

**WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ  
I ZARZĄDZANIA  
WYDZIAŁ INFORMATYKI**

**SIECI KOMPUTEROWE I ROZPROSZONE SYSTEMY OPERACYJNE  
(nazwa skrócona: sieci i systemy rozproszone: SSR)**

**WYKŁADY i ZAJĘCIA LABORATORYJNE**

**Wykładowcy: Lech Kruś, Jacek Malinowski  
Prowadzący laboratorium: J. Malinowski**

**Warszawa**

# **SIECI KOMPUTEROWE I ROZPROSZONE SYSTEMY OPERACYJNE**

## **CEL PRZEDMIOTU:**

**Wprowadzenie słuchaczy do współczesnych zagadnień systemów rozproszonych i podstaw sieci komputerowych.**

Ułatwienie rozumienia podstawowych zagadnień związanych z komunikacją w sieciach komputerowych i działaniem systemów rozproszonych, w tym takich jak: wielowarstwowe protokoły komunikacji w sieciach, działanie w układzie klient serwer, adresowanie w internecie, zdalne wykonywanie prac, przekazywanie komunikatów, synchronizacja i zarządzanie procesami, rozproszone systemy plików, zagadnienia tolerowania awarii, wprowadzenie do zarządzania w sieciach.

**Uzupełnieniem wykładu są zajęcia laboratoryjne poświęcone zagadnieniom sieci komputerowych**

## **WYMAGANE PRZYGOTOWANIE SŁUCHACZY:**

- w zakresie podstaw informatyki technicznej (cyfrowej reprezentacji informacji, podstaw arytmetyki komputerów, podstaw teorii układów logicznych):  
A. Skorupski, Podstawy budowy i działania komputerów, WKŁ 1996,
- organizacji i architektury komputerów  
(P. Metzger, Anatomia PC, Helion 1996),
- wielodostępnych systemów operacyjnych I  
(wykłady i laboratorium na Wydz. Informatyki, WSISiZ)
- wielodostępnych systemów operacyjnych II - zagadn. zaawansowane  
(wykłady i laboratorium na Wydz. Informatyki, WSISiZ)

## **ZALICZENIE PRZEDMIOTU:**

**zaliczenie laboratorium (aktywna praca, kolokwia), egzamin.**

# **ZAKRES TEMATYCZNY WYKŁADÓW**

## **I. ZAGADNIENIA BUDOWY SYSTEMÓW ROZPROSZONYCH**

### **Synchronizacja w systemach rozproszonych**

Synchronizacja czasu logicznego i czasu fizycznego

Algorytmy synchronizacji procesów

Algorytmy elekcji

### **Transakcje niepodzielne**

Założenia przetwarzania transakcyjnego

Metody realizacji: prywatna przestrzeń robocza,

Protokół zatwierdzania dwufazowego

Protokoły współbieżnego przetwarzania transakcji

### **Blokady w systemach rozproszonych**

Algorytm scentralizowanego rozpoznawania blokady

Algorytm zdecentralizowany

## **Procesy i procesory w systemach rozproszonych**

Praca wielowątkowa

Synchronizacja wątków

Modele systemów

- Model stacji roboczej

- Model puli procesorów

## **Zagadnienia tolerowania awarii**

Wady elementów systemu

Awarie systemu

Redundancja

Zwielokrotnienie aktywne

Zasoby rezerwowe

Uzgodnienia w systemach wadliwych

Przykład systemu MC Service Guard firmy Hewlett-Packard

## **Wprowadzenie do zarządzania w sieciach komputerowych (opcjonalnie)**

Model zarządzania: stacja zarządzająca – „agent” na stacji roboczej

Zmienne wykorzystywane do zarządzania

Baza MIB

Przykład rodziny produktów Open View firmy Hewlett Packard

## **Rozproszone środowisko obliczeniowe (DCE)**

## **II. PODSTAWY SIECI KOMPUTEROWYCH**

### **Praca w intersieci (internetworking)**

Bazowe techniki sieciowe (Ethernet, FDDI, ATM)

Specyfikacja Ethernetu. Metody dostępu do sieci.

Budowa ramek (Ethernet, FDDI)

Adresy logiczne i fizyczne

**Łączenie sieci w intersieć i model jej architektury.** Przyporządkowanie adresowi logicznemu adresu fizycznego (ARP). Przenoszenie datagramów w intersieci bez użycia połączenia – protokół IP.

Format datagramu, typ obsługi datagramu, kapsułkowanie datagramu, MTU sieci i fragmentacja, opcje datagramów.

Wyznaczanie trasy datagramu, podstawowe zasady.

Analizatory protokołów sieciowych.

### **Wybrane protokoły**

Komunikaty kontrolne i komunikaty o błędach - protokół ICMP. Formaty komunikatów ICMP. Wykorzystanie ICMP- ping. Przesyłanie danych niezawodnymi strumieniami – protokół TCP. Format segmentu TCP.

Realizacja niezawodności połączenia za pomocą TCP.

Porty, połączenia, pasywne i aktywne otwarcia.

Potwierdzenia i retransmisje. Protokół UDP.

## **Modele warstwowe oprogramowania protokołów.**

Zasady podziału na warstwy. 7-warstwowy wzorcowy model ISO-OSI. X.25 i jego związek z modelem ISO-OSI.

Model warstwowy TCP/IP. Podstawowe idee multiplesowania i demultiplesowania.

Programy użytkowe do pracy na odległym komputerze: telnet, rlogin.

Programy użytkowe do przesyłania plików i dostępu: ftp, tftp.

## **Konfiguracja i podstawa administrowania TCP/IP**

Konfiguracja interfejsów sieciowych. Ustawienia routingu. Pliki konfiguracyjne.

Demon inetd. Polecenie netstat.

## **Zdalne wywoływanie procedury (RPC)**

Model zdalnego wywoływania procedury.

Model proceduralny w systemach rozproszonych.

Semantyka wywołań a protokół komunikacyjny.

Format komunikatów Sun RPC.

Generowanie programów rozproszonych (generator rpcgen)

## **Sieciowy system plików NFS**

Zdalny dostęp a przesyłanie plików. Serwer i klient NFS. Montowanie systemu plików w NFS. Uchwyty do plików. Protokół montowania.

## **Obsługa nazw domenowych (DNS)**

Struktura nazw domenowych. Organizacja serwerów DNS. Rekordy zasobów RR.

Format komunikatów DNS.

Pliki konfiguracyjne na serwerach DNS.

## **LITERATURA PODSTAWOWA:**

**Tanenbaum Andrew S. , Maarten van Steen: Systemy rozproszone, zasady i paradygmaty. WNT. Warszawa, 2006.**

**Tanenbaum Andrew S. : Rozproszone systemy operacyjne. PWN. Warszawa 1997.**

**Coulouris G. , J. Dollimore, T. Kindberg : Systemy rozproszone, podstawy i projektowanie. WNT, Warszawa, 1998.**

**Douglas E. Comer: Sieci komputerowe i intersieci. WNT Warszawa 2000.**

**Douglas E. Comer: Sieci komputerowe TCP/IP. (Tom 1) Zasady, protokoły i architektura. WNT, Warszawa 1998.**

**Larry L. Peterson, Bruce S. Davie: Sieci komputerowe – podejście systemowe. Nikom, Poznań.2000.**

## **Literatura uzupełniająca:**

**W. Richard Stevens: TCP/IP Illustrated Volumel Protocols, Addison Wesley, New York, 1994**

**Craig Hunt: TCP/IP Administracja Sieci. O'Reilly&Associates Inc., Oficyna Wyd. READ ME, Warszawa, 1996.**

**Abraham Silberschatz, James.L. Peterson, Peter.B. Galvin; Podstawy systemów operacyjnych, WNT, Warszawa 1993.**

**Maurice. J. Bach; Budowa systemu operacyjnego UNIX, WNT, Warszawa, 1995.**

**Andrew. S. Tanenbaum; Modern Operating Systems, Prentice-Hall, Inc. London, 1992.**

**Abraham Silberschatz, Peter.B. Galvin; Operating System Concepts. Addison Wesley, New York, 1994**

**Douglas E. Comer, D. L. Stevens: Sieci komputerowe TCP/IP. (Tom 2) Projektowanie i realizacja protokołów. WNT, Warszawa 1997.**

**Douglas E. Comer, D. L. Stevens: Sieci komputerowe TCP/IP. (Tom 3) Programowanie w trybie klient serwer. Wersja BSD. WNT, Warszawa 1997.**

## PROBLEMY SYNCHRONIZACJI CZASU

### Cechy algorytmów rozproszonych:

- Informacje rozmieszczone na wielu maszynach
- Procesy podejmują decyzje tylko na podstawie informacji lokalnych
- Należy unikać skupiania elementów wrażliwych na awarie w jednym punkcie systemu
- W systemie rozproszonym nie istnieje jedno wspólne precyzyjne źródło czasu (wspólny zegar)

## **SYNCHRONIZACJA ZEGARÓW**

Ilustracja działania programu make w systemie dwóch maszyn o niezależnych zegarach.

### **Logiczna synchronizacja zegarów**

Zegary logiczne (logical clocks): zapewniające wewnętrzną zgodność czasu.

### **Fizyczna synchronizacja zegarów**

Zegary fizyczne (physical clocks): czas wskazywany przez zegary jest zgodny z czasem rzeczywistym (z określoną dokładnością)

## Algorytm synchronizacji zegarów logicznych (Lamport)

Rozpatrujemy system rozproszony, w którym jest wiele procesów, każdy na innej maszynie, każdy ma własny czasomierz.

**Relacja uprzedniości zdarzeń** (happens-before relation):

**Def.** Mówimy, że zdarzenie  $a$  poprzedza zdarzenie  $b$  i piszemy  $a \rightarrow b$ , wtedy i tylko wtedy, gdy wszystkie procesy są zgodne co do tego, że zdarzenie  $a$  zachodzi najpierw, a potem dopiero zdarzenie  $b$ .

**Relacja ta zachodzi bezpośrednio w przypadkach:**

1. Jeżeli  $a$  i  $b$  są zdarzeniami w tym samym procesie i  $a$  występuje przed  $b$ , to relacja  $a \rightarrow b$  jest prawdziwa.
2. Jeżeli  $a$  jest zdarzeniem wysłania komunikatu przez jeden proces i  $b$  jest zdarzeniem odebrania komunikatu przez inny proces, to relacja  $a \rightarrow b$  jest prawdziwa.

**Przechodniość relacji:** jeżeli  $a \rightarrow b$  i  $b \rightarrow c$ , to  $a \rightarrow c$ .

Zdarzenia współbieżne (concurrent): dwa zdarzenia występujące w różnych procesach, które nie wymieniają komunikatów.

**Przypisanie wartości czasu zdarzeniom** (ozn.  $C(a)$ ) powinno mieć własności:

1. Jeżeli  $a \rightarrow b$  to  $C(a) < C(b)$ .
2. Czas zegarowy  $C$  musi zawsze wzrastać.

0	1	2
0	0	0
6	8	10
12	16	20
18	24	30
24	32	40
30	40	50
36	48	60
42	56	70
48	64	80
54	72	90
60	80	100

(a)

0	1	2
0	0	0
6	8	10
12	16	20
18	24	30
24	32	40
30	40	50
36	48	60
42	61	70
48	69	80
70	77	90
76	85	100

(b)

Rys. Trzy procesy (0, 1, 2) z własnymi zegarami bez korekty (a) i z korektą (b) wg. algorytmu Lamporta.

**Warunki przypisania czasu wszystkim zdarzeniom w systemie rozproszonym**  
(zastosowane w algorytmie Lamporta)

1. Jeżeli zdarzenie  $a$  poprzedza zdarzenie  $b$  w tym samym procesie, to  $C(a) < C(b)$ .
2. Jeżeli  $a$  oznacza nadanie komunikatu, a  $b$  jego odebranie, to  $C(a) < C(b)$ .
3. Dla wszystkich zdarzeń  $a$  i  $b$ ,  $C(a) \neq C(b)$ .

## **SYNCHRONIZACJA ZEGAROW FIZYCZNYCH**

wymagana, gdy czas przypisywany zdarzeniom w systemie rozproszonym powinien się pokrywać z czasem rzeczywistym ( np. w systemach czasu rzeczywistego).

### **Problemy pomiaru czasu**

#### **Czas astronomiczny**

sekunda słoneczna: 1/86400 dnia słonecznego (między górowaniami słońca)  
średnia sekunda słoneczna (mean solar second)

#### **Międzynarodowy czas atomowy - TAI (International Atomic Time)**

czas określonej liczby przejść atomu cezu 133

Bureau International de l 'Heure w Paryżu podaje średnią z zegarów atomowych z ok. 50 laboratoriów.

## **Uniwersalny czas skoordynowany - UTC (Universal Coordinated Time)**

Czas atomowy skoordynowany z czasem astronomicznym przez dodawanie sekund przestępnych.

Wzorzec dla wszystkich współczesnych cywilnych pomiarów czasu udostępniany przez:

**NIST** (National Institute of Standard Time), Fort Collins, Colorado  
nadajnik krótkofalowy **WWV** i inne stacje radiowe.

**Satelitę GEOS** (Geostationary Environment Operational Satellite) i inne satelity.

## Przykład scentralizowanego algorytmu synchronizacji

### Założenia

System rozproszony - wiele maszyn, jedna (serwer czasu) ma odbiornik WWV.

Zakłada się, że każda maszyna ma czasomierz powodujący H przerwań na sek. Czas liczony jest jako liczba impulsów zliczanych od pewnej ustalonej chwili w przeszłości (kolejny impuls dodawany jest w momencie kolejnego wyzerowania czasomierza).

**Maksymalny współczynnik odchylenia** (maximum drift time): stała  $\rho$  taka, że

$$1 - \rho \leq \frac{dC}{dt} \leq 1 + \rho,$$

gdzie

$C(t)$  - czas wskazywany przez zegar maszyny względem czasu teoretycznego.  
(Rzeczywista liczba przerwań na sek. w różnych maszynach może odbiegać od  $H$ ).

Zapewnienie odchylenia czasu między dwiema maszynami nie większego niż  $\delta$  wymaga korekty zegarów nie rzadziej niż co  $\delta / (2\rho)$  sekund.

## Idea algorytmu synchronizacji czasu fizycznego (Cristian)

Każda maszyna okresowo (co  $\delta / (2 \rho)$  sekund) wysyła komunikat do serwera czasu z pytaniem o bieżący czas.

Serwer czasu podaje w odpowiedzi czas UTC ozn.  $C_{\text{UTC}}$ .

Każda z maszyn koryguje czas (stopniowo).

Należy uwzględnić czas przenoszenia komunikatu.

## WZAJEMNE WYŁĄCZANIE W SYSTEMACH ROZPROSZONYCH

Metody wykorzystujące pamięć dzieloną (semafora, monitory) nie są odpowiednie w systemach rozproszonych.

### Algorytm scentralizowany

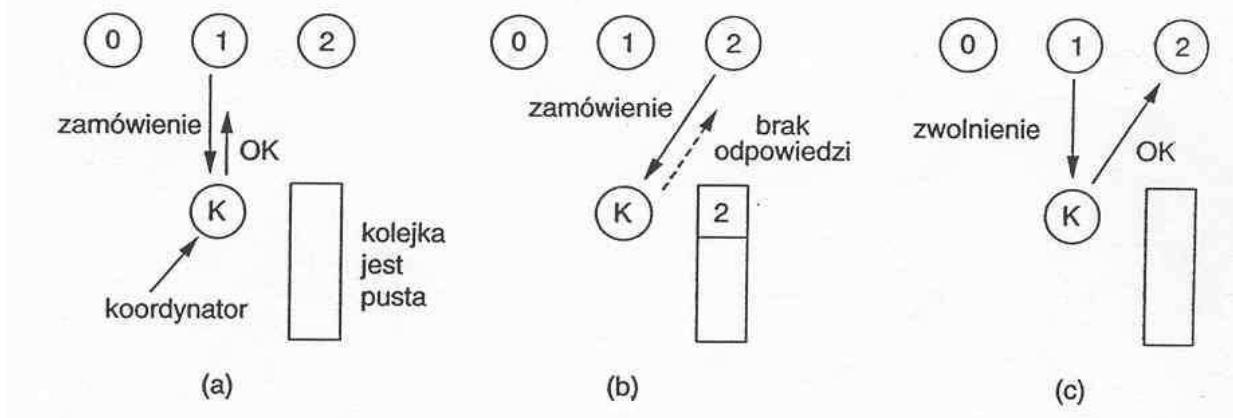
Jeden proces jest koordynatorem.

Proces, który chce wejść do sekcji krytycznej wysyła zamówienie do koordynatora.

Koordynator odpowiada (udziela zezwolenia), gdy żaden inny proces nie jest w sekcji krytycznej.

Proces, po odebraniu zezwolenia, wchodzi do sekcji krytycznej.

Proces wychodząc z sekcji krytycznej wysyła komunikat do koordynatora.



### Cechy algorytmu scentralizowanego:

zapewnia wzajemne wyłączanie, nie zachodzi głodzenie procesów,  
łatwy w realizacji, wrażliwy na awarie.

## Algorytm rozproszony

Wymagane jest całkowite uporządkowanie czasowe zdarzeń - komunikatów (np. stosując algorytm Lamporta).

Proces, który chce wejść do sekcji krytycznej wysyła do wszystkich procesów komunikat zawierający nazwę sekcji krytycznej, swój numer, bieżący czas.

Każdy komunikat jest potwierdzany (zapewnienie niezawodności).

Proces odbierający komunikat:

1. Jeśli nie jest w sekcji krytycznej i nie chce do niej wejść – wysyła do nadawcy komunikat OK.
2. Jeśli jest w sekcji krytycznej - nie odpowiada.
3. Jeśli chce wejść do sekcji krytycznej - sprawdza znacznik czasu odebranego komunikatu i komunikatu, który sam wysłał. Jeśli odebrany komunikat ma znacznik czasu mniejszy - wysyła OK.

Proces nadawca:

Czeka aż wszystkie procesy udzielą zezwolenia, wtedy wchodzi do sekcji krytycznej.

Wychodząc z sekcji krytycznej wysyła OK. do procesów, które ustawił w kolejce.



Cechy algorytmu:

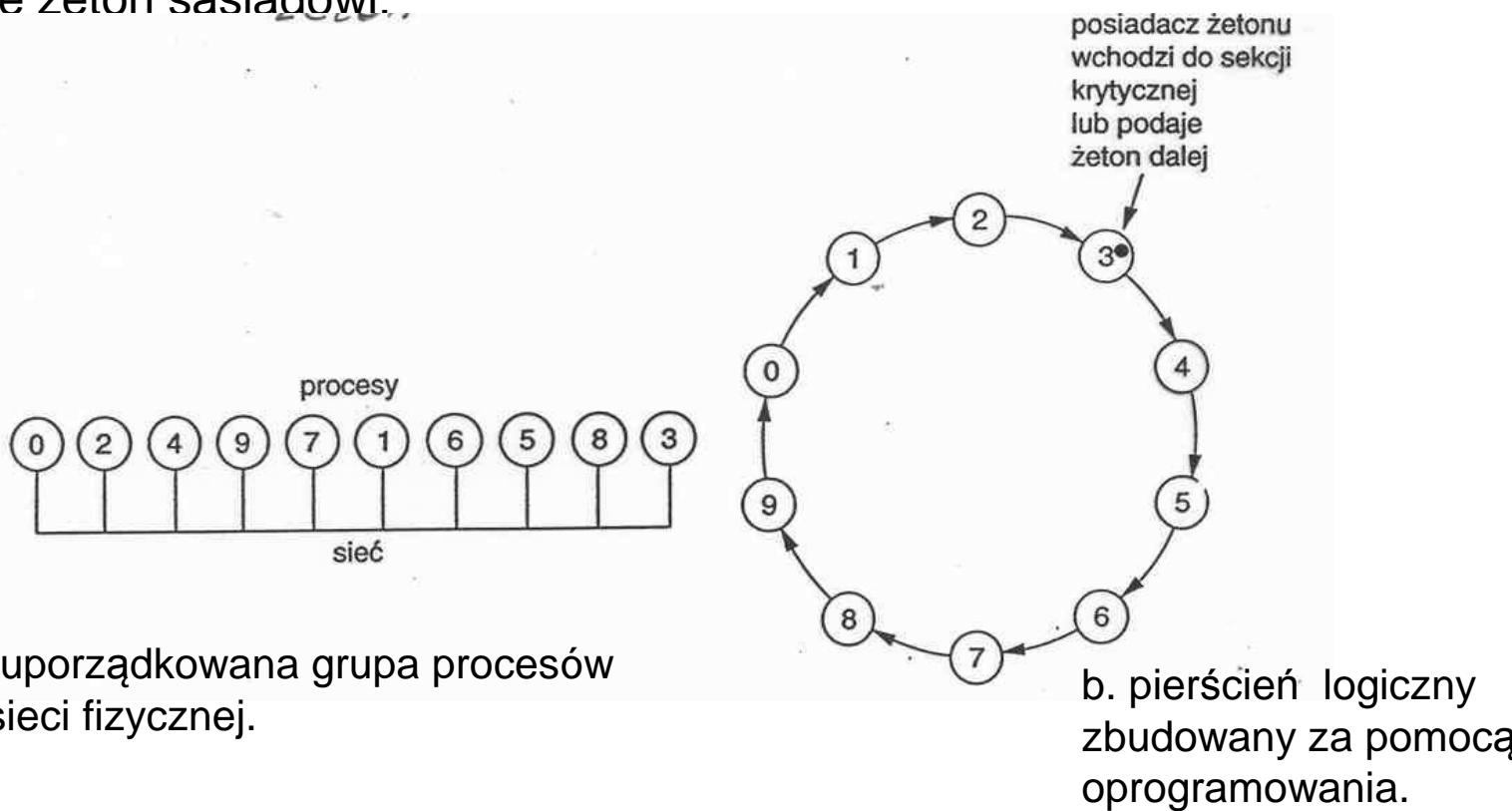
Zapewnienie wzajemnego wyłączania bez głodzenia.

Wrażliwy na awarie - brak odpowiedzi spowodowany awarią procesu jest traktowany jako brak zgody - blokowanie procesów próbujących wejść do sekcji krytycznej (jest możliwość rozwiązania). Wymagana komunikacja grupowa lub każdy proces musi utrzymywać listę procesów znajdujących się w grupie, wchodzących i wychodzących - obciążenie systemu.

## Algorytm pierścienia logicznego z żetonem

Rozpatrzmy system rozproszony, w którym zbiór procesów jest połączonych szyną. Wprowadza się logiczne (programowe) uporządkowanie procesów tworząc pierścień. W pierścieniu krąży żeton.

Proces po otrzymaniu żetonu sprawdza, czy chce wejść do sekcji krytycznej, nie - przekazuje żeton sasiadowi.  
tak - zatrzymuje żeton



### Cechy algorytmu:

Zapewnia wzajemne wyłączanie. Nie powoduje głodzenia procesów.

Powstają problemy związane z zaginięciem żetonu.

Wrażliwy na awarie procesów.

## ALGORYTMY ELEKCJI

Cel: wybór procesu, który będzie pełnił rolę koordynatora lub inicjatora w systemie rozproszonym.

### Założenia:

każdy proces ma niepowtarzalny numer,

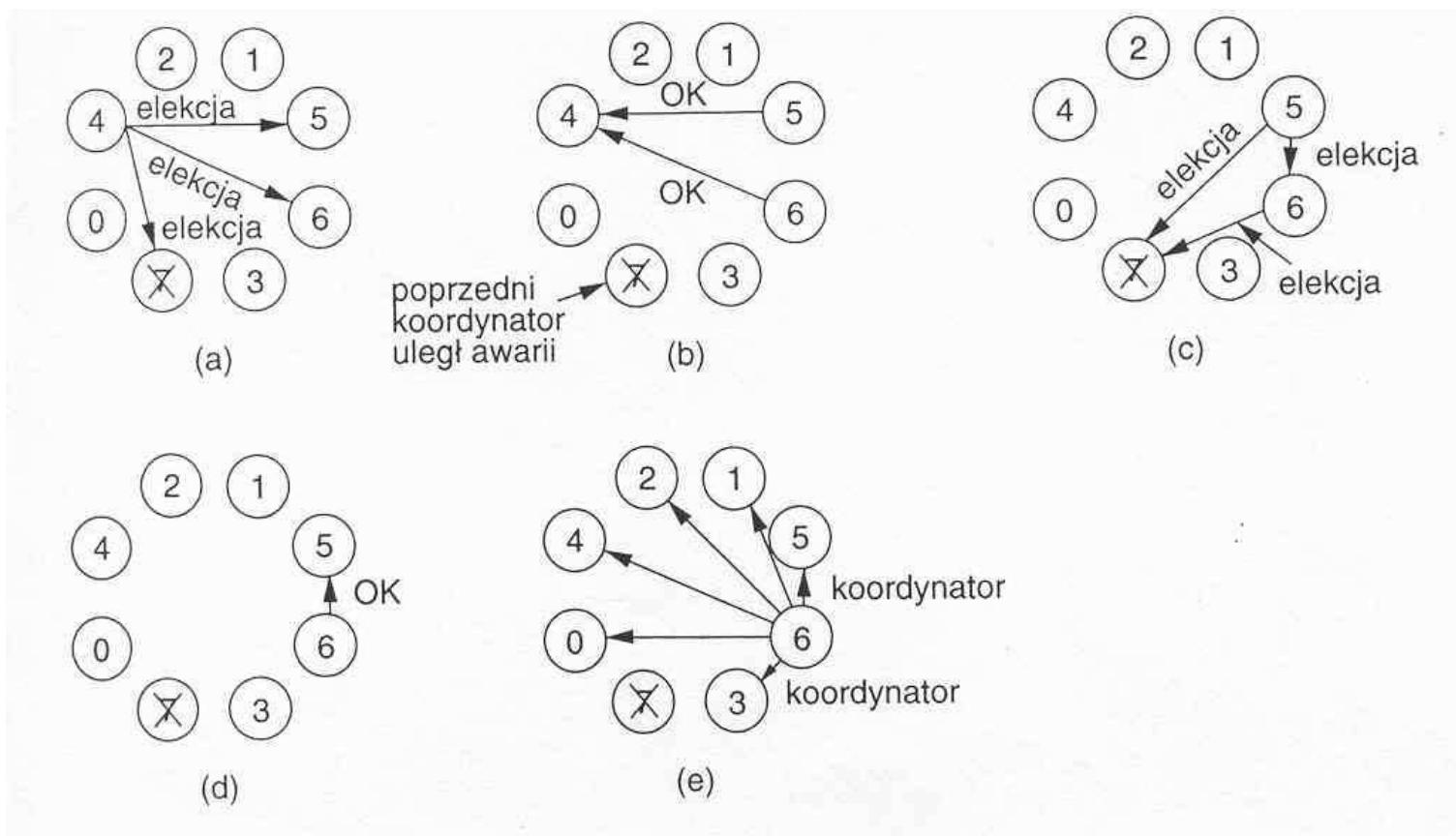
każdy proces zna numery wszystkich pozostałych,

procesy nie wiedzą, które z nich aktualnie działają, a które są unieruchomione,

próbuje się zlokalizować proces o największym numerze.

## Algorytm tyrana:

1. Proces A zauważył, że koordynator nie odpowiada.  
A wysyła komunikat ELEKCJA do wszystkich procesów z większymi numerami.
2. Brak odpowiedzi, to A zostaje koordynatorem.
3. Nadchodzi komunikat od procesu B o większym numerze.  
Proces A przestaje działać w elekcji.  
B przejmuje sterowanie i kontynuuje elekcję (zgodnie z punktami 1, 2, 3 ).
4. Proces, który wygrywa elekcję wysyła do pozostałych komunikat: KOORDYNATOR.

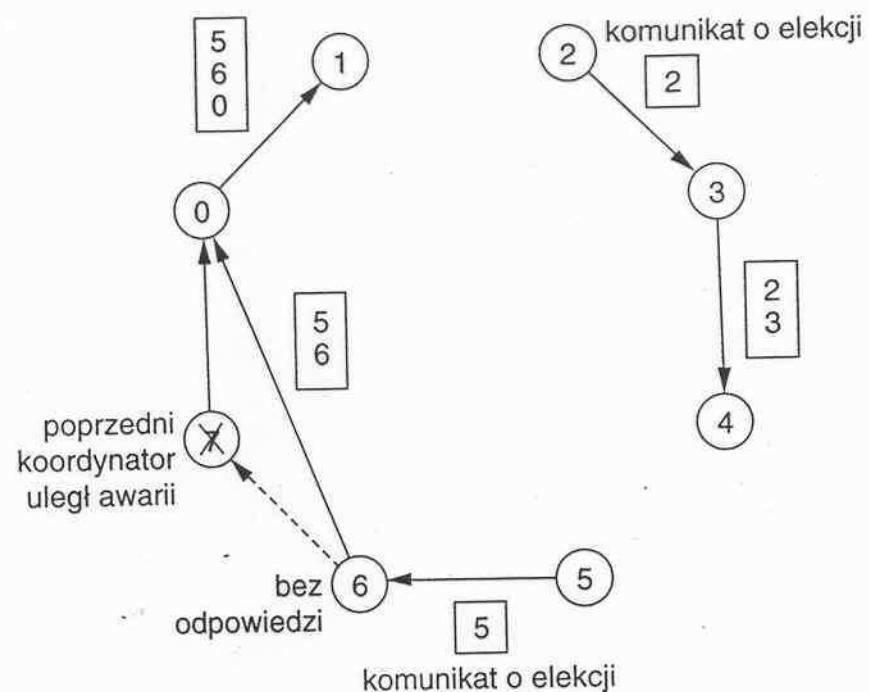


## Algorytm pierścieniowy

Założenie: procesy są fizycznie i logicznie uporządkowane. (Każdy proces przechowuje strukturę pierścienia).

### Działanie:

1. Proces A zauważył, że koordynator nie działa. Wysyła komunikat ELEKCJA do kolejnego nie wyłączonego procesu w pierścieniu. Komunikat zawiera jego numer.
2. Proces B otrzymujący komunikat ELEKCJA, dopisuje swój numer i przesyła do następnego, itd.
3. Proces A po odebraniu komunikatu z własnym numerem, wysyła komunikat: KOORDYNATOR z pełną listą procesów występujących aktualnie w pierścieniu i wskazującą proces o najwyższym numerze, który zostaje koordynatorem.
4. Koordynator rozpoczyna działanie.



## **TRANSAKCJE NIEPODZIELNE** **(atomic transactions)**

### **Wprowadzenie**

Ilustracja na przykładzie negocjacji i podpisania umowy handlowej.  
Idea transakcji w systemie komputerowym.  
Zasada wszystko albo nic.

### **MODEL TRANSAKCJI**

#### **Założenia dotyczące systemu:**

- istnieje pewna liczba niezależnych procesów .
- każdy proces może ulec awarii
- komunikacja między procesami jest zawodna

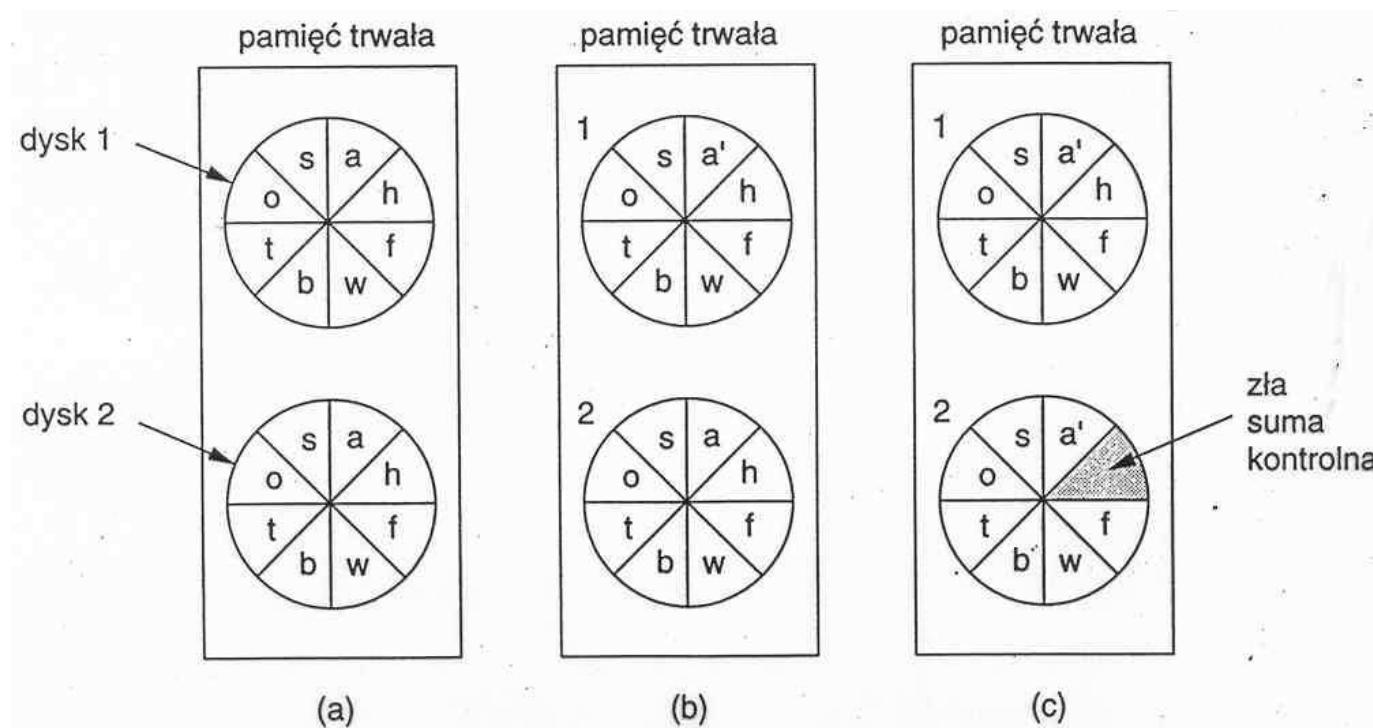
## Rodzaje pamięci

RAM

Pamięci dyskowe

**Pamięci trwałe** (stable storage) specjalnie zaprojektowane, aby mogły przetrwać możliwie wszystkie awarie (z wyjątkiem kataklizmów). Potrzebne dla realizacji transakcji niepodzielnych.

Przykład realizacji pamięci trwałej na dwóch dyskach:



(a) Pamięć trwała. (b) Awaria po zaktualizowaniu dysku 1. (c) Błędny obszar

## Transakcje elementarne

Elementarne działania niezbędne do programowania za pomocą transakcji, np.:

- Początek transakcji
- Koniec transakcji
- Zaniechanie transakcji
- Czytaj (dane z obiektu)
- Zapisz (dane do obiektu)

## Właściwości transakcji

### 1. Niepodzielność (atomicity)

Transakcja albo jest realizowana w całości, albo wcale. Jeśli następuje, to tylko jako jedno niepodzielne i natychmiastowe działanie.

### 2. Spójność (consistence)

Nie są naruszane niezmienniki systemowe.

### 3. Izolacja, uszeregowanie (isolation, serialization)

Kilka transakcji może przebiegać w tym samym czasie. Wynik końcowy powinien być taki, jakby transakcje były wykonywane po kolei (etapowo).

