

SIK = Sieci komputerowe

Tematyka całego wykładu SIK:

- Całokształt zagadnień związanych z sieciami komputerowymi, struktura intersieci, sieci fizyczne, protokoły niższych i wyższych warstw, interfejs programistyczny sieci (gniazdko), budowa i historia Internetu, konfiguracja sieci pod linux-em, narzędzia do symulowania i analizowania sieci/protokołów,
...

Plan najbliższych wykładów:

- Zarys sieci komputerowych
sieci fizyczne, intersieć, adresy węzłów, routery,
prot niższych warstw: fiz., IP, TCP, UDP,
prot wyższych warstw: FTP, HTTP, SSL, DNS, mailowe, ...

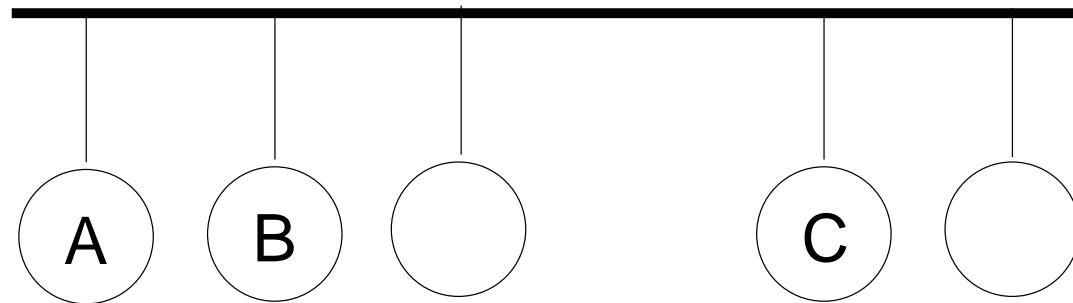
Literatura

- Comer, "Sieci komputerowe TCP/IP, tom 1", (stara książka)
- Kurose, Ross, "Sieci, od szczegółu do ogółu z internetem w tle", (nowa książka)
- Dordal, "An Introduction to Computer Networks", (pdf)
- dokumenty RFC: <https://tools.ietf.org/html/>
- materiały w wikipedii ...

Intersieć i sieci fizyczne

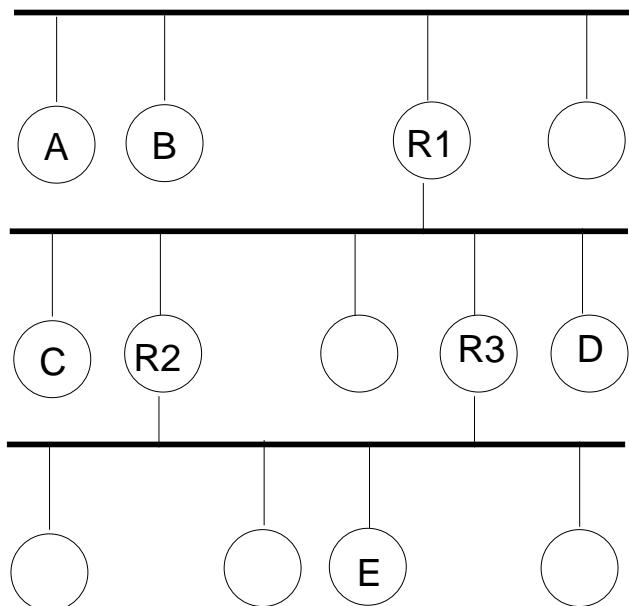
- Sieć fizyczna umożliwia wysyłanie pakietów między węzłami.
- Komputer podłączony do sieci to "węzeł".
inne nazwy węzła: host, maszyna
- Węzeł A może wysłać pakiet do węzła B (ang. unicast)
lub do wszystkich w tej samej sieci fizycznej (ang. broadcast)
- Pakiet = nagłówek + dane; inne nazwy: ramka, datagram, komunikat
nagłówek pakietu zawiera m.in. adresy sprzętowe źródłowy i docelowy...
- Interfejs sieciowy węzła (= karta sieciowa) posiada adres sprzętowy (np. eth)
patrz polecenie ifconfig ...

Sieć fizyczna:



Intersieć i sieci fizyczne

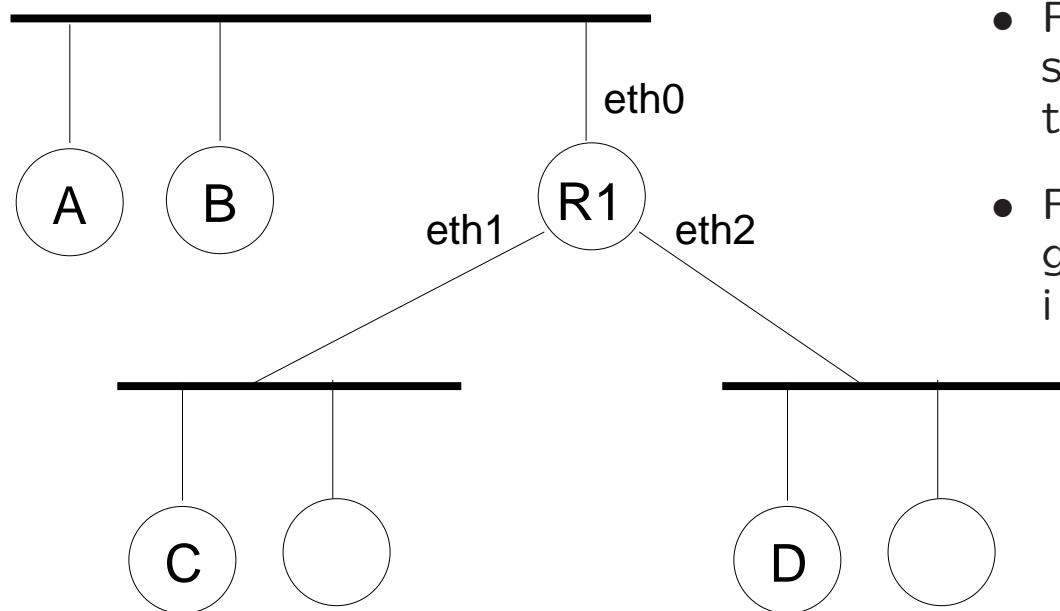
Intersieć:



- Intersieć składa się z kilku sieci fizycznych
- Przykład intersieci: Internet...
- Węzeł A może wysłać pakiet do węzła E, wtedy pakiet przeskakuje przez 2 routery, np. R1 i R2 lub R1 i R3
- Router = węzeł podłączony równocześnie do kilku sieci fizycznych, przekazuje pakiety
- Linux może działać jako router...
- "wszystkie sieci są równe"
- w Internecie: podział intersieci na AS (systemy autonomiczne) z uwagi na złożoność...

Intersieć i sieci fizyczne

Intersieć:



- Router R1 jest podłączony równocześnie do 3 sieci fizycznych
- Router decyduje przez który interfejs sieciowy wysłać pakiet to tzw. trasowanie (ang. routing).
- Router robi to na podstawie tablicy routingu i adr docelowego pakietu (adr IP)

Typy sieci fizycznych

- LAN - Sieć lokalna (ang. Local Area Network)
WAN - Sieć WAN (z ang. Wide Area Network, rozległa sieć komputerowa)
WLAN - Bezprzewodowa LAN (ang. Wireless Local Area Network)
MAN - Miejska sieć komputerowa (ang. Metropolitan Area Network)
- topologia sieci: magistrala, gwiazda, ring, drzewo - dotyczy sieci fizycznej!!
- sieć lokalna typu **Ethernet**
 - standard IEEE 802.3, Ethernet II
 - skrętka (kabel) + switch (urządzenie sieciowe)
 - skrętka nieekranowana kategori e5, do 100metrów, 100Mbit/s = Fast Ethernet, są jeszcze szybsze!, wtyczka RJ-45, nie przejmować się przeplotem!
 - maszyny w sieci mają karty sieciowe Ethernet, >2 maszyny łączymy przy pomocy switch-a => topologia gwiazdy/drzewa
 - dawniej używano kabla koncentrycznego lub skrętki + hub (koncentrator/repeater)
- bezprzewodowa sieć lokalna typu **WiFi**
 - standardy IEEE 802.11, 802.11b/g/n
 - punkt dostępowy (ang. Access Point), klienci WiFi (czyli maszyny z kartami sieciowymi WiFi)

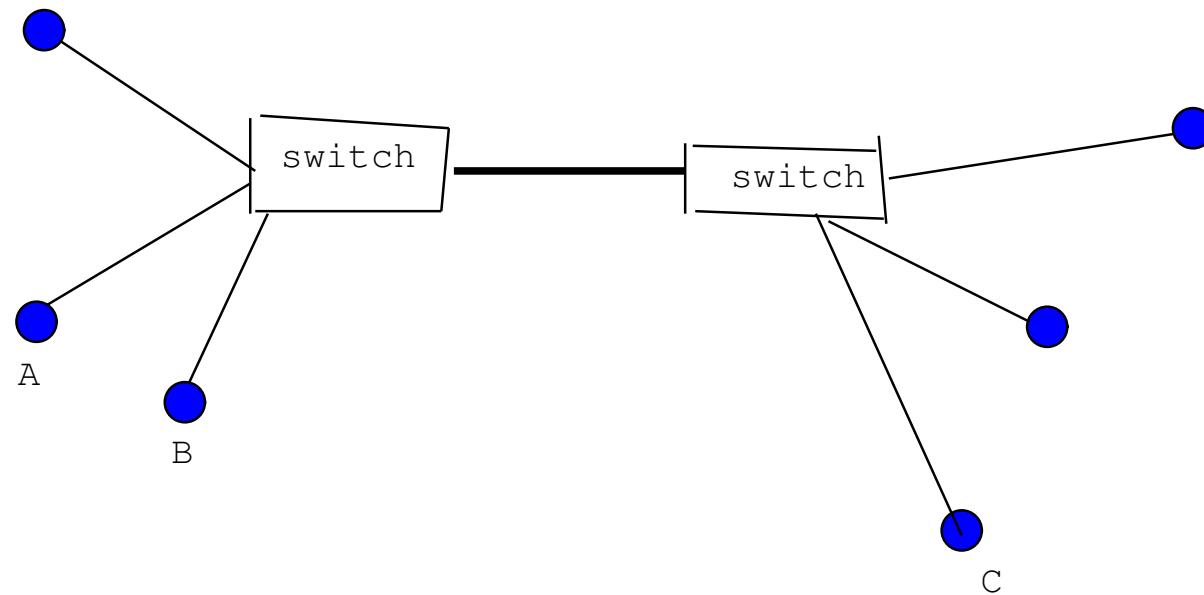
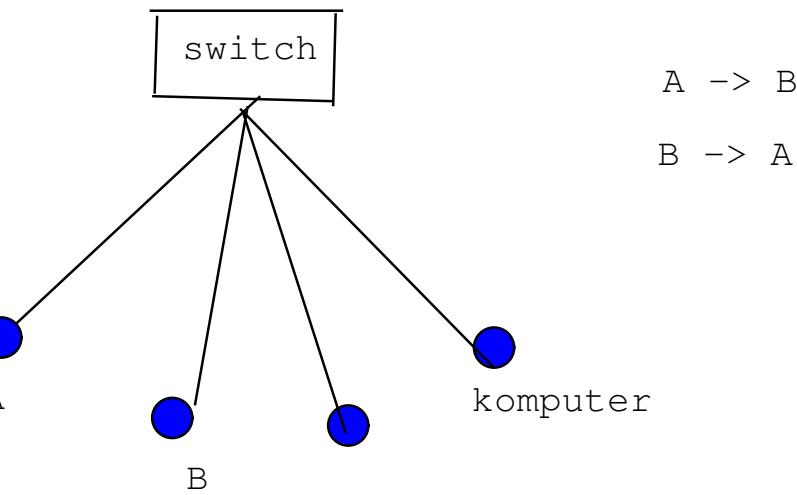
- sieci dwuwęzłowe nad łączem szeregowym
 - połączenie telefonicze, modemy telefoniczne/akustyczne, 56Kbitów/s, bardzo długie łącze szeregowe
 - prot PPP (ang. Point to Point Protocol) przenosi pakiety IP nad łączem szeregowym (demony pppd + moduł kernelowy)
 - to jest prosty przykład sieci WAN !
 - łącze szeregowe może być emulowane (nad bluetooth/rfcomm lub nad USB)
- bezprzewodowa "personalna" sieć **Bluetooth**
 - standardy IEEE 802.15
 - prot rfcomm (łącze szeregowe nad bt), profile (usługi): DUN, PAN, OBEX, ...
- technologie sieci dostępowych (dostęp do Internetu):
ADSL (kabel tel, szerokopasmowy),
HFC ("kablówka", światło + kabel koncentryczny),
FTTH (GPON, światłowód)

Ethernet - C.D.

RJ-45 (8P8C),
skrętka:



- switch - pol. przełącznik, operuje na ramkach ethernetowych, dawniej bridge (pol. most), ma kilka "portów" (gniazdek RJ-45)
- switch-e można w prosty sposób łączyć kablem typu skrętka, tworząc "drzewo"
- **zasada działania switch-a:** jeśli nie wie gdzie wysłać ramkę eth, to wysyła wszędzie; poza tym dla każdego portu (gniazdka RJ-45) pamięta jakie adresy eth się za nim kryją ...
- switch vs router ?!?!?!
- tzw "routery WiFi" zawierają switch + access point WiFi (połączone), pojedyncza sieć fizyczna ...



A → B

B → A

Adresy węzłów

właściwie nie węzłów tylko interfejsów sieciowych węzłów...

typy adresów: sprzętowe, IP, domenowe

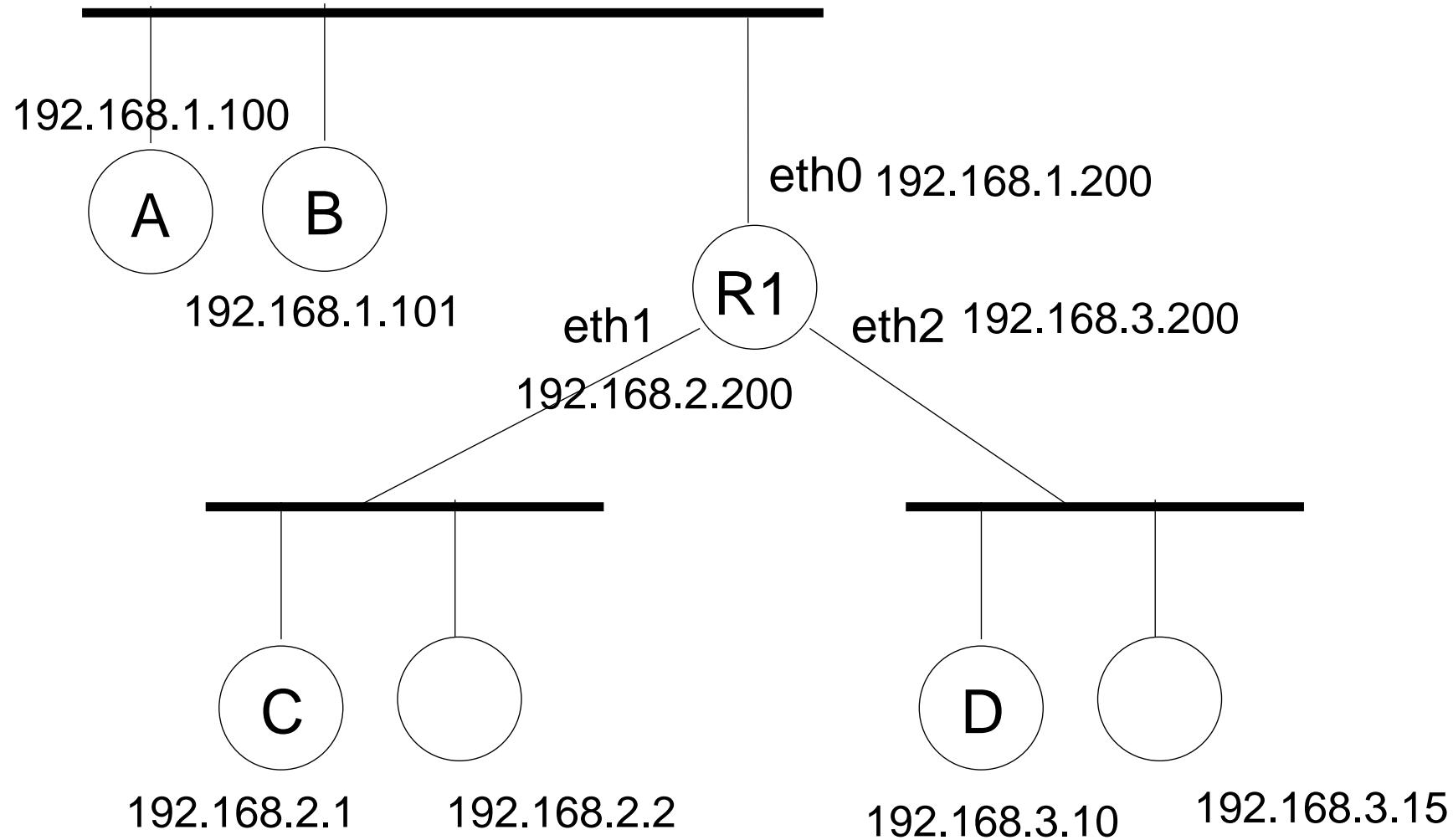
- adresy sprzętowe
np. ethernetowe, 08:9E:01:1C:9C:70, inna nazwa: adr MAC
nadawane przez producenta karty sieciowej
- adresy IP
np. 192.168.1.100, 150.254.77.44, $4 \times 8 = 32$ bity (IPv4),
przydzielanie adresów IP do interfejsu sieciowego: ręcznie, DHCP, ...
- adresy domenowe
np. wp.pl, onet.pl; serwer DNS zamienia adr domenowy na adr IP

zasady przydzielania adresów IP:

- adres IP składa się z "nr sieci" (prefiks) i z "nr hosta"
- wszystkie węzły w danej sieci fizycznej powinny mieć ten sam "nr sieci"
oraz inne "nr hosta"
- wniosek: węzły intersieci mają różne adresy IP
(uwaga na NAT - to jest wyjątek !!!)
- jeśli węzeł należy do kilku sieci to będzie miał kilka adr IP

- które bity adresu IP są nr sieci, a które nr hosta?
to zależy od "klasy adresu" (dawniej) i "maski podsieci" (obecnie)
- klasa adresu X1.X2.X3.X4; decyduje prefiks bajtu X1 w zapisie binarnym!
 - klasa A: 0..., nr sieci to X1
 - klasa B: 10..., nr sieci to X1.X2
 - klasa C: 110..., nr sieci to X1.X2.X3
 - klasa D: 1110..., multicasting
- maska podsieci
 - określa jawnie które bity adresu IP są nr sieci (jedynki w masce)
 - np. maska 255.255.255.0 dla adresu IP klasy B oznacza, że nr sieci to X1.X2.X3
 - jedynki w masce nie muszą koniecznie być spójne ani obejmować całych bajtów
 - wszystkie hosty w danej sieci fizycznej powinny mieć tą samą maskę
- adresy specjalne
 - nr hosta same jedynki: broadcast
 - nr host same zera: "ten host" (2 nr hosta są zakazane !!)
 - 127.0.0.1 = local loopback, localhost, lokalna maszyna
 - adresy prywatne (gdy nie mamy przydzielonego nr sieci w Internecie)
192.168.0.0 -> 192.168.255.255
10.0.0.0 -> 10.255.255.255
172.16.0.0 -> 172.16.255.255
- FLSM vs VLSM (Fixed/Variable Length Subnet Mask)
jeden nr sieci nie powinien być prefiksem drugiego ?!?!
adr pryw: FLSM, adr pub VLSM dla oszczędności
(pokazać rysunek)

Intersieć z przypisanym adresami IP (klasy C):



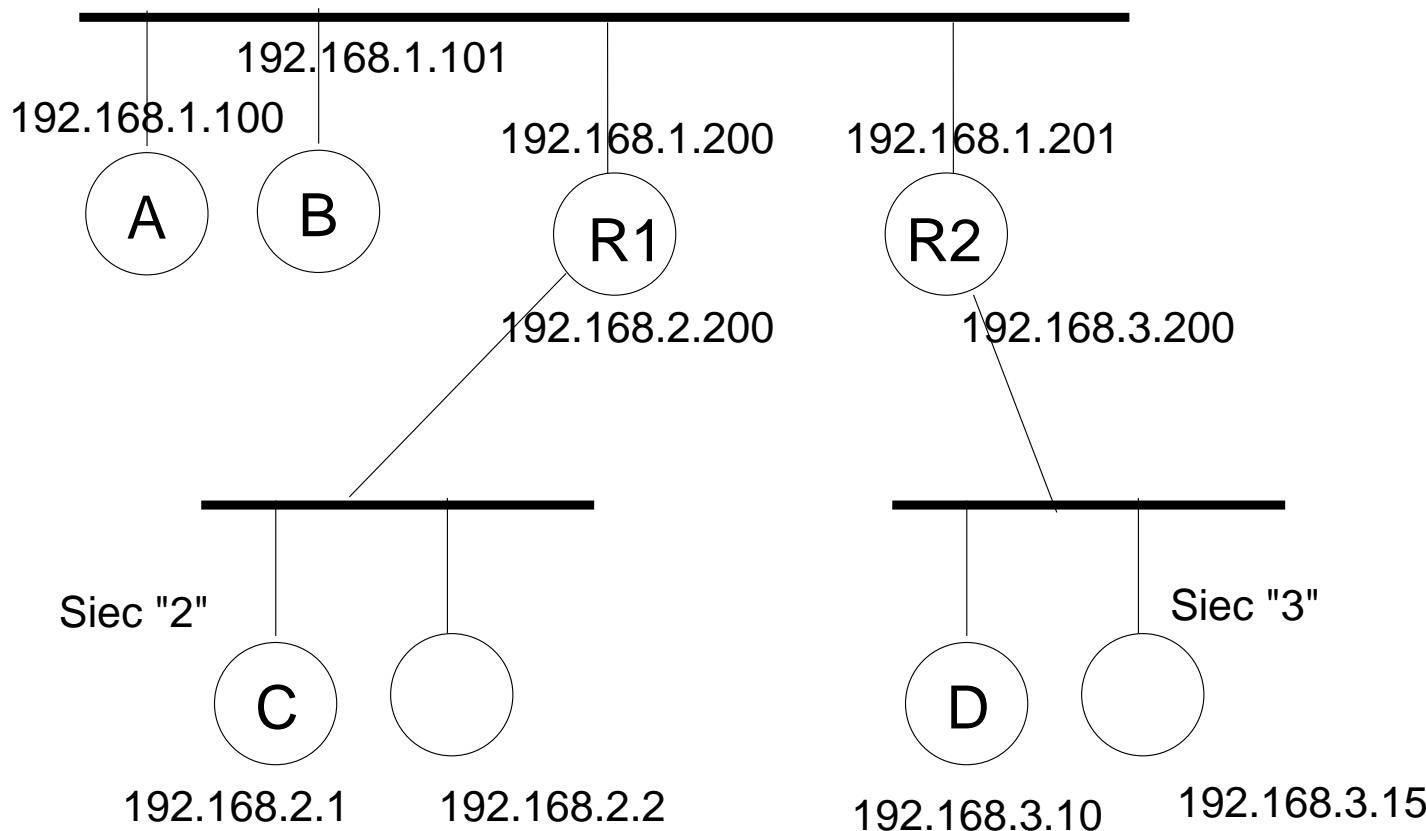
Intersieć z przypisanyem adresami IP (klasy C):

pojęcie sieci "bliskiej" i "dalekiej"...

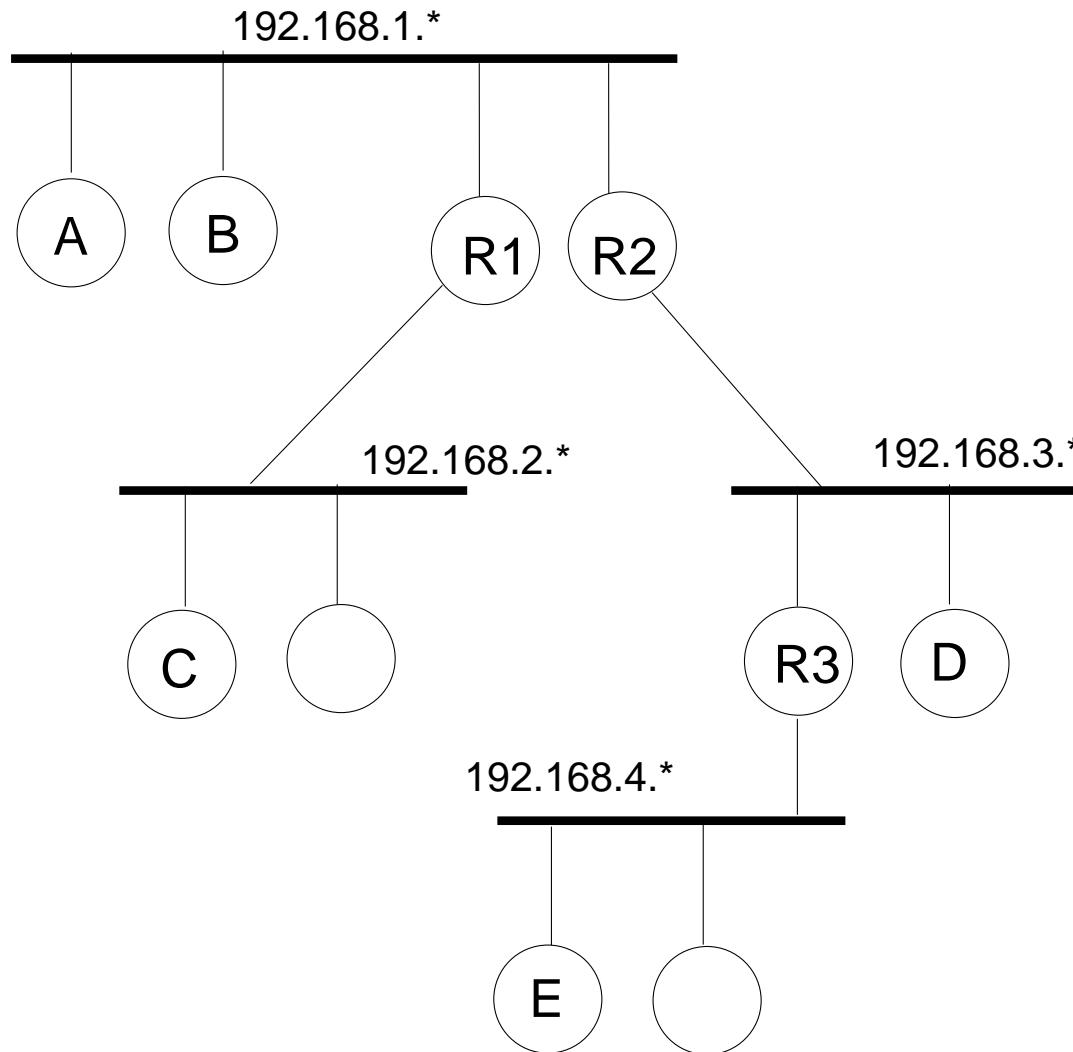
sieć bliska to ta, do której jesteśmy bezpośrednio podłączeni,

węzeł A musi wiedzieć, przez który ruter (gateway) dostać się do sieci "2" i "3" !!

w tabl. routingowej są: reg. dla sieci bliskich, dalekich, default gw



jakie wpisy muszą być w tablicach routingowych węzłów i routerów ?



"Net-tools" = polecenia liniuxa do konfigurowania sieci

ifconfig - konfigurowanie interfejsu sieciowego

route - odczyt/modyfikowanie tablicy routingowej

iptables - filtrowanie pakietów, zapora + NAT

ping, traceroute - testowanie sieci

dhcpcd - automat konfig interfejsu sieciowego

(patrz http://150.254.78.111:20002/zajecia/_xowiki2/SIK_cw Tematy A i F)

Protokoły IP, TCP, UDP

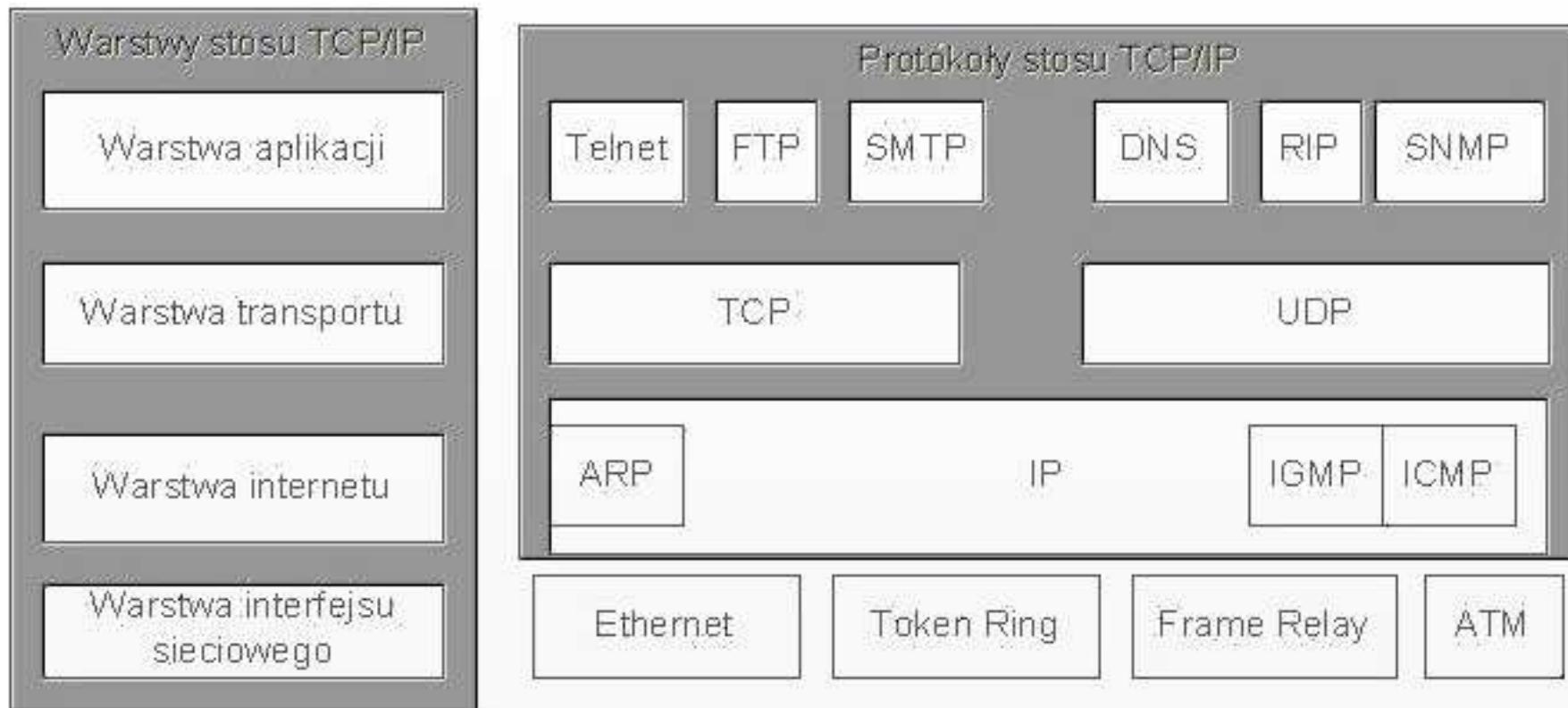
- Co to jest "protokół" ?
 - sposób w jaki hosty rozmawiają przez jakiś kanał komunikacyjny
 - m.in. definiuje format komunikatów
- prot IP - warstwa internetowa
 - przenoszenie pakietów IP przez intersiec
- prot UDP - warstwa transportowa
 - przenoszenie datagramów UDP (są nr portów)
 - niepewne
- prot TCP - warstwa transportowa
 - (wirtualne) połączenie TCP
 - można przesyłać strumien danych/bajtów
 - jest pewne
- aplikacje używają prot warstwy transportowej za pomocą gniazdek BSD (API, fun. systemowe)

Warstwy protokołów

Architektura warstwowa:

wyższa warstwa używa niższej warstwy (patrz enkapsulacja)

Podział protokołów Tcp/Ip na warstwy:



Warstwy ISO vs warstwy Tcp/Ip:



nieoczywiste warstwy ISO:

fizyczna (kabel)

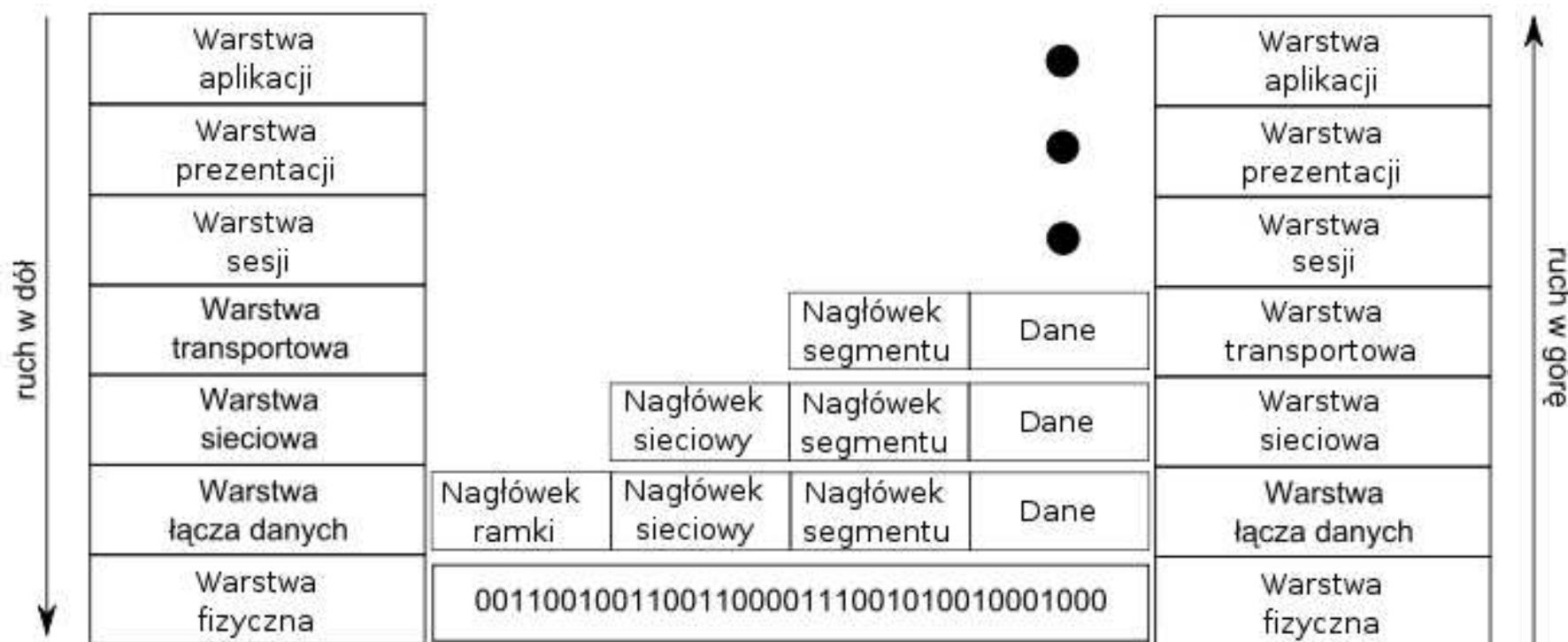
łącza danych (ramka, suma kontrolna, ack(?))

sesji (terminal, identyfikacja użytkownika)

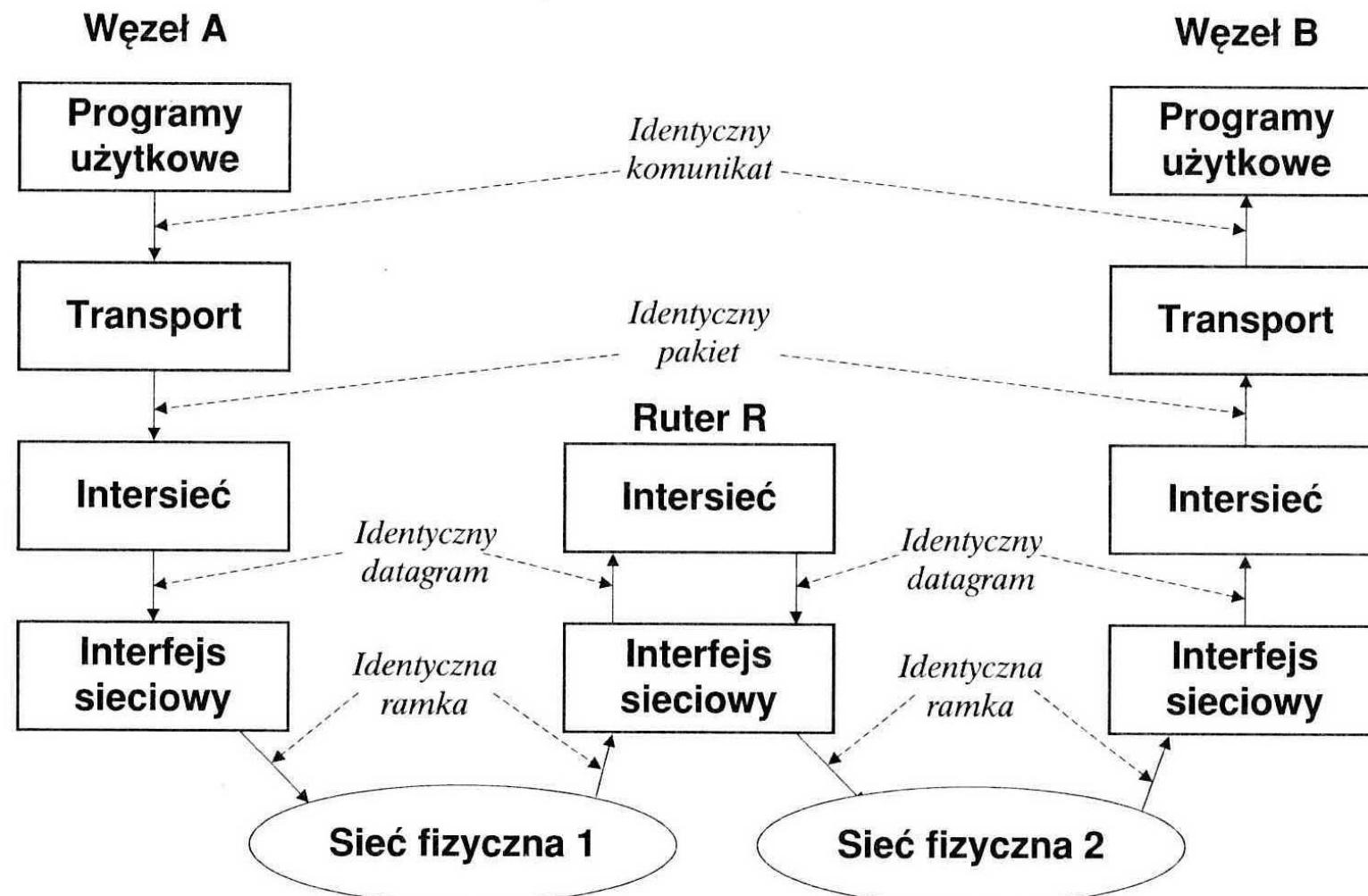
prezentacji (format danych, np. xdr, json, ASN.1)

aplikacji (mail, ftp, ...)

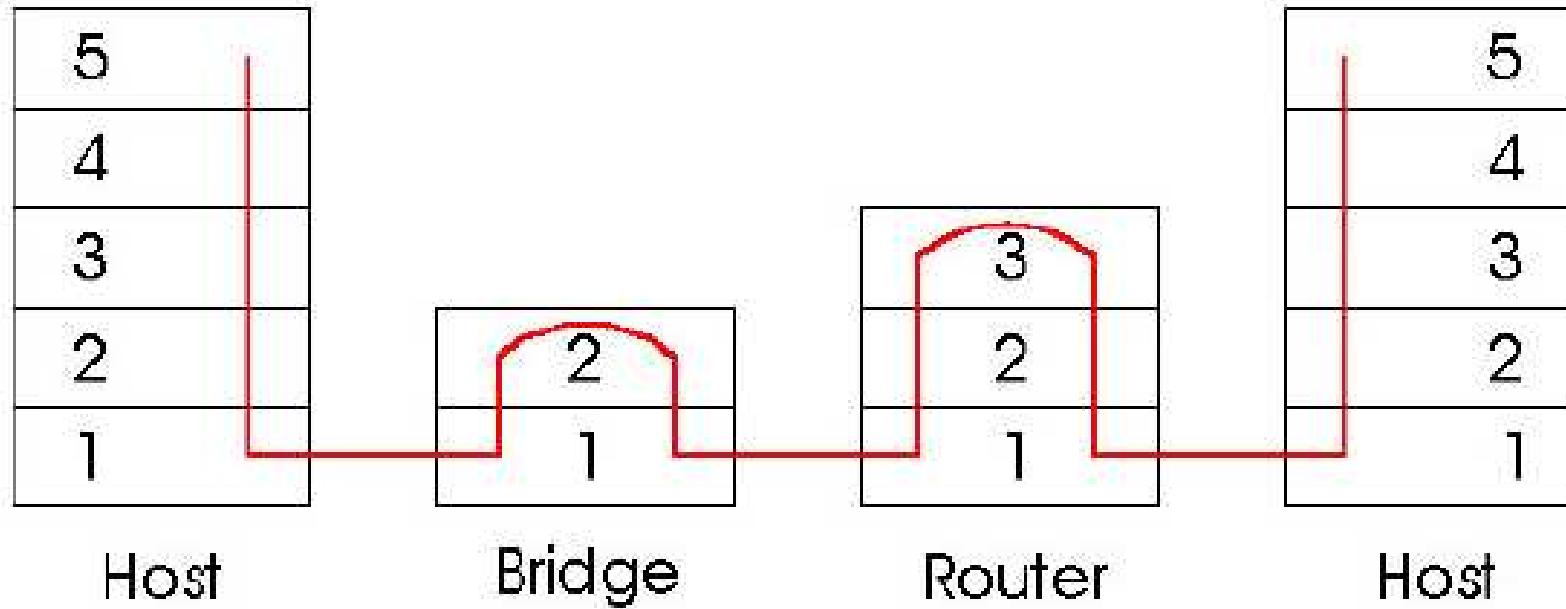
Warstwy ISO / enkapsulacja:



Jak pakiet przechodzi przez warstwy wędrując przez intersieć...



ile sieci fizycznych jest na tym rysunku?



Protokoły niskopoziomowe

- ARP (ang. Address Resolution Protocol)
 - zamiana adresu IP na sprzętowy
 - tablica/ cache ARP, zawiera pary (adres IP, adres sprzętowy)
 - zasada działania: broadcast sprzętowy z poszukiwanym adr IP
 - przenoszony w ramce eth
- DHCP (ang. Dynamic Host Configuration Protocol)
 - przydzielanie adresu IP dla interfejsu sieciowego hosta, oraz inne sprawy: maska, default router, serwery DNS
 - w datagramie UDP
- ICMP (ang. Internet Control Message Protocol)
 - zastosowania: jest ich wiele; powszechnie znane: echo/ polecenie ping komunikaty icmp posiadają pole TYP, KOD i inne w zależności od TYP:
TYP=8 TYP=0 echo pytanie/odp,
TYP=11 ttl spadł do 0 (ma zastosowanie w traceroute !)
TYP=4 tłumienie nadawcy, wysyłany przez przeciążony router
TYP=13 TYP=14 pytanie o czas i odp
TYP=17 prośba o maskę
 - w pakiecie IP

Protokoły warstwy aplikacji

- FTP, TELNET, DNS, HTTP, SMTP, POP3/IMAP, ...
- model klient-serwer; usługa, klient, serwer (świeradczy usługę), klient rozmawia z serwerem przy pomocy powyższych prot
- "nr portu", wprowadzony w TCP i UDP, serwer oczekuje na klientów na danym nr portu, np. FTP - 21, patrz /etc/services, wiele serwerów na jednej maszynie
- FTP - przesyłanie plików
TELNET, SSH - terminal do zdalnej maszyny
DNS - zamiana adresów domenowych na IP i odwrotnie
HTTP - strony www, rozmowa między przeglądarką a serwerem www
SMTP, POP3/IMAP - prot mailowe, wysyłanie/odbieranie maili ze skrzynki
...

Protokół IP

Nagłówek pakietu IP:

0 - 3	4 - 7	8 - 15	16 - 18	19 - 23	24 - 31					
Wersja	IHL	Typ usługi	Długość całkowita							
Identyfikator		Flagi		Przesunięcie fragmentu						
Czas życia	Protokół		Suma kontrolna nagłówka							
Adres źródłowy										
Adres docelowy										
Opcje		Dopełnienie								
Dane										

- pakiet IP zawiera adresy IP wezłów: źródłowego (src) i docelowego (dst)
- "czas życia", TTL, Time To Live, ile raz może przeskoczyć przez router
- IHL - długość nagłówka pakietu IP (w słowach 32bit)
- fragmentacja, gdy długość pakietu > MTU sieci fizycznej (max długość ramki)

- opcje IP - rozszerzenie IP ? różne ciekawe zastosowania...
 - opcje wydłużają nagłówek pakietu ip
 - opcje mają "klasę" i "nr"
 - opcje zajmują ≥ 1 bajt (w zależności od nr opcji)
 - klasa=0 - kontrola pakietów i sieci
 - klasa=2 - pomiary
 - nr opcji=3 i 9 - swobodne/rygorystyczne trasowanie wg nadawcy (nadawca podaje listę adr ip)
 - nr opcji=7 - zapisuj trasę (w pkg jest miejsce na pewną liczbę adr ip)
 - nr opcji=4 - zapisywanie czasów wzdłuż ścieżki

Protokół UDP

- nagłówek datagramu UDP zawiera nr portu źródłowy i docelowy
- datagram UDP jest transportowany w pakiecie IP
- broadcasting , "jeden do wielu", jedynki jako nr hosta
- multicasting, "jeden do wielu", przeskakiwanie przez routery (TTL), adres docelowy ip klasy D, grupy multicastowe
- do czego służą nr portów ? (na przykładzie udp)
2 procesy na 1 maszynie oczekują na datagram udp, jak je odróżnić??
odp: przy pomocy (różnych) nr portów...
- (prot, adr IP src, port src, adr IP dst, port dst)
jeśli prot=UDP to to jest "powiązanie"
jeśli prot=TCP to to jest "połączenie"
te liczby są zawarte w nagłówku pakietów i pozwalają odróżnić połączenia/powiązania!

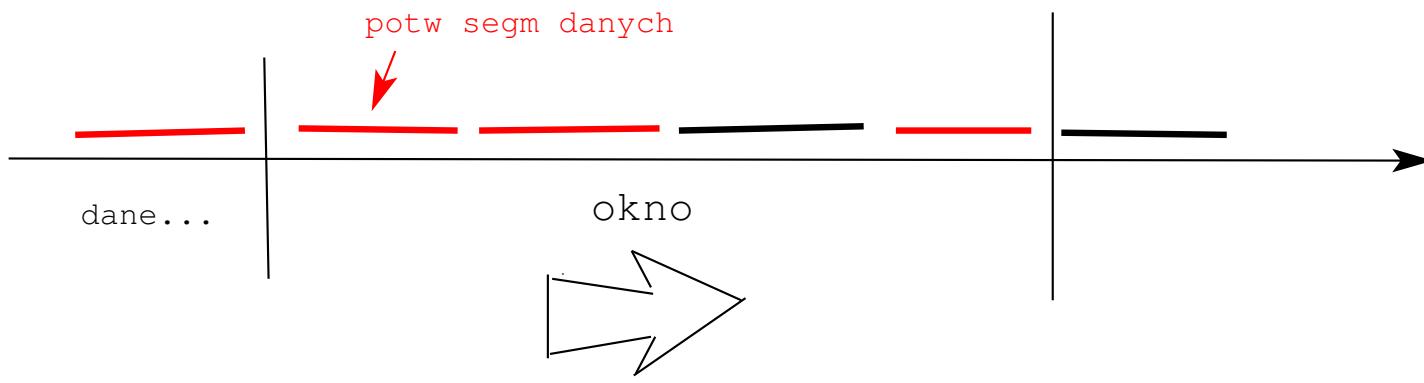
Protokół TCP

Dygresja na temat łączy (nie)nazwanych unix-a:

- łącza służą do komunikacji między dwoma procesami na jednej maszynie
- łącze to rozwiązanie "problemu producenta i konsumenta" (kontrola przepływu)
- prawa rządzące łączem ...
patrz <http://mhanckow.students.wmi.amu.edu.pl/sop322B.htm>
- połaczenie TCP zachowuje się dokładnie tak jak łącza !!!

Cechy połączenia TCP:

- połączenia TCP są pewne (dane się nie gubią - w przeciwieństwie do UDP ...)
- podobnie jak w UDP, używa się nr portów;
serwer oczekuje na klientów na danym nr portu
- połączenia TCP są dwukierunkowe
- koñczenie / zrywanie połączenia (fun. sys. close(desk) vs problemy sieciowe)
- implementacja połączenia TCP:
segmenty TCP, wysyłanie z potwierdzaniem,
przesuwające się okno z segmentami (ang. sliding window),
kontrola przepływu za pomocą zmiany rozmiaru tego okna



Gniazda BSD

- patrz <http://mhanckow.students.wmi.amu.edu.pl/sop322D.htm>
pokazać "dziedzine internetową/ gniazdka strumieniow"
pokazac dziedzine internetową/ gniazdka datagramowe
- gniazda BSD w językach skryptowych/ dynamicznych (język Tcl)
pokazać zachowanie połączenia TCP ...
- rola nr portu w połączeniach TCP
(zwł. po stronie serwera, gniazdko passywne i gniazdka aktywne)
rola nr portu w datagramach UDP

```
## serwer (w j. Tcl)
#
socket -server obsluga 10000

proc obsluga {s args} {
    puts "obsluga: podlaczył sie $s"
    fileevent $s readable "obslugaKli $s"
}
proc obslugaKli s {
    if {[eof $s]} { puts "obslugaKli: close $s"; close $s; return }
    set linia [gets $s]
    puts "obslugaKli: linia od $s, $linia"
    puts $s "odp na $linia"; flush $s
}

## klient
#
set s [socket localhost 10000]

puts $s "A ku ku !!!"; flush $s
gets $s
#% odp na A ku ku !!!

close $s
```

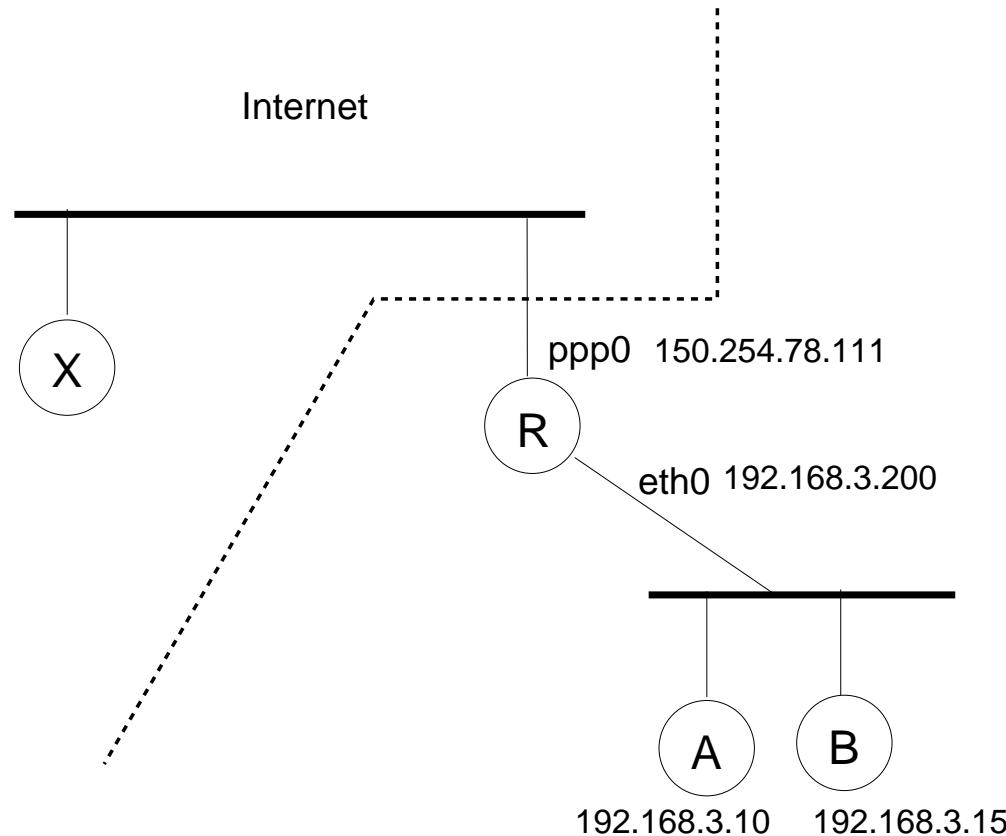
Więcej o routerach ...

- NAT (ang. Network Address Translation)
zamiana adresów IP i/lub nr portów pakietów przechodzących przez router
gdy wraca "odpowiedź" wykonuje się na pakiecie operacje odwrotną !
- SNAT, MASQ, modyfikowanie adresów IP i nr portów **źródłowych**
umożliwia dostęp do internetu z sieci lokalnej, z adresami prywatnymi!
MASQ jak SNAT, ale gdy router ma zmienny adres IP
- DNAT, modyfikowanie adresów IP i nr portów **docelowych**
umożliwia udostępnianie w internecie serwerów, pracujących na maszynach w sieci
lokalnej z adresami prywatnymi
(o ile router ma publiczny adres IP ...)
- zapora sieciowa, czyli odrzucanie niektórych pakietów IP
- linux: wszystko (NAT i zapory) robimy poleceniem *iptables* !

```
iptables -A INPUT -p tcp --dport 8080 -j ACCEPT; # wpuszczamy tcp, dport=8080
iptables -A INPUT -j DROP; # odrzucamy wszystkie pakiety
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
iptables -t nat -A POSTROUTING -o ppp0 -j SNAT --to 150.254.78.111
iptables -t nat -A PREROUTING -i ppp0 -p tcp --dport 80 -j DNAT \
--to 192.168.3.10:8015
```

Jak działa SNAT/ DNAT ??

```
iptables -t nat -A POSTROUTING -o ppp0 -j SNAT --to 150.254.78.111  
iptables -t nat -A PREROUTING -i ppp0 -p tcp --dport 80 -j DNAT \  
--to 192.168.3.10:8015
```



Prot. nad warstwą transportową: FTP

- służy do kopирования plików File Transfer Protocol
- model klient/serwer
- zasada działania: używa 2 połączeń TCP;
 1. połączenie dla komend, przez to połączenie klient wysyła rozkazy do serwera i otrzymuje odpowiedzi (patrz wydruk z sockspy, ftp01.txt)
 2. połączenie dla danych służy do kopowania plików; tworzone gdy to jest potrzebne wiele razy...
- dwa tryby tworzenia połączenia dla danych:
aktywne FTP:
klient tworzy połączenie dla komend do serwera czekającego na porcie 21, port klienta w to N; klient wysyła komendę "PORT N+1" i czeka na połączenie od serwera na porcie N+1,
serwer tworzy połączenie dla danych do klienta czekającego na porcie N+1
pasywne FTP:
klient tworzy połączenie dla komend do serwera czekającego na porcie 21, klient wysyła komendę "PASV" a serwer odpowiada z nr portu M, na którym będzie oczekiwał na połączenie dla danych (i robi to),
klient tworzy połączenie dla danych do serwera czekającego na porcie M

wydruk z socksproxy - prot FTP

14:40:45
220 FTP server ready.

14:40:45
USER anonymous

14:40:45
230 Anonymous user logged in

14:40:45
PWD

14:40:45
257 "/" is your current location

14:40:45
PASV

14:40:45
227 Entering Passive Mode (127,0,0,1,226,167)

14:40:45
TYPE A

14:40:45
200 TYPE is now ASCII

14:40:45
LIST -la ./

14:40:45
150 Accepted data connection

...

Prot. nad warstwą transportową: HTTP

- przeglądarka ściąga strony z serwera WWW za pomocą prot. HTTP
- ma także inne zastosowania, np. tzw REST API...
- model klient-serwer, połączenie TCP (wielokrotnie tworzone)
- opisane w dokumencie RFC 2616 (HTTP/1.1), <http://tools.ietf.org/html/rfc2616>

- żądanie HTTP:

```
GET /index.html HTTP/1.0 [CRLF]
Accept: image/gif, image/jpeg [CRLF]
User-Agent: Mozilla/4.0 [CRLF]
Host: www.cs.huji.ac.il:80 [CRLF]
Connection: Keep-Alive [CRLF]
[CRLF]
```

.....

pierwsza linia zawiera: metode, url, wersje protokołu
następne linie to tzw "nagłówki"
potem "pusta linia" i ew. dane (dane metody POST)

- metody w żądaniu HTTP:

GET - pobieranie zasobu na który wskazuje URL w żądaniu HTTP
(nie powinno niczego modyfikować na serwerze!!)

POST - przyjęcie danych od klienta (np. z formularza HTML)

HEAD - jak GET, ale nie pobiera danych zasobu (same nagłówki)

PUT - podobne do POST, ale URL oznacza co innego ("obiekt", a nie "metode")

DELETE - usuwanie zasobu

- ważne nagłówki w żądaniu HTTP:
 - "Host: ??" - umożliwia tworzenie "wirtualnych hostów" (1 adres IP, wiele adresów domenowych), obowiązkowy w HTTP/1.1
 - "Connection: Keep-Alive" - jedno połączenie używane do wielu zapytań HTTP
 - "Authorization: Basic cXFxOnFxcQ==" - uwierzytelnianie klienta typu "basic" (jeśli kod odp http = 401 to przeglądarka otwiera okno user/passwd)
 - "Cookie: ??" - ciasteczka wysyłane przez przeglądarkę do serwera

- odpowiedź HTTP:

```
HTTP/1.0 200 OK [CRLF]
Date: Fri, 31 Dec 1999 23:59:59 GMT [CRLF]
Content-Type: text/html [CRLF]
Content-Length: 1354 [CRLF]
[CRLF]
<html> [CRLF]
<body> [CRLF]
<h1>Hello World</h1> [CRLF]
.....
```

pierwsza linia zawiera: wersję prot, kod odpowiedzi i jej słowny opis
następne linie zawierają nagłówki odpowiedzi HTTP
potem "pusta linia" i dane odpowiedzi, np. HTML lub coś innego...

- kody w odpowiedzi HTTP:

200 OK - prawidłowa odpowiedź
302 Found - tzw "redirekt", przeglądarka powinna przełączyć się na inny URL
401 Unauthorized - strona wymaga, aby użytkownik się uwierzytelnił
404 Not Found - serwer nie znalazł zasobu

- ważne nagłówki w odpowiedzi HTTP:

"Content-Type: text/html" - typ odpowiedzi jako mime
"Content-Type: text/html; charset=utf-8"
"Content-Length: 1354" - długość odpowiedzi HTTP
"Set-Cookie" - serwer zmusza przeglądarkę żeby utworzyła ciasteczko

- przekazywanie dodatkowych parametrów do żądania HTTP:

1. zmienne w url-u, met. GET

```
http://localhost:8001/np02/plik1.tcl?x=1234&y=4321  
# kodowanie znaków przy pomocy %kod, tzw "x-url-encoding"
```

2. "dane POST" za pusta linia, met. POST

żądania http typu POST (lub GET) są tworzone przez formularz w pliku HTML, w przeglądarce, guzik submit lub przez biblioteki http, pokazać przykład http02.tcl ... kwestia kodowania znaków (utf-8 ? iso8859-2 ?)

```
POST /np02/plik1.tcl HTTP/1.0[CRLF]  
Accept: */*[CRLF]  
Host: localhost:8001[CRLF]  
User-Agent: Tcl http client package 2.5.2[CRLF]  
Content-Type: application/x-www-form-urlencoded[CRLF]  
Content-Length: 15[CRLF]  
[CRLF]  
x=12345&y=54321[CRLF]
```

- ciasteczka czyli Cookies, session_id ...
 - w żądaniu HTTP:
Cookie: nazwa1=wartość1; nazwa2=wartość2; ...
 - w odpowiedzi HTTP:

```
Set-Cookie: nazwa=wartość; expires=DATA; path=ŚCIEŻKA; secure
# expires - czas życia ciasteczka u klienta
# path - jaki url-i na hostie to ciasteczko dotyczy
# secure - tylko dla HTTPS
```
 - jak działają ciasteczka?
 - + tworzone przez odpowiedź http serwera www (Set-Cookie:)
 - + dopóki się nie przeterminują, wysyłane przez przeglądarkę do serwera www (Cookie:) w każdym żądaniu http
 - + fizycznie ciasteczka są przechowywanymi w plikach u klienta (przez przeglądarkę)
 - **zastosowanie ciasteczek:** umożliwiają przechowywanie "zmiennych sesyjnych" na serwerze www;
 - + co to są "zmienne sesyjne" ? zmienne związane z sesją użytkownika
 - + co to jest "sesja użytkownika" ? ciąg kliknięć (w przeglądarce), przez danego użytkownika, które nie są zbytnio oddzielone w czasie...
 - + dlaczego ciasteczka są niezbędne? bo serwer http jest "bezstanowy"
 - + identyfikator sesji jest przechowywany w ciasteczku u klienta ...

