SIECI KOMPUTEROWE

wykład dla kierunku informatyka

semestr 4 i 5

dr inż. Michał Sajkowski

Instytut Informatyki PP

pok. 227G PON PAN, Wieniawskiego 17/19

Michal.Sajkowski@cs.put.poznan.pl

tel. +48 (61) 8 582 100

http://www.man.poznan.pl/~michal/

sieci komputerowe
wykład 4
współdziałanie sieci

literatura podstawowa

wykład prawie w całości przygotowany na podstawie tekstu i rysunków

z rozdziału 5 w książce:

L.L. Peterson, B.S. Davie

„Sieci komputerowe. Podejście systemowe”

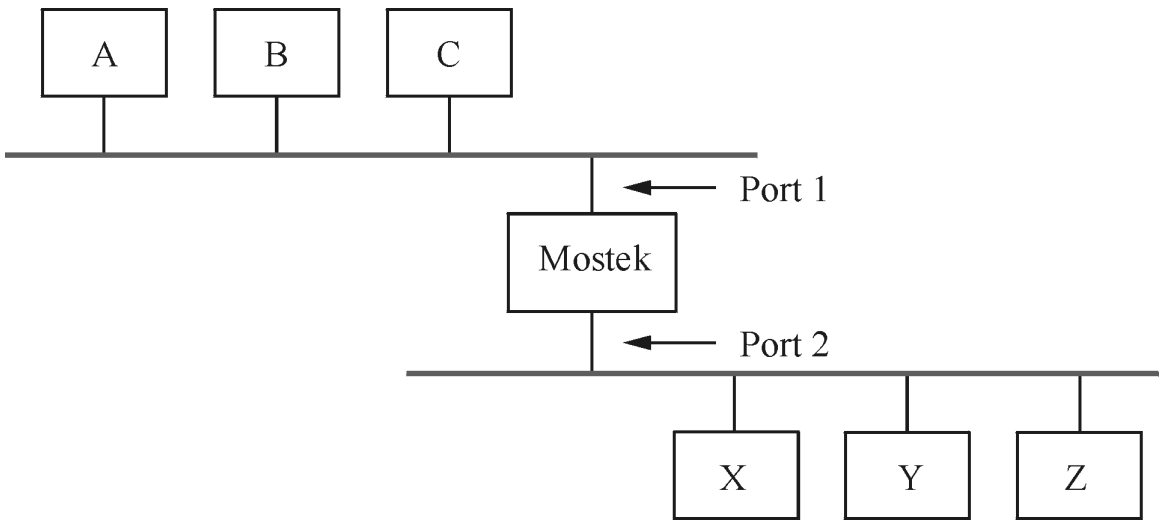
Wydawnictwo Nakom, Poznań 2000

problemy

- *wzajemne łączenie różnych sieci*
- *heterogeniczność*: użytkownicy sieci jednego typu mogą porozumiewać się z użytkownikami sieci innych typów
- *skalowalność*: łatwe powiększanie sieci do dowolnego rozmiaru
- *problemy szczegółowe*: rozszerzone sieci lokalne, protokół intersieci, globalna intersieć, protokół IP następnej generacji, rozsyłanie grupowe, nazwy komputerów

mostki i rozszerzone sieci lokalne

- połączenie dwóch sieci Ethernet: *wzmacniakiem* (maks. 2 wzmacniaki i 2500 m między komputerami)
- połączenie dwóch sieci Ethernet: *mostkiem*, który przekazuje ramki na podstawie nagłówka w warstwie łącza danych DLL, działa w LLC albo w MAC
- *mostek nieinteligentny* (przekazuje wszystkie ramki)

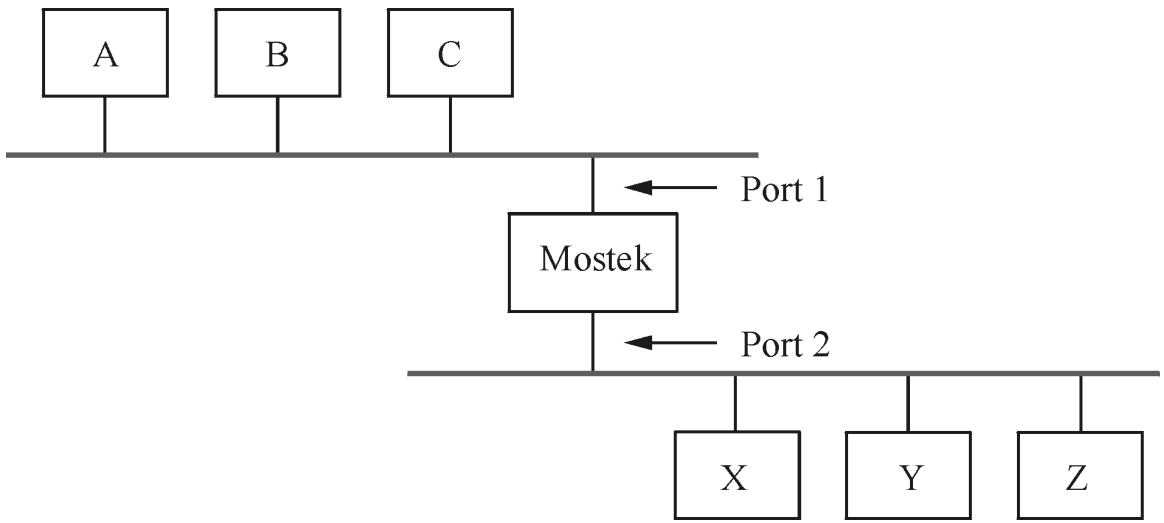


mostek inteligentny

- przekazuje tylko wybrane ramki, zgodnie z tablicą kierującą utrzymywaną w mostku:

komputer|port

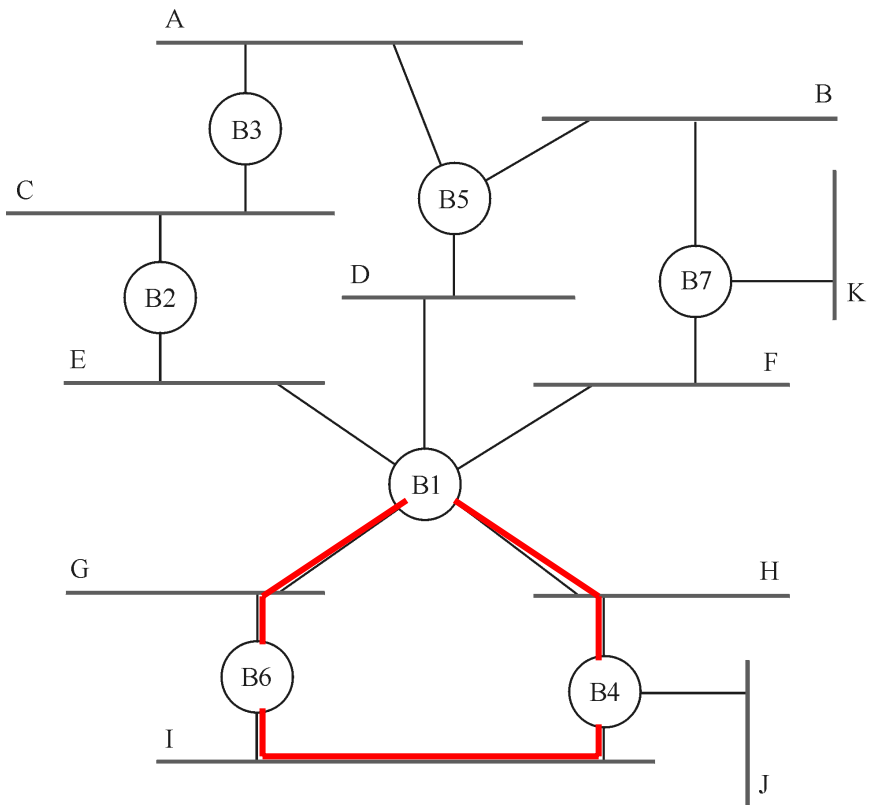
A	1
B	1
C	1
X	2
Y	2
Z	2



gdy mostek w porcie 1 *odbiera ramkę do* A *to nie przekazuje jej do portu* 2, *a gdy odbiera ramkę do* A *w porcie* 2 *to przekazuje do portu* 1

rozszerzona sieć lokalna z pętlami

- poprzednia strategia działa, dopóki nie wystąpi *pętla*:

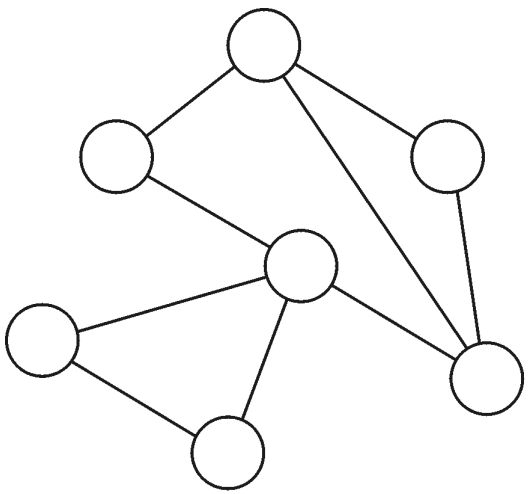


wybór portu przez mostek

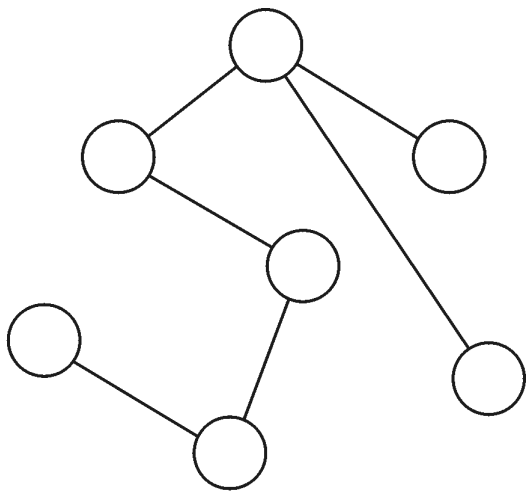
- niezależnie od przyczyn, prowadzących do pętli, *mostki muszą radzić sobie z pętlami*
- problem rozwiązano za pomocą uruchomienia w mostkach (rozproszonego) *algorytmu drzewa rozpinającego* - *drzewo rozpinające* jest podgrafem grafu cyklicznego zawierającego wszystkie wierzchołki, ale żadnego cyklu
- *protokół* stosowany przez zbiór mostków do uzgodnienia drzewa rozpinającego dla rozszerzonej sieci lokalnej - Radia Perlman (DEC, 1985)
- *mostek decyduje, na których portach przekazuje ramki*, korzystając z algorytmu drzewa rozpinającego

drzewo rozpinające grafu cyklicznego

graf cykliczny



odpowiadające mu drzewo rozpinające



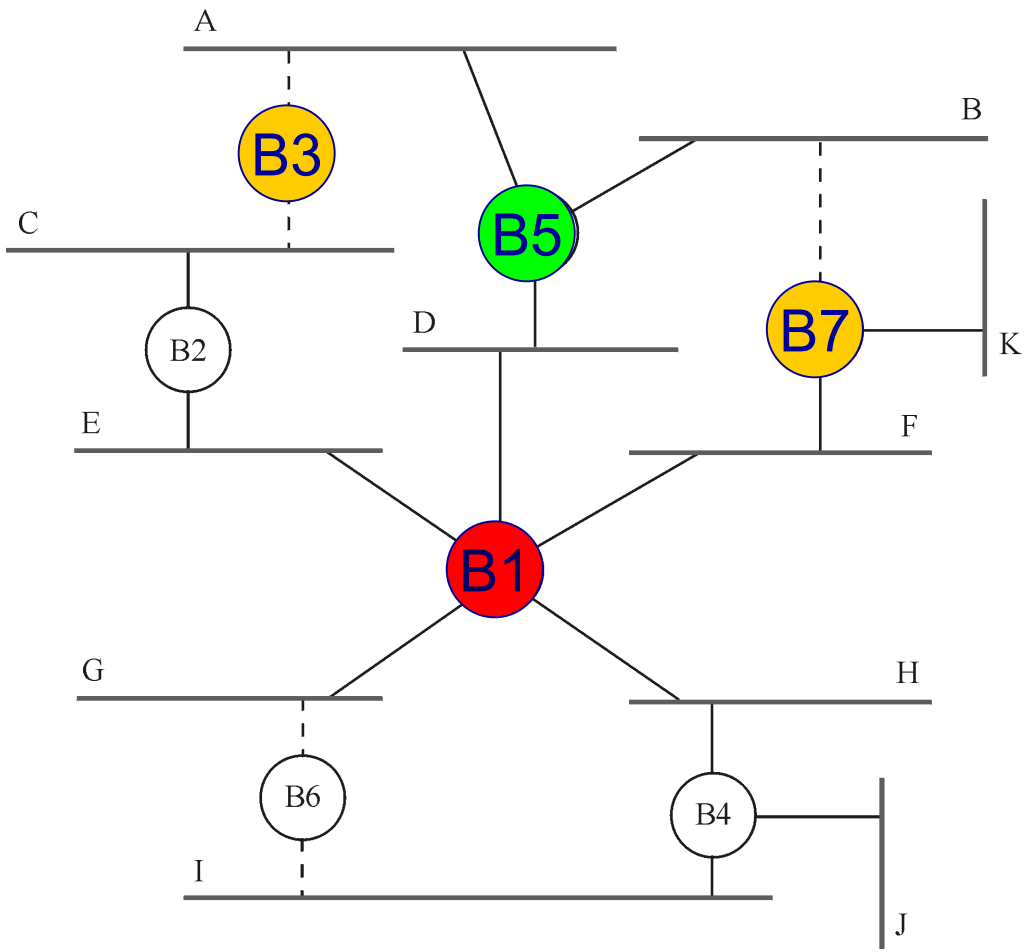
działanie algorytmu drzewa rozpinającego

każdy mostek ma unikalny identyfikator, n.p. B1, B2...

1. *wybierz* mostek z najniższym identyfikatorem na *korzeń drzewa rozpinającego* (oddzielną procedurą)
2. *dla każdego mostka oblicz najkrótszą trasę do korzenia* drzewa i zapisz porty na trasie
3. ze zbioru mostków dołączonych do danej sieci lokalnej *wybierz mostek desygnowany*, odpowiedzialny za przesyłanie ramek do mostka będącego korzeniem drzewa (*desygnowany* to mostek najbliższy korzeniowi, gdy dwa lub więcej mostki są w równej odległości to wygrywa mostek z mniejszym identyfikatorem)
4. *mostek przekazuje ramki przez porty, dla których jest mostkiem desygnowanym*

drzewo rozpinające dla rozszerzonej sieci lokalnej

- B1 jest korzeniem
- B5 jest *mostkiem desygnowanym* dla sieci A, bo bliżej niż B3
- B5 jest mostkiem desygnowanym dla sieci B, gdyż B5 ma mniejszy identyfika-tor od B7



procedura wyboru mostka na korzeń drzewa rozpinającego

- wymiana między mostkami *komunikatów z konfiguracją*: {identyfikator mostka nadającego, identyfikator mostka uznawanego przez mostek nadający za korzeń drzewa, odległość od mostka nadającego do korzenia}
- mostek zapisuje *najlepszy* komunikat z konfiguracją
- komunikat z konfiguracją *jest lepszy, gdy*: identyfikuje korzeń z mniejszym id, identyfikuje korzeń z takim samym id ale mniejszą odległością, id korzenia i odległość są takie same, ale mostek nadający ma mniejszy id
- *zdolność do rekonfiguracji* drzewa rozpinającego po awarii

procedura wyboru mostka na korzeń drzewa rozpinającego

1. *mostek zakłada, że jest korzeniem*, wysyła komunikat do każdego swojego portu z odległością 0
2. po odbiorze komunikatu przez port, *mostek sprawdza czy nowy komunikat lepszy od starego*, jeżeli tak to go zapisuje i dodaje 1 do odległości od korzenia
3. kiedy mostek odbiera komunikat, wskazujący że *on sam nie jest korzeniem*, odbiera tylko komunikaty od innych mostków, zwiększając o 1 pole odległości
4. kiedy mostek odbiera komunikat, wskazujący że *on sam nie jest mostkiem desygnowanym dla danego portu*, mostek kończy nadawanie komunikatów przez ten port

ograniczenia rozwiązań dla rozszerzonych sieci lokalnych

- *skalowalność*: ograniczenie do kilkudziesięciu sieci
- *heterogeniczność*: mostki między sieciami o takim samym formacie adresów (48 bitów): Ethernet - Ethernet, FDDI - FDDI, Ethernet - FDDI
- *przezroczystość mostków* (nieświadomość w wyższych warstwach istnienia mostków w warstwie drugiej)
- *wprowadzenie mostków zmienia charakter sieci*: sieć może tracić ramki, zmieniać ich kolejność, wprowadzać znaczne opóźnienia

proste współdziałanie sieci (IP)

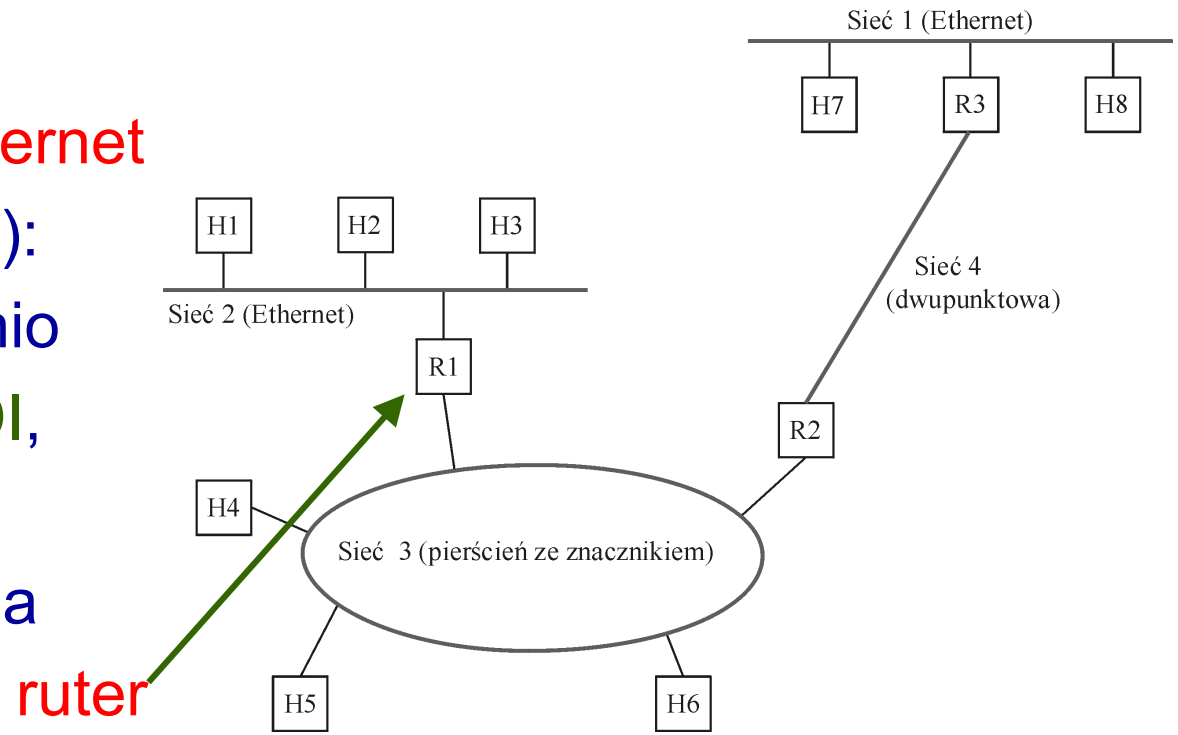
- usunięcie ograniczeń nakładanych przez mostki
- budowa dużych heterogenicznych sieci
- funkcje protokołu intersieci (IP)
- rozszerzenie skalowalności Internetu
- IP następnej generacji (IPv6)

co to jest intersieć?

