

sieci komputerowe
wykład 3 - komutacja pakietów
część druga

literatura podstawowa

wykład prawie w całości przygotowany na podstawie
tekstu i rysunków
z rozdziału 4 w książce:

L.L. Peterson, B.S. Davie
„Sieci komputerowe. Podejście systemowe”
Wydawnictwo Nakom, Poznań 2000

komutacja komórek (ATM)

- *tryb przesyłania asynchronicznego* - ATM (Asynchronous Transfer Mode)
- technika komutacji, *podobna* do komutacji kanałów wirtualnych, będąca przykładem *komutacji komórek*
- faza nawiązanie połączenia - *sygnalizacja* (Q.2931):
 - znalezienie właściwej trasy w sieci
 - przydział zasobów w komutatorach w kanale w celu uzyskania określonej *jakości usługi* (QoS)
- pakiety mają *stałą długość: 53 bajty* (nagłówek 5 bajtów + dane 48 bajtów), nazwano je *komórkami*

rozmiar komórki

- *pakiety o zmiennej długości mają wiele zalet:*
 - *jeden bajt do nadania* - nie trzeba znaków wypełniających
 - *dużo danych do nadania* - podział na pakiety - maleje stosunek bajtów nagłówka do bajtów danych, rośnie efektywność, minimalizacja ogólnej liczby przesyłanych pakietów
- *dlaczego stosuje się pakiety o stałej długości?*
 - ułatwienie implementacji szybkich, wysoce skalowalnych komutatorów sprzętowych: *łatwiej przetwarzać pakiet o stałym rozmiarze, łatwiej zrównoleglić funkcje*
 - zachowanie się kolejek - *dokładniejsza ich kontrola*

kontrola kolejek

- kolejka w komutatorze powstaje, gdy ruch kierowany *na jedno wyjście*
- wyjście kolejki *jest zajęte maksymalnie przez czas*, jaki zajmuje nadanie pakietu o maksymalnym rozmiarze, przy pakiecie o stałej długości, czas *jest zwykle krótszy*
- *przykład:*
- pakiet o zmiennej długości, maks. 4kB, łącze 100Mb/s: czas transmisji: $4096 \cdot (8/100) \cdot 10^{-6}\text{s} = 327,68\mu\text{s}$
- pakiet o stałej długości 53 B: $53 \cdot (8/100) \cdot 10^{-6}\text{s} = 4,24\mu\text{s}$

kontrola kolejek

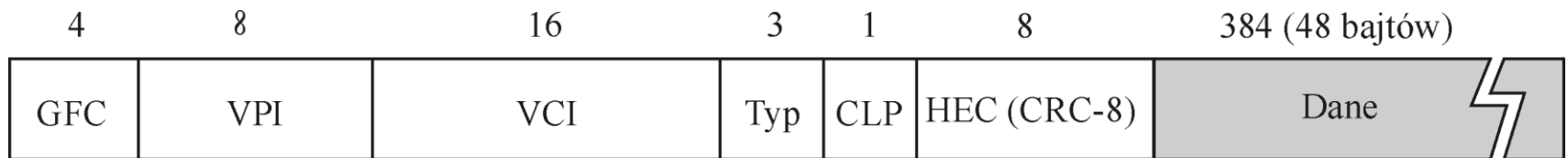
- kiedy pakiet przychodzi do pustej kolejki, *komutator czeka na cały pakiet*, zanim zacznie nadawać pakiet do łącza wyjściowego
- gdy duży pakiet zastąpimy przez „*pociąg*” złożony z *małych komórek*, to gdy pierwsza z komórek jest w kolejce, można ją nadawać
- *przykład*: dwa pakiety 4kB przychodzą do kolejki prawie w tej samej chwili, wtedy łącze bezczynne przez 327,68 μs i na końcu tego czasu jest 8 kB w kolejce, gdy te same pakiety w postaci „pociągu” komórek po 53B, wtedy transmisja po 4,24 μs

jaka jest prawidłowa długość komórki?

- *czynniki mające na to wpływ:*
- maksymalizacja *efektywności wykorzystania łącza*
- ATM wyrasta z telefonii: efektywne przesyłanie głosu:
 - cyfrowe kodowanie głosu 64kb/s (8 bitowe próbki pobierane z częstotliwością 8kHz)
 - 1 B próbkowany co $125 \mu\text{s} = 1 / (8 \times 10^3)\text{s} = 125 \times 10^{-6}\text{s}$
 - zebranie komórki n bajtów zajmie $n \times 125 \mu\text{s}$, n.p. przy 1000 bajtach wynosi 125ms, opóźnienie zauważalne dla głosu, powoduje *echo* → *układy kasowania echa*
- wybór długości komórki przyjęty w normie ATM jest *kompromisem*: 48 B danych (64B US, 32B Eu) + 5B nagłówka

format UNI komórki ATM

- *dwa formaty komórek*: **UNI** między komputerem a komutatorem, **NNI** między komutatorami (VPI=12b)



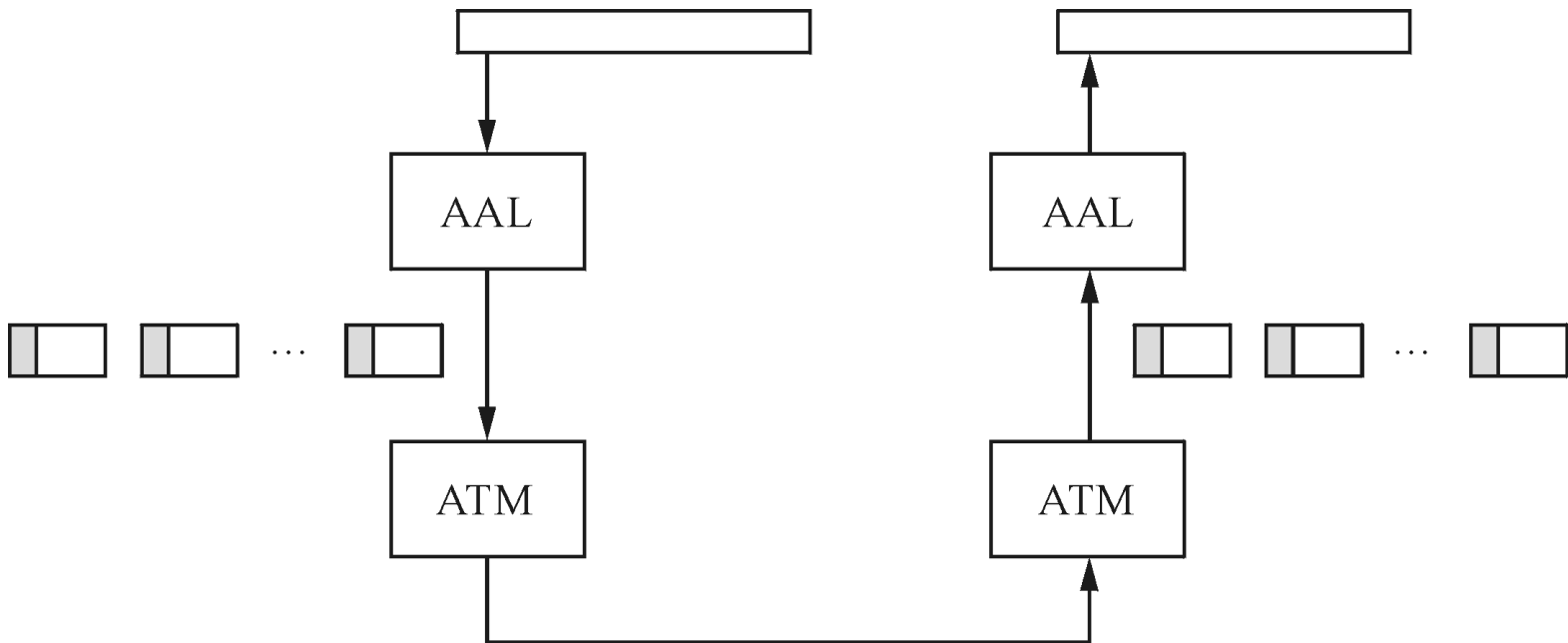
- **GFC**=ogólne sterowanie przepływem (4 bity)
- **VPI**=identyfikator ścieżki wirtualnej (8 bitów)
- **VCI**=identyfikator kanału wirtualnego (16 bitów)
- **typ**=typ (3 bity) = **1xx** zarządzanie, **0yz** dane użytkownika
y jawny wskaźnik przeciążenia, z sygnalizacja użytkownika
- **CLP**=priorytet straty komórki (1 bit, =1 komórka do usunięcia)
- **HEC**=sprawdzenie błędu nagłówka (CRC, 8 bitów)

segmentacja i składanie

- dotychczas protokół z niższego poziomu akceptował pakiet z wyższego poziomu i jedynie dołączał nagłówek - *w ATM to jest niemożliwe*, gdyż pakiet z wyższego poziomu zwykle > 48 bajtów
- rozwiązanie: *fragmentacja* u nadawcy, *składanie* u odbiorcy (w ATM: *segmentacja* i *składanie*)
- w tym celu dodano nową warstwę między ATM a warstwą stosującą pakiety o zmiennej długości, n.p. IP, o nazwie *AAL* (ATM Adaptation Layer)

