Sieci komputerowe

Tadeusz Kobus, Maciej Kokociński
Instytut Informatyki, Politechnika Poznańska

Organizacja – studia stacjonarne

Plan:

1. wstęp, model warstwowy,
2. adresacja IP,
3. podstawowe narzędzia i warstwa fizyczna,
4. warstwa łącza danych,
5. warstwa sieciowa (protokoły IP, ICMP, ARP, DHCP),
6. routing statyczny w Linuxie,
7. routing statyczny w Cisco,
8. routing dynamiczny w Cisco,
9. sieci VLAN,
10. warstwa transportowa,
11. iptables – zapora ogniowa,
12. iptables – NAT,
13. kolokwium,

14-15. tematy ekstra.

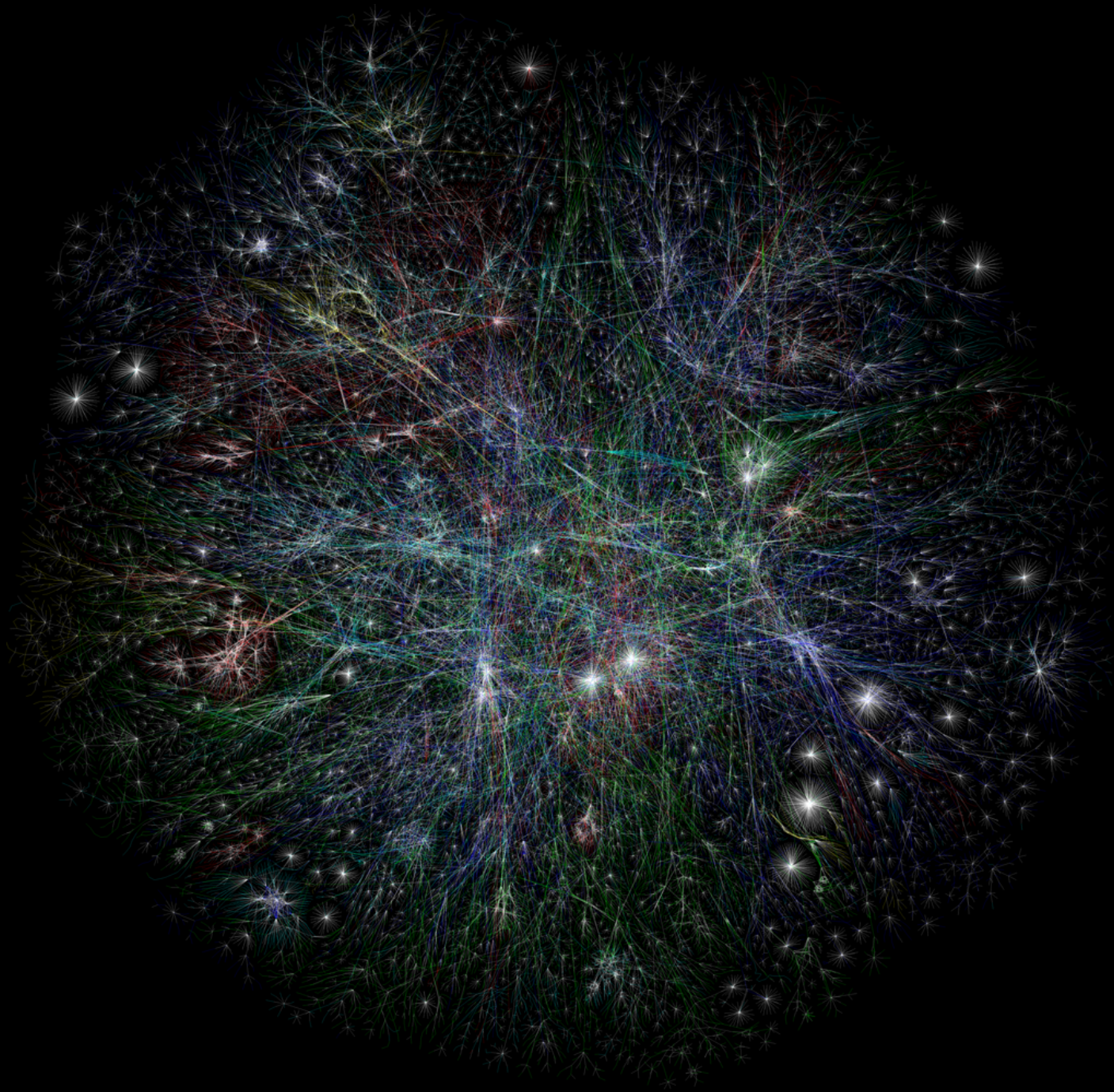
Zaliczenie:

- aktywność na zajęciach,
- kartkówki,
- kolokwium.

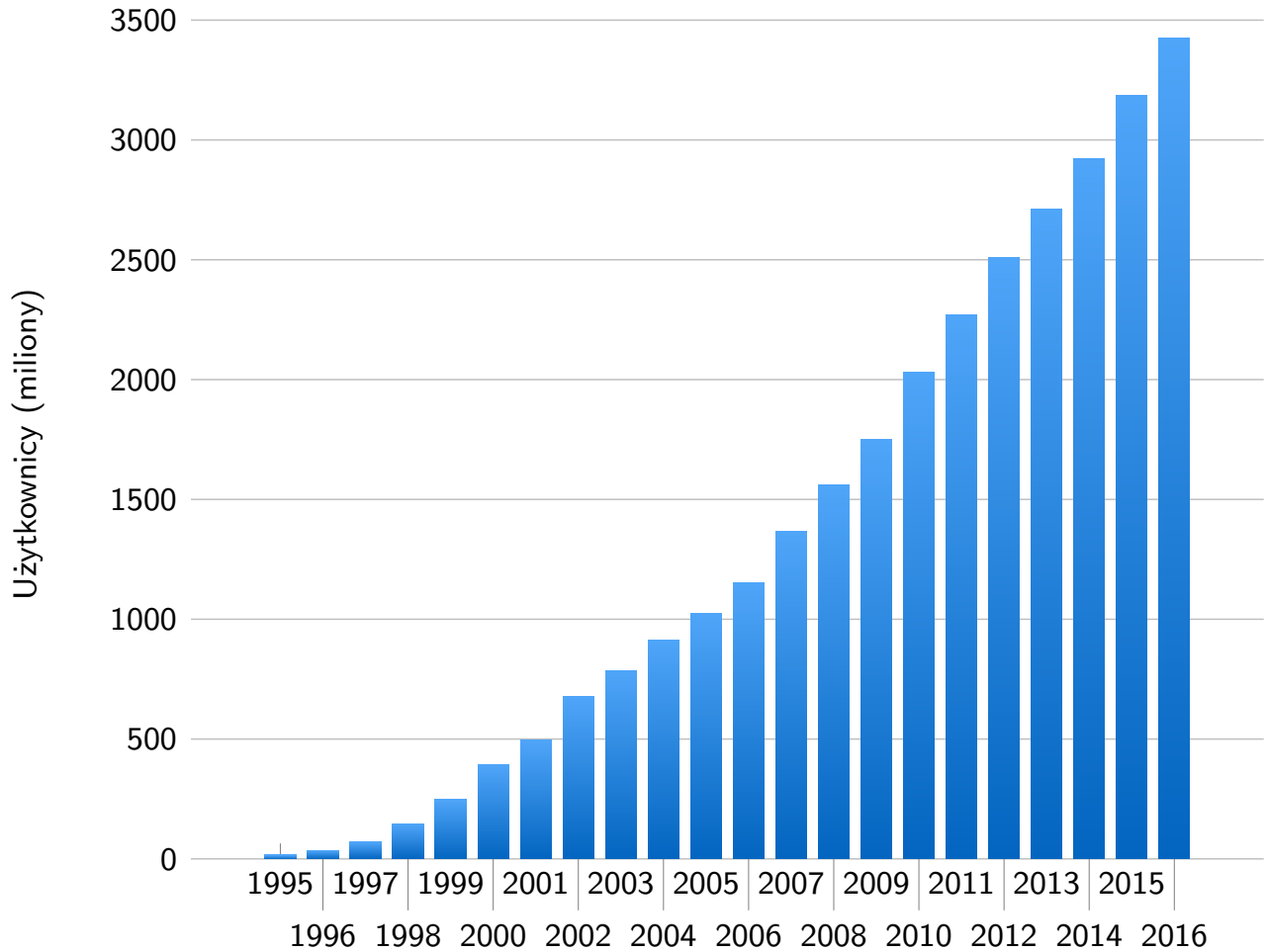
Materiały

http://www.cs.put.poznan.pl/tkobus/
students/sk1/sk.html

http://www.cs.put.poznan.pl/jkonczak/sk1
http://www.cs.put.poznan.pl/mkalewski/documents/sk.php
http://www.cs.put.poznan.pl/ksiek/sk/sk.html

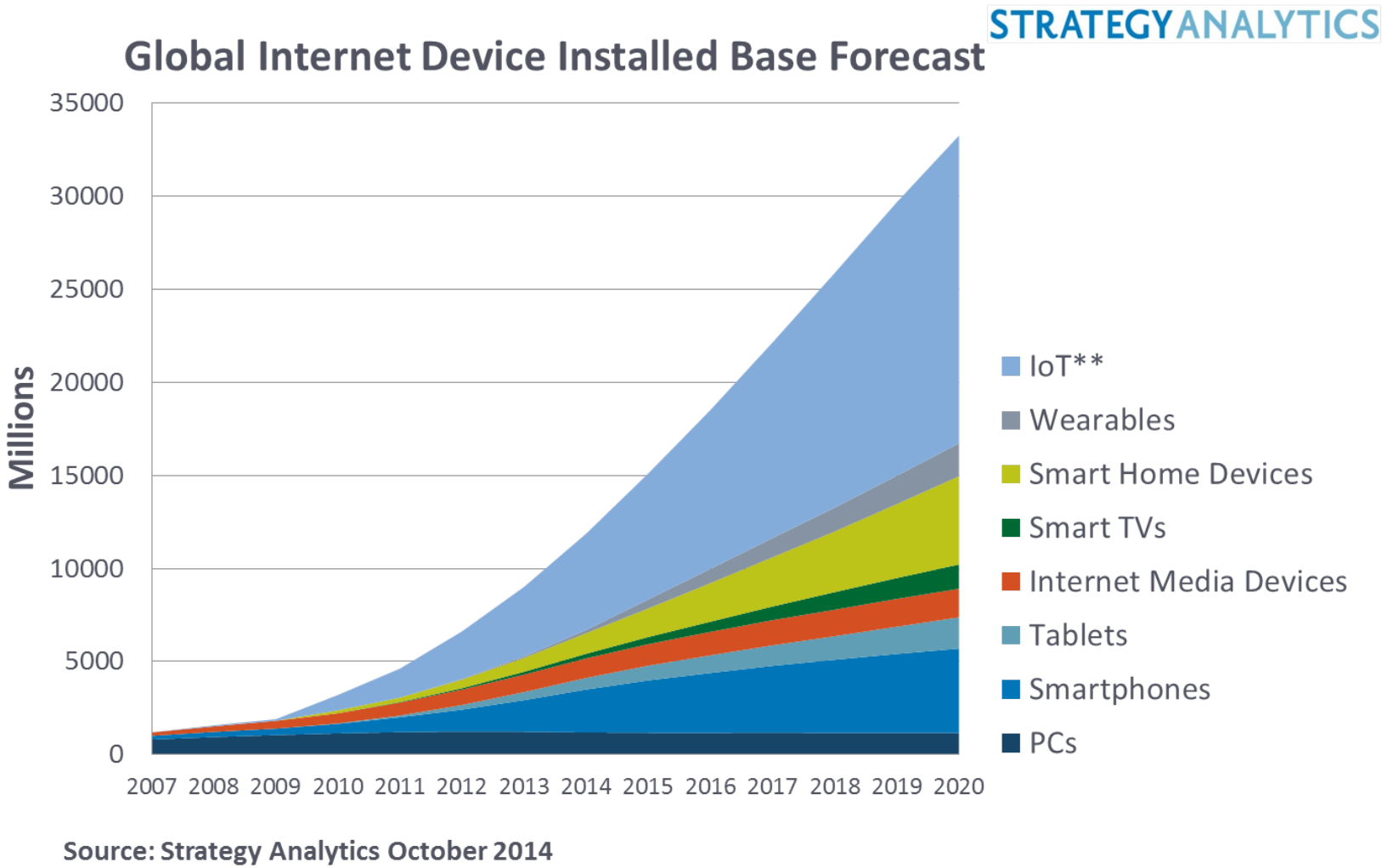


Liczba użytkowników Internetu*



*http://www.internetlivestats.com/internet-users/

Liczba urządzeń podłączonych do Internetu

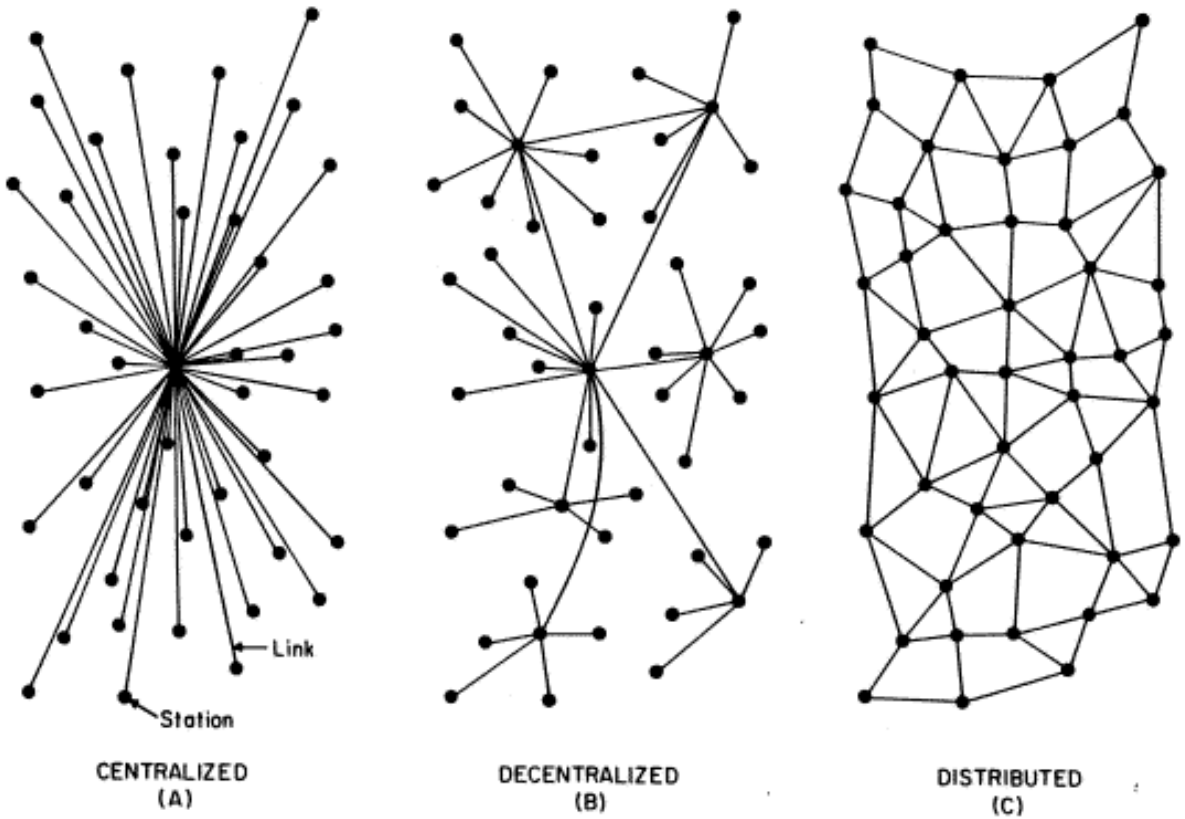


Internet – statystyki 2014

2.9 mld	liczba użytkowników
2.2 mld	liczba użytkowników mobilnych
0.97 mld	liczba witryn (75% to witryny nieaktywne)
4.0 mld	liczba kont mailowych
193 mld	liczba maili wymienianych codziennie (spam: ok. 43% ruchu między serwerami pocztowymi)
3.61 mld	liczba kont społecznościowych
1.44 mld	liczba aktywnych kont na Facebooku
12 mld	liczba wiadomości dziennie przesyłanych na Facebooku
16 lat	łączna długość filmów wrzucanych codziennie do serwisu Youtube

Internet – krótka historia (1)

1969 DARPA finansuje prace badawcze prowadzące do powstania sieci z komutacją pakietów (ang. *packet switching* zamiast *circuit switching*) → ARPANET



Internet – krótka historia (2)

1971 R. Tomlinson tworzy program do przesyłania poczty elektronicznej (adres: user@server)

1973 powstają sieci w W.Brytanii i Norwegii połączone z siecią ARPANET łączami satelitarnymi

1979 powstają pierwsze grupy dyskusyjne

1981 opracowanie protokołów komunikacyjnych TCP (Transmission Control Protocol) oraz IP (Internet Protocol)

1983 protokoły TCP/IP zostały przyjęte jako Standardy Wojskowe; implementacja TCP/IP w systemie operacyjnym UNIX BSD; ARPANET staje się siecią TCP/IP

Internet – krótka historia (3)

1983 z ARPANET wydzielona zostaję sieć MILNET (sieć Departamentu Obrony;) termin **Internet** służył do określenia obu tych sieci

1983 powstaje EARN (European Academic and Research Network)

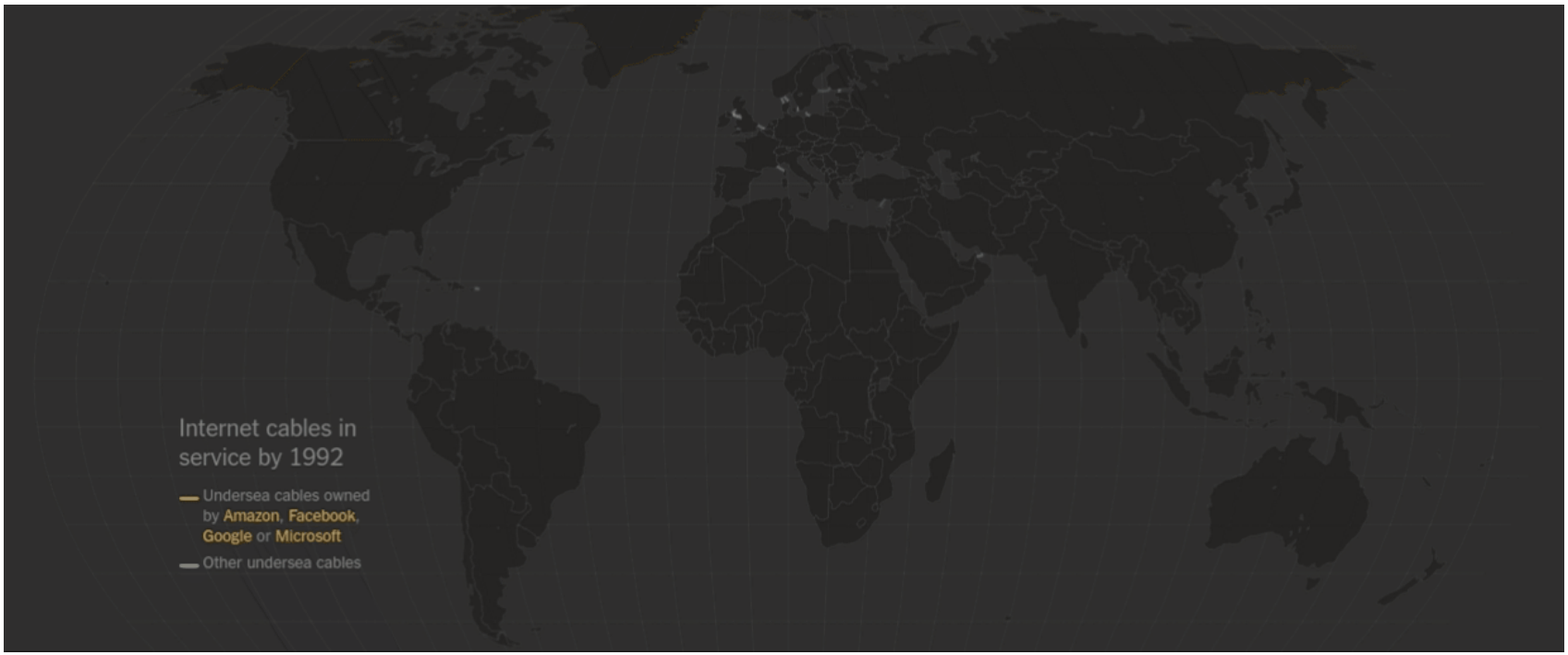
1984 wprowadzenie usługi DNS (Domain Name System); w sieci około 1000 serwerów

1986 powstaje NSFNET (National Science Foundation NET), amerykańska sieć szkieletowa o przepustowości 56 kb/s

1991 T. Berners-Lee tworzy HTML (Hyper-Text Markup Language), co daje początek WWW (World Wide Web)

1995 NFSNET przekształca się w sieć badawczą, Internet się komercjalizuje

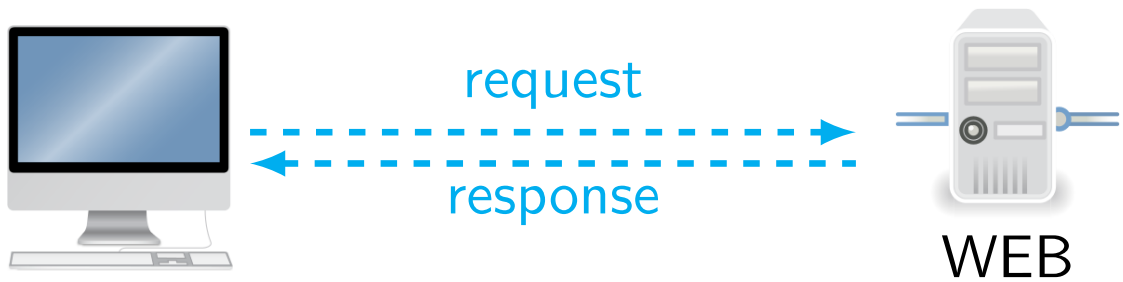
Podwodne połączenia światłowodowe



Jak działa Internet?

Komputery wymieniają komunikaty.

Strukturę komunikatów określa protokół: http, ftp, smtp, ipp, ...



Rodzaje sieci i połączeń (1)

Zasięg:

- Personal Area Network (PAN),
- Storage Area Network (SAN),
- Local Area Network (LAN),
- Campus Area Network (CAN),
- Metropolitan Area Network (MAN),
- Wide Area Network (WAN),
- Internet Area Network (IAN – Cloud).

Dostęp:

- intranet,
- ekstranet.

Rodzaje sieci i połączeń (2)

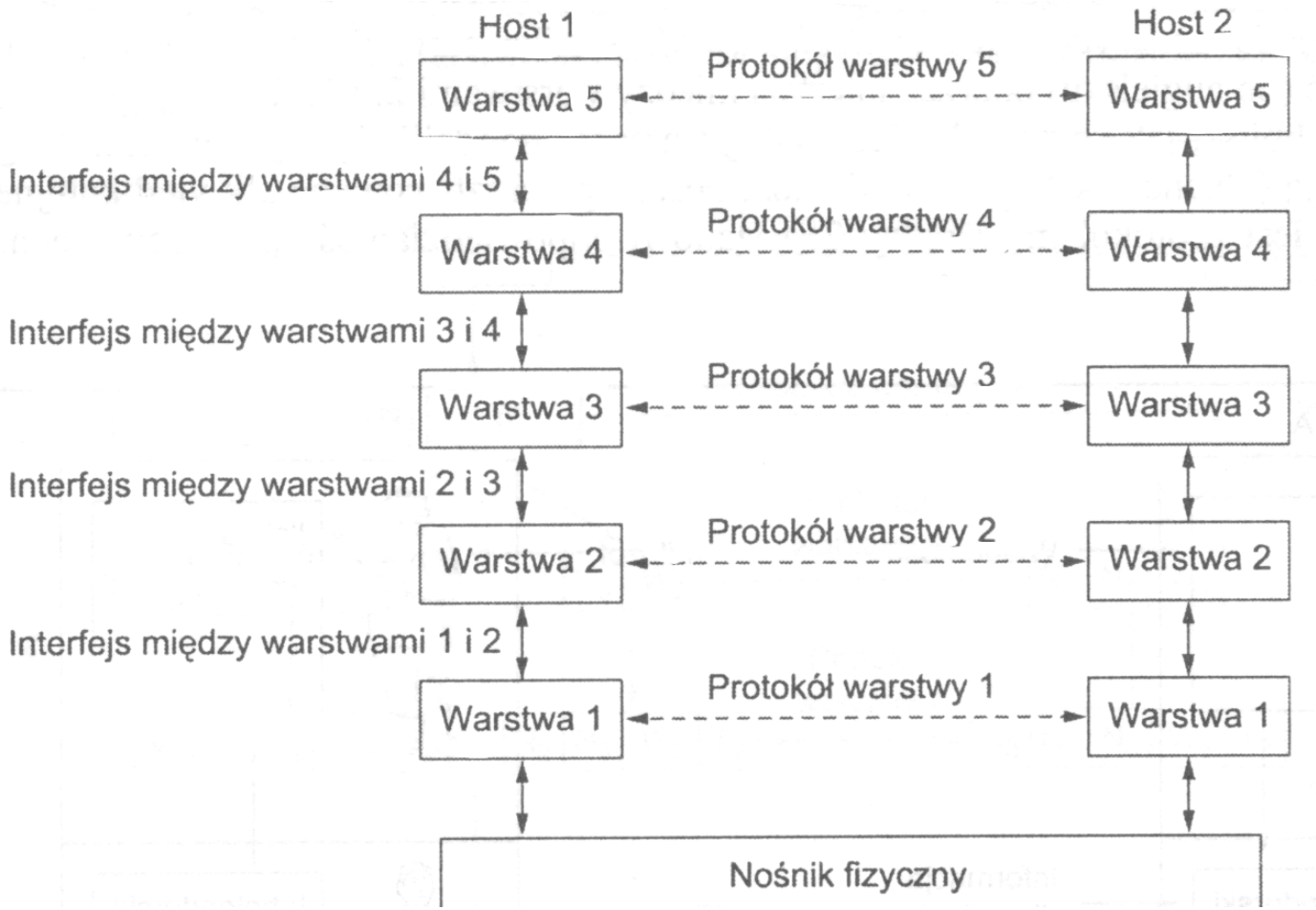
Topologia (fizyczna, logiczna):

- punkt-punkt (point-to-point),
- magistrala (bus),
- pierścień (ring),
- gwiazda (star),
- hierarchiczna, drzewiasta (tree),
- siatki (mesh).

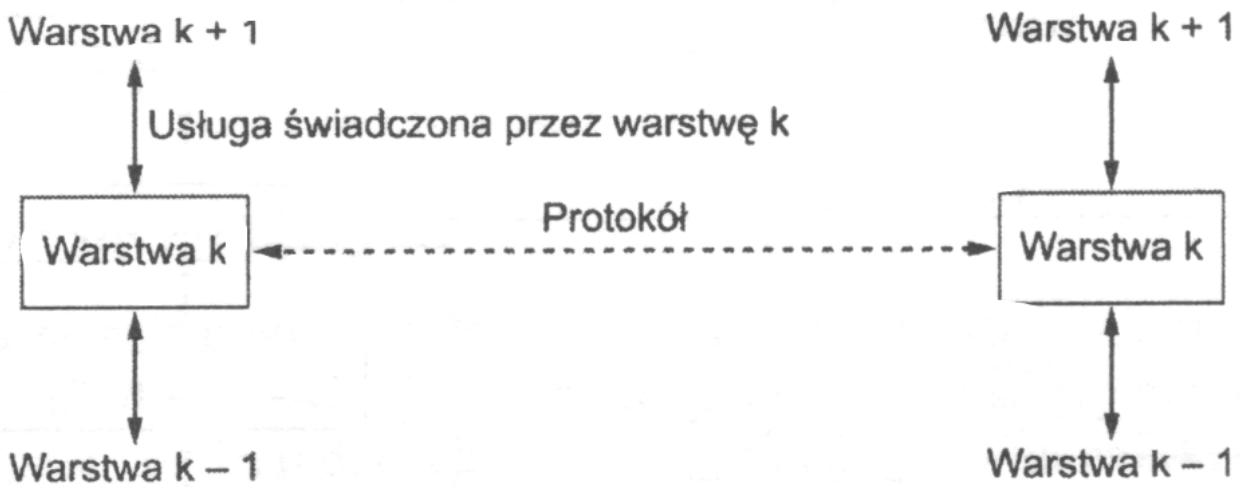
Liczba adresatów transmisji:

- unicast – jeden odbiorca,
- anycast – dowolny z wielu odbiorców,
- multicast – wielu odbiorców,
- broadcast – wszyscy możliwi odbiorcy.