- *dowolny zbiór sieci połączonych ze sobą* dla zapewnienia określonego rodzaju dostawy pakietów między komputerami (*sieć sieci*, *sieć logiczna*)
- globalna sieć światowa: Internet
- sieć (*fizyczna*):

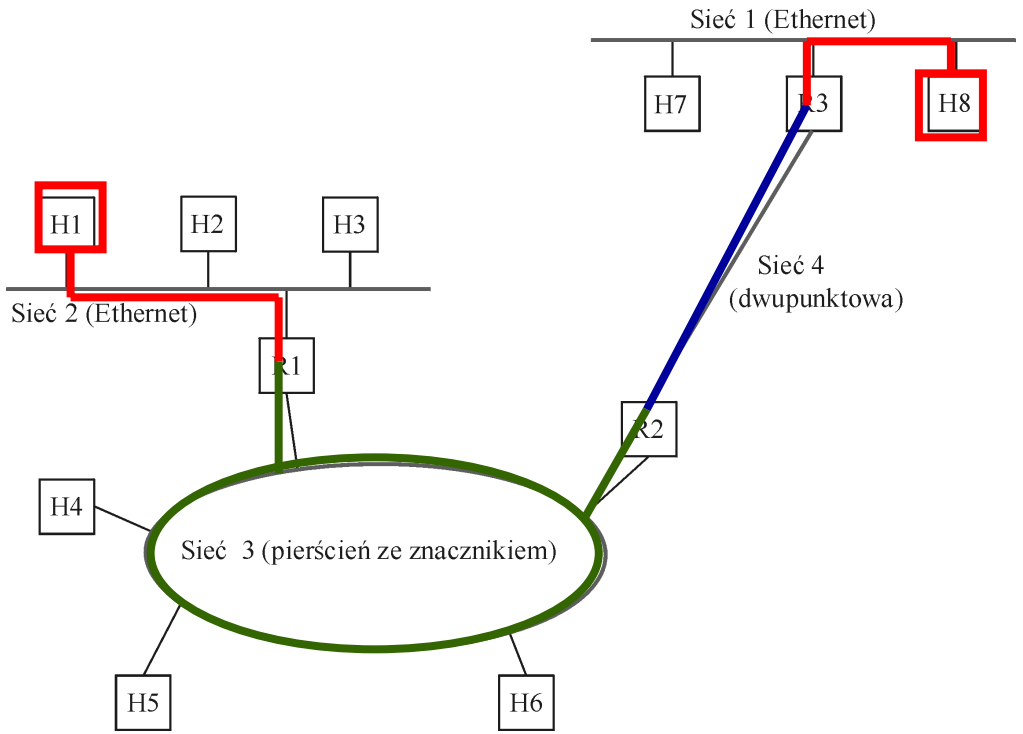
sieć bezpośrednio połączona (FDDI, Ethernet),

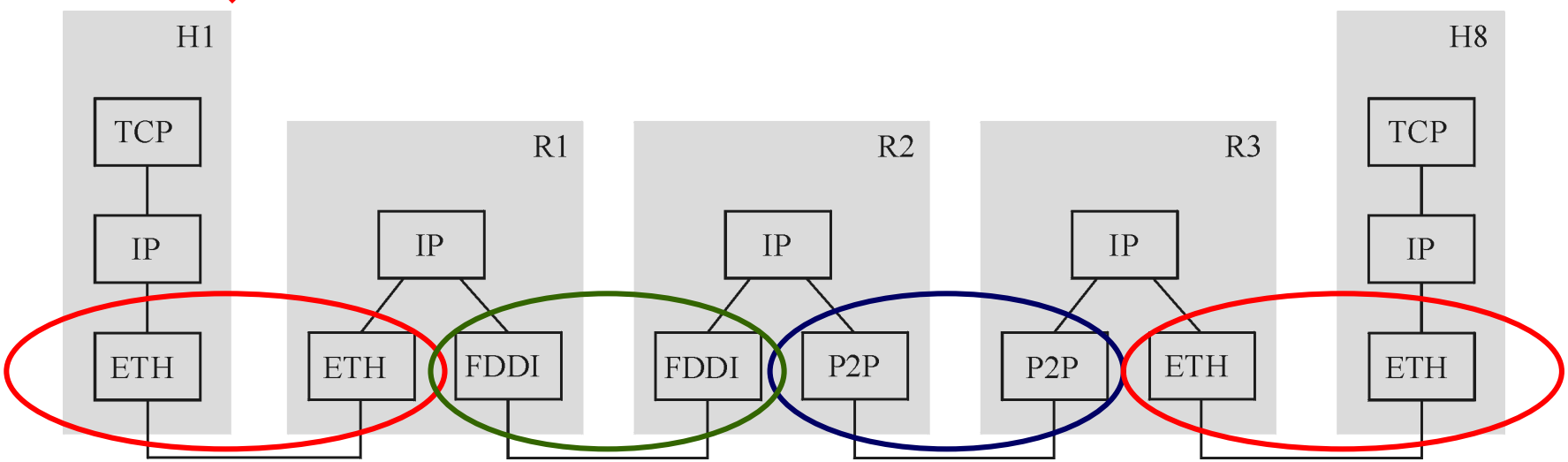
sieć komutowana (ATM)



prosta intersieć

- *topologia sieci* ➔
- *architektura warstwowa*



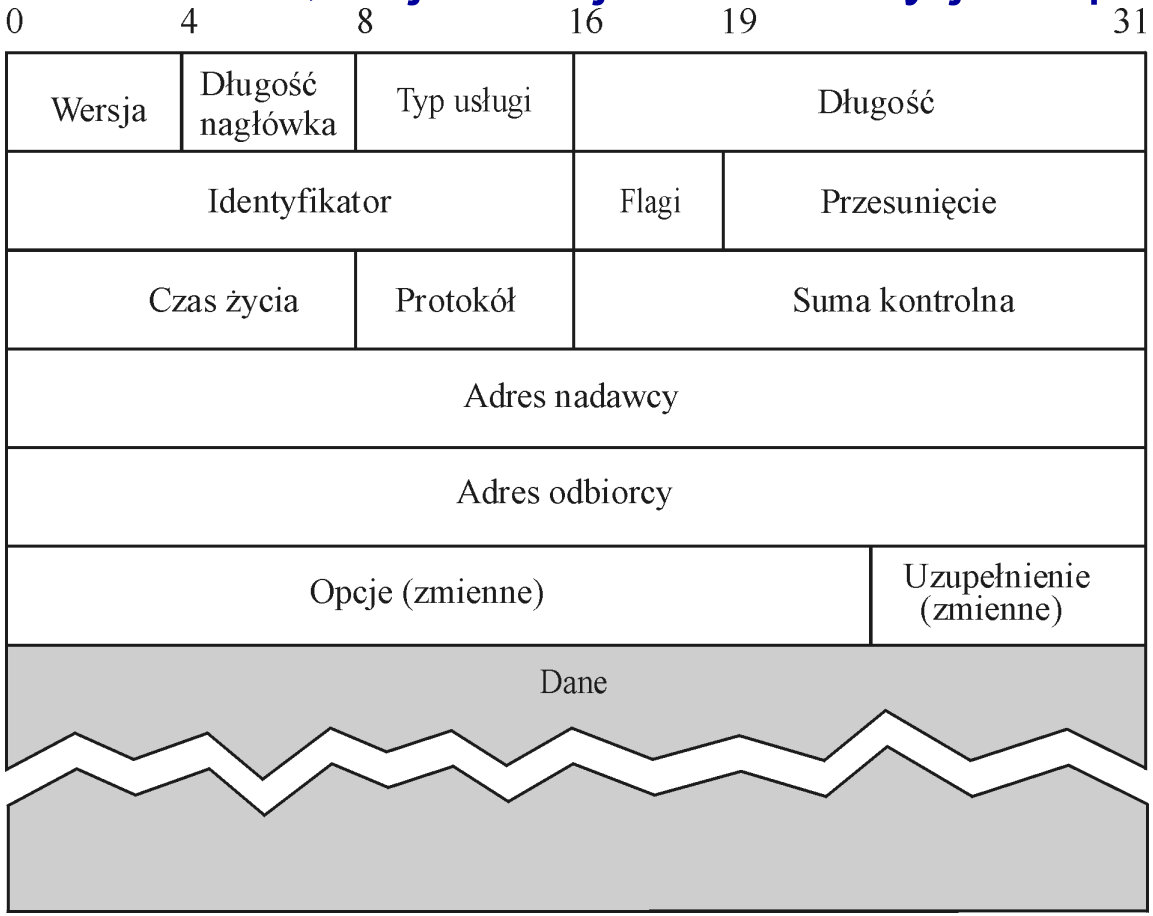


model usługi

- model usługi IP = schemat adresacji + model datagramowy dostawy danych:
- *model dostępnych możliwości* (IP robi co może, aby dostarczyć datagram, ale nie daje gwarancji)
- oznacza to, że jak coś idzie źle, to sieć nie zrobi nic, gdyż zrobiła już co mogła - *zawodna usługa*
- protokoły z wyższych warstw muszą być tego świadome

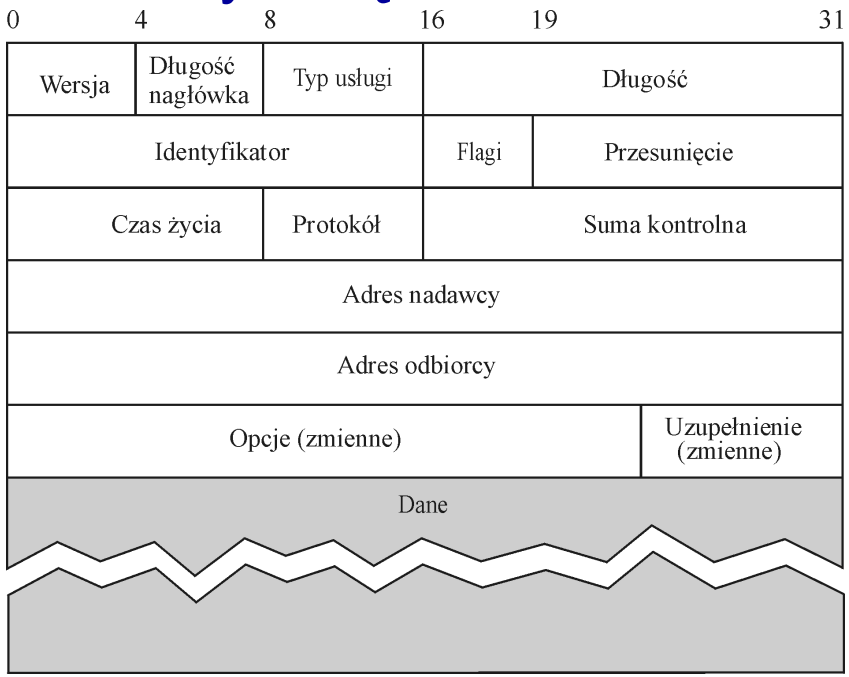
nagłówek datagramu IP (IPv4)

- słowa 32 bitowe, bajt wersji nadawany jako pierwszy



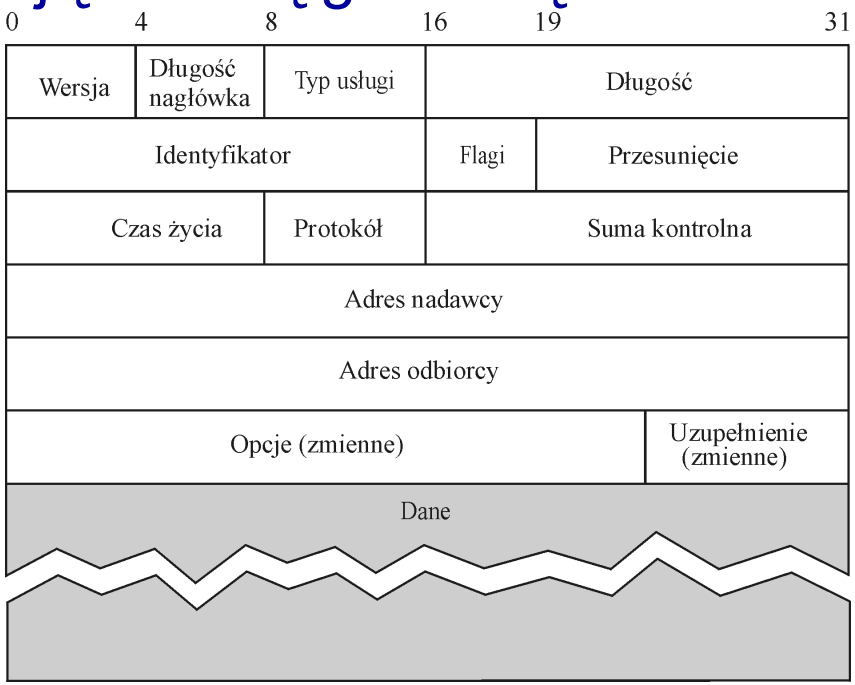
nagłówek datagramu IP (IPv4)

- *pole wersji*, n.p. IPv4
- *długość nagłówka* = 5 słów = 20 bajtów (bez opcji)
- *typ usługi*: 3 bity na priorytet, 3 bity na: małe opóźnienie, dużą przepustowość, wysoką niezawodność, oraz fragmentacja
- *długość datagramu* w bajtach z nagł. wł. maks.65535B
- czas życia (pakietu) =64
- protokół=klucz demultiplek.
- suma kontrolna (z nagł. wł.)



adres IP nadawcy i odbiorcy

- 32 bity adres nadawcy, 32 bity adres odbiorcy
- *adres odbiorcy kluczowy dla datagramu*
- *protokół* IP definiuje swoją własną globalną przestrzeń adresową, niezależną od tego, na jakich sieciach fizycznych uruchomiono IP

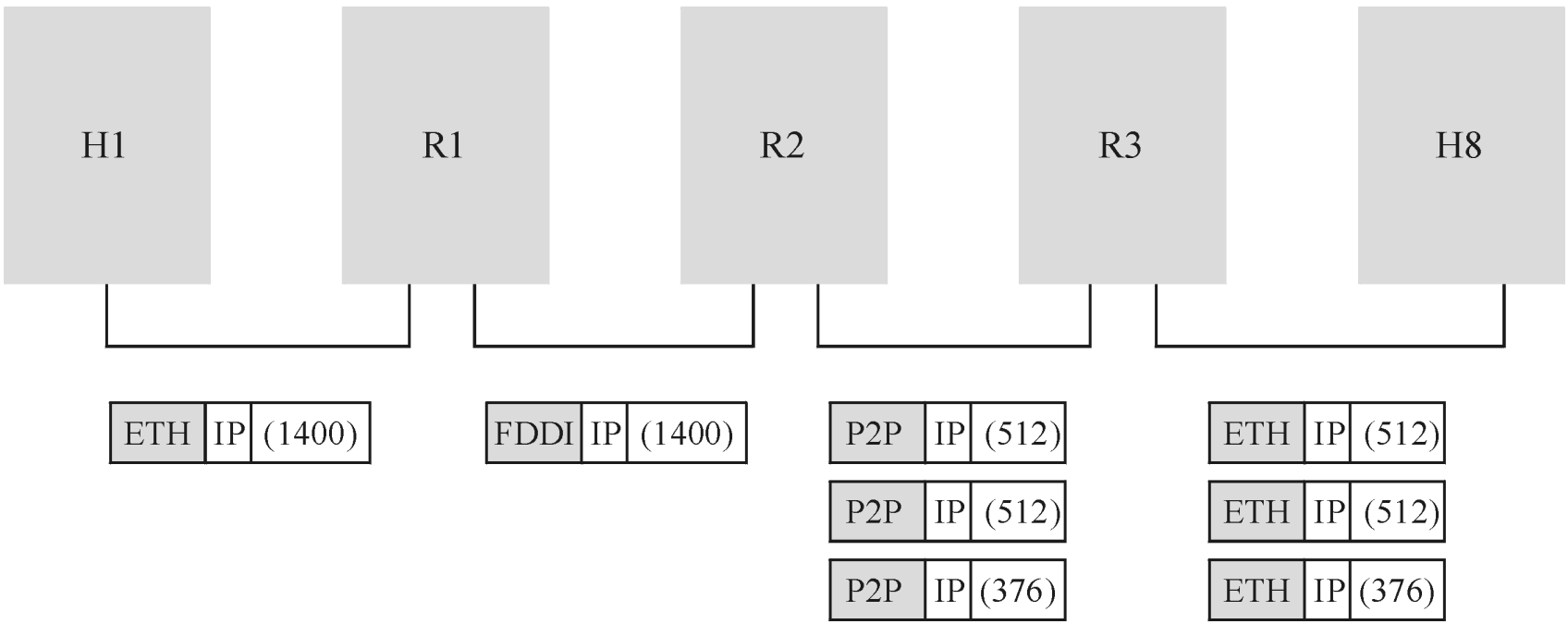


format datagramu IP

- *wybrane pola z pozostałych :*
- wersja, n.p. IPv4, IPv6
- typ usługi: *ważność* datagramu IP (określenie wymaganego opóźnienia, przepustowości, niezawodności)
- długość datagramu (maks. 65535 bajtów)
- identyfikator, flagi, przesunięcie (do *fragmentacji*)
- czas życia: usuwa datagramy „zapętlone” w pętli wyboru trasy
- protokół: *klucz demultipleksacji* (TCP=6, UDP=17)
- suma kontrolna (w słowach 16-bitowych)

