

# Mechanizmy routingu w systemach wolnodostępnych



**Łukasz Bromirski**  
[lukasz@bromirski.net](mailto:lukasz@bromirski.net)

# Agenda

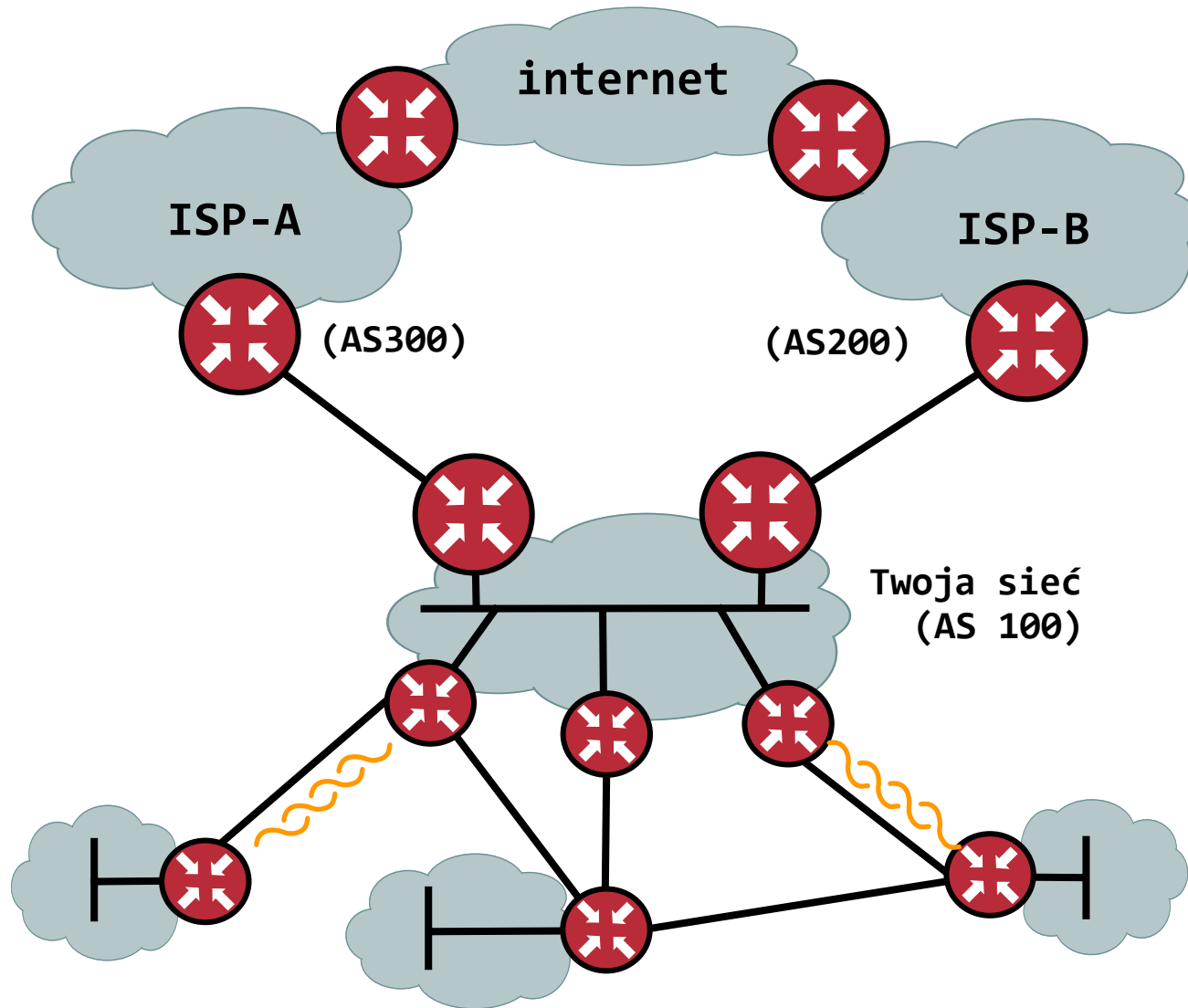
- **Routing IP**
- **Quagga/OpenBGPD/OpenOSPF/XORP**
- **Gdzie i dlaczego OSPF?**
  - OSPF w praktyce
- **Gdzie i dlaczego BGP?**
  - BGP w praktyce
- **Routing dynamiczny a inne zagadnienia**
  - NAT, QoS, multicasty i VPNy, redundancja
- **Q&A**

## **Wymagana** będzie...



- ...znajomość chociażby na minimalnym poziomie zagadnień związanych z routingiem dynamicznym i protokołów OSPF i BGP
- ...pakietu Quagga i/lub OpenSPFd/BGPd
- Na końcu tej prezentacji znajdują się odnośniki do materiałów uzupełniających – również tych bardzo podstawowych

# O czym będzie ta sesja?



# POWTÓRKA Z ROZRYWKI: ROUTING IP



# Routing IP

O czym mówimy?

- Routing IP to decyzja (standardowo) podejmowana 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

# Routing IP

O czym mówimy?

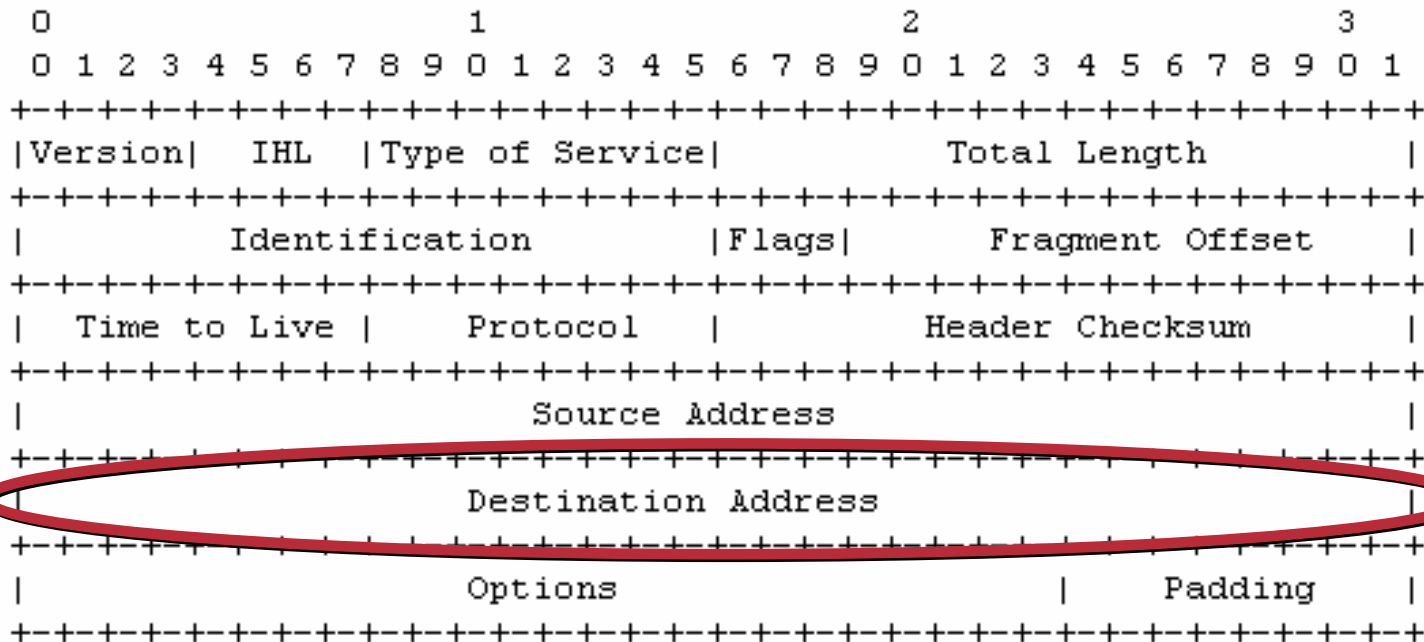
- Narzędzia systemowe wpływają na **FIB**  
opcje **FIB\_HASH/FIB\_TRIE** w kernelach 2.6.x
- Narzędzia konkretnej aplikacji wpływają na **RIB**  
właściwy dla pakietu
- Dodatkowo Linux posiada bogate opcje routingu  
na podstawie adresu źródłowego  
pakiet iproute2  
integracja z aplikacjami zewnętrznymi – tablice, realms

Realms: <http://vcalinus.gemenii.ro/quaggarealms.html>

# Routing IP

Czym zajmuje się router?

- Router otrzymuje datagramy IPv4 w postaci:



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



# Routing IP

## Budowa FIB

- Zawartość FIB:

```
[me@slack ~]$ ip route list
```

```
10.0.0.0/24          dev eth0  proto kernel src 10.0.0.100
169.254.0.0/16      dev eth0  scope link
default via 10.0.0.1 dev eth0
```

```
[me@slack ~]$ route -n
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	If
10.0.0.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	0	0	0	eth0
0.0.0.0	10.0.0.1	0.0.0.0	UG	0	0	0	eth0

**QUAGGA/ZEBRA vs  
XORP vs  
OpenOSPFd**



## Quagga/Zebra (0.99.5/0.98.6)

- 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 jednocześnie”
- Quagga jest przygotowana do przechowywania wielu takich samych tras w RIB:

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

# XORP

## (1.3)

- XORP również posiada budowę modułarną
- Router manager (**rtrmngrr**) nadzoruje pracę grupy procesów
- Dwie osobne ścieżki:
  - unicast: BGP4, RIPv1/2 i RIPvng, OSPFv2
  - multicast: PIM-SM, IGMPv1/v2, MLDv1
- Wydzielony RIB dla wszystkich protokołów
- Wydzielona FEA, pozwalająca uniezależnić się od systemu/dostępnych interfejsów
- Dostępna powłoka **xorpsd** do zarządzania
- Wiele rozmaitych problemów

# OpenOSPFd

- Projekt zespołu OpenBSD

Henning Brauer, Claudio Jeker & Esen Norby

- Projekt w trakcie dopracowywania

IPv6

problemy z redystrybucją/wstrzyknięciem trasy default

- Tradycyjny zestaw narzędzi i plików:

`ospfd` – demon odpowiedzialny za protokół

`ospfctl` – narzędzie do kontroli

`/etc/ospfd.conf` - konfiguracja

# OpenBGPd

- Ten sam team zespołu OpenBSD

Henning Brauer, Claudio Jeker & Esen Norby i inni

- Projekt w trakcie dopracowywania

IPv6 + różne zagadnienia z  
filtrowaniem/utrzymywaniem atrybutów tras

- Podobnie jak OpenOSPFd:

**bgpd** – demon odpowiedzialny za protokół

**bgpctl** – narzędzie do kontroli

**/etc/bgpd.conf** - konfiguracja

Linux port PoC: <http://hasso.linux.ee/doku.php/english:network:openbgpd>

## Podsumowanie obecnego stanu

- Wiele możliwości, potęgowanych możliwościami pakietu **iproute2**
- Nadal brak jasnej wizji dotyczącej zarządzaniem przez kernel różnymi źródłami informacji routingowej
- Ten sam problem mają wszystkie dystrybucje Linuksa oraz Free/Net/OpenBSD

# GDZIE I DLACZEGO OSPF?





## **Dlaczego OSPF?**

- **Najszybszy **standardowy** protokół IGP**  
IGP = wewnętrzny (ang. internal gateway protocol)
- **Większość małych i średnich sieci ma strukturę hierarchiczną, lub da się ją stworzyć**
- **Dostosowywalny do większości scenariuszy za pomocą standardowych mechanizmów i narzędzi**
- **Dostępny ‘for free’ w implementacjach:**  
Quagga – ospfd  
OpenOSPFd

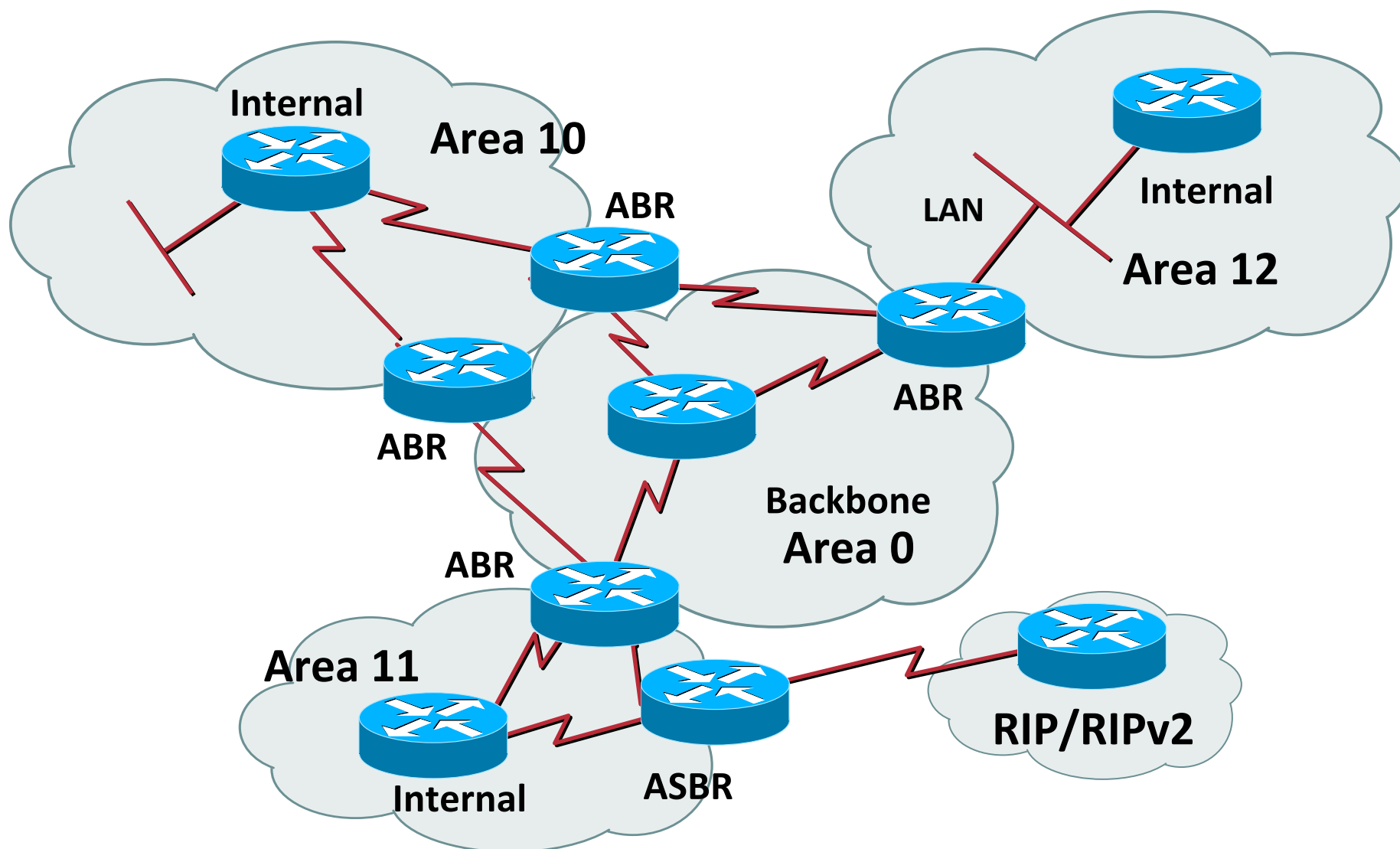
# Protokół routingu OSPF

Jak działa?

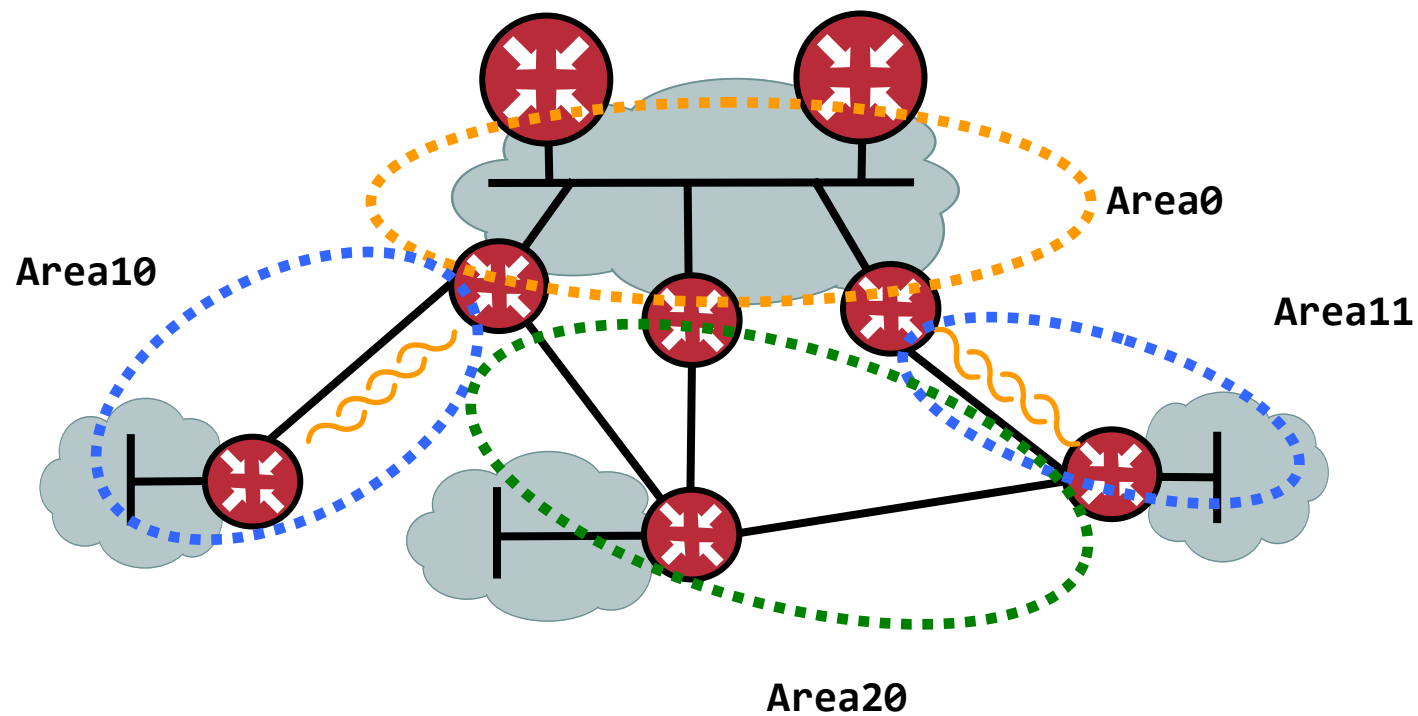
- **OSPF posługuje się hierarchiczną strukturą sieci:**
  - obszar backbone (area 0)
  - obszary podłączone różnych typów
- **Każdy z obszarów musi być połączony do obszaru 0**
  - jeśli nie może – przez link wirtualny
- **Routery identyfikowane są za pomocą **router-id****
  - najwyższy adres IP ze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

# Protokół routingu OSPF

Przykład topologii



# Jak podzielić sieć na obszary OSPF?



# Jak właściwie wybrać DR/BDRa dla segmentu?

- 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!)

```
interface eth0
ip ospf priority 253
router ospf
router-id 172.16.254.253
```



BDR

```
interface eth0
ip ospf priority 254
router ospf
router-id 172.16.254.254
```



DR

drother



```
interface eth0
ip ospf priority 0
router ospf
router-id 172.16.254.49
```

# Protokół OSPF

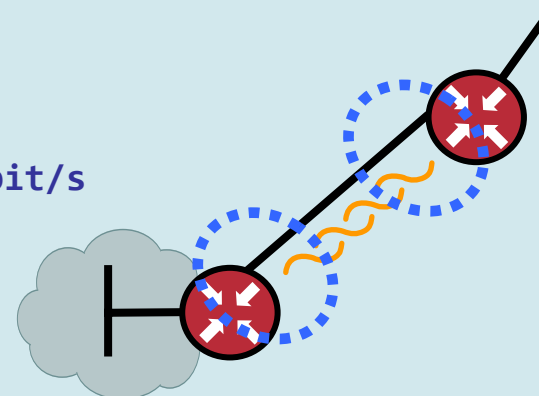
Kiedy point-to-point?

- Zalecane w przypadku sieci używających jako warstwy transportowej 802.11:

oszczędzamy czas potrzebny na elekcję DR/BDR

- Dwa interfejsy:

```
interface eth0
  ! interfejs podłączony do mostu 802.11
  ip ospf network point-to-point
  ip ospf cost 20 ! interfejs 100Mbit/s pracuje jak 5Mbit/s
interface eth1
  ! interfejs podłączony do linku 100Mbit/s
  ip ospf network point-to-point
  ip ospf cost 1 ! interfejs 100Mbit/s
```



# Protokół OSPF

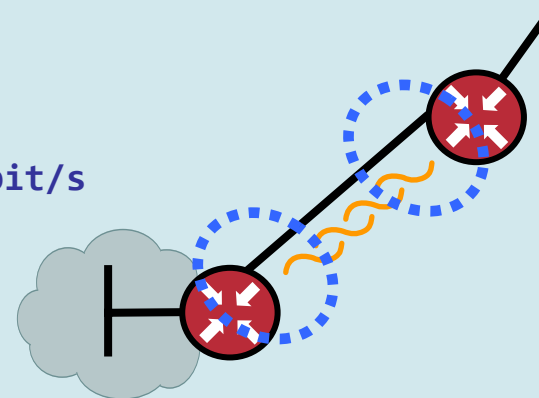
Kiedy point-to-point?

- Zalecane w przypadku sieci używających jako warstwy transportowej 802.11:

oszczędzamy czas potrzebny na elekcję DR/BDR

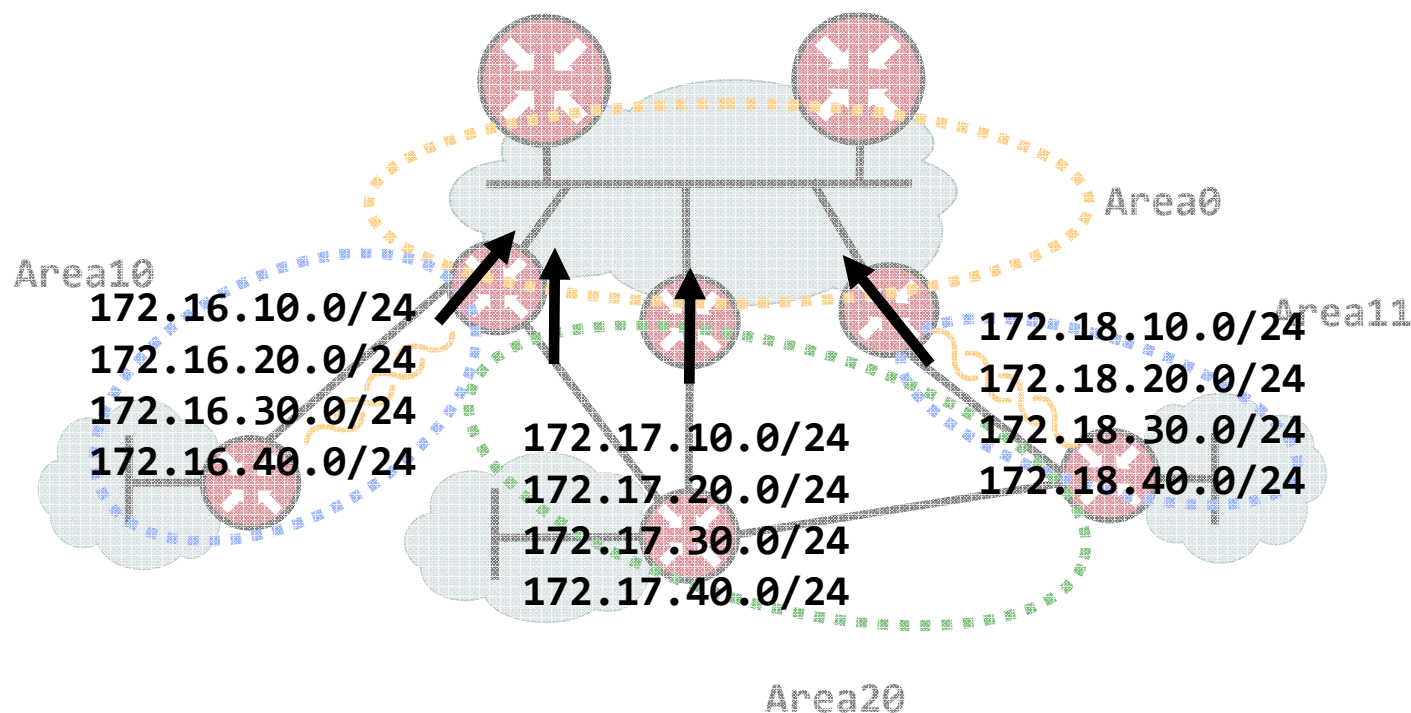
- Jeden interfejs do bridge'a i łącza 100Mbit/s:

```
interface eth0.10
  ! interfejs podłączony do mostu 802.11
  ip ospf network point-to-point
  ip ospf cost 20 ! interfejs 100Mbit/s pracuje jak 5Mbit/s
interface eth0.20
  ! interfejs podłączony do linku 100Mbit/s
  ip ospf network point-to-point
  ip ospf cost 1 ! interfejs 100Mbit/s
```



# Protokół routingu OSPF

## Sumaryzacja - brak

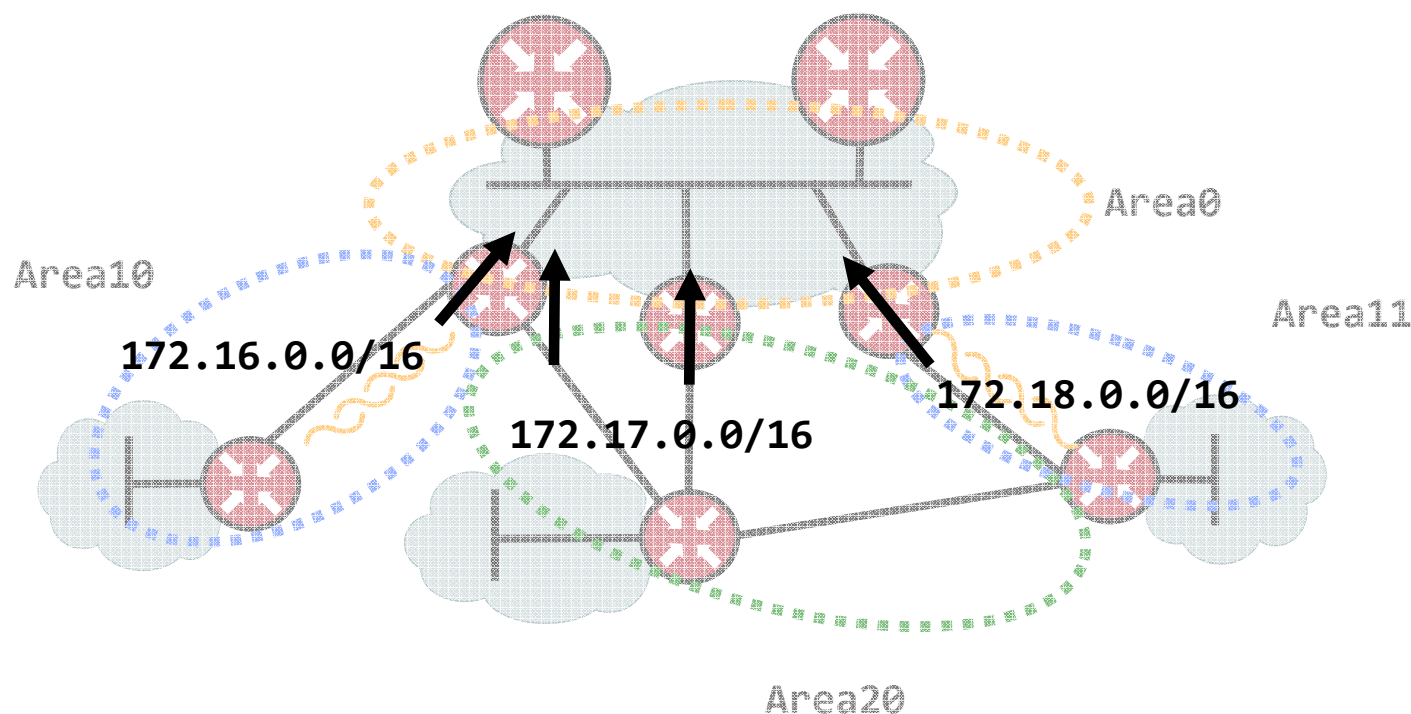


- Wiele prefiksów w Area 0
- Dodatkowe obciążenie dla algorytmu SPF i routerów backbone
- „up/down prefiksu” – obciążamy CPU routerów w A0



# Protokół routingu OSPF

## Sumaryzacja - skonfigurowana



- Pojedyncze prefiksy w Area 0
- Zmniejszamy obciążenie CPU, nie propagujemy problemów w warstwie dostępowej do szkieletu/rdzenia

# Protokół routingu OSPF

## Sumaryzacja - konfiguracja

- **Obszar 10 rozgłasza sieć 172.16.0.0/16, zamiast:**

172.16.10.0/24

172.16.20.0/24

...

```
router ospf
```

```
ospf router-id 172.16.254.11
```

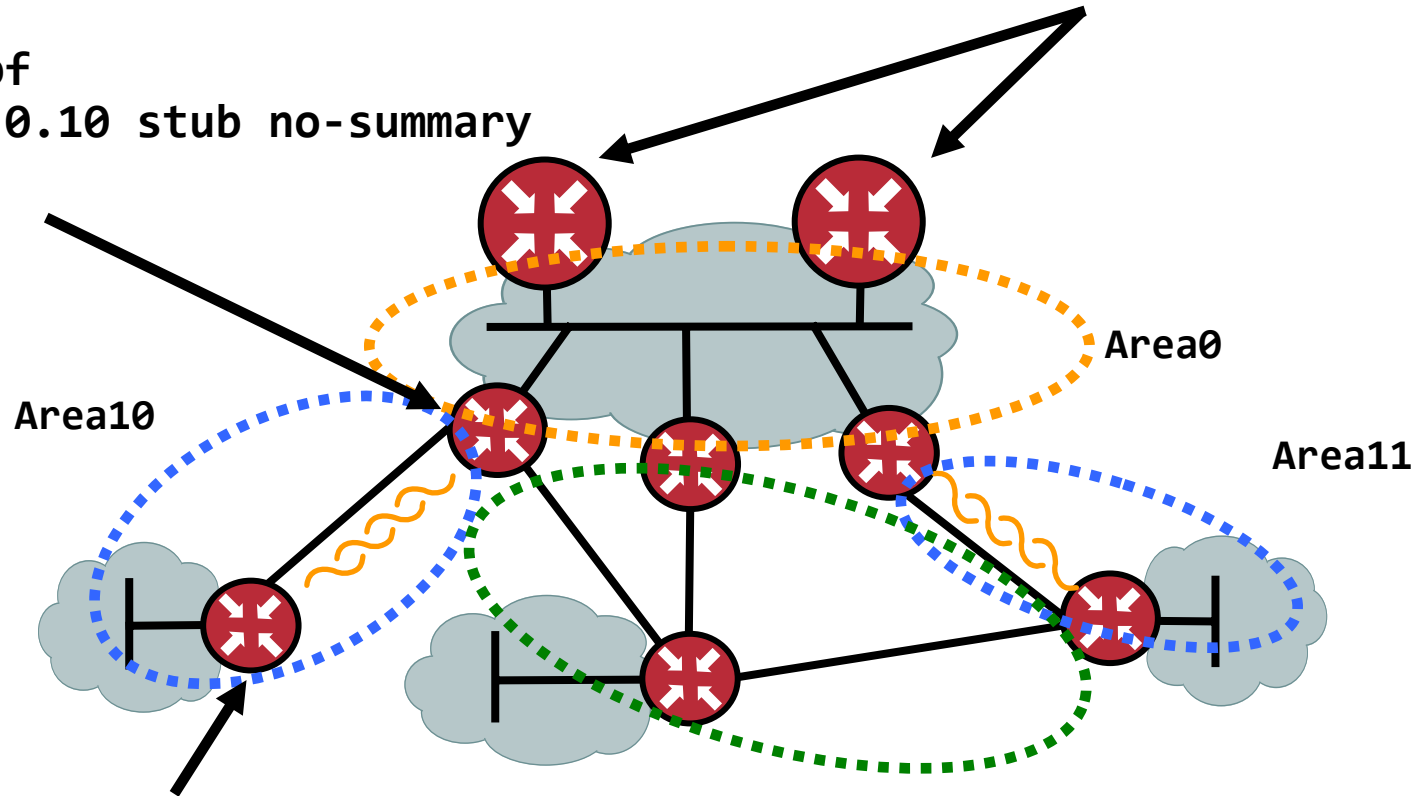
```
[...]
```

```
area 0.0.0.10 range 172.16.0.0/16
```

# Trasa domyślna

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

