

DATA RATE LIMITS

A very important consideration in data communications is how fast we can send data, in bits per second, over a channel. Data rate depends on three factors:

1. The bandwidth available
2. The level of the signals we use
3. The quality of the channel (the level of noise)

Capacity of a System

- The bit rate of a system increases with an increase in the number of signal levels we use to denote a symbol.
- A symbol can consist of a single bit or “n” bits.
- The number of signal levels = 2^n .
- As the number of levels goes up, the spacing between level decreases -> increasing the probability of an error occurring in the presence of transmission impairments.

Nyquist Theorem

- Nyquist gives the upper bound for the bit rate of a transmission system by calculating the bit rate directly from the number of bits in a symbol (or signal levels) and the bandwidth of the system (assuming 2 symbols/per cycle and first harmonic).
- Nyquist theorem states that for a noiseless channel:

$$C = 2 B \log_2 2^n$$

C= capacity in bps

B = bandwidth in Hz

Example

Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with two signal levels. The maximum bit rate can be calculated as

$$\text{BitRate} = 2 \times 3000 \times \log_2 2 = 6000 \text{ bps}$$

Example

Consider the same noiseless channel transmitting a signal with four signal levels (for each level, we send 2 bits). The maximum bit rate can be calculated as

$$\text{BitRate} = 2 \times 3000 \times \log_2 4 = 12,000 \text{ bps}$$

Example:

We need to send 265 kbps over a noiseless channel with a bandwidth of 20 kHz. How many signal levels do we need?

Solution

We can use the Nyquist formula as shown:

$$265,000 = 2 \times 20,000 \times \log_2 L$$

$$\log_2 L = 6.625 \quad L = 2^{6.625} = 98.7 \text{ levels}$$

Shannon's Theorem

- Shannon's theorem gives the capacity of a system in the presence of noise.

$$C = B \log_2(1 + \text{SNR})$$

Example:

Consider an extremely noisy channel in which the value of the signal-to-noise ratio is almost zero. In other words, the noise is so strong that the signal is faint. For this channel the capacity C is calculated as

$$C = B \log_2(1 + \text{SNR}) = B \log_2(1 + 0) = B \log_2 1 = B \times 0 = 0$$

This means that the capacity of this channel is zero regardless of the bandwidth. In other words, we cannot receive any data through this channel.

Example:

A telephone line normally has a bandwidth of 3000. The signal-to-noise ratio is usually 3162. For this channel the capacity is calculated as

$$C = B \log_2(1 + \text{SNR}) = 3000 \log_2(1 + 3162) = 3000 \log_2 3163$$

$$= 3000 \times 11.62 = 34,860 \text{ bps}$$

Example:

The signal-to-noise ratio is often given in decibels. Assume that $\text{SNR}_{\text{dB}} = 36$ and the channel bandwidth is 2 MHz. The theoretical channel capacity can be calculated as

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \text{SNR} \quad \rightarrow \quad \text{SNR} = 10^{\text{SNR}_{\text{dB}}/10} \quad \rightarrow \quad \text{SNR} = 10^{3.6} = 3981$$

$$C = B \log_2(1 + \text{SNR}) = 2 \times 10^6 \times \log_2 3982 = 24 \text{ Mbps}$$

Example:

We have a channel with a 1-MHz bandwidth. The SNR for this channel is 63. What are the appropriate bit rate and signal level?

Solution

First, we use the Shannon formula to find the upper limit.

$$C = B \log_2 (1 + \text{SNR}) = 10^6 \log_2 (1 + 63) = 10^6 \log_2 64 = 6 \text{ Mbps}$$

The Shannon formula gives us 6 Mbps, the upper limit. For better performance we choose something lower, 4 Mbps, for example. Then we use the Nyquist formula to find the number of signal levels.

$$4 \text{ Mbps} = 2 \times 1 \text{ MHz} \times \log_2 L \quad \rightarrow \quad L = 4$$

The Shannon capacity gives us the upper limit; the Nyquist formula tells us how many signal levels we need.

////////////////////////////////////

PERFORMANCE

One important issue in networking is the performance of the network.

- Bandwidth - capacity of the system
- Throughput - no. of bits that can be pushed through
- Latency (Delay) - delay incurred by a bit from start to finish
- Bandwidth-Delay Product

In networking, we use the term bandwidth in two contexts.

- The first, bandwidth in hertz, refers to the range of frequencies in a composite signal or the range of frequencies that a channel can pass.
- The second, bandwidth in bits per second, refers to the speed of bit transmission in a channel or link. Often referred to as Capacity.

Example:

A network with bandwidth of 10 Mbps can pass only an average of 12,000 frames per minute with each frame carrying an average of 10,000 bits. What is the throughput of this network?

Solution

We can calculate the throughput as

$$\text{Throughput} = \frac{12,000 \times 10,000}{60} = 2 \text{ Mbps}$$

Propagation & Transmission delay

- Propagation speed - speed at which a bit travels through the medium from source to destination.
- Transmission speed - the speed at which all the bits in a message arrive at the destination. (difference in arrival time of first and last bit)
 - Propagation Delay = Distance/Propagation speed
 - Transmission Delay = Message size/bandwidth bps
 - Latency = Propagation delay + Transmission delay + Queueing time + Processing time

Example:

What is the propagation time if the distance between the two points is 12,000 km? Assume the propagation speed to be 2.4×10^8 m/s in cable.