Model warstwowy sieci

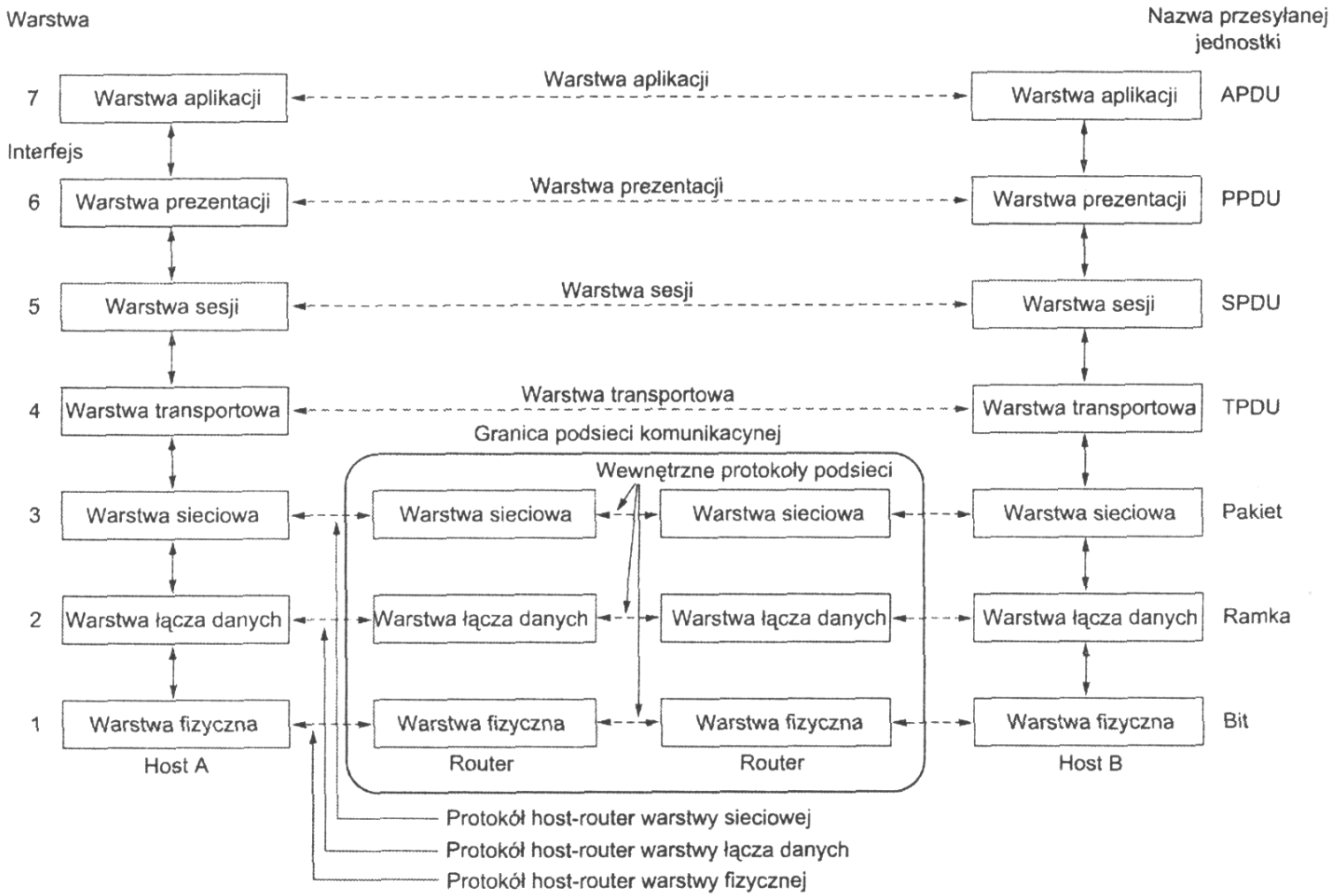


Związek między usługą i protokołem



Model odniesienia OSI

(Open System Interconnection)



Funkcje warstw modelu OSI (1)

Zastosowań (application layer)

oferuje usługi sieciowe użytkownikom lub programom,
np. protokołowi realizującemu usługę poczty elektronicznej
(nie dostarcza usług żadnej innej warstwie)

Prezentacji (presentation layer)

zapewnia przekazywanie danych (tekstowych, graficznych,
dźwiękowych) w odpowiednim (wspólnym) formacie,
dokonuje ich kompresji oraz ew. szyfrowania

Sesji (session layer)

ustanawia, zarządza i kończy połączeniami (sesjami)
pomiędzy współpracującymi aplikacjami, m.in. ustala sposób
wymiany danych (jednokierunkowy (*half-duplex*) lub
dwukierunkowy (*full-duplex*))

Funkcje warstw modelu OSI (2)

Transportowa (transport layer)
zapewnia bezbłędną komunikację pomiędzy komputerami w sieci (*host to host*), dzieli dane na fragmenty, kontroluje kolejność ich przesyłania, ustanawia wirtualne połączenia, utrzymuje je i likwiduje (TCP, UDP)

Sieciowa (network layer)
definiuje pakiety, ustala drogę transmisji danych i przekazuje dane pomiędzy węzłami sieci (IP, IPX, ICMP, ARP, DDP)

Funkcje warstw modelu OSI (3)

Łącza danych (data link layer)

zapewnia niezawodne dostarczanie danych przez znajdującą się poniżej fizyczną sieć (MAC/LLC, PPP, ATM, Frame Relay, HDLC, 802.1q, 802.3, 802.11a/b/g/n MAC/LLC)

Fizyczna (physical layer)

umożliwia przesyłanie poszczególnych bitów (ramek) przez dane fizyczne łącze, kontroluje przepływ bitów, powiadamia o błędach (RS232C, V.35, RJ45, 802.11 a/b/g/n PHY, 10BASE-T, 100BASE-TX, 1000BASE-T, T1, E1, SONET, SDH, DWDM)

Model TCP/IP

<table>
<tr><th colspan="2">model OSI</th><th colspan="2">model TCP/IP</th></tr>
<tr><td>warstwa aplikacji</td><td>(7)</td><td rowspan="3">(4)</td><td rowspan="3">warstwa aplikacji</td></tr>
<tr><td>warstwa prezentacji</td><td>(6)</td></tr>
<tr><td>warstwa sesji</td><td>(5)</td></tr>
<tr><td>warstwa transportowa</td><td>(4)</td><td>(3)</td><td>warstwa transportowa</td></tr>
<tr><td>warstwa sieciowa</td><td>(3)</td><td>(2)</td><td>warstwa internetowa</td></tr>
<tr><td>warstwa łącza danych</td><td>(2)</td><td rowspan="2">(1)</td><td rowspan="2">warstwa dostępu do sieci (host-sieć)
wg AT wielkie nic</td></tr>
<tr><td>warstwa fizyczna</td><td>(1)</td></tr>
</table>

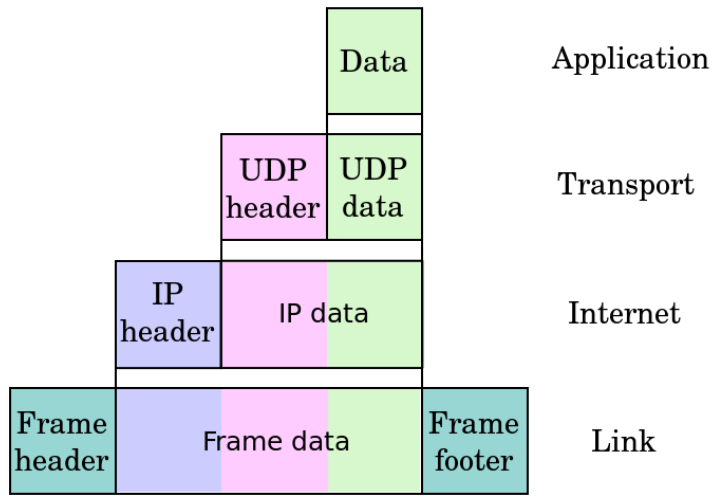
Modele warstwowe – założenia (1)

- Komunikacja równorzędna (*peer-to-peer*) – każda warstwa jednego hosta komunikuje się wyłącznie z tą samą warstwą u drugiego hosta.
- Warstwy tego samego poziomu wymieniają tzw. Protocol Data Units (PDUs).
- Maksymalny rozmiar PDU jest określony przez Maximum Transmission Unit (MTU).
- Wielkość MTU wpływa proporcjonalnie na opóźnienia transmisji i odwrotnie proporcjonalnie na narzut transmisji.

model TCP/IP	PDU
warstwa aplikacji	dane
warstwa transportowa	segmenty/datagramy
warstwa internetowa	pakiety
warstwa dostępu do sieci	ramki (i bity dla w. fizycznej w OSI)

Modele warstwowe – założenia (2)

- Wędrówce danych w dół/górę stosu warstw towarzyszy proces enkapsulacji/dekapsulacji.
- Urządzenie sieciowe *pracuje w warstwie x*, jeśli najwyższą warstwą, w ramach której przetwarza dane, to *x*.



Przed ćwiczeniami (1)

Interfejsy każdego z komputerów (kolejność wpięcia *do stołu*):

- `em1` – główny interfejs z adresem z DHCP
- `p4p1` – dolny *prawy*
- `p4p2` – dolny *lewy* (niepodpięty)
- `ttyS0` – złącze szeregowe (RS232)

Patch panele na zapleczu:

- `p4p1`
- `p4p2`
- `ttyS0`
- `em1`

Przed ćwiczeniami (2)

tablica

28	**27**	**26**	**25**
.130	.131	.132	.133
32	**31**	**30**	**29**
.134	.135	.136	.137
36	**35**	**34**	**33**
.138	.139	.140	.141
40	**39**	**38**	**37**
.142	.143	.144	.145

Numery `em1` na patch panelu

IP: `150.254.32.X`

mask: `255.255.255.192`

gateway: `150.254.32.191`

Adresacja IPv4

Adres IP komputera

- Ciąg 32 bitów (4 bajty).
- Dwie części (punkt podziału zależy od konwencji):
  - część sieciowa – adres sieci, w której znajduje się komputer (początkowa część adresu),
  - część komputerowa – adres komputera wewnątrz sieci (końcowa część adresu).
- Przykład: `11000000 01110000 01001100 00000011`
  - część sieciowa: `11000000 01110000` (2B)
  - część komputerowa: `01001100 00000011` (2B)
- Zapis dziesiętny (dotted decimal):
  - 4 jednobajtowe segmenty rozdzielone kropką,
  - każdy bajt zapisany jako liczba dziesiętna z przedziału 0-255 (00000000–11111111),
  - przykład: `192.56.76.3`.

Adres sieci i adres rozgłoszeniowy

Adres sieci: taki jak adres IP komputera, ale z wyzerowana częścią komputerową:

- przykład: `11000000 01110000 00000000 00000000` (część sieciowa = 2B),
- zapis dziesiętny: `192.58.0.0`.

Adres rozgłoszeniowy dla danej sieci – część komputerowa złożona z jedynek:

- przykład: `11000000 01110000 11111111 11111111` (część sieciowa = 2B),
- zapis dziesiętny: `192.58.255.255`.

Maska sieci

- Ciąg 32 bitów składający się z bloku jedynek i bloku zer, np. `11111111 11111111 11110000 00000000`.
- Określa punkt podziału adresu na część komputerową i sieciową.
- Część sieciową uzyskuje się poprzez wykonanie operacji AND na adresie i masce, np.:
  `11000000 01110000 01001100 00000011` (adres)
  `11111111 11111111 11110000 00000000` (maska)
  `11000000 01110000 01000000 00000000` (adres sieci)
- Zapis dziesiętny: `255.255.240.0`.
- Zapis skrócony: `/n`, np., `/20`.
- Przykład adresu IPv4 z maską: `192.56.76.3/16`.

Zakres adresów

- Adresy komputerów mieszczą się w przedziale wyznaczonym przez rozmiar części komputerowej adresów pomniejszonym o adresy specjalne.
- Przykład:
  sieć: 150.150.0.0/16

10010110 10010110 00000000 00000000	150.150.0.0	adres sieci
10010110 10010110 00000000 00000001	150.150.0.1	#1
10010110 10010110 00000000 00000010	150.150.0.2	#2
...	...	
10010110 10010110 11111111 11111110	150.150.255.254	#65534
10010110 10010110 11111111 11111111	150.150.255.255	ad. rozgłoszeniowy

  zakres adresów: 150.150.0.1 – 150.150.255.254
  liczba komputerów, które można zaadresować:
  $2^{16} - 2 = 65534$

Zadanie 1

Dany jest adres `100.0.100.50/28`.

1. Oblicz adres sieci dla powyższego adresu.
2. Wskaż adres rozgłoszeniowy sieci.
3. Ile komputerów można zaadresować w tej sieci?

Przydział adresów (1)

Dawniej – przydział klas adresów:

Klasa	Zakres adresów IP (pierwszy oktet)		Rozmiar sieci	Zastosowanie:
A	1–126	(00000001-01111110)	16 777 216	b. duże instytucje
B	128–191	(10000000-10111111)	65 534	duże instytucje
C	192–223	(11000000-11011111)	254	małe instytucje
D	224–239	(11100000-11101111)	–	multicast
E	240–255	(11111000-11111111)	–	eksperymentalne

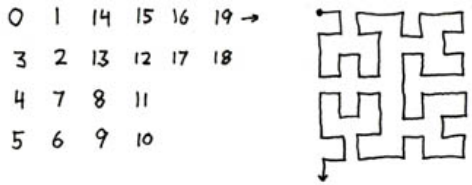
Od 1993 – Classless Inter-Domain Routing (CIDR) – liczba adresów w bloku CIDR jest dowolną (w ustalonym zakresie) potęgą liczby 2.

Prefiks CIDR	Maska	Rozmiar sieci
/32	255.255.255.255	1
/31	255.255.255.254	2
/30	255.255.255.252	4
...	...	...
/13	255.255.248.0	524 288
...	...	...
/1	128.0.0.0	2 147 483 648

Przydział adresów (2)

Hierarchia:

- IANA – Internet Assigned Numbers Authority,
- 5 x RIR – Regional Internet Registry,
- organizacje (komercyjne, naukowe, rządowe), ISPs.



Klasy adresów prywatnych

- Dla niektórych zastosowań nie jest konieczne posiadanie unikalnych adresów.
- Zakres adresów, które można wykorzystywać bez rejestracji, to adresy prywatne.
- Klasy adresów prywatnych:
  - A – adresy `10.0.0.0` – `10.255.255.255`, maska `/8`,
  - B – adresy `172.16.0.0` – `172.31.255.255`, maska `/12`,
  - C – adresy `192.168.0.0` – `192.168.255.255`, maska `/16`.
- Adresy zarezerwowane:
  - `127.0.0.0/8` – pętla (loopback),
  - `192.0.2.0/24` – przykład (TEST-NET),
  - inne (link).

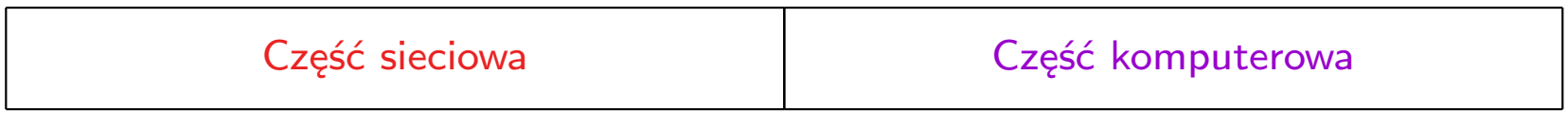
Podział sieci (1)

Problem:

- firma dysponuje jednym adresem sieci np. 150.10.0.0/16,
- firma posiada kilka budynków – w każdym znajduje się sieć LAN,
- jak zaplanować adresację w firmie?

Podział sieci (2)

Część sieciowa ma długość s bitów (maska wynosi /s).
Część komputera ma długość k bitów.



Podział sieci na p podsieci, wyznaczanie adresów podsieci:

1. *Pożyczamy* z początku części komputerowej tyle bitów, żeby móc ponumerować p podsieci: $x = \lceil \log_2(p) \rceil$.
2. Używamy *pożyczonych* bitów, żeby ponumerować podsieci.
3. Powiększamy maskę o x.

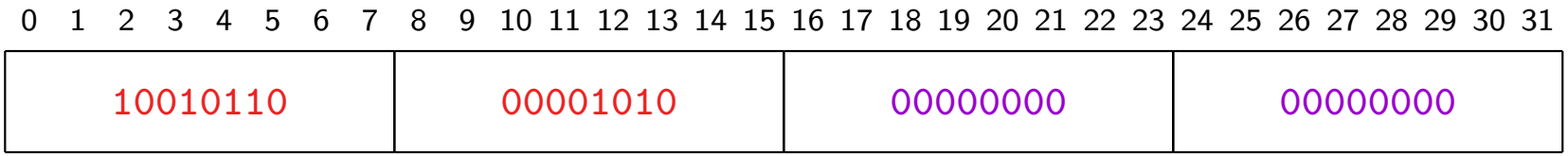
Podział sieci (3)

Uzyskujemy:

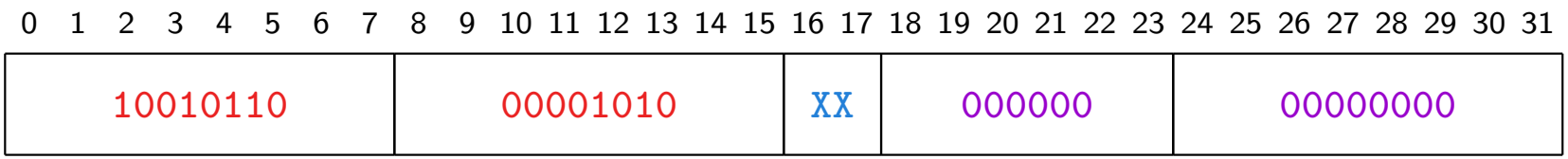
- każda podsieć ma własny adres sieci i rozgłoszeniowy,
- w każdej sieci można zaadresować $2^{k-x} - 2$ komputerów,
- maska podsieci jest dłuższa niż maska oryginalnej sieci: $s + x$.

Podział sieci – przykład (1)

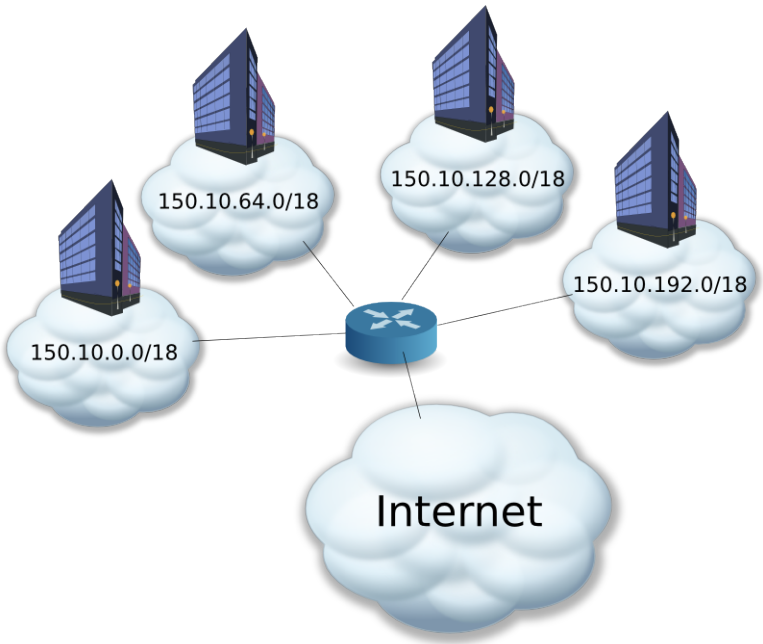
Adres sieci: 150.10.0.0/16, podział na 4 podsieci:



- adresacja podsieci wymaga $\lceil \log_2(4) \rceil = 2$ bitów,
- numeracja podsieci:
  - 00 – podsieć pierwsza,
  - 01 – podsieć druga,
  - 10 – podsieć trzecia,
  - 11 – podsieć czwarta.



Podział sieci – przykład (2)



Nr	Adres podsieci	Maska	Adres min.	Adres maks.	Adres rozgł.
00	150.10.0.0	/18	150.10.0.1	150.10.63.254	150.10.63.255
01	150.10.64.0	/18	150.10.64.1	150.10.127.254	150.10.127.255
10	150.10.128.0	/18	150.10.128.1	150.10.191.254	150.10.191.255
11	150.10.192.0	/18	150.10.192.1	150.10.255.254	150.10.255.255

Zadanie 2

Dana jest sieć `158.75.136.0/22`:

1. Podziel sieć na 8 równych sieci.
2. Dla każdej podsieci podaj adres sieci, maskę, adres rozgłoszeniowy oraz zakres adresów.

Podział sieci na podsieci różnej wielkości

Problem:

- firma posiada kilka budynków znacznie różniących się liczbą komputerów.

Podział sieci na podsieci różnej wielkości

Problem:

- firma posiada kilka budynków znacznie różniących się liczbą komputerów.

Variable Length Subnet Masking (VLSM):

- zasada postępowania:
  1. podział sieci na równe podsieci,
  2. podział podsieci na jeszcze mniejsze podsieci.

Podsieci różnej wielkości – przykład (1)

Podział sieci 150.10.0.0/16 na:

- 3 sieci po 15 000 komputerów oraz
- 4 sieci po 4000 komputerów.

Podział sieci 150.10.0.0/16 na 4 podsieci /18:

Nr	Adres podsieci	Maska	Adres min.	Adres maks.	Adres rozgł.
00	150.10.0.0	/18	150.10.0.1	150.10.63.254	150.10.63.255
01	150.10.64.0	/18	150.10.64.1	150.10.127.254	150.10.127.255
10	150.10.128.0	/18	150.10.128.1	150.10.191.254	150.10.191.255
11	150.10.192.0	/18	150.10.192.1	150.10.255.254	150.10.255.255

Podział sieci 150.10.64.0/18 na 4 podsieci /20:

Nr	Adres podsieci	Maska	Adres min.	Adres maks.	Adres rozgł.
00	150.10.64.0	/20	150.10.64.1	150.10.79.254	150.10.79.255
01	150.10.80.0	/20	150.10.80.1	150.10.95.254	150.10.95.255
10	150.10.96.0	/20	150.10.96.1	150.10.111.254	150.10.111.255
11	150.10.112.0	/20	150.10.112.1	150.10.127.254	150.10.127.255

Podsieci różnej wielkości – przykład (2)

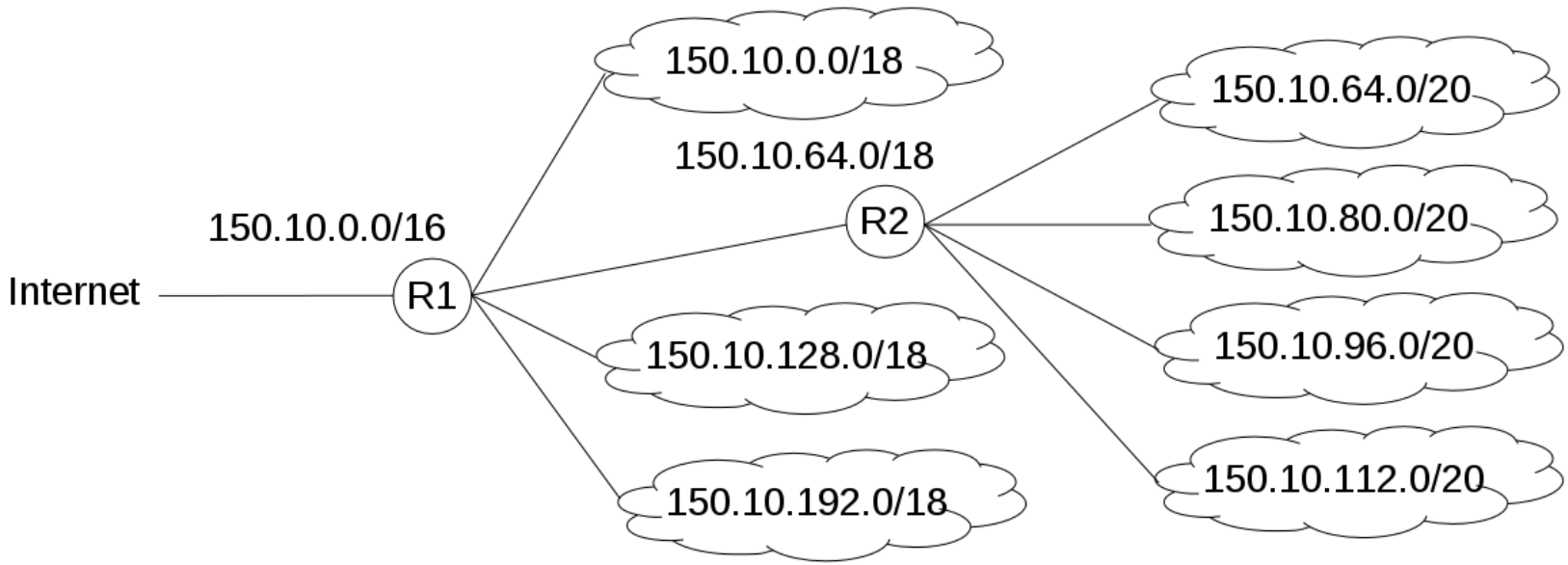
Wszystkie podsieci:

- 150.10.0.0/18
- 150.10.128.0/18
- 150.10.192.0/18
- 150.10.64.0/20
- 150.10.80.0/20
- 150.10.96.0/20
- 150.10.112.0/20

Routery znajdujące się poza siecią firmową mogą utrzymywać tylko jeden wiersz w swoich tablicach tras, zawierający wyłącznie sieć 150.10.0.0/16.

Trasy do podsieci muszą być zapamiętane w routerach wewnątrz sieci firmowej.

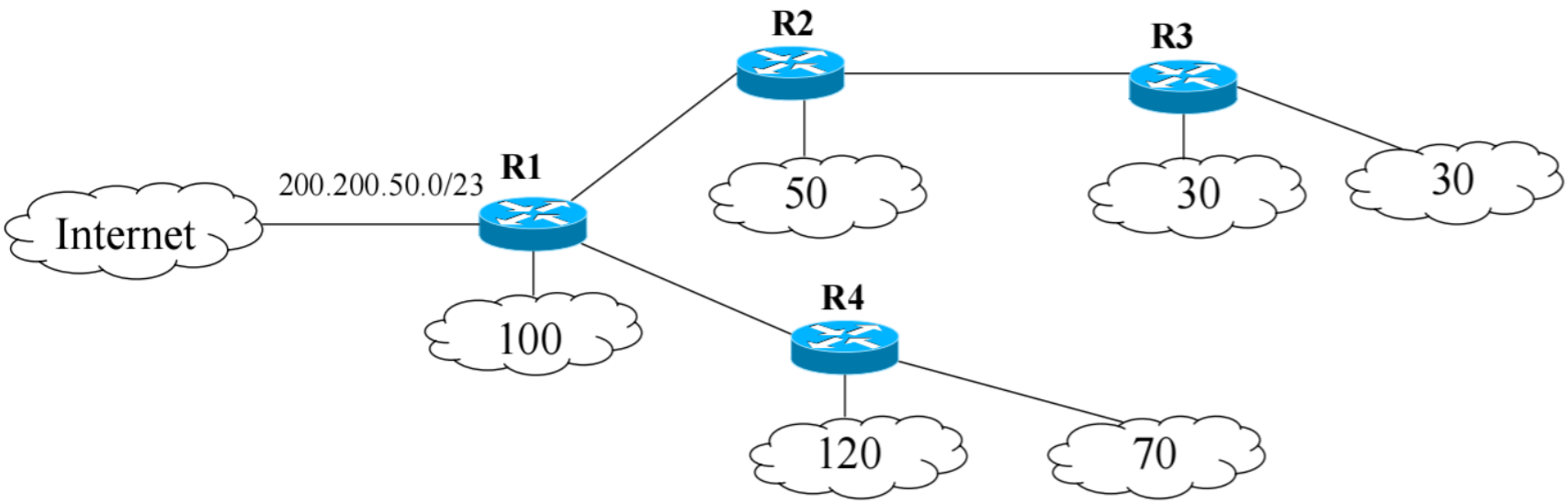
Podsieci różnej wielkości – przykład (3)



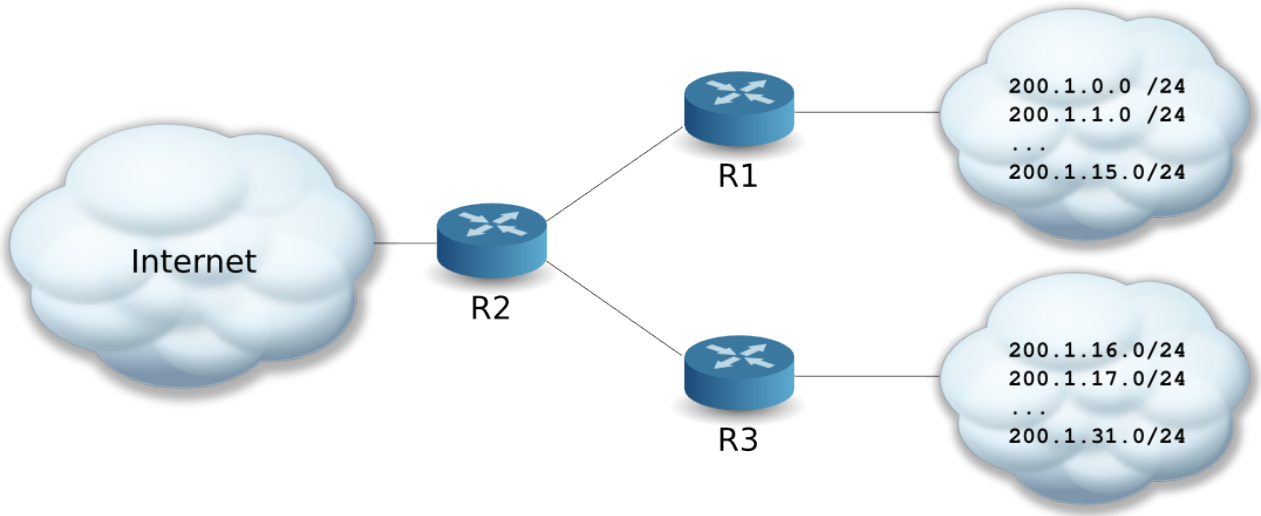
Zadanie 3

Zaproponuj schemat adresacji dla sieci z rysunku (rozmiary sieci w chmurkach):

- dostawca usług R1 (jego nazwa bierze się od nazwy routera R1) dysponuje adresem CIDR `200.200.50.0/23`,
- R1 część przestrzeni adresowej przeznacza dla własnych sieci IP, a resztę oddaje swoim poddostawcom R2 i R4,
- R2 postępuje analogicznie jak R1 względem swojego poddostawcy.



