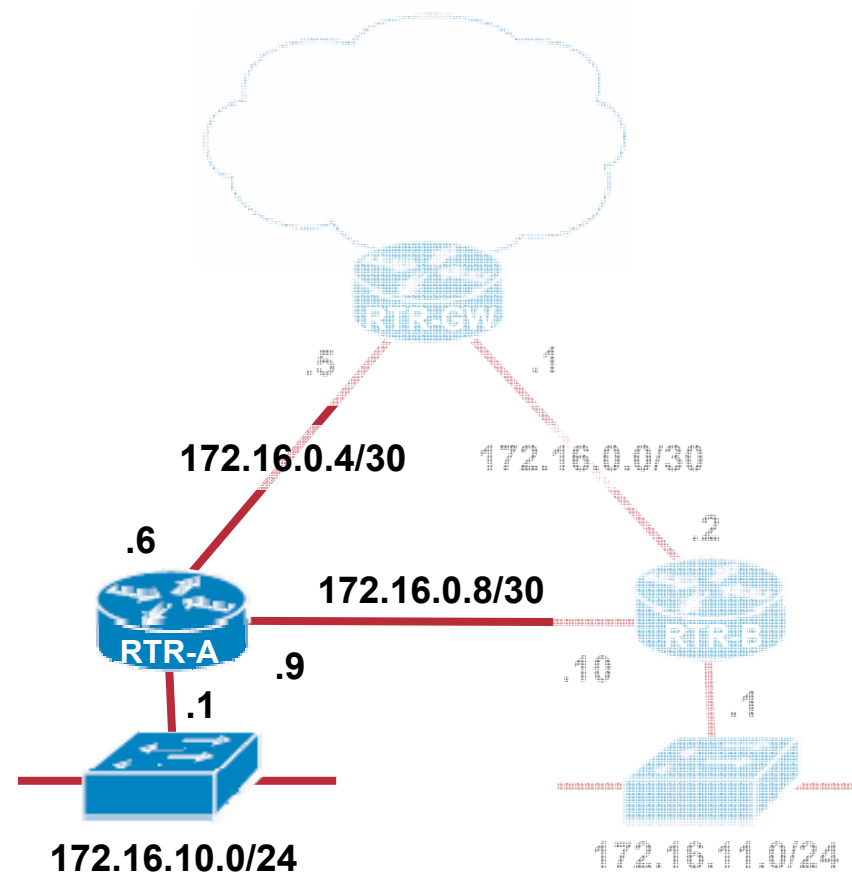
```
router rip
```

```
version 2
```

```
network 172.16.0.4/30
```

```
network 172.16.0.8/30
```

```
network 172.16.10.0/24
```

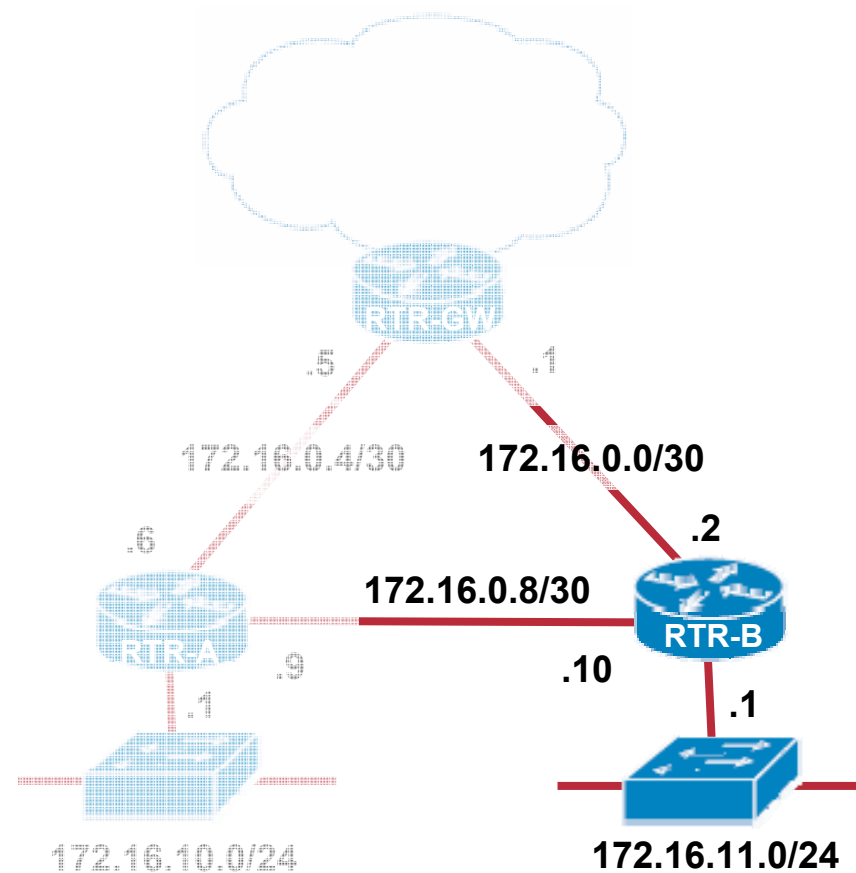


Protokół routingu RIP

Scenariusz #2 - konfiguracja

- Na RTR-B:
rozgłosić widziane sieci

```
router rip
  version 2
  network 172.16.0.0/30
  network 172.16.0.8/30
  network 172.16.11.0/24
```



Protokół routingu RIP

Scenariusz #2 – sprawdzenie konfiguracji

```
rtr-a-ripd# show ip rip
```

Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP

Sub-codes:

(n) - normal, (s) - static, (d) - default, (r) - redistribute,
(i) - interface

	Network	Next Hop	Metric From	Tag	Time
R(n)	0.0.0.0/0	172.16.0.5	1 172.16.0.5	0	02:43
C(i)	172.16.0.4/30	0.0.0.0	1 self	0	
C(i)	172.16.0.8/30	0.0.0.0	1 self	0	
C(i)	172.16.10.0/24	0.0.0.0	1 self	0	
R(n)	172.16.11.0/24	172.16.0.8	1 172.16.0.8	0	02:39

Protokół routingu RIP

Scenariusz #2 – sprawdzenie konfiguracji po awarii łącza do 172.16.0.5

```
rtr-a-ripd# show ip rip
```

Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP

Sub-codes:

(n) - normal, (s) - static, (d) - default, (r) - redistribute,
(i) - interface

	Network	Next Hop	Metric From	Tag	Time
R(n)	0.0.0.0/0	172.16.0.8	2 172.16.0.8	0	03:15
C(i)	172.16.0.8/30	0.0.0.0	1 self	0	
C(i)	172.16.10.0/24	0.0.0.0	1 self	0	
R(n)	172.16.11.0/24	172.16.0.8	1 172.16.0.8	0	03:11