```
router ospf  
area 0.0.0.10 stub no-summary
```



```
router ospf  
area 0.0.0.10 stub no-summary
```

# GDZIE I DLACZEGO BGP?



# Protokół BGP

## Wstęp i rozwinięcie

- **Protokół routingu używany do wymiany informacji o osiągalności sieci (prefiksów) pomiędzy systemami autonomicznymi**

**klasy EGP – Exterior Gateway Protocol**

- **Do niedawna RFC1771 + rozszerzenia, od stycznia 2006 obowiązuje RFC4271:**

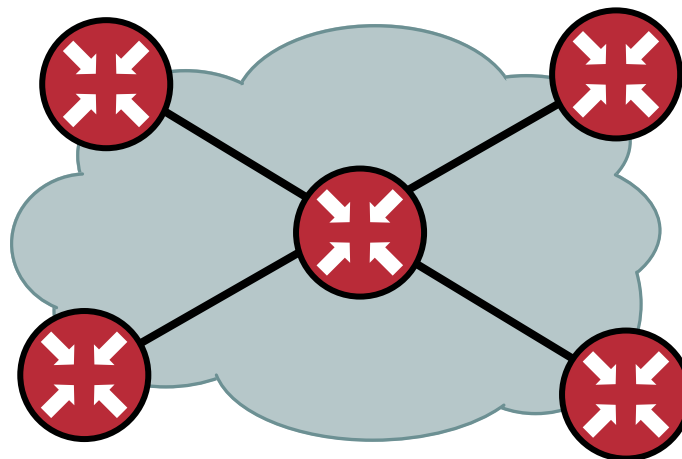
**<http://www.ietf.org/rfc/rfc4271.txt>**

**RFC4276 opisuje raport implementacyjny RFC4271**

**RFC4277 dodatkowo opisuje doświadczenia z różnymi implementacjami**

# Protokół BGP

## System Autonomiczny

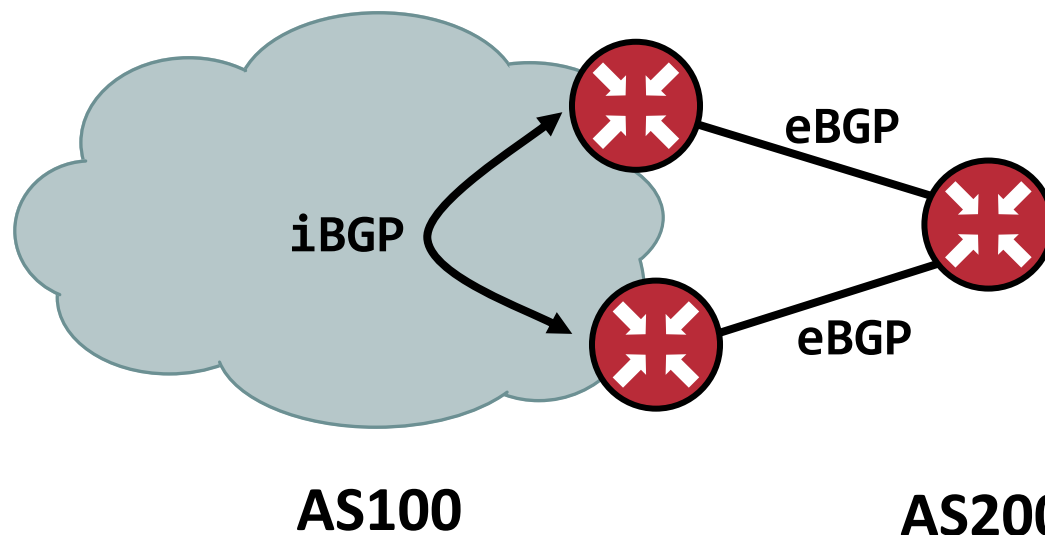


**AS100**

- Zwykle jedna firma/organizacja, zarządzana przez jedną grupę ludzi
- Jedna polityka routingu wewnętrznego i zewnętrznego
- ASN **0** i **65535** zarezerwowane, **23456** również (do przejścia na 32-bitową notację), natomiast **1-64511** zarządzane centralnie (publiczne)
- **64512-65534** do prywatnego użytku
- Obecnie zarejestrowano trochę ponad 39935 ASN, z czego około 21000 jest widocznych w globalnych tablicach routingu

# Protokół BGP

## internal BGP vs external BGP



- **iBGP – sesje pomiędzy routerami w tym samym AS**  
wszystkie routery wewnątrz AS muszą nawiązać sesje każdy z każdym (można to obejść przez konfederacje/klastry)
- **eBGP – sesje pomiędzy routerami w różnych AS**  
domyślnie połączenie bezpośrednie, należy wprost wskazać że połączenie jest multihop

# Protokół BGP

Jak BGP wybiera najlepszą trasę? (wersja skrócona)

- **Najwyższy local preference (w ramach AS)**
- **Najkrótsza ścieżka AS-Path**
- **Najniższy kod pochodzenia (Origin)**
  - IGP < EGP < incomplete
- **Najniższa wartość MED (Multi-Exit Discriminator)**
- **Lepiej trasa z eBGP niż z iBGP**
- **Najpierw trasa z niższym kosztem wg. IGP do next-hop**
- **Najniższy router-id routera BGP**
- **Najniższy adres peer'a**

(pełna lista w RFC4271)



# Protokół BGP

Atrybuty pozwalające wpływać na trasę

**1: ORIGIN**

**2: AS-PATH**

**3: NEXT-HOP**

**4: MED**

**5: LOCAL\_PREF**

**6: ATOMIC\_AGGREGATE**

**7: AGGREGATOR**

**8: COMMUNITY**

**9: ORIGINATOR\_ID**

**10: CLUSTER\_LIST**

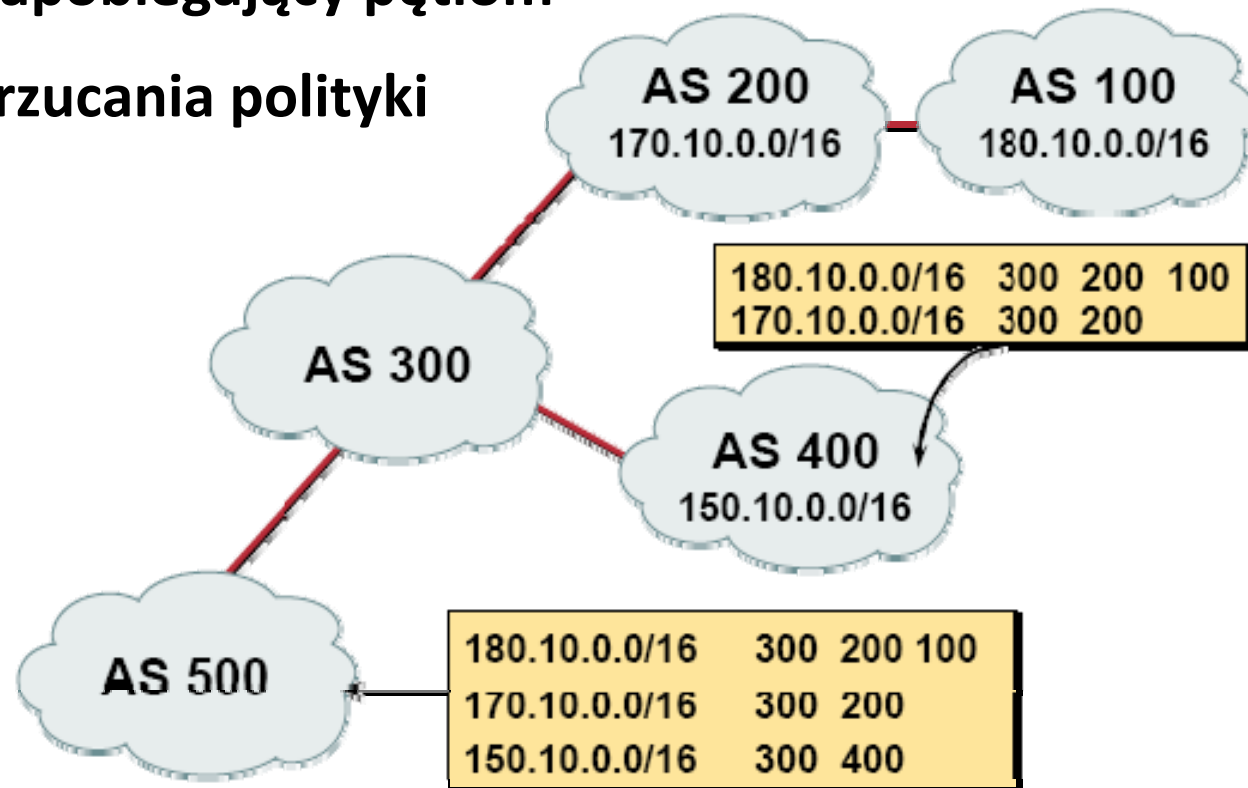
14: MP\_REACH\_NLRI

15: MP\_UNREACH\_NLRI

# Protokół BGP

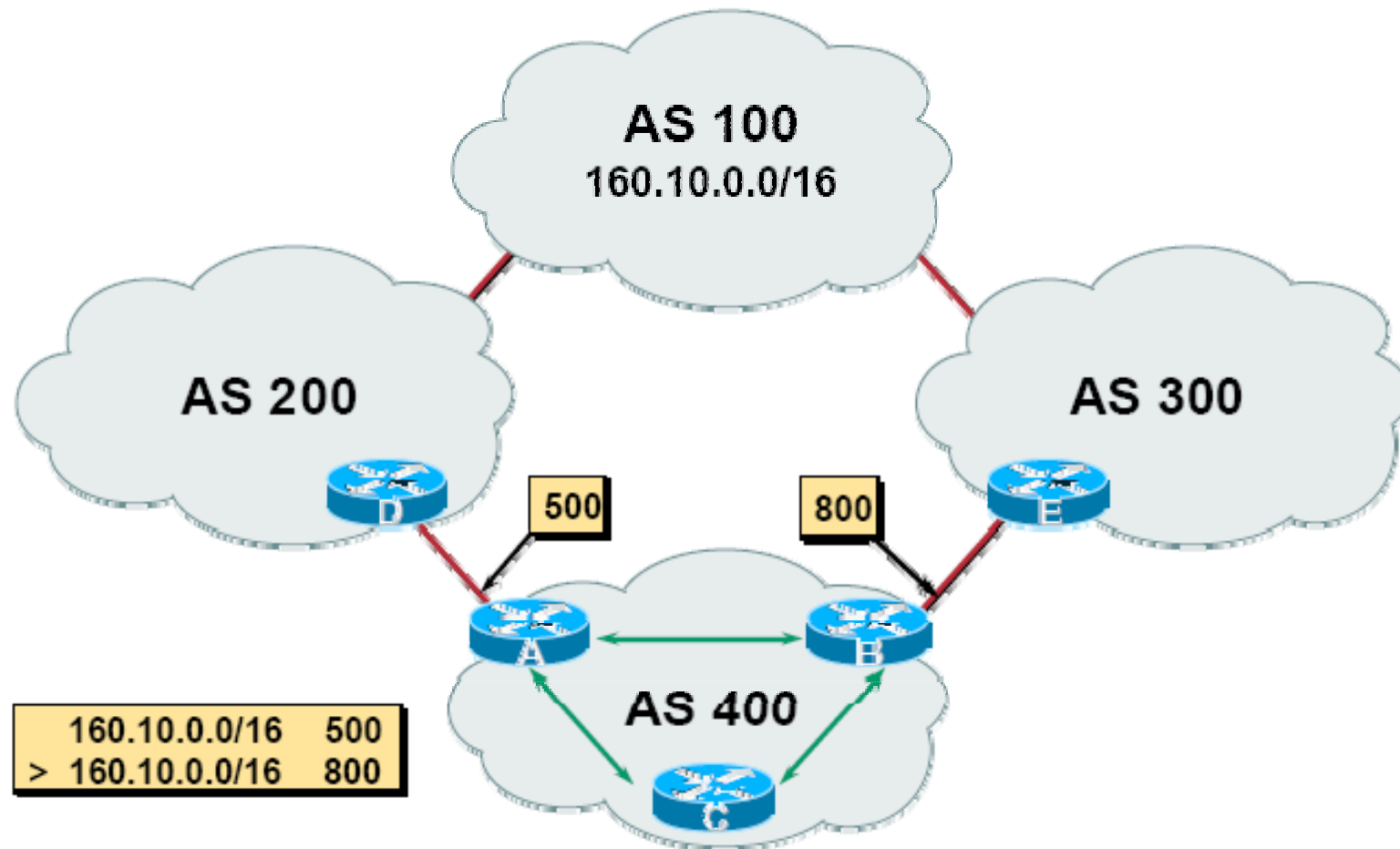
## Atrybuty – AS-Path

- Trasa, jaką prefiks przeszedł
- Mechanizm zapobiegający pętlom
- Narzędzie narzucania polityki



# Protokół BGP

Atrybuty – local preference



# Protokół BGP

## Atrybuty – community

- Opisane w RFC1997, używane między ASami (Community ID) – 32 bitowa wartość
- Dla wygody zapisywane w postaci wartości rozdzielonej kropką. Standardowo <NUM>. Na przykład: 64990
- Bardzo przydatne w konfiguracji publicznych peerów, dotycząca oznaczenia

```
Internet Partners BGP community support
e-mail contact: <bgp4@ipartners.pl>
-----
Communities to control traffic (settable by peers):

8246:2000 Do not announce to GTS CE (AS5588)
8246:2001 Prepend +1 when announcing to GTS CE
8246:2002 Prepend +2 when announcing to GTS CE
8246:2003 Prepend +3 when announcing to GTS CE

8246:2011 Prepend +1 when announcing to T-Systems
8246:2012 Prepend +2 when announcing to T-Systems
8246:2013 Prepend +3 when announcing to T-Systems

8246:2100 Do not announce to TPNET (AS5617)
8246:2101 Prepend +1 when announcing to TPNET
8246:2102 Prepend +2 when announcing to TPNET
8246:2103 Prepend +3 when announcing to TPNET

8246:2200 Do not announce to NASK (AS8308)
8246:2201 Prepend +1 when announcing to NASK
8246:2202 Prepend +2 when announcing to NASK
8246:2203 Prepend +3 when announcing to NASK