Solution

We can calculate the propagation time as

$$\text{Propagation time} = \frac{12,000 \times 1000}{2.4 \times 10^8} = 50 \text{ ms}$$

Example:

What are the propagation time and the transmission time for a 2.5-kbyte message (an e-mail) if the bandwidth of the network is 1 Gbps? Assume that the distance between the sender and the receiver is 12,000 km and that light travels at 2.4×10^8 m/s.

Solution

$$\begin{aligned} \text{Propagation time} &= \frac{12,000 \times 1000}{2.4 \times 10^8} = 50 \text{ ms} \\ \text{Transmission time} &= \frac{2500 \times 8}{10^9} = 0.020 \text{ ms} \end{aligned}$$

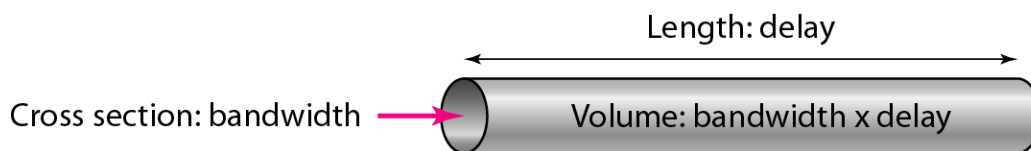
Example:

What are the propagation time and the transmission time for a 5-Mbyte message (an image) if the bandwidth of the network is 1 Mbps? Assume that the distance between the sender and the receiver is 12,000 km and that light travels at 2.4×10^8 m/s.

Solution

$$\text{Propagation time} = \frac{12,000 \times 1000}{2.4 \times 10^8} = 50 \text{ ms}$$
$$\text{Transmission time} = \frac{5,000,000 \times 8}{10^6} = 40 \text{ s}$$

The bandwidth-delay product defines the number of bits that can fill the link.



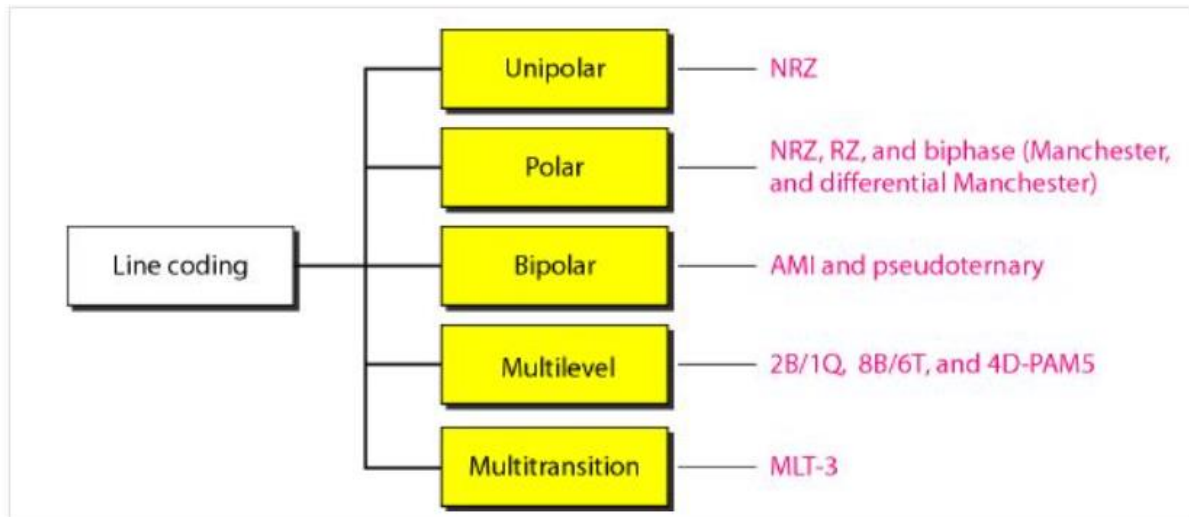
////////////////////////////////////

Line Coding

Line coding refers to the process of converting digital data into digital signals. Whenever we transmit data it is in the form of digital signals, so with the help of line coding, we can convert a sequence to bits (or encoding) into a digital signal which then again converted into bits by the receiver (or can be said as decoded by the receiver). For all this to happen we need line coding schemes which could also be able to avoid overlapping and distortion of signals.

Some necessary characteristics of line coding schemes:

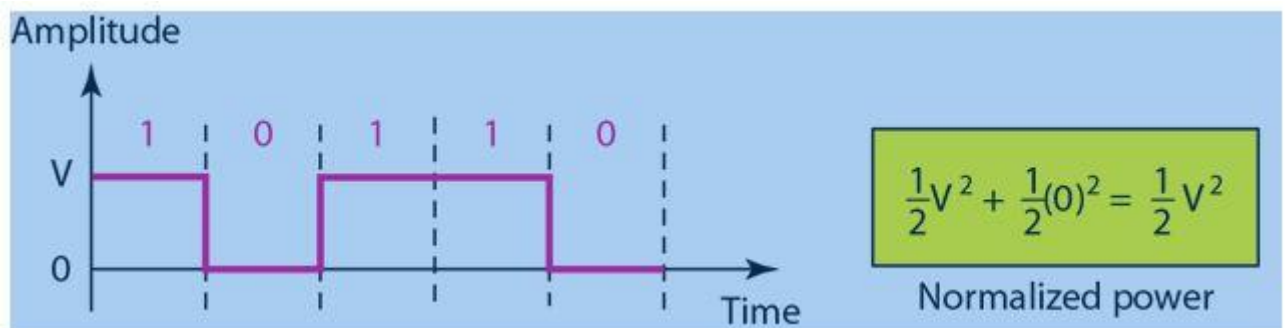
- Less complexity.
- Should have noise and interference tolerance.
- No DC component (or say low-frequency component) should be there because it can't be transferred to larger distances.
- Least baseline wandering should be there (baseline wander: low-frequency noise having nonlinear and non-stationary nature).
- Should have error detection capability.
- Should be self-synchronized.