PROTOKÓŁ ROUTINGU OSPF



Protokół OSPF

Historia

- Tworzony od 1987 przez IETF
- OSPFv2 opublikowana w RFC 1247 w 1991
- Ostatnia wersja OSPFv2 – RFC2328
- Wersja obsługująca IPv6 – OSPFv3 – RFC2740
- Metryką jest koszt trasy
 - 100Mbit/s = 10
 - 1Gbit/s = 1
- Szybka konwergencja i relatywnie małe wymagania
 - hierarchiczny podział na obszary

Protokół routingu OSPF

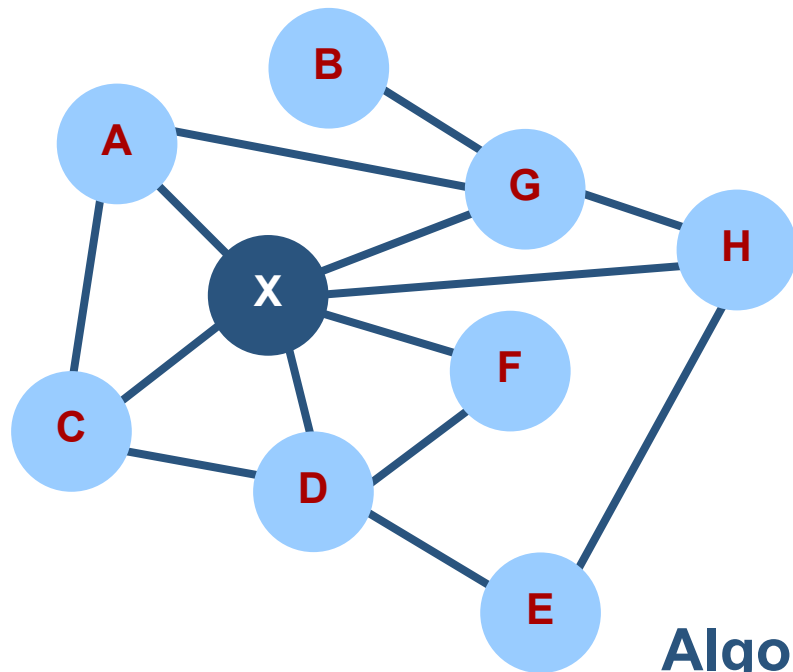
Jak działa?

- **OSPF posługuje się hierarchiczną strukturą sieci:**
 - obszar backbone (area 0)
 - obszary podłączone różnych typów
- **Każdy z obszarów musi być połączony do obszaru 0**
 - jeśli nie może – przez link wirtualny
- **Routery identyfikowane są za pomocą router-id**
 - najwyższy adres IP ze wszystkich interfejsów
 - pierwszeństwo mają interfejsy loopback – użyj ich!

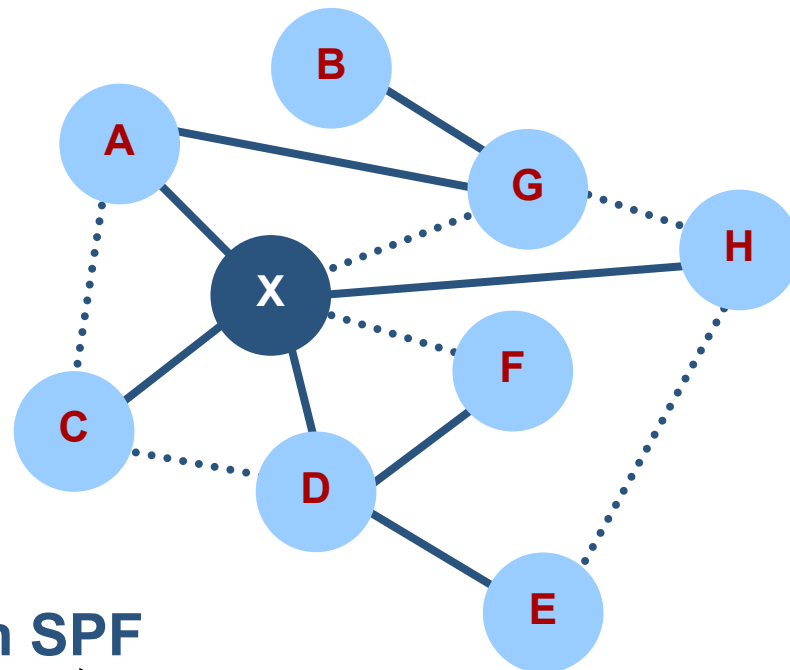
Protokół routingu OSPF

Jak działa?

Baza topologii (łącze-stan)



Drzewo najkrótszych ścieżek



Algorytm SPF
→

Protokół routingu OSPF

Jak działa?

- Routery wymieniają się **LSA** – Link State Advertisement
- W zależności od obszaru i roli routera, zestaw LSA może być różny

Typ	Nazwa LSA
1	Router
2	Network
3	Summary Network
4	Summary ASBR
5	External
7	NSSA

Protokół routingu OSPF

Algorytm SPF

- Router utrzymuje osobną bazę topologii dla każdego obszaru, do którego należy
- Routery w tym samym obszarze posiadają **tą samą** bazę topologii
- SPF działa osobno dla każdego obszaru
- Flooding LSA odbywa się tylko w granicach obszaru