AAL

- *segmentacja i składanie w ATM*



warstwy AAL

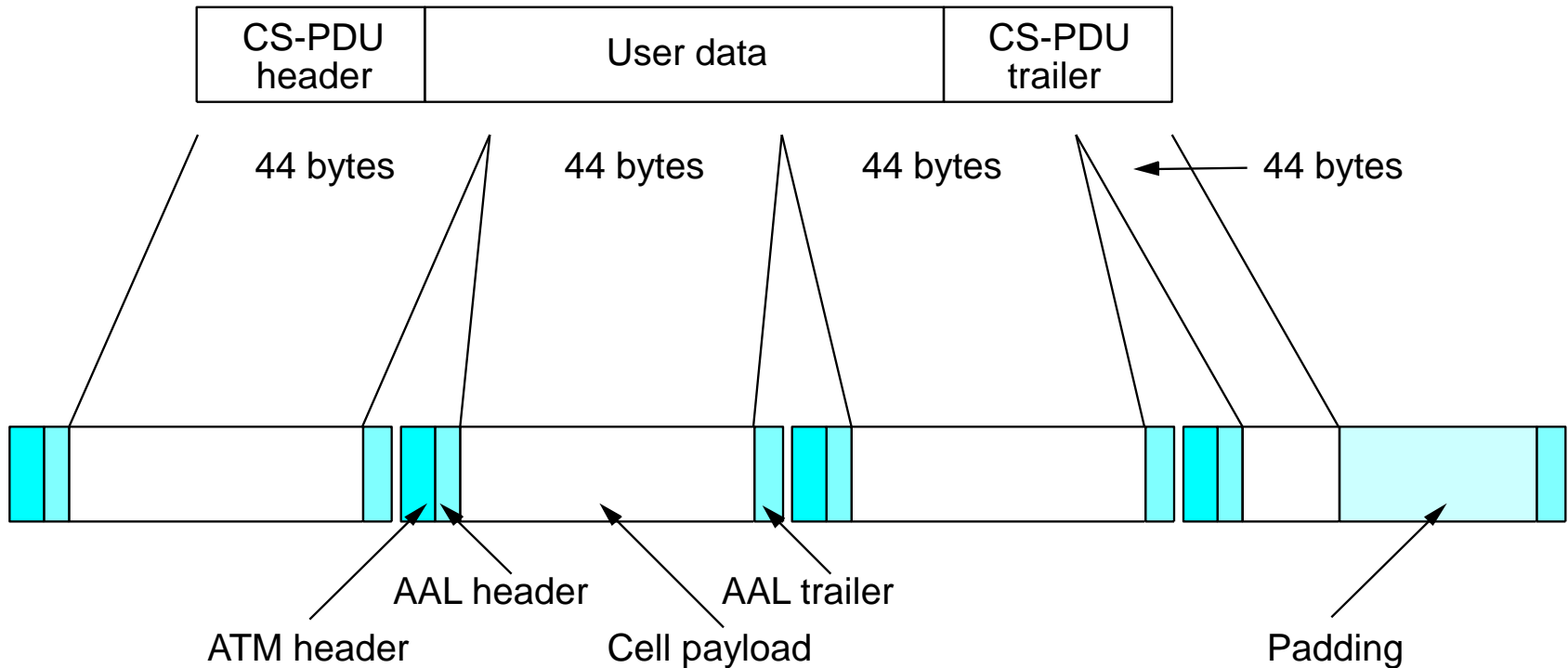
- *warstwy AAL*: 1, 2, 3/4, 5:
- 1, 2 aplikacje głosowe,
- 3 i 4 transmisja pakietów danych (3 dla X25, 4 dla IP)
- 5 usuwa wady 3 i 4

warstwa adaptacyjna AAL 3/4

- *obsługuje proces* segmentacji i składania i angażuje dwa różne formaty *pakietów* (nazwanych tutaj *jednostkami danych protokołu - PDU*):
- *jednostka danych protokołu podwarstwy zbieżności*, zwana w skrócie CS-PDU, definiuje sposób kapsułkowania jednostek PDU o zmiennej długości przed segmentacją ich na komórki
- drugą część AAL3/4 stanowią *nagłówek i ogon*, przenoszone w każdej komórce
- CS-PDU poddany segmentacji na *44 B porcje*, do każdej z nich dołączony nagłówek i ogon AAL3/4, *razem 48 B*, przenoszone jako dane w *komórce ATM*

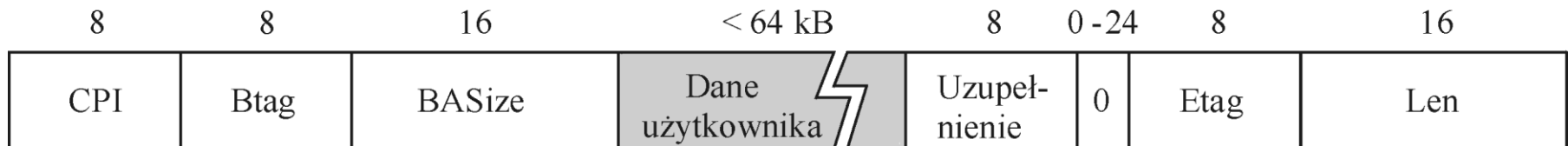
kapsułkowanie i składanie w AAL3/4

Peterson & Davie, Computer Networks, 2nd ed, MKP 2000



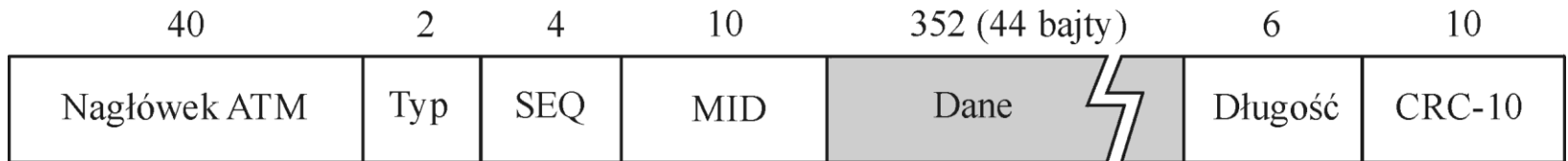
format pakietu warstwy adaptacyjnej ATM 3/4

- jednostka danych protokołu podwarstwy zbieżności (CS-PDU)



- **CPI** - wskaźnik części wspólnej (=0, pole wersji CSPDU)
- **Btag** - identyfikator początkowy
- **BASize** - pole rozmiaru alokacji bufora, (wskazówka)
- **uzupełnienie** (same 0)
- **Etag** - identyfikator końcowy = **Btag**
- **Len** - faktyczna długość PDU

format komórki ATM dla AAL3/4



- **Typ** = BOM, COM, EOM, SSM (2 bity, patrz dalej)
- **SEQ** = *numer sekwencyjny* (dla wykrycia straty komórki albo zmiany kolejności)
- **MID** = *identyfikator* (do) *multipleksacji* (kilku PDU na jedno połączenie)
- **Długość** = liczba bajtów w komórce (44 dla BOM i COM)
- **CRC-10** = 10 bitowa CRC do kontroli błędów w komórce