- obsługa sesji użytkownika na przykładzie framework'a webowego "OpenACS": ciasteczko z identyfikatorem sesji: *ad_session_id*
parametry obsługi sesji:
SessionRenew = 5min (czas po którym "odnawia się" ciasteczko sesji)
SessionTimeout = 20min (bez odnawiania sesja znika po tym czasie)
SessionLifetime = 7dni (sesja znika)
pokazać przykład oacs_session.tcl ...
- *ad_session_id*/ pytanie 1: jak jest minimalny czas między kliknięciami, po którym sesja może zniknąć???
- *ad_session_id*/ pytanie 2: dlaczego *SessionRenew > 0* ? wskazówka: strona www z 1000 obrazków ...

Gniazdka "bezpieczne" - SSL/TLS

- SSL = Secure Socket Layer, TLS = Transport Layer Security,
OpenSSL = implementacja SSL/TLS (biblioteka programistyczna i polecenie openssl)
- Skrócony opis pojęć kryptograficznych:
klucz symetr, asymetr (pub/pryw), fun haszująca, podpis elektr, certyfikat SSL
(zawiera klucz pub, podpisany przez CA)
- Co zapewnia SSL/TLS?
szylfowanie danych, żłośliwe zmiany niemożliwe, uwierzytelnianie serwera, uwierzytel-
nianie serwera i klienta
- Co jest potrzebne po stronie serwera ?
certyfikat SSL serwera (z kluczem pub serwera), klucz pryw serwera
- Co jest potrzebne po stronie klienta ?
jeśli klient chce sprawdzić certyfikat SSL serwera, to musi podać certyfikat SSL CA
(który podpisał elektronicznie certyfikat serwera)
podobnie w drugą stronę...
- Zasada działania ...
 1. serwer wysyła do klienta swój cert+klucz pub
 2. klient wymyśla klucz symetryczny X do szyfrowania danych, szyfruje X kluczem
pub serwera i wysyła go do serwera
 3. serwer odszyfrowuje swoim kluczem pryw X; teraz oba końce mają X służący do
szylfowania danych płynących przez połączenie...

- polecenie openssl:

```
# openssl: szyfrowanie/deszyfrowanie metoda Blowfish
# "-e" encode, "-d" decode, "-a" base64, "-bf" Blowfish
echo "tekst do zaszyfrowania" | openssl enc -e -a -bf -k haslo > qqq.txt
cat qqq.txt | openssl enc -d -a -bf -k haslo
```

- bezpieczna odmiana HTTP: HTTPS
to samo co http, ale używa połączenia TCP nad SSL/TLS ...
- pakiet "tls" j. Tcl, komenda **tls::socket** zamiast **socket**, pokazać przykład...

```
## klient, socket/tls, j. Tcl
#
package re tls

set s [tls::socket localhost 10000]
# + bez uwierzytelniania serwera

set s [tls::socket -require 1 -cafile cacert.pem localhost 10000]
# + wymagamy uwierzytelnienia serwera
#   "-require 1" klient chce sprawdzenia certyfikatu serwera
#   "-cafile" certyfikat CA (chodzi m.in. o klucz publiczny CA w tym pliku
#   dzięki któremu klient sprawdza podpis na certyfikacie serwera)

tls::handshake $s
# ustalanie klucza symetrycznego do szyfrowania danych...

puts $s "A ku ku !!!"; flush $s

close $s
```

```
## serwer, socket/tls, j. Tcl
#
package re tls

proc haslo {} {return "qwerty"}; # haslo do klucza pryw serwera

tls::socket -server obsluga -password haslo \
    -keyfile privkey_2.pem -certfile cert_2.pem \
    10000
    # "-keyfile" klucz pryw serwera
    # "-certfile" cert serwera (zawiera klucz pub serwera)

proc obsluga {s args} {
    puts "obsluga: $s sie podlaczył"
    tls::handshake $s
    puts "obsluga: $s sie podlaczył 2"
    fileevent $s readable "obslugaKli $s"
}

proc obslugaKli s {
    if {[eof $s]} { puts "obslugaKli: $s close"; close $s; return }
    set linia [gets $s]
    puts "obslugaKli: od $s: $linia"
    puts $s "odp na $linia"; flush $s
}
```

cert ssl wygenerowany przez CA.sh/openssl, wersja tekstowa:

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 1 (0x1)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=AU, ST=Some-State, O=Internet Widgits Pty Ltd, CN=autorytet

Validity

Not Before: Jun 23 21:53:29 2020 GMT

Not After : Jun 23 21:53:29 2021 GMT

Subject: C=AU, ST=Some-State, O=Internet Widgits Pty Ltd, CN=serwer nr 1

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:a5:4d:31:66:c4:66:46:1d:0c:e4:54:6e:d9:23:
f8:27:a1:2f:f7:99:87:59:d9:4d:3f:f4:e4:c9:f7:
05:72:0c:96:0e:2c:53:8b:db:76:eb:ee:80:60:53:
f0:6b:01:f9:85:aa:48:70:b0:d3:a3:6f:66:a2:84:
c8:fd:2c:b2:f7:33:4b:9d:d0:df:21:60:7a:56:8d:
ee:3a:28:c5:eb:6a:ea:be:d6:10:fe:29:6a:74:b3:
f2:34:36:a0:c2:26:ff:3a:d1:7f:2c:d7:b3:78:69:
cb:67:9c:b1:d4:9b:18:a9:d6:80:ba:82:5e:ab:c0:
ec:ce:33:be:f8:57:b9:ab:f7

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

Netscape Comment:

OpenSSL Generated Certificate

X509v3 Subject Key Identifier:

57:76:47:18:16:5D:E0:E5:71:9A:A4:98:75:5F:2A:DC:87:5E:AD:76

X509v3 Authority Key Identifier:

keyid:44:72:CA:52:51:4F:02:85:55:6C:76:0A:BA:4C:BD:4F:CA:67:16:A5

Signature Algorithm: sha1WithRSAEncryption

1a:ad:a1:6c:d2:fd:99:82:f5:19:fd:af:82:28:20:cc:94:68:
bc:27:84:83:2a:c6:aa:5d:db:a6:27:87:cc:50:90:d0:19:8b:
e5:de:e4:aa:0f:1d:e1:10:fe:f6:af:fe:f1:84:fb:86:29:62:
76:00:da:26:d6:62:7d:4f:e9:15:d9:d8:5c:d1:ee:c8:03:18:
c8:19:e2:01:a0:35:bb:ef:dd:fe:2c:b0:f3:e2:43:82:40:24:
f4:9d:6c:cc:7a:65:9f:00:0c:38:84:21:ee:ee:22:44:da:22:
f3:a0:bf:1c:1f:93:76:5d:1b:11:c4:33:b7:f3:ac:d6:9d:1e:
28:09

-----BEGIN CERTIFICATE-----

MIICpTCCAg6gAwIBAgIBATANBgkqhkiG9w0BAQUFADBZMQswCQYDVQQGEwJBVTET
MBEGA1UECBMKU29tZS1TdGF0ZTEhMB8GA1UEChMYSW50ZXJuZXQgV21kZ210cyBQ
dHkgTHRkMRIwEAYDVQQDEwlhdXRvcn10ZXQwHhcNMjAwNjIzMjE1MzI5WhcNMjEw
NjIzMjE1MzI5WjBbMQswCQYDVQQGEwJBVTETMBEGA1UECBMKU29tZS1TdGF0ZTEh

MB8GA1UEChMYSW50ZXJuZXQgV21kZ210cyBQdHkgTHRkMRQwEgYDVQQDEwtzZXJ3
ZXIgbnIgMTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEApU0xZsRmRh0M5FRu
2SP4J6Ev95mHWd1NP/TkyfcFcgyWDixTi9t26+6AYFPwawH5hapIcLDTo29mooTI
/Syy9zNLndDfIW6Vo3u0ijF62rqvtYQ/ilqdLPyNDagwib/0tF/LNezeGnLZ5yx
1JsYqdaAuoJeq8Dszej0++Fe5q/cCAwEAaAaN7MHkwCQYDVR0TBAIwADAsBglghkgB
hvhCAQOEHxYdT3B1b1NTTCBHZW51cmFOZWQgQ2VydGlmaWNhdGUwHQYDVR00BBYE
FFd2RwgWXeDlcZqkmHVfKtyHXq12MB8GA1UdIwQYMBaAFERyy1JRTwKFVWx2CrpM
vU/KZxa1MA0GCSqGSIB3DQEBBQUAA4GBABqtOWzS/ZmC9Rn9r4IoIMyUaLwnhIMq
xqpd26Ynh8xQkNAZi+Xe5KoPHeEQ/vav/vGE+4YpYnYA2ibWYn1P6RXZ2FzR7sgD
GMgZ4gGgNbvv3f4ssPPiQ4JAJP SdbMx6ZZ8ADDiEIe7uIkTaIv0gvxwf k3ZdGxHE
M7fzrNadHigJ
-----END CERTIFICATE-----

DNS

Główne zadanie: zamiana nazw domenowych na adr IP

Zawiera także inne informacje o nazwach domenowych...

Usługa DNS utrzymuje tablice rekordów (**RR**) tej postaci:

(nazwa_domenowa, typ, wartość, ?), gdzie typ to ...

A (adr ip), MX (MTA), NS (autorytatywny ser dns domeny),

CNAME (nazwa kanon), i inne ... patrz: **rfc 1035**

Działa nad prot UDP lub TCP (większe ser dns...)

Zasada działania:

- mój serwer DNS ma informacje o wielu nazw dom (w cache!!)

- jeśli nie ma informacji np. o „lts.wmi.amu.edu.pl” to wtedy:

pyta korzeń o adres ser dns obsługującego domenę „pl”

ten serwer powinien wiedzieć wszystko o nazw „*.pl” (subdomeny)

czyli powinien znać adr ser dns obsługującego domenę „edu.pl”

ten powinien znać adres ser dns obsługującego „amu.edu.pl”

...itd... aż dojdziemy do tzw *serwera autorytatywnego*,

znającego nazwę „lts.wmi.amu.edu.pl”

„Autorytatywny ser dns” dla danej nazwy to ten, w którym zarejestrowano (a nie zcacheowano) nazwę hosta lub domeny

Hierarchia DNS...

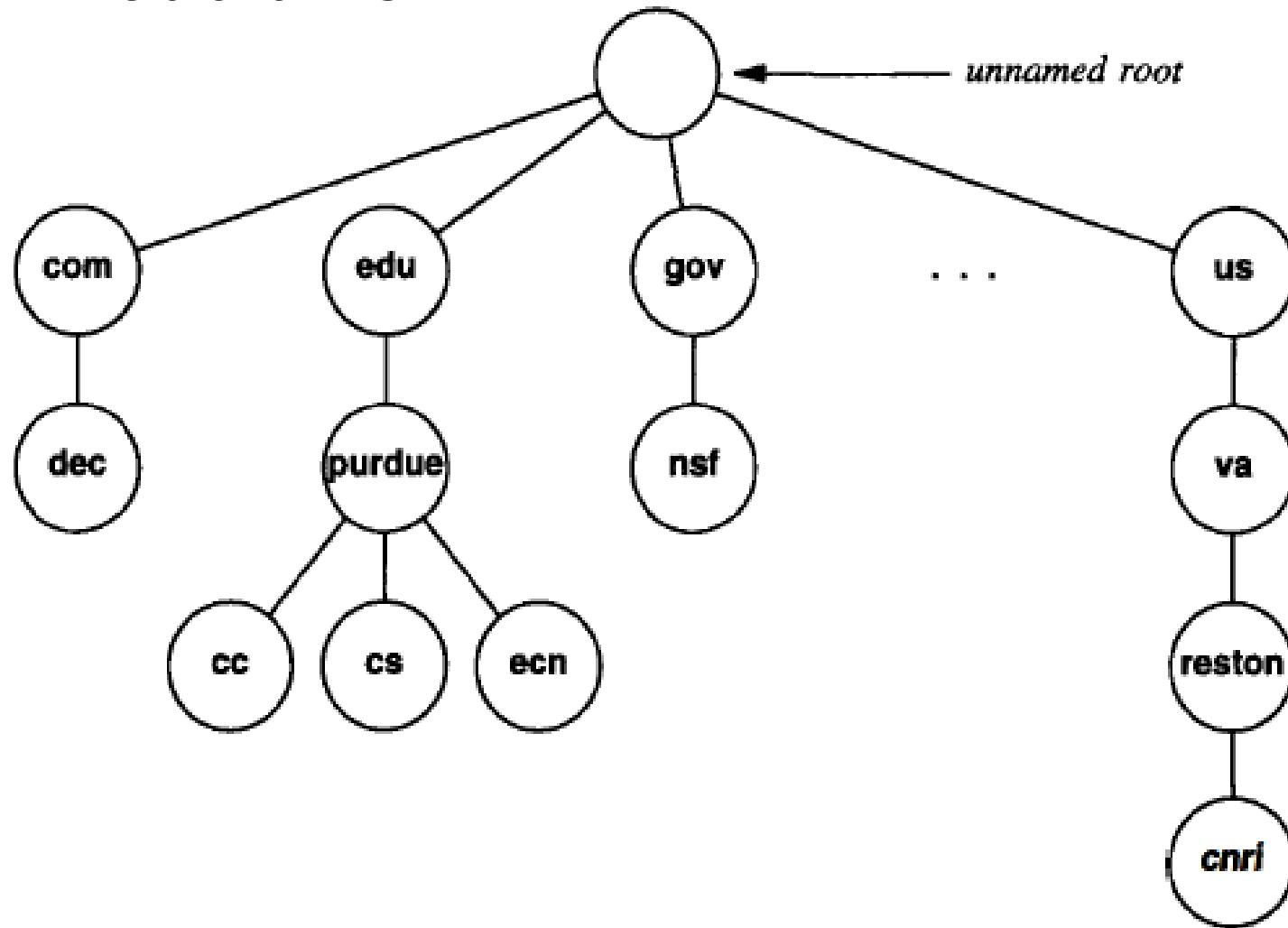
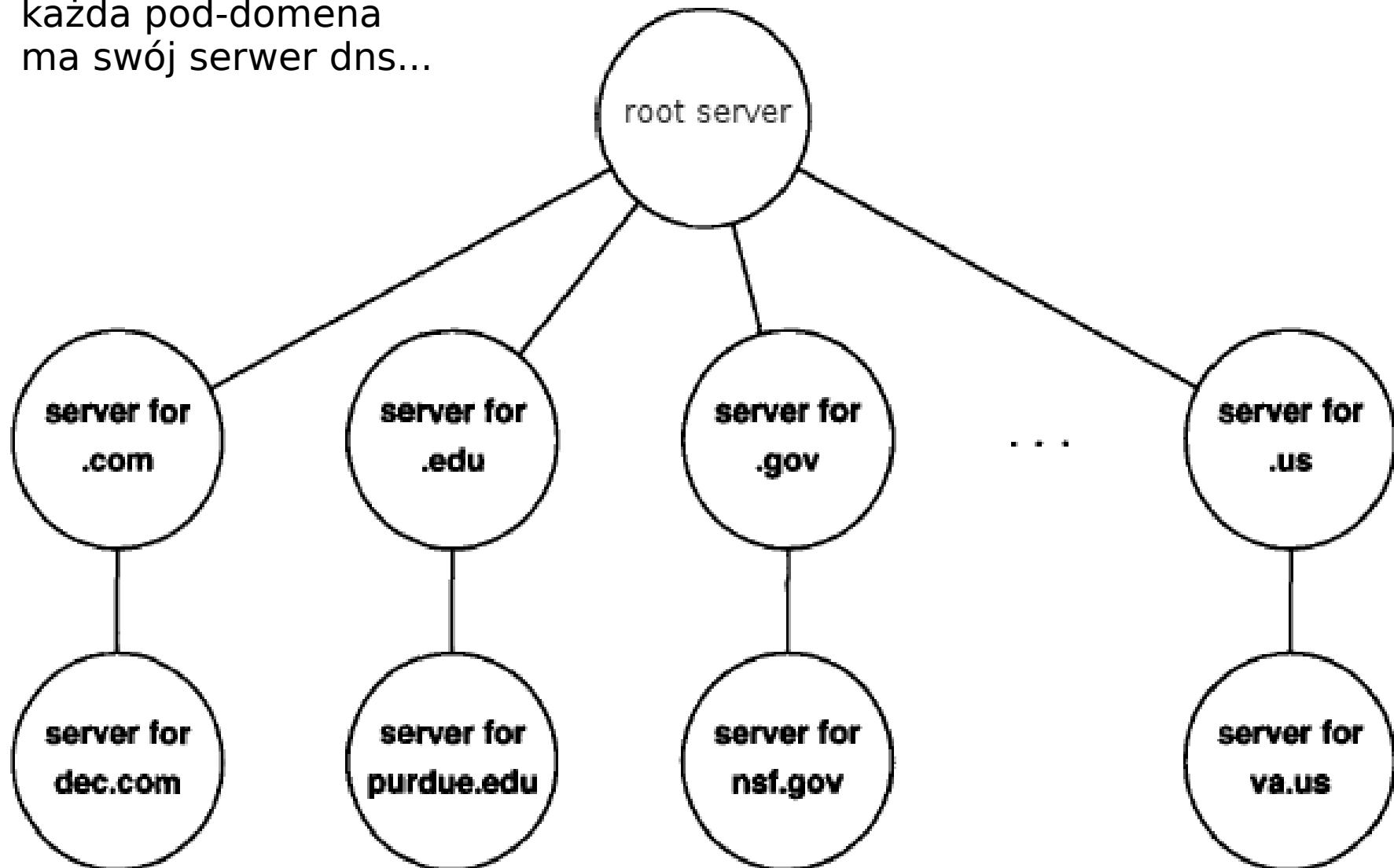


Figure 24.2 A small part of the Internet domain name hierarchy (tree). In practice, the tree is broad and flat; most host entries appear by the fifth level.

Podejście wyidealizowane:
każda pod-domena
ma swój serwer dns...



Podejście realistyczne:
bardziej „spłaszczone” ...

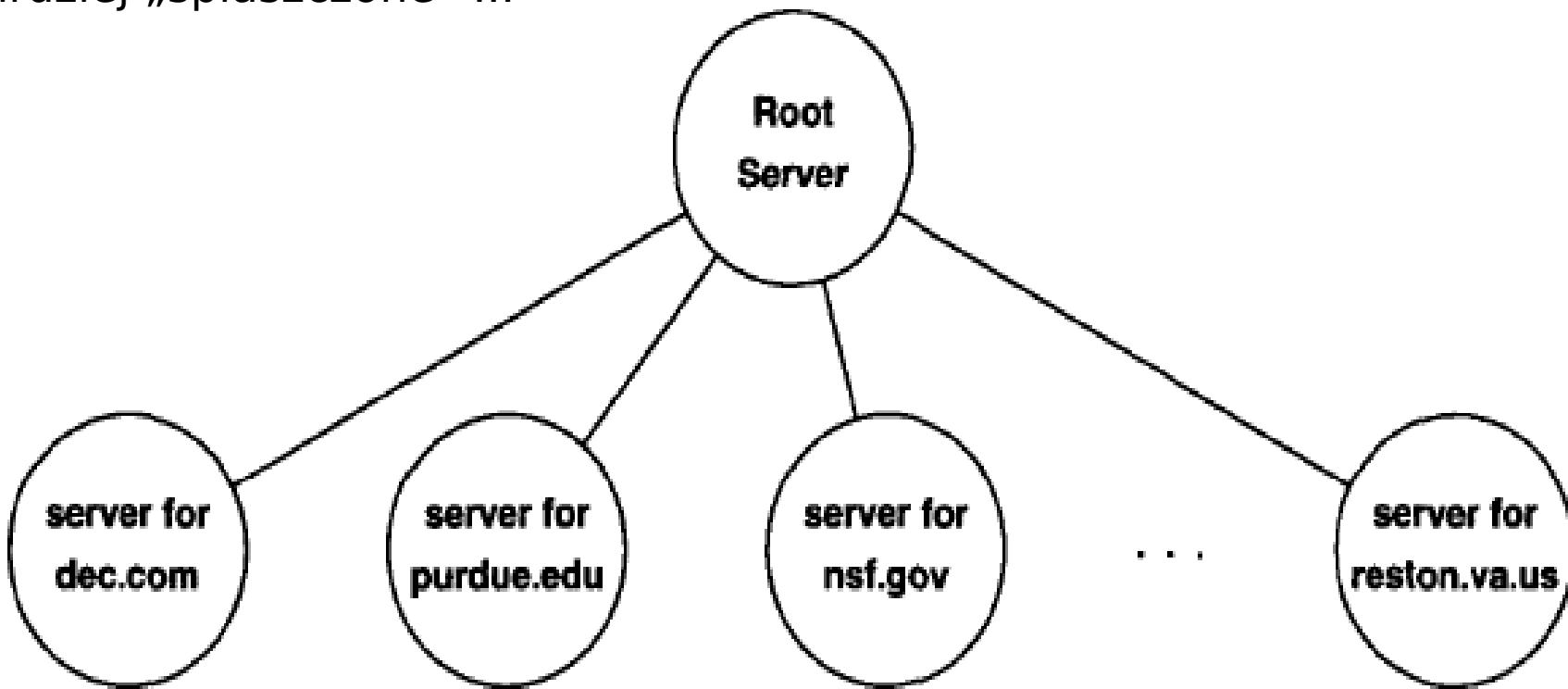
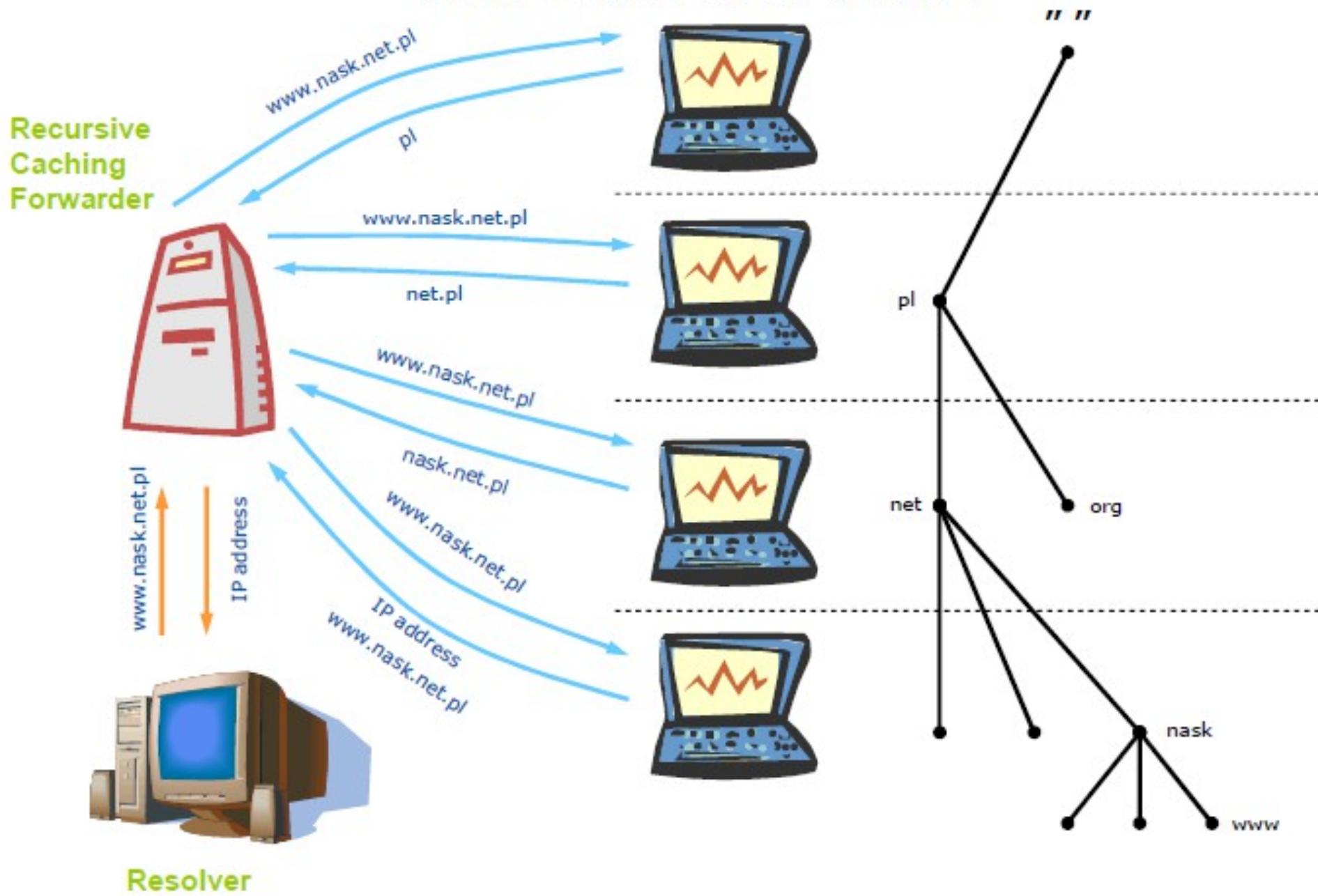
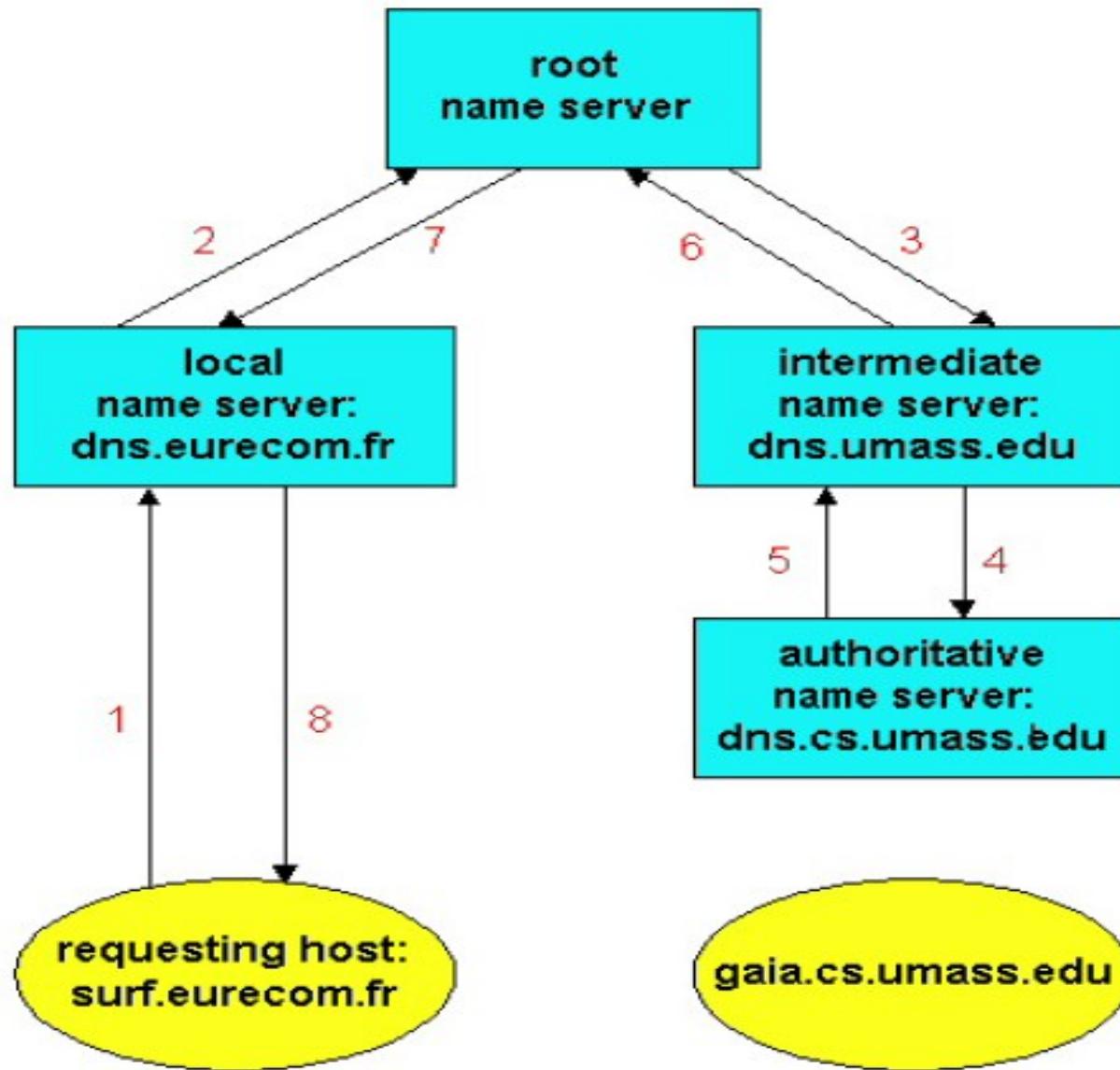


Figure 24.4 A realistic organization of servers for the naming hierarchy of Figure 24.2. Because the tree is broad and flat, few servers need to be contacted when resolving a name.

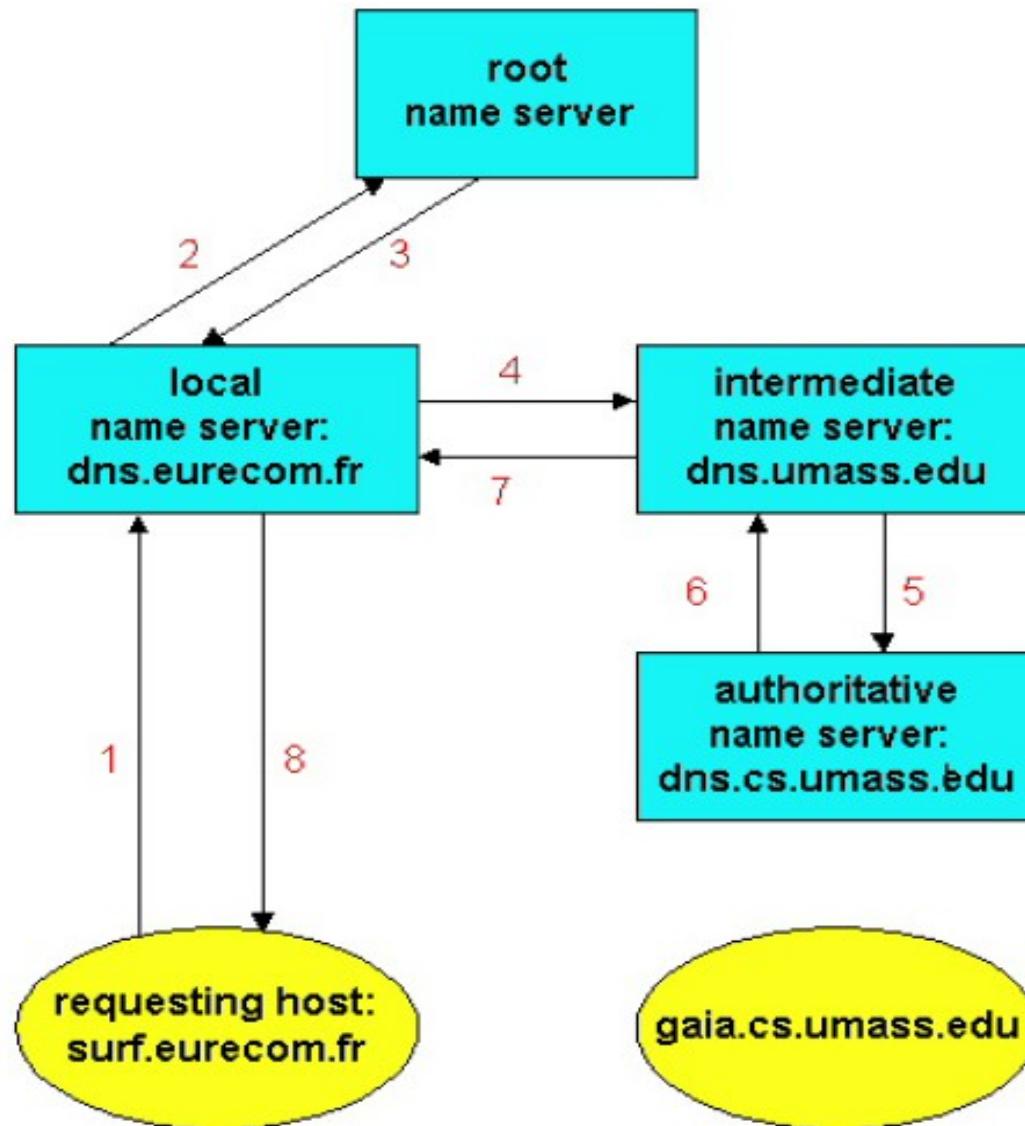
How Does DNS work ?



Szukamy gaia.cs.umass.edu; tryb pracy: rekurencyjny



Tryb pracy: iteracyjny oraz rekurencyjny



```
## pakiet dns, j. Tcl
#
package re udp; # !!
#% 1.0.11
package re dns
#% 1.3.3

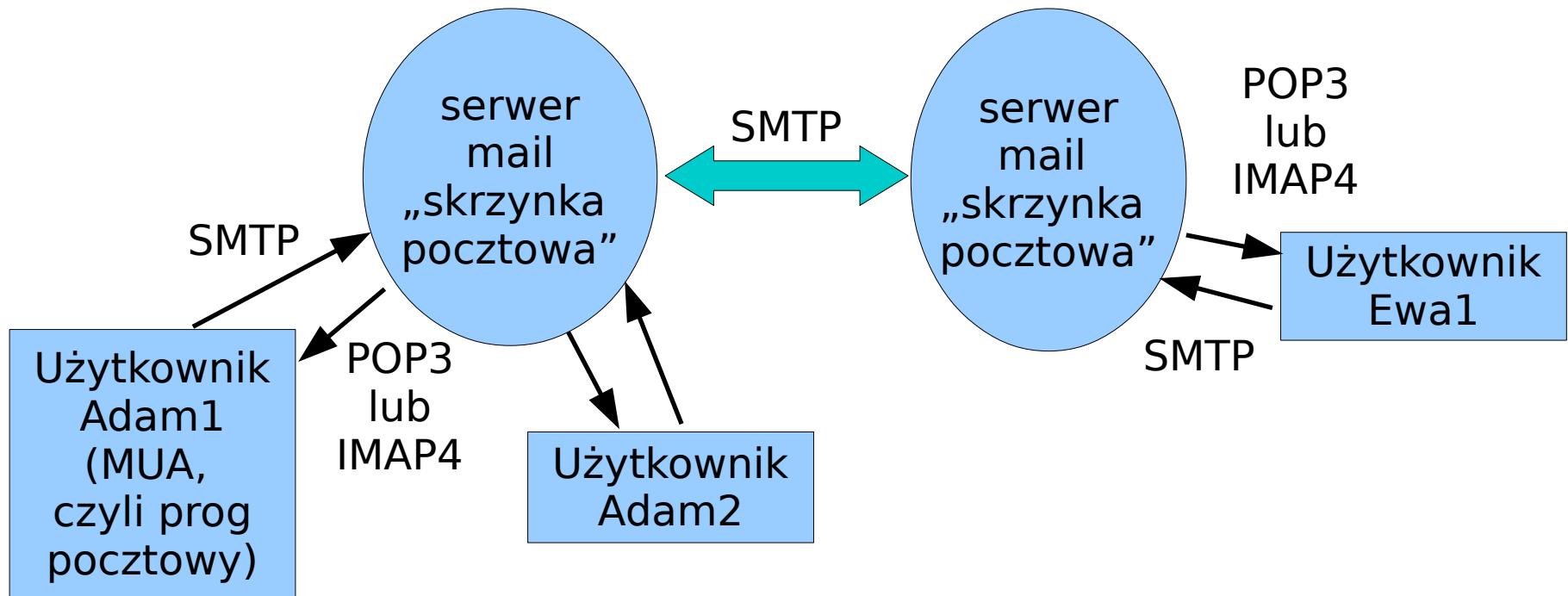
set n1 [dns::resolve "wp.pl"]
dns::wait $n1
dns::address $n1
#% 212.77.98.9
join [dns::result $n1] \n
#% name wp.pl type A class IN ttl 231 rdlenth 4 rdata 212.77.98.9

set n1 [dns::resolve "amu.edu.pl" -type NS]; # serwery DNS dla tej domeny
dns::wait $n1
join [dns::result $n1] \n
#% name amu.edu.pl type NS class IN ttl 3192 rdlenth 7 rdata dns2.amu.edu.pl
name amu.edu.pl type NS class IN ttl 3192 rdlenth 6 rdata dns.amu.edu.pl
name amu.edu.pl type NS class IN ttl 3192 rdlenth 7 rdata dns3.amu.edu.pl

set n1 [dns::resolve "amu.edu.pl" -type MX]; # mail/MTA dla tej domeny
dns::wait $n1
join [dns::result $n1] \n
#% name amu.edu.pl type MX class IN ttl 5 rdlenth 8 rdata {10 mx2.amu.edu.pl}
name amu.edu.pl type MX class IN ttl 5 rdlenth 8 rdata {10 mx1.amu.edu.pl}
```

Mail (email)

(mocno uproszczony opis...)



Adres mailowy: user@serwer, user@nazwa_domenowa

DNS wyciąga z nazwy dom adr ser mailowego (MX)

Budowa maila:

nagłówki (from:, to:, date:, subject:, content-type:),
pusta linia,
treść (w mailach złożonych: części mime; to jest drzewko!!)

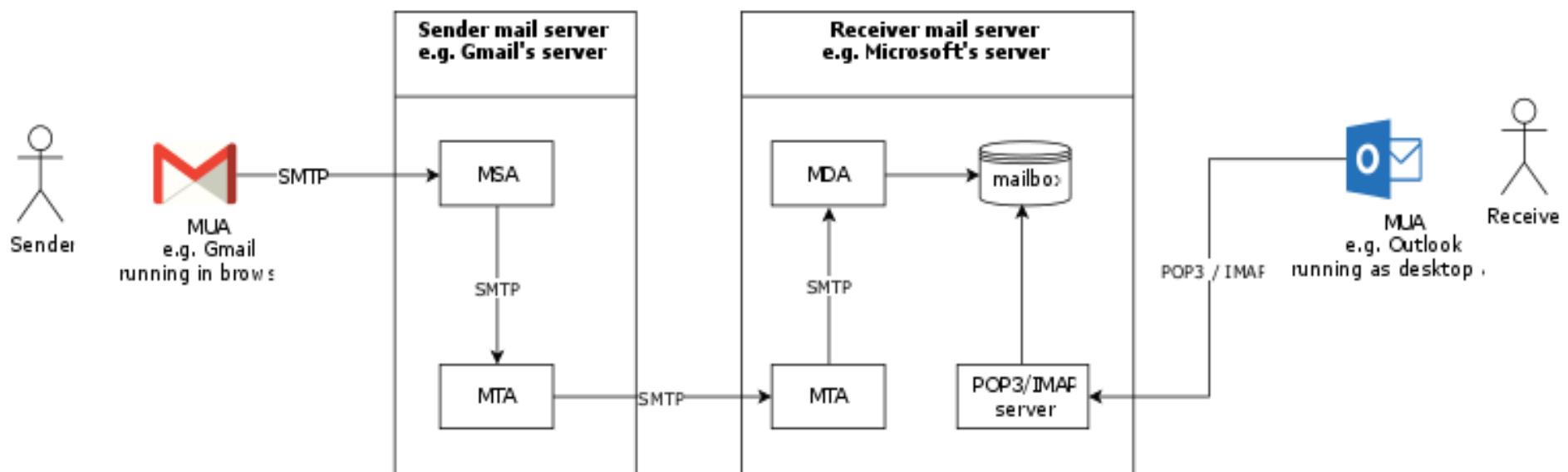
Serwery mail - więcej szczegółów...

dłaczego MX ma inny adr ip niż ser SMTP ???

MTA = Mail Transfer Agent

MSA = Mail Submission Agent

MDA= Mail Delivery Agent



```
## pakiet pop3, j. Tcl
#
package re tls
package re pop3
source pop3_tls.tcl; # dodaje obsługę tls...

set password ???
set p [pop3::open -ssl 1 pop3.amu.edu.pl mhanckow $password 995]

set maile [split [pop3::list $p]\n]; llength $maile; # mam tyle maili...
#% 694
lrange $maile end-5 end
#% {689 7011} {690 11797} {691 4464} {692 30434} {693 44913} {}

set raw0 [pop3::top $p 693 10]
# + naglowki maila + 10 linijek tresci maila...

set raw1 [pop3::retrieve $p 693]; string len $raw1
#% 44286
# + jednak sciagamy caly mail...
```

```
## pakiet mime, j. Tcl
#
package re mime

set m1 [mime::initialize -string $raw1]
mime::getheader $m1 from
    #%" {Alicja Adamczyk <aliada2@st.amu.edu.pl>}
mime::getheader $m1 date
    #%" {Wed, 3 Jun 2020 10:56:32 +0000}
mime::getproperty $m1 content
    #%" multipart/mixed
set m2 [mime::getproperty $m1 parts]
    #%" ::mime::1-1 ::mime::1-2

mime::getproperty [lindex $m2 0] content
    #%" multipart/alternative
mime::getproperty [lindex $m2 1] content
    #%" text/plain

set m3 [lindex $m2 0]
    #%" ::mime::1-1
mime::getproperty $m3 content
    #%" multipart/alternative
set m4 [mime::getproperty $m3 parts]
    #%" ::mime::1-1-1 ::mime::1-1-2
mime::getproperty [lindex $m4 0] content
    #%" text/plain
mime::getproperty [lindex $m4 1] content
    #%" text/html
```

Usługa katalogowa, X.500, LDAP

usługa katalogowa czyli opis zasobów w sieci...

zasoby: użytkownicy, grupy użytkowników, komputery,
zakłady, departamenty, drukarki, ...

drzewo obiektów/folderów

ścieżka od korzenia do wierzchołka (obiektu lub folderu)

ścieżka jest postaci "attr1=value1, attr2=value2, ..."

obiekty także mają atrybuty (np. cn, mail, description, displayName, ...)

można zadawać "zapytania o obiekty" z warunkiem logicznym

LDAP = Lightweight Directory Access Protocol, dostęp do usł. kat.

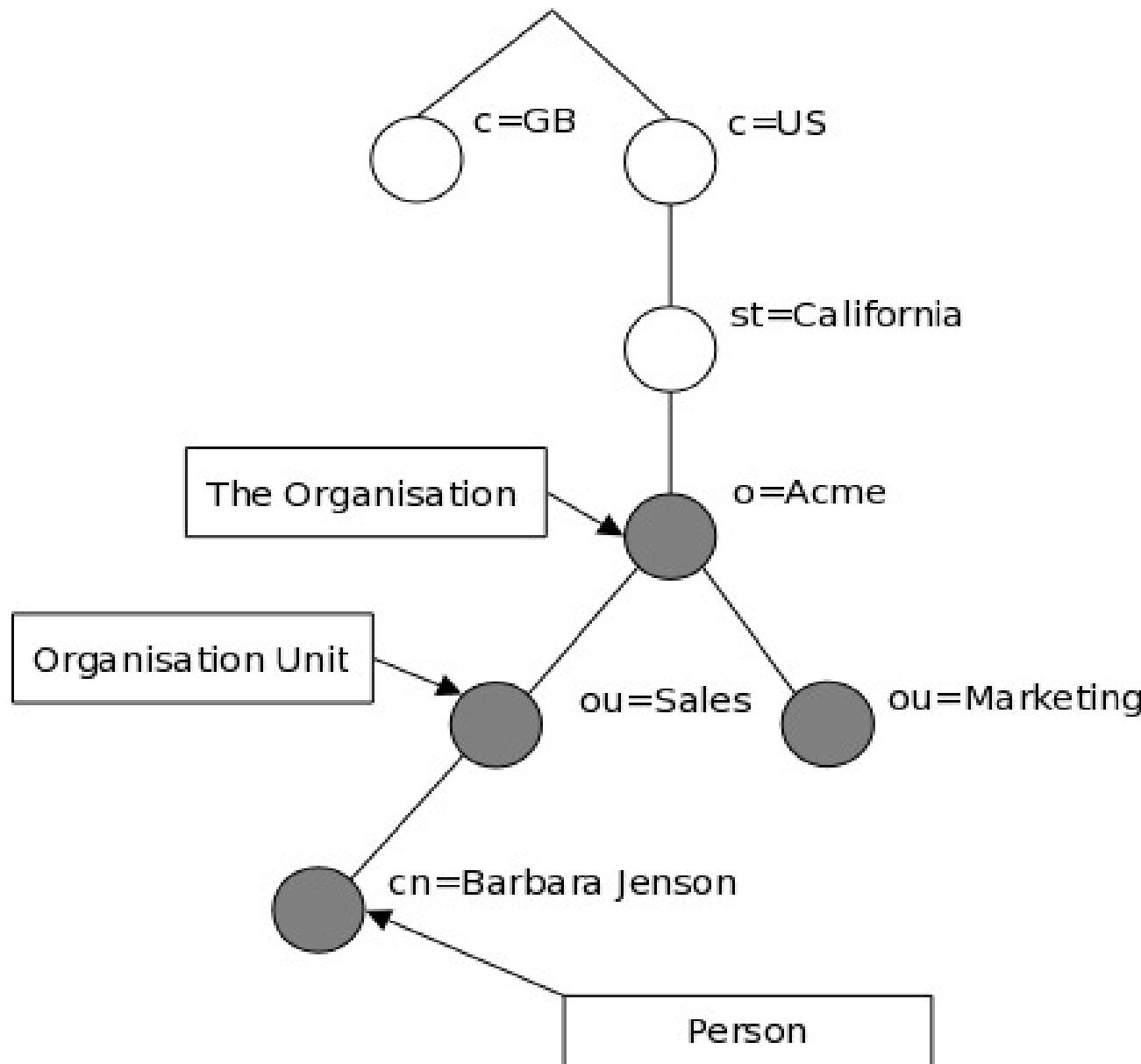
docs: X.500, rfc2251, **rfc4511 (LDAP)**,

<http://www.openldap.org/doc/admin24/intro.html>

Drzewo LDAP/X.500...

ścieżka do „Barbara Jenson”:

„cn=Barbara Jenson, ou=Sales, o=Acme, st=California, c=US”



```
package re ldap
catch {package re dict}

# tunel SSH jest niezbędny jeśli działamy spoza WMI !!!
#ssh -L 5000:ldap.wmi.amu.edu.pl:389 ???@lts.wmi.amu.edu.pl
set h [ldap::connect localhost 5000]

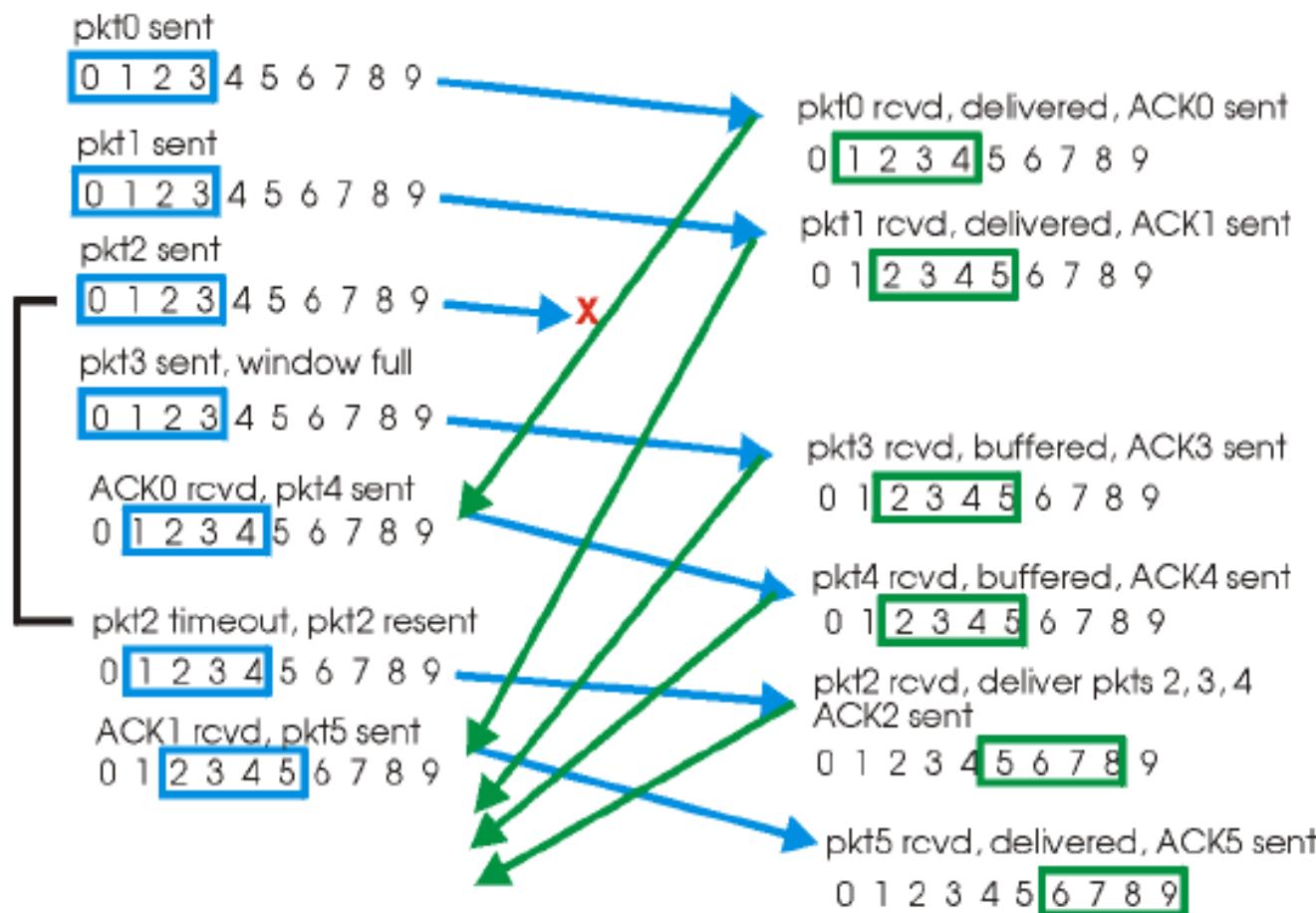
set x1 [ldap::search $h "DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl" "(CN=mhanckow)"""]
llength $x1
    #%
lindex $x1 0 0
    #%
    # CN=mhanckow,OU=Faculty,OU=People,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
    # ^ distinguished name
lsort [dict keys [lindex $x1 0 1]]
    #%
    # accountExpires badPasswordTime badPwdCount c cn co codePage company countryCode ...
    # ^ atrybuty obiektu

dict get [lindex $x1 0 1] description
    #%
    # Pracownik
dict get [lindex $x1 0 1] distinguishedName
    #%
    # CN=mhanckow,OU=Faculty,OU=People,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
dict get [lindex $x1 0 1] mail
    #%
    # mhanckow@amu.edu.pl
encoding convertfrom utf-8 [dict get [lindex $x1 0 1] displayName]
    #%
    # Michał Hańćkowiak
join [dict get [lindex $x1 0 1] memberOf] \n
    #%
    # CN=Zaklad_Teorii_Algorytmow_i_Bezpieczenstwa_Danych,OU=Departments,
    # OU=Groups,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
CN=public,OU=Groups,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
CN=faculty_research,OU=Groups,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
CN=faculty,OU=Groups,DC=labs,DC=wmi,DC=amu,DC=edu,DC=pl
```

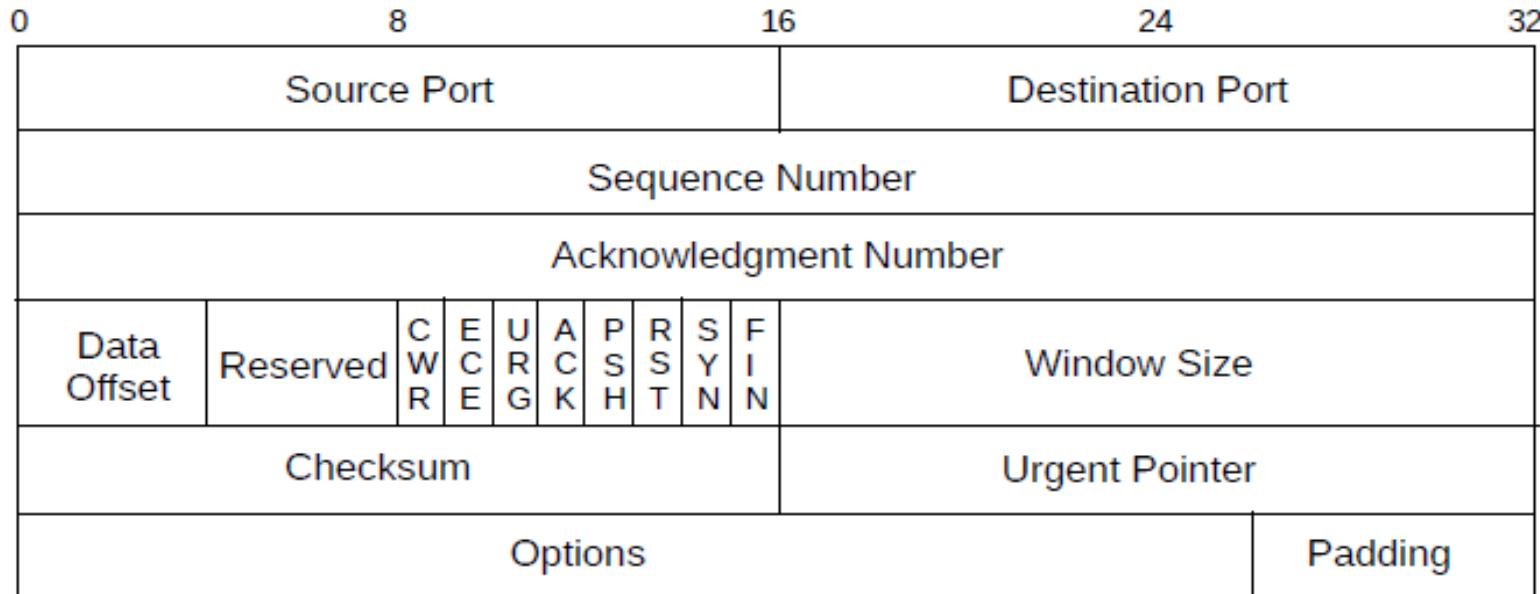
TCP

Co już wiemy ?

1. **strumień danych**, brak struktury komunikatów
2. **połączenie wirtualne**, dwukierunkowe, ident przez 4 liczby: saddr/daddr/sport/dport
3. **bufory** nadawczy i odbiorczy na obu końcach poł. TCP (flush/psh)
4. zachowuje się jak łącza unix-owe (close(), prod/kons)
5. **przesuwające się okno** (ang. sliding window), segm TCP z danymi i ack...



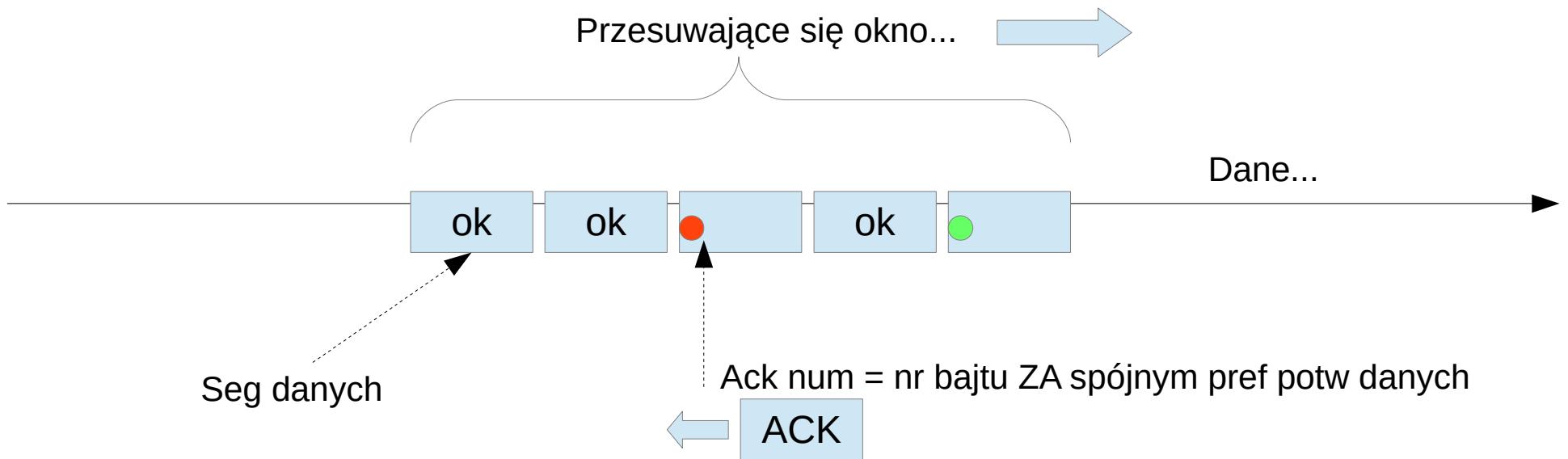
format segmentu TCP



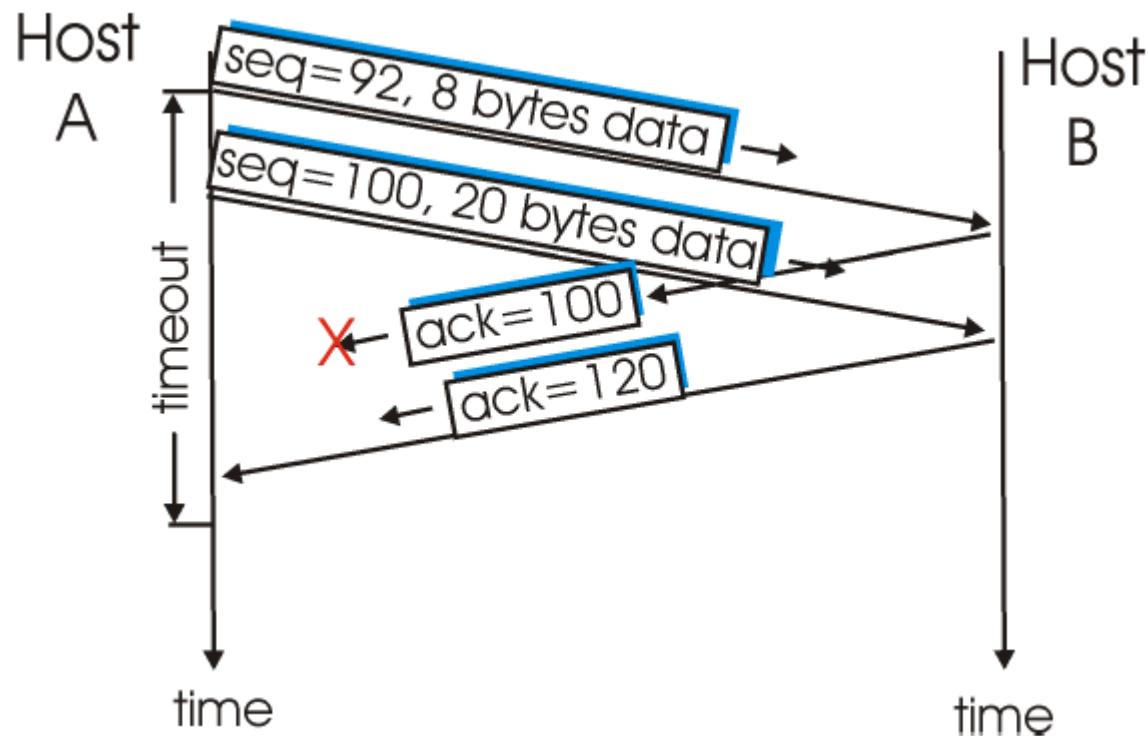
Najważniejsze pola:

1. sport, dport
2. seq num (pol. nr porządkowy), A->B
nr pierwszego bajtu w seg danych z pkt widzenia nadawcy
3. ack num (pol. nr potwierdzenia), B->A
nr bajtu ZA potwierdzonym spójnym ciągiem danych u odbiorcy,
1 seg ACK może potwierdzić wiele seg danych !!
4. flagi: ACK, PSH, RST, SYN, FIN, URG, ...
określają znaczenie seg, może być kilka naraz,
ACK seg potwierdzający dostarczenie danych,
PSH ma związek z flush(), dotyczy buf odbiorczego
SYN/FIN nawiązywanie i zamykanie poł TCP
5. win size, proponowany przez odbiorcę rozmiar „sliding window” (wolna pamięć odbiorcy)

okno, seg z danymi, seg ack...



okno, seg z danymi, seg ack...



widać, kiedy 1 seg ack może potwierdzać dane z 2 seg danych...

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.6	192.168.1.200	TCP	74	filenet-rpc > 34512 [SYN] Seq=0 Win=5840 Len=0
2	0.000035	192.168.1.200	192.168.1.6	TCP	74	34512 > filenet-rpc [SYN, ACK] Seq=0 Ack=1 Win=8
3	0.000290	192.168.1.6	192.168.1.200	TCP	66	filenet-rpc > 34512 [ACK] Seq=1 Ack=1 Win=5840
4	0.001535	192.168.1.6	192.168.1.200	TCP	1514	filenet-rpc > 34512 [ACK] Seq=1 Ack=1 Win=5840
5	0.001609	192.168.1.200	192.168.1.6	TCP	66	34512 > filenet-rpc [ACK] Seq=1 Ack=1449 Win=8
6	0.002772	192.168.1.6	192.168.1.200	TCP	1514	filenet-rpc > 34512 [ACK] Seq=1449 Ack=1 Win=5
7	0.002894	192.168.1.200	192.168.1.6	TCP	66	34512 > filenet-rpc [ACK] Seq=1 Ack=2897 Win=1

Frame 4: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
Ethernet II, Src: AmbitMic_37:5f:d8 (00:d0:59:37:5f:d8), Dst: AsustekC_66:a8:75 (00:0e:a6:66:a8:75)
Internet Protocol Version 4, Src: 192.168.1.6 (192.168.1.6), Dst: 192.168.1.200 (192.168.1.200)
Transmission Control Protocol, Src Port: filenet-rpc (32769), Dst Port: 34512 (34512), Seq: 1, Ack: 1, Len: 1448
Source port: filenet-rpc (32769)
Destination port: 34512 (34512)
[Stream index: 0]
Sequence number: 1 (relative sequence number)
[Next sequence number: 1449 (relative sequence number)]
Acknowledgement number: 1 (relative ack number)
Header length: 32 bytes
Flags: 0x010 (ACK)
Window size value: 5840
[Calculated window size: 5840]

0020 01 c8 80 01 86 d0 74 6d 09 64 af 53 56 15 80 10tm .d.SV...
0030 16 d0 5d b8 00 00 01 01 08 0a 00 01 52 9b 00 08 ...].....R...
0040 2b 4d 67 67 67 67 67 67 67 67 67 67 67 67 67 +Mgggggg gggggggg

Seg danych nr 1 w wireshark...

seg ten ma ok 1.5kb oraz seq num NIE jest w rzeczywistosci =1 (!)

