

SIECI KOMPUTEROWE

wykład dla kierunku informatyka

semestr 4 i 5

dr inż. Michał Sajkowski

Instytut Informatyki PP

pok. 227G PON PAN, Wieniawskiego 17/19

Michal.Sajkowski@cs.put.poznan.pl

tel. +48 (61) 8 582 100

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

sieci komputerowe

wykład 2

sieci łączy bezpośrednich

literatura podstawowa

wykład prawie w całości przygotowany na podstawie
tekstu i rysunków
z rozdziału 3 w książce:

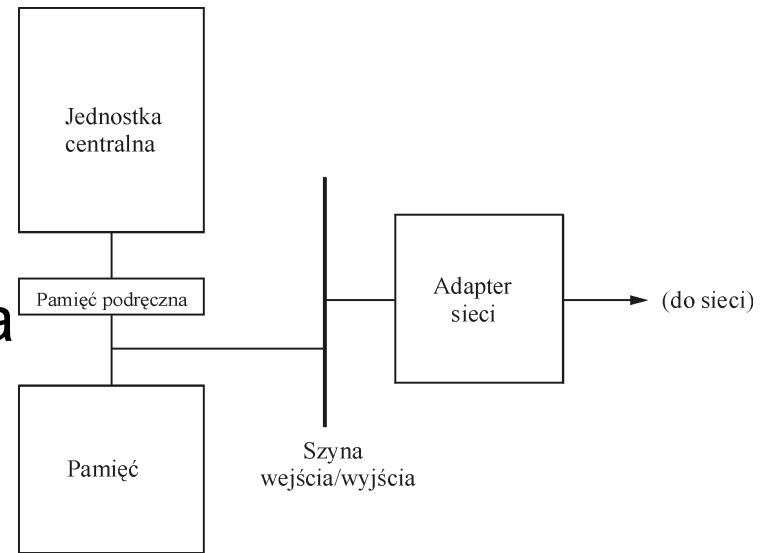
L.L. Peterson, B.S. Davie
„Sieci komputerowe. Podejście systemowe”
Wydawnictwo Nakom, Poznań 2000

problemy

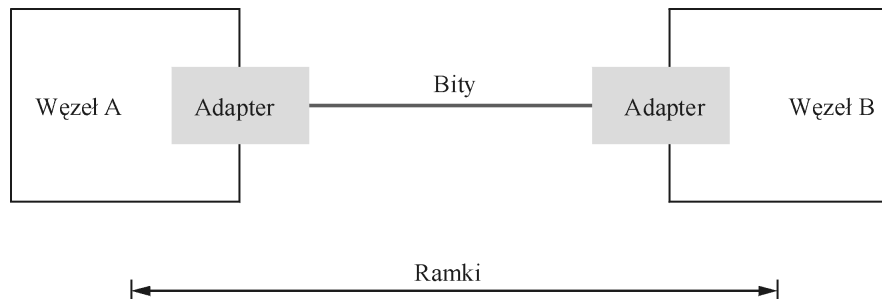
- *łączenie komputerów* bezpośrednio przez fizyczne medium to pierwszy krok, pozostają *problemy*:
 1. *kodowanie bitów* w przewodzie tak, aby były rozumiane przez komputer odbiorczy
 2. *rozpoznawanie ramek* - identyfikacja komunikatu w ciągu bitów, zgodnego z przyjętym formatem
 3. *wykrywanie błędów* transmisji ramek
 4. *niezawodna transmisja* - mimo uszkodzeń łącza
 5. *sterowanie dostępem do medium* - mediacja dostępu do współdzielonego łącza

adapter sieci

- wspomniane pięć funkcji realizuje się w *adapterze sieci*:
karcie podłączonej z jednej strony do szyny wejścia/wyjścia w komputerze, a z drugiej strony do łącza fizycznego



- między adapterami wymieniane są bity*, między komputerami *ramki*:



węzeł

- komputer ogólnego przeznaczenia (*stacja robocza*)

komputer

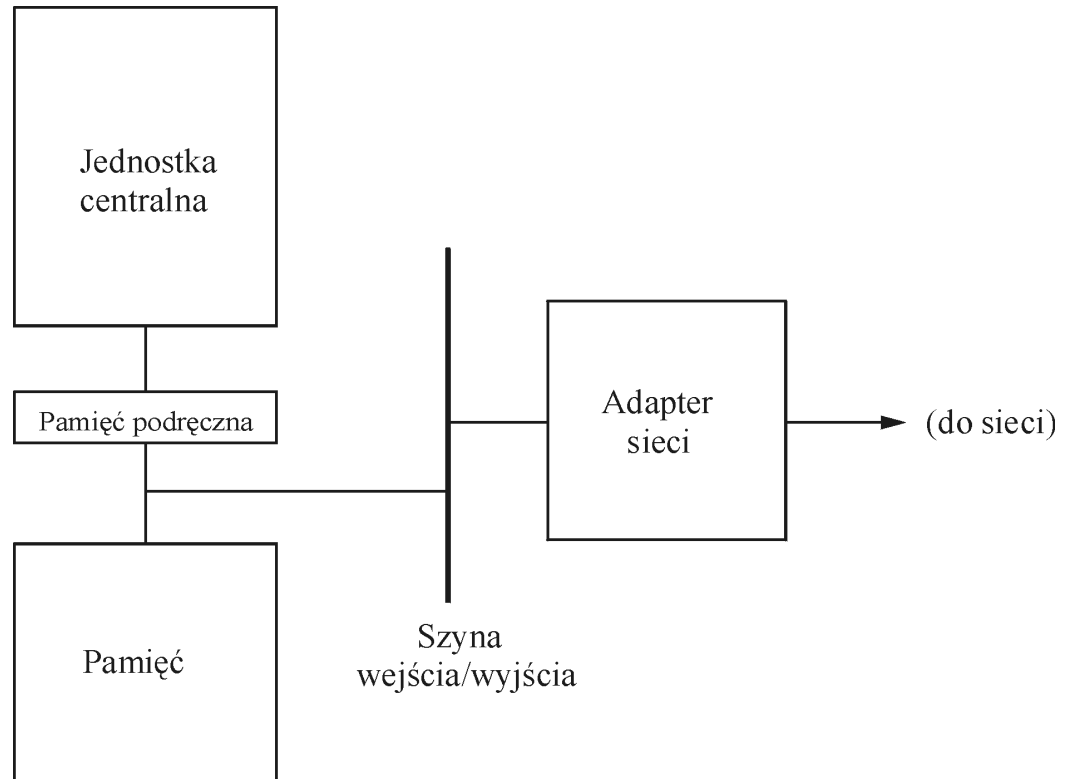
(programy aplikacyjne)

*komutator**

(komunikatów
między łączami)

*ruter**

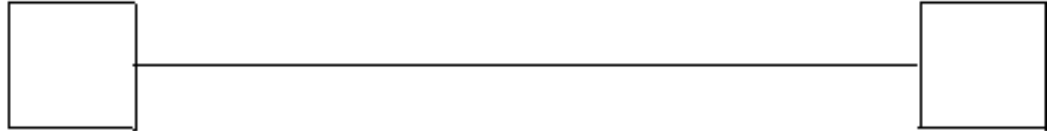
(pakietów między
sieciami)



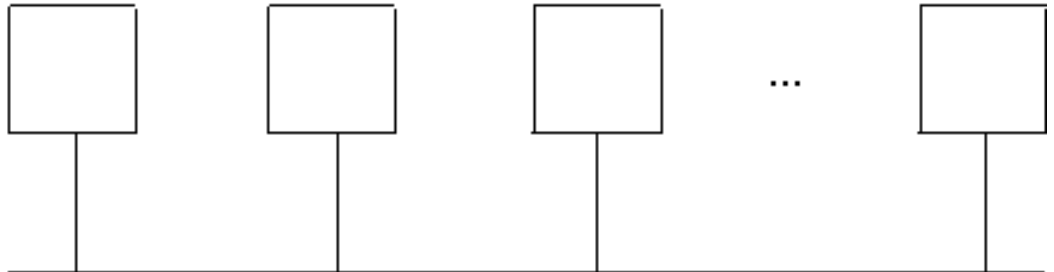
*także komputer specjalnego zastosowania

łącza bezpośrednie

- dwupunktowe:



- wielodostępne:



łącza

- *media fizyczne*: skrętka, kabel koncentryczny, światłowód, przestrzeń...
- *propagacja sygnałów*: cyfrowych i analogowych
- *kodowanie* danych binarnych w sygnale analogowym (*modem* - *modulator/demodulator*)
- *kodowanie* głosu w łączy cyfrowe: *codec* (*koder/dekoder*), np. w ISDN
- *nadajnik cyfrowy* (dane binarne do łączy cyfrowego)
- *łącza dwupunktowe*: pełnodupleksowe i półdupleksowe
- *szerokość pasma* i *zasięg* łączy lokalnych
- o dostępie do *łącza wielodostępnego* będzie później

typy kabli i światłowodów dostępne powszechnie dla łączy lokalnych

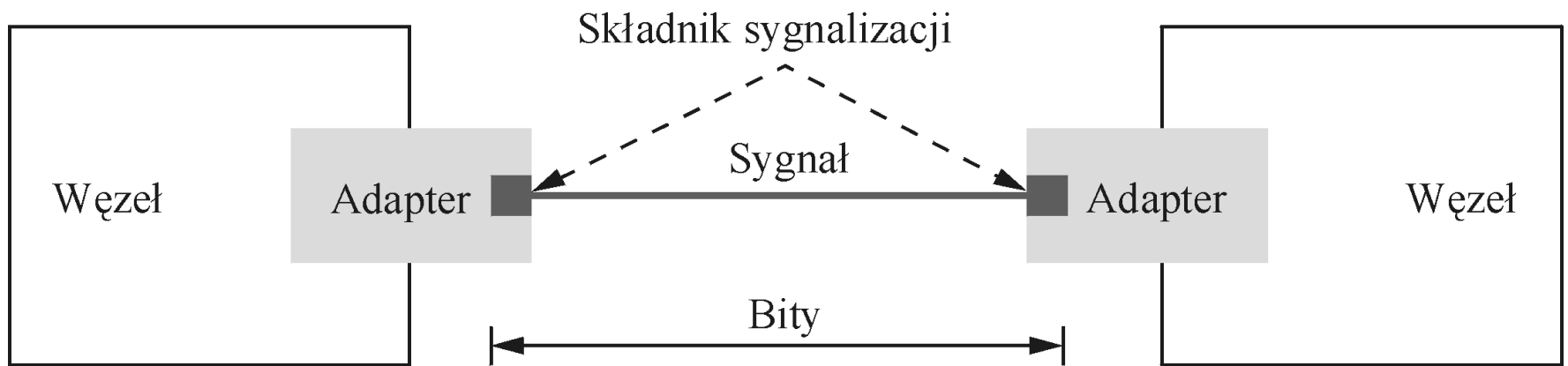
Kabel	Typowe szerokości pasma i odległości
Skęćka kategorii 5	10 – 100 Mb/s, 100 m
Koncentryczny 50 Ω (cienki)	10 – 100 Mb/s, 200 m
Koncentryczny 75 Ω (gruby)	10 – 100 Mb/s, 500 m
Światłowód wielopasmowy	100 Mb/s, 2 km
Światłowód jednopasmowy	100 – 2400 Mb/s, 40 km

szerokości pasma łączy cyfrowych świadczonych przez operatorów usług telekomunikacyjnych

usługa	szerokość pasma	
ISDN	64 kb/s	komutowane
T1	1,544 Mb/s	dzierżawione
T3	44,736 Mb/s	
STS-1	51,840 Mb/s	
STS-3	155,251 Mb/s	
STS-12	622,080 Mb/s	
STS-24	1,244 160 Gb/s	
STS-48	2,488 320 Gb/s	

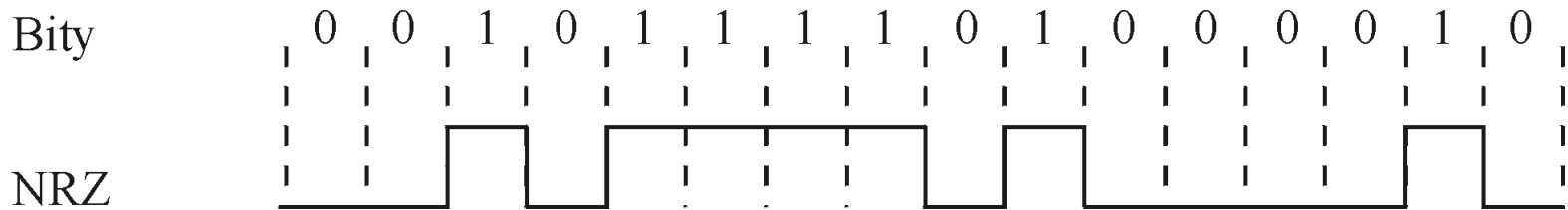
kodowanie

- *przepływ sygnałów między składnikami sygnalizacji* a bitów między adapterami



kodowanie NRZ

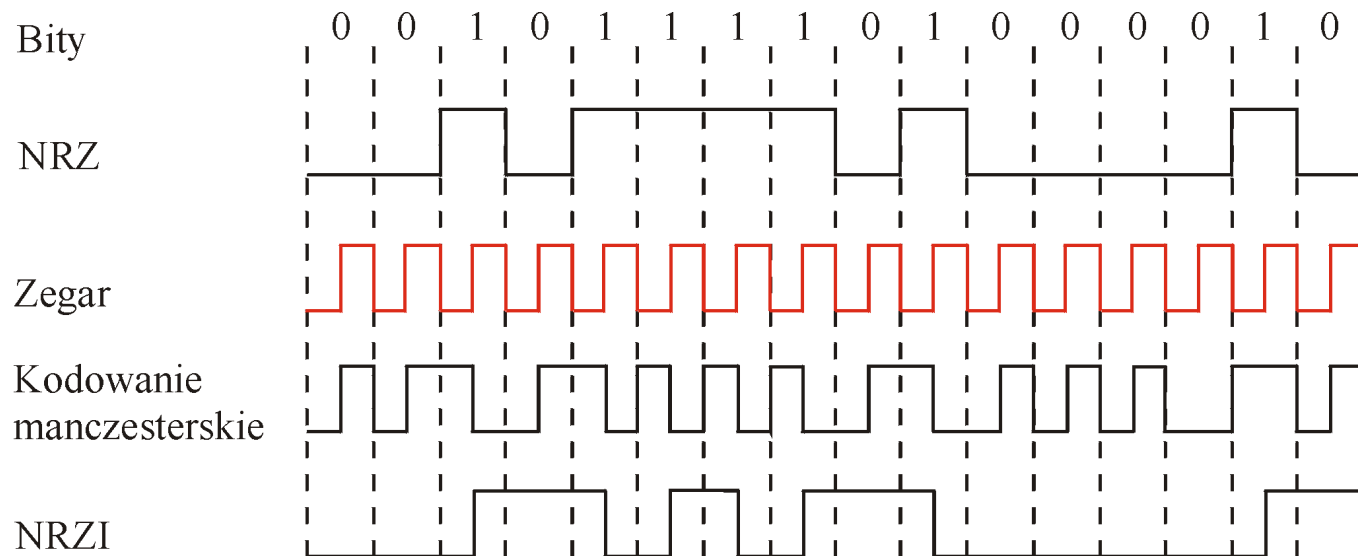
- odwzorowanie wartości danych *1 na sygnał wysoki* a wartości danych *0 na sygnał niski*






- *problemy:*
 - odbiornik nie rozróżnia *długiego ciągu zer* od *braku napięcia*
 - *długi ciąg jedynek* zmienia średnią wartość sygnału (*wędrówka poziomu odniesienia*)
 - brak zmian sygnału nie pozwala *odzyskać zegara*

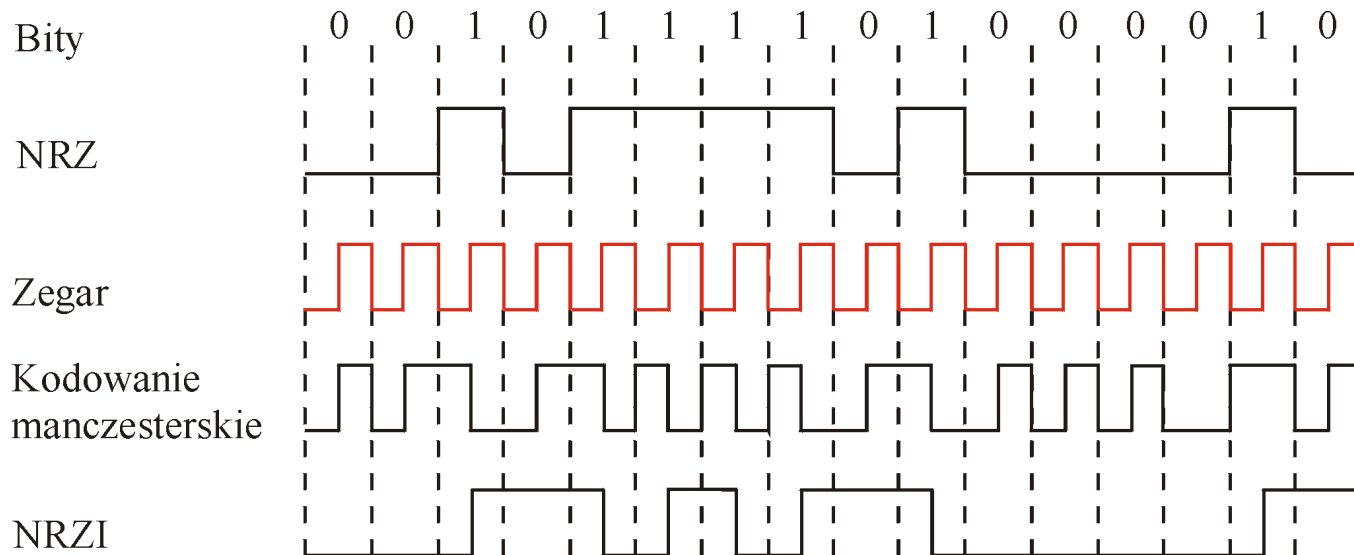
kodowanie NRZI

- podejmuje problem *dryfowania* zegara, spowodowanego przez długi okres bez zmiany sygnału
- *zmiana* poziomu sygnału dla 1, *ten sam* poziom dla 0
- rozwiązuje problem kolejnych jedynek, ale nie kolejnych zer !



kodowanie manczesterskie

- podejmuje problem kolejnych zer i jedynek z NRZ
- *scalenie zegara* z sygnałem za pomocą operacji różnicy symetrycznej XOR na danych kodowanych w schemacie NRZ i na zegarze: zegar , 0 , 1 
- *problem*: szybkość transmisji = $1/2 \times$ szybkości modulacji (szybkości zmiany sygnału), sprawność kodowania 50%



kodowanie 4B/5B

4-bitowy symbol danych	5-bitowy kod
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111

kodowanie 4B/5B

4-bitowy symbol danych	5-bitowy kod
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

kodowanie 4B/5B

- *dodatkowe bity w strumieniu bitów*
- ważne kody:
 - *na początku*: nie więcej niż jedno zero
 - *na końcu*: nie więcej niż dwa zera
- z pozostałych 16 kodów: 7 nieważne oraz:
 - 11111 - łącze bezczynne
 - 00000 - łącze martwe
 - 00100 - zatrzymanie
 - pozostałe 6: symbole sterujące
- *sprawność kodu: 80%*

prawo Shannona

- *twierdzenie Shannona* podaje górną granicę przepływności binarnej C łącza (w b/s) jako funkcję stosunku mocy sygnału do szumu (w dB)

$$C = B \log_2(1 + S/N)$$

gdzie: B szerokość pasma łącza (3300-300=3000 Hz)

S średnia moc sygnału, N średnia moc szumu

S/N stosunek sygnału do szumu, wyrażany w dB

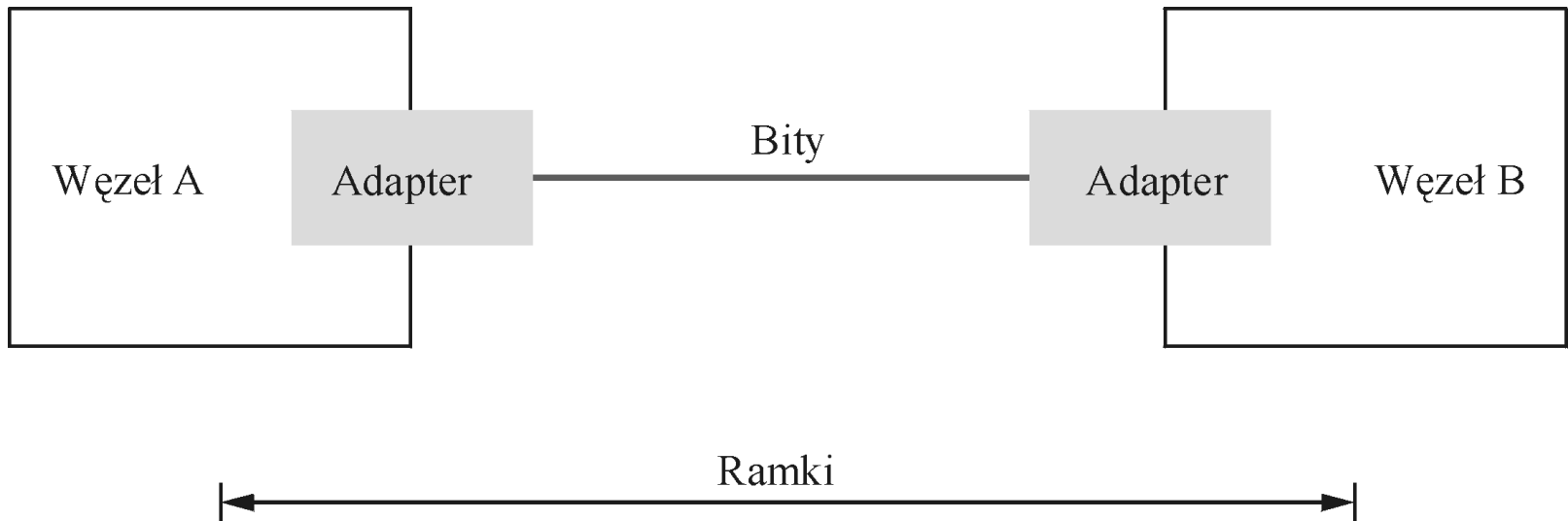
spełnia wzór: $\text{dB} = 10 \log_{10}(S/N)$ i zwykle wynosi 30, czyli: $S/N=1000$, stąd:

$$C = 3000 \log_2(1001) \sim 30 \text{ kb/s}$$

co dla modemu oznacza 28,8 kb/s

rozpoznawanie ramek

- określenie, *który zbiór bitów* stanowi ramkę



protokoły znakowe

- podejście *z wyróżnionym znakiem*
- podejście *zliczające znaki*

znaki sterujące:

- SYN: *znak synchronizacji*, początek ramki
- STX: *początek tekstu*
- ETX: *koniec tekstu*
- DLE: *opuszczenie łącza danych*

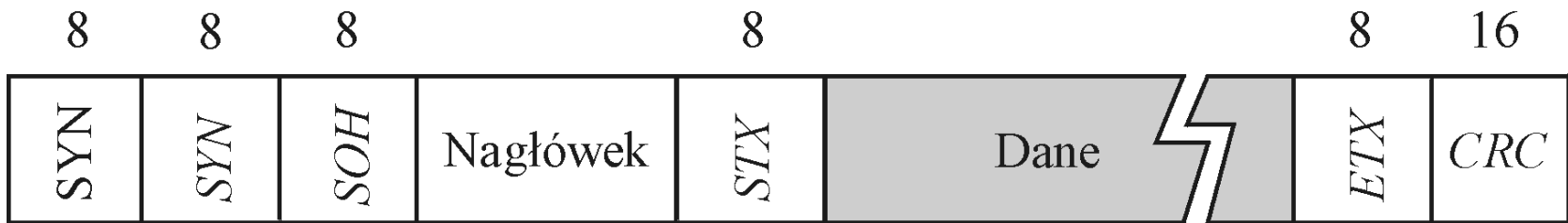
unikanie znaku ETX (poprzedzenie znakiem DLE)

- DLE/ETX, DLE/DLE

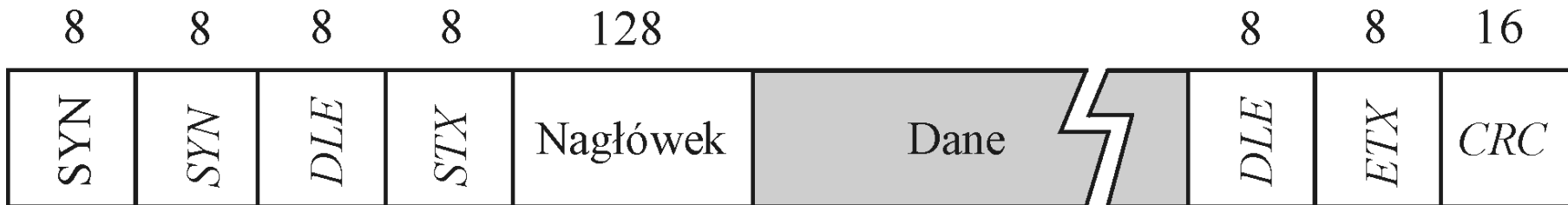
cykliczna kontrola nadmiarowa CRC

podjęcie z wyróżnionym znakiem

- format ramki w protokole **BISYNC (BSC)** firmy IBM:

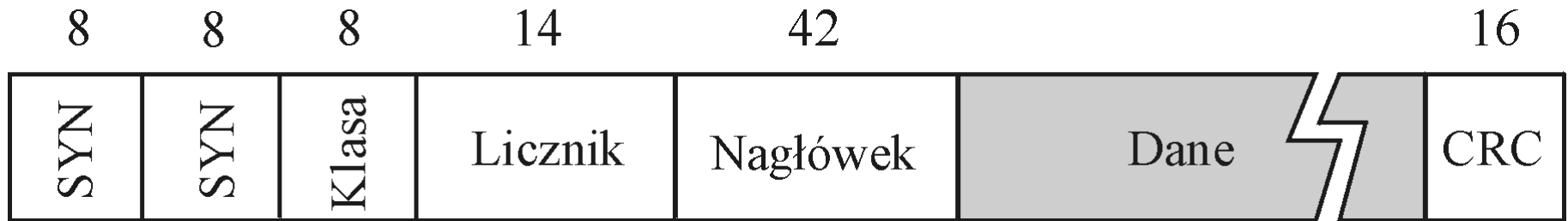


- format ramki w protokole **IMP-IMP** w sieci ARPANET:



podejście zliczające znaki

- format ramki w protokole **DDCMP**:



- protokół warstwy łącza w sieci DECNET firmy DEC

protokoły bitowe

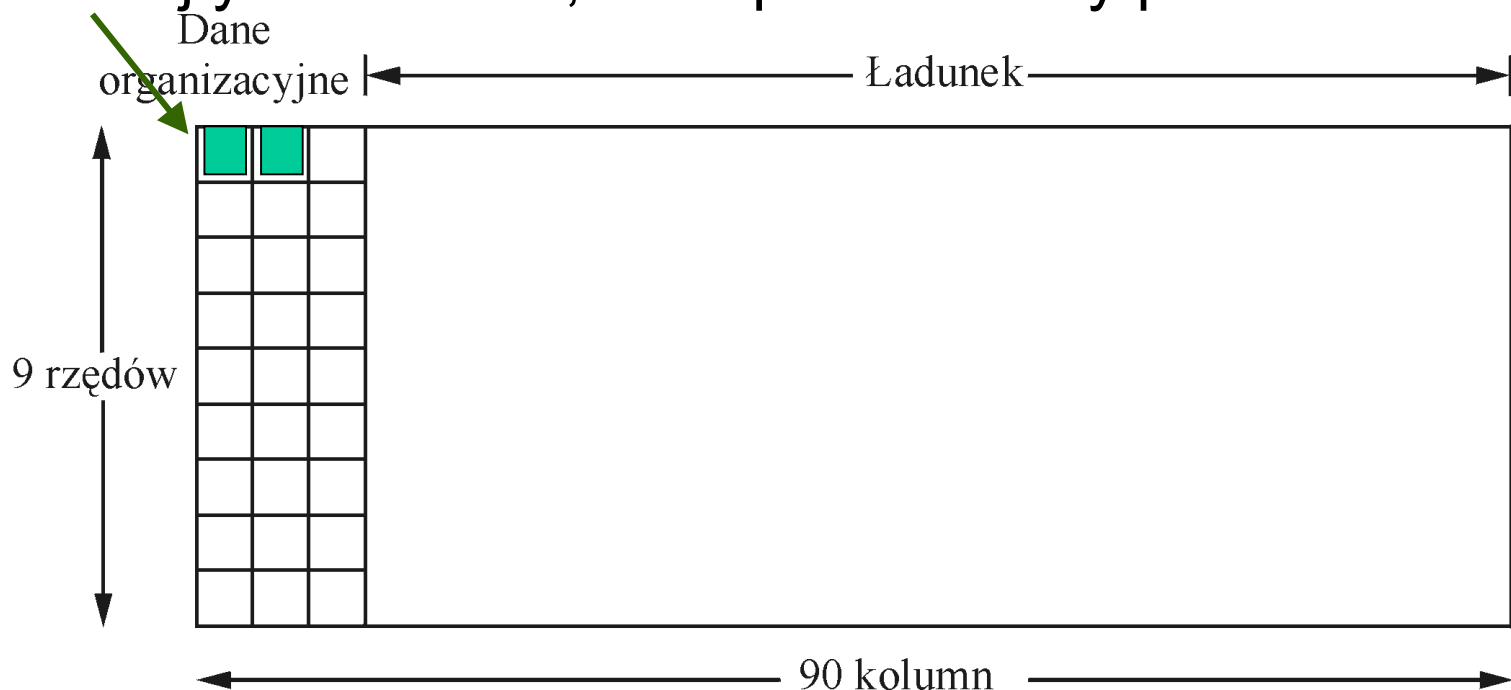
- *ramka* = zbiór bitów
- protokół **SDLC** firmy IBM, protokół **PPP ~ HDLC**
- protokół **HDLC** - norma w architekturze OSI:



- wyróżniona sekwencja **01111110**: początek i koniec ramki
- wstawianie bitów: 0 po każdym 5 jedynkach (**11111**) z wnętrza komunikatu
- **01111110** koniec ramki, **01111111** błąd (2 ramki stracone)

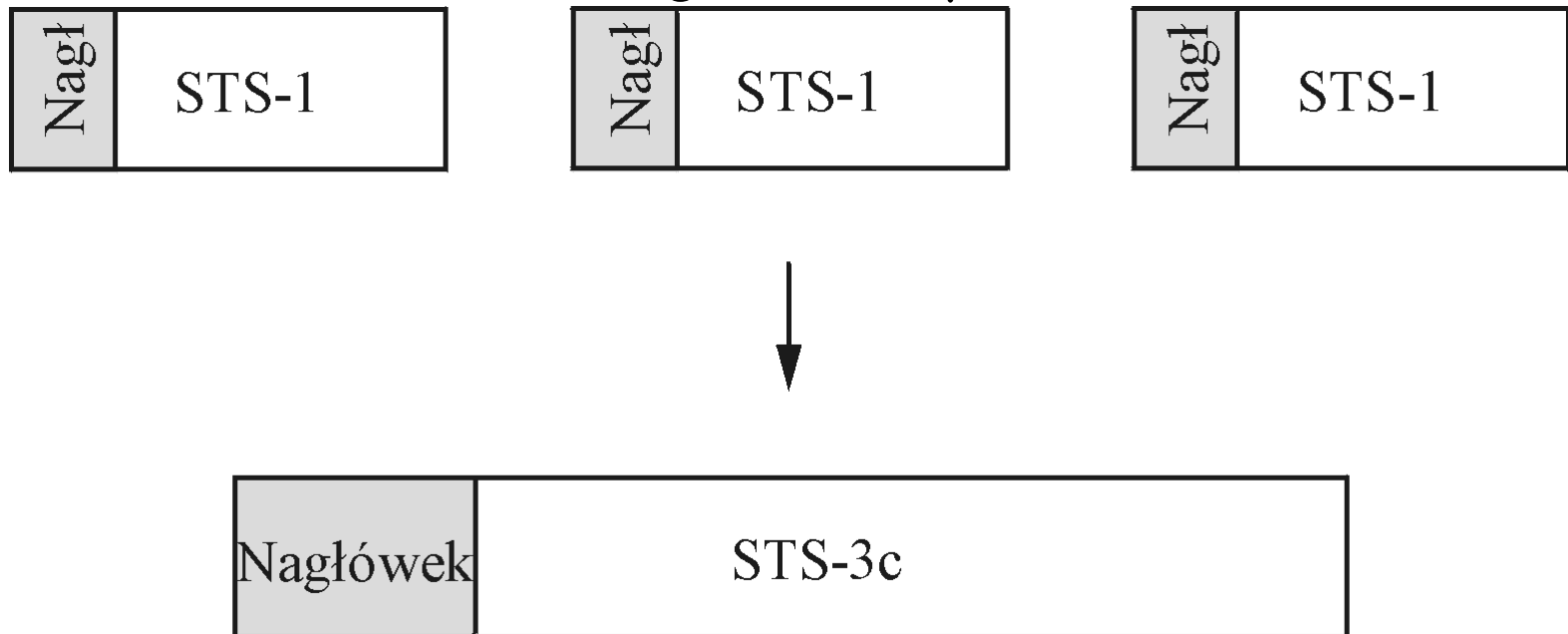
rozpoznawanie ramek za pomocą zegara

- *ramka* STS-1 w sieci SONET, 9 wierszy po 90 bajtów
- 3 bajty organizacyjne w wierszu
- 2 bajty wzór bitów, stale poszukiwany przez odbiornik



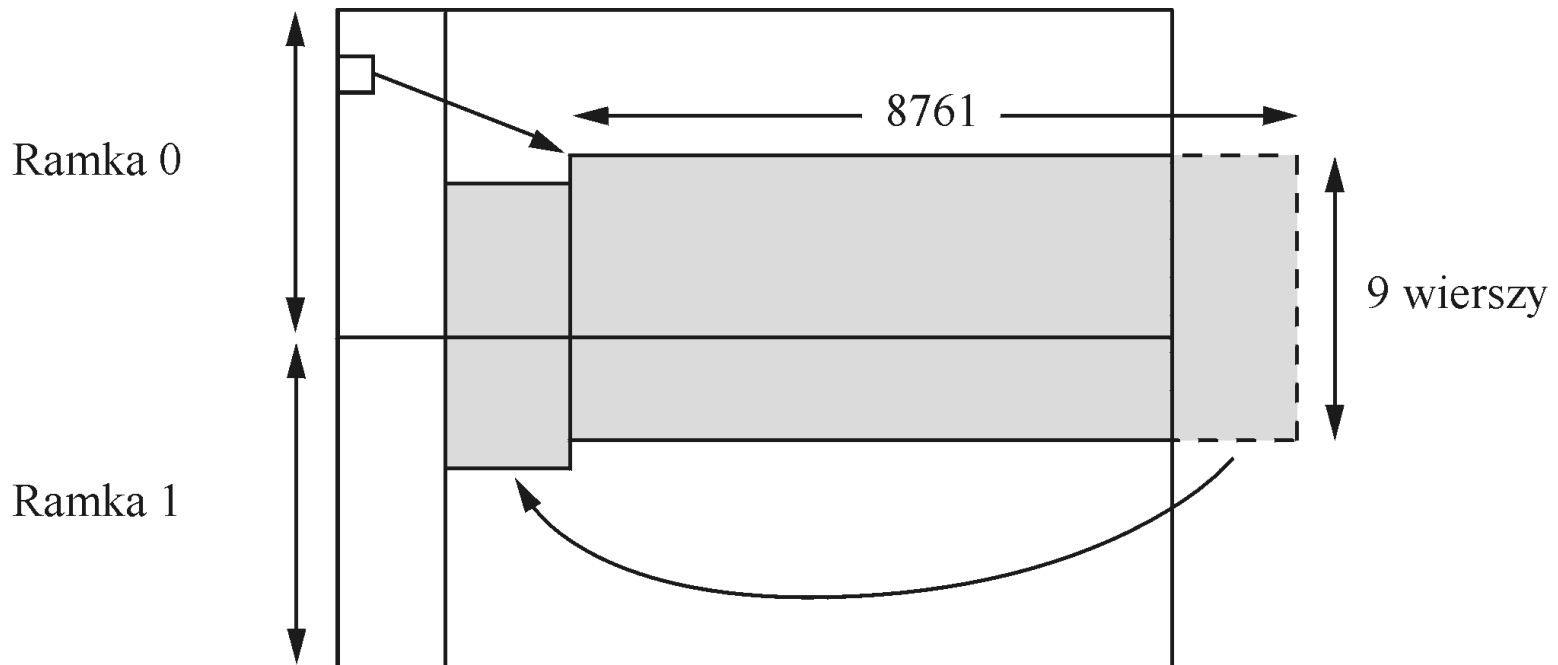
konkatenacja trzech ramek STS-1 w jedną ramkę STS-3c

- *obsługa multipleksacji łączy*: 3 ramki STS-1 (810 B, 51,84 Mb/s), mieszczą się w jednej ramce STS-3c (2430 B, 155,25 Mb/s = $3 \times 51,84$ Mb/s)
- każda ramka ma „długość” 125μs



ramki sieci SONET nie będące w fazie

- ładunek ramki *pływa* między granicami ramki
- ładunek przesunięty w prawo i *zawinięty* na początek



wykrywanie błędów

- *cykliczna kontrola nadmiarowa CRC:*
IMP-IMP, BISYNC (znaki EBCDIC), HDLC (CRC-CCITT), DDCMP, CSMA(CRC-32), pierścień ze znacznikiem (CRC-32), FDDI (CRC-32), ATM (CRC-8, CRC-10)
- *parzystość dwuwymiarowa:*
BISYNC (znaki ASCII)
- *sumy kontrolne:*
protokoły w Internecie

cykliczna kontrola nadmiarowa

- *dodanie informacji nadmiarowej do ramki*, pozwalające stwierdzić, czy do ramki wprowadzono błędy
- k nadmiarowych bitów dla n bitowego komunikatu, $k \ll n$
- *Ethernet*: CRC-32 dla ramki 12 000 bitów (1500 B)
- n bitowy komunikat jako wielomian stopnia $n-1$:
n.p. komunikat 10011010 odpowiada wielomianowi, gdzie wartość bitu to współczynnik wielomianu, czyli $M(x) = x^7 + x^4 + x^3 + x^1$
- k stopniem wielomianu dzielnika $C(x)$, założmy, że $C(x) = x^3 + x^2 + 1$, czyli $k=3$

cykliczna kontrola nadmiarowa

- *przesyłanie wielomianu* $P(x)$, podzielnego bez reszty przez $C(x)$
- gdy wystąpi błąd, to tak jakby do $P(x)$ dodano błędny człon $E(x)$
- gdy odbiorca komunikatu dzieli wielomian $P(x) + E(x)$ przez $C(x)$, *to wynik jest zerem gdy*: $E(x)$ jest zerem, (nie ma błędu), albo $E(x)$ jest podzielne przez $C(x)$ (co jest rzadkie, gdy dobrze dobrany $C(x)$)
- wynik dzielenia $P(x) + E(x)$ przez $C(x)$ równy 0 oznacza komunikat bezbłędny, wynik $\neq 0$ oznacza błąd w komunikacie

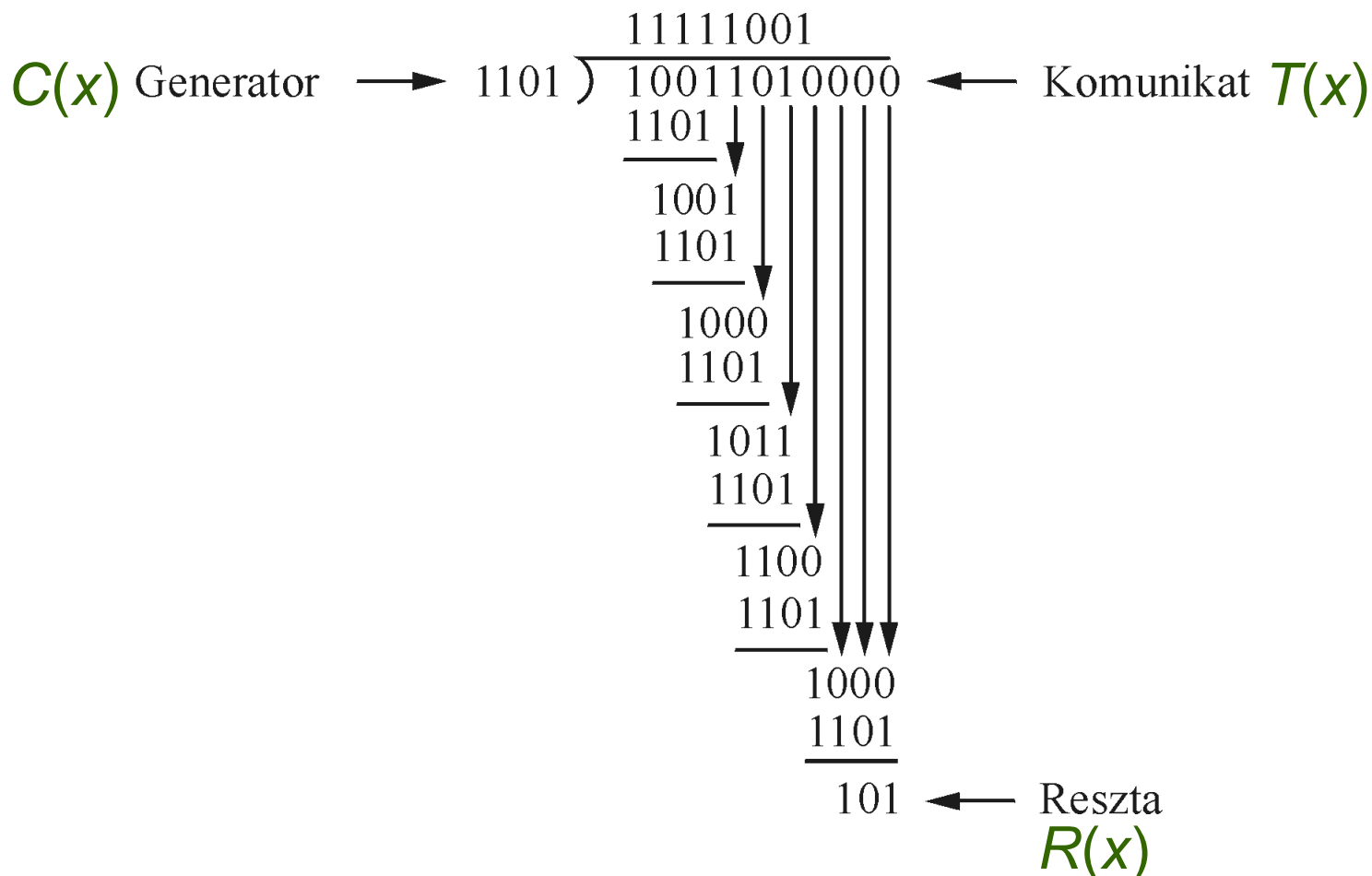
cykliczna kontrola nadmiarowa

- *przykład*: chcemy utworzyć wielomian $P(x)$ do nadania, który jest wyliczony z oryginalnego komunikatu $M(x)$, jest k bitów dłuższy od $M(x)$ i jest dokładnie podzielny przez $C(x)$:

$$M(x) = x^7 + x^4 + x^3 + x^1, k=3, C(x) = x^3 + x^2 + 1$$

- *mnożymy* $M(x)$ przez x^k , czyli x^3 , czyli dodajemy 3 zera na końcu komunikatu: czyli z 10011010 otrzymujemy 10011010000, nazywamy go $T(x)$
- *dzielimy* $T(x)$ przez $C(x)$, czyli przez 1101, i znajdujemy resztę $R(x)$, wynoszącą 101
- *odejmujemy* resztę $R(x)$ od $T(x)$:
- $10011010000 \text{ XOR } 101 = 10011010101 = P(x)$

dzielenie wielomianów $T(x)$ przez $C(x)$ za pomocą dzielenia pisemnego

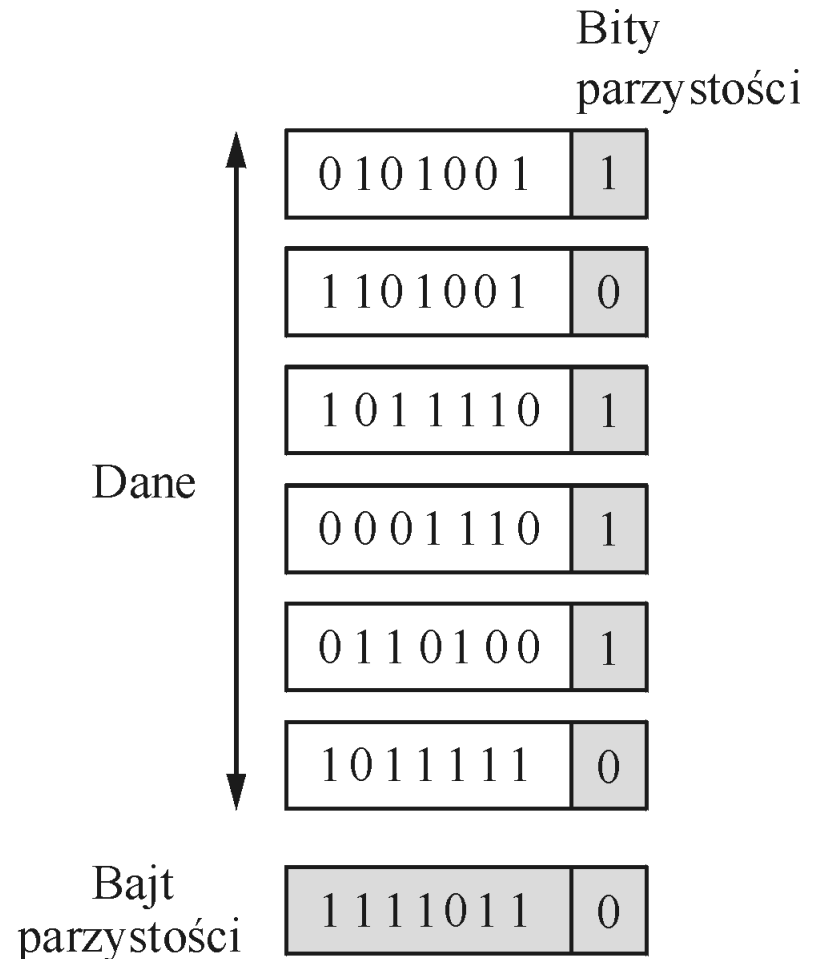


powszechnie stosowane wielomiany CRC

CRC	C(x)
CRC-8	$x^8 + x^2 + x^1 + 1$
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^1 + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-CCITT	$x^{16} + x^{12} + x^5 + 1$
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} +$ $+ x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

parzystość dwuwymiarowa

- tutaj *parzysta parzystość* dla każdego bajtu (*w poziomie, bit*) oraz *parzysta parzystość* dla pozycji każdego bitu przez wszystkie bajty (*w pionie, bajt*)
- wykrywa wszystkie błędy na* jednym, dwóch, trzech i *najwyżej czterech bitach*



suma kontrolna

- *w nadajniku*: dodajemy wszystkie nadawane słowa i nadajemy wynik sumowania (*sumę kontrolną*)
- *w odbiorniku*: dodajemy wszystkie otrzymane słowa i porównujemy z sumą kontrolną otrzymaną od nadajnika
- gdy wyniki do siebie *nie pasują*, to dane i/lub suma kontrolna są uszkodzone
- podejście stosowane w Internecie

niezawodna transmisja

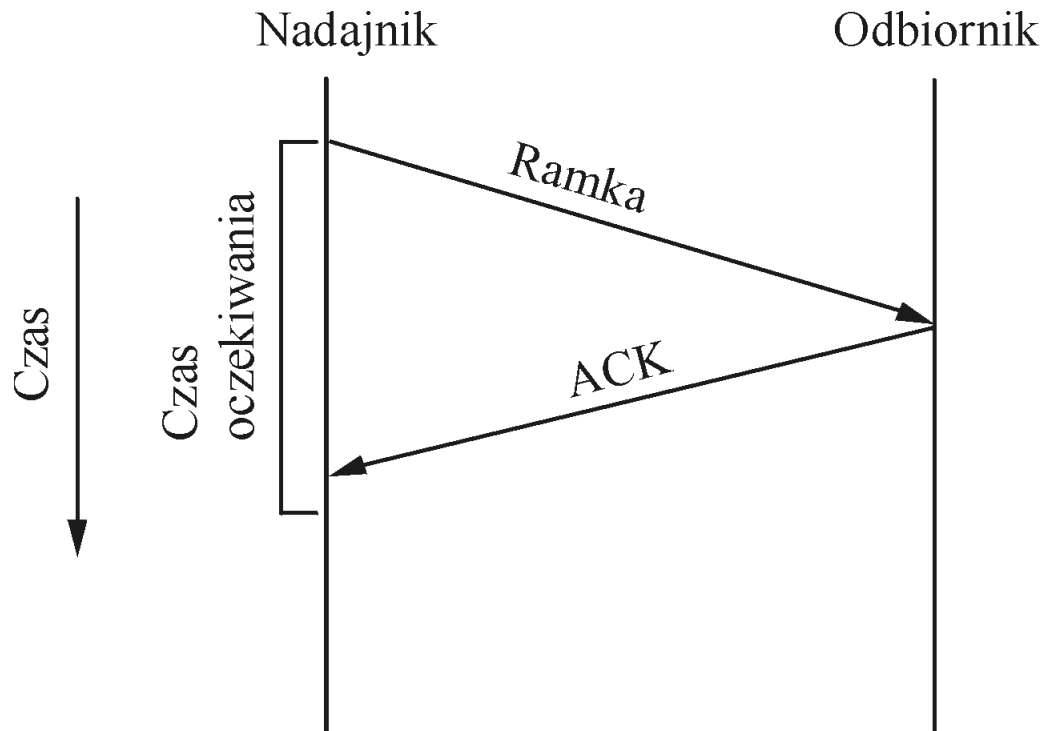
- *nie wszystkie błędy udaje się naprawić* - ramki takie są odrzucane
- powstaje *problem odzyskania utraconych ramek*
- rozwiązanie: kombinacja *mechanizmów potwierdzenia i czasu oczekiwania*
- *brak odbioru potwierdzenia przez czas oczekiwania powoduje retransmisję ramki*
- ogólna strategia: *automatyczne żądanie powtórzenia*
ARQ → algorytmy ARQ

algorytmy ARQ: algorytm stój i czekaj

- po nadaniu ramki, nadajnik *czeka na potwierdzenie* (ACK), przed nadaniem następnej
- jeżeli potwierdzenie nie przyjdzie po wyznaczonym czasie, *retransmituje oryginalną ramkę*

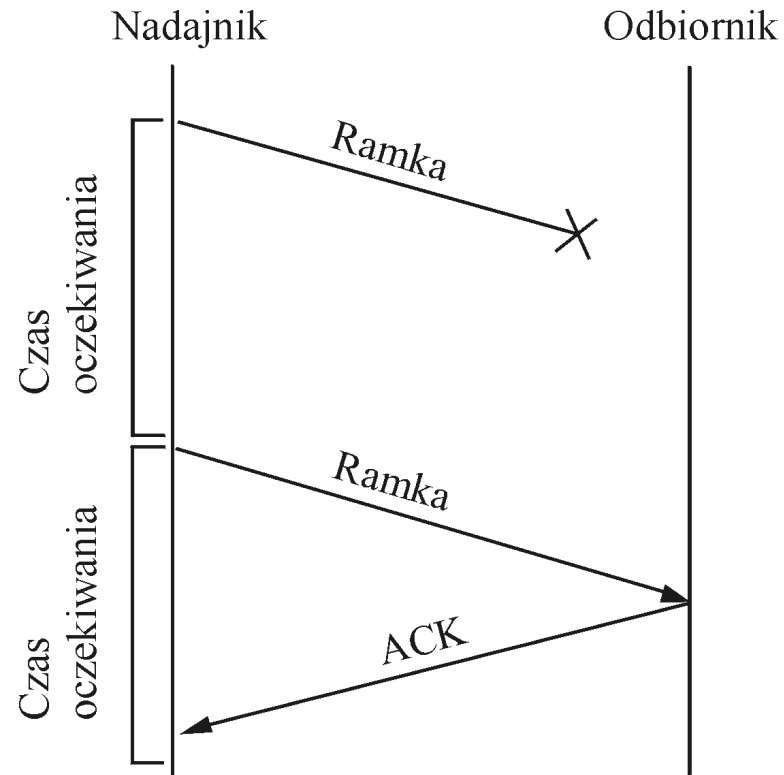
algorytm stój i czekaj

- ACK odebrane *przed upływem* czasu oczekiwania



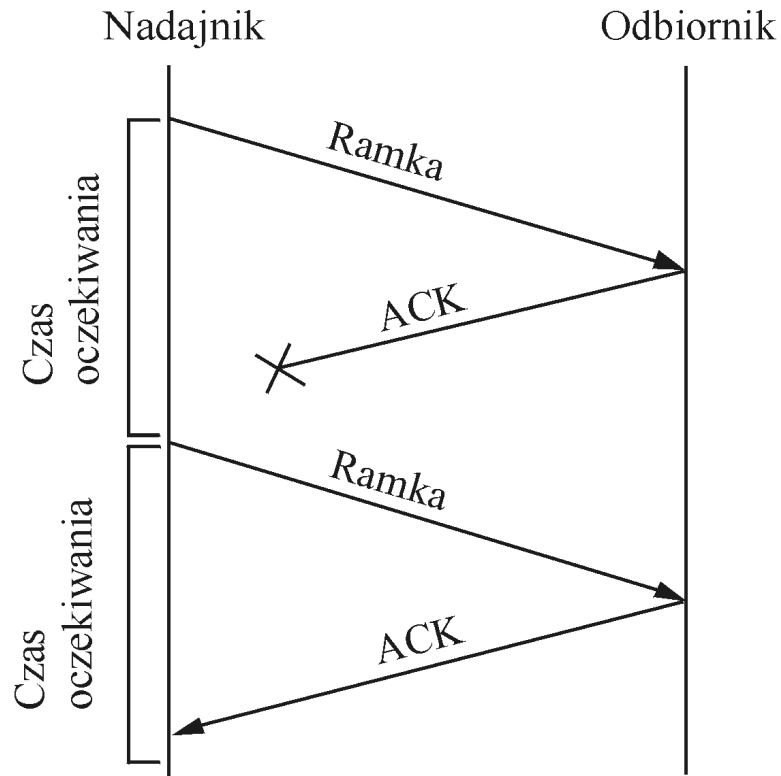
algorytm stój i czekaj

- strata oryginalnej ramki



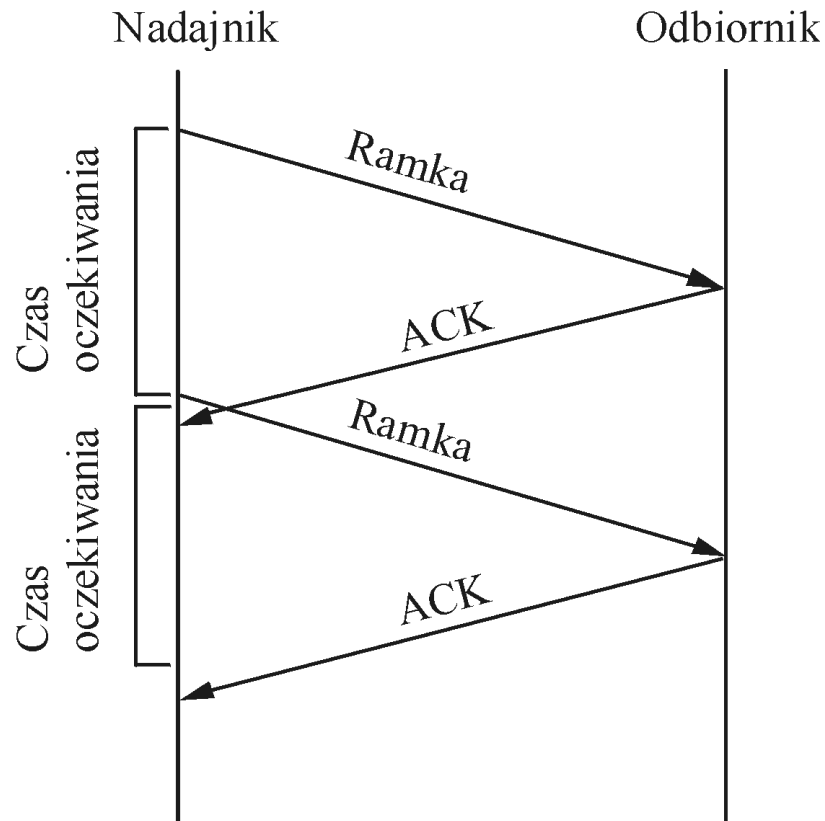
algorytm stój i czekaj

- strata ACK



algorytm stój i czekaj

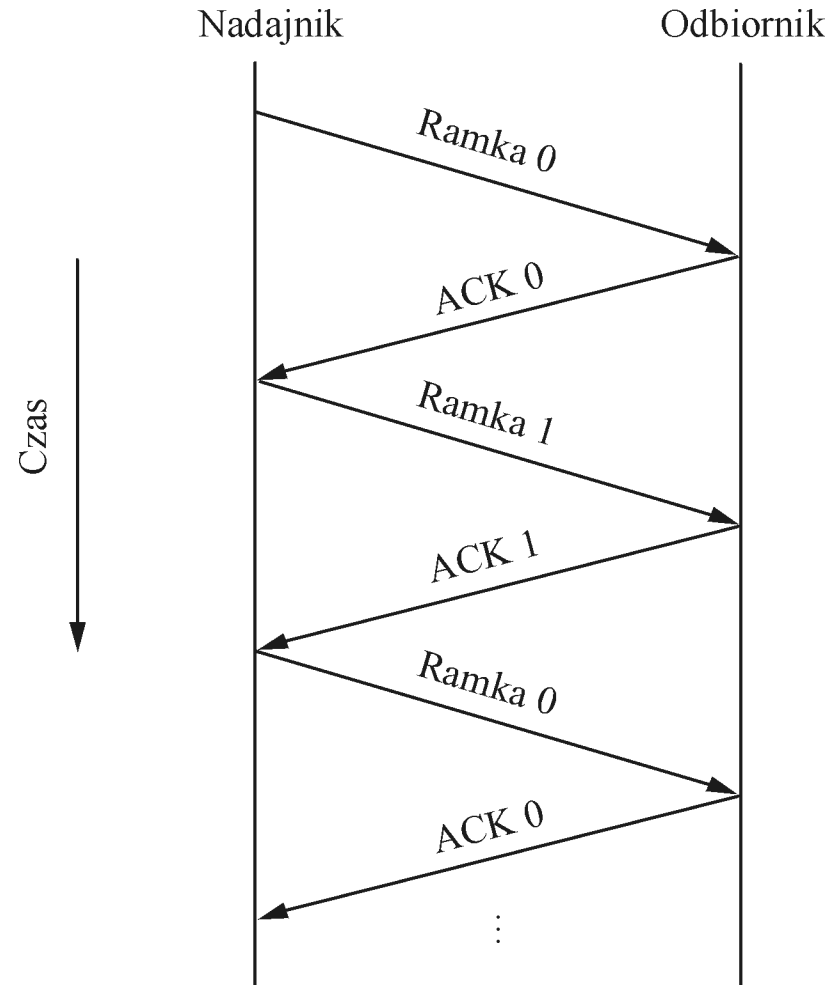
- sygnał czasu oczekiwania *przedwczesny*



algorytm stój i czekaj

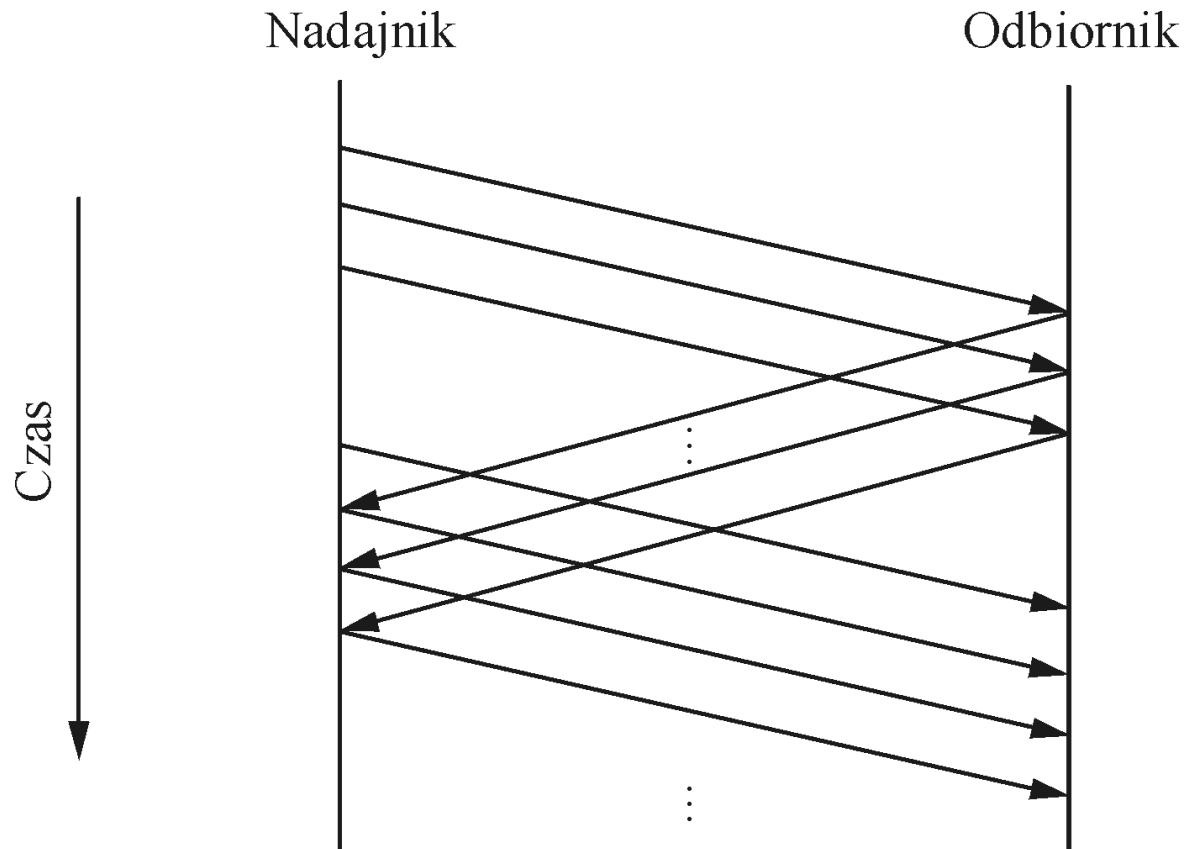
- w dwóch ostatnich przypadkach odbiornik sądzi, *że retransmitowana ramka jest już następną ramką*, ponieważ poprawnie odebrał i potwierdził pierwszą ramkę
- dlatego wprowadzono 1 bitowy numer sekwencyjny, przyjmujący na przemian wartości 0 i 1, w ramach i w ACK
- odbiornik może wtedy *zignorować duplikaty* ramek, ale nadal je potwierdza
- *wada algorytmu stój i czekaj*: nadajnik wysyła *tylko jedną* ramkę w danej chwili, znacznie poniżej przepustowości łącza

algorytm stój i czekaj z 1-bitowym numerem sekwencyjnym



algorytmy ARQ: przesuwne okno

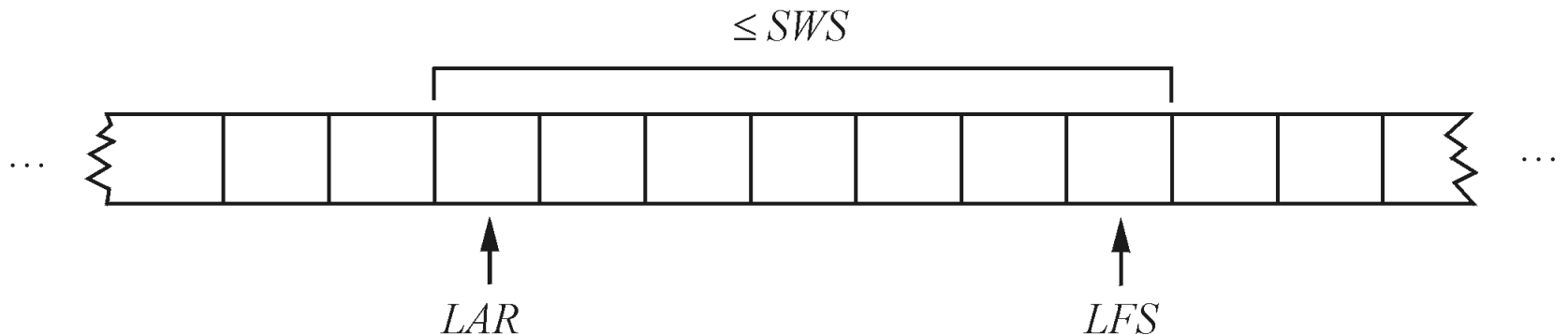
- nadajnik gotowy do przesłania n.p. dziewiątej ramki, w chwili przyjęcia potwierdzenia na pierwszą ramkę



przesuwne okno - nadajnik

- *zmienne w nadajniku*: numer sekwencyjny **SeqNum**, rozmiar okna nadawczego **SWS**, numer sekwencyjny ostatniego odebranego potwierdzenia **LAR**, numer sekwencyjny ostatniej nadanej ramki **LFS**
- nadajnik utrzymuje zależność między zmiennymi:

$$LFS - LAR \leq SWS$$



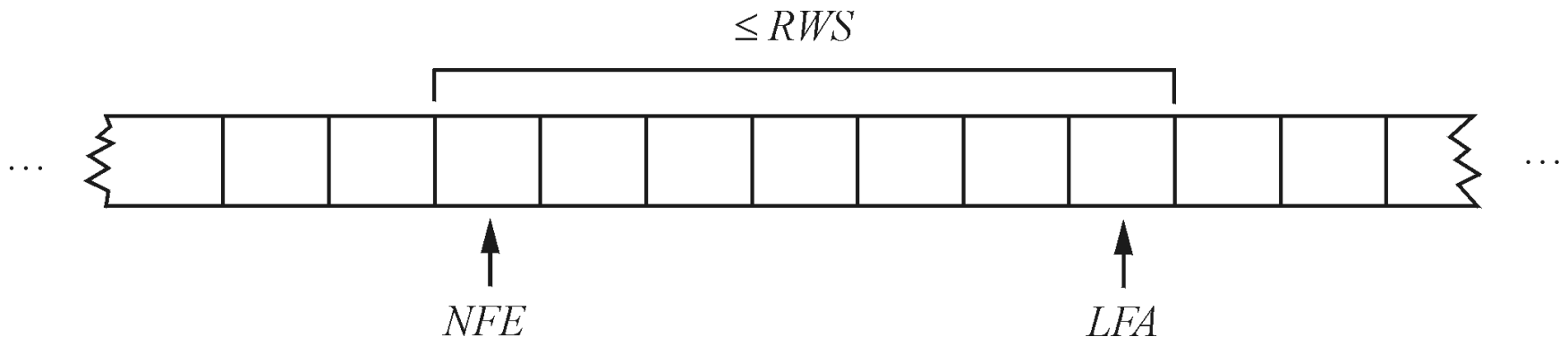
przesuwne okno - nadajnik

- *działanie algorytmu w nadajniku:*
- nadajnik przydziela zegar każdej nadanej ramce
- kiedy przychodzi potwierdzenie, nadajnik przesuwają **LAR** w prawo, pozwalając nadać następną ramkę
- kiedy zegar upłynął przed odebraniem potwierdzenia, nadajnik retransmituje ramkę
- nadajnik zdolny do buforowania ramek w ilości do rozmiaru okna nadawczego **SWS**

przesuwne okno - odbiornik

- *zmienne w odbiorniku*: rozmiar okna odbiorczego RWS, numer sekwencyjny ostatniej akceptowalnej ramki **LFA**, numer sekwencyjny następnej oczekiwanej ramki **NFE**
- odbiornik utrzymuje zależność między zmiennymi:

$$LFA - NFE + 1 \leq RWS$$



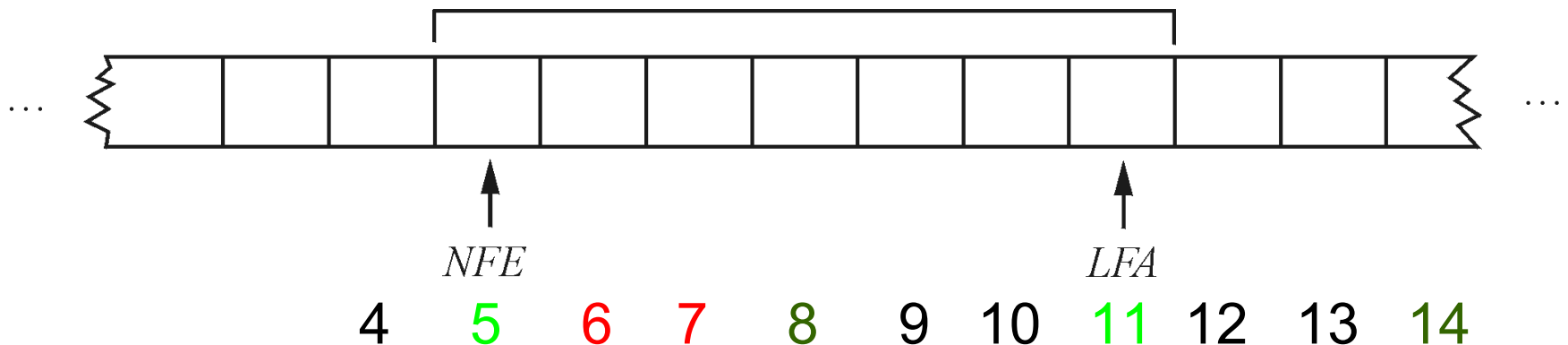
przesuwne okno - odbiornik

- *działanie algorytmu w odbiorniku:*
- ramka przychodząca *akceptowana* gdy jest w oknie odbiornika: $NFE \leq SeqNum \leq LFA$, inaczej *odrzucona*
- SeqNumtoAck to *największy niepotwierdzony numer sekwencyjny*, taki że wszystkie ramki o numerach mniejszych od niego są już odebrane
- odbiornik potwierdza odbiór ramki z SeqNumtoAck (*potwierdzenie kumulacyjne*)
- odbiornik ustawia: $NFE = SeqNumtoAck + 1$ oraz $LFA = SeqNumtoAck + RWS$

przesuwne okno - odbiornik

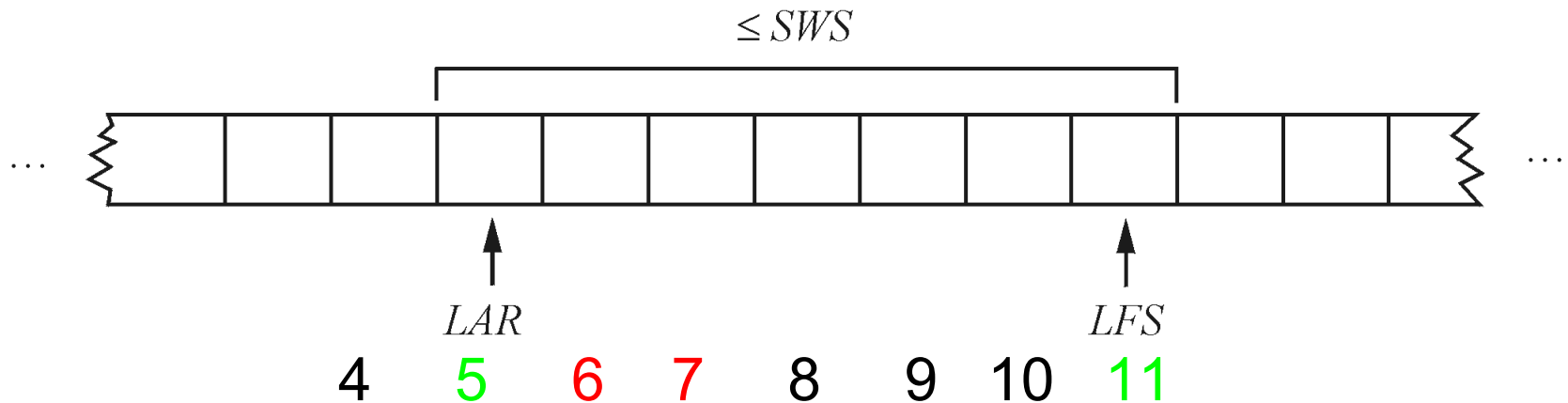
przykład

- niech $NFE=5$ i $RWS=7$, to implikuje, że $LFA=11$
- gdyby ramki 6 i 7 przyszły, to są buforowane (są *poza kolejnością*, gdyż ramka 5 jeszcze nie dotarła)
- gdy przychodzi ramka 5 , odbiornik potwierdza ramkę 7 , podwyższa NFE na 8 i ustawia LFA na 14
 $\leq RWS$



przesuwne okno - nadajnik warianty

- jeżeli ramka 5 jest rzeczywiście stracona, nadajnik po upływie czasu oczekiwania nada ją ponownie (cofa się o 3 ramki, bo nadał już 6 i 7)
- nadajnik nie jest w stanie powiększyć swojego okna, dopóki ramka 5 nie zostanie potwierdzona



przesuwne okno - odbiornik - warianty

- odbiornik może nadać *negatywne potwierdzenie* (NAK) dla ramki 5 kiedy przyszła ramka 6 (nie ma potrzeby, jest mechanizm czasu oczekiwania)
- odbiornik może nadać dodatkowe potwierdzenie ACK na ramkę 4, jak przychodzą ramki 6 i 7 (wskazówka dla nadajnika o stracie ramki 5)
- odbiornik potwierdza ramki 6 i 7, a nie tylko ostatnią odebraną w kolejności ramkę 4 - *selektywne potwierdzenie*

dostęp do łącza wielodostępnego

- CSMA/CD (Ethernet)
- pierścień ze znacznikiem (FDDI)

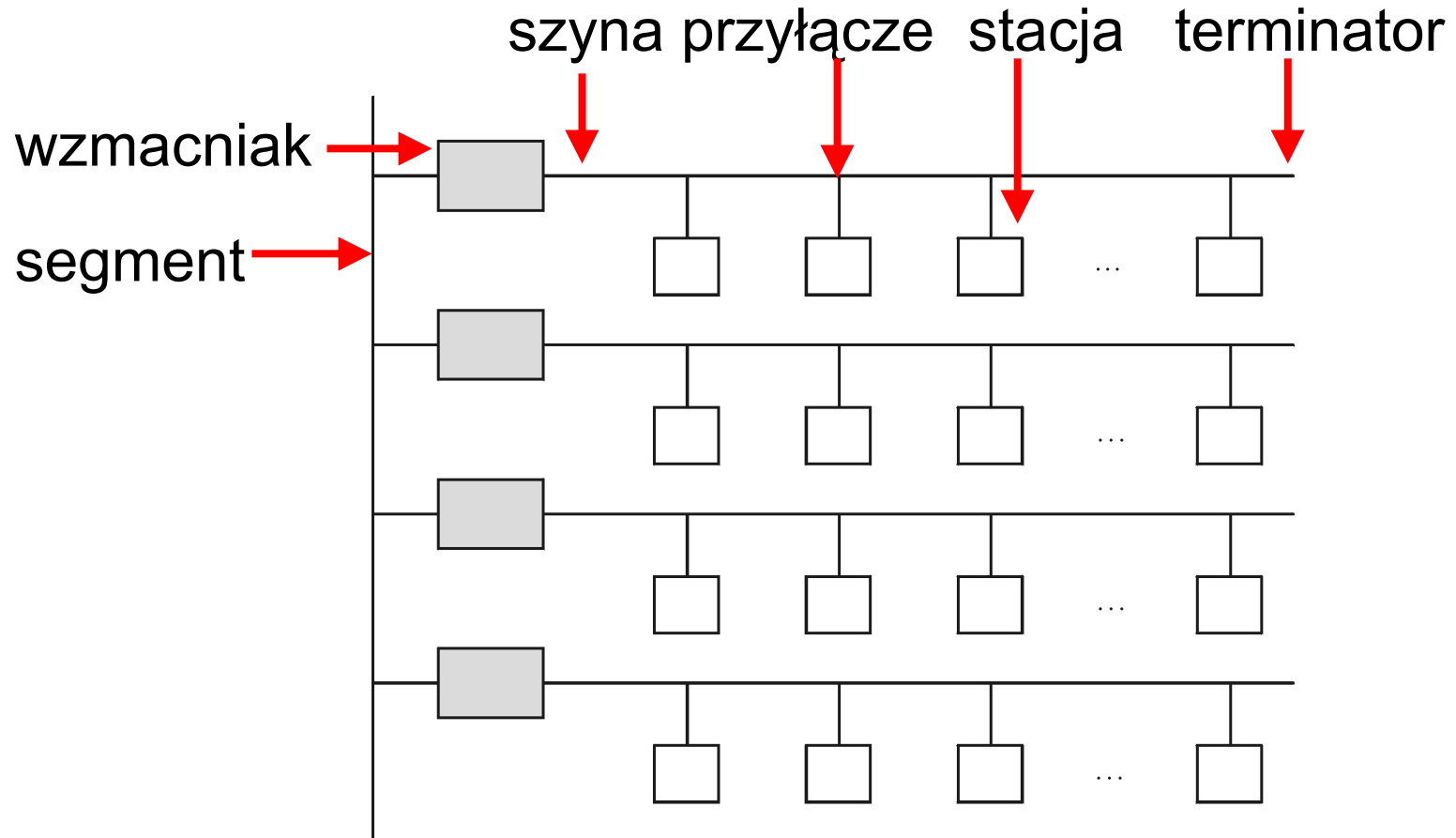
CSMA/CD (Ethernet)

- CSMA/CD *metoda dostępu do łącza sieci z badaniem stanu kanału i wykrywaniem kolizji*
- szybkość transmisji 10Mb/s
- *sieć wielodostępna*: zbiór węzłów nadaje i odbiera ramki na łączu współdzielonym (szyna, wiele stacji)
- *badanie stanu kanału*: wszystkie węzły są w stanie rozróżnić między łączem *nieczynnym* a *zajętym*
- *wykrywanie kolizji*: węzeł *nastuchuje* podczas nadawania i może *wykryć*, kiedy nadana przez niego ramka interferuje (koliduje) z ramką nadawaną przez inny węzeł
- *geneza*: ALOHA

Ethernet - cechy fizyczne

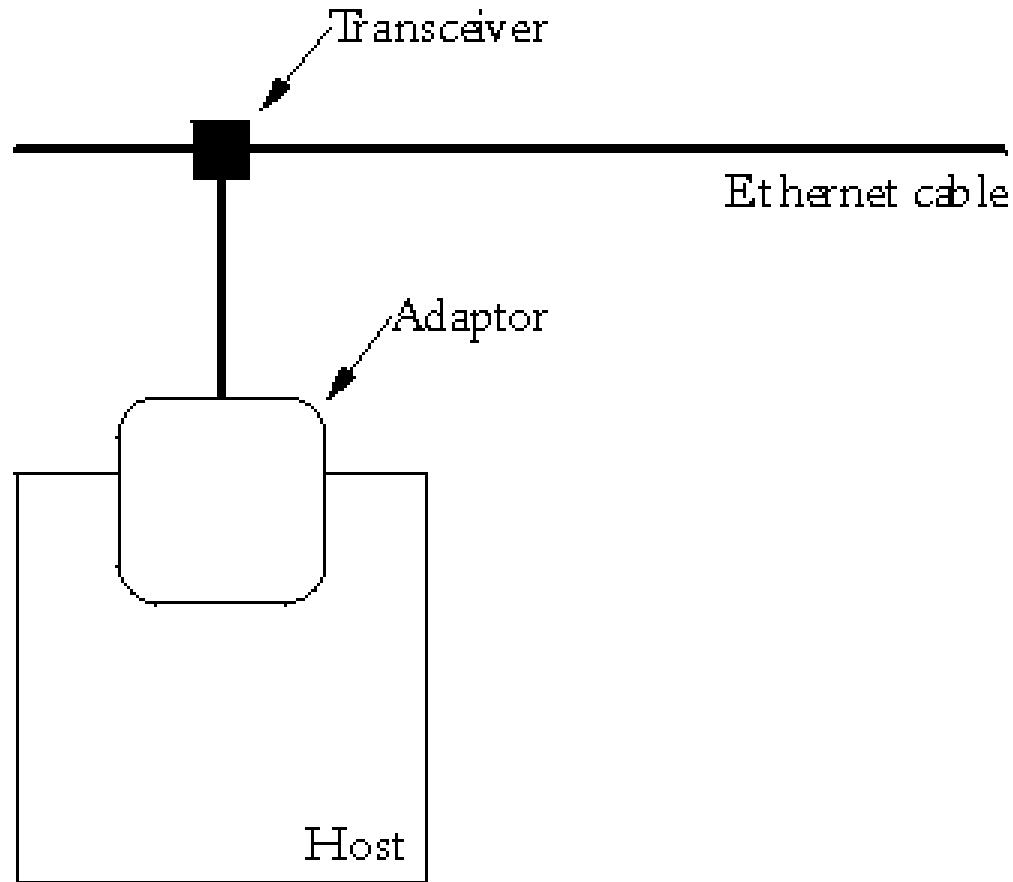
- *segment Ethernetu*: kabel koncentryczny 500m, impedancja 50Ω , 10Base5, kodowanie manczesterskie
- komputer dołączony do segmentu przez *przylącze*, min. co 2,5 m, maks. 1024 komputery
- między adapterem Ethernetu w komputerze a przylączem *nadajnik-odbiornik* (transceiver): nasłuch łączy i przekazywanie sygnału
- *wzmacniak* do łączenia segmentów, maks. 4 wzmacniaki między komputerami, stąd zasięg 2500m
- *rozgłaszanie sygnału* - terminator na końcu segmentu absorbuje sygnały, unikanie odbić i interferencji

przykładowa konfiguracja Ethernetu



nadajnik/odbiornik w Ethernetie

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



wynalazca Ethernetu

Robert M. Metcalfe (1973)

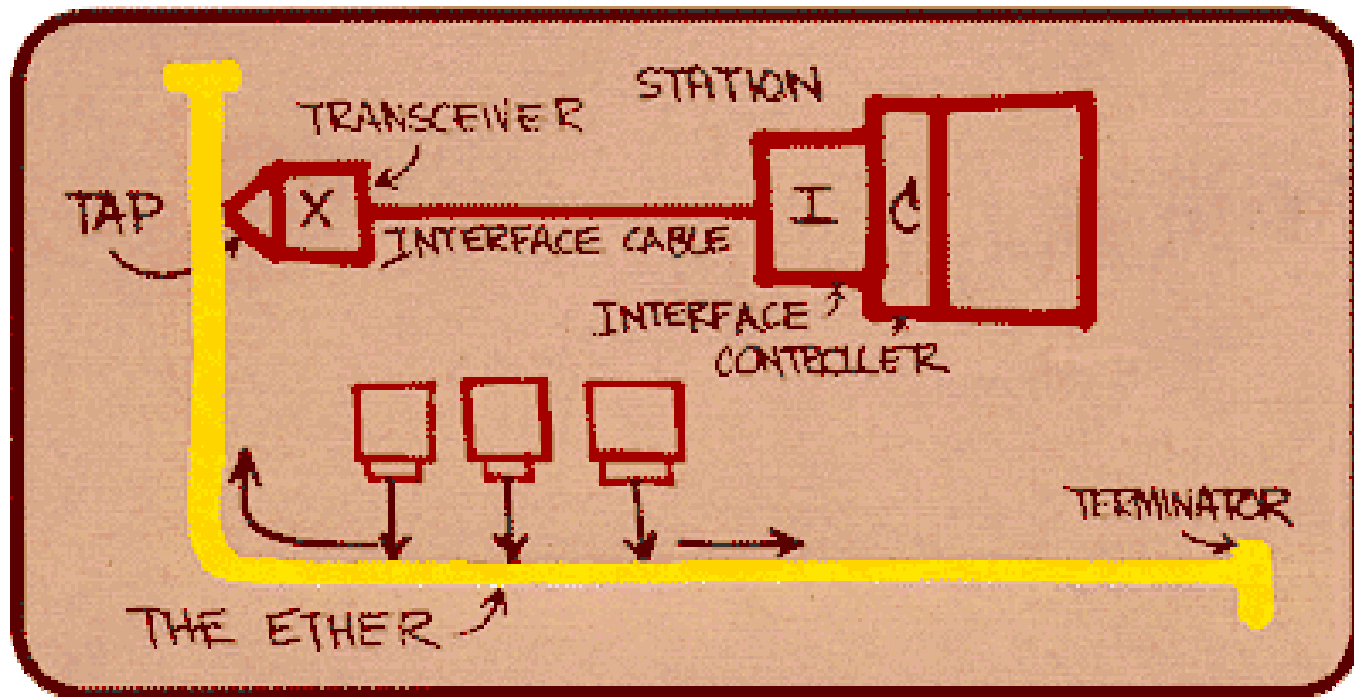
<http://www.ibiblio.org/pioneers>



rysunek Ethernetu (Bob Metcalfe)

<http://www.ots.utexas.edu/ethernet>


Charles Spurgeon's Ethernet Web site



protokół dostępu

- implementowany *sprzętowo* w adapterze sieci
- *format ramki* (podana ilość bitów każdego pola):

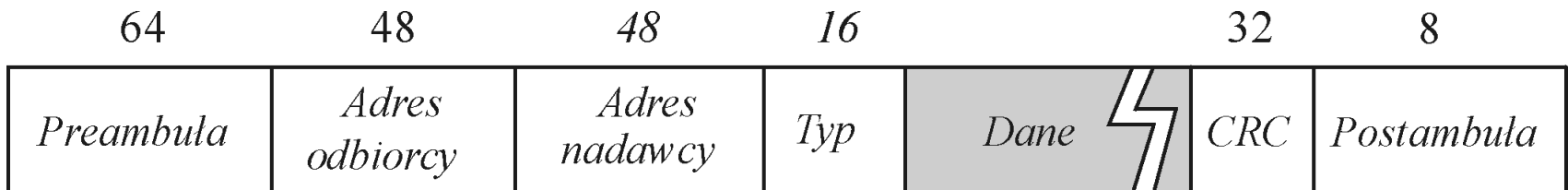
64 48 48 16 32 8

<i>Preambula</i>	<i>Adres odbiorcy</i>	<i>Adres nadawcy</i>	<i>Typ</i>	<i>Dane</i> 	<i>CRC</i>	<i>Postambula</i>
------------------	-----------------------	----------------------	------------	-------------------------------------------------------------------------------------------------	------------	-------------------

- *preambuła*: naprzemiennie 0 i 1
- *typ* pakietu: klucz *demultipleksacji* (do którego protokołu wyższego poziomu dostarczyć ramkę)
- *dane*: do 1500 bajtów
- CRC-32

protokół dostępu

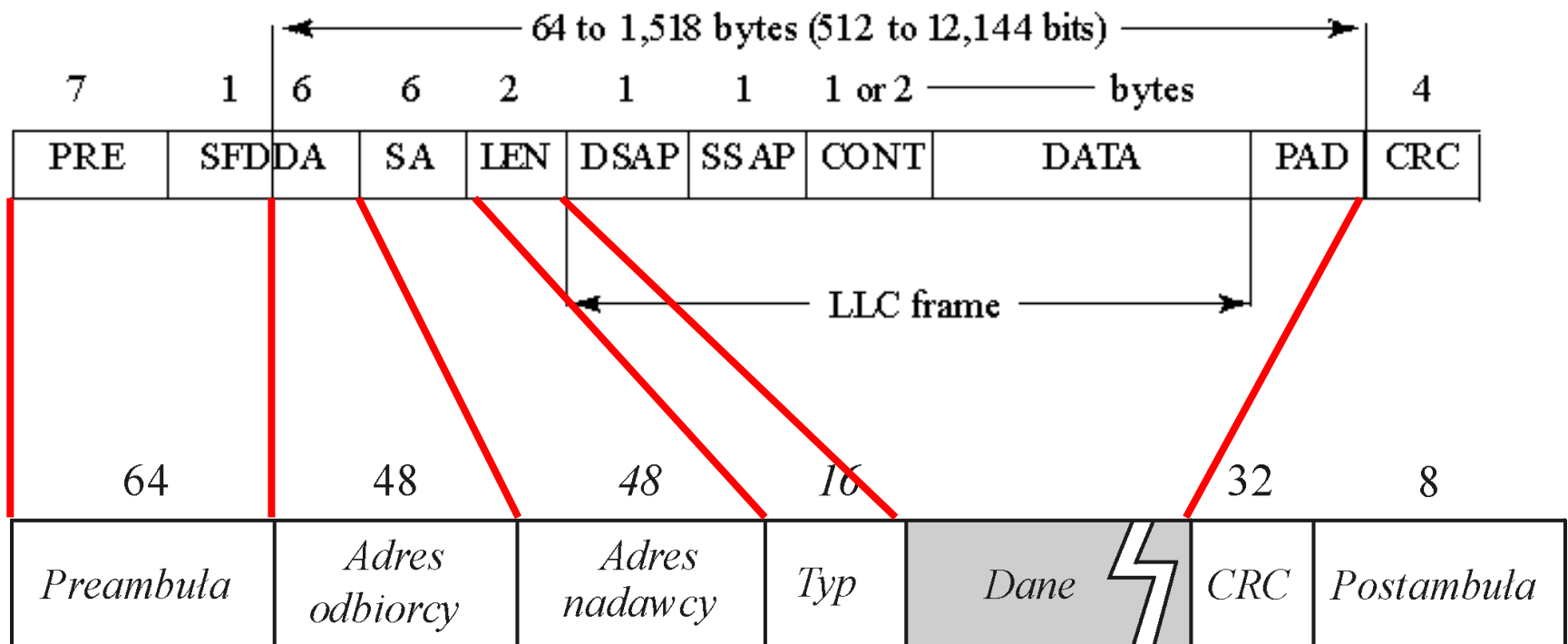
- Ethernet jest protokołem *bitowym*!
- *z punktu widzenia komputera*, ramka Ethernetu ma 14-bajtowy nagłówek: dwa adresy po 6 bajtów i pole typu 2-bajtowe
- adapter nadający *dołącza* preambułę, CRC i postambułę, adapter odbierający *usuwa je*



normy, ramka Ethernetu a ramka LLC

Walrand&Varaiya, High-Performance Communication Networks,
http://www.mkp.com/books_catalog/catalog.asp?ISBN=1-55860-574-6

- norma IEEE 802.3



- norma Ethernetu (Xerox, DEC, Intel 1978)

adresy w Ethernetie

- *adres ethernetowy*, zapisany w pamięci ROM adaptera, *unikalny w skali światowej*, gdyż zawiera przedrostek przydzielony producentowi karty sieciowej
- 6 bajtów zapisanych parami cyfr heksadecymalnych, n.p. 8:0:2b:e4:b1:2, odpowiada adresowi Ethernetu:

00001000 00000000 00101011
11100100 10110001 00000010

- adapter przekazuje komputerowi ramki, w których rozpoznał swój adres - *adres jednostkowy*
- *adres rozgłoszeniowy* ff:ff:ff:ff:ff:ff
- *adres grupowy*, pierwszy bit 1 - ale nie ff:ff:ff:ff:ff:ff

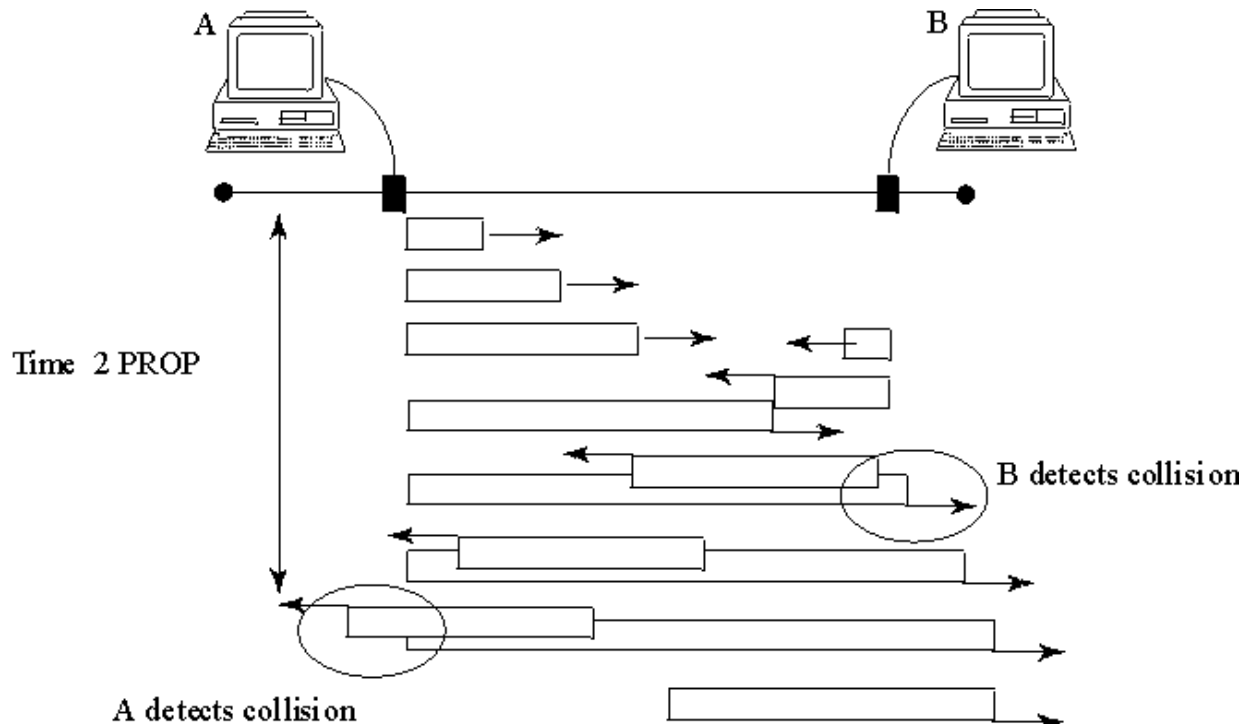
algorytm Ethernetu po stronie nadajnika

- nadajnik ma ramkę do nadania *a linia wolna* - *nadaje*
- nadajnik ma ramkę do nadania *a linia zajęta* - *czeka na zwolnienie linii i natychmiast nadaje*
- gdy więcej niż dwa adaptory nadają ramki jednocześnie - *kolizja ramek w sieci*
- nadajnik stwierdza kolizję - nadaje 64-bitową preambułę i 32-bitową *sekwencję zagłuszającą*
- by być pewnym, że nadana ramka nie koliduje z inną ramką nadajnik *musi nadać 512 bitów*, (14B nagłówek, 46B danych, 4B CRC) czyli wypełnić łącze, opóźnienie od końca do końca wynosi 51,2μs

wykrywanie kolizji

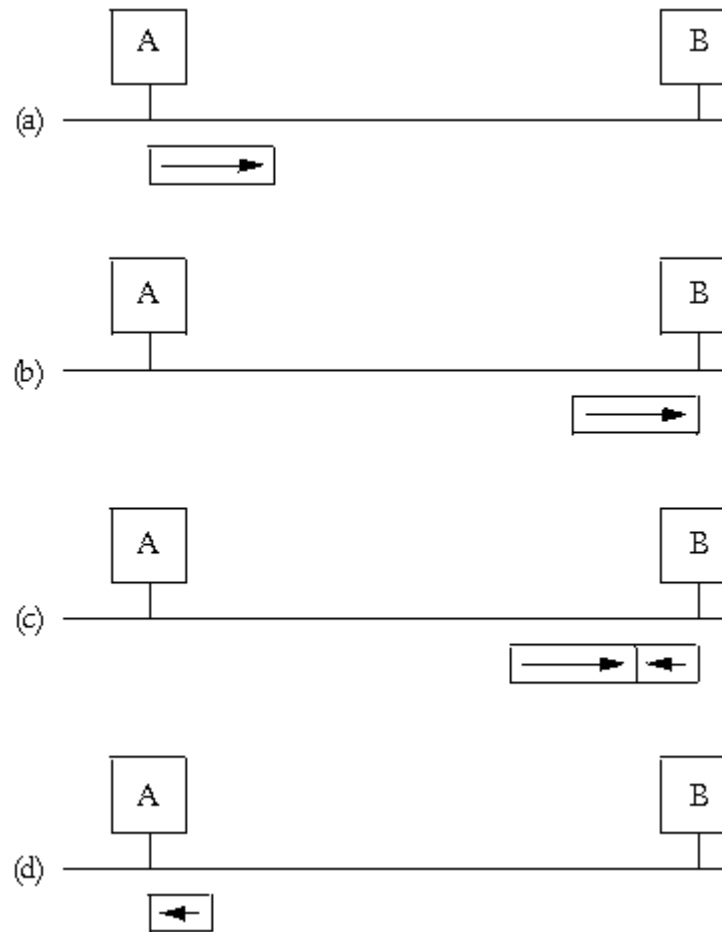
Walrand&Varaiya, High-Performance Communication Networks,
http://www.mkp.com/books_catalog/catalog.asp?ISBN=1-55860-574-6

komputer A musi nadawać przez czas równy *podwojonej opóźnieniu* ($51,2\mu\text{s}$) łącza o długości 2500m aby *stwierdzić kolizję*



wykrywanie kolizji - najgorszy przypadek

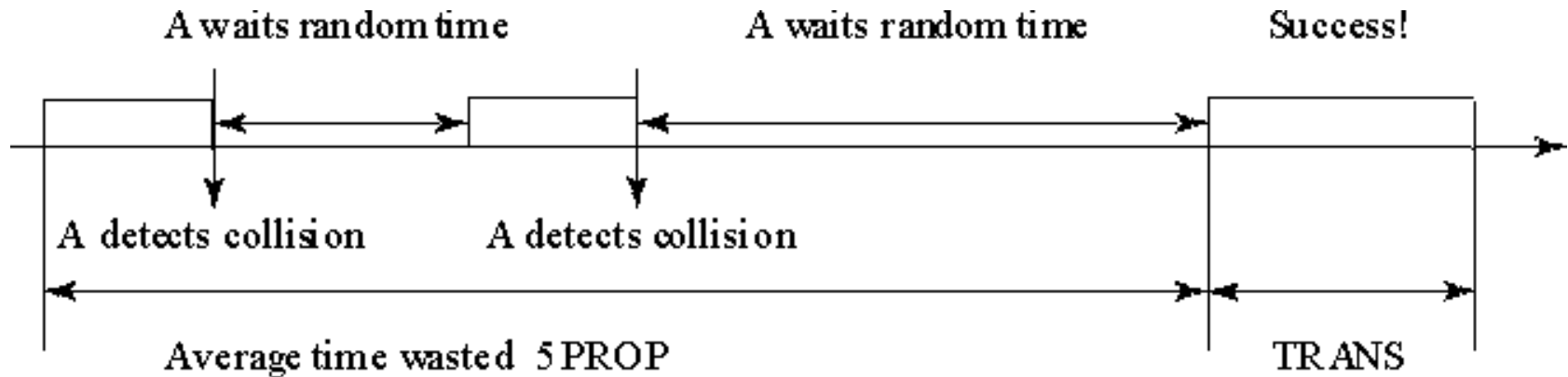
Peterson&Davie, Computer Networks, 2nd ed. KMP 2000



retransmisja ramki po kolizji

Walrand&Varaiya, High-Performance Communication Networks,
http://www.mkp.com/books_catalog/catalog.asp?ISBN=1-55860-574-6

- przed kolejną próbą nadawnik *podwaja czas oczekiwania*: 0, 51,2 μ s, 102,4 μ s, ...



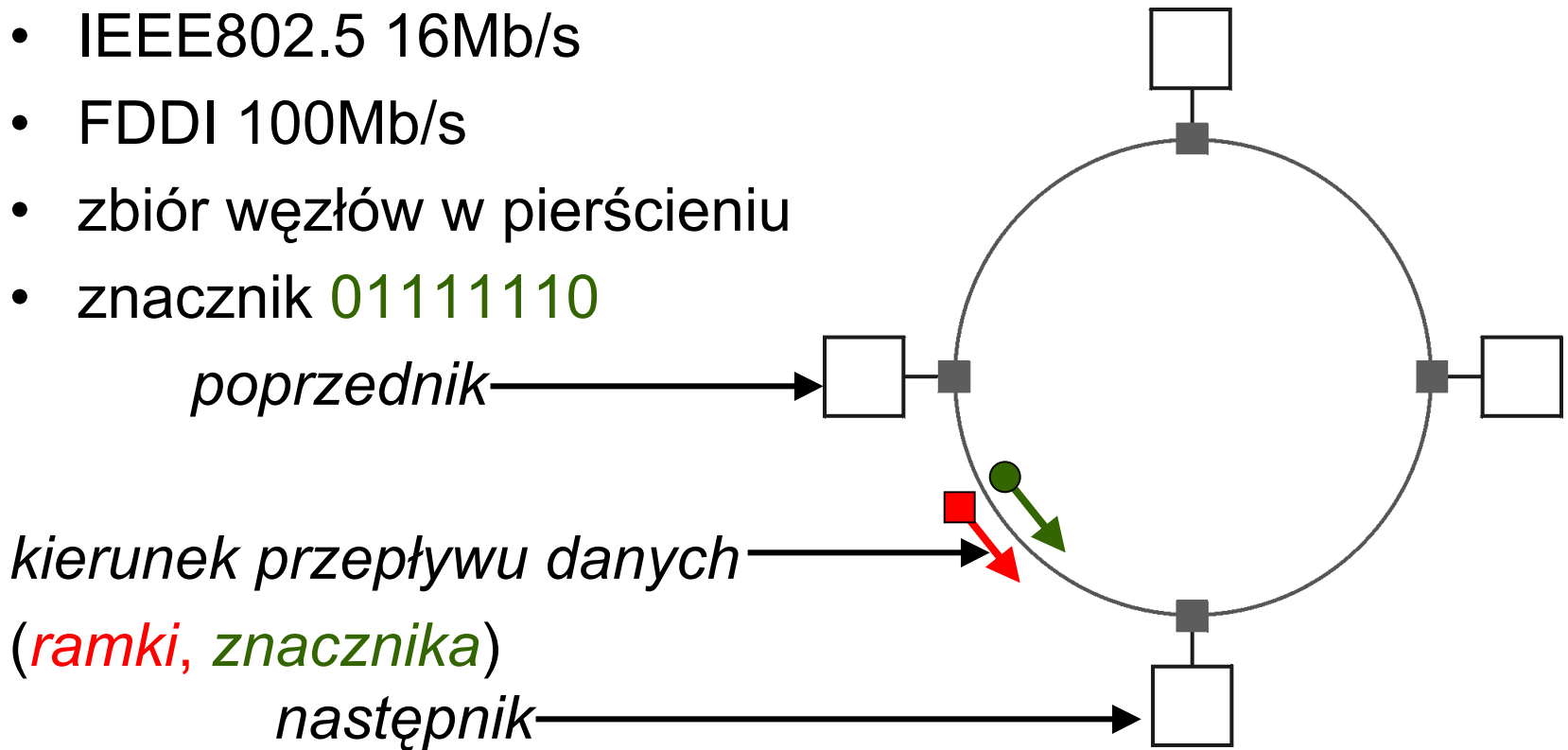
- adapter *rezygnuje* po 16 próbach i informuje komputer o błędzie nadawania

wnioski z pracy Ethernetem

- pracuje najlepiej przy niewielkim obciążeniu
- sieci do 200 komputerów
- < 500m
- opóźnienie $\sim 5\mu s$
- tani, nie ma komutatorów
- łatwy w administracji
- *nie nadaje się* do systemów czasu rzeczywistego - brak gwarancji nie przekroczenia określonego opóźnienia

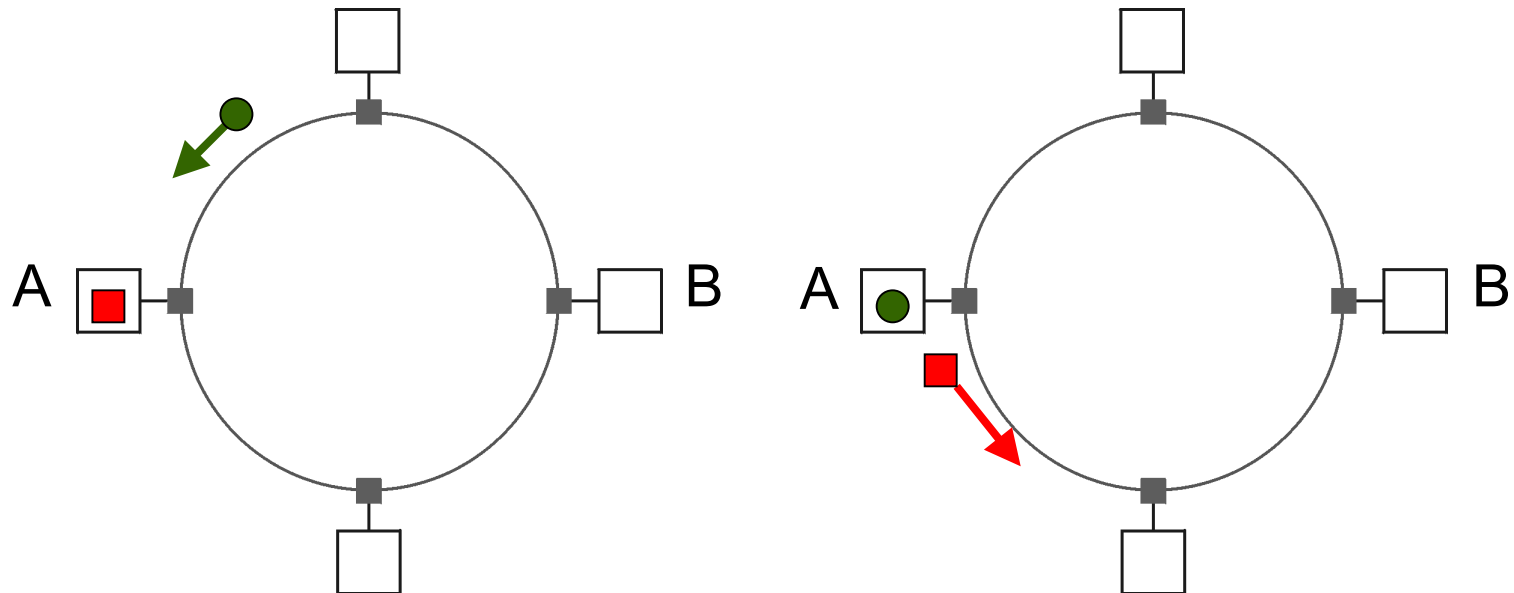
pierścień ze znacznikiem (FDDI)

- PRONET 10Mb/s i 80Mb/s, IBM 4Mb/s
- IEEE802.5 16Mb/s
- FDDI 100Mb/s
- zbiór węzłów w pierścieniu
- znacznik 01111110



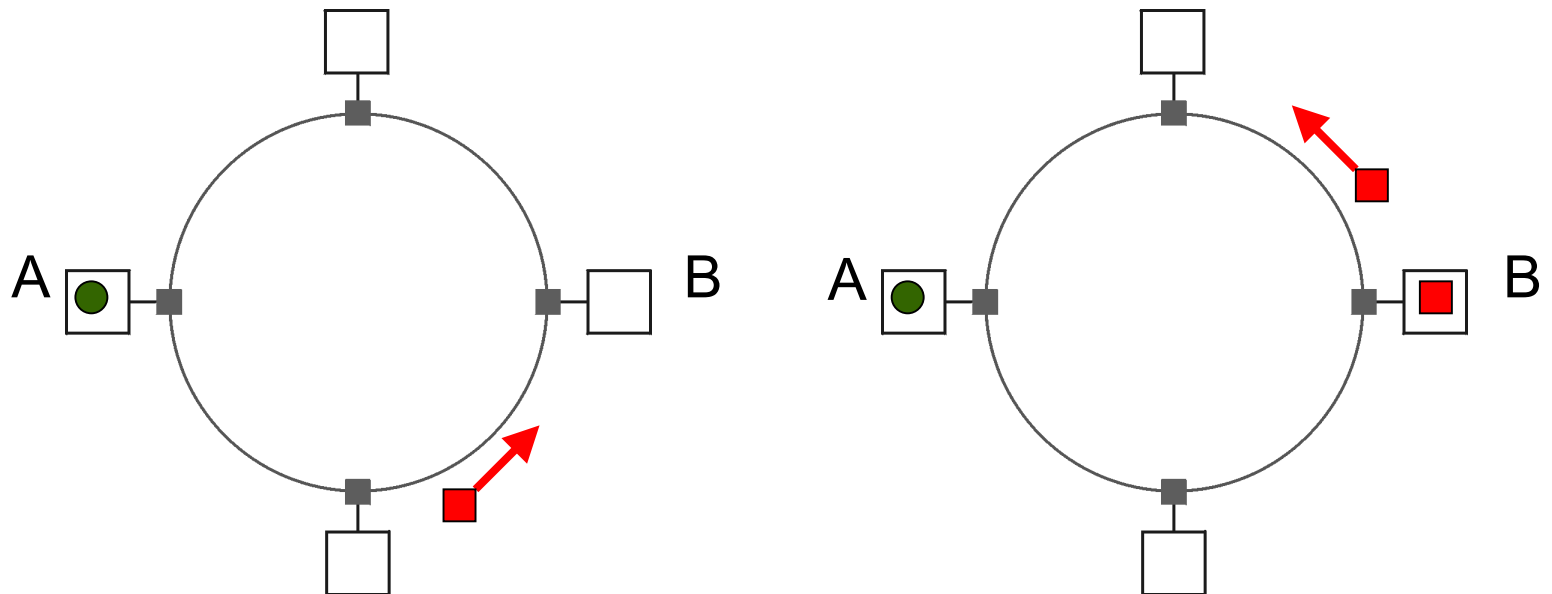
dostęp do współdzielonego pierścienia

- *znacznik* porusza się wokół pierścienia
- każdy węzeł odbiera *znacznik* i przekazuje dalej
- węzeł, który ma *ramkę* do nadania pobiera *znacznik* z pierścienia, a zamiast niego umieszcza w pierścieniu własną *ramkę*



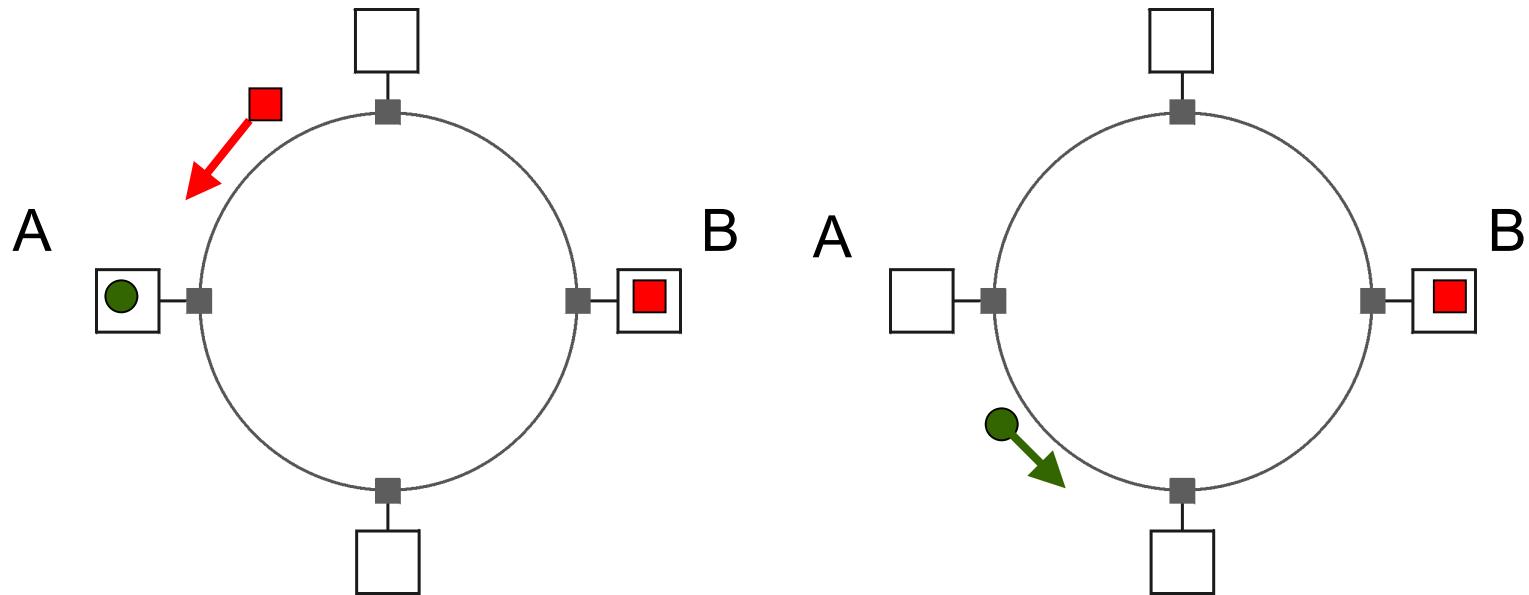
dostęp do współdzielonego pierścienia

- każdy węzeł przekazuje dalej *ramkę*
- odbiorca kopiuje *ramkę* i przekazuje *ramkę* dalej...



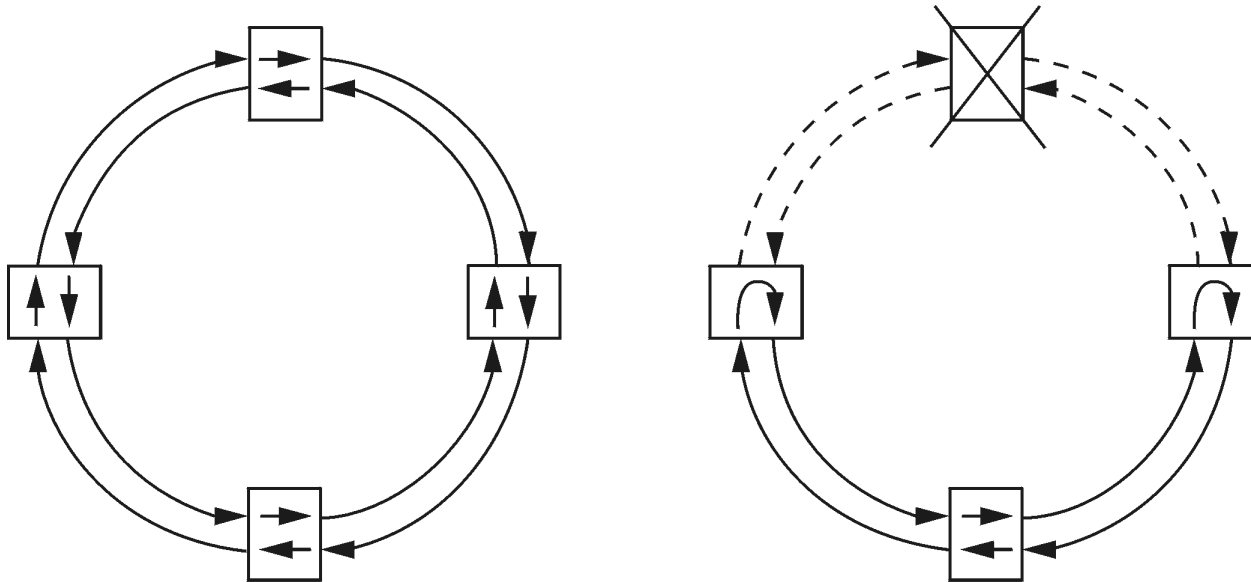
dostęp do współdzielonego pierścienia

- ... aż do nadawcy
- nadawca usuwa *ramkę* i umieszcza ponownie *znacznik* w pierścieniu



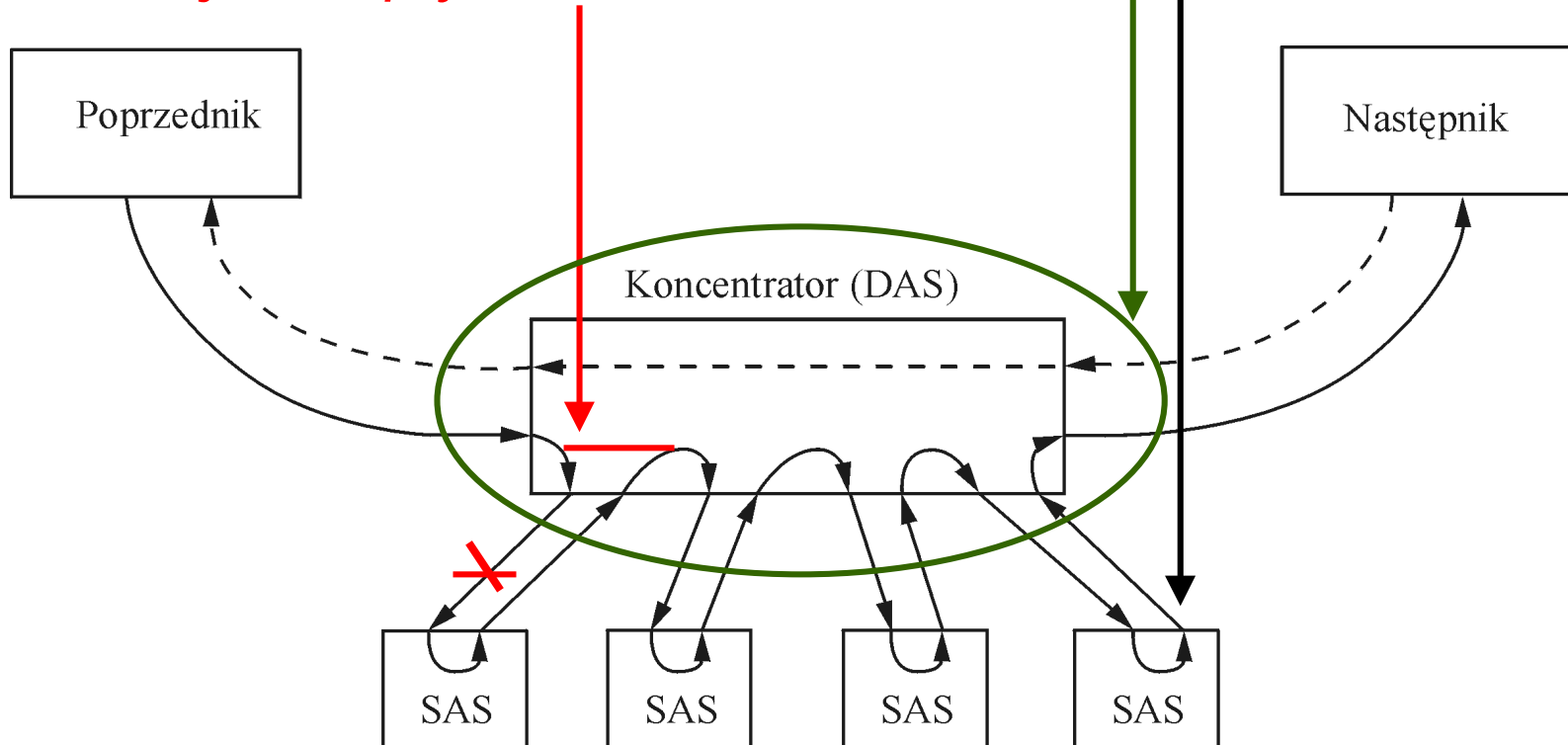
cechy fizyczne (FDDI)

- podwójny pierścień - dwa niezależne pierścienie, pierwotny i wtórny, nadające w przeciwnych kierunkach
- gdy pierwotny ulega awarii, następuje zamknięcie pętli na wtórnym



sposoby dołączenia węzła do pierścienia

- *stacja z podwójnym przyłączeniem (DAS)*
- *stacja z pojedynczym przyłączeniem (SAS)*
- *obejście optyczne*



cechy fizyczne FDDI

- każda stacja implementuje bufor o zmiennej pojemności, 9-80 bitów, odseparowujący odbiór od poprzednika od nadawania do następnika, wprowadza opóźnienie 10ns na bit
- maks. 500 komputerów
- maks. 2km między dowolną parą stacji
- maks. 200km światłowodu (100km kabla)
- kodowanie 4B/5B

czas posiadania znacznika THT

- *jak wiele danych węzeł może nadać, gdy posiada znacznik?*
- decyduje o tym *czas posiadania znacznika* THT
- strategia *tylko tyle danych ile ma* faworyzuje węzły, które mają dużo danych, tymczasem mały ale ważny komunikat w innym węźle czeka długo
- n.p. w IEEE 802.5 THT=10ms
- z czasu THT można wyprowadzić inny ważny czas, zwany czasem powrotu znacznika TRT

czas powrotu znacznika TRT

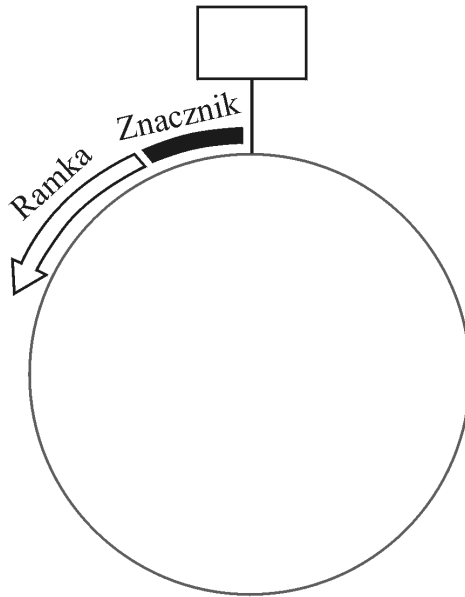
- *czas powrotu znacznika* TRT, jest to czas przejścia znacznika przez pierścień widziany z określonego węzła: $TRT \leq \text{ActiveNodes} \times THT + \text{RingLatency}$
- *węzeł aktywny*: ma dane do nadania
- *opóźnienie pierścienia*: czas przejścia znacznika przez pierścień, gdy nie ma danych do nadania
- uzgadnia się *docelowy czas powrotu znacznika* TTRT, aby nałożyć górną granicę na TRT
- gdy $\text{zmierzony TRT} \geq \text{TTRT}$ to znacznik opóźniony i węzeł nie nadaje, gdy $\text{zmierzony TRT} < \text{TTRT}$ to dopuszcza się, że węzeł posiada znacznik przez czas $\text{TTRT} - \text{zmierzony TRT}$

zwalnianie znacznika przez węzeł nadający

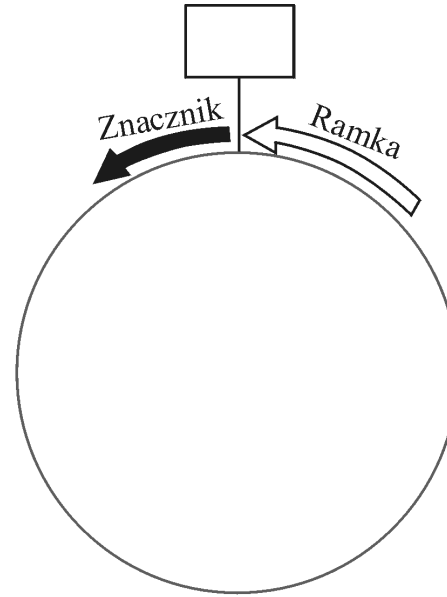
nadajnik umieszcza znacznik w pierścieniu:

- natychmiast po swojej ramce
- po przejściu ramki przez pierścień i jej usunięciu

zwolnienie natychmiastowe



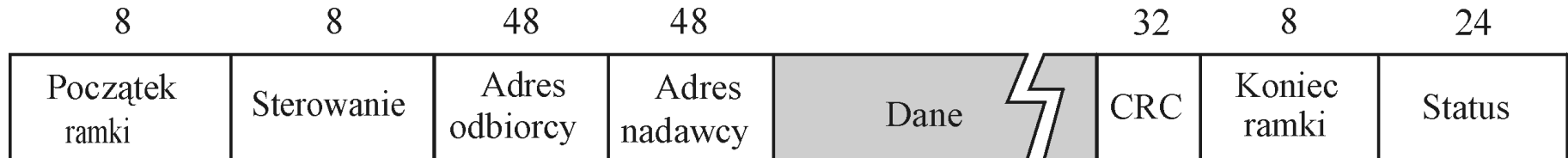
zwolnienie opóźnione



konserwacja znacznika

- *problem*: znacznik można stracić, przez błąd bitu, lub awarię węzła posiadającego znacznik
- *generacja nowego znacznika po stracie*:
- uzgadnia się nową wartość **TTRT**-ramką uzgadniania
- znacznik uzyskuje węzeł proponujący *najniższy czas TTRT* - czyli ten węzeł, którego ramka uzgadniania wraca do niego
- *monitorowanie ważności znacznika*: maks. czas bezczynności między transmisjami to **2,5ms** (równy opóźnieniu pierścienia + czas transmisji pełnej ramki)
- gdy upływa, węzeł nadaje ramkę uzgadniania

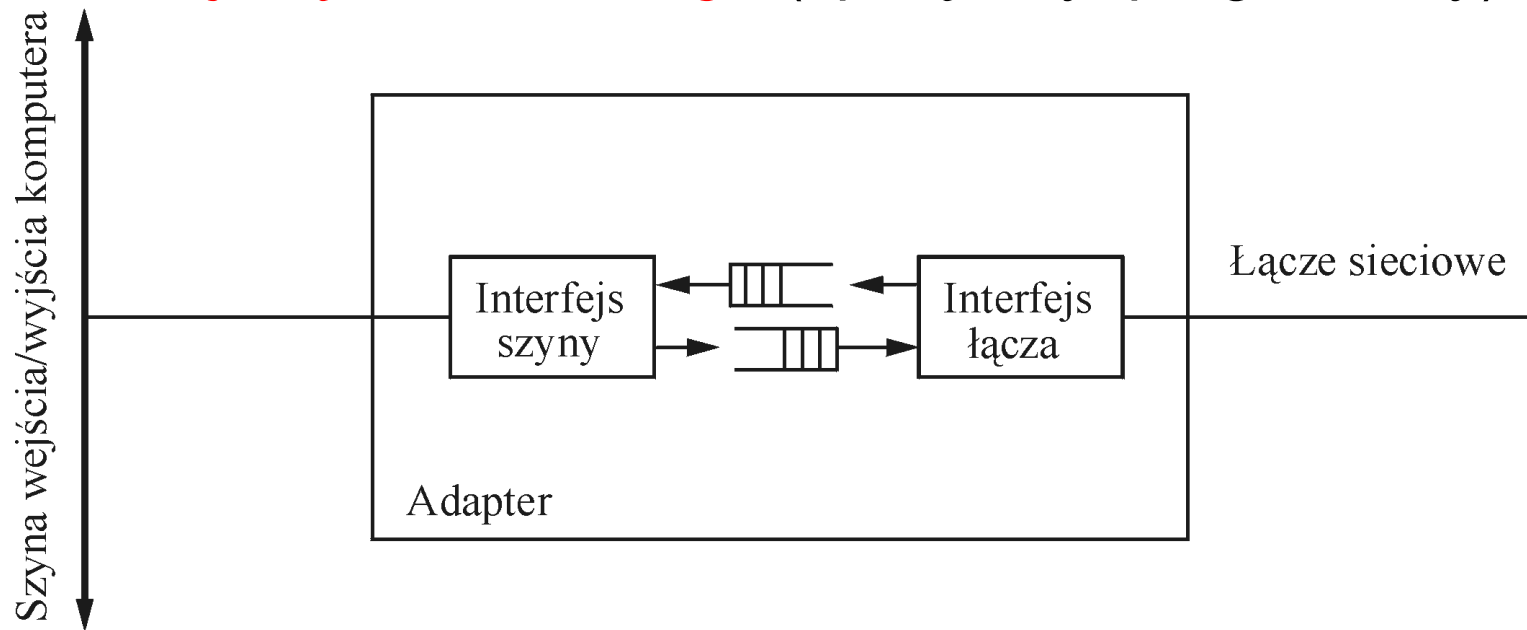
format ramki FDDI



- kodowanie 4B/5B
- adresy 48 bitowe
- *bitowe rozpoznawanie ramek* (unikalne sekwencje początku i końca ramki)
- CRC-32
- maks. 4500 bajtów, w tym 4478 bajtów danych
- *pole sterujące* (do wyróżnienia ramki)
- *status ramki* (zwracany do nadawcy - to pierścień!)

adapter sieci

- adapter jest *interfejsem* między komputerem a siecią
- *interfejs szyny* (szerokość szyny 32b, $f=25\text{MHz}$, cykl 40ns, co daje szybkość transmisji 800Mb/s)
- *interfejs łączy sieciowego* (sprzętowy, programowy)



spojrzenie na adapter od strony komputera

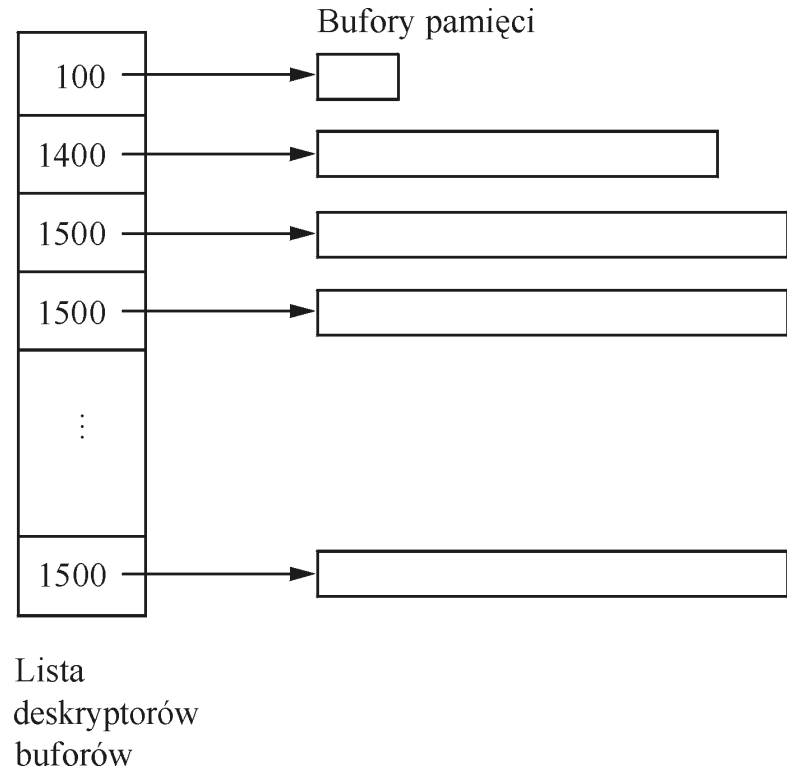
- z punktu widzenia jednostki centralnej, adapter udostępnia jej *rejestr stanu sterowania CSR*, odczytywalny i zapisywalny przez jednostkę centralną, umieszczany pod pewnym adresem w pamięci
- jednostka centralna *zapisuje CSR*, aby nakazać adapterowi nadanie lub odbiór ramki i *odczytuje CSR* aby poznać stan adaptera
- *odpytywanie* - sprawdzanie bitu w *CSR* czy nadeszła ramka
- *przerwanie od adaptera* do komputera - *obsługa przerwania* w systemie operacyjnym komputera

sposób przesyłania ramek między adapterem a pamięcią komputera

- bezpośredni dostęp do pamięci (DMA)
- programowane wejście/wyjście (PIO)

bezpośredni dostęp do pamięci (DMA)

- *nie ma potrzeby buforowania* ramek w adapterze, adapter odczytuje i zapisuje pamięć komputera
- *jednostka centralna przekazuje adapterowi dwie listy deskryptorów plików*



programowane wejście/wyjście (PIO)

- adapter sieci zawiera *pewną ilość buforów* - jednostka centralna *kopiuje ramki* między pamięcią komputera a pamięcią adaptera (*dwuwejściową, stąd drogą*)