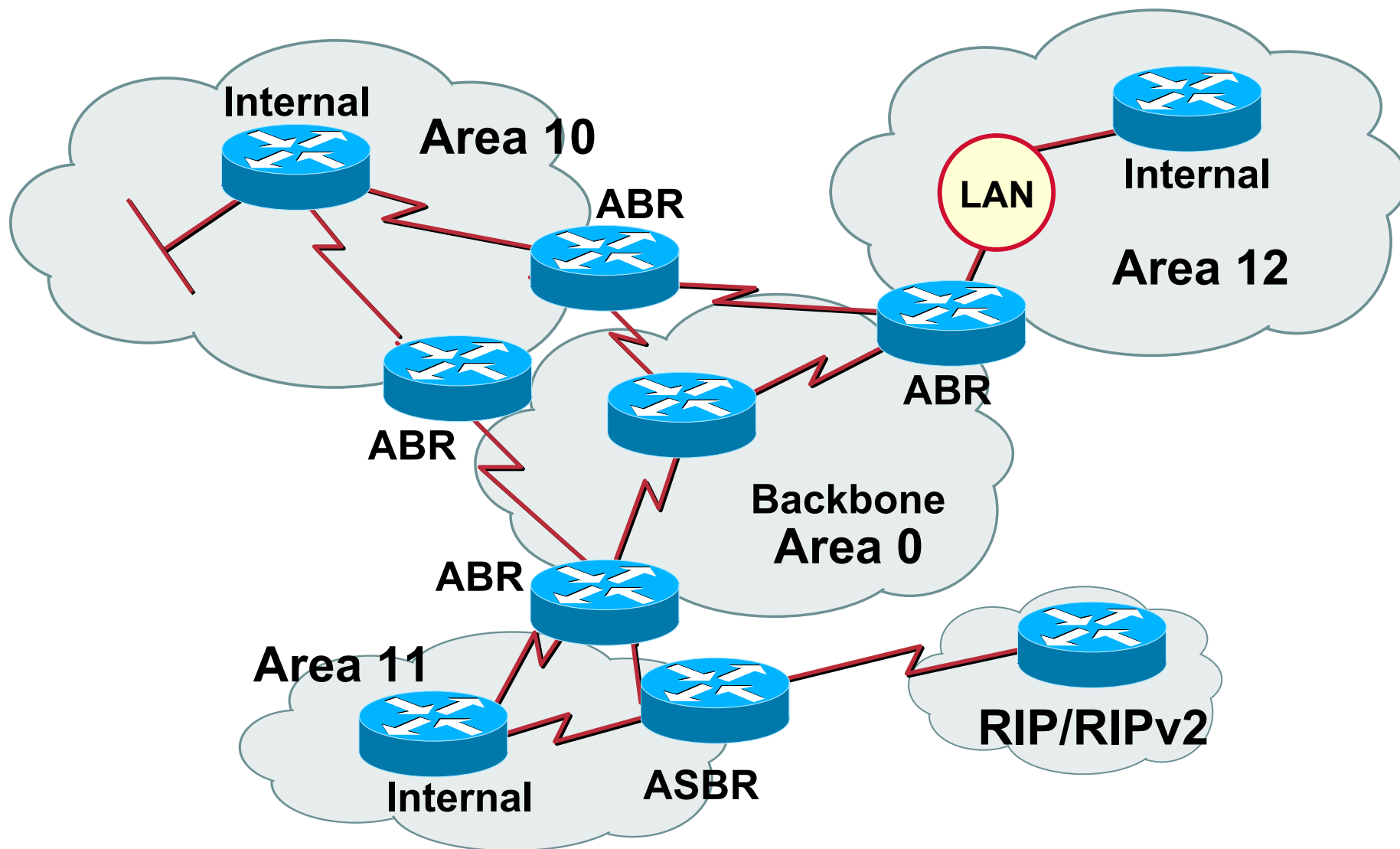
Protokół routingu OSPF

Role routerów

- **Internal router** – wszystkie interfejsy w tym samym obszarze
- **Backbone router** – wszystkie interfejsy w obszarze 0
- **Area Border Router (ABR)**
 - router posiada interfejsy przynajmniej w dwóch obszarach
- **Autonomous System Boundary Router (ASBR)**
 - redystrybucja informacji z innych RIB – tras połączonych, statycznych, RIP itp.

Protokół routingu OSPF

Przykład topologii



Protokół OSPF

Rodzaje łączy

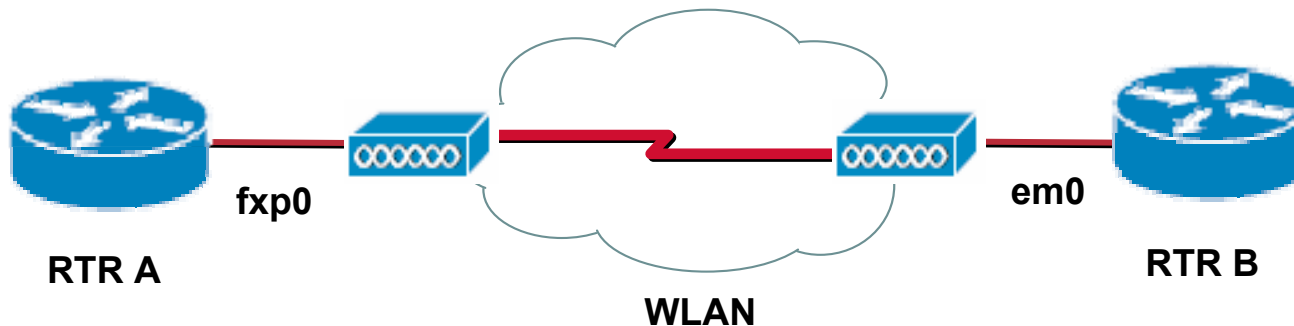
- **Broadcast**
 - np. Ethernet
 - wybór DR i BDR dla segmentu
- **Non-Broadcast Multi-Access (NBMA)**
 - np. Frame Relay
 - konieczność wskazania sąsiadów poleceniem neighbour
- **Point-to-Point, Point-to-Multipoint, Point-to-Multipoint Non-Broadcast**

Protokół OSPF

Kiedy point-to-point?

- Zalecane w przypadku sieci używających jako warstwy transportowej 802.11:

oszczędzamy czas potrzebny na elekcję DR/BDR



UWAGA: koszt trasy przez AP jest równy wynegocjowanej z AP prędkości na interfejsie – zwykle 100Mbit/s (koszt 10). Jeśli posiadasz inne, równoległe i szybsze łącze warto zastanowić się nad zmianą (zwiększeniem) kosztu przez ten interfejs!

Protokół OSPF

Konfiguracja point-to-point

```
interface fxp1
  ip ospf network point-to-point
  ip ospf cost 20 ! zawyżenie kosztu do odpowiadającego łącza 50Mbit/s
                  ! aby inne łącze o realnej przepustowości 100Mbit/s
                  ! było atrakcyjniejsze
q-ospfd# show ip ospf interface fxp1
fxp1 is up
  Internet Address 192.168.50.1/30, Broadcast 192.168.50.3, Area 0.0.0.50
  Router ID 172.16.254.10, Network Type POINTOPOINT, Cost: 20
  Transmit Delay is 1 sec, State Point-To-Point, Priority 1
  No designated router on this network
  No backup designated router on this network
  [...]
  Neighbor Count is 1, Adjacent neighbor count is 1
```