pole typu AAL3/4

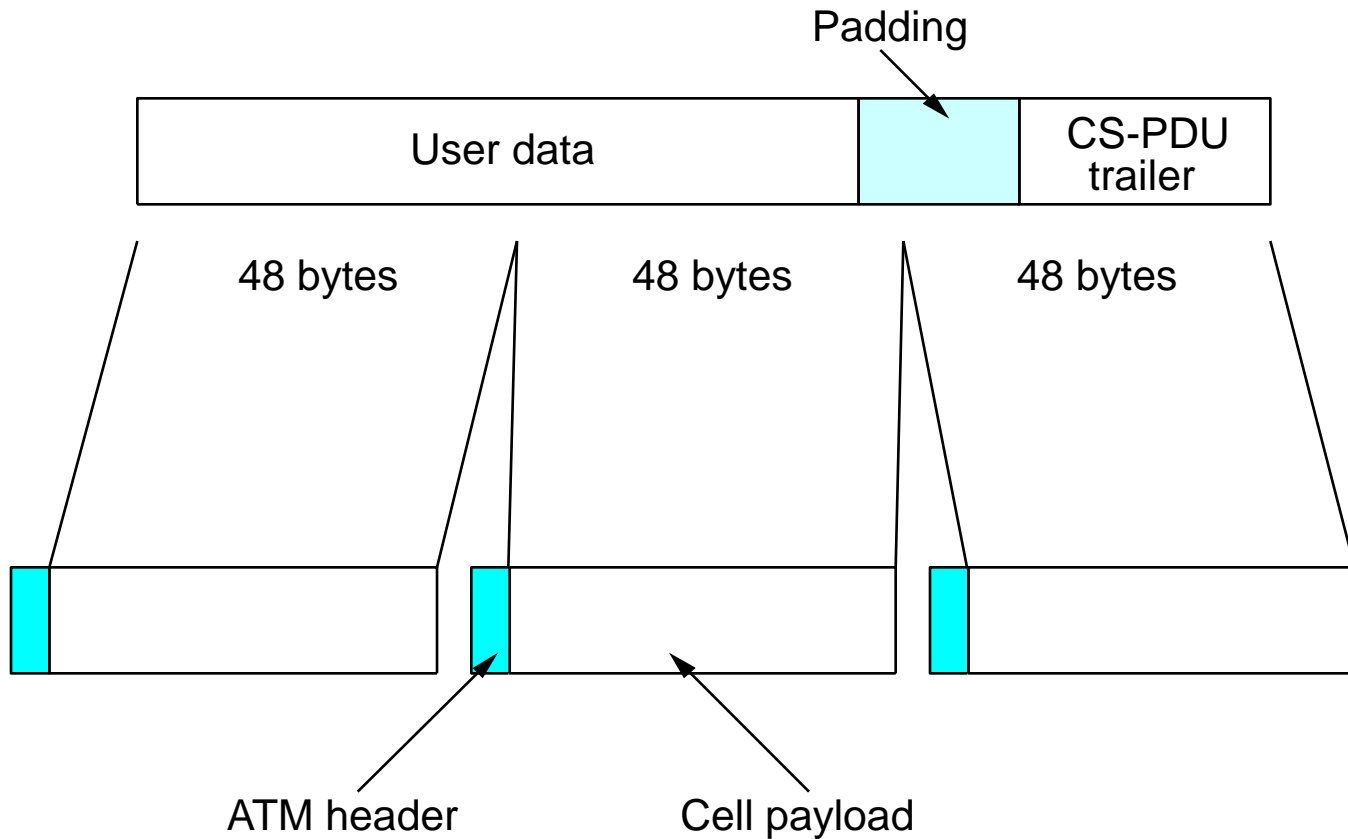
wartość	nazwa	znaczenie
10	BOM	początek komunikatu
00	COM	kontynuacja komunikatu
01	EOM	koniec komunikatu
11	SSM	komunikat jedno-segmentowy

warstwa adaptacyjna ATM 5

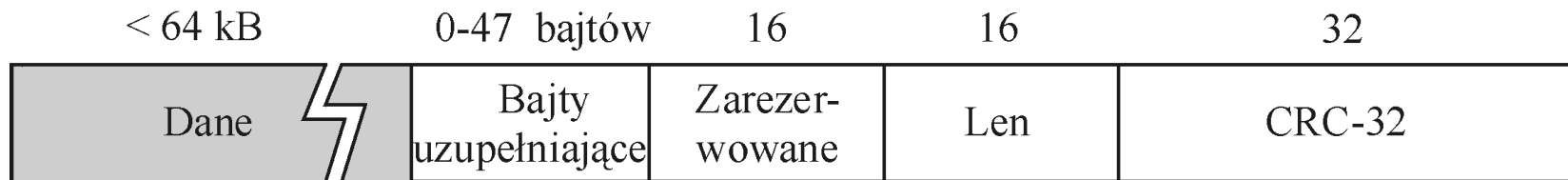
- *krytyka AAL3/4* (zbyt wiele pól do prostej funkcji segmentacji i składania)
- *ruch „postaw na bit”* - 1 bit w nagłówku ATM (zamiast nagłówka AAL) do wyznaczenia końca ramki, to segmentację i składanie można uzyskać bez użycia jakiegokolwiek z 48 bajtów danych ATM
- *zastąpienie 2 bitów pola typu z AAL3/4 przez 1 bit w nagłówku komórki ATM* -ustawienie tego bitu identyfikuje ostatnią komórkę w PDU
- kolejna komórka jest pierwszą komórką następnej PDU

kapsułkowanie i składanie w AAL5

Peterson & Davie, Computer Networks, 2nd ed, MKP 2000



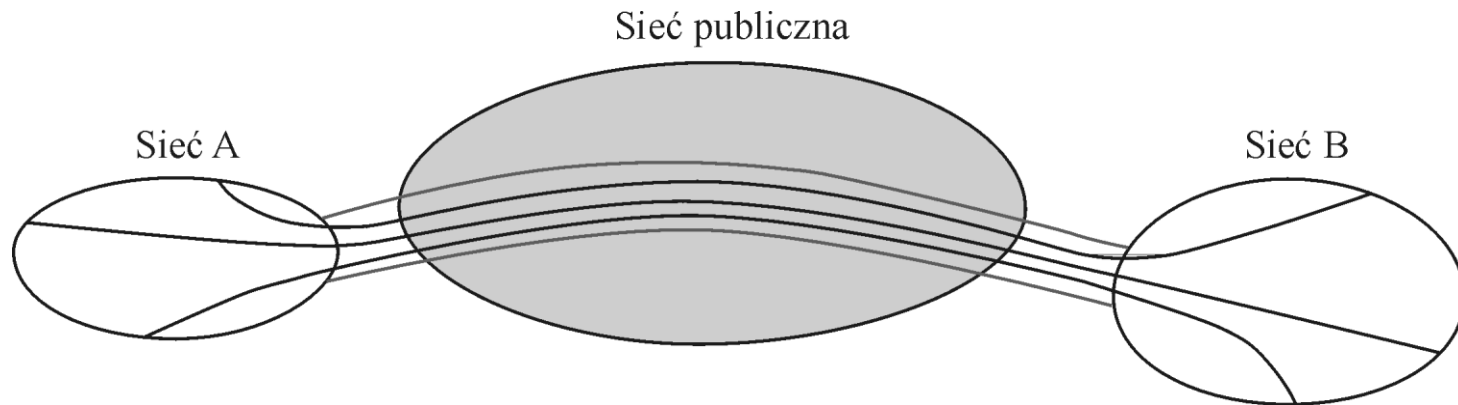
format pakietu warstwy adaptacyjnej ATM 5



- *dane + bajty uzupełniające + 8 bajtowy ogon* (2 bajty zarezerwowane=0, 2 bajty długości PDU, 4 bajty CRC-32)
- *0 - 47 bajtów uzupełniających* (zapewnia, że ogon zawsze przypada na koniec komórki ATM)
- AAL5 to taki zakres funkcji jak AAL3/4, z wyjątkiem multipleksacji - zaletą AAL5 jest prostota w stosunku do AAL3/4
- AAL5 zastosowana do nadawania datagramów IP z wykorzystaniem ATM