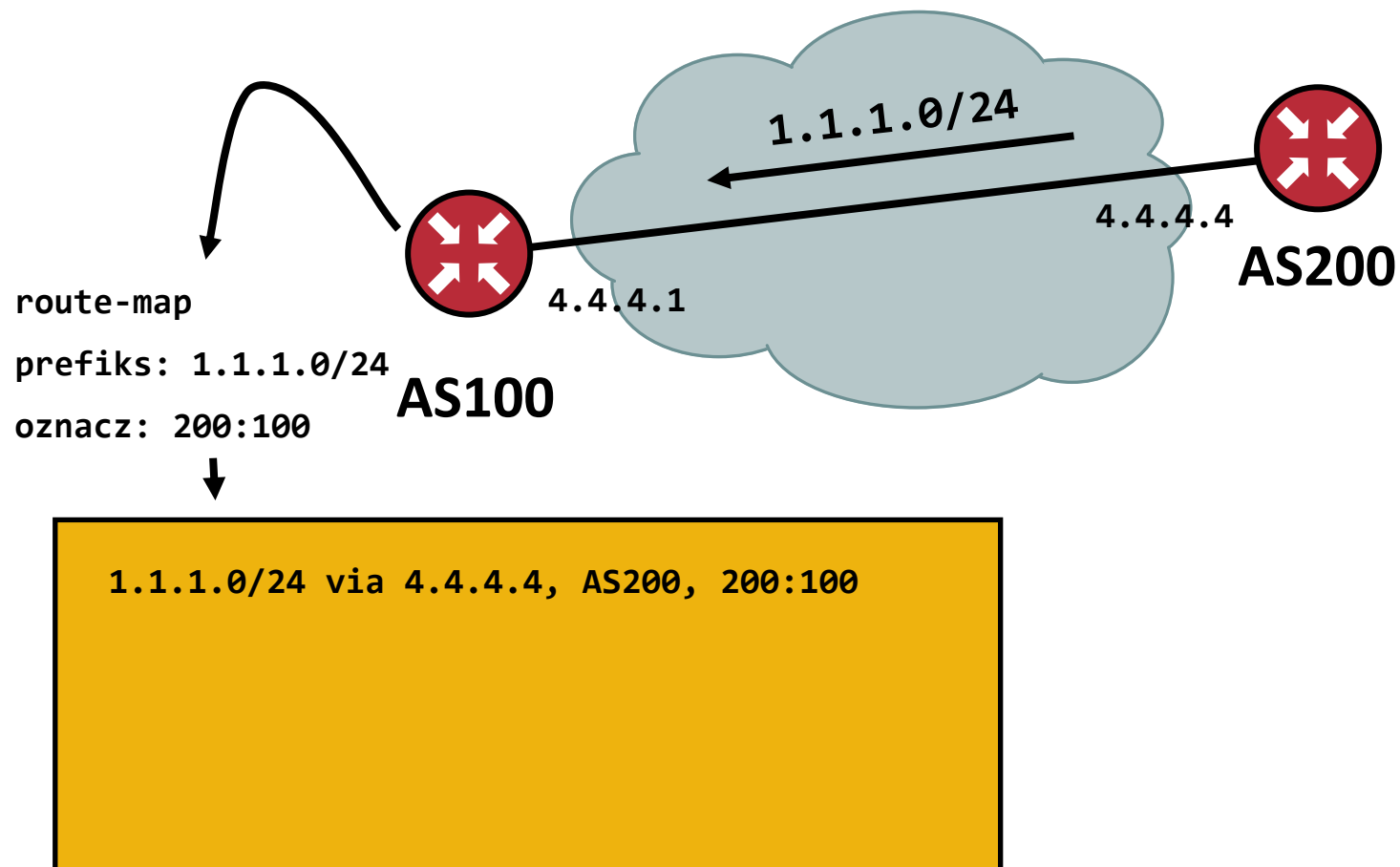
8246:2300 Do not announce to POL34 (AS8501)

8246:2400 Do not announce to Sunsite ICM (AS8664)
8246:2401 Prepend +1 when announcing to Sunsite ICM
8246:2402 Prepend +2 when announcing to Sunsite ICM
8246:2403 Prepend +3 when announcing to Sunsite ICM

8246:2411 Prepend +1 when announcing to WP
8246:2412 Prepend +2 when announcing to WP
8246:2413 Prepend +3 when announcing to WP
```

# Protokół BGP

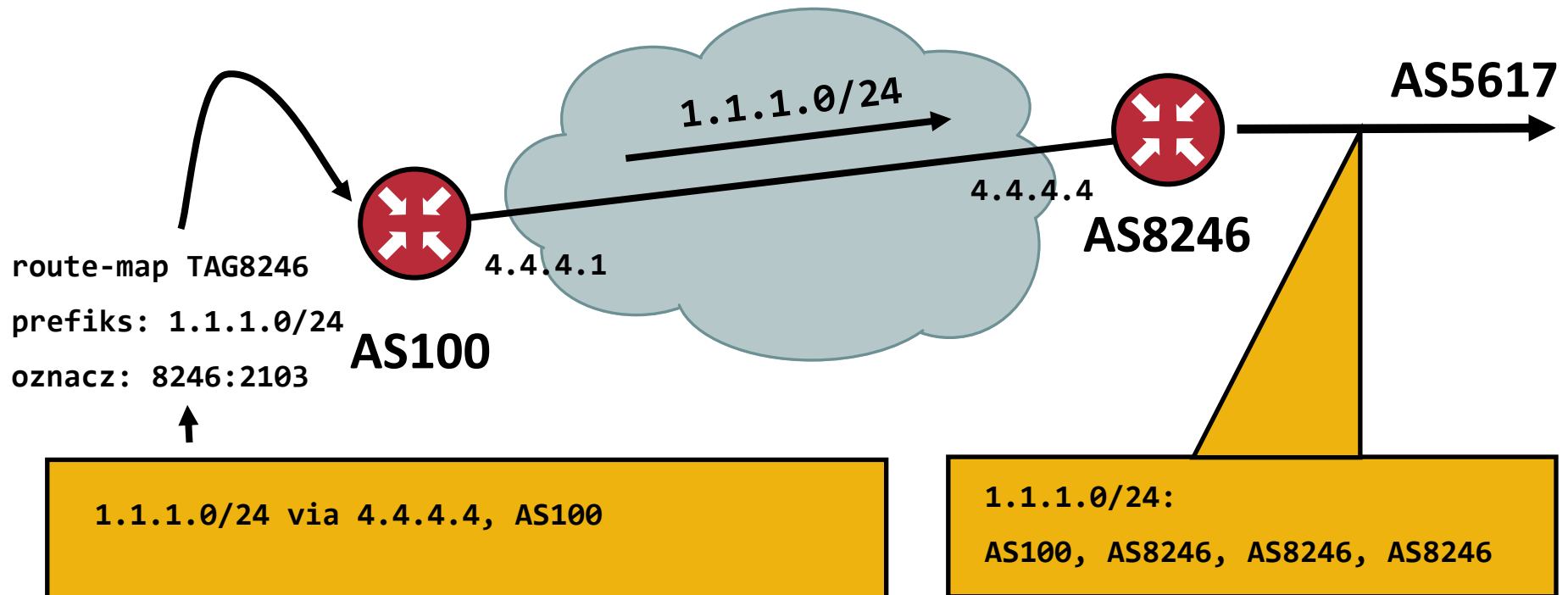
Ilustracja działania – nadawanie community otrzymywanym prefiksom



# Protokół BGP

Ilustracja działania – wysyłanie odpowiednio oznaczonych community

```
8246:2100 Do not announce to TPNET (AS5617)
8246:2101 Prepend +1 when announcing to TPNET
8246:2102 Prepend +2 when announcing to TPNET
8246:2103 Prepend +3 when announcing to TPNET
```

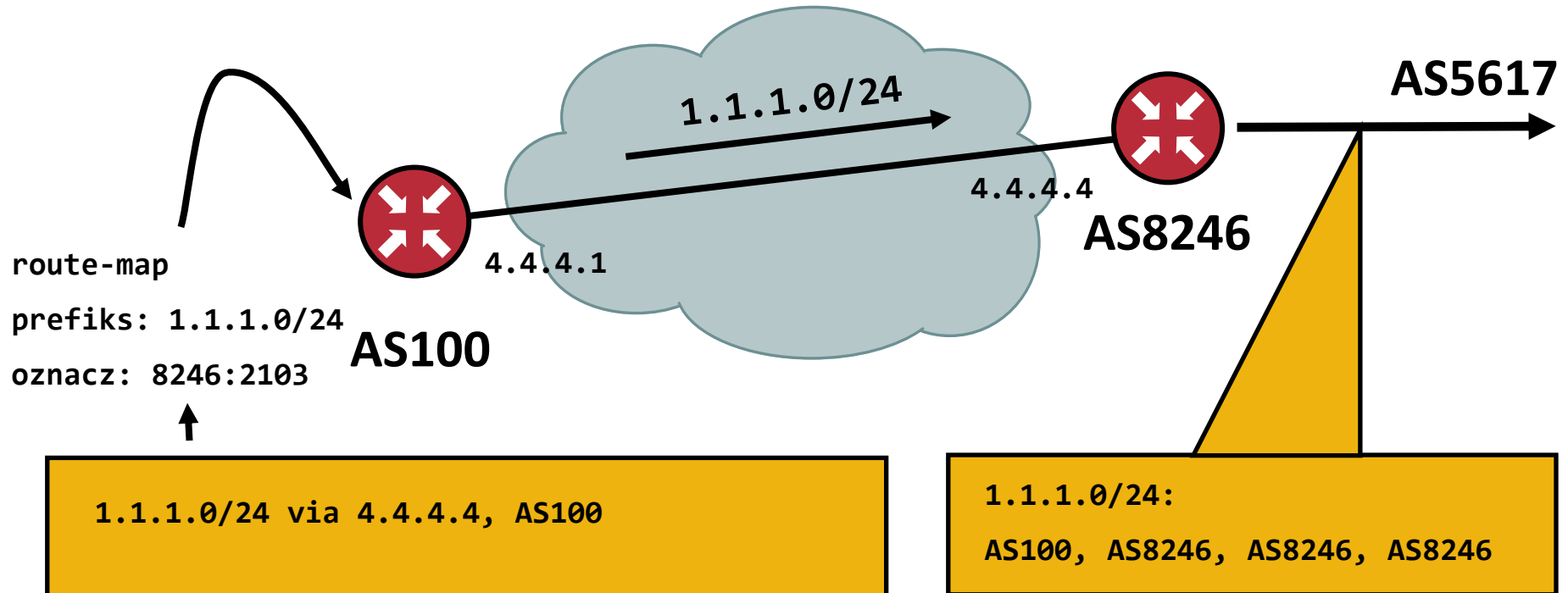


```
router bgp 100
  neighbor 4.4.4.4 route-map 8246 out
```

# Protokół BGP

Ilustracja działania – wysyłanie odpowiednio oznaczonych community