POCZĄTEK\_TRANSAKCJI

$x = 0;$   
 $x = x + 1;$

KONIEC\_TRANSAKCJI

(a)

POCZĄTEK\_TRANSAKCJI

$x = 0;$   
 $x = x + 2;$

KONIEC\_TRANSAKCJI

(b)

POCZĄTEK\_TRANSAKCJI

$x = 0;$   
 $x = x + 3;$

KONIEC\_TRANSAKCJI

(c)

czas →

plan 1

$x=0; \quad x=x+1; \quad x=0; \quad x=x+2; \quad x=0; \quad x=x+3;$

dopuszczalny

plan 2

$x=0; \quad x=0; \quad x=x+1; \quad x=x+2; \quad x=0; \quad x=x+3;$

dopuszczalny

plan 3

$x=0; \quad x=0; \quad x=x+1; \quad x=0; \quad x=x+2; \quad x=x+3;$

niedozwolony

(d)

(a)-(c) Trzy transakcje. (d) Możliwe plany

#### **4. Trwałość (durability)**

Po zatwierdzeniu transakcji jej wynik jest nieodwracalny i trwały.

#### **Uwagi dotyczące zagnieżdżania transakcji**

Transakcja może się rozwidlać. Tworzone są wtedy transakcje pochodne, wykonywane równolegle na różnych maszynach.

Transakcje te również mogą się rozwidlać.

Zasada trwałości dotyczy tylko transakcji pierwotnej.

Mожет быть viele transakcji pochodnych, zagnieżdżonych dowolnie głęboko.

Jeden ze sposobów realizacji transakcji: każda rozpoczęta transakcja dostaje prywatną kopię obiektów systemu i działa na kopiach (w swoim prywatnym świecie).

## METODY REALIZACJI TRANSAKCJI

### PRYWATNA PRZESTRZEŃ ROBOCZA

Proces rozpoczynający transakcję dostaje przydzieloną prywatną przestrzeń roboczą zawierającą kopie rzeczywistych obiektów.

W przypadku zatwierdzenia transakcji - zmiany przenoszone są obiekty rzeczywiste.

Zaniechanie transakcji - usunięcie prywatnej przestrzeni roboczej.

Problem - wysoki koszt kopiowania wszystkich obiektów.

Możliwe rozwiązanie: rozróżnienie obiektów do czytania i do aktualizacji, zastosowanie indeksowania.

## Wykorzystanie indeksowania

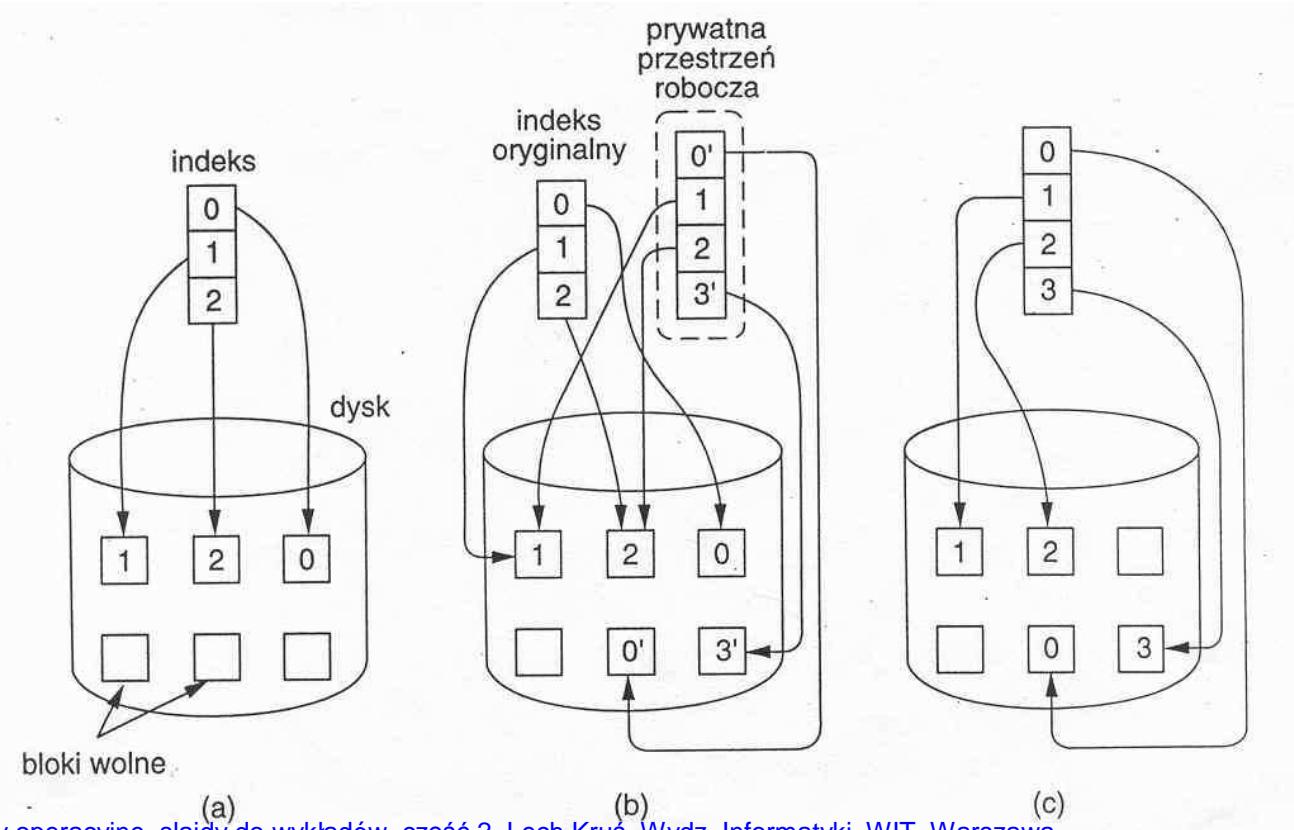
Indeks (i-węzeł w syst. UNIX) zawiera adresy dyskowe bloków pliku.

Do prywatnej przestrzeni kopiuje się tylko indeks.

Czytanie pliku - odwołanie do oryginalnego pliku.

Aktualizacja bloku - stworzenie kopii bloku, wstawienie adresu do prywatnego indeksu, aktualizacja bloku. Dodanie bloku (shadow block) - dostawienie adresu nowego bloku do prywatnego indeksu.

Zatwierdzenie transakcji - przemieszczenie prywatnego indeksu do przestrzeni procesu rodzicielskiego.



## REJESTR ZAPISÓW WYPRZEDZAJĄCYCH – LISTA ZAMIARÓW (writeahead log, intentions lists)

Idea:

Pliki są modyfikowane w miejscu ich występowania, ale przed zmianą jakiegokolwiek bloku następuje zapisanie rekordu w specjalnym rejestrze zapisów wyprzedzających, przechowywanym w pamięci trwałej.

Zapisy obejmują: transakcję, która dokonuje zmiany, jaki plik i blok jest zmieniany, starą i nową zawartość bloku.

Ilustracja zapisów w rejestrze transakcji używającej dwóch obiektów x, y.

x = 0; y=0;	rejestr	rejestr	rejestr
POCZĄTEK_TRANSAKCJI	x = 0/1	x = 0/1	x=0/1
x = x + 1;		y= 0/2	y = 0/2
y = y + 2;			x = 1/4
x = y * y;			
KONIEC_TRANSAKCJI			

## **Postępowanie w różnych sytuacjach:**

### **Zatwierdzenie transakcji**

- do rejestru wpisywany jest rekord zatwierdzenia. Zmiany w plikach są już dokonane.

### **Zaniechanie transakcji**

- wycofanie (rollback), t.j. przywrócenie stanu początkowego na podstawie zapisów w rejestrze.

**Awaria** - rejestr umożliwia rekonstrukcję danych, możliwe jest kontynuowanie transakcji lub jej odwołanie

## **PROTOKÓŁ DWUFAZOWEGO ZATWIERDZANIA TRNASAKCJI (two-phase commit protocol)**

Rozwiążanie problemu zapewnienia niepodzielności transakcji w sytuacji współpracy wielu procesów na różnych maszynach.

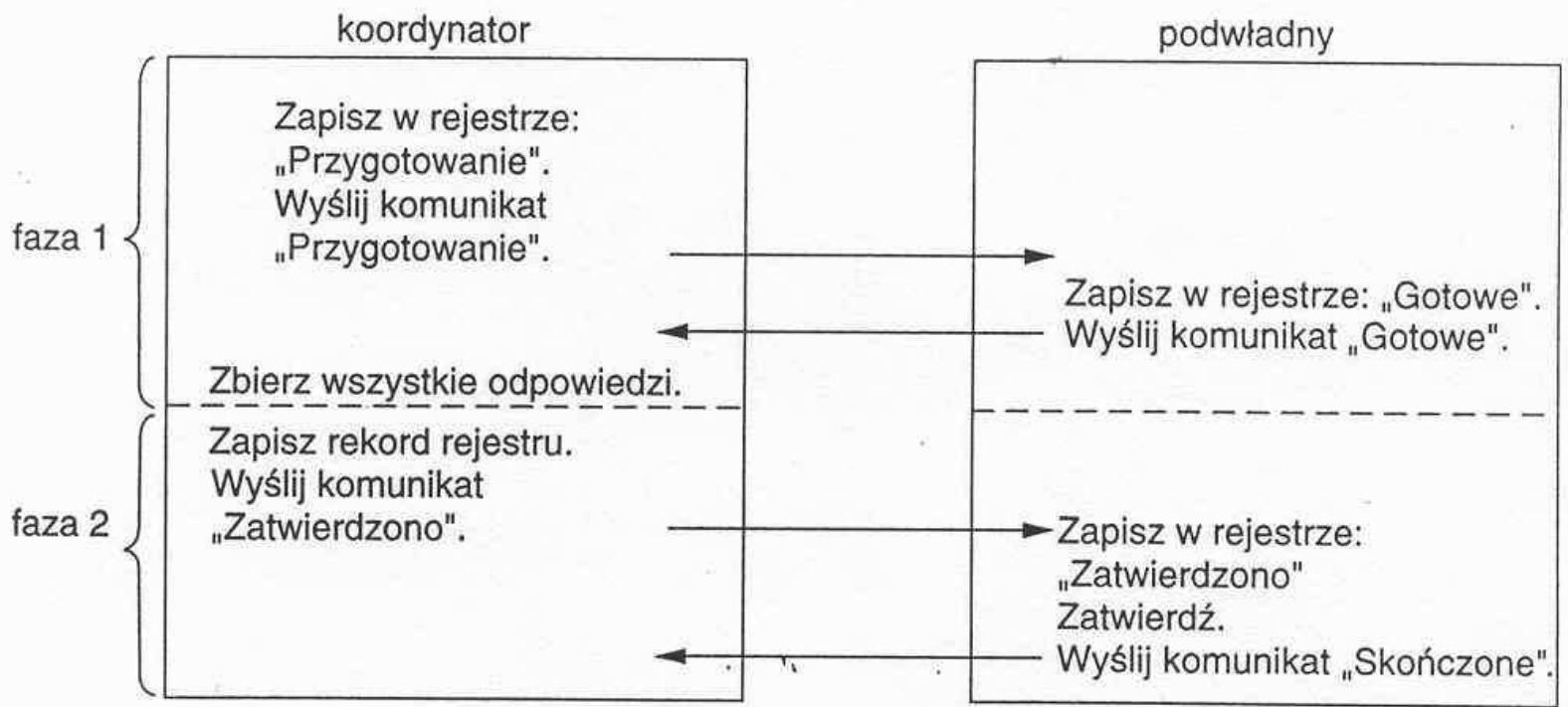
Każdy z procesów może przechowywać część obiektów modyfikowanych przez transakcję.

### **Idea:**

Jeden z procesów jest koordynatorem, pozostałe - podwładnymi.

Zastosowanie specjalnego protokołu zatwierdzania, wykorzystującego wymianę komunikatów między procesem koordynatorem, a procesami podwładnymi.

Potwierdzanie wszystkich działań zapisami w rejestrze przechowywanym w pamięci trwałej.



Ilustracja wymiany komunikatów dwufazowego zatwierdzania transakcji.

W fazie 1 - przygotowania - koordynator zleca podwładnemu uczestnikom głosowanie za lub przeciw zatwierdzeniu transakcji.

W fazie 2 – podjęcie decyzji.

### **Uwagi:**

Wszystkie decyzje zapisywane są w rejestrze.

Rejestr przechowywany jest w pamięci trwałej.

Zapis aktualnego stanu w rejestrze umożliwia kontynuację działań także w przypadku awarii.

## Postępowania w przypadku awarii

**Stany:**

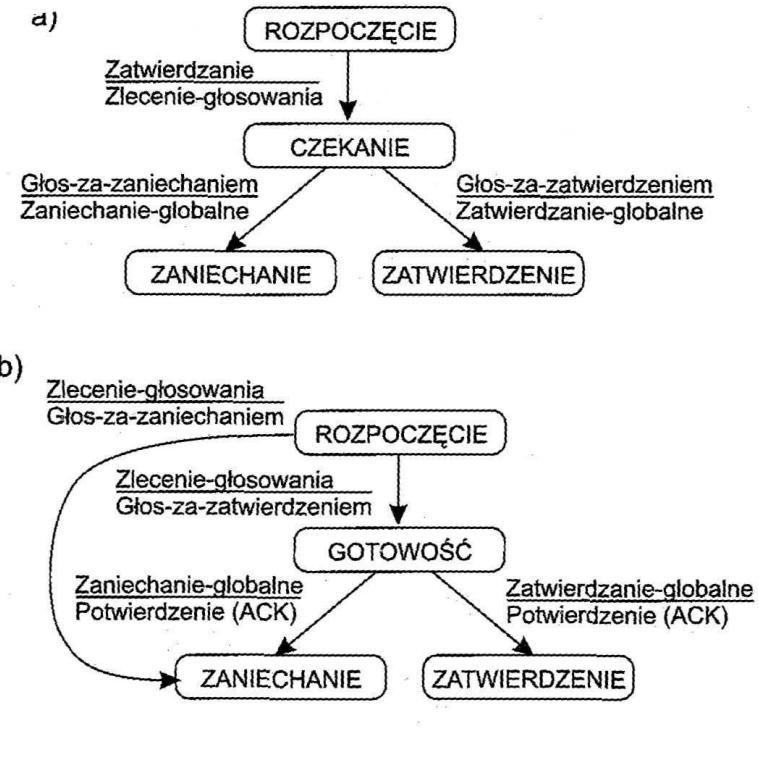
- a. maszyny koordynatora oraz
- b. maszyny uczestnika w protokole zatwierdzania dwufazowego

### Możliwe sytuacje blokowania

- blokowanie się koordynatora czekającego na głos uczestnika – przekroczenie ustalonego czasu – prowadzi do decyzji o zaniechaniu,
- blokowanie się uczestnika, długo czekającego po rozpoczęciu na zlecenie głosowania – decyzja o zaniechaniu i odpowiedni komunikat do koordynatora,
- blokowanie uczestnika czekającego na komunikat z wynikiem głosowania – jeśli komunikat nie nadjejdzie w określonym czasie – decyzja o zaniechaniu.

Przechowywanie wszystkich decyzji w pamięci trwałej umożliwia rekonstrukcję procesu z przesyaniem dodatkowych komunikatów.

Inne rozwiązanie - protokół zatwierdzanie trzyfazowego (Skeen 1981).



# ALGORYTMY NADZOROWANIA WSPÓŁBIEŻNOŚCI (concurrency control algorithms)

Mechanizmy nadzorujące współbieżne wykonywanie transakcji korzystających z tych samych danych, realizowanych przez wiele procesów na różnych maszynach.

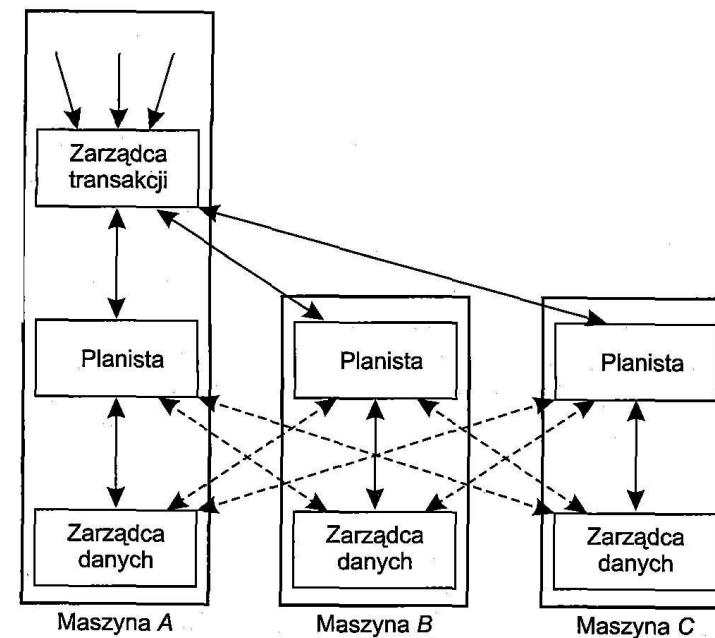
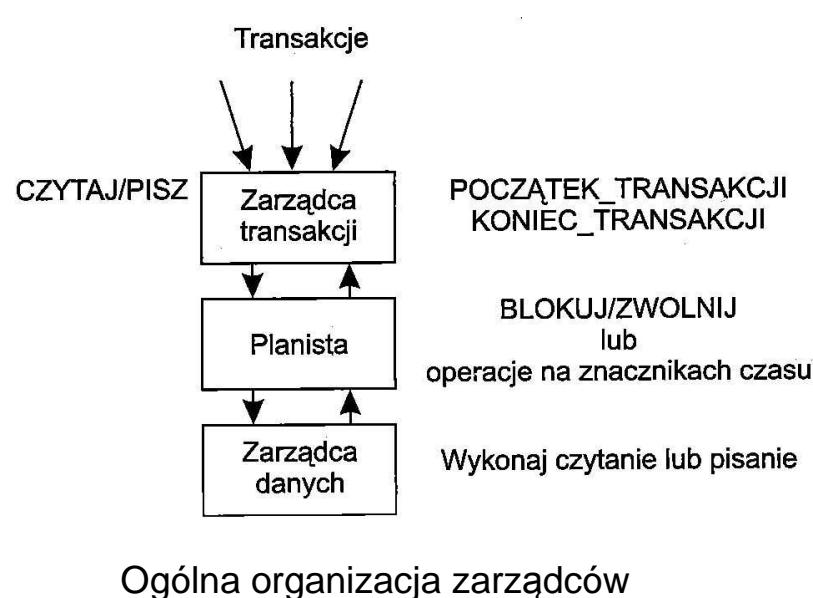
## Zajmowanie – blokowanie (locking)

Proces wykonujący transakcję zamyka – blokuje obiekt, np. plik przed korzystaniem z niego przez inne procesy. Operacja blokowania zarządzana jest przez procesy zarządzające:

Zarządcą transakcji - odpowiedzialny za niepodzielność transakcji.

Planista – odpowiada za właściwe sterowanie współbieżnością.

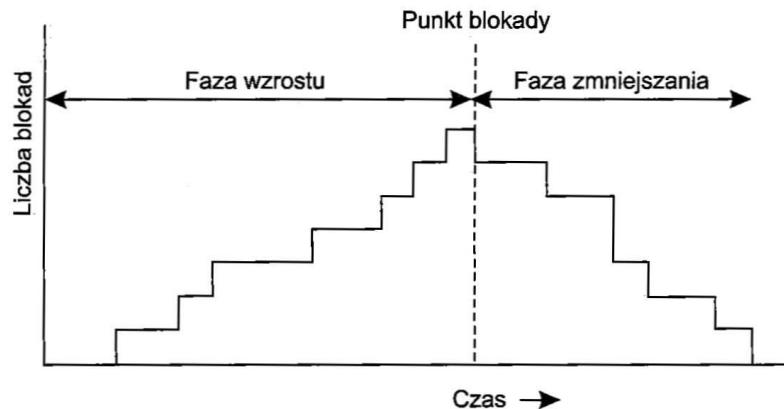
Zarządca danych – wykonuje rzeczywiste operacje czytania i zapisywania danych.



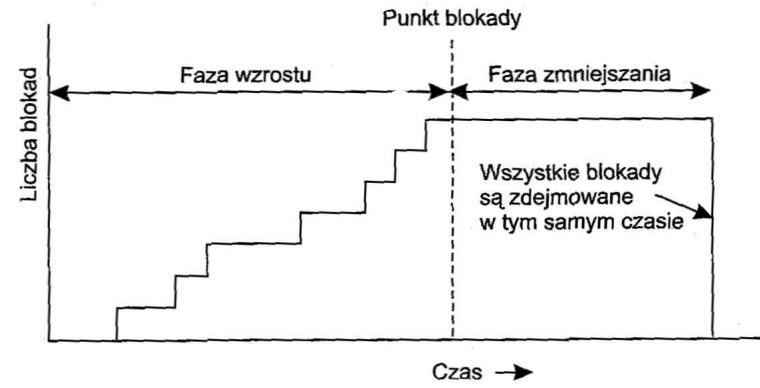
Ogólna organizacja zarządców obsługi transakcji rozproszonych

## Blokowanie dwufazowe (two-phase locking)

Schemat zajmowania obiektów, wg którego w pierwszej fazie proces tylko zajmuje – blokuje wszystkie niezbędne obiekty (faza wzrostu), a w drugiej fazie (faza malenia) - tylko zwalnia.



Blokowanie dwufazowe



Ścisłe blokowanie dwufazowe

### Uwagi:

Udowodniono, że jeśli do wszystkich transakcji stosuje się blokowanie dwufazowe, to plany realizacji tych transakcji są szeregowalne. Mogą jednak wystąpić zakleszczenia.

Zalety ścisłego blokowania dwufazowego: eliminacja „zaniechania kaskadowego” – konieczności anulowania zatwierdzonej transakcji z tego powodu, że miała kontakt z daną, której oglądać nie powinna, uproszczenie zarządzania.

Warianty rozwiązań w systemach rozproszonych :

- blokowanie scentralizowane - za nakładanie blokad odpowiada centralny zarządca,
- blokowanie z kopią podstawową – dla każdej danej jest określona kopia podstawowa, zarządca blokowania w maszynie z tą kopią odpowiada za nakładanie i zdejmowanie blokad,
- blokowanie rozproszone ze zwieleniem danych na wielu maszynach – planiści na każdej maszynie odpowiadają za blokowanie , a także za przekazywanie operacji lokalnemu zarządcy danych.

## Optymistyczne nadzorowanie współbieżności ze znacznikami czasu (optimistic concurrency control)

Metoda stosowana zwłaszcza przy wykorzystywaniu prywatnych przestrzeni roboczych.

Polega na zapisywaniu informacji, które obiekty były czytane i zapisywane, ze znacznikami czasu operacji. Wymagana jest logiczna synchronizacja czasu w systemie.

Wykonuje się transakcję nie zważając na inne.

W chwili zatwierdzania transakcji sprawdza się, czy inna transakcja nie zmodyfikowała danych po jej rozpoczęciu (na podstawie znaczników czasu):

- jeśli tak – zaniechanie transakcji,
- jeśli nie - zatwierdzenie.

Zalety w sytuacjach, gdy rzadko występują konflikty.

- Odporna na zakleszczenia, żaden proces nie musi czekać na zajęte obiekty.
- Możliwość maksymalnego zrównoleglenia działań.

Wada: konieczność powtórzenia całej transakcji, gdy po jej rozpoczęciu, inna transakcja zmodyfikowała wykorzystywane obiekty danych.

## **Pesymistyczne nadzorowanie współbieżności z zastosowaniem znaczników czasu**

Każdej transakcji przypisany jest znaczek czasu operacji elementarnej "Początek transakcji".

Zapewniona jest niepowtarzalność znaczników czasu (logiczna synchronizacja czasu - algorytm Lamporta).

Każdy plik ma skojarzony znaczek czasu czytania i znaczek czasu pisania przez ostatnią zatwierdzoną transakcję.

Znaczek czasu czytania i pisania do pliku mniejsze od znacznika czasu danej transakcji - nie ma problemu.

Sytuacja odwrotna oznacza, że po rozpoczęciu transakcji, inna, późniejsza transakcja miała dostęp do pliku.

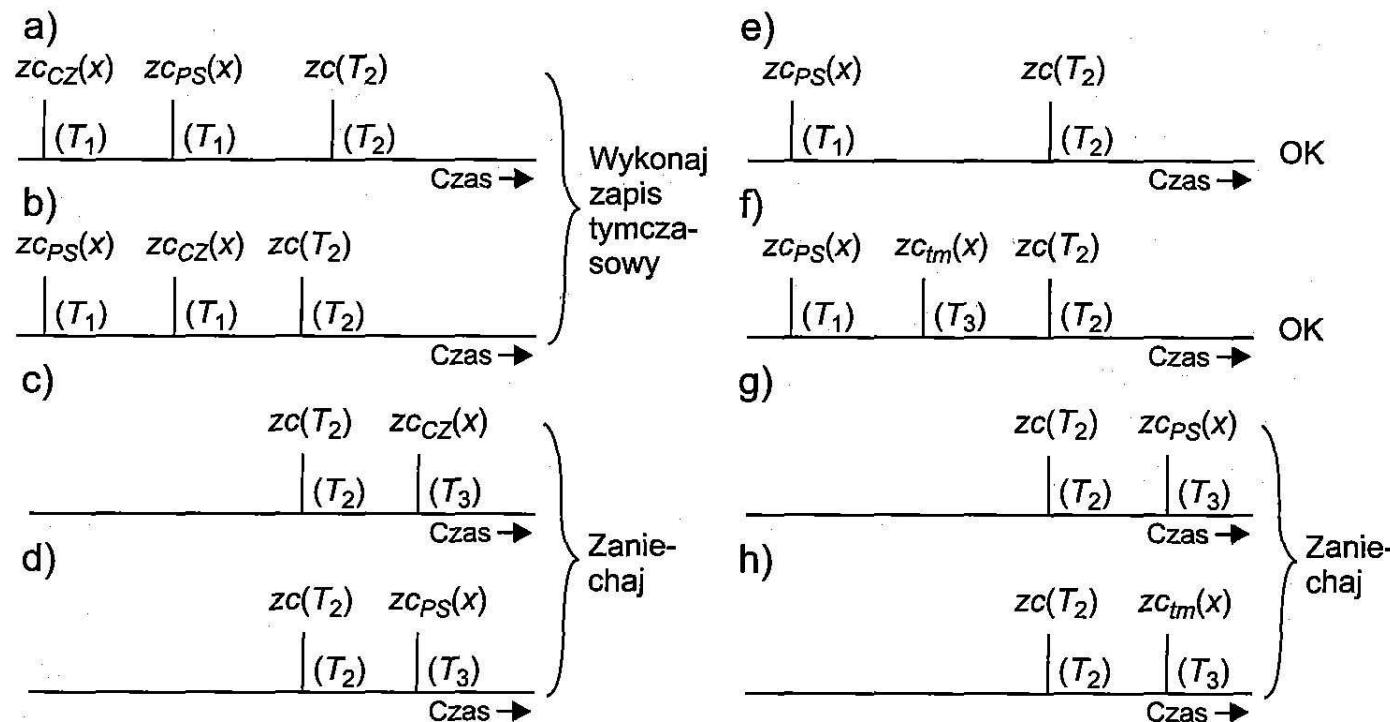
### **Uwagi:**

Stosowanie znaczników powoduje odmienne postępowanie niż w przypadku metody blokowania. Transakcja, która spotka późniejszy znaczek musi być zanegowana.

Stosowanie znaczników jest bezpieczniejsze od metod blokowania - zapobiega powstawaniu zakleszczeń.

## Przykład

Trzy transakcje  $T_1$ ,  $T_2$ ,  $T_3$  ze znacznikami czasu rozpoczęcia  $zc$ , czytania  $zc_{cz}$ , pisania  $zc_{ps}$



Sytuacje:

- a i b: późniejszy znacznik czasu transakcji  $T_2$  umożliwia jej dokonanie tymczasowego zapisu,
- c i d: działania późniejszej transakcji  $T_3$  powodują zaniechanie transakcji  $T_2$ , która chciała dokonać zapisu,
- e i f: późniejszy znacznik czasu transakcji  $T_2$  umożliwia jej dokonanie czytania,
- g i h: działania późniejszej transakcji  $T_3$  powodują zaniechanie transakcji  $T_2$ , która chciała dokonać odczytu.

## BLOKADY W SYSTEMACH ROZPROSZONYCH

Metody postępowania - analogiczne jak w przypadku systemów jednoprocesorowych:

- . Zapobieganie.
- . Unikanie.
- . Wykrywanie i usuwanie skutków.
- . Ignorowanie problemu.

### Przykłady postępowania

#### 1. Scentralizowane wykrywanie blokady

System: zbiór maszyn, jeden proces - koordynator.

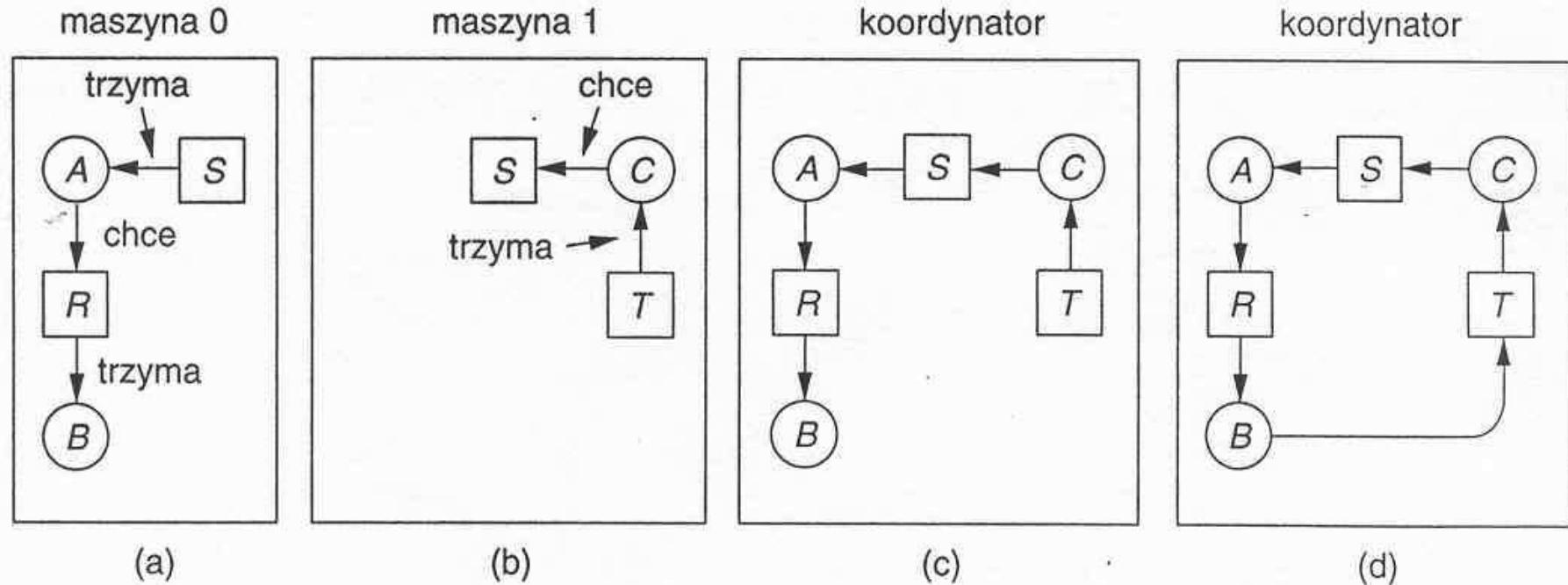
Każda maszyna utrzymuje graf własnych procesów i zasobów.

Koordynator tworzy graf dla całego systemu.

Sposoby przesyłania informacji:

1. Maszyna wysyła komunikat po każdej zmianie krawędzi w grafie.
2. Maszyna wysyła okresowo wykaz dodanych i usuniętych krawędzi:
3. Koordynator prosi maszyny o przesłanie informacji, gdy będzie mu to potrzebne.

## Przykład ilustrujący możliwość powstania tzw. fałszywej blokady:



Proces B zwolnił zasób R i zamówił zasób T. Komunikat o zwolnieniu zasobu nie dotarł na czas do koordynatora. Koordynator stwierdza cykl w grafie.

### Poprawne rozwiązanie:

Logiczna synchronizacja czasu (algorytm Lamporta).

Komunikaty zawierają znaczniki czasu.

Koordynator wykorzystując znaczniki czasu może wyeliminować tworzenie fałszywych blokad.

## 2. Rozproszone wykrywanie blokady

Przykład algorytmu Chandy-Misra- Haasa.

Proces może zamawiać wiele zasobów jednocześnie.

### Idea:

Proces oczekujący na zasób wysyła komunikat do procesu przetrzymującego ten zasób.

Komunikat zawiera:      Nr procesu rozpoczynającego czekanie.

                                  Nr procesu wysyłającego komunikat.

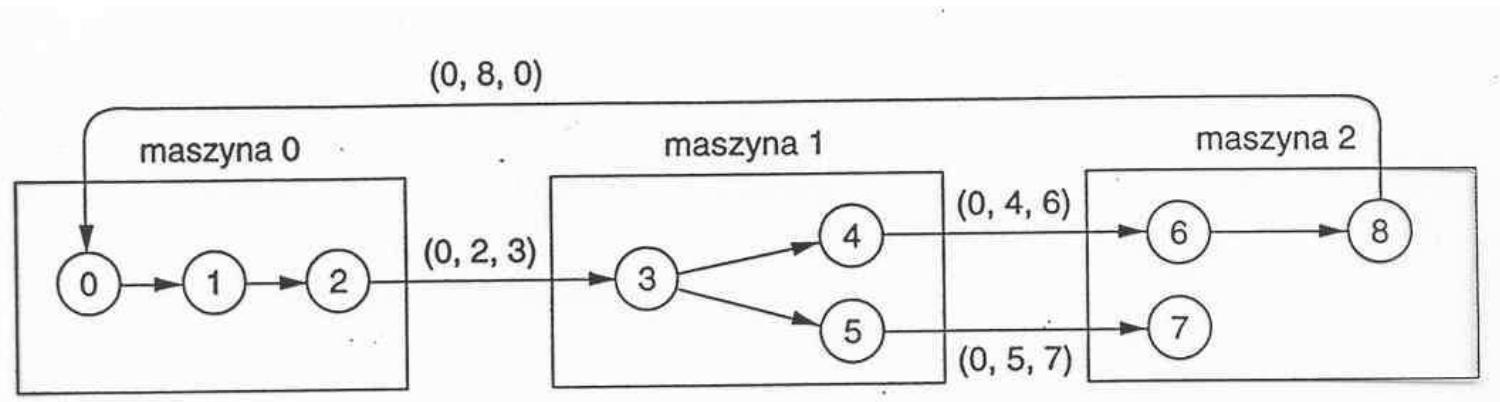
                                  Nr procesu, do którego komunikat jest wysyłany.

Odbiorca komunikatu sprawdza czy sam nie czeka, jeśli czeka to wysyła kolejny komunikat aktualizując 2-ie i 3-e pole.

Powrót komunikatu do pierwotnego nadawcy oznacza blokadę.