I. Unipolar Scheme:

In a unipolar scheme, all the signal levels are on one side of the time axis, either above or below.

NRZ (Non-Return-to-Zero): Traditionally, a unipolar scheme was designed as a non-return-to-zero (NRZ) scheme in which the positive voltage defines bit 1 and the zero voltage defines bit 0. It is called NRZ because the signal does not return to zero at the middle of the bit. The following figure shows a unipolar NRZ scheme.



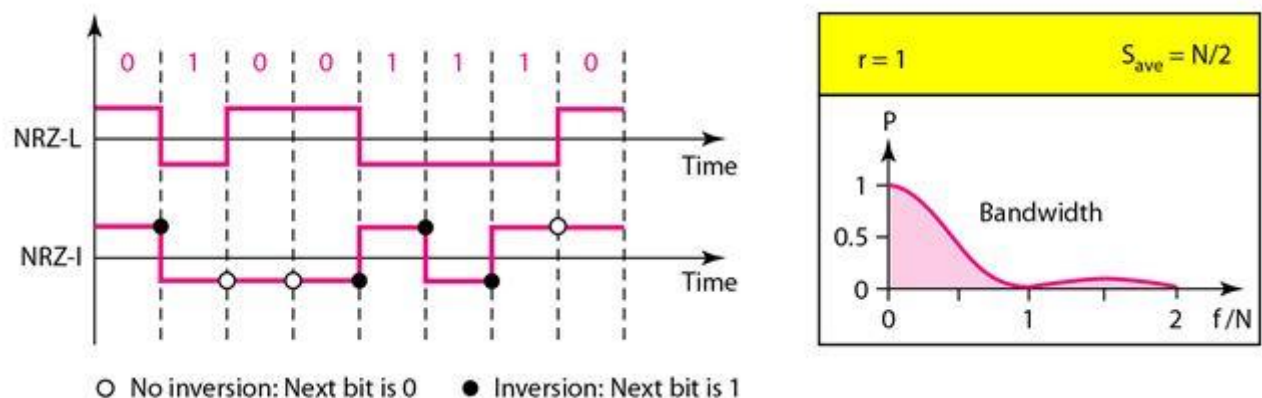
Compared with its polar counterpart, the normalized power (power needed to send 1 bit per unit line resistance) is double that for polar NRZ. For this reason, this scheme is normally not used in data communications today.

II. Polar Schemes

In polar schemes, the voltages are on the both sides of the time axis. For example, the voltage level for 0 can be positive and the voltage level for 1 can be negative.

a). Non-Return-to-Zero (NRZ):

In polar NRZ encoding, we use two levels of voltage amplitude. We can have two versions of polar NRZ: NRZ-L and NRZ-I, as shown in the following Figure. The figure also shows the value of r , the average baud rate, and the bandwidth.



In the first variation, NRZ-L (NRZ-Level), the level of the voltage determines the value of the bit. In the second variation, NRZ-I (NRZ-Invert), the change or lack of change in the level of the voltage determines the value of the bit. If there is no change, the bit is 0; if there is a change, the bit is 1.

Drawbacks:

1. Baseline wandering is a problem for both variations; it is twice as severe in NRZ-L. If there is a long sequence of 0s or 1s in NRZ-L, the average signal power becomes skewed. The receiver might have difficulty discerning the bit value.

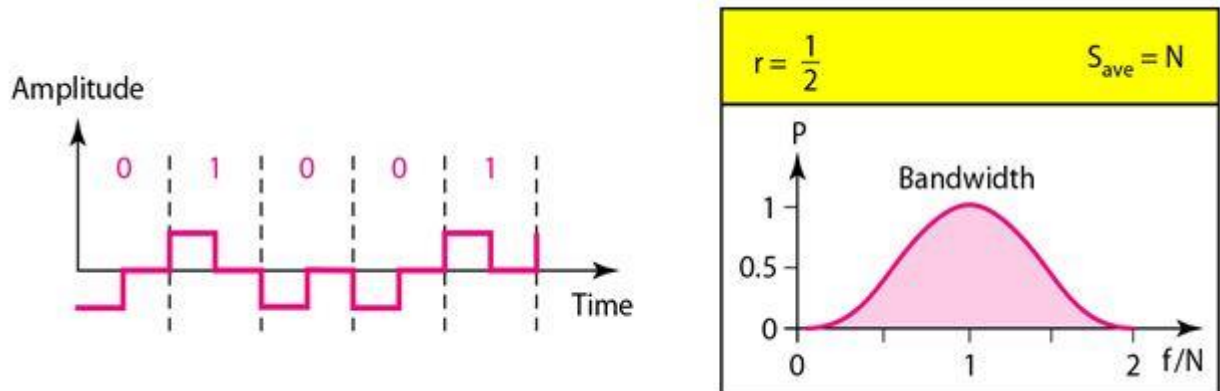
In NRZ-I this problem occurs only for a long sequence of 0s. If we eliminate the long sequence of 0s, we can avoid baseline wandering.