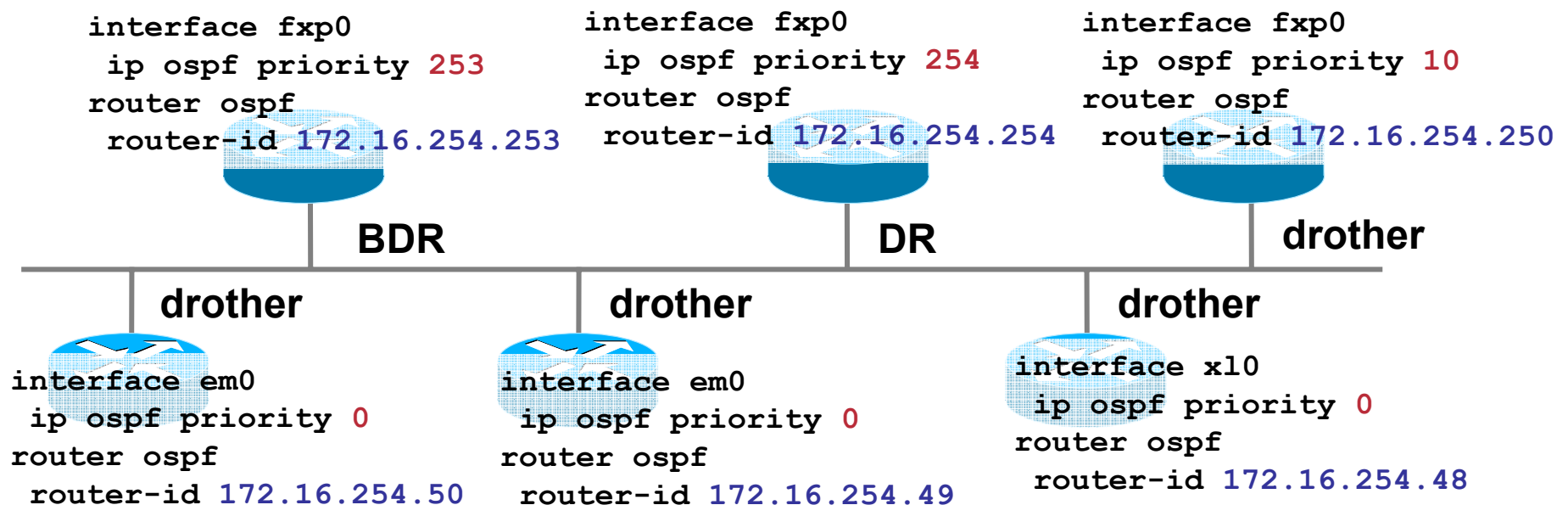
Protokół OSPF

Kiedy broadcast?

- We wszystkich topologiach, w których wiele routerów łączy wspólny segment Ethernet

priority routera (0-254, 254 najwyższy, 0 – nie zostanie DR)

wyższe router-id (zalecane stabilne rozplanowanie numeracji interfejsów loopback!)



Protokół OSPF

Konfiguracja broadcast

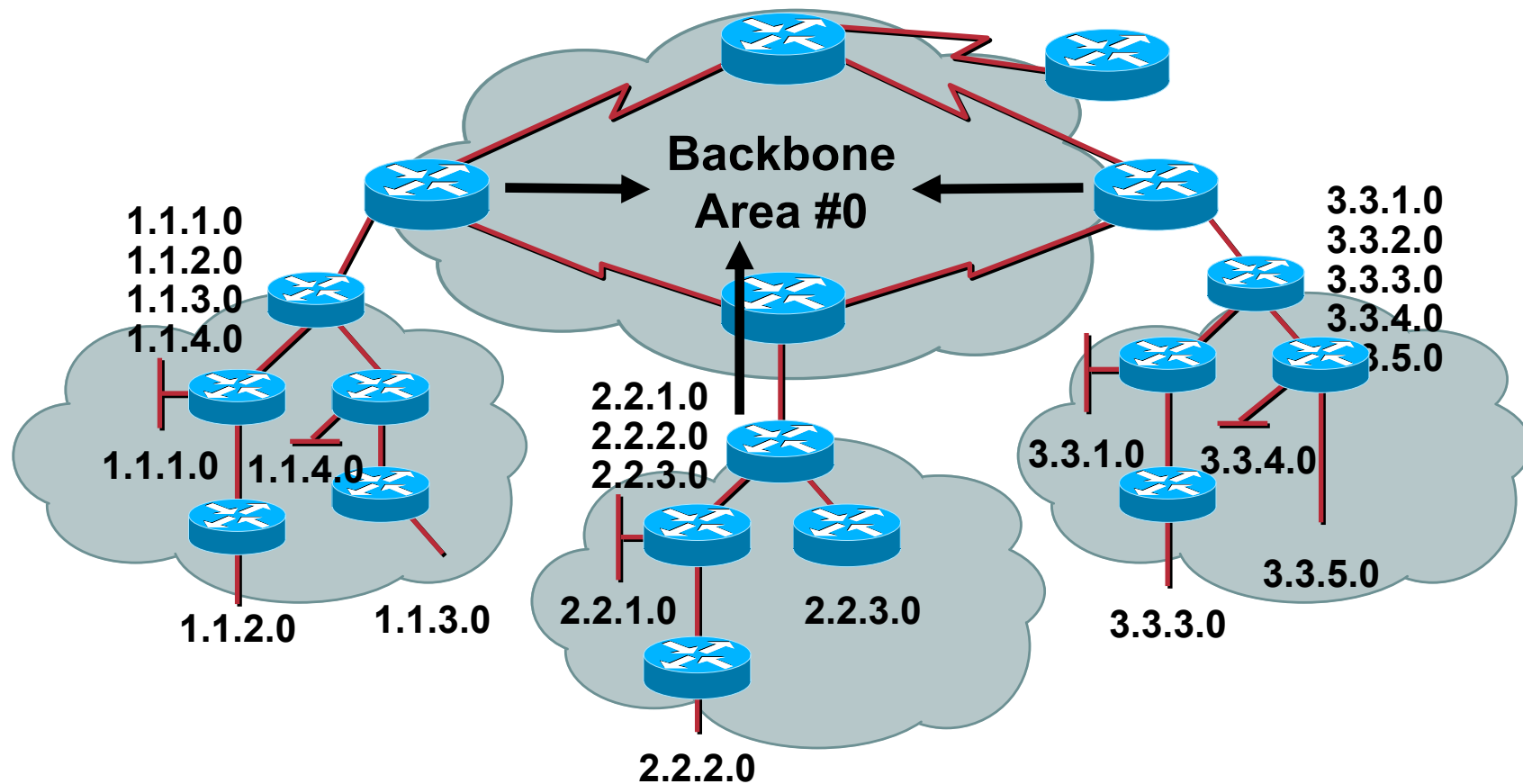
```
interface fxp0
  ip ospf network broadcast ! domyślnie dla interfejsów Ethernet
```

```
q-ospfd# show ip ospf interface fxp0
```

```
fxp0 is up
[...]
Internet Address 192.168.0.100/24, Broadcast 192.168.0.255, Area 0.0.0.0
Router ID 172.16.254.10, Network Type BROADCAST, Cost: 10
Transmit Delay is 1 sec, State DROther, Priority 1
Designated Router (ID) 172.16.254.201, Interface Address 192.168.0.201
Backup Designated Router (ID) 172.16.254.200, Interface Address 192.168.0.200
Multicast group memberships: OSPFAllRouters
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Hello due in 00:00:04
Neighbor Count is 2, Adjacent neighbor count is 2
```