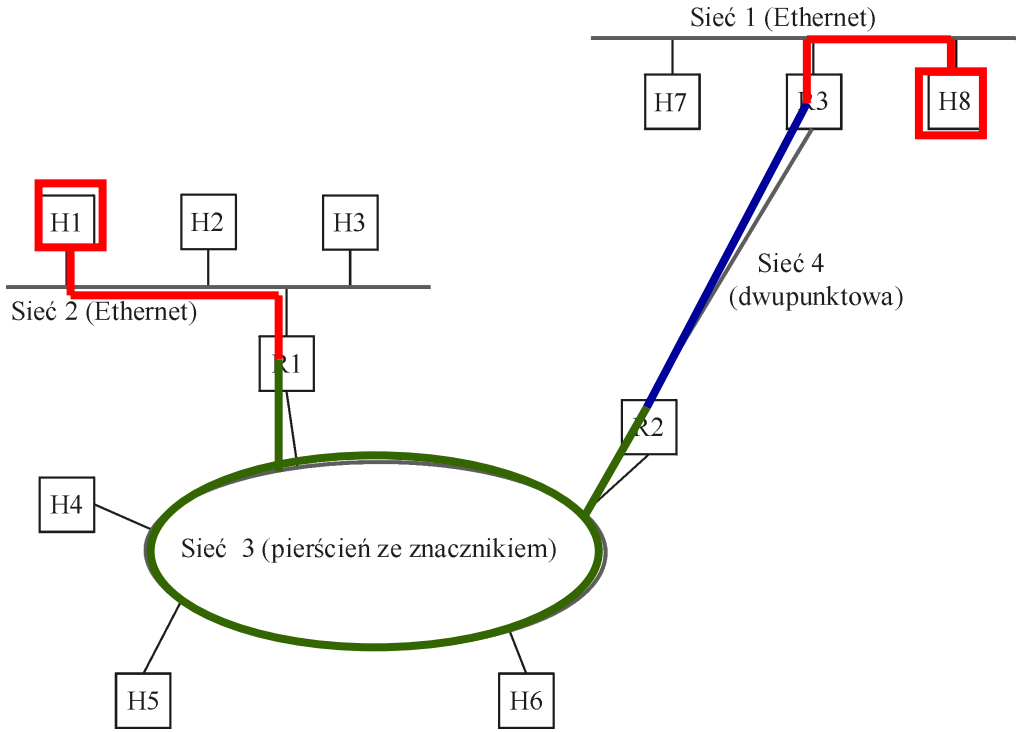
fragmentacja i składanie

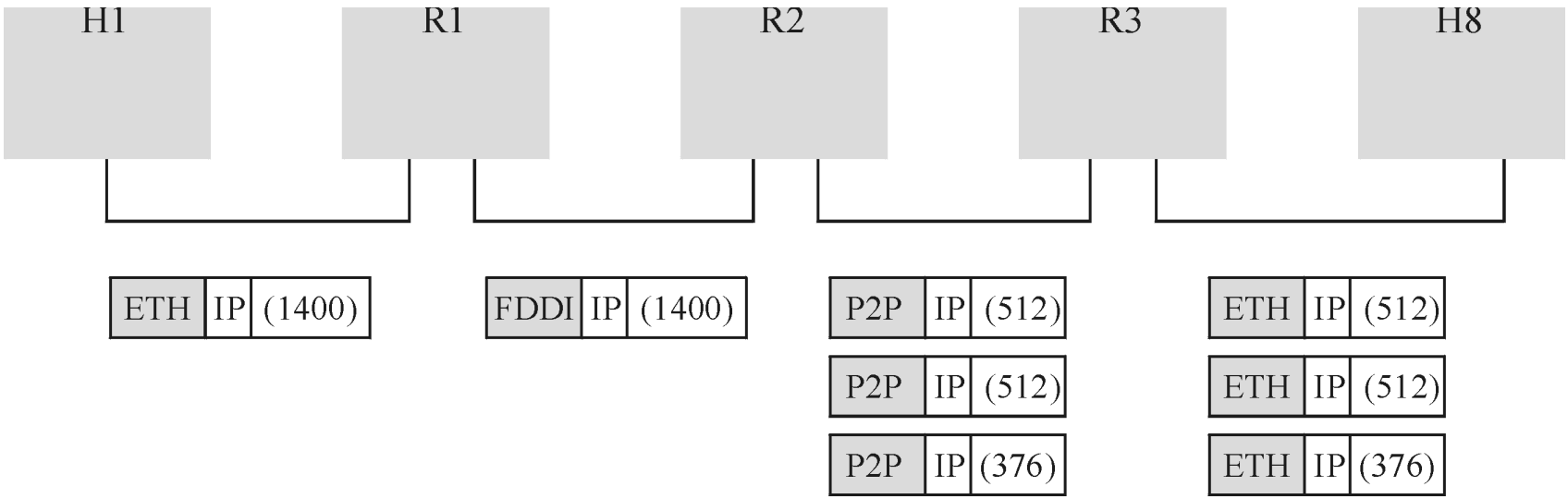
- sieć fizyczna *nie obsługuje* pakietów dł. 65535 bajtów
- stąd *fragmentacja* (Ethernet 1500 bajtów, FDDI 4500 bajtów, P2P 532 bajty)
- *przykład*: przesłanie datagramu IP dł. 1420 bajtów



fragmentacja i składanie

- *topologia sieci* ➔
- *datagram IP w sek-wencji sieci fizycz-nych* ➔





fragmentacja i składanie

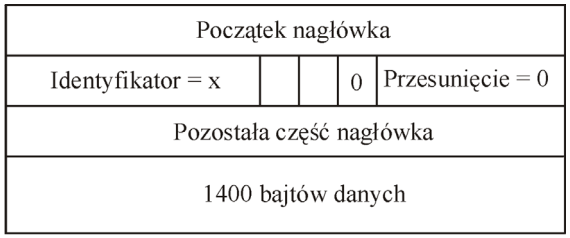
- w nagłówku datagramu IP (drugie słowo):

identyfikator (x, ten sam dla wszystkich fragmentów)

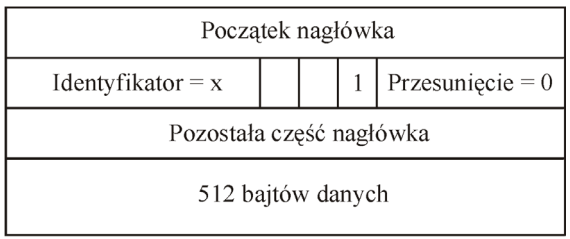
flaga M=1 w polu flagi (będą dalsze fragmenty)

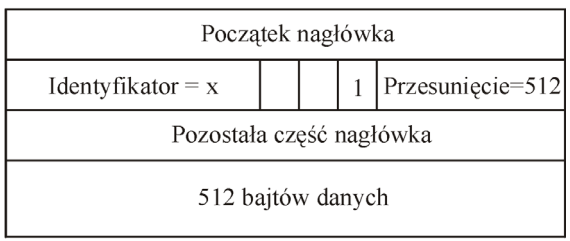
pole przesunięcie (w pierwszym 0, drugim n.p. 512, trzecim 1024)

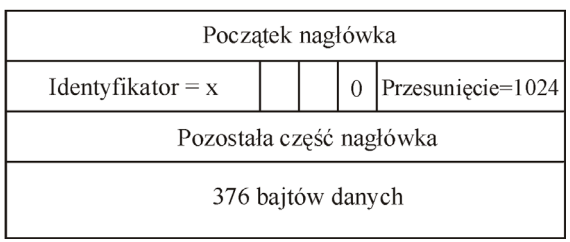
(a)



(b)

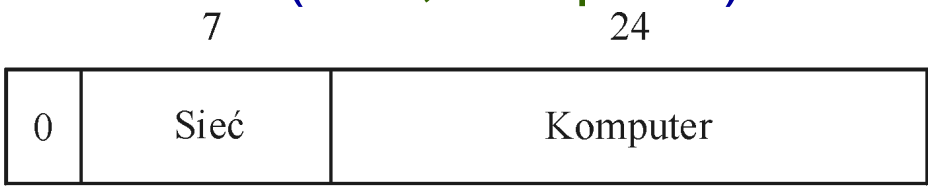




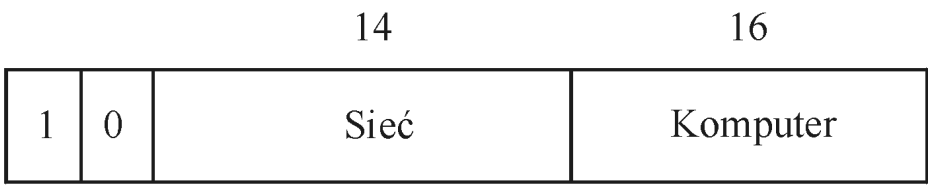


adresy globalne

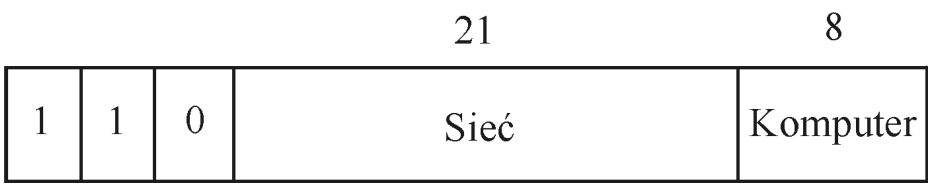
- adresy Ethernetu są globalnie niepowtarzalne, ale niestety są *proste*
- adresy IP są globalnie niepowtarzalne, ale na szczęście są *hierarchiczne* (sieć, komputer)
- klasa A



- klasa B

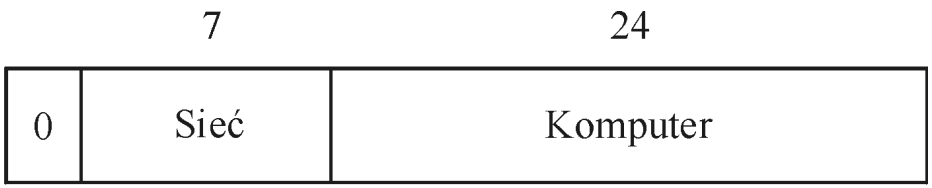


- klasa C

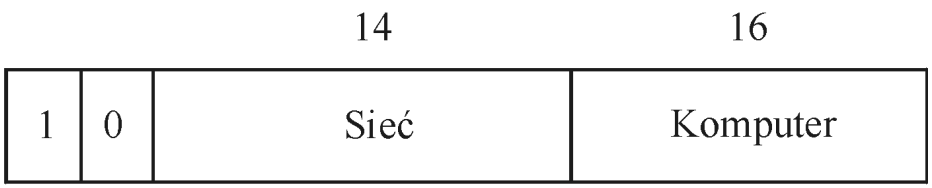


adresy IP

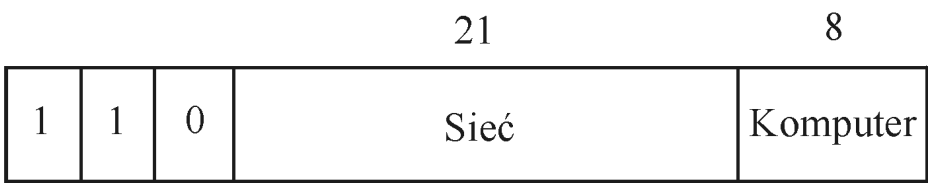
- ~4 miliardy adresów(adresowanie klasowe)
- 2^7-2 (126) sieci klasy A, po 2^{24} -2 (16177214) komputerów
- 2^{14} -2 (16382) sieci klasy B, po 2^{16}-2 (65534)komputerów
- 2^{21}-2 (2097150) sieci klasy C, po 2^8 -2 (254) komputerów
- klasa A
  1-126



- klasa B
  128-191



- klasa C
  192-223



adresy IP

- faktycznie jest mniej niż 4 miliardy adresów:
- klasa A - około 2mld adresów:

  126 ×16 177 214 = 2 038 328 964
- klasa B - około 1mld adresów:

  16 382 × 65534 = 1 073 577 988
- klasa C - ponad 500 mln adresów:

  2 097 150 × 254 = 532 176 100
- inne klasy:

  klasa D, adresy od 1110, 224-239, *rozsyłanie grupowe*

zapis adresu IP

- zapis adresu IP:

  adres IP: cztery *dziesiętne* liczby całkowite, oddzielone kropkami

- n.p.: 150.254.173.3

- *nazwa* tego komputera: rose.man.poznan.pl

adresy prywatne

- w każdej klasie są adresy IP *nie przydzielone*:
  10.0.0.0 - 10.255.255.255
  172.16.0.0 - 172.31.255.255
  192.168.0.0 - 192.168.255.255

- zastosowanie: w sieciach nie przyłączonych do Internetu, np. w sieciach bankowych
- z definicji, ruch kierowany do tych adresów nie jest obsługiwany w Internecie

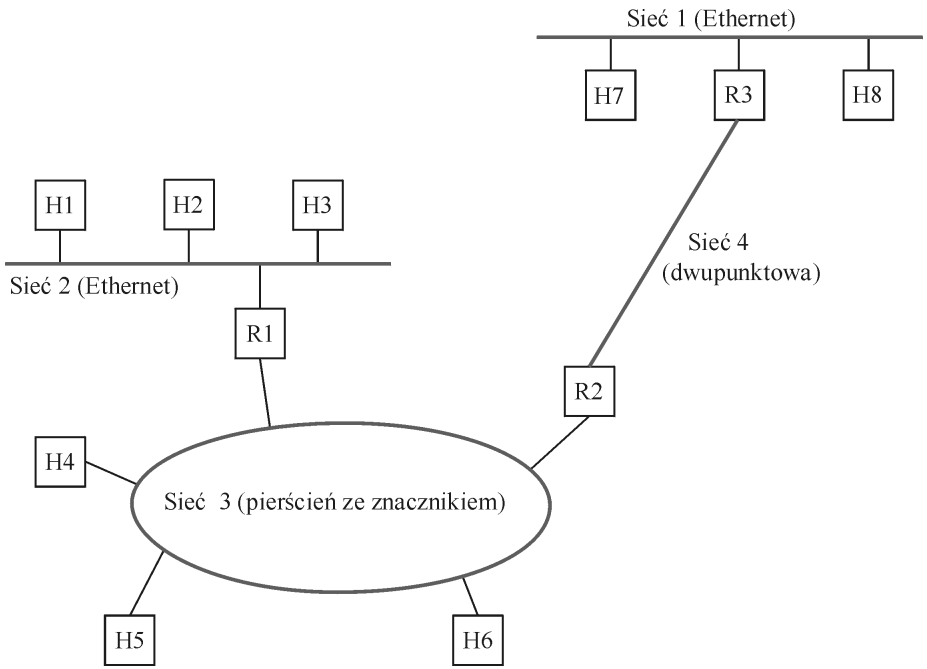
kierowanie datagramów przez protokół IP

- *kierowanie*: pobieranie datagramu z wejścia i nadanie do właściwego wyjścia
- datagram zawiera adres IP komputera odbiorczego
- węzeł (komputer lub ruter) *ustala*, czy jest dołączony do tej samej sieci co odbiorca, porównując *części dotyczące sieci* w adresie odbiorcy i w adresach interfejsu komputera i interfejsów rutera
- zgodność oznacza, że jest to ta sama sieć
- niezgodność oznacza przesłanie datagramu IP do *rutera, wybranego z tablicy kierującej węzła*, albo przesłanie do *rutera domyślnego*

kierowanie datagramów przez protokół IP

- *przykłady*: H1 nadaje do H2, ta sama sieć, co H1 (jak H1 znajduje poprawny adres Ethernetu dla H2?→ARP)
- H1 nadaje do H8, inna sieć, jedyny wybór to ruter R1
- R1 do R2 (R2 to ruter domyślny dla R1)
- R2 bada numer sieci w adr. H8 i kieruje do R3:

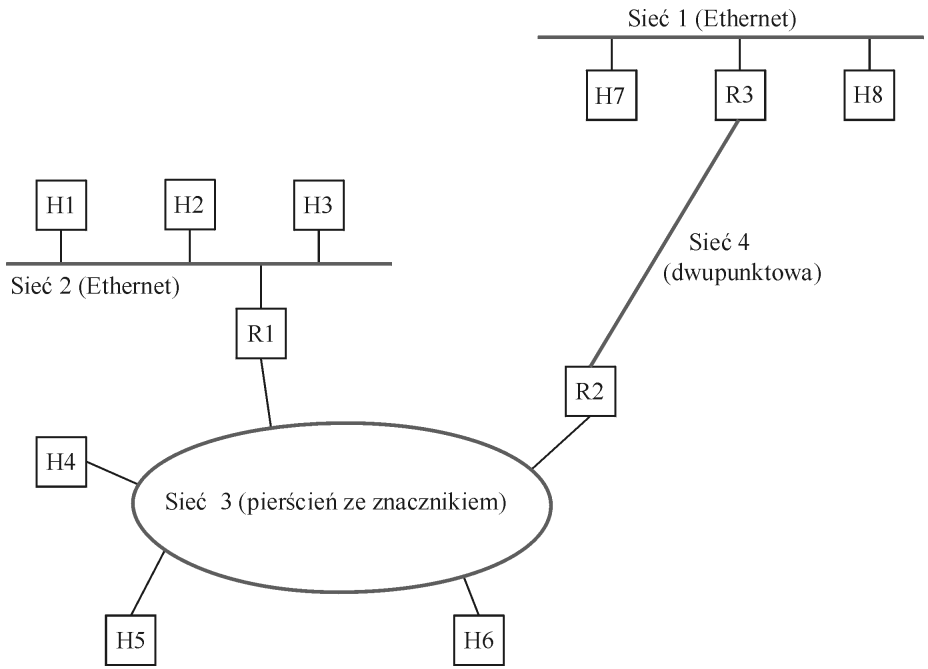
NumerSieci	NastEtap
1	R3
2	R1



kierowanie datagramów przez protokół IP

- *kompletna tablica kierująca dla rutera* R2

NumerSieci	NastEtap
1	R3
2	R1
3	interfejs 1
4	interfejs 0

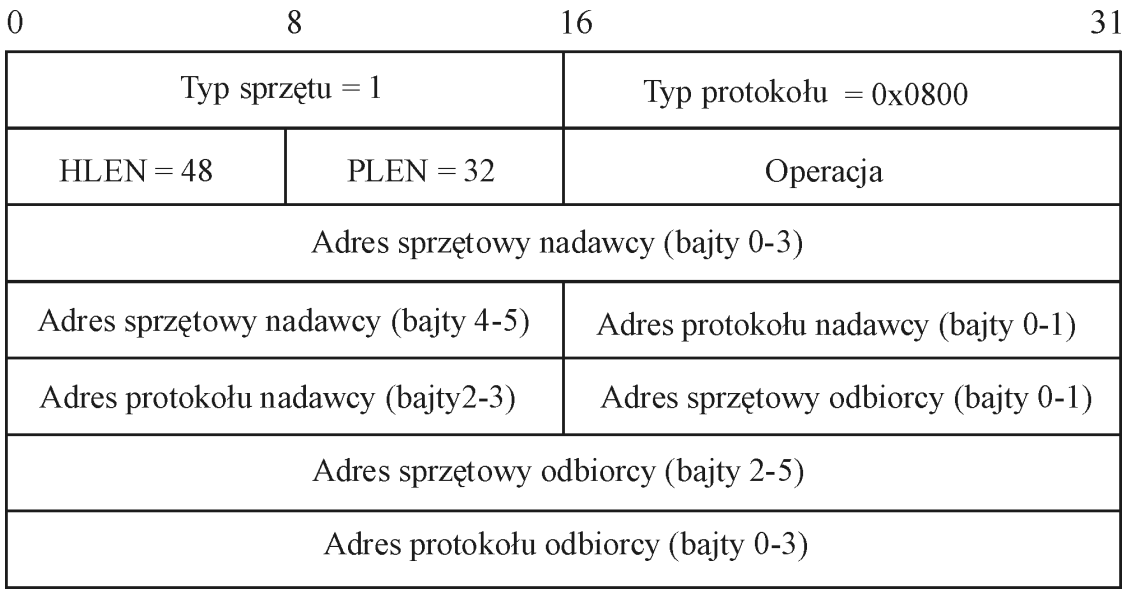


translacja adresów (ARP)