Agregacja adresów sieci (1)



	Sieć	Maska	Brama
R2	200.1.0.0	/24	R1
	200.1.1.0	/24	R1
	200.1.2.0	/24	R1
	...	...	...
	200.1.15.0	/24	R1
	200.1.16.0	/24	R3
	200.1.17.0	/24	R3
	200.1.18.0	/24	R3
	...	...	...
	200.1.31.0	/24	R3

Agregacja adresów sieci (2)

Zasada agregacji (mając dany zakres adresów IP):

1. Znajdź najdłuższy wspólny prefiks ich części sieciowych.
2. Skróć część sieciową (maskę) i utwórz uogólniony adres IP sieci (tzw. adres nadsieci lub *CIDR – supernet address* lub *CIDR address*).

Uwaga: nadsieć nie może zawierać adresów spoza danego zakresu adresów IP!

Agregacja adresów sieci (3)

Agregacja adresów dla routera R2:

200.1.0.0/24	11001000.00000001.00000000.00000000
200.1.1.0/24	11001000.00000001.00000001.00000000
200.1.2.0/24	11001000.00000001.00000010.00000000
200.1.3.0/24	11001000.00000001.00000011.00000000
...	...
200.1.15.0/24	11001000.00000001.00001111.00000000
200.1.16.0/24	11001000.00000001.00010000.00000000
200.1.17.0/24	11001000.00000001.00010001.00000000
200.1.18.0/24	11001000.00000001.00010010.00000000
200.1.19.0/24	11001000.00000001.00010011.00000000
...	...
200.1.31.0/24	11001000.00000001.00011111.00000000

Nowa tablica routingu na R2:

Sieć	Maska	Brama
200.1.0.0	/20	R1
200.1.16.0	/20	R3

Agregacja adresów sieci (4)

Często nie można dokonać agregacji do jednego adresu nadrzędnego.

200.1.48.0/24	11001000.00000001.00110000.00000000
200.1.49.0/24	11001000.00000001.00110001.00000000
...	...
200.1.63.0/24	11001000.00000001.00111111.00000000
200.1.64.0/24	11001000.00000001.01000000.00000000
200.1.65.0/24	11001000.00000001.01000001.00000000
...	...
200.1.79.0/24	11001000.00000001.01001111.00000000

Agregacja do jednego adresu 200.1.0.0/17 ujęłaby nieistniejące adresy sieci, np: 200.1.127.0/24.

Agregacja adresów sieci (5)

W tym przypadku należy wykonać dwie osobne agregacje:

```
200.1.48.0/24   11001000.00000001.00110000.00000000
200.1.49.0/24   11001000.00000001.00110001.00000000
      ...                          ...
200.1.63.0/24   11001000.00000001.00111111.00000000
```

Nadsieć: 200.1.48.0/20.

```
200.1.64.0/24   11001000.00000001.01000000.00000000
200.1.65.0/24   11001000.00000001.01000001.00000000
      ...                          ...
200.1.79.0/24   11001000.00000001.01001111.00000000
```

Nadsieć: 200.1.64.0/20.

Zadanie 4

Dokonaj agregacji poniższych wpisów na routerze R1 tak bardzo jak się da:

Sieć	Maska	Brama
150.254.80.0	/24	R2
150.254.81.0	/24	R2
150.254.82.0	/24	R2
...		
150.254.111.0	/24	R2
150.254.112.0	/24	R2

Podstawowe narzędzia sieciowe w Linuksie

Pakiety narzędzi net-tools i route2

Pakiet net-tools:

- od 1983, początkowo w BSD, nie rozwijany dalej,
- arp,
- ifconfig,
- hostname,
- netstat,
- inne: dnsdomainname, domainname, ipmaddr, iptunnel, mii-tool, nameif, nisdomainname, plipconfig, rarp, route, slattach, ypdomainname.

Pakiet route2:

- od 2000 roku, miał zastąpić net-tools,
- ip,
- inne: arpd, ctstat, genl, ifcfg, ifstat, lnstat, nstat, routef, routel, rtacct, rtmon, rtpr, rtstat, ss, tc.

Inne narzędzia

Pakiet `iputils`:

- `arping`,
- `ping`,
- `tracepath`,
- inne: `clockdiff`, `ipg`, `ping6`, `rarpd`, `rdisc`, `tftpd`, `tracepath6`, `traceroute6`.

Przydatne są również `mtr` i `traceroute`.

Karty i interfejsy sieciowe

Karta sieciowa – fizyczne urządzenie pozwalające komputerowi na utrzymywanie połączenia z siecią komputerową (realizowaną w określonej technologii, np. Ethernet, FDDI, Token Ring).

Interfejs sieciowy – punkt dostępu do sieci komputerowej utrzymywany przez system operacyjny:

- interfejs fizyczny – punkt dostępu związany z fizycznym urządzeniem, np. kartą ethernetową czy kartą wifi,
- interfejs logiczny – punkt dostępu do sieci logicznej,
  - z jednym interfejsem fizycznym może być związanych wiele interfejsów logicznych,
  - interfejs logiczny może być związany też z urządzeniami wirtualnymi, mostkami, tunelami, etc.

Nazwy interfejsów sieciowych

Tradycyjne nazwy (rozszerzone o numery, od 0):

- loopback: `lo` (bez numeru),
- przewodowe karty sieciowe: `eth`,
- bezprzewodowe karty sieciowe: `wlan`, `ath`, `wifi`, `radio`,
- `firewire`, infiniband: `ib`,
- urządzenia wirtualne: `dummy`, mostki: `br`, tunele: `tun`, `tap`, `sit`, `tnl`, `ppp`, `vpn`, `gre`.

Consistent Network Device Naming – powiązanie nazwy z konkretnym urządzeniem fizycznym (lub jego umiejscowieniem), np.:

- port ethernetowy na płycie głównej: `em`,
- port ethernetowy na PCI: `p<numer slotu>p<numer portu>`,
- karta wifi: `wlp3s0`.

Interfejsy sieciowe i ustawienia sieci (1)

```
# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
    state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500
    qdisc mq state DOWN group default qlen 1000
    link/ether 84:3a:4b:71:94:5c brd ff:ff:ff:ff:ff:ff
# ip addr add 192.168.1.103/24 dev eth0
# ip addr show eth0
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500
    ...
    inet 192.168.1.103/24 scope global eth0
       valid_lft forever preferred_lft forever
# ip addr del 192.168.1.103/24 dev eth0
```

Interfejsy sieciowe i ustawienia sieci (2)

```
# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 6018  bytes 520408 (508.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6018  bytes 520408 (508.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 84:3a:4b:71:94:5c  txqueuelen 1000  (Ethernet)
        RX packets 7381654  bytes 8531909613 (7.9 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3764753  bytes 476654769 (454.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
# ifconfig eth0 192.168.1.103 netmask 255.255.255.0
# ifconfig eth0
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.1.103  netmask 255.255.255.0  broadcast 192.168.1.255
...
# ifconfig eth0 0.0.0.0
```

Interfejsy sieciowe i ustawienia sieci (3)

```
# ip addr add 192.168.1.103/24 dev eth0
# ip addr add 172.16.0.103/16 dev eth0 label eth0:1
# ip addr add 10.0.0.103/8 dev eth0 label eth0:kokoszka
# ip addr show eth0
...
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500
    qdisc mq state DOWN group default qlen 1000
    link/ether 84:3a:4b:71:94:5c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 scope global eth0
       valid_lft forever preferred_lft forever
    inet 172.16.0.103/16 scope global eth0:1
       valid_lft forever preferred_lft forever
    inet 10.0.0.103/8 scope global eth0:kokoszka
       valid_lft forever preferred_lft forever
# ip addr del 10.0.0.103/8 dev eth0
# ip addr flush dev eth0
```

Interfejsy sieciowe i ustawienia sieci (4)

```
# ifconfig eth0 192.168.1.103 netmask 255.255.255.0
# ifconfig eth0:1 172.16.0.103 netmask 255.255.0.0
# ifconfig eth0:kokoszka 10.0.0.103 netmask 255.0.0.0
# ifconfig
...
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.1.103  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 84:3a:4b:71:94:5c  txqueuelen 1000  (Ethernet)
        RX packets 7381654  bytes 8531909613 (7.9 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3764753  bytes 476654769 (454.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
eth0:1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.16.0.103  netmask 255.255.0.0  broadcast 172.16.255.255
        ether 84:3a:4b:71:94:5c  txqueuelen 1000  (Ethernet)
eth0:kokoszka: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 10.0.0.103  netmask 255.0.0.0  broadcast 10.255.255.255
        ether 84:3a:4b:71:94:5c  txqueuelen 1000  (Ethernet)
# ifconfig eth0:kokoszka down
# ifconfig eth0 down
```

Podglądanie ruchu sieciowego

Pakiety `wireshark` i `wireshark-gnome`.

Interesujące funkcje:

- logowanie i filtrowanie ruchu,
- widok stosu protokołów $\longrightarrow$ enkapsulacja wiadomości,
- śledzenie strumienia TCP/UDP,
- statystyki, przebiegi, grafy,
- ...

Uwagi:

- przenumerowywanie różnych wartości (np. numery sekwencyjne) dla większej czytelności,
- domyślnie preambuła i FCS nie są pokazywane.

`wireshark` – **uwagi**

Uruchamianie:

- na OpenSUSE do prawidłowej pracy wymaga roota,
- jeśli pojawi się błąd: `Can't open display:`, należy wykonać `export DISPLAY=:0`,
- jeśli pojawi się błąd: `Unable to open display :0` lub `Cannot connect to X server`, należy z prawami użytkownika `student` wykonać `xhost +`.

Sprawdzanie łączności IP (1)

```
# ping 192.168.1.101
PING 192.168.1.101 (192.168.1.101) 56(84) bytes of data.
64 bytes from 192.168.1.101: icmp_seq=1 ttl=64 time=0.690 ms
64 bytes from 192.168.1.101: icmp_seq=2 ttl=64 time=0.357 ms
64 bytes from 192.168.1.101: icmp_seq=3 ttl=64 time=0.416 ms
64 bytes from 192.168.1.101: icmp_seq=4 ttl=64 time=0.448 ms
^C
--- 192.168.1.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.357/0.477/0.690/0.129 ms
```

Sprawdzanie łączności IP (2)

```
# traceroute -I wikipedia.org
traceroute to wikipedia.org (91.198.174.192), 30 hops max, 60 byte packets
 1 gateway (150.254.31.1) 1.227 ms 1.532 ms 1.819 ms
 2 hellfire.put.poznan.pl (150.254.6.36) 1.532 ms 1.533 ms 1.534 ms
 3 mx-pp-wilda-irb-1.man.poznan.pl (150.254.163.29) 1.804 ms 1.805 ms 1.805 ms
 4 mx-pcss-1-AE0-1.man.poznan.pl (150.254.255.64) 2.056 ms 2.063 ms 2.063 ms
 5 212.191.237.217 (212.191.237.217) 2.062 ms 2.062 ms 2.313 ms
 6 ae2.cr2-esams.wikimedia.org (80.249.209.176) 30.667 ms 27.952 ms 23.770 ms
 7 text-lb.esams.wikimedia.org (91.198.174.192) 23.770 ms 23.772 ms 24.033 ms
# mtr wikipedia.org
 dcs-rw-2.cs.put.poznan.pl (0.0.0.0)
                                                                Thu Mar  9
13:05:48 2017
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                Packets               Pings
 Host                         Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 150.254.31.1               0.0%     9    1.0   1.4   1.0   2.6   0.0
 2. hellfire.put.poznan.pl     0.0%     9    0.6   0.8   0.5   1.6   0.0
 3. mx-pp-wilda-irb-1.man.poznan.pl 0.0%     9    1.1   1.0   0.9   1.3   0.0
 4. mx-pcss-1-AE0-1.man.poznan.pl   0.0%     8    1.0   1.9   1.0   5.9   1.5
 5. 212.191.237.217            0.0%     8    1.1  11.4   1.0  52.0  17.2
 6. ae2.cr2-esams.wikimedia.org     0.0%     8   24.5  24.3  24.0  25.2   0.0
 7. text-lb.esams.wikimedia.org     0.0%     8   26.0  24.4  23.9  26.0   0.7
```

Zadanie 5

1. Wspólnie z koleżankami/kolegami z rzędu ustal adres sieci, który pozwoli na zaadresowanie wszytkich komputerów w rzędzie. Sieci są unikalne między rzędami.
2. Przypisz jeden z adresów do interfejsu `em1` (każdy komputer w rzędzie ma mieć inny adres).
3. Przeanalizuj ruch na interfejsie `em1` przy pomocy programu `wireshark`.
4. Zbadaj przy pomocy polecenia `ping` połączenie z innymi komputerami w rzędzie.
5. Zbadaj przy pomocy polecenia `ping` połączenie komputerami w innych rzędach.
6. Parami połącz się z innym komputerem z rzędu przy pomocy komendy `telnet` i zbadaj treść wymienianych pakietów.

Warstwa fizyczna

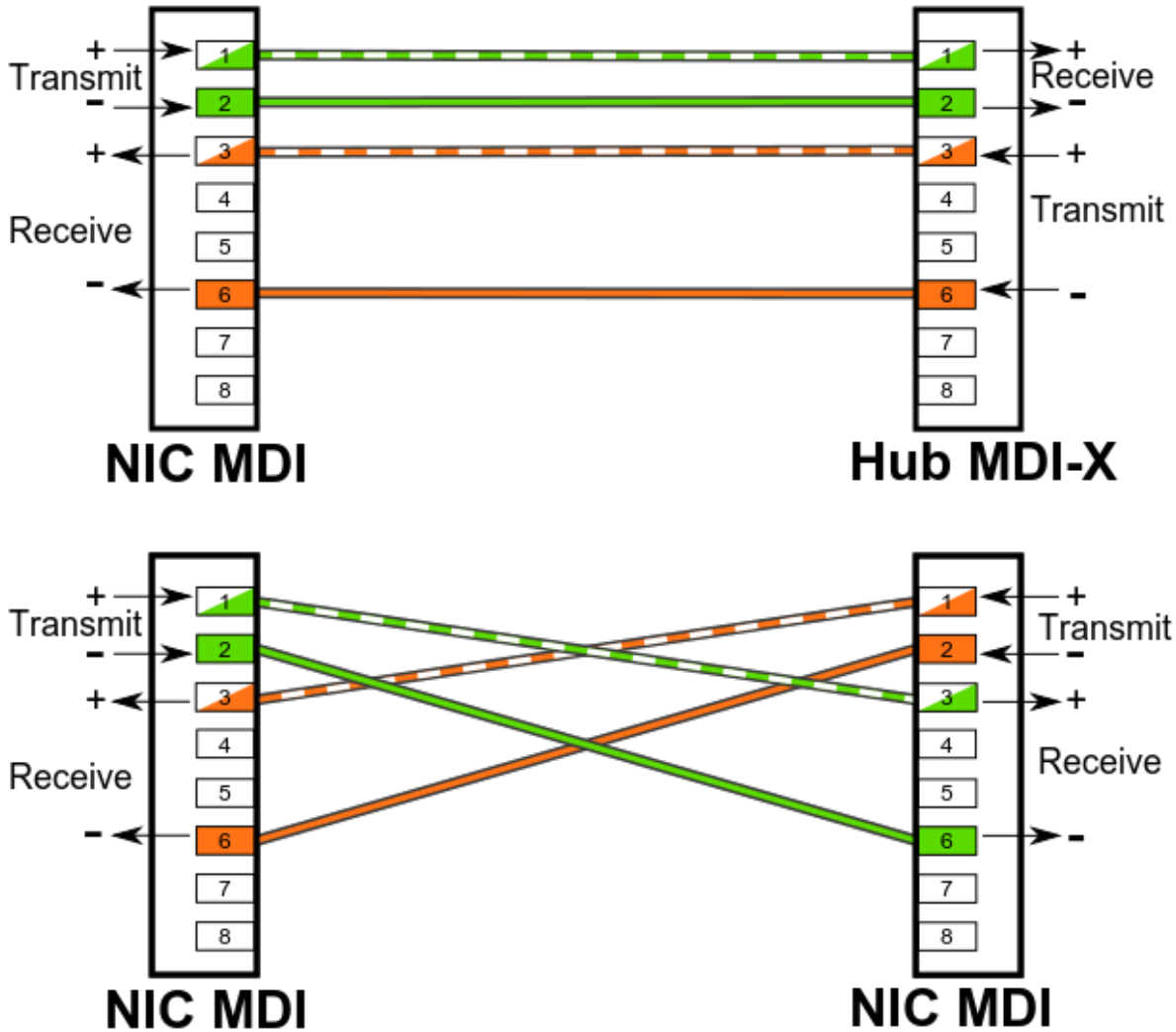
Warstwa fizyczna

- Umożliwia przesyłanie poszczególnych bitów (ramek) przez dane fizyczne łącze.
- Określa jak bity zamieniane są na sygnał (elektryczny, optyczny, radiowy) propagowany przez łącze fizyczne.
- Standaryzuje mechanizmy obsługi transmisji danych, techniki sygnalizacji, powiadamiania o błędach, fizyczne własności kart sieciowych i nośników fizycznych.
- Nie uwzględnia mechanizmów badania integralności danych.
- Przykładowe standardy:
  - RS232C
  - V.35
  - 8P8C (RJ45)
  - 802.11 a/b/g/n PHY
  - 802.3*
  - 10BASE-T
  - 100BASE-TX
  - 1000BASE-T
  - T1
  - E1
  - SONET
  - SDH
  - DWDM

Standard Ethernet (IEEE 802.3)

- Medium: skrętka, światłowód, kabel koncentryczny.
- Maksymalne odległości: do 100 metrów (podwójna skrętka), do 100 km (światłowód).
- Topologia: punkt-punkt, gwiazda, szyna.
- Typowe złącza: 8P8C (popularnie RJ45), LC, SC, ST,.
- Kable (skrętka, 8P8C):
  - proste (ang. *straight-through*),
  - krosowane (ang. *cross-over*)– urządzenia tej samej warstwy,
  - konsolowe (ang. *roll over*) – podłączenie komputera do konsoli routera,
  - auto MDI-MDIX (medium dependent interface (cross over)).
- Więcej informacji (link).

Kabel prosty vs kabel krosowany



Protokoły rozwiązywania problemu kolizji

Carrier Sense Multiple Access/Collision Detection (CSMA/CD) – w Ethernecie w ramach jednej domeny kolizyjnej:

- transmisja jest zamykana po wykryciu kolizji,
- wysyłany jest tam sygnał *zagłuszający*,
- następuje retransmisja po czasie losowej długości.

Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) – w sieciach bezprzewodowych:

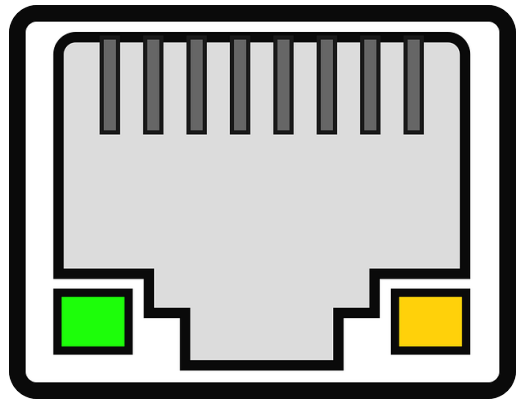
- przed próbą transmisji wysyłany jest sygnał próbny (pilot),
- nadawanie tylko jeśli żadne inne urządzenie nie nadaje,
- gdy kolizja jest wykryta, następuje próba retransmisji po czasie losowej długości,
- powoduje duże straty czasowe $\rightarrow$ CSMA/AMP i CSMA/CA+AMP (z priorytetami wiadomości).

Kable – oznaczenia

Na przykład: 10BASE-T, 1000BASE-LX, 40GBASE-LR4:

- nominalna prędkość: 10, 100, 1000, 40G, etc.,
- wykorzystywane pasmo: baseband, broadband, passband,
- medium:
  - T – skrętka (ang. twisted pair),
  - S – światło podczerwone 850 nm (ang. short wavelength, multi-mode fiber),
  - L – światło podczerwone 1300 nm (ang. long wavelength, mostly single-mode fiber),
  - E/Z – światło podczerwone 1500 nm (ang. extra long wavelength, single-mode fiber),
  - C – miedziany, dwużyłowy kabel koncentryczny (ang. copper/twinax),
- modulacje i kodowanie, np. X - kodowanie PCS 8b/10b, R - kodowanie PCS 64b/66b,
- liczba linii per łącze (dla WAN PHY), np. 1, 2, 4, 10.

RJ45 – rozwiązywanie problemów



- Zazwyczaj jedna dioda sygnalizuje podłączenie kabla (ciągłe światło), a druga transmisję danych (przerywane światło).
- Czasem kolor światła związany jest z wynegocjowaną prędkością przesyłania danych.

Parametry interfejsu sieciowego

```
# ethtool eth0
Settings for eth0:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Advertised pause frame use: Symmetric
        Advertised auto-negotiation: Yes
        Speed: 100Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 16
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: pg
        Wake-on: p
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: no
```

Zmiana ustawień interfejsu sieciowego

```
# ifconfig eth0 down
# ethtool -s eth0 speed 10 duplex half autoneg off
# ethtool eth0
  ...
        Speed: 10Mb/s
        Duplex: Half
        Auto-negotiation: off
  ...
# ifconfig eth0 up
```

- Zmiana ustawień powoduje, że interfejs trzeba ponownie *podnieść*.
- Duplex:
  - full – równoczesne przesyłanie danych w obie strony (podłączenie do przełącznika),
  - half – w danej chwili przesyłanie danych tylko w jedną stronę (podłączenie do koncentratora),
  - auto-negotiation – urządzenie samo decyduje o trybie działania.

Statystyki interfejsu sieciowego

```
# ethtool -S eth0
NIC statistics:
     rx_bytes: 74356477841
     rx_error_bytes: 0
     tx_bytes: 110725861146
     tx_error_bytes: 0
     rx_ucast_packets: 104169941
     rx_mcast_packets: 138831
     rx_bcast_packets: 59543904
     tx_ucast_packets: 118118510
     tx_mcast_packets: 10137453
     tx_bcast_packets: 2221841
     ...
     rx_oversize_packets: 0
     rx_64_byte_packets: 61154057
     rx_65_to_127_byte_packets: 55038726
     rx_128_to_255_byte_packets: 426962
     rx_256_to_511_byte_packets: 3573763
     rx_512_to_1023_byte_packets: 893173
     rx_1024_to_1522_byte_packets: 42765995
     rx_1523_to_9022_byte_packets: 0
     ...
```

Informacje o sterowniku i identyfikacja interfejsu sieciowego

```
# ethtool -i enp0s26u1u2
driver: asix
version: 22-Dec-2011
firmware-version: ASIX AX88772 USB 2.0 Ethernet
bus-info: usb-0000:00:1a.0-1.2
supports-statistics: no
supports-test: no
supports-eeprom-access: yes
supports-register-dump: no
supports-priv-flags: no
```

```
(czasem nie wspierane)
# ethtool -p enp0s26u1u2
```

Zadanie 6

1. Zmierz osiąganą maksymalną prędkość transmisji przy użyciu usługi `speedtest.net`.
2. Zmień parametry `em1` ograniczając prędkość transmisji i zmieniając parametry dupleksu.
3. Ponownie zmierz maksymalną prędkość transmisji.
4. Wróć do pierwotnych ustawień.
5. Przy pomocy odpowiedniego parametru komendy `ethtool` zidentyfikuj interfejsy `p4p1` oraz `p4p2`.

Trwałe zmiany ustawień interfejsu sieciowego

Fedora/RedHat/CentOS:

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
...
ETHTOOL_OPTS="speed 100 duplex full autoneg off"
```

OpenSUSE:

```
# cat /etc/sysconfig/network/ifcfg-eth0
POST_UP_SCRIPT='eth0'
# cat /etc/sysconfig/network/scripts/eth0
#!/bin/bash
/sbin/ethtool -s speed 1000 duplex full autoneg off
```

Ubuntu:

```
# cat /etc/network/interfaces
post-up ethtool -s eth0 speed 1000 duplex full autoneg off
```

Ustawienia interfejsu wifi

Odpowiednik `ethtool` dla radiowych kart sieciowych.

```
# iwlist wlan0 scan
Cell 01 - Address: 0e:51:9d:33:19:83
          ESSID:"Kukuczka"
          Mode:Master
          Channel:11
          Frequency:2.462 GHz (Channel 11)
          Quality=100/100  Signal level:-47dBm  Noise level=-100dBm
          Encryption key:off
...
# iwconfig wlan0 essid Kukuczka
# iwconfig wlan0 channel 11 # freq 2.462G
# iwconfig wlan0 retry 16
# iwconfig wlan0 commit

# iwconfig wlan0 mode ad-hoc # managed,master
```

Zadanie 7

1. Usuń nadane wcześniej adresy IP na `em1` (mają pozostać tylko standardowe adresy, tj. uzyskiwane przez DHCP).
2. Podłącz swój komputer poprzez port `p4p1` do przełącznika na zapleczu (wszystkie komputery są podłączane do tego samego przełącznika).
3. Skonfiguruj adres IP na porcie `p4p1`, tak by wszystkie komputery w laboratorium mogły komunikować się między sobą poprzez nowo utworzoną sieć.

Warstwa łącza danych

Warstwa łącza danych

- Zapewnia niezawodne dostarczanie danych przez znajdującą się poniżej fizyczną sieć.
- Komunikacja wyłącznie z urządzeniami połączonymi bezpośrednio do tej samej sieci (segmentu sieci).
- Dwie podwarstwy:
  - logical link control (LLC) – sterowanie połączeniem logicznym,
  - media access control (MAC) – sterowanie dostępem do mediów.
- Przykładowe standardy:
  - MAC/LLC
  - PPP
  - ATM
  - Frame Relay
  - HDLC
  - 802.1q
  - 802.3
  - 802.11a/b/g/n MAC/LLC

Podwarstwy LLC i MAC

Logical Link Control (LLC) – sterowanie połączeniem logicznym:

- rozdzielanie i łączenie danych transmitowanych różnymi protokołami (np. IP, IPX, Appletalk) przez warstwę MAC (multiplexing, demultiplexing; np. na podstawie nagłówka, określa do jakiego protokołu warstwy 3 przekazać otrzymaną),
- sterowanie przepływem, wykrywanie błędów, retransmisja danych (w zależności od technologii i protokołów).

Media Access Control (MAC) – sterowanie dostępem do mediów:

- kontrola (wielo)dostępu do medium transmisji (interfejs dla LLC),
- dzielenie na ramki i kontrola poprawności,
- adresacja komputerów w segmencie i przekazywanie informacji adresowych.

Adresacja w warstwie łącza danych

- Sześć par hexadecymalnych cyfr rozdzielonych dwukropkiem (lub myślnikiem):
  - `2f:de:47:00:32:27`,
  - `ff:ff:ff:ff:ff:ff` – adres rozgłoszeniowy,
  - `01:00:5e:00:00:00` – `01:00:5e:7f:ff:ff` – adresy multicast (najmniej znaczący bit pierwszego oktetu równy 1, mapowanie multicast IP-MAC $\longrightarrow$ link).
- Urządzenie odbiera ramki, które są przeznaczone dla jego adresu MAC, na adres rozgłoszeniowy, lub na adres multicast w którym się zawiera.
- Karta sieciowa w trybie `promiscuous` odbiera ramki kierowane do wszystkich adresów MAC.
- Producenci sprzętu posiadają przydzielone klasy adresów (link 1, link 2).

Adres MAC karty sieciowej (1)

```
# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536
    qdisc noqueue state UNKNOWN mode DEFAULT group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    qdisc mq state UP mode DORMANT group default qlen 1000
    link/ether 84:3a:4b:71:94:5c brd ff:ff:ff:ff:ff:ff
# ip link set dev eth0 down
# ip link set dev eth0 address aa:bb:cc:dd:ee:ff
# ip link show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    qdisc mq state UP mode DORMANT group default qlen 1000
    link/ether aa:bb:cc:dd:ee:ff brd ff:ff:ff:ff:ff:ff
# ip link set dev eth0 up
```

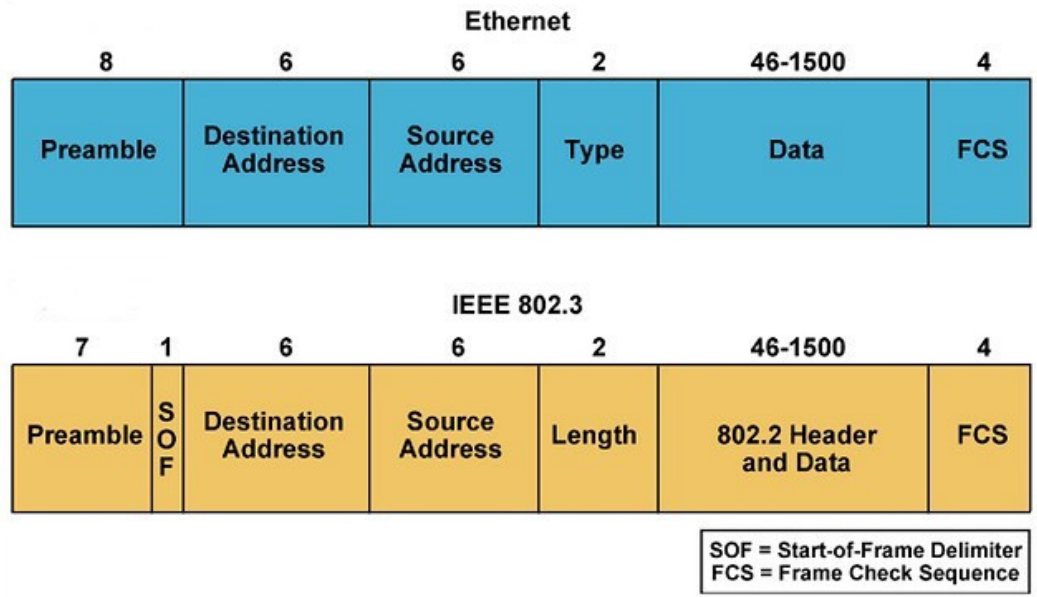
```
# ethtool -P eth0
Permanent address: 84:3a:4b:71:94:5c
```

Adres MAC karty sieciowej (2)

