NETWORKS AND TELECOMMUNICATIONS SERIES

Digital Communication Techniques

Christian Gontrand

iSTE
WILEY

Digital Communication Techniques

Series Editor
Guy Pujolle

Digital Communication Techniques

Christian Gontrand

ISTE

WILEY

First published 2020 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

Library of Congress Control Number: 2019953804

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN 978-1-78630-540-4

Contents

Acknowledgements

Acknowledgements are owed to the non-exhaustive list below:

Chafia Yahiaoui from the *Ecole Supérieure d'Informatique d'Alger* (Technical University of Algeria), and my telecom colleagues at INSA Lyon: Guillaume Villemaud, Jean-Marie Gorce, Hugues Benoit-Cattin, Attila Baskurt, Stéphane Frenot, Thomas Grenier, Jacques Verdier, Gérard Couturier, Patrice Kadionic, Alexandre Boyer and Carlos Belaustegui Goitia among others, for their detailed observations, as well as their helpful commentaries. Kind acknowledgements also go to Omar Gaouar, my kindly mate at INSA FES, a networker, but also a music buff.

This work is supported by the UpM (*Union pour la Méditerranée* – Mediterranean Union). It has been accomplished at the *Centre d'Intégration en Télécommunication et Intelligence Artificielle* (Center of integration in telecommunications and artificial intelligence), INSA FES, UEMF.

Preface

Impressive developments in Information and Communications Technologies (ICT) have naturally led universities and technical schools to develop the electrical engineering (EI) training they provide. This is particularly true in the wireless communications sector. In fact, communications as part of the transmission of data, whether verbal or in video form, is finding more and ever more varied applications. It is becoming necessary for future graduates to understand and master problems linked to the implementation of radio links, depending on the environment, formatting and source data flow, on the power available to the antenna and on the receiver's selectivity and sensitivity.

This book only requires an introductory level of understanding in mathematics. It does not aim to suffice in and of itself, but rather to convince the reader of the wealth of this domain and its future, to provide good building blocks that will lead to fruition elsewhere. Manufacturers' concise application notes also seem vital for any researcher/engineer.

Technological innovation plays a very important role in the ICT domain. It therefore seems necessary for training courses now to provide well-adapted and innovative content in teaching and associated tools, while still mastering, as well as possible, the fundamental nature of teaching, which is the only guarantee of a solid and lasting education.

This book is aimed at professional diploma students and engineering and masters students. However, it could also perhaps be aimed at researchers in related domains, such as that of hardware, with, for example, phase-locked loops and their central components: voltage-controlled oscillators, and the

famous associated phase noise. Of course, there is an entire domain linked to what is known as firmware, which must be taught, but there are also mathematical tools already in use, for relativity for example, or cryptography, indeed, older forms of coding must be revisited, such as that of Claude Shannon.

Christian GONTRAND
November 2019

Introduction

The word “communication” is now a catch-all in modern society; in its most basic sense, it makes it possible to share information. A department that in any French university or technical school might historically have been labeled as “humanities” (at the end of the 1960s, particularly focused on human resources or sociology); was often later reduced to “communication and humanities”, both terms having become interchangeable in the meantime. Perhaps, now devoid of a clear meaning, nothing will be left apart from the term communication?

This word must not be amalgamated into others: information (transport), (en)coding. Perhaps later semantics are involved in this book, in a strict, technical sense, certainly not in any modernistic sense.

I.1. Why digitize the world?

For broadband communications, transmissions are limited by physical constraints, such as noise or interference, resulting from system imperfections and physical components modifying the transmission of the signal sent. Distortion of the signal over the course of the broadcast is, similarly, a concern. Hence, there is a need for a clear separation of the signals sent, so that, in practice, they remain distinct when they are received.