Filter: | Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.6	192.168.1.200	TCP	74	filenet-rpc > 34512 [SYN] Seq=0
2	0.000035	192.168.1.200	192.168.1.6	TCP	74	34512 > filenet-rpc [SYN, ACK]
3	0.000290	192.168.1.6	192.168.1.200	TCP	66	filenet-rpc > 34512 [ACK] Seq=1
4	0.001535	192.168.1.6	192.168.1.200	TCP	1514	filenet-rpc > 34512 [ACK] Seq=1
5	0.001609	192.168.1.200	192.168.1.6	TCP	66	34512 > filenet-rpc [ACK] Seq=1
6	0.002772	192.168.1.6	192.168.1.200	TCP	1514	filenet-rpc > 34512 [ACK] Seq=1
7	0.002894	192.168.1.200	192.168.1.6	TCP	66	34512 > filenet-rpc [ACK] Seq=1
8	0.003100	192.168.1.6	192.168.1.200	TCP	1514	filenet-rpc > 34512 [ACK] Seq=1

```

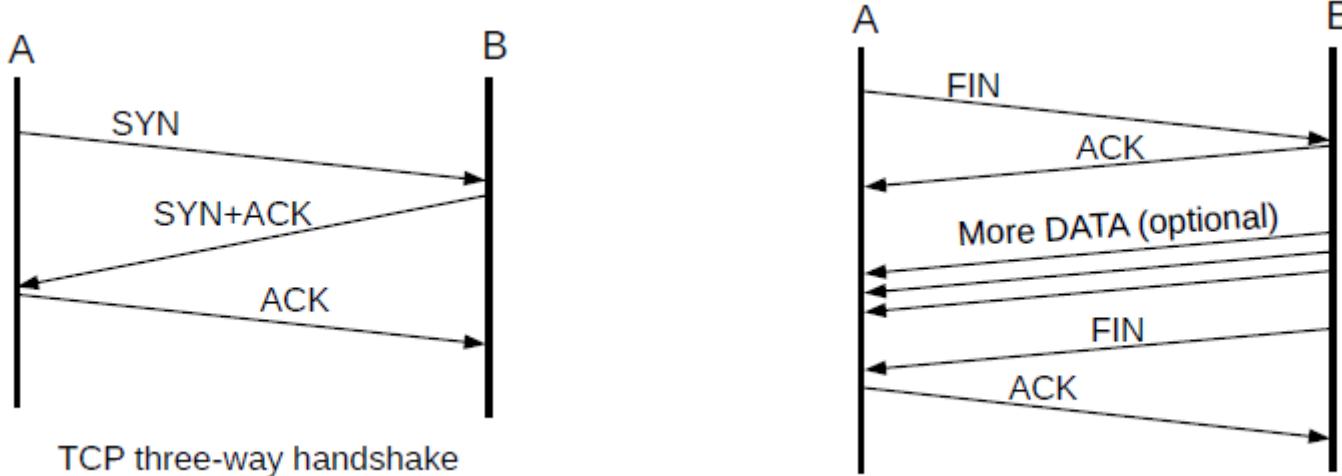
Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
Ethernet II, Src: AsustekC_66:a8:75 (00:0e:a6:66:a8:75), Dst: AmbitMic_37:5f:d8 (00:d0:59:37:5f:d8)
Internet Protocol Version 4, Src: 192.168.1.200 (192.168.1.200), Dst: 192.168.1.6 (192.168.1.6)
Transmission Control Protocol, Src Port: 34512 (34512), Dst Port: filenet-rpc (32769), Seq: 1, Ack: 1449, Len: 1514
    Source port: 34512 (34512)
    Destination port: filenet-rpc (32769)
    [Stream index: 0]
    Sequence number: 1      (relative sequence number)
    Acknowledgement number: 1449      (relative ack number)
    Header length: 32 bytes
Flags: 0x010 (ACK)
Window size value: 8688
[Calculated window size: 8688]
[Window size scaling factor: 1]

```

0020	01	06	86	d0	80	01	af	53	56	15	74	6d	0f	0c	80	10S V.tm...
0030	21	f0	c3	08	00	00	01	01	08	0a	00	08	2b	4d	00	01	!..... .+M..
0040	52	9b															R.

Seg ack dla poprzedniego seg danych...
ack num wskazuje na następny bajt którego spodziewa się odbiorca

Tworzenie/ niszczenie poł TCP



	A, ISN=1000	B, ISN=7000
1	SYN, seq=1000	
2		SYN+ACK, seq=7000, ack=1001
3	ACK, seq=1001, ack=7001	
4	“abc”, seq=1001, ack=7001	
5		ACK, seq=7001, ack=1004
6	“defg”, seq=1004, ack=7001	
7		seq=7001, ack=1008
8	“foobar”, seq=1008, ack=7001	
9		seq=7001, ack=1014, “hello”
10	seq=1014, ack=7006, “goodbye”	

ISN=initial seq num, potw z wartością „isn+1”, isn to nie 1 !!,
obie strony poznają numerację bajtów z obu stron...

Automat skonczony dla TCP, klient, (czynności: niebieski/ kli, czerwony/ ser)

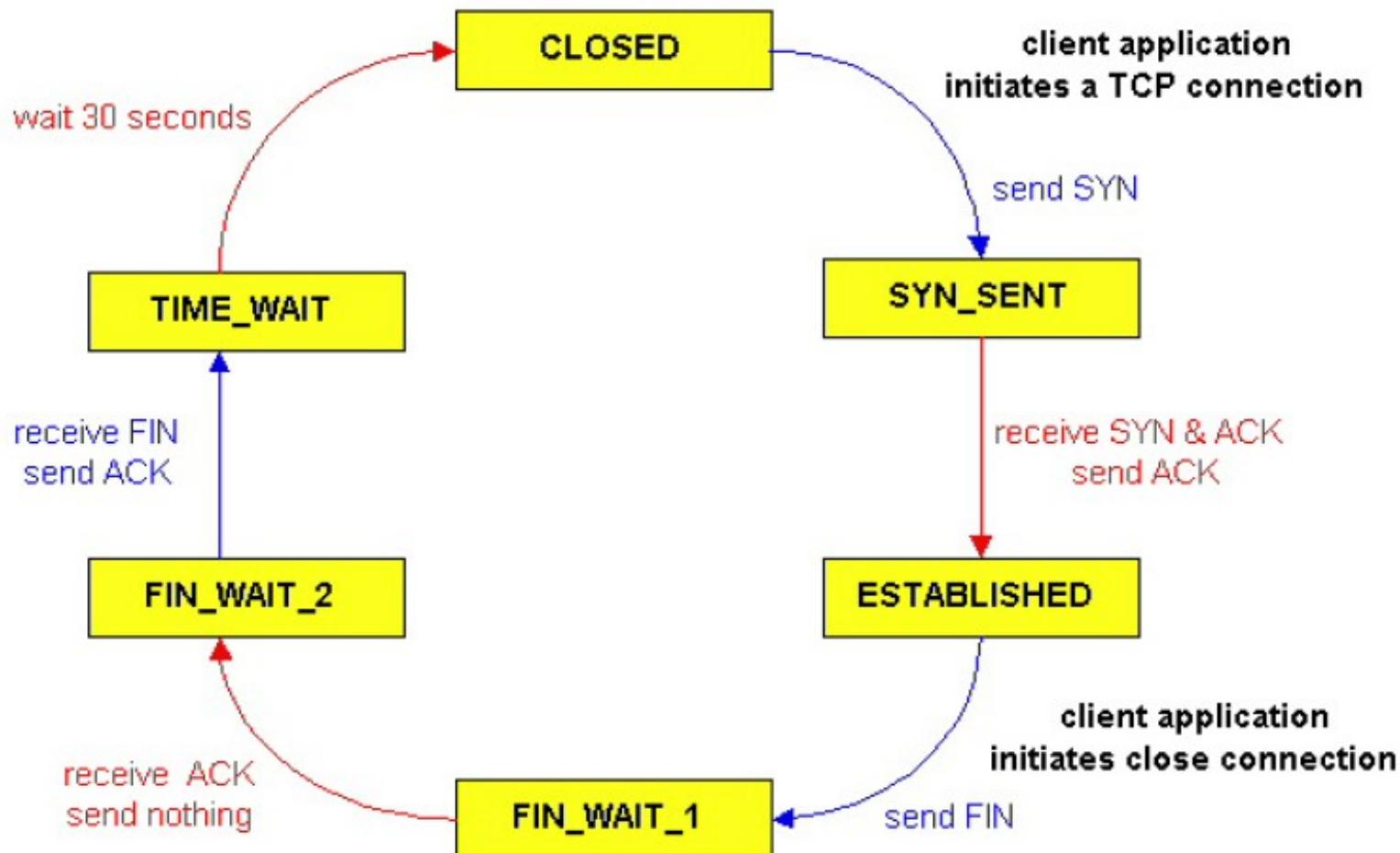


Figure 3.5-11: A typical sequence of TCP states visited by a client TCP

Automat skonczony dla TCP, serwer, (czynności: niebieski/ kli, czerwony/ ser)

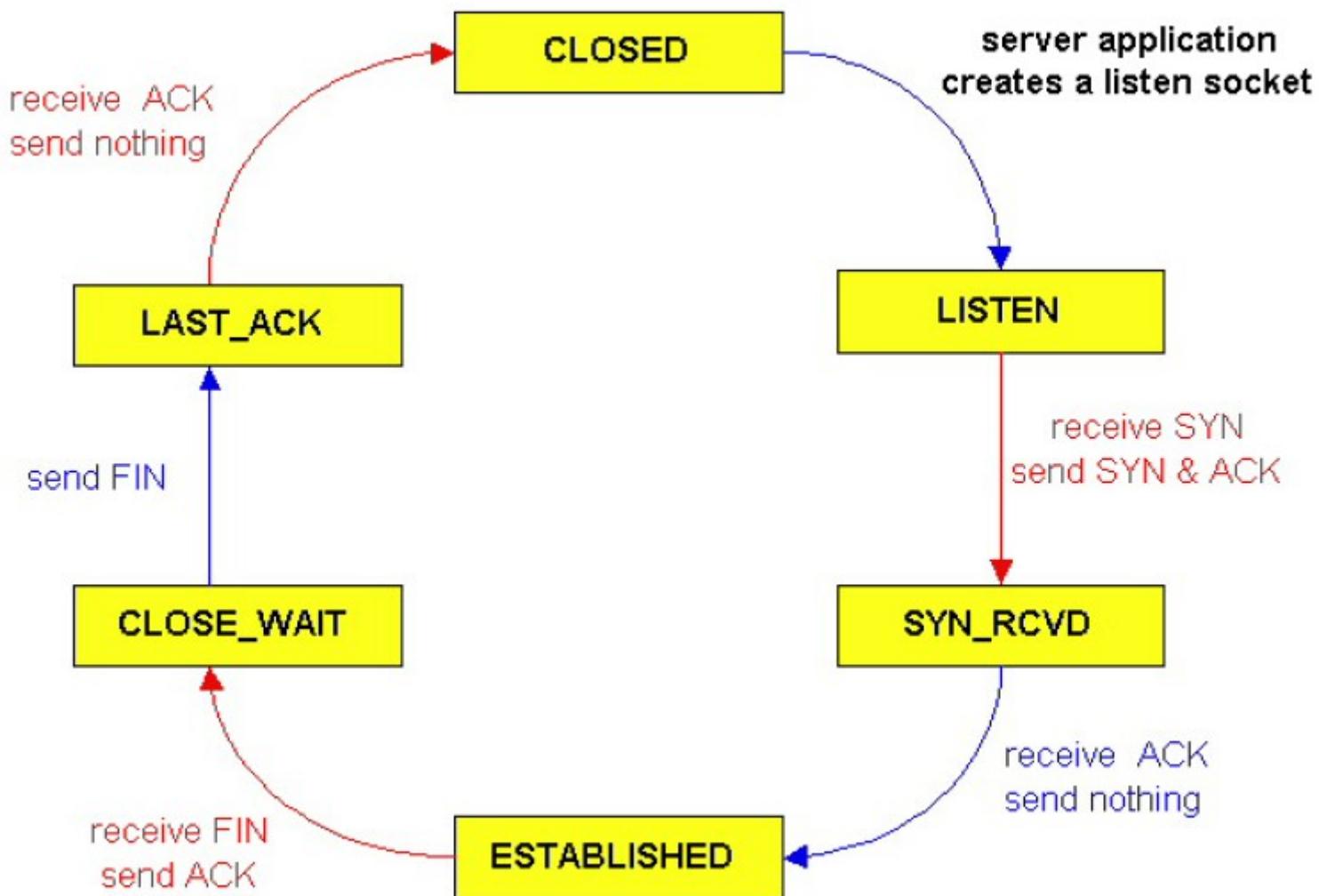


Figure 3.5-12: A typical sequence of TCP states visited by a server-side TCP

Pytania

Jak prot TCP zapewnia „niezawodność połączenia” ?

Potwierdzanie i retransmisja seg danych...

Po jakim czasie retransmisja?? tzw. timeout

Jak prot TCP walczy z „przeciążeniem routerów” ?

Zmniejszanie okna, zwiększenie timeout

Potem trzeba wrócić do normalnej przepustowości...

Problemy z którymi zmaga się prot TCP:

Długość seg danych: na ogół 1.5kb, wg Comera trudne...

Timeout retransmisji: na podstawie szacowanego RTT (ang. round trip time)

Rozmiar okna: powoli rośnie (tzw „powolny start”), gwałtownie maleje przy przeciążeniu

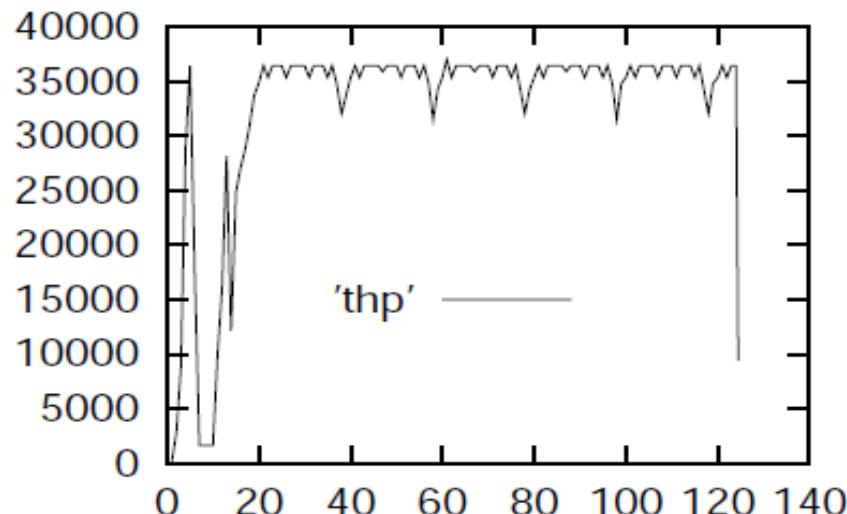


Figure 4.1: Throughput of TCP connection

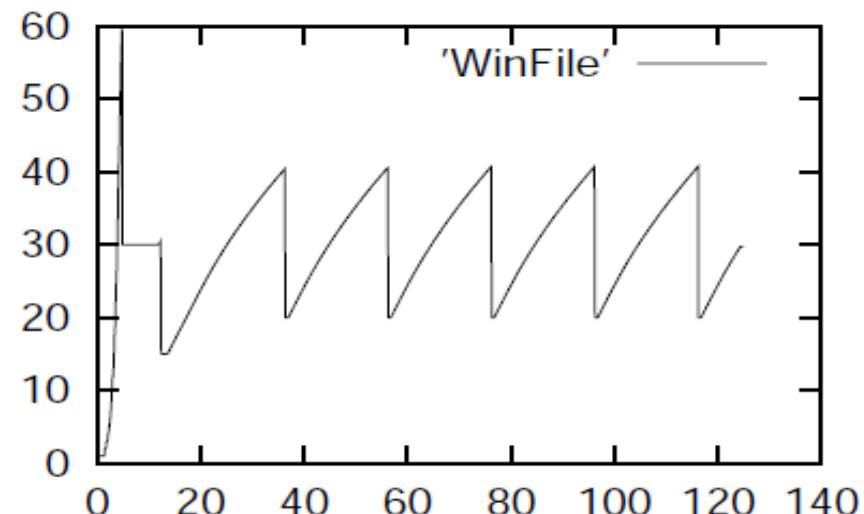


Figure 4.2: Window size of TCP connection

Szacowanie param dla TCP/ szczegóły

Obliczanie „timeout” retransmisji: $x \in (0,1)$, np. $x=0.1$

EstimatedRTT = $(1-x)$ EstimatedRTT + x SampleRTT

RTT = round trip time

Deviation = $(1-x)$ Deviation + $x |$ SampleRTT – EstimatedRTT|

odchylenie

Timeout = EstimatedRTT + $4 *$ Deviation

po tym czasie retransmitujemy seg danych (jeśli nie ma „ack”)

Obliczanie rozmiaru okna:

Dozwolone_okno= min(propozycja_odbiorcy, okno_przeciążeniowe)

Okno przeciążeniowe: normalnie jest ono równe prop_odbiorcy

1. jeśli zgubi się seg danych => zmniejsz okno_przeciążeniowe o połowę
2. rozpocznij od okna= 1 seg, po przyjściu ack podwajaj dla każdego seg danych,
„powolny start” (wcześniej nie taki powolny bo wykładniczy!),
istnieje próg „win_th”, po jego przekroczeniu okno rośnie wolniej
(to faza „unikania przeciążenia”)

„alg Nagle-a”: mechanizm? unikanie krótkich seg danych o ile to możliwe,
dzięki temu prot tcp jest przystosowany do usługi „telnet” jak i „ftp”
wynika z opóźnienia ack...

„alg Karna”: heurystyka? wydłużanie timeout
nie brać pod uwagę RTT przy retransmisji...

„Sieć fizyczna” = warstwa 1 i 2

Dwa typy kanałów/ łączy warstwy 2:

- **rozgłoszeniowe**, z broadcastingiem
 - wiele węzłów (>2), prot dostępu do łącza, np. eth, wifi
- **punkt do punktu**, dokładnie 2 węzły, np. łącze szeregowe + ppp

Usługi warstwy 2:

1. ramkowanie (format ramki, granice międzyramkowe)
2. MAC (media access control), prot dostępu do łącza (rozgłoszeniowego!)
3. niezawodne dostarczanie (potw + retransmisja, **czasami!!**)
4. kontrola przepływu
5. wykrywanie błędów (np. suma kontrolna, crc, parzystosc w rs232, ...)
6. usuwanie błędów (np. kody Hamminga, hdd? cd?)
7. pełny/pół duplex (komunikacja dwu/jedno kierunkowa)

UWAGA: 3, 4 i 5 robi TCP w warstwie 4 (ale to dotyczy wirt poł TCP) !!!

Typy prot (wielo)dostępu do łącza rozgłoszeniowego:

- dzielące kanał (TDMA, FDMA, CDMA, ...)
Time/Freq/Code Division Multiple Access
- dostęp losowy (CSMA/CD – dawny eth, CSMA/CA - wifi)
Carrier Sense Multiple Acces with Collision Detection/Avoidance
- cykliczne (nadaje ten kto ma token, wirt. ring, TokenRing, FDDI)

Ethernet (eth)

Historia eth:

1. „gruby eth”, 10BASE-5, IEEE 802.3, kabel koncentryczny, transceiver
2. „cieńki eth”, 10BASE-2, 802.3a, kabel koncentryczny, złącza BNC
3. skrętka + hub, 10BASE-T, 802.3j, repeater wieloportowy
4. skrętka + switch, 100BASE-TX = „Fast Eth”, 802.3u, są też inne rodzaje...
100BASE-FX, na światłowodzie
1000BASE-LX, światłowód jednomodowy, do 10km !, gigabit eth
1000BASE-T, skrętka
10GBASE-LX4, 10 GB eth, światłowód 1 lub wielo-modowy, 10Gb/s !

Oznaczenia rodzajów eth:

[przepustowość][rodzaj transmisji]-[rodzaj kabla][dodatkowe oznaczenia]

przepustowość – Mb/s

rodzaj transmisji – BASE = baseband, BROAD = broadband

baseband – bez wysokiej częstotliwości nośnej (sygnał modulowany bezpośrednio przez bity użytkownika)
w eth: zawsze baseband...

rodzaj kabla – T = twisted pair (skrętka); F = fiber (światłowód),
dodatk ozn – X = podobno „full duplex” ?!

Kategorie skrętki (max częstotliwość):

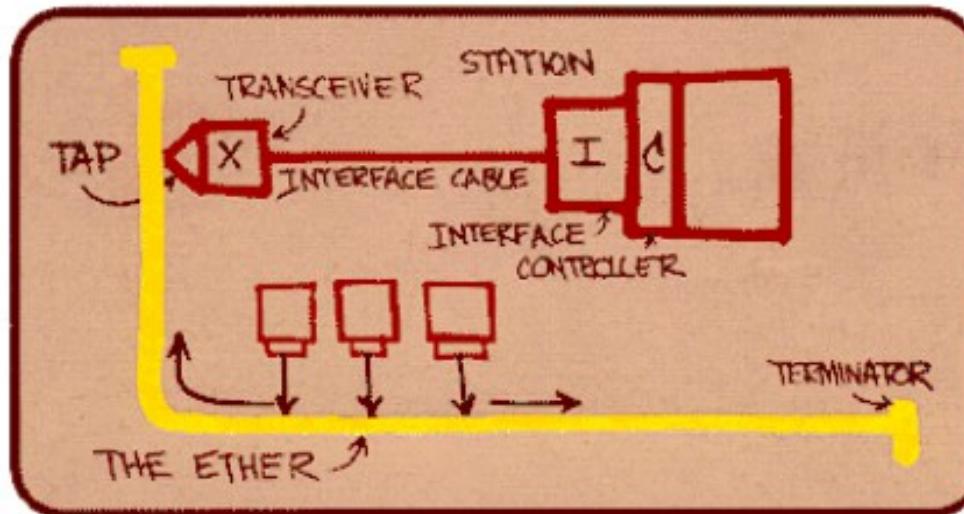
Kat 1, zwykła skrętka telefoniczna, 1 para

Kat 3, dla 10BASE-T, 10MHz, 4 pary

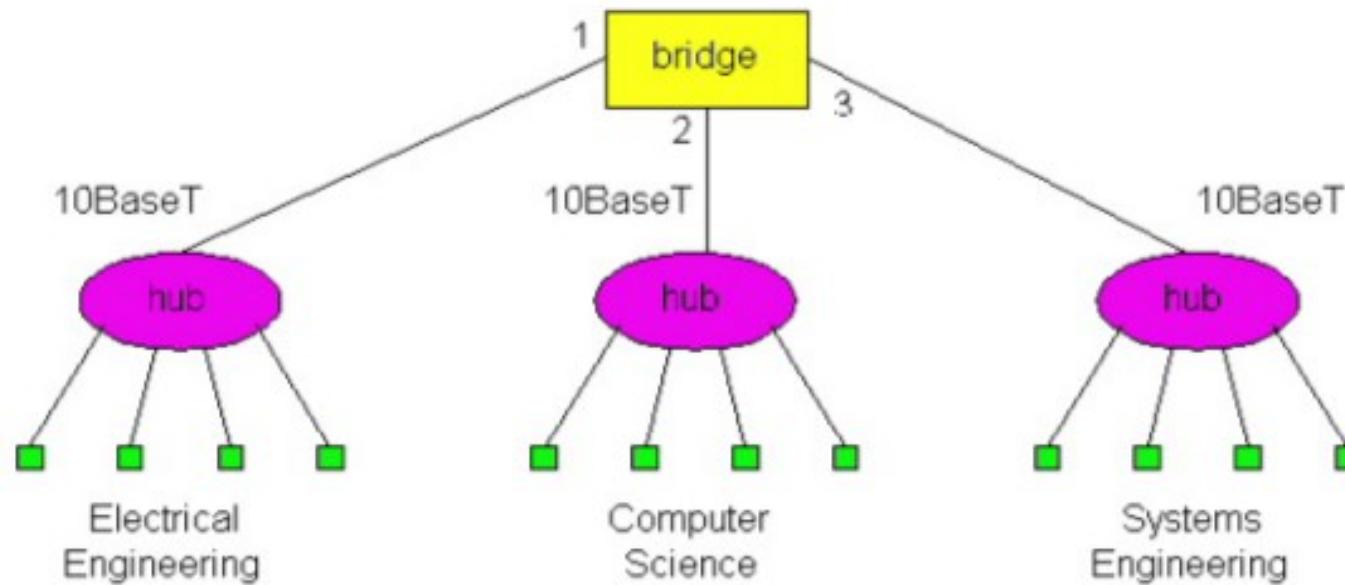
Kat 5, 5e, 100MHz (trochę więcej!), 100m, 100BASE-TX, 4 pary

*Typy skrętki: UTP, FTP, STP
(nieekranowana, foliowana, ekranowana)*

Oryginalny szkic „grubego eth”, Metcalfe, 1976:



Dawniejsza konfiguracja sieci eth: bridge (=switch) i hub-y ...



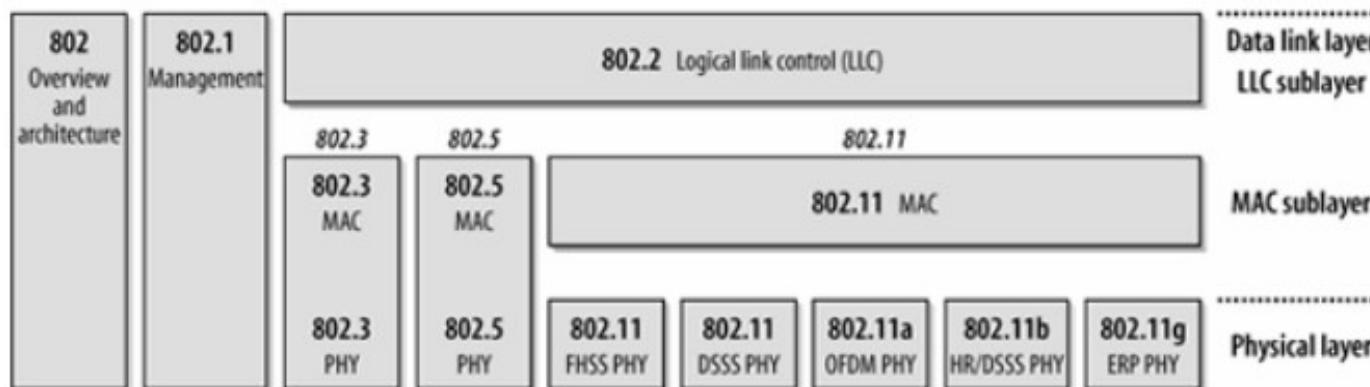
Ramka eth

Preamble	Destination Address	Source Address	Frame Type	Frame Data	CRC
8 octets	6 octets	6 octets	2 octets	46–1500 octets	4 octets

Jest kilka rodzajów ramek: Eth II (DIX), Novellraw, 802.2 LLC, 802.2 SNAP

- najpopularniejsza obecnie: Eth II, z typem za src adr
typy: 0x0800 ipv4, 0x0806 arp, 0x86DD ipv6, ... wikipedia/EtherType
- ramki 802.2 * mają dodatkowy nagłówek w danych !!
wtedy zamiast „frame type” jest „length” <= 1500
dodatkowy nagłówek 802.2 jest wspólny dla eth, wifi i innych sieci...
- podwarstwy warstwy 2:
LLC = Logical Link Ctrl, 802.2, eth: głównie EtherType
MAC = Media Access Ctrl, eth (magistralowy): CSMA/CD, dostęp do medium

Figure 2-1. The IEEE 802 family and its relation to the OSI model

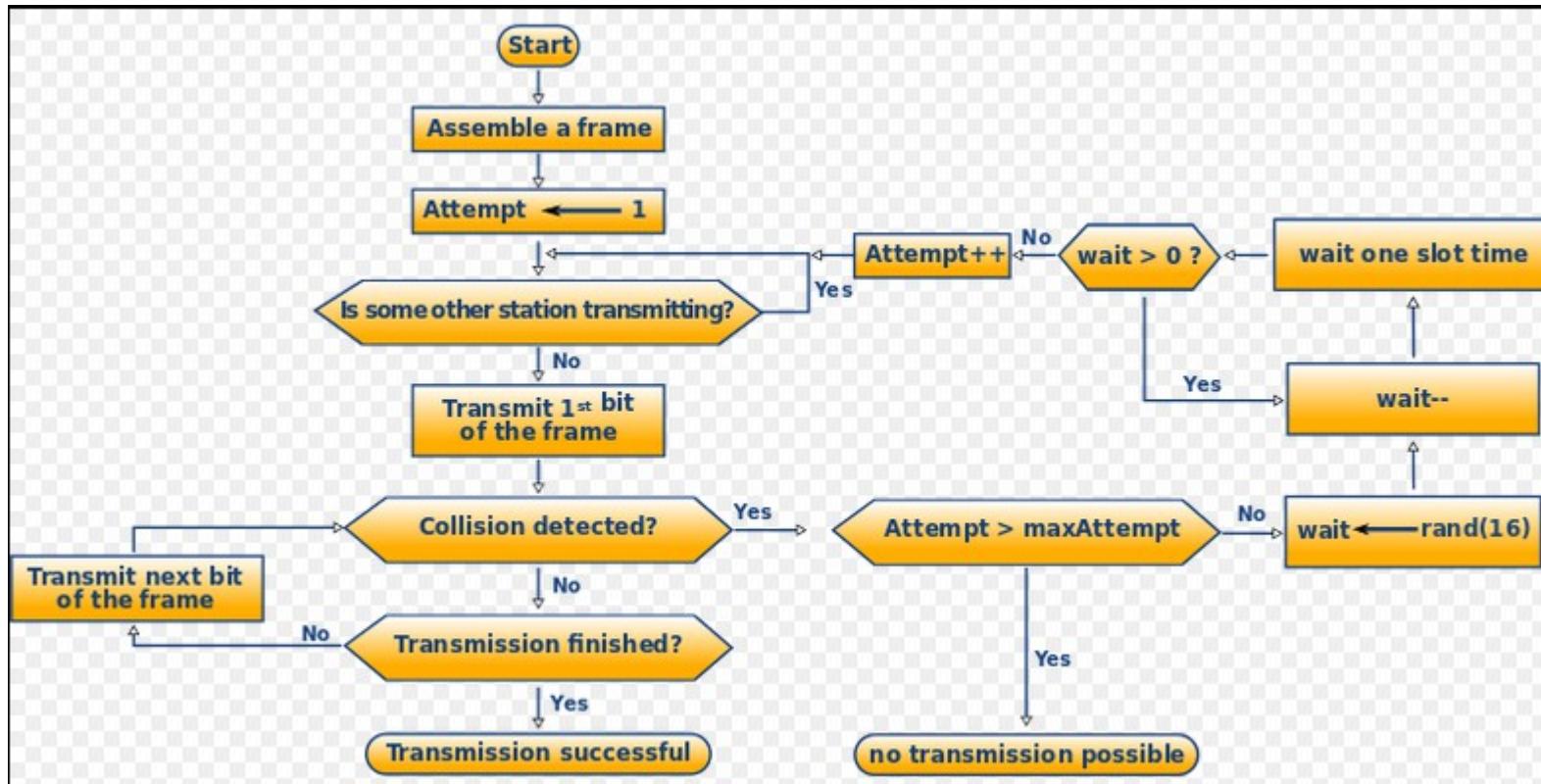


Warstwa 2
dwie podwarstwy:
LLC i MAC

Warstwa 1

Eth „magistralowy”/ zasada działania: CSMA/CD

Carrier Sense Multiple Access w. Collision Detection



Jeśli łącze zajęte to wybiera się losowo czas z przedziału [0, 2^N)

dla N=0..10 dla każdej próby, dla N=11..15, z przedziału[0, 1024)
tzw „backoff time”; czeka się k jedn czasu (k * 51.2 mikro-sek)

Min długość ramki eth = 64 bajty, z tego wynika max średnica eth magistralowego,
bo wykryć kolizję można tylko w czasie nadawania ramki...

(kolizja musi dotrzeć do nadawcy zanim skończy nadawać)

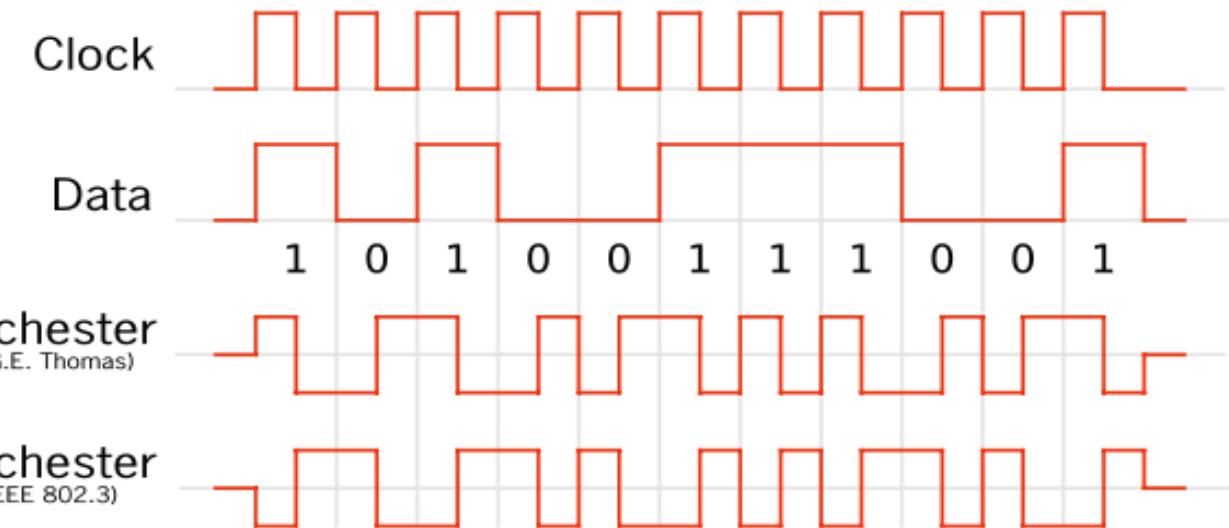
to NIE dotyczy eth „skrętka + switch „

Skrętka: kolizje tylko na 1 kablu, w trybie half-duplex...

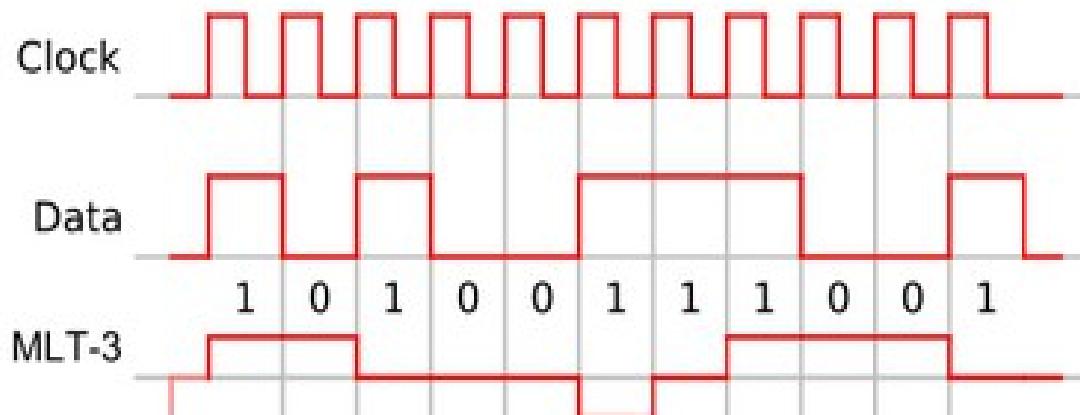
Eth, warstwa 1, baseband (kodowanie)

jak wygląda sygnał na kablu ???

Eth „magistralowy”:
kod Manchester
chodzi o „przemienność” prądu



Eth „skrętka + switch” (100BASE-TX):
„4B5B” - 4 bity zapisane za pomocą 5 bitów oraz...
„MLT-3” - przechodzi cyklicznie przez -1,0,1,0; „1” nast. wartość, „0” ta sama:



Switch eth (przełącznik)

Typy switch-y:

store-and-forward, tańszy, wczytuje całą ramkę do pamięci

cut-throught, czyta nagłówek (adr dst), bezpośrednio przekazuje bity

Pamięć switch-a:

adr eth kryjące się za portem, bufor ramek (walka z przeciążeniem),
specjalna pamięć CAM = content addressable mem, droga

Kontrola przepływu:

half duplex – sztuczne kolizje, full duplex – ramka PAUSE
(ramka PAUSE zawsze zawiera czas wstrzymania nadawcy)

Drzewo spinające:

musi być, z uwagi na broadcast; STP (spanning tree prot), 802.1D

Inne uwagi o switch-u:

- może przyjmować/nadawać ramki na wsz portach równocześnie
- nie ma kolizji (o ile na skrótce jest full duplex, Fast eth – TAK)
- gdy wiele ramek idzie do tego samego portu to są buforowane
- switch „uczy się” przy pomocy adr src w ramkach

Eth/ uzupełnienia

Tłumienie nadawcy/ obrona przed przeciążeniem łącza
ramka PAUSE (802.3x) lub wymuszanie kolizji w starszym eth...

Drzewo spinające, STP, 802.1D
STP = spanning tree prot...

Link aggregation, 802.3ad
kilka kabli między dwoma switch-ami (poza tym drzewo...)
2 kable 100Mb/s = 200Mb/s
można też łączyć wewn magistrale switch (tych dużych)

FCS = Frame Check Sequence w eth: CRC
Jakie są inne możliwości ?
Rs232/bit parzystości i uogólnienia, kody Hamminga, ...

VLAN = podział sieci fizycznej eth na wirtualne sieci fizyczne
np. na 1 dużym switch-u
pozwala zmniejszyć „domenę rozgłoszeniową”
przełączanie w warstwie 3 (sprzętowe routery ?)

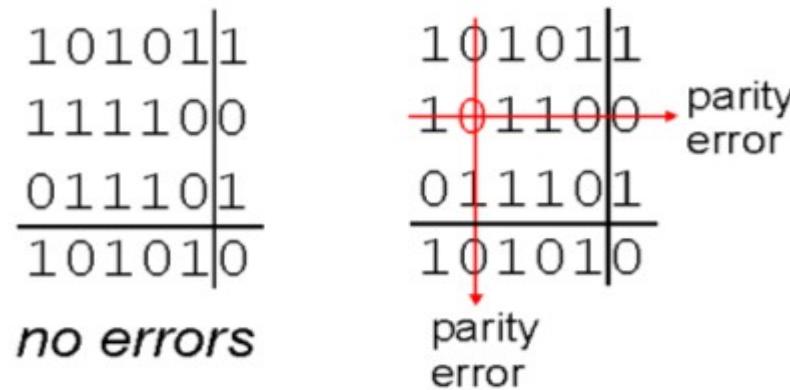
Częste pomyłki:
- domena kolizyjna vs domena rozgłoszeniowa
- różnica między routerem a switchem, jeśli chodzi o budowę
(switch bardziej „sprzętowy”)

Error Detection and Correction (1)

Oprócz danych, przesyłamy/ przechowujemy dodatkowe dane: redundancja

RS232, łącze szeregowe, bit parzystości:

parzystość liczby jedynek w bajcie, wykrywa 1 błąd (przekłamanie bitu)
uogólnie pozwalające naprawić ten błąd:



Suma kontrolna, checksum, RFC 1071

używane przez IP, TCP, UDP

dane traktuje jako słowa 16bit, ich suma obliczona i zapisana w sys U1

CRC, używane przez eth,

dane jako wsp wielomianu nad ciałem GF(2), dzielone przez spec wielomian
crc to reszta z tego dzielenia (wsp)

można to łatwo realizować sprzętowo (karta eth to robi !!)

jak duży błąd zostanie zauważony przez crc? crc16, crc32 ??

Error Detection and Correction (2)

Kody Hamminga...

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11	p16	d12	d13	d14	d15	
Parity bit coverage	p1	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
	p2		x	x		x	x		x	x		x	x		x	x		x	x		...	
	p4			x	x	x	x				x	x	x	x						x		
	p8					x	x	x	x	x	x	x	x	x								
	p16															x	x	x	x	x		

p_i - bit parzystości dla oznaczonych bitów w tabeli, d_j – j-ty bit danych,
Tutaj: 15 bitów danych, 5 bitów parzystości,
Przykład: jeśli p_1, p_2, p_8 wskazują błąd, to jest on na bieżie p_{11} ($11=1+2+8$)

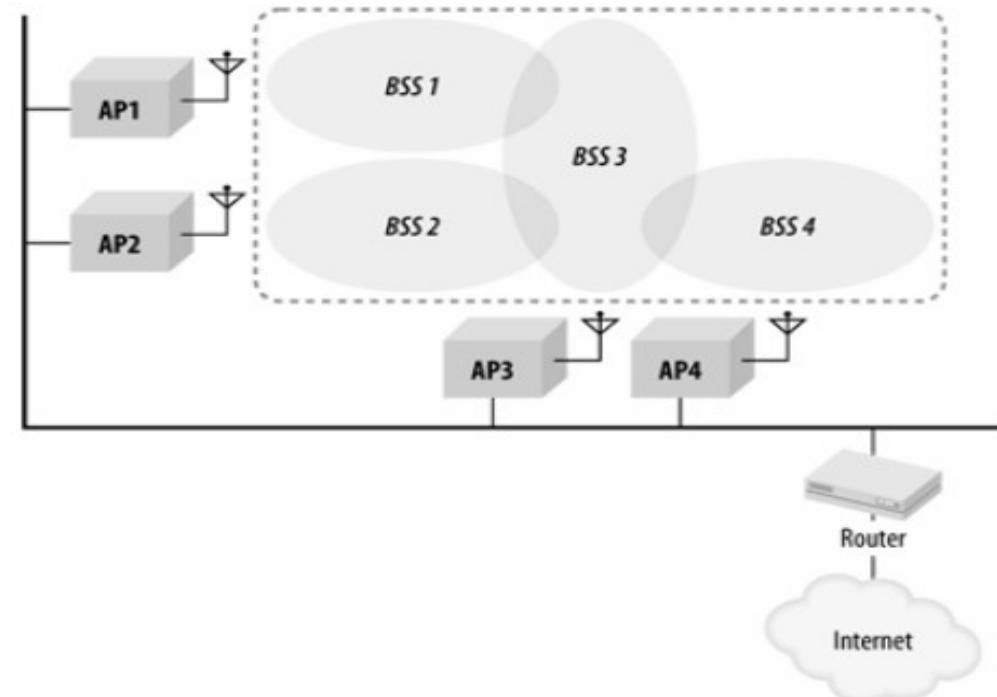
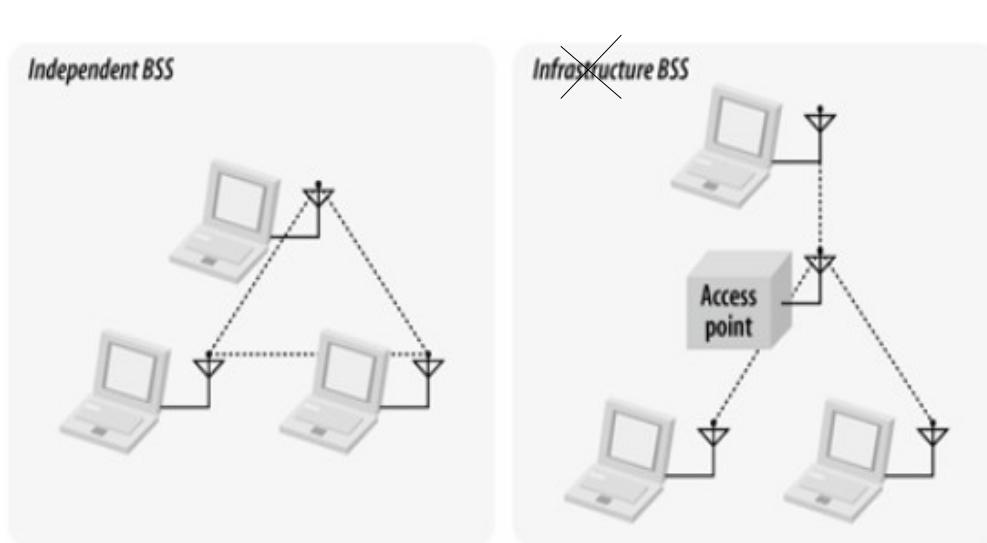
WiFi – podstawowe pojęcia

Bezprzewodowa sieć fizyczna...

używa jako medium przestrzeni i fal elektromagnetycznych,
„fale radiowe”, mikrofale, okolice 2.4GHz lub 5GHz,
standardy IEEE 802.11, 802.11a/b/g/n/ac (z literką dotyczą warstwy fizycznej)
WEP, WPA/WPA2, 802.11i – bezpieczeństwo (szyfrowanie, istotne z powodu...)

Access point (AP), stacja (STA, klient wifi),
BSS(Basic Service Set, 1x AP + stacje, dane za pośrednictwem AP) ,
IBSS (Independent BSS, „ad-hoc”, bez AP),
ESS (Extended Service Set, wiele BSS + system dystrybucji, **jeden „nr sieci”**),
System dystrybucji (zazw. eth; most łączący wifi i eth ?),

Figure 2-5. Extended service set



WiFi – podstawowe pojęcia c.d.

(E)SSID - nazwa sieci, ma ją każdy AP, w ESS powinny być identyczne, ramka beacon
BSSID – adr sprzętowy AP, stacje wifi także mają adr sprzętowy

Adr sprzętowy wifi: tak sam jak eth!! 6 bajtów... też możliwy multi/broad-casting...

Kanały wifi, których używa BSS... jest ich 13, w BSS używa się 1 kanału !!,

bliskie BSSy powinny używać innych kanałów !!! kanały nie są odseparowane...

Bezpieczeństwo: stare złe rozwiązańe WEP (4 klucze),

nowe dobre rozwiązańe WPA/WPA2=802.11i (wpa_supplicant)

Linux: interfejsy sieciowe wifi mają nazwy postaci: wlan0, ...

Wyświetlanie widocznych AP: root# iwlist wlan0 scan

Podłączanie się do AP: root# iwconfig wlan0 essid „SSID/nazwa sieci”

Wyświetlanie kanałów: root# iwlist wlan0 chan

wlan0 13 channels in total; available frequencies :

Channel 01 : 2.412 GHz

Channel 02 : 2.417 GHz

Channel 03 : 2.422 GHz

Channel 04 : 2.427 GHz

Channel 05 : 2.432 GHz

Channel 06 : 2.437 GHz

Channel 07 : 2.442 GHz

Channel 08 : 2.447 GHz

Channel 09 : 2.452 GHz

Channel 10 : 2.457 GHz

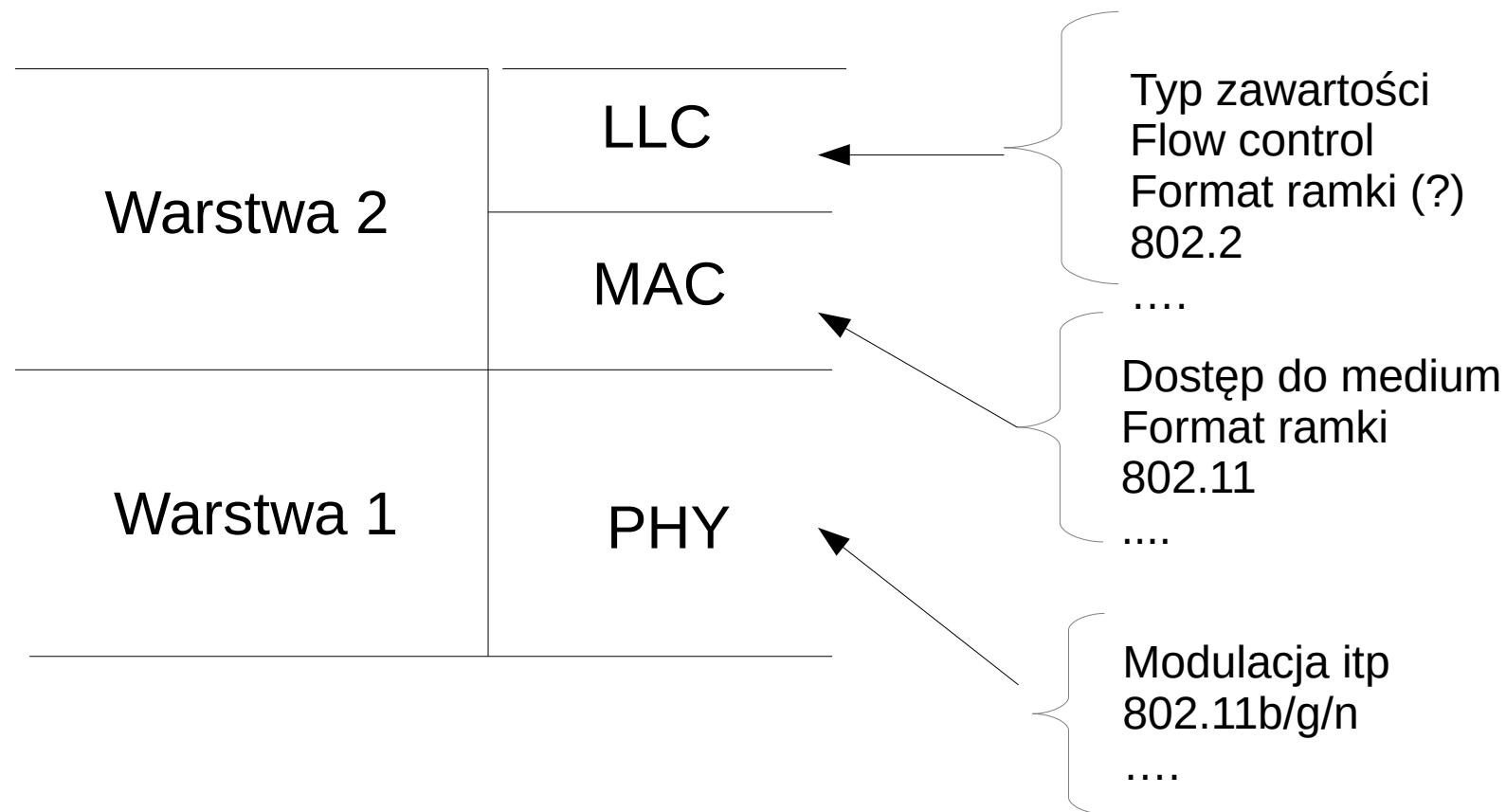
Channel 11 : 2.462 GHz

Channel 12 : 2.467 GHz

Channel 13 : 2.472 GHz

Current Frequency: 2.427 GHz (Channel 4)

Podział warstw 1 i 2 na podwarstwy...



Wifi, warstwa 2, podwarstwa MAC

Dostęp do medium:

DCF = Distributed Coordination Function = CSMA/CA

PCF = Point -"- bez rywalizacji o dostęp, HCF = posiada elem QoS

CSMA/CA = Carrier Sense Multiple Access with Collision Avoidance

w eth było CSMA/CD ! w wifi **NIE MA** wykrywania kolizji w czasie nadawania !!

są pozytywne potwierdzenia ACK otrzymania ramki

jest okno Backoff losowej rywalizacji o dostęp do łącza (contention window),

stacje losowo wybierają slot z „contention window” i czekają odp czas,

okno zwiększa długość przy każdej nieudanej próbie wysłania ramki

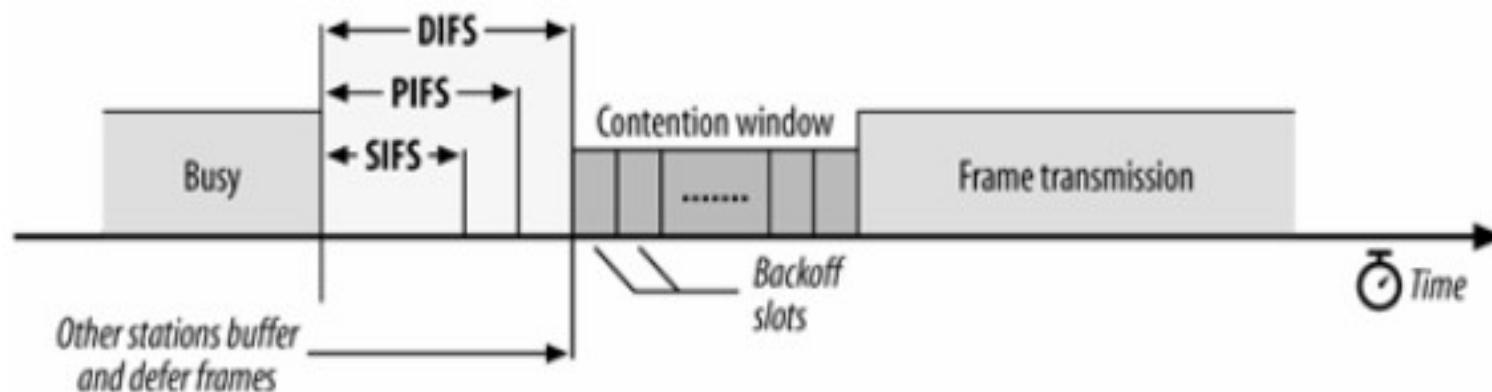
są odstępy czasowe między ramkami o różnej długości: DIFS, SIFS, EIFS, ...

jeśli nośnik był wolny przez czas DIFS + czas w oknie rywalizacji

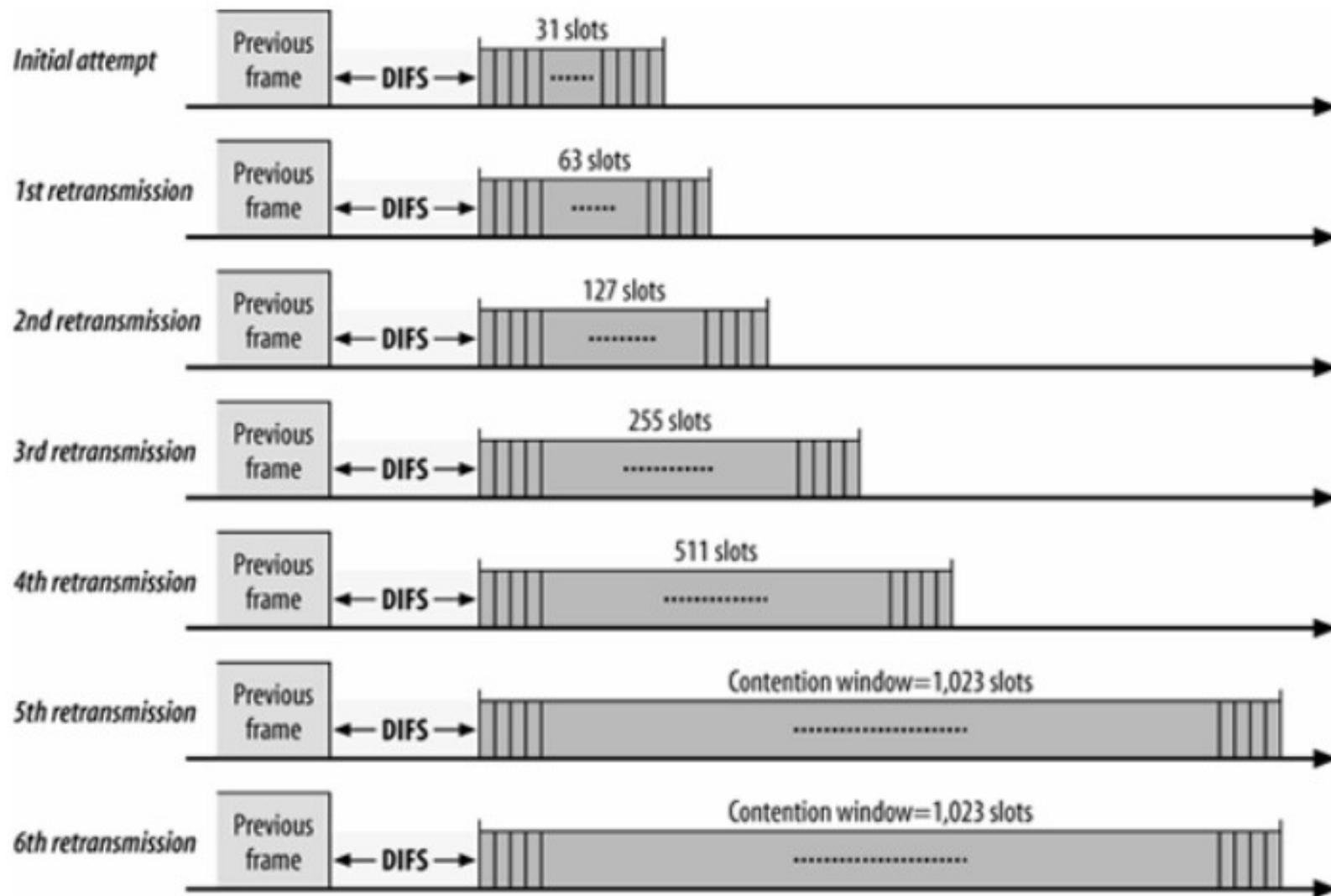
to można nadawać (jeśli był błąd to EIFS zamiast DIFS)

chodzi o to aby 1 stacja zaczęła nadawać z wielu próbujących...

są dodatkowe mechanizmy „rezerwowania” medium: NAV i RTS/CTS



Zmiana długości okna rywalizacji przy retransmisji (brak ack)



Wifi, warstwa 2, podwarstwa MAC, c.d.

Inne rozwiązania pozwalające uniknąć błędnych ramek (rezerwowanie medium):

prot RTS/CTS (Request To Send, Clear To Send), krótkie ramki do rezerw łącza inaczej: do „uciszania” innych stacji, znane z łącza szeregowego rs232...

Linux: próg RTS, chodzi o długość ramki, iwconfig wlan0 rts; jest też próg fragm!

wektor NAV: ramki mają pole Duration, w którym informują o czasie nadawania...

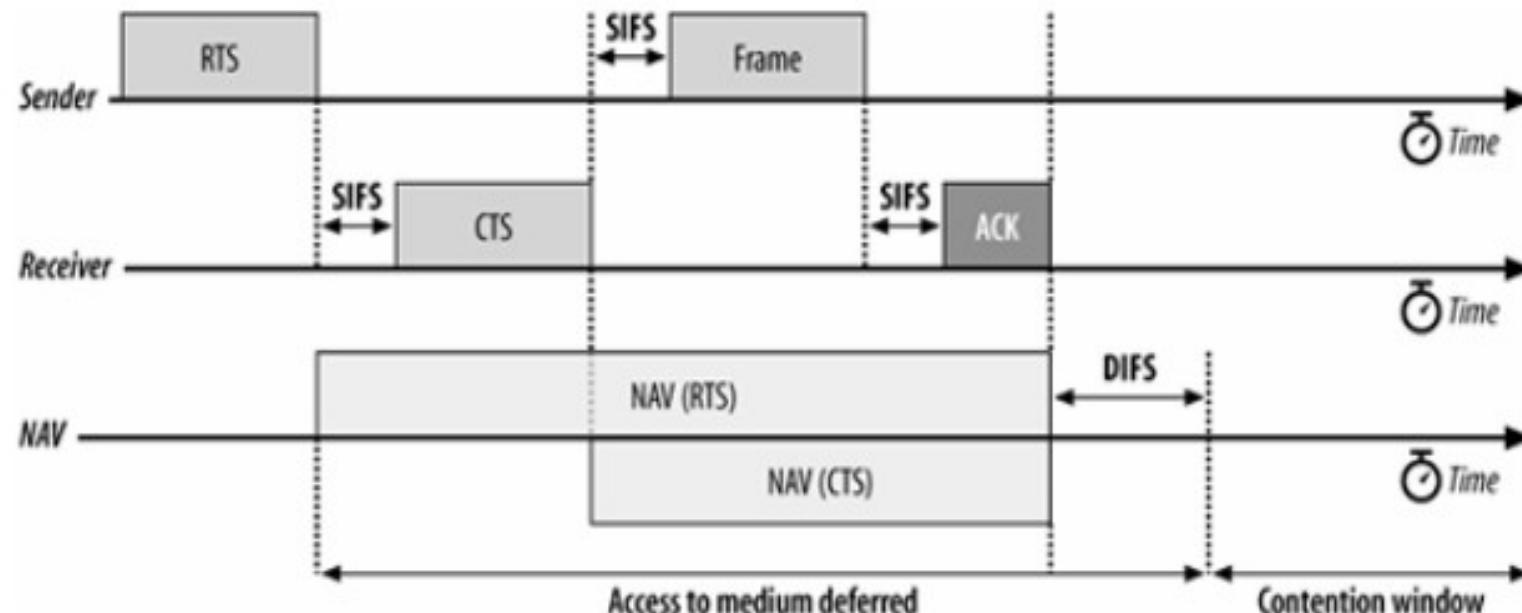
wszystkie stacje które widzą taką ramkę modyfikują swój wektor NAV który mówi jak długo medium będzie zajęte przez inną stację...

komunikacja nie tylko stacja – AP, ale także stacja – stacja !!!