Protokół routingu OSPF

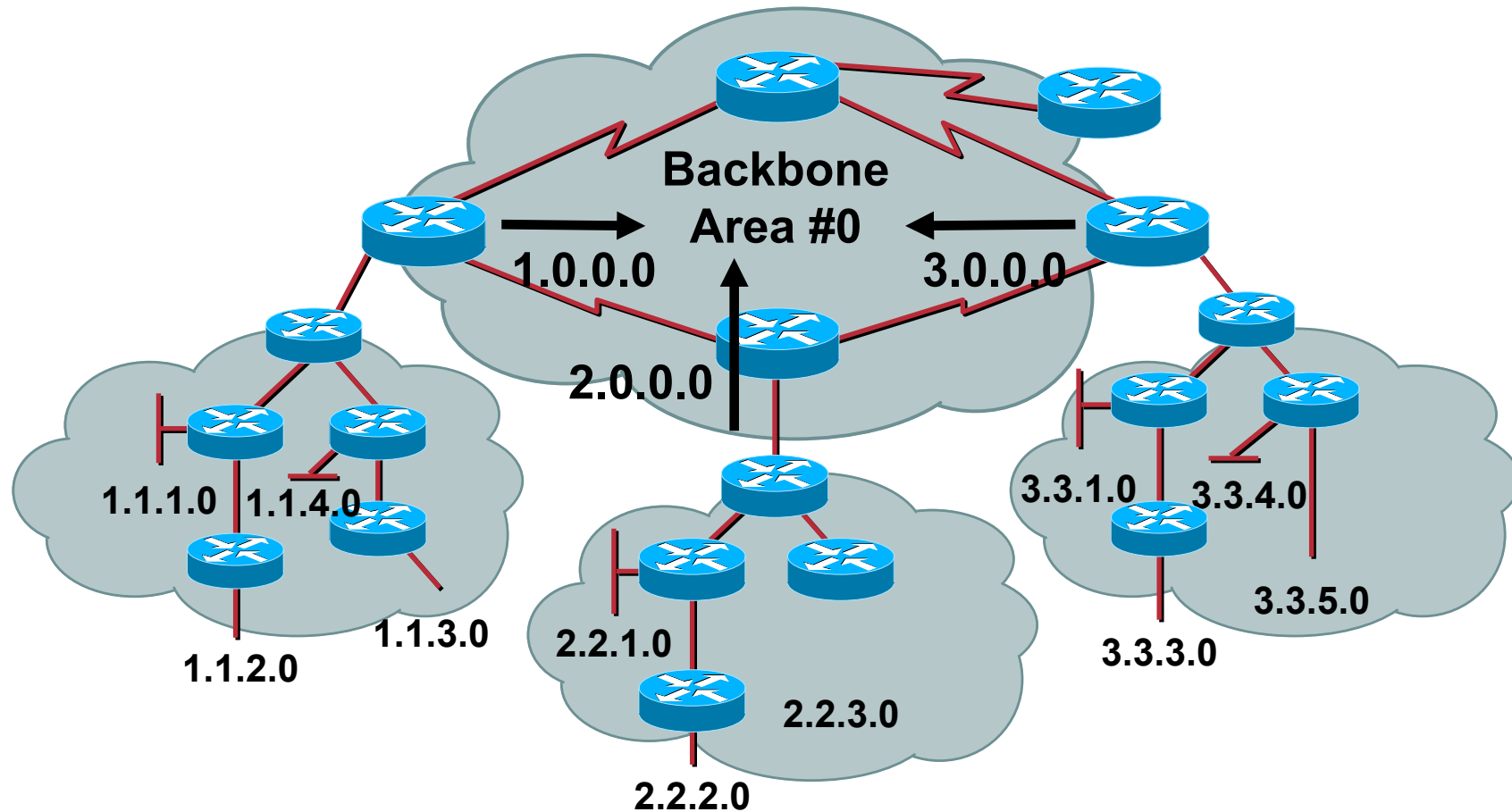
Sumaryzacja - brak



- **Wiele prefiksów w Area 0**
- **Dodatkowe obciążenie dla algorytmu SPF i routerów backbone**

Protokół routingu OSPF

Sumaryzacja - poprawnie



- Rozgłaszamy tylko summary LSA
- Zmiany stanów łącz nie propagują się pomiędzy obszarami

Protokół routingu OSPF

Sumaryzacja - konfiguracja

```
router ospf
  ospf router-id 172.16.254.11
  network 192.168.0.0/24 area 0.0.0.0
  network 192.168.50.0/30 area 0.0.0.50
  area 0.0.0.50 range 172.16.10.0/24
```


Protokół routingu OSPF

Scenariusz #1 – OSPF i wstrzyknięcie trasy domyślnej - Quagga

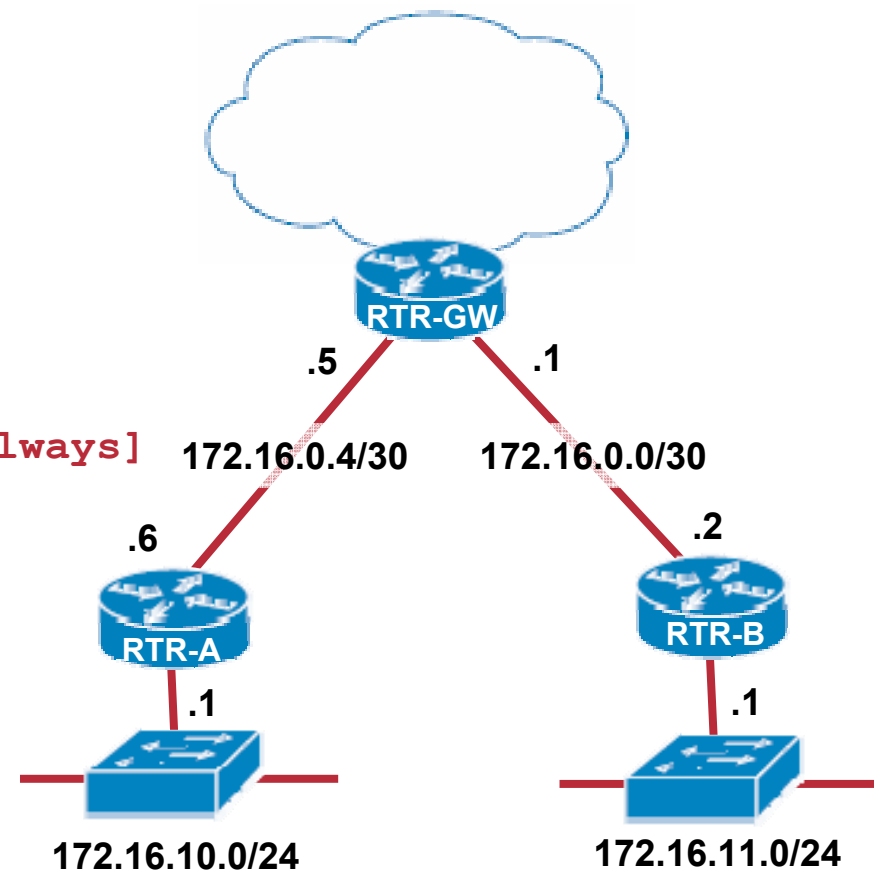
- Na RTR-GW:
rozgłosić default

```
router ospf
```

```
network 172.16.0.0/30 area 10
```

```
network 172.16.0.8/30 area 11
```

```
default-information originate [always]
```