- *przełumaczenie* adresu IP na adres fizyczny na poziomie łącza, mający sens w danej sieci (np. adres Ethernetu)
- dokonuje się tego korzystając z *tablic odwzorowania adresów*, tworzonych dynamicznie za pomocą *protokołu odwzorowania adresów* (ARP)
- gdy komputer, który chce nadać datagram IP nie ma w tablicy odwzorowania adresu IP na adres na poziomie łącza, *rozgłasza zapytanie ARP z adresem* IP - komputer z tym adresem IP odpowiada nadawcy zapytania, podając swój adres na poziomie łącza
- zapytanie zawiera też adres IP i adres na poziomie łącza *nadawcy* (przydatne przy odpowiedzi)

format pakietu ARP

- typ sprzętu (1=Ethernet), typ protokołu (n.p. IP)
- HLEN i PLEN: długości adresów sprzętu i protokołu
- operacja (żądanie albo odpowiedź)
- adresy protokołu (IP, 32 bity) i adresy sprzętowe (Ethernet, 48 bitów) nadawcy i odbiorcy



mechanizmy IP do uzyskania...

- *heterogeniczności:*
- *model usługi dostępnych możliwości* (zawodne datagramy) +

+ wspólny format pakietu (*fragmentacja i składanie*) +

+ globalna przestrzeń adresowa dla identyfikacji wszystkich komputerów (ARP w różnych sieciach fizycznych)

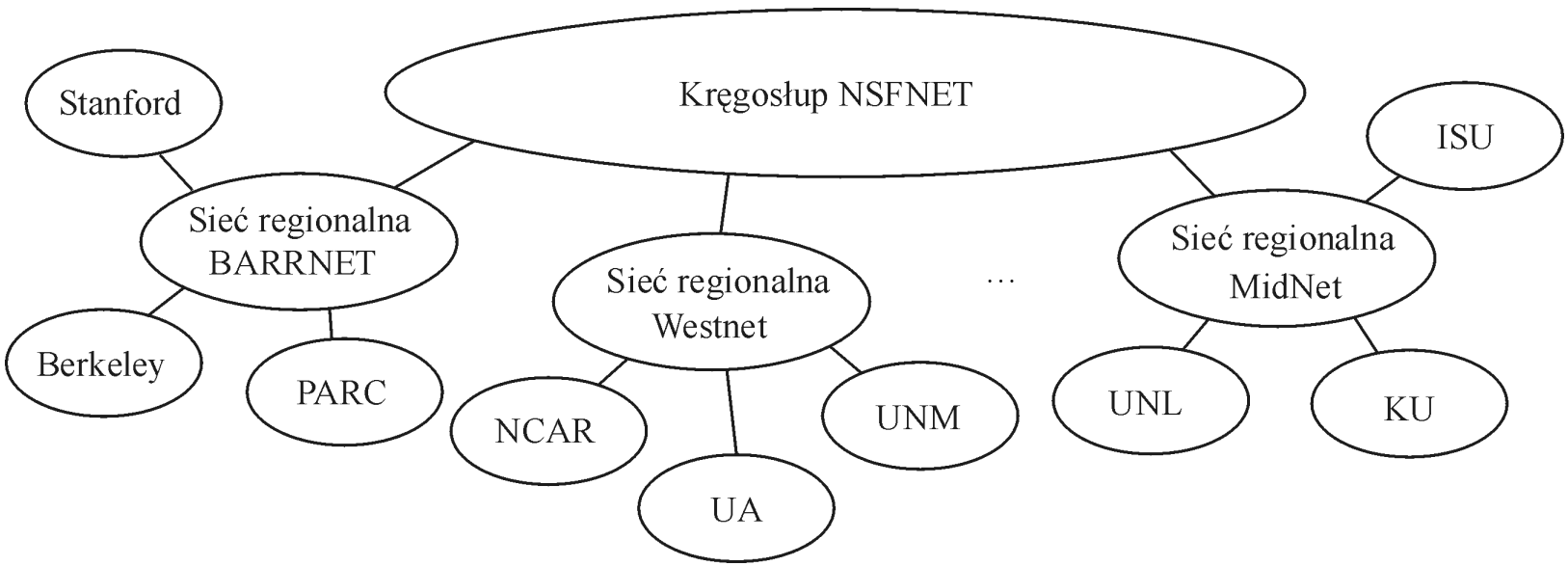
- *skalowalności:*
- *agregacja hierarchiczna* w celu ograniczenia informacji potrzebnej do *kierowania pakietów* (podział adresu IP na część sieciową i część odnoszącą się do komputera) - kierowanie do sieci, potem do komputera

sprawozdanie o błędach

- *protokół komunikatów kontrolnych Internetu* ICMP, skojarzony z protokołem IP
- zestaw komunikatów o błędach, przesyłanych do komputera nadawczego, kiedy ruter lub komputer niezdolny do przetworzenia datagramów IP
- n.p.: komputer nieosiągalny, składanie się nie powiodło, suma kontrolna się nie zgadza, TTL=0, ...
- komunikaty kontrolne od rutera do komputera nadawczego: n.p. skieruj inną trasą (ICMP-Redirect)

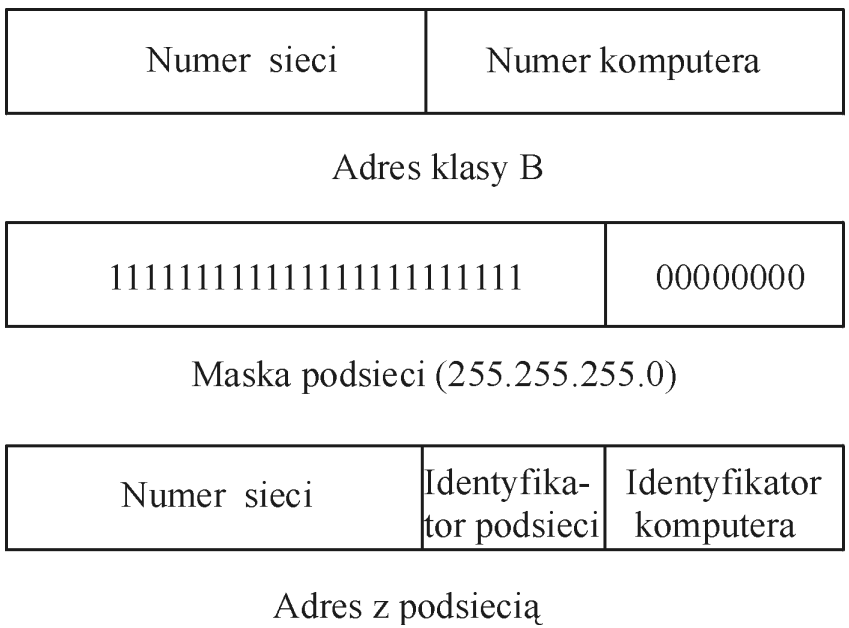
globalna intersieć

- struktura drzewa Internetu (1990)
- miejsca połączone w *sieć regionalną*
- sieci regionalne połączone *siecią kręgosłupową*



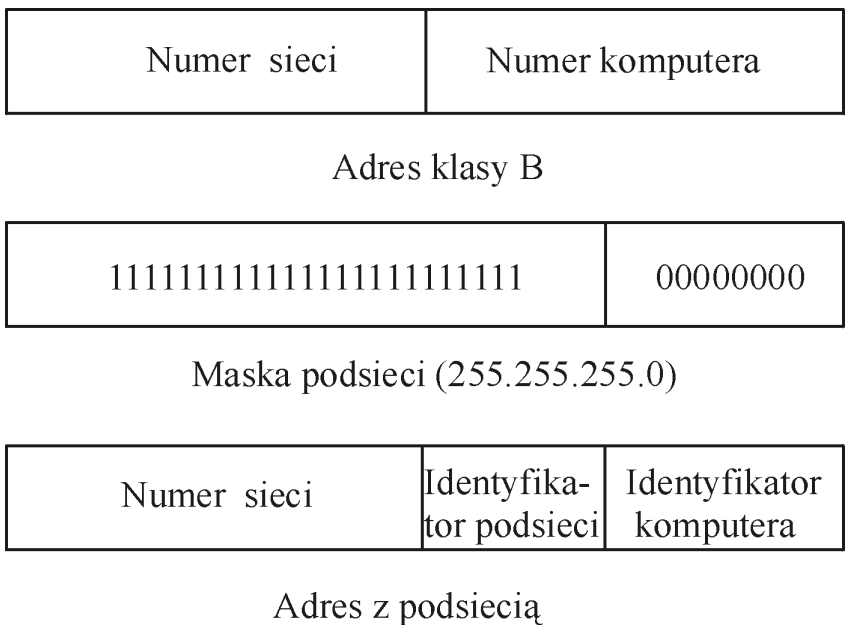
podział sieci na podsieci

- jeden numer sieci jest *współdzielony* przez wiele sieci
- wprowadzenie dodatkowego *poziomu hierarchii*
- wprowadzenie *numeru podsieci* za pomocą *maski podsieci*



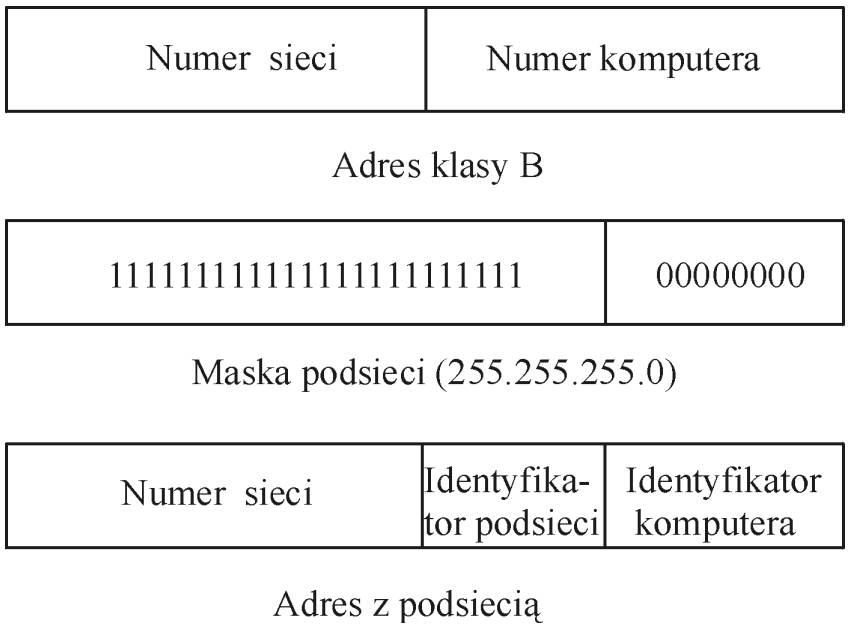
przykład podziału sieci na podsieci

- adres IP klasy B, z maską podsieci 255.255.255.0 powoduje podział dotychczasowego numeru komputera na 8 bitów na podsieć i 8 bitów na komputer w podsieci
- 2^8-2 podsieci, po 2^8-2 komputerów w każdej podsieci, 2^8-2=254, co daje: 254×254=64516 komput.
- przed podziałem było 2^{16}-2=65534 komputerów w sieci
- zmniejszenie do (w %): 64516/65534= 98,47%



przykład podziału sieci na podsieci

- n.p. gdy adresem sieci klasy B jest 128.96.0.0, to *po podziale mamy podsieci*:
- 128.96.1.0 z komputerami:
  128.96.1.1 do 128.96.1.254
- 128.96.2.0 z komputerami
  128.96.2.1 do 128.96.2.254
  ...
- 128.96.254.0 z komputerami
  128.96.254.1 do 128.96.254.254
- 255 na ostatnim bajcie: *rozgłaszanie* w podsieci



przykład podziału sieci na podsieci

- adres IP klasy B, z maską podsieci 255.255.255.128 powoduje podział dotychczasowego numeru komputera na 9 bitów na podsieć i 7 bitów na komputer w podsieci
- 2^9-2 podsieci, po 2^7-2 komputerów w każdej podsieci, 2^9-2=510, a 2^7-2=126, co daje $510 \times 126 = 64260$ komputerów
- przed podziałem w sieci było 2^{16}-2=65534 komputerów
- zmniejszenie do (w %): 64260/65534= 98,06%

przykład podziału sieci na podsieci

- n.p. gdy adresem sieci klasy B jest 128.96.0.0, to *po podziale mamy podsieci*:
- 128.96.1.0 z komputerami:

  od 128.96.1.1 do 128.96.1.126

  rozgłaszanie w podsieci: 128.96.1.127
- 128.96.1.128 z komputerami:

  od 128.96.1.129 do 128.96.1.254

  rozgłaszanie w podsieci: 128.96.1.255

  ...
- 128.96.254.128 z komputerami

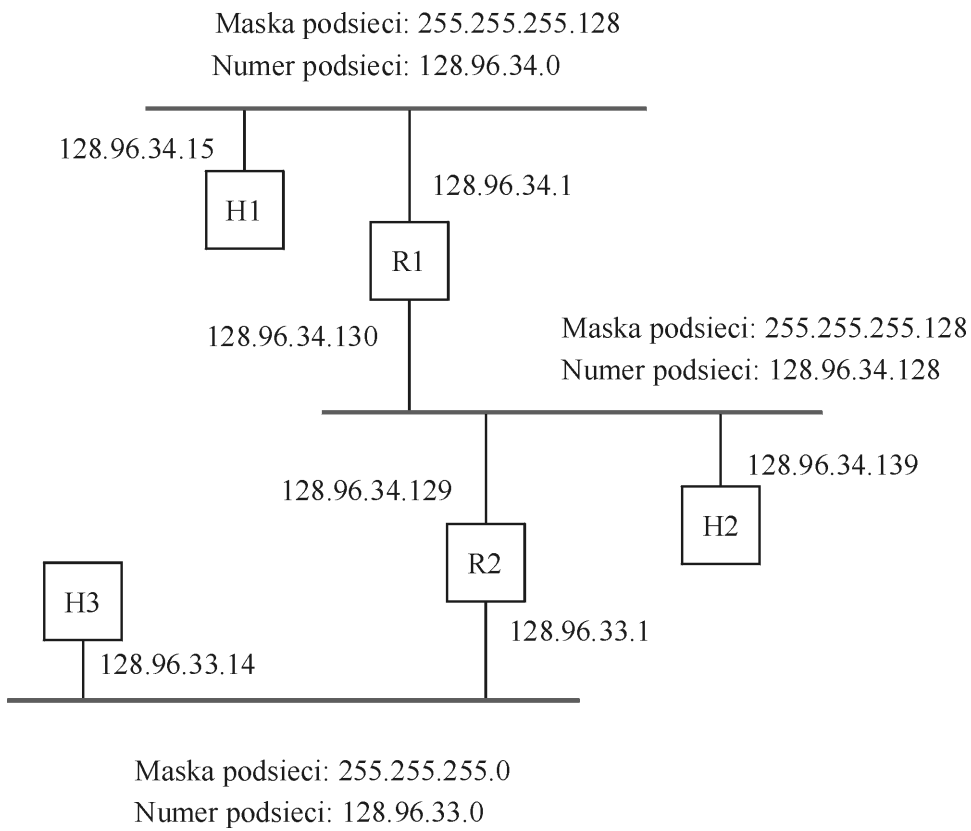
  od 128.96.254.129 do 128.96.254.254

nadanie datagramu przez H1 do H2

- operacja AND na masce podsieci H1 i adresie H2:

255.255.255.128 AND 128.96.34.139 = 128.96.34.128

- nie pasuje do numeru podsieci dla H1
- H1 nadaje pakiet do rutera R1



nadanie datagramu przez H1 do H2

numer podsieci	Maska podsieci	NastepnyEtap
128.96.34.0	255.255.255.128	Interfejs 0
128.96.34.128	255.255.255.128	Interfejs 1
128.96.33.0	255.255.255.0	R2

- R1 dokonuje operacji AND na adresie H2 i na masce podsieci pierwszego elementu w tablicy kierującej R1

128.96.34.139 AND 255.255.255.128 = 128.96.34.128

- wynik nie pasuje do numeru podsieci pierwszego elementu (128.96.34.0)

nadanie datagramu przez H1 do H2

numer podsieci	Maska podsieci	NastepnyEtap
128.96.34.0	255.255.255.128	Interfejs 0
128.96.34.128	255.255.255.128	Interfejs 1
128.96.33.0	255.255.255.0	R2