ścieżka wirtualna

- w ATM *24 bitowy identyfikator* kanałów wirtualnych:
VPI 8bitów + VCI 16 bitów



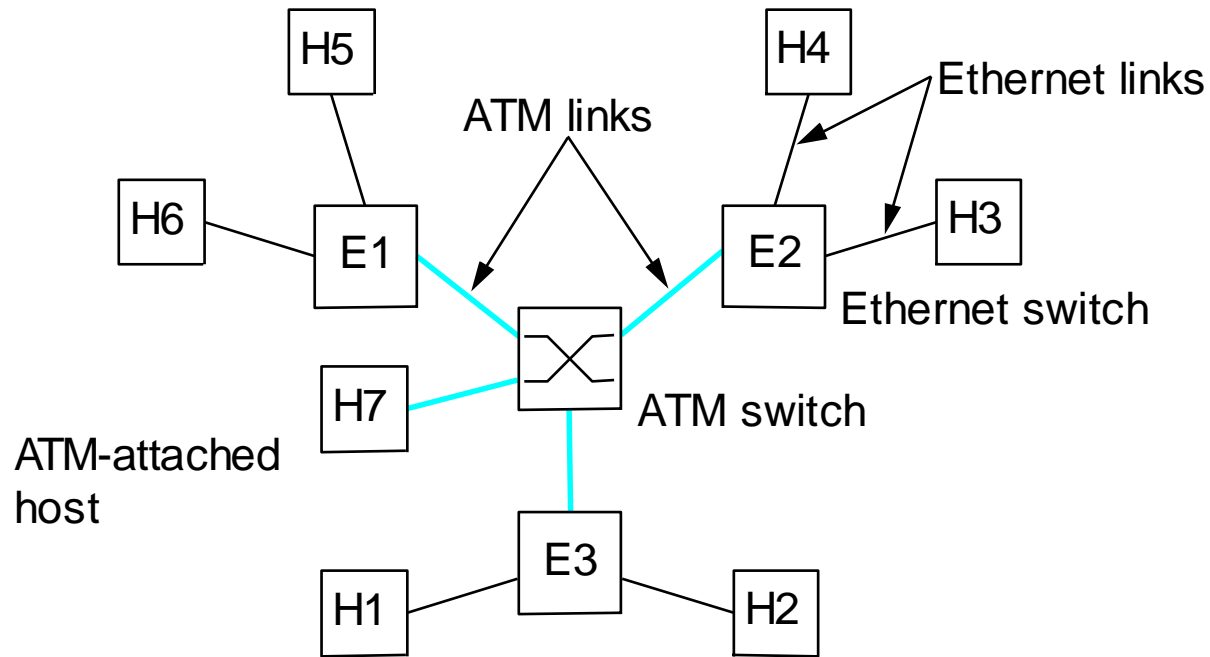
- *ścieżka wirtualna między oddziałami A i B* ustalona w sieci publicznej za pomocą VPI (8 bitów)
- w oddziałach korporacji **A** i **B** do komutacji VPI + VCI (24 bity) w sieci komutatorów ATM (hierarchia)

warstwy fizyczne dla ATM

- ATM może być uruchomiony na kilku mediach i z kilkoma protokołami warstwy fizycznej
- początkowo była to sieć optyczna SONET
- problem znalezienia granic komórki w sieci SONET (problem wyznaczania ramek)

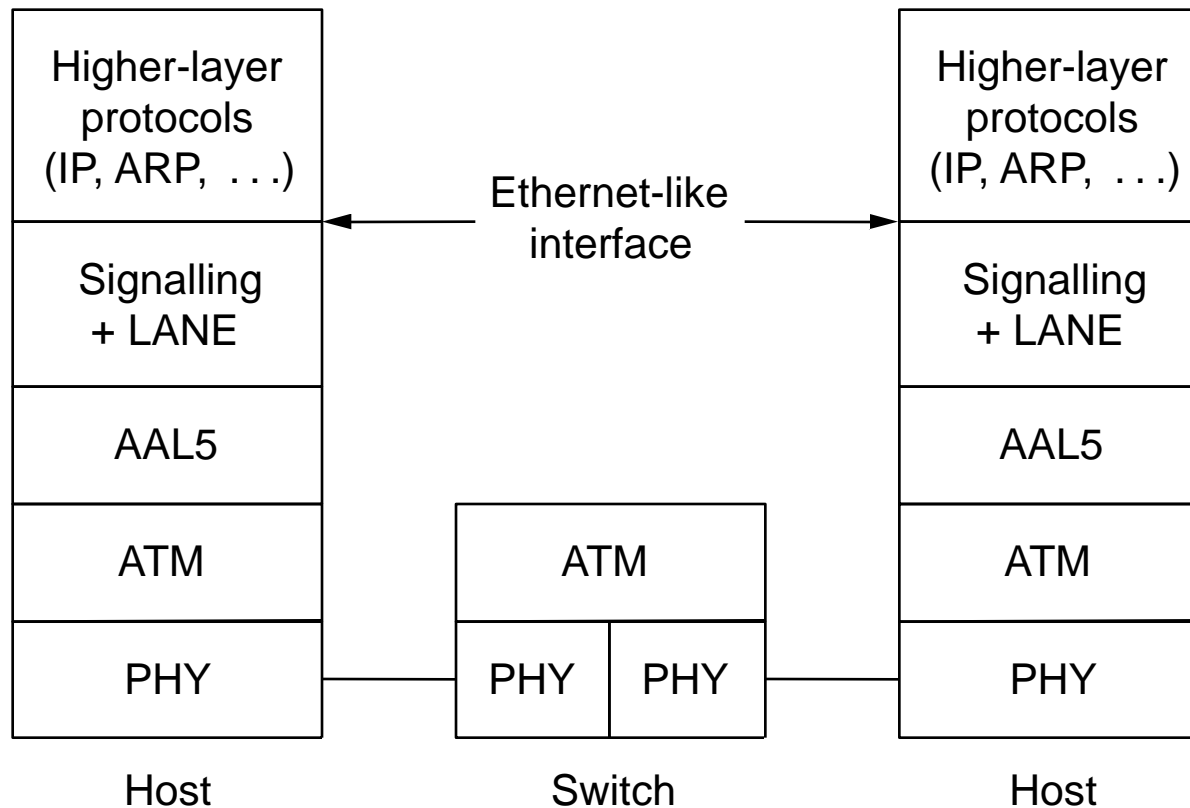
ATM jako kręgosłup sieci lokalnej

Peterson & Davie, Computer Networks, 2nd ed, MKP 2000



warstwy protokołów przy emulacji sieci lokalnej

Peterson & Davie, Computer Networks, 2nd ed, MKP 2000

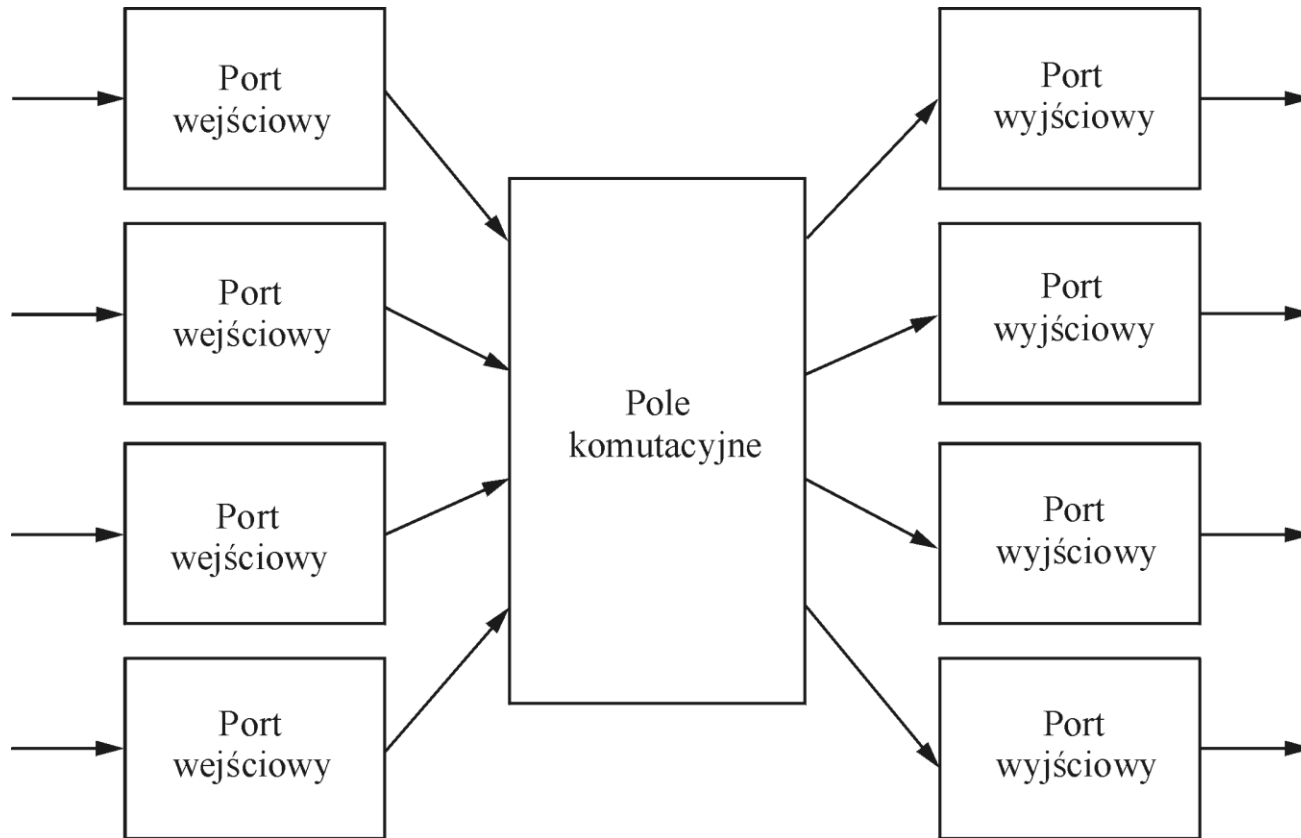


sprzęt komutujący

- komutator jest urządzeniem *wieloweściowym i wielowyściowym*
- zadaniem komutatora jest *przekazanie jak największej liczby pakietów z wejść na odpowiednie wyjścia*
- dwa wyzwania przy projektowaniu komutatora:
 - *przepustowość* - nie większa niż minimum z sumy przepustowości na wejściach i sumy przepustowości na wyjściach (albo na wyjściu...) - problem z definicją, problem z modelowaniem ruchu, rywalizacja
 - *skalowalność* - wzrost kosztów w funkcji liczby wejść i wyjść, maksymalny rozmiar komutatora