Sposób usunięcia blokady: np. usunięcie procesu, który zapoczątkował próbę.

**Na rysunku ilustracja dla 3 maszyn (tylko grafy oczekiwania).**



### **3. Zapobieganie blokadom w systemach rozproszonych**

przykład praktycznego podejścia wykorzystującego niepodzielne transakcje i znaczniki czasu

Każda transakcja ma jednoznacznie przyporządkowany niepowtarzalny znacznik czasu.

#### **Idea:**

Jeśli proces zamawia zasób przetrzymywany przez inny proces - sprawdza się znaczniki czasu.

Sposób postępowania:

Proces starszy zamawia zasób przetrzymywany przez proces młodszy  
- proces starszy czeka.

Proces młodszy zamawia zasób przetrzymywany przez proces starszy  
-proces młodszy ginie.

Własność algorytmu: znaczniki procesów oczekujących w łańcuchu są rosnące, zapobiega to powstaniu cyklu.

## **Sieć komputerowa**

Siecią komputerową nazywamy system informatyczny składający się z dwóch lub więcej komputerów połączonych w celu wymiany danych między nimi. Sieć może być zbudowana z wykorzystaniem urządzeń takich jak koncentratory, mosty lub/i przełączniki. Dwa ostatnie typy urządzeń (mosty i przełączniki) stosowane są do podziału sieci na segmenty. Urządzenia te służą do przekazywania informacji między segmentami, oraz ich separacji – dzięki tej funkcjonalności dane przesyłane w obrębie danego segmentu nie wydostają się poza ten segment, a dane zaadresowane do komputera znajdującego się w innym segmencie zostają wysłane tylko do tego segmentu. Zasady działania mostów (przełączników) opisuje standard RFC 802.1D.

Komputery należące do jednego segmentu współdzielą to samo medium transmisyjne. Może to prowadzić do tzw. kolizji występujących wtedy, gdy dwa lub więcej komputerów z tego samego segmentu wysyła dane w tej samej chwili. Transmisja danych nie dochodzi wówczas do skutku, a w segmencie rozchodzi się sygnał kolizji. Sygnał ten nie wydostaje się poza segment, w którym kolizja wystąpiła. Z tego powodu segmenty sieci komputerowej nazywane są niekiedy domenami kolizyjnymi. Istnieją mechanizmy, które w zależności od rodzaju sieci zapobiegają kolizjom, albo – w przypadku ich wystąpienia – zapewniają powtórzenie nieudanej transmisji. Komunikacja w obrębie sieci przebiega na bazie adresów sprzętowych.

## **Adresy sprzętowe**

Adresy sprzętowe, zwane też adresami fizycznymi, adresami MAC (ang. Media Access Control), lub adresami warstwy łącza danych modelu OSI (ang. data link layer) służą do identyfikowania hostów w komunikacji wewnętrzniejsiowej. Zwyczajowo, adresy MAC są zapisywane w postaci sześciu dwucyfrowych liczb szesnastkowych oddzielonych dwukropkami. Pierwsze trzy liczby oznaczają producenta interfejsu sieciowego (z wyjątkiem adresów broadcast i multicast). Adres taki może być typu unicast, multicast, albo broadcast. Adres unicast identyfikuje pojedynczy komputer, multicast – grupę komputerów (np. realizujących określoną usługę), broadcast – wszystkie komputery w danej sieci. Adres MAC typu broadcast to ff:ff:ff:ff:ff:ff, natomiast adresy MAC typu multicast zawierają się w przedziale 01:80:c2:00:00:00 – 01:80:c2:ff:ff:ff. Dane z adresem docelowym broadcast trafiają się do wszystkich komputerów w segmencie lokalnym oraz pozostałych segmentach, dane z adresem multicast – do komputerów w segmencie lokalnym oraz do tych segmentów,

w których znajdują się komputery należące do danej grupy multicast. Uwaga, dane z adresami multicast z przedziału 01:80:c2:00:00:00 – 01:80:c2:00:00:0f nie są przekazywane (przez mosty lub przełączniki) do innych segmentów, zatem trafiają tylko do komputerów w segmencie lokalnym. Należy podkreślić, że adresy sprzętowe umożliwiają przesyłanie danych tylko w obrębie jednej sieci, nie jest natomiast możliwe przesłanie danych między różnymi sieciami w oparciu o same adresy fizyczne.

## **Adresy logiczne**

Służą do identyfikacji hostów w komunikacji międzysieciowej. Adresy logiczne umożliwiają przesłanie danych między różnymi sieciami. Urządzenia łączące dwie lub więcej sieci noszą nazwę routerów. Analizują one adresy logiczne i zajmują się przekazywaniem danych między sieciami w sytuacji, gdy host docelowy znajduje się w innej sieci niż host źródłowy. Router posiada po jednym interfejsie do każdej z podłączonych do niego sieci. Na podstawie adresu logicznego hosta docelowego i tabeli routingu, zwanej też tabelą trasowania, router wysyła dane bezpośrednio do tego hosta (jeśli host znajduje się w jednej z przyłączonych do routera sieci), albo do jednego z routerów sąsiednich, czyli takiego, który ma interfejs do jednej z przyłączonych do routera sieci.

## **System adresacji IP (wersja IPv4)**

Jest to system adresacji logicznej stosowany jeszcze w przeważającej części Internetu. Każdy komputer podłączony do (publicznego) Internetu musi mieć unikalny adres IP. W najczęściej stosowanym zapisie adres IP składa się z czterech liczb dziesiętnych z przedziału od 0 do 255 oddzielonych kropkami. W pewnych sytuacjach stosuje się również zapis binarny (dwójkowy) i heksadecymalny (szesnastkowy). Przykładem adresu IP jest 10.1.1.20. Ten sam adres w zapisie dwójkowym to 00001010.00000001.00000001.00010100. Adres IP składa się z dwóch części – sieciowej i hostowej. Pierwsza z nich identyfikuje sieć, natomiast druga – hosta w danej sieci. Podział na część sieciową i hostową jest realizowany przy pomocy tzw. maski, która jest ciągiem 32 bitów składającym się z samych jedynek po lewej i z samych zer po prawej stronie. Przykładem maski jest adres 11111111.11111111.11110000.00000000 (w zapisie binarnym) lub 255.255.240.0 (w zapisie dziesiętnym). Liczbę jedynek występujących w postaci binarnej maski nazywamy jej długością. Maska określa, które bity adresu tworzą jego część sieciową, a które hostową. Mianowicie, część sieciową adresu tworzą te bity, które w masce są jedynkami, a część hostową – te, które w masce są zerami. W powyższym

przykładzie część sieciowa adresu składa się z pierwszych 20 bitów, natomiast hostowa – z pozostałych 12 bitów.

Maska służy do określenia sieci, w której znajduje się host o danym adresie IP, a konkretnie do określenia adresu i rozmiaru tej sieci. Adres sieci uzyskujemy w wyniku boolowskiego pomnożenia kolejnych bitów adresu odpowiednie bity maski. Ta operacja nazywa się nakładaniem maski na adres. W naszym przykładzie adres sieci to 10.1.0.0. Adres ten powstał z pomnożenia kolejnych bitów adresu 10.1.1.20 przez odpowiadające im bity maski 255.255.240. Ilustruje to poniższy rysunek.

Adres binarnie	Adres dziesiętnie
00001010.00000001.00000001.00010100	10.1.1.20
Maska binarnie	Maska dziesiętnie
11111111.11111111.11110000.00000000	255.255.240.0
Adres sieci binarnie	Adres sieci dziesiętnie
00001010.00000001.00000000.00000000	10.1.0.0

Z kolei rozmiar sieci to liczba wszystkich adresów w danej sieci. Jest ona równa  $2^n$ , gdzie n jest liczbą bitów części hostowej. Z kolei maksymalna liczba hostów, które można umieścić w danej sieci wynosi  $2^n - 2$  (zapis  $x^n$  oznacza x do potęgi n). Może się wydawać, że liczba ta powinna być równa rozmiarowi sieci, ale dwa adresy mają specjalne znaczenie. Adres IP zawierający w części hostowej same zera (w zapisie binarnym) jest adresem całej sieci, natomiast adres zawierający w części hostowej same jedynki jest tzw. adresem broadcast. Służy on do adresowania informacji przeznaczonej dla wszystkich komputerów w danej sieci, a nie tylko dla jednego z nich. Jest to tzw. broadcast skierowany (inny rodzaj adresu broadcast jest omówiony w dalszym ciągu). Pakiet z takim adresem docelowym może być przekazywany przez routery. Każdy z pozostałych  $2^n - 2$  adresów jest tzw. adresem unicast, tzn. służy do adresowania informacji przeznaczonej dla jednego, określonego tym adresem komputera.

Maska nie była potrzebna, kiedy przynależność komputera do sieci była określana tylko na podstawie tzw. klasy, do której należał dany adres. Jednak wraz z rozwojem Internetu wprowadzono możliwość podziału pełnej sieci na podsieci, przy zastosowaniu maski dłuższej niż domyślna dla pełnej sieci. Zasady takiego podziału opublikowano w dokumencie

RFC 950. Maska stała się wówczas nieodłącznym atrybutem adresu, ponieważ nie sam adres IP, ale dopiero para składająca się z adresu IP i maski daje pełną информацию o logicznej lokalizacji komputera. Poniższa tabela przedstawia podział przestrzeni adresowej IPv4 na klasy.

### Klasy adresów IPv4

Klasa	Pierwszy bajt adresu	Liczba sieci danej klasy	Część sieciowa	Część hostowa	Maska domyślna	Liczba adresów unicast w sieci
A	1-126 (127)	126 (127)	1 oktet	3 oktety	255.0.0.0	$2^{24} - 2$
B	128-191	$2^6 * 2^8 = 2^{14}$	2 oktet	2 oktety	255.255.0.0	$2^{16} - 2$
C	192-223	$2^5 * 2^{16} = 2^{21}$	3 oktety	1 oktet	255.255.255.0	$2^8 - 2 = 254$
D	224-239	Adresy typu multicast				
E	240-255	Zarezerwowane dla uprawnionych podmiotów				

**Uwaga 1:** Do adresowania komputerów są używane adresy z pierwszych trzech klas, czyli klas A, B i C. Adresy z pozostałych klas mają inne przeznaczenie.

**Uwaga 2:** Formalnie, adresy z pierwszym bajtem równym 127 należą do klasy A, ale nie są one używane do adresowania rzeczywistych interfejsów sieciowych. Za ich pomocą adresowany jest tzw. interfejs pętli zwrotnej (ang. loopback interface) używany w sytuacjach, gdy komputer (a ściślej – oprogramowanie realizujące protokół IP) wysyła informację do samego siebie.

**Uwaga 3:** Adres 255.255.255.255 (formalnie należący do klasy E) to tzw. broadcast globalny. Pakiet z tym adresem docelowym jest przeznaczony dla wszystkich komputerów w sieci lokalnej, ale nie jest przekazywany przez routery do innych sieci, czym różni się funkcjonalnie od broadcastu skierowanego.

**Uwaga 4:** Pakiety z adresami multicast z przedziału 224.0.0.0 – 224.0.0.255 nie są przekazywane (przez routery) do innych sieci, zatem trafiają tylko do komputerów w sieci lokalnej (porównaj z analogczną uwagą dotyczącą adresów MAC typu multicast).

Pewne grupy adresów IPv4 są wydzielone jako tzw. adresy prywatne. Adresy te są stosowane w sieciach prywatnych oddzielonych od publicznego Internetu specjalnymi bramkami (ang. gateway). Na tych bramkach jest wykonywana tzw. translacja adresów. W pakiecie wysyłanym z sieci prywatnej do publicznego Internetu bramka zamienia prywatny adres źródłowy (adres komputera źródłowego w sieci prywatnej) na publiczny adres źródłowy (adres bramki od strony Internetu publicznego). W pakiecie przesyłanym z publicznego Internetu do sieci prywatnej bramka zamienia publiczny adres docelowy (adres bramki od strony Internetu publicznego) na prywatny adres docelowy (adres komputera docelowego w sieci prywatnej).

### **Grupy adresów prywatnych (określone w dokumencie RFC 1918)**

10.0.0.0 - 1 sieć klasy A

od 172.16.0.0 do 172.31.0.0 - 16 sieci klasy B

od 192.168.0.0 do 192.168.255.0 - 256 sieci klasy C

Adresy z powyższych grup nie mogą być nadawane komputerom podłączonym do publicznego Internetu. Służą one do adresowania komputerów w tzw. sieciach prywatnych, które nie wchodzą w skład publicznego Internetu, ale są z nim pośrednio połączone przez tzw. bramki internetowe. Na bramkach tych działają mechanizmy translacji adresów umożliwiające obustronną komunikację między komputerami z sieci prywatnej, a tymi z publicznego Internetu.

Sieci klas A, B i C mogą być dzielone na równe rozmiarowo podsieci. W celu podzielenia sieci danej klasy na podsieci wydłuża się jej maskę domyślną o pewną liczbę bitów – k. Skutkuje to wydzieleniem  $2^k$  grup adresów (podsieci) takich, że adresy każdej grupy są jednakowe w części sieciowej, różnią się między sobą w części hostowej, pierwszy z nich ma same zera (w zapisie binarnym) w części hostowej, a ostatni ma same jedynki (w zapisie binarnym) w części hostowej.

**Tabela podziału sieci klasy C na równe podsieci**

Liczba podsieci	Maska podsieci	Liczba bitów maski	Adresy podsieci	Zakresy dostępnych adresów unicast	Adres broadcast
$2^1 = 2$	255.255.255.128	$24 + 1 = 25$	w.x.y.0 w.x.y.128	w.x.y.1 - w.x.y.126 w.x.y.129 - w.x.y.254	w.x.y.127 w.x.y.255
$2^2 = 4$	255.255.255.192	$24 + 2 = 26$	w.x.y.0 w.x.y.64 w.x.y.128 w.x.y.192	w.x.y.1 - w.x.y.62 w.x.y.65 - w.x.y.126 w.x.y.129 - w.x.y.190 w.x.y.193 - w.x.y.254	w.x.y.63 w.x.y.127 w.x.y.191 w.x.y.255
$2^3 = 8$	255.255.255.224	$24 + 3 = 27$	w.x.y.0 w.x.y.32 w.x.y.64 w.x.y.96 w.x.y.128 w.x.y.160 w.x.y.192 w.x.y.224	w.x.y.1 - w.x.y.30 w.x.y.33 - w.x.y.62 w.x.y.65 - w.x.y.94 w.x.y.97 - w.x.y.126 w.x.y.129 - w.x.y.158 w.x.y.161 - w.x.y.190 w.x.y.193 - w.x.y.222 w.x.y.225 - w.x.y.254	w.x.y.31 w.x.y.63 w.x.y.95 w.x.y.127 w.x.y.159 w.x.y.191 w.x.y.223 w.x.y.255

### **Metoda VLSM (Variable Length Subnet Mask – RFC 1009)**

Sieć danej klasy można również podzielić na podsieci o różnej wielkości. W tym celu stosowana jest tzw. metoda VLSM (ang. Variable Length Subnet Mask). Nazwa metody bierze się stąd, że do wydzielenia poszczególnych podsieci używa się masek o różnych długościach. Taki sposób podziału jest niezbędny w sytuacjach, kiedy metoda podziału na równe podsieci nie może być zastosowana ze względu na wymagania odnośnie rozmiarów poszczególnych podsieci. Założymy, że chcemy podzielić sieć klasy C o adresie 192.168.1.0 na cztery podsieci, do których ma należeć odpowiednio 100, 50, 25 i 20 komputerów. Metoda podziału na równe podsieci nie ma w tym przypadku zastosowania, ponieważ nie da się umieścić 100 adresów unicast w podsieci o rozmiarze 64, która powstaje z podziału sieci klasy C na 4 równe części. Problem ten daje się jednak rozwiązać, jeśli do podziału sieci wyjściowej zastosujemy maski o różnych długościach – 25, 26 i dwa razy po 27 bitów. Oto tabela przedstawiająca rozwiązanie:

Numer podsieci	Maska podsieci	Adres podsieci	Zakres dostępnych Adresów unicast	Adres broadcast
1	255.255.255.128	192.168.1.0	192.168.1.1 – 126	192.168.1.127
2	255.255.255.192	192.168.1.128	192.168.1.129 – 190	192.168.1.191
3	255.255.255.224	192.168.1.192	192.168.1.193 – 222	192.168.1.223
4	255.255.255.224	192.168.1.224	192.168.1.225 – 254	192.168.1.255

**Powyższy podział zrealizowany metodą podziału odcinka:**

[0-----127][128-----191][192----223][224----255]

**Uwaga:** Każda z podsieci wydzielonych metodą VLSM musi być podsiecią powstałą z równego podziału. W powyższym przykładzie pierwsza podsieć, czyli 192.168.1.0/25, jest pierwszą podsiecią powstałą z podziału na 2 równe podsieci, druga podsieć jest trzecią powstałą z podziału na 4 równe podsieci, a dwie ostatnie podsieci to siódma i ósma powstałe z podziału na 8 równych podsieci. Przy podziale metodą VLSM istotna jest kolejność wydzielania podsieci. Wynika to z faktu, że nie każdy adres z całej sieci może być adresem podsieci z niej wydzielanej. W powyższym przykładzie nie można zamienić kolejnością pierwszej i drugiej podsieci. Po takiej zamianie druga podsieć miałaby adres 192.168.1.64 i 25-bitową maskę, ale nie byłaby to podsieć powstała z równego podziału sieci 192.168.1.0 na dwie części. Ilustruje to poniższy rysunek.

[0-----63][64-----191][192----223][224----255]

Widać na nim, że grupa adresów 192.168.1.64/25 nie jest ani podsiecią 192.168.1.0/25 ani 192.168.128/25, które to podsieci powstają z równego podziału na dwie części. Poza tym, grupa adresów powstała z nierównego podziału musi być siecią IP. Grupa 192.168.1.64/25 nie jest siecią IP, ponieważ pierwszy adres grupy, czyli 192.168.1.0 1000000 (ostatni oktet zapisany binarnie, spacja oddziela część sieciową od hostowej) nie ma samych zer w części hostowej. **Dzieląc sieć na podsieci metodą VLSM należy przestrzegać następującej zasady: w zapisie bitowym adres podsieci musi mieć same zera w części hostowej.** Wynika z niej, że podsieć może być „odsunięta” od początku całej sieci tylko o wielokrotność swojego rozmiaru.

## **Zadanie do samodzielnego rozwiązania**

Podzielić pełną sieć klasy C o adresie w.x.y.0 na trzy jak najmniejsze podsieci w taki sposób, aby w kolejnych podsieciach można było umieścić odpowiednio 30, 60 i 120 komputerów.

**Wskazówka:** kolejne podsieci muszą mieć adresy w.x.y.0 albo w.x.y.32 (pierwsza), w.x.y.64 (druga), oraz w.x.y.128 (trzecia).

## **CIDR (Classless Inter-Domain Routing – RFC 1517...1520)**

Jest to wydzielanie bloków adresów będących (w nawiązaniu do podziału na klasy) podsieciami, całymi sieciami lub sieciami zagregowanymi (nadsieciemi). Blok taki jest oznaczany przez w.x.y.z/d gdzie w.x.y.z jest początkowym adresem bloku, a zarazem adresem podsieci, całej sieci, lub sieci zagregowanej, natomiast d jest liczbą bitów maski dzielącej adresy bloku na część sieciową i hostową. Maksymalna liczba adresów unicast w takim bloku wynosi  $2^d - 2$ . Pierwszy adres bloku składa się z samych zer w części hostowej i jest adresem sieci, a ostatni składa się z samych jedynek w części hostowej i jest adresem broadcast w danej sieci.

## **Przykładowa agregacja czterech sieci klasy C do jednej sieci bezklasowej**

Weźmy następujące 4 sieci klasy C: 192.168.0.0, 192.168.1.0, 192.168.2.0, 192.168.3.0

W każdej z powyższych sieci są 254 adresy unicast, łącznie 1016 adresów unicast.

Wypiszmy po kolejno wszystkie adresy z tych 4 sieci **z maską domyślną skróconą o 2 bity**:

192.168.0—0 00.0—0

...

192.168.0—0 00.1—1

192.168.0—0 01.0—0

...

192.168.0—0 01.1—1

192.168.0—0 10.0—0

...

192.168.0—0 10.1—1

192.168.0—0 11.0—0

...

192.168.0—0 11.1—1

Powyższe 1024 adresy tworzą sieć IP, ponieważ spełniają odpowiednie warunki.

Adres CIDR sieci zagregowanej: 192.168.0.0/22

Maska sieci zagregowanej binarnie: 1—1.1—1.11111100.0—0

Maska sieci zagregowanej dziesiętnie: 255.255.252.0

Zakres adresów unicast: 192.168.0.1 – 192.168.3.254 (1022 adresy)

Adres broadcast w sieci zagregowanej: 192.168.3.255

Adresy 192.168.1.0, 192.168.2.0, 192.168.3.0, oraz 192.168.0.255, 192.168.1.255, 192.168.2.255 mogą być nadawane komputerom w sieci zagregowanej, natomiast przy zachowaniu podziału na klasy są odpowiednio adresami drugiej, trzeciej i czwartej sieci, oraz adresami broadcast w pierwszej, drugiej i trzeciej sieci.

**Uwaga 1:** Nie każdy adres sieci klasy A, B lub C może być początkowym adresem bloku CIDR.

### **Przykład niepoprawnie określonego bloku CIDR**

Przykładowo, następujących ośmiu sieci klasy C: 192.168.4.0,...,192.168.11.0 nie można połączyć w sieć IP. Powstałby wtedy blok adresów o adresie 192.168.4.0/21 (8 sieci klasy C łączymy skracając domyślną maskę o 3 bity, czyli skracając ją z 24 do 21 bitów). Żeby blok był siecią IP, to pierwszy adres bloku musi mieć same zera (w zapisie binarnym) w części hostowej, czyli na ostatnich 11 bitach. Tymczasem w zapisie dwójkowym adres 192.168.4.0 przedstawia się następująco: 11000000.10101000.00000 100.00000000, więc jedenasty bit, licząc od prawej strony, jest jedynką, co przeczy powyższej zasadzie. Poprawnie natomiast jest określony blok 192.168.8.0/21 reprezentujący sieć zagregowaną z ośmiu sieci klasy C o adresach 192.168.8.0,...,192.169.15.0, a także blok 192.168.4.0/22 reprezentujący sieć zagregowaną z czterech sieci klasy C o adresach 192.168.4.0,...,192.168.7.0.

**Uwaga 2:** Sposób adresowania metodą CIDR jest w istocie zerwaniem z podziałem na klasy (stąd określenie „classless” – oznaczające „bezklasowy”). Przy zachowaniu podziału na klasy przykładowy adres 192.168.0.0/22 jest niepoprawny, ponieważ 192.168.0.0 jest adresem sieci klasy C, do której mogą należeć maksymalnie 254 komputery, natomiast maksymalna liczba adresów unicast w bloku 192.168.0.0/22 wynosi 1022 ( $2^{10} - 2$ ).

## Protokół ARP (Address Resolution Protocol)

Służy do ustalania nieznanego adresu sprzętowego (ang. MAC address) maszyny o znanym adresie IP znajdującej się w tej samej sieci. Znajomość samego adresu IP jest bowiem niewystarczająca do realizowania komunikacji wewnętrznej, która wykorzystuje adresy MAC. Protokół ARP może być stosowany w sieciach umożliwiających transmisję w trybie broadcast (Ethernet, Token Ring, FDDI). Jego działanie przebiega następująco:

**1 etap:** Stacja A wysyła w trybie rozgłoszeniowym (ang. broadcast), czyli na adres sprzętowy FF:FF:FF:FF:FF:FF, żądanie (ang. ARP request), aby stacja o danym adresie IP odpowiedziała swoim adresem MAC. Pakiet z tym żądaniem zawiera adres MAC i adres IP stacji A.

**2 etap:** Jeśli stacja o danym adresie IP jest aktywna - nazwijmy ją B, wówczas wysyła do stacji A odpowiedź (ang. ARP reply) zawierającą własny adres MAC i własny adres IP. Uzupełnia przy tym swoją tablicę ARP (ang. ARP cache) znajdująca się w pamięci podręcznej o wpis z odwzorowaniem adresu IP stacji A na jej adres MAC. Może to zrobić, ponieważ informacje te są zawarte w pakiecie z żądaniem ARP wysłanym w 1 etapie przez stację A.

**3 etap:** Po otrzymaniu odpowiedzi od stacji B, stacja A wpisuje do swojej tablicy ARP odwzorowanie adresu IP stacji B na jej adres MAC. Wpis w pamięci podręcznej ARP jest przechowywany przez określony czas (zależny od systemu operacyjnego i konfiguracji odpowiedniego parametru), a po upływie tego czasu jest usuwany.

W systemie Linux czas ten jest określany w sekundach parametrem jądra `net.ipv4.neigh.default.gc_stale_time` i można go ustawić poleceniem `sysctl`, np.

```
sysctl -w net.ipv4.neigh.default.gc_stale_time=1800
```

Protokół ARP jest uruchamiany zawsze wtedy, kiedy stacja źródłowa ma wysłać dane na określony adres IP **w tej samej sieci**, ale nie ma w swojej tablicy ARP wpisu odwzorowującego adres IP stacji docelowej na jej adres MAC. Po uzyskaniu adresu MAC stacji docelowej, stacja źródłowa umieszcza ten adres w odpowiednim polu adresowym ramki z danymi i przystępuje do ich wysyłania.

W systemie operacyjnym Linux protokół ARP jest zaimplementowany jako jeden ze składników jądra. Dostępne jest również polecenie `arp` służące do zarządzania pamięcią podręczną ARP. Oto niektóre opcje tego polecenia:

`arp -a` : Wypisuje zawartość tablicy ARP, podaje nazwy hostów. Użycie dodatkowo opcji `-n` powoduje wypisanie adresów IP zamiast nazw.

`arp -a -i <nazwa interfejsu>` : Wypisuje zawartość tablicy ARP dla wskazanego interfejsu. **W przypadku, gdy maszyna ma wiele interfejsów, dla każdego z nich istnieje osobna tablica ARP.**

`arp -d <nazwa lub adres hosta>` : Usuwa z tablicy ARP wpis dla wskazanego hosta

`arp -d *` : Usuwa wszystkie wpisy z tablicy ARP (\* zastępuje dowolny ciąg znaków)

*arp -s <nazwa lub adres IP> <adres MAC>* : Dodaje wpis do tablicy ARP odwzorowujący nazwę lub adres IP wskazanego hosta na jego adres MAC. Wpis taki nie jest automatycznie usuwany z pamięci podręcznej ARP, można go usunąć tylko ręcznie.

*ifconfig eth0 -arp* : Wyłącza ARP na lokalnym interfejsie eth0, który przestaje wysyłać i odpowiadać na zapytania ARP.

*ifconfig eth0 arp* : Włącza ARP na lokalnym interfejsie eth0

W nowszych wersjach systemu Linux jest instalowany pakiet *iproute*, udostępniający polecenie „*/sbin/ip*” służące do konfigurowania protokołu IP. Rolę polecenia *arp* spełnia polecenie „*ip neighbor*”, dające więcej możliwości. Oto kilka przykładów:

*ip neighbor show* : wypisuje zawartość tablicy ARP

*ip neighbor add <adres IP> lladdr <adres MAC> dev eth0* : dodaje wpis do tablicy ARP odwzorowujący adres IP wskazanego hosta na jego adres MAC. Jeśli chcemy, aby wpis nie był usuwany automatycznie, należy dodać opcję „*nud permanent*”.

*ip neighbor del <adres IP> dev eth0* : usuwa wpis dla hosta o podanym adresie IP z tablicy ARP interfejsu eth0

**Uwaga:** zamiast „*ip neighbor*” wystarczy pisać „*ip neigh*”.

Działanie protokołu ARP można „wymusić” wydając polecenie *arping*:

*arping <adres IP>* : wysyła nieskończoną liczbę zapytań ARP ze wskazanym adresem IP. Można to przerwać wciskając Ctrl-C (jeśli polecenie *arping* działa w pierwszym planie). Użycie opcji –f powoduje wysłanie tylko jednego zapytania.

Śledzenie działania protokołu ARP umożliwia polecenie *arpwatch*. Nie jest ono dostępne w standardowych wersjach znanych dystrybucji systemu Linux. Wymaga instalacji ręcznej (np. poleceniem *yum* w dystrybucji Fedora, albo *apt-get* w dystrybucjach Ubuntu czy Debian).

## **Protokół DHCP (Dynamic Host Configuration Protocol)**

Służy do automatycznej konfiguracji parametrów sieciowych na komputerze-kliencie. Serwer DHCP zazwyczaj znajduje się w tej samej sieci co komputer-klient. Protokół ten jest realizowany w następujących etapach:

1. Klient wysyła w sieć komunikat DHCP Discover (broadcast MAC, czyli ff:ff:ff:ff:ff:ff)
2. Serwer odpowiada komunikatem DHCP Offer, w którym są m.in. adres IP, maska, adres routera, adres serwera DNS.
3. Klient akceptuje proponowane parametry wysyłając komunikat DHCP Request
4. Serwer potwierdza przydzielenie parametrów komunikatem DHCP Ack

Parametry są dzierżawione na pewien czas zwany okresem dzierżawy. Po upływie połowy tego okresu klient wysyła do serwera żądanie odnowienia dzierżawy.

## Trasowanie (Routing)

Przed wysłaniem pakietu na określony adres IP, host źródłowy sprawdza, czy maszyna docelowa znajduje się w sieci lokalnej, czy też poza nią. Sprawdzenie to polega na nałożeniu maski, będącej lokalnym parametrem konfiguracyjnym maszyny źródłowej, na adres IP maszyny docelowej. Jeśli w wyniku tej operacji otrzymany zostanie adres sieci lokalnej, host źródłowy umieszcza w odpowiednim polu wysyłanej ramki adres MAC maszyny docelowej, a następnie wysyła ramkę bezpośrednio do niej. **Jeśli docelowy adres MAC jest nieznany (nie ma go w tablicy ARP), to do jego ustalenia stosowany jest protokół ARP.** Jeśli po nałożeniu maski na docelowy adres IP otrzymany zostanie adres inny niż sieci lokalnej, to pakiet kierowany jest do jednego z routerów znajdujących się w sieci lokalnej. Wybór routera dokonywany jest na podstawie adresu docelowego. Jeśli w sieci znajduje się tylko jeden router, to jego adres IP jest konfigurowany na hostach w tej sieci jako adres tzw. routera domyślnego (ang. default gateway). Router domyślny może być też skonfigurowany na hostach w sieci z więcej niż jednym routerem, ale wtedy niektóre pakiety będą przesyłane między routerami przed opuszczeniem sieci lokalnej. Adresy IP routerów są, podobnie jak własny adres IP i maska, lokalnymi parametrami konfiguracyjnymi każdego hosta działającego w oparciu o protokół IP. **Adresy MAC interfejsów routerów, tak samo jak adresy MAC wszystkich hostów z sieci lokalnej, ustalane są za pomocą protokołu ARP.**

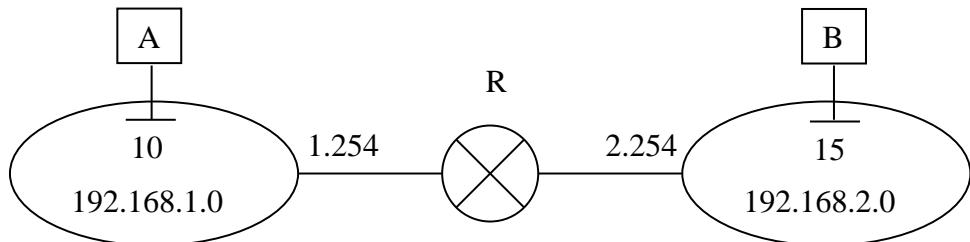
Proces przekazywania pakietów między sieciami nazywa się trasowaniem (ang. routing). Jest ono realizowane przez urządzenia trasujące (ang. router) w oparciu o docelowy adres sieciowy pakietu, oraz informacje zapisane w tablicy trasowania (ang. routing table). Po otrzymaniu pakietu router sprawdza zgodność adresu docelowego z kolejnymi wierszami tablicy i na tej podstawie przesyła pakiet do kolejnego routera albo bezpośrednio do komputera docelowego. Należy podkreślić, że tablice trasowania mają również hosty (w oparciu o nią host decyduje czy pakiet z danym adresem docelowym ma wysyłać bezpośrednio do odbiorcy w sieci lokalnej czy do routera).

Tabela trasowania routera może być konfigurowana ręcznie przez administratora – jest to tzw. routing statyczny, albo automatycznie w wyniku działania protokołu routingu – tzw. routing dynamiczny. Istnieje kilka podstawowych protokołów routingu różniących się pod względem kryterium wyboru trasy pakietów, czy obszaru działania. Jednak wszystkie te protokoły mają jedną wspólną cechę – router konfiguruje (automatycznie) swoją tablicę trasowania bazując na informacji uzyskiwanej od routerów sąsiednich, czyli znajdujących się w tych sieciach, do których jest on bezpośrednio przyłączony.

**Uwaga:** Przy zwykłym trasowaniu źródłowy i docelowy adres logiczny nie ulegają zmianie na całej trasie pakietu. Routery zmieniają natomiast adresy MAC. Podczas przekazywania pakietu router zmienia docelowy adres MAC (MAC interfejsu, na którym router odebrał pakiet) na adres MAC interfejsu następnego routera albo adres MAC interfejsu stacji docelowej. Natomiast źródłowy adres MAC (MAC interfejsu, z którego pakiet był wysłany do routera) jest zmieniany na adres MAC wyjściowego interfejsu routera.

## Przykład ilustrujący zmianę adresów MAC przez router

W sieci klasy C o adresie 192.168.1.0 znajduje się host A o adresie 192.168.1.10, natomiast w sieci klasy C o adresie 192.168.2.0 – host B o adresie 192.168.2.15. A wysyła do B pakiet danych.



### Adresy IP na całej trasie z A do B:

źródłowy: 192.168.1.10

docelowy: 192.168.2.15

### Adresy MAC na trasie z A do interfejsu 192.168.1.254 routera R:

źródłowy: MAC hosta A

docelowy: MAC interfejsu 192.168.1.254

### Adresy MAC na trasie z interfejsu 192.168.2.254 routera R do B:

źródłowy: MAC interfejsu 192.168.2.254

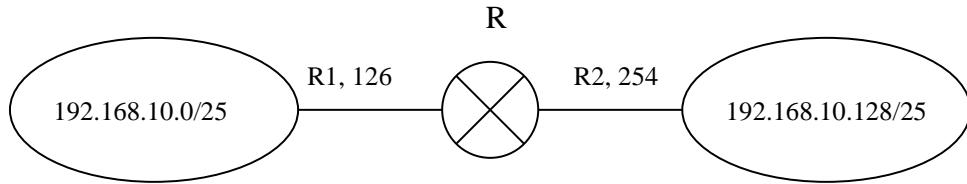
docelowy: MAC hosta B

Router R nie zmienia adresów IP – pozostają one stałe na całej trasie pakietu, natomiast musi zmienić adresy MAC, ponieważ pakiet jest przekazywany z jednej sieci do drugiej.