The transmission of a set of signals undergoes data dispersion over time, leading to intersymbol interference. Signals reflected from buildings, the ground or vehicles cause this dispersion, depending on the length of the paths traveled. The significance of this phenomenon depends on the

frequency (above all high frequency), which can vary stochastically, via, for example, the signal's phases over time (after reflection of obstacles: echoes). They often generate signals, added destructively, or at reception. The resulting signal will therefore be very weak, or sometimes almost non-existent. These signals can also be added constructively; the final signal will therefore be more powerful than one that arrives via a direct path. We note that multiple paths do not present only drawbacks, since they enable communication even when the transmitter and receiver are not in direct contact (for example, Transcontinental Communications).

A signal is often corrupted when it crosses different paths between transmitter and receiver: data bits that reach the receiver are subject to delays. This distorted signal will be interpreted poorly by the receiver.

In broadband communications, signals are limited by constraints: transmission errors are attenuated when the signal is digitized. For example, for the voice, the amplitude of the signal is typically measured 8,000 times per second and its value is coded in an 8-bit sequence (of 0s and 1s) – we refer here to *sampling*. The receiver decodes the sequence of the original signal, thus reconstructing the signal sent. Using only 0s and 1s leads to a low (or indeed non-existent) probability of error. The propagation channel can be modeled via an impulse response (see: linear system, Dirac comb); the signal received r(t) is therefore none other than the filtering of the signal sent x (T) through the propagation channel c (t) and can therefore be written in baseband, via a convolution to which noise is often added (see Langevin term added), modeling the system imperfections. Reference is made to frequency-selective channels when the signal transmitted x (t) occupies a [−W / 2, W / 2] frequency band, which is wider than the propagation channel's coherence bandwidth, c (t), (propagation channel defined as the inverse of the propagation channel's maximum delay spread Tr).

In this case, the frequential components of x(t) separated from the coherence bandwidth undergo different attenuations. In broadband digital systems, symbols are often sent at a regular interval of time T, at a maximum path delay time Tr; the signal received at an instant t can be expressed as a weighted sum (affected by path attenuations) of the signal transmitted simultaneously (the propagation time for the electromagnetic waves is often neglected, as these propagate at the speed of light) and signals sent at previous instants, a multiple of the (sampling) period.

I.2. Temporal representation of a channel

The coefficients of the propagation channel are given by the values taken for various multiple moments of T: [|c(0)|, |c(T)|, |c(2T)|,|c(3T)|, |c(4T)|, |c(5T)|]. If we focus on mobile radio, between buildings, at 5Ghz, T is in the order of 50 ns; Tr equates to 450 ns.

Designers need to reduce interference caused by multiple reflections of the signal and extract the signal. Equalization means balancing the effects of distortions resulting from these multiple paths. To do this, it is necessary to identify the attenuation coefficients that model the effect of the propagation channel c (t).

Current technologies, used in industrial applications, call on training sequences; a "chosen sequence" is sent regularly, known by the sender and the intended recipient. This method makes it possible to know the channels' different phase shifts and delays, and gives good results in practice. On the other hand, if the sampling period is too short in relation to the delay Tr (as is the case with high flow transfers; the number of coefficients c(iT) (typically: $0 \leq i \leq 5$) to be determined can be great, see matrix inversion). Thus, the transmission of high flows when there are several paths present can quickly increase the complexity and therefore the cost of the terminals.

A channel's selective frequency: the signal to be transmitted has frequency components attenuated differently through the propagation channel. This phenomenon is produced when the signal has a broader frequency band than the propagation channel's consistent band. A channel's consistent band is defined as the minimum pass band for which losses from the two channels are independent. This phenomenon is one of the main obstacles to transmission reliability: in fact, it is necessary to estimate the channel (which triggers a loss of flow in moving environments) and also to equalize it (which increases receiver complexity).

Digital equalizer complexity depends on the number of the propagation channel's paths (determined by the relationship between the duration of equalization, Tr, and the sampling period, T), but also the type of constellation transmitted – see Fresnel diagram. The bits are transmitted in the form of symbols rather than as they are. The number of bits contained in each symbol indicates the size of the constellation; the greater this size, the higher the flow. The average size of these constellations generally has a fixed threshold because of the power limits at the terminals.