budowa komutatora

porty wejściowe pole komutacyjne porty wyjściowe



buforowanie w komutatorze

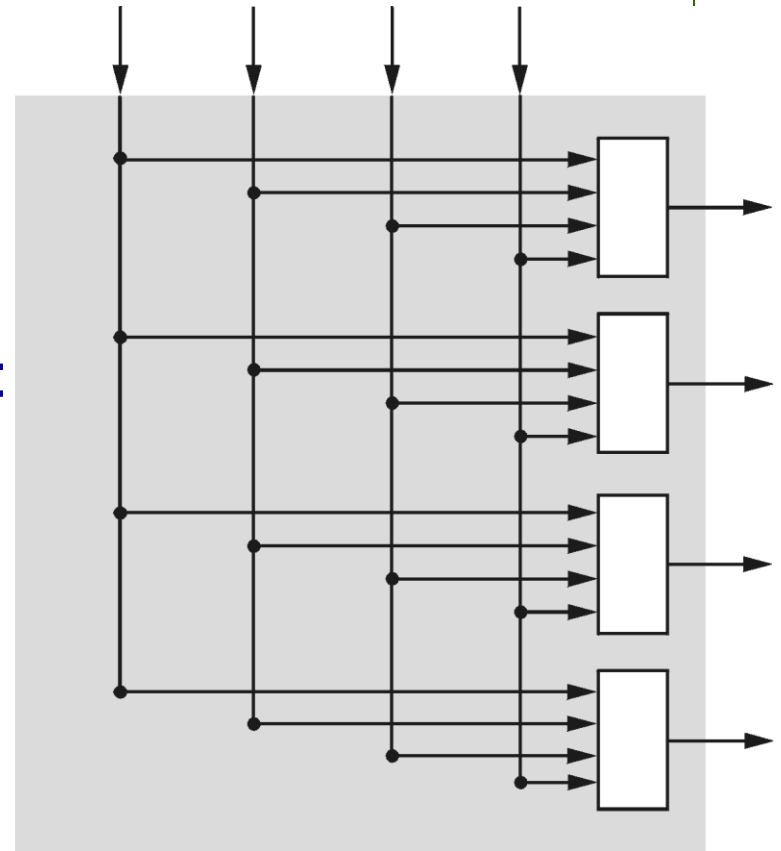
- *port*: odbiornik, bufory, nadajnik, tablice VC
- buforowanie w portach wejściowych - nie stosuje się, ze względu na *blokowanie od czoła łącza*:



- buforowanie w polu komutacyjnym (*wewnętrzne*)
- buforowanie w portach wyjściowych

komutatory krzyżowe $n \times n$

- n wejść, n wyjść
- rywalizacja o port wyjściowy
- każde wejście połączone z każdym wyjściem
- *pudełko* związane z wyjściem:
 - rozpoznaje pakiety do tego wyjścia
 - obsługuje rywalizację
- złożoność komutatora $\geq n^2$

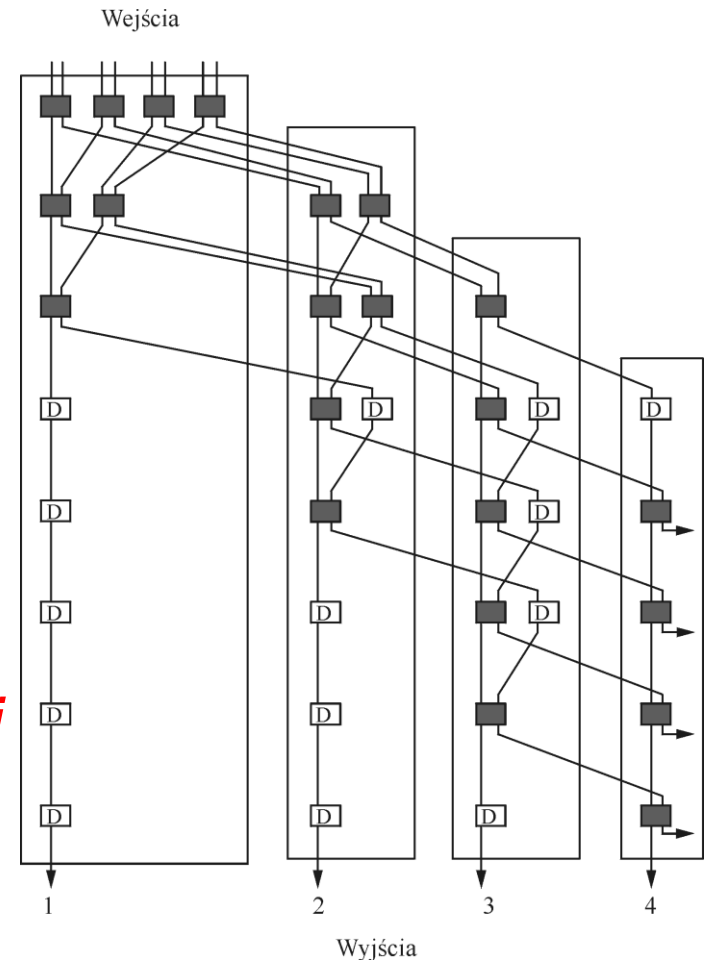


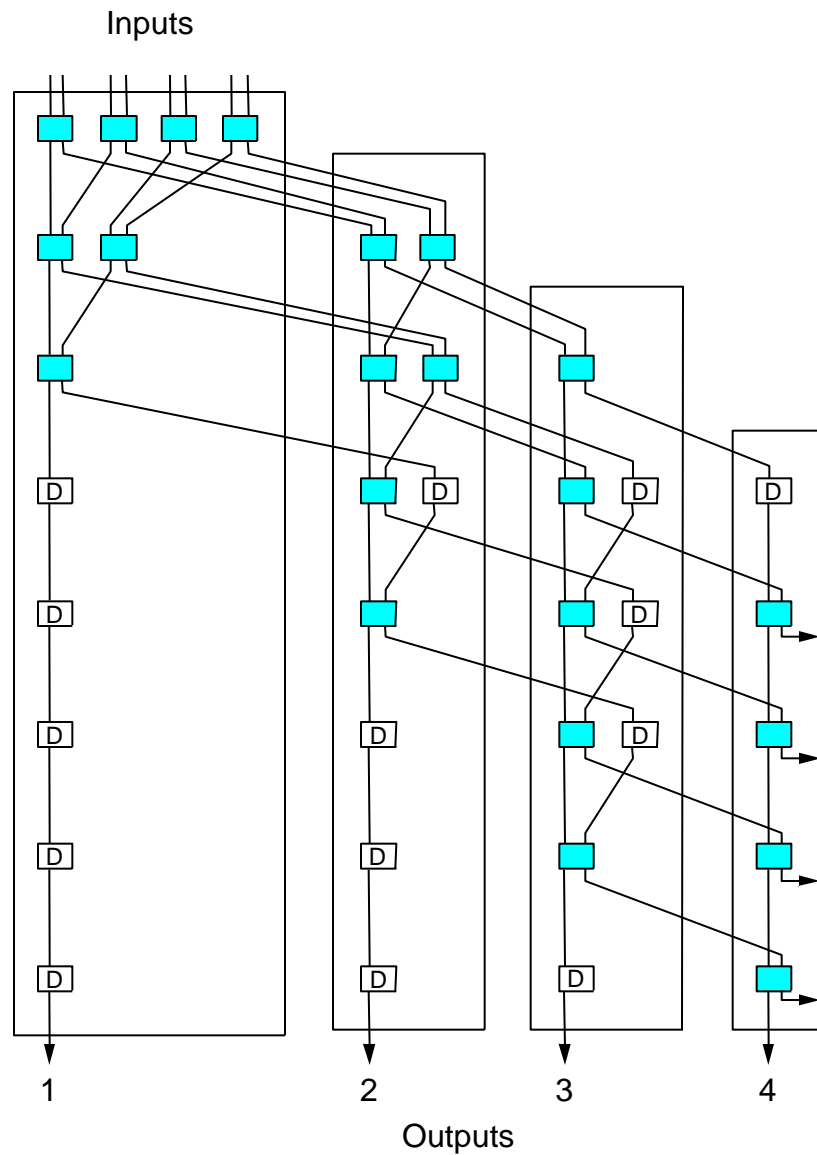
komutator działający w systemie pucharowym