```
8246:2100 Do not announce to TPNET (AS5617)
8246:2101 Prepend +1 when announcing to TPNET
8246:2102 Prepend +2 when announcing to TPNET
8246:2103 Prepend +3 when announcing to TPNET
```



# CZY MUSZĘ UŻYWAĆ BGP?





# Czy na pewno muszę mieć BGP?

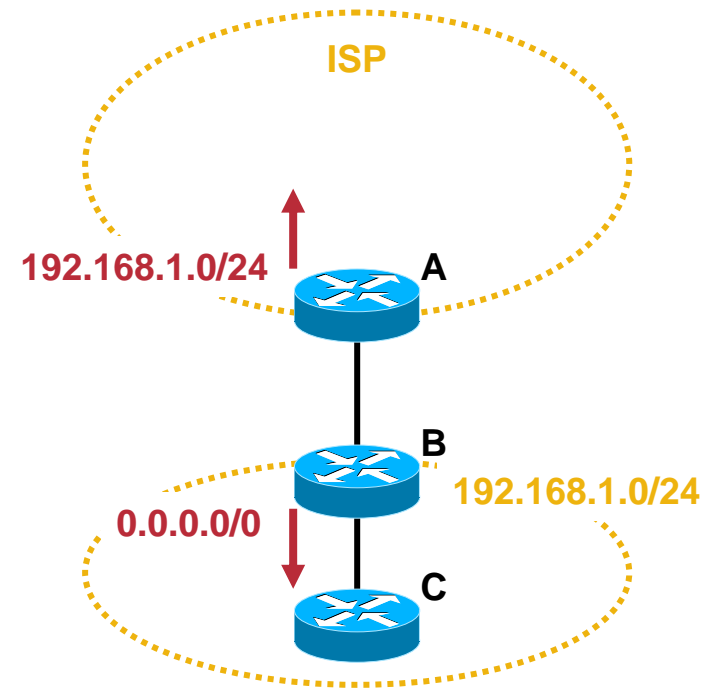
- **Pojedyncze połączenie do Internetu - NIE**

w sieci używasz domyślnej trasy (statycznie lub protokół routingu)

Twój ISP zapewnia widoczność i osiągalność przydzielonej Ci adresacji IP

- **Nawet jeśli łączy są dwa lub trzy do tego samego ISP, BGP nie jest potrzebne**

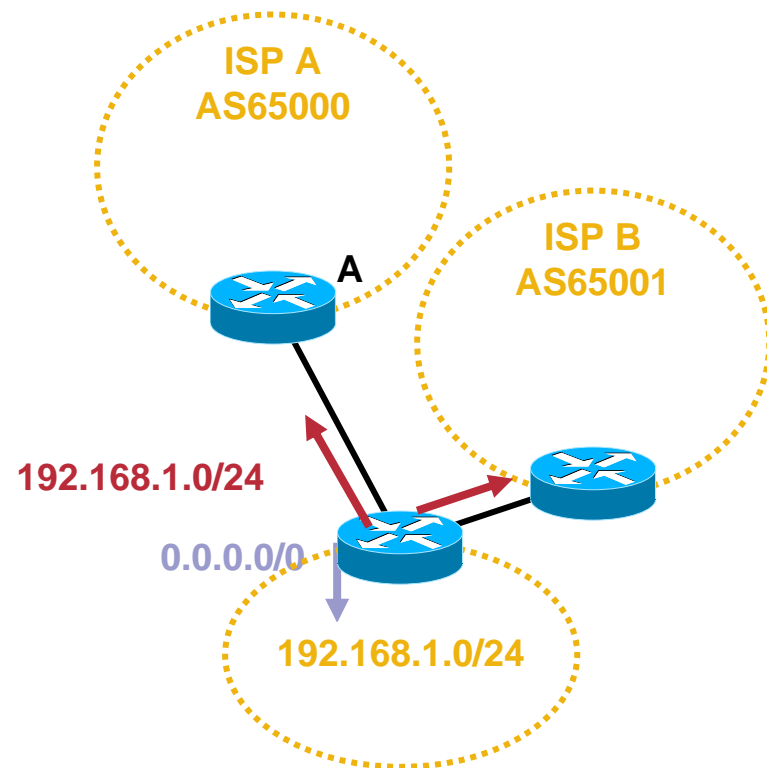
wiele tras domyślnych na różne IP po stronie Twojej i ISP



# Czy na pewno muszę mieć BGP?

- Jeśli jesteś połączony do dwóch różnych ISP – BGP jest pożądane – rozgłasza do obu swoją pulę adresów, Internet widzi ją przez obu ISP
- Nie oznacza to, że musisz pobierać wszystkie światowe prefiksy

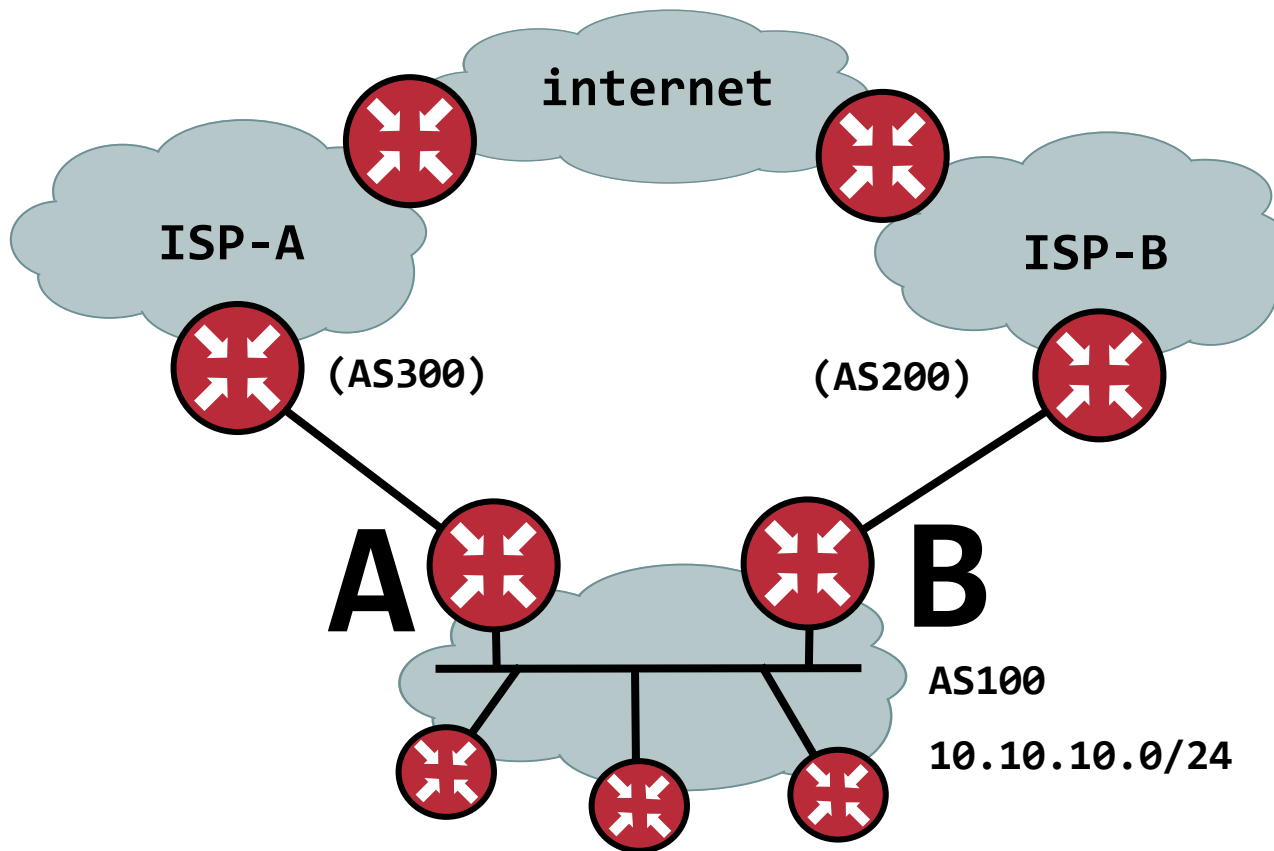
...pozwala to jednak 'świadomie' kształtować własną politykę routingu dużo dokładniej



# PRZYKŁAD ZASTOSOWANIA BGP



...nasza sieć:



# Protokół BGP

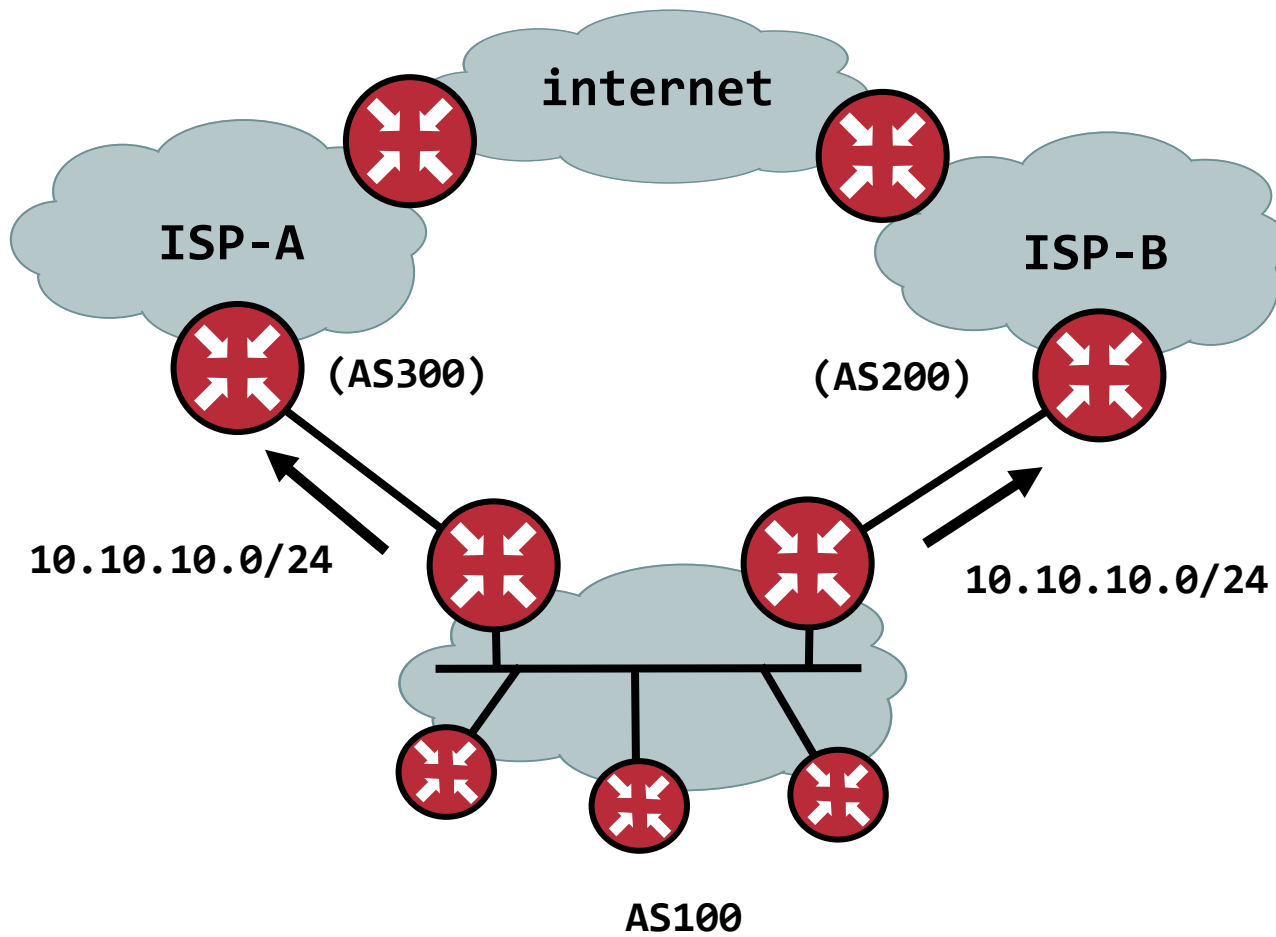
Przykładowa konfiguracja – router A – prefix oraz sesje eBGP

```
router bgp 100
  bgp router-id 10.10.10.253
  network 10.10.10.0 mask 255.255.255.0
  aggregate-address 10.10.10.0 255.255.255.0 summary-only

  neighbor 172.16.10.1 remote-as 200
  neighbor 172.16.10.1 description AS200
  neighbor 172.16.10.1 version 4
  neighbor 172.16.10.1 password AJAX*PERSIL*E