Why is not it possible to increase the flow indefinitely by increasing constellation size? The transmission rate can be increased by enlarging the constellation. But, if we speak of the rate as the number of bits per second arriving perfectly at the receiver, then this is impossible; the greater the size of the constellation (at a fixed power, which is always normalized for questions of transmission cost), then the closer the values of the symbols transmitted. It is not easy therefore for the receiver to discriminate between two values riddled with errors resulting from noise. We can really increase the flow (i.e. transmission speed) by increasing the constellation. The rate therefore has a threshold called *channel capacity*. The idea of an error-free transmission was scarcely imagined by scientists at the end of the 1950s. At this time, it was natural to reduce the probability of transmission errors by reducing binary flow, thus defining channel capacity. It was only with the work of Claude Shannon at the start of the 1920s that encoding emerged to solve this dilemma.

I.3. The need for coding

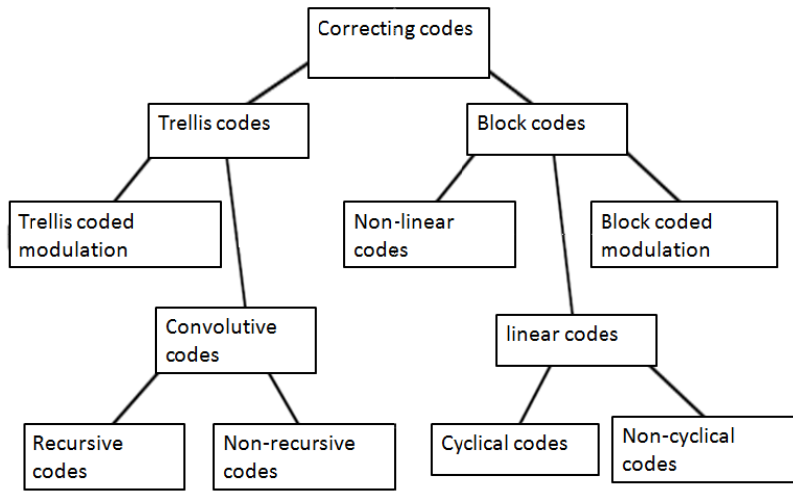

Figure I.1. *Different types of codes*

So that the intended recipient can understand the message broadcast, it must be as close as possible to the initial message. Whatever the principle

behind the broadcast, disruption will be added to the information and will distort it. It is therefore necessary to eliminate this interference; this is the first goal of coding.

I.4. Synoptic bases on information theory

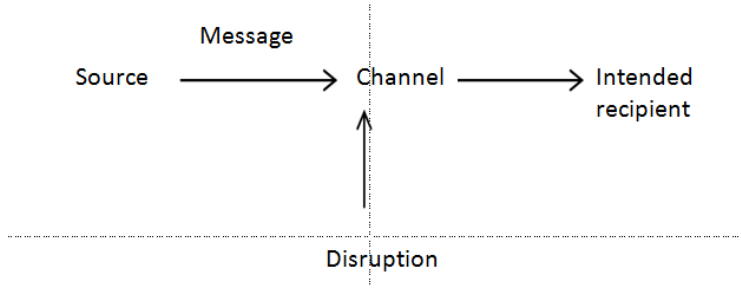

Figure I.2. *Shannon diagram (source: item of interest for the recipient. Channel: origin of the phenomenon of propagation but also of disruption)*

We will consider a discrete channel without memory.

The word "discrete" refers to the fact that the real signal has already been transformed, if it is analog, into a binary digital signal, which is no longer continuous. "Without memory" means that the noise is modeled via a conditional probability of B given that A is independent of time.

From a theoretical perspective, we will approximate this channel using a white Gaussian channel, which means that all the bits have the same broadcast probability, whatever their position.