- *przyjęto założenia dotyczące ruchu*, w celu redukcji złożoności portów wyjściowych
- prawdopodobieństwo, że wiele wejść nadaje do jednego wyjścia (*punkt zapalny*) jest bardzo małe
- *pomysł*: zaprojektować port akceptujący l pakietów jednocześnie, $l < n$
- *port wyjściowy*: filtr pakietów, koncentrator wybierający l pakietów i odrzucający resztę, kolejka długości l
- pakiety walczą *w systemie pucharowym*, wybierając l zwycięzców z n , w l rundach

komutator działający w systemie pucharowym

- $n = 8, l = 4$
- *element komutujący* 2×2 wybiera losowo zwycięzcę
- *element opóźniający* D
- *zwycięzca* gra $\log_2 n$ rund
- *przeegrany* przechodzi do kolejnej *sekcji* i gra baraże, gdy *wygrywa*, pozostaje w *sekcji*
- pakiet *przeegrany* w l sekcjach *usuwany*

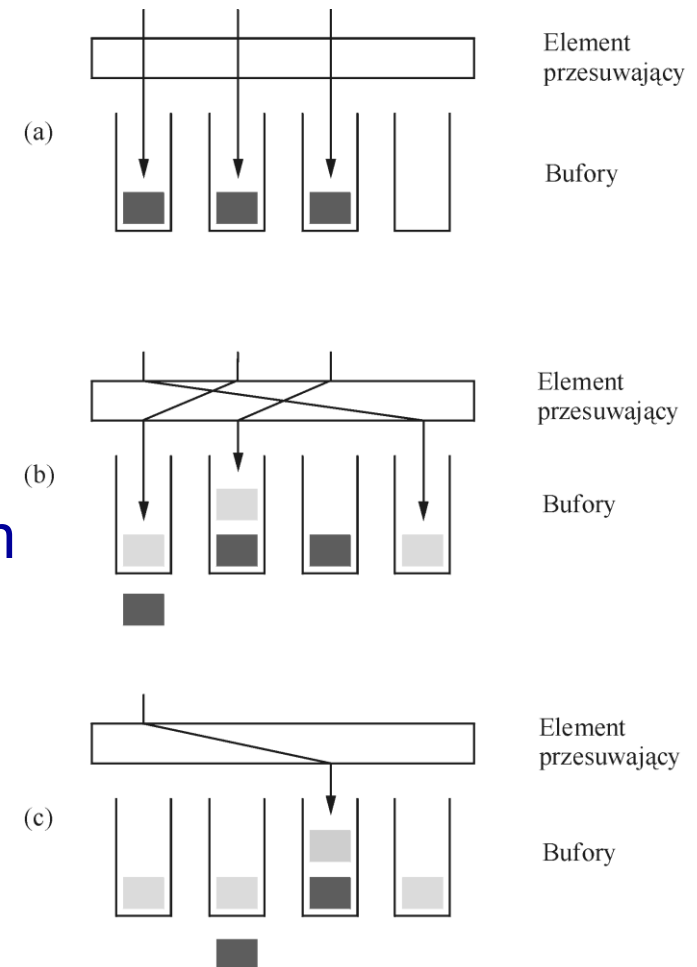


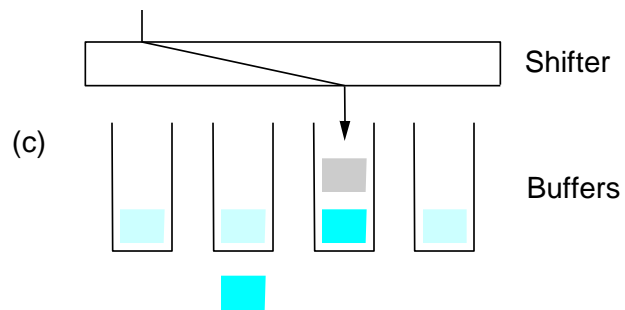
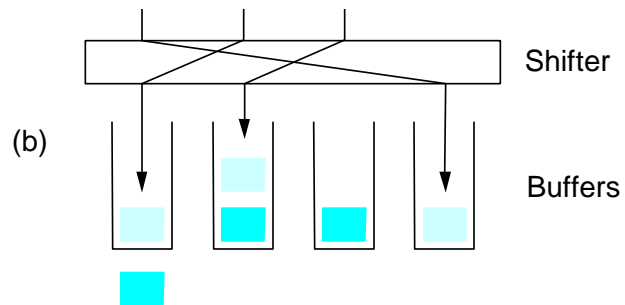
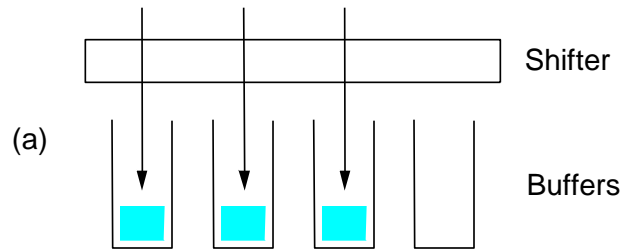


bufor współdzielony komutatora działającego w systemie pucharowym

bufor wyjściowy:

- akceptuje l pakietów i nadaje jeden pakiet w jednym cyklu
- tablica l buforów poprzedzona przez element przesuwający, zapełniana w trybie karuzelowym
- pakiety odczytywane po jednym, w trybie karuzelowym
- złożoność komutatora n^2



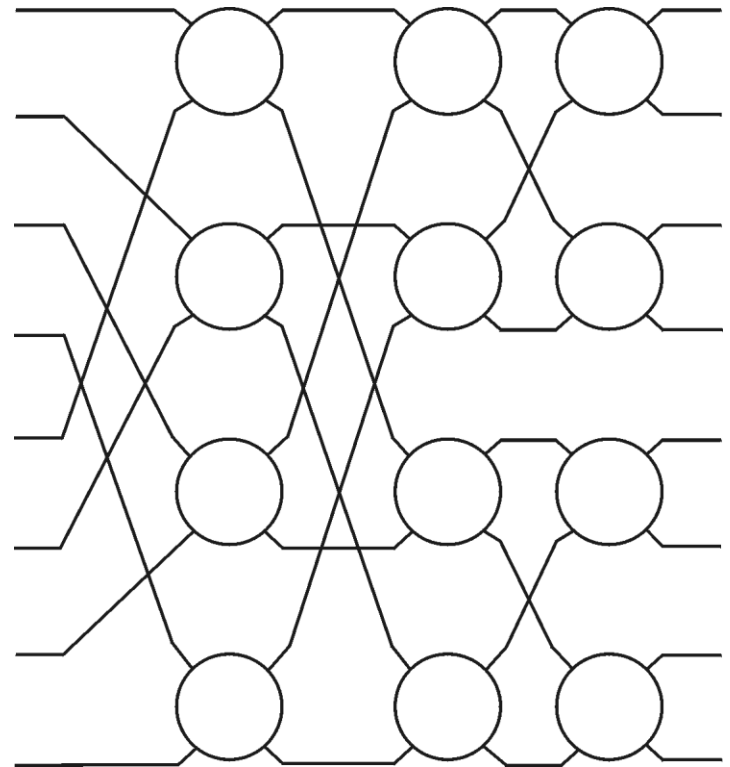


pola komutacyjne same wybierające trasę

- *definicja*: klasa komutatorów utworzonych z wielu małych, wzajemnie połączonych elementów komutujących, w których podejmowane są lokalne decyzje komutujące
- pakiety znajdują swoją *własną trasę* przez pole komutacyjne na podstawie *sekwencji takich decyzji*
- *ogólna zasada*: każdy pakiet przenosi wystarczającą informację w swoim nagłówku, aby pozwolić małym elementom komutującym podejmować decyzje
- *nagłówek* ten dodaje się przed wejściem w pole komutacyjne i usuwa w porcie wyjściowym