2. The synchronization problem (sender and receiver clocks are not synchronized) also exists in both schemes. Again, this problem is more serious in NRZ-L than in NRZ-I. While a long sequence of 0s can cause a problem in both schemes, a long sequence of 1s affects only NRZ-L.

3. Another problem with NRZ-L occurs when there is a sudden change of polarity in the system. For example, if twisted-pair cable is the medium, a change in the polarity of the wire results in all 0s interpreted as 1s and all 1s interpreted as 0s. NRZ-I does not have this problem. Both schemes have an average signal rate of $N/2$.

b). Return to Zero (RZ):

The main problem with NRZ encoding occurs when the sender and receiver clocks are not synchronized. The receiver does not know when one bit has ended and the next bit is starting. One solution is the return-to-zero (RZ) scheme, which uses three values: positive, negative, and zero. In RZ, the signal changes not between bits but during the bit. In the following figure, we see that the signal goes to 0 in the middle of each bit. It remains there until the beginning of the next bit.



Drawbacks:

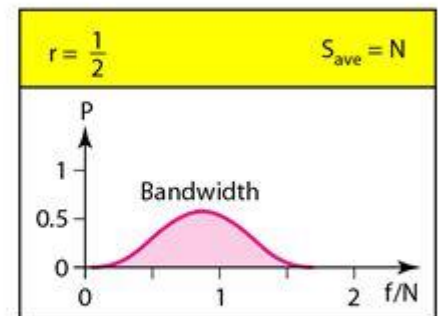
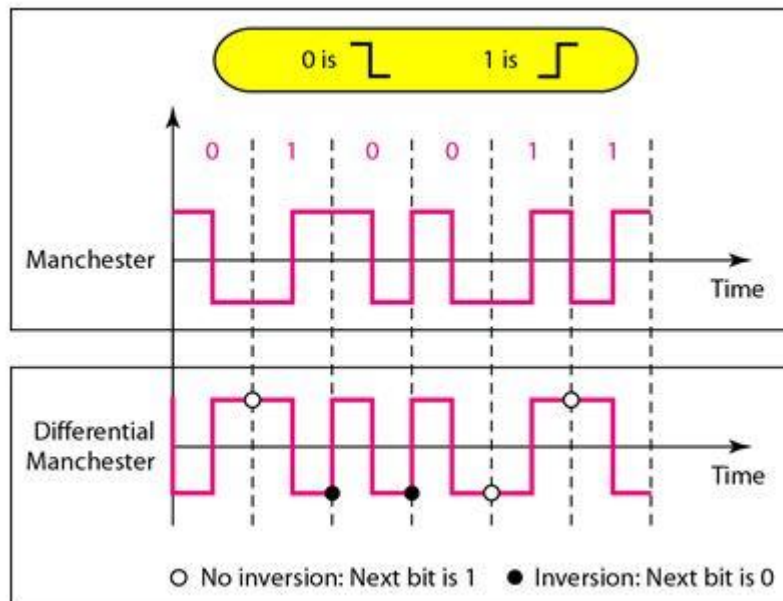
1. The main disadvantage of RZ encoding is that it requires two signal changes to encode a bit and therefore occupies greater bandwidth.
2. Sudden change of polarity resulting in all 0s interpreted as 1s and all 1s interpreted as 0s, still exist here, but there is no DC component problem.
3. Another problem is the complexity: RZ uses three levels of voltage, which is more complex to create and discern. As a result of all these deficiencies, the scheme is not used today.

Biphase Manchester and Differential Manchester:

The idea of RZ (transition at the middle of the bit) and the idea of NRZ-L are combined into the Manchester scheme.

In Manchester encoding, the duration of the bit is divided into two halves. The voltage remains at one level during the first half and moves to the other level in the second half. The transition at the middle of the bit provides synchronization.

Differential Manchester, on the other hand, combines the ideas of RZ and NRZ-I. There is always a transition at the middle of the bit, but the bit values are determined at the beginning of the bit. If the next bit is 0, there is a transition; if the next bit is 1, there is none. The following figure shows both Manchester and differential Manchester encoding.

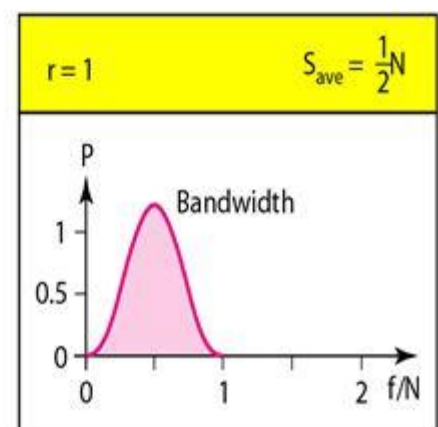
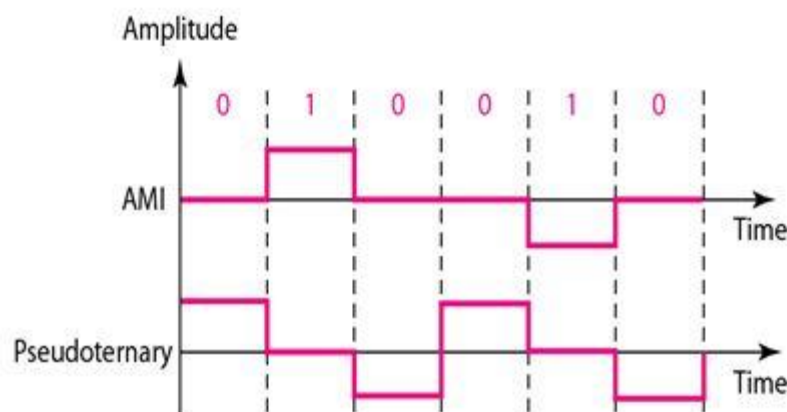