**Uwaga:** adresy IP nie ulegają zmianie przy zwykłym routingu, ale zmieniają się przy przechodzeniu pakietu z sieci prywatnej do publicznej albo z publicznej do prywatnej. Jest to tzw. mechanizm translacji adresów. Router, który realizuje ten mechanizm jest bramą (ang. gateway) między siecią prywatną a publiczną częścią internetu.

## Przykłady ilustrujące routing IPv4

**Przykład 1:** Sieć klasy C o adresie 192.168.10.0 jest podzielona na dwie równe podsieci maską 25 bitową, czyli 255.255.255.128. Fizyczne interfejsy routera R do pierwszej i drugiej podsieci mają nazwy systemowe R1 i R2.



Tablica routingu routera R:

Sieć (host) przeznaczenia	Maska	Adres następnego routera	Nazwa systemowa interfejsu wyjściowego
192.168.10.0	255.255.255.128	0.0.0.0	R1
192.168.10.128	255.255.255.128	0.0.0.0	R2

Wpisy dla sieci, do których router jest bezpośrednio przyłączony, są dodawane do jego tablicy trasowania automatycznie, w wyniku skonfigurowania interfejsów łączących go z tymi sieciami. Bieżący przykład przedstawia taką właśnie sytuację.

Tablica routingu przykładowego hosta z pierwszej podsieci, z interfejsem o nazwie eth0:

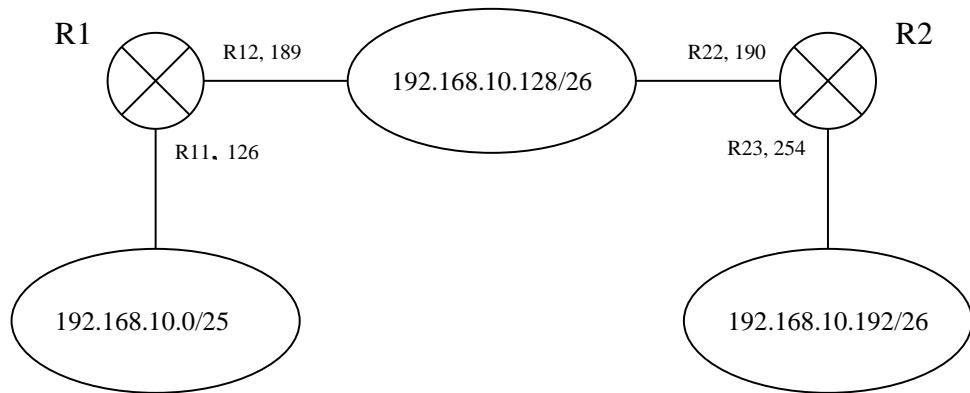
Sieć/host przeznaczenia	Maska	Adres routera	Nazwa systemowa interfejsu wyjściowego
127.0.0.0	255.0.0.0	0.0.0.0	lo (loopback)
0.0.0.0	0.0.0.0	192.168.10.126	eth0
192.168.10.0	255.255.255.128	0.0.0.0	eth0

Tablica routingu przykładowego hosta z drugiej podsieci, z interfejsem o nazwie eth0:

Sieć/host przeznaczenia	Maska	Adres routera	Nazwa systemowa interfejsu wyjściowego
127.0.0.0	255.0.0.0	0.0.0.0	lo (loopback)
0.0.0.0	0.0.0.0	192.168.10.254	eth0
192.168.10.128	255.255.255.128	0.0.0.0	eth0

## Przykład 2.

Sieć klasy C o adresie 192.168.10.0 jest podzielona metodą VLSM na trzy podsieci:



Maska pierwszej podsieci ma 25 bitów, drugiej i trzeciej – 26 bitów. Odpowiednie zakresy adresów unicast są następujące: 1 – 126, 129 – 190, 193 – 254 (ostatni bajt adresu), natomiast adresy broadcast (ostatni bajt adresu) to 127, 191 i 255. R11 i R12 to nazwy interfejsów routera R1 do pierwszej i drugiej podsieci; R21, R22 to nazwy interfejsów routera R2 do drugiej i trzeciej podsieci.

Tablica routingu R1:

Sieć/host przeznaczenia	Maska	Adres następnego routera	Nazwa systemowa interfejsu wyjściowego
192.168.10.0	255.255.255.128	0.0.0.0	R11
192.168.10.128	255.255.255.192	0.0.0.0	R12
192.168.10.192	255.255.255.192	192.168.10.190	R12

Tablica routingu R2:

Sieć/host przeznaczenia	Maska	Adres następnego routera	Nazwa systemowa interfejsu wyjściowego
192.168.10.0	255.255.255.128	192.168.10.189	R22
192.168.10.128	255.255.255.192	0.0.0.0	R22
192.168.10.192	255.255.255.192	0.0.0.0	R23

Przykładowy host z interfejsem o nazwie eth0 i adresie 192.168.10.130, znajdujący się w drugiej podsieci, powinien mieć w swojej tablicy routingu następujące wpisy:

Tabela trasowania hosta z podsieci 192.168.10.128

Sieć/host przeznaczenia	Maska	Adres routera	Nazwa systemowa interfejsu wyjściowego
192.168.10.0	255.255.255.128	192.168.10.189	eth0
192.168.10.128	255.255.255.192	0.0.0.0	eth0
192.168.10.192	255.255.255.192	192.168.10.190	eth0

Gdyby rozważany host wszystkie pakiety adresowane poza sieć lokalną wysyłał do routera domyślnego, na przykład do R2, zgodnie z następującym wpisem:

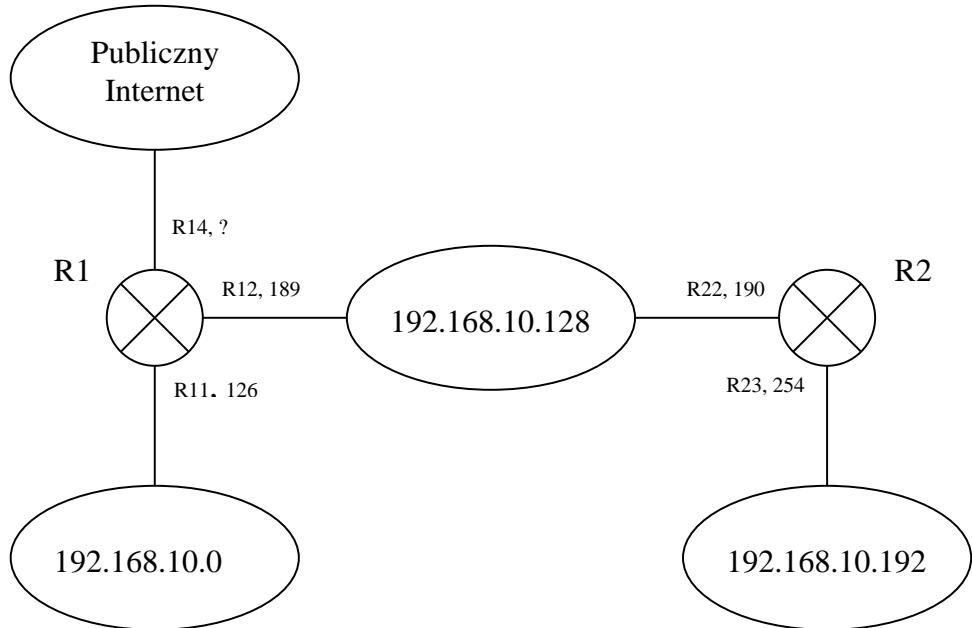
Sieć/host przeznaczenia	Maska	Adres routera	Nazwa systemowa interfejsu wyjściowego
0.0.0.0	0.0.0.0	192.168.10.190	eth0

wówczas pakiet do sieci 192.168.10.0, przykładowo do hosta o adresie 192.168.10.1, byłby najpierw skierowany do R2, stamtąd zgodnie z jego tablicą routingu – do R1, który wysłałby pakiet bezpośrednio do 192.168.10.1. W takim przypadku pakiet niepotrzebnie trafia do R2, zamiast od razu do R1.

**Uwaga:** R2 otrzymując od 192.168.10.130 pakiet skierowany do sieci 192.168.10.0 wysyła ten pakiet do R1, oprócz tego informuje hosta 192.168.10.130 (przy pomocy specjalnego komunikatu ICMP), że pakiety do sieci 192.168.10.0 powinien kierować bezpośrednio do R1.

### Przykład 3.

Sieć klasy C o adresie 192.168.10.0 jest podzielona tak jak w przykładzie 2. Dodatkowo, router R1 ma połączenie z Internetem – przez interfejs R14 o nieznanym publicznym IP.



Tablica routingu R1:

Sieć/host przeznaczenia	Maska	Adres następnego routera	Nazwa systemowa interfejsu wyjściowego
192.168.10.0	255.255.255.128	0.0.0.0	R11
192.168.10.128	255.255.255.192	0.0.0.0	R12
192.168.10.192	255.255.255.192	192.168.10.190	R12
0.0.0.0	0.0.0.0	Nieznany, publiczny IP (np. IP routera dostawcy Internetu)	R14

**Uwaga:** Sieć 192.168.10.0 (podzielona na 3 podsieci) jest siecią adresów prywatnych. W związku z tym w pakiecie wysyłanym np. z podsieci 192.168.10.128/26 do publicznego Internetu, przy przejściu przez router R1 źródłowy adres IP zmieni się na adres IP interfejsu R14, natomiast nie zmieni się adres docelowy. Z kolei w pakiecie wysyłanym z Internetu publicznego, a przeznaczonym np. dla hosta 192.168.10.150, docelowy adres IP (adres interfejsu R14) zmieni się na adres 192.168.10.150, natomiast nie zmieni się adres źródłowy.

Tablica routingu R2:

Sieć/host przeznaczenia	Maska	Adres następnego routera	Nazwa systemowa interfejsu wyjściowego
192.168.10.128	255.255.255.192	0.0.0.0	R22
192.168.10.192	255.255.255.192	0.0.0.0	R23
0.0.0.0	0.0.0.0	192.168.10.189	R22

Zauważmy, że tabeli trasowania R2, oprócz wpisów dla sieci do niego przyłączonych, jest jeszcze tylko trasa domyślna. Wynika to z tego, że jeśli pakiet ma trafić z R2 do sieci 192.168.10.0 albo do Internetu, to w obu przypadkach R2 musi wysłać ten pakiet do R1.

Przykładowy host z interfejsem o nazwie eth0 i adresie 192.168.10.130, znajdujący się w drugiej podsieci, powinien mieć w swojej tablicy routingu następujące wpisy:

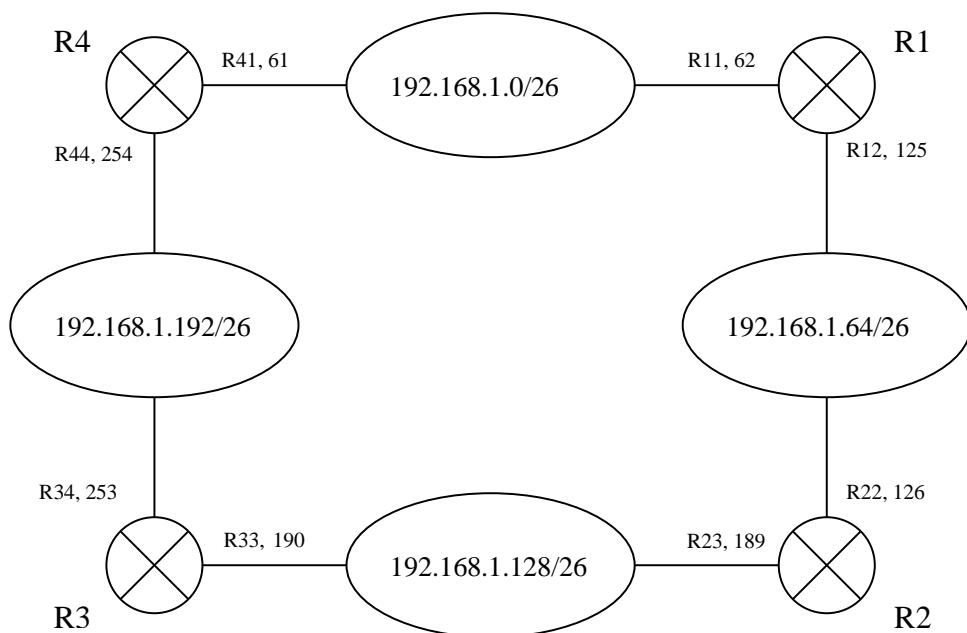
Tabela trasowania hosta z podsieci 192.168.10.128

Sieć/host przeznaczenia	Maska	Adres routera	Nazwa systemowa interfejsu wyjściowego
192.168.10.128	255.255.255.192	0.0.0.0	eth0
0.0.0.0	0.0.0.0	192.168.10.189	eth0
192.168.10.192	255.255.255.192	192.168.10.190	eth0

W odróżnieniu od przykładu 2, trasa domyślna jest w tym przypadku konieczna, przy czym uzasadnienie jest takie samo, jak dla trasy domyślnej routera R2.

#### Przykład 4.

Sieć klasy C o adresie 192.168.1.0 jest podzielona na 4 równe podsieci maską 26 bitową.  
 Nazwy interfejsów routera R1 do pierwszej i drugiej podsieci: R11, R12;  
 nazwy interfejsów routera R2 do drugiej i trzeciej podsieci: R22, R23;  
 nazwy interfejsów routera R3 do trzeciej i czwartej podsieci: R33, R34;  
 nazwy interfejsów routera R4 do czwartej i pierwszej podsieci: R44, R41.



Tablica routingu R1:

Sieć/host przeznaczenia	Maska	Adres następnego routera	Nazwa systemowa interfejsu wyjściowego
192.168.1.0	255.255.255.192	0.0.0.0	R11
192.168.1.64	255.255.255.192	0.0.0.0	R12
192.168.1.128	255.255.255.192	192.168.1.126	R12
192.168.1.192	255.255.255.192	192.168.1.61	R11
192.168.1.200	255.255.255.255	192.168.1.126	R12

**Uwaga 1:** Wpis w piątym wierszu określa trasę do pojedynczej stacji, a nie do sieci. Zgodnie z zasadą dłuższej maski, R1 wybierze trasę do 192.168.10.200 prowadzącą przez R2, a nie przez R4, ponieważ dla trasy przez R2 maska jest 32 bitowa, a dla trasy przez R4 – 26 bitowa.

**Uwaga 2:** Założymy, że działają wszystkie routery i rozważmy następującą sytuację:

Przykładowy host z interfejsem o nazwie eth0 i adresie 192.168.1.70, znajdujący się w drugiej podsieci, ma w swojej tablicy routingu następujące wpisy (routerem domyślnym jest R2):

Sieć/host przeznaczenia	Maska	Adres routera	Nazwa systemowa interfejsu wyjściowego
192.168.1.64	255.255.255.192	0.0.0.0	eth0
0.0.0.0	0.0.0.0	192.168.1.126	eth0

czyli R2 jest routerem domyślnym dla 192.168.1.70, natomiast R2 ma w swojej tablicy routingu następujący wpis (trasa do czwartej sieci przez R1):

Sieć/host przeznaczenia	Maska	Adres następnego routera	Nazwa systemowa interfejsu wyjściowego
192.168.1.192	255.255.255.192	192.168.1.125	R22

Przy tak skonfigurowanych tablicach pakiety wysłane przez 192.168.1.70, przeznaczone dla 192.168.1.200, będą „odbijane” między routerami R1 i R2 (ściślej – między interfejsami R12 i R22), bo zgodnie z 5 wierszem swojej tabeli R1 kieruje do R2 pakiety przeznaczone dla 192.168.1.200). W rezultacie pakiety te nigdy nie trafią do adresata. Można temu zaradzić dodając do tablicy routingu R2 następujący wpis:

Sieć/host przeznaczenia	Maska	Adres następnego routera	Nazwa systemowa interfejsu wyjściowego
192.168.1.200	255.255.255.255	192.168.1.190	R23

**Uwaga 3:** W protokole IP istnieje mechanizm (wykorzystujący pole TTL nagłówka IP) zapobiegający krążeniu pakietów w pętli, co może być efektem niepoprawnej konfiguracji tablic trasowania, m.in. takiej jak w powyższym przykładzie.

## Mechanizm Proxy ARP

Mechanizm ten służy do ukrycia przed hostami faktu, że znajdują się one w różnych sieciach IP (np. rozdzielonych tunelem VPN). Jest konfigurowany na routeraх i działa w ten sposób, że router odpowiada własnym adresem MAC na żądania ARP wysłane na adres IP z innej sieci. Ilustruje to następujący przykład. W środowisku przedstawionym na poniższym rysunku hosty w podsieciach 192.168.1.0/25 i 192.168.1.128/25 są skonfigurowane z maską 255.255.255.0, tak jakby znajdowały się w jednej sieci. Jeśli host o adresie 192.168.10.10 wyśle żądanie ARP na adres 192.168.10.150, to router R1 odpowie mu adresem MAC interfejsu R11. W rezultacie host ten wyśle do R1 pakiet przeznaczony dla 192.168.10.150, tak jakby wysyłał pakiet bezpośrednio do stacji docelowej, a nie do routera. Zgodnie z tabelami trasowania routerów, pakiet zostanie przesłany z R1 do R2, a następnie do hosta 192.168.1.150

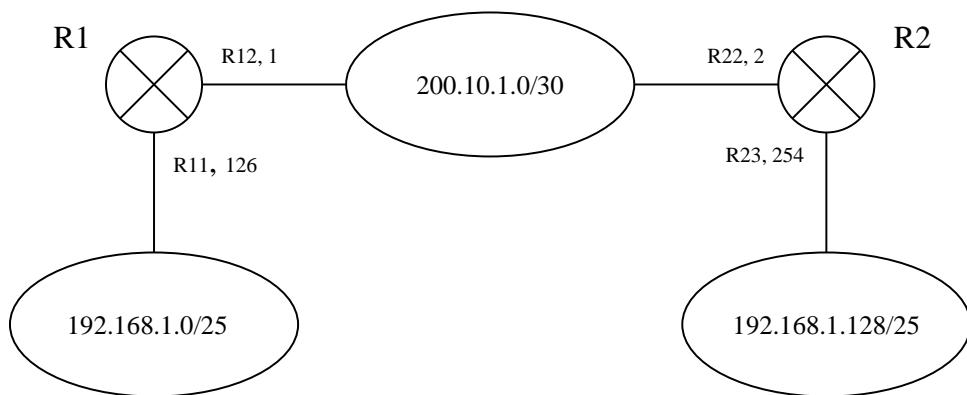


Tabela routingu R1

Sieć/host przeznaczenia	Maska	Adres routera	Nazwa systemowa interfejsu wyjściowego
192.168.1.0	255.255.255.128	0.0.0.0	R11
200.10.1.0	255.255.255.252	0.0.0.0	R12
192.168.1.128	255.255.255.128	200.10.1.2	R12

Tabela routingu R2

Sieć/host przeznaczenia	Maska	Adres routera	Nazwa systemowa interfejsu wyjściowego
192.168.1.128	255.255.255.128	0.0.0.0	R23
200.10.1.0	255.255.255.252	0.0.0.0	R22
192.168.1.0	255.255.255.128	200.10.1.1	R22

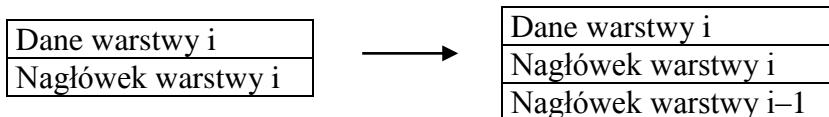
## **Model warstwowy komunikacji sieciowej. Budowa nagłówka ramki Ethernet II i pakietu IP.**

### Model warstwowy OSI: (Open Systems Interconnection)

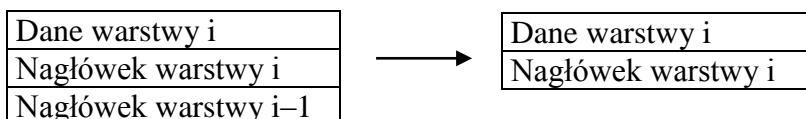
Aplikacji
Prezentacji
Sesji
Transportu
Sieci
Łącza danych
Fizyczna

Każdy protokół sieciowy można opisać przy pomocy powyższego modelu, przy czym nie zawsze do opisu protokołu używane są wszystkie warstwy. Np. protokół IP składa się tylko z warstwy sieci, natomiast nie może funkcjonować bez wsparcia dwóch warstw niższych.

Ogólna zasada stanowi, że protokół umiejscowiony w warstwie n wymaga wsparcia wszystkich warstw od 1 do n–1. Poszczególne warstwy (oprócz fizycznej) realizowane są przez oprogramowanie. W trakcie wysyłania dane są przekazywane z warstw wyższych do niższych, przy czym każda warstwa dodaje swój nagłówek. W rezultacie dane przekazywane przez warstwę i do warstwy i–1 składają się z danych i nagłówka warstwy i, co przedstawia następujący rysunek:



Po dotarciu do celu, dane przekazywane są w odwrotną stronę, mianowicie od warstw niższych do wyższych, przy czym każda warstwa usuwa swój nagłówek. Jest to przedstawione na kolejnym rysunku ilustrującym przekazywanie danych z warstwy i–1 do warstwy i:



Po dotarciu do warstwy n, usuwany jest jej nagłówek i dane ulegają przetworzeniu przez oprogramowanie tej warstwy.

### Warstwy MAC i LLC

Warstwa łączna danych podzielona jest na dwie podwarstwy – MAC (Media Access Control) i LLC (Logical Link Control). Funkcje pierwszej z nich to: odczytywanie i zapis adresów sprzętowych (MAC), definiowanie formatu ramki, realizowanie metody dostępu do medium transmisyjnego (żeton, CSMA/CD), kontrola błędów. Podstawową funkcją podwarstwy LLC jest tzw. multipleksowanie i de-multipleksowanie protokołów warstwy sieci.

## Model warstwowy TCP/IP:

Jest uproszczeniem modelu ISO, zawiera tylko 5 warstw:

Aplikacji
Transportu
Sieci
Łącza danych
Fizyczna

## Ethernet II:

W sieciach Ethernet stosowane jest kodowanie typu Manchester. Polega ono na tym, że bit 1 jest kodowany zmianą napięcia z wyższego na niższe, a bit 0 – odwrotnie. Zmiana zachodzi w połowie czasu trwania bitu. Ethernet II jest najczęściej stosowaną odmianą protokołu Ethernet. Oto budowa ramki Ethernet II:

Preambuła + SFD	Adres odb.	Adres nad.	Typ	Dane	FCS
8 oktetów	6 oktetów	6 oktetów	2 oktety	46-1500	4 oktety

Preambuła – ciąg 56 bitów (na przemian jedynki i zera) umożliwiający synchronizację nadawcy i odbiorcy. Chociaż karty sieciowe po obu stronach łącza ustawione są na tę samą prędkość transmisji, zazwyczaj między ich rzeczywistymi prędkościami jest niewielka różnica, która musi być zniwelowana w wyniku synchronizacji.

SFD – znacznik początku ramki (ang. Start-of-Frame Delimiter), czyli ciąg następujących 8 bitów: 1 0 1 0 1 0 1 1

Adres odbiorcy/nadawcy – docelowy/źródłowy adres MAC składający się z 6 oktetów

Typ – jeśli wartość w tym polu jest większa od 1500 (0x5DC), jest w nim kod protokołu warstwy sieci. Dla IP jest to 0x800, dla ARP – 0x806, dla IPX – 0x8137. W przeciwnym przypadku pole zawiera informację o **długości** pola danych i ramka nie jest typu Ethernet II, ale należy do typu Raw (standard IEEE 802.3) albo typu LLC (standard IEEE 802.2). Typ Raw nie zawiera informacji o protokole warstwy sieci i jest stosowany w sieciach Novell. W przypadku LLC, dwa pierwsze bajty znajdujące się za polem „Typ” zawierają informację o protokole warstwy sieci (DSAP, SSAP). Rozszerzeniem typu LLC jest typ SNAP.

Dane – ze względu na fizyczne parametry sieci Ethernet, wprowadzone są ograniczenia na całkowitą długość ramki. Jeśli w trakcie nadawania wystąpi kolizja, to stacja musi mieć możliwość stwierdzenia tego faktu jeszcze przed zakończeniem nadawania. Wynika stąd dolne ograniczenie na długość ramki. Ramki nie mogą być też zbyt długie, m.in. ze względu na możliwość utraty synchronizacji między stacją nadającą i odbierającą.

FCS – suma kontrolna (Frame Control Sequence), wykorzystywana wówczas, jeśli oprogramowanie warstwy łączącej danych zawiera mechanizmy sprawdzania poprawności transmisji. Jest obliczana za pomocą algorytmu CRC (ang. Cyclic Redundancy Check).

### Budowa nagłówka IPv4:

0	4	8	16	19	31							
Version	IHL	Type of Service	Total Length									
Identification		Flags		Fragment Offset								
Time To Live	Protocol		Header Checksum									
Source IP Address												
Destination IP Address												
Options			Padding									

Opis pól:

Version – numer wersji protokołu IP (4 – IPv4, 6 – IPv6)

IHL – długość nagłówka IPv4 (Internet Header Length) w słowach 32-bitowych (czyli 4-bajtowych). Jest konieczne, ponieważ w skład nagłówka IP wchodzi pole opcji o zmiennej długości. Minimalna długość nagłówka IP wynosi 20 bajtów (brak opcji), natomiast maksymalna –  $15 \times 4 = 60$  bajtów. Górnne ograniczenie na długość nagłówka IP wynika z faktu, że pole IHL składa się z 4 bitów i 15 jest największą liczbą, jaką można w nim zapisać.

TOS – typ obsługi (Type of Service), może zawierać wskazania dla routerów odnośnie wyboru trasy dla pakietu, ale zazwyczaj składa się z samych zer. Pierwsze sześć bitów tego pola tworzy tzw. pole DS CS (ang. Differentiated Services Code Point). Bit 0, 1 i 2 oznaczają ważność pakietu (precedence), bit 3 – żądanie małego opóźnienia (delay), bit 4 – żądanie dużej przepustowości (throughput), bit 5 – żądanie dużej niezawodności (reliability). W tabeli routingu też występuje rubryka TOS. Router wybiera daną trasę, jeśli zawartość pola TOS pakietu jest zgodna z zawartością rubryki TOS dla tej trasy. Bit 7 i 8 tworzą tzw. pole ECN (Explicit Congestion Notification) i są używane do powiadamiania o zatorach w sieci.

Total Length – całkowita długość pakietu IP (łącznie z nagłówkiem) mierzona w bajtach (słowach 8-bitowych). Pole to ma długość 16 bitów, z czego wynika, że maksymalna długość pakietu IP wynosi  $2^{16} - 1 = 65535$  oktetów.

Trzy kolejne pola wykorzystywane są przez mechanizmy fragmentacji i składania pofragmentowanych pakietów IP. Fragmentacja polega na dzieleniu pakietu na części o długości nie przekraczającej MTU (maksymalna długość pola danych ramki w bajtach) tej sieci, do której pakiet ma być wysłany. Przykładowo, MTU wynosi 1500 dla sieci Ethernet, a 4470 dla FDDI. Fragmentacji mogą dokonywać zarówno hosty jak i routery. Jeśli jest to konieczne, router fragmentuje pakiety przekazywane z sieci o większym MTU do sieci o mniejszym MTU. Łączenie pakietu w całość odbywa się tylko w komputerze docelowym.

**Identification** – liczba identyfikująca pakiet, jest kopowana z nagłówka całego pakietu do nagłówka każdego fragmentu, który powstaje w wyniku podziału danego pakietu. Informuje komputer docelowy, które fragmenty składają się na dany pakiet.

**Flags** – pierwszy bit tego pola jest zawsze zerem. Drugi bit (DF – do not fragment) służy do informowania routerów, czy mogą fragmentować pakiet; jeśli ten bit jest jedynką, to w przypadku konieczności fragmentowania, router odrzuca pakiet i wysyła do jego nadawcy odpowiedni komunikat. Trzeci bit (MF – more fragments) informuje, czy dany fragment jest ostatni (wartość 0), czy nie (wartość 1).

**Fragment Offset** – odsunięcie początku pola danych fragmentu od początku pola danych całego pakietu. Jest podawane w słowach 8-bajtowych (64-bitowych). Zgodnie z definicją, dla pierwszego fragmentu ma wartość zero.

**Time to Live** – oznacza maksymalny czas w sekundach, przez jaki pakiet może przebywać w sieci. Pole to jest wypełniane wartością początkową przez nadawcę pakietu. Wartość ta zależy od systemu operacyjnego. Każdy router na trasie pakietu sprawdza wartość TTL i przed wysłaniem pakietu zmniejsza ją o liczbę większą lub równą 1 (czas w sekundach, przez jaki pakiet był przetwarzany). Jeśli router otrzyma pakiet z TTL równym 1, to odrzuca go i wysyła do nadawcy komunikat o przekroczeniu TTL. W ten sposób z sieci usuwane są np. pakiety krażące w pętli, która może powstać w wyniku błędnej konfiguracji routerów.

**Protocol** – określa protokół następnej warstwy, zgodnie z opisem w dokumencie RFC 1060. Przykładowe wartości to 6 dla TCP, 17 dla UDP, 2 dla IGMP, 1 dla ICMP.

**Header Checksum** – suma kontrolna nagłówka. Przy jej obliczaniu uwzględniane są wyłącznie pola nagłówka, bez pola danych. Musi być aktualizowana na każdym routerze, ponieważ zawartość pola TTL ulega, a niektórych innych pól (np. TOS, Options) – może ulec zmianie, po przejściu pakietu przez router.

Następne dwa pola zawierają źródłowy i docelowy adres IP

**Options** – opcjonalne pole opcji. Każda opcja składa się z oktetu, w którym zapisany jest jej kod, opcjonalnego oktetu określającego jej długość, oraz opcjonalnego ciągu oktetów zawierających dane opcji. Pierwszy bit kodu opcji określa, czy opcje powinny być kopiowane do wszystkich fragmentów (wartość 1), czy tylko do pierwszego fragmentu (wartość 0). Dwa kolejne bity określają klasę opcji, natomiast pięć pozostałych – numer opcji. W poniższej tabeli przedstawione są najważniejsze opcje IP.

<b>Klasa</b>	<b>Numer</b>	<b>Długość</b>	<b>Opis</b>
0	0	1	Koniec listy opcji. Zajmuje tylko 1 oktet.
0	3	zmienna	Swobodne trasowanie według nadawcy.
0	9	zmienna	Rygorystyczne trasowanie według nadawcy
0	7	zmienna	Zapisywanie trasy pakietu.
2	4	zmienna	Zapisywanie stempli czasowych wzduż trasy.

**Padding** – wypełnienie do pełnego słowa 32-bitowego. Składa się z samych zer.

Trasowanie według nadawcy (ang. source routing) polega na określeniu trasy pakietu (adresy kolejnych routerów) przez jego nadawcę. Trasa jest więc z góry narzucona, a nie wytyczana na każdym jej etapie przez kolejne routery. Trasowanie swobodne polega na podaniu adresów tych routerów, przez które pakiet musi przejść na swojej trasie, ale oprócz nich może też przechodzić przez inne. Z kolei trasowanie rygorystyczne polega na podaniu adresów wszystkich routerów na trasie pakietu, czyli musi on przejść przez wszystkie podane routery i nie może przechodzić przez inne. Trasowanie wg nadawcy jest wykorzystywane w celach diagnostycznych (np. sprawdzenie możliwości przesłania pakietu daną trasą), albo w celu ominięcia pewnych routerów mogących znaleźć się na trasie pakietu.

#### Fragmentacja przy wysyłaniu danych w sieć Ethernet:

Struktura ramki Eth. II przenoszącej pakiet IP z nagłówkiem w podstawowej wersji (1 kreska to 1 bajt):



Jeśli pakiet IP nie mieści się w jednej ramce, to jest dzielony na fragmenty, z których każdy jest przesyłany w osobnej ramce. Parametry fragmentu znajdują się w drugim słowie nagłówka IP. Składa się ono z 3 pól: 16-bitowego pola Identyfikacja, 3-bitowego pola flag, oraz 13-bitowego pola Offset.

W pole Identyfikacja każdego fragmentu jest wstawiana zawartość pola Identyfikacja całego pakietu. Pozwala to stacji docelowej rozpoznać fragmenty pochodzące z tego samego pakietu.

Flagi: pierwsza to 0, druga to DF (ang. don't fragment), trzecia to MF (ang. more fragments)

MF=1: dany fragment nie jest ostatni  
MF=0: dany fragment jest ostatni

MTU - maximum transfer unit, maksymalna długość (w bajtach) pola danych ramki protokołu realizującego komunikację wewnętrznie sieciową (MTU=1500 dla Ethernet II)

Router znajdujący się na trasie pakietu może łączyć sieci o różnych MTU. Jeśli pakiet ma być przekazany z sieci o większym MTU do sieci o mniejszym MTU, to może być konieczna jego fragmentacja. Jeśli DF=1, to router nie przekaże dalej pakietu (bo nie zezwala na to wartość flagi DF), a do jego nadawcy wyśle odpowiedni komunikat ICMP.

Offset: odsunięcie pola danych fragmentu od początku pola danych całego pakietu, podawane w słowach 8-bajtowych

**Uwaga: Liczba bitów pola Offset to 13, więc największa liczba, którą można w nim zapisać to  $2^{13} - 1 = 8191$ . Z tego względu offset jest podawany w słowach 8-bajtowych, bo całkowita długość pakietu IP może wynosić  $2^{16} - 1 = 65535$  bajtów, więc musi być możliwe zapisywanie odstępów większych niż 8191 bajtów. W konsekwencji, długość w bajtach pola**

danych każdego fragmentu, z wyjątkiem ostatniego, musi być całkowitą wielokrotnością ośmiu!

Przykłady fragmentacji:

### Przykład 1

Przedstawić fragmenty powstałe przy wysyłaniu w sieć Ethernet (MTU=1500) pakietu IP o maksymalnej długości ( $2^{16} - 1 = 65\ 535$  bajtów razem z nagłówkiem), jeśli nagłówek IP ma standardową długość wynoszącą 20 bajtów.

Ponieważ

długość nagłówka IP + Długość pola danych fragmentu  $\leq$  MTU,

więc

**długość pola danych fragmentu  $\leq 1480$ .**

Czy pole danych fragmentu może mieć długość 1480 bajtów?

Tak, bo 1480 jest całkowitą wielokrotnością ośmiu ( $1480 = 185 \times 8$ ).

Z kolei

długość pola danych całego pakietu =  $65535 - \text{długość nagłówka IP} = 65535 - 20 = 65515$ ,  
więc przy wysyłaniu pakietu powstaną 44 fragmenty o długości pola danych 1480 bajtów,  
oraz 45-ty (ostatni) fragment o długości pola danych 395 bajtów ( $65515 = 44 \times 1480 + 395$ ). W  
standardowej notacji, czyli „długość @ przesunięcie MF/LF” (długość i przesunięcie bajtach),  
rezultat powyższej fragmentacji jest zapisywany następująco:

1 fragment: 1480 @ 0 MF

2 fragment: 1480 @ 1480 MF

3 fragment: 1480 @ 2960 MF

...

44 fragment: 1480 @ 63640 MF

45 fragment: 395 @ 65120 LF

**Uwaga:** Jeśli transmisja odbywa się w sieci Ethernet, wówczas minimalna długość pola danych ramki wynosi 46 oktetów. Przy założeniu, że nagłówek IP zajmuje 20 oktetów, dane fragmentu muszą mieć długość co najmniej 26 oktetów. **Zatem, dla pakietu IP ze standardowym (brak opcji) nagłówkiem, minimalna długość pola danych fragmentu IP wysyłanego w sieć Ethernet wynosi 26.** W razie konieczności pole danych ostatniego fragmentu uzupełniane jest bitami zerowymi.

## Przykład 2

W łącze PPP (MTU=296) wysyłany jest segment TCP zawierający 1200 bajtów danych. Zakładamy, że nagłówki IP i TCP mają po 20 oktetów. Przedstaw powstałe fragmenty używając notacji „długość @ przesunięcie MF/LF”. Długość i przesunięcie mają być podane w bajtach (w polu offset przesunięcie jest podawane w słowach 8-bajtowych).

Segment TCP = Nagłówek TCP + Dane TCP

Pakiet IP = Nagłówek IP + Dane IP = Nagłówek IP + Nagłówek TCP + Dane TCP

Długość pola danych pakietu IP = Długość nagłówka TCP + Długość pola danych TCP

**Długość pola danych całego pakietu IP =  $20 + 1200 = 1220$**

Z kolei każdy fragment zawiera nagłówek IP i pole danych, a ponieważ MTU=296, więc długość nagłówka IP + długość pola danych fragmentu  $\leq 296$ , skąd wynika, że

długość pola danych fragmentu  $\leq 276$

Czy pole danych fragmentu może mieć długość 276 bajtów?

Nie, bo  $276 = 34 \cdot 8 + 4$ , więc 276 nie jest całkowitą wielokrotnością ośmiu.

**Maksymalna długość pola danych fragmentu to  $272 = 34 \cdot 8$ .**

Powstanie zatem 5 fragmentów, czyli 4 fragmenty z polem danych o długości 272 bajty oraz piąty fragment z polem danych o długości 132 ( $1220 = 4 \cdot 272 + 132$ ). W standardowej notacji rezultat powyższej fragmentacji zapisuje się w następujący sposób:

272 @ 0 MF  
272 @ 272 MF  
272 @ 544 MF  
272 @ 816 MF  
132 @ 1088 LF

## Protokół ICMP

ICMP, opisany w dokumencie RFC 792, jest protokołem pomocniczym dla IP. Chociaż nie jest protokołem transportowym, to w modelu OSI znajduje się w warstwie czwartej, ponieważ komunikaty ICMP są przesyłane w pakietach IP. Protokół ten służy do przesyłania danych informacyjnych i komunikatów o problemach z transmisją pakietów. Ponieważ IP nie jest protokołem niezawodnym, tzn. nie zawiera mechanizmów gwarantujących dostarczenie pakietu do stacji docelowej, więc zaistniała konieczność stworzenia narzędzia pozwalającego w pewnych sytuacjach wykrywać przyczyny problemów z transmisją. Jeśli pakiet IP zawiera komunikat ICMP, to w polu „Protocol” nagłówka pakietu wpisana jest wartość 1. Nagłówek ICMP składa się z 4 lub więcej bajtów – długość nagłówka jest zależna od typu komunikatu – i znajduje się w polu danych pakietu IP, zaraz za nagłówkiem IP.

### Ogólna budowa komunikatu ICMP:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+																						
	Nagłówek IP																					
+-----+																						
	Typ		Kod		Suma Kontrolna																	
+-----+																						
	Dane - zawartość pola zależna od typu komunikatu																					
+-----+																						

### Rodzaje niektórych komunikatów ICMP

(pełna lista na stronie <https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml>):

#### **Przeznaczenie nieosiągalne (Destination Unreachable)**

Typ: 3

Kod: 0 <- sieć nieosiągalna

1 <- host nieosiągalny

2 <- protokół nieosiągalny (określony w polu "Protocol" nagłówka IP)

3 <- port nieosiągalny (określony w polu 'Destination Port' nagłówka TCP lub UDP)

4 <- konieczna fragmentacja, ale jest ustawiony bit DF

5 <- niepowodzenie trasowania źródłowego

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+																						
	Typ		Kod		Suma kontrolna																	
+-----+																						
	Dane																					
+-----+																						

Komunikat wysyłany przez router do maszyny źródłowej, jeśli w tablicach trasowania routera nie ma informacji pozwalającej na przekazanie pakietu, albo jeśli nieosiągalny jest host docelowy, do którego router ma bezpośrednio przekazać pakiet. Host docelowy również może wysłać taki komunikat, jeżeli np. port docelowy jest nieaktywny. Jeszcze jedną przyczyną wysłania tego komunikatu przez router może być konieczność fragmentacji pakietu, ale ustawienie flagi DF na to nie pozwala.

### **Przekroczenie czasu** (Time Exceeded)

Typ: 11

Kod: 0 <- przekroczenie górnej granicy TTL

1 <- przekroczenie czasu składania pakietu z fragmentów

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+		Typ		Kod		Suma kontrolna																
+-----+		Dane																				
+-----+																						

Kod 0 oznacza przekroczenie czasu życia pakietu (Time-to\_Live exceeded), czyli wyzerowanie się pola TTL po dotarciu pakietu do n-tego routera na trasie, gdzie n jest początkową wartością pola TTL umieszczaną tam przez stację źródłową.

Komunikat z kodem 1 jest wysyłany do nadawcy przez stację docelową, ponieważ tylko ona składa z powrotem pakiet z fragmentów (nie robią tego routery).

### **Problem z parametrem** (Parameter Problem)

Typ: 12

Kod: 0, problem jest identyfikowany przez wskaźnik

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+		Typ		Kod		Suma Kontrolna																
+-----+		Wskaźnik		Bity nieużywane																		
+-----+																						

Oznacza problem z przetwarzaniem parametrów nagłówka IP. Wskaźnik identyfikuje pierwszy oktet pola parametru, np. 1 odpowiada polu ToS, 20 – kodowi pierwszej opcji.

### **Wygaszanie źródła** (Source Quench)

Typ: 4

Kod: 0

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+		Typ		Kod		Suma kontrolna																
+-----+		Dane																				
+-----+																						

Komunikat wysyłany przez router lub host docelowy w przypadku, gdy nie nadąża on z przetwarzaniem napływających pakietów i następuje przepełnienie bufora. Po otrzymaniu takiego komunikatu maszyna źródłowa powinna zmniejszyć prędkość nadawania.

## **Przekierowanie (Redirect)**

Typ: 5

Kod: 0 <- przekieruj te z następujących wysyłanych pakietów, które są zaadresowane do komputera, którego adres spowodował komunikat przekierowania, oraz te, które są zaadresowane do dowolnego komputera w jego sieci (obecnie nie jest stosowany ze względu na to, że adres sieci znajdującej się za routерem nie wynika z klasy adresu docelowego i nie jest możliwy do ustalenia);

Kod 1 <- przekieruj tylko te z następujących wysyłanych pakietów, które są zaadresowane do komputera, którego adres spowodował komunikat przekierowania;

Kod 2 <- dotyczy pakietów określonych dla kodu 0, ale tylko tych, w których pole ToS jest takie samo jak w pakiecie, który spowodował komunikat przekierowania (obecnie nie jest stosowany z tego samego względu co komunikat z kodem 0);

Kod 3 <- dotyczy pakietów określonych dla kodu 1, ale tylko tych, w których pole ToS jest takie samo jak w pakiecie, który spowodował komunikat przekierowania.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+---+		Typ		Kod		Suma Kontrolna																
+----+																						
	Adres IP routera																					
+-----+		Wskaźnik		Bity nieużywane																		
+-----+																						

Komunikat wysyłany przez router R1 w sytuacji, gdy maszyna źródłowa wysyła pakiet poza sieć lokalną, kierując go do routera R1, a router R1 przekazuje ten pakiet do interfejsu routera R2 znajdującego się w tej samej sieci co maszyna źródłowa. Komunikat ten informuje maszynę źródłową, że powinna kierować pakiet bezpośrednio do R2.

## **Żądanie echo lub odpowiedź na echo (Echo Request, Echo Reply)**

Typ: 8, Kod: 0 <- Echo Request

Typ: 0, Kod: 0 <- Echo Reply

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1		
+-----+		Typ		Kod		Suma kontrolna																
+-----+																						
	Identyfikator		Numer sekwencyjny																			
+-----+		Dane ...																				
+-----+-----+																						

Identyfikator pozwala zestawić komunikaty Echo Request z odpowiadającymi im komunikatami Echo Reply. Na przykład, identyfikator może odgrywać rolę taką, jak numer portu w protokole TCP lub UDP, czyli identyfikować połączenie, natomiast numer sekwencyjny może być zwiększyony o 1 w każdym kolejnym pakiecie Echo Request. W pakiecie Echo Reply są zwracane te same wartości.

**Stempel czasowy lub odpowiedź na stempel czasowy** (Timestamp Request, Timestamp Reply)

Typ: 13 <- Timestamp  
14 <- Timestamp Reply

Kod: 0

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+		Typ		Kod		Suma kontrolna																
+-----+		Identyfikator		Numer sekwencyjny																		
+-----+		Czas wysłania																				
+-----+		Czas odebrania																				
+-----+		Czas transmisji																				
+-----+																						

**Żądanie informacji lub odpowiedź na żądanie informacji** (Information Request, Information Reply)

Typ: 15 <- Information Request  
16 <- Information Reply

Kod: 0

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+		Typ		Kod		Suma Kontrolna																
+-----+		Identyfikator		Numer sekwencyjny																		
+-----+																						

**Wykrywanie routerów** (Router Discovery)

Typ: 9 <- Router advertisement  
10 <- Router solicitation

Kod: 0

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
+-----+		Typ		Kod		Suma Kontrolna																
+-----+		Dalsze pola zajmowane tylko przez komunikat „Advertisement”																				
+-----+																						

Powyższe komunikaty mają na celu poinformowanie hostów o obecności routera w sieci, obecnie są rzadko stosowane, ponieważ zastępuje je protokół DHCP.

## Testowanie komunikacji na poziomie IP - polecenie ping

Polecenie to wykorzystuje komunikaty ICMP „Echo Request” i „Echo Reply” do stwierdzenia, czy istnieje komunikacja w warstwie IP między maszyną lokalną, a wskazaną maszyną zdalną. Składnia polecenia w systemie Linux jest następująca:

```
ping opcje adres_lub_nazwa_maszyny_zdalnej
```

Oto niektóre przydatne opcje

-c liczba\_pakietów : wysyła podaną liczbę komunikatów „Echo Request”

-n : wypisywane są adresy IP zamiast nazw, stosuje się w przypadku nie działających mechanizmów rozwiązywania nazw (DNS)

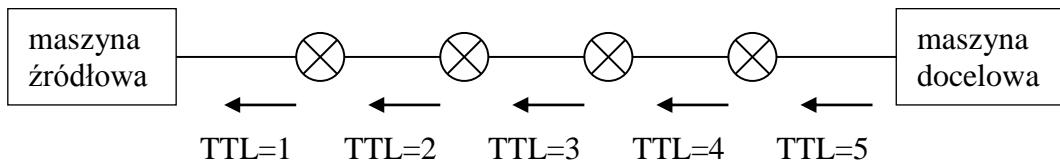
-s rozmiar\_pakietu : wysyła komunikaty ICMP o wskazanej długości pola danych

-R : żądanie zapisywania trasy w polu opcji nagłówka IP

pozostałe opcje są opisane w podręczniku man (man ping).

## Śledzenie trasy pakietów - polecenie traceroute

Polecenie traceroute generuje pakiety z małymi wartościami pola TTL. Dla przypomnienia, TTL (Time To Live) jest polem nagłówka IP mającym zapobiec powstawaniu pętli. Każdy router przekazujący pakiet zmniejsza wartość tego pola o 1. Router odrzuca każdy pakiet z wartością TTL równą zero i wysyła do nadawcy pakiet komunikat ICMP o przekroczeniu czasu (Time Exceeded). Przez wysyłanie po kolejni pakietów z małymi wartościami TTL (1,2,3,...) traceroute powoduje, że routery znajdujące się na trasie pakietu wysyłają pakiety ICMP, identyfikując się w ten sposób. Pakiet z TTL równym 1 powoduje wysłanie komunikatu przez pierwszy router, z TTL równym 2 – przez drugi, itd.



Router wysyła komunikat ICMP przez ten sam interfejs, przez który otrzymał pakiet pochodzący od polecenia traceroute.. Adres tego interfejsu jest podawany w komunikacie ICMP jako adres źródłowy. Oprócz adresów IP, traceroute raportuje czas przesłania pakietu tam i z powrotem. Dodatkowe komunikaty ICMP messages (np. o nieosiągalnej maszynie lub sieci docelowej) są przedstawiane w postaci zakodowanej, na przykład !N oznacza nieosiągalną sieć, !H oznacza nieosiągalnego hosta, itd.

Składnia polecenia:

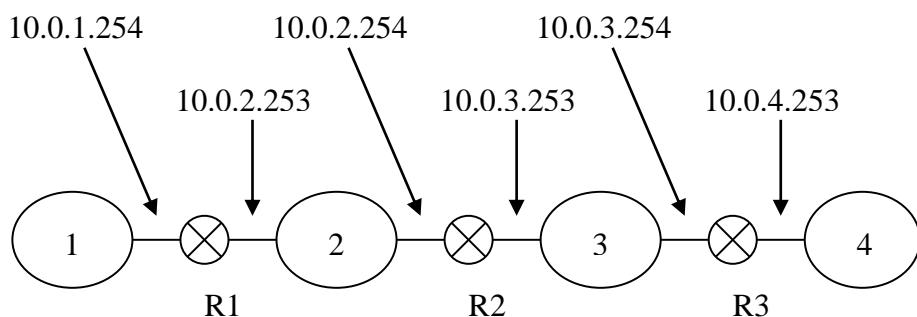
traceroute <opcje> <nazwa lub adres IP stacji docelowej>

niektóre opcje:

- f czas\_TTL : ustawia początkową wartość wstawianą w pole TTL pierwszego wysyłanego pakietu (domyślnie 1)
- F : ustawia bit „nie fragmentuj”
- I : używa pakietów ICMP ECHO zamiast datagramów UDP
- m czas\_TTL: ustawia maksymalną wartość wstawianą w pole TTL ostatniego wysyłanego pakietu (domyślnie 30)
- n : wypisuje adresy IP zamiast nazw
- p bazowy\_port\_UDP : ustawia bazowy port UDP w pakietach wysyłanych jako datagramy UDP (domyślnie 33434), zakłada się, że w maszynie docelowej na portach od bazowy do bazowy + liczba\_skoków – 1 nie nasłuchuje żadna aplikacja, a w konsekwencji host docelowy odpowie pakietem ICMP „PORT UNREACHABLE” (Port nieosiągalny)
- t wartość\_TOS : ustawia żądaną wartość (dziesiętną) w polu TOS (domyślnie 0), opcja przydatna do badania, czy dla różnych wartości TOS pakiety przesyłane są różnymi trasami, przydatne wartości to 16 (małe opóźnienie), czy 8 (wysoka przepustowość)

pozostałe opcje są opisane w podręczniku man (man traceroute).

Konfiguracja sprzętu dla przykładów 1 i 2:



Polecenia konfigurujące router R1:

Konfiguracja interfejsów sieciowych:

```
ifconfig eth0 10.0.1.254 netmask 255.255.255.0 broadcast 10.0.1.255
```

```
ifconfig eth0:1 10.0.2.253 netmask 255.255.255.0 broadcast 10.0.2.255
```

Konfiguracja tablicy trasowania:

```
route add default gw 10.0.2.254
```

Polecenia konfigurujące router R2:

Konfiguracja interfejsów sieciowych:

```
ifconfig eth0 10.0.2.254 netmask 255.255.255.0 broadcast 10.0.2.255
```

```
ifconfig eth0:1 10.0.3.253 netmask 255.255.255.0 broadcast 10.0.3.255
```

Konfiguracja tablicy trasowania:

```
route add -net 10.0.1.0 netmask 255.255.255.0 gw 10.0.2.253
```

```
route add -net 10.0.4.0 netmask 255.255.255.0 gw 10.0.3.254
```

Polecenia konfigurujące router R3:

Konfiguracja interfejsów sieciowych:

```
ifconfig eth0 10.0.3.254 netmask 255.255.255.0 broadcast 10.0.3.255
```

```
ifconfig eth0:1 10.0.4.253 netmask 255.255.255.0 broadcast 10.0.4.255
```

Konfiguracja tablicy trasowania:

```
route add default gw 10.0.3.253
```

### Przykład 1. Śledzenie pakietów przy pomocy polecenia ping

Wydaj polecenie ping na maszynie z sieci 10.0.1.0 do maszyny z sieci 10.0.4.0. Zaobserwuj otrzymywane komunikaty w przypadku, gdy włączone są wszystkie interfejsy routerów, oraz interfejsy maszyn źródłowej i docelowej. Następnie wyłącz kolejno: interfejs maszyny docelowej, interfejs 10.0.4.253 routera R3, interfejs 10.0.3.254 routera R3, interfejs 10.0.3.253 routera R2, interfejs 10.0.2.254 routera R2, interfejs 10.0.2.253 routera R1, oraz interfejs 10.0.1.254 routera R1, obserwując przy tym otrzymywane komunikaty.

### Przykład 2. Śledzenie pakietów przy pomocy polecenia traceroute

Na maszynie z sieci 10.0.1.0 wydaj polecenie traceroute, wskazując jako docelową, maszynę z sieci 10.0.4.0. Zaobserwuj otrzymywane komunikaty. Następnie, posługując się programem Wireshark, przeprowadź analizę pakietów generowanych przez polecenie traceroute.

### Przykład 3. Analiza komunikatów Echo Request i Echo Reply

Skonfiguruj analizator Wireshark tak, aby przechwytywał tylko pakiety ICMP wymieniane między komputerami twoim i sąsiada. W tym celu użyj następującego wyrażenia filtrującego:

icmp and host <nazwa lub IP własnego komp.> and host <nazwa lub IP komp. sąsiada>

Następnie wydaj polecenie

ping -c3 <nazwa lub IP komp. sąsiada>

Przeanalizuj wysyłane i otrzymywane komunikaty ICMP (w sumie powinno ich być 6), szczególną uwagę zwróć na pola „Identifier” i „Sequence Number” występujące w nagłówku ICMP.

## Protokół TCP

### Uwagi ogólne:

TCP jest protokołem transportowym zapewniającym niezawodną wymianę danych między dwiema częściami aplikacji sieciowej. Jest protokołem połączniowym, tzn. tworzy połączenie wirtualne między dwoma punktami końcowymi. Punkt taki jest parą składającą się z adresu IP oraz numeru portu. TCP działa trójetapowo, przy czym poszczególne etapy to: otwieranie, utrzymywanie i zamknięcie połączenia. Efektywnie, połączenie wirtualne jest równoważne niezawodnemu połączeniu fizycznemu.

Segmenty (jednostki danych protokołu TCP) przesyłane między dwiema częściami aplikacji sieciowej mogą ulec zniekształceniu, zagubieniu, powtórzeniu, opóźnieniu, oraz dotrzeć w kolejności innej niż ta, w jakiej zostały wysłane. Protokół TCP zapewnia, że wysłane segmenty dotrą do właściwej aplikacji w komplecie, bez zniekształceń, w prawidłowej kolejności i bez powtórzeń.

Połączenie TCP jest realizowane między parą tzw. punktów końcowych (ang. connection endpoints lub sockets), gdzie punkt końcowy (gniazdo) jest parą złożoną z adresu IP i numeru portu identyfikującego aplikację sieciową, w ramach której połączenie jest nawiązywane.

Porcja informacji przesyłana w ramach sesji TCP jest nazywana segmentem. Segment TCP składa się z nagłówka TCP oraz danych TCP.

Dane TCP przesyłane w jedną stronę tworzą strumień, czyli uporządkowany, spójny ciąg bajtów. Każdy oktet danych należący do strumienia ma swój numer porządkowy zwany numerem sekwencyjnym oktetu. Strumień danych jest identyfikowany przez numer generowany losowo podczas otwierania połączenia TCP. Chodzi przy tym o to, aby uniknąć przemieszania danych należących do różnych sesji TCP. Jeśli strumień identyfikowany jest numerem x, to x+1 jest numerem sekwencyjnym pierwszego oktetu strumienia. Numer sekwencyjny pierwszego oktetu danych segmentu jest zarazem numerem sekwencyjnym tego segmentu.

1 2 3 4 5 <- to jest strumień (bajtów)

1 3 5 4 2 <- to nie jest strumień (zmieniona kolejność)

1 2 4 6 7 <- to też nie jest strumień (są ubytki)

1 2 2 3 3 <- to też nie jest strumień (są powtórzenia)

## Budowa nagłówka TCP

<u>16-bit source port number</u>	<u>16-bit destination port number</u>
<u>32-bit sequence number</u>	
<u>32-bit acknowledgement number</u>	
<u>4-bit header length</u>	<u>6-bits reserved</u>
	U A P R S F
<u>16-bit TCP checksum</u>	<u>16-bit window size</u>
<u>16-bit urgent pointer</u>	
<u>Options (if any) Max 40-bytes</u>	
<u>Data</u>	

Znaczenie pól:

Numer portu źródłowego (Source port number): 16 bitów  
 Identyfikuje aplikację źródłową.

Numer portu przeznaczenia (Destination port number): 16 bitów  
 Identyfikuje aplikację docelową.

Numer sekwencyjny (Sequence number): 32 bity  
 Numer porządkowy pierwszego oktetu danych przesyłanych w bieżącym segmencie (jeśli nie jest ustawiona flaga SYN). Kiedy flaga SYN jest ustawiona, wtedy numer sekwencyjny jest numerem identyfikującym strumień danych wysyłanych do strony przeciwnej w ramach bieżącej sesji (ISN - Initial Sequence Number), a pierwszy oktet tego strumienia danych ma numer ISN+1. Przykładowo, jeśli ISN=52, to SeqNo(1 segm.) = 53

**Numerowanie segmentów w nagłówku TCP za pomocą numerów sekwencyjnych:**

ISN (numer identyfikacyjny strumienia) jest generowany losowo z przedziału  $[0, 2^{32} - 1]$ ;

Numer sekwencyjny pierwszego segmentu = seqNo(1 segm.) = ISN + 1

Numer sekwencyjny następnego segmentu = numer sekwencyjny poprzedniego + liczba bajtów w polu danych poprzedniego;

Przykładowo, jeśli SeqNo(1 segm.) = 53, Dług(1 segm.) = 3, to SeqNo(2 segm.) = 56

Numer potwierdzenia (Acknowlegment number): 32 bity

Jeśli jest ustawiona flaga ACK, wówczas pole to zawiera numer sekwencyjny następnego oczekiwanej segmentu, który jest potwierdzeniem odbioru segmentu poprzedniego. Po zestawieniu połączenia flaga ACK jest zawsze ustawiona. Pole to służy do informowania strony przeciwej (potwierdzania), że otrzymany został od niej segment poprzedni w stosunku do tego, którego numer sekwencyjny jest przesyłany w tym polu.

#### Potwierdzanie segmentów za pomocą numerów potwierdzeń w nagłówku TCP:

Numer potwierdzenia segmentu k = Numer sekwencyjny segmentu k+1

Przykładowo, jeśli odbiorca potwierdza odebranie segmentu o numerze sekwencyjnym 56 i pole danych tego segmentu ma długość 7, to w polu „Acknowlegment numer” odbiorca wysyła do nadawcy wartość 63.

TCP header length: 4 bity

Liczba 32-bitowych (4-bajtowych) słów tworzących nagłówek TCP. Wskazuje, gdzie zaczyna się pole danych. Jest konieczny, ponieważ nagłówek TCP może zawierać pole opcji o zmiennej długości. Długość podstawowego nagłówka TCP (bez opcji) wynosi 20 bajtów i w tym przypadku wartość tego pola jest równa 5.

Reserved: 6 bitów

Pole zarezerwowane dla przyszłych zastosowań. Składa się z samych zer.

Bitы kontrolne, inaczej flagi (Control bits, TCP flags): 6 bitów

URG: Jeśli ustawiony, to pole „Wskaźnik pilnych danych” jest znaczące

ACK: Jeśli ustawiony, to pole “Numer potwierdzenia” jest znaczące

PSH: Jeśli ustawiony, oznacza żądanie natychmiastowego przekazania danych do właściwej aplikacji. Jest to istotne w przypadku buforowania danych i przekazywania ich do aplikacji dopiero po zapełnieniu bufora.

RST: Jeśli ustawiony, oznacza jednostronne zamknięcie połączenia

SYN: Jeśli ustawiony, oznacza synchronizację numerów początkowych, jest przesyłany na etapie otwierania połączenia

FIN: Jeśli ustawiony, oznacza żądanie zamknięcia połączenia, jest przesyłany na etapie obustronnego zamykania połączenia

Długość okna przesuwnego (Window size): 16 bitów

Liczba oktetów danych, łącznie z oktetem określonym w polu “Numer potwierdzenia”, które strona przeciwna może wysłać bez otrzymania potwierdzenia ich odebrania. Jeśli np. strona B otrzymała od strony A segment z numerem potwierdzenia równym x i długością okna przesuwnego równą s, to B może wysłać do A segmenty o łącznej długości pola danych nie przekraczającej s, bez oczekiwania na potwierdzenie otrzymania tych segmentów. Pierwszy z segmentów wysłanych przez stronę B będzie miał numer kolejny równy x. Długość okna przesuwnego równa zero oznacza żądanie zaprzestania wysyłania danych.

Przykładowo, jeśli nadawca wysyła segmenty z polem danych o długości 100 B i otrzymał od odbiorcy potwierdzenie odbioru piątego segmentu, w którym w polu „długość okna” jest wartość 1024, to przed otrzymaniem kolejnego potwierdzenia może wysłać maksymalnie 10 segmentów (od 6 do 15).

#### Suma kontrolna (Checksum): 16 bitów

Jest liczona z uwzględnieniem wszystkich bajtów nagłówka TCP (oprócz bajtów pola sumy kontrolnej), pola danych TCP, a także tzw. pseudo-nagłówka TCP, w skład którego wchodzą trzy pola z nagłówka IP, a mianowicie: źródłowy IP, docelowy IP i protokół warstwy 4, oraz całkowita długość segmentu TCP (nagłówek wraz z polem danych).

Zauważmy że

długość segmentu TCP = długość pakietu IP – długość nagłówka IP,  
czyli długość segmentu TCP łatwo obliczyć z odpowiednich pól nagłówka IP.

Pseudo-nagłówek jest stosowany w celu wyeliminowania błędnie trasowanych pakietów.

#### Wskaźnik pilnych danych (Urgent pointer): 16 bitów

Liczba oktetów tzw. „pilnych danych” przenoszonych w bieżącym segmencie. Jeśli numer sekwencyjny bieżącego segmentu jest równy x, a wskaźnik pilnych danych jest równy s, to  $x+s$  jest numerem porządkowym pierwszego oktetu danych znajdujących się bezpośrednio za danymi pilnymi. Pole to jest znaczące tylko w przypadku ustawienia flagi URG. Pilne dane są przekazywane do warstwy aplikacji poza kolejnością wynikającą z ich umiejscowienia w strumieniu.

#### Opcje (Options): zmienna długość, max 40 bajtów

Opcja zajmuje przestrzeń za podstawowym nagłówkiem TCP. Jedna opcja może zajmować dowolną liczbę oktetów (nie większą niż 40), nie musi to być wielokrotność czterech.

Opcje są uwzględniane przy liczeniu sumy kontrolnej nagłówka TCP. Istnieją dwa rodzaje opcji różniące się liczbą składników:

Rodzaj 1 (1 składnik): Pojedynczy oktet „Typ opcji”.

Rodzaj 2 (3 składniki): Oktet „Typ opcji”, oktet „Długość opcji”, oraz oktety zawierające dane opcji.

Długość opcji drugiego rodzaju jest liczona z uwzględnieniem oktetów „Typ” i „Długość”.

Jeśli pole opcji ma długość nie będącą wielokrotnością czterech bajtów, wówczas następuje wypełnienie pola opcji bitami zerowymi do pełnego słowa 32-bitowego.

Obecnie używane są następujące opcje:

Rodzaj	Typ	Długość	Znaczenie
1	0	-	Koniec listy opcji.
1	1	-	Brak operacji.
2	2	4	Maksymalna długość segmentu danych.

Pierwsza z wymienionych opcji umieszczana jest za wszystkimi pozostałymi opcjami; powinna być stosowana tylko wówczas, gdy koniec pola opcji nie pokrywa się z końcem nagłówka TCP. W takim przypadku stosowane jest wypełnienie zerami pozostałych bitów nagłówka TCP.

Opcja „Brak operacji” może być stosowana do rozdzielania opcji, na przykład w sytuacji, gdy pożądane jest, aby kolejna opcja zaczynała się od początku 32-bitowego słowa

nagłówka TCP. Ponieważ nie jest wymagane, aby początek opcji pokrywał się z początkiem 32-bitowego słowa nagłówka TCP, oprogramowanie realizujące protokół TCP musi prawidłowo przetwarzać również opcje nie spełniające tego warunku.

Kod binarny kolejnej opcji przedstawia się następująco:

|00000010|00000100| maksymalna długość segmentu (4 bajty) |

Typ=2 Dług.=4

Służy ona do informowania strony przeciwej o maksymalnej długości, jaką może mieć pole danych w odbieranych segmentach. Strona przeciwna nie powinna wysyłać większych segmentów, gdyż zostaną odrzucone. Może wystąpić tylko w segmentach używanych do inicjowania połączenia, tzn. mających ustawioną flagę SYN. Brak tej opcji oznacza zgodę na przyjmowanie dowolnie długich segmentów.

Dane (Data)

Pole danych TCP.

### Przykładowa sesja TCP

#### Otwieranie połączenia TCP

A: żądanie otwarcia połączenia (wysłanie segmentu inicjującego połączenie):

SYN=1, SeqNo=x

(x jest losowo generowanym identyfikatorem strumienia danych płynącego od A)

B: zgoda na otwarcie połączenia, a zarazem potwierdzenie odebrania powyższego segmentu od A:

SYN=1, ACK=1, SeqNo=y, AckNo=x+1

(y jest losowo generowanym identyfikatorem strumienia danych płynącego od B)

A: wysłanie potwierdzenia odebrania powyższego segmentu od B, czyli potwierdzenie odebrania zgody na otwarcie połączenia:

ACK=1, SeqNo=x+1, AckNo=y+1

**Uwaga:** na etapie otwierania połączenia numer potwierdzenia (AckNo) otrzymujemy dodając jedynkę do numeru sekwencyjnego (SeqNo) segmentu potwierdzanego.

### Utrzymywanie połączeniach (transmisja danych)

A: wysłanie pierwszego segmentu z danymi:

ACK=1, SeqNo =  $x + 1$ ,  $a_1$  bajtów danych, AckNo =  $y + 1$ ,

B: wysłanie do A  $b_1$  bajtów danych oraz potwierdzenia odbioru powyższego segmentu:

ACK=1, SeqNo =  $y + 1$ ,  $b_1$  bajtów danych, AckNo =  $x + 1 + a_1$

A: wysłanie do B  $a_2$  bajtów danych oraz potwierdzenia odbioru powyższego segmentu:

ACK=1, SeqNo =  $x + 1 + a_1$ ,  $a_2$  bajtów danych, AckNo =  $y + 1 + b_1$

.

.

.

A: wysłanie do B ostatniego (o numerze m) segmentu z danymi oraz potwierdzenia odbioru (n-1)-ego segmentu od B:

ACK=1, SeqNo =  $x + 1 + (a_1 + \dots + a_{m-1})$ ,  $a_m$  bajtów danych, AckNo =  $y + 1 + (b_1 + \dots + b_{n-1})$ ,

B: wysłanie do A ostatniego (o numerze n) segmentu z danymi oraz potwierdzenia odbioru m-tego segmentu od A:

ACK=1, SeqNo =  $y + 1 + (b_1 + \dots + b_{n-1})$ ,  $b_n$  bajtów danych, AckNo =  $x + 1 + (a_1 + \dots + a_m)$ ,

A: wysłanie do B potwierdzenia odbioru n-tego segmentu od B:

ACK=1, SeqNo =  $x + 1 + (a_1 + \dots + a_m)$ , AckNo =  $y + 1 + (b_1 + \dots + b_n)$

**Uwaga:** na etapie utrzymywania połączenia numer potwierdzenia (AckNo) otrzymujemy dodając do numeru sekwencyjnego (SeqNo) segmentu potwierdzanego długość jego pola danych.

### Obustronne zamykanie połączenia

Zakładamy, że w etapie transmisji danych A wysłał do B strumień danych o długości  $d_A = a_1 + \dots + a_m$ , a B wysłał do A strumień danych o długości  $d_B = b_1 + \dots + b_n$ , oraz że A inicjuje zamknięcie połączenia.

A: wysłanie segmentu inicjującego zamknięcie połączenia:

FIN=1, ACK=1, SeqNo =  $x + 1 + d_A$ , AckNo =  $y + 1 + d_B$

B: wysłanie zgody na zamknięcie połączenia:

FIN=1, ACK=1, SeqNo =  $y + 1 + d_B$ , AckNo =  $x + 2 + d_A$

A: wysłanie potwierdzenia odebrania zgody B na zamknięcie połączenia:

ACK=1, SeqNo =  $x + 2 + d_A$ , AckNo =  $y + 2 + d_B$

**Uwaga:** na etapie zamykania połączenia odebranie segmentu z ustawioną flagą FIN potwierdzamy wstawiając w pole AckNo numer sekwencyjny (SeqNo) segmentu potwierdzanego powiększony o 1.

**Ćwiczenie 1:** wykorzystując analizator sieciowy *Wireshark* obejrzyj i poddaj analizie pakiety przesyłane między klientem a serwerem usługi *telnet* działającej w oparciu o protokół transportowy TCP. Zastosuj następujący filtr przechwytywania:

**host a and host b and tcp port telnet**

gdzie a i b to adresy albo nazwy DNS klienta i serwera usługi telnet. Jest to skrócony (ale mimo to równoważny) zapis następującego filtru

**(src host a and dst host b and tcp dst port telnet) or (src host b and dst host a and tcp src port telnet)**

Nawiasy są tu konieczne ze względu na jednakowy priorytet operatorów and i or.

Zwróć uwagę, że po otwarciu połączenia TCP następuje negocjacja parametrów właściwych dla protokołu *telnet*, następnie użytkownik jest wzywany do podania nazwy logowania i hasła. Po wciśnięciu znaku na klawiaturze, jest on przesyłany w osobnym segmencie do serwera, serwer odpowiada odsyłając ten znak do klienta (echo), również w osobnym segmencie, następnie znak jest wypisywany na terminalu użytkownika. Na znaki hasła serwer nie odpowiada echem. Zarówno nazwa logowania jak i hasło są możliwe do odczytania.

**Ćwiczenie 2:** wykorzystując analizator sieciowy *ethereal* obejrzyj i poddaj analizie pakiety przesyłane między klientem a serwerem usługi *ftp* działającej w oparciu o protokół transportowy TCP. Zastosuj następujący filtr przechwytywania:

**host a and host b and tcp**

Zwróć uwagę na to, że podczas sesji otwierane są dwa połączenia – jedno sterujące, a drugie do przesyłania danych. Usługa ftp może działać w tzw. trybie aktywnym albo pasywnym. Do przełączania między trybami służy polecenie „*passive*”. Zaobserwuj różnicę w działaniu obu trybów.

Tryb aktywny ftp:

1. Klient wysyła żądanie „PORT” zgłaszaając numer swojego portu dla połączenia do przesyłania danych
2. Po zaakceptowaniu żądania „PORT” przez serwer, klient wysyła żądanie pobrania (RECV) lub wysłania (STOR) pliku
3. Serwer inicjuje połączenie do przesyłania danych. Dla tego połączenia port serwera ma numer 20 (ftp-data).

Tryb pasywny ftp:

4. Klient wysyła żądanie „PASV”
5. Serwer odpowiada zgłaszaając numer „wysokiego portu”. Będzie to port serwera dla połączenia do przesyłania danych
6. Klient inicjuje połączenie do przesyłania danych. Dla tego połączenia port klienta ma „wysoki numer”, tj. większy od 1023.
7. Klient wysyła żądanie pobrania (RECV) lub wysłania (STOR) pliku.

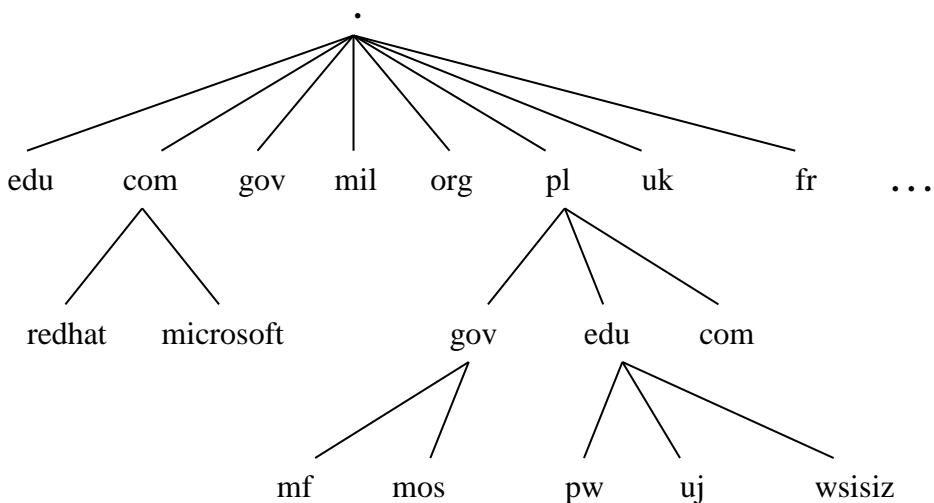
Tryb pasywny stosowany jest zazwyczaj wtedy, kiedy serwer ftp znajduje się poza siecią, do której są przyłączeni klienci, natomiast sieć chroniona jest przez firewall blokujący wszystkie połączenia TCP inicjowane z zewnątrz. W trybie aktywnym połączenie do przesyłania danych inicjuje serwer ftp, a więc zostanie ono zablokowane przez firewall. Możliwym wyjściem z sytuacji jest zainicjowanie tego połączenia przez klienta, czyli przejście w tryb pasywny.

## Usluga DNS

Zadaniem usługi DNS (Domain Name Service) jest ustalenie adresu IP maszyny na podstawie jej nazwy domenowej, albo operacja odwrotna – ustalenie nazwy domenowej maszyny na podstawie jej adresu IP. Usługa ta działa w oparciu o protokół transportowy UDP, a jej serwery używają portu o numerze 53.

### Przestrzeń nazw DNS

Nazwa domenowa jest elementem hierarchicznej struktury zwanej przestrzenią nazw DNS. Najwyższym elementem przestrzeni nazw DNS jest domena główna (root domain) oznaczana samą kropką. Na pierwszym poziomie umiejscowione są domeny organizacji edukacyjnych, komercyjnych, rządowych, wojskowych, itd. mających swoje siedziby w Stanach Zjednoczonych (historycznie, system DNS wywodzi się z USA), oraz 224 domeny krajowe. W nomenklaturze angielskojęzycznej nazywane są one „top level domains”, czyli domenami pierwszego poziomu.



### Budowa drzewa nazw DNS

Pełna nazwa domenowa (ang. Fully Qualified Domain Name, w skrócie FQDN) składa się z nazwy maszyny oraz tzw. prefiku DNS, jednoznacznie określającego położenie domeny, do której maszyna ta należy. W skład prefiku DNS wchodzi nazwa domeny, oraz nazwy wszystkich domen położonych nad nią. Formalnie, prefiks powinien być zakończony kropką, jednak często jest ona opuszczana. Przykładowo, sz123.wsisiz.edu.pl. jest pełną nazwą domenową (FQDN) komputera o nazwie sz123 należącego do domeny wsisiz będącej pod-domeną domeny edu, która z kolei jest pod-domeną domeny pl. Domena pl, znajdująca się na pierwszym poziomie hierarchii DNS, jest pod-domeną domeny głównej.

### Strefy i delegacje

Usługa DNS działa w oparciu o rozproszoną bazę danych, której fragmenty umiejscowione są na wielu serwerach DNS rozlokowanych po całym świecie. Przestrzeń nazw DNS podzielona jest na obszary zwane strefami. Strefą nazywamy zbiór domen posiadających wspólną domenę nadzczną, również należącą do tej strefy. Domena taka

nazywana jest domeną główną strefy. Zgodnie z powyższą definicją, zbiór składający się z jednej domeny również jest strefą. Za każdą strefę odpowiedzialny jest co najmniej jeden serwer DNS będący tzw. serwerem autorytywnym tej strefy. Serwer autorytywny przechowuje pliki zawierające m.in. odwzorowania nazw DNS na adresy IP dla wszystkich maszyn należących do domen strefy, oraz inne istotne informacje. W celu zapewnienia niezawodnego działania usługi DNS, wymaga się, aby pliki te były przechowywane na więcej niż jednym serwerze, zatem oprócz podstawowego serwera strefy powinien istnieć co najmniej jeden serwer zapasowy, który przejmuje funkcje serwera podstawowego w przypadku awarii bądź planowego wyłączenia. Informacja przechowywana na serwerach zapasowych jest uaktualniana za pomocą tzw. transferów strefy. Transfery strefy mogą być pełne (AXFR) lub przyrostowe (IXFR). Transfery przyrostowe uwzględniają tylko zmiany, które zaszły od ostatniej zmiany numeru seryjnego (patrz opis rekordu SOA).

Kluczowe dla działania systemu DNS są serwery strefy „.” która składa się z samej domeny głównej. Obecnie strefa ta jest obsługiwana przez 13 serwerów o następujących nazwach domenowych A.ROOT-SERVERS.NET., B.ROOT-SERVERS.NET. ...., M.ROOT-SERVERS.NET. Należy zwrócić uwagę na fakt, że serwery strefy „.” nie należą do domeny głównej, ale do domeny ROOT-SERVERS.NET.

Domena główna każdej strefy jest zarazem korzeniem poddrzewa DNS. Zbiór domen należących do tego poddrzewa może być zbyt liczny, aby tworzył jedną strefę, a to ze względu na rozmiar plików bazy DNS dla takiej strefy. W związku z powyższym stosuje się tzw. delegowanie stref. Strefa delegowana to taka, której domena główna znajduje się bezpośrednio pod jedną z domen strefy delegującej. Przykładowo, wszystkie domeny pierwszego poziomu są domenami głównymi stref delegowanych ze strefy „.”.

## Rekordy zasobów

Pliki bazy DNS składają się z rekordów zasobów (ang. Resource Record, w skrócie RR). Istnieje ok. 20 typów tych rekordów, z których najważniejsze to:

Nazwa typu	Skrót nazwy	Funkcja
Adres (Address)	A	Odwzorowanie nazwy domenowej maszyny na adres IPv4 (obsługa zapytań prostych)
Adres (Address)	AAAA	Odwzorowanie nazwy domenowej maszyny na adres IPv6 (obsługa zapytań prostych)
Serwer nazw (Name Server)	NS	Odwzorowanie nazwy DNS strefy na nazwę DNS jednego z serwerów strefy (podstawowego lub jednego z zapasowych)
Początek autorytatywnej informacji (Start of Authority)	SOA	Odwzorowanie nazwy DNS strefy na nazwę DNS <b>podstawowego</b> serwera strefy, adres e-mail administratora strefy, oraz parametry określające harmonogram transferów strefy
Wskaźnik (Pointer)	PTR	Odwzorowanie adresu IP maszyny na jej nazwę DNS (obsługa zapytań odwrotnych)
Nazwa kanoniczna(Canonical Name)	CNAME	Odwzorowanie innej nazwy DNS (alias) maszyny, której podstawowa nazwa jest podana w rekordzie typu A.
Wymiennik poczty (Mail Exchange)	MX	Odwzorowanie nazwy DNS strefy na nazwę DNS serwera pocztowego (SMTP) tej strefy
Informacja o sprzęcie (Hardware Information)	HINFO	Określenie platformy sprzętowej i systemu operacyjnego maszyny

Rekord zasobów składa się z następujących pól:

Owner: nazwa maszyny lub pełna nazwa DNSdomeny; informacja wiodąca według której wyszukiwany jest potrzebny rekord

TTL: data i czas określające do kiedy informacja w danym rekordzie jest aktualna; po upływie tego czasu klient albo serwer działający w trybie rekurencyjnym, który pobrał tę informację powinien usunąć ją z pamięci podręcznej

Class: klasa rekordu; najczęściej używaną klasą jest IN (Internet)

Type: typ rekordu

Record-specific data: informacja podawana w rekordzie

Rekord SOA:

Rekord SOA musi być pierwszym rekordem każdego pliku strefy. Pole „Owner” zawiera nazwę DNS strefy, dla której dany serwer jest autorytatywny. Informacja podawana w rekordzie: pełna nazwa domenowa **podstawowego** serwera strefy (Primary DNS Server),

adres e-mail administratora strefy (Zone Admin e-mail), numer seryjny sterujący transferami strefy (Serial Number), okres odświeżania (refresh interval), czas do ponownej próby połączenia z serwerem podstawowym (Retry Interval), czas przez jaki informacja na serwerze wtórnym pozostaje aktualna (Expire Interval), minimalny czas życia (Minimum TTL).

**Uwaga!** W rekordzie SOA znak „@” w adresie pocztowym administratora strefy jest zamieniany na „.”, a każdy znak „.” występujący przed @ – na „\.”. Przykładowo, adres zone.master@firma.com jest zamieniany na zone\.master.firma.com. Jeśli zapis w rekordzie SOA ma być zamieniony na standardowy adres pocztowy, to znaki „\.” są zamieniane na „..”, a pierwszy znak „..” na „@”.

Przykład rekordu SOA:

wsisiz.edu.pl. 86400 IN SOA see-you-later.wsisiz.edu.pl. admin.wsisiz.edu.pl.  
2021100603 28800 7200 1814400 86400

Rekord typu A:

Pole „Owner” zawiera nazwę domenową maszyny. Informacja podawana przez rekord to adres IPv4 maszyny. Jest to informacja podawana w odpowiedzi na zapytanie proste.

Przykład rekordu A:

sz022.wsisiz.edu.pl. 86400 IN A 213.135.47.132

Rekord typu AAAA:

Pole „Owner” zawiera nazwę domenową maszyny. Informacja podawana przez rekord to adres IPv6 maszyny. Jest to informacja podawana w odpowiedzi na zapytanie proste.

Przykład rekordu AAAA:

oceanic.wsisiz.edu.pl. 86400 IN AAAA 2001:1a68:a::33

Rekord typu NS

Pole „Owner” zawiera nazwę DNS strefy. Informacja podawana przez rekord to pełna nazwa domenowa jednego z serwerów danej strefy. Uwaga: jeden z rekordów NS zawiera nazwę serwera podstawowego, ale nie ma tam informacji, że jest to serwer podstawowy.

Przykład rekordu NS:

wsisiz.edu.pl. 86400 IN NS see-you-later.wsisiz.edu.pl.

Rekord typu PTR:

Pole "Owner" zawiera nazwę DNS składającą się z adresu IP zisanego "na odwrót" oraz prefiksu in-addr.arpa . Informacja podawana przez rekord to pełna nazwa domenowa maszyny, której adres IP jest zapisany "na odwrót" w polu "Owner". Jest to informacja podawana w odpowiedzi na zapytanie odwrotne.

Przykład rekordu PTR:

132.47.135.213.in-addr.arpa. 86400 IN PTR sz022.wsisiz.edu.pl.

## Delegowanie stref przy użyciu rekordów zasobów

Jak już wcześniej wspomniano, delegowanie strefy jest zabiegiem mającym na celu utrzymanie rozmiarów plików bazy DNS w rozsądnych granicach. Aby delegowanie strefy powiodło się, w plikach strefy delegującej muszą się znaleźć następujące informacje: nazwa strefy delegowanej, oraz nazwa DNS i adres IP serwera autorytatywnego dla strefy delegowanej. Pierwsze dwa parametry są określane przy pomocy rekordu NS wiążącego nazwę strefy delegowanej z nazwą serwera tej strefy, natomiast trzeci – przy pomocy rekordu A wiążącego nazwę serwera strefy delegowanej z jego adresem IP. Oczywiście, w pliku strefy delegowanej musi znajdować się rekord SOA, wiążący nazwę strefy delegowanej z nazwą DNS podstawowego serwera tej strefy. Istotne jest, aby nazwa serwera strefy delegowanej w tym rekordzie była zgodna z nazwą serwera w odpowiednim rekordzie NS pliku strefy delegującej. Do zilustrowania powyższych warunków niech posłuży następujący przykład:

Załóżmy, że z hipotetycznej strefy mojafirma.com.pl. delegowana jest strefa wawa.mojafirma.com.pl. . Plik strefy mojafirma.com.pl. musi zawierać następujące rekordy:

wawa.mojafirma.com.pl.	IN	NS	dns.wawa.mojafirma.com.pl.
dns.wawa.mojafirma.com.pl.	IN	A	193.0.0.1

(193.0.0.1 jest oczywiście tylko przykładowym publicznym adresem IP), natomiast plik strefy wawa.mojafirma.com.pl. – następujące rekordy:

wawa.mojafirma.com.pl.	IN	SOA	dns.wawa.mojafirma.com.pl.
dns.wawa.mojafirma.com.pl.	IN	A	193.0.0.1

Zakładamy tu, że podstawowy serwer strefy wawa.mojafirma.com.pl. należy do domeny głównej tej strefy, ale **w ogólnym przypadku nie jest wymagane, aby serwer autorytatywny dla strefy należał do domeny głównej tej strefy.**

## Zasady działania usługi DNS

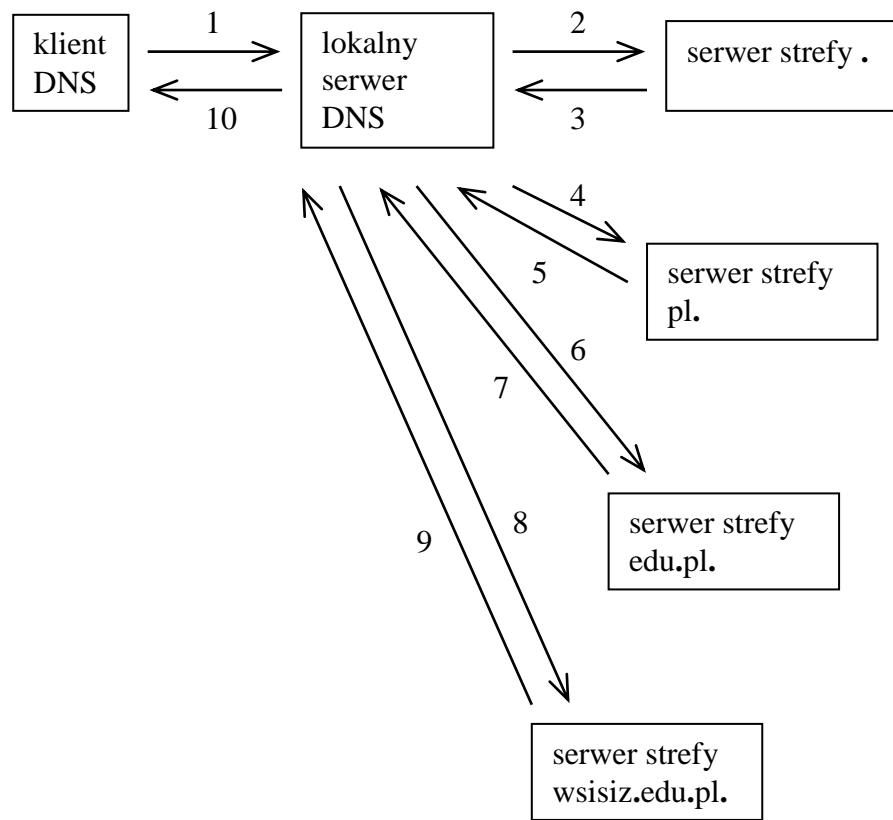
Klientem DNS może być dowolny komputer pracujący w oparciu o protokół TCP/IP. Użycie nazwy domenowej w poleceniu uruchamiającym aplikację sieciową, np. telnet sz123.wsisz.edu.pl, albo wpisanie nazwy domenowej w polu adresu przeglądarki internetowej powoduje, w sposób niewidoczny dla klienta, wysłanie zapytania DNS do lokalnego serwera tej usługi. Serwer lokalny udziela odpowiedzi klientowi, w oparciu o własne zasoby, bądź zasoby innych serwerów. Istnieją dwa rodzaje zapytań DNS - zapytania proste, czyli o adres IP maszyny o podanej nazwie domenowej, oraz zapytania odwrotne, czyli o nazwę domenową maszyny o podanym adresie IP. Te drugie używane są zazwyczaj w celu wyeliminowania tzw. "DNS spoofing", czyli zabiegu polegającego na udzielaniu przez podstawiony serwer DNS fałszywych odpowiedzi na zapytania proste, aby umożliwić innej maszynie udawanie maszyny o określonej nazwie domenowej.

Chcąc ustalić adres IP maszyny o znanej nazwie domenowej, przykładowo - sz123.wsisz.edu.pl., klient DNS wysyła zapytanie do lokalnego serwera DNS (Adres lokalnego serwera jest jednym z parametrów konfiguracyjnych klienta DNS). Serwer lokalny wysyła zapytanie do serwera strefy . . Serwer strefy . udziela tzw. odpowiedzi referencyjnej (ang. Referral Answer), podając adres serwera strefy pl. (delegowanej ze strefy . ). W kolejnym kroku serwer lokalny wysyła zapytanie do serwera strefy pl. , który również udziela odpowiedzi referencyjnej, podając adres serwera strefy edu.pl. (zakładamy, że strefa

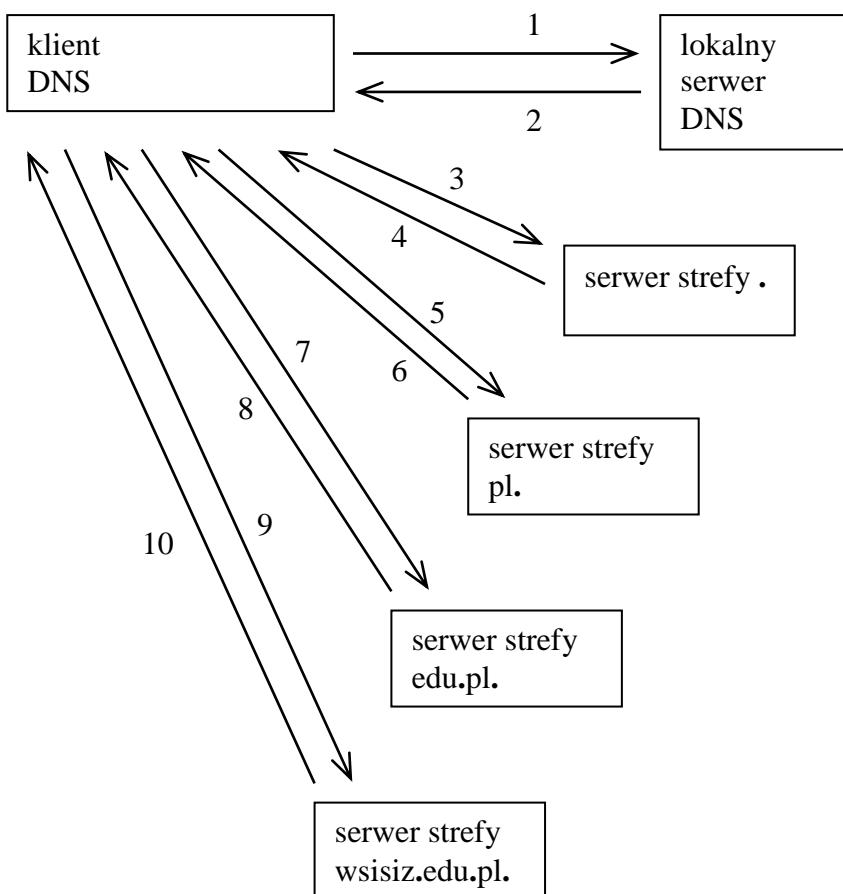
edu.pl. jest delegowana ze strefy pl. ). W kolejnym etapie serwer lokalny wysyła zapytanie do serwera strefy edu.pl. , który też udziela odpowiedzi referencyjnej, podając adres serwera strefy wsisiz.edu.pl. (zakładamy, że strefa wsisiz.edu.pl. jest delegowana ze strefy edu.pl. ). W ostatnim kroku serwer lokalny wysyła zapytanie do serwera strefy wsisiz.edu.pl. , który podaje adres IP maszyny sz123.wsisiz.edu.pl. udzielając tzw. odpowiedzi autorytywnej (ang. Authoritative Answer). Po uzyskaniu adresu IP maszyny sz123.wsisiz.edu.pl., lokalny serwer DNS przesyła go klientowi.

W miarę możliwości, lokalny serwer zapisuje do pamięci podręcznej (ang. DNS cache) uzyskiwane z innych serwerów informacje, aby przy kolejnych zapytaniach klientów nie musiał powtarzać procedury uzyskiwania adresów IP serwerów niektórych stref. Przechowywanie odpowiedzi DNS w pamięci podręcznej serwerów przyspiesza działanie usługi DNS, oraz zmniejsza obciążenie serwerów, a zwłaszcza serwerów strefy . i stref pierwszego poziomu. Ze swojej strony, klienci DNS zapisują w pamięci podręcznej informacje uzyskane z lokalnego serwera DNS, co zmniejsza ruch w sieci lokalnej, jak również obciążenie serwera lokalnego.

Opisany powyżej schemat działania usługi DNS nosi nazwę trybu rekurencyjnego. Polega on na tym, że lokalny serwer DNS otrzymuje od klienta zadanie ustalenia adresu IP maszyny o podanej nazwie domenowej (zapytanie proste), albo nazwy domenowej maszyny o podanym adresie IP (zapytanie odwrotne) i po wykonaniu tego zadania przesyła klientowi żądaną informację. W przypadku, gdy serwer DNS odmawia działania w trybie rekurencyjnym, usługa DNS działa w trybie iteracyjnym. Polega on na tym, że, zadanie odpytywania kolejnych serwerów DNS spoczywa na kliencie, a serwer lokalny przesyła klientowi tylko adres następnego serwera, z którym klient ma się komunikować (odpowiedź referencyjna). Administrator serwera DNS ma możliwość wyłączenia trybu rekurencyjnego i zastąpienia go trybem iteracyjnym. Należy zauważyć, że w podanym przykładzie, lokalny serwer DNS uzyskuje końcową odpowiedź w trybie iteracyjnym, gdyż właśnie na nim spoczywa zadanie odpytywania kolejnych serwerów DNS. Przyjęto tu założenie, że w celu uniknięcia zbyt dużego obciążenia, serwery stref . , pl. i edu.pl. najprawdopodobniej odmówią działania w trybie rekurencyjnym.



Schemat działania DNS w trybie rekurencyjnym



Schemat działania DNS w trybie iteracyjnym

#### Strefa wyszukiwania odwrotnego in-addr.arpa.

Specjalna strefa in-addr.arpa. została utworzona w przestrzeni DNS w celu dostarczenia informacji, w oparciu o którą serwery DNS mogą odpowiadać na zapytania odwrotne, tj. zapytania o nazwę domenową maszyny o podanym adresie IP. Nazwy DNS stref delegowanych z in-addr.arpa są to adresy zarejestrowanych sieci IP zapisane w porządku odwrotnym, do których dodany jest prefiks in-addr.arpa. . Jeśli, na przykład, jakiejś organizacji przyznano sieć klasy A o adresie 20.0.0.0, to odpowiada jej strefa 20.in-addr.arpa. Założymy, że sieć tę podzielono maską 16-bitową na 256 podsieci (każda podsieć ma rozmiar sieci klasy B) i dla każdej z nich ze strefy 20.in-addr.arpa. delegowano osobną strefę. Nazwy DNS delegowanych stref będą wówczas następujące: 0.20.in-addr.arpa. , 1.20.in-addr.arpa. , 2.20.in-addr.arpa. ...., 255.20.in-addr.arpa. Jeśli sieć 20.1.0.0/16 zostanie następnie podzielona maską 24-bitową na 256 podsieci i dla każdej z nich ze strefy 1.20.in-addr.arpa. będzie delegowana osobna strefa, to powstaną w ten sposób strefy 0.1.20.in-addr.arpa. , 1.1.20.in-addr.arpa. , 2.1.20.in-addr.arpa. ...., 255.1.20.in-addr.arpa.

Pliki bazy DNS znajdujące się w serwerach stref delegowanych z in-addr.arpa. muszą składać się m.in. z rekordów typu PTR. Rekordy PTR zawierają informację, na podstawie której wspomniane wyżej serwery udzielają odpowiedzi na zapytania odwrotne.

Założymy że chcemy włączyć strefę x w globalny system DNS. Procedura ta polega m.in. na rejestracji domeny głównej strefy x, w wyniku czego staje się ona strefą delegowaną z innej strefy, już włączonej w globalny DNS. Oprócz tego, dla x należy zarejestrować strefę

wyszukiwania odwrotnego, czyli strefę ois(x).in-addr.arpa. , gdzie ois(x) oznacza odwrócony identyfikator (pod)sieci IP przydzielonej organizacji, która chce zarejestrować domenę x. Wówczas strefa ois(x).in-addr.arpa. stanie się strefą delegowaną, pośrednio lub bezpośrednio, ze strefy in-addr.arpa. Zagwarantuje to dotarcie do serwera strefy ois(x).in-addr.arpa. przez serwery stref położonych wyżej w hierarchii DNS, a tym samym umożliwia uzyskiwanie odpowiedzi na zapytania odwrotne dotyczące maszyn ze strefy x.

### Klient DNS w systemie Red Hat Linux

Funkcje klienta DNS realizuje program o angielskiej nazwie „resolver”. Działanie tego programu zależy od zawartości dwóch plików konfiguracyjnych - /etc/host.conf i /etc/resolv.conf . Pierwszy z tych plików podaje ustawienia opcji programu resolver. Oto niektóre z nich:

order: kolejność stosowania różnych metod odwzorowywania nazw

hosts: ustalenie adresu IP na podstawie wpisu w /etc/hosts

bind: ustalenie adresu IP w oparciu o usługę DNS

nis: ustalenie adresu IP w oparciu o usługę NIS

trim: usunięcie nazwy domeny przy sprawdzaniu pliku /etc/hosts

multi: możliwość związania kilku adresów IP z jedną nazwą DNS w /etc/hosts

Przykładowa zawartość /etc/host.conf :

order hosts, bind

multi off

Plik /etc/resolv.conf zawiera pewne ustawienia domyślne programu resolver:

domain: nazwa DNS domeny lokalnej

search: lista nazw DNS automatycznie dołączanych do przekazywanej programowi resolver nazwy maszyny

nameserver: adres IP lokalnego serwera DNS; może być więcej niż jeden taki wpis

Przykładowa zawartość pliku /etc/resolv.conf :

domain wsisiz.edu.pl

search wsisiz.edu.pl ibspan.waw.pl

nameserver 213.135.44.40

nameserver 213.135.34.24

nameserver 217.17.34.10

nameserver 212.87.0.37

### Polecenia diagnostyczne nslookup, host i dig

Polecenie host służy do uzyskiwania informacji z serwerów stref. Domyślnie, tzn. bez wskazania serwera strefy, informacja pobierana jest z serwera strefy lokalnej. Przykłady działania polecenia host:

host ns.icm.edu.pl : wypisuje adres IP maszyny o podanej nazwie DNS (ns.icm.edu.pl)

host -v ns.icm.edu.pl : wypisuje w formacie plików strefy informacje dotyczące maszyny ns.icm.edu.pl, m.in. zawartość rekordu typu A, na podstawie którego ustalany jest adres IP. Dodatkowo wypisywane są nazwy DNS serwerów autorytatywnych strefy icm.edu.pl

host -t ns redhat.com : wypisuje nazwy DNS serwerów autorytatywnych strefy redhat.com

host -t soa redhat.com : wypisuje nazwę DNS serwera podstawowego strefy redhat.com, oraz inne informacje zawarte w rekordzie SOA tej strefy

host 66.187.233.210 : wypisuje nazwę DNS maszyny o podanym adresie IP (66.187.233.210)

host -v 66.187.233.210 : wypisuje w formacie plików strefy informacje dotyczące maszyny o adresie 66.187.233.210, m.in. zawartość rekordu typu PTR, na podstawie którego ustalana jest nazwa DNS. Dodatkowo wypisywane są nazwy DNS serwerów autorytatywnych strefy wyszukiwania odwrotnego 233.187.66.in-addr.arpa , oraz adres IP serwera podstawowego tej strefy.

host -l wsisiz.edu.pl : wypisuje pełną zawartość plików wskazanej strefy (wsisiz.edu.pl). Polecenie to korzysta z pełnego transferu strefy (AXFR). Należy mieć na uwadze, że serwer strefy może odmówić jej pełnego transferu dla nieuprawnionego klienta.

host -l -v wsisiz.edu.pl : wypisuje pełną zawartość plików strefy, zachowując oryginalny format.

## **PROCESY I WĄTKI W SYSTEMACH ROZPROSZONYCH**

Proces z jednym wątkiem sterowania:  
własny licznik rozkazów, stos, zbiór rejestrów, przestrzeń adresowa;  
komunikacja między procesami - systemowe mechanizmy komunikacji np. komunikaty  
(ew. semafory ale tylko wtedy gdy dostępna jest pamięć współdzielona).

Wiele wątków sterowania w procesie:  
każdy wątek ma własny licznik rozkazów, stos, rejesty;  
ale wszystkie wątki mają wspólną przestrzeń adresową, ten sam zbiór otwartych.  
plików, procesów pochodnych itp.

Cechy analogiczne (wątków i tradycyjnych procesów jednowątkowych) :  
wykonywanie sekwencyjne, działanie współbieżne - podział czasu procesora,  
stany wątków: gotowy, wykonywany, czekający (blokowany), likwidowany;  
tworzenie wątków pochodnych.

**Tabela. Elementy procesu (w którego skład wchodzi wiele wątków) wspólnie dla wszystkich wątków oraz elementy odrębne dla każdego pojedynczego wątku.**

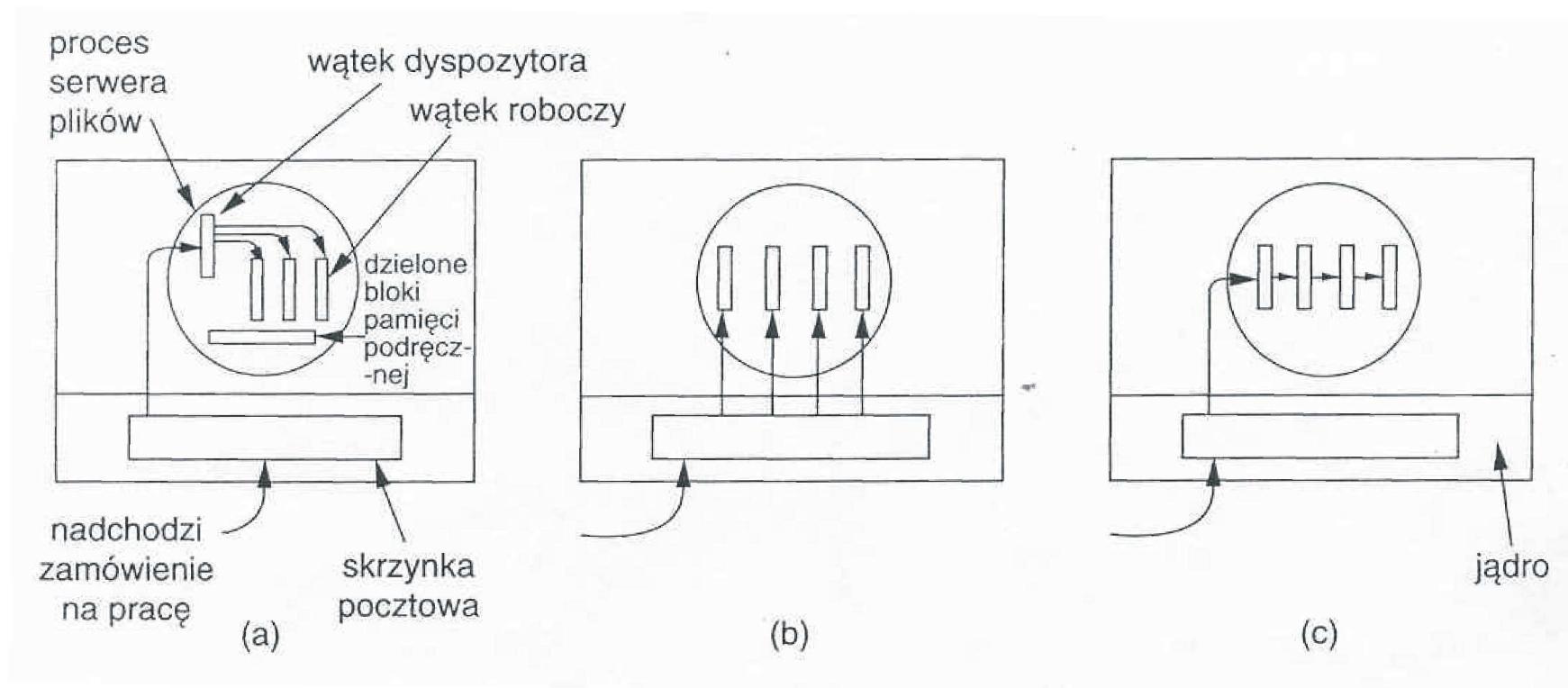
<b>Proces</b>	<b>Wątek</b>
przestrzeń adresowa	licznik rozkazów
zmienne globalne	stos
otwarte pliki	zbiór rejestrów
procesy pochodne	wątki pochodne
czasomierze	stan
sygnały	
semafony	
informacje rozrachunkowe	

## Sposoby organizacji wątków w procesie

Dyspozytor - pracownik (Dispatcher – worker)

Model zepołowy (Team model)

Model potokowy (Pipeline model)



Na rysunku trzy sposoby organizacji wątków w procesie:

(a) dyspozytor-pracownik, (b) model zespołowy, (c) model potokowy

## **Pakiety wątków**

Pakiet wątków: zbiór elementarnych działań - wywołań bibliotecznych dostępnych dla programistów.

### **Przykłady działań elementarnych:**

tworzenie nowego wątku

likwidacja wątku

czekanie na zakończenie innego wątku

informowanie planisty, że w tym momencie powinien być wykonany inny wątek.

### **Organizacja wzajemnego wyłączania**

zastosowanie wirujących blokad,

elementów synchronizacji:

**zamek (mutex)** - zmienna do wyłącznego dostępu,

typowe operacje:

lock(zamek);

unlock(zamek);

### **zmienna warunkowa**

typowe operacje:

wait(zmienna warunkowa);

wakeup( zmienna warunkowa);

## **Przykład: zapewnienie wyłączności dostępu do zasobu:**

**lock(zamek);**

    sprawdź struktury danych;

**while(zasób zajęty)**

**wait(zmienna warunkowa);**

    oznacz zasób jako zajęty;

**unlock(zamek);**

.....

korzystanie z zasobu

.....

**lock(zamek);**

    oznacz zasób jako wolny;

**unlock(zamek);**

**wakeup(zmienna warunkowa);**

## **Problemy planowania wątków na przykładzie systemu Mach**

### **Założenia i cele projektu Mach**

System operacyjny przeznaczony do pracy w systemach rozproszonych, zgodny z systemem BSD UNIX.

Możliwość pracy w systemach heterogenicznych.

Możliwość pracy w systemach komputerowych o różnej architekturze sprzętowej, (w tym z wieloprocesorami).

Możliwość pracy w sieciach komputerowych o różnej prędkości.

Zapewnienie klientom przezroczystości sieci i obiektowej organizacji.

Zintegrowane zarządzanie pamięcią i komunikacją międzyprocesową.

Pojęcie wielowątkowego procesu - zadania (task) i wątku.

### **Charakterystyka planowania w systemie Mach**

Problem planowania: wiele procesów-zadań, wiele wątków, wiele procesorów.

Planuje się tylko przydział procesorów do wątków.

System priorytetów przypisanych wątkom.

Kolejki globalne wykonywanych wątków

Kolejki lokalne przypisane procesorom.

Rozproszona koordynacja przydziału wątków do procesorów.

Zmienny kwant czasu w systemie.

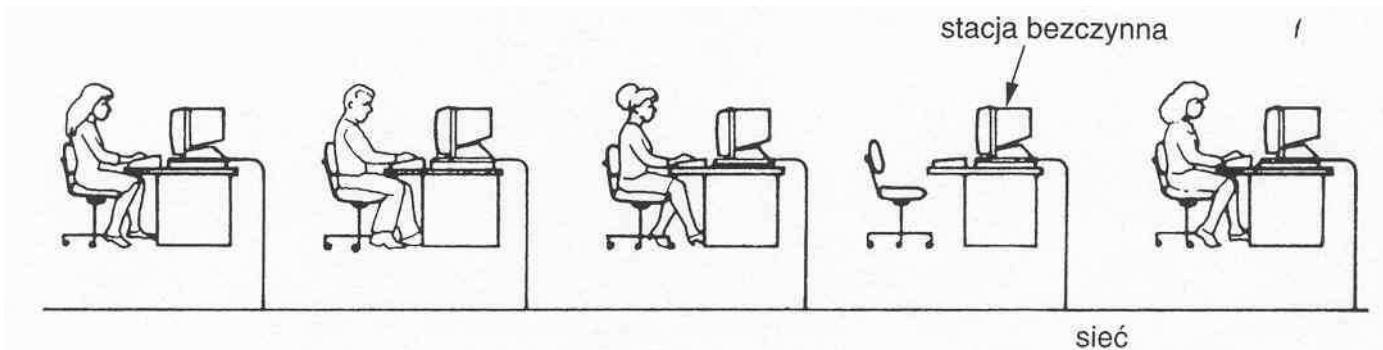
# PROCESY I WĄTKI W SYSTEMACH ROZPROSZONYCH

## MODELE SYSTEMÓW

Różne sposoby organizacji systemów

### Model stacji roboczych

Wiele stacji połączonych siecią LAN



Rys. 4.10. Sieć osobistych stacji roboczych wyposażonych w lokalne systemy plików

Możliwe wykorzystanie stacji z prywatnymi dyskami, ale także stacji bezdyskowych.  
Stacje bezdyskowe - systemy plików realizowane na zdalnych serwerach.

Zalety stacji bezdyskowych:

- niskie koszty,
- łatwość eksploatacji,
- symetria wykorzystania,
- niski hałas.

## **Sposoby wykorzystania prywatnych dysków stacji roboczych**

### **Stronicowanie i przechowywanie plików tymczasowych**

pliki tymczasowe, tworzone w czasie sesji np. w trakcie komplikacji, nie muszą być przesyłane do serwera plików.

### **Stronicowanie i przechowywanie plików tymczasowych, oraz systemowych plików binarnych**

na dyskach lokalnych przechowuje się dodatkowo najczęściej wykorzystywane binaria - kompilatorów, edytorów tekstu, programy obsługi.

### **Stronicowanie i przechowywanie plików tymczasowych, systemowych plików binarnych, oraz podręczna pamięć plików**

w czasie sesji użytkownik ściąga potrzebne pliki z serwera na dysk lokalny, pracuje wykorzystując dysk lokalny, odsyła ostateczne wersje plików do serwera przed zakończeniem sesji.

Zalety: redukcja obciążenia sieci, utrzymuje się zcentralizowaną pamięć długoterminową.

Wady: Problem utrzymania spójności pamięci podręcznych.

## **Kompletny lokalny system plików**

Każda maszyna ma własny system plików z możliwością montowania systemów plików innych maszyn.

### Zalety:

gwarantowany czas odpowiedzi,  
małe obciążenie sieci.

### Wady:

utrudnione dzielenie informacji,  
realizuje idee operacyjne system sieciowy, a nie przezroczystego systemu rozproszonego.

## **Wykorzystanie bezczynnych stacji**

Ogólny problem zdalnego wykonywania procesów w sposób przezroczysty.

### **Pierwsza próba - UNIX BSD**

rsh maszyna polecenie

wady: trzeba określić maszynę, środowisko zdalne na ogólnie niż lokalne.

### **Problemy**

znalezienie bezczynnej maszyny,  
zapewnienie przezroczystości wykonania,  
czynności po powrocie właściciela.

### **Znalezienie bezczynnej stacji**

Definicja bezczynności stacji.

### **Algorytm lokalizacji bezczynnej stacji sterowany za pomocą serwera**

#### **Stacja robocza**

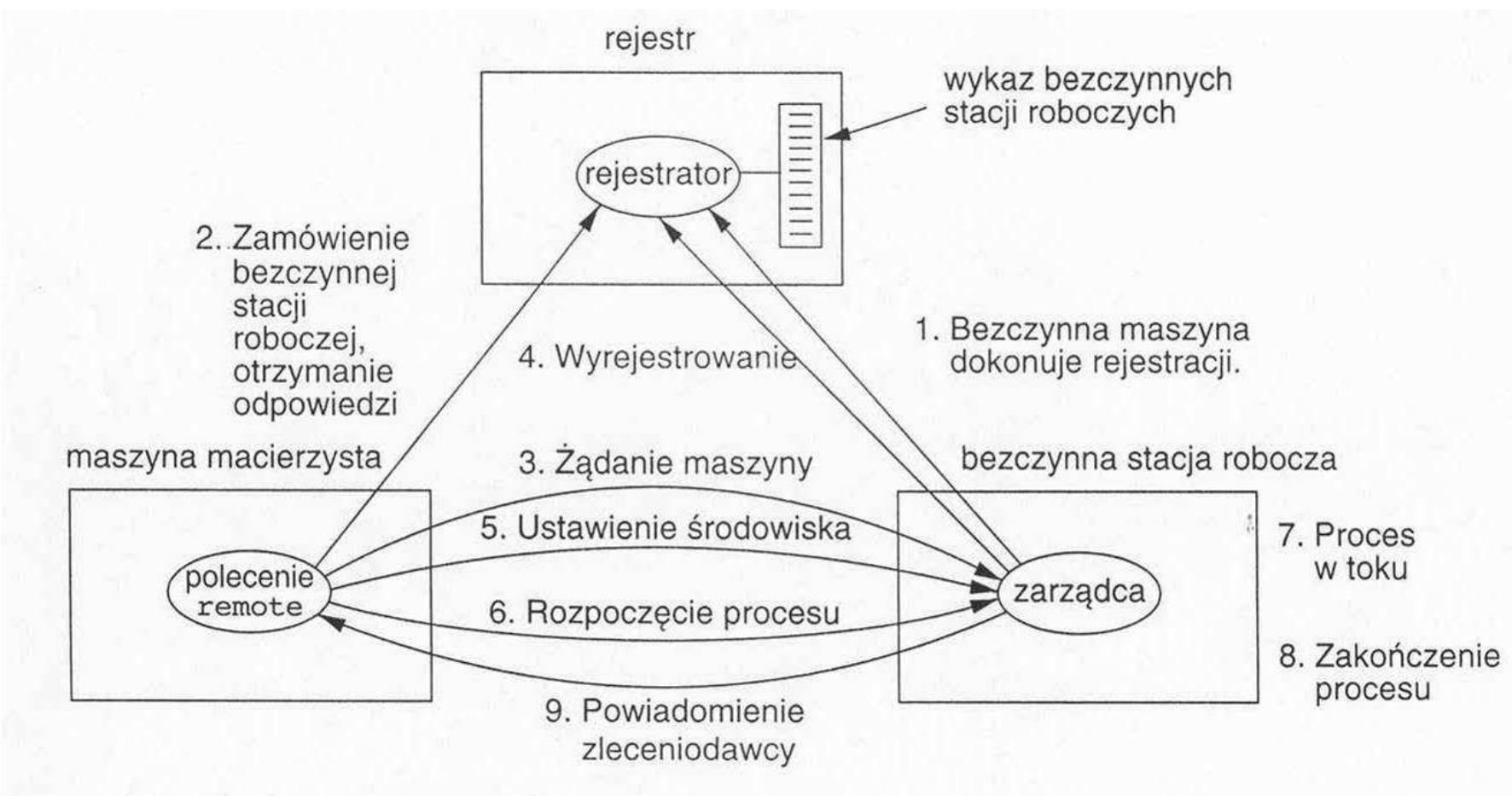
stwierdza swoją bezczynność

ogłasza swoją dostępność - niezbędne informacje (dane stacji) są wpisywane do pliku rejestracyjnego

#### **Użytkownik**

wykonuje: remote polecenie, program remote sam sprawdza rejestr

## Algorytm znajdowania i zatrudniania bezczynnej stacji roboczej wykorzystujący centralne rejestrowanie



## **Algorytm sterowany przez klienta**

Program `remote` rozgłasza zamówienie, podaje jako informacje:  
program, który potrzebuje stację,  
wielkość potrzebnej pamięci,  
zapotrzebowanie na obliczenia, ...

Po nadaniu odpowiedzi program `remote` wybiera stację.

### **Zdalne wykonanie procesu:**

przenieszczenie kodu,  
zapewnienie tego samego środowiska  
potrzebny ten sam obraz plików  
ten sam katalog roboczy  
te same zmienne środowiska.

## **Problemy działania jądra systemu**

Odwołania do systemu plików np. operacja `read`:

system bezdyskowy - zamówienie do serwera plików  
dyski lokalne z kompletnymi systemami plików - do stacji macierzystej.

Odwołania dot. klawiatury i monitora: przesyłane do stacji macierzystej.

Inne odwołania, np. dot. priorytetu, segmentu danych, nazwy maszyny, adresu sieciowego itp.: wykonywane zdalnie.

Problemy synchronizacji czasu.

## **Powrót właściciela do stacji**

Jakie podjąć działania

Żadne - stacja przestaje być osobista

Zlikwidować proces zewnętrzny

nagle - utrata całej pracy, system w chaosie  
ostrzec proces, aby mógł sam się zamknąć  
przenieść proces na inną maszynę  
tzn. kod i dane użytkownika, jądrowe struktury danych

Oczyścić maszynę źródłową.

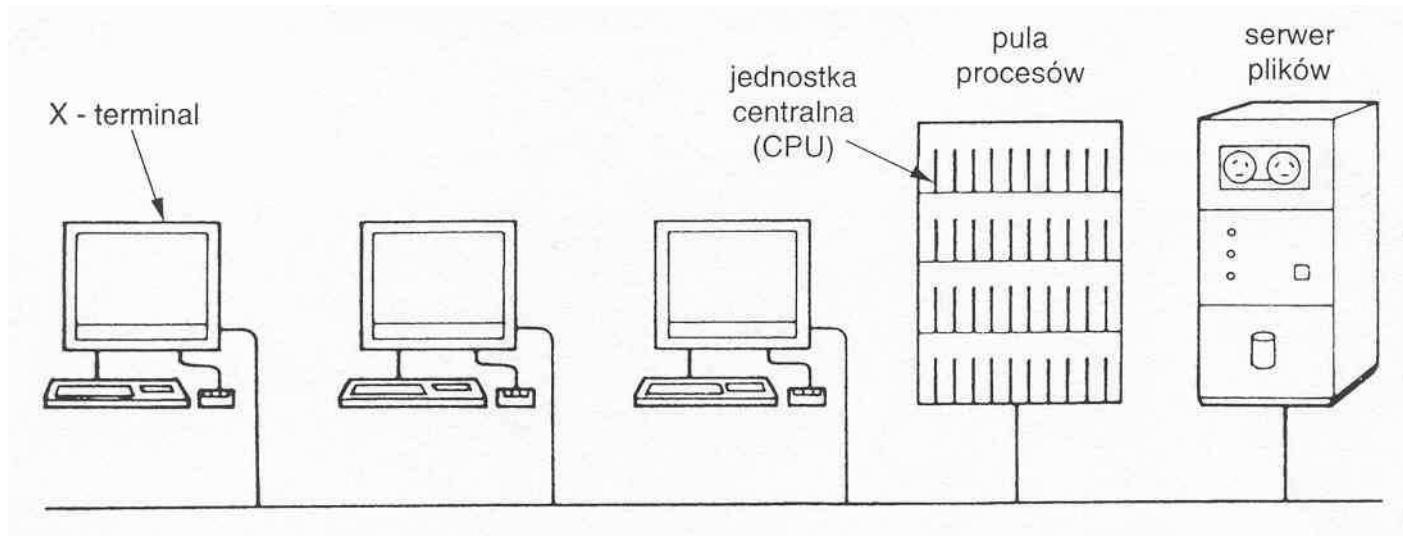
Podstawowa zasada:

kończący proces musi zostawić maszynę w takim samym stanie, w jakim ją zastał.

## Model puli procesów

Wiele jednostek centralnych w jednej szafie.

Użytkownicy mają szybkie terminale graficzne.



Rys. Przykład systemu rozproszonego wg. modelu puli procesorów

Zalety:

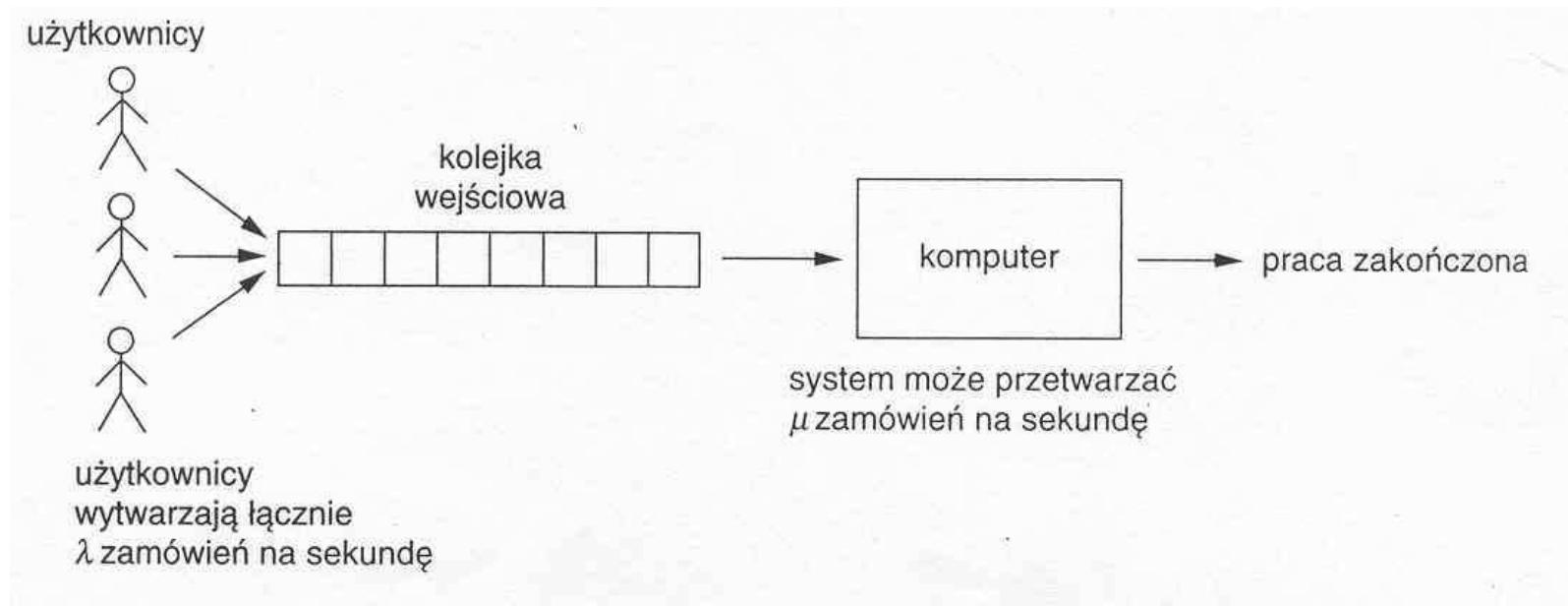
redukcja kosztów - wspólny system zasilania, obudowa, ...

łatwość powiększania mocy obliczeniowej,

możliwość udostępnienia użytkownikowi tylu procesorów, ile potrzebuje.

## System masowej obsługi

użytkownicy generują losowo zamówienia,  
zamówienia ustawiane są w kolejce do obsługi.



Rys. Elementarny system masowej obsługi

# NIEZAWODNOSC - TOLEROWANIE AWARII W SYSTEMACH ROZPROSZONYCH

Systemy komputerowe zawodzą z powodu wad elementów składowych.

**wada (fault)** - niewłaściwe działanie elementu, które może wynikać z różnych powodów: błędu projektanta, błędu w produkcji, błędu w programie, . . .

Klasyfikacja wad

## **Wady przejściowe (transient faults)**

Pojawiają się i znikają. Przy powtórzeniu operacji wada zwykle się już nie pojawia.

## **Wady nieciągłe (intermittent faults)**

Wielokrotnie pojawiają się i znikają w sposób przypadkowy.

## **Wady trwałe (permanent faults)**

Po pojawieniu się nie ustępują, aż uszkodzony element zostanie naprawiony.

**Cel projektowania i budowy systemu tolerującego awarie: uzyskanie pewności, że system będzie działał nawet w przypadku obecności wad.**

Tradycyjne badania tolerowania uszkodzeń - analiza statystyczna wad elementów elektronicznych.

## Awarie w systemie rozproszonym

W systemie rozproszonym jest wiele elementów składowych.

Niewłaściwe działanie procesora może być spowodowane zarówno fizyczną wadą produkcyjną, uszkodzeniem, błędem programu. I.

Tolerowanie awarii przez system rozproszony polega bardziej na takiej jego budowie, aby **mógł przetrwać uszkodzenia elementów składowych** (zwłaszcza procesorów), niż na całkowitym wyeliminowaniu prawdopodobieństwa wystąpienia wad.

### Formy uszkodzeń

#### **Uszkodzenie wyciszające (fail-silent fault)**

Procesor się zatrzymuje i nie odpowiada.

Następuje wadliwe zatrzymanie (fail-stop fault).

#### **Wady bizantyjskie (Byzantine fault)**

Procesor po wystąpieniu takiej wady dalej działa, ale błędnie odpowiada na pytania i niewłaściwie współpracuje z innymi. Stwarza wrażenie poprawnej pracy.

## **Redundancja**

Rozproszone systemy tolerujące awarie buduje się wykorzystując redundancję.

Przykłady:

### **Redundancja informacji**

Przesyłanie dodatkowych bitów informacji, umożliwiających odtworzenie zniekształconych bitów. Kod Hamminga stosowany w transmisji.

### **Redundancja czasu**

Wykonanie operacji, a jeśli wykonana błędnie, powtórzenie jej wykonania.  
Przykład - użycie transakcji niepodzielnych.

### **Redundancja fizyczna**

Specjalna budowa, dodatkowe wyposażenie, zwielokrotnienie elementów składowych, aby system działał mimo awarii niektórych elementów.

### **Sposoby realizacji:**

aktywne zwielokrotnienie,  
zasoby rezerwowe.

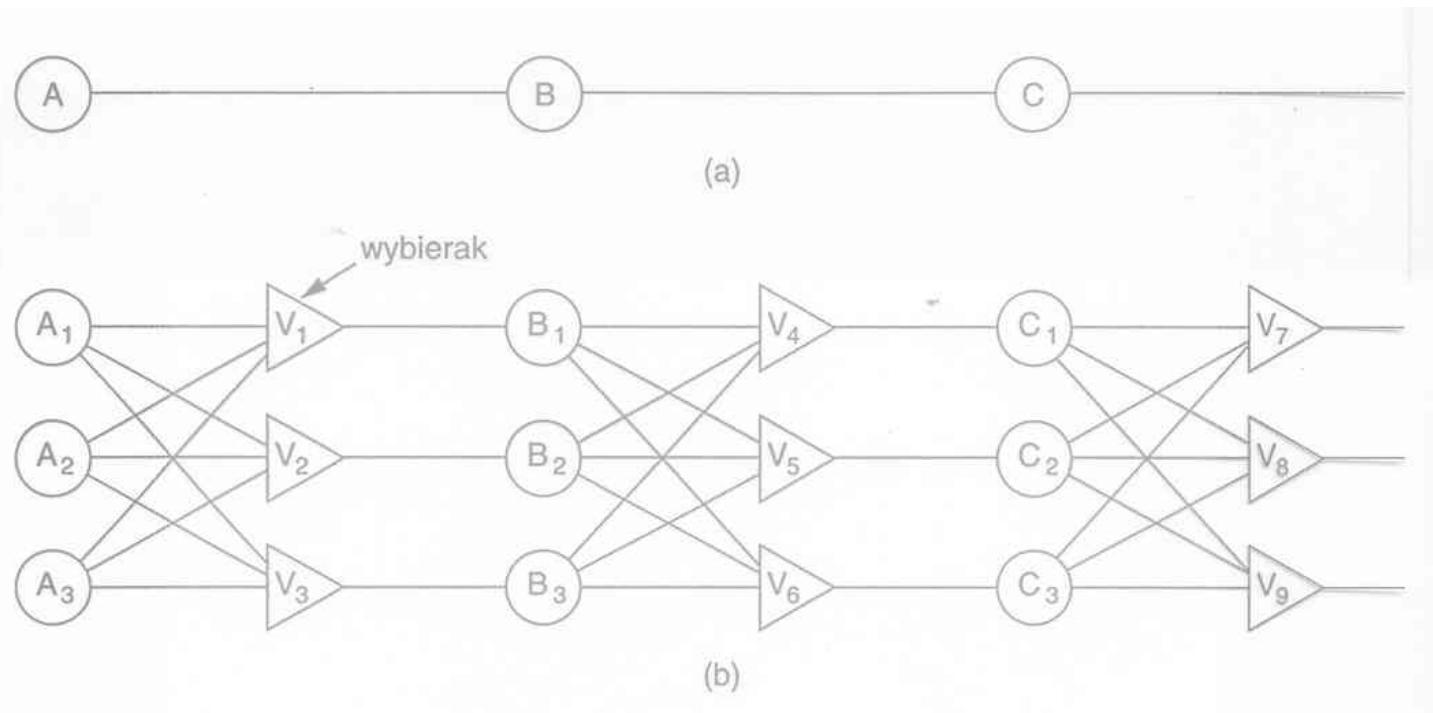
### **Zagadnienia analizy projektowej:**

wymagany stopień zwielokrotnienia,  
działanie systemu, gdy nie ma uszkodzeń - średnie i najgorsze,  
działanie systemu, gdy uszkodzenia występują - średnie i najgorsze.

## Aktywne zwielokrotnienie (active replication)

Zwielokrotnienie elementów działających równolegle.  
Podejście autonomiczne (state machine approach).

Przykład zwielokrotnienia urządzenia  
Technika potrójnej redundancji modularnej (ang. TMR - Triple Modular Redundancy).



## Zagadnienia z wielokrotnienia serwerów w systemach rozproszonych

Serwer - maszyna skończenie stanowa: przyjmuje zamówienia i generuje odpowiedzi.

Zamówienia od klienta wysyłane do wielu serwerów. Jeśli zostaną odebrane i przetworzone w tym samym porządku, to po przetworzeniu wszystkie sprawne serwery będą w tym samym stanie i wygenerują te same odpowiedzi. Wyniki można połączyć, aby wyeliminować uszkodzenia.

Jakie wielokrotnienie?

Odpowiedź zależy od założenia projektowego stopnia odporności systemu na uszkodzenia.

Def.

### **System tolerujący k-uszkodzeń (k-fault tolerance)**

jest to system, który przetrwa uszkodzenia k elementów i będzie działał właściwie.

### **Problem niepodzielnego rozgłaszania -**

wymaganie, aby wszystkie zamówienia dochodziły do serwerów w tej samej kolejności.

Realizacji przetwarzana zamówienia w tej samej kolejności na wszystkich serwerach

- . globalne ponumerowanie - zastosowanie globalnego serwera numerów ,
- . logiczne zegary Lamporta - każdy komunikat ma znacznik czasu, przetwarzanie w serwerach zgodnie ze znacznikami czasu.

## Zasoby rezerwowe

Aktywnie wykorzystywane są zasoby podstawowe (serwer podstawowy). W przypadku awarii, funkcje uszkodzonego zasobu (serwera) przejmuje zasób (serwer) rezerwowy.

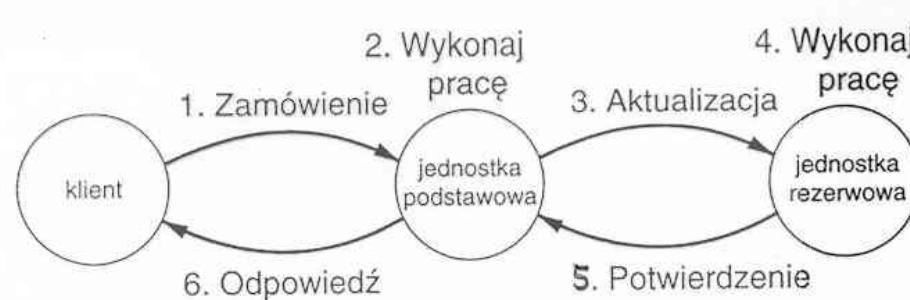
### Zalety

prostsza realizacja - komunikaty są przesyłane tylko do jednego serwera, nie trzeba ich porządkować,  
potrzeba mniej maszyn niż w przypadku aktywnego . z wielokrotnienia

### Wady

mała odporność na wady bizantyjskie  
czasochłonne, złożone przywracanie serwera podstawowego do pracy

### Przykład realizacji Protokół operacji zapisu



Rozwiążanie bardziej zaawansowane

Wspólny dysk dla jednostki podstawowej i rezerwowej z oddzielnymi partycjami.  
Zamówienia i wyniki zapisywane są na dysku.

## Przykład zastosowania redundancji

### **Multi Computer Service Guard** firmy Hewlett Packard

System odporny na (tolerujący) awarie sprzętu i oprogramowania, przeznaczony dla aplikacji wymagających wysokiej niezawodności (mission critical applications).

System rozproszony, składający się z kilku węzłów zorganizowanych jako klaster (cluster). Węzłami mogą być systemy jedno lub wieloprocesorowe.

Węzły w klastrze mają wspólny dostęp do dysków z wykorzystaniem szyny (bus).

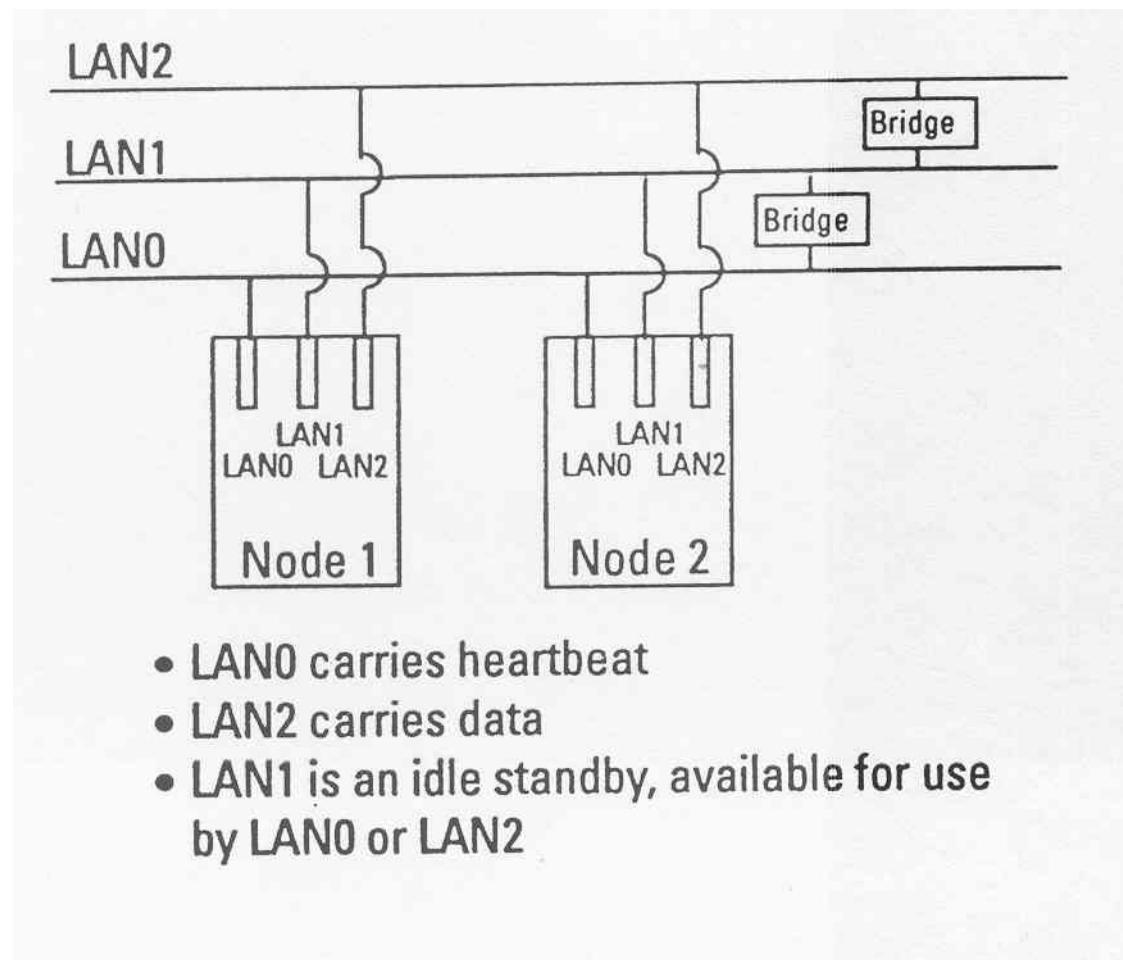
Połączone są również przez sieć LAN wykorzystywaną do:

przesyłania informacji związanych z wykonywaniem aplikacji (dostęp klientów),  
przesyłania sygnałów monitorujących pracę węzłów (heartbeat).

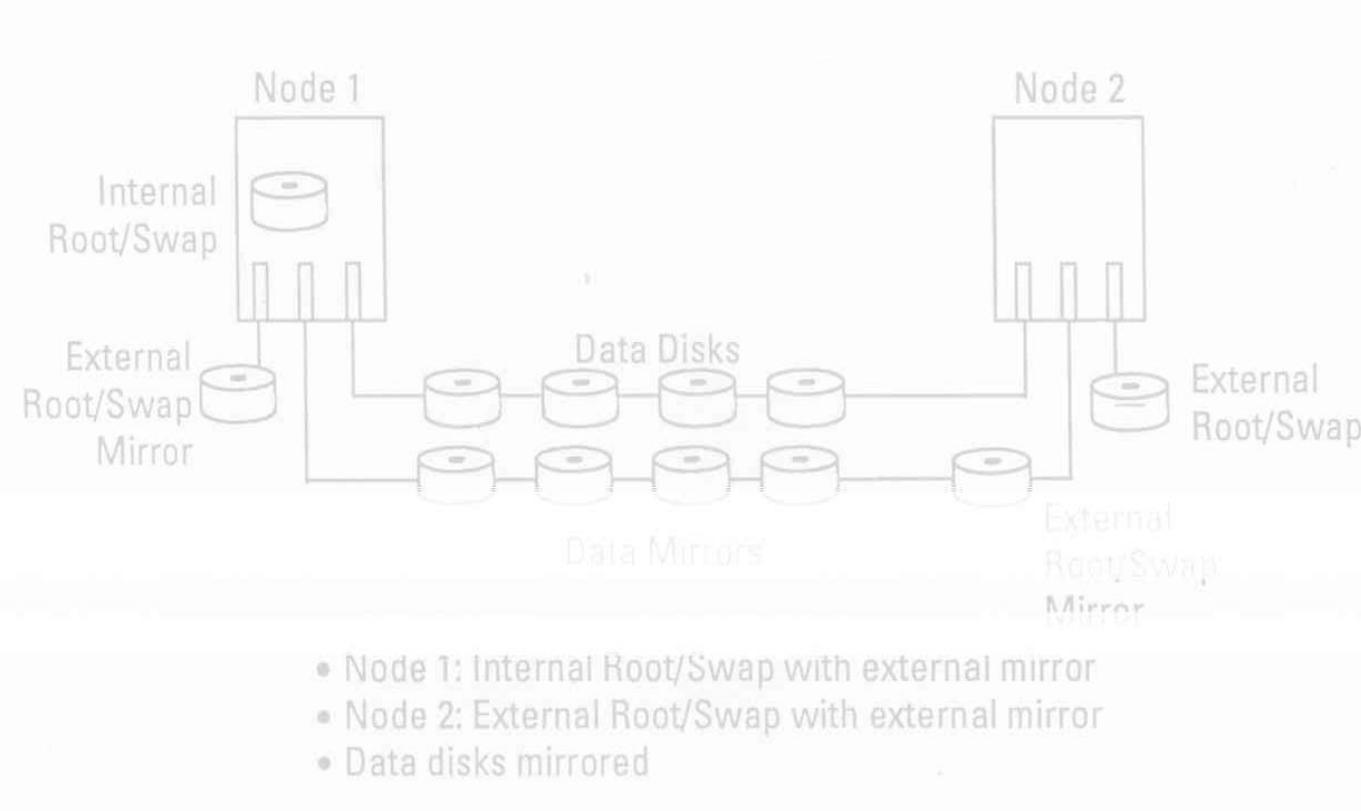
MC Service Guard monitoruje prawidłowość działania (stanu) różnych elementów składowych systemu. W przypadku wykrycia wad podejmuje działanie – automatycznie eliminuje skutki wad, ewentualnie pozwala zminimalizować czas przerwy. Wykrywa i reaguje na wady związane z pracą : jednostek centralnych, pamięci systemowych, sieci LAN, interfejsów sieciowych, procesów aplikacyjnych i systemowych.

Zasoby klastra (wszystkie zasoby niezbędne do wykonania określonych usług aplikacyjnych – pamięć dyskowa, zasoby sieciowe, procesy aplikacyjne i systemowe) organizowane są jako tzw. pakiety aplikacyjne (application packages). Pakiety te stanowią jednostki zarządzane w ramach klastra.

## Przykład Redundantnej konfiguracji sieci LAN



## Przykład redundantnej konfiguracji root/swap



## **W MC/Service Guard stosuje się redundancję w zakresie:**

systemów komputerowych tworzących węzły,  
linii sieci,  
interfejsów sieciowych,  
dysków: root, swap, danych,  
szyn (bus).

Każdy system (węzeł klastra) wykonuje określone aplikacje, ale w przypadku awarii jednego z nich - inny przejmuje wykonanie - kontynuacje zadania.

Korzystając z MC/Service Guard można tworzyć pełne środowisko wykonywania aplikacji odporne na uszkodzenia.

Zaleca się stosowanie, razem z MC/Service Guard następujących rozwiązań:

Mirror Disk/UX

RAID

Power Trust Uninterruptible Power Supplies (UPS),

HP Process Resource Manager

HP Open View Admin Center

HP Open View Operation Center

## TEORIA MASOWEJ OBSŁUGI TEORIA KOLEJEK

mgr inż. Daria Jagieło

### Literatura

- B. von der Veen: *Wstęp do teorii badań operacyjnych*. PWN, Warszawa 1970.
- Gniedenko B. W., Kowalenko I. N.: *Wstęp do teorii obsługi masowej*. PWN, Warszawa 1971.
- Jędrzejczyk Z., Kukula K.: *Badania operacyjne w przykładach i zadaniach*. PWN, Warszawa 1999.
- Kożubski J.: *Wprowadzenie do badań operacyjnych*. Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk 1999.

### Geneza

Początek XX wieku - w związku z silnym rozwojem sieci telefonicznych oraz problemami praktycznymi związanymi z rozbudową już istniejących lub tworzeniem nowych sieci. Pojawiły się problemy dotyczące m.in. odpowiedniego doboru sprzętu, warunków tworzenia centrów telekomunikacyjnych itp. Nurt ten zapoczątkowany był przez duńskiego naukowca AGNERA KRARUPA ERLANGA.

### Teoria masowej obsługi

- Teoria kolejek zajmuje się budową modeli matematycznych, które można wykorzystać w racjonalnym zarządzaniu dowolnymi systemami działania, zwany**m****asow****ej obsług**i.
- Przykładami takich systemów są: sklepy, porty lotnicze, podsystem użytkowania samochodów przedsiębiorstwa transportowego, podsystem obsługiwanego środków transportu.

### Teoria masowej obsługi

- praktyczna dziedzina wiedzy wykorzystywana coraz częściej w zinformatyzowanym świecie,
- związana z rachunkiem prawdopodobieństwa i teorią procesów stochastycznych,
- używająca jako narzędzi badawczych analizy zespołowej, teorii równań różniczkowych i całkowych i innych kierunków matematycznych.

### Cele masowej obsługi

#### Klient

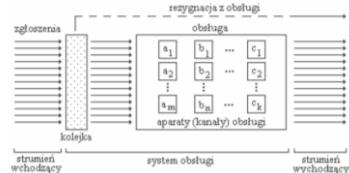
- wybór momentu zgłoszenia,
- średni koszt,
- średni czas obsługi,
- długość kolejki,

#### Zarządzający

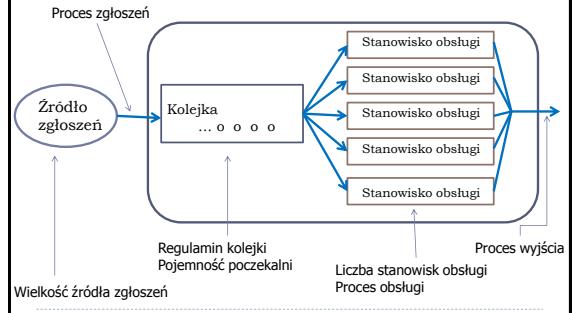
- usprawnienie systemu,
- zwiększenie liczby kanałów obsługi,
- rozbudowa poczekalni,
- zwiększenie atrakcyjności systemu,

## Podstawowe pojęcia

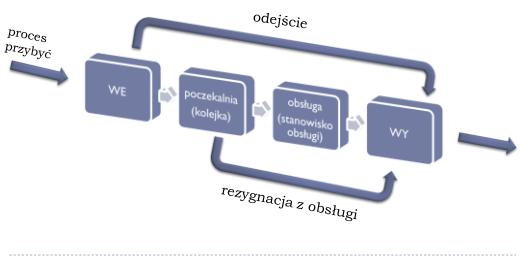
**Zgłoszenie** – obiekt (jednostka) zgłaszający się lub oczekujący na zaspokojenie określonej potrzeby.  
**Obsługa** – zaspokojenie określonej potrzeby.  
**Aparat obsługi** – obiekt umożliwiający zrealizowanie obsługi zgłoszenia.  
**Układ obsługi** – zbiór jednorodnych aparatów obsługi i czynności realizowanych przy pomocy aparatów.  
**Kolejka** – zbiór zgłoszeń oczekujących na obsługę.  
**System masowej obsługi** – zbiór elementów: układ masowej obsługi, zgłoszenia.



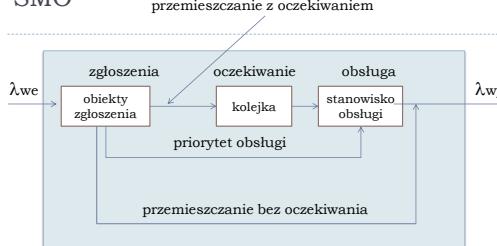
## System masowej obsługi



## Schemat SMO



## SMO

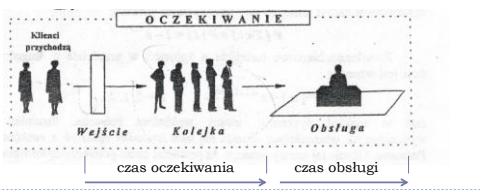


$\lambda_{we}$  – strumień wejściowy zgłoszeń,  
 $\lambda_{wy}$  – strumień wyjściowy obsłużonych obiektów

## Struktura zjawiska oczekiwania

W najprostszej formie zjawisko lub proces oczekiwania składa się z trzech faz zasadniczych:

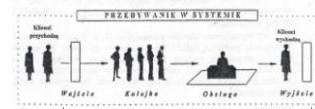
- ▶ wejście jednostek (lub klientów) do systemu obsługi
- ▶ oczekивание (kolejka)
- ▶ obsługa.



## Pojęcie przebywania w systemie

Proces przebywania w systemie obsługi składa się z czterech podstawowych etapów:

- ▶ wejście jednostek (lub klientów) do systemu obsługi
- ▶ oczekивание (kolejka)
- ▶ obsługa,
- ▶ wyjście jednostek (lub klientów).



Czas przebywania w systemie składa się z dwóch części:

- ▶ czas oczekiwania na obsługę,
- ▶ czas obsługi.

## Charakterystyki systemu obsługi

System obsługi można opisać za pomocą trzech charakterystyk:

- ▶ **Strumień zgłoszeń** – będący statystycznym opisem procesu przybywania zgłoszeń.
- ▶ **Proces obsługi** – opisujący obsługę zgłoszeń.
- ▶ **Sposób obsługi kolejek** – określający metodę wybierania następnego zgłoszenia do obsługi.

## Procesy SMO

### proces zgłoszeń

- losowy lub zdeterminowany,

### obsługa

- jeden lub kilka równoległych kanałów obsługi,  
 - obsługa pojedyncza lub grupowa,  
 - obsługa składająca się z jednej lub kilku faz (etapów obsługi),

### poczekalnia

- kolejka o różnych możliwych regulaminach,  
 - kolejki uprzywilejowane,

## Podstawowe parametry systemu obsługi:

- ▶  $\lambda$  - liczba zgłoszeń napływających do systemu obsługi w jednostce czasu
- ▶  $\mu$  - liczba zgłoszeń obsługiwanych w ustalonej jednostce czasu
- ▶ Parametrem charakteryzującym stabilność systemu jest:

$$\rho = \frac{\lambda}{S\mu} \quad \text{- intensywność ruchu}$$

## Proces zgłoszeń (strumień)

$\lambda$  - liczba zgłoszeń napływających do systemu obsługi w jednostce czasu

### Przykład

Do sklepu zgłasza się średnio 5 klientów w ciągu 10 minut

$$\lambda = 5 \text{ [klientów]}/10[\text{minut}]$$

$$\lambda = 0,5 \text{ [klienta/minutę]}$$

## Strumień zgłoszeń

- ▶ charakter losowy → przedział czasu pomiędzy kolejnymi zgłoszeniami jest zmieniącą ciągłą
- ▶ zgłoszenia mogą nastąpić w dowolnej chwili (ale musi być zachowane średnie natężenie zgłoszeń)
- ▶ czas nadania zgłoszenia jest niezależny od poprzednich zgłoszeń
- ▶ prawdopodobieństwo zgłoszenia w przedziale  $\Delta t$  jest proporcjonalne do tego przedziału

## Proces obsługi

$\mu$  - liczba zgłoszeń obsługiwanych w ustalonej jednostce czasu

### Przykład

Kasjer obsługuje w ciągu 12 minut 6 klientów

$$\mu = 6 \text{ [klientów]}/12[\text{minut}]$$

$$\mu = 0,5 \text{ [klienta/minutę]}$$

## Intensywność ruchu

- ▶ Intensywność ruchu (stała Erlanga) – stosunek średniej liczby zgłoszeń jaka napływa do systemu w jednostce czasu do średniej liczby zgłoszeń jaka może być obsłużona w jednostce czasu

$$\rho = \frac{\lambda}{s\mu}$$

s – liczba stanowisk przeznaczonych do obsługi



## Przykład

$$\lambda = 0,5 \text{ [klienta/minutę]}$$

$$\mu = 0,5 \text{ [klienta/minutę]}$$

$\rho = 1$  na granicy stabilności

$$\rho = \frac{\lambda}{s\mu} < 1 \quad \text{System stabilny}$$

$$\rho = \frac{\lambda}{s\mu} = 1 \quad \text{System na granicy stabilności}$$

$$\rho = \frac{\lambda}{s\mu} > 1 \quad \text{System niestabilny}$$



W systemie masowej obsługi mamy do czynienia z napływającymi w miarę upływu czasu zgłoszeniami (np. uszkodzony pojazd, klient, statek), z kolejką obiektów oczekujących na obsługę oraz za stanowiskami obsługi (np. stanowiska diagnostowania pojazdu, sprzedawca, stanowisko wyladunku).

Rozróżnia się systemy masowej obsługi:

- z oczekiwaniem;
- bez oczekiwania.

W SMO z oczekiwaniem zgłoszenie (obiekt zgłoszenia) oczekuje w kolejce na obsługę, zaś w systemie bez oczekiwania, wszystkie stanowiska obsługi są zajęte i obiekt zgłoszenia wychodzi z systemu nie obsłużony.



## Systemy obsługi

- jednokanałowe systemy obsługi
- wielokanałowe systemy obsługi



## Optymalizacja wszelkiego rodzaju jednostek usługowych

### Główne zadania:

- Minimalizacja czasu oczekiwania na obsługę
- Optymalne określenie stanowisk obsługi
- Określanie parametrów obsługi



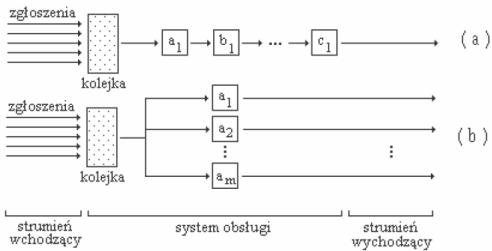
## Klasyfikacje systemów masowej obsługi

- 1. Organizacja obsługi**
2. Zachowanie się zgłoszenia
3. Istnienie kolejki
4. Rozmiary kolejki
5. Organizacja kolejki

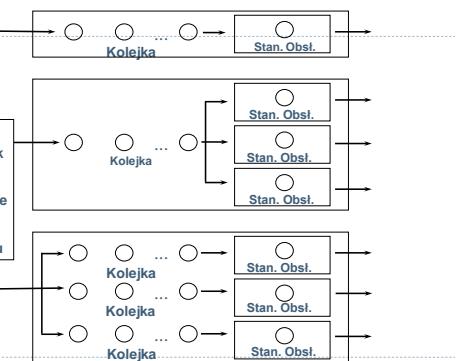


### Klasyfikacja systemów masowej obsługi ze względu na organizację obsługi:

Szeregowe (a), równoległe (b)



### mieszane



### Klasyfikacje systemów masowej obsługi

1. Organizacja obsługi
- 2. Zachowanie się zgłoszenia**
3. Istnienie kolejki
4. Rozmiary kolejki
5. Organizacja kolejki

### Zachowanie się zgłoszenia

- ze stratami (zgłoszenie opuszcza po upływie pewnego czasu system rezygnując z obsługi);
- bez strat (zgłoszenie w systemie przebywa do czasu obsłużenia).

### Klasyfikacje systemów masowej obsługi

1. Organizacja obsługi
2. Zachowanie się zgłoszenia
- 3. Istnienie kolejki**
4. Rozmiary kolejki
5. Organizacja kolejki

### Istnienie kolejki

- zabroniona
- dozwolona

## Klasyfikacje systemów masowej obsługi

1. Organizacja obsługi
2. Zachowanie się zgłoszenia
3. Istnienie kolejki
- 4. Rozmiary kolejki**
5. Organizacja kolejki

## Rozmiary kolejki

### systemy z kolejką

- ograniczoną
- nieograniczoną

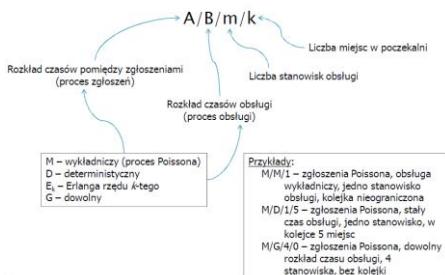
## Klasyfikacje systemów masowej obsługi

1. Organizacja obsługi
2. Zachowanie się zgłoszenia
3. Istnienie kolejki
4. Rozmiary kolejki
- 5. Organizacja kolejki**

## Organizacja kolejki

- ▶ **FIFO** (*first in first out*), czyli kolejność obsługi według przybycia;
- ▶ **SIRO** (*selection in random order*) czyli kolejność obsługi losowa;
- ▶ **LIFO** (*last in first out*), czyli ostatnie zgłoszenie jest najpierw obsłużone;
- ▶ priorytet dla niektórych usług, np. bezwzględny priorytet obsługi oznacza, że zostaje przerwana aktualnie wykonywana obsługa obiektu, a na jego miejsce wchodzi obiekt z priorytetem.

## Notacja Kendalla



## System z jednym stanowiskiem obsługi

Nieograniczona liczba zgłoszeń oczekujących na obsługę SMO-1  
 $M/M/1/\infty$

System tworzą:

- ▶ jeden aparat obsługi o wykładniczym czasie trwania obsługi i intensywności obsługi  $\mu$ ,
- ▶ prosty strumień zgłoszeń o intensywności  $\lambda$ ,
- ▶ intensywność obsługi
- ▶ warunek stabilności systemu:  $0 \leq \rho < 1$        $\rho = \frac{\lambda}{s\mu}$

## Charakterystyki

- procent czasu zajętości wszystkich stanowisk obsługi
- prawdopodobieństwo, że system nie jest pusty
- średnia liczba klientów czekających
- średnia liczba klientów czekających i obsługiwanych
- średni czas czekania
- średni czas czekania i obsługi
- prawdopodobieństwo, że przybywający klient czeka
- prawdopodobieństwo, że n klientów jest w systemie

## Przykład:

Na poczcie obok innych stanowisk jedno jest przeznaczone do obsługi wpłat i wypłat gotówkowych osób fizycznych.

Ruch w godzinach 14-18 jest tak duży, że rozwija się możliwość uruchomienia dodatkowego stanowiska obsługi.

Sprawdzić, czy jest to słuszna decyzja.  
Poniżej podano obserwacje poczynione w czasie jednej z godzin szczytowych.

	Czas przyjęcia liczony od przybycia poprzedniego klienta (w min)	Czas obsługi klienta (w min)		Czas przyjęcia liczony od przybycia poprzedniego klienta (w min)	Czas obsługi klienta (w min)
	<b>0</b>	1,5		<b>1</b>	5,5
	<b>0,5</b>	2,5		<b>1,5</b>	4,5
	<b>1</b>	1		<b>2</b>	4
	<b>1,5</b>	2		<b>1,5</b>	3
	<b>1</b>	3		<b>1</b>	2
	<b>2,5</b>	5		<b>2,5</b>	1,5
	<b>0,5</b>	0,5		<b>3</b>	3
	<b>6</b>	1,5		<b>3,5</b>	4
	<b>2</b>	2,5		<b>4</b>	4
	<b>1,5</b>	6		<b>3,5</b>	3
				<b>40</b>	<b>60</b>

## Rozwiązanie

Proces zgłoszeń  $\lambda = \frac{20}{40} = \frac{1}{2}$

Proces obsługi  $\mu = \frac{20}{60} = \frac{1}{3}$

Intensywność ruchu  $\rho = \frac{\lambda}{\mu} = \frac{\frac{1}{2}}{\frac{1}{3}} = \frac{3}{2} = 1,5$

Zachodzi nierówność  $\rho > 1$ , czyli liczba zgłoszeń przewyższa możliwości obsługi takiej liczby zgłoszeń.

**SYSTEM NIESTABILNY**

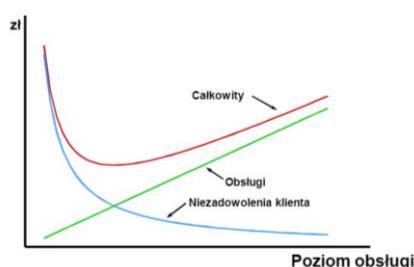
Oznacza to, że prawdopodobieństwo długiej kolejki się zwiększa.

Osiągnięcie stanu równowagi jest tylko możliwe dzięki podjęciu radikalnych działań:

skróceniu czasu obsługi klienta

zainstalowaniu dodatkowego stanowiska obsługi.

## Koszty w systemach masowej obsługi



Model matematyczny funkcjonowania SMO opiera się na teorii procesów stochastycznych

W modelu tym występują zmienne losowe:

- czas upływający między wejściem do systemu dwóch kolejnych zgłoszeń;
- czas obsługi jednego zgłoszenia przez stanowisko obsługi;
- liczba stanowisk;
- liczebność miejsc w kolejce zgłoszeń oczekujących na obsługę.

## System M/M/s

- s stanowisk obsługi
- Strumień wejściowy Poissona z parametrem  $\lambda$
- Obsługa wykładnicza z parametrem  $\mu$
- Dyscyplina obsługi FIFO
- Pojedyncza kolejka
- Kolejka nieograniczona
- $\lambda < s\mu$

## Założenia w teoretycznym modelu

Rozpatrywane są tylko sytuacje, w których klienci obsługiwani są według kolejności przybywania do punktu świadczącego usługę, zatem wszyscy klienci są traktowani na równi.

## Przykład

W prywatnej przychodni stomatologicznej czynne są dwa gabinety lekarskie. Pacjenci zgłaszały się z częstotliwością ok. 3,8 pacjenta na godz., a intensywność ich obsługi wynosi 2 pacjentów na godz.

$$\lambda = 3,8$$

$$\mu = 2$$

$$s = 2$$

$$\rho = \frac{\lambda}{\mu s} = \frac{3,8}{2 \cdot 2} = 0,95$$

## Jakie jest prawdopodobieństwo, że nie będzie kolejki?

Kiedy wystąpi kolejka?

- Gdy wszystkie stanowiska obsługi są zajęte.

Jakie są sytuacje, dla których ta kolejka nie wystąpi?

- Żadne stanowisko obsługi nie jest zajęte.
- Zajętych jest  $s-1$  stanowisk obsługi.

## Ille wynosi prawdopodobieństwo, że nie będzie kolejki?

$$p(i \geq s) = \sum_{j=s}^{\infty} p_j = \frac{p_s}{1-\rho}$$

Pr-two, że ustawi się kolejka

$$p_i = \begin{cases} \frac{(sp)^i}{i!} p_0 & \text{dla } i = 1, 2, \dots, s-1 \\ \frac{\rho^i s^i}{s!} p_0 & \text{dla } i = s, s+1, \dots, \infty \end{cases} \quad \begin{matrix} i < s \\ i \geq s \end{matrix} \quad \begin{matrix} \text{Pr-two, że w systemie jest} \\ i \text{ obiektów} \end{matrix}$$

$$p_0 = \frac{1}{\sum_{i=0}^{s-1} \frac{(sp)^i}{i!} + \frac{(sp)^s}{(1-\rho)s!}} \quad \begin{matrix} \text{Pr-two, że system jest pusty} \end{matrix}$$

## Ille wynosi prawdopodobieństwo, że nie będzie kolejki?

$$p(i \geq s) = \sum_{j=s}^{\infty} p_j = \frac{p_s}{1-\rho} = \frac{0,0462}{1-0,95} = \frac{0,0462}{0,05} = 0,9240$$

$$p_i = \begin{cases} \frac{(sp)^i}{i!} p_0 & \text{dla } i = 1, 2, \dots, s-1 \\ \frac{\rho^i s^i}{s!} p_0 & \text{dla } i = s, s+1, \dots, \infty \end{cases} \quad \begin{matrix} i < s \\ i \geq s \end{matrix}$$

$$p_s = \frac{\rho^s s^s}{s!} p_0 = \frac{0,95^2 \cdot 2^2}{2!} \cdot 0,0256 = 0,0462$$

$$p_0 = \frac{1}{\sum_{i=0}^{s-1} \frac{(sp)^i}{i!} + \frac{(sp)^s}{(1-\rho)s!}} = \frac{1}{\frac{(2 \cdot 0,95)^0}{0!} + \frac{(2 \cdot 0,95)^1}{1!} + \frac{(2 \cdot 0,95)^2}{2!}} = 0,0256$$

Ille wynosi prawdopodobieństwo, że nie będzie kolejki?

$$1 - p(i \geq s) = 1 - 0,9240 = 0,076$$

Prawdopodobieństwo, że nie będzie kolejki w poradni stomatologicznej wynosi ok. 8%.

Ille wynosi prawdopodobieństwo, że pacjent będzie musiał oczekiwac w kolejce dłużej niż 0,5 godz.?

$$P(\tau_k > T) = \frac{p_s}{1-\rho} e^{-T(s\mu-\lambda)}$$

$$P(\tau_k > 0,5) = \frac{p_s}{1-\rho} e^{-T(s\mu-\lambda)} = \frac{0,0462}{1-0,95} e^{-0,5(2 \cdot 2 - 3,0)} = 0,8360$$

Prawdopodobieństwo, że pacjent będzie musiał oczekiwac w kolejce dłużej niż 0,5 godz. wynosi ok. 84%.

Jak wygląda sytuacja z punktu widzenia właściciela poradni?

- Sytuacja z punktu widzenia właściciela poradni dla pacjentów nie jest komfortowa.
- Prawdopodobieństwo bezkolejkowego przyjęcia jest małe, bo wynoszące 0,076.
- Prawdopodobieństwo, że pacjent będzie czekał dłużej niż pół godziny, jest bardzo duże i wynosi 0,84.

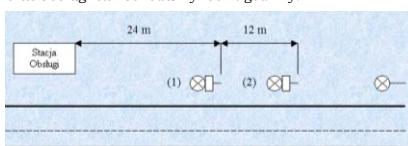
System M/M/s/N

- s stanowisk obsługi.
- Strumień wejściowy Poisson z parametrem  $\lambda$
- Obsługa wykładnicza z parametrem  $\mu$
- Dyscyplina obsługi FIFO.
- Pojedyncza kolejka.
- Kolejka ograniczona.
- $\lambda < s\mu$ .

### Przykład

Stacja obsługi samochodów posiada 1 stanowisko naprawcze (schemat). Z uwagi na duże zainteresowanie wśród klientów prowadzonymi usługami, właściciel postanowił utworzyć dodatkowe stanowisko naprawcze. Wymaga to zmiany usytuowania znaku zakazu zatrzymywania.

Właściciel sądzi, że wprowadzona zmiana przyniesie mu dwukrotny wzrost klientów. Dotychczas zgłaszało się 6 klientów w ciągu dnia pracy (8 godzin). Średni czas obsługi samochodu wynosi 2 godziny.



Założenia:  
Średnia długość samochodu 4 m

M/M/1/N → M/M /s/N

$$s = 1$$

$$N = 6$$

$$\lambda = \frac{6}{8} = 0,75$$

$$\mu = \frac{1}{2} = 0,5$$

$$\rho = \frac{\lambda}{s\mu} = \frac{0,75}{0,5} = 1,5$$

$$\bar{v} = 5,82$$

$$p_0 = 0,0203$$

$$s = 2$$

$$N = 9$$

$$\lambda = \frac{12}{8} = 1,5$$

$$\mu = \frac{1}{2} = 0,5$$

$$\rho = \frac{\lambda}{s\mu} = \frac{1,5}{2 \cdot 0,5} = 1,5$$

$$\bar{v} = 7,3195$$

$$p_0 = 0,002$$

## System M/G/1

### Model :

Strumień wejściowy Poissona z param. 1.  
Czas obsługi o dowolnym rozkładzie, średniej  $m$  i odchyleniu standardowym  $s$ .

Jedno stanowisko obsługi.

➤ Czas obsługi nie musi mieć rozkładu wykładniczego brak założeń o rozkładzie) np.:

- Naprawa telewizora
- Badanie wzroku
- Fryzjer

## System M/D/1

➤ Czas obsługi może być ustalony np.:

- Taśma produkcyjna.
- Myjnia automatyczna.

➤ Czas obsługi deterministyczny

➤ Aby uzyskać system M/D/1 w systemie M/G/1 trzeba przyjąć odchylenie standardowe równe 0 ( $s = 0$ ).

Warunek stabilności  $\lambda < \mu$

## Przykład

Stacja benzynowa ma 3 dystrybutory - ON, Pb95, Pb98. Dojazdy przed dystrybutoremami pozwalały na tworzenie trzech niezależnych kolejek. Samochody tankujące ON odjeżdżają średnio co 10 minut, a czas tankowania wynosi średnio 8 minut, zaś samochody tankujące Pb95 i Pb98 przejeżdżają co 8 minut, a czas tankowania wynosi 1 minut. Stacja zatrudnia jednego pracownika. Ocenić (pod względem czasu przebywania klientów na stacji) dwa warianty organizacji pracy stacji benzynowej:

Pracownik sam obsługuje dystrybutory i inkasuje pieniądze. Samochody są tankowane zgodnie z kolejnością przybycia.

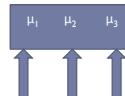
Stacja zostaje zamieniona na samoobsługowa, a pracownik zajmuje się jedynie inkasowaniem pieniędzy.

W obu przypadkach średni czas placenia za paliwo wynosi 1 minute.

**Wskazówka:** Jeśli w systemie M/M/s (M/M/1) strumień wejściowy jest procesem Poissona o intensywności  $\lambda$ , to strumień wyjściowy jest również procesem Poissona o intensywności  $\lambda$ .

### A) Pracownik

M/M/1/ $\infty$



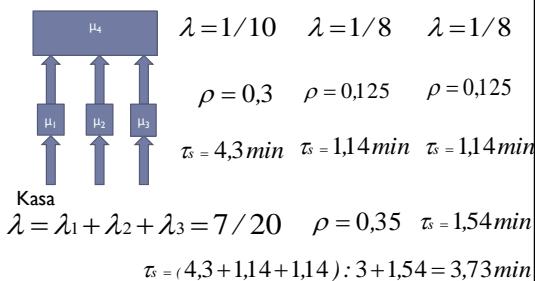
$$\lambda = \lambda_1 + \lambda_2 + \lambda_3$$

$$\lambda = 1/10 + 1/8 + 1/8 = 7/20$$

$$\rho = \frac{18}{20} = 0,9$$

$$\tau_s = 25,7 \text{ min}$$

### B) Samoobsługa      3 x M/M/1/ $\infty$



## Aplikacje

- ▶ WINQSB
- ▶ FLEXSIM
- ▶ CAST
- ▶ VISSIM
- ▶ Java Modelling Tools

**WINQSB**

Problem Title: Colas	Time Unit: hour	
Entry Format:		
<input checked="" type="radio"/> Simple M/M System <input type="radio"/> General Queueing System		
OK	Cancel	Help
Random Seed: <input checked="" type="radio"/> Use default random seed <input type="radio"/> Enter a seed number <input type="radio"/> Use system clock  Queue Discipline: <input checked="" type="radio"/> FIFO <input type="radio"/> LIFO <input type="radio"/> Random  Random seed number: 27437 Simulation time: 1000 hours Start collection time: 0 hours Queue capacity: M Max. number of data collections: M		
OK	Cancel	Help

**Data Description**

Number of servers	ENTRY
Service rate [per server per hour]	M
Customer arrival rate [per hour]	M
Queue capacity (maximum waiting space)	M
Customer population	M
Busy server cost per hour	M
Idle server cost per hour	M
Customer waiting cost per hour	M
Customer being served cost per hour	M
Cost of customer being balked	M
Unit queue capacity cost	M

**Data Description**

Number of servers	ENTRY
Number of servers	2
Service rate [per server per hour]	15
Customer arrival rate [per hour]	20
Queue capacity (maximum waiting space)	M
Customer population	150
Busy server cost per hour	100
Idle server cost per hour	100
Customer waiting cost per hour	100
Customer being served cost per hour	100
Cost of customer being balked	100
Unit queue capacity cost	100

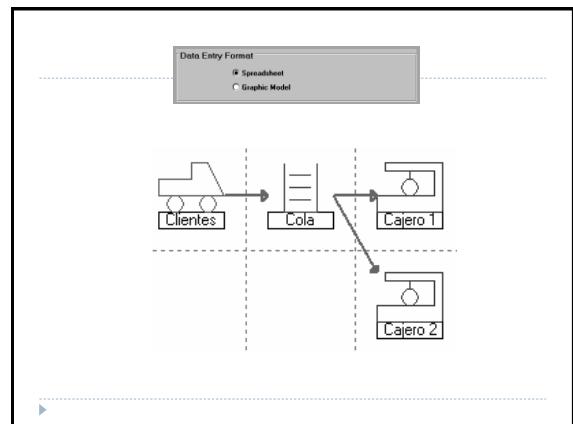
Simulating...

Press the "Q" key to quit the simulation if you wish.  
The program will retain the result up to the moment.

12-01-2005	Performance Measure	Result
1	System: M/M/2	From Simulation
2	Customer arrival rate (lambda) per hour =	20,0000
3	Service rate per server (mu) per hour =	15,0000
4	Overall system effective arrival rate per hour =	0,9807
5	Overall system effective service rate per hour =	19,8677
6	Overall system utilization =	66,3079 %
7	Average number of customers in the system (L) =	2,2767
8	Average number of customers in the queue (Lq) =	0,9528
9	Average number of customers in the queue for a busy system (Lb) =	1,8000
10	Average time customer spends in the system (W) =	0,1146 hours
11	Average time customer spends in the queue (Wq) =	0,0478 hours
12	Average time customer spends in the queue for a busy system (Wb) =	0,0506 hours
13	The probability all servers are idle (P0) =	20,1912 %
14	The probability an arriving customer waits (Pw) or system is busy (Pb) =	\$2,8070 %
15	Average number of customers being balked per hour =	0,0000
16	Total cost of busy server per hour =	\$198,9230
17	Total cost of idle server per hour =	\$101,0720
18	Total cost of customer waiting per hour =	\$190,1147
19	Total cost of customer being served per hour =	\$265,2332
20	Total cost of customer being balked per hour =	\$0
21	Total queue space cost per hour =	\$0
22	Total system cost per hour =	\$755,3479
23	Simulation time in hour =	1000,0000
24	Starting data collection time in hour =	0
25	Number of observations collected =	198688
26	Maximum number of customers in the queue =	7
27	Total simulation CPU time in second =	3,3034

**12-02-2005**

	Result	Cientes
1	Total Number of Arrival	1123
2	Total Number of Balking	260
3	Average Number in the System (L)	2,2144
4	Maximum Number in the System	17
5	Current Number in the System	1
6	Number Finished	871
7	Average Process Time	0,0663
8	Std. Dev. of Process Time	0,0071
9	Average Waiting Time (Wq)	0,1879
10	Std. Dev. of Waiting Time	0,1560
11	Average Transfer Time	0
12	Std. Dev. of Transfer Time	0
13	Average Flow Time (W)	0,2542
14	Std. Dev. of Flow Time	0,1562
15	Maximum Flow Time	0,6007
	Data Collection: 0 to 100 Minutos	
	CPU Seconds =	1,6250



## CAST

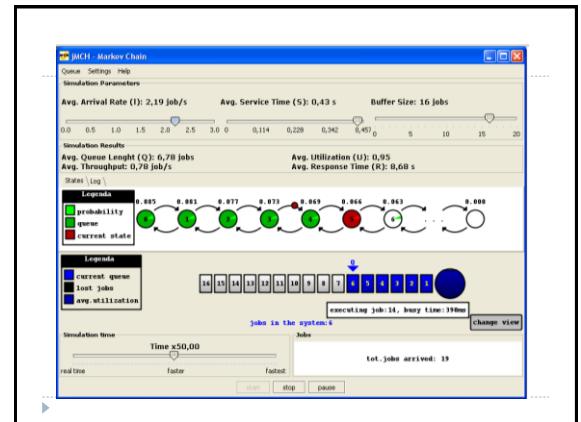
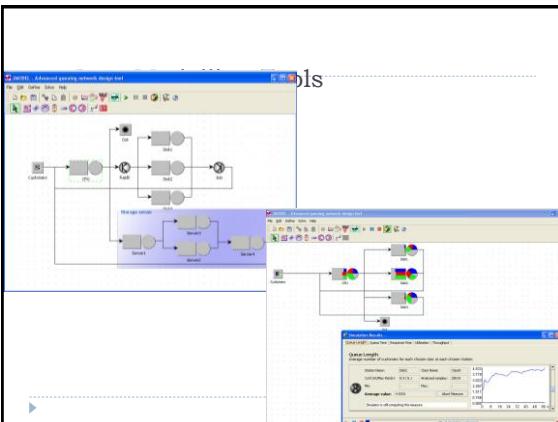
- ▶ Symulacyjny model do analizy systemów kolejkowania

[Airport](#)

[Terminal](#)

## VISSIM

- ▶ [Movie](#)



## TEORIA MASOWEJ OBSŁUGI TEORIA KOLEJEK

mgr inż. Daria Jagielo