Protokół routingu OSPF

Scenariusz #1 – OSPF i wstrzyknięcie trasy domyślnej - OpenOSPFd

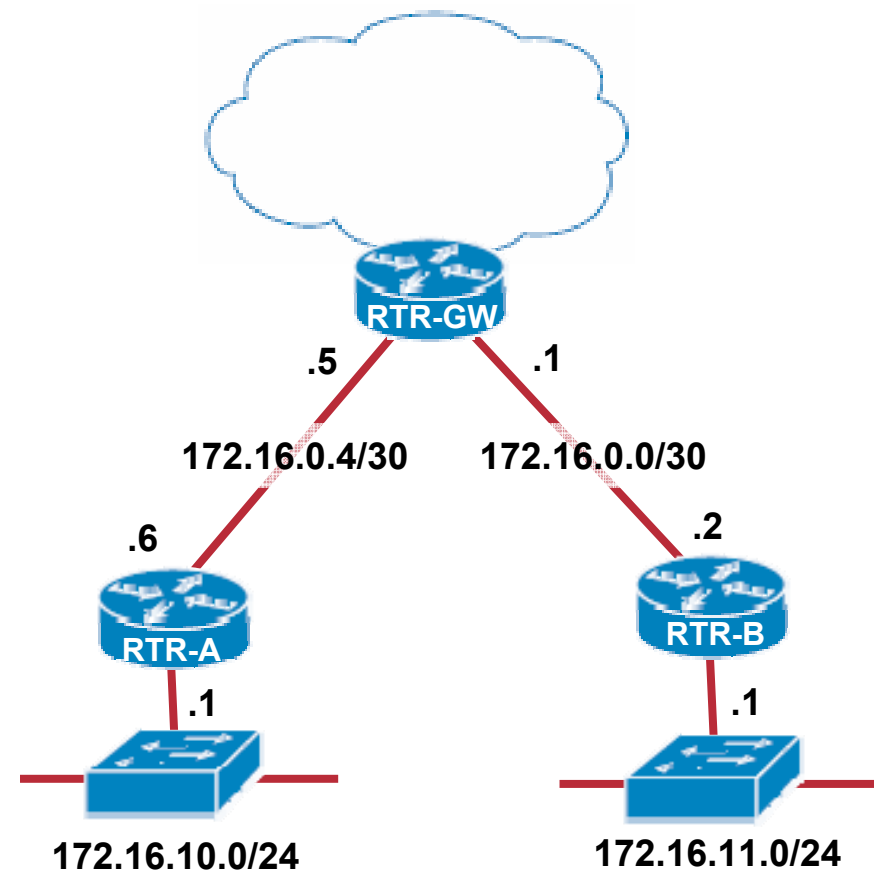
- Na RTR-GW:
rozgłosić default

[...]

```
fib-update yes
```

```
redistribute default
```

[...]



Protokół routingu OSPF

Scenariusz #2 – dwa OSPF ASBR

- Na RTR-G1:
rozgłosić default

```
router ospf
```

```
network 172.16.0.0/25 area 0
```

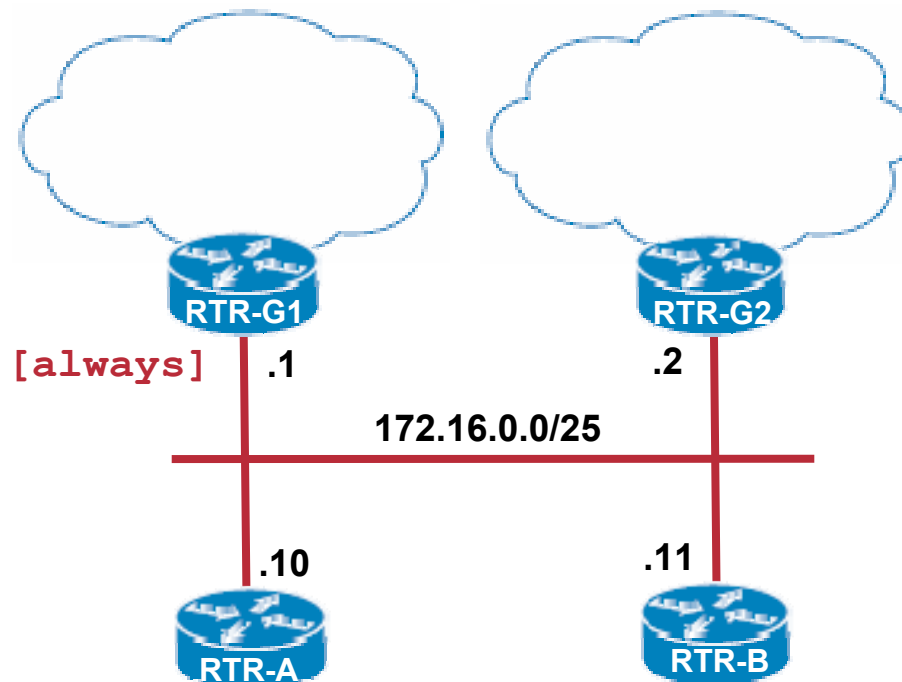
```
default-information originate [always]
```

- Na RTR-G2:
rozgłosić default

```
router ospf
```

```
network 172.16.0.0/25 area 0
```

```
default-information originate [always]
```



Protokół routingu OSPF

Scenariusz #2 – dwa OSPF ASBR

- Na RTR-A:
który default?

```
rtra-ospfd# show ip ospf database
```

```
OSPF Router with ID (172.16.254.203)
```

```
AS External Link States
```

Link ID	ADV Router	Age	Seq#	CkSum	Route
0.0.0.0	172.16.254.254	342	0x80000002	0x347c	E2 0.0.0.0/0 [0xa]
0.0.0.0	172.16.254.253	1862	0x80000002	0x2e81	E2 0.0.0.0/0 [0xa]

Protokół routingu OSPF

Scenariusz #2 – dwa OSPF ASBR

- Na RTR-A:
który default?

```
rtra-ospfd# show ip ospf route
```

```
[...]
```

```
===== OSPF external routing table =====
```

```
N E2 0.0.0.0/0                [20/10] tag: 10  
                               via 172.16.0.1, fxp0  
                               via 172.16.0.2, fxp0
```