The Manchester scheme overcomes several problems associated with NRZ-L, and differential Manchester overcomes several problems associated with NRZ-I. First, there is no baseline wandering. There is no DC component because each bit has a positive and negative voltage contribution. The only drawback is the signal rate. The signal rate for Manchester and differential Manchester is double that for NRZ. The reason is that there is always one transition at the middle of the bit and maybe one transition at the end of each bit.

III. Bipolar Schemes

In bipolar encoding (sometimes called multilevel binary), there are three voltage levels, positive, negative, and zero. The voltage level for one data element is at zero, while the voltage level for the other element alternates between positive and negative.

AMI and Pseudoternary: The following figure shows two variations of bipolar encoding: AMI and pseudo ternary.



A common bipolar encoding scheme is called bipolar alternate mark inversion (AMI). In alternate mark inversion, a neutral zero voltage represents binary 0. Binary 1s are represented by alternating positive and negative voltages.

A variation of AMI encoding is called Pseudoternary in which the 1 bit is encoded as a zero voltage and the 0 bit is encoded as alternating positive and negative voltages.

The bipolar scheme was developed as an alternative to NRZ. The bipolar scheme has the same signal rate as NRZ, but there is no DC component. The NRZ scheme has most of its energy concentrated near zero frequency, which makes it unsuitable for transmission over channels with poor performance around this frequency. The concentration of the energy in bipolar encoding is around frequency $N/2$.

IV. Multilevel Schemes:

The desire to increase the data speed or decrease the required bandwidth has resulted in the creation of many schemes. The goal is to increase the number of bits per baud by encoding a pattern of m data elements into a pattern of n signal elements. We only have two types of data elements (0s and 1s), which means that a group of m data elements can produce a combination of 2^m data patterns.

We can have different types of signal elements by allowing different signal levels. If we have L different levels, then we can produce L^n combinations of signal patterns.

If $2^m = L^n$, then each data pattern is encoded into one signal pattern. If $2^m < L^n$, data patterns occupy only a subset of signal patterns. The subset can be carefully designed to prevent baseline wandering, to provide synchronization, and to detect errors that occurred during data transmission. Data encoding is not possible if $2^m > L^n$ because some of the data patterns cannot be encoded.

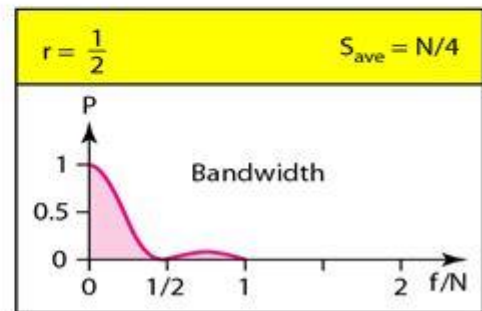
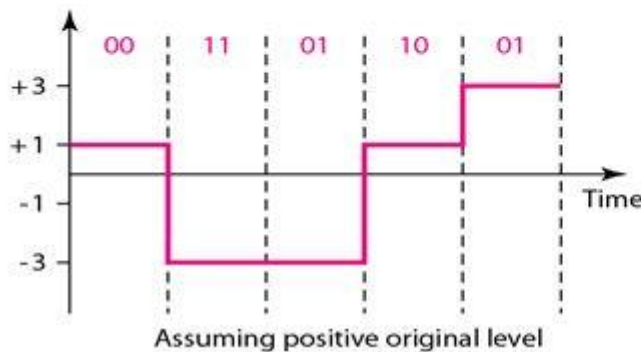
The code designers have classified these types of coding as $mBnL$, where m is the length of the binary pattern, B means binary data, n is the length of the signal pattern, and L is the number of levels in the signaling. A letter is often used in place of L : B (binary) for $L=2$, T (ternary) for $L=3$, and Q (quaternary) for $L=4$. Note that the first two letters define the data pattern, and the second two define the signal pattern.

a). 2BIQ:

The first $mBnL$ scheme we discuss, two binary, one quaternary (2BIQ), uses data patterns of size 2 and encodes the 2-bit patterns as one signal element belonging to a four-level signal. In this type of encoding $m=2$, $n=1$, and $L=4$ (quaternary). The following figure shows an example of a 2BIQ signal.

