

Szymon Łukasik  
KAiT

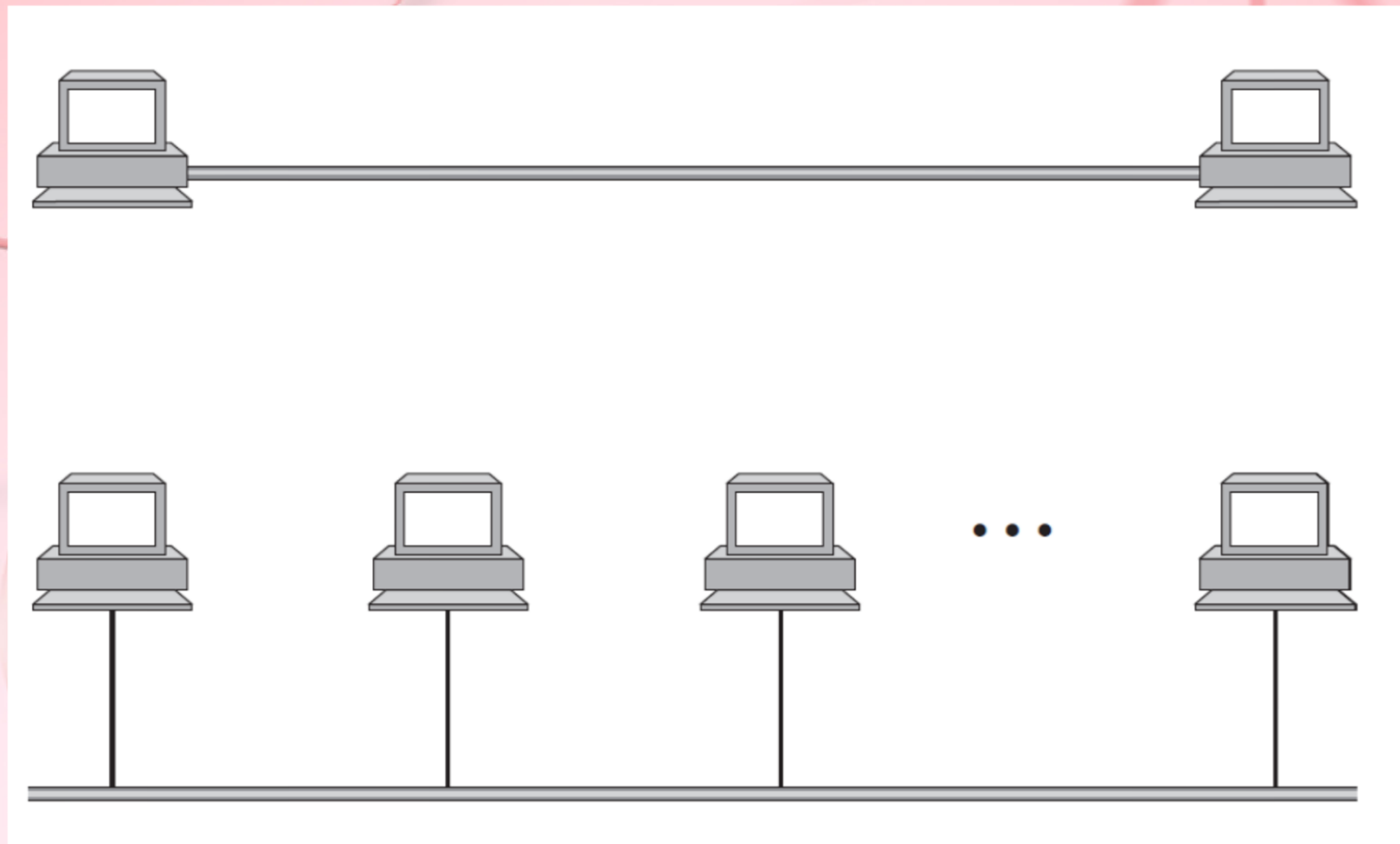


# Sieci komputerowe - podstawowe technologie

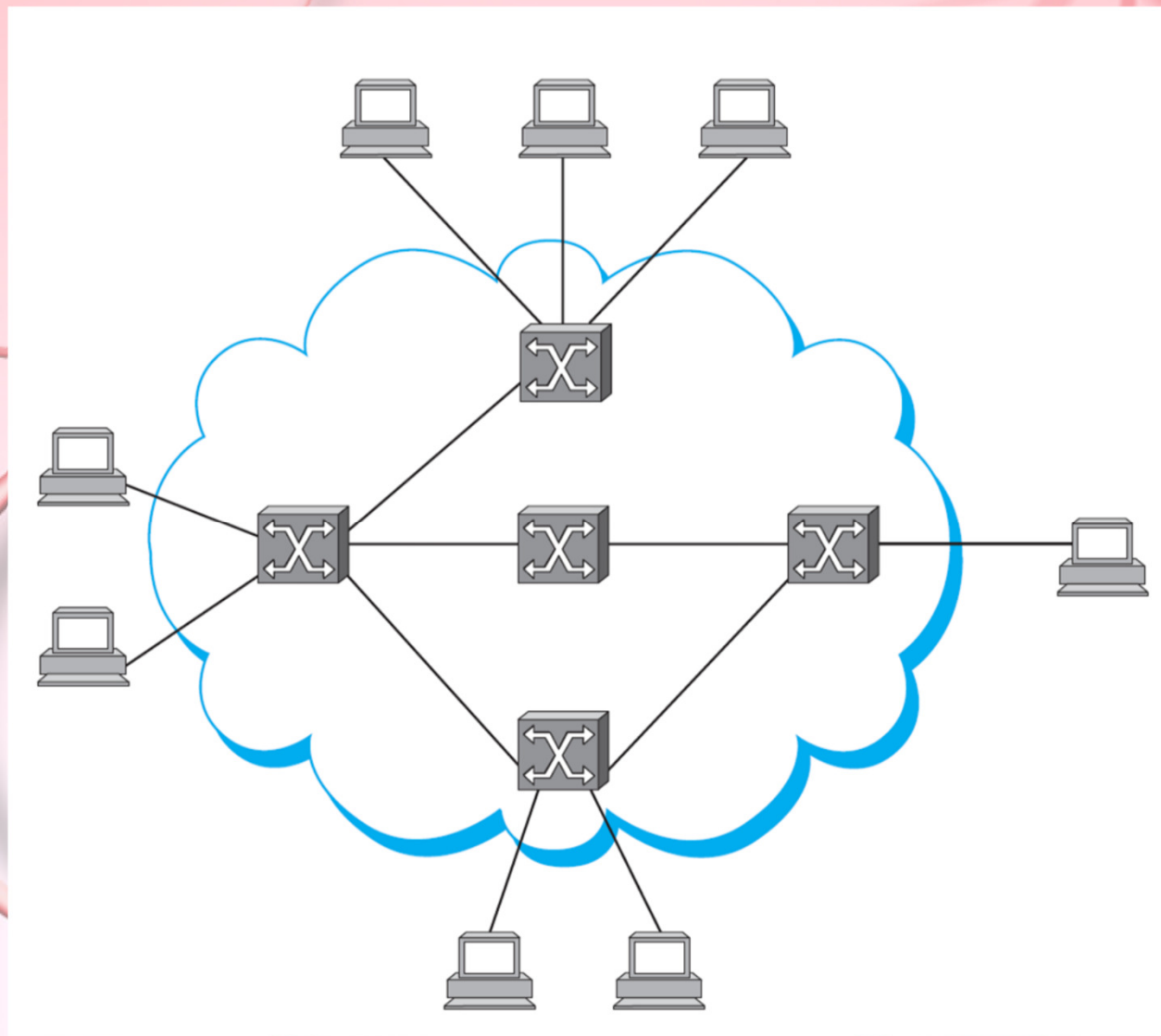
# Plan wykładu

1. Wymagania – czyli czego tak naprawdę od sieci komputerowych oczekujemy.
2. Modele warstwowe sieci komputerowych – czyli 40 lat historii pracuje dla nas.
3. Aspekty wydajnościowe – czyli opóźnienie i przepustowość (fakty i mity).
4. Podstawy transmisji danych i media transmisyjne – od kabla koncentrycznego po światłowody.
5. Kodowanie bitów.
6. Wykrywanie i korekcja błędów.

