$$C_{60dB} = 0{,}333 \cdot 3000 \cdot 60 = 60 \text{ kbps}$$

Z twierdzenia Nyquista wynika konieczność kodowania bitów za pomocą sygnałów w celu przesłania większej liczby bitów w jednostce czasu.

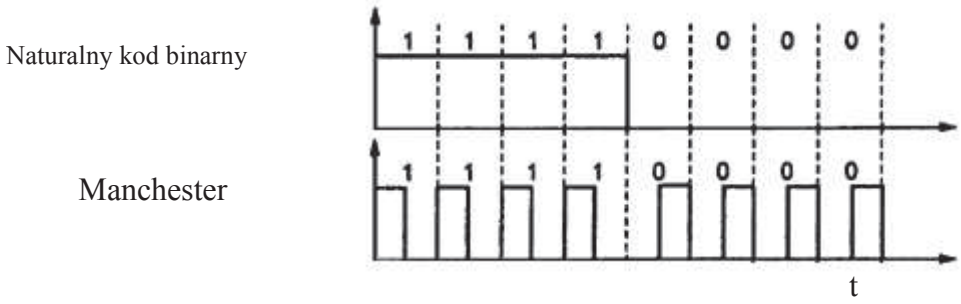
Z twierdzenia Shannona wynika, że żadna metoda kodowania informacji nie pozwala na złamanie praw fizyki, które stawiają ograniczenia liczby bitów, które można przesłać w ciągu sekundy za pomocą konkretnego systemu komunikacyjnego. W praktyce określa ono, jak szybko można przesłać dane przez łącze telefoniczne. Systemy telefonii analogowej zapewniają stosunek sygnału do szumu na poziomie 30 dB i pasmo około 3000 Hz. Maksymalna liczba bitów, które można przesłać w ciągu sekundy jest ograniczona do:

$$C_{30dB} = 3000\log_2(1 + 1000) \qquad \text{czyli } 30000 \text{ bps}$$

2.9.Algorytmy kodowania danych

2.9.1 Manchester

Zasada działania kodu Manchester polega na zmianie poziomu sygnału w środku każdego bitu sygnału wejściowego. Bitowi „1” odpowiada zmiana poziomu od wyższego do niższego, a „0” - od niższego do wyższego:



Przejście między poziomami sygnału występują przy każdym bicie, w związku z czym możliwa jest ciągła kontrola synchronizacji detektora ze strumieniem danych, nawet w przypadku nadawania długiej sekwencji zer lub jedynek. Fakt ten może być również wykorzystywany do detekcji błędów – brak oczekiwanej zmiany poziomu sygnału oznacza przekłamanie. Kod Manchester wymaga impulsów dwukrotnie krótszych niż kod NRZ. Oznacza to dwukrotne zwiększenie szybkości modulacji, a więc i dwukrotny wzrost wymaganego pasma transmisyjnego przy tej samej szybkości transmisji danych. Korzystną cechą sygnału przesyłanego w kodzie Manchester jest fakt, że jego wartość średnia jest równa zero.

2.9.2 4B/5B

Kodowanie 4B/5B zostało zaprojektowane oryginalnie na potrzeby sieci FDDI, gdzie pozwoliło na 80% wykorzystanie przepustowości łącza. Zaadaptowano je do standardu 100BaseTX, gdzie służy jako wstępny skrambler danych przed kodowaniem MLT-3. Zabieg ten ma na celu zapobieganie powstawaniu długich ciągów logicznych zer, co skutkowałoby utratą synchronizacji (patrz kodowanie MLT-3). Kodowanie zostało zmienione jedynie w nieznacznym stopniu w stosunku do wersji FDDI, w celu uwzględnienia kontroli ramek Ethernet.

W kodowaniu 4B/5B ciągi czterobitowe kodowane są pięciobitowymi symbolami. Do każdych czterech bitów dodawany jest piąty – za pomocą 4 bitów można utworzyć $2^4 = 16$ ciągów, natomiast pięć bitów daje ich już $2^5 = 32$. Analizując zamieszczoną tabelę kodową można zauważyć, że uzyskana w ten sposób nadmiarowość umożliwia takie zakodowanie sygnału, że nawet ciąg samych zer będzie zawierał jedynkę (i analogicznie ciąg samych jedynek będzie zawierał zero), co zapewnia utrzymanie synchronizacji. Poniższa tabela przedstawia wszystkie możliwe ciągi zer i jedynek wraz z ich interpretacją:

	PCS code-group [4:0] 4 3 2 1 0	Name	MII (TXD/RXD) <3:0> 3 2 1 0	Interpretation
DATA	1 1 1 1 0	0	0 0 0 0	Data 0
	0 1 0 0 1	1	0 0 0 1	Data 1
	1 0 1 0 0	2	0 0 1 0	Data 2
	1 0 1 0 1	3	0 0 1 1	Data 3
	0 1 0 1 0	4	0 1 0 0	Data 4
	0 1 0 1 1	5	0 1 0 1	Data 5
	0 1 1 1 0	6	0 1 1 0	Data 6
	0 1 1 1 1	7	0 1 1 1	Data 7
	1 0 0 1 0	8	1 0 0 0	Data 8
	1 0 0 1 1	9	1 0 0 1	Data 9
	1 0 1 1 0	A	1 0 1 0	Data A
	1 0 1 1 1	B	1 0 1 1	Data B
	1 1 0 1 0	C	1 1 0 0	Data C
	1 1 0 1 1	D	1 1 0 1	Data D
	1 1 1 0 0	E	1 1 1 0	Data E
	1 1 1 0 1	F	1 1 1 1	Data F
	1 1 1 1 1	I	undefined	IDLE; used as inter-stream fill code
CONTROL	1 1 0 0 0	J	0 1 0 1	Start-of-Stream Delimiter, Part 1 of 2; always used in pairs with K
	1 0 0 0 1	K	0 1 0 1	Start-of-Stream Delimiter, Part 2 of 2; always used in pairs with J
	0 1 1 0 1	T	undefined	End-of-Stream Delimiter, Part 1 of 2; always used in pairs with R
	0 0 1 1 1	R	undefined	End-of-Stream Delimiter, Part 2 of 2; always used in pairs with T
INVALID	0 0 1 0 0	H	Undefined	Transmit Error; used to force signaling errors
	0 0 0 0 0	V	Undefined	Invalid code
	0 0 0 0 1	V	Undefined	Invalid code
	0 0 0 1 0	V	Undefined	Invalid code
	0 0 0 1 1	V	Undefined	Invalid code
	0 0 1 0 1	V	Undefined	Invalid code
	0 0 1 1 0	V	Undefined	Invalid code
	0 1 0 0 0	V	Undefined	Invalid code
	0 1 1 0 0	V	Undefined	Invalid code
	1 0 0 0 0	V	Undefined	Invalid code
	1 1 0 0 1	V	Undefined	Invalid code

W nadawanej sekwencji znaków nigdy nie wystąpi ciąg dłuższy niż 8 jedynek. Piąty bit w niewielkim zakresie umożliwia ponadto wykrywanie błędów. Wadą tego kodowania, np. w stosunku do 8B/10B, jest brak zrównoważenia wystąpień sygnałów 0 i 1, w związku z czym wymagana do zakodowania energia będzie większa w przypadku wysyłania większej liczby 1 niż 0. Należy zauważyć, że 25% nadmiarowość oznacza konieczność użycia zegara o odpowiednio wyższej częstotliwości, np. 125MHz przy 100Mb/s. Kod ten używany jest min. w standardach Fast Ethernet, FDDI czy HIPPI-6400

2.9.3 5B/6B

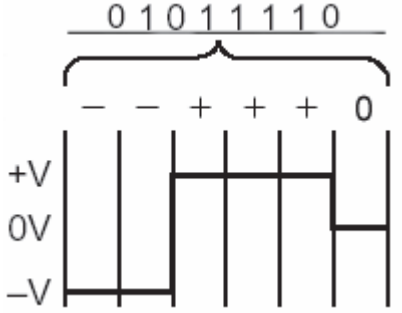
Zasada działania jest taka sama jak w przypadku kodowania 4B/5B. Dodatkowo wprowadzona została zasada równoważenia składowej stałej w celu zapobiegania polaryzacji sygnału (3 zera i 3 jedynki w każdej grupie sześciu bitów). Umożliwia to także prostsze wykrywanie błędów – niepoprawny jest każdy ciąg, w którym występuje więcej niż 3 zera lub 3 jedynki pod rząd.

Nadmiarowość wynosi tu 20% (co pięć bitów dodawany jeden dodatkowy). Oznacza to, że przy prędkości transmisji 100Mb/s, stosowany jest zegar o częstotliwości 120 MHz. Używany m.in. w 100VGAnyLAN.

2.9.4 8B/6T

Kodowanie 8B/6T zaprojektowane zostało w celu wykorzystania skrętki kategorii 3 do transmisji sygnału 100Mb/s. Kodowanie przebiega w ten sposób, że każdej sekwencji ośmiu bitów ze strumienia danych wejściowych przyporządkowany zostaje ciąg sześciu symboli trzystanowych (o trzech możliwych poziomach napięć: –V, 0, +V). Możliwych jest więc $3^6 = 729$ ciągów, z czego wykorzystywanych jest 2^8=256 ciągów. Ciągi kodowe zostały tak dobrane, aby zapewnić możliwość dobrej detekcji błędów, zmniejszyć efekty wysokoczęstotliwościowe oraz wyeliminować składową stałą. Przyjęto założenie, że w każdym ciągu muszą wystąpić co najmniej dwa poziomy napięć (niezbędne do celów synchronizacji). Ponadto mogą być używane specjalne ciągi kodowe, np. jako znaczniki.
Kodowanie wielopoziomowe umożliwia zakodowanie więcej niż jednego bitu informacji w pojedynczej zmianie poziomu – tym sposobem sygnał o częstotliwości 12,5MHz przenosi strumień danych o szybkości 33,3Mb/s. Każdy cykl sygnału 12,5MHz zawiera dwa poziomy, co daje 25 milionów zmian poziomów na sekundę na pojedynczej parze skrętki. Na trzech parach sumarycznie daje to 75 milionów zmian w każdej sekundzie. Dzieląc przez 6 symboli w każdym ciągu kodowym, otrzymujemy 12,5 miliona ciągów kodowych na sekundę, z których każdy odpowiada ośmiu bitom danych – daje to sygnał o szybkości 100Mb/s. Warto zauważyć, że częstotliwość 12,5MHz mieści się w limicie 16MHz dla skrętki kategorii 3.

Przykładowo, osiem bitów danych *01011110* zostanie zakodowane jako następujące sześć symboli: – – + + + 0 co zostało zilustrowane poniżej:

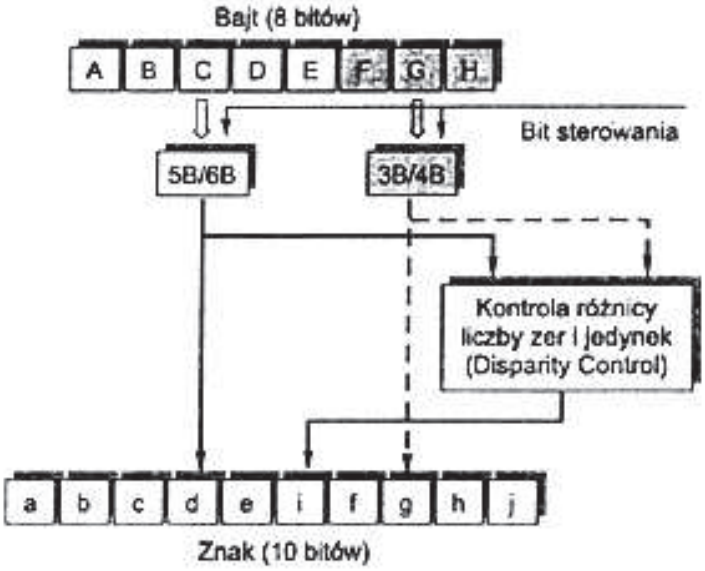


2.9.5 8B/10B

Aby możliwe było wiarygodne przesyłanie danych z prędkościami gigabitowymi i większymi (standardy Gigabit Ethernet czy 10 Gigabit Ethernet), konieczna jest kolejna zmiana w metodzie kodowania danych. Strumień napływających danych dzielony jest na bloki ośmiobitowe (kolejne bity oznaczone są *HGFEDCBA*, gdzie H – najbardziej znaczący bit, A – najmniej znaczący bit), do których następnie dodawane są dwa nadmiarowe bity w celu otrzymania dziesięciobitowego ciągu kodowego. Ponadto założono istnienie tzw. zmiennej sterującej (bit sterowania) – blok ośmiobitowy zawiera dane jeżeli zmienna ta ma wartość *D*, lub jest bajtem kontrolnym jeżeli ma wartość *K*. Kodowanie przebiega w ten sposób, że najpierw każde 8 bitów dzielone jest na 3 najbardziej znaczące bity (*HGF*) oraz 5 najmniej znaczących bitów (*EDCBA*). Następnie osiem bitów przekształcanych jest na dziesięć bitów o postaci *abcdeifghj*. 10-cio bitowe ciągi kodowe zostają tak dobrane, aby zawierały:

- 5 jedynek i 5 zer lub
- 4 jedynki i 6 zer lub
- 6 jedynek i 4 zera

Zapobiega to występowaniu dłuższych sekwencji takich samych bitów co ułatwia synchronizację. Kolejne bloki ośmiobitowe kodowane są w ten sposób, aby pierwszy miał więcej bitów 1, następny więcej bitów 0, itd. Proces kodowania przedstawia poniższy schemat:



Bajt niezakodowany	Bajt zakodowany
	9 → j
7 → H	8 → h
6 → G	7 → g
5 → F	6 → f
	5 → i
4 → E	4 → e
3 → D	3 → d
2 → C	2 → c
1 → B	1 → b
0 → A	0 → a

Sposób konwersji 8B/10B

Każdy blok ośmiobitowy można zapisać w postaci *Dxx.y* (bajt danych – ang. data character) lub *Kxx.y* (bajt kontrolny – ang. special character), gdzie *xx* to zapis dziesiętny pięciu najmniej znaczących bitów, a *y* pozostałych. Np. bajt *10100110* zostanie zapisany jako *D6.5*. Za pomocą bajtu kontrolnego oraz 3 bajtów danych można utworzyć tzw. zestawy uporządkowane (ang. Ordered Set) – oznaczające przykładowo początek (SOF – Start of Frame to K28.5 D21.5 D23.2 D23.2) i koniec ramki (EOF – End of Frame – K28.5 D10.4 D21.4 D21.4).
Ja już napisano kolejne bajty kodowane są tak, aby pierwszy zawierał więcej jedynek niż zer. Drugi zawiera więcej zer i jedynek, w trzecim występuje więcej jedynek itd. Liczba zer i jedynek w transmitowanym bajcie określona jest jako dysparytet (ang. running disparity, RD). Jeżeli liczba zer jest równa liczbie jedynek, wówczas mówimy o dysparytecie neutralnym. Jeżeli w bajcie przeważa liczba jedynek, wówczas mówimy o dysparytecie dodatnim (RD+), a jeżeli przeważa liczba zer to o dysparytecie ujemnym (RD-).
Wartość parametru RD dla podgrup określa się według następujących zasad:

- parametr RD jest dodatni (RD+), gdy liczba jedynek jest większa niż liczba zer oraz na końcu 6-bitowej podgrupy 000111 oraz 4-bitowej podgrupy 0011
- parametr RD jest ujemny (RD-), gdy liczba jedynek jest mniejsza niż liczba zer oraz na końcu 6-bitowej podgrupy 111000 oraz 4-bitowej podgrupy 1100
- w innych przypadkach wartość dysparytetu na końcu podgrupy jest taka sama jak na początku podgrupy.

Przed wysłaniem danych nadajnik dla każdego bajtu wyszukuje na podstawie bieżącej wartości RD odpowiedni wpis w tabeli. Wpis ten staje się grupą kodową dla danego bajtu. Po wysłaniu bajtu obliczona zostaje nowa wartość RD, która użyta zostanie do wysłania kolejnego bajtu. Dodatek B przedstawia wszystkie ciągi kodowe.

W kodzie 8B/10B nadmiarowość wynosi 25%, więc by uzyskać prędkość przesyłu danych 1Gb/s, faktyczna prędkość transmisji musi wynosić 1,25GHz.

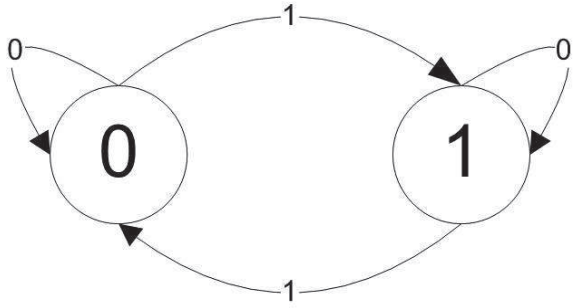
2.9.6 MLT-3

Jest to trójpoziomowy sygnał (Multi-Level Threshold) wykorzystywany do reprezentacji strumienia bitów zakodowanego jako 4B/5B (dla 100BaseTX). Zaprojektowany został z myślą o transmisji z prędkościami 100Mb/s i większymi. Jak już było powiedziane wcześniej, przy okazji 8B/6T, kodowanie wielopoziomowe umożliwia zakodowanie więcej niż jednego bitu informacji w pojedynczej zmianie poziomu. Uzyskuje się dzięki temu ograniczenie widma sygnału, lecz kosztem mniejszego odstępu sygnału od zakłóceń.
Najpierw każde 4 bity danych wejściowych zamieniane jest na 5-cio bitowy ciąg, zgodnie z kodem 4B/5B. Tym samym strumień danych o szybkości 100Mb/s zostaje zamieniony na 125Mb/s. Użycie MLT-3 pozwala na przenoszenie strumienia danych 125Mb/s, sygnałem o częstotliwości 31,25MHz.
MLT-3 używa trzech różnych poziomów napięć: -1, 0, +1. Kodowanie odbywa się według następujących reguł:

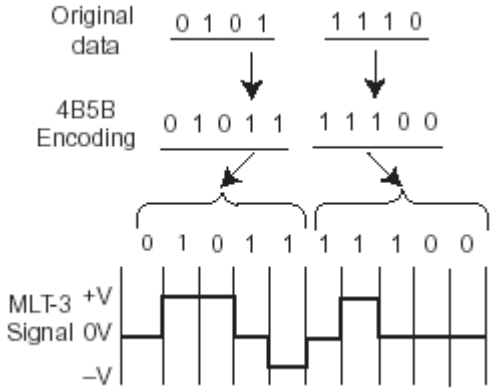
- Jeżeli następny bit wejściowy jest równy 0, to następna wartość wyjściowa jest taka sama, jak poprzednio.
- Jeżeli następny bit wejściowy jest równy 1, to nastąpi zmiana poziomu wartości wyjściowej:
  - Jeżeli wartość poprzednia była równa +1 lub –1, to następna wartość wyjściowa jest równa 0.

- Jeżeli wartość poprzednia była równa 0, to następna wartość wyjściowa będzie niezerowa, o znaku przeciwnym do ostatniej niezerowej wartości.

Na poniższym grafie stan 0 oznacza brak zmiany wartości wyjściowej, natomiast stan 1 oznacza zmianę wartości wyjściowej zgodnie z warunkiem podanym powyżej.

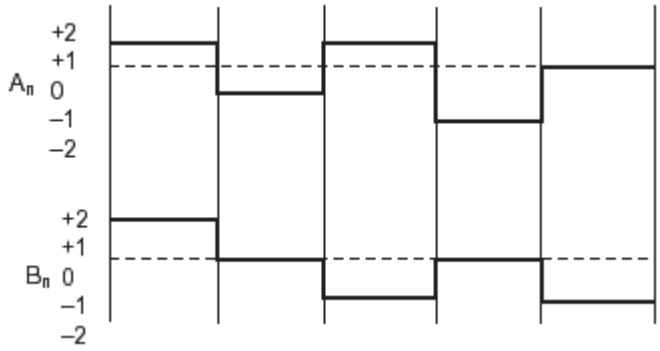


Innymi słowy poziom pozostaje niezmieniony dla logicznych zer, a jedynka oznacza zmianę poziomu. Zmiany następują wg cyklu 0, +1, 0, -1, 0, +1. Przykładowy ciąg danych zakodowany MLT-3:



2.9.7 PAM-5

W celu zaadaptowania dwuparowej skrętki kategorii 3 do większych szybkości transmisji, zastosowano kodowanie 5 level Pulse Amplitude Modulation. Jest to kolejny kod wielopoziomowy. W 100BaseT2 przesyłane są dwa 5-cio poziomowe sygnały PAM o częstotliwości 12,5MHz. Każdy cykl sygnału dostarcza dwóch zmian poziomów, jest więc 25 milionów zmian poziomów na sekundę na parę w skrętce. Każda z par sygnału PAM (A i B) koduje inny, 4-bitowy ciąg kodowy (25mln*4b=100Mb/s), przy użyciu pięciu różnych poziomów: -2, -1, 0, +1, +2 (odpowiednio: -1V, -0.5V, 0V, 0.5V, 1V). Poniżej widać przykładowy kod PAM-5:



W Gigabit Ethernecie zastosowano kodowanie PAM-5. Główną różnicą podczas transmisji sygnału pomiędzy 10/100 Mbps Ethernetem a Gigabit Ethernetem jest fakt, że 1000BASE-T wykorzystuje cztery pary do równoczesnego wysyłania i odbierania sygnału, podczas gdy w 10/100 Mbps Ethernecie używane są tylko dwie pary – jedna do nadawania i jedna do odbioru.

Zarówno MLT-3 jak i PAM-5 zostały zaprojektowane jako kody pseudookresowe, dzięki czemu składowa stała sygnału jest bliska lub równa zeru.