	Previous level: positive	Previous level: negative
Next bits	Next level	Next level
00	+1	-1
01	+3	-3
10	-1	+1
11	-3	+3

Transition table



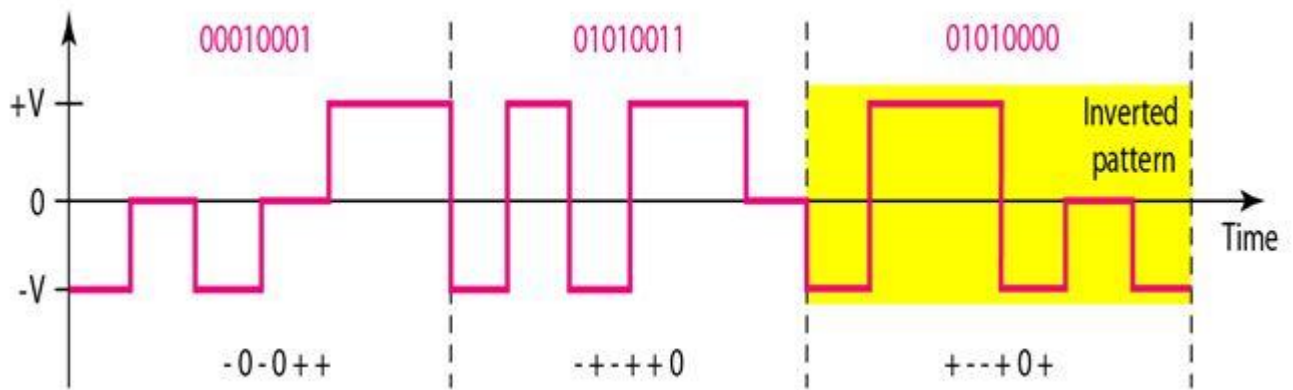
The average signal rate of 2B1Q is $S = N/4$. This means that using 2B1Q, we can send data 2 times faster than by using NRZ-L. However, 2B1Q uses four different signal levels, which means the receiver has to discern four different thresholds. The reduced bandwidth comes with a price. There are no redundant signal patterns in this scheme because $2^2=4$.

b). 8B6T:

A very interesting scheme is eight binary, six ternary (8B6T). The idea is to encode a pattern of 8 bits as a pattern of 6 signal elements, where the signal has three levels (ternary). In this type of scheme, we can have $2^8=256$ different data patterns and data patterns and $3^6=729$ different signal patterns.

There are $729 - 256 = 473$ redundant signal elements that provide synchronization and error detection. Part of the redundancy is also used to provide DC balance. Each signal pattern has a weight of 0 or +1 DC values. This means that there is no pattern with the weight -1. To make the whole stream Dc-balanced, the sender keeps track of the weight. If two groups of weight 1 are encountered one after another, the first one is sent as is, while the next one is totally inverted to give a weight of -1. The following figure shows an example of three data patterns encoded as three signal patterns.

The three possible signal levels are represented as -, 0, and +. The first 8-bit pattern 00010001 is encoded as the signal pattern -0-0++ with weight 0; the second 8-bit pattern 010 10011 is encoded as - + - + 0 with weight +1. The third bit pattern should be encoded as + - - + 0 + with weight +1. To create DC balance, the sender inverts the actual signal. The receiver can easily recognize that this is an inverted pattern because the weight is -1. The pattern is inverted before decoding.



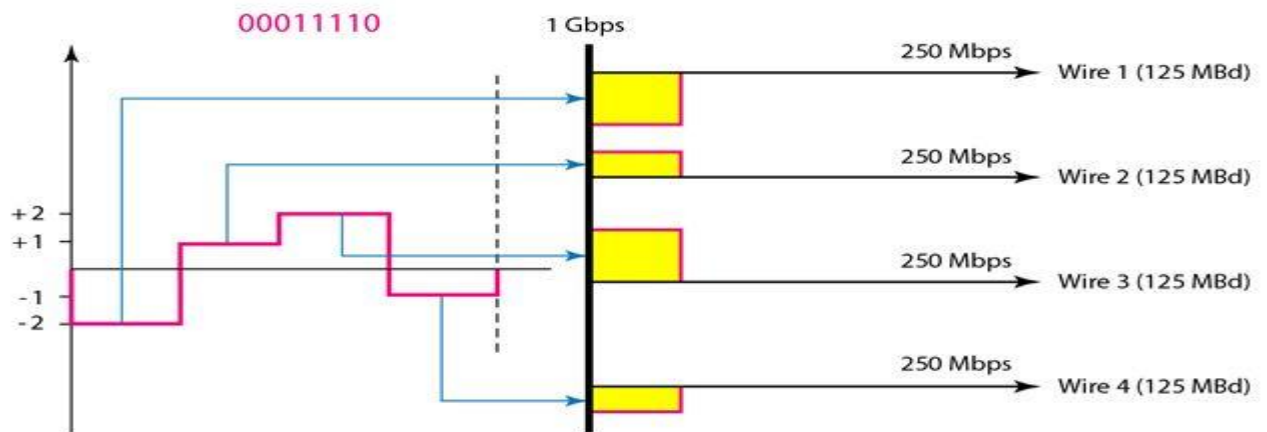
C).4D-PAM5:

The last signaling scheme we discuss in this category is called four dimensional five-level pulse amplitude modulation (4D-PAM5). The 4D means that data is sent over four wires at the same time. It uses five voltage levels, such as -2, -1, 0, 1, and 2.

However, one level, level 0, is used only for forward error detection. If we assume that the code is just one-dimensional, the four levels create something similar to 8B4Q. In other words, an 8-bit word is translated to a signal element of four different levels. The worst signal rate for this imaginary one-dimensional version is $N \times 4/8$, or $N/2$.