```
# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 4755  bytes 407972 (398.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4755  bytes 407972 (398.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.103  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 84:3a:4b:71:94:5c  txqueuelen 1000  (Ethernet)
        RX packets 3020949  bytes 3781419997 (3.5 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2068340  bytes 289950531 (276.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
# ifconfig eth0 down
# ifconfig eth0 hw ether aa:bb:cc:dd:ee:ff
# ifconfig eth0 up
# ifconfig
...
        ether aa:bb:cc:dd:ee:ff  txqueuelen 1000  (Ethernet)
```

Standard Ethernet (IEEE 802.3) (1)

- Obejmuje zarówno warstwę fizyczną jak i łącza danych.
- Struktura ramki (długości pól w bajtach/oktetach):



- Rozmiar: od 64B do 1500B.

Standard Ethernet (IEEE 802.3) (2)

- W Ethernet (v2) długość ramki 'zakodowana' jest w danych interpretowanych przez warstwę wyższą.
- Jak zbadać z ramką jakiego typu mamy do czynienia?

Standard Ethernet (IEEE 802.3) (2)

- W Ethernet (v2) długość ramki 'zakodowana' jest w danych interpretowanych przez warstwę wyższą.
- Jak zbadać z ramką jakiego typu mamy do czynienia?
- Kodowanie typu – wartości większe niż 1500:
  `0x0800`: IPv4
  `0x0806`: ARP
  `0x86DD`: IPv6
  ...
- FCS – algorytm Cyclic Redundancy Check (CRC):
  - detekcja wszystkich pojedynczych, podwójnych, potrójnych przekłamań,
  - detekcja ciągów długości n jednobitowych przekłamań, $n \leqslant 32$,
  - prawdopodobieństwo, że losowe przekłamanie bitów będzie niewykryte jest równe 2^{-10}.

Zadanie 8

1. Przy pomocy programu `wireshark` przeanalizuj ruch w sieci pod kątem występowania ramek Ethernet 802.3 i Ethernet v2.
2. W przypadku jakich pakietów daje się zaobserwować ramki Ethernet 802.3?

Urządzenia warstwy łącza danych

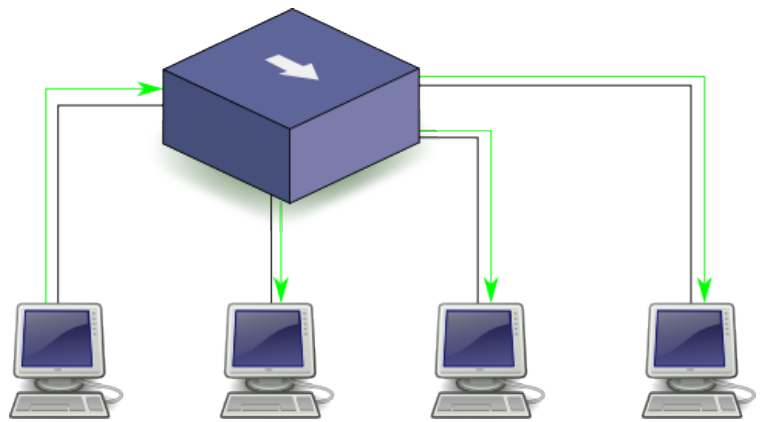
Podstawowe urządzenia warstwy łącza danych (w tym Ethernetu):

- koncentratory (hubs)[†],
- przełączniki (switches),
- mostki (bridges),
- ...

[†]Koncentratory tak naprawdę funkcjonują w warstwie fizycznej, bo w żaden sposób nie dokonują analizy zawartości ramek.

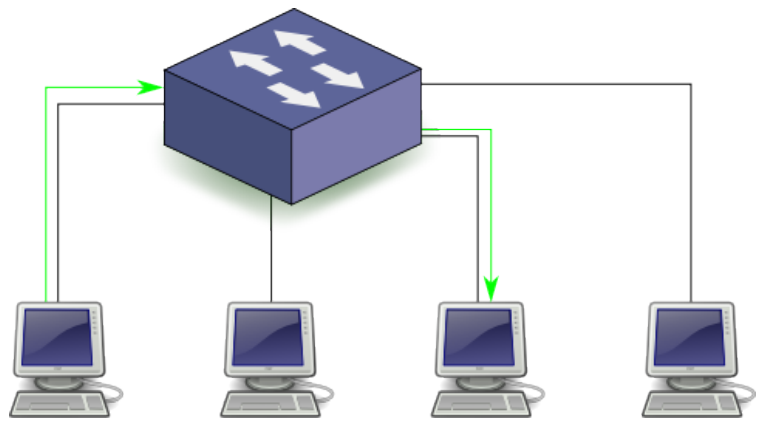
Koncentrator (hub)

- Wzmacnia sygnał i kieruje ramkę na wszystkie porty.
- Komputery współdzielą pasmo – pojedyncza domena kolizyjna (i za razem pojedyncza domena rozgłoszeniowa):
  - karty sieciowe urządzeń podłączonych do koncentratora muszą radzić sobie z występowaniem kolizji,
  - Carrier Sense Multiple Access with Collision Detection (CSMA/CD) (na kartach sieciowych, a nie na koncentratorze):
    - transmisja jest zamykana po wykryciu kolizji,
    - wysyłany jest tam sygnał *zagłuszający*,
    - następuje retransmisja po czasie losowej długości.



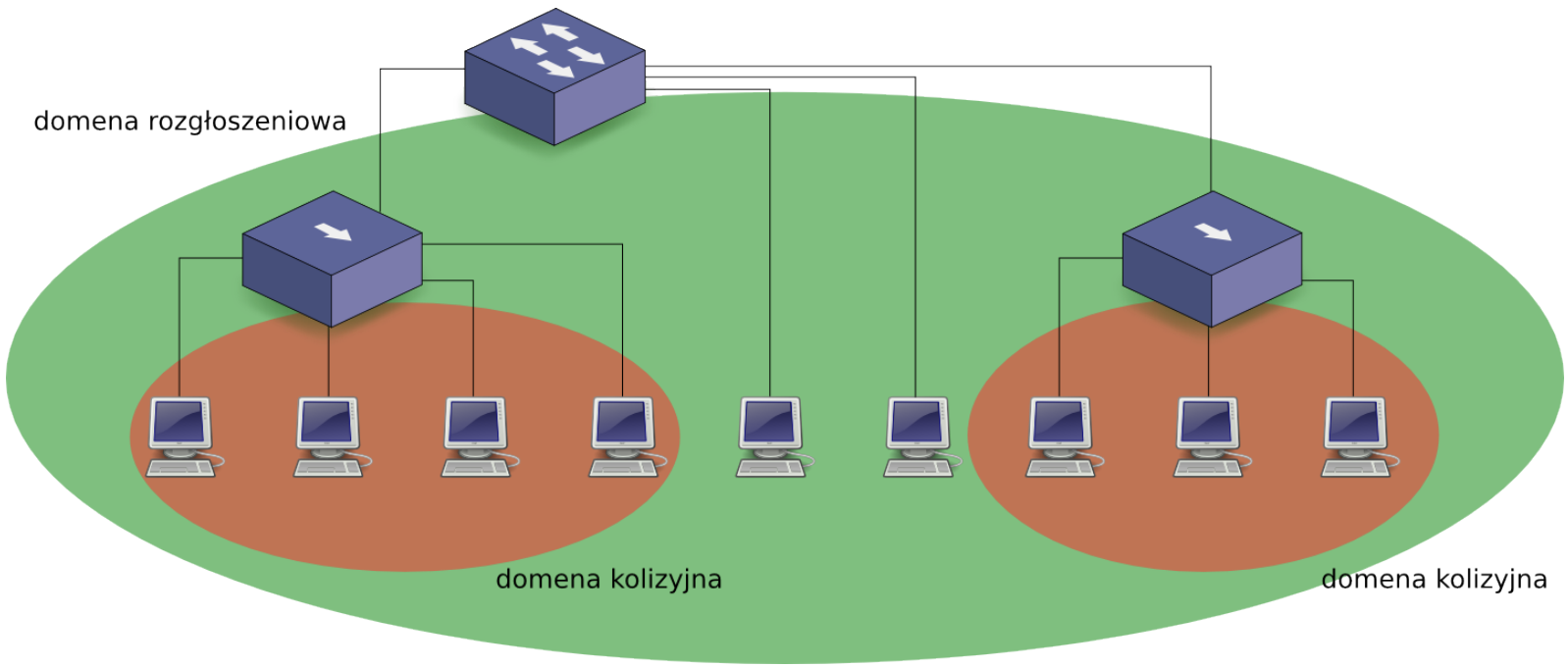
Przełącznik (switch) (1)

- Kieruje ramkę do właściwego portu.
- Uczy się adresów MAC kryjących się za określonymi portami (algorytm Transparent Bridging lub Backward Learning):
  - tablica par (adres MAC, numer portu), gdzie zapamiętuje adresy źródłowe wyciągnięte z ramek,
  - jeśli docelowy adres MAC nie jest zapisany, ramka wysyłana jest na wszystkie porty oprócz źródłowego pod adres rozgłoszeniowy (`ff:ff:ff:ff:ff:ff`),
  - przestarzałe adresy są usuwane.



Przełącznik (switch) (2)

- Jedna domena rozgłoszeniowa – wiadomość wysłana pod adres rozgłoszeniowy MAC trafi do każdego komputera,
- Separacja domen kolizyjnych (w przeciwieństwie do koncentratora).



Zadanie 9

1. Podłącz swój komputer (poprzez port `p4p1`) do koncentratora (na zapleczu).
2. Skonfiguruj interfejs `p4p1`, tak by wszystkie komputery w rzędzie działały w jednej sieci (unikalne sieci między rzędami).
3. Przy pomocy programu `wireshark` przeanalizuj ruch na interfejsie `p4p1`. Czy daje się zaobserwować pakiety z innych sieci (w szczególności komunikaty protokołu telnet)?

Pomiar prędkości

```
# netserver -D
Starting netserver with host 'IN(6)ADDR_ANY' port '12865' and
family AF_UNSPEC


(na innym komputerze)
# netperf -H 192.168.1.101
MIGRATED TCP STREAM TEST from 192.168.1.103 port 0 AF_INET to 192.168.1.101
port 0 AF_INET
Recv   Send    Send
Socket Socket  Message  Elapsed
Size   Size    Size     Time     Throughput
bytes  bytes   bytes    secs.    10^6bits/sec

 87380  16384  16384    10.00    941.83
```

Zadanie 10

1. W jednej chwili (wraz z koleżankami/kolegami) dokonaj pomiaru prędkości transmisji z sąsiednim komputerem z tego samego rzędu.
2. Wraz z innymi studentami zbadaj sumaryczną przepustowość koncentratora.

Zadanie 11

1. Przepnij (na zapleczu) swój komputer do wspólnego przełącznika i dokonaj ponownie pomiaru prędkości (jak w poprzednim zadaniu).
2. Skąd wynikają różnice w osiąganych prędkościach?
3. Czy w programie `wireshark` można zobaczyć pakiety z innych sieci? Jeśli tak to jakie?

Zadanie 12

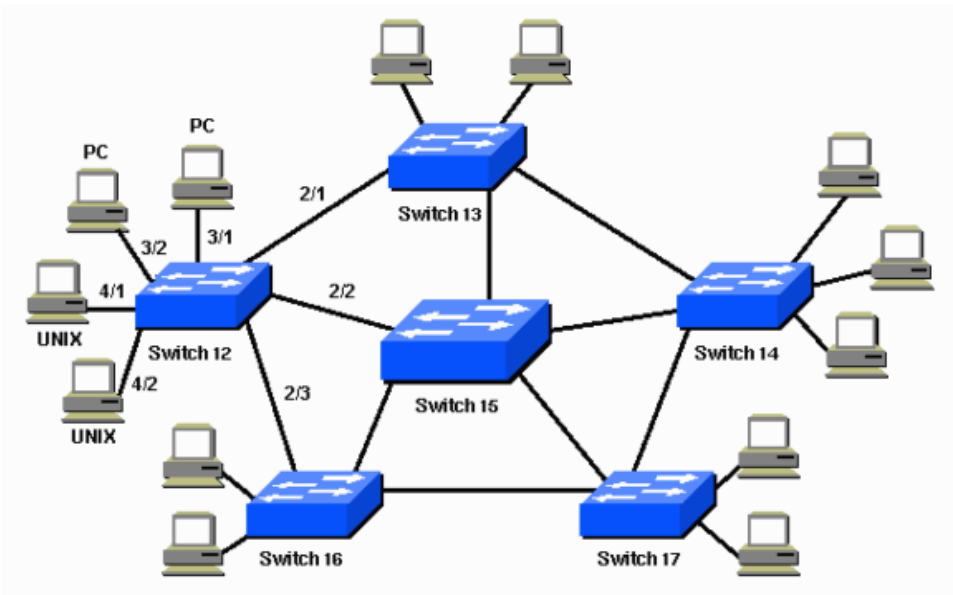
1. Wraz z koleżankami/kolegami przepnij połowę komputerów do drugiego przełącznika, tak by komputery w parach były podłączone do różnych przełączników.
2. Przełączniki połącz ze sobą kablem (krosowanym).
3. Dokonaj ponownie pomiaru prędkości (jak w poprzednim zadaniu).
4. Skąd wynikają różnice w osiąganych prędkościach?
5. Czy można łatwo zwiększyć przepustowość?

Zadanie 13

1. Dokonaj pomiaru prędkości transmisji do adresu `127.0.0.1`.
2. Czy do tej transmisji angażowana jest karta sieciowa?

Spanning Tree Protocol (1)

- W celu zwiększenia niezawodności, między przełącznikami używa się więcej połączeń niż trzeba.
- Skąd wiadomo którędy powininna wędrować ramka?



Spanning Tree Protocol (2)

- Spanning Tree Protocol (STP) – protokół drzewa rozpinającego graf $\longrightarrow$ eliminacja cykli z sieci połączonych ze sobą przełączników,
- Algorytm:
  1. Wybór korzenia: ten przełącznik, którego numer seryjny (unikalny w skali światowej) jest najmniejszy.
  2. Wyznaczanie najkrótszych ścieżek od korzenia do wszystkich innych przełączników.
  3. Deaktywacja połączeń, które nie występują w zbiorze najkrótszych ścieżek.
  4. W przypadku awarii któregoś z połączeń, wykonanie algorytmu ponownie.

Zadanie 14 (1)

1. Podepnij swój komputer na porcie `p4p1` do przełącznika, tak by jeden przełącznik przypadał na jeden rząd w laboratorium. Upewnij się, że działa połączenie z innymi komputerami w rzędzie.
2. Zepnij dwoma kablami przełącznik, do którego jest podłączony Twój komputer z przełącznikiem odpowiadającym sąsiedniemu rzędowi.
3. Odpowiednio skonfiguruj interfejs `p4p1` by móc skomunikować się z siecią z sąsiedniego rzędu lub by dało się zaobserwować jakikolwiek ruch w tamtej sieci.
4. Podglądaj ruch na interfejsie `p4p1` przy pomocy programu `wireshark` (najlepiej robić to tylko na jednym lub dwóch wybranych komputerach w rzędzie).

Zadanie 14 (2)

1. Jeden z komputerów w rzędzie podepnij do wejścia konsolowego przełącznika i uruchom (jako root) polecenie `picocom /dev/ttyS0`. Zmień ustawienia/wykonaj polecenia na przełączniku:
   - `terminate auto install: yes`
   - `enter auto configuration: no`
   - `enable`
   - `configure terminal`
   - `no spanning-tree vlan 1`
2. Jeśli nie działoby się nic spektakularnego, wykonaj komendę `ping` na adres z sieci zdefiniowanej dla rzędu, ale który jest nieprzydzielony żadnemu komputerowi w laboratorium.

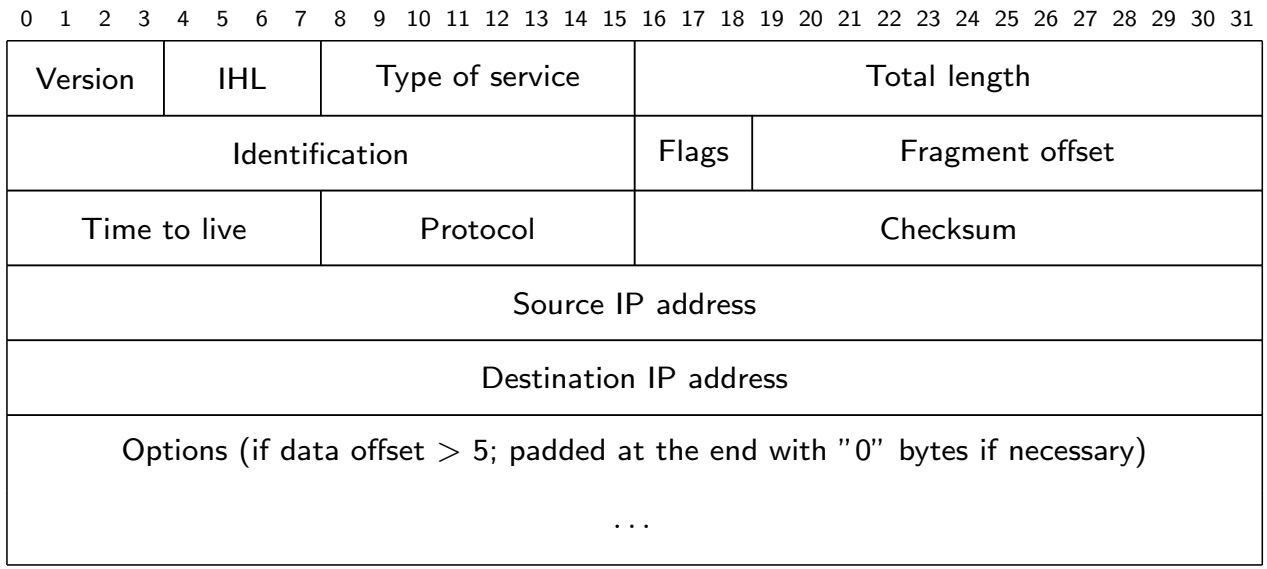
Warstwa sieciowa

Warstwa sieciowa

- Definiuje pakiety.
- Definiuje schemat adresowania używanego w Internecie.
- Ustala drogę transmisji danych między oddalonymi węzłami sieci.
- Odpowiada za fragmentacje pakietów (ze względu na Maximum Transmission Unit, MTU).
- Działa w modelu bezpołączeniowym.
- Przykładowe standardy:
  - IP
  - IPX
  - ICMP
  - ARP
  - DDP

Internet Protocol (IPv4)

- Rola – jednoznaczna adresacja urządzeń i trasowanie pakietów przez wiele sieci.
- Adres IP identyfikuje urządzenie i określa jego logiczną lokalizacje.
- Struktura nagłówka pakietu:



Flags (3b) + Fragment offset (13b) – jeśli datagram to fragment większego komunikatu

Time to live (TTL) – maksymalna liczba przeskoków (hops), którą może pokonać pakiet

Fragmentacja pakietów

- Pierwotny pakiet: 4000B (20B nagłówka + 3980B danych).
- Router przekazuje pakiet do sieci o mniejszym MTU: 1500B.

···	length = 4000	ID = x	fragflag = 0	offset = 0	···

- Pakiet dzielony jest na 3 fragmenty
  (dane: 1480B + 1480B + 1020B = 3980B).
- Offset jest podawany nie w bajtach, ale w bajtach podzielonych przez 8 $\longrightarrow$ redukcja rozmiaru nagłówka:
  - $1480/8 = 185$
  - $(1480 + 1480)/8 = 370$

···	length = 1500	ID = x	fragflag = 1	offset = 0	···

···	length = 1500	ID = x	fragflag = 1	offset = 185	···

···	length = 1040	ID = x	fragflag = 0	offset = 370	···

Zadanie 15

1. Zaloguj ruch przy pomocy programu `wireshark`. Jakie zazwyczaj są ustawione flagi w pakietach?
2. Powtórz ćwiczenie otwierając w międzyczasie jakąś stronę internetową przy użyciu protokołu `https`.
3. Powtórz ćwiczenie w międzyczasie wykonując polecenie `ping -s 4000` na wybrany adres IP.

Czas życia pakietu

W warstwie łącza danych ramki nie krążą w nieskończoność bo jest używany STP.

W warstwie sieciowej wykorzystywany jest mechanizm TTL:

- pole TTL nagłówka IP jest inicjalizowane na np. 64,
- pole TTL jest dekrementowane przez każdy router na drodze pakietu,
- router nie przekazuje dalej pakietu gdy TTL $= 0$.

Dlaczego wartość 64 jest wystarczająca?

Zadanie 16

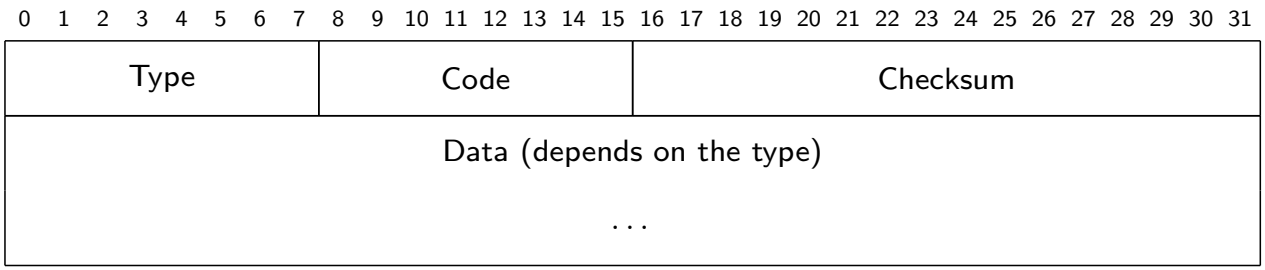
1. Jak działa program `traceroute` (ewentualnie skorzystaj z opcji `-I` lub `-T`)?
2. Zaloguj ruch przy pomocy programu `wireshark` i zbadaj nagłówki pakietów generowanych przez program `traceroute`.

Internet Control Message Protocol (ICMP)

- Protokół diagnostyczny przesyłający informacje o stanie i błędach w sieci.
- Musi być realizowany w każdej implementacji protokołu IP.
- Po co?
  - IP jest protokołem bezpołączeniowym.
  - IP nie ma wbudowanego mechanizmu informacji o sytuacjach wyjątkowych, np. o braku hosta docelowego, błędnym routowaniu, etc.
- Komunikaty ICMP mogą być generowane przez dowolne urządzenie *napotkane* na drodze pakietu (karta sieciowa, router, etc.)
- ICMP funkcjonuje na pograniczu warstw 3 i 4:
  - ICMP realizuje zadania warstwy sieciowej (3),
  - ICMP korzysta z IP do przesyłania komunikatów $\longrightarrow$ może być zaliczany do protokołów warstwy transportowej (4).

Internet Control Message Protocol (ICMP)

- Struktura komunikatu:

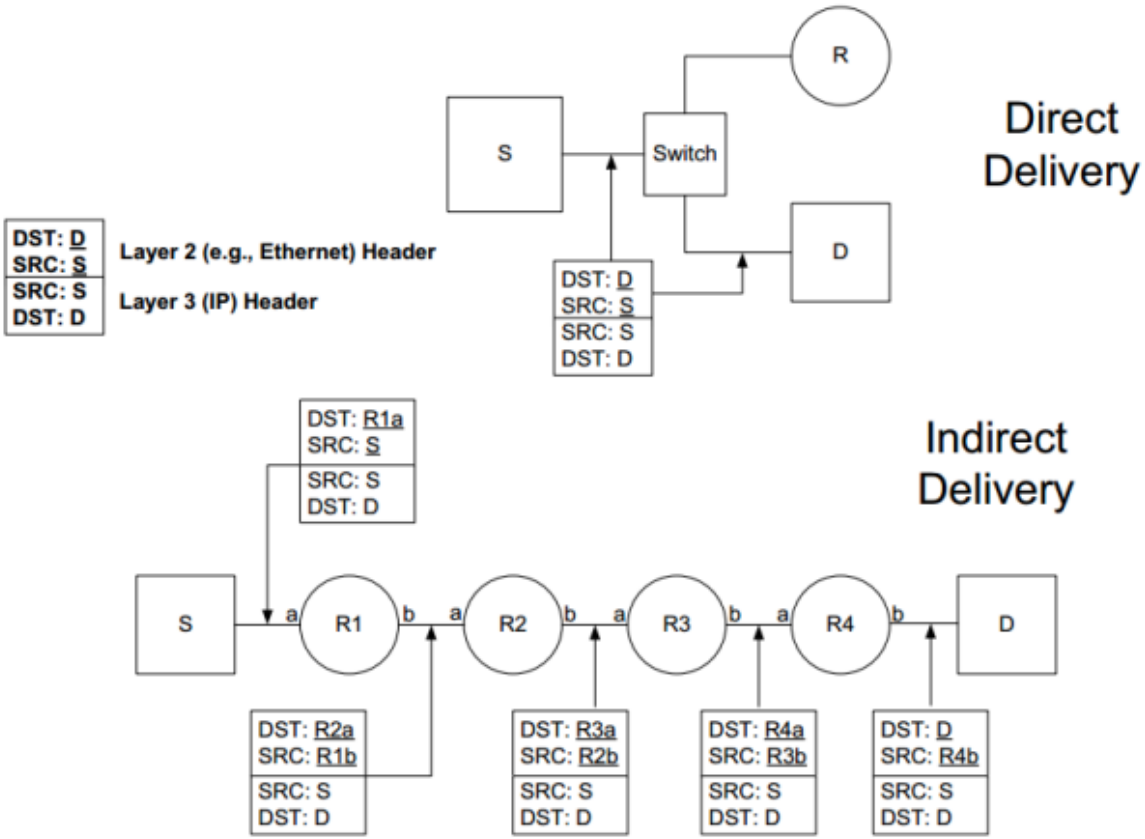


- Typy komunikatów ICMP (link).
- `ping`:
  - Echo Reply (0),
  - Destination Unreachable (3),
  - Echo Request (8).
- `traceroute` $\longrightarrow$ sprawdzanie TTL:
  - Echo Request (8),
  - Time Exceeded (11).

Internet Group Management Protocol (IGMP)

- Zarządzanie grupami multicastowymi w danej sieci:
  - dołączanie/odłączanie od grupy,
  - badanie stanu grupy.
- Komunikaty podobne do komunikatów ICMP.

Komunikacja w sieci globalnej



- Pakiet IP przechodzi przez sieć w (prawie) niezmienionej formie.
- Czas życia ramki to czas transmisji przez sieć lokalną.

Address Resolution Protocol (ARP) (1)

Problem:

- W sieciach IP komputery identyfikowane są przez adresy IP (warstwa 3).
- W sieciach Ethernetowych komputery identyfikowane są poprzez adresy MAC (warstwa 2).
- Skąd wiadomo jak skojarzyć adres MAC z adresem IP danego komputera?

Trywialne rozwiązanie:

- statyczna konfiguracja mapowania MAC–IP,
- nieskalowalne i problematyczne.

Address Resolution Protocol (ARP) (2)

Dynamiczne odwzorowanie adresów IP w adresy MAC
(protokół warstwy 2.5).

Protokół:

- zapytanie: nadawca rozgłasza w sieci fizycznej pytanie o adres MAC urządzenia o adresie IP `X.X.X.X`; pytanie zawiera adres MAC `aa:aa:aa:aa:aa:aa` nadawcy,
- odpowiedź: urządzenie, które dostało zapytanie i którego adres IP to `X.X.X.X` wysyła na `aa:aa:aa:aa:aa:aa` swój adres MAC `bb:bb:bb:bb:bb:bb`,
- (optymalizacja) nadawca zapisuje w prywatnej tablicy ARP-cache odzworowanie `X.X.X.X` do `bb:bb:bb:bb:bb:bb`. Wpisy w ARP-cache są tymczasowe.

Address Resolution Protocol (ARP) (3)

Reverse ARP:

- tłumaczenie MAC na IP,
- obecnie nieużywane.

Proxy ARP:

- używany, gdy niektóre komputery w sieci nie mają zdefiniowanej bramy,
- gdy komputer próbuje wysłać komunikat do komputera spoza sieci, router może odpowiedzieć na zapytanie ARP swoim adresem MAC (adresem MAC interfejsu znajdującego się w tej sieci, z której pochodzi zapytanie),
- router działa jako brama, choć komputer o tym nie wie.

ARP – narzędzia (1)

```
# ip neigh show
192.168.1.1 dev eth0 lladdr 98:fc:11:bc:f1:a0 REACHABLE
# ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=83.1 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=128 ms
^C
--- 192.168.1.100 ping statistics ---
# ip neigh show
192.168.1.100 dev eth0 lladdr 90:72:40:67:24:c7 REACHABLE
192.168.1.1 dev eth0 lladdr 98:fc:11:bc:f1:a0 REACHABLE

(minute pozniej)

# ip neigh show
192.168.1.100 dev eth0 lladdr 90:72:40:67:24:c7 STALE
192.168.1.1 dev eth0 lladdr 98:fc:11:bc:f1:a0 REACHABLE
```

ARP – narzędzia (2)