(choć dane są zawsze przesyłane za pośrednictwem AP)

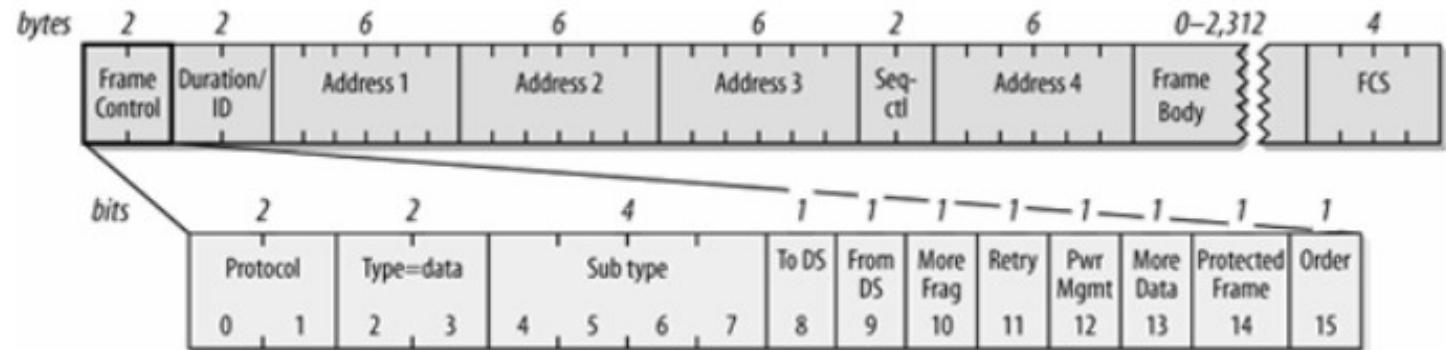
RTS/CTS rozwiązuje problem dalekich stacji:

1 stacja wysła rts, AP wysyla cts, 2 stacja widzi cts (choć nie widzi rts...)



Wifi, warstwa 2, podwarstwa MAC, c.d.

Format ramki wifi:



Co się znajduje w ramce ?

Typ i podtyp, b. dużo podtypów, typy ramek:

mgmt (beacon, auth, assoc, ...)

control (rts/cts, ack, ...),

data (max d<ług> danych= 2304, 2296 z powodu nagł LLC/802.2)

Adresy:

dst, src, transmitter(?), bssid; interpret zależy od typu ramki !!

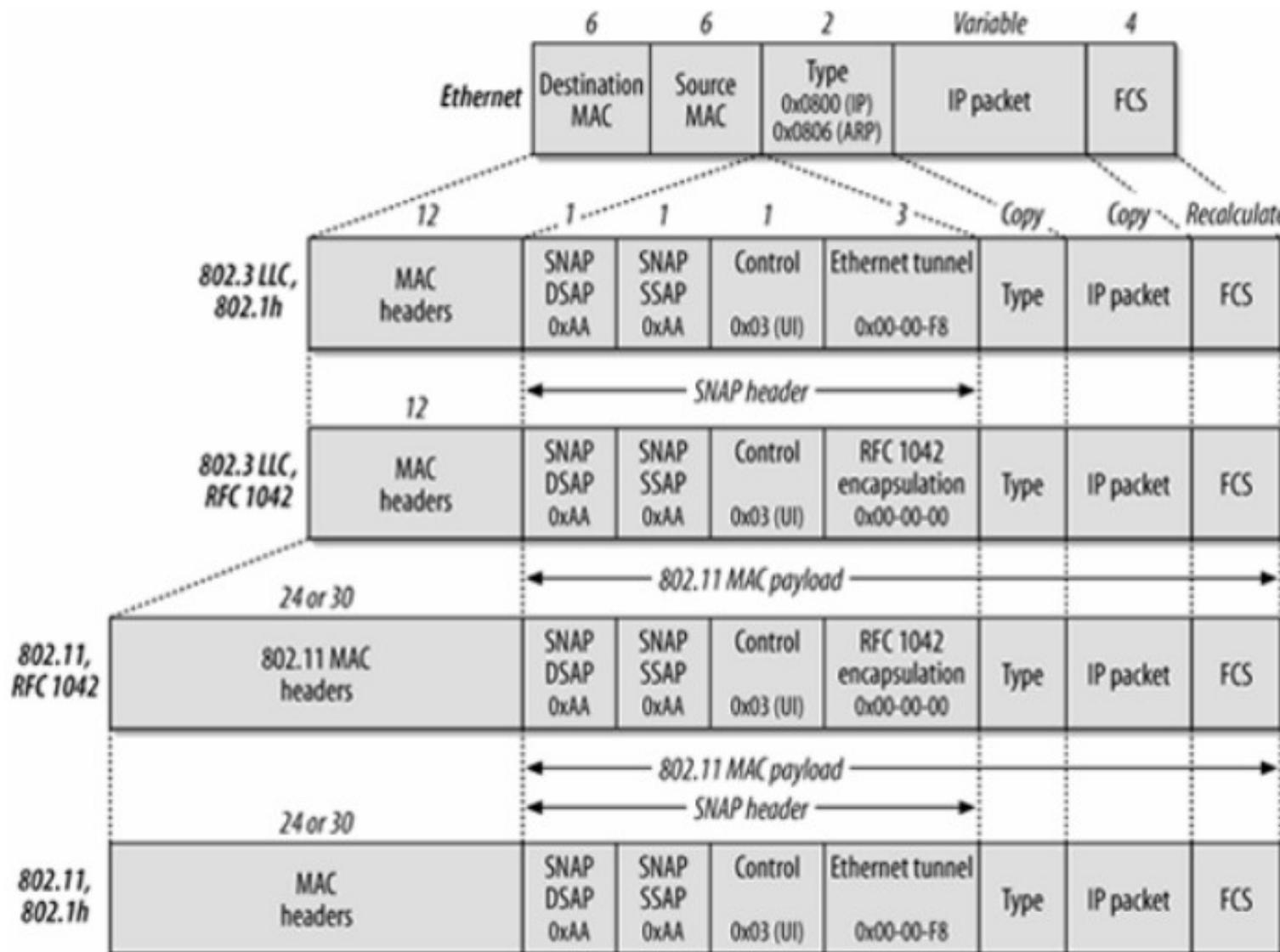
dst może być w tym samym BSS lub innym w ramach ESS

bssid jest po to aby odróżniać ramki naszego i cudzego BSS...

ramka może przeskakiwać między: wifi → eth → wifi (zmienia się adr3)

Zmiana ramki przy przeskakiwaniu wifi → eth → wifi w ESS...

Figure 3-13. IP encapsulation in 802.11



Wifi, warstwa 1 = PHY

Uzyskaliśmy dostęp do mediu, jak przesyłać ramkę ???

Table 17.4 IEEE 802.11 Physical Layer Standards

	802.11	802.11a	802.11b	802.11g
Available bandwidth	83.5 MHz	300 MHz	83.5 MHz	83.5 MHz
Unlicensed frequency of operation	2.4–2.4835 GHz DSSS, FHSS	5.15–5.35 GHz OFDM 5.725–5.825 GHz OFDM	2.4–2.4835 GHz DSSS	2.4–2.4835 GHz DSSS, OFDM
Number of non-overlapping channels	3 (indoor/outdoor)	4 indoor 4 (indoor/outdoor) 4 outdoor	3 (indoor/outdoor)	3 (indoor/outdoor)
Data rate per channel	1, 2 Mbps	6, 9, 12, 18, 24, 36, 48, 54 Mbps	1, 2, 5.5, 11 Mbps	1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48, 54 Mbps
Compatibility	802.11	Wi-Fi	Wi-Fi	Wi-Fi at 11 Mbps and below

Wifi, warstwa 1 = PHY

„SS” = Spread Spectrum, rozpraszanie widma,
użycie szerokiego zakresu częstotliwości fal radiowych (vs 1 częstotliwość nośna...)
w celu 1. uniknięcia zakłóceń, 2. zwiększenia przepustowości

FHSS = Frequency Hopping SS

skakanie po częstotliwościach wg pewnego wzorca, nadal używane przez Bluetooth !!

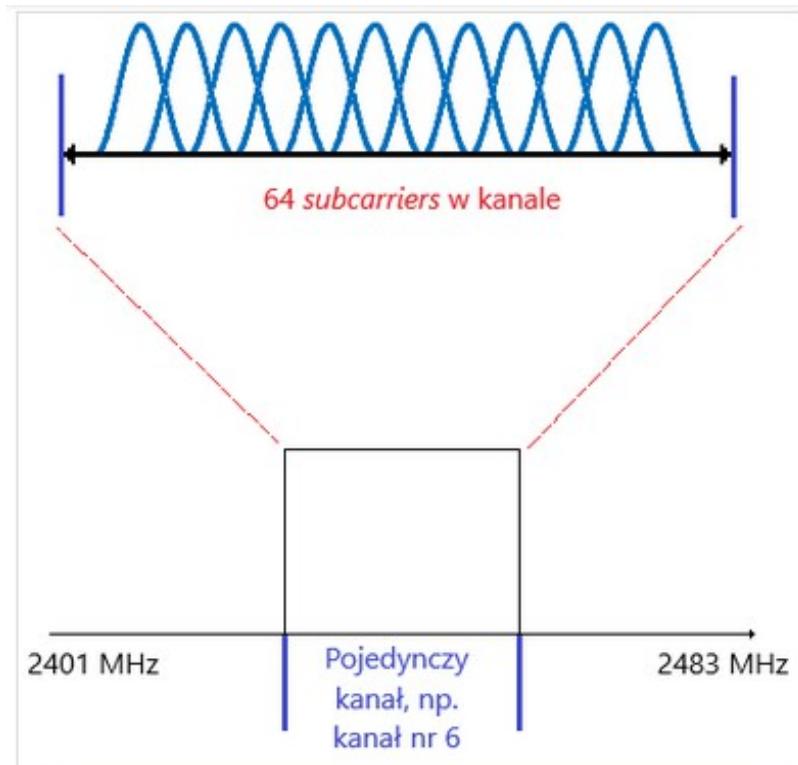
DSSS = Direct Sequence SS

bit zamieniamy na ciąg 11 bitów, kod Barkera „0” = 1 0 1 1 0 1 1 1 0 0 0, „1” negatyw
przesyłamy je równolegle 11 pod-kanałami (dlatego kanały DS są szerokie...)

OFDM = jak FDM (Frequency Division Multiplexing),
ale częstotliwości nośne są ortogonalne...

mogą być bliżej siebie! jest ich więcej !!
mamy więcej kanałów,
którymi możemy przesyłać dane...

OFDM →



Podział kanału na 64 subcarriers, każda o szerokości 312,5 KHz

Wifi, warstwa 1 = PHY, c.d.

Jak przesyłać ciąg danych (bitów) przez pod-kanał?

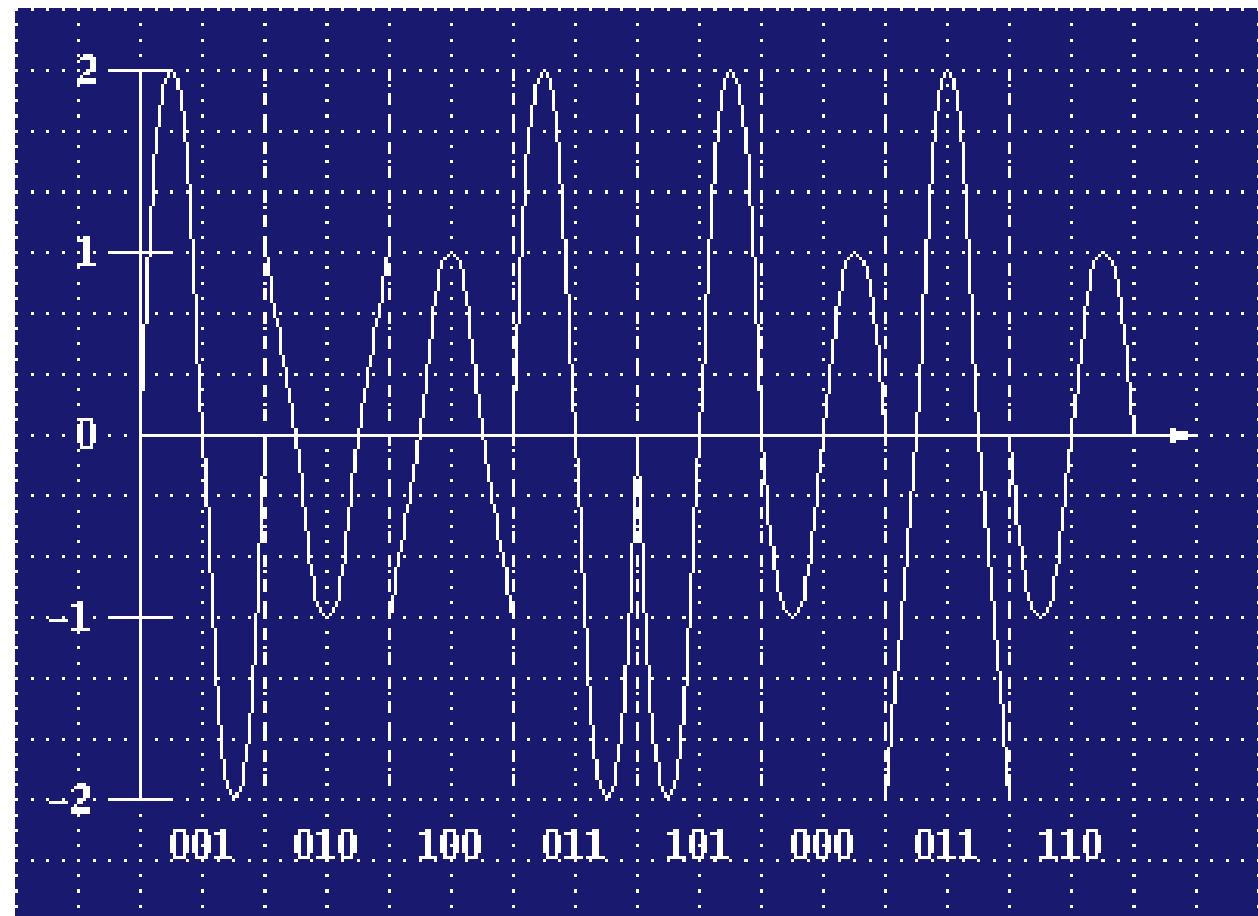
czyli modulowanie częstotliwości nośnej przez ciąg bitów...

są różne rozwiązania: AM (modulacja amplitudowa), FM (częstotliwościowa)

PSK (fazowa), mieszane, np. QAM (amplitudowo/fazowa, patrz tzw „konstelacje”):

Bit value	Amplitude	Phase shift
000	1	None
001	2	None
010	1	1/4
011	2	1/4
100	1	1/2
101	2	1/2
110	1	3/4
111	2	3/4

8-QAM



Wifi, bezpieczeństwo

Zawsze na początku:

1. autentykacja (open, shared key/ **tego nie używać**)
2. asocjacja (powiązanie z jednym AP)

WEP (**tego nie używać**), Wired Equivalent Privacy

szłyfr strumieniowy RC4, zbyt dużo danych się ujawnia w ramkach !!!
4 klucze, nr klucza jest w ramce !

WPA/WPA2, 802.11i (tego używać), Wifi Protected Access

„Personal” WPA/WPA2-PSK PreShared Key

pojedynczy klucz używany przez AP i stacje

„Enterprise” używa się EAP (znanego też z PPP), 802.1X

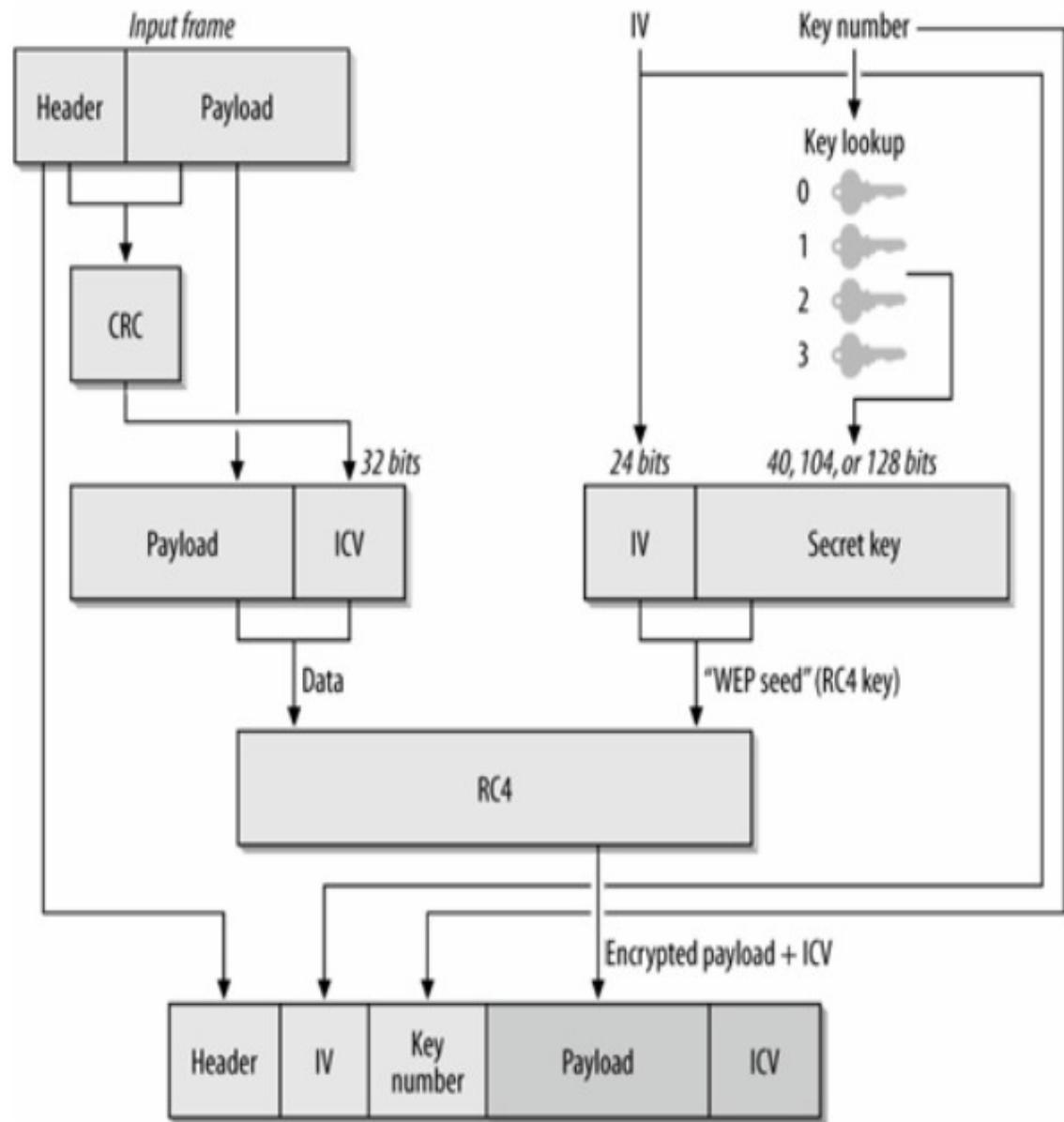
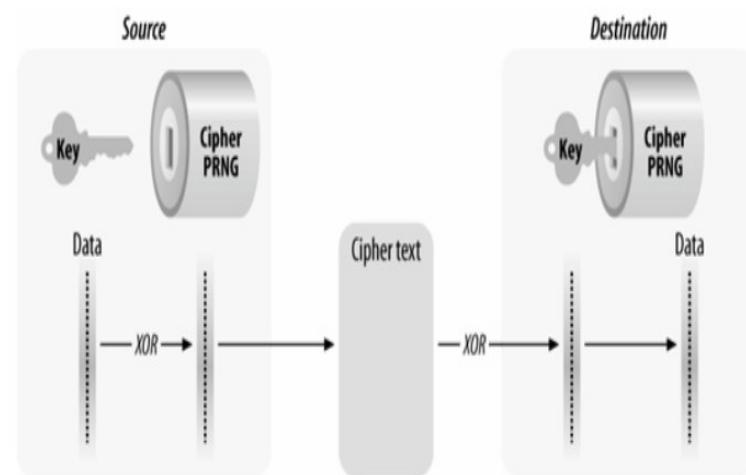
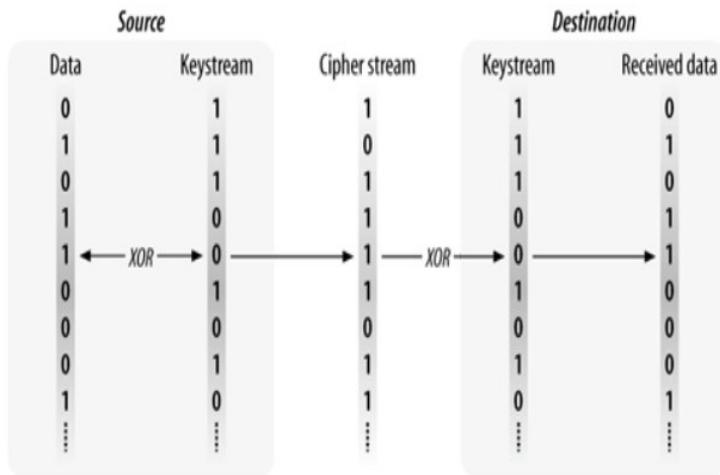
komunikaty EAP, negocjacja sposobu uwierzytelnienia użytkowników,

wpuszczanie użytkowników do sieci wifi na podstawie danych z ser aut,

3 składniki:

1. suplikant (wpa_supplicant),
2. autentykator (AP),
3. serwer autentykacji (RADIUS)

Wifi, WEP



Wifi w Linux-ie

Polecenia linuxowe:

```
ifconfig wlan0 up  
iwconfig wlan0  
iwconfig wlan0 mode managed/ad-hoc/monitor  
Iwconfig wlan0 essid <nazwa sieci>  
Iwconfig wlan0 chan <nr kanału>  
iwlist wlan0 X/ X=scan, chan, keys, ...  
wpa_supplicant, wpa_passwd (łatwa obsługa wpa/personal)
```

Jak się używa trybu ad-hoc ?

wł mode=ad-hoc, nadać nazwę 1 węzłów przez subcmd essid,
pozostałe węzły podłączają się też przez subcmd essid...
można używać WEP... (nie można WPA ?!)

Jeśli chodzi o implementacje w linuxie...

Wext (stare), nl80211, moduły cfg80211 mac80211 (nowe),
używa się gniazdek „netlink” (komunikacja user space - kernel)

Bluetooth

Spec IEEE 802.15, RF 2.4GHz, zasięg 10m (klasa 2), zastępuje kabel rs232...

Przepustowość: ok 1Mb/s (721kb/s, w nowszych spec więcej...)

Warstwa fiz/radiowa: FHSS (skakanie po kanałach, 79 kanałów)

Zabezpieczenia? Parowanie urządzeń bt (linux: bluetooth-agent <kod>)

Master/Slave: 1 master <=7 slaves, komunikacja wyłącznie M-S (nie S-S),

adr sprzętowe 6 bajtów (jak eth/wifi), adr M determinuje schemat FH !!!

wymiana komunikatów między M a S (kilka slotów czasowych, TDM)

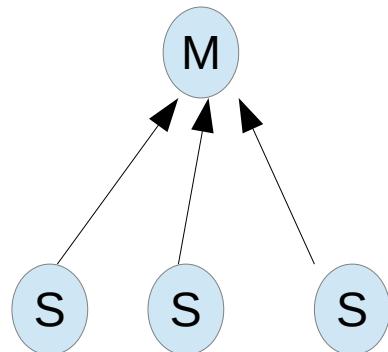
nadaje M lub jeden z S (wszystkim rządzi M)

Podstawowa sieć: **piconet**, wiele piconet-ów to **scatternet**

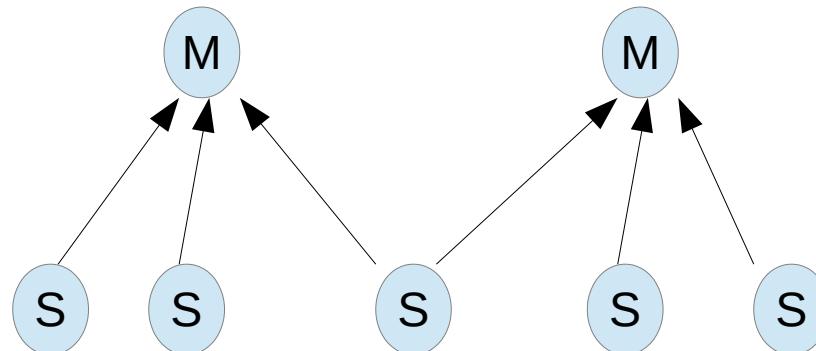
1 węzeł może być w wielu piconetach!!

problem formowania/obsługi scatternet nierożw. !!

dzięki FH (freq hop) jest szansa, że bliskie pikonetły używają innej częstotliwości w chwili t



piconet



scatternet

Bluetooth c.d.

Stos proto (nietypowy!!), profile (=usługi)

RF – warstwa radiowa, 2.4 GHz, 79 kanałów,
FHSS,

Baseband – odp MAC, ramki, *nazwa myląca!!*
sloty czasowe 625 mikrosek,
ramki mogą być wielokrotnie slotów,
wymiana ramek między M i S (runda?),
2 rodzaje logicznych kanałów:

ACL (asynch datagramy) i SCO (synch, audio)

LMP ???

L2CAP = Logical Link Control Adaptation Prot
pkg o rozmiarze do 64kB, podobne do UDP??

RFCOMM – emulacja łącza szeregowego nad bt
nad poł rfcomm można uruchomić prot PPP
podobne do TCP??

Profile – rodzaj usług...

DUN dial-up networking

OBEX FTP przesył plików, prot OBEX

SDP ogłoszanie dostępnych profili

PAN, PANU sieć nad bt...

Audio – np. słuchawki,
obsługiwane bez l2cap, bezp nad baseband...

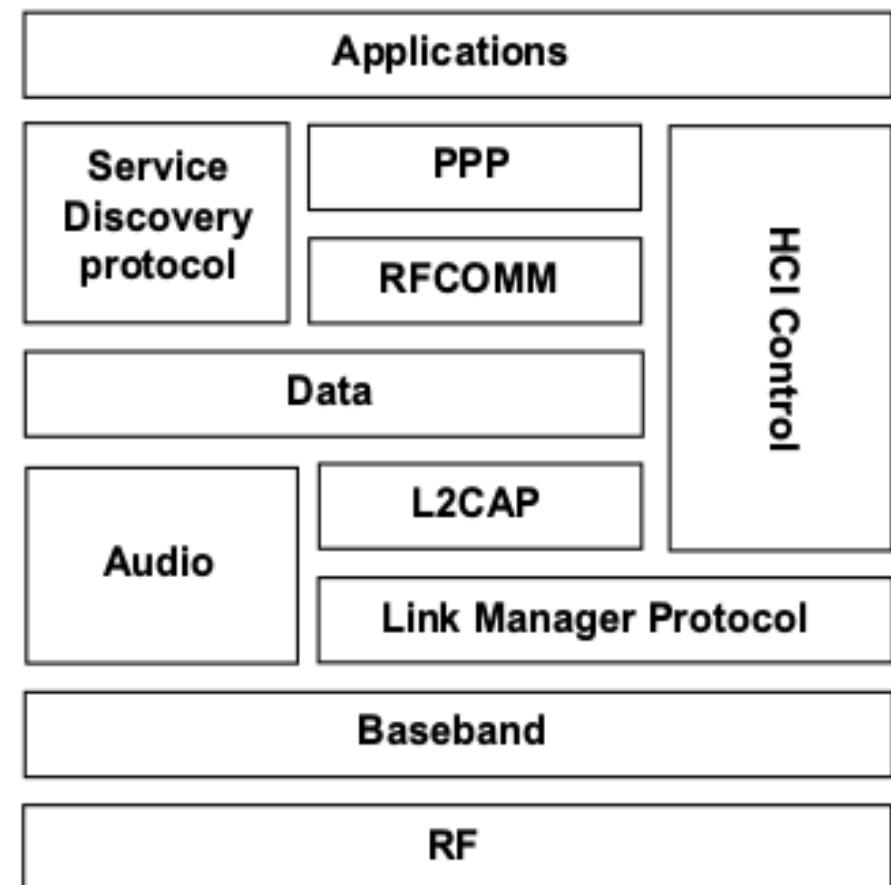


Fig. 1. Organisation of the BT stack.

Bluetooth c.d.

Impl linuxowa prot bt: BlueZ...

Komendy linuxowe:

bluetoothd

bluetooth-agent <kod>

hcitool dev; # adr_BT naszego urządzenia

hcitool scan; # wykrywanie sąsiednich urządzeń

hcitool con; # pokazuje połączenia M-S i typ kanału (ACL/SCO)

hcitool sr; # zamiana M/S

rfcomm connect <X> <adr_BT> <kanal>; # powstanie plik /dev/rfcommX

rfcomm listen rfcommX <kanal>

sdptool browse <adr_BT>; # pokazuje profile/usługi dostępne na urządzeniu...

Service Name: Dial-Up Networking

Service RecHandle: 0x10026

Service Class ID List:

"Dialup Networking" (0x1103)

Protocol Descriptor List:

"L2CAP" (0x0100)

"RFCOMM" (0x0003)

Channel: 22; # MH: rfcomm/ kanał 22 – to jest modem GPRS !!!

Language Base Attr List:

code_ISO639: 0x454e

encoding: 0x6a

base_offset: 0x100

Profile Descriptor List:

"Dialup Networking" (0x1103)

Version: 0x0100

Routing

Słowo „routing” dwa znaczenia:

1. przekazywanie pakietów, ”forwarding”
2. modyfikowanie tablicy routingowej (tym się teraz zajmiemy...)

Uproszczony obraz „intersieci”: każda sieć fiz ma 2 routery, istnieje „koszt łącza”

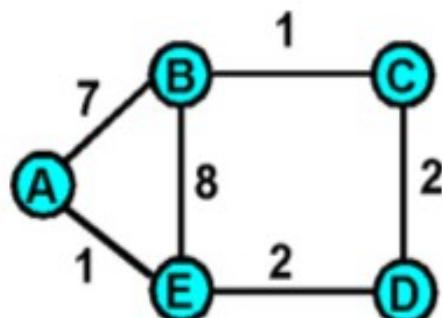
Dwa podejścia: **DV** (Distance Vector) „wektor odległość”, **LS** (Link State) „stanu łącza” opierają się na „algorytmie najkrótszych ścieżek między wsz parami wierz” (relaksacja)

Tablica routingowa danego węzła zawiera wiersze:
(węzeł_docelowy, sąsiad, odległość/metryka)

DV: każdy wierz wymienia z sąsiadami „tablicę routingową”
oraz aktualizuje swoją tablicę na podstawie otrzymanych informacji
(nowe lub lepsze trasy przez innego sąsiada/gw, itp.)
alg Bellman-Ford, relaksacja, „najkrótsze ścieżki z 1 źródłem do wszy wierz”

LS: każdy wierz sprawdza stan swoich połączeń z sąsiadami,
oraz rozsyła te informacje po całej sieci (czyli każdy wierz ma obraz całej sieci),
następnie każdy wierz lokalnie oblicza przez którego sąsiada prowadzi
najlepsza trasa do danego węzła (alg Dijkstry, relaksacja)

Routing – uproszczenie intersieci...



		cost to destination via			
		D()	A	B	D
d e s t i n a t.	A	(1)	14	5	
	B	7	8	(5)	
	C	6	9	(4)	
	D	4	11	(2)	

Uproszczony graf reprezentujący „intersieć”

Wierz grafu to routery (nie ma tu hostów!!)

Łącza mają wagę (mogą być =1), odległość jest ważona

Globalna macierz odległości (ważonych) z pkt widzenia wierz E

Kółkiem oznaczono minimalną trasę

Można ją łatwo przerobić na tabl routingową E...

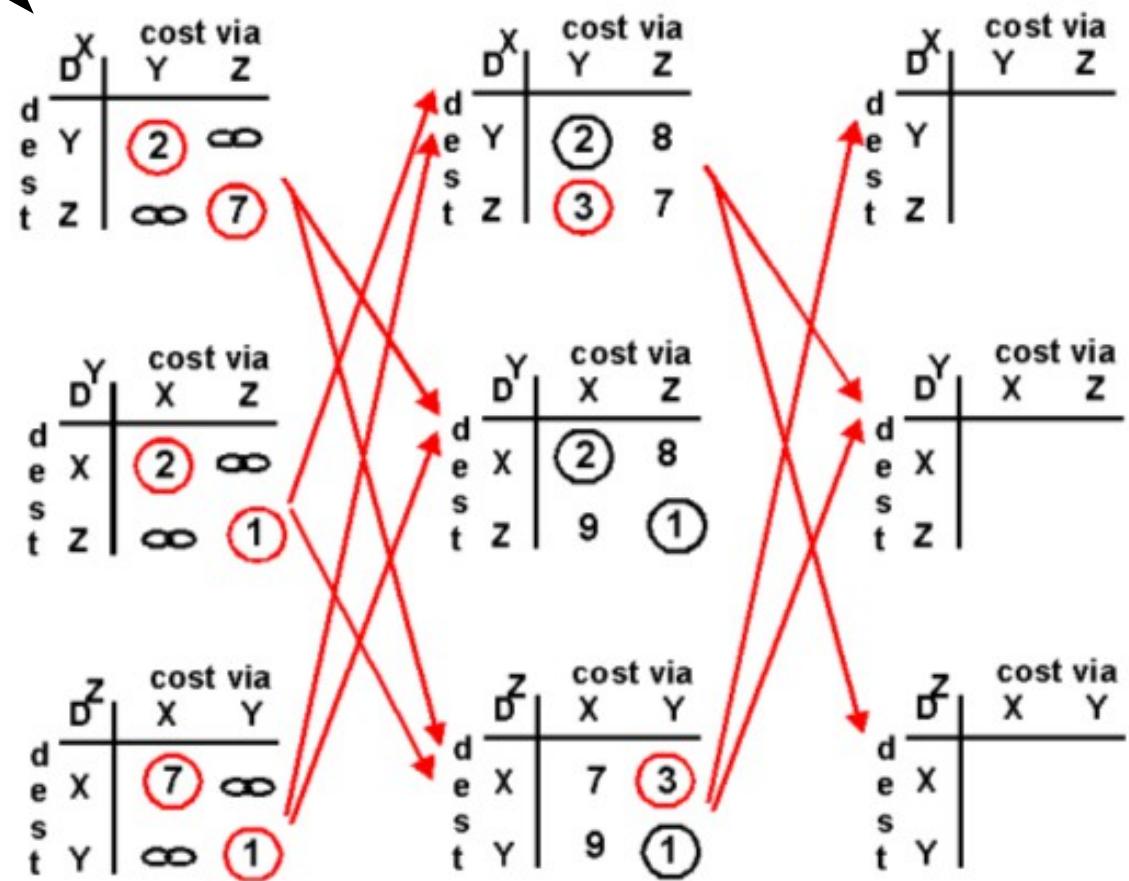
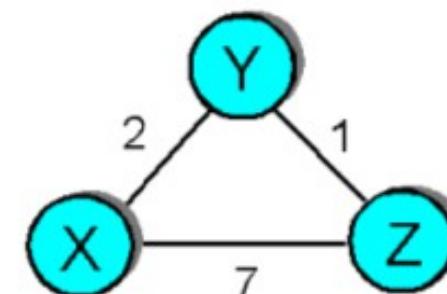
Routing DV

jak uaktualniemy tablice routingowe?

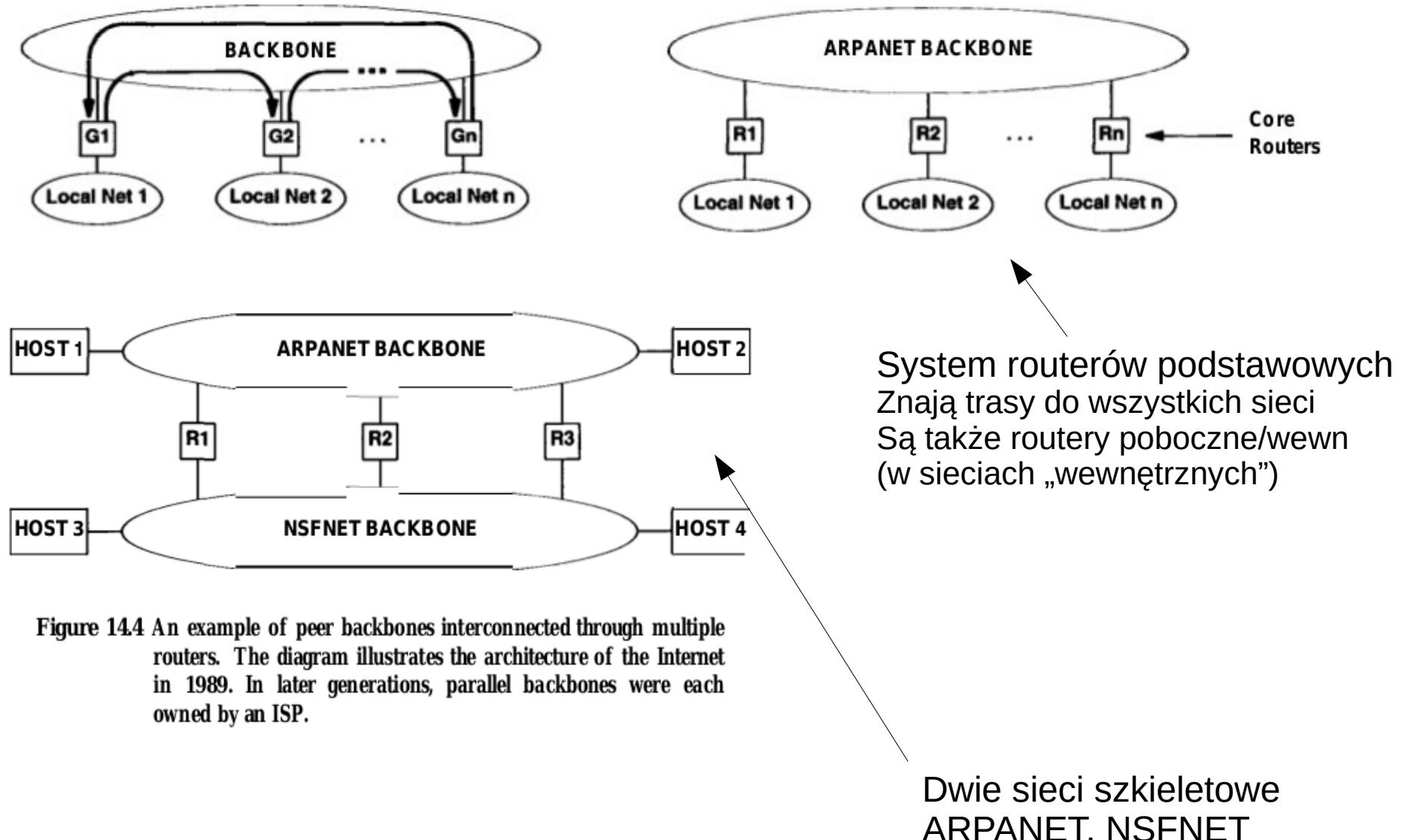
obraz pokazuje zmiany tabl routing
w czasie dzialania alg...
(tabl routing wyznaczaj kolkka!)

oznaczenie: $D^Z(X, Y)$ to odleglosc
od Z do X via sasiad Y
(Y jest sasiadem Z)

$D^Z(X, Y) = \text{nieskonczenosc};$
dostalismy info ze $\text{dist}(Y, X) = 2$
to wtedy uaktualniemy $D^Z(X, Y)$
 $D^Z(X, Y) := c(Z, Y) + \text{dist}(Y, X) = 3$,
bo $3 < \text{nieskonczenosc}$...

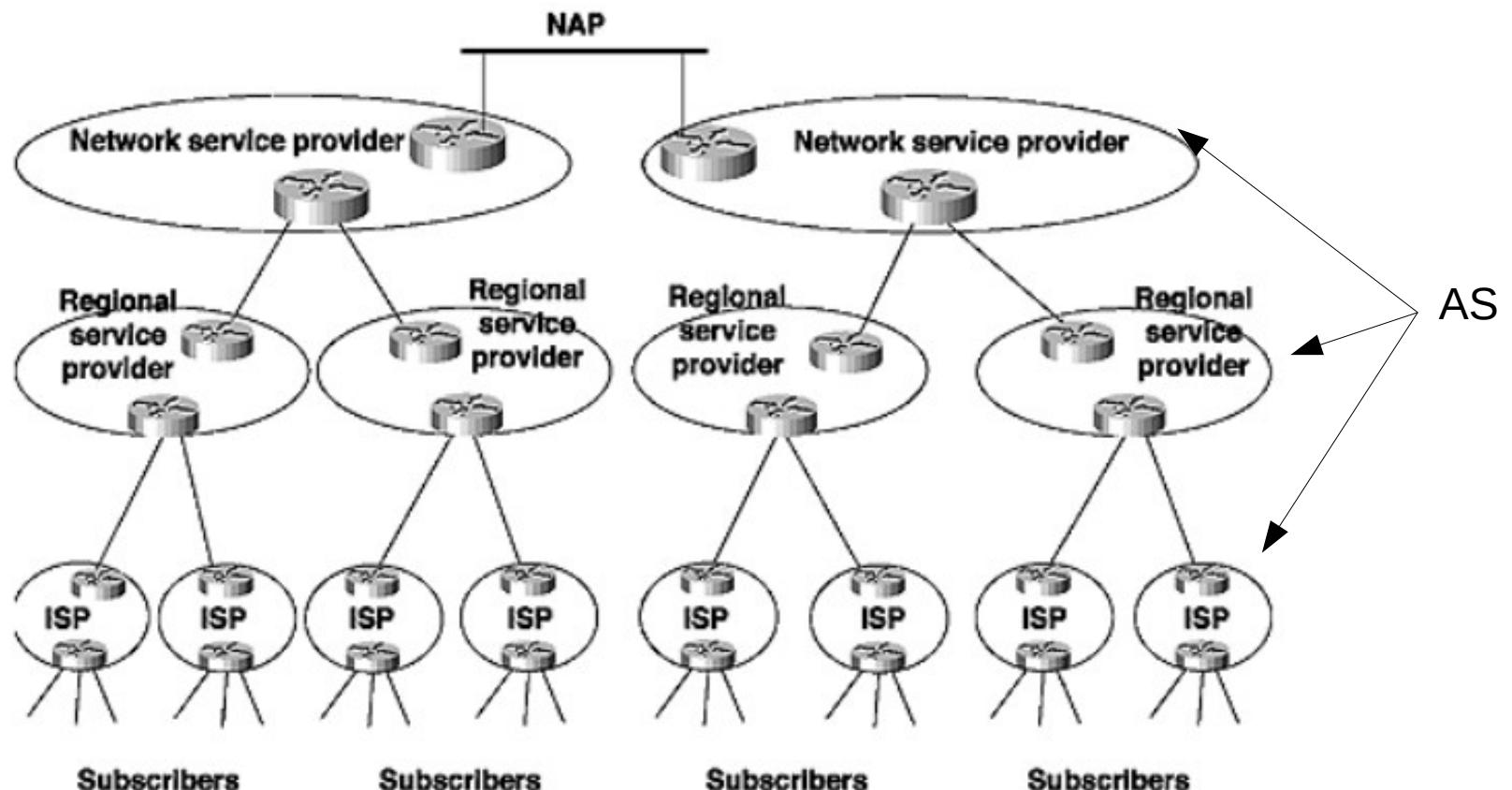


Routing - historia



Routing - stan obecny

Figure 2-6. ISP/ NAP Hierarchy



3 poziomy ISP (Internet Service Provider), NAP to obecnie IXP (Internet Exchange Point)

Routing i systemy autonomiczne (AS-y)

AS = ang. **Autonomous System**, zbiór sieci fizycznych (intersieć), zarządzana przez jedna instytucję, ze wspólną „polityką routingową”, z tym samym algorytmem routingu wewn

AS-y mają spec nr ASN (2 lub 4 bajty), nadawane przez RIR (dawniej IANA)

RIR nadaje też (publiczne adr IP, bloki adresów), osobne RIRy dla kontynentów (?)

Potrzebne dwa rodzaje routingu:

- **wewnętrzny**, wewnątrz AS, typu DV (np. RIP) lub LS (np. OSPF(?)
- **zewnętrzny**, między AS-owy, dostarcza info o osiągalności sieci przez sąsiednie AS-y... (BGP-4, dawniej „EGP”, uwaga na EGP/ nazwę ogólną !!)

Tablica routingowa routera pochodzi z 2 źródeł: od routingu wewn ORAZ zewn...

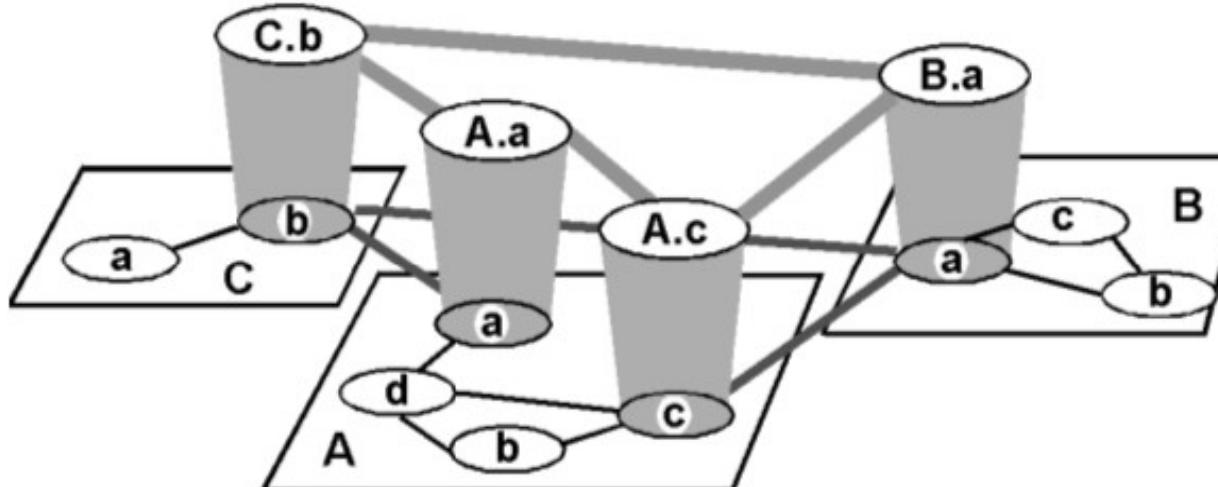


Figure 4.3-1: Intra-AS and Inter-A S routing.

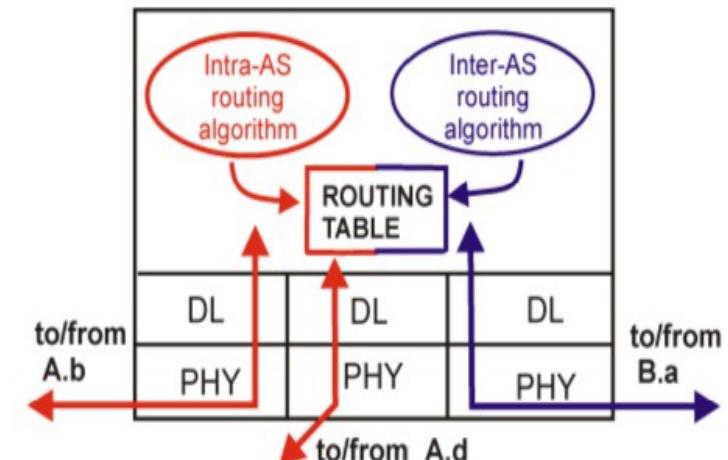


Figure 4.3-2: Internal architecture of gateway router A.c

Routing wewn/ RIP

Algorytm routingu typu DV; routing dynamiczny; używa UDP port 520
RIPv1 (IPv4 klasowe), RIPv2 (IPv4 bezklasowe), RIPng (Ipv6)

Węzły RIP aktywne i pasywne

Węzły aktywne co 30sek wymieniają tabl routingową z sąsiadami

Sąsiad? Węzeł RIP w sieci bliskiej; broadcast (same „1” w adr ip dst)

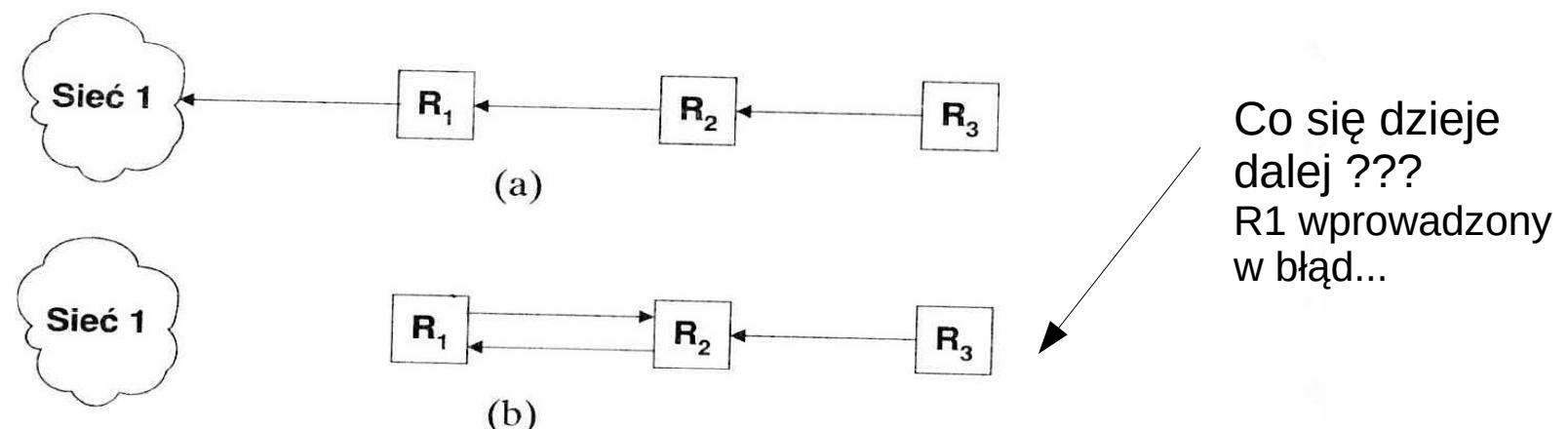
Koszt trasy = liczba etapów, tzn waga połączenia = 1

Max średnica = 15, koszt 16 oznacza „nieskończoność”

Implementacje: routed, gated, **quagga** (nowsze!), gated i quagga zaw też inne prot!!

Problemy RIP...

1. nie wykrywa pętli
2. max średnica AS = 15
3. „powolna zbieżność” oraz powodowane przez nią kłopoty:



Rys. 16.4. Problem powolnej zbieżności. W przypadku (a) 3 routery znają trasę do sieci 1. W przypadku (b) znika połączenie z siecią 1, ale R_2 nadal je oferuje, powodując powstanie pętli

Routing wewn/ RIP

Jak unikać oscylacji między trasami o tym samym koszcie?

zasada: zmieniać trasę tylko jeśli pojawi się trasa o „<” koszcie...

Rozwiązań problemu „powolnej zbieżności”:

1. „uaktualnianie z podzielonym horyzontem” nie propaguje się informacji przez interf, przez który się tą informację otrzymało
2. „wstrzymanie” zmian; przez 60sek ignoruje się informacje o nieosiągalności danej sieci
3. „metoda odtrutki” po informacji o zniknięciu połączenia zachowuję się informacje o nim ale z wagą 16 (nieskończoność)

Format komunikatów RIP v1:

0	8	16	24	31
COMMAND (1-5)	VERSION (1)	MUST BE ZERO		
FAMILY OF NET 1		MUST BE ZERO		
	IP ADDRESS OF NET 1			
	MUST BE ZERO			
	MUST BE ZERO			
	DISTANCE TO NET 1			
FAMILY OF NET 2		MUST BE ZERO		
	IP ADDRESS OF NET 2			
	MUST BE ZERO			
	MUST BE ZERO			
	DISTANCE TO NET 2			
	...			

Command	Meaning
1	Request for partial or full routing information
2	Response containing network-distance pairs from sender's routing table
3	Turn on trace mode (obsolete)
4	Turn off trace mode (obsolete)
5	Reserved for Sun Microsystems internal use
9	Update Request (used with demand circuits)
10	Update Response (used with demand circuits)
11	Update Acknowledge (used with demand circuits)

Routing wewn/ RIP

Format komunikatu RIP v2:

0	8	16	24	31			
COMMAND (1-5)	VERSION (1)	MUST BE ZERO					
FAMILY OF NET 1		ROUTE TAG FOR NET 1					
IP ADDRESS OF NET 1							
SUBNET MASK FOR NET 1							
NEXT HOP FOR NET 1							
DISTANCE TO NET 1							
FAMILY OF NET 2		ROUTE TAG FOR NET 2					
IP ADDRESS OF NET 2							
SUBNET MASK FOR NET 2							
NEXT HOP FOR NET 2							
DISTANCE TO NET 2							
...							

Routing wewn/ OSPF

- działa nad IP, proto 89, RFC 2328, używany przez „dużych” ISP
- podział AS na obszary, w tym 1 obszar szkieletowy...
- w zasadzie LS (wewn obszarów), między obszarami DV...
LS: rozgłasza zmiany o stanie łączy wsz routerom swojego obszaru...
- wiele tras do danego celu/sieci wybieranych na podst TOS w nagł ip! oraz adr ip dst (?)
- „load balancing” gdy jest wiele tras do tego samego celu
- komunikacja między routerami wymaga uwierzyteln (inaczej niż RIP)
- trasy do konkretnych maszyn są możliwe
- obsługuje CIDR (adr ip bezklasowe)
- wykorzystuje broadcast, o ile możliwe

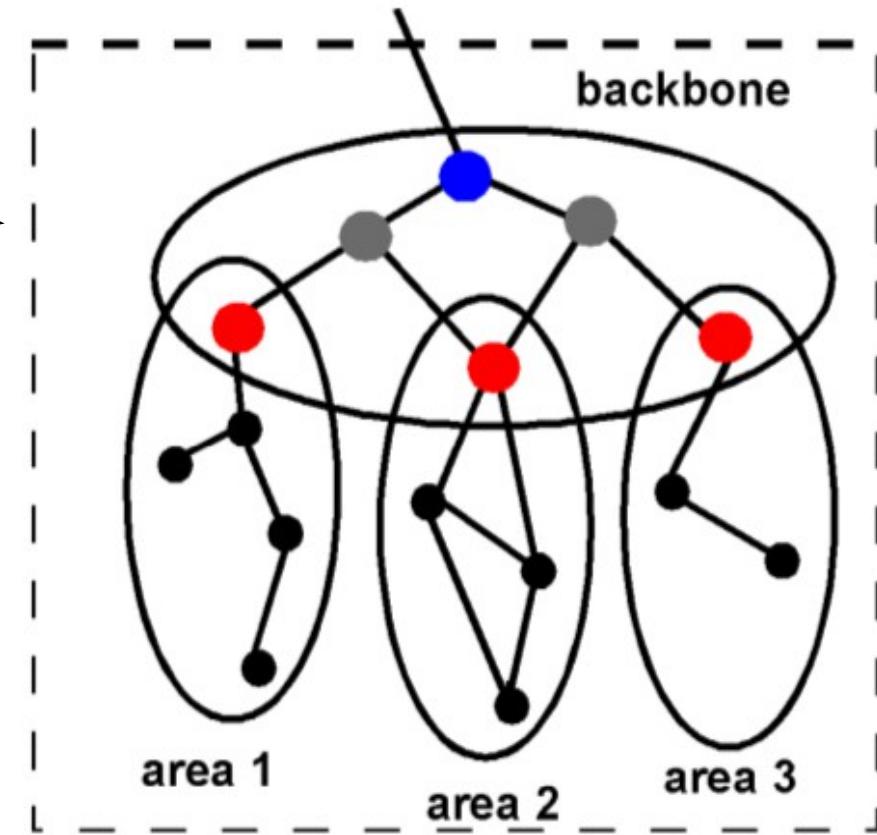


Figure 4.5-7: Hierarchically structured OSPF AS with four areas.

Niebieski – router graniczny do innego AS
Czerwony – router brzegowy obszaru (area)

Routing zewn/ EGP i BGP-4

Odcinki trasy między hostami h1 i h2 pochodzące od routingu wewn i zewn...

Połączenia „wirt” między routerami BGP

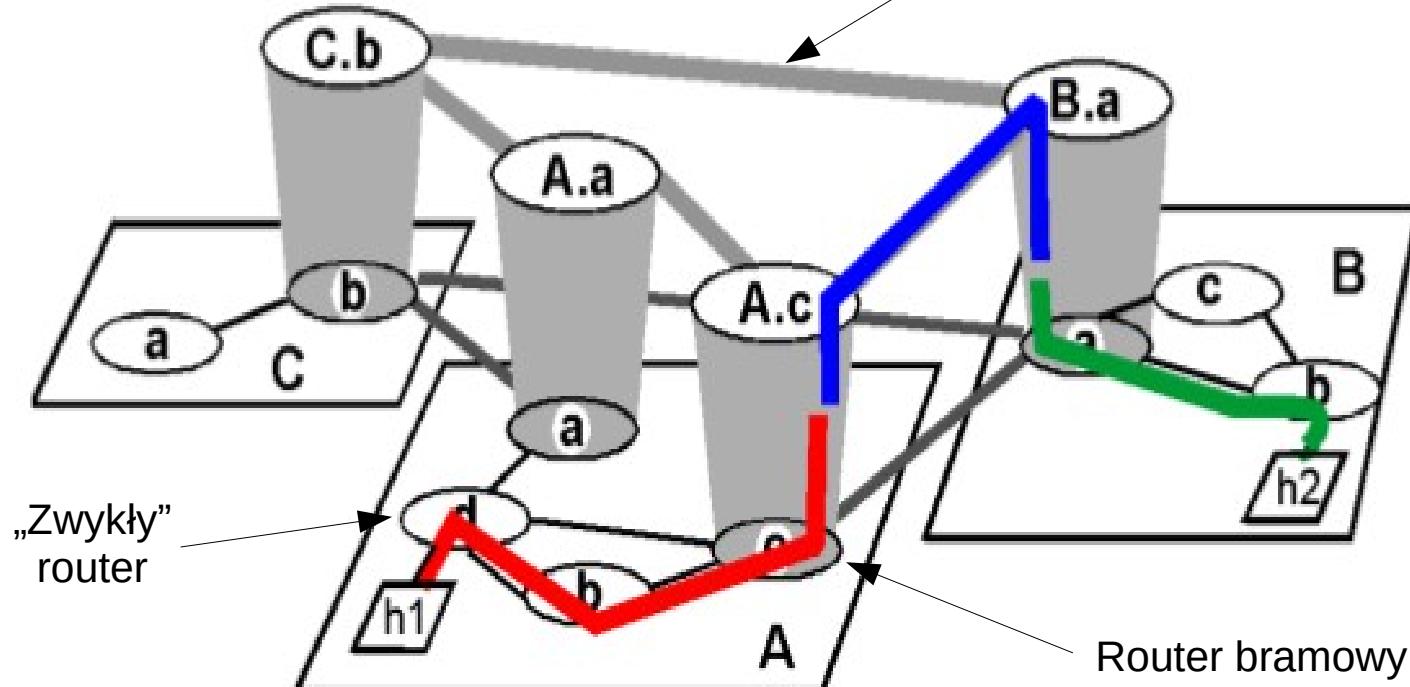


Figure 4.3-3: The route from A.d to B.b: intra-AS and inter-AS path segments.

Routing zewn/ EGP i BGP-4

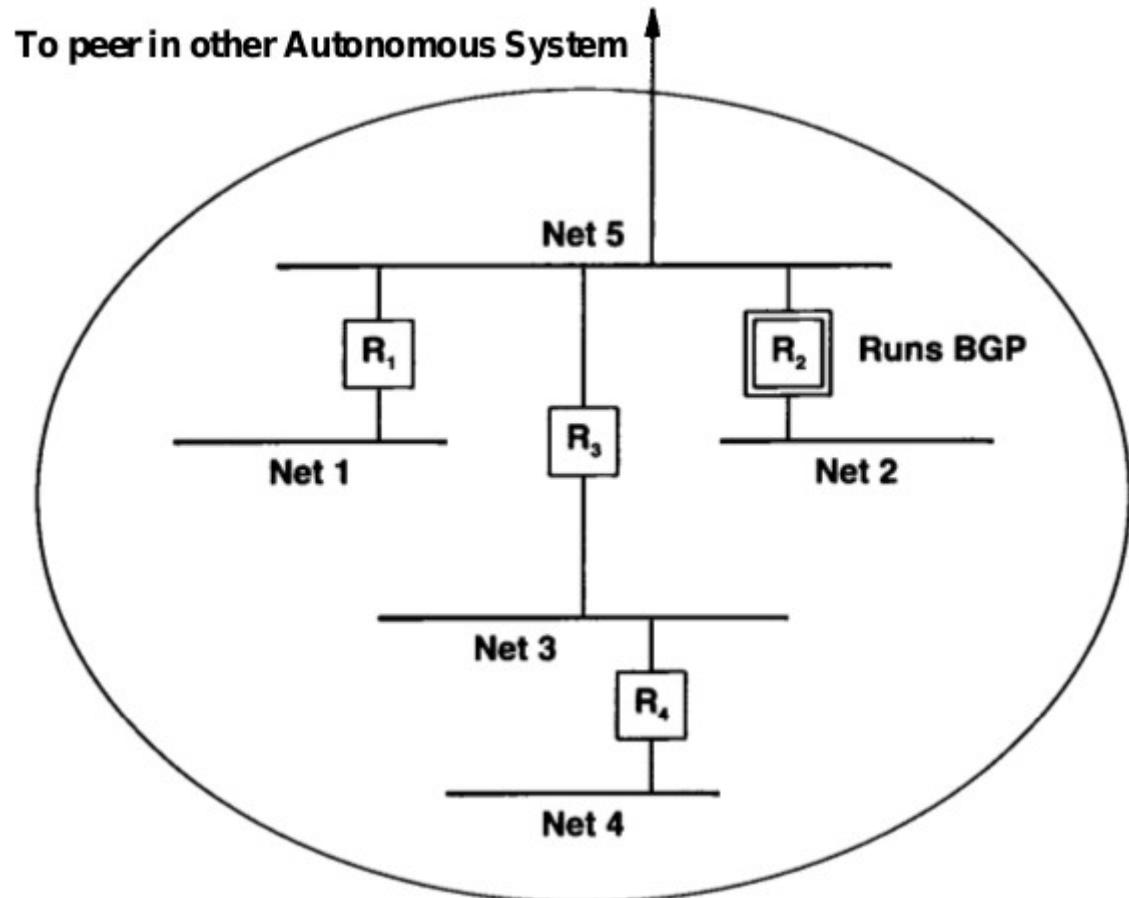
- „routing hierarchiczny”:
podział na AS vs krótki prefiks nr sieci

- „EGP” vs EGP (nazwa ogólna!)

- „EGP” przestarzały prot rout zewn;
router EGP informuje tylko
o sieciach w swoim AS !!!
„zakaz informowania o obcych”
wymagany jest „syst routerów podst”,
które znają wszystkie sieci...
brak info o odległości do obcych sieci
które są znane w AS...

- komunikaty „EGP” z aktualnymi trasami
zawierają odległości między podstawową
siecią (Net 5) a innymi sieciami AS-u...

- "BGP-4" likwiduje te ograniczenia...
informuje sąsiednie routery BGP
o sieciach **osiągalnych** przez własny AS...



Routing zewn/ BGP-4

- opisany w RFC 1771, 1772, 1773; działa nad prot TCP, port 179
- cele BGP:
 1. uzyskanie z sąsiednich AS informacji o osiągalności sieci
 2. przekazanie tych info wszystkim routerom własnego AS
 3. wyznaczanie dobrych tras do sieci, na podstawie info o osiągalności sieci oraz zasady zdef dla AS („polityka”)
- sesje BGP: iBGP (wewn 1 AS), eBGP (między różnymi AS-ami)
sesje eBGP służą do wymiany info o osiągalności sieci poprzez sąsiedni AS...
- BGP jest prot „wektor ścieżka” (vs „wektor odległość” w RIP)
dla docelowej sieci podaje się nie tylko odległość ale trasę/ścieżkę...
router BGP przekazują sobie trasy do sieci (nie tylko info o osiągalności)...
są to m.in. numery ASN przez które przebiega trasa...
jest więcej informacji o różnych trasach prowadzących do celu...
można lepiej podejmować decyzję...

Telekom, WAN, sieci ATM, ...

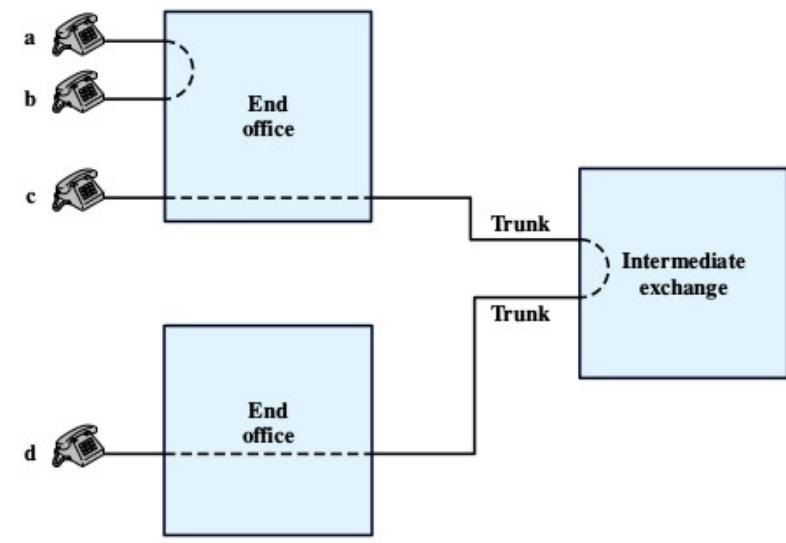
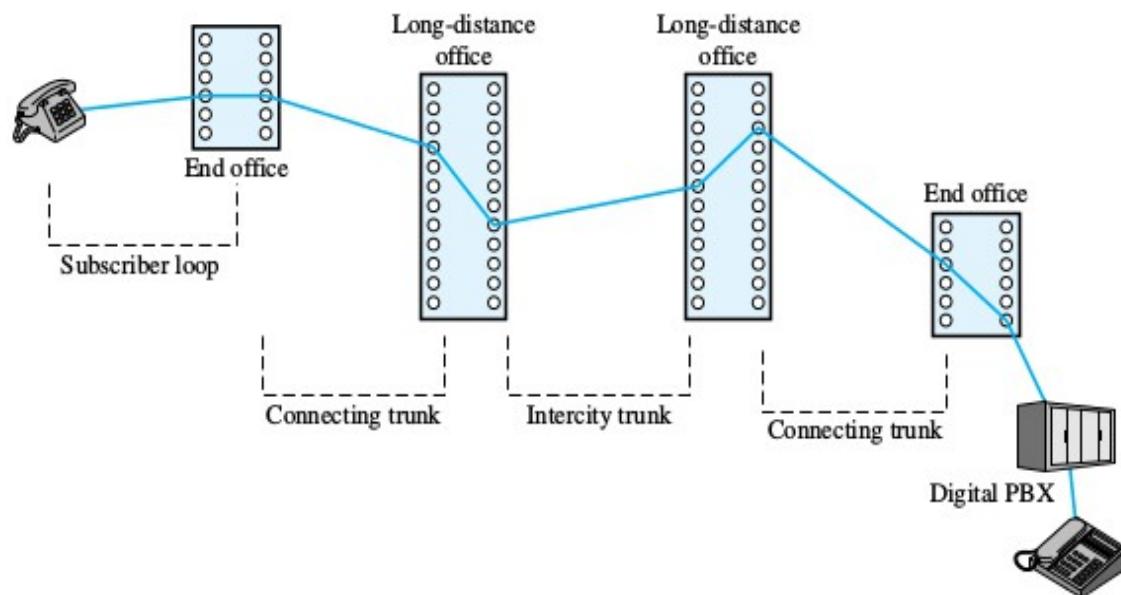
„Packet switching” vs „circuit switching” (przełączanie pakietów vs obwodów)
Przełączanie obwodów wywodzi się z sieci telefonicznych (POTS=Plain Old Teleph. Sys.)
QoS – połączenie („obwód”) daje gwarancje przepustowości i inne...

jest jednak rozrzucone w przypadku przesyłania danych...

Połączenia mogą być mniej/bardziej wirtualne (niekoniecznie chodzi o poł fizyczne/elektr.)
Są też połączenia w 4 warstwie („poł. TCP”) ?!?!?

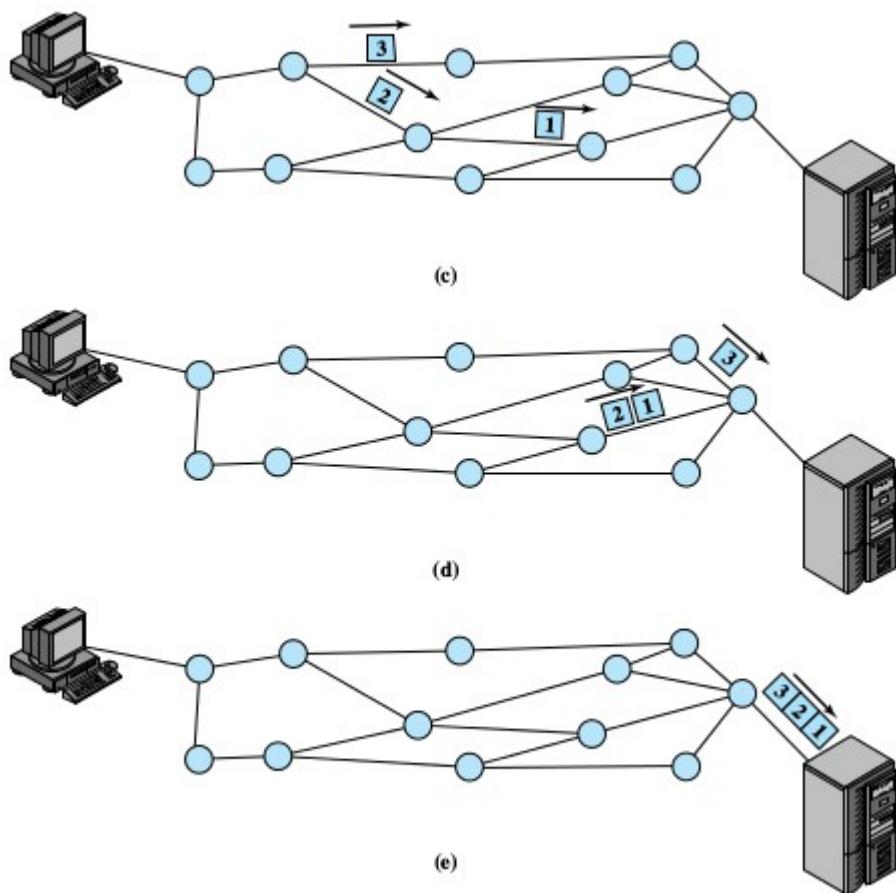
jednak na trasie **nie rezerwuje się** żadnych zasobów dla takiego połączenia
Internet, intersieć – przełączanie pakietów...

POTS - aboneci, pętle abonenckie, centrale telefoniczne, sygnalizacja (nawiązywanie poł),
między centralami przesyła się jednym kablem wiele rozmów, dawniej FDM, potem TDM,
telefonia „cyfrowa”: rozmowa < 4kHz, próbkowanie, PCM, 8kHz, 1 bajt/próbkę, stąd 64kb/s

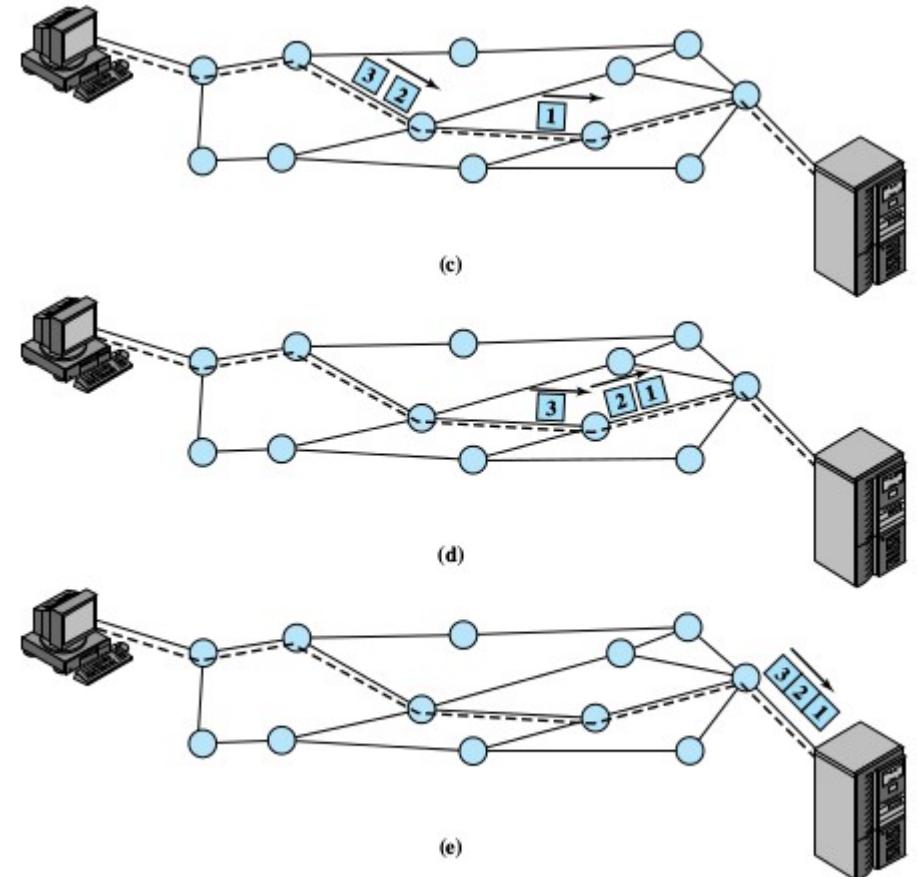


Dwie odmiany „przełączania pakietów”: 1. datagramowa, 2. wirtualnych obwodów

Pakiety są niezależne



Trasa zaplanowana zanim się wyśle pakiety



przełączania obwodów, pakietów/datagramowe, pakietów/wirtualnych obwodów

Table 10.1 Comparison of Communication Switching Techniques

Circuit Switching	Datagram Packet Switching	Virtual Circuit Packet Switching
Dedicated transmission path	No dedicated path	No dedicated path
Continuous transmission of data	Transmission of packets	Transmission of packets
Fast enough for interactive	Fast enough for interactive	Fast enough for interactive
Messages are not stored	Packets may be stored until delivered	Packets stored until delivered
The path is established for entire conversation	Route established for each packet	Route established for entire conversation
Call setup delay; negligible transmission delay	Packet transmission delay	Call setup delay; packet transmission delay
Busy signal if called party busy	Sender may be notified if packet not delivered	Sender notified of connection denial
Overload may block call setup; no delay for established calls	Overload increases packet delay	Overload may block call setup; increases packet delay
Electromechanical or computerized switching nodes	Small switching nodes	Small switching nodes
User responsible for message loss protection	Network may be responsible for individual packets	Network may be responsible for packet sequences
Usually no speed or code conversion	Speed and code conversion	Speed and code conversion
Fixed bandwidth	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each packet	Overhead bits in each packet

Przykłady sieci pakietowych z wirtualnymi obwodami: X.25, Frame Relay, ATM

ATM

ATM = Asynchronous Transfer Mode, wyraźne pochodzenie od sieci telefonicznych, atm switch (łącznica), interf UNI (user-switch) i NMI (switch-switch), para światłowodów, obwody: SVC komutowane (tworzone przez proto), PVC stałe (tworzone „ręcznie”), „ID obwodu” w UNI to 2 liczby: VPI 8b i VCI 16b (Virtl Path/Circ Id), w NNI VPI ma 12b, **zamiast adr dst w komórce jest „id obwodu” ?!?! obowdy tworzy się na podst adr dst Komórki (nie ramki!) ATM: 53 bajty = 5 nagłówków, 48 dane, stała długość!** Warstwa adaptacyjna: AAL1: CBR, np. audio; AAL2: VBR; AAL5: do 64Kb, dane

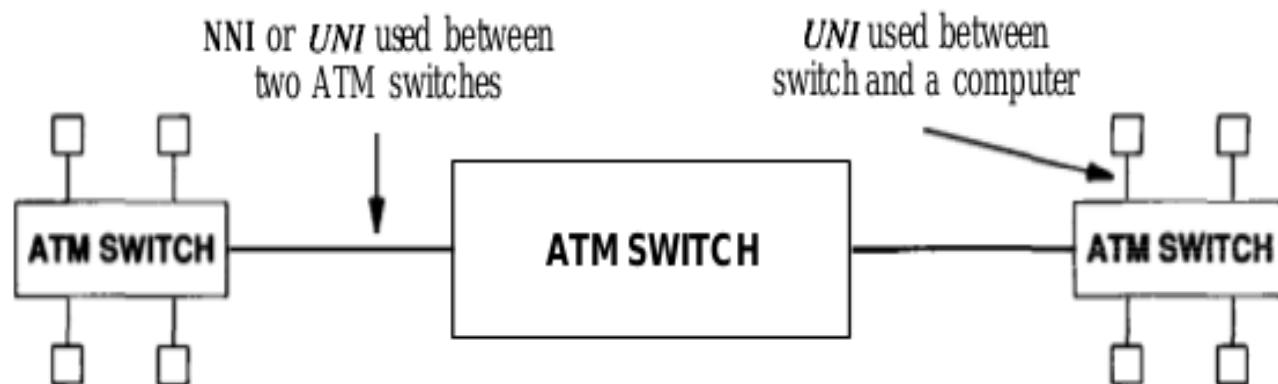


Figure 18.2 Three ATM switches combined to form a large network.
Although an NNI interface is designed for use between switches,
UNI connections can be used between ATM switches in a
private network.

Warstwy i „plaszczynny” ATM:
plaszczyna? dotyczy wielu warstw

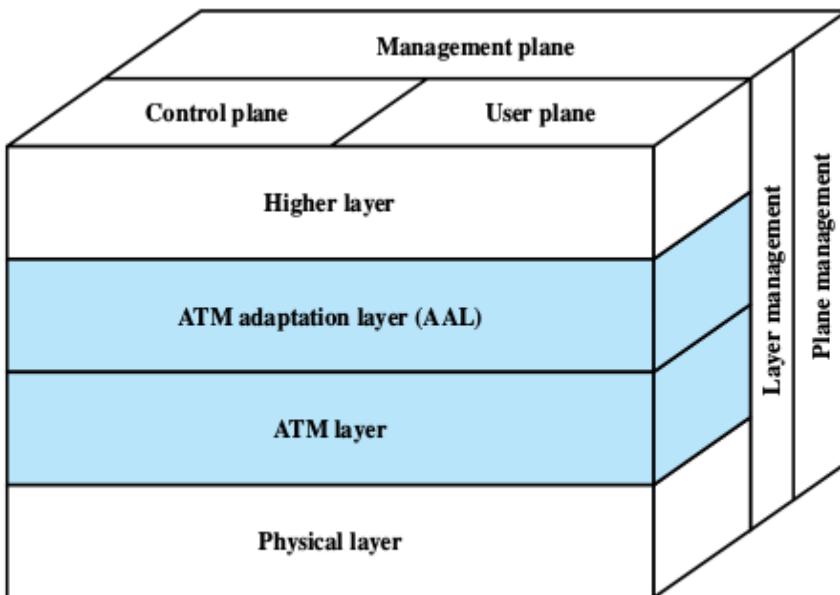


Figure 11.1 ATM Protocol Architecture

Miejsce AAL (warst adaptacji)
jest na końcach obwodu...
gwarancje co do połączenia
AAL1: CBR, audio
AAL2: VBR, ?
...
AAL5: dane, „best effort”

Komórki ATM, UNI i NNI:
- komórki nie posiadają adr dst,
a jedynie id obwodu, który się zmienia!!

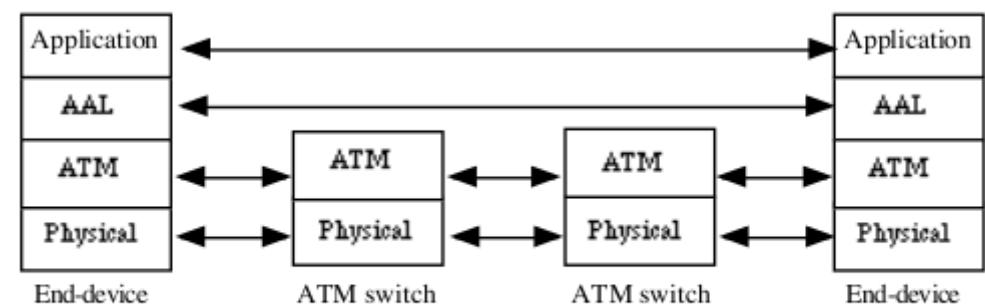
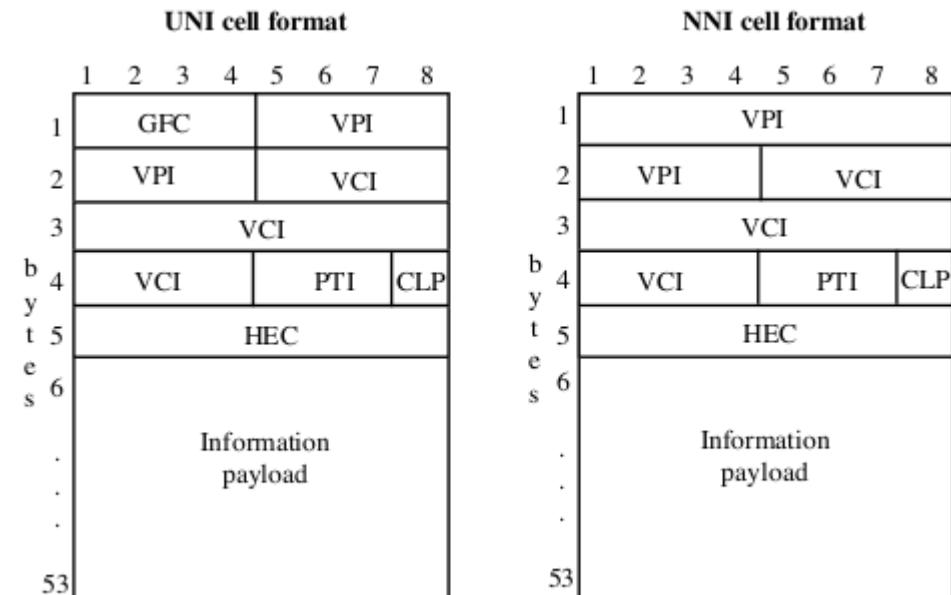


Figure 4.6: Cell switching in an ATM network

Rola „id obwodu ATM” VPI/VCI :

mamy obwody: A-> B i C-> D

Jak widać VPI/VCI zmienia się na odcinkach obwodu...

Tablice „label switching” decydują co dalej z pakietem oraz zmieniają VPI/VCI...

wpis y w tabl pojawiają się przy tworzeniu połączenia...

VP (Virt Path) vs VC (Virt Chann) ?

związek: VP zawiera wiele VC
chodzi o ułatwienie zarządzania,
zarządza się całą grupą VC,
a nie pojedynczymi VC...

to zmniejsza czas tworzenia
poł logicznego między końcami

Brak adresów sprzętowych ATM?

jak się definiuje końce poł log??
jest kilka rozw, adr różnej dług,
ATMARP...

Podobna do ATM koncepcja: **MPLS**
także rezerwuje się połączenie,
etykieta MPLS między nagł eth/ip
pełni rolę VPI/VCI...

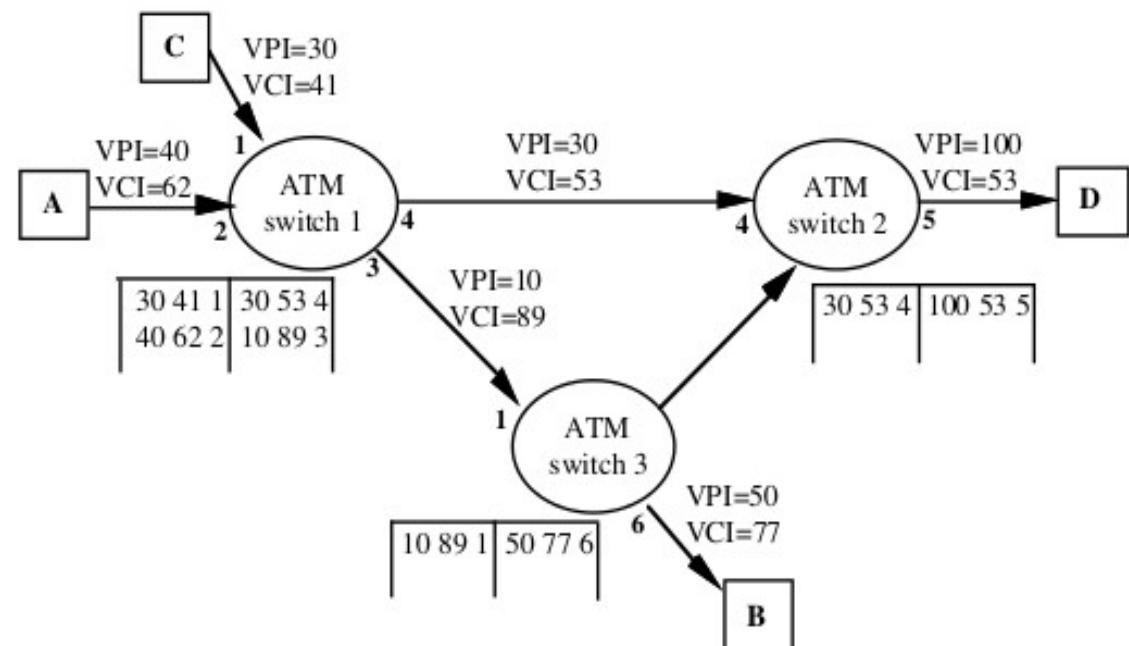


Figure 4.3: An example of label swapping

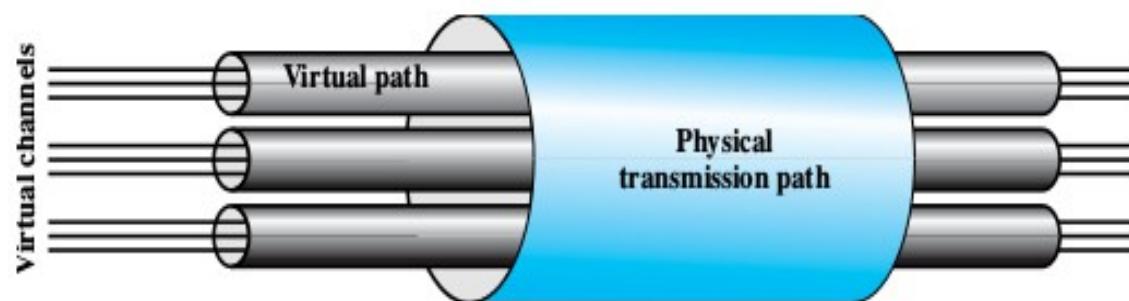


Figure 11.2 ATM Connection Relationships

SONET/SDH

Multipleksowanie wielu wolniejszych kanałów na światłowodzie...

np. rozmów telefon 64kb/s, ale także kanałów dla danych, także „ATM nad SDH”
SONET – Synch Optical Net (USA), SDH = Synch Digital Hierarchy (reszta świata)

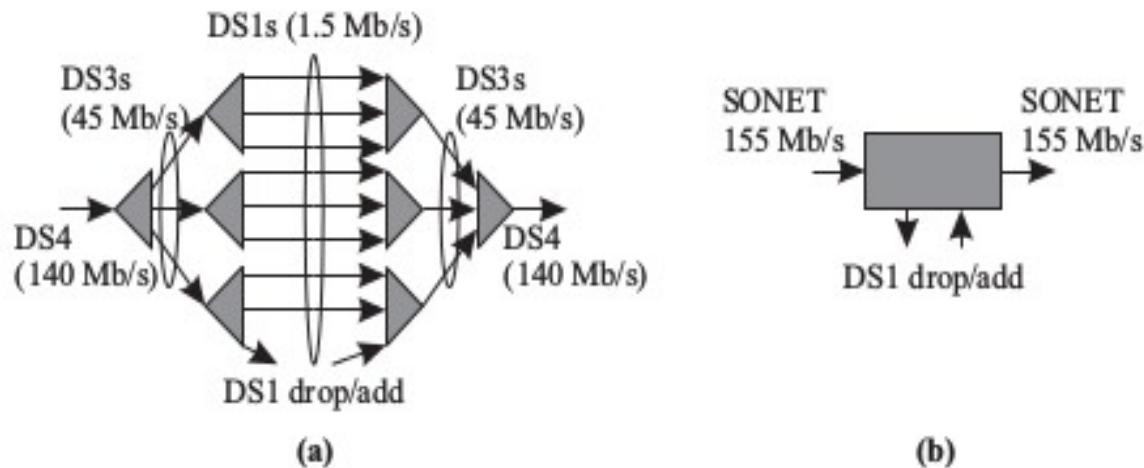
„hierarchia” ??? chodzi o składanie z kilku wolniejszych kanałów 1 szybszego
SDH działa w trybie **synch** (vs asnych), np. rs232 jest asynch (wykrywanie ramki)

SDH: dokładne zegary, wiadomo kiedy przyjdzie pierwszy bit ramki...

ramki SDH są specyficzne, z przeplotem

Poprzednik historyczny SDH: PDH – słabsza synchronizacja kanałów/ramek

zaleta SDH nad PDH: można wyciągnąć wolniejszy kanał w tańszy sposób...



Ramki SDH...

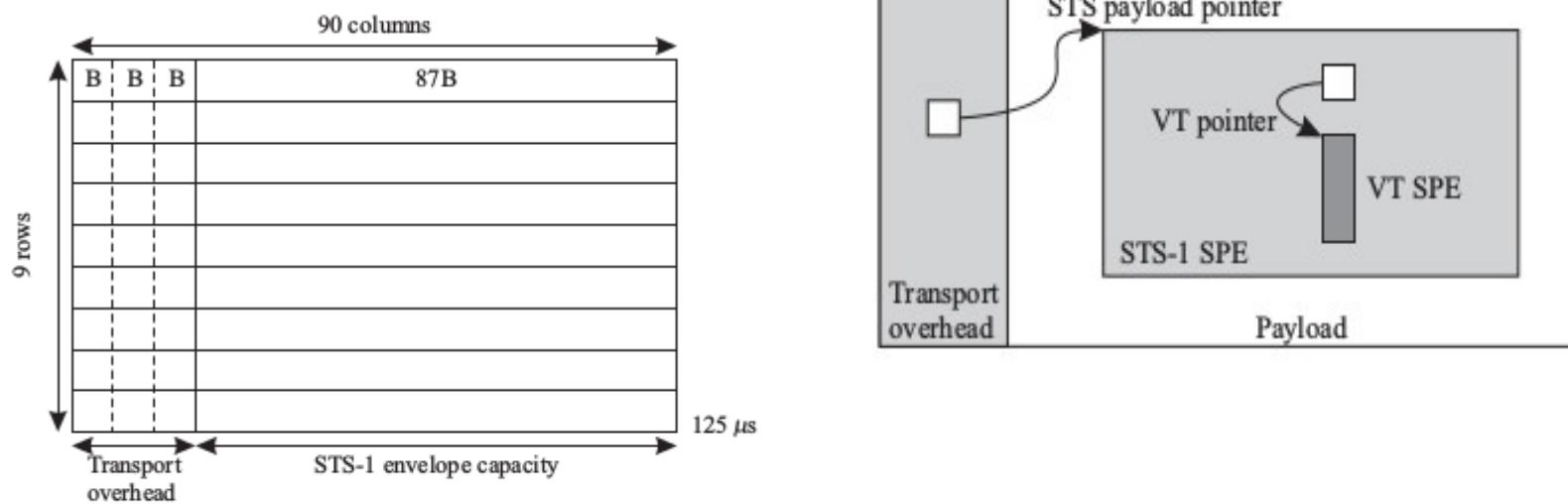


Figure 6.5 Structure of an STS-1 frame. B denotes an 8-bit byte.

Budowa sieci SDH i wydobywanie wolniejszych kanałów: ADM Add/Drop Multiplexer

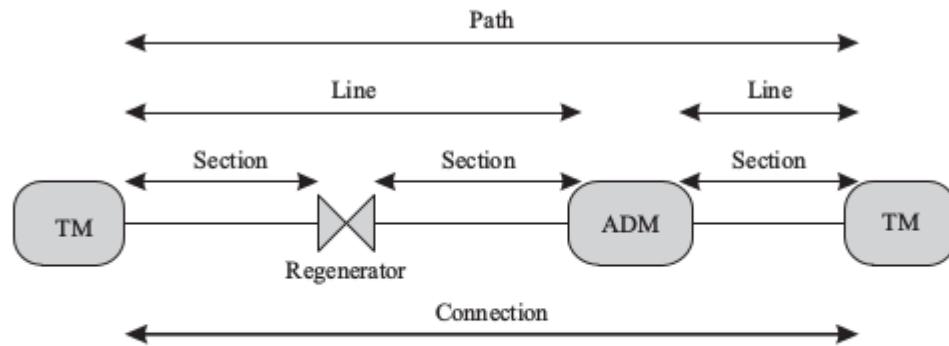


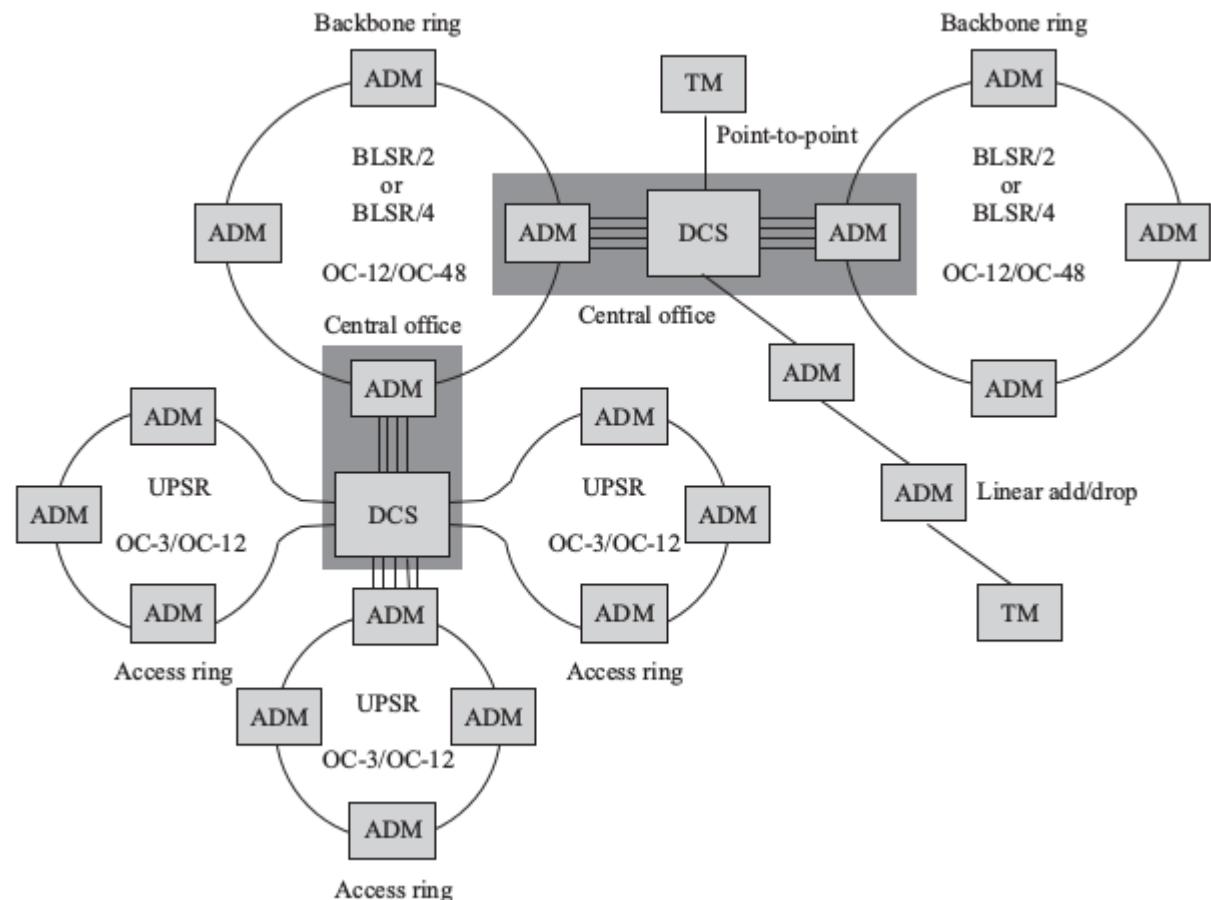
Figure 6.4 SONET/SDH layers showing terminations of the path, line, and section layers for a sample connection passing through terminal multiplexers (TMs) and add/drop multiplexers (ADMs). The physical layer is not shown.

Przepływność kanałów SDH:

SONET name	SDH name	Line rate (Mbps)	Synchronous Payload Envelope rate (Mbps)	Transport Overhead rate ⁷ (Mbps)
STS-1	None	51.84	50.112	1.728
STS-3	STM-1	155.52	150.336	5.184
STS-12	STM-4	622.08	601.344	20.736
STS-48	STM-16	2,488.32	2,405.376	84.672
STS-192	STM-64	9,953.28	9,621.504	331.776
STS-768	STM-256	39,813.12	38,486.016	1,327.104

Table 1· SONET/SDH digital hierarchy

Globalne spojrzenie na sieci SDH:
ringi, poł p2p, ADMy



PPP

PPP prot przenoszący pakiety IP lub inne, np. IPX, DECnet (czyli warstwa 2) ...

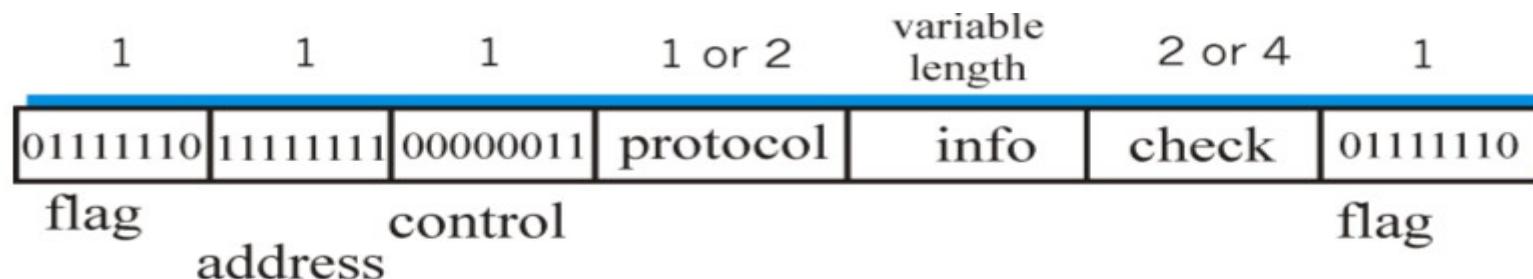
Działa nad „łączem szeregowym” : poł tel+modemy 56kbs, poł SDH, poł X.25, poł ISDN
czyli obsługuje połączenia pkt-do-pkt...

Opisane w RFC 1661, RFC 1331 (stare), RFC 1334 (auth)

Jakie wymagania postawiono przed PPP ?

- def ramki, odbiorca musi wiedzieć gdzie jest początek i koniec rami
- przezroczystość, dowolne dane (żadnych ograniczeń, np. że \n zakazany)
- wiele prot warstwy 3 (nie tylko IP)
- różne rodzaje połączenia fizycznego (także wirt kanały, szybkie i wolne)
- error detection (ale nie correction)
- connection liveness (sprawdzanie czy poł fiz ciągle działa)
- negocjacje prot warstwy 3 (np. ustalanie adr IP obu końców)
- uwierzytelnianie obu końców połączenia (dlaczego tego nie podano u Kurose?)
- NIE wymaga się „flow control” (robią to prot wyższych warstw)
- NIE wymaga się „error correction”

Format ramki PPP:



Flag oznacza początek ramki, ten znak musi być „escaped” (01111101) jeśli jest w info !!
„info” to dane ramki, check to CRC, „protocol” to typ danych ramki... np. IP, IPX, LCP, NCP

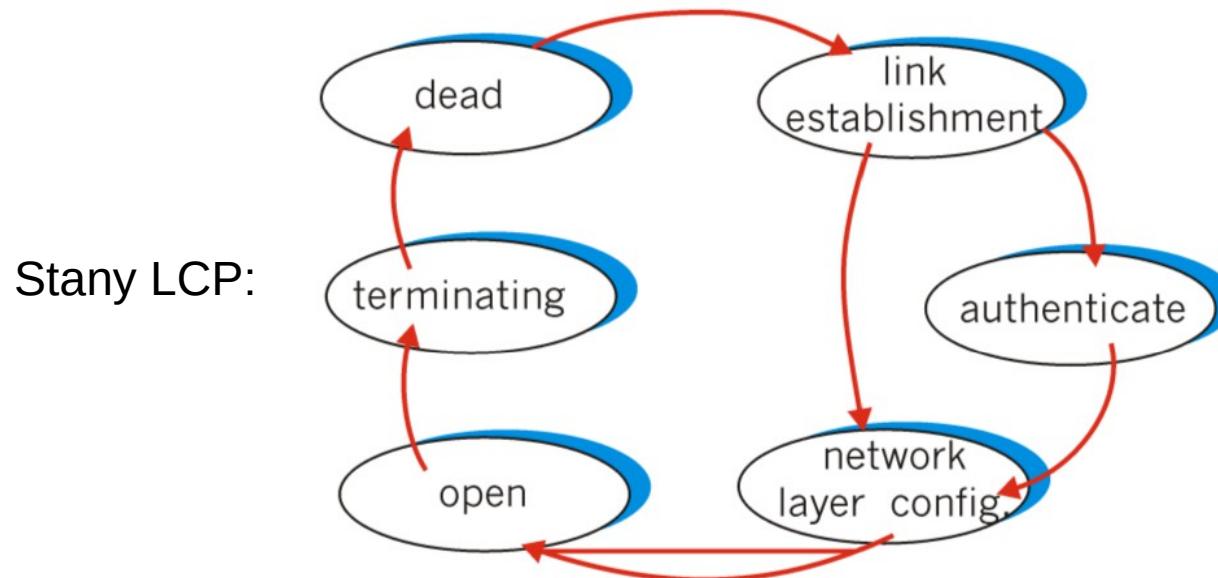
Co to są komunikaty prot LCP i NCP ???

LCP = Link Control Protocol,

jest odp za: konfig połączenia „handshake”, np. ustala max dług danych/info, negocjuje metodę uwierzytelnienia użytkownika oraz ją wykonuje, jest kilka sposobów auth: PAP, CHAP, EAP (znane też z wifi/WPA), po ustanowieniu połączenia i uwierzytelnieniu końców, czas na konfg warstwy 3...

NCP = Network Control Protocol, „N”= IP, DECnet, IPX, ...

np. w przypadku IP: ustalenie adr ip iterf sieciowych, default gw, ser DNS, ...



Polecenia „pppd” oraz moduł „ppp” impl prot PPP w linuxie...

przykład *Tcl*-owy: zamiast /dev/tty??? używa stdin/out (notty), s – gniazdko tcp
exec pppd notty noauth passive local 192.168.10.1:192.168.10.2 <@\$s>@\$s &
exec pppd notty <@\$s>@\$s &

Multimedia

Cechy charakterystyczne strumienia audio/video (a/v):

- wrażliwość na opóźnienie pakietu
- tolerancja strat pakietu (większa niż w innych usługach)

Typy strumienia a/v (inne wymagania):

- transmisja zapisanego a/v
- transmisja „na żywo”, interaktywna, np. telefon internetowy voip, telekonf itp

Prot Tcp/Ip: czego używać do a/v ??

- UDP lepsze niż TCP (dlaczego?)
- standardowy sposób przesyłania a/v przez UDP: prot RTP/RTCP
- problemy z „best effort” (brak QoS w Internecie !!!)
- transmisja „1 do wielu”, podobna do rtv: multicasting, grupy multicastowe

Sposoby radzenia sobie z brakiem QoS w Internecie:

- best effort (to co jest teraz, czyli brak QoS...)
- Intserv (radykalna propozycja, zmiana prot, na podobieństwo ATM / MPLS ?)
- Diffserv (mniej radykalna prop, klasy ruchu: ekonomiczna/biznesowa, TOS)

Telefonia internetowa: Voip

- H.323, SIP+RTP, oprogram „asterisk” (centrala tel ip)
- połączenie z POTS

Ogólne sposoby radzenia sobie z a/v bez QoS

Kompresja danych a/v (mniejsze wymagania co do przepustowości)

- audio: zapis PCM (bez kompresji, 64kbs), GSM (13kbs), MP3

- video: kompresja wideo MPEG1 (VCD), MPEG2 (DVD),

MPEG4 chap 2 = Xvid

MPEG4 chap 10 = H.264

MPEG4 chap 14 = spec kontenera multimed .mp4 (wiele strumieni a+v)

Problemy wynikające z „best effort”:

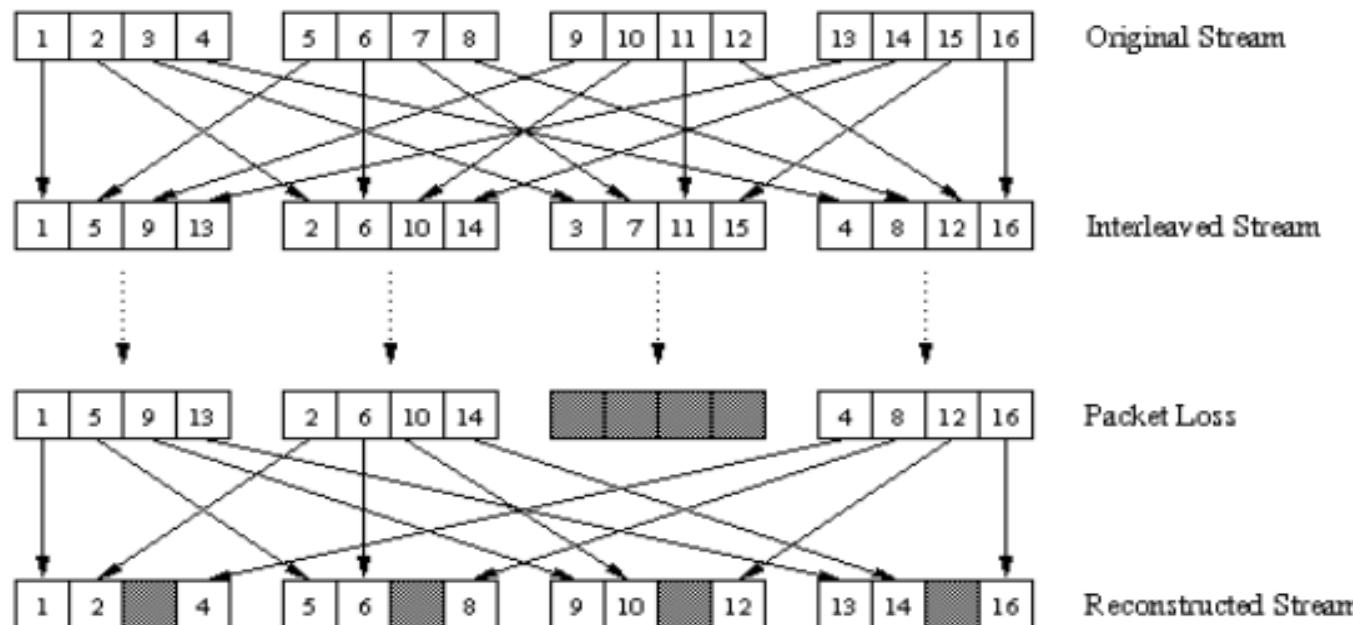
- opóźnienie pakietów UDP, i co za tym idzie...

- fluktuacje (ang. jitter), pakiety przychodzą w niewłaściwej kolejności

- straty pakietów

Jak sobie z tym radzić ???

- zapisany a/v: buforowanie, redundancja strumienia, interleaving+interpolacja:



Transmisja zapisanego a/v

Film ściągany przez http (czyli nad tcp)

- długa odpowiedź, content-type=video/mpeg
- przeglądarka może uruchomić zewn media player lub html5/video zamiast url-a do pliku a/v może być url do metapliku...
- prot http pozwala odtwarzać film od pewnego miejsca (?!?!), spec nagłówek w żądaniu http...

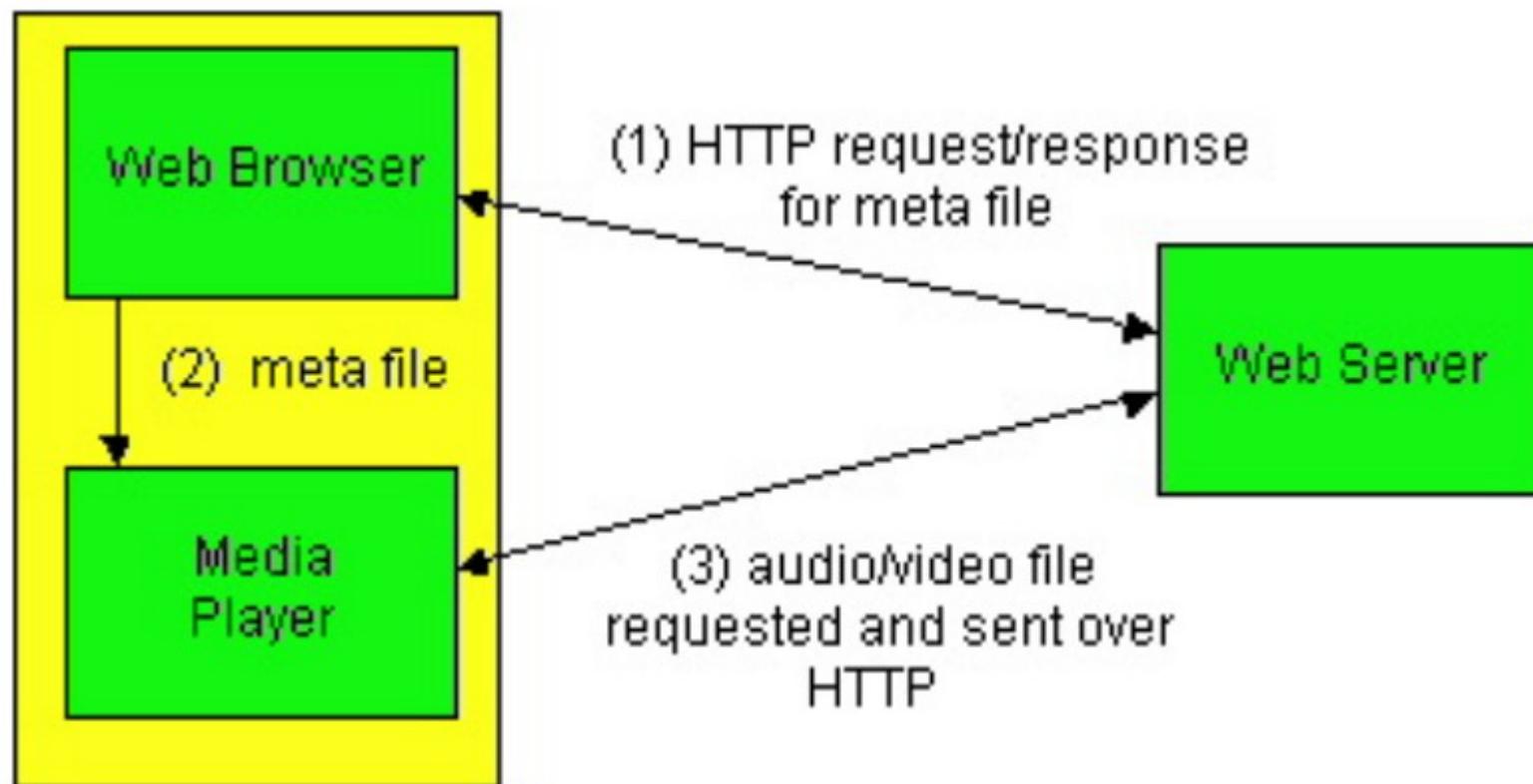
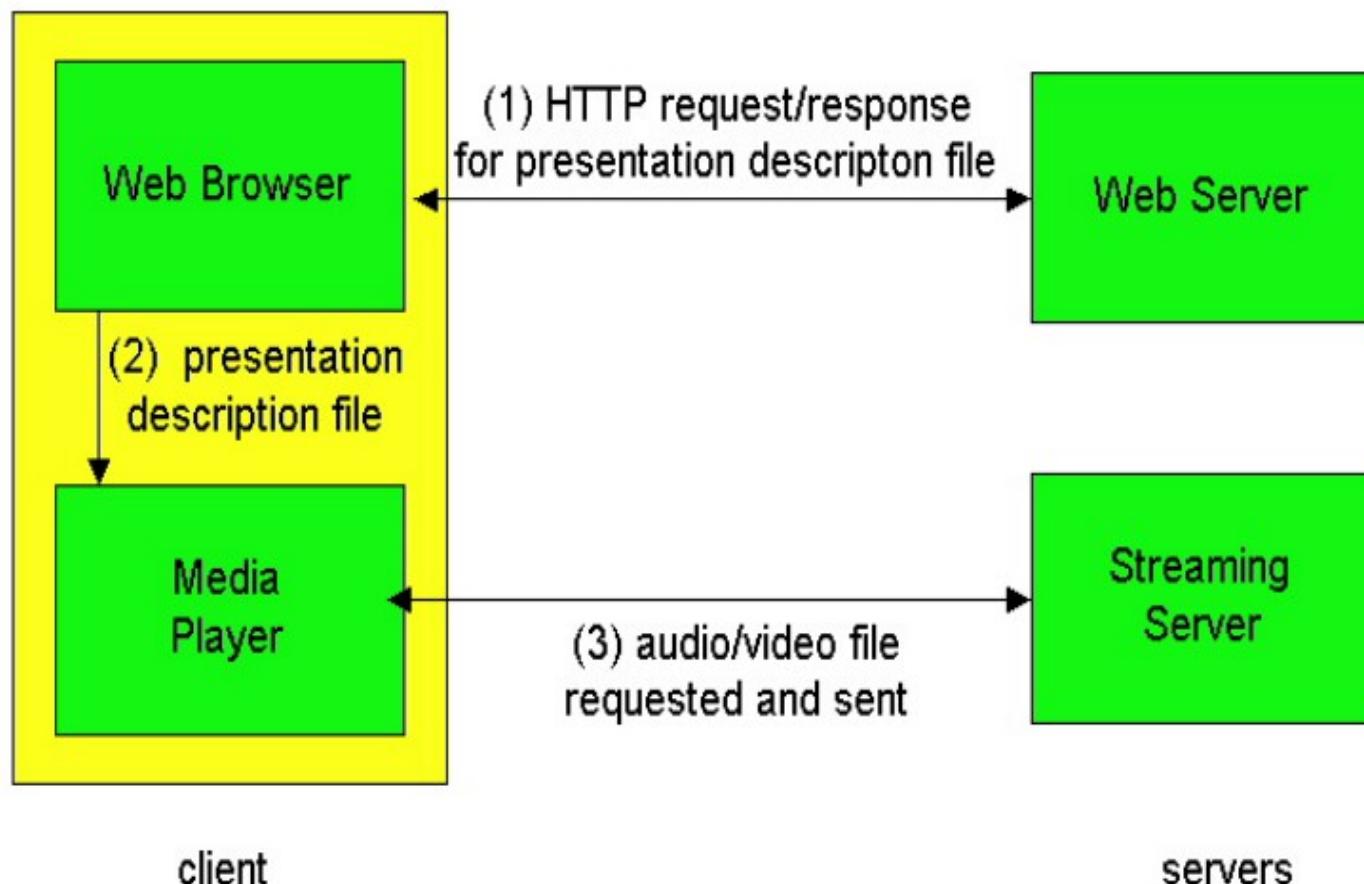


Figure 6.2-2 Web server sends audio/video directly to the media player.

Transmisja zapisanego a/v

Media player odtwarza film wysyłany przez streaming server,

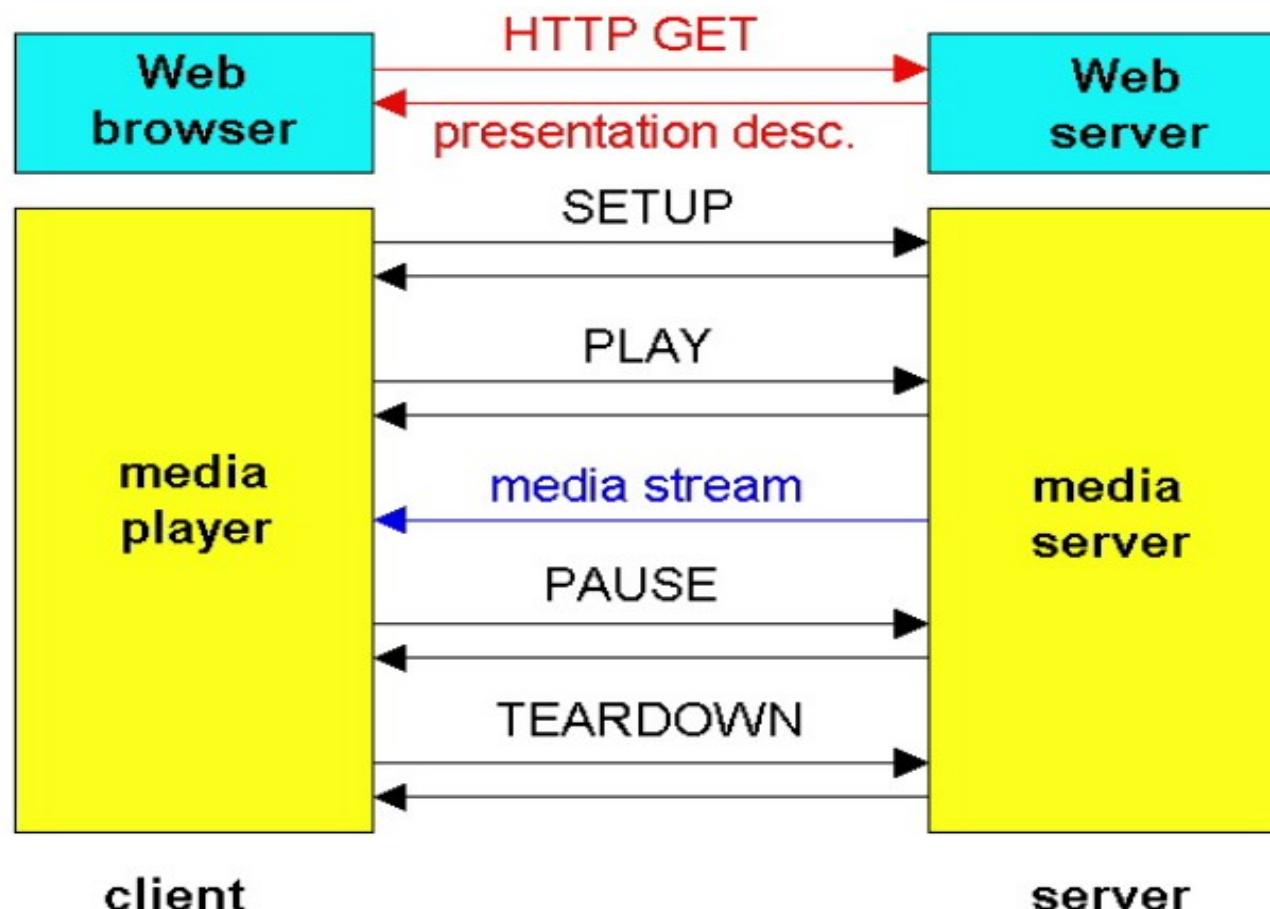
- strumien a/v przesyłany za pomocą prot UDP i RTP/RTCP
- buforowanie u klienta, przez co unika się fluktuacji
- plik SMIL opisujacy jak „media player” łączy się z „stream ser”



Transmisja zapisanego a/v

Jak poprzednio, jest serwer strumienowy, ale

- mamy obsługę prot RTSP (RFC 2326),
RTSP: podobny do http, odpowiednik „pilota” odtwarzacza dvd
- media stream przesyłany być może przez RTP...
- gdzie można spotkać ? Youtube + stary telefon (np. Nokia 500)



RTP

RTP (port p) i RTCP (port p+1), RFC 3550, generalnie: transmisja a/v...

RTP przesyła klatki filmu albo dane audio

RTCP przesyła „statystyki” (ich użycie zależy od app)

Działa nad prot UDP (czyli możliwe straty/ opóźnienia/ fluktuacje pakietów)

RTP może działać w trybie multicastingu („1 do wielu”) lub unicastingu

Używane przez app multimedialne, np. voip (SIP/RTP, H.323), vic/vat, ...

NIE zapewnia QoS, ale pozwala naprawić fluktuacje itp

Co jest w nagłówku RTP?

nr sekwencyjny, znacznik czasowy, id sesji, typ zawartości,

nr sekw pozwala wykryć straty pakietów!

Payload Type	Sequence Number	Timestamp	Synchronization Source Identifier	Miscellaneous Fields
--------------	-----------------	-----------	-----------------------------------	----------------------

RTP Header

Payload Type Number	Audio Format	Sampling Rate	Throughput
0	PCM mu-law	8 KHz	64 Kbps
1	1016	8 KHz	4.8 Kbps
3	GSM	8 KHz	13 Kbps
7	LPC	8 KHz	2.4 Kbps
9	G.722	8 KHz	48-64 Kbps
14	MPEG Audio	90 KHz	-
15	G.728	8 KHz	16 Kbps

Figure 6.4-4 Some audio payload types supported by RTP.

Payload Type Number	Video Format
26	Motion JPEG
31	H.261
32	MPEG1 video
33	MPEG2 video

Figure 6.4-5 Some video payload types supported by RTP.

Multicasting

Wysyłanie 1 pakietu do wielu odbiorców (przydatne w transmisji a/v typu rtv)

Działa z prot UDP, specjalne adr ip dst klasy D; 224.0.0.0-239.255.255.255
adr ip src są zwykłe...