The technique is designed to send data over four channels (four wires). This means the signal rate can be reduced to $N/8$, a significant achievement. All 8 bits can be fed into a wire simultaneously and sent by using one signal element. The point here is that the four signal elements comprising one signal group are sent simultaneously in a four-dimensional setting.

The following figure shows the imaginary one-dimensional and the actual four-dimensional implementation. Gigabit LANs use this technique to send 1-Gbps data over four copper cables that can handle 125 Mbaud. This scheme has a lot of redundancy in the signal pattern because 28 data patterns are matched to $4^4 = 256$ signal patterns. The extra signal patterns can be used for other purposes such as error detection.

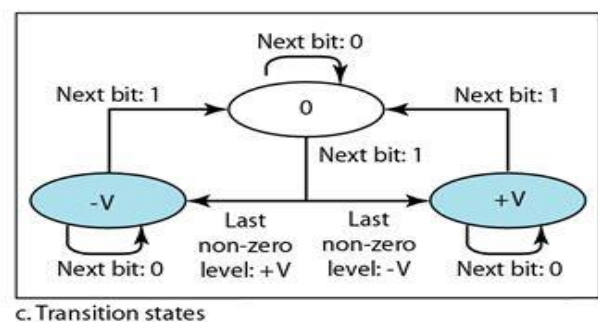
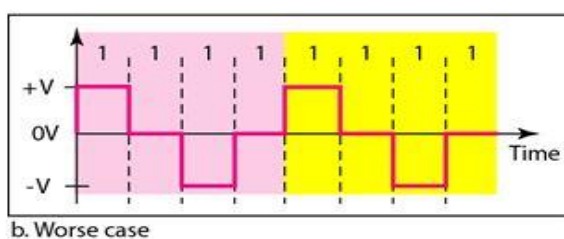
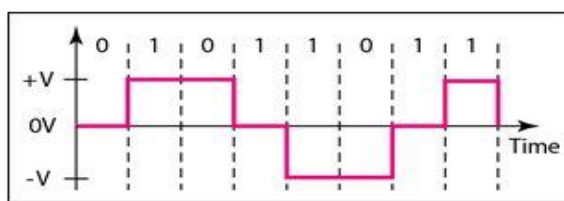


V. Multiline Transmission: MLT-3:

NRZ-I and differential Manchester are classified as differential encoding but use two transition rules to encode binary data (no inversion, inversion). If we have a signal with more than two levels, we can design a differential encoding scheme with more than two transition rules. MLT-3 is one of them. The multiline transmission, three level (MLT-3) scheme uses three levels (+V, 0 and -V) and three transition rules to move between the levels.

1. If the next bit is 0, there is no transition.
2. If the next bit is 1 and the current level is not 0, the next level is 0.
3. If the next bit is 1 and the current level is 0, the next level is the opposite of the last nonzero level.

The behavior of MLT-3 can best be described by the state diagram shown in the following figure. The three voltage levels (-V, 0, and +V) are shown by three states (ovals). The transition from one state (level) to another is shown by the connecting lines. The following figure also shows two examples of an MLT-3 signal.



The signal rate is the same as that for NRZ-I, but with greater complexity (three levels and complex transition rules). It turns out that the shape of the signal in this scheme helps to reduce the required bandwidth. Let us look at the worst-case scenario, a sequence of Is. In this case, the signal element pattern +VO - VO is repeated every 4 bits.

A non-periodic signal has changed to a periodic signal with the period equal to 4 times the bit duration. This worst-case situation can be simulated as an analog signal with a frequency one-fourth of the bit rate. In other words, the signal rate for MLT-3 is one-fourth the bit rate. This makes MLT-3 a suitable choice when we need to send 100 Mbps on a copper wire that cannot support more than 32 MHz (frequencies above this level create electromagnetic emissions).

////////////////////////////////////

BLOCK CODING

We need redundancy to ensure synchronization and to provide some kind of inherent error detecting. Block coding can give us this redundancy and improve the performance of line coding. block coding changes a block of m bits into a block of n bits, where n is larger than m . Block coding is referred to as an mB/nB encoding technique.