```
# ip neigh add 192.168.1.101 lladdr aa:bb:cc:dd:ee:ff dev eth0
# ip neigh show
192.168.1.100 dev eth0 lladdr 90:72:40:67:24:c7 REACHABLE
192.168.1.101 dev eth0 lladdr aa:bb:cc:dd:ee:ff PERMANENT
192.168.1.1 dev eth0 lladdr 98:fc:11:bc:f1:a0 DELAY
# ip neigh del 192.168.1.101 dev eth0
# ip neigh show
192.168.1.100 dev eth0 lladdr 90:72:40:67:24:c7 REACHABLE
192.168.1.101 dev eth0  FAILED
192.168.1.1 dev eth0 lladdr 98:fc:11:bc:f1:a0 REACHABLE

# ip neigh flush dev eth0
# ip link set arp off dev eth0
# ip link set arp on dev eth0
```

ARP – narzędzia (3)

```
# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
ap11                     ether   98:fc:11:bc:f1:a0   C                     eth0

# arp -i eth0 -s 192.168.1.101 aa:bb:cc:dd:ee:ff
# arp -i eth0 -d 192.168.1.101

# cat /proc/sys/net/ipv4/neigh/eth0/gc_stale_time
60

# arping -I eth0 192.168.1.100
ARPING 192.168.1.100 from 192.168.1.103 eth0
Unicast reply from 192.168.1.100 [90:72:40:67:24:C7]  106.945ms
Unicast reply from 192.168.1.100 [90:72:40:67:24:C7]  51.238ms
Unicast reply from 192.168.1.100 [90:72:40:67:24:C7]  74.105ms
^CSent 4 probes (1 broadcast(s))
Received 3 response(s)
```

Zadanie 17

1. Podłącz swój komputer (poprzez port `p4p1`) do koncentratora (na zapleczu).
2. Skonfiguruj interfejs `p4p1`, tak by wszystkie komputery w rzędzie działały w jednej sieci (unikalne sieci między rzędami).
3. Zbadaj jak zmienia się tablica ARP, gdy uruchamiasz program `ping` z argumentami będącymi adresami IP komputerów z Twojej sieci i adresami komputerów w innych rzędach (należących do innych sieci).

Dynamic Host Conf. Protocol (DHCP) (1)

Protokół pozwalający urządzeniu na otrzymanie od serwera w (lokalnej) sieci danych do konfiguracji, np.:

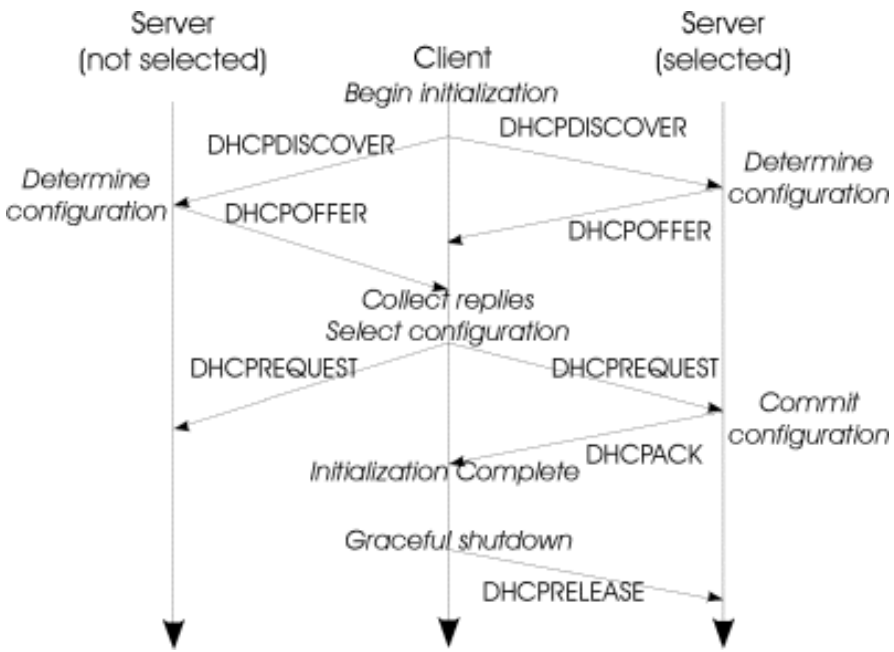
- adres IP, maska,
- adres IP bramy sieciowej,
- wartość MTU,
- nazwa DNS (Domain Name Server),
- adresy serwerów DNS,
- inne: adresy IP serwera SMTP, serwera TFTP, ...

Dynamic Host Conf. Protocol (DHCP) (2)

Sposoby przydzielania adresu z DHCP (możliwe konfiguracje serwera):

- przypisanie na stałe adresu IP do konkretnego urządzenia (po adresie MAC),
- przydział automatyczny z wcześniej zdefiniowanej puli adresów,
- przydział dynamiczny pozwalający urządzeniom na ponowne wykorzystanie poprzednio przydzielonych adresów:
  - pula adresów zdefiniowana jak w przydziale automatycznym,
  - każdy adres jest przydzielany (dzierżawiony) na określony czas,
  - urządzenia starają się *odświeżać* dzierżawę.

Dynamic Host Conf. Protocol (DHCP) (3)



Inne komunikaty:

- `DHCPNAK` – odmowa przydziału parametrów konfiguracji,
- `DHCPDECLINE` – informacja, że adres sieciowy jest już używany,
- `DHCPINFORM` – żądanie przydziału parametrów konfiguracji bez przydziału adresu IP.

DHCP jest protokołem wykorzystującym transakcje!

dhclient – klient DHCP (1)

```
# dhclient -r -v eth0
Killed old client process
Internet Systems Consortium DHCP Client 4.3.4
...
Listening on LPF/eth0/1c:6f:65:82:11:8e
Sending on   LPF/eth0/1c:6f:65:82:11:8e
Sending on   Socket/fallback
DHCPRELEASE on eth0 to 150.254.31.62 port 67 (xid=0xb4d0356)

# dhclient -v eth0
Internet Systems Consortium DHCP Client 4.3.4
...
Listening on LPF/eth0/1c:6f:65:82:11:8e
Sending on   LPF/eth0/1c:6f:65:82:11:8e
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 7 (xid=0x1b07331)
DHCPOFFER from 150.254.31.62
DHCPREQUEST on eth0 to 255.255.255.255 port 67 (xid=0x1b07331)
DHCPACK from 150.254.31.62 (xid=0x1b07331)
bound to 150.254.31.4 -- renewal in 1781 seconds.
```

dhclient – klient DHCP (2)

```
# dhclient eth0
(rownolegle: journalctl [-e][-f] lub tail [-f] /var/log/messages)
# journalctl -f
...
Mar 24 14:44:45 tux dhclient[6363]: Created duid
                          "\000\004q\212W6$\374A\275\237W)\265\375\0138U".
Mar 24 14:44:45 tux dhclient[6363]: DHCPDISCOVER on eth0 to \
                       255.255.255.255 port 67 interval 8 (xid=0x8519f52c)
Mar 24 14:44:45 tux dhclient[6363]: DHCPREQUEST on eth0 to \
                                    255.255.255.255 port 67 (xid=0x8519f52c)
Mar 24 14:44:45 tux dhclient[6363]: DHCPOFFER from 150.254.31.62
Mar 24 14:44:45 tux dhclient[6363]: DHCPACK from 150.254.31.62 (xid=0x8519f52c)
Mar 24 14:44:45 tux NET[6391]: /usr/sbin/dhclient-script : \
                                                  updated /etc/resolv.conf
Mar 24 14:44:47 tux dnsmasq[1272]: reading /etc/resolv.conf
Mar 24 14:44:47 tux dnsmasq[1272]: using nameserver 150.254.30.30#53
Mar 24 14:47:47 tux dnsmasq[1272]: using nameserver 150.254.5.4#53
Mar 24 14:44:47 tux dhclient[6363]: bound to 150.254.31.4 -- \
                                                    renewal in 1577 seconds.
# ip addr show eth0
...
    inet 150.254.31.4/25 brd 150.254.31.127 scope global dynamic eth0
       valid_lft 2898sec preferred_lft 2898sec
```

dhclient – klient DHCP (3)

```
(czasami inna sciezka dostepu, np. /var/lib/dhcp/dhclient.leases)
$ more /var/lib/dhclient/dhclient.leases
default-duid "\000\004q\212W6$\374A\275\237W)\265\375\0138U";
lease {
  interface "eth0";
  fixed-address 150.254.31.4;
  option subnet-mask 255.255.255.128;
  option routers 150.254.31.1;
  option dhcp-lease-time 3600;
  option dhcp-message-type 5;
  option domain-name-servers 150.254.30.30,150.254.5.4;
  option dhcp-server-identifier 150.254.31.62;
  option broadcast-address 150.254.31.127;
  option domain-name "cs.put.poznan.pl";
  renew 5 2017/03/24 14:40:12;
  rebind 5 2017/03/24 15:02:59;
  expire 5 2017/03/24 15:10:29;
}
```

Zadanie 18

1. Przy pomocy programu `wireshark` loguj ruch na interfejsie `p4p1`.
2. Wymuś ponowne przydzielenie adresu z DHCP.
3. Porównaj dane z zalogowanego ruchu z zawartością logów systemowych.
4. Porównaj zawartość plików `/etc/resolv.conf` oraz `/var/lib/dhclient/dhclient.leases` (w innych dystrybucjach ścieżka może się różnić, np. `/var/lib/dhcp/dhclient.leases`).

Zeroconf

Protokół automatycznej konfiguracji urządzeń do pracy w sieci TCP/IP – Zero Configuration Networking:

- tworzenie adresów IP dla urządzeń $\longrightarrow$ `169.254.0.0/16`,
- przypisywanie nazw domenowych,
- wykrywanie urządzeń i usług w sieci, m.in., drukarek, serwerów FTP.

Popularne implementacje:

- Avahi – Linux, BSD,
- Apple Bonjour.

Routing statyczny w Linuksie

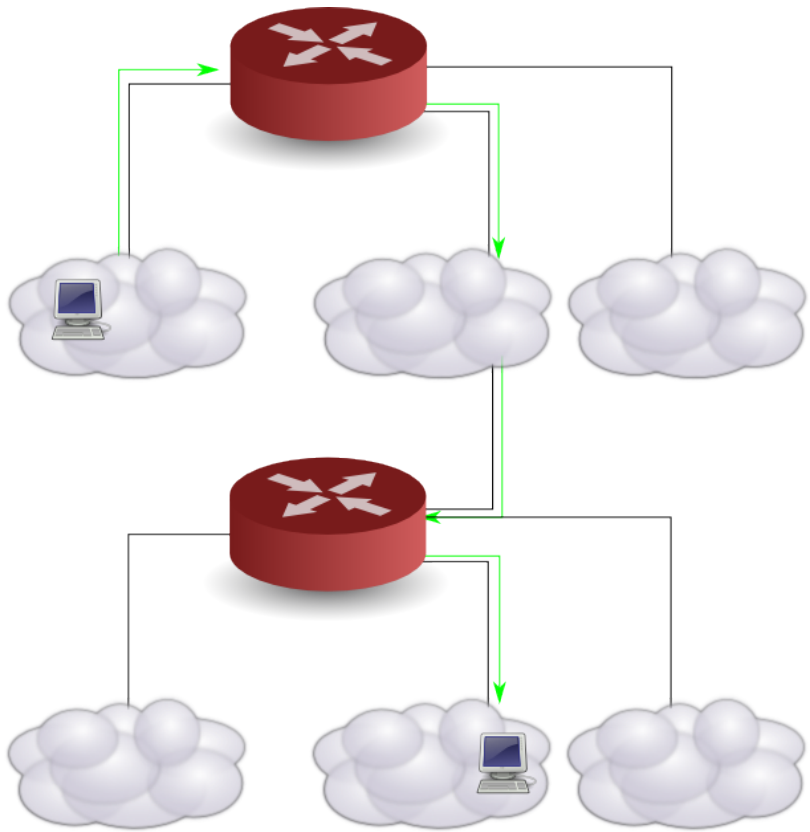
Routing

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

Sieci łączone są przy pomocy routerów.

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

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



Tablica routingu

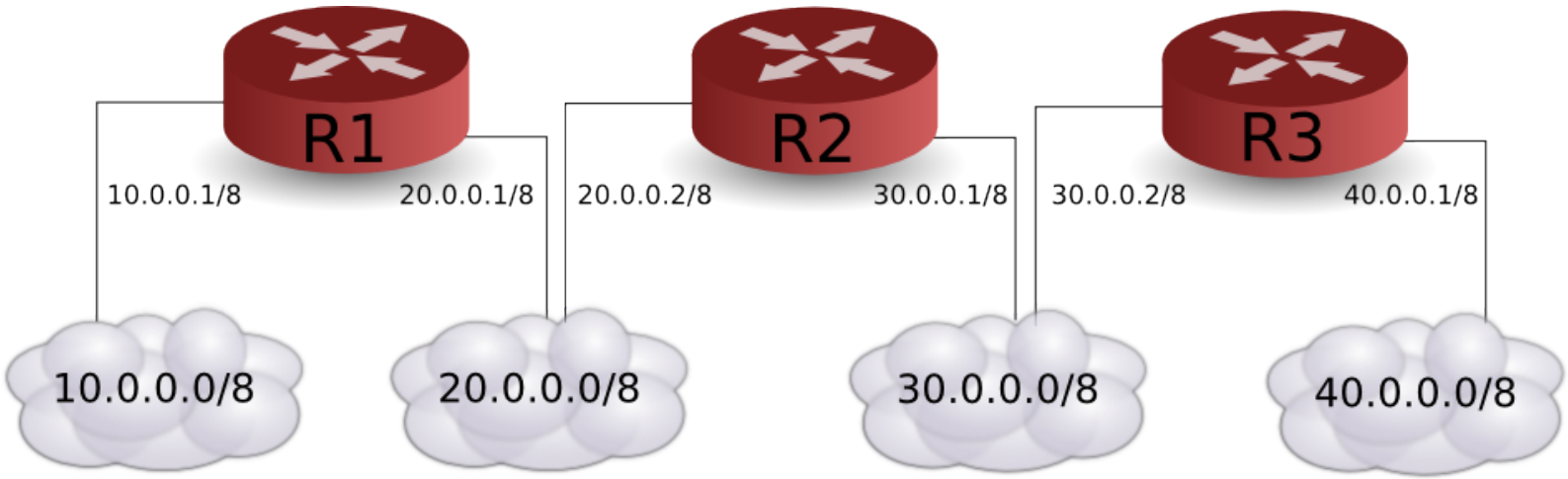
Sieć	Maska	Brama
192.168.1.0	255.255.255.0	10.0.0.1
192.168.2.0	255.255.255.0	10.0.0.2
192.168.3.0	255.255.255.0	10.0.0.3
192.168.4.0	255.255.255.0	10.0.0.4
10.0.0.0	255.0.0.0	-

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

Typy routingu

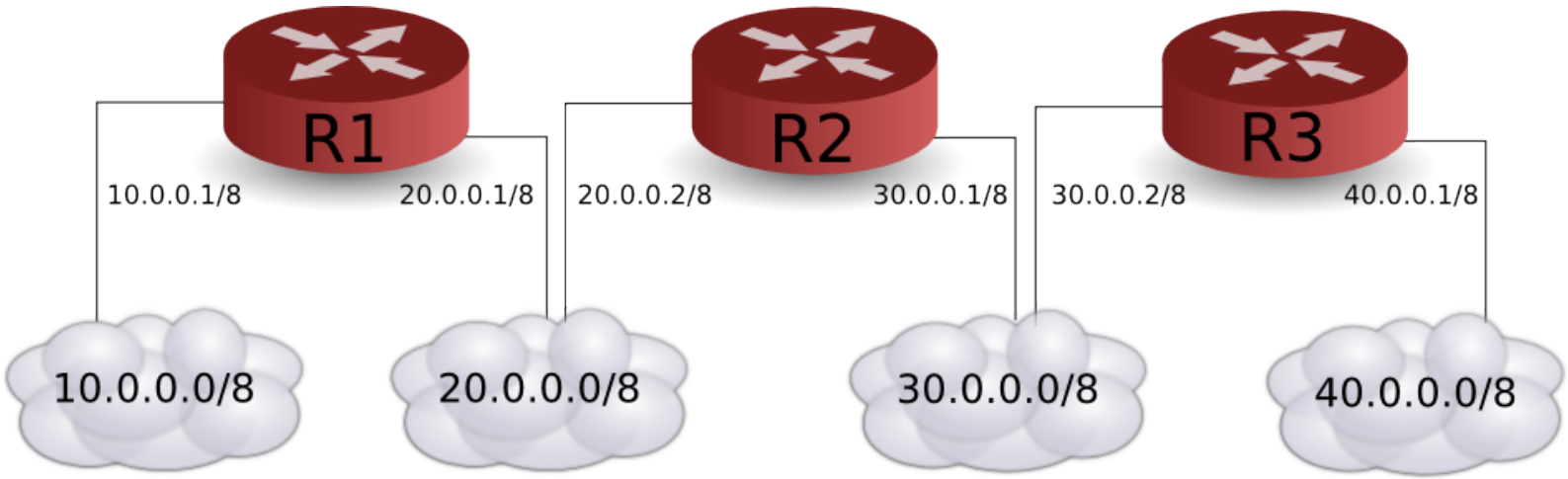
- Routing statyczny (static routing) – zawartość tablicy routingu zdefiniowana przez administratora (na stałe).
- Routing dynamiczny (dynamic routing) – tablica routingu wyznaczona jest przez protokół routingu (np. RIP, OSPF, IS-IS, IGRP) w czasie działania urządzenia.
- Routing sprzętowy – realizowany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

Routing – przykład (1)



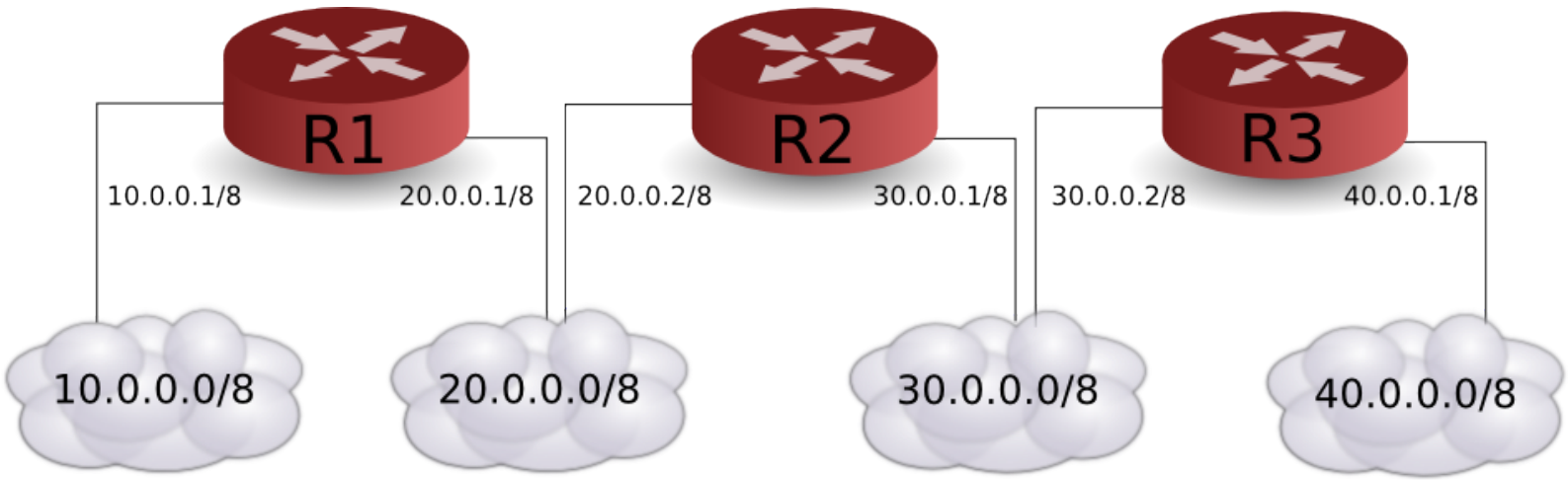
	Sieć	Maska	Brama
R1	10.0.0.0	255.0.0.0	-
	20.0.0.0	255.0.0.0	-
	30.0.0.0	255.0.0.0	20.0.0.2
	40.0.0.0	255.0.0.0	20.0.0.2

Routing – przykład (2)



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

Routing – przykład (3)



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

Konfiguracja routingu statycznego (1)

```
# ip route show
default via 150.254.44.1 dev wlp3s0  proto static  metric 600
150.254.44.0/23 dev wlp3s0  proto kernel  scope link  src 150.254.45.101
                                                      metric 600
# ip addr add 192.168.1.101/24 dev wlp3s0
# ip route
default via 150.254.44.1 dev wlp3s0  proto static  metric 600
150.254.44.0/23 dev wlp3s0  proto kernel  scope link  src 150.254.45.101
                                                      metric 600
192.168.1.0/24 dev wlp3s0  proto kernel  scope link  src 192.168.1.101
# ip route add 172.16.0.0/16 via 192.168.1.111
# ip route
default via 150.254.44.1 dev wlp3s0  proto static  metric 600
150.254.44.0/23 dev wlp3s0  proto kernel  scope link  src 150.254.45.101
                                                      metric 600
172.16.0.0/16 via 192.168.1.111 dev wlp3s0
192.168.1.0/24 dev wlp3s0  proto kernel  scope link  src 192.168.1.101
                                                      metric 600
# ip route add default via 192.168.1.1
# ip route del 172.16.0.0/16
```

Konfiguracja routingu statycznego (2)

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

Procedura wyboru trasy (1)

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

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

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

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

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

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

Procedura wyboru trasy (2)

```
# ip route
default via 150.254.31.1 dev enp0s26u1u2  proto static  metric 100
default via 150.254.44.1 dev wlp3s0  proto static  metric 600
150.254.6.8 via 150.254.44.1 dev wlp3s0  proto dhcp  metric 600
150.254.31.0/25 dev enp0s26u1u2  proto kernel  scope link  src 150.254.31.15
                                                           metric 100
150.254.44.0/23 dev wlp3s0  proto kernel  scope link  src 150.254.45.101
                                                           metric 600
# route -n
Kernel IP routing table
Destination   Gateway       Genmask         Flags Metric Ref Use Iface
0.0.0.0       150.254.31.1  0.0.0.0         UG    100    0     0 enp0s26u1u2
0.0.0.0       150.254.44.1  0.0.0.0         UG    600    0     0 wlp3s0
150.254.6.8   150.254.44.1  255.255.255.255 UGH   600    0     0 wlp3s0
150.254.31.0  0.0.0.0       255.255.255.128 U     100    0     0 enp0s26u1u2
150.254.44.0  0.0.0.0       255.255.254.0   U     600    0     0 wlp3s0
```

Forwarding

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

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

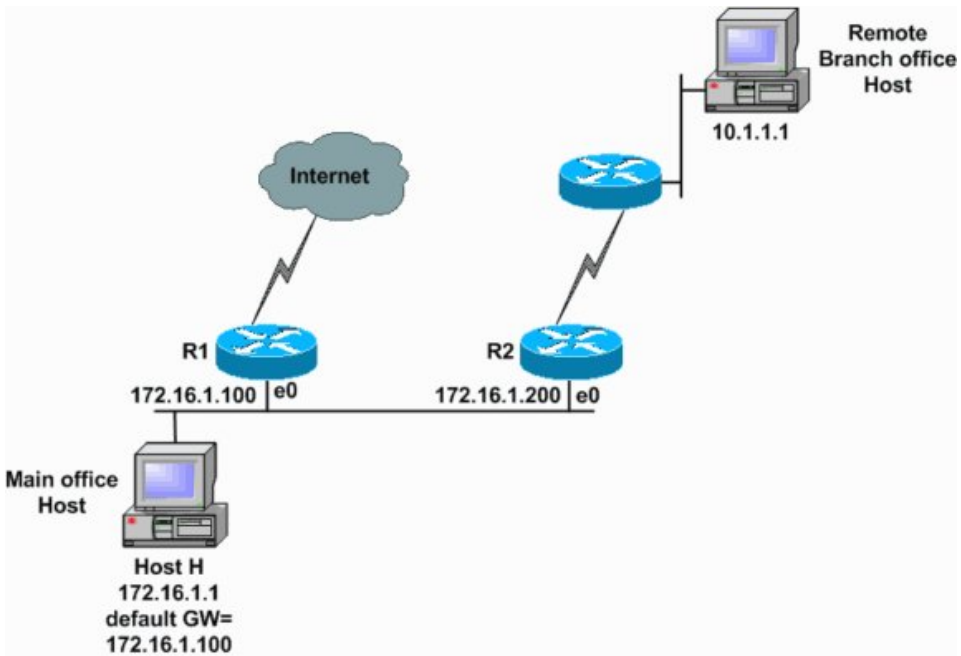
```
# cat /proc/sys/net/ipv4/ip_forward
0
# echo 1 > /proc/sys/net/ipv4/ip_forward
# cat /proc/sys/net/ipv4/ip_forward
1
# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

Zadanie 19

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

Komunikaty redirect

```
(IP hosta: 172.16.1.1)
# ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
From 172.16.1.100: icmp_seq=1 Redirect Host(New nexthop: 172.16.1.200)
64 bytes from 8.8.8.8: icmp_seq=1 ttl=64 time=82.8 ms
From 172.16.1.100: icmp_seq=2 Redirect Host(New nexthop: 172.16.1.200)
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=123 ms
```



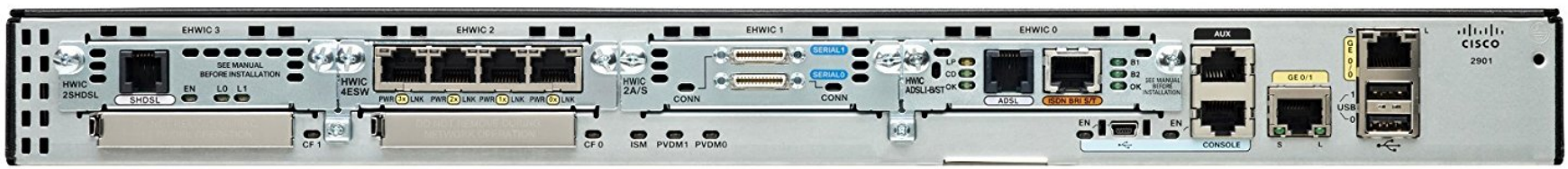
Router R1 przekazuje pakiet od hosta H przez ten sam interfejs (e0), przez który pakiet przyszedł od H.

Router R1 dodatkowo informuje H, że istnieje prostsze połączenie: H $\rightarrow$ R2.

Routing statyczny w urządzeniach Cisco

Routery Cisco

- Bardziej złożone niż przełączniki.
- System operacyjny: IOS.
- Podstawowe wyjścia:
  - FastEthernet, GigaEthernet – podłączenia z przełącznikami,
  - Serial – dawniej podłączenie od providera, łączenie routerów między sobą,
  - Console – konfiguracja,
  - inne: ADSL, USB, porty optyczne, ...
- Modularna budowa $\longrightarrow$ dodatkowe złącza sieci różnej technologii.



Połączenie z konsolą Cisco

Połączenie jest realizowane poprzez port szeregowy komputera:

- komunikacja *jeden bit na raz*, urządzenie `/dev/ttyS0`,
- niebieski kabel z tyłu komputera $\longrightarrow$ wyjście na patch panelu na zapleczu,
- port konsoli Cisco to zwykły port ethernetowy $\longrightarrow$ połączenie niebieskimi kablami (kable proste),
- obsługa:

```
picocom [-b 9600] /dev/ttyS0
minicom [-b 9600] -D /dev/ttyS0
```

Zadanie 20

1. Połącz sąsiednie komputery przy pomocy kabla szeregowego.
2. Sprawdź komunikację między komputerami poprzez użycie programu `picocom [-b 9600] /dev/ttyS0`. Wyjście z programu: `ctrl-a-q`.
3. Sprawdź co się stanie, gdy porty szeregowe połączonych komputerów będą mieć ustawiony różny baud rate.
4. Upewnij się, że niebieskie kable są ponownie podłączone!

Zadanie 21

Zadanie jest rozwiązywane w parach.

1. Podłącz komputer poprzez łącze szeregowe do wyjścia konsolowego jednego z routerów na zapleczu.
2. Wykonaj polecenie `picocom [-b 9600] /dev/ttyS0`. Jeśli połączenie jest właściwe, to coś powinno się pokazać na terminalu (patrz następny slajd).
3. Uwaga: jeśli pojawią się pytania, to odpowiadaj rozsądnie – nie wchodź w `initial configuration`!

Połączenie z konsolą Cisco

```
# picocom /dev/ttyS0
picocom v1.8

port is        : /dev/ttyS0
flowcontrol    : none
baudrate is    : 9600
parity is      : none
databits are   : 8
escape is      : C-a
local echo is  : no
noinit is      : no
noreset is     : no
nolock is      : no
send_cmd is    : sz -vv
receive_cmd is : rz -vv
imap is        :
omap is        :
emap is        : crcrlf,delbs,

Terminal ready
[enter]
Router>
```

Tryby powłoki Cisco

```
Router>

Router> enable
Router#

Router# configure terminal
Enter configuration commands, \
   one per line. End with CNTL/Z.
Router(config)#

Router(config)# do <komenda
                    nizego trybu>

Router(config)# exit
Router# disable
Router>
```

Tryb użytkownika:

- `ping`, `tracert`.

Tryb uprzywilejowany:

- normalnie wymaga hasła,
- dodatkowo `telnet`, `show interfaces`, `show running-config`, `show ip route`.

Tryb konfiguracyjny:

- wiele podtrybów,
- `router`, `interface`, `ip route`.