# Połączenia – proste sieci

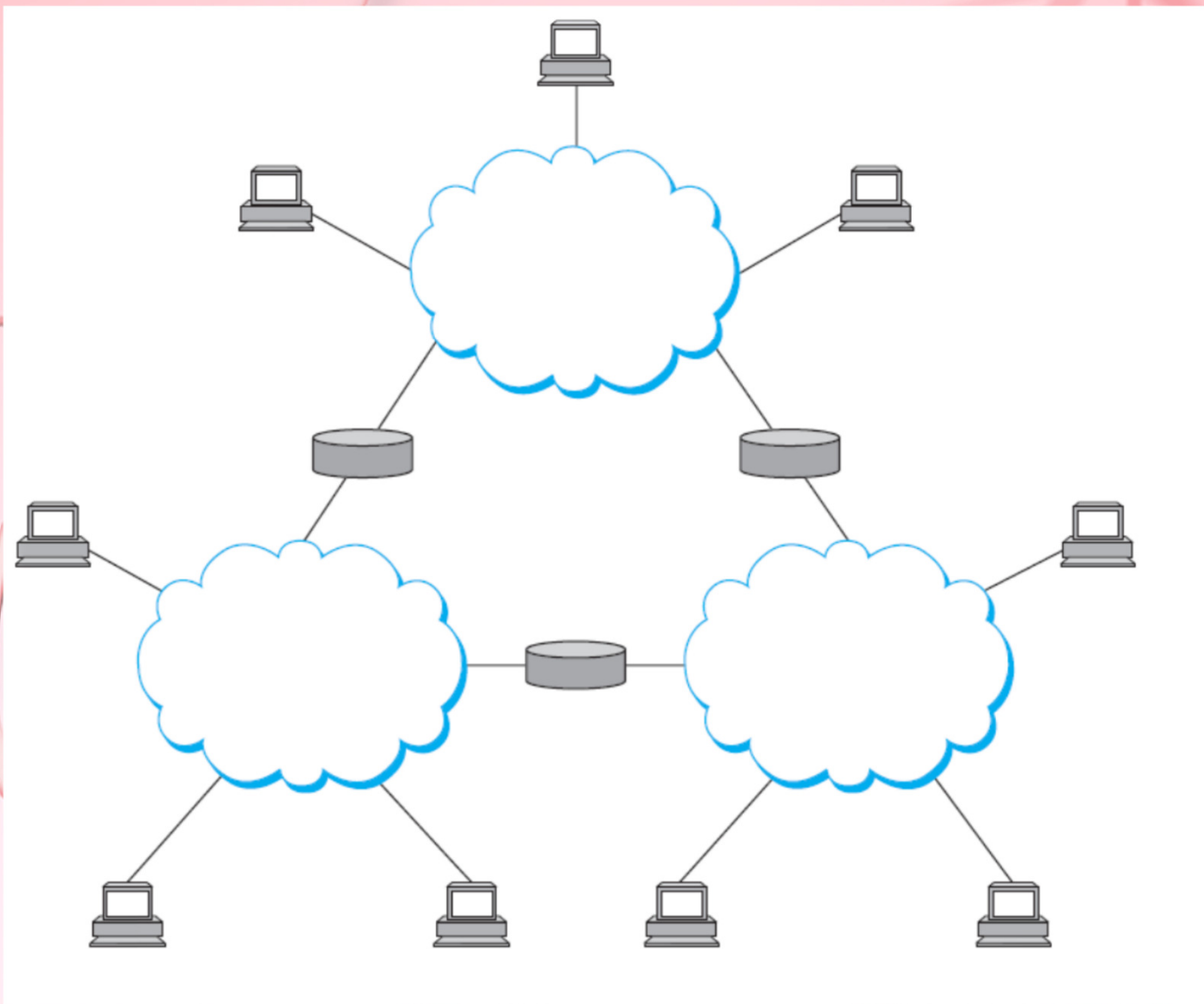


# Sieć z przełączaniem pakietów



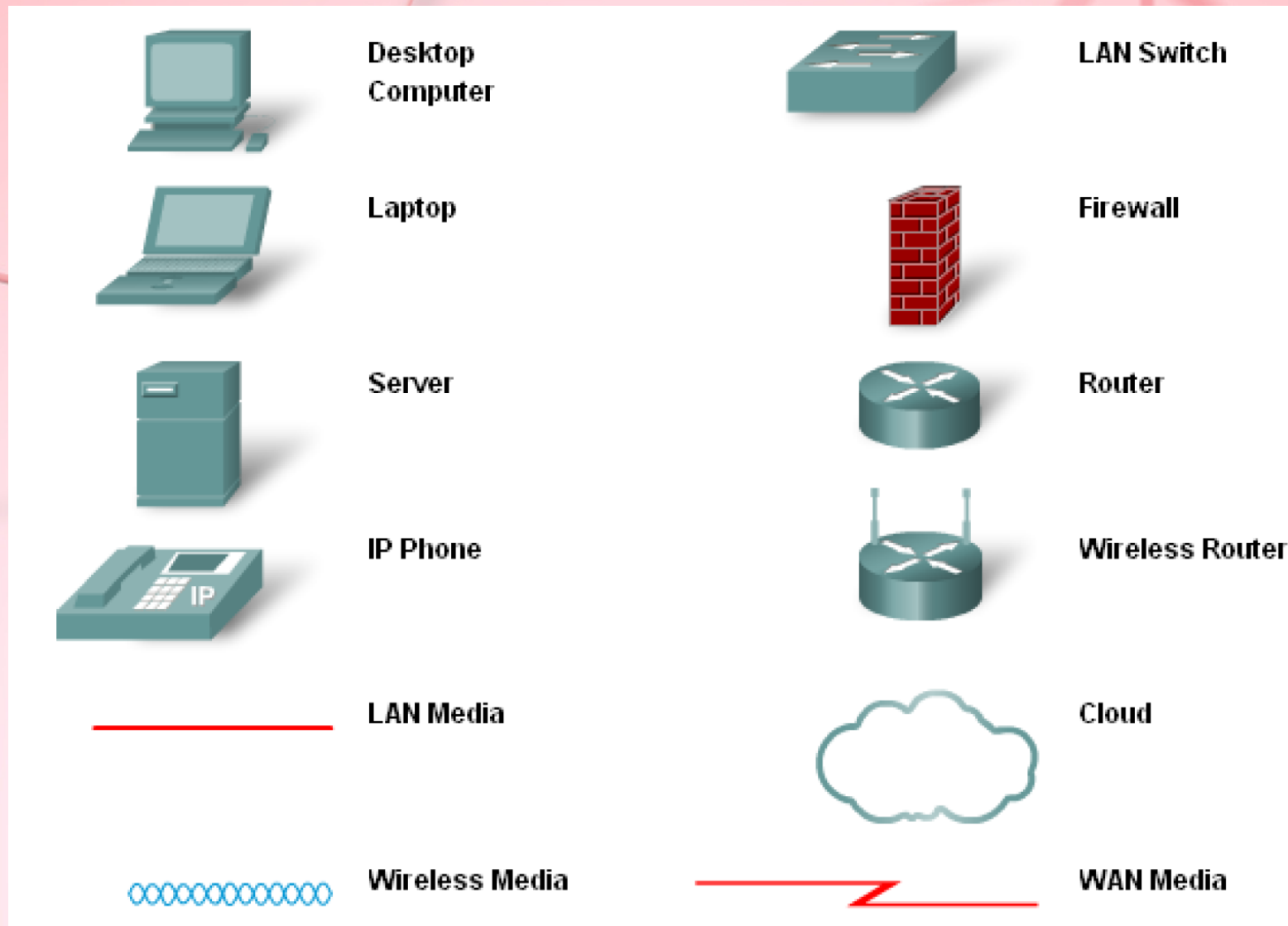
Sieci komputerowe

# Połączenia sieci – sieć typu internet

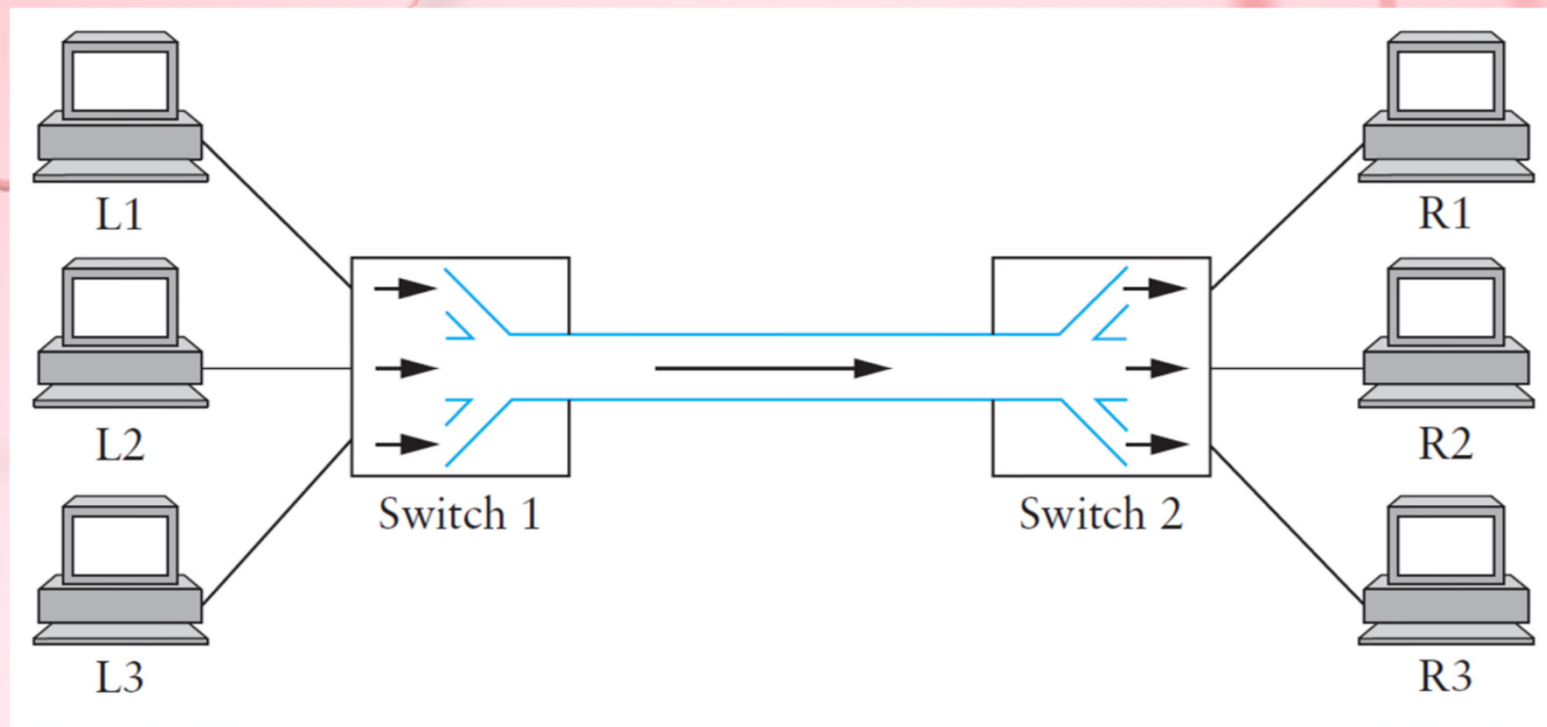


Sieci komputerowe

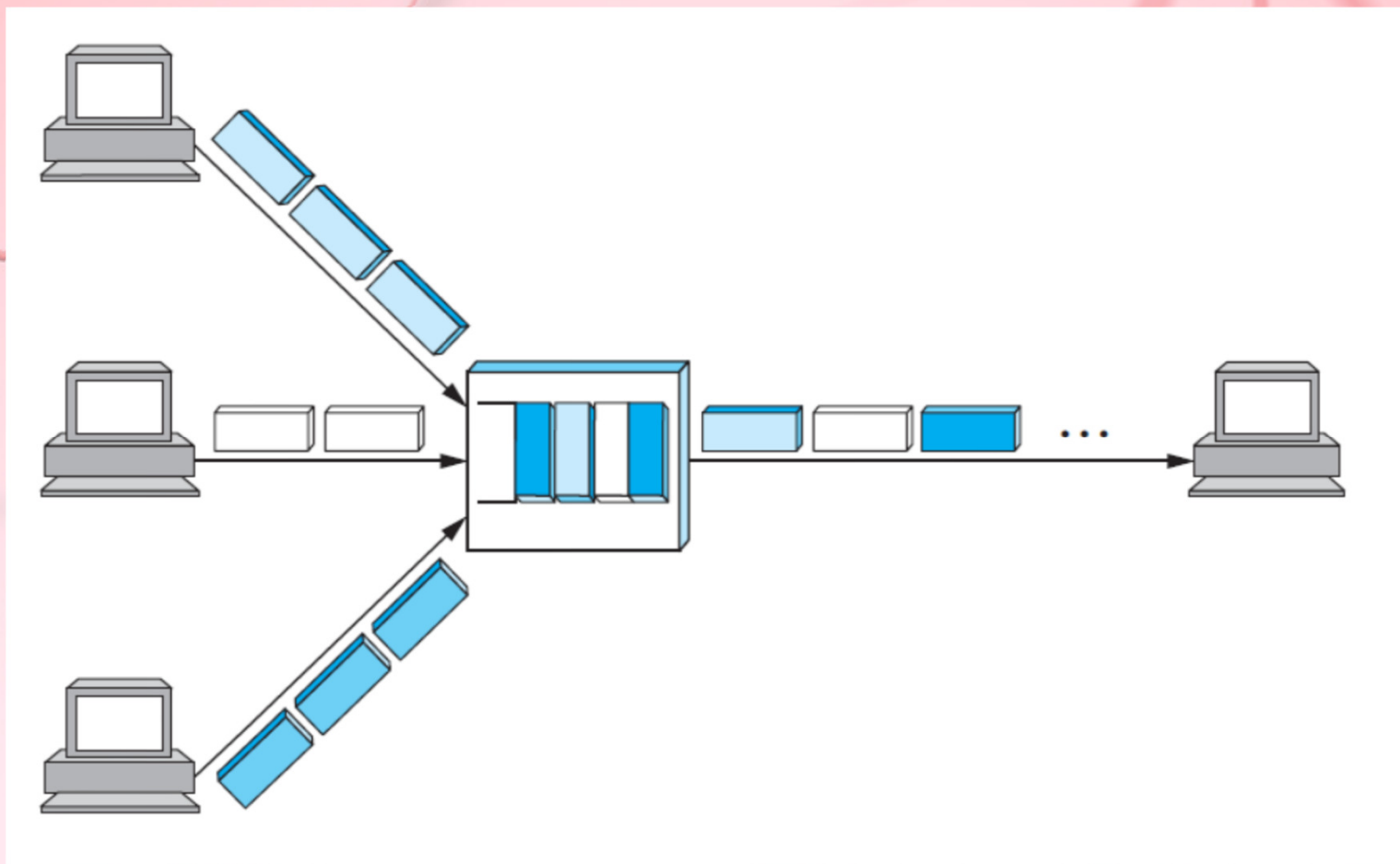
# Symbole



# Dzielenie łącza

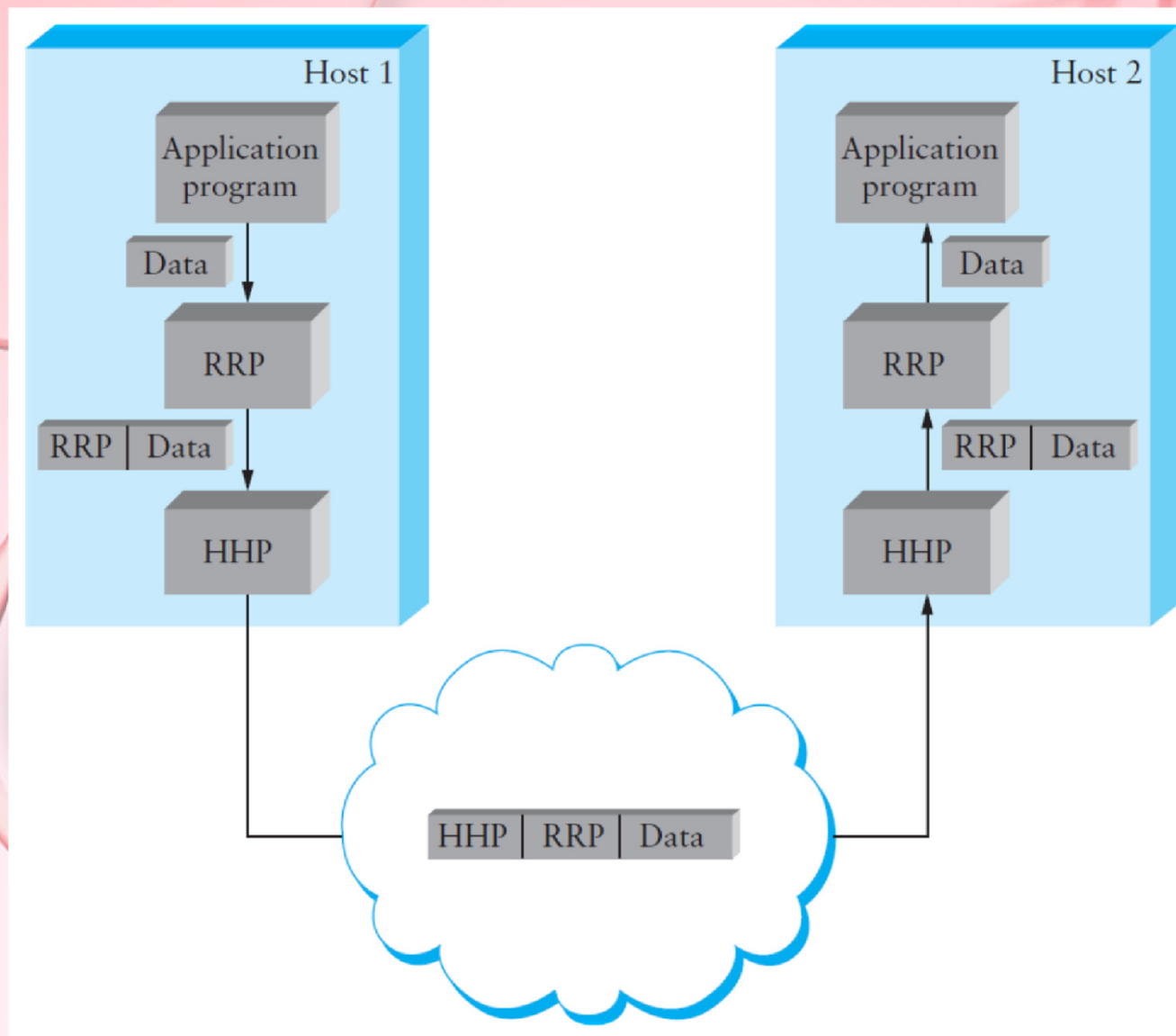


# Dzielenie łącza + transparentne przesyłanie

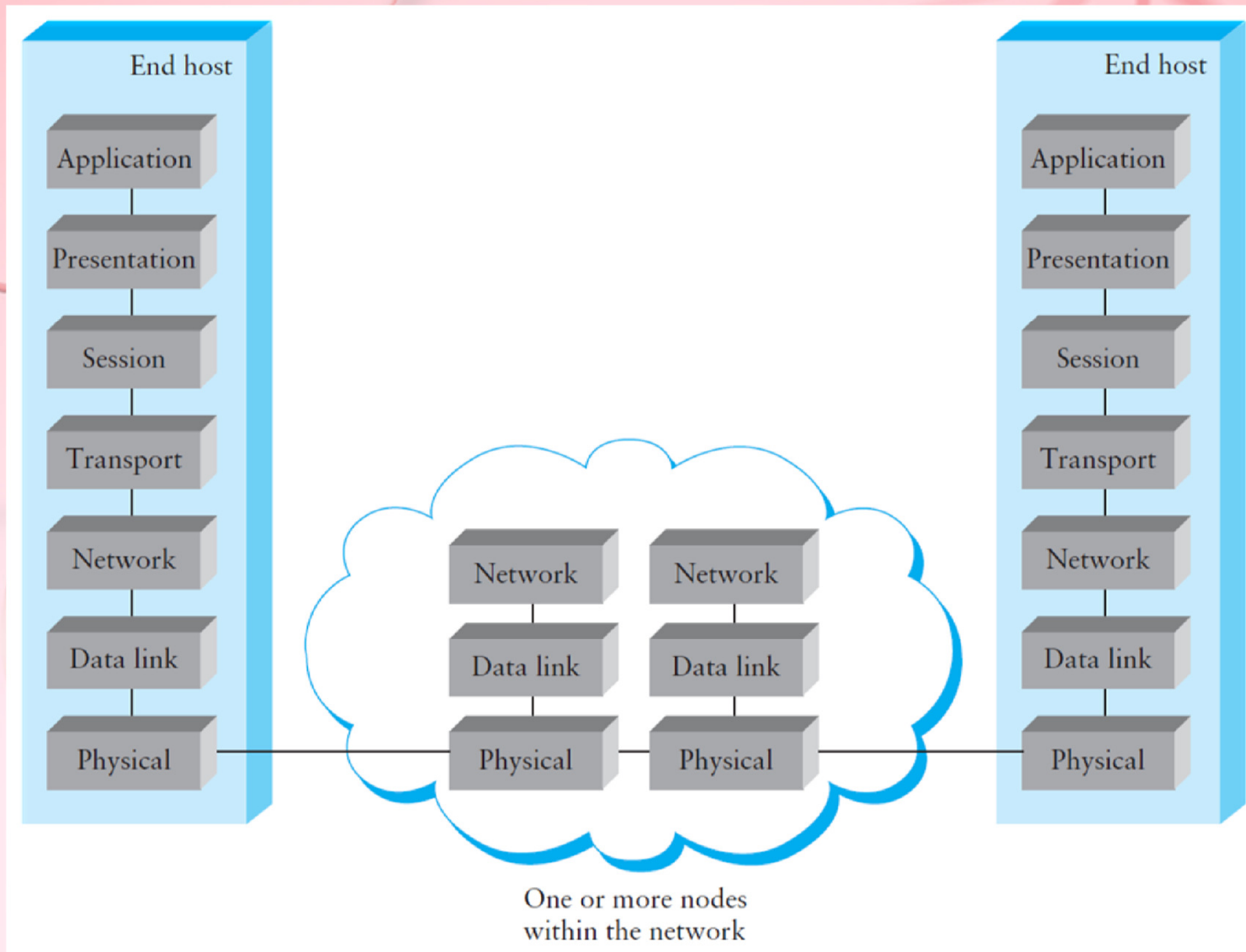




# Warstwowy model sieci i kapsułkowanie



# Warstwowy model sieci OSI



# Warstwy OSI

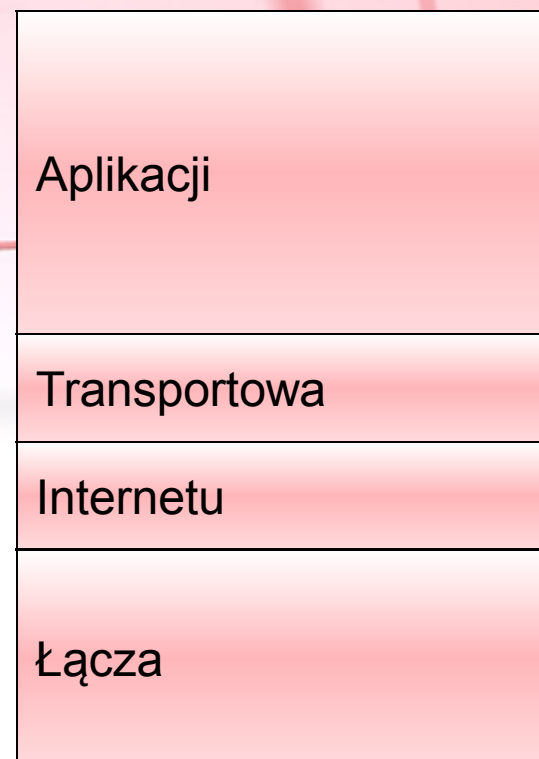
	Jednostka danych	Warstwa	Funkcja
<b>Warstwy hostów</b>	Dane	7. Aplikacji	Bezpośredni dostęp sieciowy dla aplikacji
		6. Prezentacji	Reprezentacja danych, szyfrowanie i deszyfrowanie, konwertowanie danych zależnych od platformy sprzętowej do postaci uniwersalnej
		5. Sesji	Komunikacja między hostami
	Segmenty	4. Transportowa	Połączenia końcowe, niezawodność kontrola transmisji
<b>Warstwy mediów</b>	Pakiety	3. Sieciowa	Określanie trasy i adresowanie logiczne
	Ramki	2. Łącza danych	Adresowanie fizyczne
	Bity	1. Fizyczna	Media, sygnały i transmisja binarna

# Model OSI, a sieci TCP/IP

## OSI



## TCP/IP



## Przepustowość (ang. bandwidth/throughput)

W komunikacji elektronicznej przepustowość to szerokość pasma częstotliwości która może być wykorzystana do przesyłania danych (liczy się ją jako różnicę min. i max. Użytecznej częstotliwości w Hercach – np. dla linii telefonicznej  $3300\text{Hz} - 300\text{Hz} = 3000\text{Hz}$ ).

W sieciach komputerowych przepustowość jest często utożsamiana z szybkością transferu danych tj. ilością danych jakie można przenieść z jednego punktu do drugiego w danym czasie (zwykle 1s). Jeżeli mówimy o przepustowości teoretycznej to używamy ang. terminu bandwidth, jeśli zaś mierzonej eksperymentalnie – ang. throughput. Te przepustowości nigdy nie tożsame (wbrew rozmaitym reklamom).

## Opóźnienie (ang. latency)

Drugą miarą pozwalającą na ocenę sieci jest opóźnienie – czyli czas potrzebny na dotarcie wiadomości z jednego punktu sieci do drugiego. Przykładowo sieć międzykontynentalna może mieć opóźnienie 24ms – tyle czasu potrzeba na przesłanie wiadomości z jednego kontynentu na drugi.

Czasami ważniejsze od czasu przesłania pojedynczej wiadomości jest czas dotarcia informacji zwrotnej (czyli przesłania danych tam i z powrotem). Definiuje się go jako tzw. *round-trip time (RTT)* sieci.

# Źródła opóźnień

## Opóźnienie

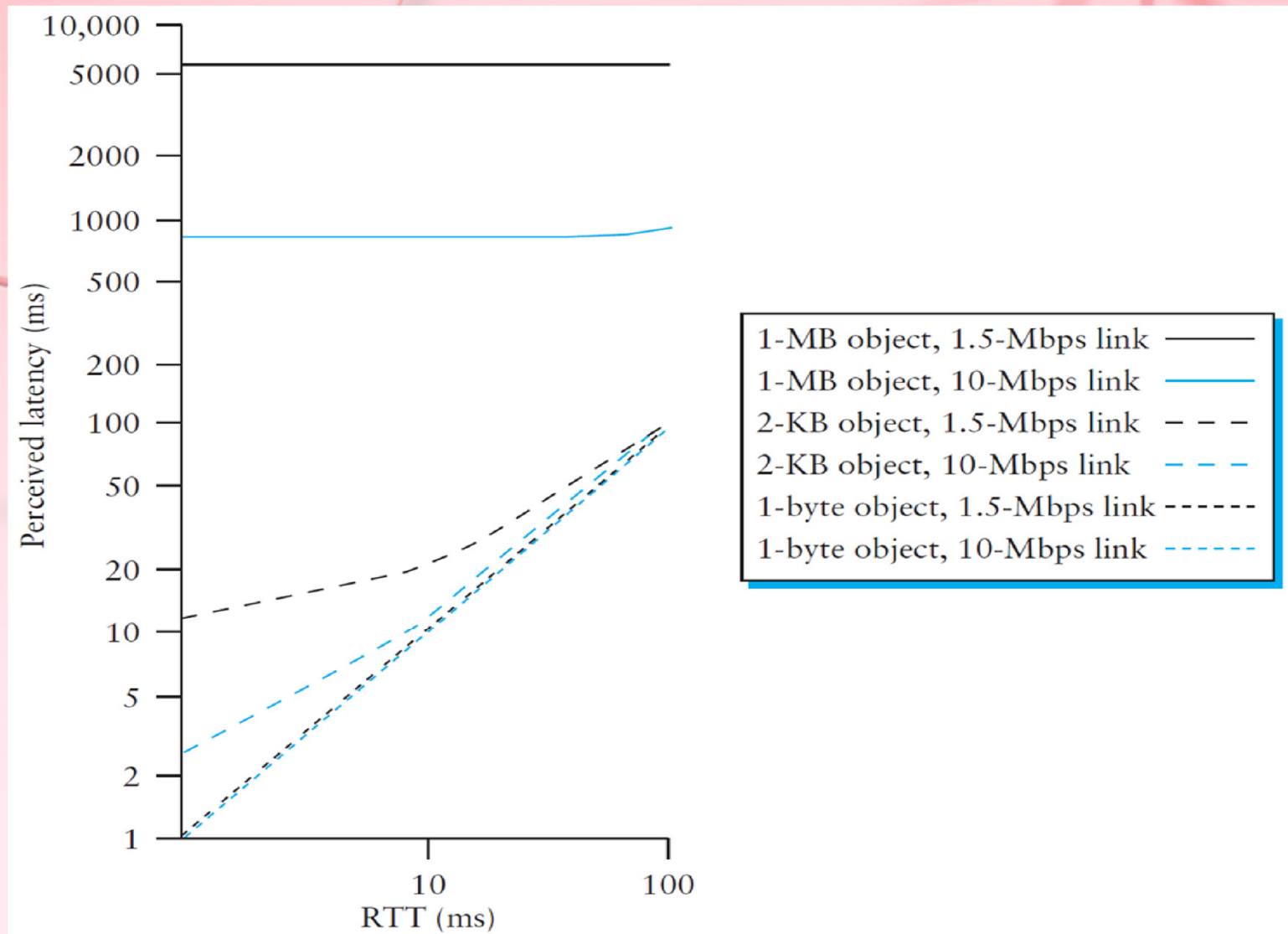
= Czas propagacji + Czas transmisji + Czas kolejkwania

**Czas propagacji** – maksymalna prędkość przesyłu ograniczana przez prędkość światła ( $3,0 \cdot 10^8$  m/s – próżnia,  $2,3 \cdot 10^8$  m/s – przewód),  $t_{propagacji} = \text{odległość} / c$

**Czas transmisji** – związany z przepustowością łącza,  $t_{transmisji} = \text{rozmiar\_pakietu} / \text{przepustowość}$

**Czas kolejkwania** – w sieci przełączanej pakiety muszą być buforowane w pamięci urządzeń przełączających, buforowanie to powoduje dodatkowe opóźnienia

# Opóźnienie, a przepustowość





## Pułapki Gbps

### **Efektywna przepustowość łącza**

= Rozmiar pakietu / Czas transferu

Czas transferu to nie tylko opóźnienie, należy wziąć pod uwagę także nawiązanie połączenia, sprawdzanie poprawności itp., ogólnie:

Czas transferu = RTT + 1/Szerokość pasma × Rozmiar pakietu.

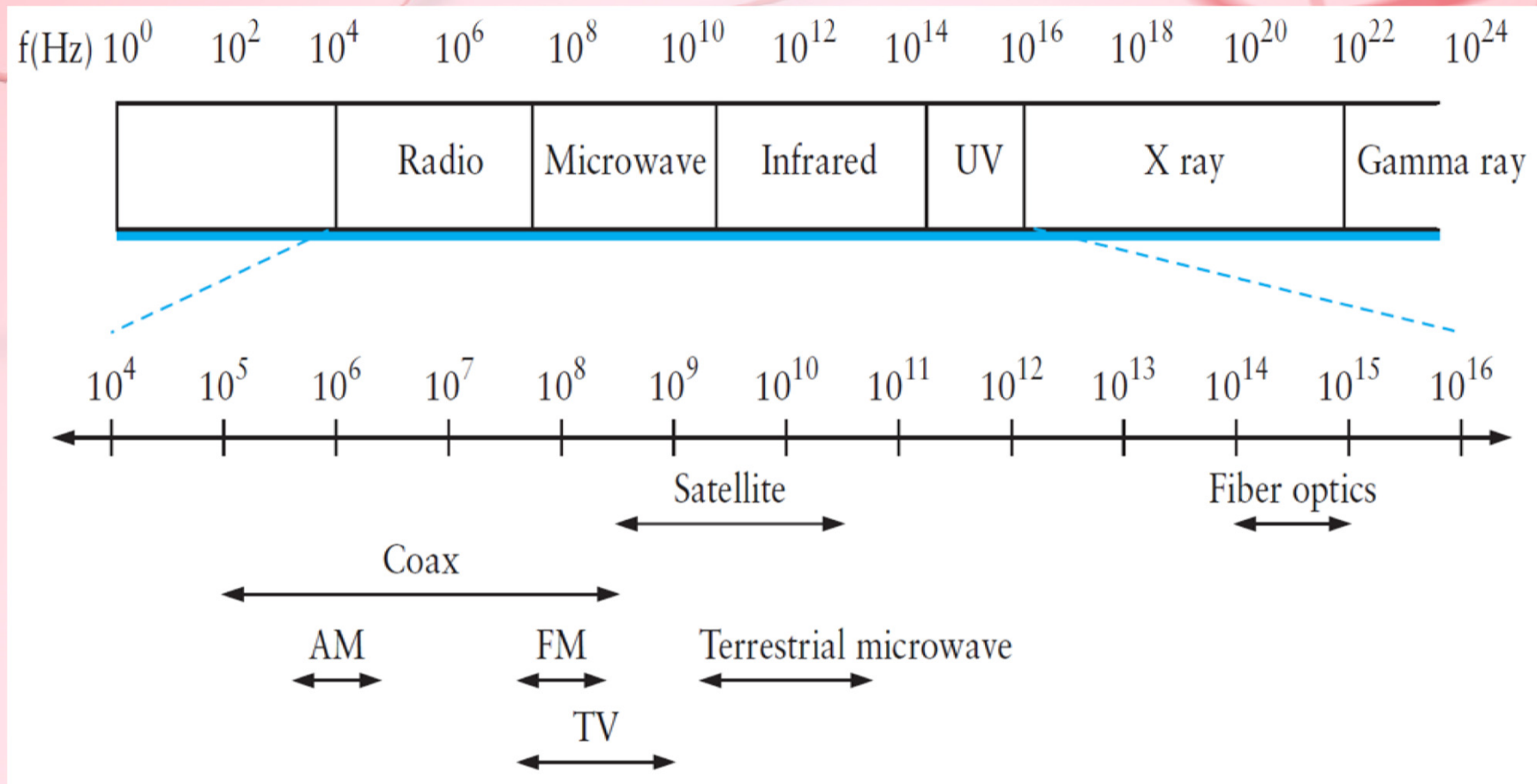
Dla łącza o szerokości pasma 1Gbps i rozmiarze pakietu danych 1MB, sieci o opóźnieniu RTT=100ms mamy:

Czas transferu = 108ms

Efektywna przepustowość = 1MB/108 ms = 74,1Mbps

# Pasmo transmisji

Dane – reprezentowane jako sygnały przesyłane poprzez fale elektromagnetyczne, o różnych częstotliwościach.



# Media transmisyjne

Medium	Typowa szerokość pasma	Odległości
Skrętka kat. 5	10–100Mbps	100m
Skrętka kat. 6	10–1000Mbps	100m
Kabel koncentryczny	10–100Mbps	500m
Światłowód wielomodowy SR	10 Gbs	300 m
Światłowód jednomodowy	100–2400 Mbps	40 km
Sieć 802.11n	150Mbps	250m
Bluetooth	1Mbps	10m

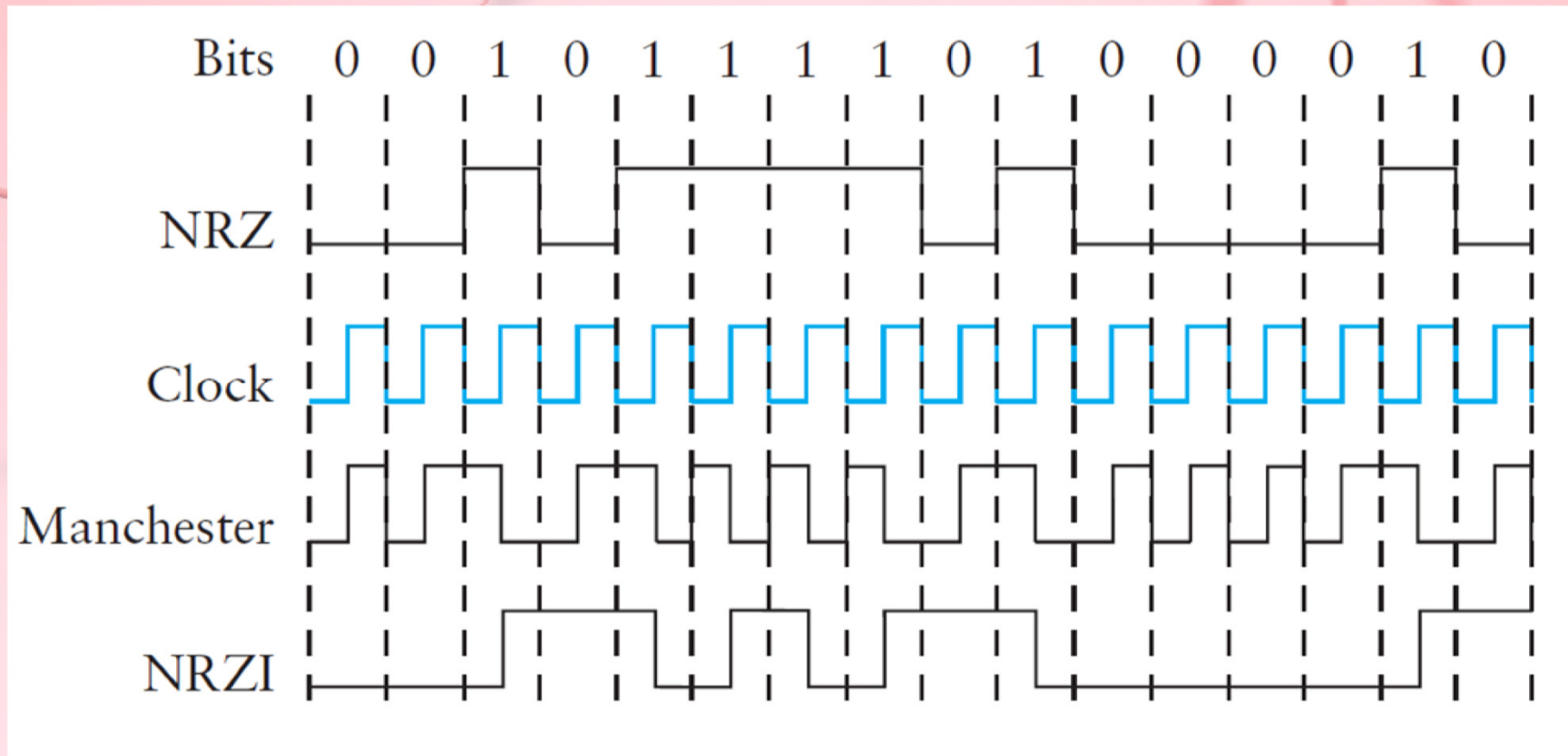
## Połączenia „ostatniej mili”

DSL (ang. Digital Subscriber Line) - cyfrowa linia abonencka umożliwiająca szerokopasmowy dostęp do Internetu.

Technologia DSL wykorzystuje wyższe, nieużywane pasmo przyłącza ( $>3300\text{Hz}$ ) poprzez tworzenie kanałów, o szerokości  $4312,5\text{ Hz}$  każdy, zaczynających się pomiędzy  $10$  a  $100\text{ kHz}$ , w zależności od konfiguracji systemu. Przydział kanałów jest kontynuowany na wyższych i wyższych częstotliwościach (dla ADSL do  $1,1\text{MHz}$ )

ADSL (Asymetryczne DSL) to obecnie najbardziej rozpowszechniona forma dostępu do Internetu w Polsce. Maksymalna prędkość pobierania danych dla ADSL to  $24\text{Mbps}$ .

# Kodowanie sygnałów cyfrowych



## Wykrywanie błędów – parzystość (dwuwymiarowa)

		Parity bits
Data	0101001	1
	1101001	0
	1011110	1
	0001110	1
	0110100	1
	1011111	0
Parity byte	1111011	0

Wykrywa wszystkie błędy 1-3 bitowe, większość – 4 bitowych

## Internetowa suma kontrolna (16-bitowa, dopełnienie do 1)

```
u_short cksum(u_short *buf, int count)
{
    register u_long sum = 0;
    while (count--)
    {
        sum += *buf++;
        if (sum & 0xFFFF0000)
        {
            /*przeniesienie - zawiń */
            sum &= 0xFFFF;
            sum++;
        }
    }
    return ~ (sum & 0xFFFF);
}
```

## Cykliczny Kod Nadmiarowy (CRC)

Ciąg bitów reprezentowany przez wielomian, np.

10011010

$$M(x) = 1 \times x^7 + 0 \times x^6 + 0 \times x^5 + 1 \times x^4 + 1 \times x^3 + 0 \times x^2 + 1 \times x^1 + 0 \times x^0$$

1. Ustalamy wielomian  $C(x)$  o rzędzie  $k$  – dzielnik, o pewnej długości (np. Ethernet – 32bitowy).
2. Jeżeli  $M(x)$  ma długość  $n+1$  dodajemy do niego  $k$ -dodatkowych bitów, tworząc wielomian  $P(x)$ .
3. Wielomian ten ma być bez reszty podzielny przez  $C(x)$  – jeśli tak będzie po stronie odbiorcy to wiadomość nie uległa zniekształceniu.



## Dzielenie „bitowych” wielomianów

Dowolny wielomian  $B(x)$  może być podzielony raz przez  $C(x)$  gdy  $B(x)$  jest takiego samego rzędu jak  $C(x)$ .

Resztę z dzielenia  $B(x)$  przez  $C(x)$  otrzymuje się odejmując  $C(x)$  od  $B(x)$  – operację tą przeprowadza się przez bitowy XOR  $B(x)$  i  $C(x)$ .

Przykład:

$$B(x) = x^3 + 1$$

$$C(x) = x^3 + x^2 + 1$$

Podzielne (ten sam rząd).

$$\text{Reszta} = 0 \times x^3 + 1 \times x^2 + 0 \times x^1 + 0 \times x^0 = x^2$$

## Realizacja CRC

Jak na podstawie  $M(x)$  stworzyć wielomian podzielny bez reszty przez  $C(x)$ ?

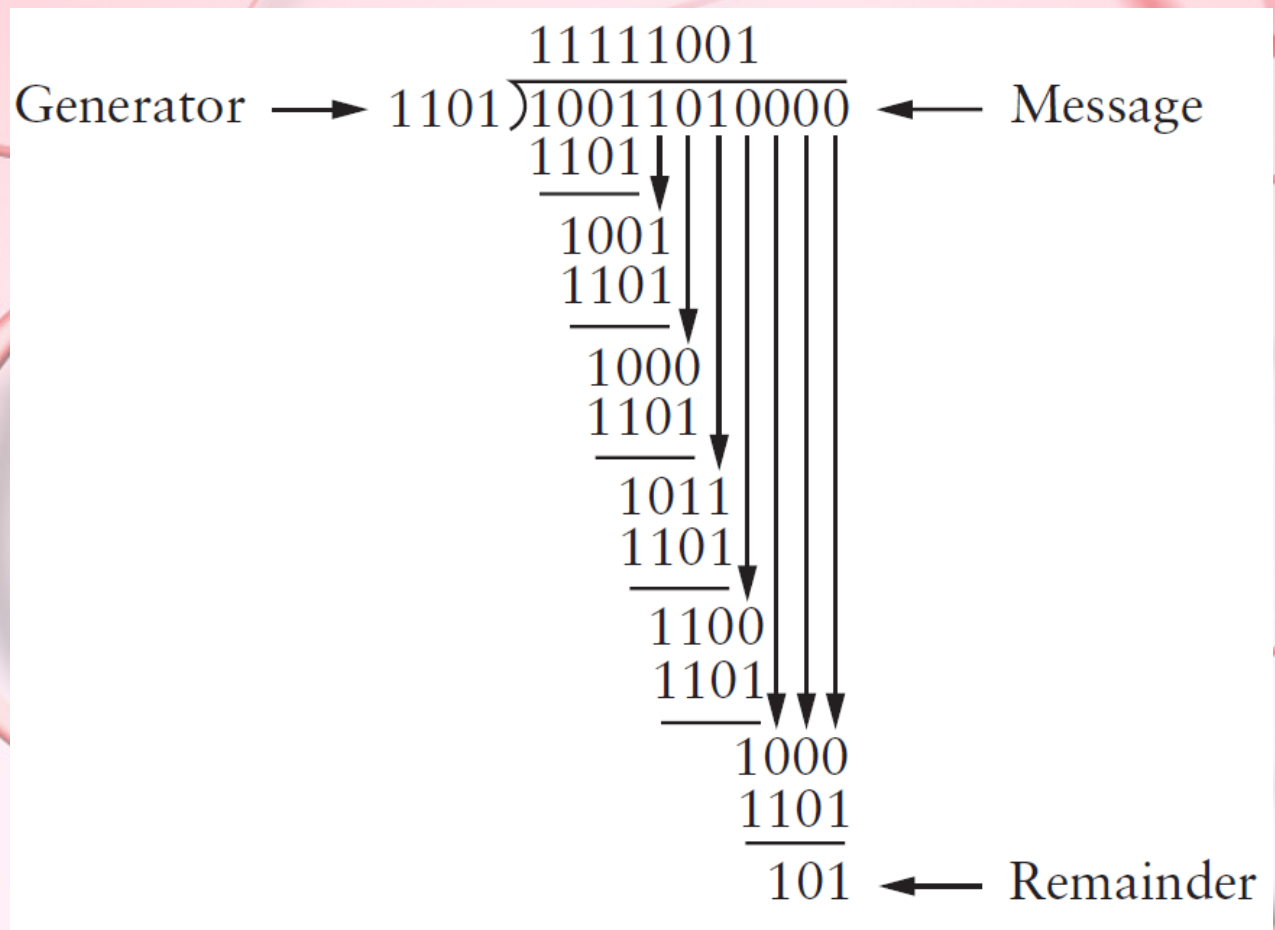
1. Wymnóż  $M(x)$  przez  $x^k$ , tj. dodaj zera na końcu wiadomości, otrzymując nowy wielomian  $T(x)$ .
2. Podziel  $T(x)$  przez  $C(x)$  i znajdź resztę
3. Odejmij resztę od  $T(x)$

Otrzymany wielomian jest bez reszty podzielny przez  $C(x)$ .

# CRC - przykład

$M(x) = x^7 + x^4 + x^3 + x^1$  tj. 10011010

$C(x) = C(x) = x^3 + x^2 + 1$  tj. 1101,  $T(x) = 10011010000$



## CRC – dzielniki

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	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11}$ $+ x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

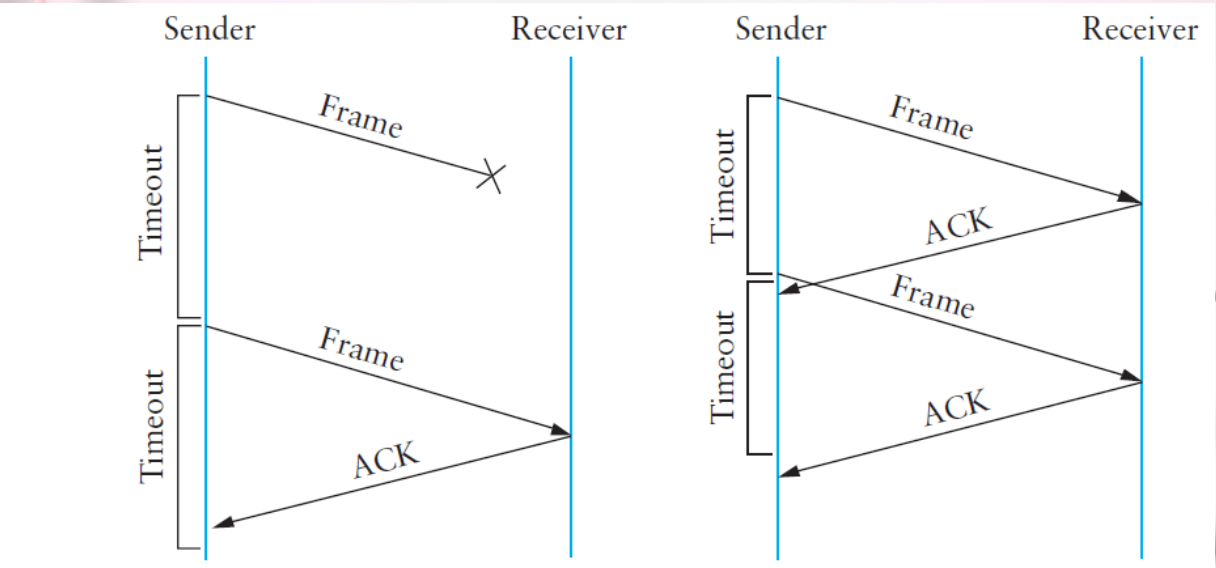
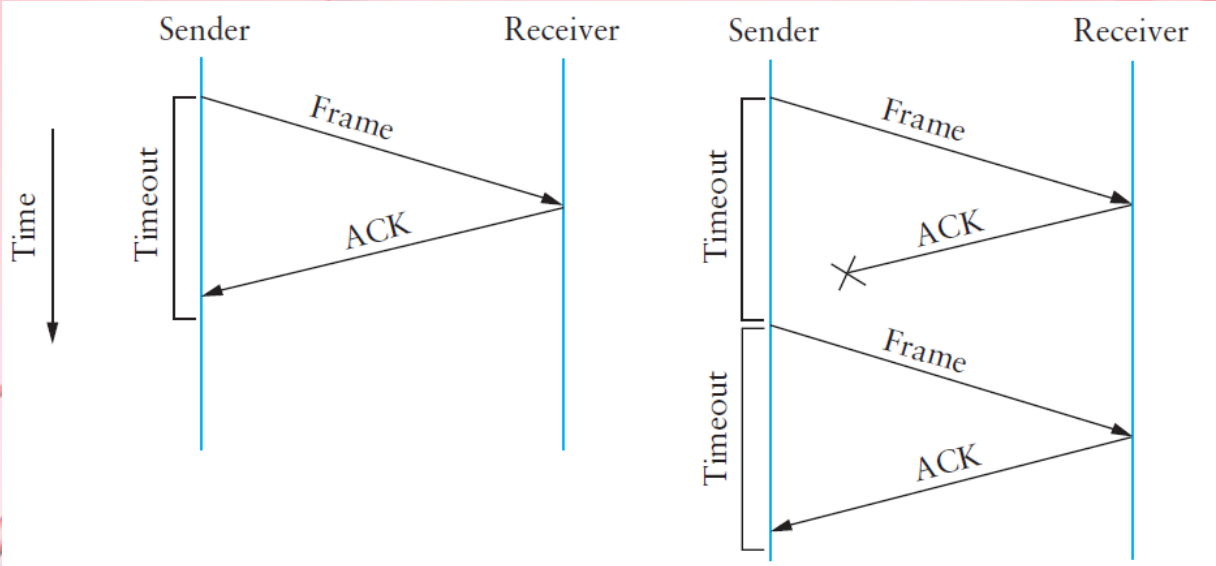
## Niezawodna transmisja

Aby zapewnić niezawodność transmisji używa się potwierdzeń (ang. acknowledgements, ACK) i czasów przekroczenia operacji (ang. timeout).

Algorytmy które wykorzystują te elementy w celu niezawodnego przesyłania komunikatów (tzw. ARQ tj. *automatic repeat request*) to między innymi:

- stop-and-wait
- sliding window

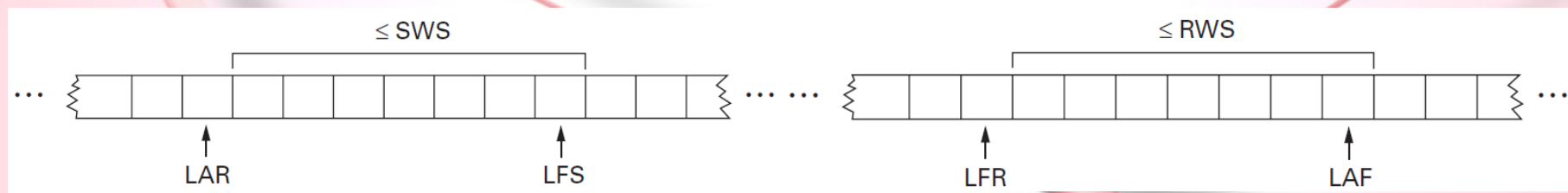
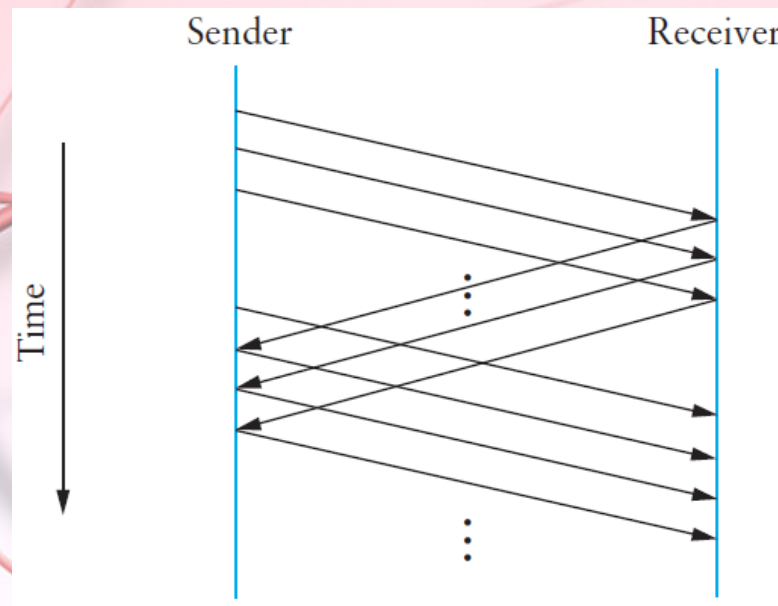
# Stop-and-wait



# Sliding window

SWS, RWS – Send/Receive Window Size

LAR, LFS – Last ACK/Frame Received/Send, LAF – Last Acceptable Frame



Nadawca ( $LFS - LAR \leq SWS$ )

Odbiorca ( $LAF - LFR \leq RWS$ )

**DZIĘKUJĘ ZA UWAGĘ!**

NASTĘPNY WYKŁAD:

**SIECI ETHERNET**