H entropy: this defines the quantity of information provided by the source; it depends on the 0's and 1's probability of appearance. If a single message is possible, the entropy is null. The entropy makes it possible to measure the quantity of information lost after noisy transmission or encryption.

I.4.1. *Shannon–Hartley theory*

There is a quantity of maximum theoretical information that can be transmitted by the channel. For any channel, there is a coding algorithm such

that the message sent by the source is received with an arbitrarily weak error rate.

I.4.1.1. *A little math*

A message X is a set of basic elements x_i characterized by their probability of occurrence. The quantity of information it transmits is a measure of its unpredictability: the more predictable a message is, the less information it provides.

If x is a basic message and p(x) its *probability of transmission*, then the *quantity of information* h(x) it transmits is defined by $h(x) = -\log(p(x))$. We note that if this message has a probability of 1, the information broadcast, h, is null.

Source entropy: this is defined by the source's average quantity of information, which translates mathematically into the expected value of the intrinsic quantity of information from each basic message. H(X) therefore depends only on the probability of broadcast of 0 or 1.

$$H(X) = E(h(xi)) = -\sum_{i=1}^{k} p(x_i) * \log_2(p(x_i))$$

– Unit: it is *shannon.*

We see that entropy is maximum for a uniform broadcast probability, i.e. for $p(x_i = 0) = p(x_i = 1) = 1/2$, which is the case in a symmetrical binary channel.

H(X/Y) is also defined, called *ambiguity* or *conditional entropy*, which is linked directly to the *probability of error* of the channel's transmission.

$$H(X/Y) = E(h(x_i/y)) = -\sum_{j=1}^{k} p(y_j) * H(X/Y = y_j)$$

with

$$H(X/Y = y_j) = -\sum_{i=1}^{k} p(x_i/y_j) * \log_2 (p(x_i/y_j))$$

In fact, a binary variable X can take only two values: 0 or 1 (Figure I.3).

In the modeling of the binary symmetrical channel, whatever its initial value, there is a probability of error $p = p_e$, so that the bit can be changed into its opposite, and there is a therefore a probability $p = 1 - p_e$ that the bit can be transmitted.

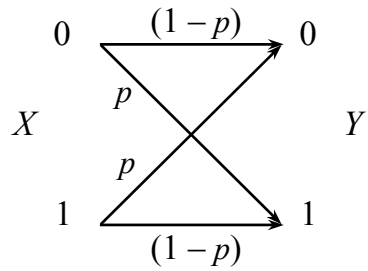

Figure I.3. *Probability of error (symmetrical binary channel)*

For entropy values, we therefore have:

$$\begin{cases} H(X/Y) = -p\log_2(p) - (1-p)\log_2(1-p) \text{ si } 0<p<1 \\ H(X/Y) = 0 \text{ if } p = 0 \text{ or } 1 \end{cases}$$

The data flow Ds corresponds to the product of the entropy by the average number of symbols sent per second.

If each symbol has an average duration of τ_m, then $Ds = H(x) / \tau_m$ in *shannons/s*

The mutual information I(X;Y) measures the quantity of information provided by x, i.e. the correct information transmitted by the channel. We have:

$$I(X;Y) = H(Y)-H(Y/X) = H(X)-H(X/Y)$$

While H(X) represents the initial quantity of information, H(X/Y) is, in a way, the quantity of information lost during the transmission.

If I(X;Y) = 0, H(Y) = H(Y/X), i.e. the probability of reception of y is independent of the transmission probability of the message received; the channel is poor.

However, the channel is considered perfect if I(X;Y) is maximum, i.e. if the negative term –H(X|Y) = 0 is I(X|Y) = H(X)

$$<=> H(X/Y) = 0$$

$$<=> -\log\{p(X/Y)\} = 0$$

$$<=> P(X/Y) = 1$$

It means that if a message Y is received, the message from the origin X is 100% certain.