Grupy multicastowe (mc), każda grupa ma swój adr klasy D

Komputer może należeć do wielu grup mc !!

Ta sama grupa może być w wielu różnych sieciach fizycznych !!!

Gdy wysyłamy pakiet do grupy mc, to może on trafić do maszyn
w różnych sieciach fizycznych (inaczej niż w broadcastingu...)

Obsługa programistyczna mc (przykład udp01b.tcl na stronie ćw Temat H)

Wysyła się pakiety na adres mc i nr portu,

Procesy mogą się podłączać do grupy mc (oraz odłączać)

Specjalne grupy mc:

224.0.0.1 grupa wszystkich węzłów

Zamiana „adr mc ip” na „adr mc eth” (w przypadku zwykłego adr ip: usługa arp)

Istnieje mc sprzętowy w eth !!! adr eth postaci 01:XX:XX:XX:XX:XX

23 najmłodsze bity adr mc ip umieszcza się w najmłodszych bitach

spec adr mc eth: 01:00:5E:XX:XX:XX

(choć adr mc ip ma 28 znaczących bitów !!!)

Multicasting c.d.

Pakiety mc mogą przeskakiwać przez routery... (rola pola TTL w nagł ip)

Prot IGMP = Internet Group Management Protocol...

komunikacja host – router w jednej sieci fizycznej

router musi widzieć jakie grupy mc są niepuste w danej sieci fizycznej

jeśli router jest połączony do wielu sieci fizycznych

to utrzymuje takie info dla każdej z nich....

Prot IGMP składa się z prostych komunikatów

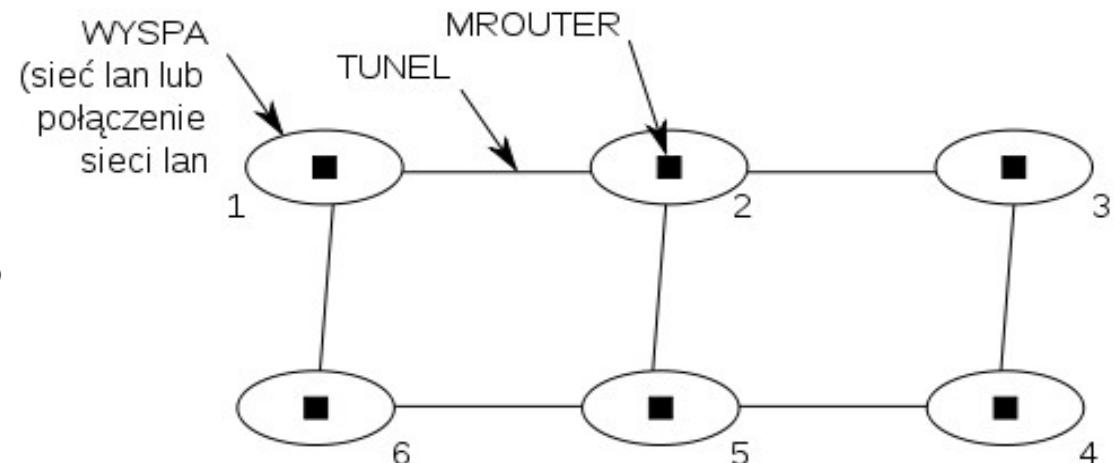
Query ($r \rightarrow h$); Report/ Join Group, Leave Group ($h \rightarrow r$)

Prot IGMP działa nad IP (służy także do komunikacji między routerami!)

Istnieje „routing mc”, info o trasach obsługiwane przez prot DVMRP (RFC 1075)

Istnieją „wyspy mc” każda będąca intersekcją z obsługą mc,

połączone tunelami przechodzącyymi przez sieci bez obsługi mc (mbones)



Mrouted = impl prot DVMRP

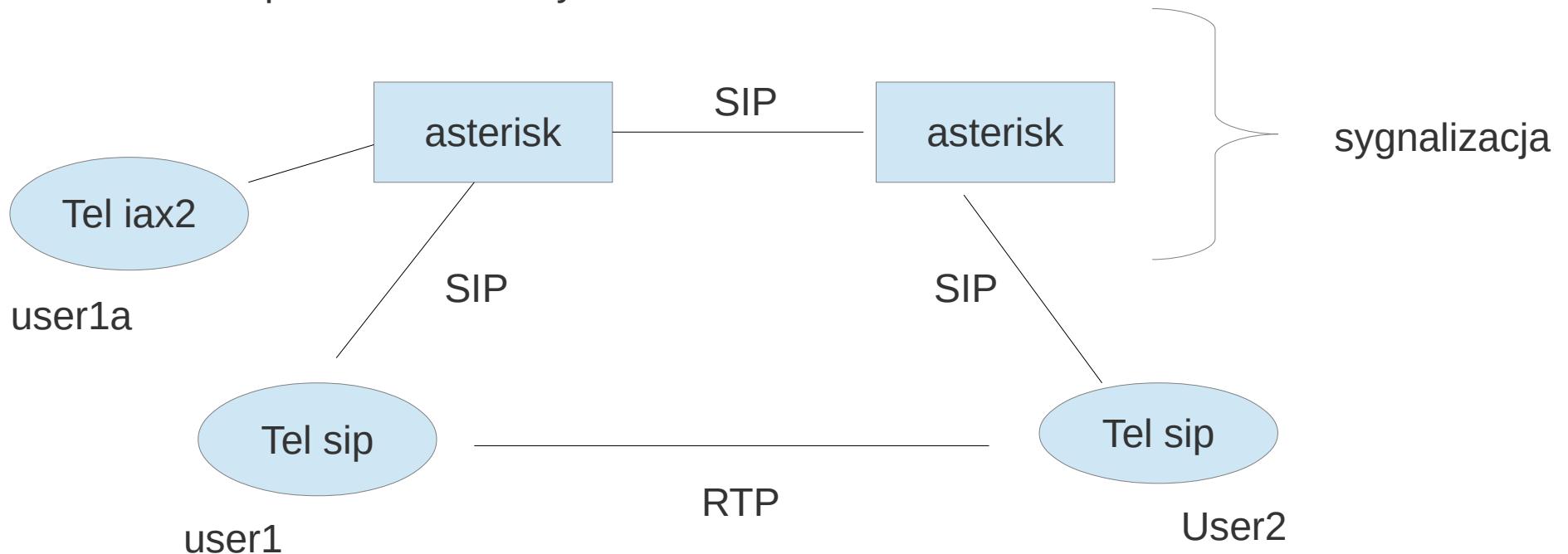
VOIP/ asterisk...

Program „asterisk” = internetowa „centralka telefoniczna”

Może być połączony z POTS oraz z analogowymi telefonami
przy pomocy spec kart rozszerzeń...

Obsługuje prot SIP/RTP oraz IAX2 (pojedynczy port w przeciwieństwie do RTP)
prot sip sygnalizacja, prot rtp transmisja audio
prot iax2 obsługuje sygnalizację + transmisję audio

Istnieją programy klienckie linphone (SIP/RTP), ??? (IAX2), także na androida,
są też biblioteki (do budowy w telefonów soft)
oraz sprzętowe telefony SIP/IAX2....



Internet vs QoS (propozycje)

Intserv

idea podobna do QoS w ATM, tworzy się połączenia, rezerwuje zasoby,
opis czego app oczekuje: Rspec, Tspec
do rezerowania połączenia używa się prot RSVP (RFC 2205)

Diffserv

różne klasy ruchu, biznesowa/ekonomiczna,
pakiety biznesowe mają wyższy priorytet niż ekonomiczne,
nie wymaga wielkich zmian w dzisiejszym internecie...

DHCP

czyli automatyczna konfig interf sieciowego

Skąd komputer bierze adr IP, maskę, oraz inne elem konfiguracji interf sieciowego?

Historia: RARP (tylko adr IP), BOOTP, DHCP (rozszerzenie BOOTP, obecnie używane...)

RARP – podobne do ARP, działa w drugą stronę, pyt / odp, broadcast sprzętowy,
losowe opóźnienie przy wysyłaniu odpowiedzi przez ser rarp (istotne w eth...)

BOOTP – jedna wymiana komunikatów z serwerem BOOTP

taki sam komunikat w pyt i odp, w odp wypełnia się nieznane pola,
możliwość użycia przekaźnika, nazwa pliku startowego (img dysku), komp bezdyskowe

DHCP – rozszerzenie BOOTP, trochę bardziej skomplikowane...

trzy sposoby przydzielania adr:

1. adr eth → adr ip, przydzielany „ręcznie”
2. adr ip przydzielany dynamicznie z puli, gdy pojawi się pyt
3. adr ip przydzielany dynamicznie/ na określony czas („leasing”)
po upłynięciu okresu wynajęcia musi być odnawiany!!
tylko ta metoda pozwala na odzyskanie adr ip !!!

BOOTP i DHCP działają nad UDP (porty 67 i 68)

jak to możliwe gdy nie ma adr IP? odp; broadcast, adr dst ip „same 1”

Linux:

klient dhcp : program **dhcpcd**; serwer dhcp: program **dhcpd** lub **dhcpd3**
po stronie kli: „dhcpcd eth0”

DHCP

Jakie informacje dostarcza dhcp?

Adr ip, maskę, default gw, adr ser DNS, wart MTU, ...

Komunikaty BOOTP i DHCP...

w DHCP jest kilka typów komunikatów (w opcjach, z uwagi na kompat z BOOTP)
kli może otrzymać kilka odpowiedzi, od kilku ser, wybiera z nich jedną...

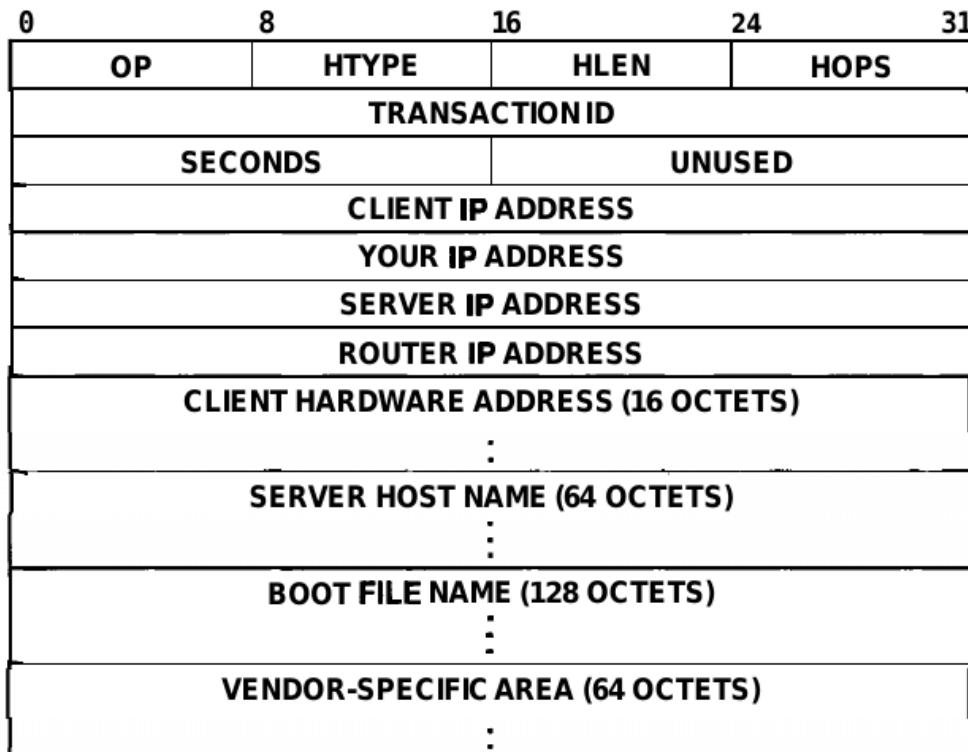


Figure 23.1 The format of a BOOTP message. To keep implementations small enough to fit in ROM, all fields have fixed length.

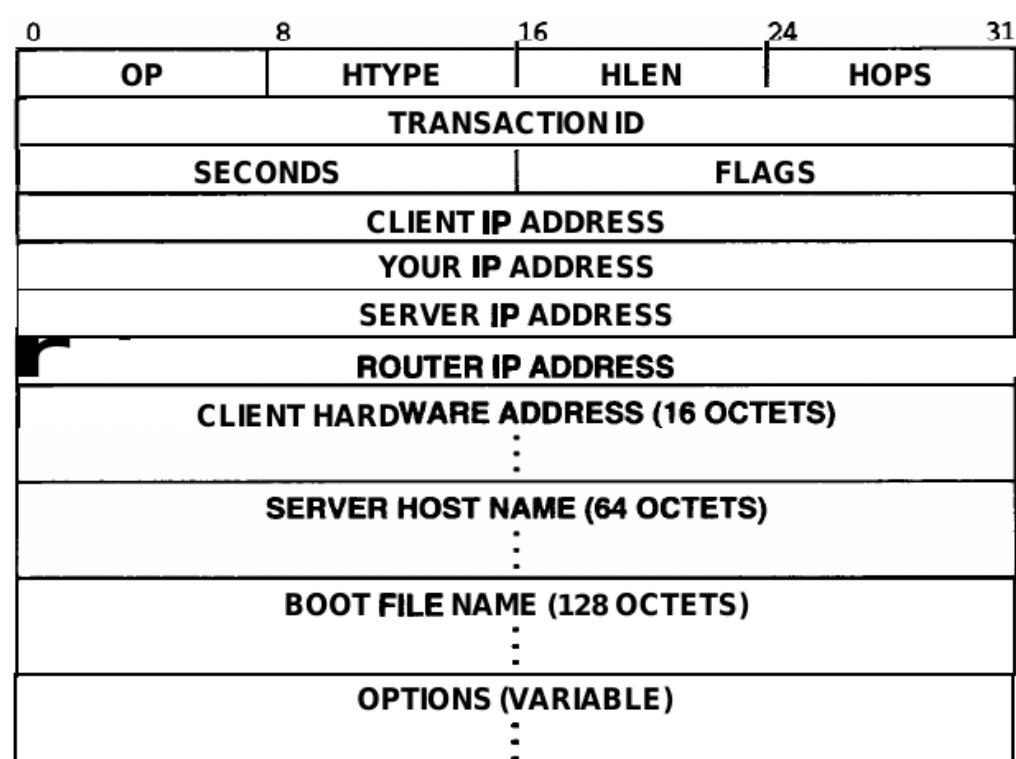


Figure 23.5 The format of a DHCP message, which is an extension of a BOOTP message. The options field is variable length; a client must be prepared to accept at least 312 octets of options.

DHCP

Stany kli dhcp...

normalny stan działania to „bound”
„DHCP*” to nazwy komunikatów

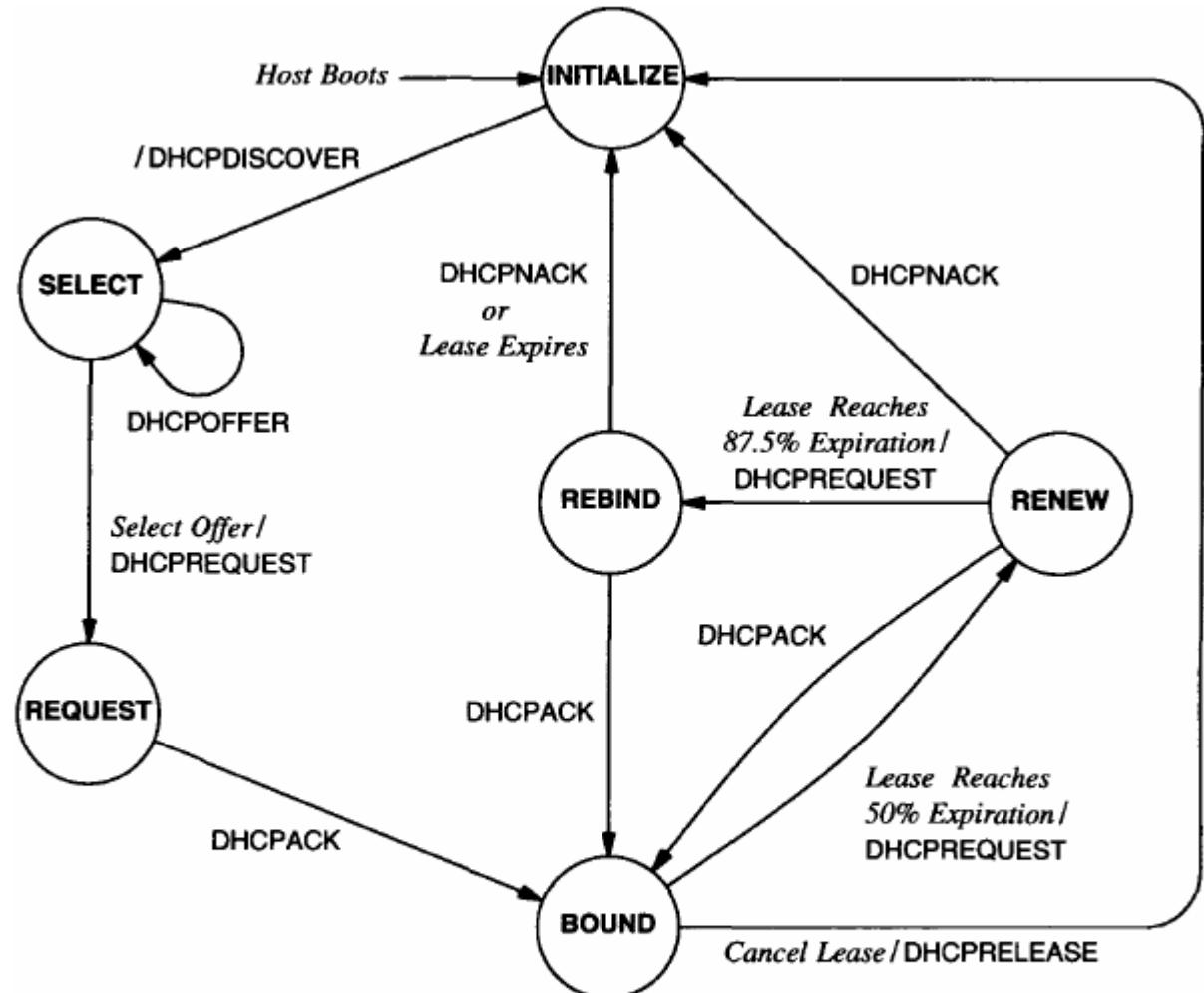


Figure 23.4 The six main states of a DHCP client and transitions among them. Each label on a transition lists the incoming message or event that causes the transmission, followed by a slash and the message the client sends.

IPv6 czyli ulepszenie war. 3 (IPv4)

IPv6 to modyfikacja IPv4 ,czyli wymiana 3 warstwy prot...
główna cecha: dłuższe adr: ipv4 32bit, ipv6 128bit (na pewno ich wystarczy...)

Uproszczony nagłówek ipv6 w porównaniu do ipv4, 40 bajtów,
hop limit = ttl, traffic class i flow label = na użytek QoS, payload len = długość danych,
next header = typ następnego nagłówka (w danych)...

może to być nagłówek dodatkowy ipv6 lub nagłówek wyższej warstwy (UDP, TCP)

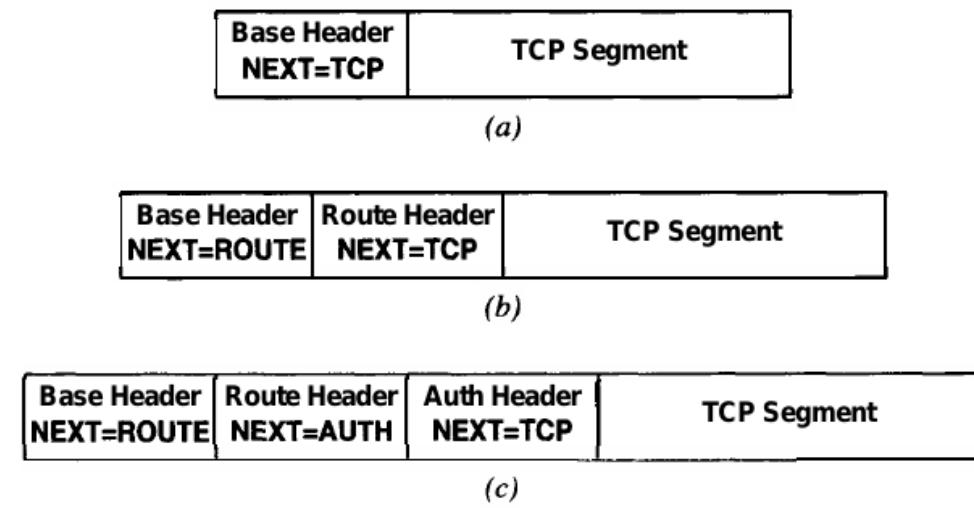
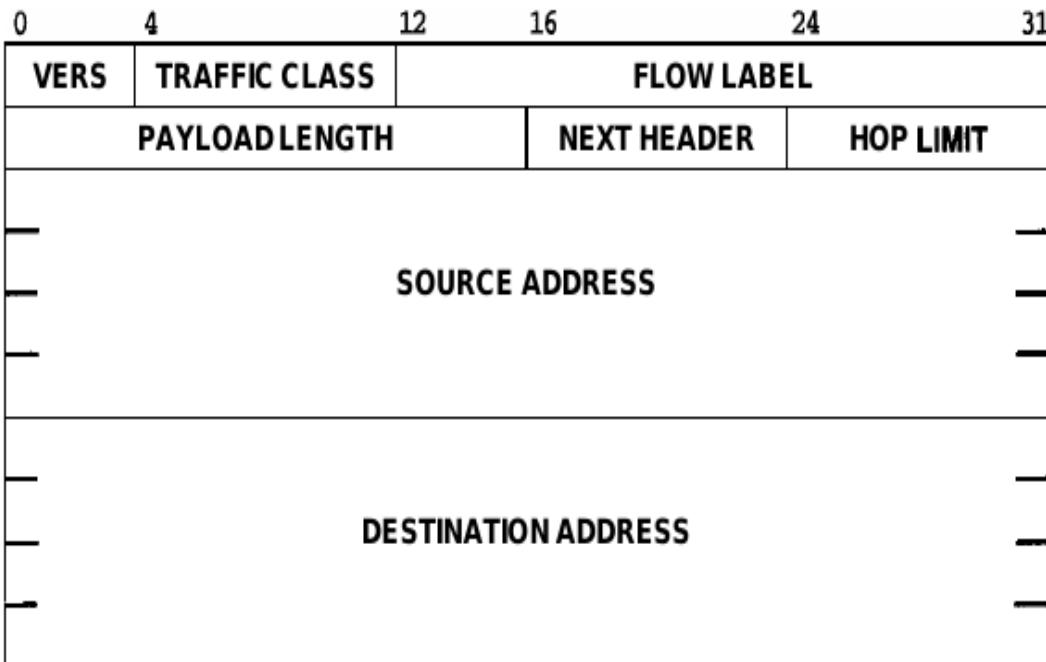


Figure 33.3 Three datagrams with (a) only a base header, (b) a base header and one extension, and (c) a base header plus two extensions. The NEXT HEADER field in each header specifies the type of the following header.

IPv6

Adresy ipv6 i ich skróty:

2001:0db8:0000:0000:0000:ff00:0042:8329

2001:db8:0:0:0:ff00:42:8329

2001:db8::ff00:42:8329

„::” oznacza ciąg zer, może wystąpić tylko raz! „::1” to localhost

Adresy budowane na podstawie innych:

Adresy LL = Local-Link, adr ipv6 na podstawie adr eth, ważne tylko w 1 sieci fizycznej,

FE80 :: X1 X2 : X3 FF : FE X4 : X5 X6

X1:X2:X3:X4:X5:X6 to adr eth ze zmodyfikowanym 1 bajtem (2 najmłodszy bit)

UWAGA: adr LL wymagają info o interfejsie bo mogą się powtarzać:

„adr_ipv6%eth0” (warto zrobic eksperyemnt !!!)

są też inne możliwości: np. ::FFFF:y1.y2.y3.y4 – adr ipv6 na bazie adr ipv4

Jak się włącza ipv6 na linuxie?

jeśli ipv6 działa to plik /proc/net/if_inet6 istnieje

„modprobe ipv6” włącza działanie ipv6 w linuxie od kernela 2.6 !!!

w linuxie, przydziela się adres LL do interf...

Współistnienie ipv4 i ipv6:

1. dwa stosey prot, z ipv4 i ipv6

2. tunelowanie pkg ipv6 w pkg ipv4

Typy adr dst:

Ipv4: unicast, broadcast, multicast

Ipv6: unicast, multicast, anycast (wysylamy do jednego hosta ze zbioru)

NIE ma pojęcia broadcasting !!

IPv6

Typy adresów ipv6 na podstawie **RFC 4291** (prefiks decyduje):

Address type	Binary prefix	IPv6 notation	Section
Unspecified	00...0 (128 bits)	::/128	2.5.2
Loopback	00...1 (128 bits)	::1/128	2.5.3
Multicast	11111111	FF00::/8	2.7
Link-Local unicast	1111111010	FE80::/10	2.5.6
Global Unicast	(everything else)		

Jeden if (np. eth0) może mieć: 1 adr ipv4, >=1 adr ipv6...

```
# ip addr add ::ffff:192.168.1.3/120 dev eth0
    # ^ ipv6 oparty na ipv4!! maska nie obejmuje ostat bajtu!
# ifconfig eth0
eth0      Link encap:Ethernet  Hwaddr 08:9E:01:1C:9C:70
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a9e:1ff:fe1c:9c70/64 Scope:Link
          inet6 addr: ::ffff:192.168.1.3/120 Scope:Global
              # ^ 1 adr ipv6 ma scope link, drugi ma scope global
```

Tabl routingowa osobna dla ipv4 i ipv6 („route -n -A inet/inet6” albo „ip route”)

```
# route -n -A inet6
Kernel IPv6 routing table
Destination „Next Hop” Flags Metric Ref Use Iface
::1/128 :: Un 0 2 0 lo
fe80::9ead:97ff:fe84:af1d/128 :: Un 0 2 0 wlan0
fe80::/64 :: U 256 1 0 wlan0
ff00::/8 :: U 256 2 0 wlan0
::/0 :: !n -1 1 0 lo
    # ^ to są reguły dla sieci bliskich...
```

SNMP

czyli zarządzanie sieciami komp

Manager i agent SNMP (działa na urządzeniu sieciowym)...

Obiekty/zmienne MIB opisane w języku SMI (syntaktyka z ASN.1)

Agent utrzymuje zmienne opisujące urządzenie

Manager może odczytywać te zmienne i modyfikować

(na tym polega proto manager <-> agent, także oparty na ASN.1/encoder/dekoder)

Jest możliwość czekania na zdarzenie od agenta (trap)

Przykładowe oprogramowanie: program „scotty”...

Std drzewo obiektów/zmiennych ISO/ITU...

ścieżka korzeń liść – zm. prosta,

ścieżka korzeń „nie liść” - zm. złożona (obiekt ?), np. tablica

ścieżka może być ciągiem liczb lub etykiet...

Poddrzewo tego drzewa zawiera „MIB-2”,

którego impl jest obowiązkowa dla agentów snmp...

SNMP

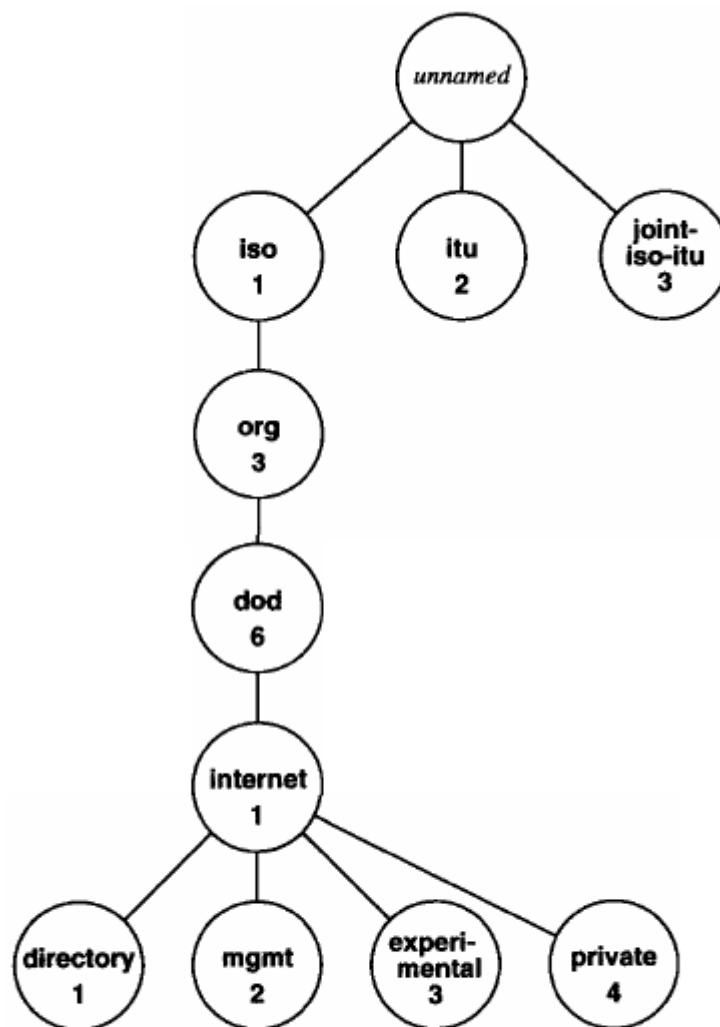


Figure 30.4 Part of the hierarchical object identifier namespace used to name MIB variables. An object's name consists of the numeric labels along a path from the root to the object.

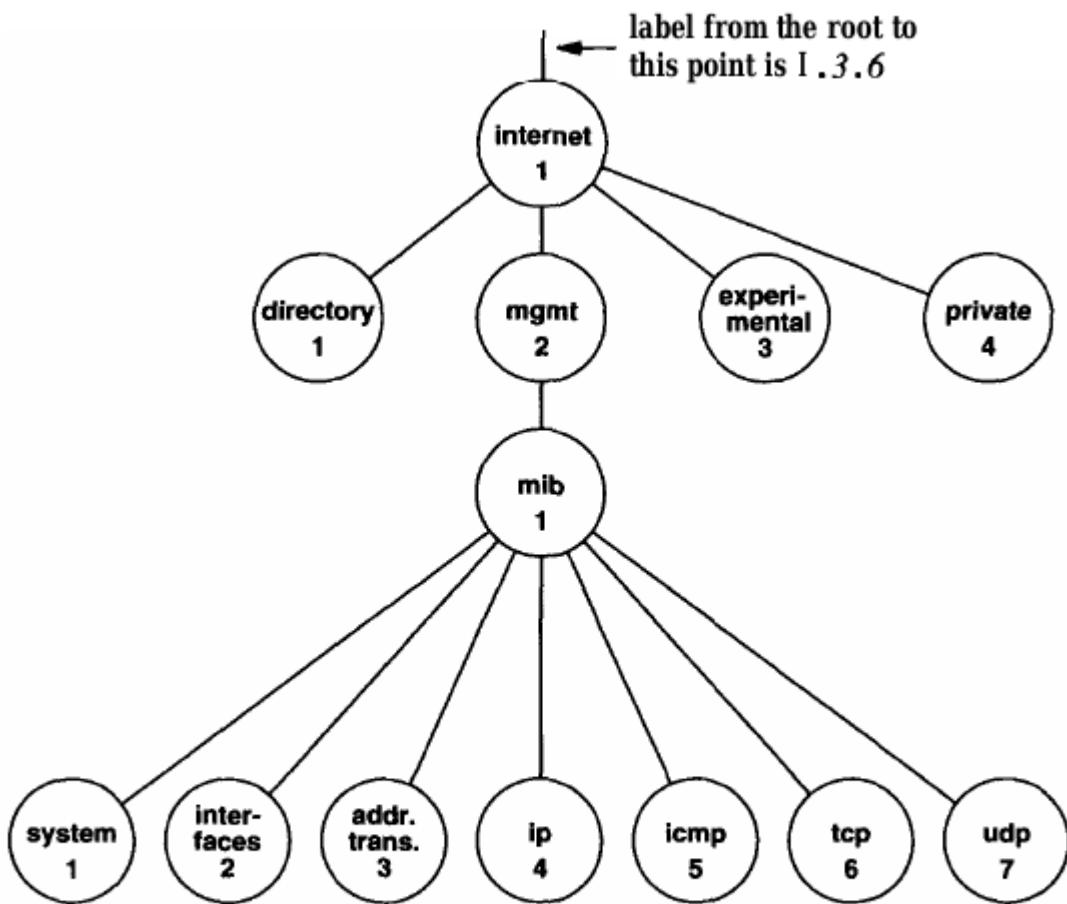


Figure 30.5 Part of the object identifier namespace under the IAB mib node. Each subtree corresponds to one of the categories of MIB variables.

Bezpieczeństwo...

Pojęcia:

poufność (ang. secrecy) – dane są zaszyfrowane

uwierzytelnianie (ang. authentication) - udowodnij kim jesteś

integralność (ang. integrity) – nie można złośliwie zmodyfikować danych

Gdzie się zapewnia powyższe rzeczy w sieci komputerowej?

nad warstwą 4 - połączenia tcp, SSL i TLS (obecnie obowiązuje TLS 1.3)

w warstwie 3 - IPsec (patrz iproute2), zabezpiecza wszystko „nad warstwą 3”

w warstwie 2 - wifi, WEP, WPA/WPA2

„warstwa app” - ssh, sftp, sshfs (wyżej niż ssl)

Inne zabezpieczenia:

bastion (wg Comera...),

szczególnie zabezpieczony komputer, do którego jest dostęp z zewnątrz

zapory sieciowe, firewalls,

w linuxie polecenie **iptables** (tabela filter),

nie wpuszczamy pewnych pakietów, router nie przekazuje pewnych pakietów

VPN = Virtual Private Network

sieć wirtualna zbudowana na publicznej sieci (ale zabezpieczona)

IPsec

Zabezpieczenie w warstwie 3...

Dwa nagłówki:

AH = Auth Header – uwierz src, integr

zawiera: next header (proto), SPI, seq num, auth data

ESP = Encapsualation Sec Payload – uwierz src, integr, poufność

zawiera: podobne pola jak AH

Security Agreement/Association, SA, logiczny kanał tworzony na początku...

zawiera: typ prot (AH, ESP), źródłowy adr ip, 32bit ID połączenia (SPI); SA jest jednokierunkowe

Nr proto w nagłówku ip: 51 i 50



Figure 7.8-1: Position of the AH header in the IP datagram.

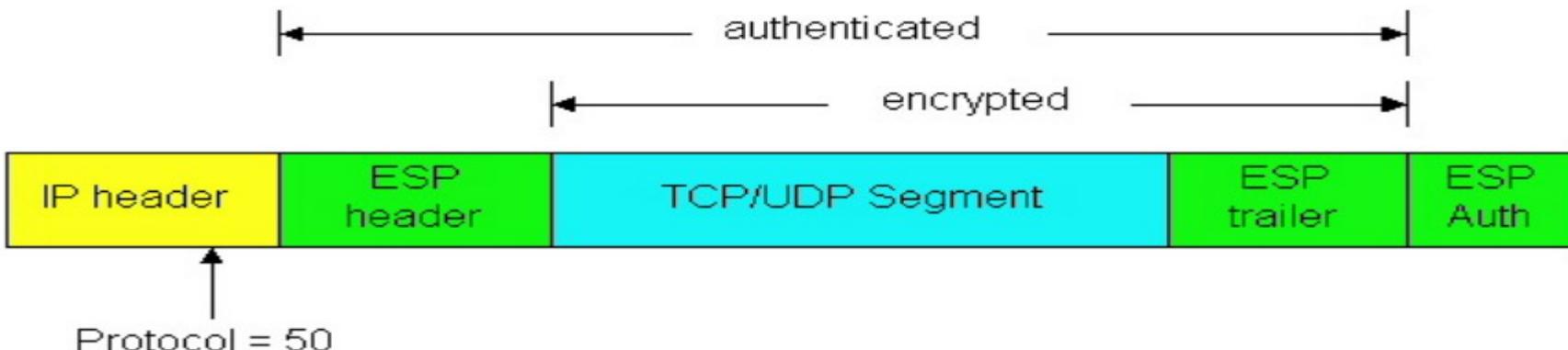


Figure 7.8-2: The ESP fields in the IP datagram.

Bastion (wg. Comera)

Bastion – maszyna między dwoma zaporami, pośrednik między internetem a siecią wewn
„zapora 1” przepuszcza pkg tylko do bastionu, „zapora 2” przepuszcza pkg tylko od bastionu;
komunikacja http intranet → internet za pośrednictwem „http proxy” na bastionie?

Sieć między zaporami: DMZ = strefa zdemilitaryzowana

Routery domowe: DMZ ma trochę inne znaczenie (?)

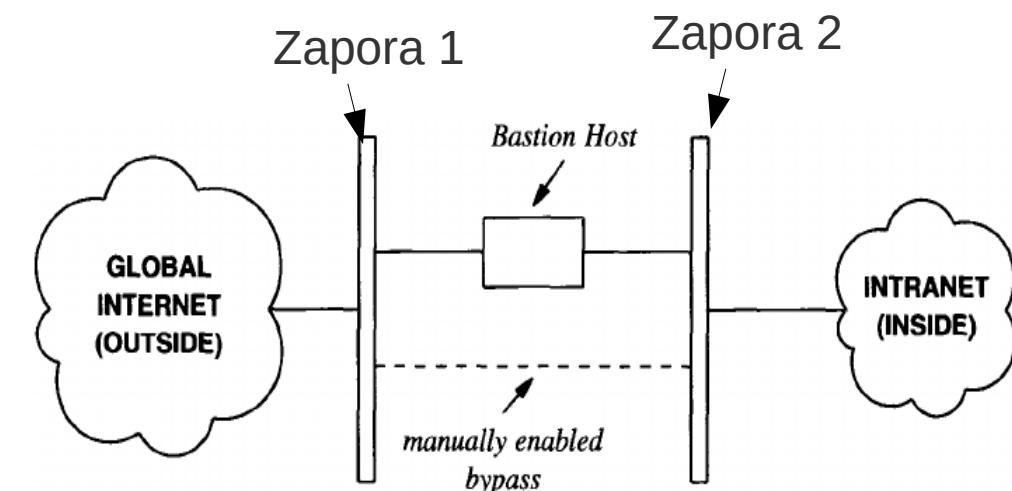


Figure 32.7 The conceptual organization of a bastion host embedded ~~in~~ a firewall. The bastion host provides secure access to outside services without requiring an organization to admit datagrams with arbitrary destinations.

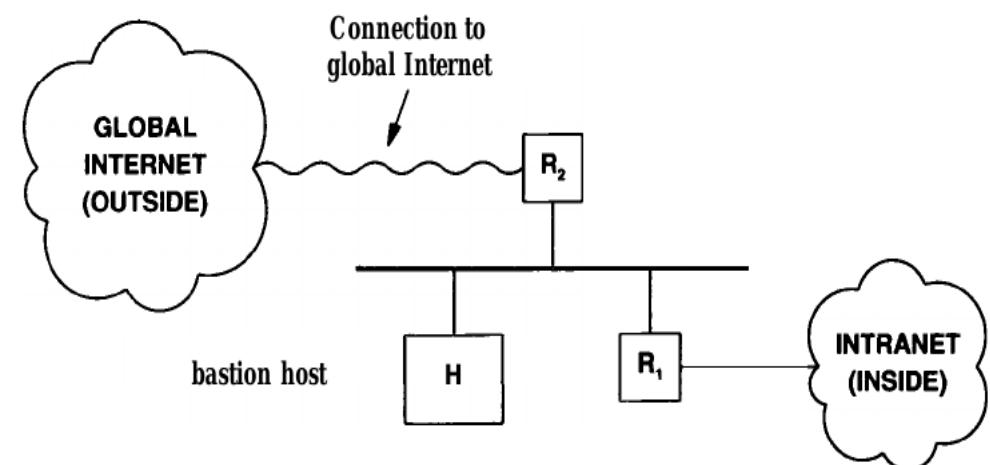


Figure 32.8 A firewall implemented with two routers and a bastion host. One of the routers has a connection to the rest of the Internet.

Zapory/ firewalls (iptables)

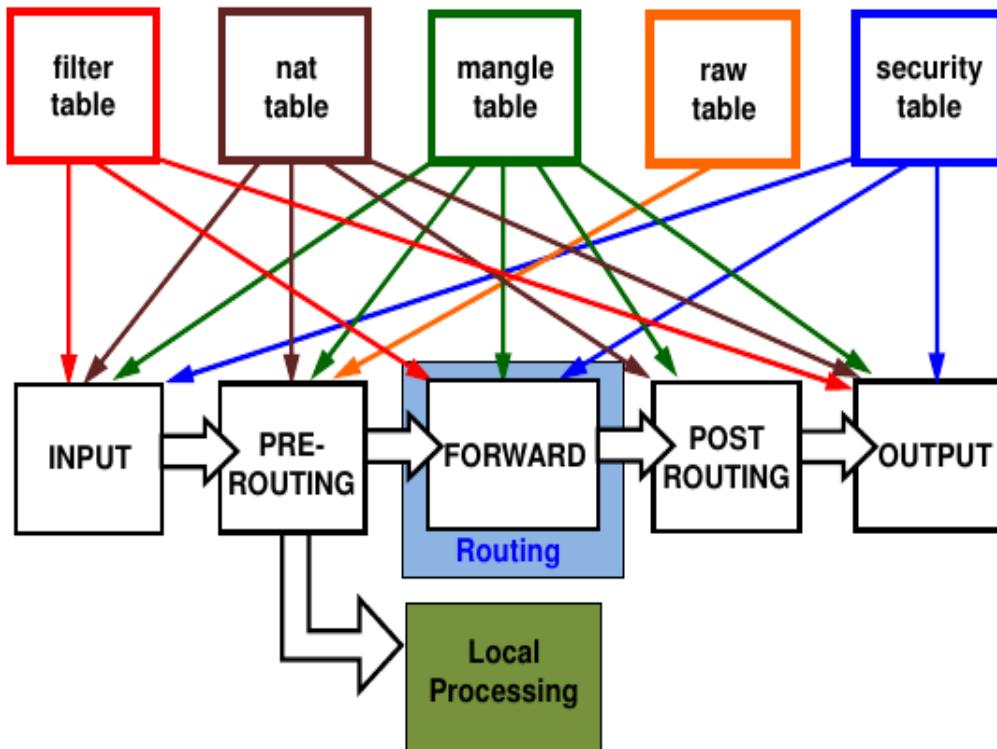


Table	Function	Chain
filter (default)	Packet filtering / firewall	INPUT FORWARD OUTPUT
NAT	Network Address Translation	PREROUTING INPUT OUTPUT POSTROUTING
mangle	Packet modification	PREROUTING INPUT FORWARD OUTPUT POSTROUTING
security	Mandatory Access Control	INPUT FORWARD OUTPUT
raw	Bypass "conntrack" for corner cases	PREROUTING OUTPUT

Table-1: Intables Tables & Chains

```
iptables -A INPUT -p tcp --dport 8080 -j ACCEPT
# wpuśczzamy pakiety tcp, dport 8080
iptables -A INPUT -j DROP
# odrzucamy wszystkie pakiety
```

iproute2

Następca net-tools...

to są komendy „ip” i „tc”
oraz wsparcie w kernelu

Co można zrobić

przy pomocy iproute2 ?

to co w net-tools,
kształtowanie ruchu (qdisc),
tuneli, ipsec,
zaawansowany routing,
routing multicastowy,

Docs:

Linux, Adv-Routing-HOWTO

Porównanie iproute2

i net-tools ...



NET-TOOLS COMMANDS	IPROUTE COMMANDS
arp -a	ip neigh
arp -v	ip -s neigh
arp -s 192.168.1.1 1:2:3:4:5:6	ip neigh add 192.168.1.1 lladdr 1:2:3:4:5:6 dev eth1
arp -i eth1 -d 192.168.1.1	ip neigh del 192.168.1.1 dev eth1
ifconfig -a	ip addr
ifconfig eth0 down	ip link set eth0 down
ifconfig eth0 up	ip link set eth0 up
ifconfig eth0 192.168.1.1	ip addr add 192.168.1.1/24 dev eth0
ifconfig eth0 netmask 255.255.255.0	ip addr add 192.168.1.1/24 dev eth0
ifconfig eth0 mtu 9000	ip link set eth0 mtu 9000
ifconfig eth0:0 192.168.1.2	ip addr add 192.168.1.2/24 dev eth0
netstat	ss
netstat -neopa	ss -neopa
netstat -g	ip maddr
route	ip route
route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0 i	ip route add 192.168.1.0/24 dev eth0

iproute2 / qdisc

Interf sieciowy (łącze) posiada kolejkę ramek do wysłania...

Qdisc = „dyscyplina kolejki”, czyli zasada działania kolejki

Polecenie iproute2 wyświetlające qdisc dla eth0, kolejka typu „**pfifo_fast**”

```
root# tc qdisc show dev eth0
qdisc pfifo_fast 0: root refcnt 2 bands 3 priomap 1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
```

Dyscyplina ta posiada 3 kolejki „bands” o priorytetach 0 (max pri), 1, 2

respektuje bity TOS pkg ip, na tej podstawie kwalifikuje pkg do odp kolejki...

Binary	Decimal	Meaning
1000	8	Minimize delay (md)
0100	4	Maximize throughput (mt)
0010	2	Maximize reliability (mr)
0001	1	Minimize monetary cost (mmc)
0000	0	Normal Service

UWAGA: obecnie pole tos ma inne znaczenie!!!
patrz: wikipedia Type_of_service, RFC 2474, DS field + ECN
(linux: dopiero w 2011 to zauważono, bit „min cost” jest zły...)

2 kolejki priorytetowe... zasada działania:
jeśli nie ma pkg z high pri to wysyłaj z low prio

TOS	Bits	Means	Linux Priority	Band
0x0	0	Normal Service	0 Best Effort	1
0x2	1	Minimize Monetary Cost	1 Filler	2
0x4	2	Maximize Reliability	0 Best Effort	1
0x6	3	mmc+mr	0 Best Effort	1
0x8	4	Maximize Throughput	2 Bulk	2
0xa	5	mmc+mt	2 Bulk	2
0xc	6	mr+mt	2 Bulk	2
0xe	7	mmc+mr+mt	2 Bulk	2
0x10	8	Minimize Delay	6 Interactive	0
0x12	9	mmc+md	6 Interactive	0
0x14	10	mr+md	6 Interactive	0
0x16	11	mmc+mr+md	6 Interactive	0
0x18	12	mt+md	4 Int. Bulk	1
0x1a	13	mmc+mt+md	4 Int. Bulk	1
0x1c	14	mr+mt+md	4 Int. Bulk	1
0x1e	15	mmc+mr+mt+md	4 Int. Bulk	1

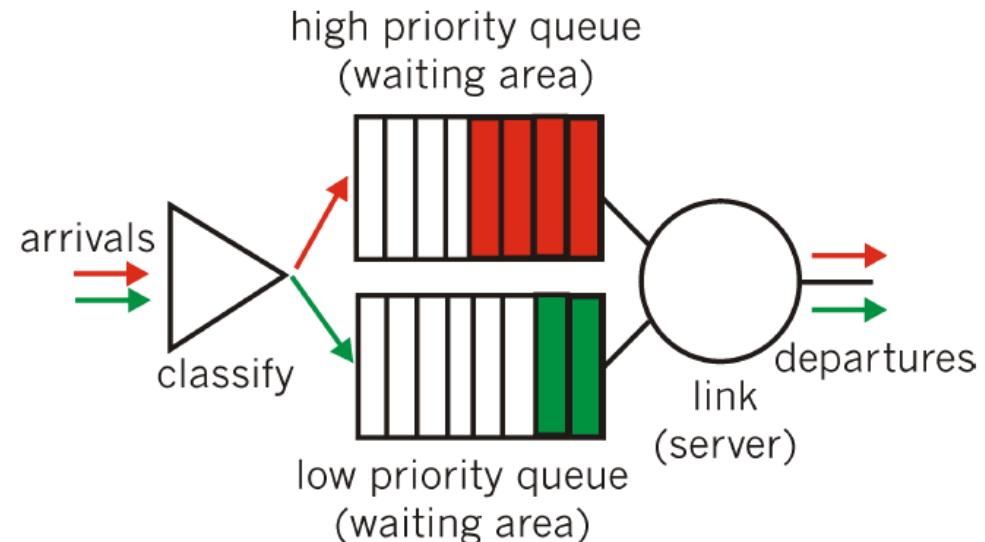


Figure 6.6-3: Priority queuing model

iproute2 / qdisc

Inne typy dyscypliny kolejek:

mq (?), używane przez wlan0... ?

SFQ (Stochastic Fairness Queue), „RR między połączaniami/strumieniami + hashi”

TFB (Token Bucket Filter), „1 dziurawe wiadro”

CBQ (Class Based Queue), pakiety są „klasyfikowane”, frakcja przepustowości na klasę

HTB (Hierarchical Token Bucket), „drzewo dziurawych wiader”, lepsze niż CBQ?

gdzie są opisy qdisc? np. <https://man7.org/linux/man-pages/man8/tc-sfq.8.html>

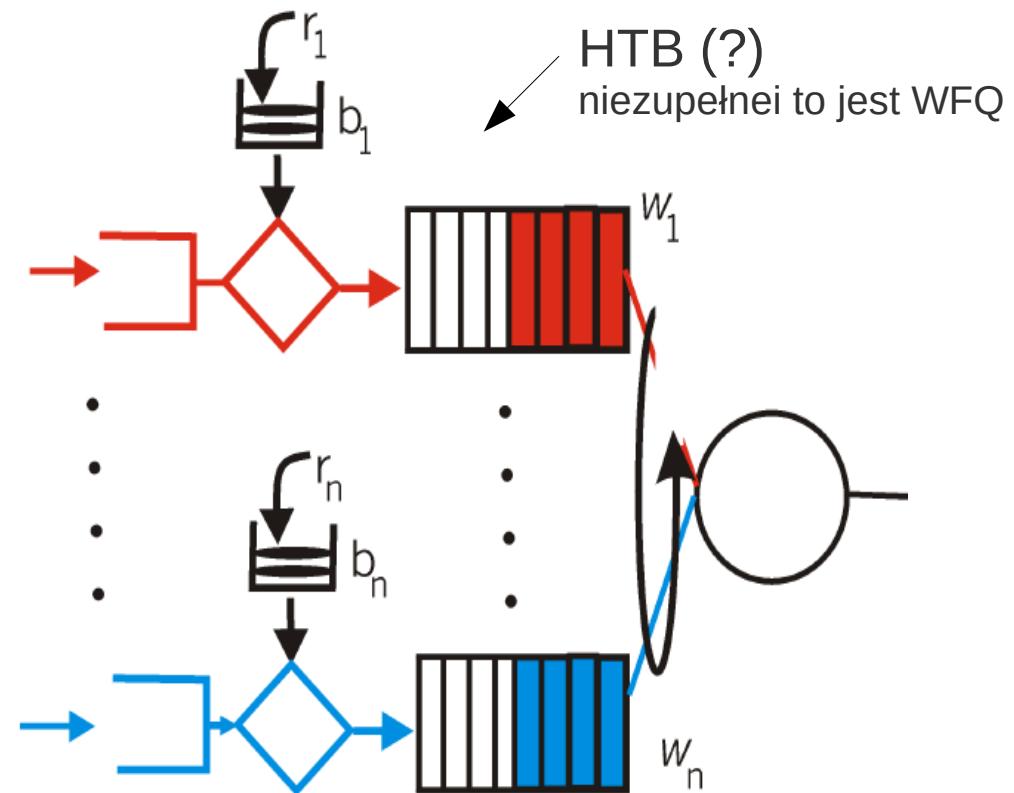
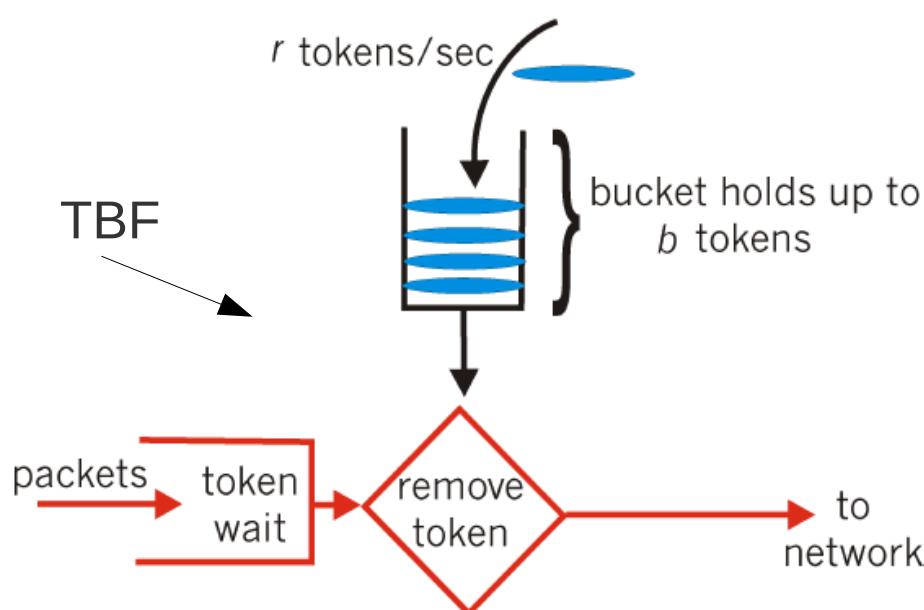


Figure 6.6-8: n multiplexed leaky bucket flows with WFQ scheduling

Figure 6.6-7: The Leaky Bucket Policer

WFQ = Weighted Fair Queue

iproute2 / qdisc

Po co właściwie te „dyscypliny kolejkowe” ? (przypomnienie)

- chcemy inaczej traktować ruch voip niż ftp...
- obrona przed atakami: „dos”, np. zalewanie udp, SFQ vs fifo...
- klasyfikacja/ oznaczanie pkg + qdisc łączy routerów = **DiffServ**

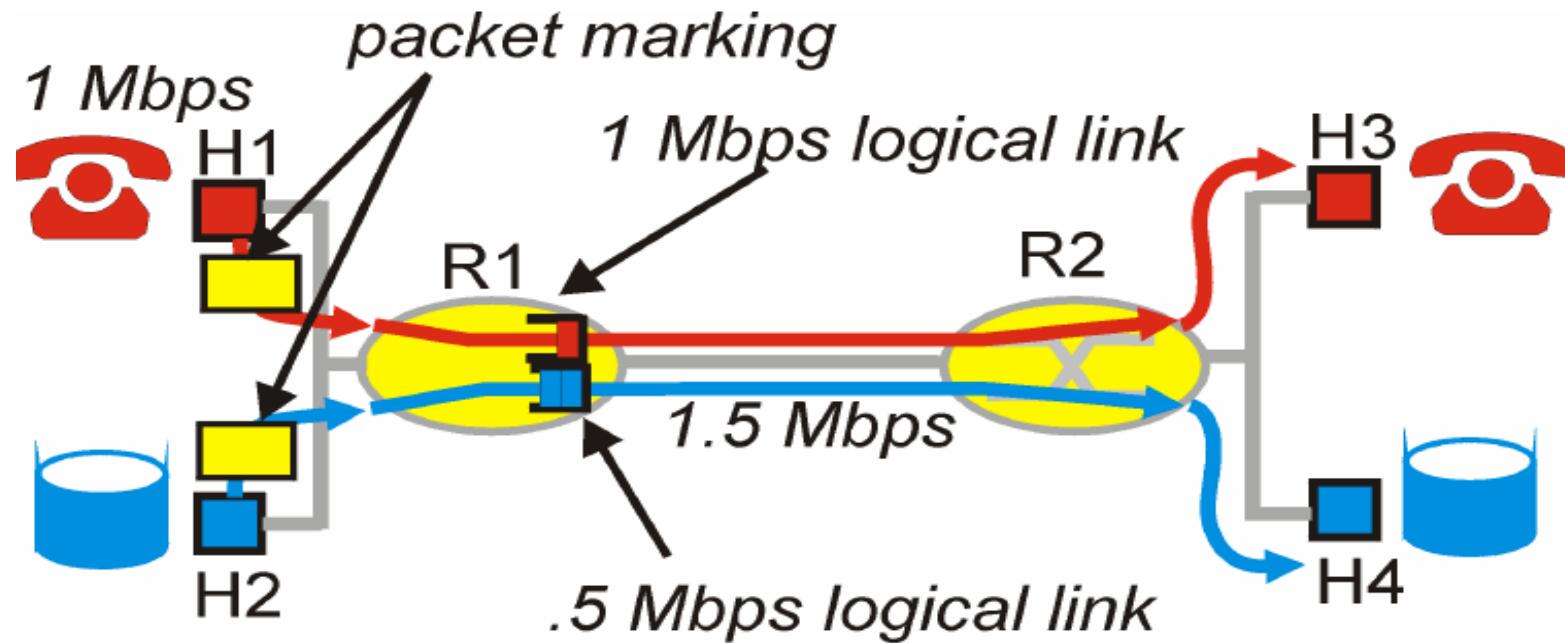


Figure 6.5-4: Logical isolation of audio and ftp application flows

Kłopotliwe pojęcia dotyczące qdisc (wg MH):

- classful vs classless:
classful = jest klasyfikacja pkg, klasy są różnie traktowane
- shaping vs scheduling:
scheduling= zmiana kolejności pkg, shaping = opóźnianie pkg

tunele

Def tunelu: wirtualne połączenie nad innymi protokołami,
które zachowuje się jak pojedyncza sieć fizyczna (2 węzlowa)
Różne typy tuneli:

„**ppp nad tcp**” lub „**ppp nad ssl nad tcp**” (ppp przenosi pkg ip)
łatwo stworzyć z uwagi na powszechność ppp...
wady: udp nad tcp?? także tcp nad tcp stwarza problemy...

Tunele SSH:

```
ssh -L local_port:host:port user@maszyna
      host:port po stronie sshd (remote)
ssh -R remote_port:host:host_port user@maszyna
      host:port po stronie ssh (local)
ssh -D port user@maszyna
      obsługa prot SOCKS4/5, dowolny port!!, ser myśli, że maszyna jest kli
```

Tunele iproute2:

„**gre**”, ipv4 nad ipv6 lub inne dowolne kombinacje, **BEZ szyfrowania !!**
tworzy się je plecieniem „**ip tunnel**”:

```
ip tunnel add mode gre local X remote Y
jako X i Y należy podać istniejące adr ip,
pojawi się wirt interf greX, który trzeba skonfig przesz ifconfig/route
można użyc adr ipv6 !!! (wtedy mamy ipv6 nad ipv4)
tunel trzeba skonfigurować na obu końcach !!
```

*Inne typy tuneli w iproute2: patrz „**ip tunnel help**” ... ipip, sit, ...*

tunel

Tunel „gre”: nagłówek gre między nagłówkami zewn a wewn pakietem, w nagłówku gre znajdują się: proto wewn pakietu i pewne flagi...



Ramka z gre przechwycona wiresharkiem...

Wireshark screenshot showing a captured GRE frame (Frame 1). The packet details pane shows the following structure:

- Ethernet II header: Src: 08:9e:01:1c:9c:70 (08:9e:01:1c:9c:70), Dst: 9c:ad:97:84:af:01 (Broadcast)
- Internet Protocol, Src: 192.168.1.3 (192.168.1.3), Dst: 192.168.1.23 (192.168.1.23)
- Generic Routing Encapsulation (IP)
 - Flags and version: 0000
 - 0... = No checksum
 - .0... = No routing
 - ..0. = No key
 - ...0 = No sequence number
 - 0.... = No strict source route
 -000 = Recursion control: 0
 - 0000 0... = Flags: 0
 -000 = Version: 0
 - Protocol Type: IP (0x0800)
- Internet Protocol, Src: 10.0.0.1 (10.0.0.1), Dst: 10.0.0.2 (10.0.0.2)
- Internet Control Message Protocol

The bytes pane at the bottom shows the raw hex and ASCII data for the captured frame.