Podpowiedzi powłoki

```
Router# show h?
hardware  history  hosts  html

Router# show hard[tab]
Router# show hardware
...
ROM: System Bootstrap, Version 12.3(8r)T9, RELEASE SOFTWARE (fc1)
Router uptime is 1 hour, 54 minutes
System returned to ROM by power-on
System image file is "flash:c2801-ipbase-mz.124-1c.bin"
Cisco 2801 (revision 6.0) with 114688K/16384K bytes of memory.
Processor board ID FCZ102422JY
2 FastEthernet interfaces
2 Low-speed serial(sync/async) interfaces
DRAM configuration is 64 bits wide with parity disabled.
191K bytes of NVRAM.
62720K bytes of ATA CompactFlash (Read/Write)

Router# sh ha
...
```

Ważne skróty klawiszowe

```
Router# show ip int
*Jan  1 01:53:02.079: %SYS-5-CONFIG_I: Configured from console by console
[ctrl-r]
Router# show ip interface brief
Interface                  IP-Address      OK? Method Status                Prot
FastEthernet0/0            192.168.0.1     YES manual up                    up
FastEthernet0/1            unassigned      YES unset  administratively down down
Serial0/1/0                unassigned      YES unset  administratively down down
Serial0/1/1                unassigned      YES unset  administratively down down

Router# ping 192.168.0.1 repeat 100000
Type escape sequence to abort.
Sending 100000, 100-byte ICMP Echos to 192.168.0.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[ctrl-alt-6]
Success rate is 100 percent (37/37), round-trip min/avg/max = 1/1/4 ms

Router# configure terminal
Router(config)#interface FastEthernet 0/1
Router(config-if)#
[ctrl-z]
Router#
```

Powłoka – podsumowanie

`enable/disable`	wejście/wyjście do/z trybu uprzywilejowanego
`configure terminal`	wejście do trybu konfiguracji
`exit`	wychodzi o jeden tryb w górę, działa do trybu uprzywilejowanego,
`end/ctrl+z`	cofa się do trybu uprzywilejowanego,
`show running-config`	aktualna konfiguracja (w trybie uprzywilejowanym),
`do <komenda>`	pozwala wykonać komendę z wcześniejszego poziomu, np. komendę `show ip route` w trybie konfiguracji
`ctrl+r`	odświeża bieżącą linię
`?`	podpowiada możliwe komendy i ich składnie
`tab`	autouzupełnia komendy
`ctrl+alt+6`	przerywa wykonywanie polecenia
`no <komenda>`	wykonaj odwrotną akcje do akcji określonej przez komendę, np. aktywacja interfejsu wymaga `no shutdown`
`undebug all`	wyłącza `debug all`

komendy można skracać (póki są jednoznaczne),
np. zamiast `configure terminal` wystarczy `conf t`

Interfejsy routera

```
Router# show interfaces
FastEthernet0/0 is up, line protocol is up
  Hardware is Gt96k FE, address is 0018.1876.e034 (bia 0018.1876.e034)
  Internet address is 192.168.0.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, 100BaseTX/FX
...

Router# show interface FastEthernet 0/0
...

Router# show ip interface brief
Interface              IP-Address      OK? Method Status                Prot
FastEthernet0/0        unassigned      YES unset  administratively down down
FastEthernet0/1        unassigned      YES unset  administratively down down
Serial0/1/0            unassigned      YES unset  administratively down down
Serial0/1/1            unassigned      YES unset  administratively down down
```

Nazwa routera i konfiguracja adresów IP (1)

```
Router# configure terminal
Router(config)# hostname MyCisco
MyCisco(config)#

MyCisco(config)# interface FastEthernet 0/1
MyCisco(config-if)# ip address 192.168.51.1 255.255.255.0
MyCisco(config-if)# no shutdown
MyCisco(config-if)#
*Jan  1 03:20:48.995: %LINK-3-UPDOWN: Interface FastEthernet0/1, \
                                      changed state to up
*Jan  1 03:20:49.995: %LINEPROTO-5-UPDOWN: Line protocol on \
                     Interface FastEthernet0/1, changed state to up

MyCisco(config-if)# do show ip interface brief
Interface          IP-Address      OK? Method Status                Prot
FastEthernet0/0    unassigned      YES unset  administratively down down
FastEthernet0/1    192.168.51.1    YES manual up                    up
Serial0/1/0        unassigned      YES unset  administratively down down
Serial0/1/1        unassigned      YES unset  administratively down down
```

Nazwa routera i konfiguracja adresów IP (2)

Łączenie przy pomocy kabla szeregowego (grube niebieskie kable):

- końcówki kabli wyglądają tak samo, ale różnią się!
- DCE (Data Circuit-terminating Equipment) – strona ustalająca częstotliwość transmisji,
- DTE (Data Terminal Equipment) – strona dostosowująca się.

```
MyCisco(config)# interface Serial 0/1/0
MyCisco(config-if)# ip address 10.1.0.1 255.0.0.0
MyCisco(config-if)# clock rate ?
...
MyCisco(config-if)# clock rate 128000
MyCisco(config-if)# no shutdown

OtherCisco(config)# interface Serial 0/1
OtherCisco(config-if)# ip address 10.222.0.1 255.0.0.0
OtherCisco(config-if)# no shutdown
```

Nazwa routera i konfiguracja adresów IP (2)

```
MyCisco# show controllers Serial 0/1/0
Interface Serial0/1/0
Hardware is GT96K
DCE V.35, clock rate 128000
idb at 0x62B17CA0, driver data structure at 0x62B1A064
...
```

```
MyCisco# show ip interface brief
Interface        IP-Address      OK? Method Status                Prot
FastEthernet0/0  unassigned      YES unset  administratively down down
FastEthernet0/1  192.168.51.1    YES manual up                    up
Serial0/1/0      10.1.0.1        YES manual up                    up
Serial0/1/1      unassigned      YES unset  administratively down down
```

Nazwa routera i konfiguracja adresów IP (3)

```
MyCisco# ping 192.168.51.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.51.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

MyCisco# ping 10.222.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.222.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms

MyCisco# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID    Local Intrfce     Holdtme    Capability  Platform  Port ID
OtherCisco   Ser 0/1/0          171         R S I      2611XM    Ser 0/0
```

Konfiguracja routingu statycznego (1)

```
MyCisco# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
...
Gateway of last resort is not set
C    10.0.0.0/8 is directly connected, Serial0/1/0
C    192.168.51.0/24 is directly connected, FastEthernet0/1

MyCisco(config)# ip route 172.16.0.0 255.255.0.0 10.222.0.1
MyCisco(config)# ip route 0.0.0.0 0.0.0.0 192.168.51.17
MyCisco(config)# do show ip route
...
Gateway of last resort is 192.168.51.17 to network 0.0.0.0
S    172.16.0.0/16 [1/0] via 10.222.0.1
C    10.0.0.0/8 is directly connected, Serial0/1/0
C    192.168.51.0/24 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 192.168.51.17
```

Konfiguracja routingu statycznego (2)

```
MyCisco# traceroute 172.16.32.11 numeric
Type escape sequence to abort.
Tracing the route to 10.222.0.1

  1 10.222.0.1    8 msec 9 msec 8 msec
  2 172.16.32.11  8 msec 9 msec 9 msec

MyCisco(config)# no ip route 172.16.0.0 255.255.0.0
MyCisco(config)# do show ip route
...
Gateway of last resort is 192.168.51.17 to network 0.0.0.0
C    10.0.0.0/8 is directly connected, Serial0/1/0
C    192.168.51.0/24 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 192.168.51.17
```

Konfiguracja routingu statycznego (3)

```
MyCisco# show ip interface brief
Interface        IP-Address      OK? Method Status                Prot
FastEthernet0/0  10.0.3.15       YES manual up                    up
FastEthernet0/1  172.16.16.172   YES manual up                    up
MyCisco# show ip route
...
Gateway of last resort is not set
C    172.16.0.0/16 is directly connected, FastEthernet0/1
     172.21.0.0/16 is variably subnetted, 3 subnets, 3 masks
S       172.21.0.0/28 [1/0] via 172.16.0.28
S       172.21.0.0/24 [1/0] via 172.16.0.24
S       172.21.0.0/16 [1/0] via 172.16.0.16
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.3.0/24 is directly connected, FastEthernet0/0
S       10.0.0.0/16 [1/0] via 10.0.3.127
```

Gdy w tablicy routingu są sieci o maskach innych niż wynikają z klas, do których by normalnie przynależały, wpisy są odpowiednio grupowane.

Dwa adresy IP na jednym interfejsie (1)

```
MyCisco(config)# interface FastEthernet 0/1
MyCisco(config-if)# ip address 192.168.51.1 255.255.255.0
MyCisco(config-if)# ip address 10.13.4.1 255.255.0.0 secondary
MyCisco(config-if)# no shutdown

MyCisco# show ip interface
FastEthernet0/1 is up, line protocol is up
  Internet address is 192.168.51.1/24
  Broadcast address is 255.255.255.255
  ...
  Secondary address 10.13.4.1/24
  ...
MyCisco# show ip route
...
C    192.168.51.0/16 is directly connected, FastEthernet0/1
C    10.13.0.0/16 is directly connected, FastEthernet0/1
```

Adresów secondary może być bardzo wiele.

Dwa adresy IP na jednym interfejsie (2)

`show ip interface brief` nie pokaże adresów secondary!

```
MyCisco# show ip interface brief
Interface        IP-Address     OK? Method Status                Prot
FastEthernet0/0  unassigned     YES unset  administratively down down
FastEthernet0/1  192.168.51.1   YES manual up                    up
Serial0/1/0      unassigned     YES unset  administratively down down
Serial0/1/1      unassigned     YES unset  administratively down down
```

Zadanie 22

1. Wraz z koleżankami i kolegami połącz routery w łańcuch poprzez porty ethernetowe (kablami krosowanymi – czarnymi).
2. Nadaj interfejsom odpowiednie adresy, zbadaj połączenia do sąsiednich routerów przy pomocy poleceń `ping` i `show cdp neighbors`.
3. Dodaj routing statyczny do innych sieci.
4. Zbadaj połączenia przy pomocy polecenia `traceroute <adres> numeric`.
5. Gdy starczy czasu:
   - Użyj portów szeregowych do stworzenia dodatkowych połączeń między routerami (np. w celu zamknięcia łańcucha).
   - Podłącz routery do przełączników (jeden przełącznik na każdą sieć), a do przełącznika podłącz komputery z Twojego rzędu.
   - Skonfiguruj adresację oraz routing na komputerach i upewnij się, że komunikacja z innymi komputerami/routerami działa.
   - Wyłap pakiety CDP przy pomocy programu `wireshark`.

Routing dynamiczny w urządzeniach Cisco

Prot. routingu dynamicznego $\rightarrow$ zasięg (1)

Protokoły routingu wewnętrznego (ang. interior gateway protocols, IGP):

- wykorzystywane w ramach pojedynczego systemu autonomicznego (ang. autonomous system) – grupy sieci, w ramach której utrzymywany jest spójny system trasowania,
- lista systemów autonomicznych (obecnie około 84000) (link),
- przykładowe protokoły:
  - IGRP/EIGRP (Interior Gateway Routing Protocol/Enhanced IGRP),
  - OSPF (Open Shortest Path First),
  - RIP (Routing Information Protocol),
  - IS-IS (Intermediate System to Intermediate System).

Prot. routingu dynamicznego $\rightarrow$ zasięg (2)

Protokoły routingu zewnętrznego (ang. exterior gateway protocols, EGP):

- wykorzystywane między systemami autonomicznymi (identyfikowanymi przy pomocy ASN. ang. autonomous system number),
- przykładowe protokoły:
  - EGP (Exterior Gateway Protocol) – obecnie wycofywany,
  - BGP (Border Gateway Protocol).

Prot. routingu dynamicznego $\longrightarrow$ działanie (1)

Protokoły stanu łącza (ang. link-state):

- (wszystkie) routery wymieniają się informacjami o stanie połączeń (topologii sieci),
- każdy router na własną rękę oblicza najkrótsze ścieżki do każdego celu i na tej podstawie tworzy tablicę routingu,
- wykorzystywany algorytm Dijkstry (link),
- przykładowe protokoły: OSPF, IS-IS.

Prot. routingu dynamicznego $\longrightarrow$ działanie (2)

Protokoły wektora odległości (ang. distance vector):

- sąsiednie routery wymieniają się gotowymi tablicami routingu wraz z metrykami (odległością do sieci/kosztem transmisji),
- każdy router przetwarza otrzymane komunikaty i wybiera *najtańszą* ścieżkę do danego celu (sieci), po czym wymienia się tą informacją z sąsiadami,
- najczęściej wykorzystywany algorytm Bellmana-Forda (link) – wyznaczanie najkrótszych ścieżek każdy do każdego,
- przykładowe protokoły: RIP, IGRP.

Prot. routingu dynamicznego → działanie (3)

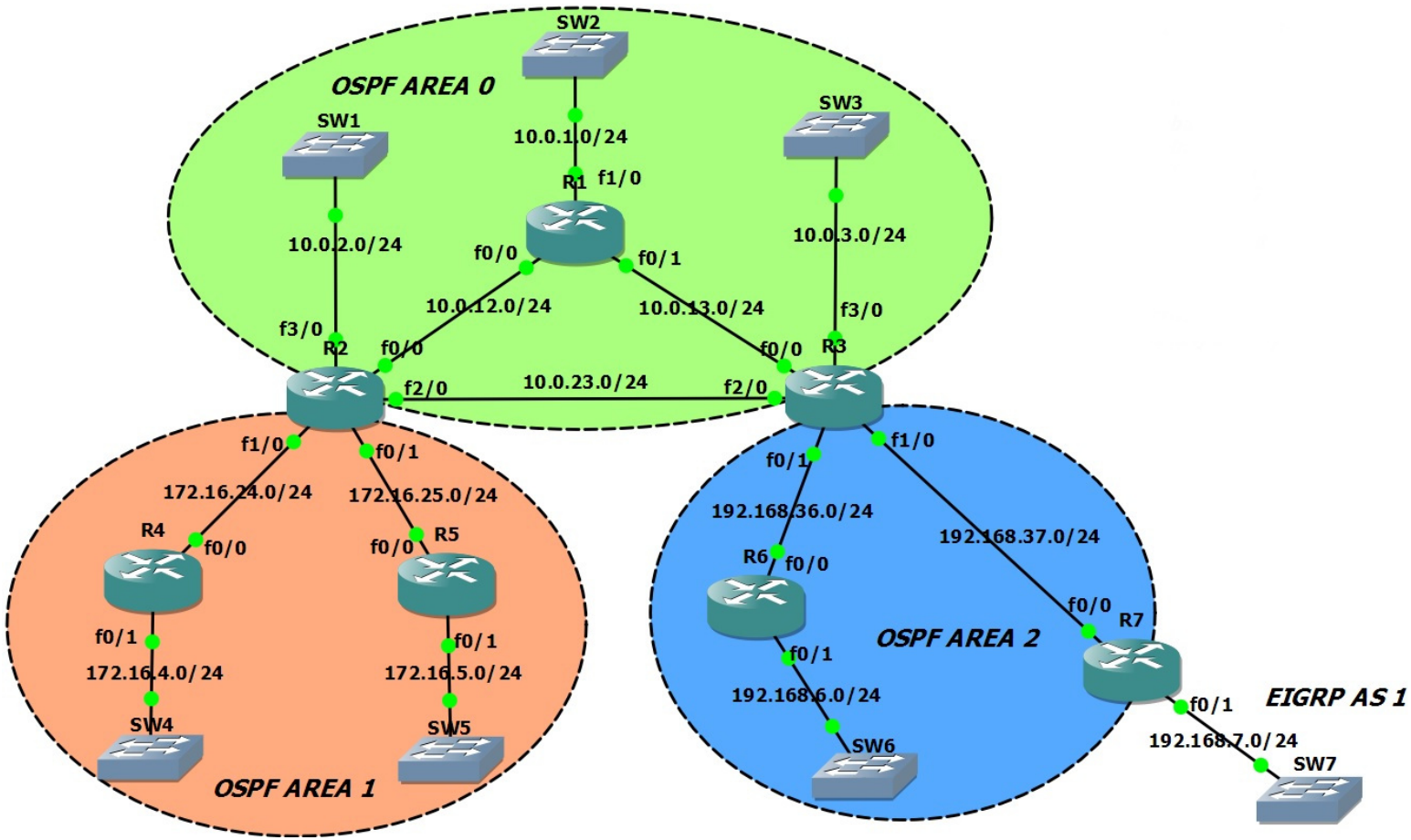
Protokoły wektora ścieżki (ang. path vector):

- routery wymieniają się ze sobą informacjami o ścieżkach, którymi można dotrzeć do danego celu,
- router przekazując ścieżkę dalej, dodaje siebie na jej koniec,
- routery ignorują otrzymane ścieżki jeśli same w niej występują (by uniknąć pętli),
- tablica routingu zawiera całą ścieżkę do celu, a nie tylko następny krok (może istnieć wiele ścieżek do jednego celu),
- wykorzystywany do routingu zewnętrznego:
  - pojedynczym elementem ścieżki jest nie router, ale system autonomiczny,
  - różnorodne polityki decydują, która ścieżka jest bardziej korzystna (np. porozumienia biznesowe),
- przykładowe protokoły: BGP.

Open Shortest Path First (OSPF) (1)

- Zastosowanie – protokół routingu wewnętrznego, tj. ograniczony do jednego systemu autonomicznego.
- Działa w sieciach do 500 routerów.
- Protokół stanu łącza w ramach pojedynczego *obszaru* (*area*):
  - każdy router buduje pełen model topologii sieci,
  - komunikacja w ramach obszaru przy pomocy multicastu.
- Protokół wektora odległości między różnymi obszarami:
  - desygnowany router, który przekazuje listę gotowych tras routerom spoza obszaru.

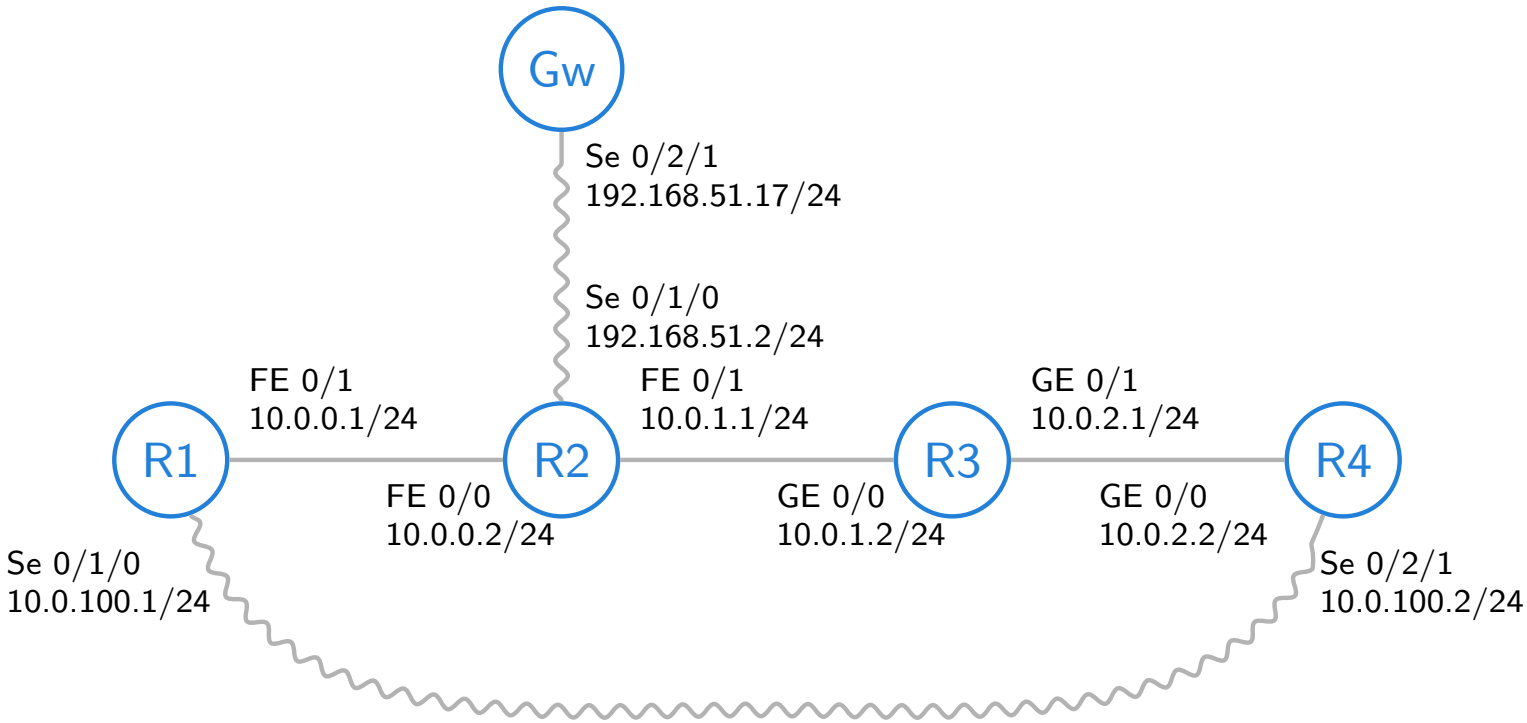
Open Shortest Path First (OSPF) (2)



Open Shortest Path First (OSPF) (3)

1. Routery nawiązują relację z sąsiadami wymieniając wiadomości *Hello*.
2. Routery wymieniają wiadomości *Database Descriptions (DD)* $\longrightarrow$ dowiadują się o istniejących połączeniach.
3. Po każdej wykrytej zmianie stanu łącza router (epidemicznie) rozgłasza komunikat *Link-State Update (LSU)* z opisem zmian.
4. Zapobieganie zalewania sieci pakietami LSU – w jednej domenie rozgłoszeniowej wybiera się router desygnowany i zapasowy router desygnowany.

Rozważana sieć



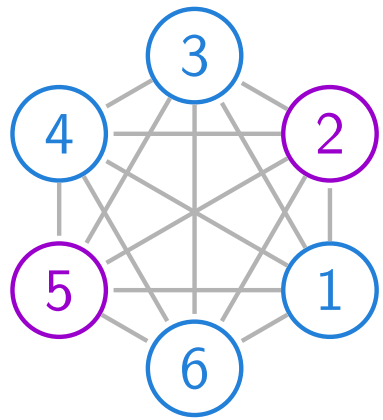
R1-R4 – wspólny obszar OSPF.

Gw – wyjście na świat (włączane później).

Konfiguracja OSPF (1)

```
R3(config)# router ospf 1
R3(config-router)# network 10.0.0.0 0.0.255.255 area 0
R3# show ip ospf neighbor
Neighbor ID  Pri   State      Dead Time  Address    Interface
10.0.2.2       1   FULL/DR    00:00:32   10.0.2.2   GigabitEthernet0/1
10.0.1.1       1   FULL/BDR   00:00:38   10.0.1.1   GigabitEthernet0/0

R4# show ip ospf neighbor
Neighbor ID  Pri   State      Dead Time  Address    Interface
10.0.2.1       1   FULL/BDR   00:00:35   10.0.2.1   GigabitEthernet0/0
10.0.0.1       0   FULL/  -   00:00:37   10.0.100.1 Serial0/2/1
```

- Identyfikator routera – jeden z adresów IP.
- Dwa wyróżnione routery na jedną sieć (domenę rozgłoszeniową): Designated Router, Backup Designated Router.



Konfiguracja OSPF (2)

```
R3# show ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.0.2.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.0.0 0.0.255.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.0.2.2             110      00:11:19
    10.0.1.1             110      00:11:19
  Distance: (default is 110)
```

- Informacja o uruchomionych procesach protokołów.
- Ważne: *wzorzec* sieci OSPF, źródła informacji OSPF.

Konfiguracja OSPF (3)

```
R3# show ip ospf data router
            OSPF Router with ID (10.0.2.1) (Process ID 1)
                Router Link States (Area 0)
  LS age: 1042
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 10.0.0.1
  Advertising Router: 10.0.0.1
  LS Seq Number: 8000000E
  Checksum: 0x851B
  Length: 60
  Number of Links: 3

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 10.0.2.2
     (Link Data) Router Interface address: 10.0.100.1
      Number of TOS metrics: 0
       TOS 0 Metrics: 7812

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 10.0.100.0
     (Link Data) Network Mask: 255.255.255.0
      Number of TOS metrics: 0
       TOS 0 Metrics: 7812

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 10.0.0.2
     (Link Data) Router Interface address: 10.0.0.1
      Number of TOS metrics: 0
       TOS 0 Metrics: 10
```

- Informacja o wymienianych komunikatach OSPF (tu wiadomość od R1).
- Pojedynczy wpis dla sieci 10.0.0.0/24 (Eth.).
- Dwa wpisy dla sieci 10.0.100.0/24 (Serial):
  - połączenie punkt-punkt, bo połączenie Serial (nie ma przełączników typu Serial),
  - połączenie do sieci 10.0.100.0/24.

Konfiguracja OSPF (4)

```
R3# show ip ospf database
            OSPF Router with ID (10.0.2.1) (Process ID 1)
                Router Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum Link count
10.0.0.1        10.0.0.1        125         0x80000003 0x00DAFE 3
10.0.1.1        10.0.1.1        1061        0x80000003 0x004E89 2
10.0.2.1        10.0.2.1        1065        0x80000003 0x008051 2
10.0.2.2        10.0.2.2        124         0x80000003 0x005A16 3

                Net Link States (Area 0)
Link ID         ADV Router      Age         Seq#       Checksum
10.0.0.2        10.0.1.1        1064        0x80000001 0x007793
10.0.1.2        10.0.2.1        1065        0x80000001 0x007B8B
10.0.2.2        10.0.2.2        1076        0x80000001 0x008181
```

Podsumowanie informacji o:

1. routerach w naszym obszarze,
2. wyróżnionych routerach w poszczególnych segmentach obszaru (`10.0.1.1` to ID wyróżnionego routera dla segmentu pierwszego; jego adres w tym segmencie to `10.0.0.2`).

Konfiguracja OSPF (5)

```
R3# show ip route
*Jan  1 01:36:57.487:
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       ...
Gateway of last resort is not set
     10.0.0.0/24 is subnetted, 4 subnets
C       10.0.2.0 is directly connected, GigabitEthernet0/1
O       10.0.0.0 [110/2] via 10.0.1.1, 00:00:38, GigabitEthernet0/0
C       10.0.1.0 is directly connected, GigabitEthernet0/0
O       10.0.100.0 [110/65] via 10.0.2.2, 00:00:38, GigabitEthernet0/1
```

Metryki wykorzystywane do wyboru trasy:

1. dystans administracyjny (ang. administrative distance, link) – stopień *wiarygodności* informacji o połączeniu:
   - 1 – trasa statyczna,
   - 5-200 – trasa dynamiczna, wartość zależna od protokołu, domyślnie 110 dla OSPF, 120 dla RIP,
   - 255 – trasa nieznana,
2. koszt – suma po połączeniach kosztów liczonych jako 100Mb/prędkość łącza (uwaga: czasami 1Gb/prędkość łącza).

Ujednolicanie metryki kosztów

```
R3(config)# router ospf 1
R3(config-router)# auto-cost reference-bandwidth 1000
R3# show ip route
...
Gateway of last resort is not set
     10.0.0.0/24 is subnetted, 4 subnets
C       10.0.2.0 is directly connected, GigabitEthernet0/1
O       10.0.0.0 [110/20] via 10.0.1.1, 00:00:15, GigabitEthernet0/0
C       10.0.1.0 is directly connected, GigabitEthernet0/0
O       10.0.100.0 [110/648] via 10.0.2.2, 00:00:15, GigabitEthernet0/1
R3# show ip ospf interface
GigabitEthernet0/1 is up, line protocol is up
  Internet Address 10.0.2.1/24, Area 0
  Process ID 1, Router ID 10.0.2.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 10.0.2.2, Interface address 10.0.2.2
  Backup Designated router (ID) 10.0.2.1, Interface address 10.0.2.1
  ...
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 10.0.1.2/24, Area 0
  Process ID 1, Router ID 10.0.2.1, Network Type BROADCAST, Cost: 10
```

Ręczna zmiana wartości kosztu

```
R4# show ip route
...
O       10.0.1.0 [110/11] via 10.0.2.1, 00:00:15, GigabitEthernet0/0
R4# configure terminal
R4(config)# interface GigabitEthernet 0/0
R4(config-if)# bandwidth 25
R4(config-if)# do show ip route
...
O       10.0.1.0 [110/50] via 10.0.2.1, 00:00:15, GigabitEthernet0/0
R4(config-if)# ip ospf cost 17
R4(config-if)# do show ip route
...
O       10.0.1.0 [110/27] via 10.0.2.1, 00:00:15, GigabitEthernet0/0
```

Do sieci 10.0.1.0/24 (transmisja przez 10.0.100.0/24 zawsze droższa):

1. łącze 1Gb/s $\rightarrow$ 100 Mb/s: $\frac{1000}{1000} + \frac{1000}{100} = 11$,
2. łącze 25Mb/s $\rightarrow$ 100 Mb/s: $\frac{1000}{25} + \frac{1000}{100} = 50$,
3. łącze o koszcie 17 $\rightarrow$ 100 Mb/s: $17 + \frac{1000}{100} = 27$.

Dodawanie domyślnej bramy

```
R2(config)# ip route 0.0.0.0 0.0.0.0 192.168.51.17
R2(config)# default-information originate

R3# show ip route
...
Gateway of last resort is 10.0.1.1 to network 0.0.0.0
     10.0.0.0/24 is subnetted, 4 subnets
C       10.0.2.0 is directly connected, GigabitEthernet0/1
O       10.0.0.0 [110/20] via 10.0.1.1, 00:07:35, GigabitEthernet0/0
C       10.0.1.0 is directly connected, GigabitEthernet0/0
O       10.0.100.0 [110/648] via 10.0.2.2, 00:07:35, GigabitEthernet0/1
O*E2 0.0.0.0/0 [110/10] via 10.0.1.1, 00:03:21, GigabitEthernet0/0
R3# show ip ospf database
...
                Type-5 AS External Link States
Link ID         ADV Router      Age         Seq#       Checksum Tag
0.0.0.0         10.0.1.1        417         0x80000001 0x002960 1
```

OSPF – podsumowanie komend (1)

Konfiguracja OSPF – tryb konfiguracji (`Router(config)#`)

`router ospf <pid>`	wejście do trybu konfiguracji procesu OSPF
`network <IP> <odw. maska> \ area <obszar>`	dodanie sieci pasujące do wzorca do OSPF, aktywacja wymiany tras; pierwszy wpis włącza proces OSPF
`auto-cost reference-bandwidth \ <prędkość w Mb/s>`	ustawia referencyjną prędkość używaną do automatycznego obliczania kosztów połączenia
`default-information originate`	włącza redystrybucję trasy domyślnej
`redistribute <co rozgłaszać>`	włącza rozgłaszanie tras podanego typu: połączonych (`connected subnets`), statycznych (`static subnets`), z innego procesu/protokołu routingu (`eigrp subnets`)

Ręczne ustalanie kosztu łącza – tryb konfiguracji interfejsu (`Router(config-if)#`)

`bandwidth <prędkość>`	zmiana postrzeganej przez protokoły routingu prędkości
`ip ospf cost <koszt>`	sztuczna zmiana kosztu danego łącza dla OSPF

OSPF – podsumowanie komend (2)

Diagnostyka – tryb uprzywilejowany (`Router#`)

`show ip protocols`	informacje o działających procesach prot.
`show ip ospf`	informacje o działających procesach OSPF
`show ip ospf interface`	informacje o procesach OSPF dla poszczególnych interfejsów
`show ip ospf neighbor`	informacje o sąsiadach
`show ip ospf database`	podsumowane dane o połączeniach
`show ip ospf database network`	podsumowane dane o sieciach
`show ip ospf database router`	pełne dane o połączeniach

Zadanie 23

1. Wraz z koleżankami i kolegami połącz 5-6 routerów w pętlę, tak by były osobne sieci między każdymi sąsiednimi routerami.
2. Na każdym routerze uruchom proces OSPF, tak by wszystkie routery pracowały w ramach jednego obszaru OSPF.
3. Zbadaj jak wyglądają tablice routingu na routerach i którędy wędrują pakiety do różnych sieci (polecenie `traceroute`).
4. Ujednolić koszty przesyłu na wszystkich połączeniach (1000 Mb/s).
5. Dodaj/usuń połączenia w sieci i zobacz jak się zmienia tablica routingu i trasy pakietów.

Sieci VLAN

Problem

W dużych sieciach lokalnych (liczących kilkaset urządzeń), narzut związany z ramkami wysyłanymi na adres rozgłoszeniowy (`ff:ff:ff:ff:ff:ff`) zaczyna być zauważalny.

Rozwiązanie: pogrupować urządzenia i tworzyć oddzielne sieci dla każdej grupy urządzeń. Wtedy też łatwiej filtrować ruch w takiej sieci. Grupowanie najczęściej uwzględnia rodzaj urządzenia lub jego przeznaczenie:

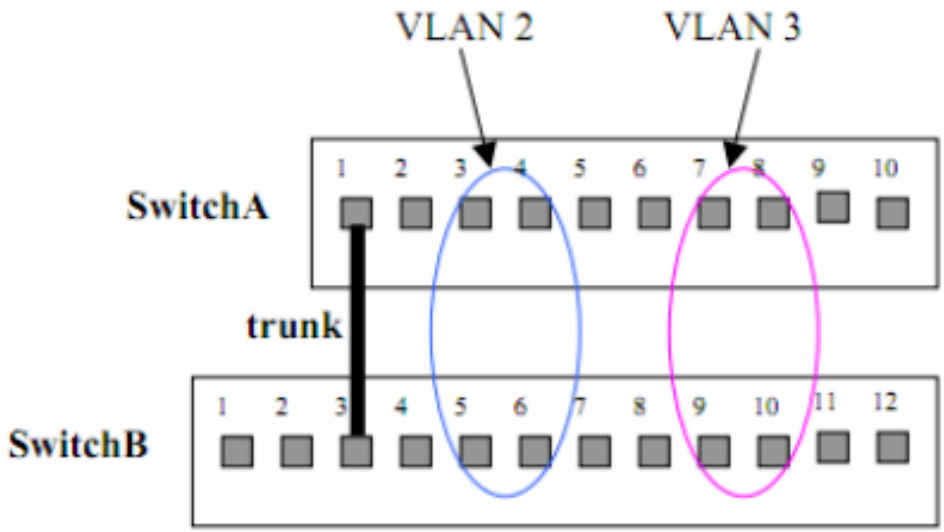
- komputery pracowników różnych działów,
- drukarki,
- serwery,
- telefony IP,
- ...

Nie chcemy mieć oddzielnej infrastruktury dla każdej grupy urządzeń.

Sieci VLAN (Virtual LAN) (1)

- Sieć VLAN to wydzielona logicznie sieć urządzeń w ramach innej, większej sieci fizycznej (mogącej obejmować wiele urządzeń sieciowych).
- VLANy działają na warstwie 2 modelu OSI $\rightarrow$ przełączniki.
- Urządzenia w ramach jednego VLANu mogą się komunikować między sobą jakby były podłączone do pojedynczego przełącznika; ruch między różnymi VLANami jest odseparowany (różne domeny rozgłoszeniowe).
- Komunikacja między VLANami wymaga użycia routera.
- Zalety:
  - ograniczenie ruchu rozgłoszeniowego,
  - łatwiejsze dostosowywanie struktury sieci do struktury organizacyjnej instytucji,
  - większe bezpieczeństwo.

Sieci VLAN (Virtual LAN) (2)



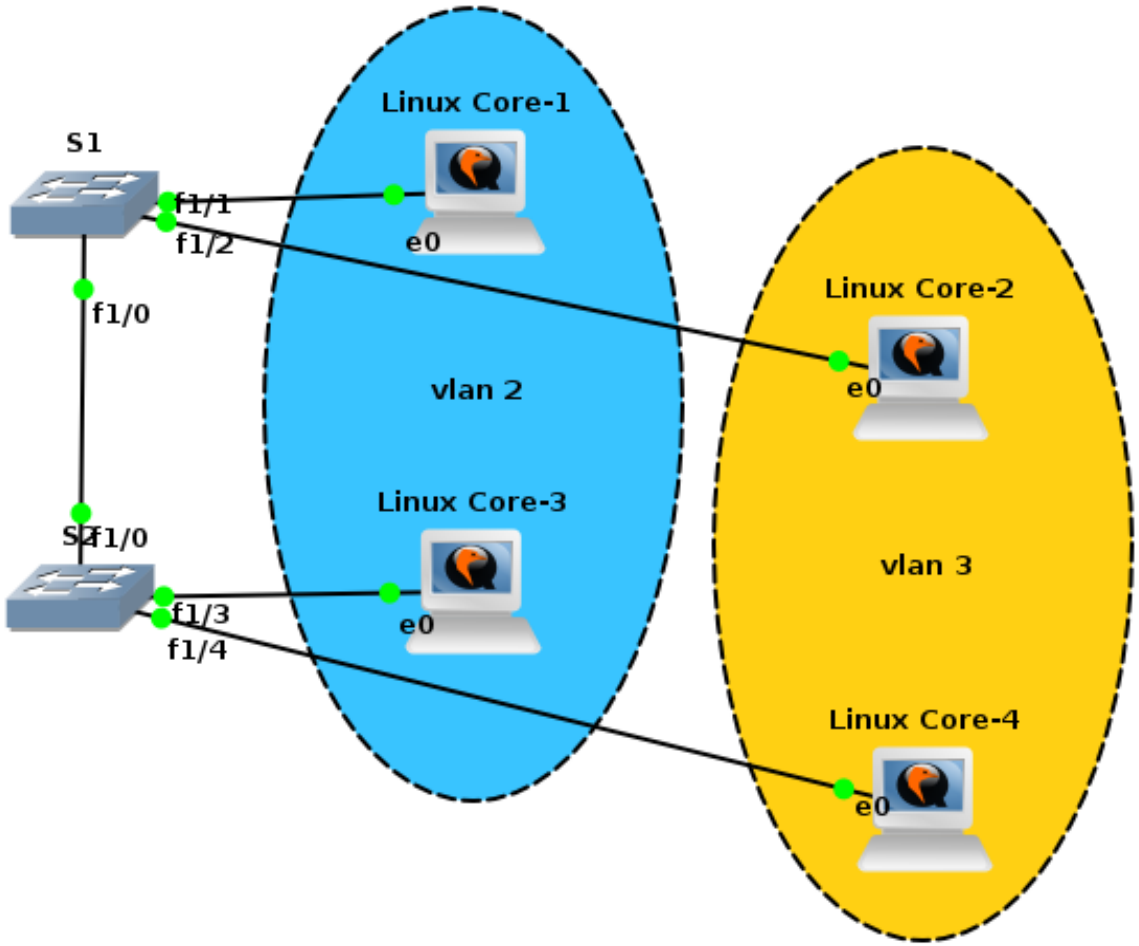
VLANy identyfikowane są liczbami całkowitymi.

Przypisywanie VLANu do portu: statycznie lub dynamicznie.

Przekazywanie ramek różnych VLANów między przełącznikami wymaga połączenia typu trunk:

- do ramek dodawana jest 12-bitowa liczba identyfikująca VLAN nadawcy,
- przełącznik docelowy usuwa etykietę przed wysłaniem ramki do portu adresata.

Rozważana sieć



Sieci VLAN:

- `vlan 2` – staff,
- `vlan 3` – students.

Adresacja:

- L1: `10.0.0.1/24`,
- L2: `10.0.100.2/24`,
- L3: `10.0.0.3/24`,
- L4: `10.0.100.4/24`.

Między S1 i S2 jest połączenie typu trunk.

W praktyce najpierw obmyślane są VLANy i adresacja projektowana jest pod nie.

Przypisywanie interfejsów do VLANów (1a)

```
S1(config)# vlan 2
S1(config-vlan)# name staff
S1(config-vlan)# vlan 3
S1(config-vlan)# name students
S1(config-vlan)# state ?
  active   VLAN Active State
  suspend  VLAN Suspended State
S1(config-vlan)# exit
APPLY completed.
Exiting....

S1(config)# interface FastEthernet 1/1
S1(config-if)# switchport access vlan 2
S1(config-if)# interface FastEthernet 1/2
S1(config-if)# switchport access vlan 3
```

Domyślny stan sieci VLAN to `active`.

`exit` zapisuje zmiany i wychodzi do poziomu uprzywilejowanego.

W niektórych wersjach oprogramowania, VLAN tworzony jest automatycznie przy przypisaniu jego numeru do pierwszego portu.

Przypisywanie interfejsów do VLANów (1b)

```
S1# vlan database
S1(vlan)# vlan 2
VLAN 2 added:
    Name: VLAN0002
S1(vlan)# vlan 2 name staff
VLAN 2 modified:
    Name: staff
S1(vlan)# vlan 3 name students
VLAN 3 added:
    Name: students
S1(vlan)# exit
APPLY completed.
Exiting....

S1(config)# interface FastEthernet 1/1
S1(config-if)# switchport access vlan 2
S1(config-if)# interface FastEthernet 1/2
S1(config-if)# switchport access vlan 3
```

W starszych wersjach oprogramowania konfiguracja VLANów odbywa się z poziomu `vlan database` (tam też można modyfikować parametry VLANów).

`exit` zapisuje zmiany i wychodzi do poziomu uprzywilejowanego.

Przypisywanie interfejsów do VLANów (2)

```
(czasami: show vlan-switch)
S1# show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa1/0, Fa1/3, Fa1/4, Fa1/5
                                                Fa1/6, Fa1/7, Fa1/8, Fa1/9
                                                ...
2    staff                            active    Fa1/1
3    students                         active    Fa1/2
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        1002   1003
2    enet  100002     1500  -      -      -        -    -        0      0
3    enet  100003     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        1      1003
1003 tr    101003     1500  1005   0      -        -    srb      1      1002
1004 fdnet 101004     1500  -      -      1        ibm  -        0      0
1005 trnet 101005     1500  -      -      1        ibm  -        0      0
```

Konfiguracja połączenia typu trunk

```
S1(config)# interface FastEthernet 1/0
S1(config-if)# switchport trunk encapsulation dot1q
S1(config-if)# switchport mode trunk
*Mar  1 00:10:02.203: %DTP-5-TRUNKPORTON: Port Fa1/0 has become dot1q trunk
S1(config-if)# switchport mode ?
  access  Set trunking mode to ACCESS unconditionally
  trunk   Set trunking mode to TRUNK unconditionally

S1# show interfaces trunk
Port      Mode         Encapsulation  Status        Native vlan
Fa1/0     on           802.1q         trunking      1

Port      Vlans allowed on trunk
Fa1/0     1-1005

Port      Vlans allowed and active in management domain
Fa1/0     1-3

Port      Vlans in spanning tree forwarding state and not pruned
Fa1/0     none
```

Zadanie 24

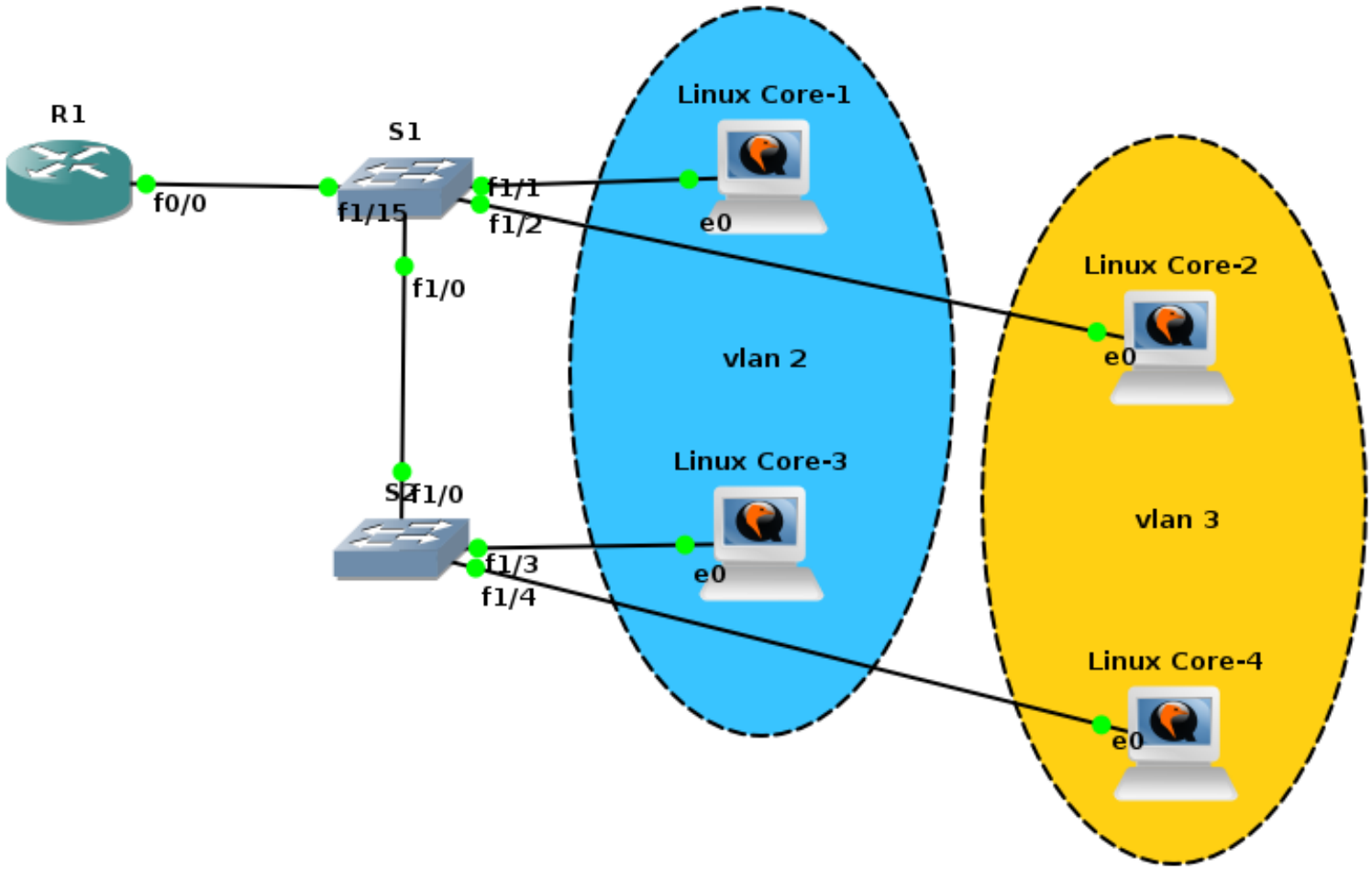
1. Wraz z koleżankami i kolegami zbuduj sieć taką, że:
   - komputery z każdego rzędu podpięte są (poprzez interfejs `p4p1`) do osobnego przełącznika,
   - wszystkie przełączniki są ze sobą połączone (np. w łańcuch).
2. Nadaj adres IP swojemu komputerowi zgodnie ze spójną adresacją, tak by wszystkie komputery w jednej kolumnie należały do jednej (unikalnej) sieci. Sprawdź połączenia z innymi komputerami (w kolumnie) w laboratorium.
3. Wraz z koleżankami i kolegami skonfiguruj przełączniki tak by komputery w każdej kolumnie należały do osobnego VLANu.
4. Przy pomocy programu `wireshark` sprawdź czy możesz zaobserwować jakiekolwiek pakiety z sieci odpowiadających innym VLANom.

Zadanie 24

Wnioski:

- Kompletna separacja ruchu w VLANach.
- Nawet zmieniając adres komputera na adres z innej sieci, nie można z niej korzystać!

Routing między VLANami



Połączenie między routerem a przełącznikiem również musi być połączeniem typu trunk.

Konfiguracja routera do pracy z VLANami

```
R1(config)# interface FastEthernet 0/0
R1(config-if)# no shutdown
R1(config-if)# interface FastEthernet 0/0.2
R1(config-subif)# encapsulation dot1Q 2
R1(config-subif)# ip address 10.0.0.200 255.255.255.0
R1(config-subif)# interface FastEthernet 0/0.3
R1(config-subif)# encapsulation dot1Q 3
R1(config-subif)# ip address 10.0.100.200 255.255.255.0

R1# show ip interface brief
Interface              IP-Address   OK? Method Status                    Prot
FastEthernet0/0        unassigned   YES unset  up                        up
FastEthernet0/0.2      10.0.0.200   YES manual up                        up
FastEthernet0/0.3      10.0.100.200 YES manual up                        up
FastEthernet0/1        unassigned   YES unset  administratively down down
```

Zadanie 25

1. Wraz z koleżankami i kolegami zmodyfikuj sieć, tak by:
   - nie było połączenia między dwoma środkowymi przełącznikami,
   - przełączniki z pierwszej grupy były podłączone do jednego routera, a przełączniki z drugiej grupy były podłączone do drugiego routera.
2. Skonfiguruj router tak, by mógł odbywać się ruch między różnymi VLANami. Dodaj statyczny routing na swoim komputerze, tak by mógł komunikować się z sieciami odpowiadającymi innym VLANom.
3. Połącz routery ze sobą (określ między nimi dodatkową sieć) i skonfiguruj routing tak by była możliwa komunikacja *każdy z każdym* w laboratorium.

Warstwa transportowa

Warstwa transportowa

- Zapewnienia bezbłędną komunikację między komputerami (end-to-end).
- Odpowiada za dzielenie danych na fragmenty i kontrolę ich przesyłania.
- Ustanawia wirtualne połączenia, utrzymuje je i poźniej likwiduje.
- Przykładowe standardy:
  - TCP
  - UDP
  - UDPlite
  - DCCP
  - SCTP

Połączenia w warstwie transportowej

Połączenie jest identyfikowane przez:

- adres lokalny, port lokalny,
- protokół,
- adres zdalny, port zdalny.

Połączeniom w systemie op. odpowiadają gniazda (sockets).

Port:

- 16-bitowy identyfikator usługi/aplikacji/procesu działajacego na konkretnym urządzeniu (na konkretnym adresie IP):

1	–	1023	uprzywilejowane (system/well-known/privileged)
1024	–	49151	zarejestrowane (user/registered)
49152	–	65535	efemeryczne (dynamic/private/ephemeral)

- lista numerów portów przydzielona do usług przez IANA (link),
- lista podstawowych numerów portów – `/etc/services`.

Aktywne połączenia (1)

```
# netstat -tunap
Active Internet connections (servers and established)
Prot RQ SQ Local Address       Foreign Address      State       PID/Program name
tcp   0  0 0.0.0.0:17500       0.0.0.0:*            LISTEN      1618/dropbox
tcp   0  0 127.0.0.1:17600     0.0.0.0:*            LISTEN      1618/dropbox
tcp   0  0 0.0.0.0:22          0.0.0.0:*            LISTEN      972/sshd
tcp   0  0 127.0.0.1:631       0.0.0.0:*            LISTEN      971/cupsd
tcp  38  0 150.254.31.4:55148  108.160.172.204:443  CLOSE_WAIT  1618/dropbox
tcp   0  0 150.254.31.4:47070  108.160.163.108:443  ESTABLISHED 1618/dropbox
tcp   0  0 150.254.31.4:55904  64.233.162.189:443   ESTABLISHED 1855/opera
tcp   0  0 150.254.31.4:55802  54.192.228.32:443    ESTABLISHED 1618/dropbox
...
tcp6  0  0 :::22               :::*                 LISTEN      972/sshd
tcp6  0  0 ::1:631             :::*                 LISTEN      971/cupsd
udp   0  0 0.0.0.0:17500       0.0.0.0:*                        1618/dropbox
udp   0  0 0.0.0.0:50321       0.0.0.0:*                        751/avahi-daemon
udp   0  0 0.0.0.0:5353        0.0.0.0:*                        751/avahi-daemon
udp   0  0 0.0.0.0:2046        0.0.0.0:*                        1277/dhclient
udp   0  0 0.0.0.0:68          0.0.0.0:*                        1277/dhclient
udp6  0  0 :::39661            :::*                             1277/dhclient
```

Aktywne połączenia (2)

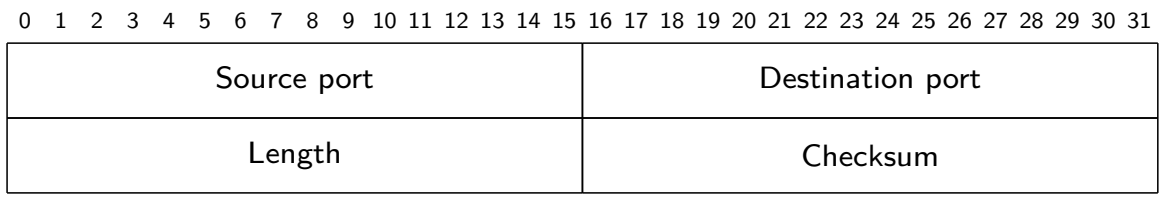
```
# netstat -tunap | grep ssh
tcp   0  0 0.0.0.0:22              0.0.0.0:*               LISTEN      972/sshd
tcp   0  0 150.254.31.4:60824      150.254.30.50:22        ESTABLISHED 3985/ssh
tcp6  0  0 :::22                   :::*                    LISTEN      972/sshd

# netstat -tuap | grep ssh
tcp   0  0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      972/sshd
tcp   0  0 dcs-rw-2.cs.put.p:60824 hpc.cs.put.poznan.p:ssh ESTABLISHED 3985/ssh
tcp6  0  0 [::]:ssh                [::]:*                  LISTEN      972/sshd
```

Zadanie 26

1. Przy pomocy programu `netstat` zbadaj, na jakich portach uruchamiane są serwery:
   - HTTP,
   - HTTPS,
   - FTP,
   - telnet,
   - DNS.

User Datagram Protocol (UDP)

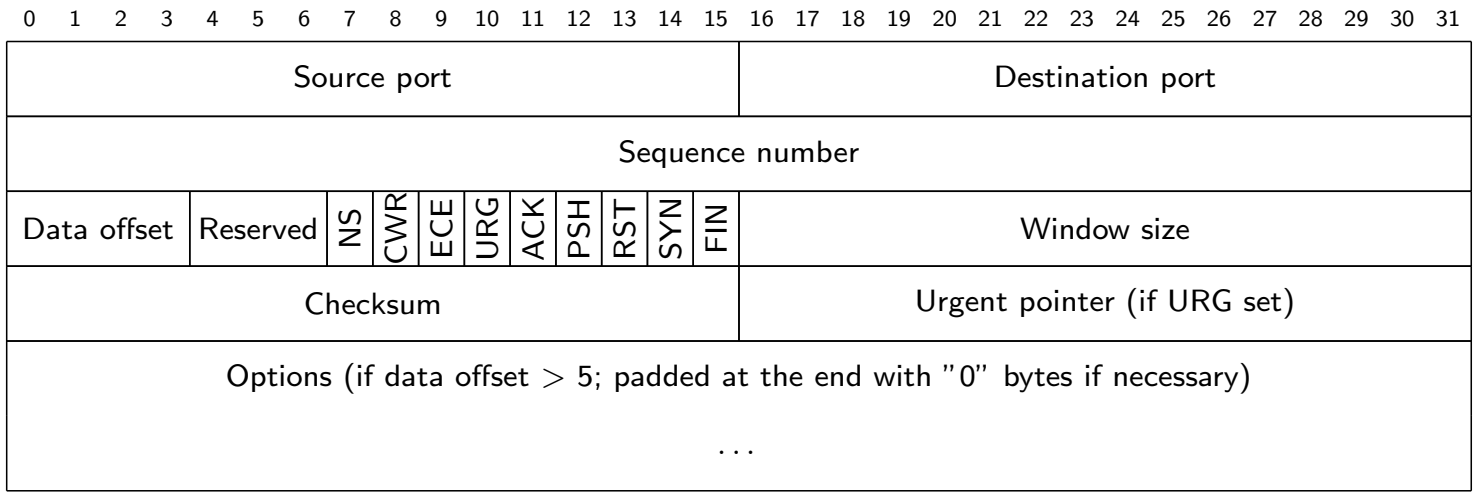
- Bezpołączeniowy protokół transportowy.
- Brak kontroli przepływu i gwarancji dostarczenia – weryfikacja niezawodności w komunikacji musi być implementowana przez aplikacje.
- Weryfikacja integralności przez sumy kontrolne.
- Struktura nagłówka datagramu:



- Protokoły/programy wykorzystujące UDP:
  - DNS
  - DHCP
  - RIP
  - strumieniowanie multimediów
  - gry sieciowe

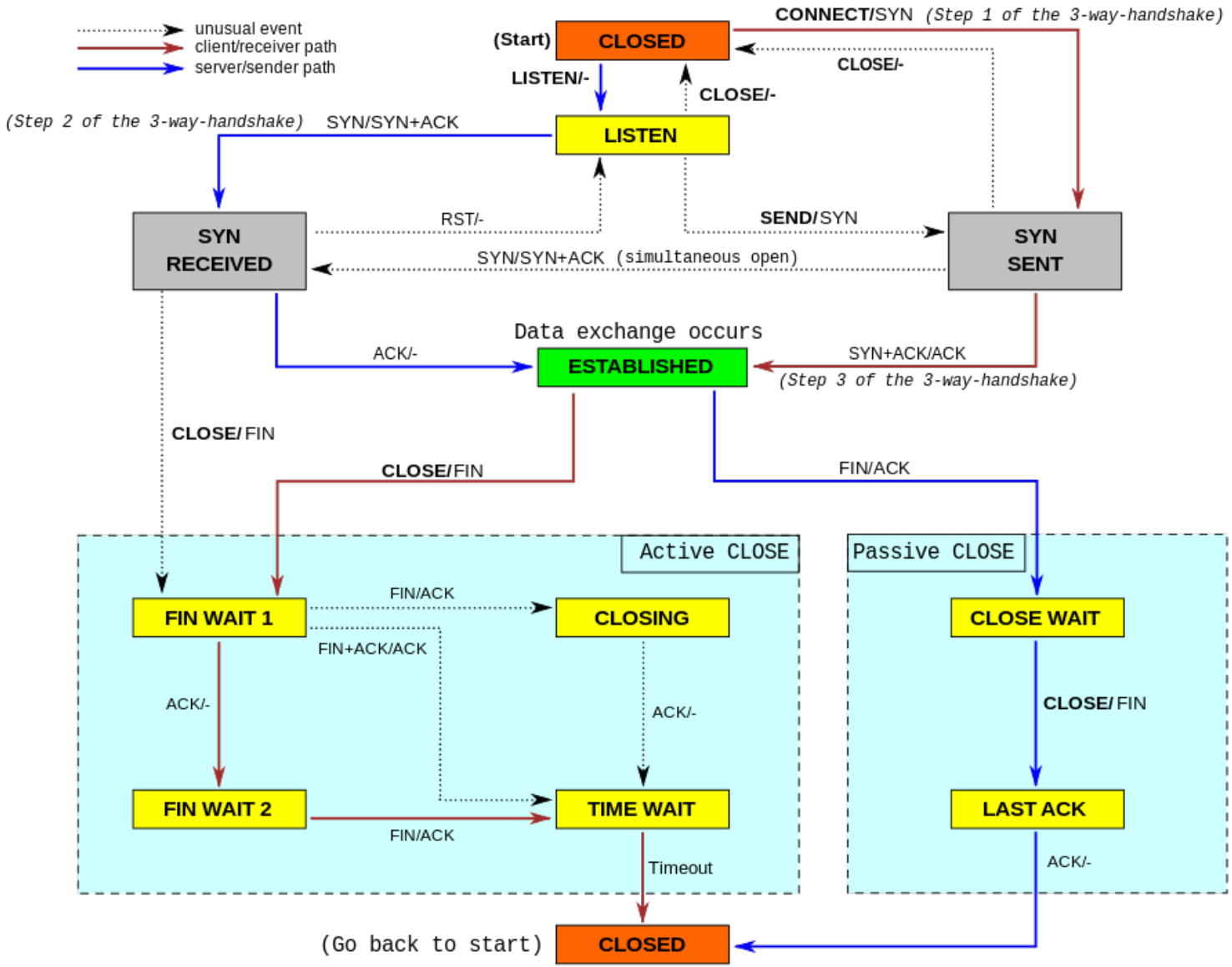
Transmission Control Protocol (TCP) (1)