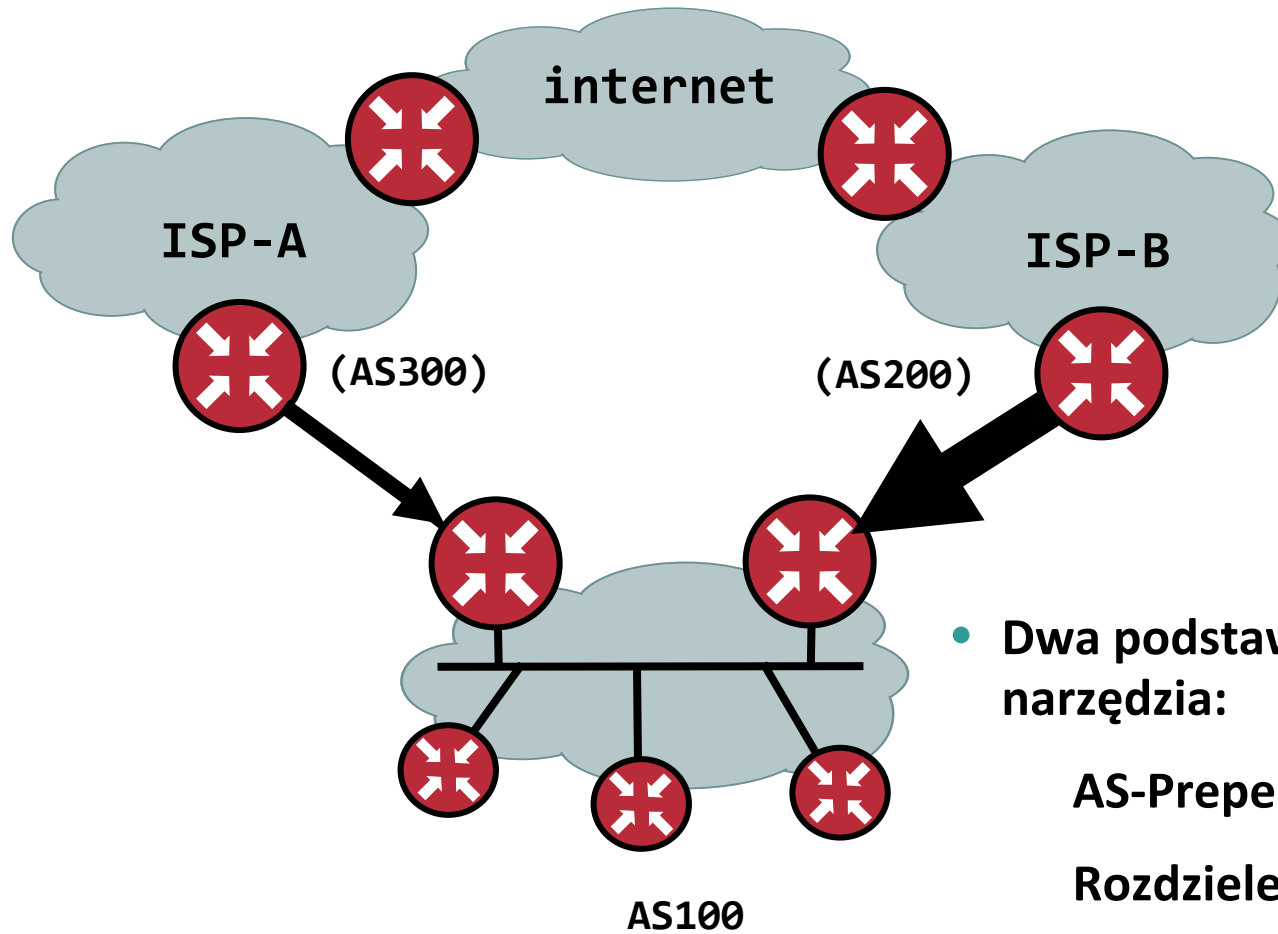
  neighbor 10.10.10.2 remote-as 100
  neighbor 10.10.10.2 description router-B
  neighbor 10.10.10.2 version 4
  neighbor 10.10.10.2 password BEZ*MD5*JEST*ZIA-ZIA
```

...nasza sieć:



# Przykładowy problem:

łącze z ISP-B jest przeciążone



- Dwa podstawowe narzędzia:

AS-Prepend

Rozdzielenie prefiksów

## Przykładowy problem:

łącze z ISP-B jest przeciążone

- „Pogorszenie” atrakcyjności własnego prefiksu 10.10.10.0/24 przez AS200

```
router bgp 100
  neighbor 172.16.10.1 remote-as 200
  neighbor 172.16.10.1 route-map KuLepszejPrzyszlosci out

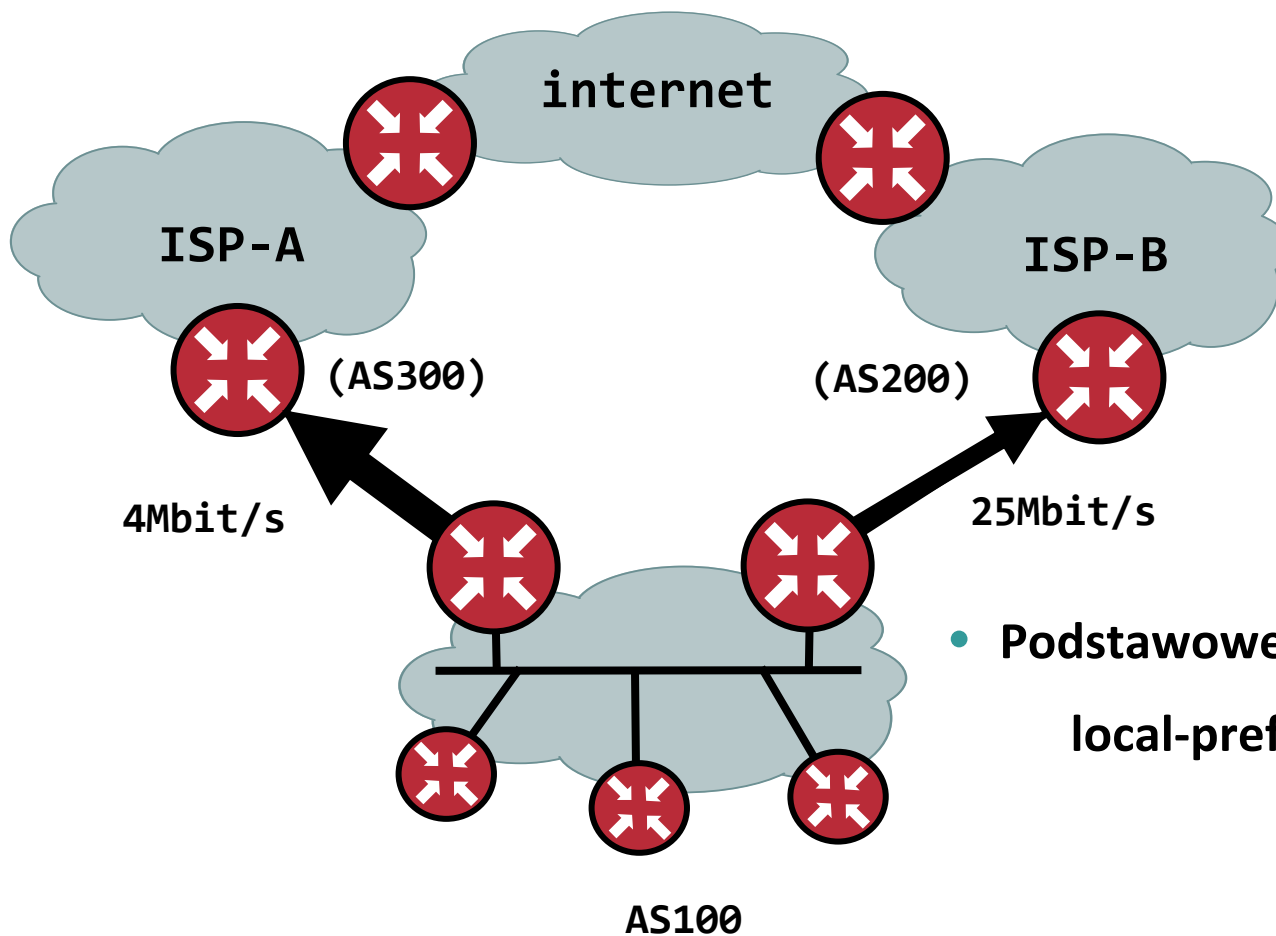
route-map KuLepszejPrzyszlosci permit 10
  set as-path prepend 100 100
```

```
AS200rtr# show ip bgp 10.0.10.0
  Network          Next Hop        LocPrf  Path
* 10.0.10.0/24    172.16.10.2    100     100 100 100 i
*>                172.16.99.1   100     300 100 i
```



# Przykładowy problem:

większość ruchu wychodzi przez ISP-A



- Podstawowe narzędzie:  
local-preference

## Przykładowy problem:

większość ruchu wychodzi przez ISP-A

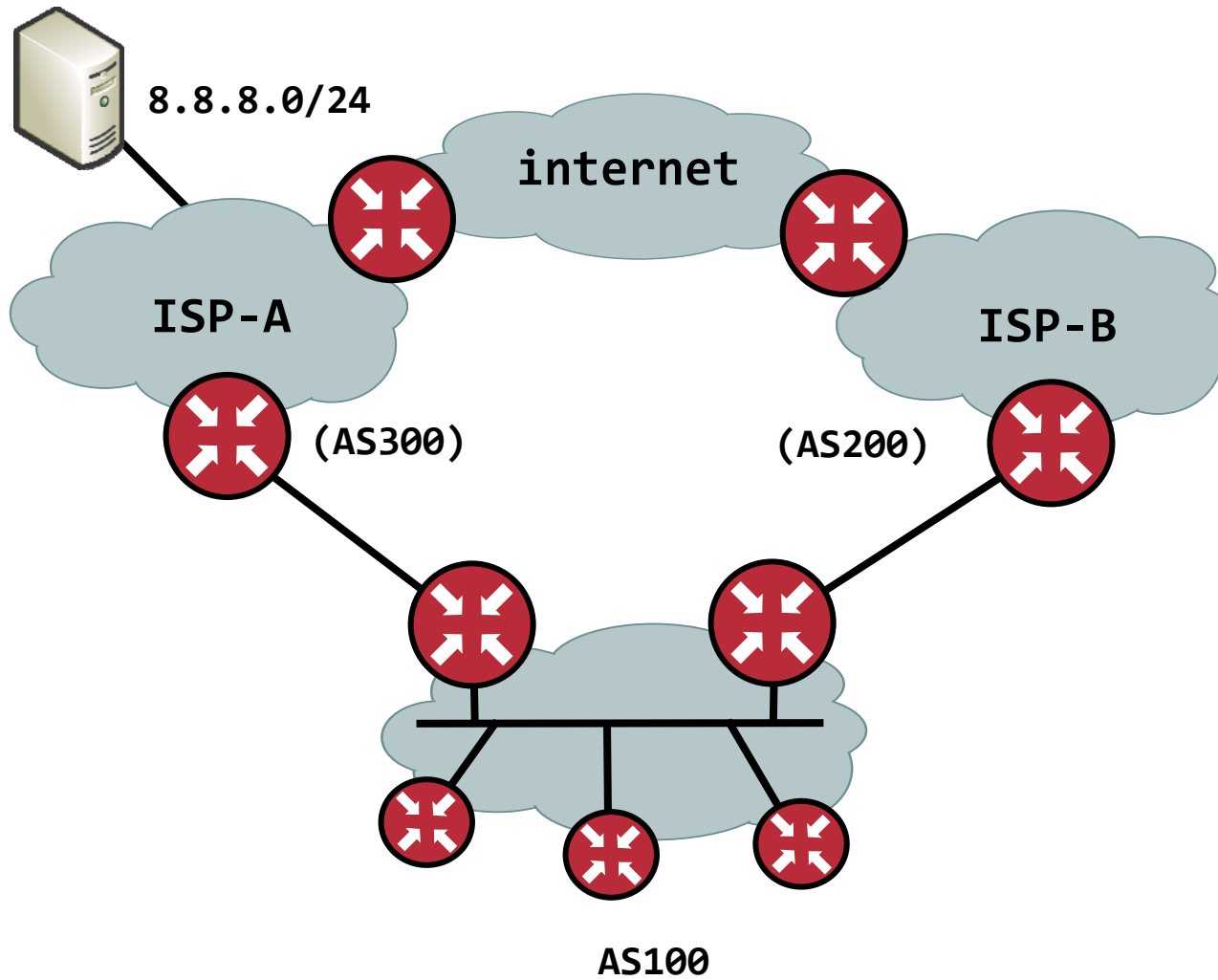
- **Wszystkie** prefiksy otrzymane z AS200 są lepsze niż inne, z domyślnym (100) local-preference