IPsec przy pomocy iproute2

Używa się cmd „ip xfrm state/policy” ...

xfrm to „mechanizm” przekształcania pakietów (kompresja, szyfrowanie)
oprócz znanych pojęć: AH, ESP, SA jest jeszcze SP = „Security Policy”

Patrz:

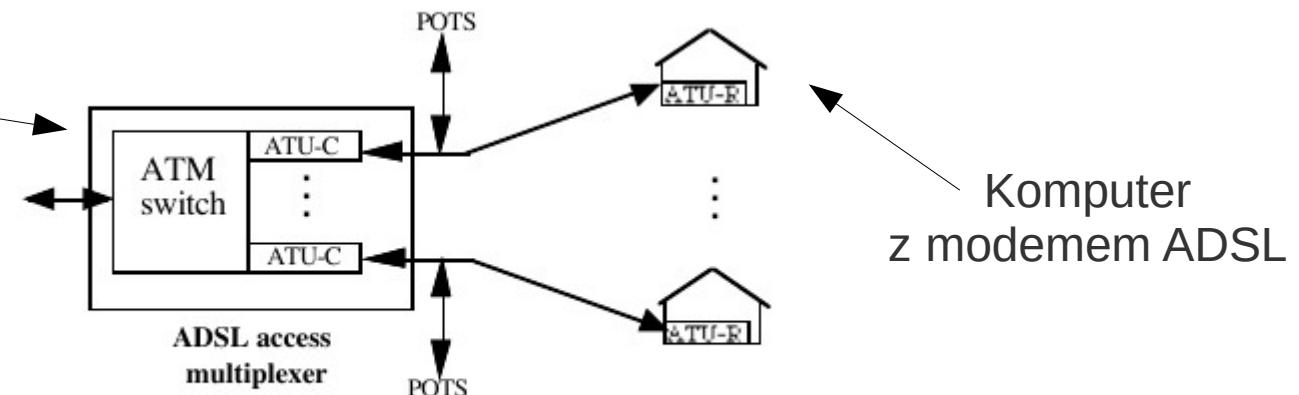
ip xfrm help
ip xfrm state help
ip xfrm policy help



Sieci dostępowe/ ADSL

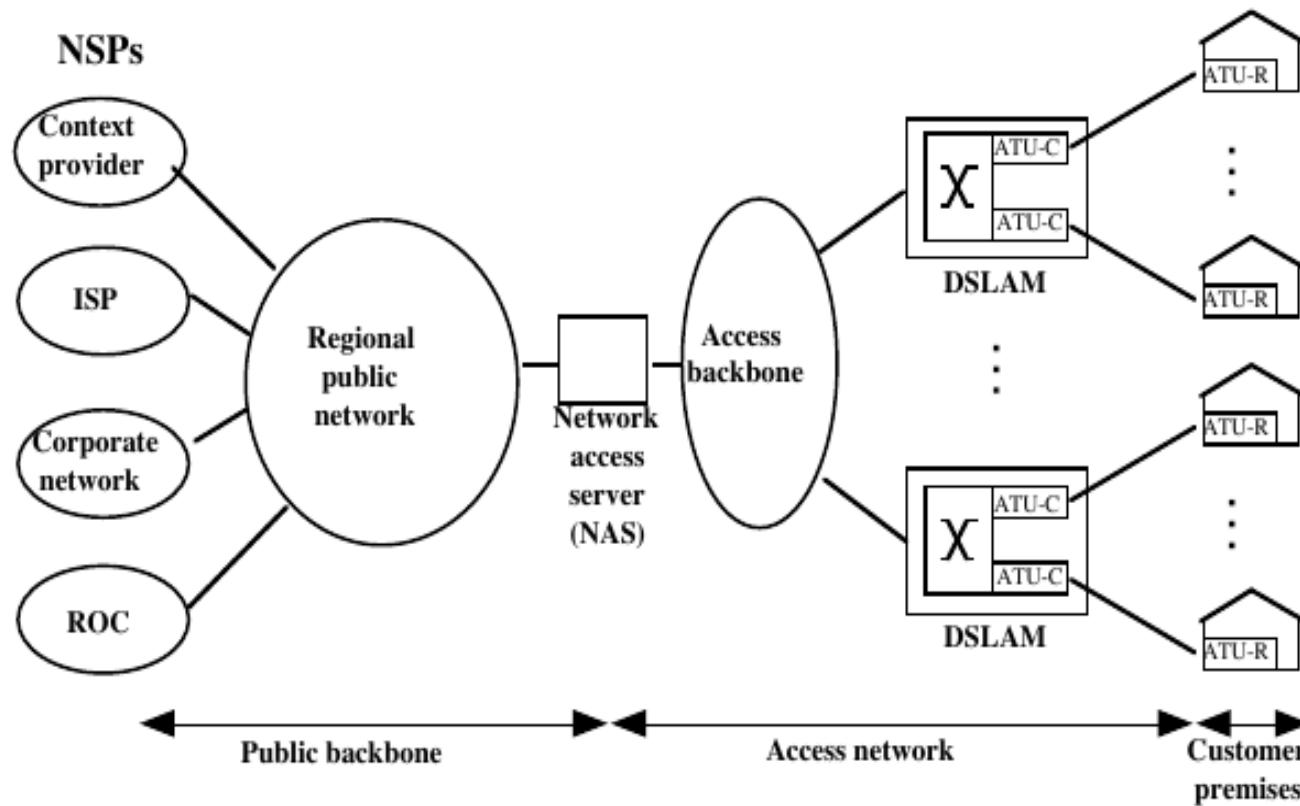
Urządzenie w centrali telefon.

DSLAM

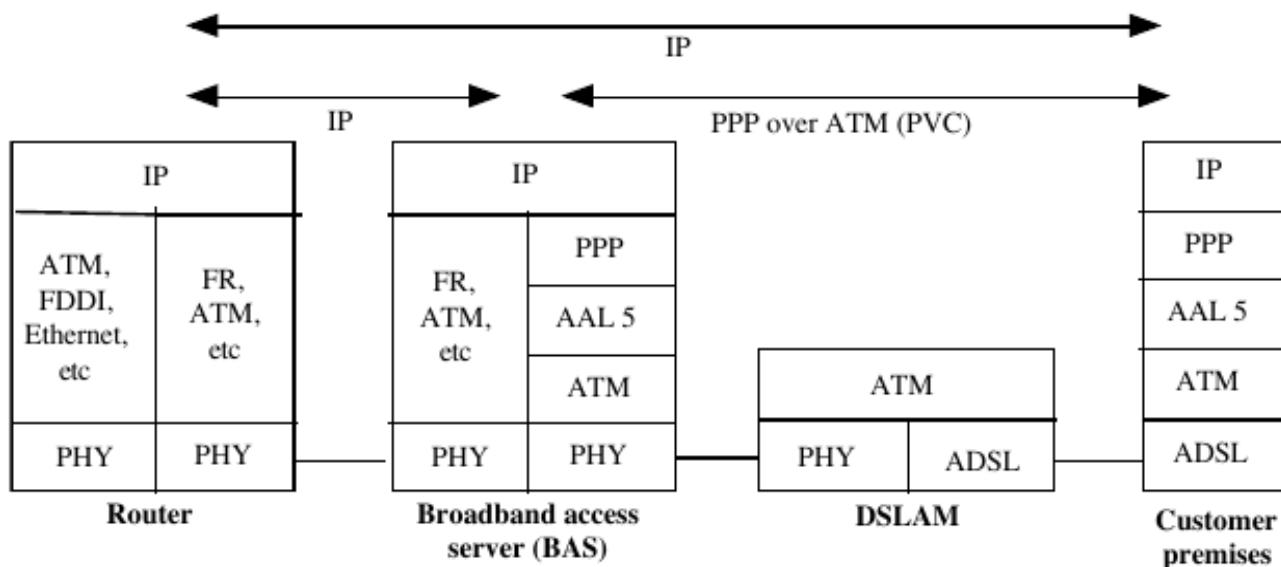
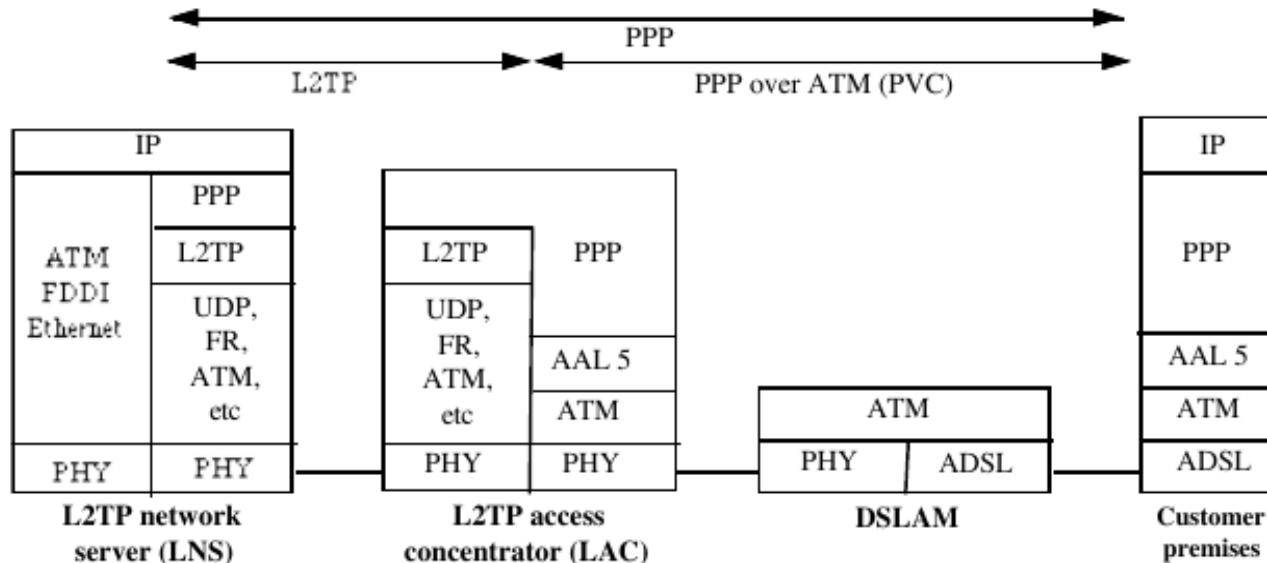


Komputer
z modemem ADSL

Figure 9.4: The ADSL access multiplexer (DSLAM)



Sieci dostępowe/ ADSL



Sieci dostępowe, światłowód

O samym światłowodzie;
budowa:

rdzeń (SM: 9um[mikrometry], MM 50-62um, domieszkowane szkło)

płaszcz (125um, szkło),

bufor

SM (1 modowy, 1 wiązka światła) vs MM (wielo modowy, wiele wiązek),
obecnie dominuje SM...

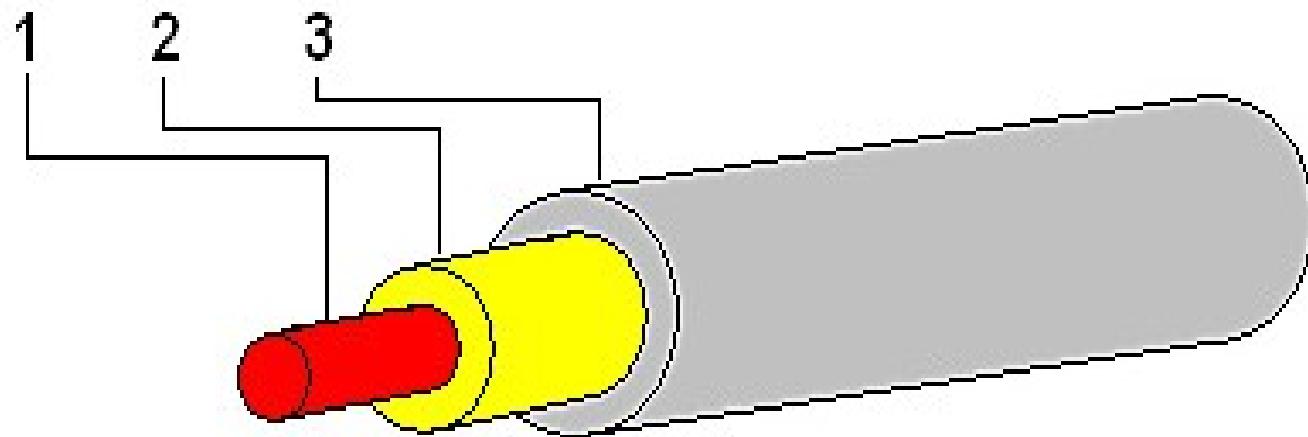
WDM: multipleksowanie po długości fali, 2 długości: 1310nm i 1550nm (nanometry)

DWDM: >2 długości fali

Światłowodowe sieci dostępowe:

FTTC = fiber to the curb, FTTH = fiber to the home

Sieci dostępowe, światłowód

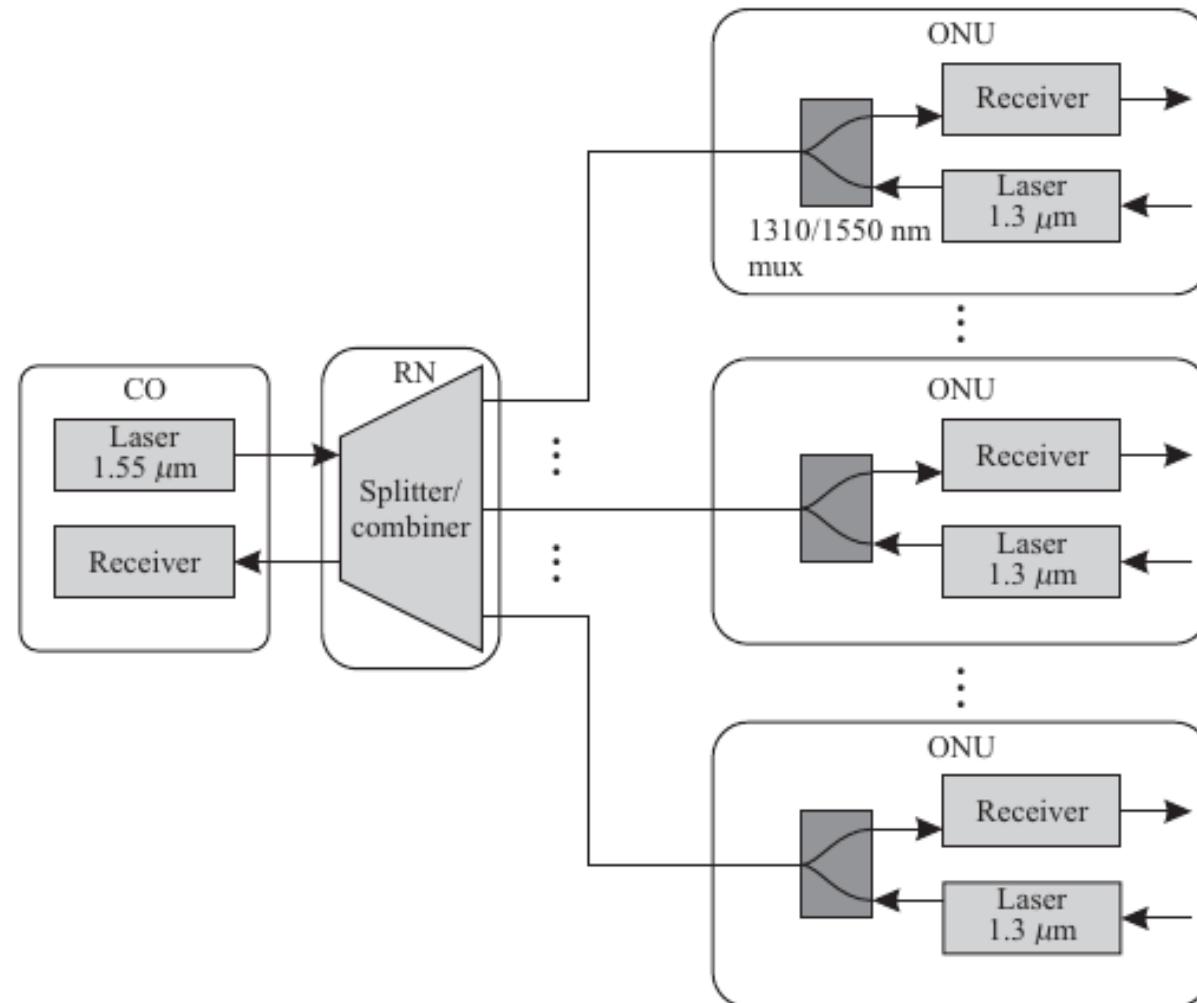


Budowa włókna światłowodowego

1. rdzeń
2. płaszcz
3. bufor

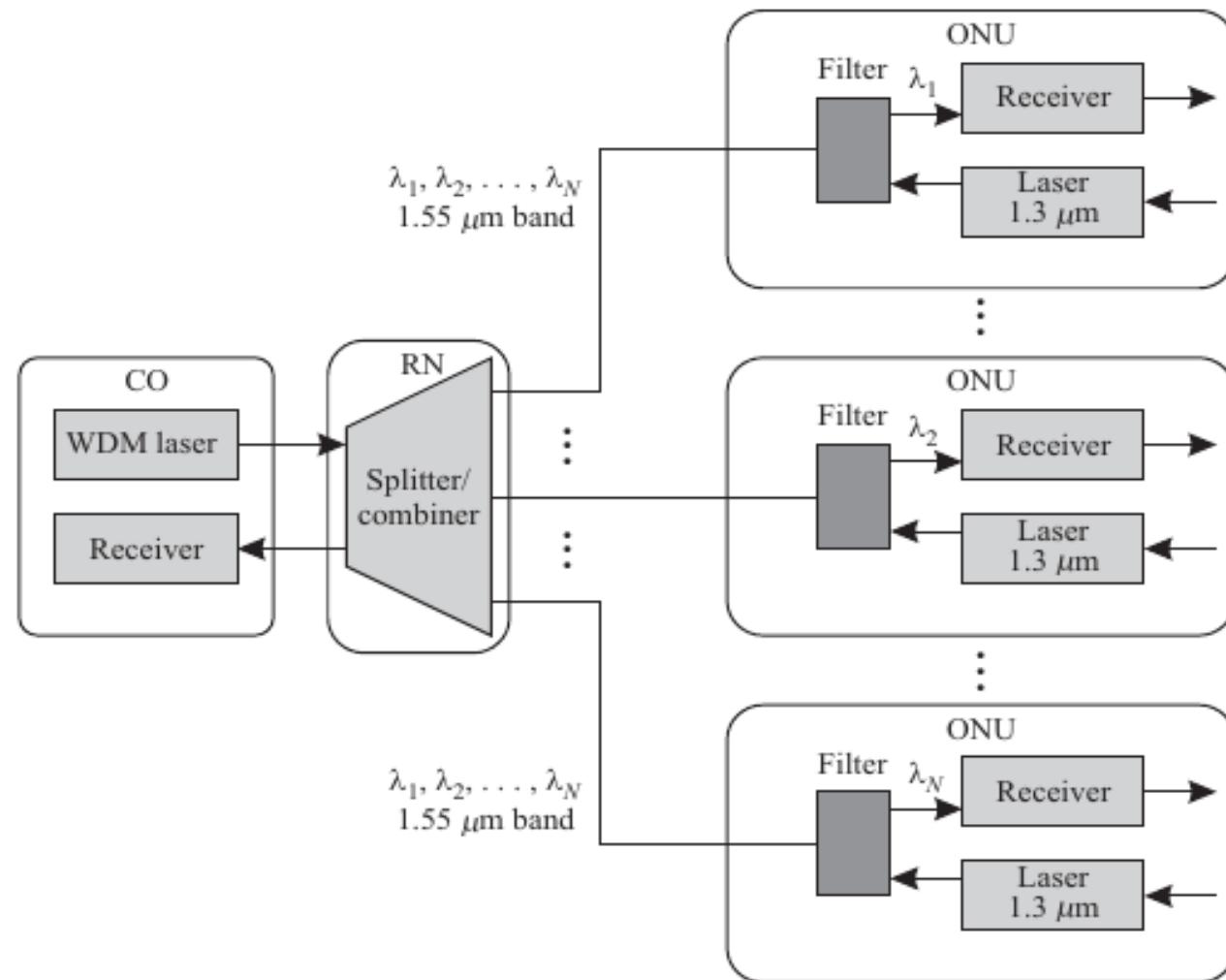
Sieci dostępowe, światłowód

Technologia; PON = Passive Optical Network, GPON (Gigabit PON)
odcinek CO-RN: → „broadcast”, ← TDM



Sieci dostępowe, światłowód

WPON...



UPnP

Możliwość sterowania urządzeniami (TV, router, ...) z obsługą upnp

Zasada działania:

1. automatyczne wykrywanie urządzeń, tzw zeroconf,
urządzenia wysyłają komunikat UDP na adr m.c. 239.255.255.250, port: 1900
specjalny komunikat... który zawiera m.in. typ urządzenia oraz url;
wykrywanie urządzeń to usługa **ssdp**...
2. pod podanym url-em jest opis serwisu w formacie xml,
te serwisy są b. podobne do WebServices SOAP (ale nie WSDL !!!)
serwisy te bywają standardowe... ale nie zawsze, patrz MainTVServer2...
3. instnieja toolkiti które potrafią wykryć urządzenia upnp,
zinterpretować opis serwisu,
oraz utworzyć pieniek, poprzez który można wywoływać met serwisu...

Niebezpieczeństwo:

np. router z włączonym upnp umożliwia swobodne konfig DNAT...

Komunikat UDP/ m.c. w którym TV ogłasza się jako MediaRenderer...

```
NOTIFY * HTTP/1.1
HOST: 239.255.255.250:1900
CACHE-CONTROL: max-age= 1800
LOCATION: http://192.168.1.5:7676/smp_14_
NT: urn:schemas-upnp-org:device:MediaRenderer:1
NTS: ssdp:alive
SERVER: SHP, UPnP/1.0, Samsung UPnP SDK/1.0
USN: uuid:0db58580-00e6-1000-9ba6-4844f7571770::urn:schemas-upnp-
org:device:MediaRenderer:1
```

„Ręczne” wykrywanie urządzeń upnp w j. Tcl...

```
load ~/tcl/tcludp.so udp; package require udp
set s [udp_open 1900]
fconfigure $s -translation crlf -buffering none
fconfigure $s -mcastadd 239.255.255.250
set licznik 0
fileevent $s readable { _puts "/// MH: licznik=$licznik\n[read $s]"; incr licznik }
```

Puszczanie na TV filmu .mp4 z serwera http...

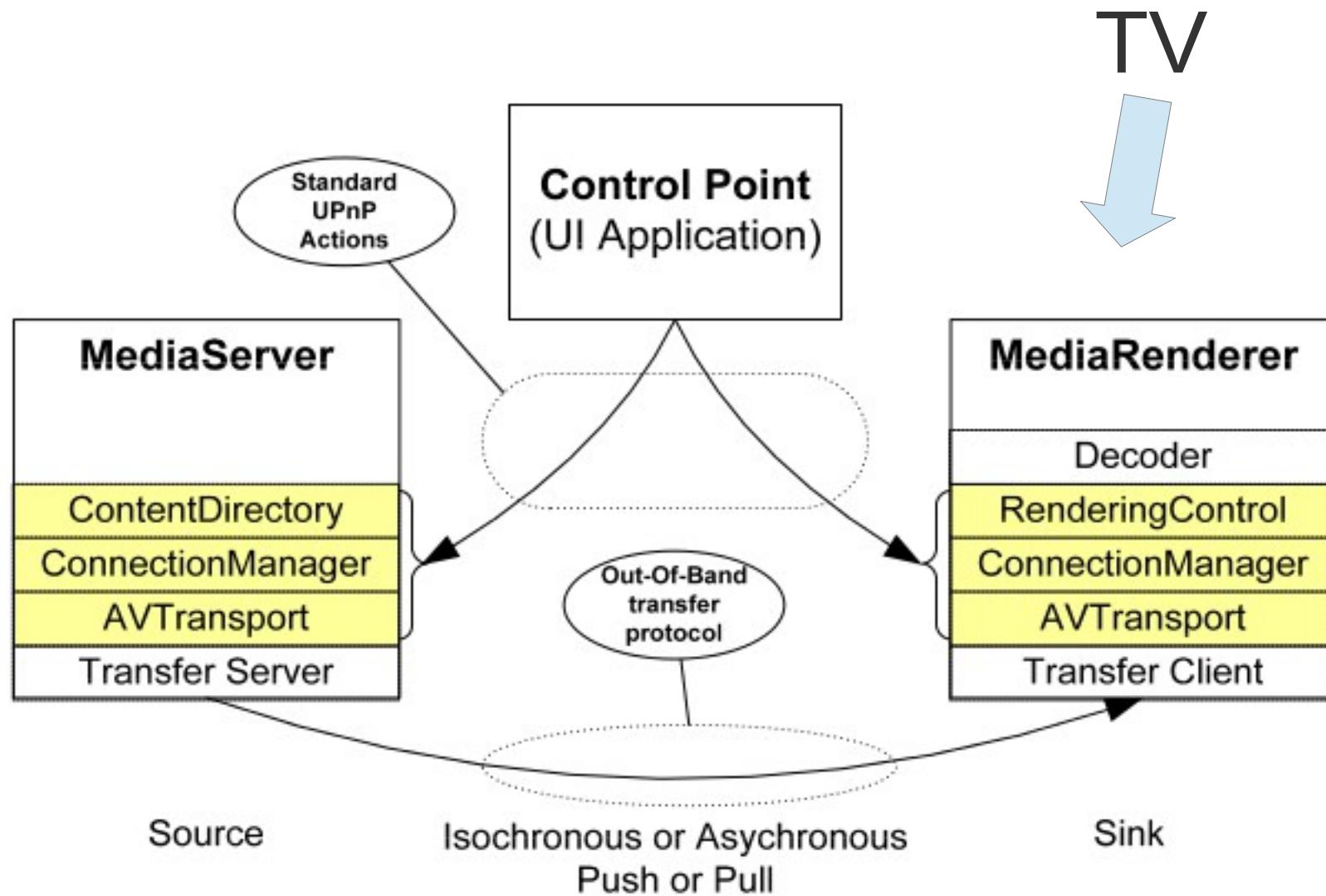
```
package require upnp
set host 192.168.1.13
proc nic args {_puts [info level 0]; set ::nic $args}
upnp discover urn:schemas-upnp-org:device:MediaRenderer:1 nic
set nic
#% 0db58580-00e6-1000-9ba6-4844f7571770

set url "http://$host:10000/filmy/jakis_tam_film.mp4"
set format "http-get:*:mpeg:"

${nic}::ConnectionManager::PrepareForConnection $format "" -1 Output
${nic}::AVTransport::SetAVTransportURI 0 $url ""
${nic}::AVTransport::Play 0 1
# + włączamy odtwarzanie filmu przez http...

${nic}::AVTransport::Stop 0
```

UPnP AV Architecture:1 – oficjalna specyfikacja...



Konfiguracja routera domowego przez upnp (**NIEBEZPIECZNE !!!**) dokładnie: konfiguracja DNAT...

```
package re upnp
proc nic args {_puts [info level 0]; set ::nic $args}
upnp discover urn:schemas-upnp-org:device:InternetGatewayDevice:1 nic

# (kod wyciągający długie namespaces, utworzone przez pkg upnp)
set dev(0) ::7177ea78-d2fa-383a-be46-8dd8683885f8
set dev(1) ::e618eb36-5d8a-3e53-bc3d-b9c9c93e10ea
set dev(2) ::3c5e0108-e3d9-3458-a437-5c5895b87cae
set dev(3) ::d8855a19-6d24-35c4-9738-5d59481e46ca
set dev(4) ::f561dcaa-cd26-3479-956c-f2b62c88f60b
set dev(5) ::86b4420e-4038-39a5-91b3-5b44117de7f8

${dev(3)}::WANPPPConn1::GetExternalIPAddress
    #% NewExternalIPAddress 81.219.205.137

${dev(3)}::WANPPPConn1::AddPortMapping {} 5000 TCP 10000 192.168.1.3 1 "serwer
www wibble" 0
    # + args: NewRemoteHost NewExternalPort NewProtocol NewInternalPort
NewInternalClient NewEnabled NewPortMappingDescription NewLeaseDuration
    # + na Netiaspot musi byc "NewLeaseDuration=0"

${dev(3)}::WANPPPConn1::GetSpecificPortMappingEntry {} 5000 TCP
    #% NewInternalPort 10000 NewInternalClient 192.168.1.3 NewEnabled 1
NewPortMappingDescription {serwer www wibble} NewLeaseDuration 0

${dev(3)}::WANPPPConn1::DeletePortMapping {} 5000 TCP
${dev(3)}::WANPPPConn1::GetSpecificPortMappingEntry {} 5000 TCP
```

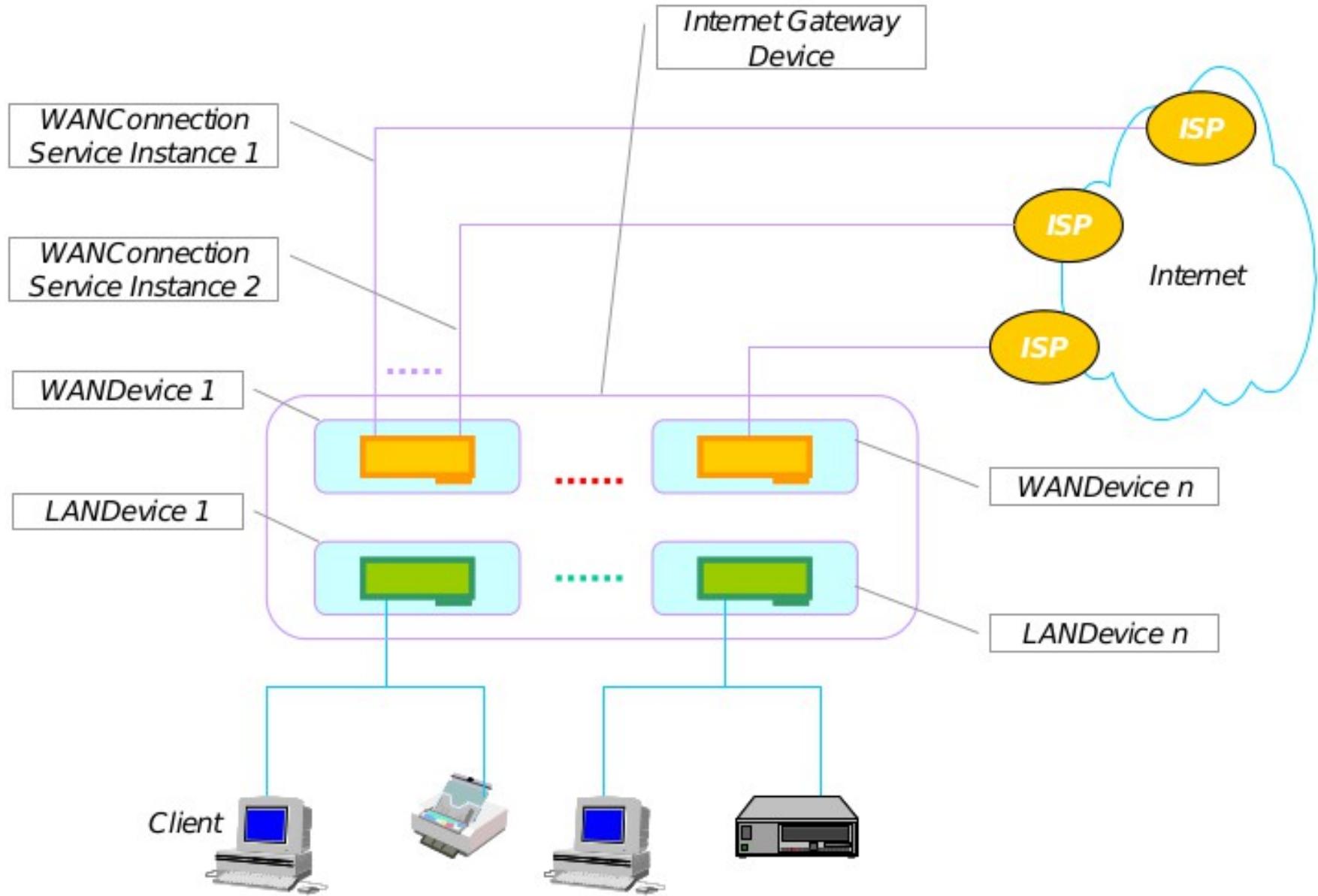
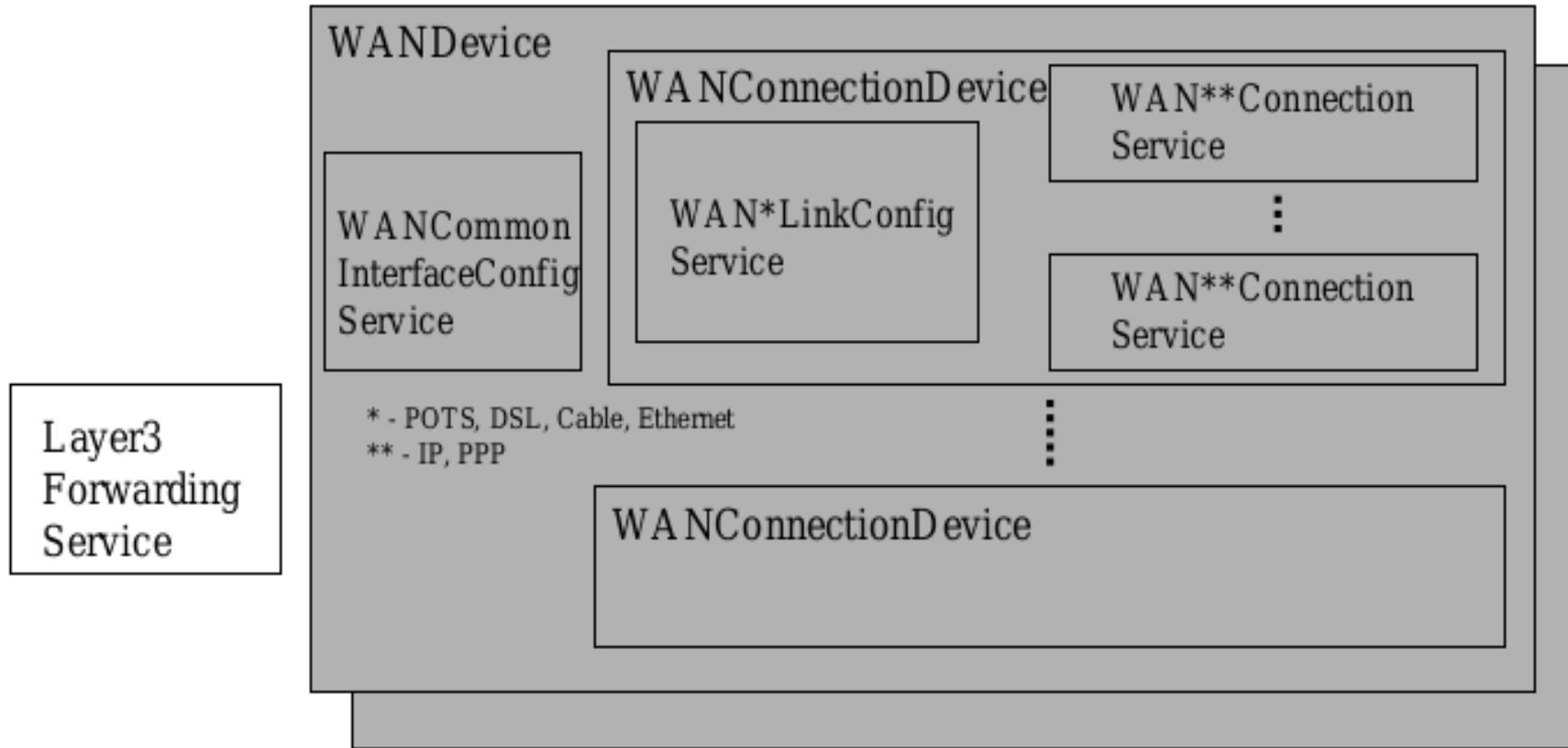


Figure 1: **InternetGatewayDevice** with LAN and WAN Interfaces

InternetGatewayDevice



LANDevice

LANHostConfigManagement
Service

Wifi/ hostapd/ dhcpcd3

Jak utworzyć AP wifi na komputerze z adapterem wifi...

Mode= managed, ad-hoc, monitor, master (?)

Hostapd to program tworzący AP wifi,

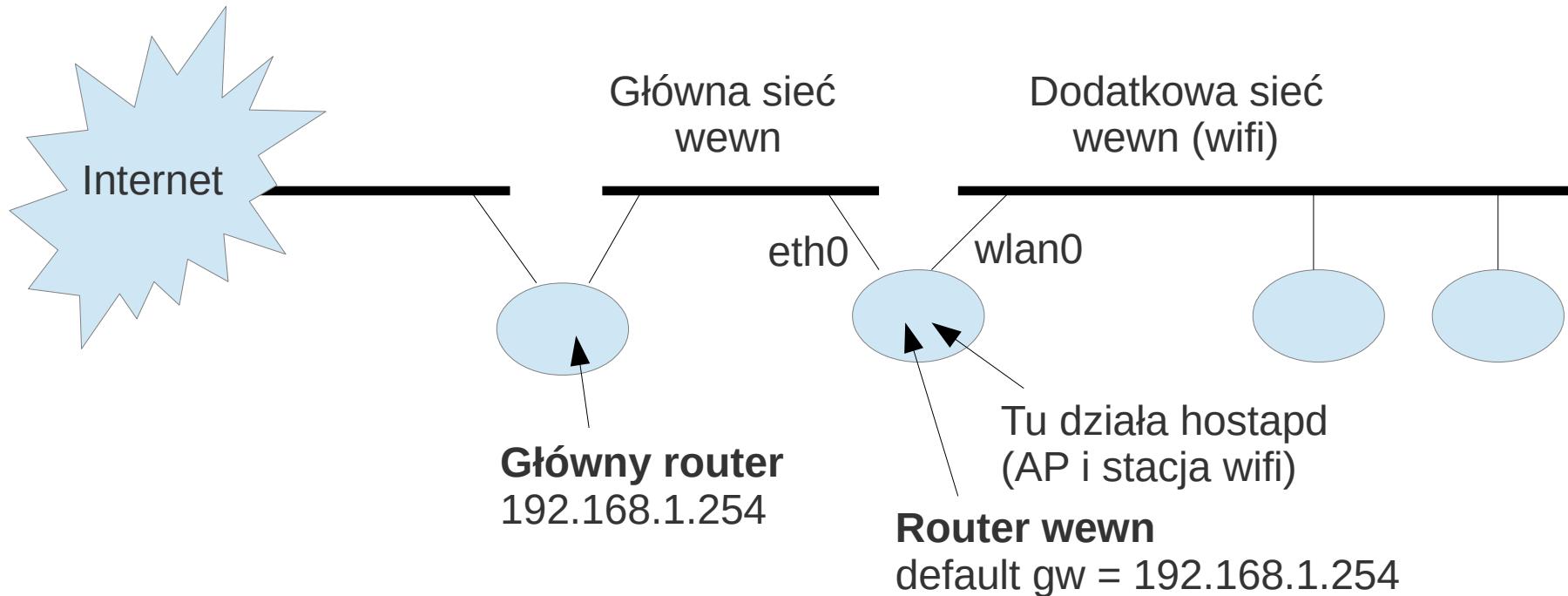
Dhcpcd3 to serwer usługi DHCP,

oba programy wymagają pliku config...

Tworzymy drugą sieć wewnętrzna...

jak dać dostęp do internetu bez konfigurowania „głównego routera” ?

Odp; 2x SNAT !!!



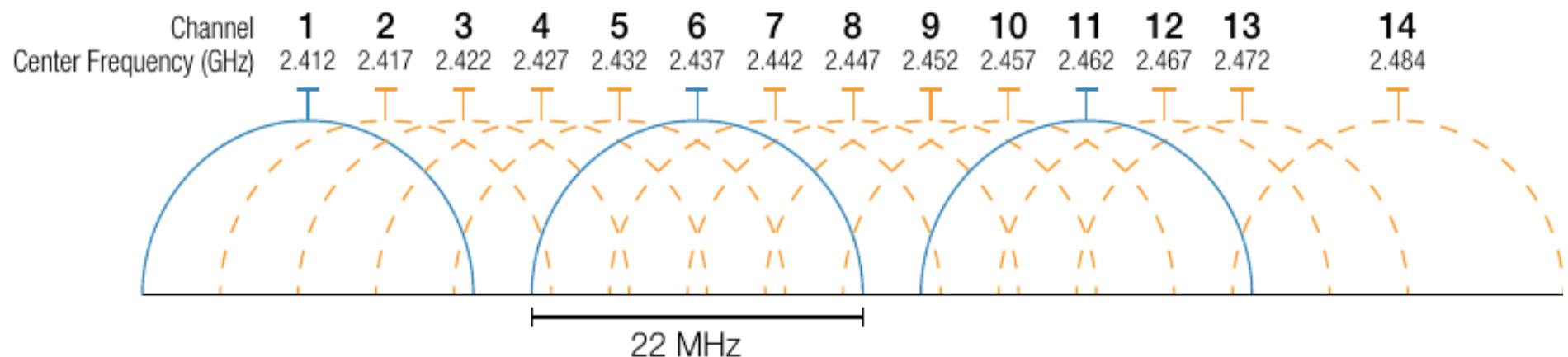
```
# włączanie hostapd i dhcpcd3 na routerze wewn...
# + są potrzebne 2 pliki konfiguracyjne: dhcpcd_sik.conf i hostapd_sik.conf
#
ifconfig wlan0 up 10.0.0.1 netmask 255.255.255.0
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -P FORWARD ACCEPT
iptables -t nat -F
iptables -t nat -A POSTROUTING -j SNAT -s 10.0.0.0/24 --to adr_ip_eth0
dhcpcd3 wlan0 -cf dhcpcd_sik.conf &
sleep 3
./hostapd -d hostapd_sik.conf
# pokazać pliki konfig hostapd i dhcpcd3...
```

Wifi/ uzupełnienia...

Stacja słucha ramek „beacon” na wszystkich kanałach!
ale potem działa na jednym z kanałów...

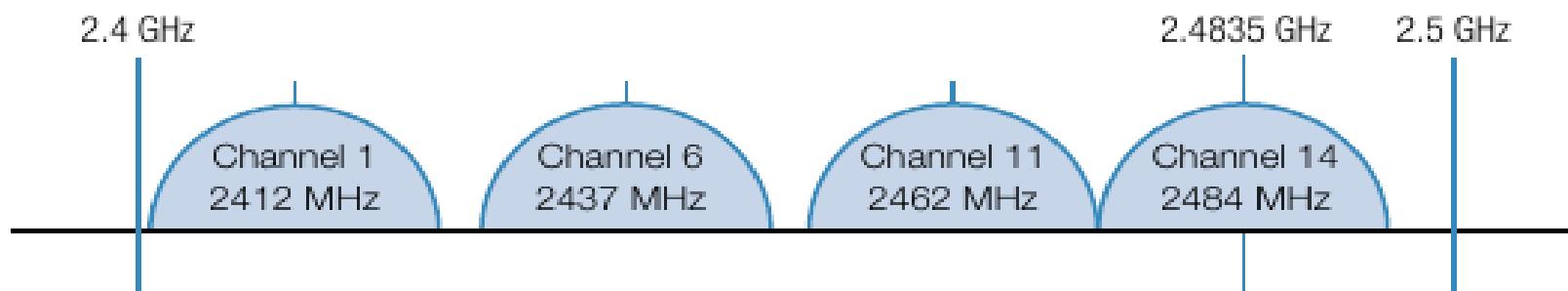
Różnica między CSMA w starych eth i wifi
nie tylko CD/CA; oczekiwanie na ciszę w medium...

Szerokość kanałów (20Mhz, 40Mhz), nakładanie się kanałów,
szerokość kanałów można ustawić np. w AP (patrz emulator)

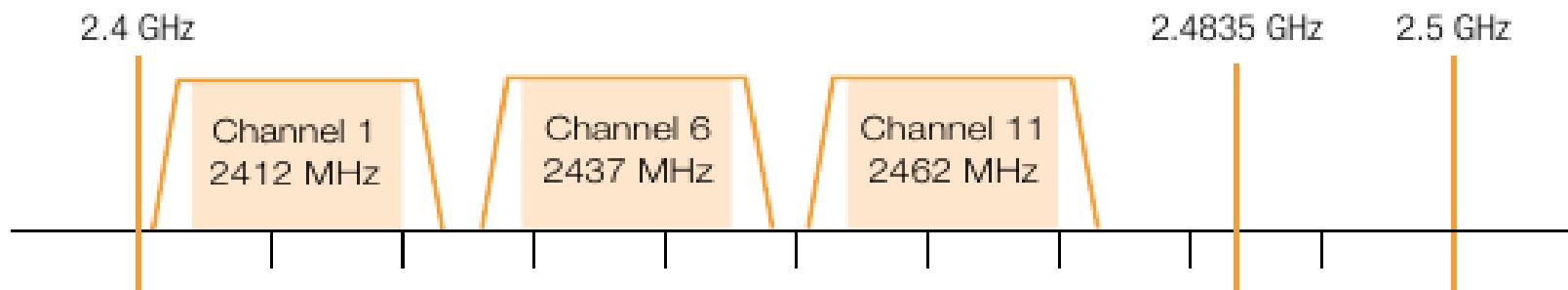


Non-Overlapping Channels for 2.4 GHz WLAN

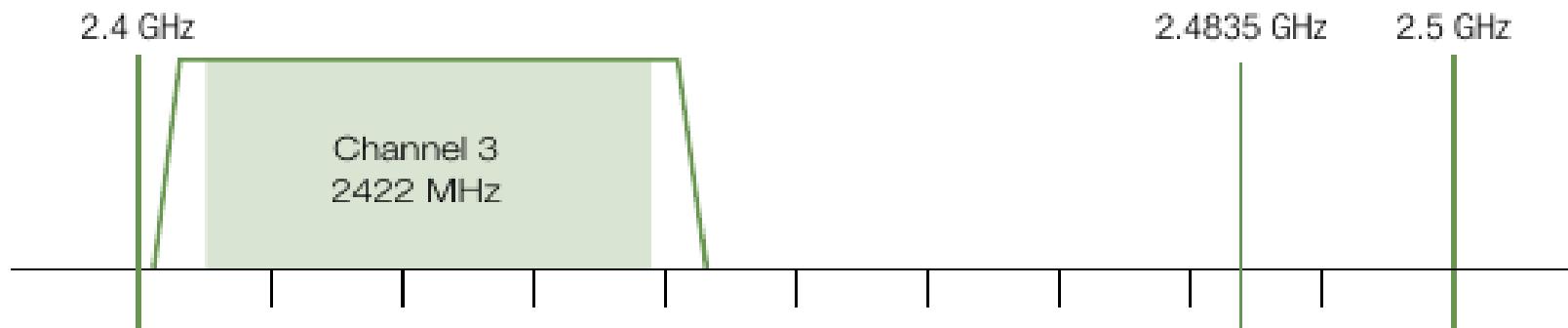
802.11b (DSSS) channel width 22 MHz



802.11g/n (OFDM) 20 MHz channel width - 16.25 MHz used by subcarriers



802.11n (OFDM) 40 MHz channel width - 33.75 MHz used by subcarriers



qdisc/ CBQ

Kolejka/ dyscyplina „classful”, dzieli łącze między różne klasy ruchu sieciowego, uzywa priorytetów, WRR (Weighted Round Robin), „pożyczania”... jest w linuxie jak i w NS-2, wynalazcy: Sally Floyd, Van Jacobson „Link-sharing and Resource Management Models for Packet Networks” 1995

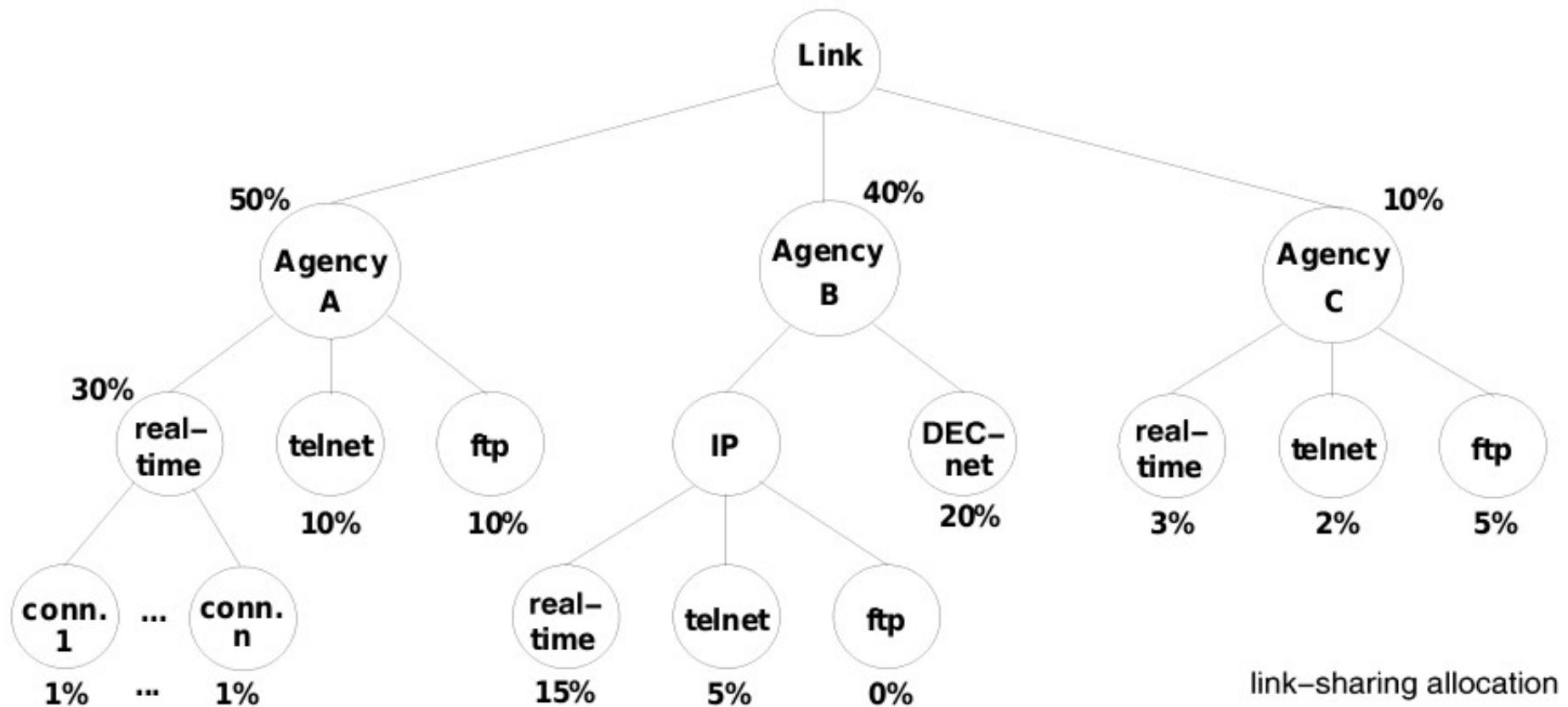
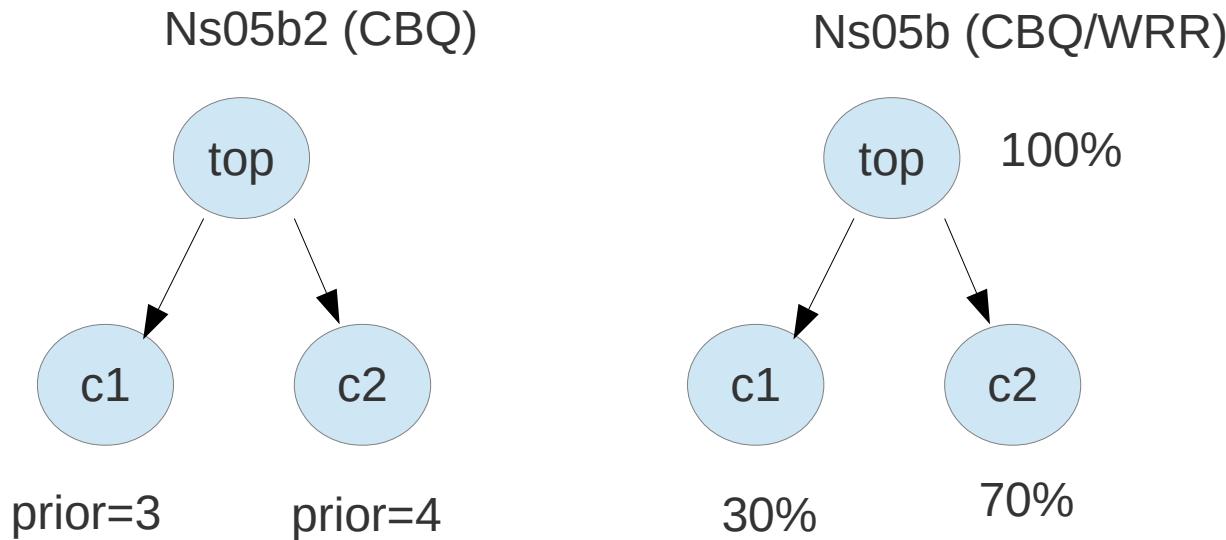


Figure 3: A hierarchical link-sharing structure.

Przykłady NS-2: CBQ i CBQ/WRR...



Sposoby wybierania pkg do wysłania (Schedulers):

1. general sched: priorytety + wagi (WRR)
2. link-sharing sched: regulacja klas ruchu

Regulacja klas: opóźnianie pakietów

Kiedy trzeba regulować klasy???

Symulator sieci komputerowych NS-2

Główna strona projektu: <http://www.isi.edu/nsnam/ns>
http://mhanckow.vm.wmi.amu.edu.pl:20002/zajecia/_xowiki2/SIK_G_zadania
(^ materiały zebrane przez MH)

NS-2 to symulator (nie emulator) sieci komputerowych...
pracuje się z nim "wsadowo", a wygląda to tak:

1. przygotowuje się skrypt (np. ns01.tcl), który:
 - + definiuje **topologie sieci** (węzły, połączenia, kolejki komunikatów)
 - + definiuje **ruch pakietów w sieci** poprzez określenie agentów, aplikacji oraz początkowych zdarzeń;
agent definiuje protokół, np. udp, tcp
aplikacja to np. ftp, cbr (constant bit rate)
2. uruchamiamy symulację programem ns: ./ns ns01.tcl
 - + program ns przeprowadza symulację i tworzy log/trace ze zdarzeniami w pliku ns01.tr oraz ns01.nam (animacje)
3. oglądamy log ns01.tr,
przetwarzamy go różnymi narzędziami np. programem awk, gnuplot,
oglądamy animację działania sieci przy pomocy ./nam ns01.nam

Sposób pracy z NS-2:

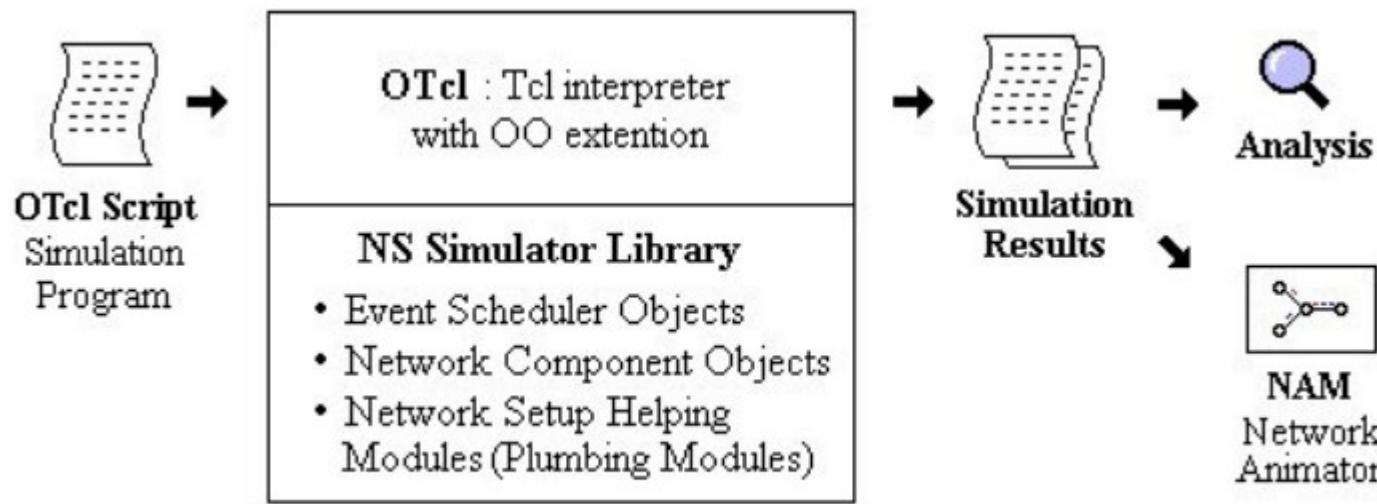


Figure 3. Simplified User's View of NS-2 [9]

Główna książka o NS-2: <http://www.isi.edu/nsnam/ns/doc/index.html>

(^ głównie potrzebna do rozszerzania symulatora...)

<http://www.isi.edu/nsnam/ns/tutorial/index.html> - tutorial 1

<http://nile.wpi.edu/NS/> - tutorial 2

T. Issariyakul, E. Hossain, "Introduction to Network Simulator NS2" (obszerna!!)

E. Altman, T. Jimenez "NS Simulator for beginners" (b. dobre!!) :

<http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf>

standardowy manual ns-2 :

http://mhanckow.vm.wmi.amu.edu.pl:20002/zajecia/_xowiki2/download/file/ns.1.pdf

Instalacja NS-2: pod lin32, skopiować .zip z folderu:

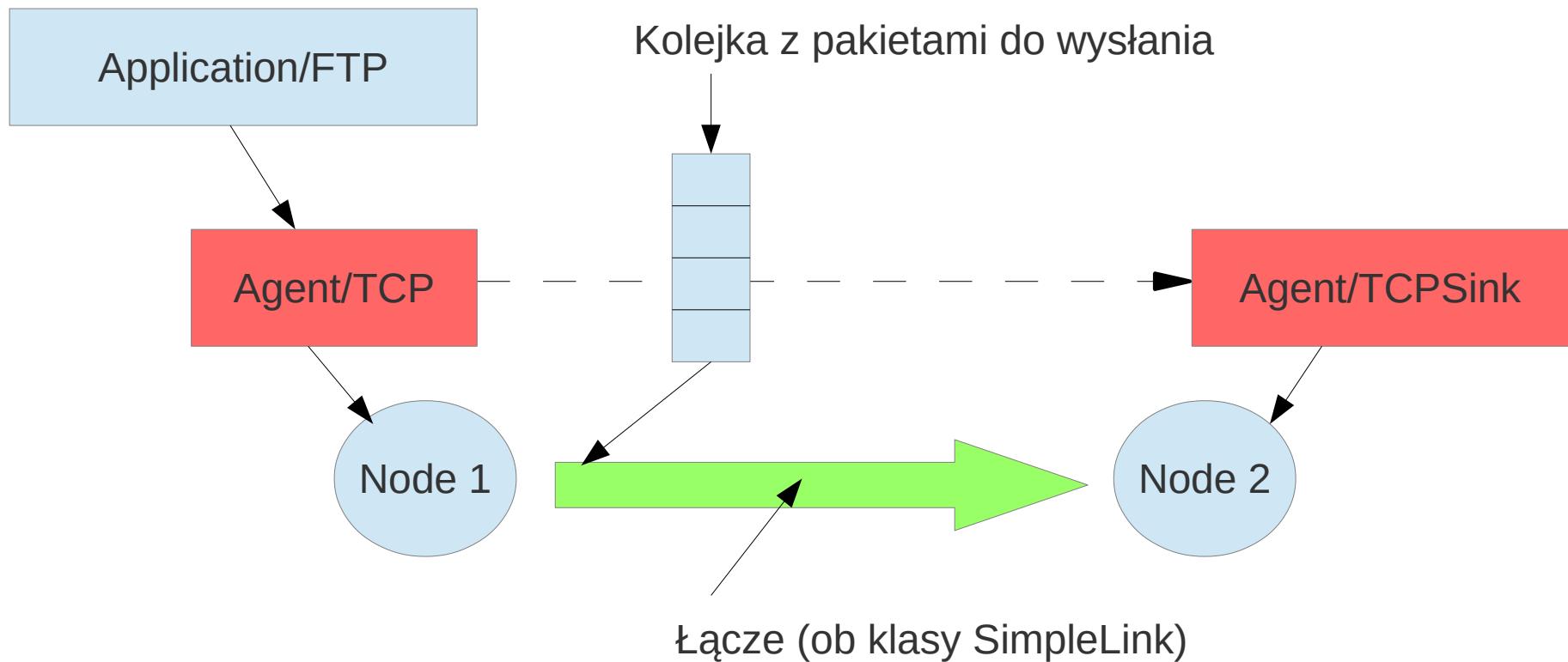
<https://mhanckow.students.wmi.amu.edu.pl/sik/ns-2/>

Pokażemy jak się definiuje sieć oraz ruch sieciowy przy pomocy ob:
Simulator, Node, SimpleLink, Agent/UDP, Agent/TCP*, Application/*, ...

Najprostszy kompletny przykład (ale z UDP i CBR):

http://mhanckow.vm.wmi.amu.edu.pl:20002/zajecia/_xowiki2/SIK_G_przyk01

Zasada łączenia aplikacji, agenta i węzła (node):



Tworzenie połączenia (2x SimpleLink):

Podaje się: 1) przepustowość, 2) opóźnienie, 3) typ kolejki (qdisc !!)

```
set ns [new Simulator]
$ns color 1 red
$ns color 2 blue
$ns namtrace-all [open ns07.nam w]
$ns trace-all [open ns07.tr w]
```

```
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
```

```
$ns duplex-link $n1 $n2 30Mb 50ms DropTail
$ns duplex-link $n2 $n3 20Mb 50ms DropTail
# kolejka DropTail (najprostsza)
# met duplex-link tworzy 2 ob. SimpleLink, w obie strony...
```

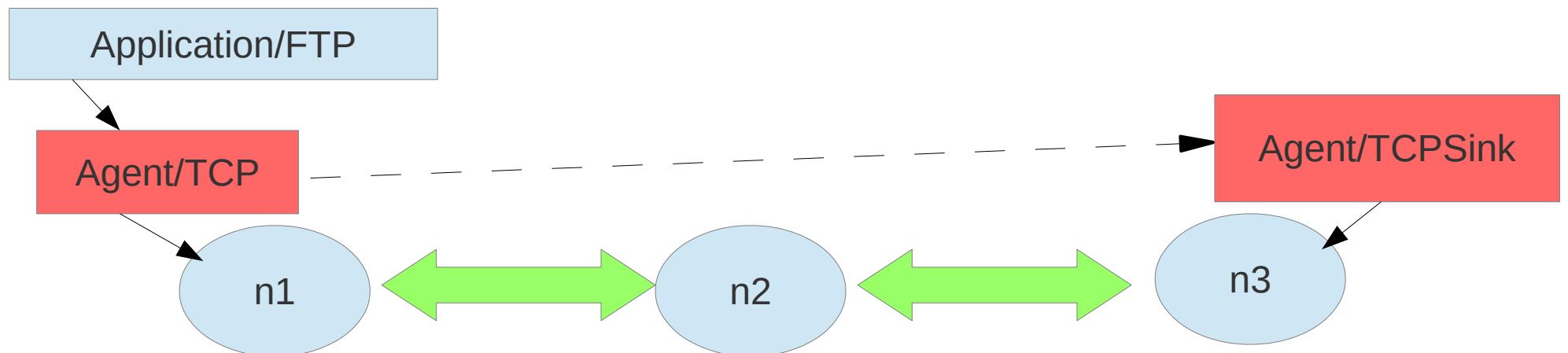
„Podczepianie” Agenta (prot 4 warstwy) oraz Aplikacji (FTP, CBR)

```
set tcp [new Agent/TCP]  
set sink [new Agent/TCPSink]
```

```
$ns attach-agent $n1 $tcp  
$ns attach-agent $n3 $sink  
# podczepiamy agentów pod węzły
```

```
$ns connect $tcp $sink  
# tworzymy połączenie
```

```
set ftp [new Application/FTP]  
$ftp attach-agent $tcp  
# instalujemy aplikację FTP nad TCP
```



Definiowanie początkowych zdarzeń:

```
$ns at 0.1 {$ftp start}  
$ns at 7.9 {$ftp stop}  
$ns at 8.0 {$ns halt}  
# początkowe zdarzenia; następne wynikają z symulacji...
```

```
$ns run  
$ns flush-trace  
puts "...koniec"  
# uruchomienie symulacji + napisy końcowe
```

Okazuje się, że NS-2 jest „sterowany zdarzeniami” a nie zegarem...
początkowe zdarzenia powodują lawinę zdarzeń...