- Połączeniowy protokół sterowania transmisją.
- Kontrola przepływu (strumień), potwierdzenia, buforowanie.
- Struktura nagłówka segmentu:

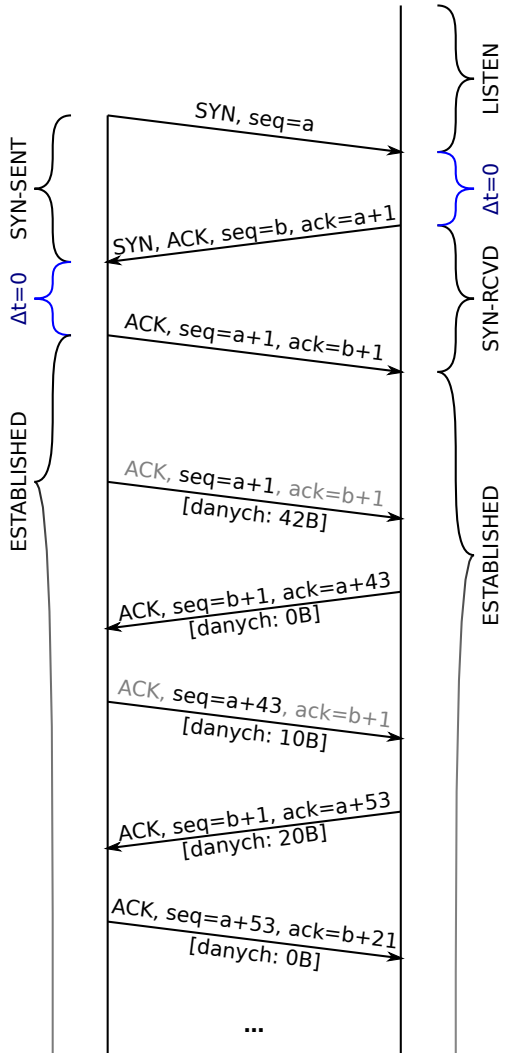


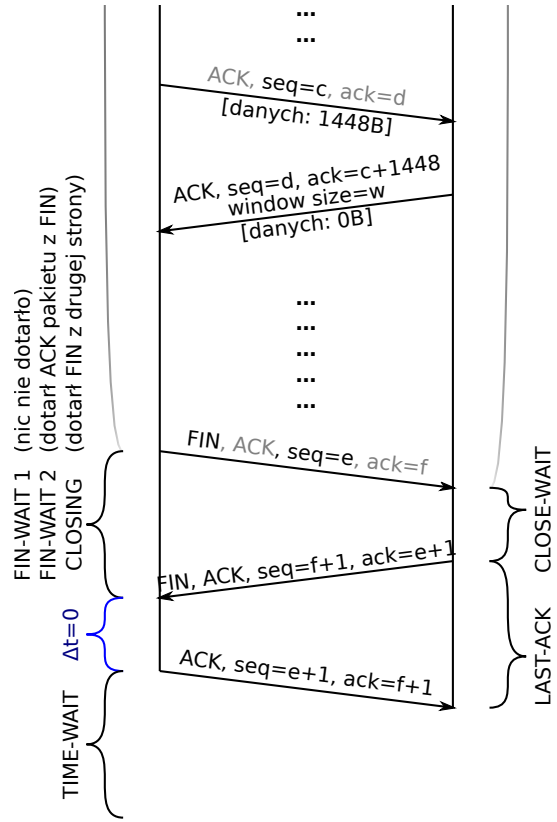
- Protokoły/programy wykorzystujące TCP:
  - HTTP
  - HTTPS
  - SMTP
  - POP3
  - IMAP
  - SSH
  - FTP

Transmission Control Protocol (TCP) (2)



Transmission Control Protocol (TCP) (3)





Rysunek w całości: link

Operacje na gniazdach

`ncat` – jak `cat`, ale dla gniazd a nie dla plików

- kopiowanie danych przez sieć,
- przekierowywanie ruchu dla istniejących połączeń,
- tworzenie prostych usług i klientów usług,
- ...

```
(server)
# ncat -l 127.0.0.1 74 > out.txt

(client)
# ncat 127.0.0.1 74 < in.txt
```

Zadanie 27

Połączenie TCP

```
(server)
# ncat -l 127.0.0.1 74
... [ENTER]

(client)
# ncat 127.0.0.1 74
... [ENTER]
```

Połączenie UDP

```
(server)
# ncat -ul 127.0.0.1 74
... [ENTER]

(client)
# ncat -u 127.0.0.1 74
... [ENTER]
```

1. Co widać w `wiresharku`?
2. Co pokazuje `netstat` (na różnych etapach uruchomienia programów)?
3. Kto pierwszy może pisać, klient czy serwer?
4. Jaka usługa jest przypisana do portu 74 (patrz `/etc/services`)?

Serwer stron WWW (1)

http://www.cs.put.poznan.pl/tkobus

```
# ncat www.cs.put.poznan.pl 80
GET /tkobus/ HTTP/1.0 [ENTER]
[ENTER]

HTTP/1.1 200 OK
Date: Fri, 04 Dec 2015 13:15:54 GMT
Server: Apache
Last-Modified: Fri, 20 Nov 2015 22:02:53 GMT
ETag: "191a67-2887-52500097d7d40"
Accept-Ranges: bytes
Content-Length: 10375
Connection: close
Content-Type: text/html

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
...
</html>
```

Serwer stron WWW (2)

hello.html:

```
<html>
  <body>
    <font color="green">Hello world!</font>
  </body>
</html>
```

```
# ncat -l 127.0.0.1 80 < hello.html
```

Serwer stron WWW (2)

hello.html:

```
<html>
  <body>
    <font color="green">Hello world!</font>
  </body>
</html>
```

```
# ncat -l 127.0.0.1 80 < hello.html
GET / HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
                                                    image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
             like Gecko) Chrome/46.0.2490.80 Safari/537.36 OPR/33.0.1990.58
Accept-Encoding: gzip, deflate, lzma, sdch
Accept-Language: en-US,en;q=0.8
```

Serwer stron WWW (3)

Wiele połączeń: `-k`, `--keep-open`

```
# ncat -lk 127.0.0.1 80 -c 'echo -e "HTTP/1.1 200 OK\n"; cat hello.html'
```

```
# ncat -o out -lk 127.0.0.1 80 -c \
                        'echo -e "HTTP/1.1 200 OK\n"; \
                         echo -e "<html><body>It is $(date)</body></html>"'
```

Inne fajne przykłady

```
# ncat -lk 127.0.0.1 8081 -c 'while true; do read i && echo [echo] $i; done'
```

```
# ncat -lk 192.168.1.101 8081 --max-conns 3 --allow 192.168.1.0/24
                                                    --exec "/bin/bash"
```

```
(server)
# ncat -l 127.0.0.1 74 --chat
(client)
# ncat 127.0.0.1 74
```

Filtracja pakietów w Linuksie

Filtracja pakietów w Linuksie

Netfilter – część jądra systemu operacyjnego Linux pozwalająca na kontrolę ruchu sieciowego.

Do kontroli mechanizmów netfilter służą następujące programy:

- `iptables`
- `ip6tables`
- `ebtables`
- `arptables`

Projekty związane z mechanizmem filtracji pakietów w Linuksie funkcjonują w ramach *Netfilter project*.

Proces filtracji pakietu (1)

Na różnych etapach przechodzenia pakietu przez jądro Linuksa, aplikowane są kolejne filtry.

Filtry pogrupowane są w tabele (tables) grupujące różne funkcjonalności:

- `filter` – odrzuca niechciane pakiety,
- `nat` – zmienia adresy źródłowe lub docelowe pakietu
- `mangle` – modyfikuje pakiety,
- `raw` – udostępnia pakiety przed ich przetworzeniem przez część kernela (np. w trybie śledzenia połączeń (conntrack)),
- `security` – używana przy zaawansowanych modelach bezpieczeństwa w Linuksie (np. SELinux).

Proces filtracji pakietu (2)

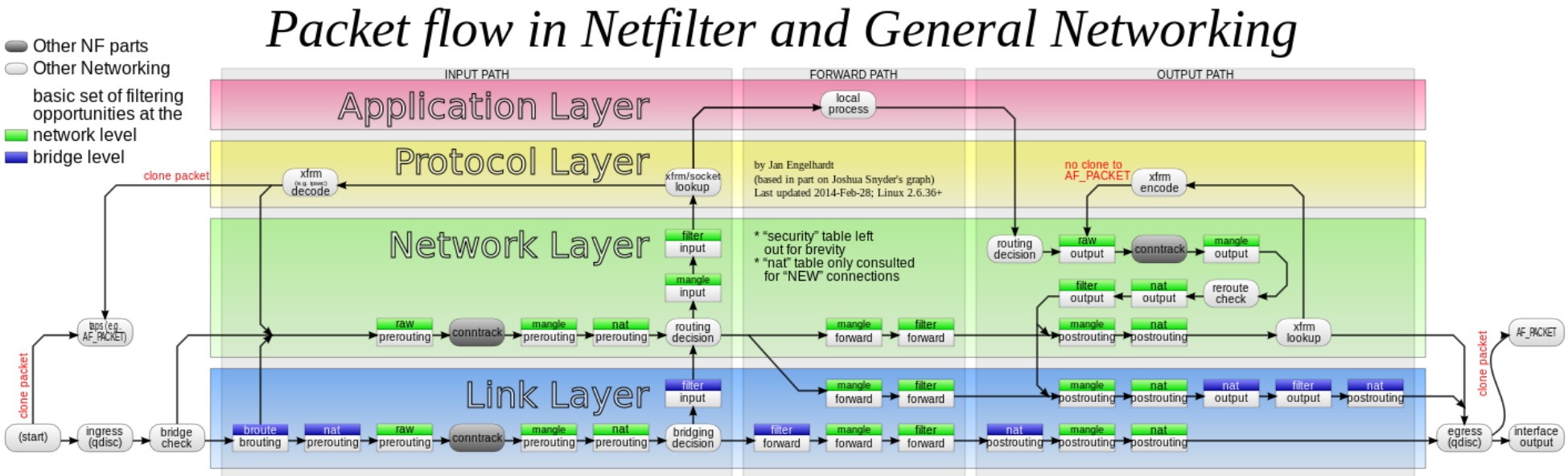
Każda z tablic ma wbudowany zestaw łańcuchów reguł odpowiadających etapom ich aktywacji:

- `filter`: `INPUT`, `FORWARD`, `OUTPUT`,
- `nat`: `PREROUTING`, `OUTPUT`, `POSTROUTING`,
- `mangle`: `PREROUTING`, `INPUT`, `FORWARD`, `OUTPUT`, `POSTROUTING`,
- `raw`: `PREROUTING`, `OUTPUT`.

Można tworzyć własne łańcuchy, ale mogą one być uruchomione tylko przez dołączenie ich do już istniejących.

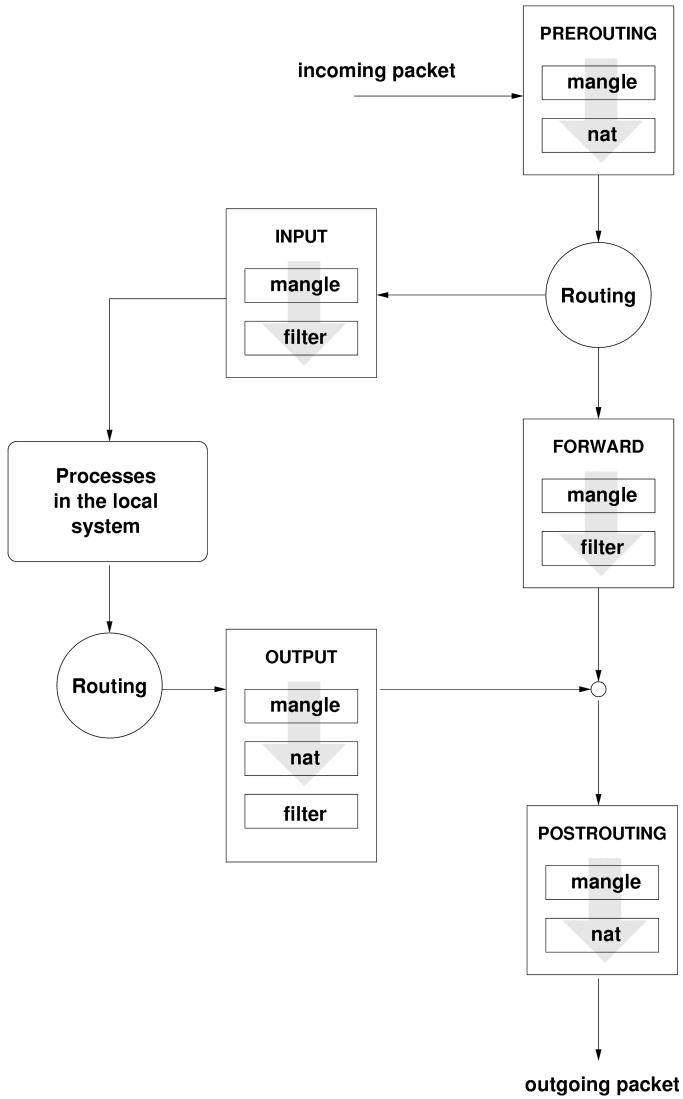
Jądro Linuksa jest zaopatrzone w mechanizm śledzenia połączeń (conntrack). Korzystanie z tego mechanizmu często znacznie ułatwia konfigurację filtracji pakietów.

Proces filtracji pakietu (3)



JK: I tak ten rysunek zawiera nieścisłości.

Proces filtracji pakietu (4)



Konfiguracja filtrów (1)

Reguły są przetwarzane w kolejności – jeśli któraś reguła zadecyduje o losie pakietu, późniejsze nie są brane pod uwagę.

Komenda `iptables` wszędzie rozróżnia wielkie i małe litery. Kolejność argumentów jest istotna!

Typowe wywołanie komendy:

```
iptables [-t tablica] [opcja + łańcuch
+ parametry pakietu] [-j akcja]
```

Operacje w ramach akcji:

- `ACCEPT` – przepuszczenie pakietu,
- `DROP` – ignorowanie pakietu,
- `REJECT` – odrzucenie pakietu (symuluje zamknięte gniazdo),
- inne akcje, np. `LOG`, `MARK`, `SET`.

Konfiguracja filtrów (2)

Wypisywanie reguł:

- `iptables [-t filter] -L`
- `iptables -t <tablica> -L`
- `iptables [-t <tablica>] -L -n -v -x --line-numbers`

Zmiana domyślnego zachowania (tj. jeśli żadna inna reguła nie określi działania):

- `iptables [-t filter] -P <łańcuch> <polityka>`
- zwykle stosowane polityki to ACCEPT lub DROP

```
# iptables -P INPUT DROP
```

Konfiguracja filtrów (4)

Modyfikacja reguł:

- `iptables -A <łańcuch> <reguła>`
  dodawanie (`--append`) reguły do łańcucha,
- `iptables -I <łańcuch> [pozycja] <reguła>`
  wstawianie (`--insert`) reguły do łańcucha na daną pozycję,
- `iptables -D <łańcuch> <reguła>/<nr>`
  usuwanie (`--delete`) reguły z łańcucha z podanej pozycji,
- `iptables -F [łańcuch]`
  usuwanie wszystkich reguł [z łańcucha] (`--flush`),
- `iptables -Z [łańcuch]`
  resetowanie (`--zero`) liczników łańcucha.

```
# iptables -A INPUT -s 1.2.3.4 -j ACCEPT
# iptables -I INPUT 1 -s 1.2.3.4 -j ACCEPT
# iptables -D INPUT 2
# iptables -D INPUT -s 1.2.3.4 -j ACCEPT
# iptables -F INPUT
```

Konfiguracja filtrów (5)

Rdzeń iptables zawiera niewiele filtrów, m.in. źródłowy i docelowy adres (`-s`, `-d`), interface (`-i`, `-o`) i protokół (`-p`). Wiecej: `man iptables`.

Filtry można negować przez użycie znaku wykrzyknika '!'.

Komenda iptables ma rozbudowaną pomoc wewnętrzną – po napotkaniu `--help` przerwie dodawanie reguły i pokaże listę opcji.

```
# iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
# iptables -A INPUT -i wlan0 -s 192.168.55.0/24 -j REJECT
# iptables -I INPUT -p tcp --help
```

Konfiguracja filtrów (6)

Wiele funkcji dostępnych jest poprzez rozszerzenia, ładowane przez `-m <nazwa>` (patrz też `man iptables-extensions`):

- `-m tcp/udp` – jest automatycznie ładowany razem z `-p udp/tcp`; pozwala ustalić m.in. port źródłowy i docelowy `--sport/--dport`,
- `-m conntrack` – wybiera stan połączenia `--ctstate`, m.in.: `INVALID`, `NEW`, `ESTABLISHED`, `RELATED`,
- `-m comment` – pozwala na dowolny komentarz `--comment`,
- `-m limit` – dzięki `--limit` ogranicza liczbę pakietów na jednostkę czasu,
- `-m time` – pozwala włączyć regułę o `--datestart` i wyłączyć o `--datestop`,
- `-m connlimit` – używając `--connlimit-above` pozwala ograniczyć liczbę połączeń z jednego adresu.

Konfiguracja filtrów (7)

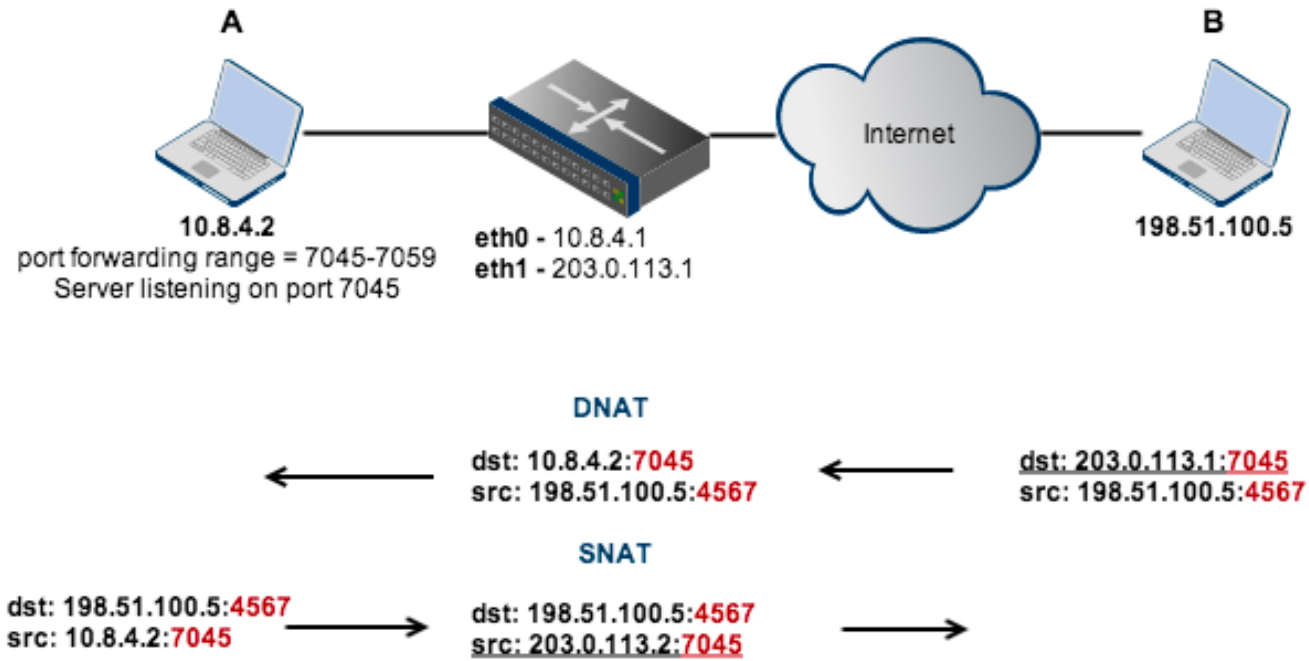
```
(ruch wychodzacy dla grupy o gid 1006 tylko do sieci 192.168.1.0/24)
# iptables -A OUTPUT -m owner --gid-owner 1006 ! -d 192.168.1.0/24 -j DROP

(ignoruj przychodzace polaczenia tcp)
# iptables -A INPUT --protocol tcp --syn -j DROP
# iptables -A INPUT --protocol tcp --tcp-flags SYN,RST,ACK,FIN SYN -j DROP

(tylko 2 polaczenia telnetowe per klient)
# iptables -A INPUT -p tcp --syn --dport 23 -m connlimit \
                                          --connlimit-above 2 -j REJECT

(dopusc ruch tylko z okreslonego adresu mac)
# iptables -A INPUT -m mac ! --mac-source 00:12:34:56:78:ab -j DROP

(umozliwienie przyjmowania komunikatow icmp)
# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

Zadanie 28

1. Zablokuj cały ruch wchodzący. Użyj do tego polityki, nie reguły.
2. Pozwól wchodzić połączeniom już ustanowionym.
3. Pozwól wchodzić połączenim ssh.
4. Zabroń wchodzenia ruchu ssh z wybranego adresu nie kasując żadnej z powyższych reguł.
5. Wyświetl liczniki, sprawdź ile razy reguła z poprzedniego zadania została uruchomiona.
6. Zbadaj (używając wiresharka) czym różni się cel DROP od REJECT.
7. Zablokuj ruch wychodzący do `www.cs.put.poznan.pl`
8. Ogranicz liczbę połączeń ssh od każdego adresu IP z osobna.
9. Ogranicz globalną liczbę połączeń ssh.

Translacja adresów w Linuksie

Network Address Translation (NAT)



Translacja źródłowa – Source NAT (SNAT)

Pozwala na dostęp do sieci z adresacją publiczną (np. Internet) urządzeniom z sieci z adresacją prywatną.

Maskarada:

- uproszczony tryb translacji źródłowej,
- adres źródłowy jest automatycznie ustawiany na adres interfejsu, którym pakiet opuszcza router,
- trzeba podać przynajmniej interfejs wyjściowy!

```
# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Klasyczny SNAT:

- wymaga podania adresu (lub adresów) na jakie ma być zmieniane źródło

```
# iptables -t nat -A POSTROUTING -j SNAT --to-source 203.0.113.2
```

Translacja źródłowa – Source NAT (SNAT)

Problematyczny scenariusz:

- dwa komputery z sieci prywatnej A (10.8.4.2) i A' (10.8.4.3),
- A i A' chcą łączyć się z tym samym sewerem B (198.51.100.5),
- A i A' losują ten sam numer portu dla połączenia z B, np. 56000,
- po SNAT na routerze, nie można byłoby okreslić, do którego komputera (A czy A') należy skierować pakiety zwrotne od B.

Rozwiązanie:

- router dokonuje czasami (np. dla A') zarówno translacji IP jak i portu źródłowego,
- po SNAT na routerze, pakiety generowane przez A' będą wyglądać tak, jakby były nadane przez router z adresu 203.0.113.2 i portu np. 56007.

Translacja docelowa – Dest. NAT (DNAT)

Pozwala na dostęp świata do urządzeń znajdujących się w sieci prywatnej.

W szczególności pozwala na przekierowywanie portów między hostami.

```
# iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j DNAT \
                                          --to-destination 10.8.4.2
# iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 22 -j DNAT \
                                          --to-destination 10.8.4.2:8008
```

SNAT vs DNAT

SNAT i DNAT to zupełnie dwie niezależne rzeczy!

- zarówno SNAT i DNAT dokonują translacji adresów źródłowych jak i adresów docelowych, ale
- dla danego żądania i odpowiedzi:
  - SNAT: najpierw translacja adresów źródłowych w żądaniach, a później translacja adresów docelowych w odpowiedziach,
  - DNAT: najpierw translacja adresów docelowych w żądaniach, a później translacja adresów źródłowych w odpowiedziach,
- może być SNAT bez DNAT (patrz punkt 2 następnego Zadania),
- może być DNAT bez SNAT (patrz punkty 6-7 następnego Zadania).

Zadanie 29 (1)

Do rozwiązania w rzędach, min. 3 komputery PC1–PC3.

Konfiguracja podstawowa:

- dwie oddzielne sieci prywatne PC1–PC2 (na `p4p1`) i PC2–PC3 (na `p4p2`)
- PC1, PC3: dla pewności wyłączony interfejs `em1`,
- PC2: ustawione przekazywanie pakietów: `sysctl -w net.ipv4.conf.all.forwarding=1`,
- PC3: PC2 jako domyślna brama (adres interfejsu `p4p2`).

Zadanie 29 (2)

1. Ustaw na PC2 maskaradę adresów tak, żeby PC1 i PC3 mogły korzystać z Internetu (PC1 nie ma routingu domyślnego, więc w tym momencie Internet nie będzie działać). Uwaga na `/etc/resolv.conf`.
2. Sprawdź czy PC3 może łączyć się z PC1. Jeśli nie, napraw to używając SNAT.
3. Używając wiresharka zobacz jakie adresy ma ustawiony pakiet wysłany z PC1 do PC3 na każdym komputerze.
4. Zmień TTL pakietów przechodzących przez PC2 z PC3 do PC1 na 1, zaobserwuj (w wiresharku) co się dzieje.
5. Włącz na PC3 program nasłuchujący na wybranym porcie.
6. Skonfiguruj PC2 tak, by PC1 mógł połączyć się z programem działającym na PC3 używając adresu PC2.
7. Skonfiguruj PC2 tak, by PC1 mógł połączyć się z serwerem SSH działającym na PC3 używając adresu PC2 i portu innego niż 22.
8. Używając wiresharka zobacz jak modyfikowane są pakiety z powyższych poleceń.

Referencje (1)

Podstawowe materiały źródłowe:

1. *Sieci komputerowe*, A. Tanenbaum
2. Skrypty do laboratoriów z Sieci Komputerowych J. Kończaka, K. Sieka i M. Kalewskiego
3. Skrypt do wykładu z Sieci Komputerowych J. Kobusa

Rysunki:

1. http://www.wired.com/wp-content/uploads/2014/05/1069646562.LGL_.2D.4000x4000.png
2. http://photos.prnewswire.com/prnfull/20141009/151326
3. http://ltwp.net/networks/images/network_types.png
4. https://www.nytimes.com/interactive/2019/03/10/technology/internet-cables-oceans.html
5. Diagramy modelu warstwowego sieci z *Sieci komputerowe*, A. Tanenbaum
6. https://commons.wikimedia.org/wiki/File:UDP_encapsulation.svg
7. http://cdn.1001freedownloads.com/vector/thumb/77770/RJ45_Female.png
8. http://mblogthumb3.phinf.naver.net/20140530_18/second_brain_1401424709357IjOUQ_PNG/053014_0438_25.png?type=w2
9. http://www.cs.put.poznan.pl/ksiek/_images/hub.png
10. http://www.cs.put.poznan.pl/ksiek/_images/switch.png
11. http://www.cs.put.poznan.pl/ksiek/_images/domains.png
12. http://www.ebrahma.com/wp-content/uploads/2014/04/STP.gif
13. http://xkcdexplained.wikia.com/wiki/Map_of_the_Internet?file=Map_of_the_internet.jpg
14. http://www.cs.put.poznan.pl/ksiek/_images/plan11.png
15. http://www.cs.put.poznan.pl/ksiek/_images/cidr1.png

Referencje (2)

Rysunki (ciąg dalszy):

16. https://notes.shichao.io/tcpv1/figure_5-16.png
17. http://www.cs.put.poznan.pl/ksiek/_images/router.png
18. http://www.cs.put.poznan.pl/ksiek/_images/routing.png
19. http://www.cs.put.poznan.pl/ksiek/_images/routing-example.png
20. https://notes.shichao.io/tcpv1/figure_13-1.png
21. http://ssfnet.org/Exchange/tcp/Graphics/tcpStateDiagram1.gif
22. https://en.wikipedia.org/wiki/Netfilter#/media/File:Netfilter-packet-flow.svg
23. fire-filter
24. http://i.stack.imgur.com/Tztkc.png
25. https://upload.wikimedia.org/wikipedia/commons/6/65/Ethernet_MDI_to_MDIX.svg
26. https://upload.wikimedia.org/wikipedia/commons/5/5b/Ethernet_MDI_crossover.svg
27. http://seth.galitzer.net/files/dhcp/final.png
28. https://images-na.ssl-images-amazon.com/images/I/61RJ3sLkncL._SL1500_.jpg
29. https://ccie4all.files.wordpress.com/2013/01/ospf-lsas-and-area-types.jpg
30. https://lh6.ggpht.com/nj6EBQBek5JAhqJ5QjsUGEeoOAdSLG7Y5x6_710ndbSQ7q44xD91jNZqMH1GnxVYiw=w300
31. http://www.cisco.com/c/dam/en/us/support/docs/ip/routing-information-protocol-rip/13714-43-01.gif
32. http://www.cs.put.poznan.pl/jkonczak/tcp_flow.svg
33. https://commons.wikimedia.org/w/index.php?curid=30810617