The capacity of the transmission channel CC is defined by the maximum of mutual information:

$$CC = \max\{I(X;Y)\}$$

The goal of any transmission system is to get closer to this value, given that the unpredictable presence of noise degrades the message sent.

I.4.1.2. *Application to a binary symmetrical channel*

We seek to calculate $Cc = \max\{I(X;Y)\} = \max\{H(Y)-H(Y/X)\}$

– Calculation of H(Y)

We seek to calculate the probability p(y = 0) of obtaining y = 0:

There is a probability p(x = 0) that x = 0 will be sent and a probability $p(y = 0/x = 0) = (1 - p_e)$ that it will be retransmitted correctly. But there is also a probability $p(y = 0/x = 1) = p_e$ that the bit 0 will be received, although it is x = 1 that has been sent, with a probability p(x = 1).

With the result that:

$$p(y = 0) = p(x = 0) * p(y = 0/x = 0) + p(x = 1) * p(y = 0/x = 1)$$
$$= 1/2 * (1 - p_e) + 1/2 * p_e$$

As in the case of a symmetrical binary channel, we know that p(x = 0) = p(x = 1) = 1/2 = 1/2 p(y = 1)

Hence:

$$H(Y) = -1/2*\log_2(1/2) - 1/2*\log_2(1/2) = -2 * 1/2 * \log_2(1/2) = \log_2(2) = 1$$

– Calculation of H(Y/X)

We have H(Y/X) = p(x = 0) * H(Y/x = 0) + p(x = 1)*H(Y/x = 1) = 1/2 * [H(Y/x = 0) + H(Y/x = 1)]

(because p(x = 0) = p(x = 1) = 1/2)

with:

$$H(Y/x = 0) = -p(y = 0/x = 0) * \log_2 (p(y = 0/x = 0)) - p(y = 1/x = 0) * \log_2 (p(y = 1/x = 0))$$
$$= -(1 - p_e) * \log_2 (1 - p_e) - p_e * \log_2 (p_e)$$

and

$$H(Y/x = 1) = -p(y = 0/x = 1) * \log_2 (p(y = 0/x = 1)) - p(y = 1/x = 1) * \log_2 p(y = 1/x = 1))$$
$$= -p_e * \log_2 (p_e) - (1 - p_e) * \log_2 (1 - p_e)$$

We have therefore obtained:

$$H(Y/x = 0) = H(Y/x = 1)$$

Hence:

$$H(Y/X) = 1/2 * [H(Y/x = 0) + H(Y/x = 1)]$$
$$= -p_e * \log_2 (p_e) - (1 - p_e) * \log_2 (1 - p_e)$$

– Calculation of I(X;Y)

By linking the different elements calculated previously:

$$I(X;Y) = H(Y)-H(Y/X)$$
$$= 1 + p_e * \log_2(p_e) + (1 - p_e) * \log_2(1 - p_e)$$

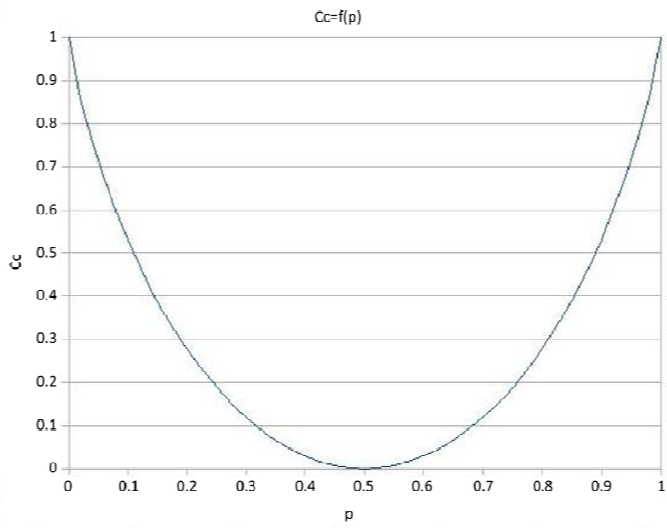

Figure I.4. *Capacity of the transmission channel depending on the probability of error it causes*

We note that the channel's capacity is maximum if the $p_e = 0$ or 1. This result seems logical since the channel is not noisy ($p_e = 0$); we can be confident of having the original message at output, if however we have $p_e = 1$ we know we should take the reverse message to the output message to recover the initial information.

According to *Shannon theory*, for any source of entropy H < C, there is a code of length N so that the probability of error per word is increased by an arbitrarily weak quantity of

$$p_e < 2^{-N * E(Rb)}$$

with $E(R_b)$ Gallagher function or random decoding.

I.5. Codes in linear blocks

To send information, we digitize it, i.e. it is transcribed in the form of a sequence of bits. To avoid the signal deteriorating due to interference present along the journey from the source to the destination, it is necessary to encode it. The most common method consists of introducing a *redundancy* into the input message to be sure of receiving all elements of it on output.

If the channel has high interference, the bits are tripled: if we have *abc* at input, we transmit *aaabbbccc.* The decoder therefore knows how to recognize the bits of the initial signal and errors. For example, if they receive *aiabbbocc,* they choose each time the character most often present in the three consecutive bits, so they read *abc.* With this type of code, we assume initially that the maximum error is one per 3-bit sequence.

If the channel has little interference, there is no need to introduce such a great redundancy, which lengthens the message and hence the transmission time. We therefore choose a *parity bit coding*: the message is broken into k bits to which is added a "parity" bit so that the number of 1s is even. Thus, if the decoder receives an odd number of 1s, they detect an error. As they cannot correct it, they receive an input message, and the cycle recommences until there are no more errors.

I.5.1. *Block codes*

Here too, the message is sequenced into blocks of k bits, treated separately by the coder. We can therefore have 2^k different messages to be sent. We create a code of 2^k collections ordered from n bits ($n > k$) whose elements are called "words": a single word, formed of n bits, is made to correspond to each potential message (see Figure I.3).

For each message received by the coder, if it does not correspond exactly to a word in the code, its *distance** from each of the words is measured to deduce the initial message from it for the smallest distance, less than or equal to the number of errors: $e = E((d - 1)/2)$ (with E as integer part function). In fact, if a message lies at a distance (d/2) from two words, it will not be possible to know what it corresponds to.

A code is then defined by three parameters: [n,k,d] with

– n: word size;

– k: code dimension;

– d: minimum Hamming distance[1] between two words.

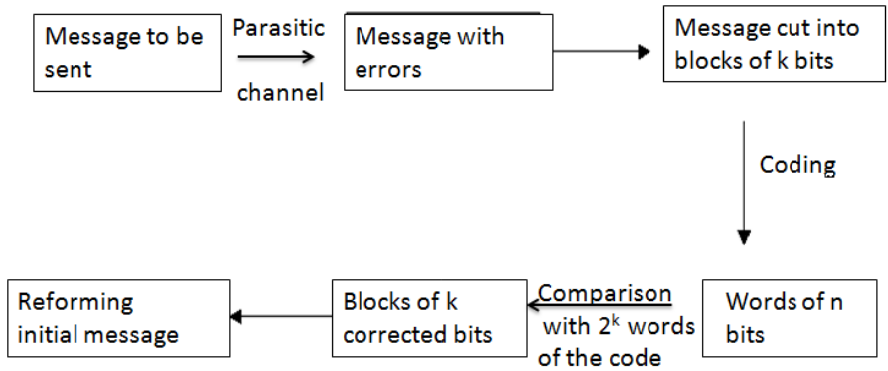

Figure I.5. *Block codes*

The *reliability* of a code is measured using *d/n* and the code rate by $R = k/n$. The smaller R is, the greater the redundancy, and so the longer the operating time. It is therefore necessary to find a good balance between these two values to correct a maximum number of errors in a minimum time.

For example, we wish to send a message over k = 2 bits, it can therefore take $2^2 = 4$ different values. We choose to code the words over 5 bits, and the Hamming distance will therefore be 3. We then have a code [5,2,3].

Message	Corresponding word
(0,0)	(0,0,1,1,0)
(0,1)	(0,1,0,1,1)
(1,0)	(1,0,1,0,1)
(1,1)	(1,1,0,0,0)

If we receive (1,0,1,1,0), we find a minimum d = 1 for the word that corresponds to the message (0,0), so we rectify the error and read (0,0).

1 Hamming distance (d): number of bits by which two words of the code differ.

How does this work in practice? A mathematical tool must be used.

– G is a n*k matrix. Any word c of the code can be obtained from any message x from all the messages using c = Gx, with dim(c) = n and dim(x) = k, $n \geq k$. G is called a coding matrix.

– We take H a (n-k)*n matrix for which Hc = 0, i.e. H(Gx) = GHx = 0, so GH = 0. We can say that H is "orthogonal" to G, their product, scalar, is null. The word c sent therefore belongs to the code if and only if the product Hc is null, otherwise there is an error in the message's reception. H is the code control matrix.

– It is therefore enough to create a matrix G, then H orthogonal to this matrix, to check if the message received is erroneous or not. If it is erroneous, we can then detect and correct this error using a simple matrix calculation.

I.5.2. *Example*

If we take a word coded in blocks of k = 3 bits, there will be $2^3 = 8$ possible different messages (from x_1 to x_8). A word needs to be made to correspond with each of these messages (from c_1 to c_8). To do this, we create a coding matrix G, 6*3 with which each word can be calculated: $c_n = Gx_n$

The messages x_n sent are:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{matrix}$$

$$x_1 = \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \quad x_2 = \begin{matrix} 1 \\ 0 \\ 0 \end{matrix} \quad x_3 = \begin{matrix} 1 \\ 1 \\ 0 \end{matrix} \quad x_4 = \begin{matrix} 1 \\ 1 \\ 1 \end{matrix} \quad x_5 = \begin{matrix} 1 \\ 0 \\ 1 \end{matrix}$$

$$x_6 = \begin{matrix} 0 \\ 0 \\ 1 \end{matrix} \quad x_7 = \begin{matrix} 0 \\ 1 \\ 1 \end{matrix} \quad x_8 = \begin{matrix} 0 \\ 1 \\ 0 \end{matrix}$$

The words of the code are then obtained by $c_n = Gx_n$

$$c_1 = \begin{matrix}0\\0\\0\\0\\0\\0\end{matrix} \quad c_2 = \begin{matrix}1\\0\\0\\1\\0\\1\end{matrix} \quad c_3 = \begin{matrix}1\\1\\0\\0\\1\\1\end{matrix} \quad c_4 = \begin{matrix}1\\1\\1\\0\\0\\0\end{matrix}$$

$$c_5 = \begin{matrix}1\\0\\1\\1\\1\\0\end{matrix} \quad c_6 = \begin{matrix}0\\0\\1\\0\\1\\1\end{matrix} \quad c_7 = \begin{matrix}0\\1\\1\\1\\0\\1\end{matrix} \quad c_8 = \begin{matrix}0\\1\\0\\1\\1\\0\end{matrix}$$

We note that the first 3 bits of each word are identical to the 3 bits of the message sent, then the four following bits serve simply to code the information; these are the parity symbols. This type of code is called "systematic".

We then seek a control matrix H so that HG = 0, and in which all the column vectors are distinct. We find:

$$H = \begin{matrix}0&1&1&1&0&1\\1&1&0&1&0&0\\1&0&1&1&1&0\end{matrix}$$

If we send object 100001, we calculate the syndrome s = Hc = 111 —> this is the fourth column of H; this means that the fourth bit of the word sent is erroneous. As we are dealing with a binary, it is enough to replace the 1 with a 0, and find the corresponding word (here c_2).

NOTE.– There are still undetectable configurations of errors.

I.6. Coding techniques

I.6.1. *Interleaving*

Interleaving is a coding technique that consists of permutating a sequence of bits to distance errors from one another as much as possible. The errors are distributed all along an s sequence; the percentage of errors at each place

is therefore not very high and they can therefore be corrected. The bits will then be put back into order to recover the initial message.

Concretely, interleaving is used, for example, over the CD: if it has a score, the errors are concentrated in the same place; they are distributed along a long sequence so that they can be detected and then corrected. This is Reed–Solomon block coding.

Today, there is no rule for interleaving; different interleavers must be tested to choose the one that gives the best result.

For *turbocodes*, the interleaver is an integral part of the code design (the interleaver is chosen depending on the code).

But this technique poses a problem: an interleaver is often designed for a precise length of code; it will no longer work if the error packet extends over a great length. In turbocodes, *Golden* interleavers are used most as they have good spreading properties.

NOTE.– Interleaving is also used for convolutive codes.

I.6.2. *Convolutive codes*

I.6.2.1. *General remarks*

The principle behind convolutive codes was invented in 1955 by Peter Elias, a professor at MIT. Unlike block codes, which cut the message into finite blocks, we will consider here a semi-infinite sequence of information that passes through several shift registers. The number of these registers is called code memory.

For example, we consider the convolutive code shown in Figure I.6.

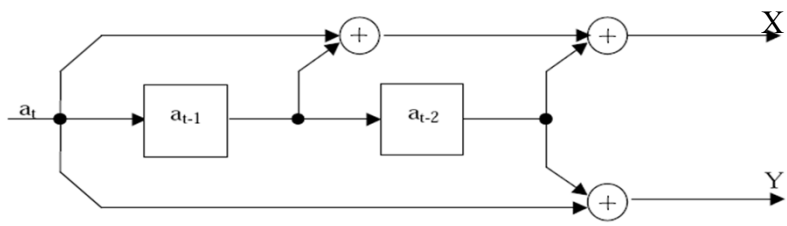

Figure I.6. *Convolutive coder*

It has a memory equal to 2. At the instant t, we therefore consider bits a_t, a_{t-1}, a_{t-2}. At the output, we will have

$$X_1 = a_t \oplus a_{t-1} \oplus a_{t-2} \qquad X_2 = a_t \oplus a_{t-2}$$

(modulo addition 2).

We represent the code using a transition diagram (Figure I.7). This schema describes, for each possible combination of shift registers, the coders' output message, depending on the input bit. Each case in the schema corresponds to a state of the shift registers. The digits beside the arrows indicate the input bits and the coded bit corresponding to the transition. For example, if the coder, initialized at 00, receives the sequence 101, the coded message leaving will be 11 10 00 (Figure I.8).

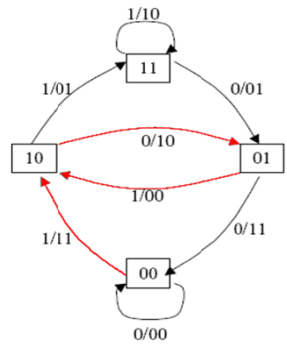

Figure I.7. *Transition diagram. For a color version of this figure, see www.iste.co.uk/gontrand/digital.zip*

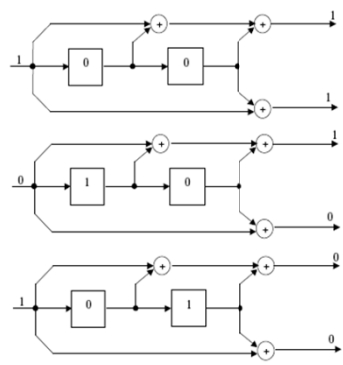

Figure I.8. *Response to message 101*