Fig. 1.3 Clock advancement in a time-driven simulation

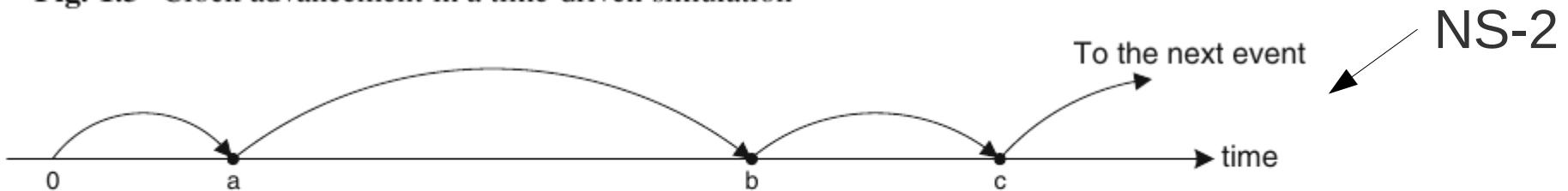


Fig. 1.4 Clock advancement in an event-driven simulation

Ogólne uwagi o NS-2:

1. skrypty definiujące siec i ruch w sieci programuje się w j. OTcl (rozszerz. OOP Tcl); programy ns, nstk, nse to po prostu interpretery Tcl z dodanymi klasami NS-2
2. NS-2 jest rozszerzalny - można dodawać nowe protokoły, robi się to w j. OTcl i C++; techn. "*split-objects*", każdy obiekt ma 2 połówki: w OTcl i w C++; dlaczego używa się 2 języków? odp: ???
3. NS-2 zawiera bardzo wiele protokołów różnych warstw (np. kilka(naście) odmian TCP)
4. j. OTcl (tak jak Tcl) posiada introspekcję,
dlatego czasami warto uruchomić konsolę: ./nstk konsola2c.tcl
możemy wtedy interaktywnie "zajrzeć" do obiektów, wyświetlić ich metody, zmienne...
możemy wyświetlić klasy, np. "info comm Agent/Tcp/*"
wyświetli wszystkie wersje prot. TCP w NS-2 ...
jednak symulacje wygodniej prowadzić wsadowo, czyli bez konsoli...
Pokazać przykłady ns10.tcl i ns12.tcl

5. Agenci vs połączenia/ powiązania

Każde połączenie lub powiązanie ma **osobną** parę agentów !!!
Nie utożsamiać gniazdek itp. z pojęciami NS-2 ...

Format pliku log/trace:

Każde zdarzenie w osobnej linii w poniższym formacie,

Typy zdarzeń:

1. (+) wstawia się pakiet do kolejki połączenia
2. (-) wyciąga się pakiet z tej kolejki, i jest on przesyłany fizycznie przez połączenie
3. (r) pakiet wychodzi drugim końcem połączenia
4. (d) jest też możliwe ze pakiet został porzucony,
np. z powodu przepełnienia kolejki

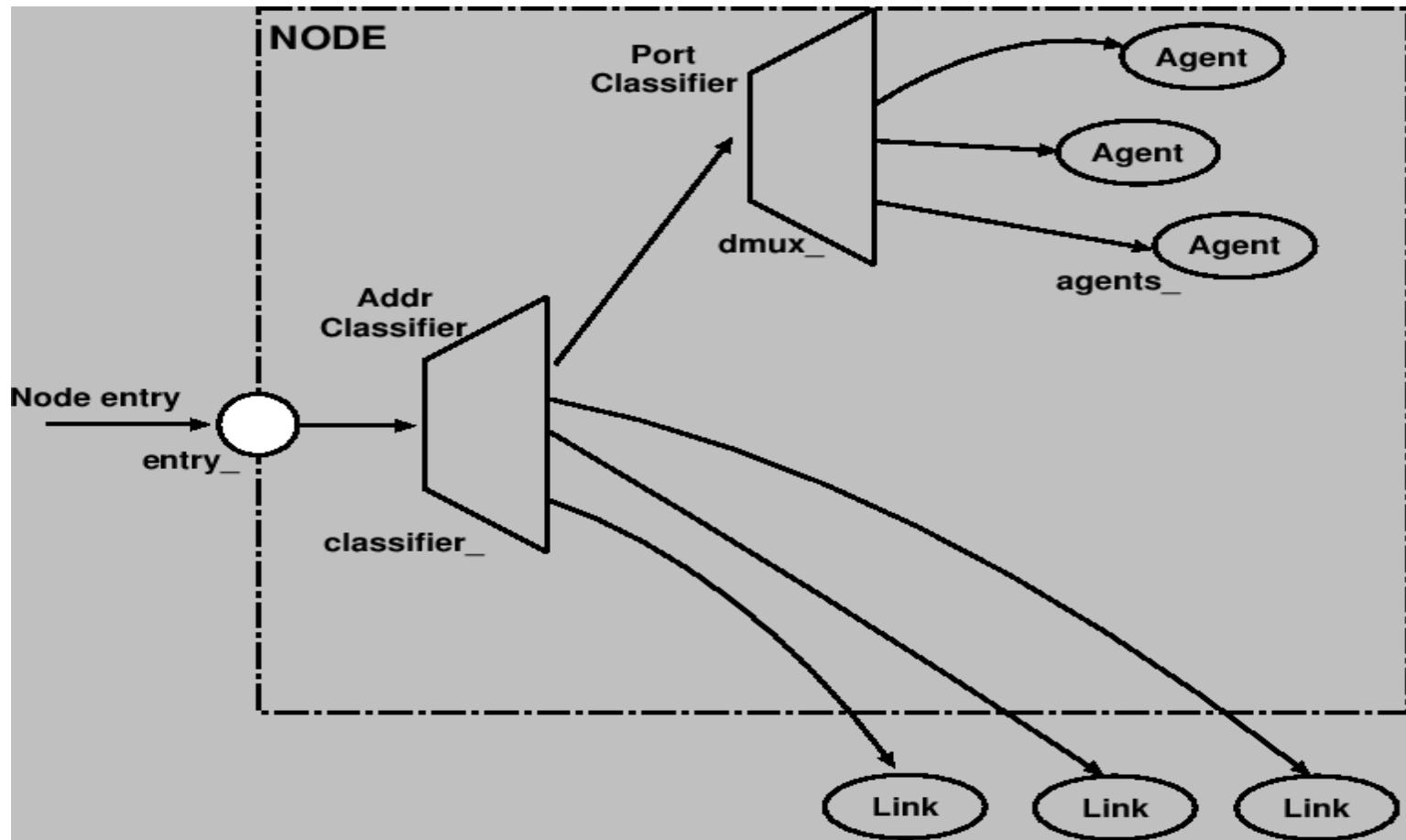
Event	Time	From node	To node	pkt type	Pkt size	Flags	Fid	Src addr	Dst addr	Seq num	Pkt id
-------	------	-----------	---------	----------	----------	-------	-----	----------	----------	---------	--------

Figure 2.5: Fields appearing in a trace

```
# przykład pliku trace (.tr)
+ 0.1 0 1 tcp 40 ----- 0 0.0 2.0 0 0
- 0.1 0 1 tcp 40 ----- 0 0.0 2.0 0 0
r 0.150107 0 1 tcp 40 ----- 0 0.0 2.0 0 0
+ 0.150107 1 2 tcp 40 ----- 0 0.0 2.0 0 0
- 0.150107 1 2 tcp 40 ----- 0 0.0 2.0 0 0
r 0.200213 1 2 tcp 40 ----- 0 0.0 2.0 0 0
+ 0.200213 2 1 ack 40 ----- 0 2.0 0.0 0 1
```

Wnętrzności NS-2 (jeśli chcemy dodać nowe protokoły...)

Klasa Node:

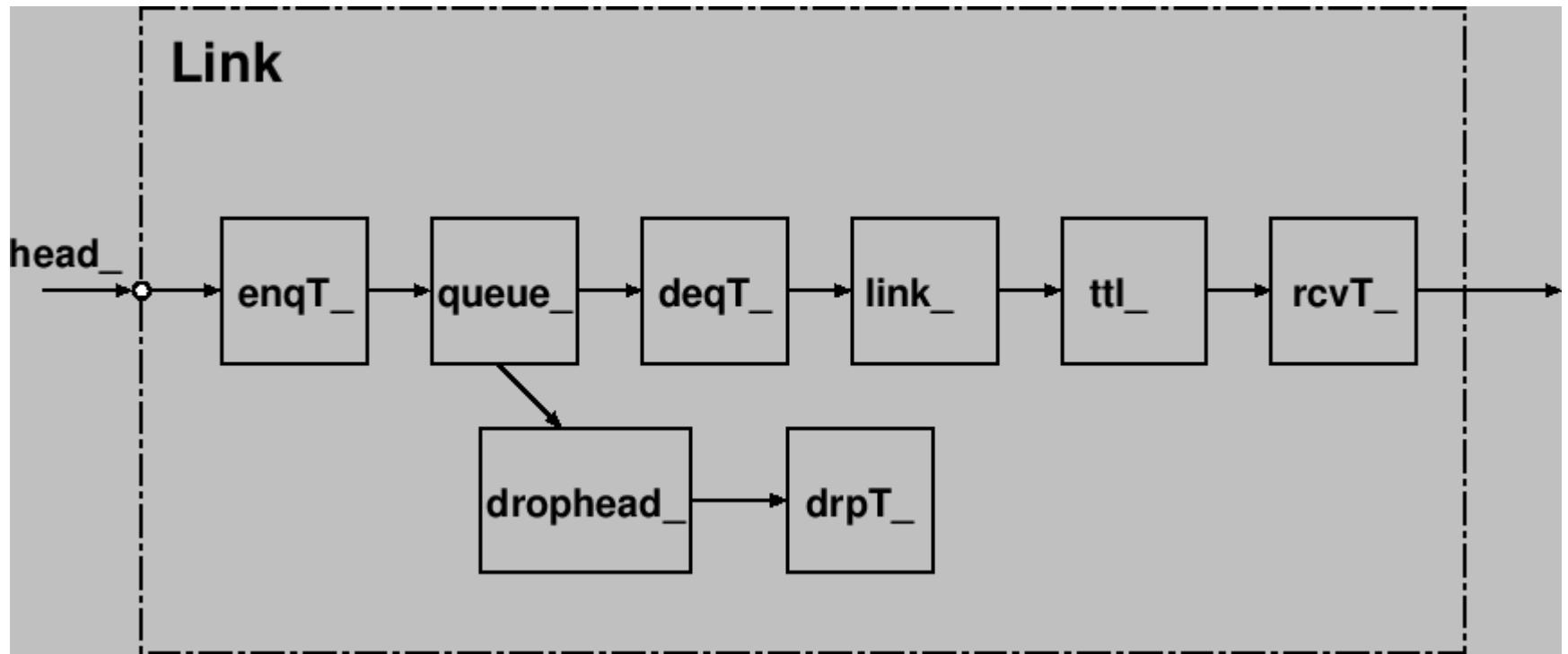


Classifier_, dmux_, i inne to zm. ob. klasy Node

Można je zobaczyć za pomocą „introspekcji”: ob info vars

Podobnie można zobaczyć metody: ob info instprocs

Klasa SimpleLink



Zm. ???T_ prowadzą do obiektów „trace”

Zm. queue_ to obiekt reprezentujący kolejkę (qdisc)

Zm. link_ to obiekt opóźniający pakiet (działanie zależy od przepust i opóźnienia)

Itd....

Inne rzeczy w NS-2:

- sieci lokalne (eth)
- sieci bezprzewodowe/ mobilne
- sieci satelitarne
- ???

Sieci komórkowe (GSM...)

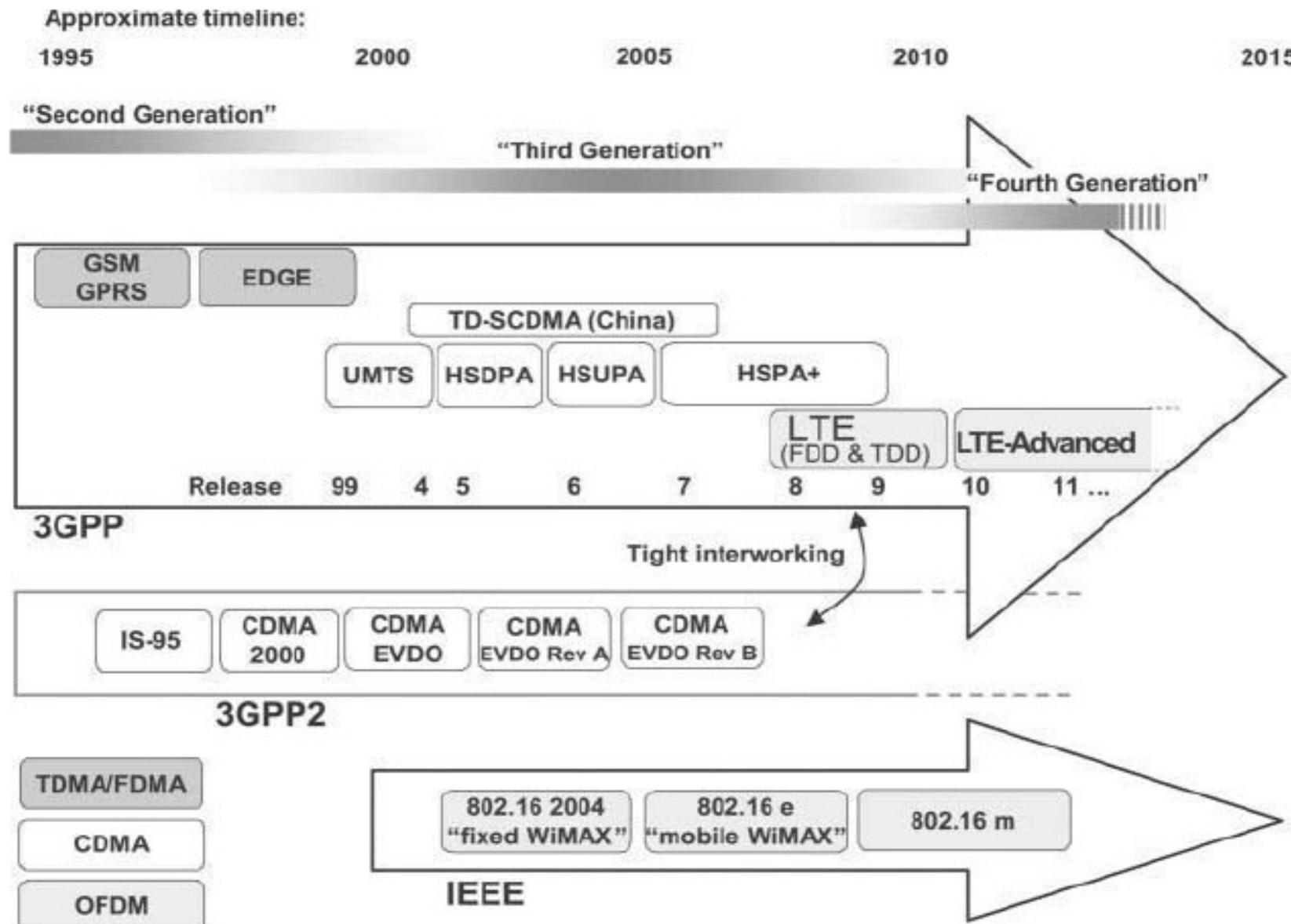


Figure 1.1: Approximate timeline of the mobile communications standards landscape.

Generacje sieci komórkowych:

1G – analogowe

2G – cyfrowe, GSM, głównie głos, przełączanie obwodów (??),

Potem dodano pakiety: GPRS/EDGE, kilka kb/s, ten sam sprzęt!

Metoda dostępu do łącza: TDMA + FDMA

(Kanały FDM dzielone na podkanały przy pomocy TDM)

3G – UMTS, nowy sprzęt !!, dalszy roz: HSPA, miał na to wpływ WiMAX

Metoda dostępu do łącza: CDMA (!!)

Przepustowość: 144kbs (samochód), 384kbs (w miejscu),

2Mbs (zamknięta przestrzeń)

4G – LTE, LTE-advanced, wyłącznie przełączanie pakietów,

Metoda dostępu do łącza: OFDMA, MIMO (wiele anten),

Setki Mbs... ta sama technika co wifi 802.11n !!

Różnica Wifi vs sieci komórkowe:

W wifi mamy „dostęp losowy” do łącza (CSMA/CA)

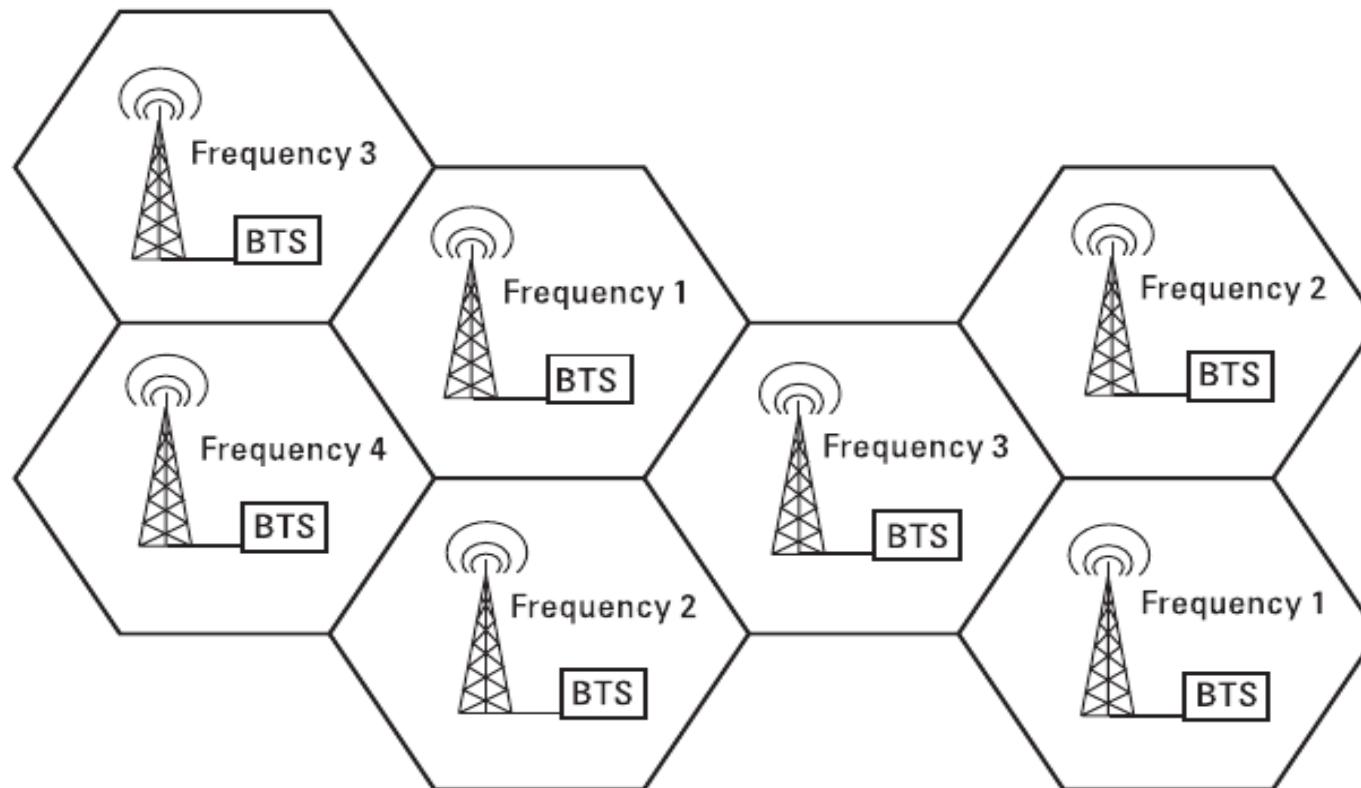
W GSM raczej współdzielenie łącza (oraz CDMA ?)

W GSM od początku obsługa mobilności użytkowników !!

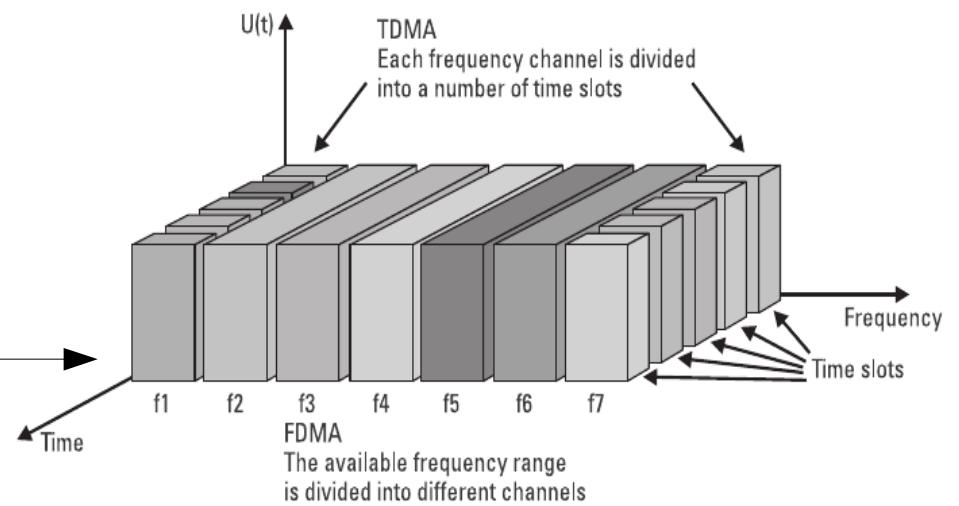
jak jest z mobilnością w Wifi ? co to jest mobilność ?

czy wireless zawsze oznacza mobilność ??

Struktura sieci GSM



Mamy komórki (przyp. BSS wifi),
Sąsiednie komórki muszą
używać innych częstotliwości
(SDMA = Space Div Multip Acc),
BTS = Base Transceiver Station
FDMA + TDMA



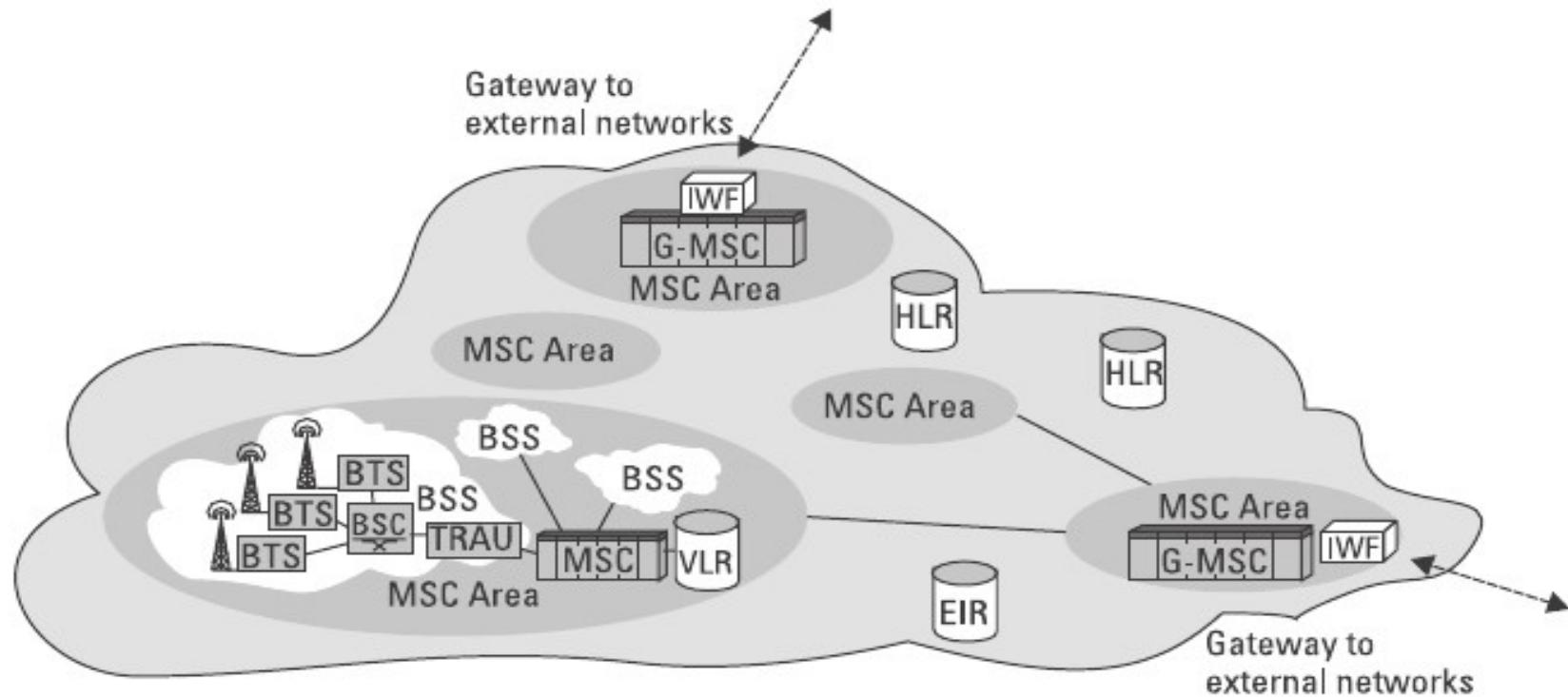


Figure 1.1 GSM network architecture

BSS = Base Station Subsystem

BTS = Base Transceiver Station

BSC = Base Station Controller

MSC = Mobile Switching Center

Centrala przełączania mobilnego

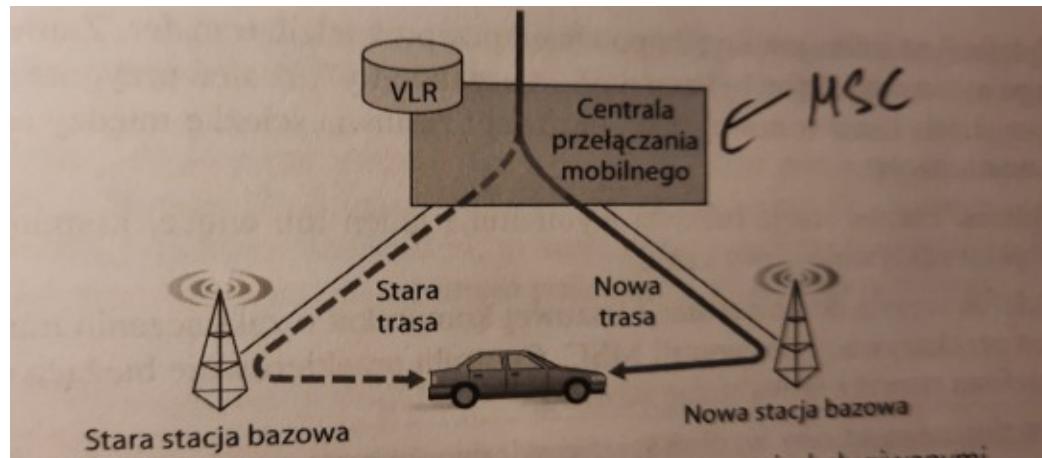
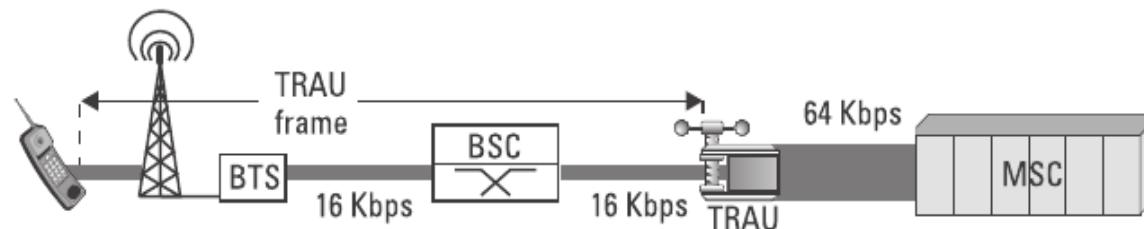
Chodzi o przełączanie między komórkami

VLR = Visitor Location Register

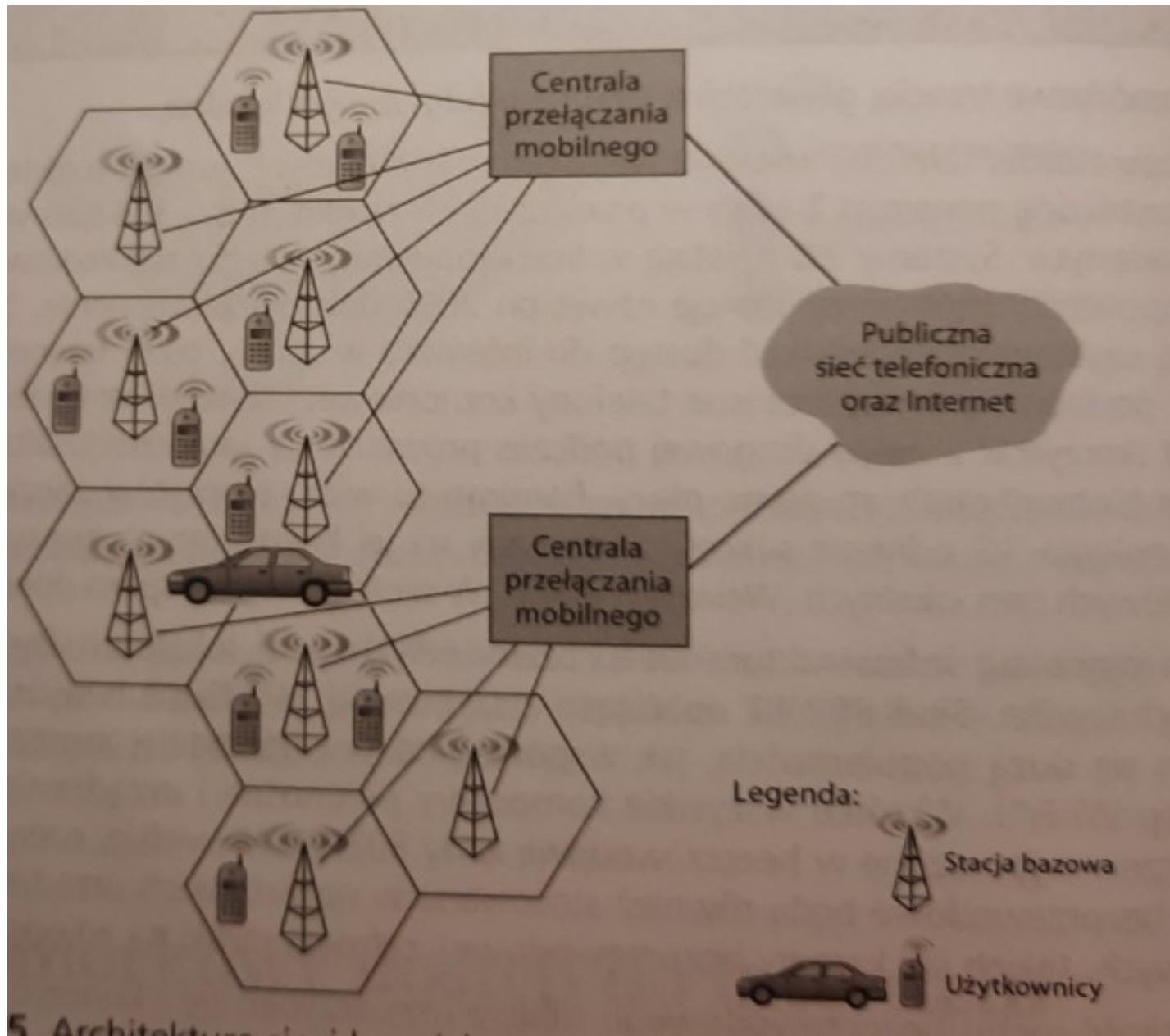
Przy MSC, dane o wizytujących użytkownikach

HLR = Home Location Register

Przy MSC, dane o domowych użytkownikach



Uproszczona wersja komórek GSM:



mobilność

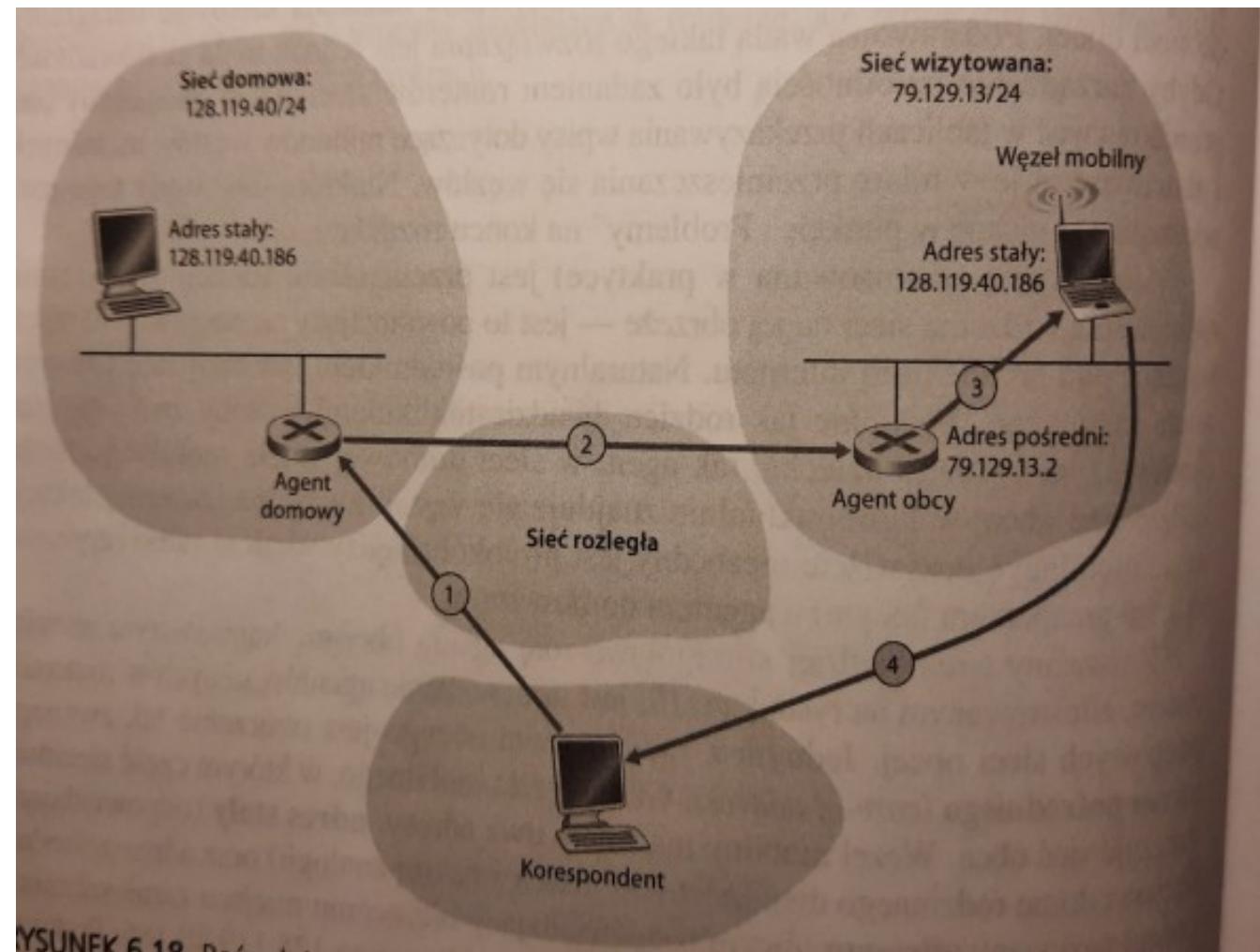
Mobilność to możliwość podłączania komputera do różnych sieci z zachowaniem tego samego adresu ip !! (niepoprawnego ?!?!)
Może dotyczyć sieci bezprzewodowych jak i przewodowych...
Obsługa mobilności była od zawsze w sieci telefonicznej GSM
ten sam nr telefonu !! jest ona dość podobna do „mobile ip” ...

Mobile IP, RFC 3220

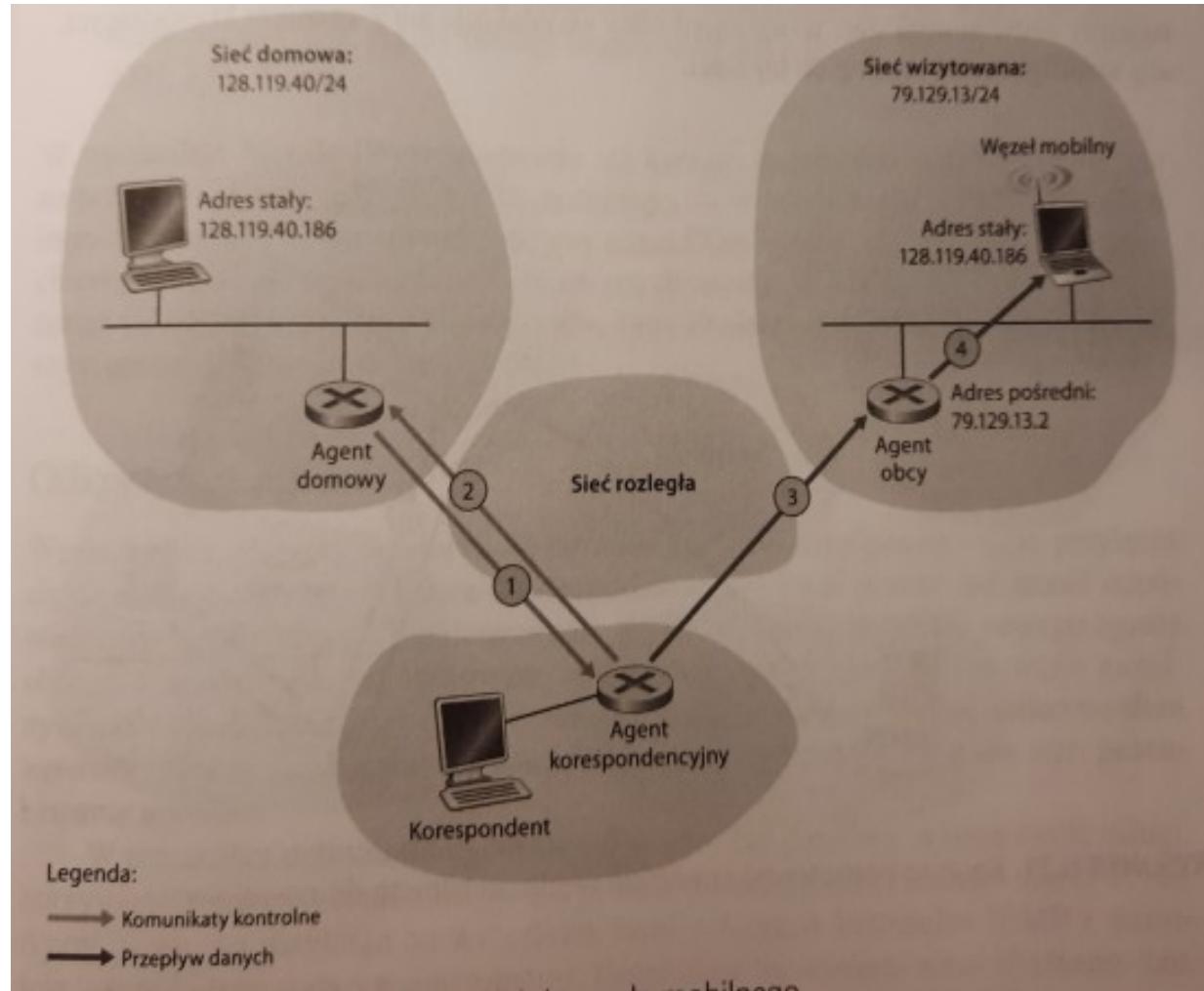
Pojęcia:

- Agent domowy (HA)
- Węzeł mobilny (MN)
- Agent obcy (FA)
- Foreign Agent
- Adres pośredni (COA)
- Care-Of Address

Ogłoszanie FA w sieci wizytowanej: przy pomocy broadcast ip kom ICMP, type=9 i 10 (albo FA się ogłasza, albo MN go szuka)



Mobile IP/ Inna opcja: z agentem korespondenta...



Mobile IP/ Inna opcja: z agentem korespondenta... C.D.

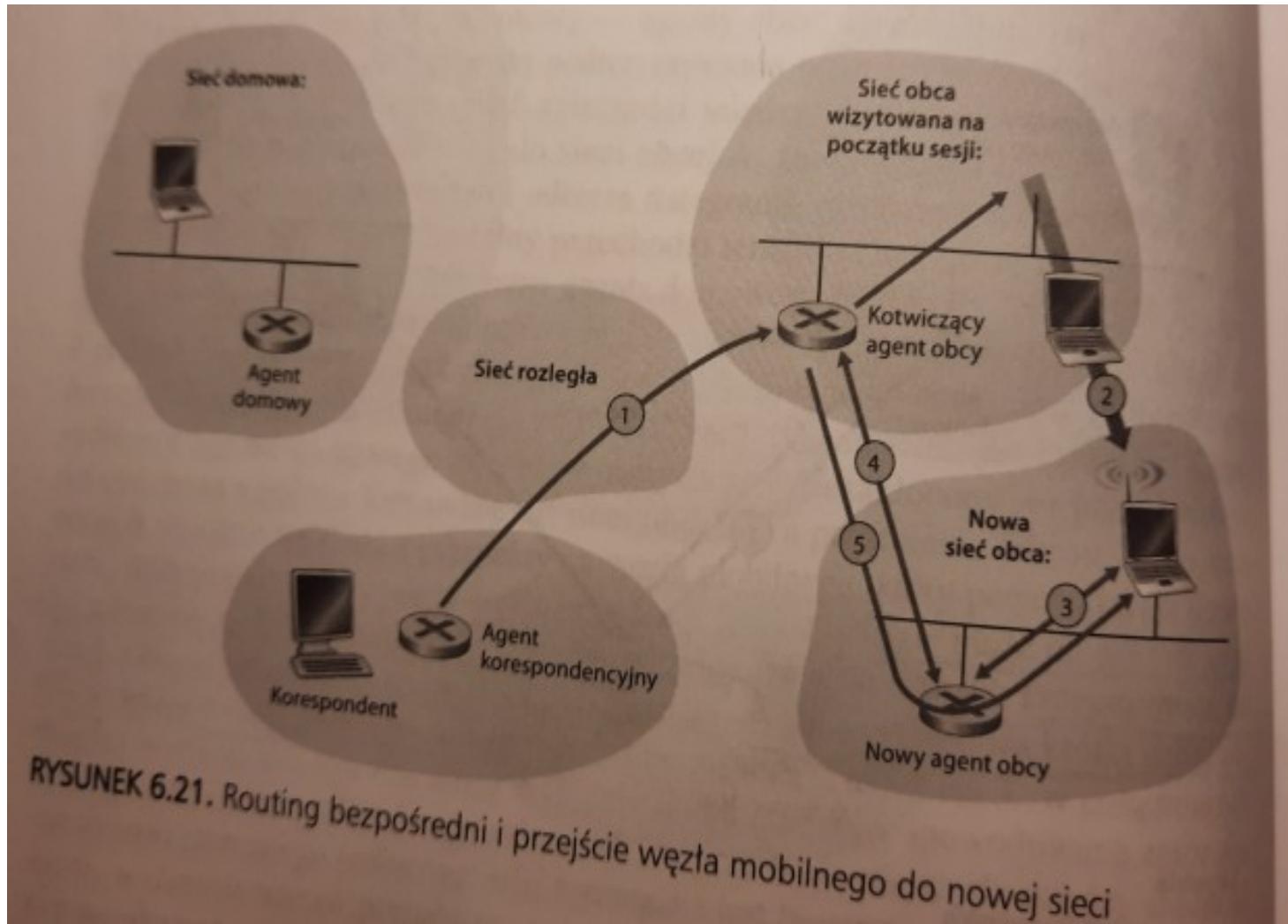
zmiana sieci obcej przez MN...

mamy pośrednictwo „agenta kotwiczącego”

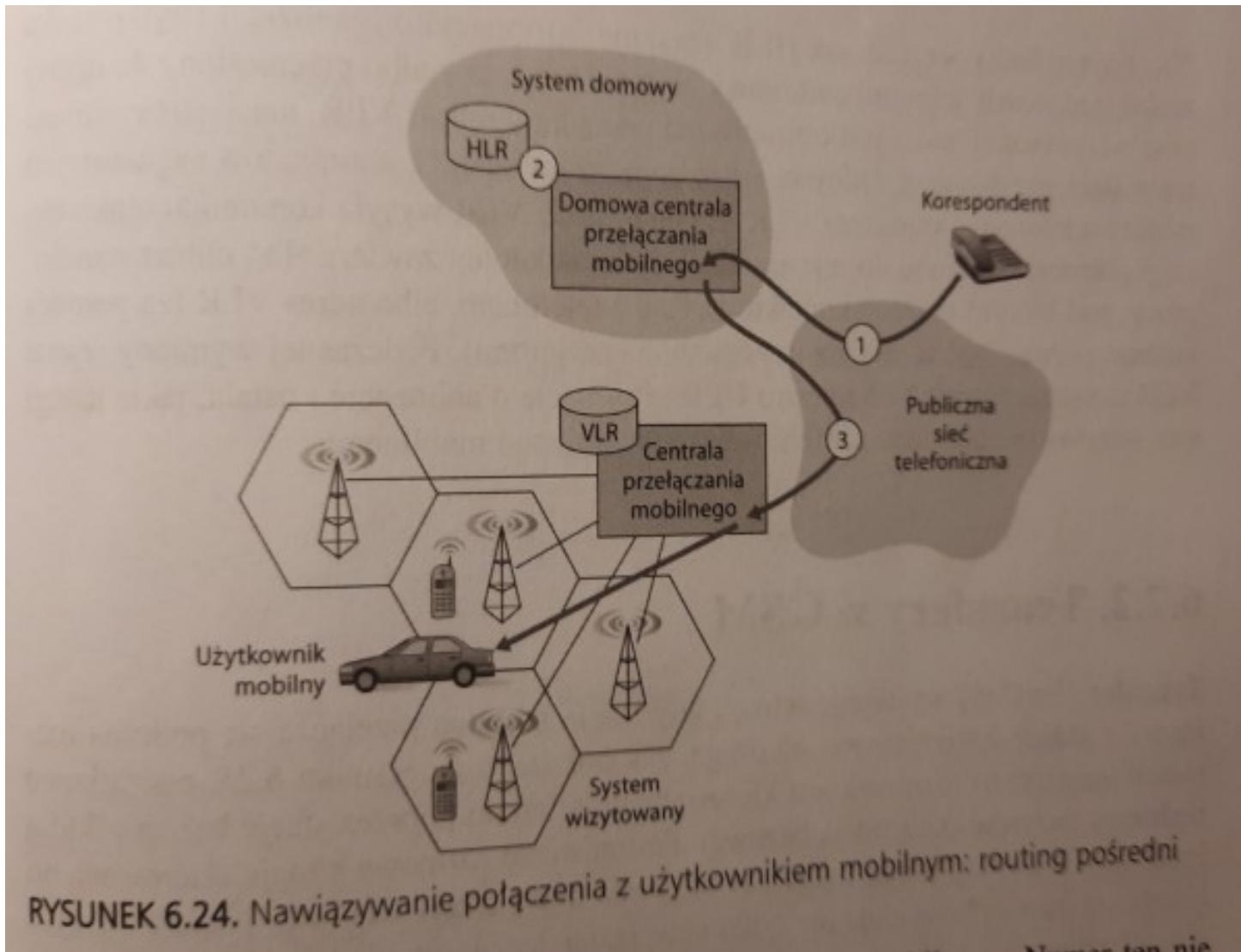
jest to bardzo podobne do zjawisk przy zmianie MSC w GSM...

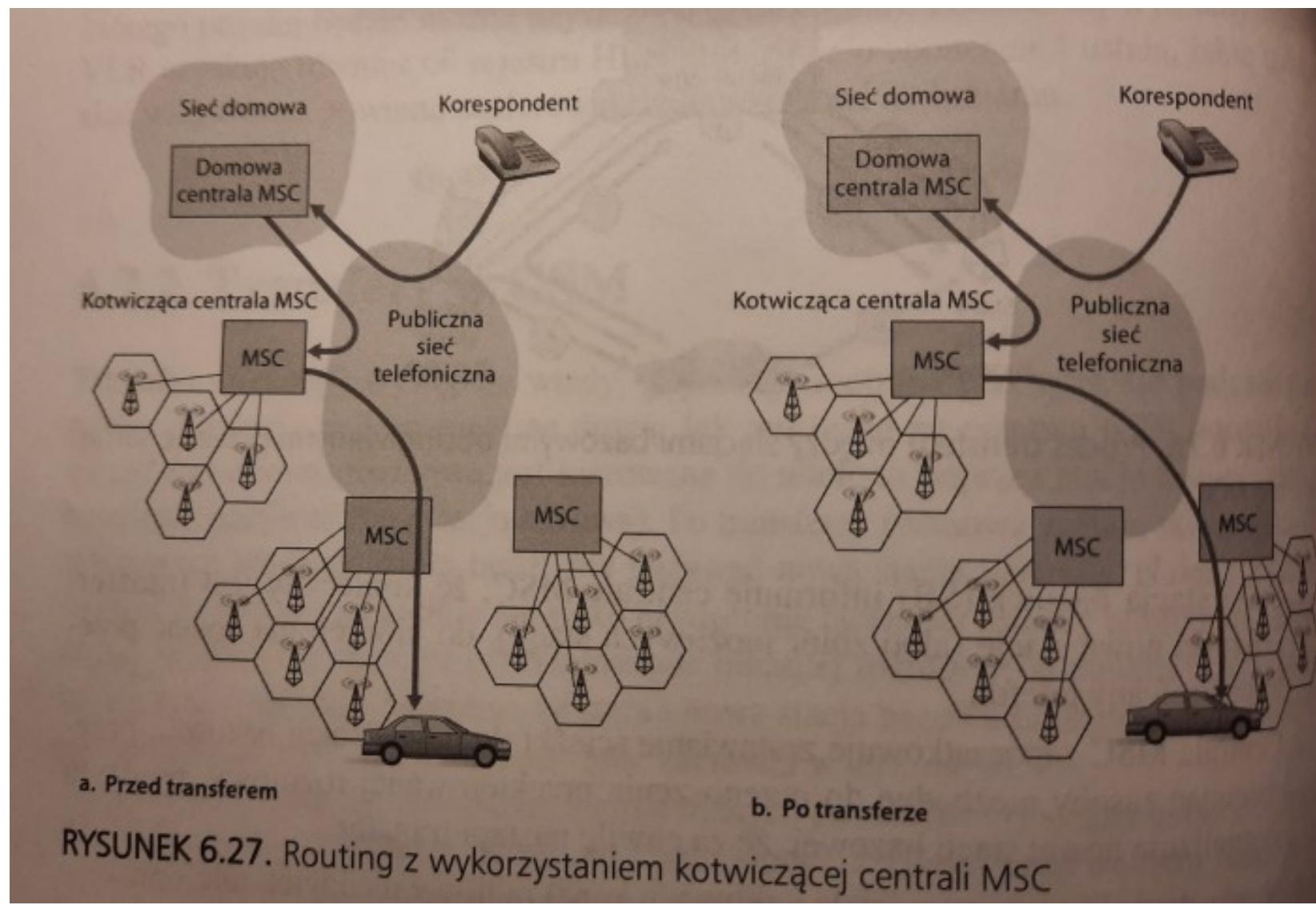
przez pewien czas komunikacja koresp-MN przechodzi

przez 3 pośredników !!



Mobilność w GSM...

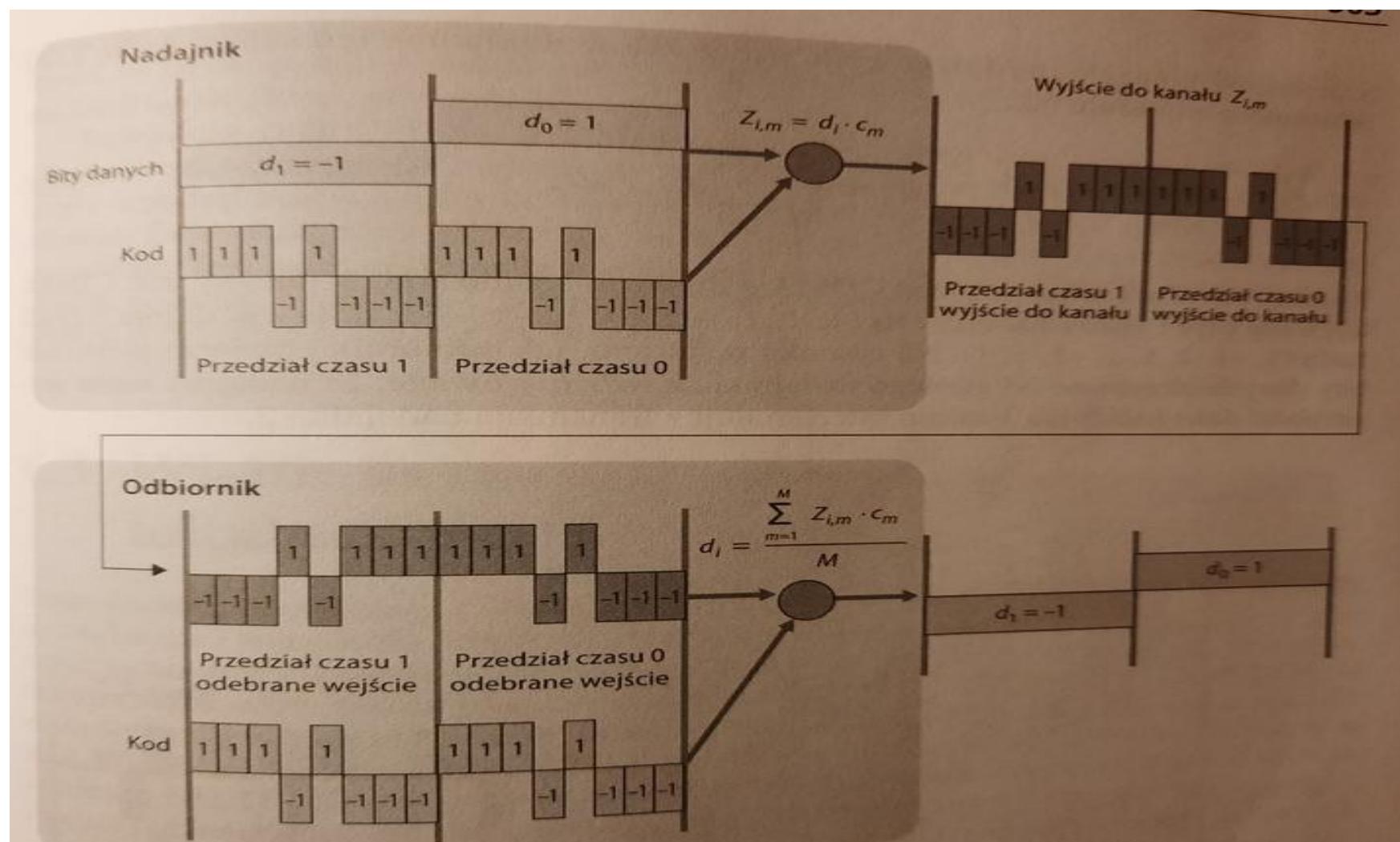




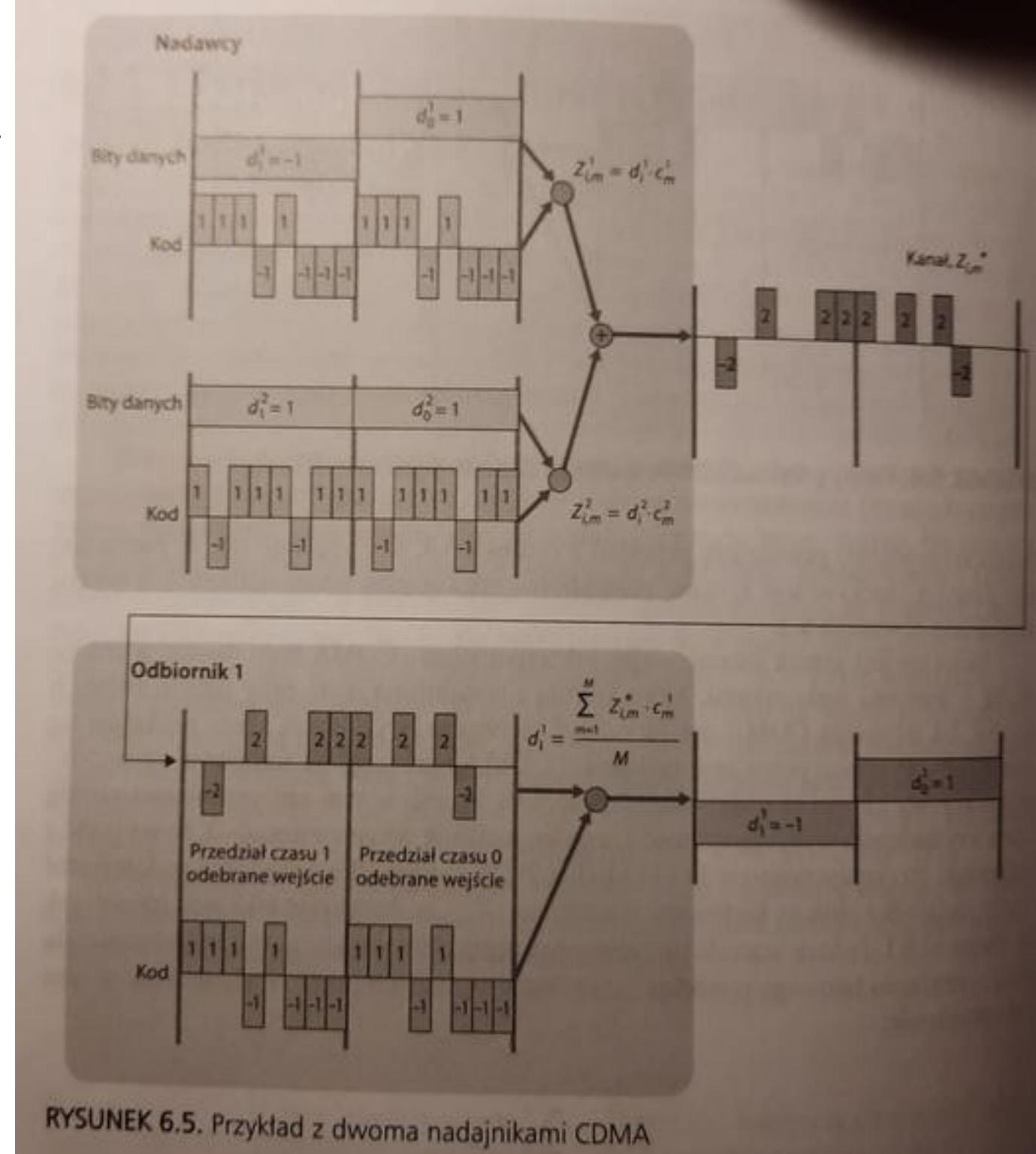
RYSUNEK 6.27. Routing z wykorzystaniem kotwiczącej centrali MSC

CDMA

Jest to metoda dostępu do łącza przypo- minająca DSSS;
Jednemu bitowi przypisuje się ciąg „małych” bitów...
Dominowała w epoce 3G w sieciach komórkowych...
Każdy użytkownik ma własny kod CDMA...



Wyodrębnianie danych od jednego użytkownika z sumy danych od wielu



Ciekawostka...

Jak przesyłać dane (ciąg bitów),
„szerokopasmowo”,
przy pomocy dźwięków...
innymi słowy: budujemy własną „warstwę fizyczną”

Pojęcia:

Dźwięk = suma tonów/składowych harmonicznych, „sinusy”
Transformata Fouriera, FFT, DFT,
Opis dźwięku za pomocą ciągu liczb zespolonych...
Moduł liczby zesp – amplituda,
Argument – przesunięcie fazowe,
Pozycja w ciągu - częstotliwość tonu,
(dziedzina która się tym zajmuje to „przetwarzanie sygnałów”)

Eksperyment numeryczny...

?!?!?!!?!