UWAGA: wiele tras o tej samej metryce pojawi się tylko wtedy, gdy Quagga została skompilowana z opcją multipath

Protokół routingu OSPF

Scenariusz #2 – dwa OSPF ASBR

- Na RTR-A:
który default?

```
$ netstat -nrf inet
```

```
Routing tables
```

```
Internet:
```

Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	172.16.0.1	UG1	0	0	fxp0	
127.0.0.1	127.0.0.1	UH	1	4816	lo0	

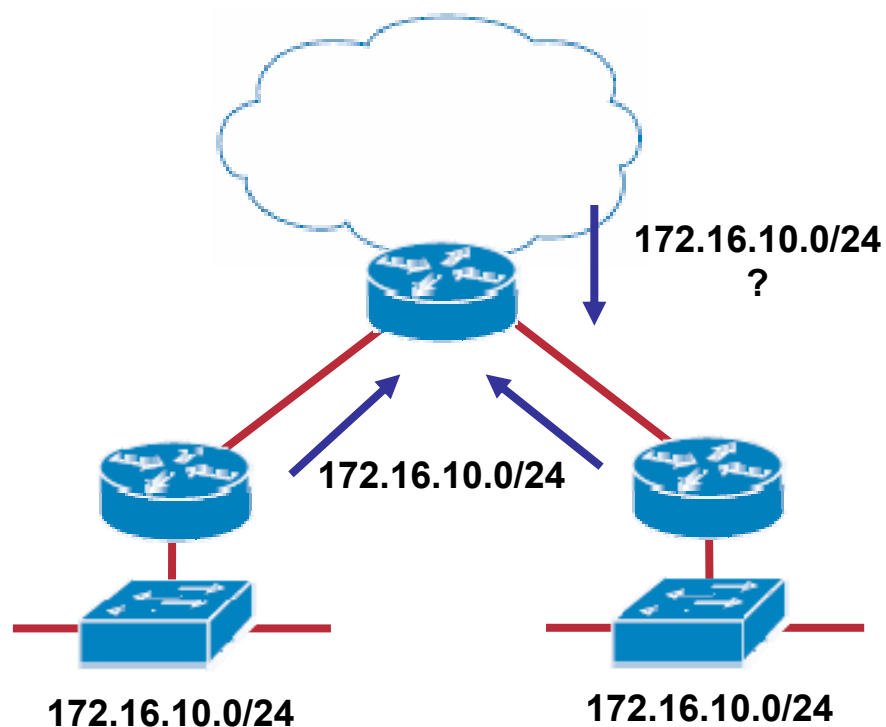
ROUTING DYNAMICZNY A INNE ZAGADNIENIA



Routing dynamiczny a inne zagadnienia

NAT

- Typowa „sieć osiedlowa” stosuje NAT



- Uwaga na nakładające się podsieci prywatne przy redystrybucji tras połączonych/statycznych do protokołów routingu dynamicznego!
- Trasy można odfiltrować nawet, jeśli redystrybucja jest włączona

Routing dynamiczny a inne zagadnienia

NAT

- **Filtrowanie sieci z RFC1918 przy redystrybucji tras połączonych/statycznych do OSPFa:**

```
router ospf
  redistribute kernel route-map NORFC1918

route-map NORFC1918 permit 10
  match ip address prefix-list 10

ip prefix-list 10 deny 10.0.0.0/8 le 32
ip prefix-list 10 deny 172.16.0.0/12 le 32
ip prefix-list 10 deny 192.168.0.0/16 le 32
ip prefix-list 10 permit any
```

Routing dynamiczny a inne zagadnienia

Filtrowanie ruchu

- Protokoły routingu mają swoje wymagania co do przepuszczanego ruchu:

RIPv1 – 520/udp

RIPv2 – multicast pod adres 224.0.0.9 na 520/udp

OSPF – protokół IP numer 89

BGP – protokół TCP na/z portu 179

- Przykładowe regułki:

```
pass in on $ext_if proto udp from any port 520 to  
224.0.0.9/32 port 520 keep state
```

```
pass out on $ext_if proto udp from $ext_if port 520 to  
224.0.0.9/32 port 520 keep state
```

```
pass in on $ext_if proto 89 from any to 224.0.0.0/24
```

```
pass out on $ext_if proto 89 from $ext_if to 224.0.0.0/24
```

```
pass in on $ext_if proto tcp from $bgp_gw port 179 to  
$ext_if flags S/SA keep state
```

```
pass out on $ext_if proto tcp from $ext_if to $bgp_gw  
port 179 flags S/SA keep state
```

Routing dynamiczny a inne zagadnienia

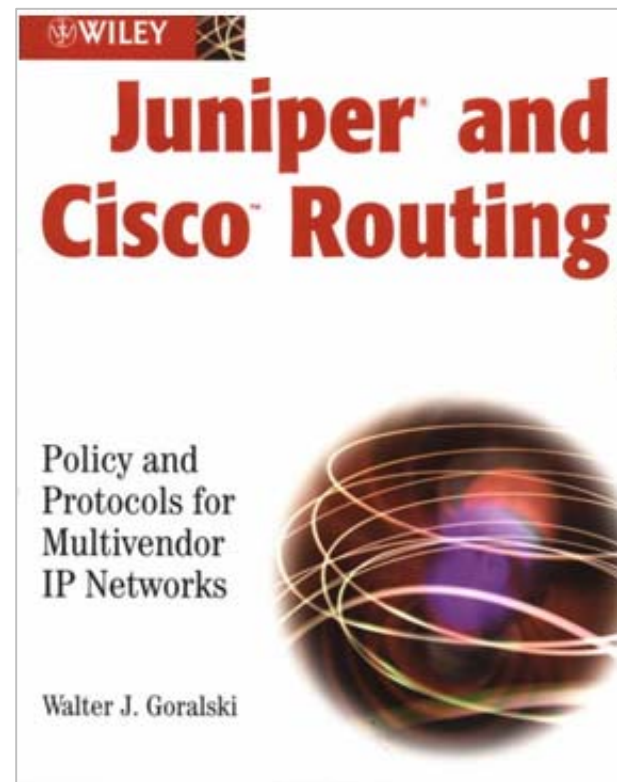
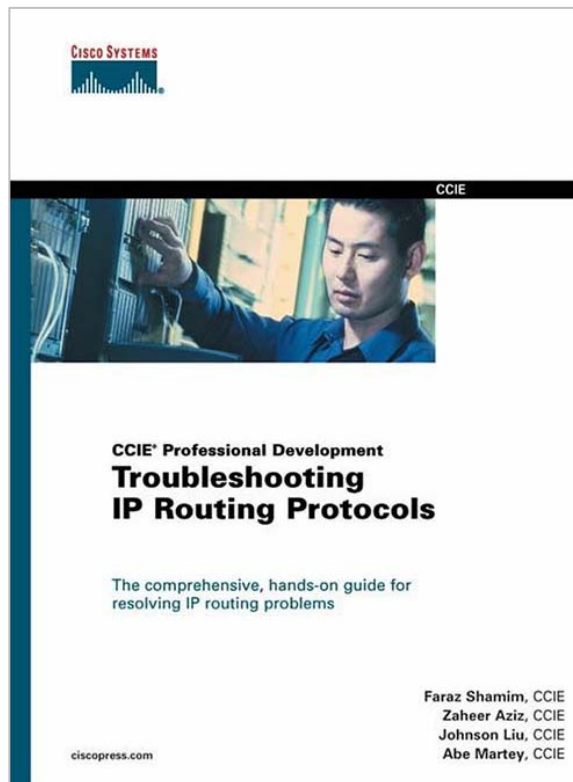
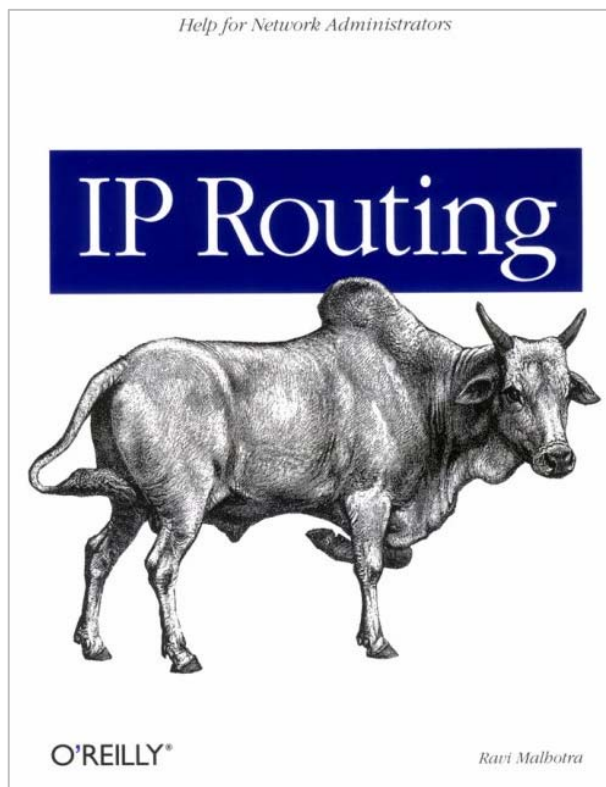
Tunelowanie IP-w-IP i GRE

- **Tunelowanie IP-w-IP (**gif**) nie przenosi żadnego ruchu poza IP (jak sama nazwa wskazuje)**
 - brak obsługi multicastów (RIPv2 i OSPF)
- **Tunelowanie GRE (**gre**) pozwala przenieść multicasty oraz inne protokoły warstwy 3 – w szczególności IPX**
 - wygodne i funkcjonalne połączenie dwóch sieci z możliwością zapewnienia działania protokołów RIPv2 i OSPF
 - quagga automatycznie konfiguruje interfejsy gre i gif jako punkt-punkt, ale tylko te, które istnieją zanim zostanie uruchomiona

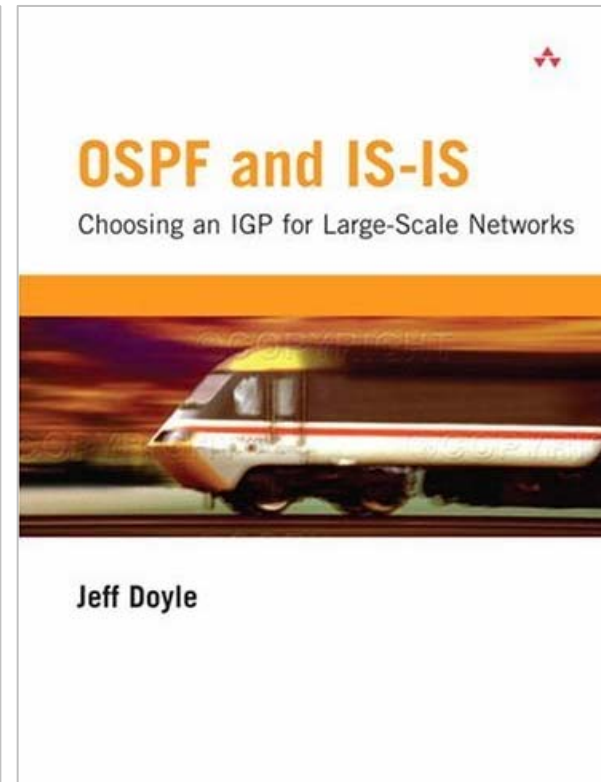
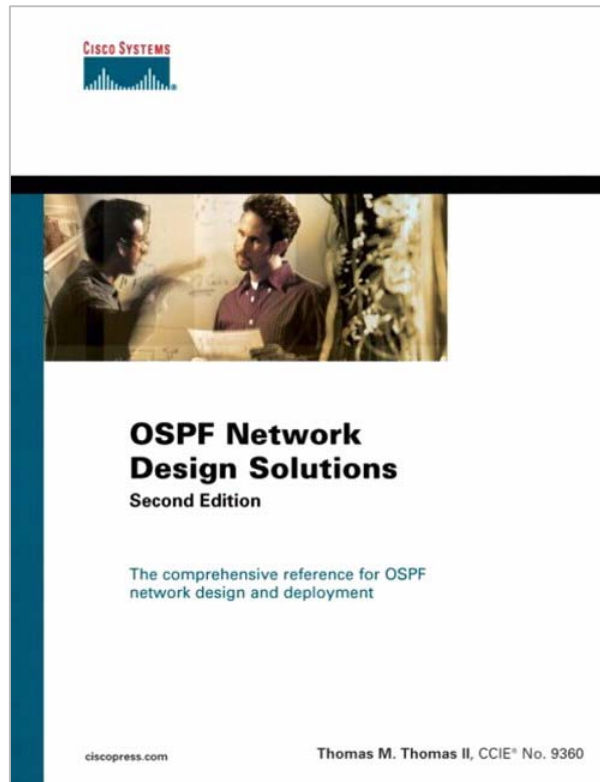
GDZIE WARTO RZUCIĆ OKIEM



Książki



Książki



Zasoby WWW

- **Pakiet Quagga:**

<http://www.quagga.net>

- **Pakiet XORP:**

<http://www.xorp.org>

- **Demon OpenOSPFd:**

<http://www.openbsd.org>

Q&A



PROTOKOŁY DYNAMICZNEGO ROUTINGU IP – RIP i OSPF PODSTAWY DZIAŁANIA ZAGADNIENIA PROJEKTOWE I WDROŻENIOWE

Dziękuję za uwagę

**lukasz@bromirski.net
lbromirski@cisco.com**