```
router bgp 100
  neighbor 172.16.10.1 remote-as 200
  neighbor 172.16.10.1 route-map KuLepszejPrzyszlosci in

route-map KuLepszejPrzyszlosci permit 10
  set local-preference 5000
```

# Przykładowy problem:

do sieci 8.8.8.0/24 zawsze lepiej przez ISP-A



# Protokół BGP

Prefiks 8.8.8.0/24 w pierwszej kolejności przez ISP-A

- Tylko prefiks **8.8.8.0/24** lub bardziej dokładny jest zawsze lepszy przez AS300

```
router bgp 100
  neighbor 172.16.20.1 remote-as 300
  neighbor 172.16.20.1 route-map KuLepszejPrzyszlosci in

ip prefix-list JedynaDroga permit 8.8.8.0/24 le 32

route-map KuLepszejPrzyszlosci permit 10
  match ip prefix-list JedynaDroga
  set local-preference 5000
```

Dobry opis jak działają prefix-listy:

<http://www.groupstudy.com/archives/ccielab/200404/msg00539.html>

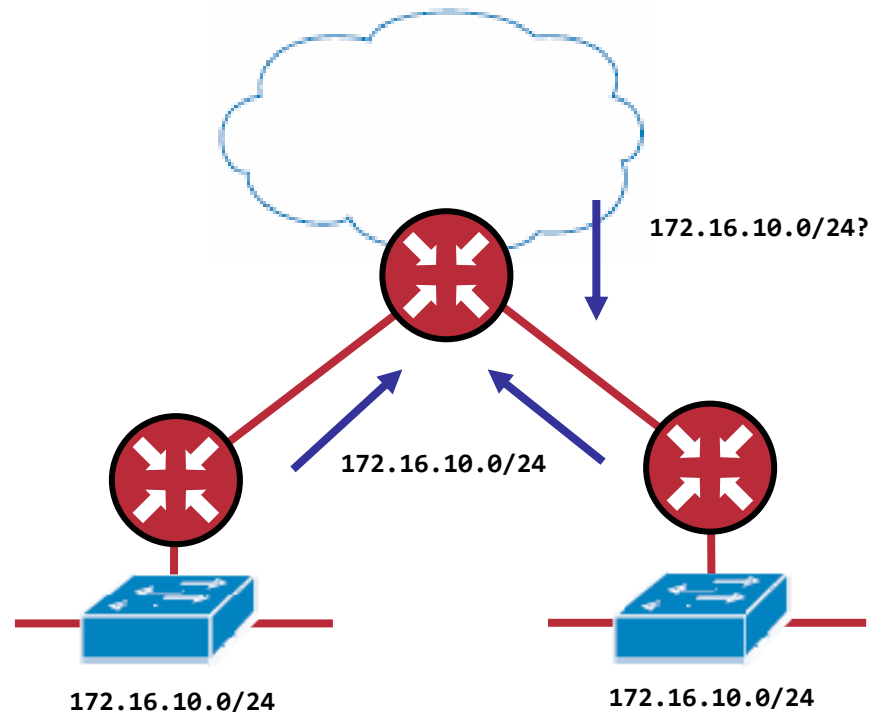
# ROUTING DYNAMICZNY A INNE ZAGADNIENIA



# Routing dynamiczny a inne zagadnienia

## NAT

- W momencie, w którym routery robią 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**

**za pomocą wskazania wprost sąsiada można dodatkowo wysyłać pakiety unicast (przydatne przy tunelach IPsec)**

**OSPF – multicast, protokół IP numer 89**

**BGP – protokół TCP na/z portu 179**



# Routing dynamiczny a inne zagadnienia

## Tunelowanie IP-w-IP i GRE

- Tunele GRE pozwalają 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 jako punkt-punkt, ale tylko te, które istnieją zanim zostanie uruchomiona

warto sprawdzić, czy interfejs jest widziany jako posiadający flagę **multicast** – różnie dla różnych kerneli i wersji pakietu

# Routing dynamiczny a inne zagadnienia

## QoS

- Mechanizmy/polityka QoS powinna **w szczególności** dotyczyć protokołów routingu
- Priorytet dla pakietów **hello**, oraz zapewniających funkcjonowanie protokołów routingu powinien być pierwszym składnikiem polityki (przed gwarancjami dla VoIP itp.)
- Zastosowanie mechanizmów nawet, jeśli nie ma potrzeby ich stosowania daje ochronę w trakcie ataków (D)DoS oraz np. infekcji wirusami/trojanami

# Routing dynamiczny a inne zagadnienia

## MPLS

- **Linux posiada już infrastrukturę wielu tabel routingu**  
iproute2 etc.
- **Implementacja architektury MPLS wymaga:**
  - wyposażenia infrastruktury sterowników sieciowych w mechanizmy odpowiedniej hermetyzacji tagów MPLS (Ethernet/FR/ATM)
  - stworzenia demona odpowiedzialnego za sygnalizację (LDP)
  - stworzenia mechanizmu dla obsługi inżynierii ruchu (RSVP lub OSPF-TE)
- **Testowane:**
  - <http://sourceforge.net/projects/mpls-linux/>
  - <http://www.cs.virginia.edu/~mngroup/projects/mpls/>

# First-Hop Redundancy

- **Standardy: VRRP**
- **Cisco-made: HSRP i GLBP**
- **Inne: CARP**
- **Idea: podstawić **jeden** wirtualny router w miejsce wielu fizycznych i zapewnić obsługę routingu**  
**wirtualny IP i wirtualny MAC**

# First-Hop Redundancy

- VRRP – standard (RFC3768)

VRRPd: <http://off.net/~jme/vrrpd/>

FreeVRRPd

- CARP/UCARP – analogiczne dla OpenBSD/innych systemów

<http://www.ucarp.org/project/ucarp>

- Integracja aplikacji i systemu operacyjnego:

LVS: <http://www.linuxvirtualserver.org/>

ct\_sync (netfilter) + keepalived + LVS/coś innego:

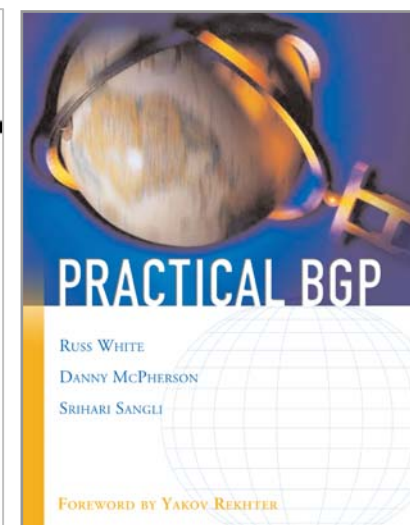
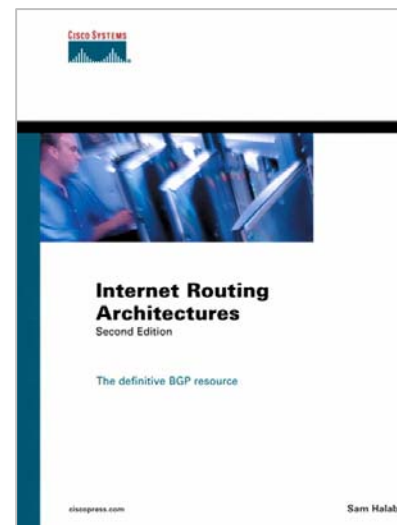
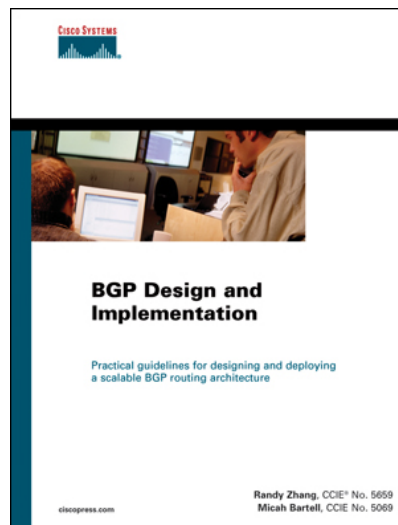
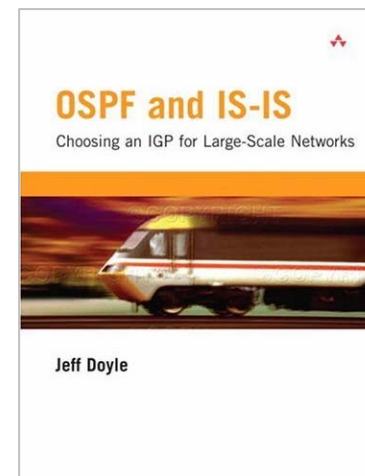
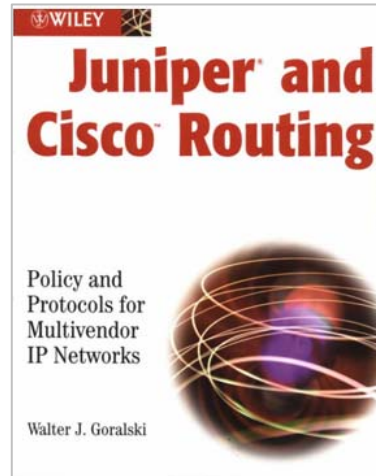
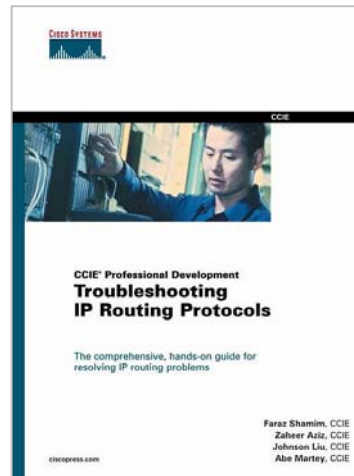
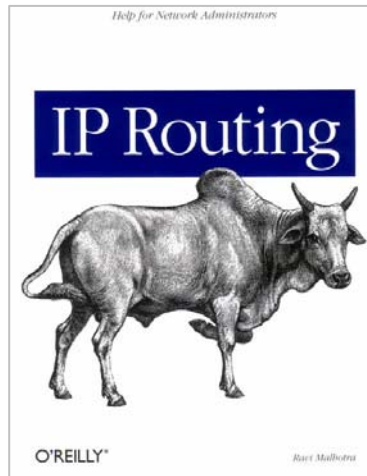
<http://svn.netfilter.org/cgi-bin/viewcvs.cgi/trunk/netfilter-ha/>

[http://www.netfilter.org/projects/libnetfilter\\_contrack/index.html](http://www.netfilter.org/projects/libnetfilter_contrack/index.html)

# GDZIE WARTO RZUCIĆ OKIEM



# Książki



## Zasoby WWW

- **Test routerów opartych o Quagę/XORP**

<http://www.networkworld.com/reviews/2006/100906-quagga-router-test.html>

- **Pakiet Quagga:**

<http://www.quagga.net>

- **Pakiet XORP:**

<http://www.xorp.org>

- **Demon OpenOSPFd/OpenBGPd:**

<http://www.openbsd.org>



## Zasoby WWW

- Einar – LiveCD wykorzystujące Xen i Quagę:

<http://www.isk.kth.se/proj/einar/>

- IP routing protocols home page (Cisco):

[http://www.cisco.com/en/US/tech/tk365/tsd\\_technology\\_support\\_protocol\\_home.html](http://www.cisco.com/en/US/tech/tk365/tsd_technology_support_protocol_home.html)

- OSPF MANET:

<http://folk.uio.no/kenneho/index.php?page=studies&subpage=wospf>

# Q&A