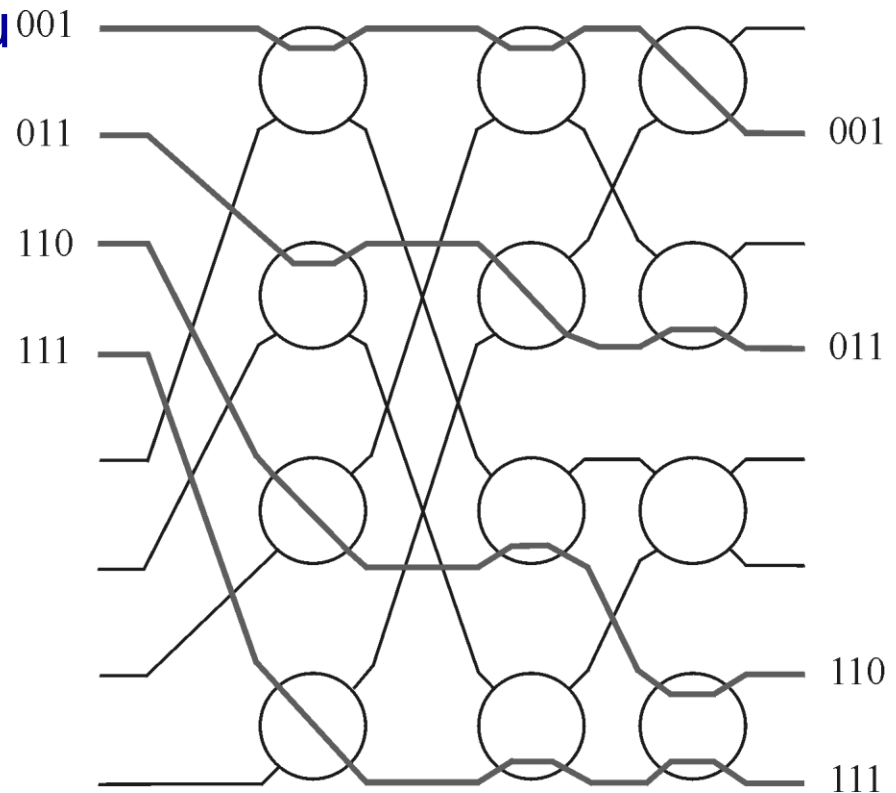
pole typu *banyan*

- wykonane z elementów 2×2 , dokonujących przełączenia na podstawie *jednego bitu w nagłówku*
- kierują pakiet do dolnego albo górnego wyjścia, w zależności od wartości tego bitu
- unikają kolizji, jeżeli pakiety występują w porządku *rosnącym* (ich numery *wzrastają w dół rysunku*)



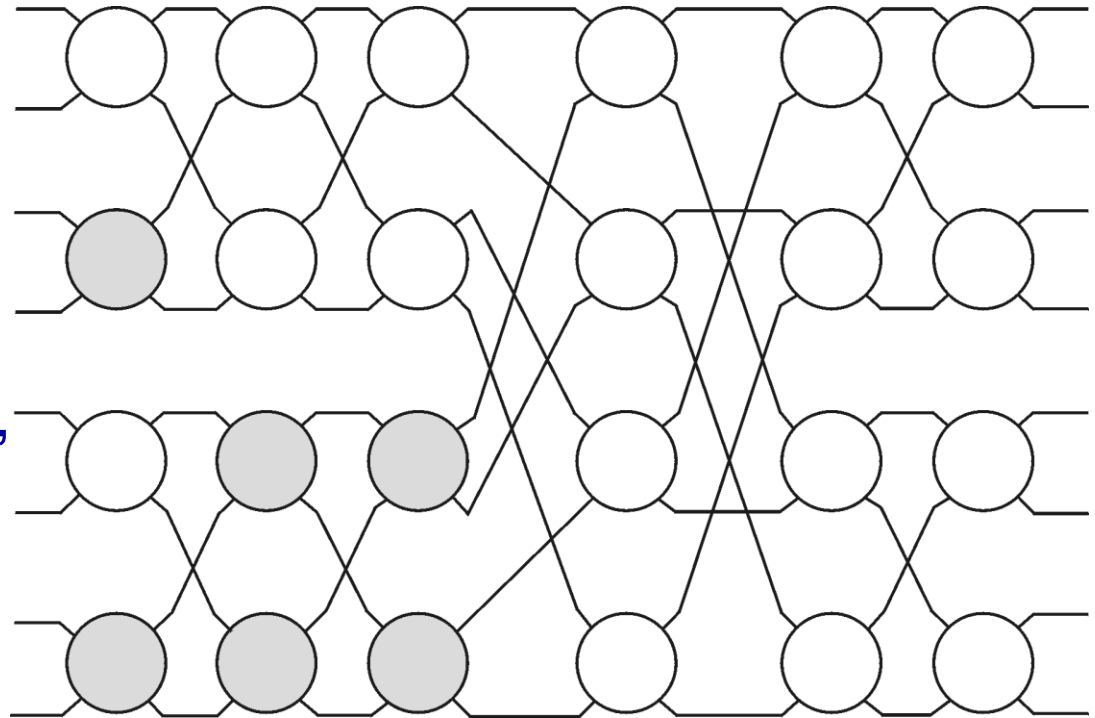
kierowanie pakietów przez pole typu *banyan*

- *pierwsza kolumna* elementów sprawdza najbardziej znaczący bit numeru portu wyjściowego w nagłówku
- bit=0 do *górnego wyjścia*
- bit=1 do *dolnego wyjścia*
- wszystkie *górne* wyjścia prowadzą do właściwej *połówki* sieci
- następną kolumną do właściwej *ćwiartki* sieci
- ostatnią kolumną do właściwego *portu wyjściowego*



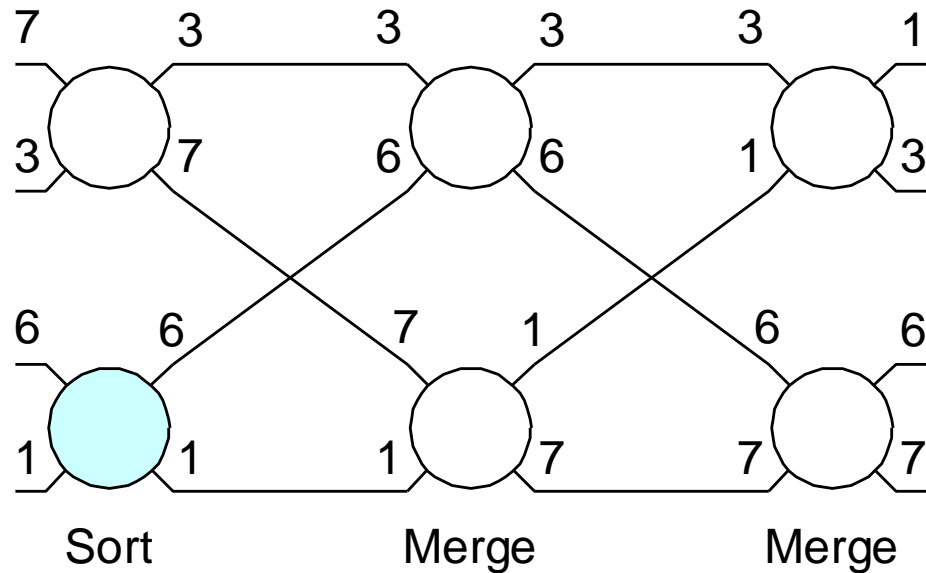
sieć Batchera

- *sieć sortująca* pakiety w kolejności
- sieć Batchera przed polem *banyan*=*pole nieblokujące* (nie ma kolizji, gdy do innych portów wyjściowych)
- *pełne porównanie nagłówków*, gdy elem.zaciemniony: *mniejszy wyróżnik do dolnego wyjścia, większy wyróżnik do górnego wyjścia*
równe: losowo