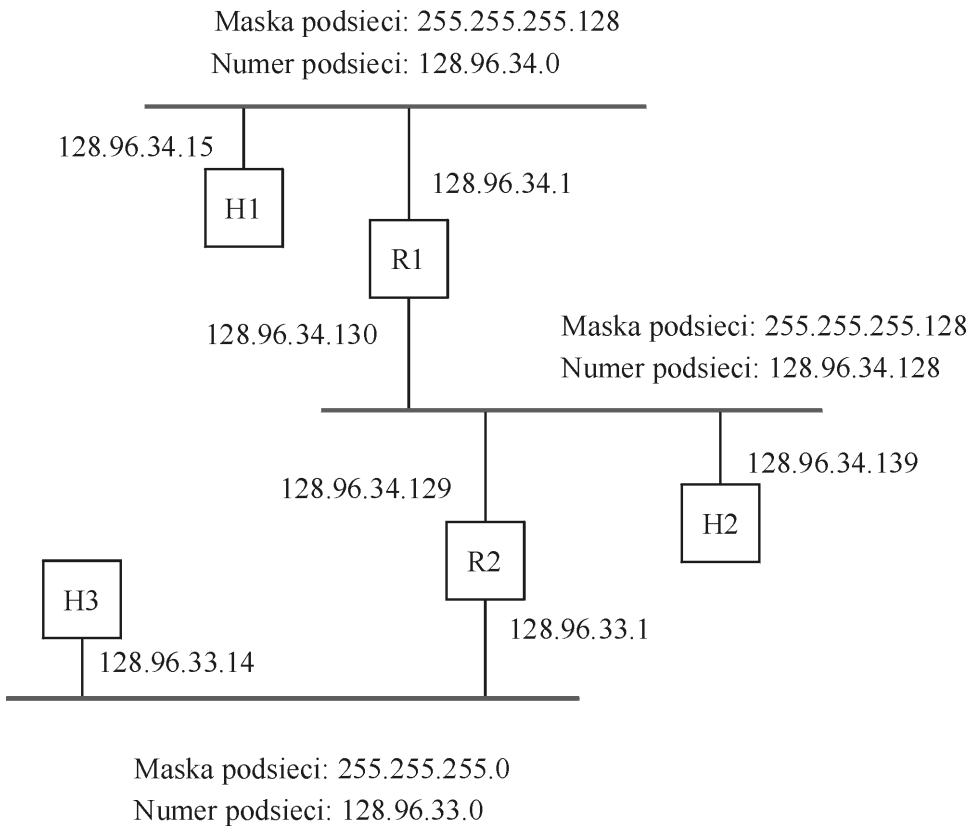
- R1 dokonuje operacji AND na adresie H2 i na masce podsieci drugiego elementu w tablicy kierującej R1

128.96.34.139 AND 255.255.255.128 = 128.96.34.128

- wynik pasuje do numeru sieci drugiego elementu (128.96.34.128)
- R1 dostarcza datagram do H2 za pomocą interfejsu 1, dołączonego do tej samej sieci co H2

nadanie datagramu przez H1 do H2

- datagram trafia do H2



podział sieci na podsieci (RFC950)

- zadanie: podzielić sieć 192.10.12.0 na 6 podsieci:
- adres klasy C, ilość bitów pożyczonych na podsieć od numeru komputera: 3, gdyż $2^3 = 8 > 6$
- na numer komputera pozostaje 5 bitów (8-3=5)
- stąd maska podsieci ma 27 bitów (24+3=27)
- maska podsieci: 255.255.255.224 (128+64+32=224)
- adres podsieci, ostatni bajt, zakres komputerów, adres rozgłaszania:

  podsieć #1: 192.10.12.32, (00100000)

  komputery od 192.10.12.33 do 192.10.12.62

  rozgłaszanie 192.10.12.63

podział sieci na podsieci (RFC950)

- adres podsieci, ostatni bajt, zakres komputerów, adres rozgłaszania:

  podsieć #2: 192.10.12.64, (01000000)

  komputery od 192.10.12.65 do 192.10.12.94

  rozgłaszanie 192.10.12.95
- podsieć #3: 192.10.12.96, (01100000)

  komputery od 192.10.12.97 do 192.10.12.126

  rozgłaszanie 192.10.12.127
- podsieć #4: 192.10.12.128, (10000000)

  komputery od 192.10.12.129 do 192.10.12.158

  rozgłaszanie 192.10.12.159

podział sieci na podsieci (RFC950)

- adres podsieci, ostatni bajt, zakres komputerów, adres rozgłaszania:

  podsieć #5: 192.10.12.160, (10100000)

  komputery od 192.10.12.161 do 192.10.12.190

  rozgłaszanie 192.10.12.191
- podsieć #6: 192.10.12.192, (11000000)

  komputery od 192.10.12.193 do 192.10.12.222

  rozgłaszanie 192.10.12.223
- w sieci 192.10.12.0 *przed podziałem* : 2^8-2 (254) komputery, *po podziale* 6 $\times$ 30=180 komputerów, co daje: 180/254=71%

podział sieci na podsieci (obecnie)

- zadanie: podzielić sieć 192.10.12.0 na 8 podsieci:
- adres klasy C, ilość bitów pożyczonych na podsieć od numeru komputera: 3, gdyż $2^3 = 8$
- na numer komputera pozostaje 5 bitów (8-3=5)
- stąd maska podsieci ma 27 bitów (24+3=27)
- maska podsieci: 255.255.255.224 (128+64+32=224)
- adres podsieci, ostatni bajt, zakres komputerów, adres rozgłaszania:

  podsieć #0: 192.10.12.0, (00000000)

  komputery od 192.10.12.1 do 192.10.12.30

  rozgłaszanie 192.10.12.31

podział sieci na podsieci (obecnie)

- adres podsieci, ostatni bajt, zakres komputerów, adres rozgłaszania:

  podsieć #1: 192.10.12.32, (00100000)

  komputery od 192.10.12.33 do 192.10.12.62

  rozgłaszanie 192.10.12.63
- podsieć #2: 192.10.12.64, (01000000)

  komputery od 192.10.12.65 do 192.10.12.94

  rozgłaszanie 192.10.12.95
- podsieć #3: 192.10.12.96, (01100000)

  komputery od 192.10.12.97 do 192.10.12.126

  rozgłaszanie 192.10.12.127

podział sieci na podsieci (obecnie)

- adres podsieci, ostatni bajt, zakres komputerów, adres rozgłaszania:

  podsieć #4: 192.10.12.128, (10000000)

  komputery od 192.10.12.129 do 192.10.12.158

  rozgłaszanie 192.10.12.159
- podsieć #5: 192.10.12.160, (10100000)

  komputery od 192.10.12.161 do 192.10.12.190

  rozgłaszanie 192.10.12.191
- podsieć #6: 192.10.12.192, (11000000)

  komputery od 192.10.12.193 do 192.10.12.222

  rozgłaszanie 192.10.12.223

podział sieci na podsieci (obecnie)

- adres podsieci, ostatni bajt, zakres komputerów, adres rozgłaszania:
- podsieć #7: 192.10.12.224, (11100000)

  komputery od 192.10.12.225 do 192.10.12.254

  rozgłaszanie 192.10.12.255
- w sieci 192.10.12.0 *przed podziałem* : 2^8-2 (254) komputery, *po podziale* 8 $\times$ 30=240 komputerów, co daje: 240/254=94%

system autonomiczny (AS)

- *system autonomiczny*: region Internetu, będący pod administracyjną kontrolą jednej stacji, np. dostawcy usług Internetu. Nazywany też *domeną wyboru trasy*
- każdy AS ma unikalny 16 bitowy identyfikator, nadawany przez Network Operation Center (NOC)
- złożoność AS *nie jest ujawniana* reszcie Internetu
- *dodatkowa* hierarchiczna agregacja informacji o wyborze trasy
- wybór trasy *wewnątrz domeny* i *między domenami*
- redukcja informacji o wyborze trasy: *trasy domyślne* i *rutery graniczne*

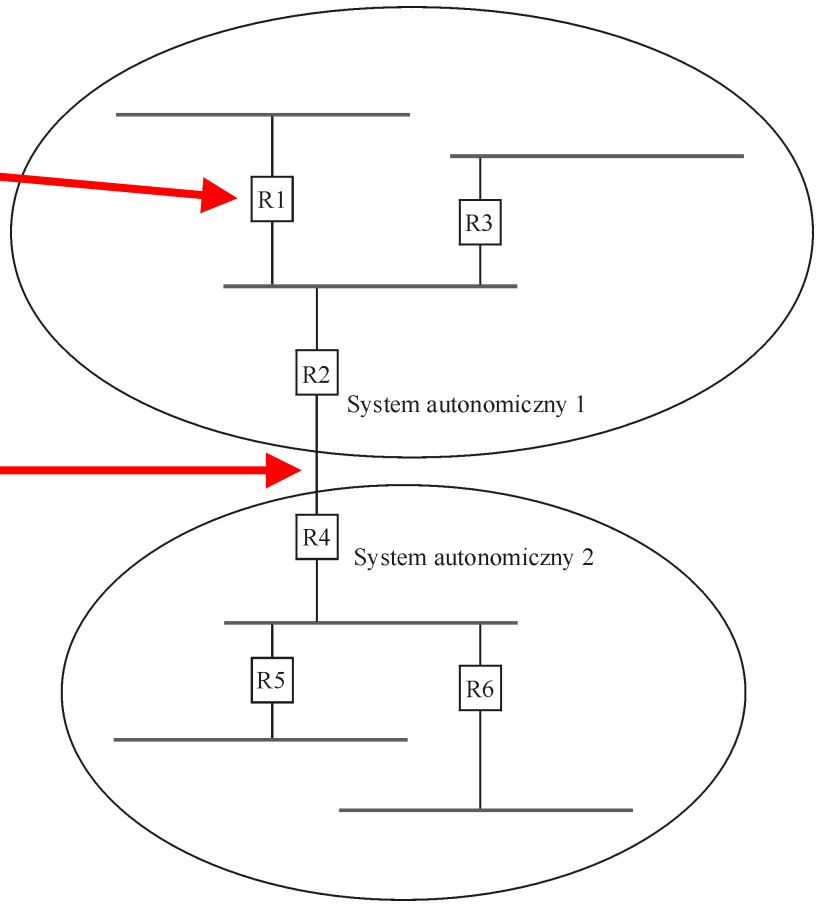
przykład systemów autonomicznych

- wybór trasy wewnątrz domeny:

RIP, IGRP, EIGRP, OSPF

- wybór trasy między domenami:

EGP, BGP

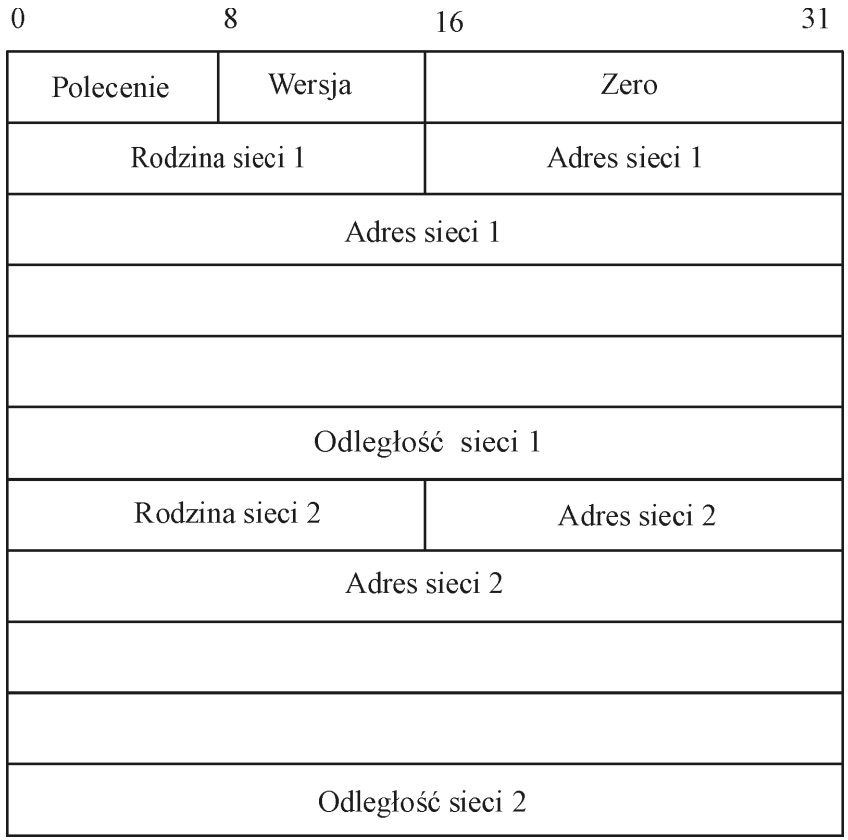


protokół wyboru trasy wewnątrz domeny

- **protokół informowania o trasach RIP**
- rozpowszechniany z Unix BSD
- wybór trasy na podstawie *wektora odległości*
- ruter przesyła informację o kosztach osiągnięcia innych ruterów (odległościach do innych sieci)
- aktualizacja co 30 s
- dozwolone odległości do 15 etapów
- 16 etapów oznacza *nieosiągalność sieci* (= ∞)

format pakietu RIP

- ⟨adres sieci, odległość⟩
  na adres sieci 14 bajtów
- w wersji RIPv1:
  4 bajty adres IP
- w wersji RIPv2:
  4 bajty adres IP
  + maska podsieci
  (obsługa podziału sieci
  na podsieci)



protokół wyboru trasy wewnątrz domeny

- **protokół „wpierw najkrótsza ścieżka” OSPF**
- wybór trasy na podstawie *stanu łącza*,

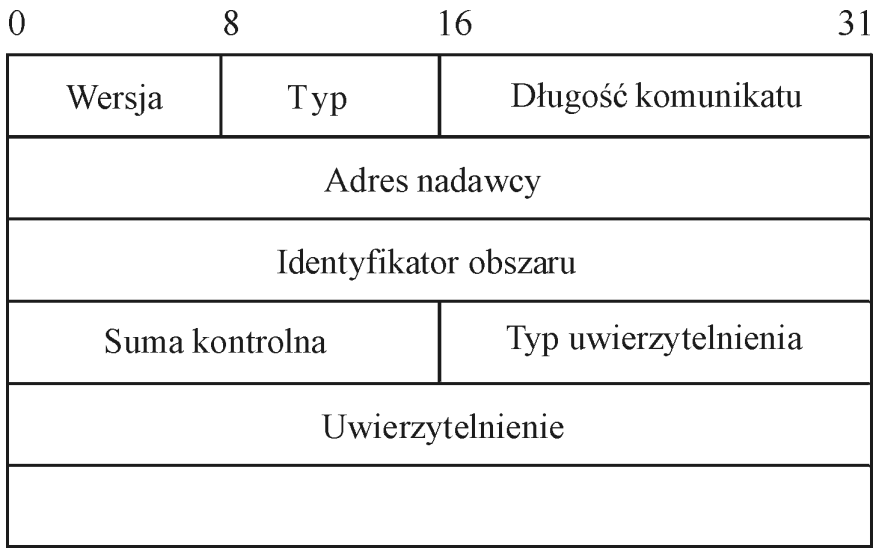
plus następujące cechy:

- *uwierzytelnienie komunikatów wyboru trasy za pomocą 8-bajtowego hasła*
- *dodatkowa hierarchia* (podział domeny na *obszary*):

  zmniejszenie ilości przesyłanych informacji, każdy obszar ma 32 bitowy identyfikator
- *zrównoważenie obciążenia*: trasom do tego samego miejsca przydziela się ten sam koszt

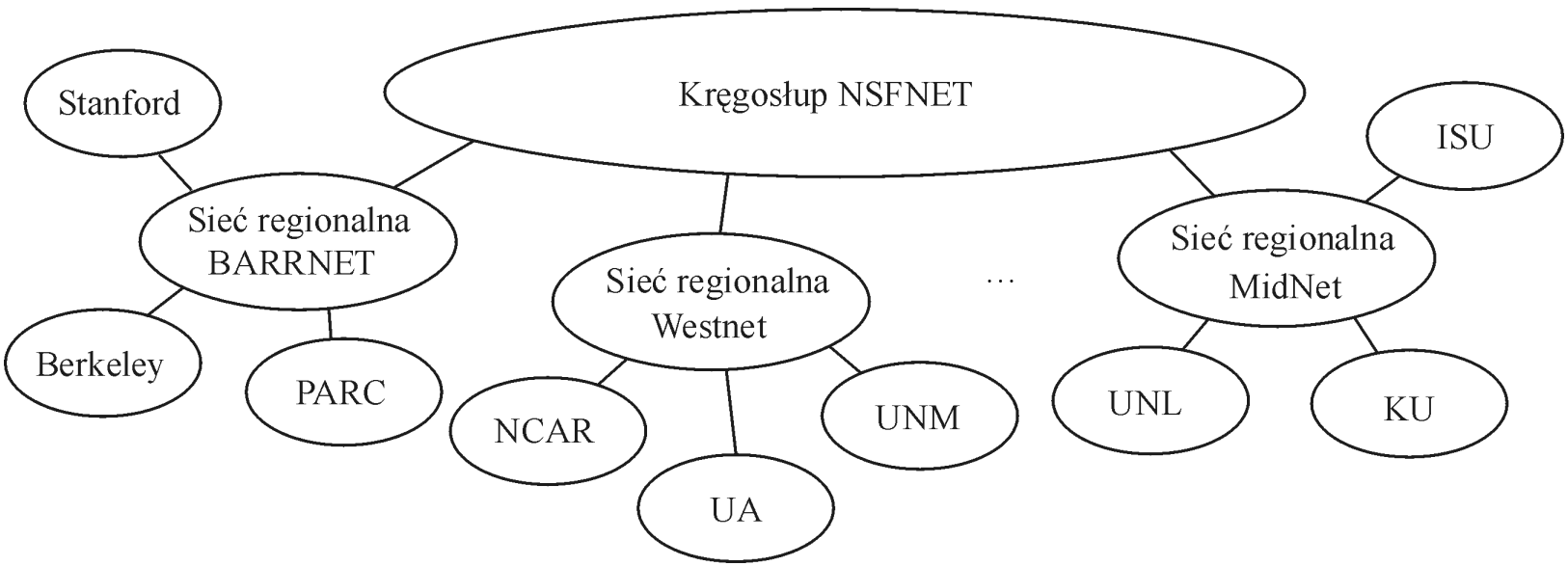
format nagłówka pakietu OSPF

- *wersja* = 2
- *typ*, n.p. = 1, komunikat halo
- *typ uwierzytelnienia*=0 (brak), =1 (jest)
- *identyfikator obszaru*, w którym znajduje się węzeł



protokoły wyboru trasy między domenami

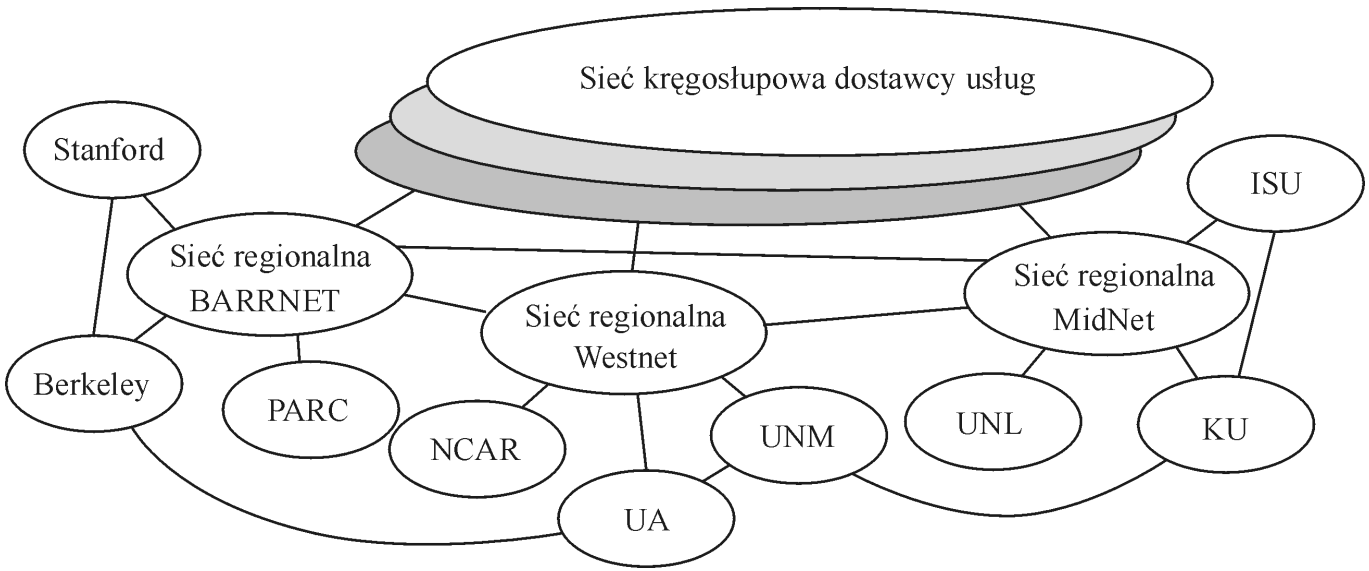
- **protokół bramki zewnętrznej EGP**
  ogranicza topologię Internetu do drzewa



Internet z jednym kręgosłupem (1990)

protokoły wyboru trasy między domenami

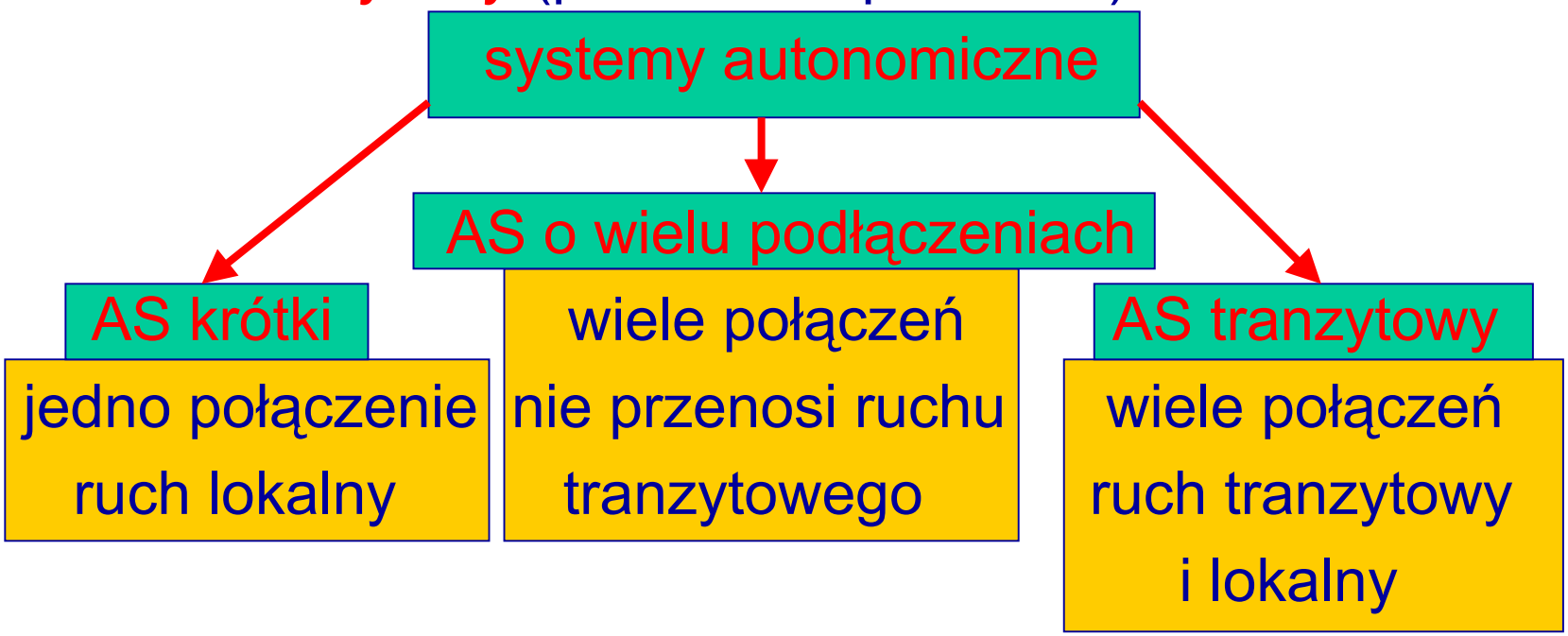
- **protokół bramki granicznej BGP**



Internet o wielu kręgosłupach (obecnie)

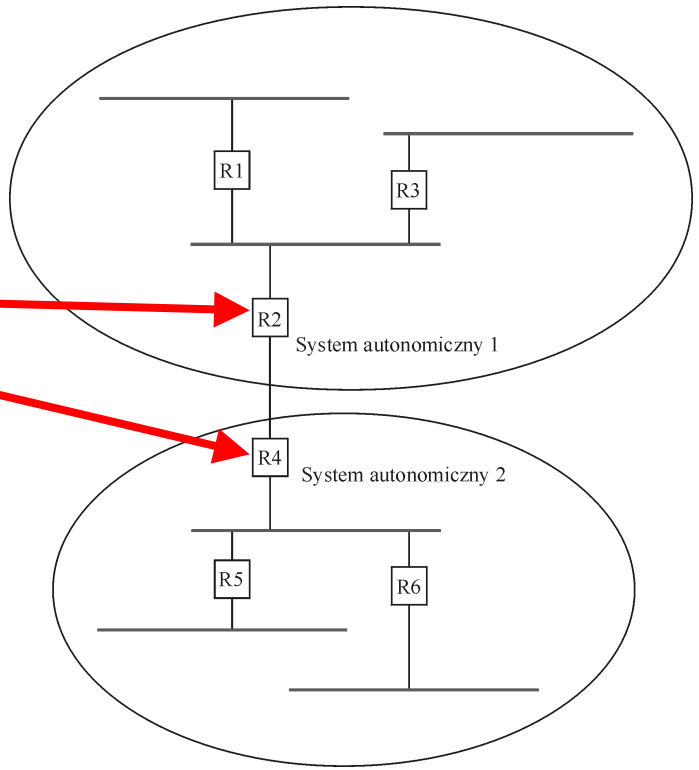
protokół bramki granicznej BGP

- obecnie wersja BGP-4
- *założenie*: Internet jest strukturą niedrzewiastą
- *ruch lokalny* (powstaje albo kończy się w AS)
- *ruch tranzytowy* (przechodzi przez AS)



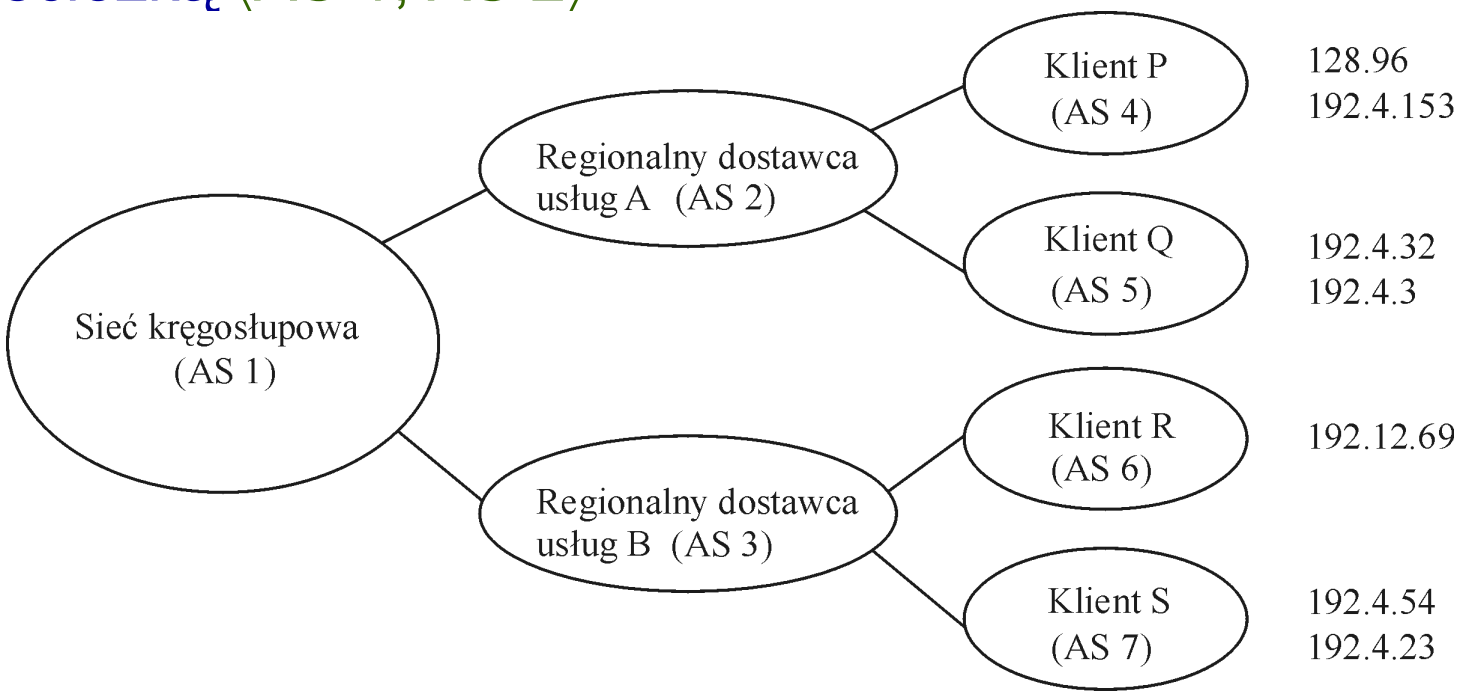
protokół bramki granicznej BGP

- *znajduje dowolną ścieżkę, bez pętli, do danej sieci*
- *nie polega* ani na wektorze odległości, ani na stanie łącza - *podaje kompletne ścieżki* (listy AS) pozwalające osiągnąć daną sieć
- *bramki graniczna*: ruter przez który pakiet osiąga dany AS
- *spiker*: rzecznik całego AS (różny od bramki granicznej)



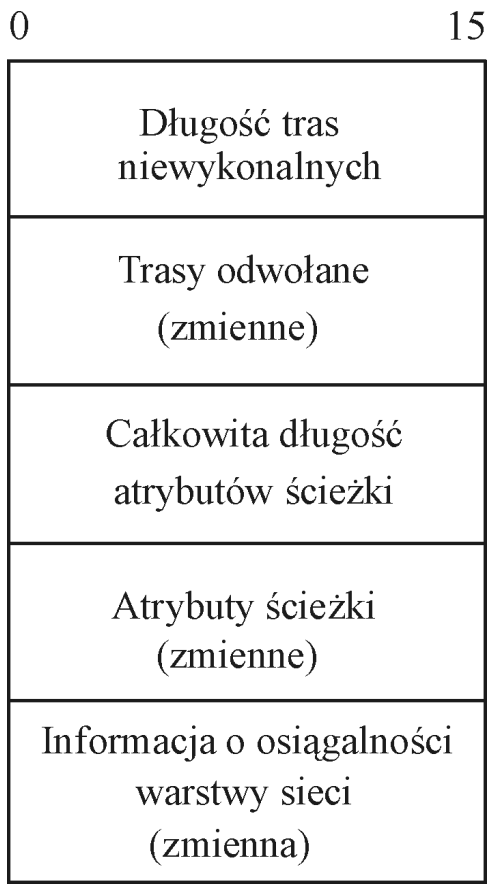
protokół bramki granicznej BGP - przykład

- spiker BGP dla AS2 podaje, że z AS2 można osiągnąć sieci 128.96, 192.4.153, 192.4.32 i 192.4.32
- sieć kręgosłupowa podaje, że sieci te osiąga się ścieżką ⟨AS 1, AS 2⟩



format pakietu aktualizującego BGP-4

- BGP *poleca* ścieżki do danej sieci
- BGP również *odwołuje* zalecane ścieżki
- *złożoność wyboru trasy* między domenami jest rzędu liczby AS
- *słaby punkt*: w ruterze granicznym ilość tras równa liczbie przydzielonych sieci



bezklasowy wybór trasy między domenami

- *problem* wyczerpywania się adresów IP
- *problem* za małej pamięci w ruterach
- *rozwiązanie*: rezygnacja ze ścisłych granic między klasami adresów i *agregacja tras* w technice CIDR
- *przykład*: przydział ciągłego bloku adresów klasy C od 192.4.16 do 192.4.31, czyli od

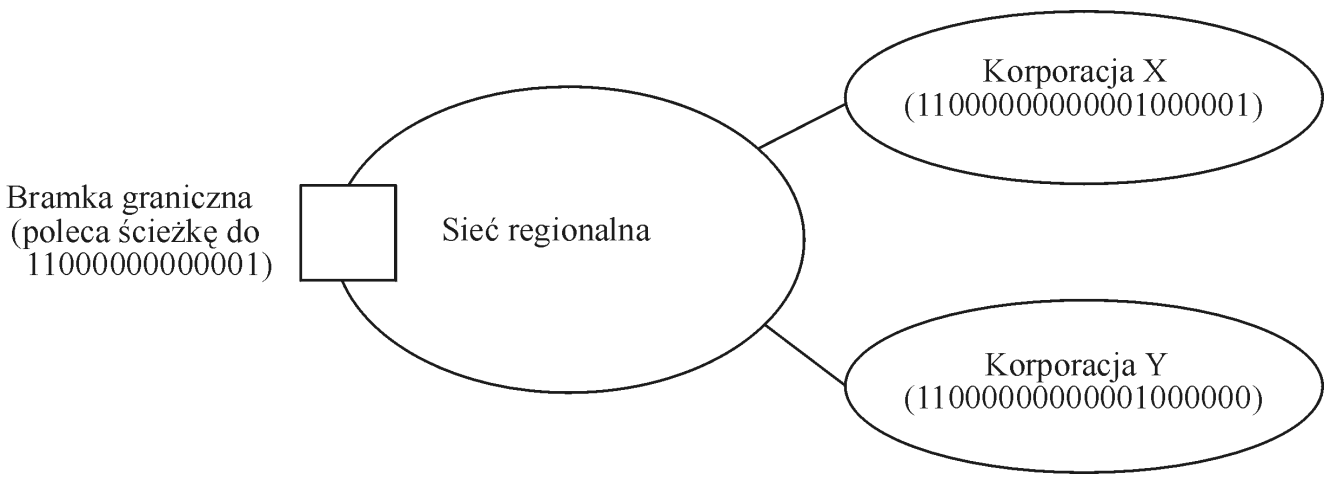
  11000000 00000100 00010000 xxxxxxxx do

  11000000 00000100 00011111 xxxxxxxx

  (górne 20 bitów takie same)
- CIDR składa adresy, które mogą być przydzielone pojedynczemu AS w jeden adres - *łączenie w nadsieć*

bezklasowy wybór trasy - CIDR

- korporacjom X i Y przydzielono sąsiednie 20 bitowe przedrostki sieci
- 19 bitów wskazuje na podsieci X i Y razem
- 20 bitów 11000000 00000100 0001 wskazuje na podsieć X
- 20 bitów 11000000 00000100 0000 wskazuje na podsieć Y



rozsyłanie grupowe

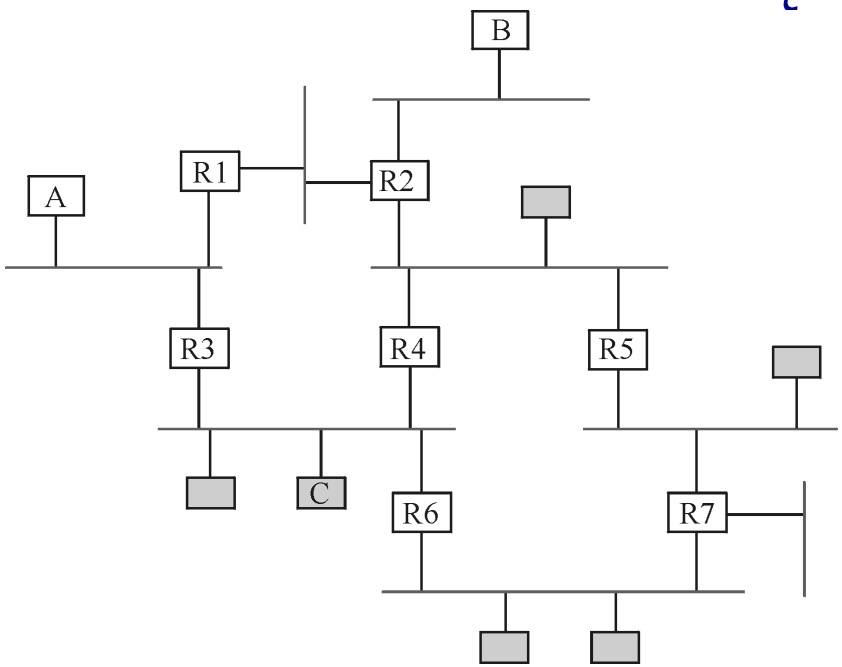
- *implementacja* rozsyłania grupowego w IPv4
- nadawanie pojedynczego pakietu pod *adres grupowy*
- decyzja o przynależności komputera do danej grupy jest *autonomiczną decyzją komputera*
- implementacja rozsyłania grupowego poprzez *rozszerzenie funkcji kierowania w ruterach*
- rozszerzenie protokołu *wektora odległości* (RIP)
- rozszerzenie protokołu *stanu łącza* (OSPF)
- *typ* adresu rozsyłania grupowego dodany do IPv4

rozsyłanie grupowe na podstawie stanu łącza

- każdy ruter monitoruje stan łączy do niego *bezpośrednio* do niego podłączonych i nadaje komunikat aktualizujący do *wszystkich* ruterów, ilekroć stan się zmienia
- komputer informuje sieć LAN, *do której grupy należy*
- ruter monitoruje sieć LAN, aby to sprawdzić
- brak informacji oznacza, że komputer *opuścił grupę*
- posiadając wiedzę, które grupy mają członków na których łączach, każdy ruter oblicza *drzewo rozsyłania grupowego o najkrótszych ścieżkach* od dowolnego nadawcy do dowolnej grupy (algorytmem Dijkstry)

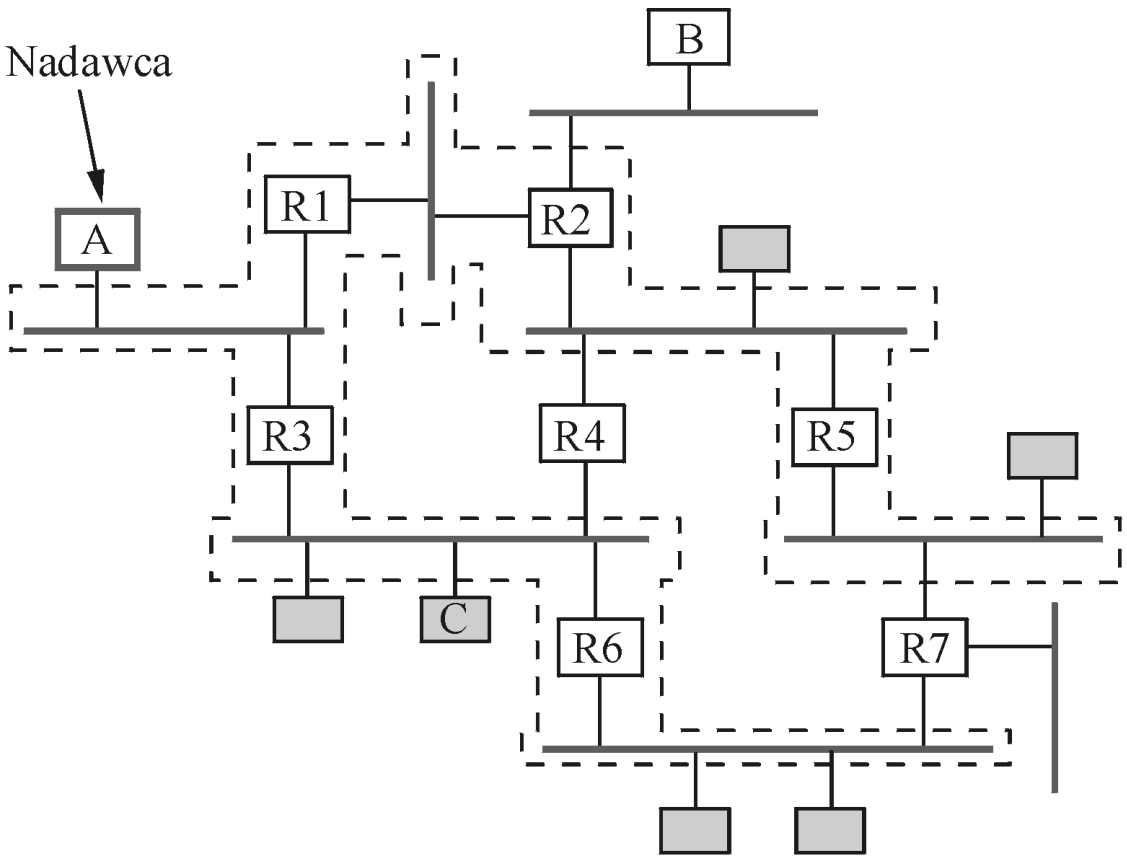
rozsyłanie grupowe na podstawie stanu łącza

- komputery oznaczone na szaro należą do grupy G

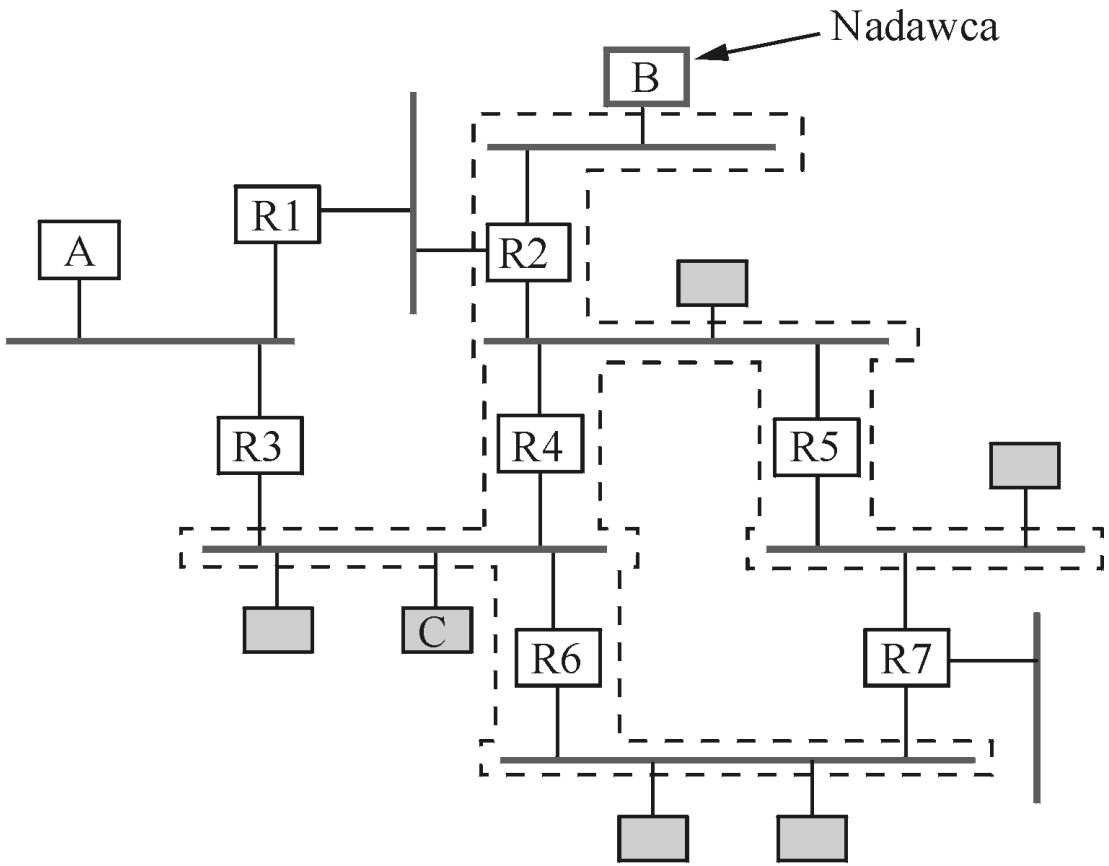


zadanie: oblicz drzewo rozsyłania grupowego o najkrótszych ścieżkach dla nadawców A, B i C do grupy G

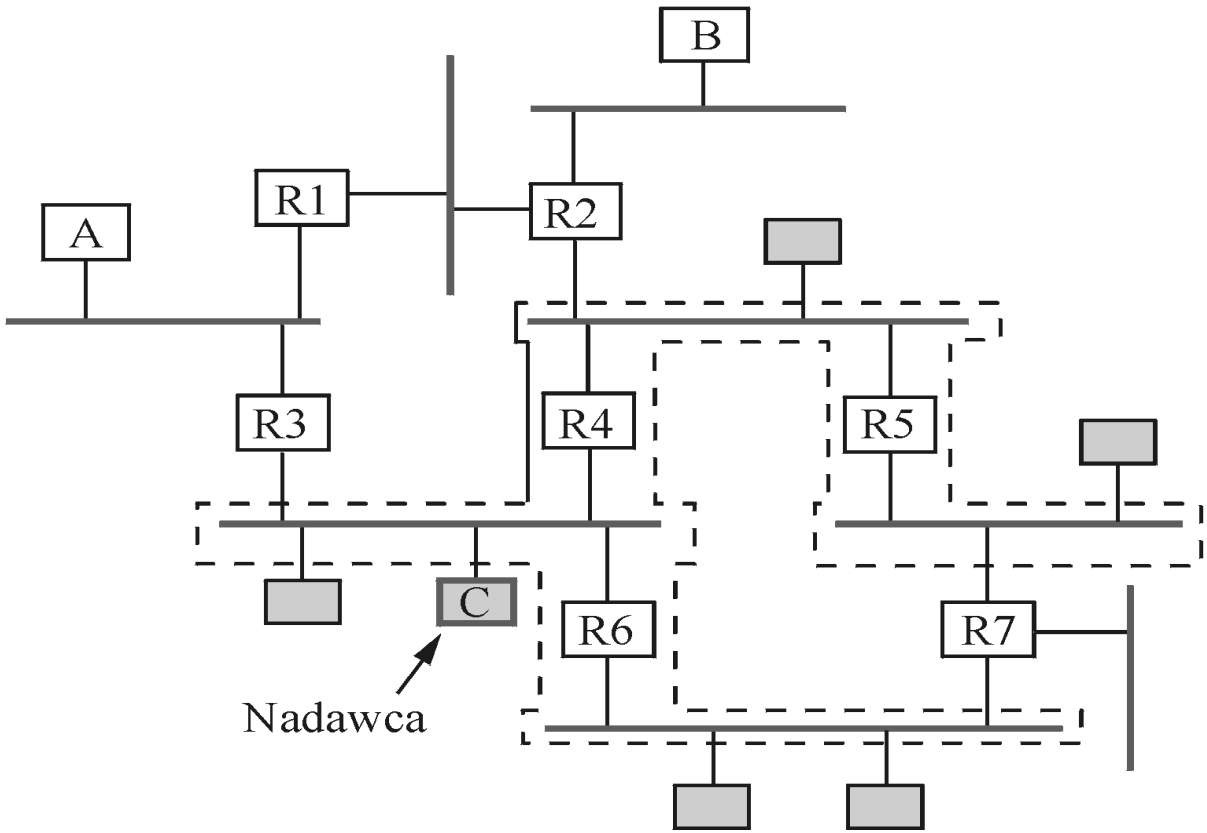
drzewo rozsyłania grupowego o najkrótszych ścieżkach dla nadawcy A



drzewo rozsyłania grupowego o najkrótszych ścieżkach dla nadawcy B



drzewo rozsyłania grupowego o najkrótszych ścieżkach dla nadawcy C



rozsyłanie grupowe na podstawie wektora odległości

- *dodanie rozsyłania grupowego do algorytmu wektora odległości* jest skomplikowane, ponieważ rutery nie znają całej topologii sieci
- realizowane *w dwóch etapach*:

1º mechanizm rozgłaszania, kierujący pakiet do wszystkich sieci w intersieci, nazwany *rozgłaszaniem uwzględniającym ścieżkę odwrotną*

2º mechanizm odcinający te sieci, które nie zawierają komputerów należących do grupy, nazwany *rozsyłaniem grupowym, uwzględniającym ścieżkę odwrotną*

rozgłaszanie uwzględniające ścieżkę odwrotną

- kiedy ruter odbiera pakiet rozsyłania grupowego od nadawcy S, kieruje go do wszystkich łączy wychodzących, tylko wtedy, gdy pakiet przyszedł na łączu należącym do najkrótszej ścieżki do nadawcy S
- ruter należący do najkrótszej ścieżki do nadawcy S w danej sieci nazywa się *ruterem macierzystym*
- *tylko* ruter macierzysty może przekazywać pakiety rozsyłane grupowo od tego nadawcy do danej sieci
- ruter trzyma dla każdej pary *nadawca/łącze* po jednym bicie, wskazującym czy jest dla niej ruterem macierzystym

rozsyłanie grupowe uwzględniające ścieżkę odwrotną

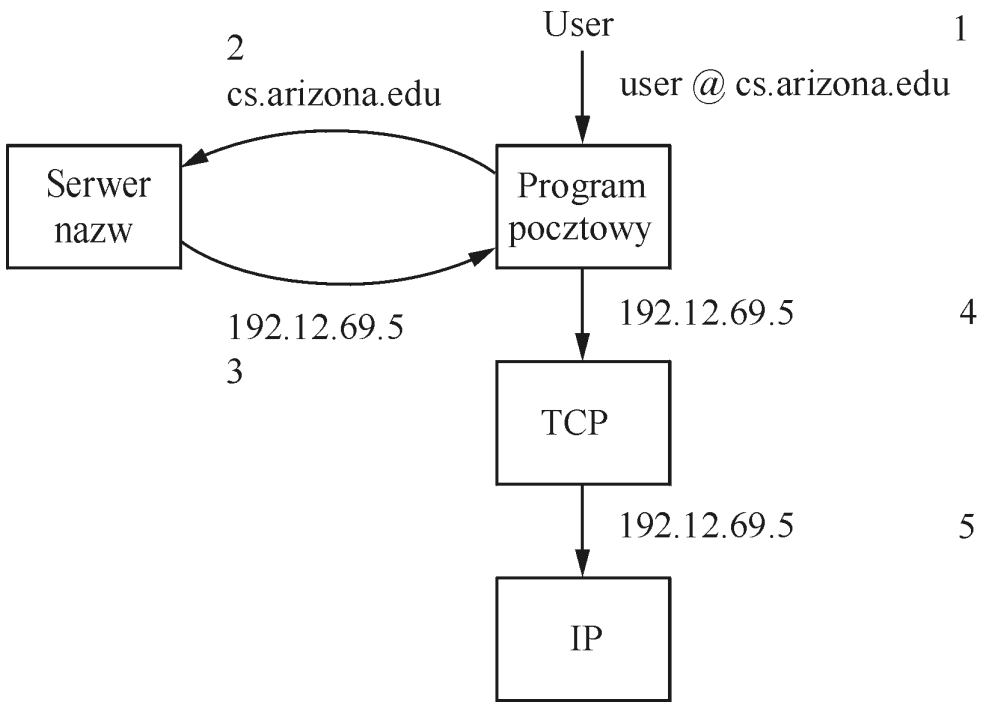
- *rozpoznanie*, kiedy *sieć typu liść* (w której ruter macierzysty jest jedynym ruterem) nie zawiera żadnych komputerów należących do grupy G, przez analizę w ruterze informacji przekazywanej okresowo przez komputery
- *propagacja informacji* „nie ma tutaj członków grupy G” *w górę drzewa najkrótszych ścieżek*, od rutera do rutera

nazwy komputerów (DNS)

- *nazwa komputera*: zmienna długość, mnemoniczna
- *przestrzeń nazw*: hierarchiczna (podzielona na składniki) albo prosta
- *system nazewnictwa*: zestaw wiązań nazw i wartości (w Internecie: *system nazw domen* (DNS))
- *wartość*: to, co zwraca system nazewnictwa, gdy mu podaje się nazwę (wartością może być adres)
- *mechanizm odwzorowania*: (procedura wywołana z nazwą zwraca wartość)
- *serwer nazw*: konkretna implementacja mechanizmu odwzorowania

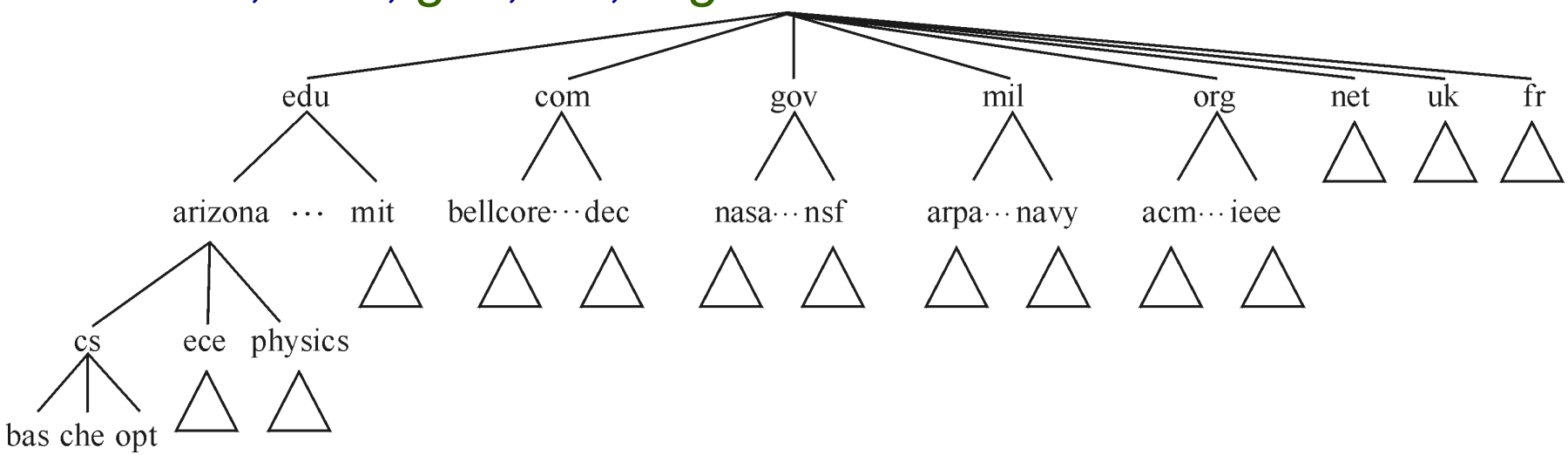
system nazw domen (DNS)

- *poprzednio* tablica wiązań nazw z adresami **`hosts.txt`** zarządzana przez NIC
- *teraz* DNS: sposób przekształcania nazwy w adres:



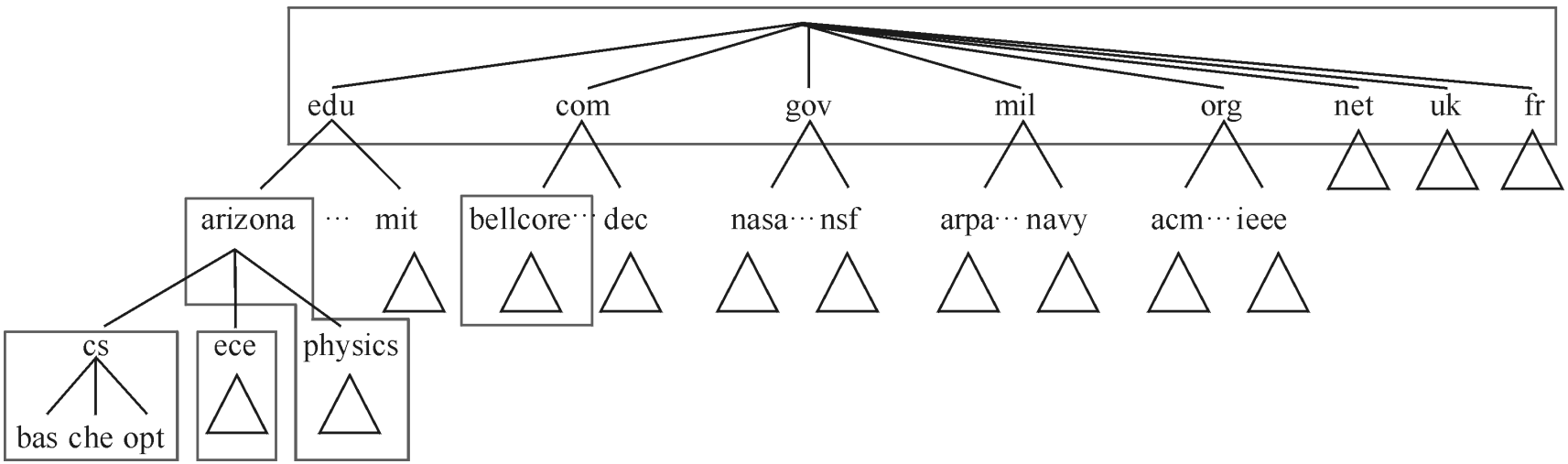
hierarchia domen

- nazwy DNS czytane od lewej, przetwarzane od prawej
- n.p.: che.cs.arizona.edu
- DNS odwzorowuje nazwy domen w wartości
- domeny dla każdego kraju + „wielka szóstka”:
  edu, com, gov, mil, org i net



serwery nazw

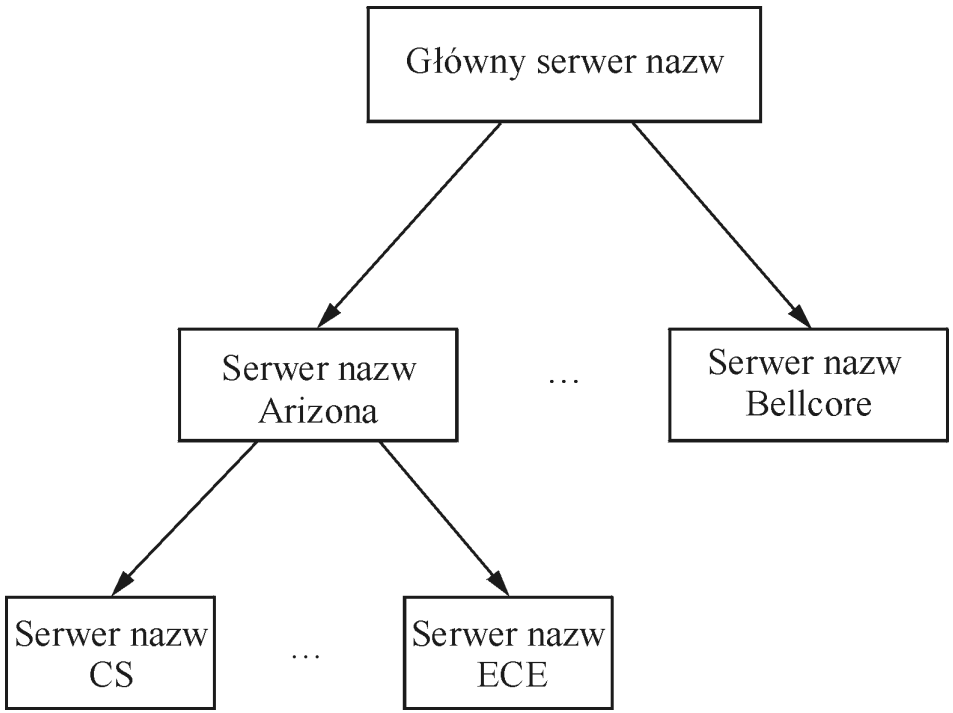
- hierarchia domen podzielona na strefy:



- *strefa* odpowiada zakresowi władzy administracyjnej i w DNS jest zarządzana przez jeden lub więcej *ser-wer nazw* (redundancja)
- górnym poziomem zarządza NIC

hierarchia serwerów nazw

- DNS - hierarchia serwerów nazw



implementacja serwera nazw

- *informacja o strefie*: zbiór rekordów zasobów:

  ⟨**`Nazwa, Wartość, Typ, Klasa, TTL`**⟩
- pole **`Wartość`** *zależne od typu*: typ A: adres IP

  typ NS: nazwa komputera, na którym jest uruchomiony serwer nazw

  typ CNAME: kanoniczna nazwa danego komputera (alias)

  typ MX: nazwa komputera, na którym uruchomiony jest serwer poczty elektronicznej dla domeny
- **`Klasa`**: inne typy rekordów (nie NIC), IN (Internet)
- **`TTL`**: czas życia zasobu

przykład głównego serwera nazw

- zawiera rekord NS dla każdego serwera z drugiego poziomu oraz rekord A do translacji nazwy w adres IP:
- ⟨arizona.edu, telcom.arizona.edu, NS, IN⟩
- ⟨telcom.arizona.edu, 128.196.128.233, A, IN⟩

- ⟨bellcore.com, thumper.bellcore.com, NS, IN⟩
- ⟨thumper.bellcore.com, 128.96.32.20, A, IN⟩

przykład serwera nazw na 2 poziomie

- domena arizona.edu posiada serwer nazw dostępny w komputerze telcom.arizona.edu, z rekordami:

  ⟨cs.arizona.edu, optima.cs.arizona.edu, NS, IN⟩

  ⟨optima.cs.arizona.edu, 192.12.69.5, A, IN⟩

  ⟨ece.arizona.edu, helios.ece.arizona.edu, NS, IN⟩

  ⟨helios.ece.arizona.edu, 128.196.28.166, A, IN⟩

  ⟨jupiter.physics.arizona.edu, 128.196.4.1, A, IN⟩

  ⟨saturn.physics.arizona.edu, 128.196.4.2, A, IN⟩

  ⟨mars.physics.arizona.edu, 128.196.4.3, A, IN⟩

  ⟨venus.physics.arizona.edu, 128.196.4.4, A, IN⟩

przykład serwera nazw na 3 poziomie

- domena cs.arizona.edu posiada serwer nazw dostępny w komputerze optima.cs.arizona.edu:

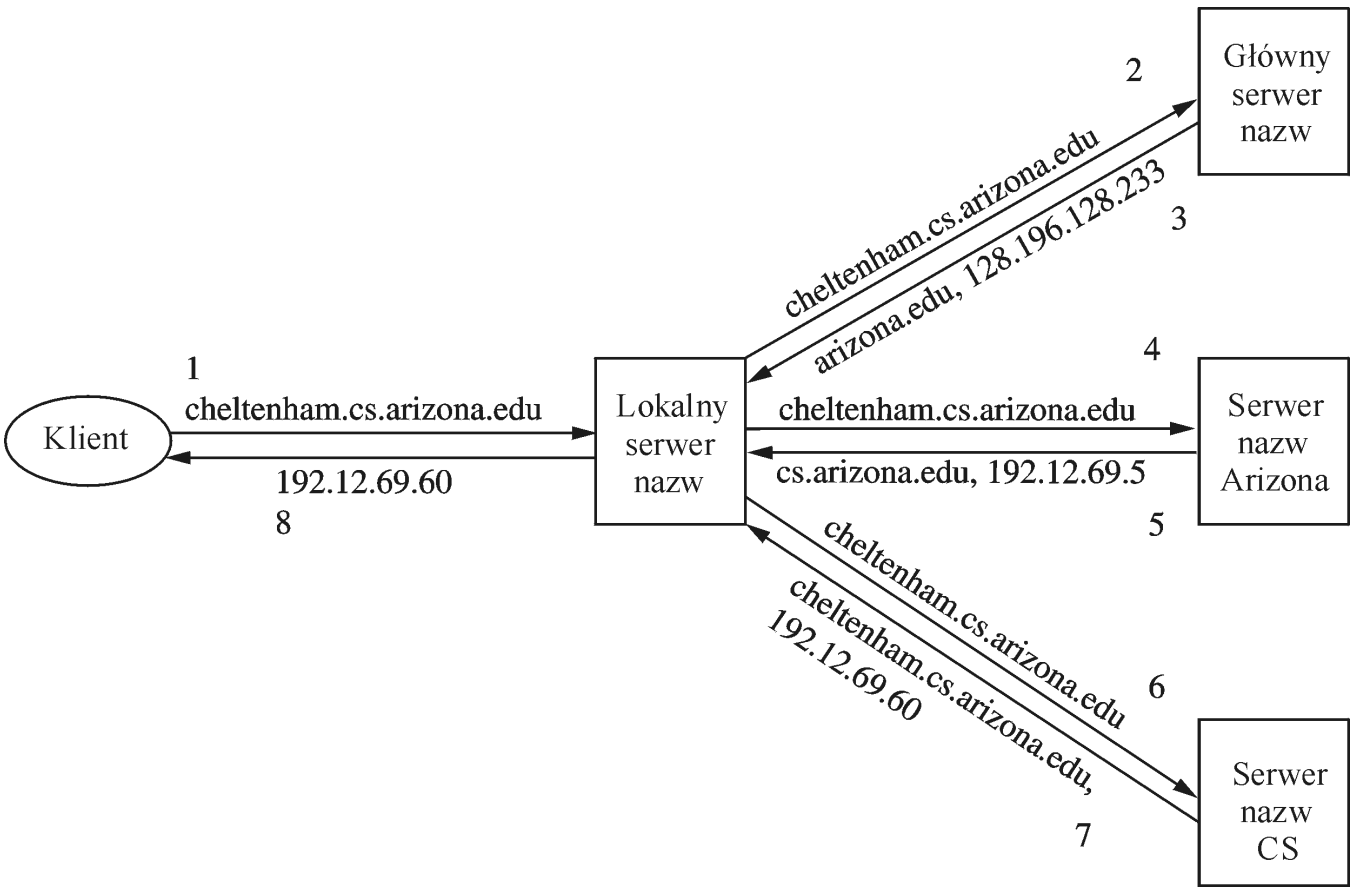
  ⟨cs.arizona.edu, optima.cs.arizona.edu, MX, IN⟩

  ⟨cheltenham.cs.arizona.edu, 192.12.69.60, A, IN⟩
  ⟨che.cs.arizona.edu, cheltenham.cs.arizona.edu, CNAME, IN⟩

  ⟨optima.cs.arizona.edu, 192.12.69.5, A, IN⟩
  ⟨opt.cs.arizona.edu, optima.cs.arizona.edu, CNAME, IN⟩

  ⟨baskerville.cs.arizona.edu, 192.12.69.35, A, IN⟩
  ⟨bas.cs.arizona.edu, baskerville.cs.arizona.edu, CNAME, IN⟩

odwzorowanie nazw w praktyce



konwencja nazewnictwa

- uniwersytety amerykańskie w domenie edu:
  cs.stanford.edu
- uniwesytety angielskie w poddomenie ac domeny uk:
  csl.cam.ac.uk
- uniwersytety polskie w poddomenie miasto, n.p.:
  cs.put.poznan.pl
  ale też w poddomenie edu domeny pl, n.p.:
  wmid.amu.edu.pl

IP następnej generacji

- *motywacja*: wzrost Internetu, problem skalowania
- *niemożliwe* jest uzyskanie 100% efektywności wykorzystania adresów
- *pierwsze prace* nad zwiększeniem przestrzeni adresów: IETF 1991
- *zmiana rozmiaru adresu do 128 bitów* $\Rightarrow$ zmiana w nagłówku pakietu IP $\Rightarrow$ nowe oprogramowanie w komputerach i ruterach
- *efekt kuli śniegowej* - lista życzeń IP nowej generacji:

  obsługa czasu rzeczywistego, bezpieczeństwo, autokonfiguracja (adresu IP i domeny), wybór trasy komputerów ruchomych

adresy w IPv6

- *długość adresu* 128 bitów (16 bajtów)
- *przestrzeń adresów* $2^{128} = 3{,}4 \times 10^{38}$
- 16000 efektywnych adresów na m^2

przydział przedrostków adresu IPv6

przedrostek	zastosowanie
0000 0000	zarezerwowany
0000 0001	nie przydzielony
0000 001	zarezerwowany dla alokacji NSAP (protokoły ISO)
0000 010	zarezerwowany dla alokacji IPX (Novell)
0000 011	nie przydzielony

przydział przedrostków adresu IPv6

przedrostek	zastosowanie
0000 1	nie przydzielony
0001	nie przydzielony
001	nie przydzielony
010	adres jednostkowy odnoszący się do dostawcy
011	nie przydzielony

przydział przedrostków adresu IPv6

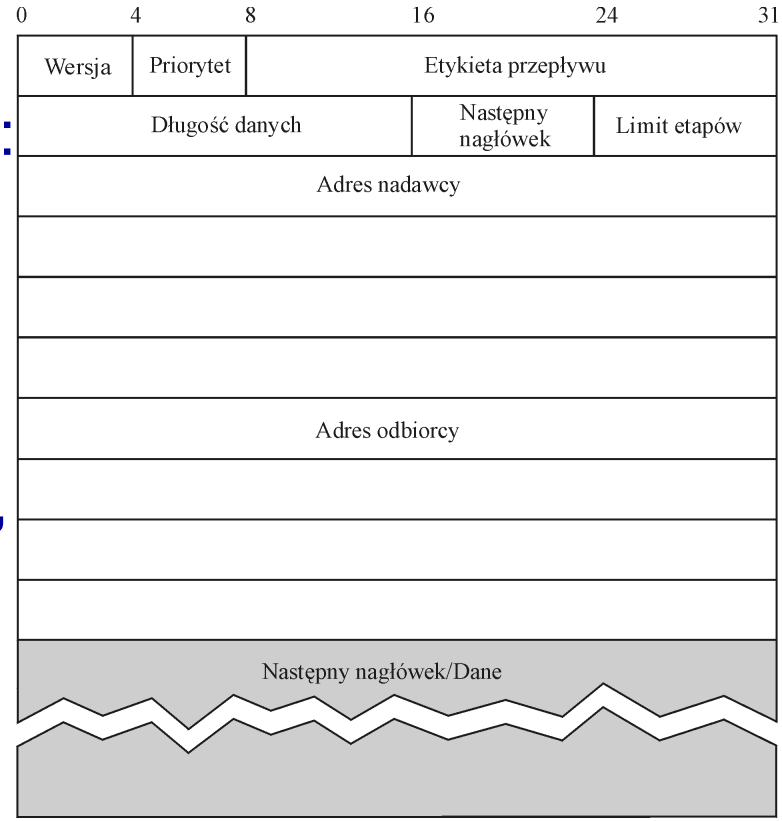
przedrostek	zastosowanie
100	zarezerwowany dla adresu jednostkowego odnoszącego się do obszaru geograficznego
101, 110,1110, 11110, 111110, 1111110, 111111100	nie przydzielony
1111 1110 10	adres wykorzystywany lokalnie w łączu
1111 1110 11	adres wykorzystywany lokalnie w miejscu
1111 1111	adres grupowy

notacja adresu

- x:x:x:x:x:x:x:x:, gdzie każdy x jest szesnastkową reprezentacją 16 bitowego (2 bajtowego) fragmentu:
  47CD:1234:4422:AC02:0022:1234:A456:0124
- zapis zwarty 47CD::A456:0124 adresu:
  47CD:0000:0000:0000:0000:0000:A456:0124
- adres IPv6 odwzorowany w IPv4:
  ::00FF:128.96.33.81

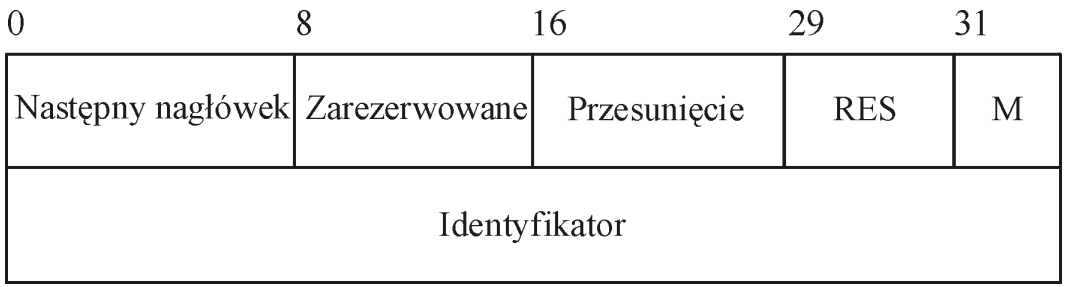
format pakietu IPv6

- wersja=6
- priorytet, etykieta przepływu: jakość usługi
- długość danych: (w bajtach)
- następny nagłówek: (po nagłówku IP), jak go nie ma, to pole protokół (TCP, UDP)
- limit etapów (to co TTL)
- dł. nagłówka 40 bajtów



przetwarzanie opcji

- *efektywniejsze* niż w IPv4 (przyjazne dla ruterów)
- *nagłówki rozszerzeń* w ustalonym porządku
- opcje dowolnej długości
- nagłówek rozszerzenia dotyczącego fragmentacji



- następny nagłówek:

  44=fragmentacja, 51=poświadczanie tożsamości, 6=TCP, i.t.p.

Autokonfiguracja adresu IP

- *podejście stanowe*: komputer rozmawia z serwerem konfiguracji
- *podejście bezstanowe*: komputer konstruuje swój adres IP na własną rękę: etapy:
  - uzyskanie identyfikatora interfejsu, unikalnego na łączu, do którego komputer jest dołączony (Ethernet)
  - uzyskanie przedrostka adresu dla podsieci (nadawany przez ruter)

3	m	n	o	p	125-m-n-o-p
010	Identifikator rejestru	Identyfikator dostawcy	Identyfikator abonenta	Identyfikator podsieci	Identyfikator interfejsu

zaawansowane możliwości wyboru trasy

- brak nagłówka wyboru trasy = CIDR w IPv4
- *nagłówek wyboru trasy*:
  - obszary topologiczne, które pakiet ma odwiedzić na trasie do odbiorcy
  - n.p. kręgosłupowa sieć dostawcy, sieć dostawcy taniego, niezawodnego, bezpiecznego ...
- *adres elementu topologicznego* (anycast) = zbiór interfejsów

przejście od IPv4 do IPv6

- trudny okres przejściowy
- mechanizmy przejścia:,
  - *operacja na podwójnym stosie* (IPv4 i IPv6 w węźle), adres IPv6 może być niezwiązany z adresem IPv4
  - *tunelowanie* (pakiet IPv6 nadany jako dane w pakiecie IPv4), stosowany może być adres IPv6 odwzorowany w IPv4