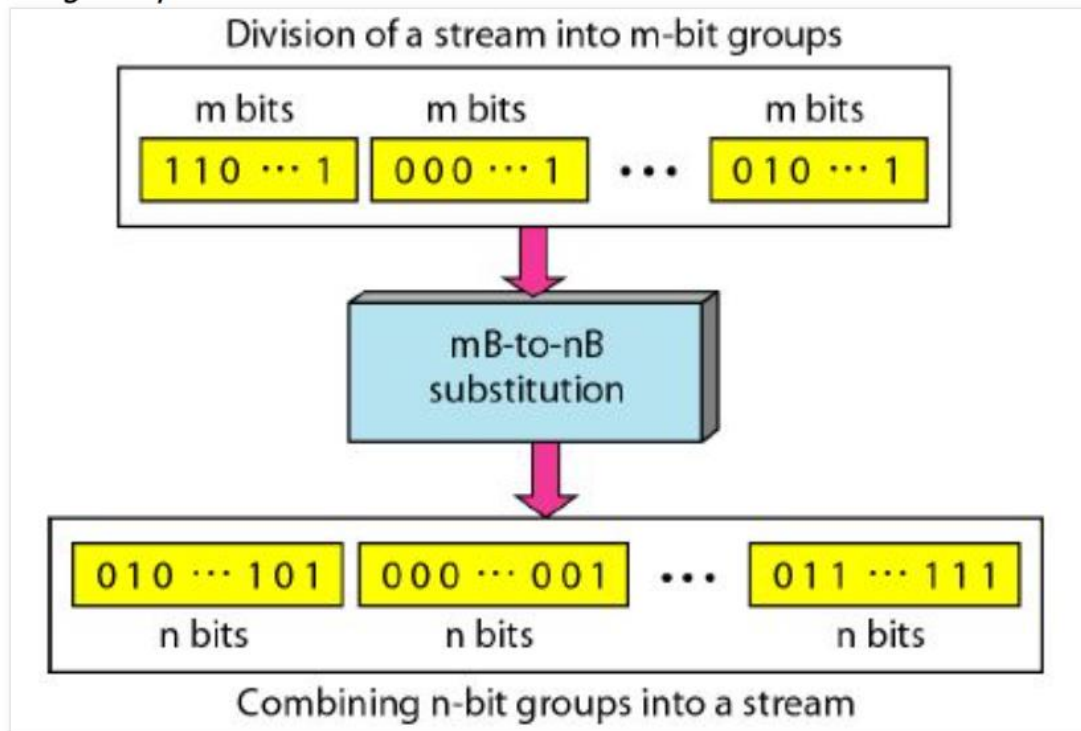
Block coding normally involves three steps: division, substitution,

1. and combination.
2. In the division step, a sequence of bits is divided into groups of m bits.
3. In the substitution step, we substitute an m -bit group for an n -bit group.
4. Finally, the n -bit groups are combined together to form a stream.

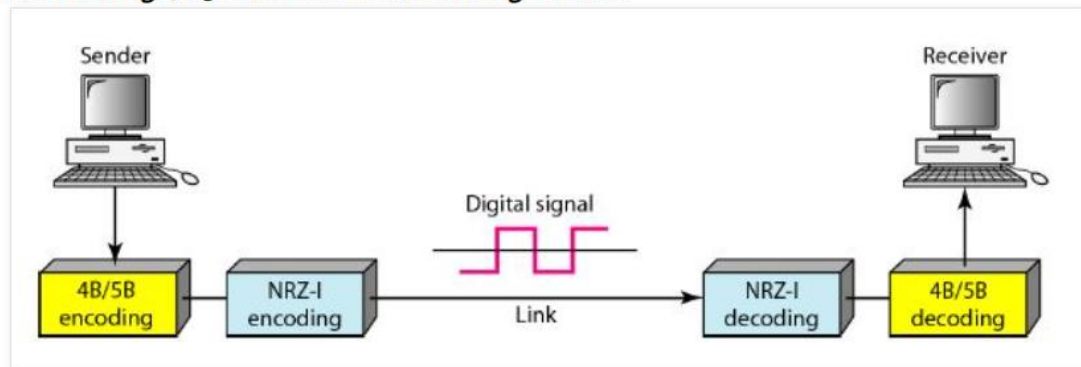
Note

Block coding is normally referred to as mB/nB coding; it replaces each m-bit group with an n-bit group.

Block coding concept



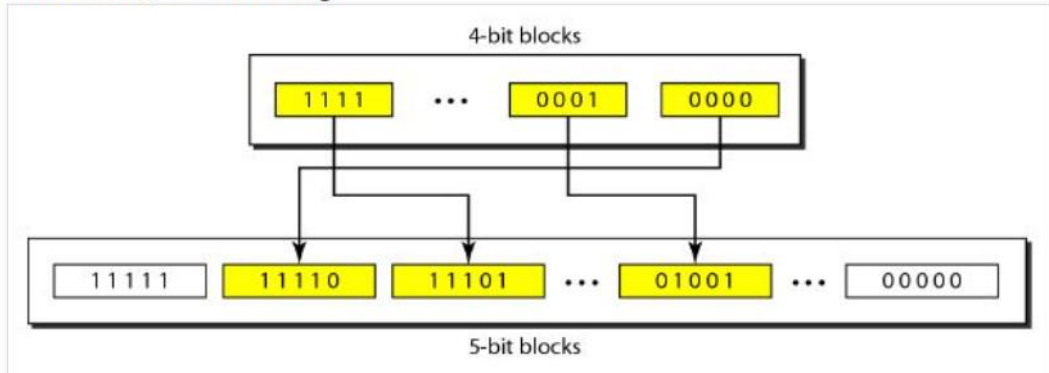
Using block coding 4B/5B with NRZ-I line coding scheme



4B/5B mapping codes

Data Sequence	Encoded Sequence	Control Sequence	Encoded Sequence
0000	11110	Q (Quiet)	00000
0001	01001	I (Idle)	11111
0010	10100	H (Halt)	00100
0011	10101	J (Start delimiter)	11000
0100	01010	K (Start delimiter)	10001
0101	01011	T (End delimiter)	01101
0110	01110	S (Set)	11001
0111	01111	R (Reset)	00111
1000	10010		
1001	10011		
1010	10110		
1011	10111		
1100	11010		
1101	11011		
1110	11100		
1111	11101		

Substitution in 4B/5B block coding



8B/10B block encoding