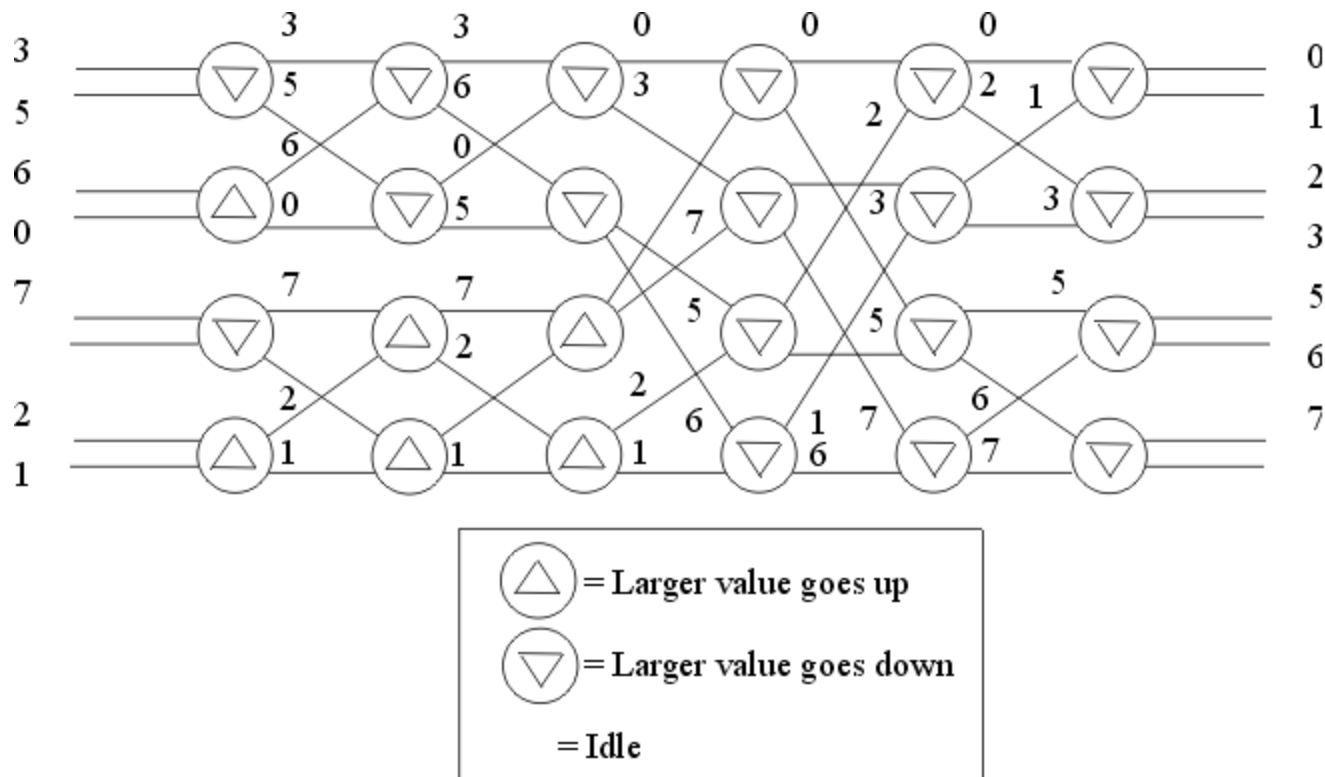
sortowanie przez scalanie

Peterson & Davie, Computer Networks, 2nd ed, MKP 2000



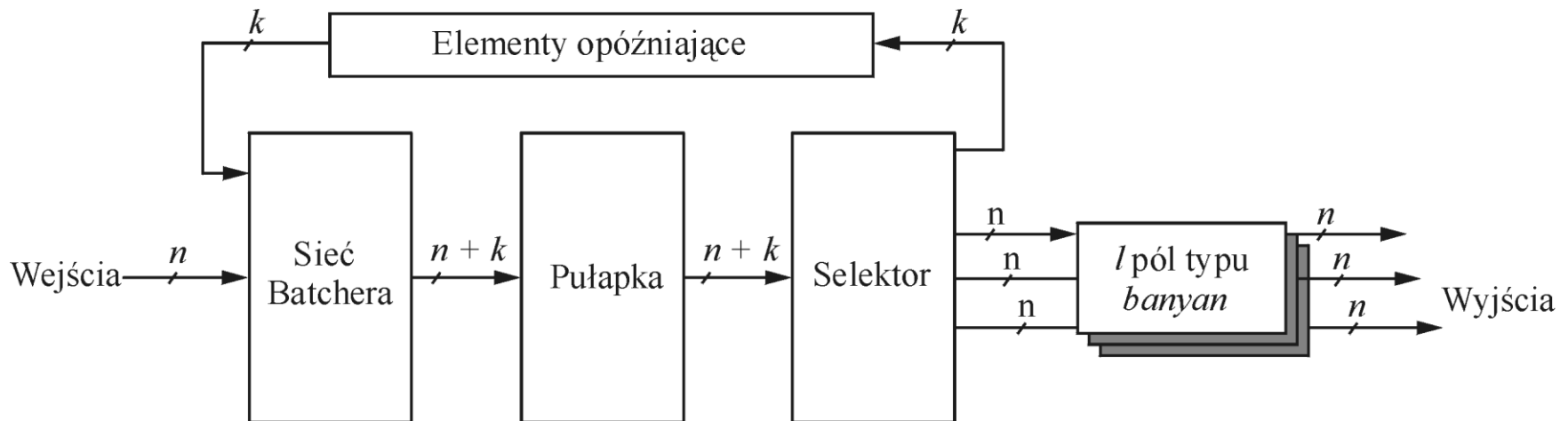
sieć Batchera

Walrand&Varaiya "High Performance Communication Networks",
MKP 1996



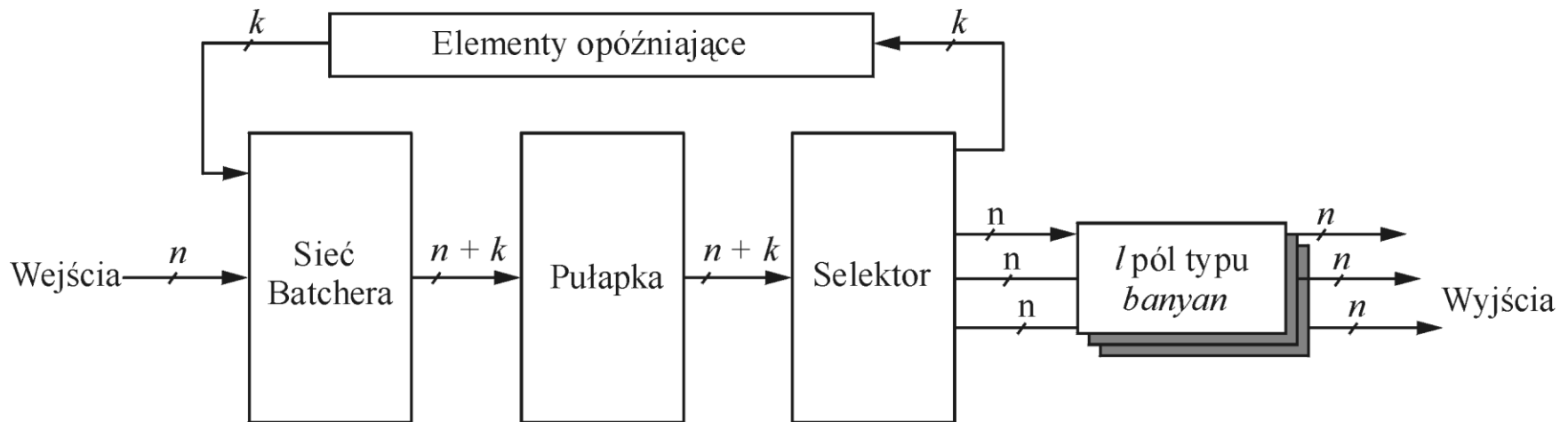
komutator *Sunshine*

- *usuwa ograniczenie*, że nie ma żadnych pakietów kierowanych do tego samego portu wyjściowego
- czyli unika *kolizji*
- *sieć Batchera + pole banyan + pułapka + selektor + elementy opóźniające*



komutator *Sunshine*

- *sieć Batchera*: sortowanie $n+k$ pakietów, n z portów wejściowych i k zawróconych
- *pułapka*: wyszukuje pakiety do zawrócenia ($>l$)
- *selektor*: l pakietów do tego samego wyjścia kieruje do różnych pól *banyan*
- *element opóźniający*: opóźnia pakiety zawracane



komutator współdzielący medium

- *zwinięcie sieci* współdzielącej medium do wnętrza komutatora i uznanie jej za pole komutacyjne
- *komutator współdzielący pamięć:*

jeden duży bufor
- gdy brakuje
bufora dla mocno
obciążonego
wyjścia, pożyczają
się go od mniej
obciążonego:

