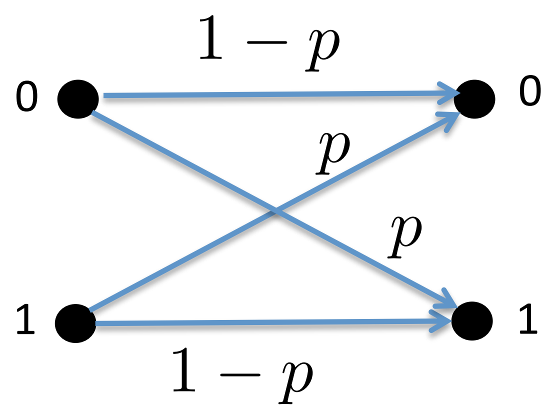
Intro – Channel Codes and Block codes

Binary Symmetric Channel (BSC)

- We will consider



$$C = 1 - H_b(p)$$

Since $0 \leq H_b(p) \leq 1$

Then $0 \leq C \leq 1$

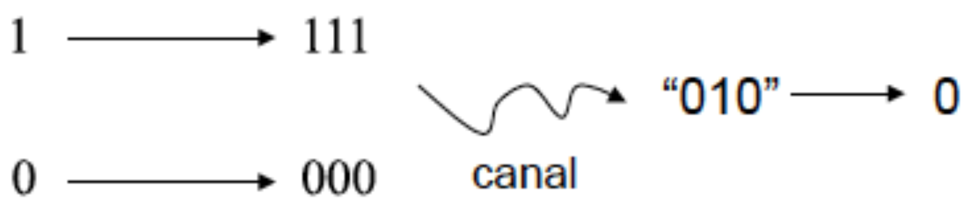
$2^0 = 1 \qquad 2^1 = 2$

$$H_b(p) = -p\log_2(p) - (1-p)\log_2(1-p)$$

- **GOAL: Have a "Pe" (prob. error in detection) smaller than "p" (prob. error of the BSC)**

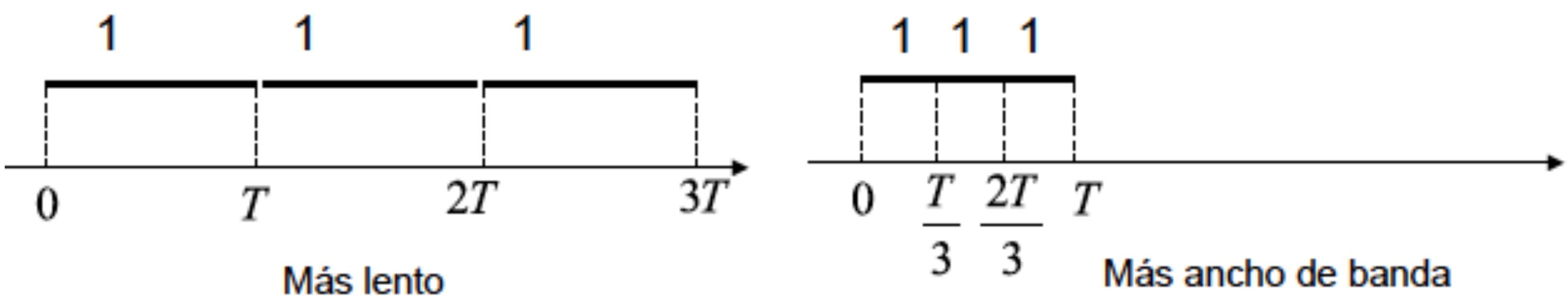
Repetition Code

- Simplest idea: Repeat and “majority vote”



$$R = \frac{k}{k+m} = \frac{1}{1+2} = \frac{1}{3}$$

I use the channel 3 times: slower (less velocity) or I use a “greater bandwidth” (more “frequencies”, more space in the “frequency domain”)



Repetition Code

Sent

$0 \longrightarrow 000$

$1 \longrightarrow 111$

Received		decision
000	**No error**	0 We believe that there is no error. (or 3 errors)
100 010 001	**1 error (we correct 1 error)**	0 **We think there is was at least one error and we correct it**
110 011 101	**2 errors**	1 We believe that there was at least one error. **(And try to correct it)**
111	**3 errors**	1 We believe that there is no error. (or 3 errors)

In 4 cases, we make an error in detection…

Repetition Code

- Assume 1/3 repetition
- What is the probability of

error ?

$$0 \rightarrow 000$$
$$1 \rightarrow 111$$

$$P_e = \underbrace{3p^2(1-p)}_{\text{2 errors}} + \underbrace{p^3}_{\text{3 error}}$$

- If crossover probability of the channel p = 0.01, and we obtain Pe ≈ 0.0003 (if we increase the repetitions we can obtain Pe smaller and smaller)
- Here coding rate R = 1/3. Can we do better? How much better?

Repetition Code

Source	Code
0	000
1	111

Decoder : majority vote.

Example of transmission : b = 0010110.

b	0	0	1	0	1	1	0
c	000	000	111	000	111	111	000
e	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000

Just an example of noise vector

(e noise vector)

Received stream

Decoding : $\hat{b}$ = 0010<u>0</u>10

Decision with one error

P_e (per source bit) : $p^3 + 3p^2(1-p) = 0.028$ and code rate : $R = 1/3$

NB: to reach $P_e \leq 10^{-15}$ we need $R \leq 1/60 \ldots$

Other properties : correction of single errors, detection of double errors.

En general en codificación de canal:

$k =$ longitud palabras de información

$n =$ longitud de las palabras códigos

$$k \leq n$$

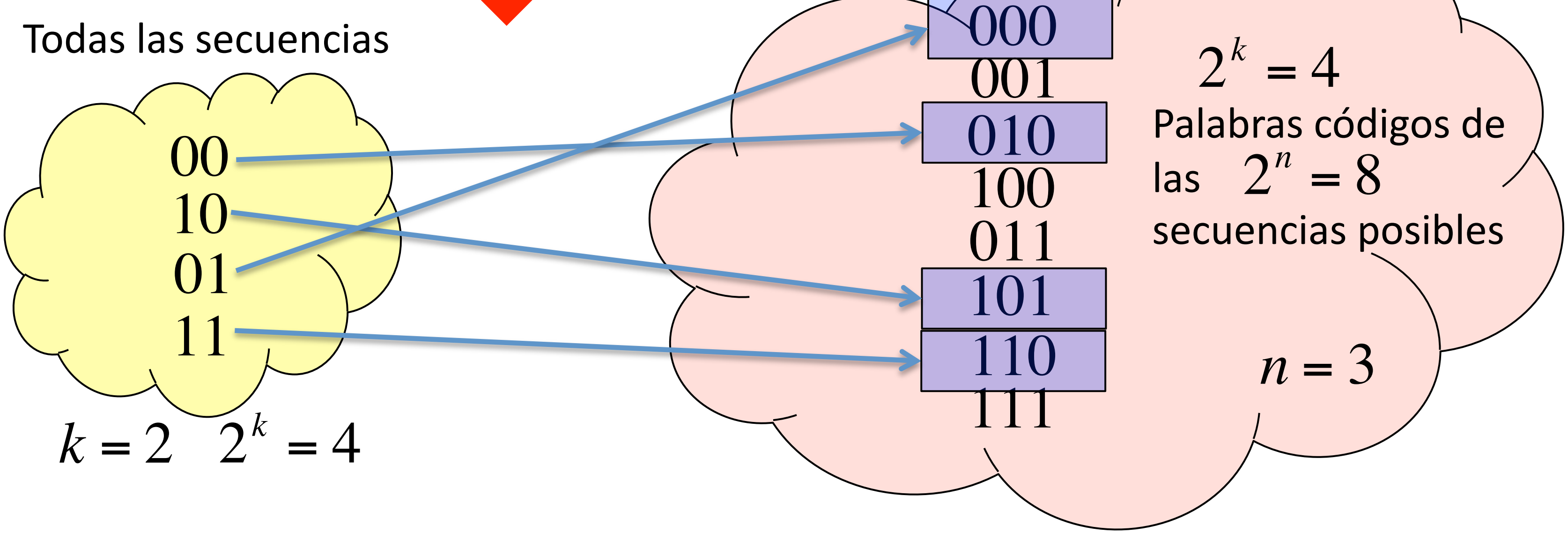
$2^k =$ numero de las posibles secuencias en entrada

$2^k =$ numero palabras código

Tasa del código:

$$R = \frac{k}{n}$$

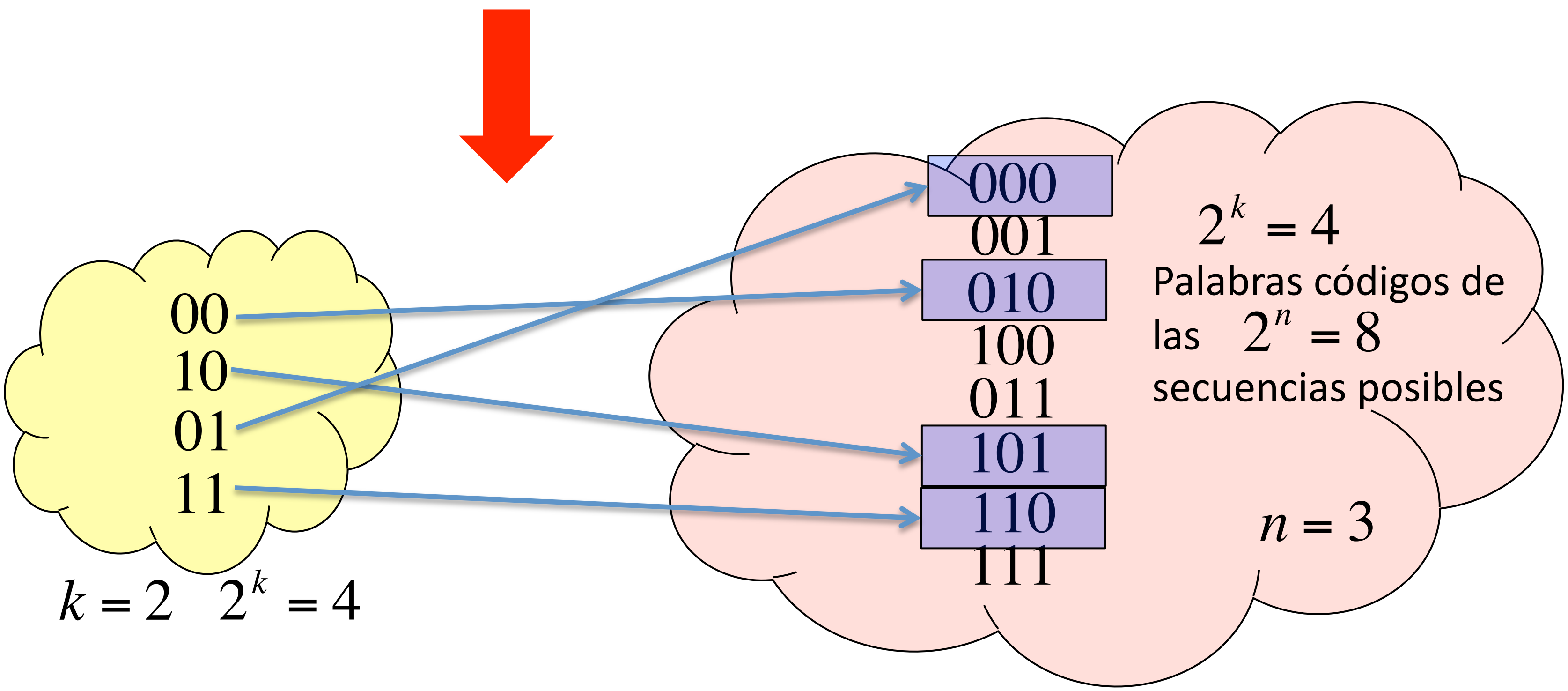
Por ejemplo:



“Channel Coding” means:

➔ Find 2^k sequences of n bits (or better a “mapping”)

En general, la tarea de un codificador consiste en ***elegir 2^k secuencias de n bits***.



Generally, we could construct a table:

Input b	Output c
00	010
10	101
01	000
11	110

Random Codes

- Si la relación biunívoca está elegida aleatoriamente, para decodificar necesitaríamos **la comparación con todas las posibles *k* palabras códigos** (hay que almacenar 2^k palabras código).

- Complejidad creciente en decodifica: no útil.

And we do not take into account the HAMMING DISTANCE (see next slide) among the codewords!

Hamming distance

The **Hamming distance** d between two codewords is the number of positions by which they differ. For example, the codewords 110101 and 111001 have a distance of $d = 2$.

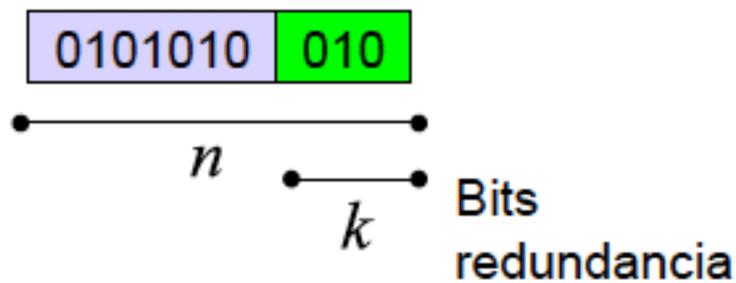
Redundancy: properties

- Para disminuir la complejidad del decodificador, hay que añadir bits (**redundancia**) a los mensajes en modo "inteligente". *Los bits de redundancia* tienen que tener estas propiedades:

1. Ser fácil en generarse (*baja complejidad en codifica*).
2. Maximizar la distancia (diferencia en bits) entre dos palabras códigos. **MAX possible HAMMING DISTANCE**
3. Tener una cierta "estructura" que, a lo mejor, permita individuar donde se han producido los errores.
4. Permitir la decodifica sin comparar con todas las posibles palabras códigos (*baja complejidad en decodifica*).

Ideal channel code

- Tasa de un código: $R = \frac{k}{n}$



1. complejidad lineal en codifica.
2. complejidad lineal en decodifica.
3. probabilidad de error que tiende a cero $P_e \to 0$ por $n \to \infty$. (Possible if R<C)
4. Una tasa *R* más alta posible (más cerca posible del máximo C).

LINEAR CODES

Linear codes

- Dentro de los *lineales*, hay 2 grupos principales que interpretan dos filosofías distintas:

Without memory

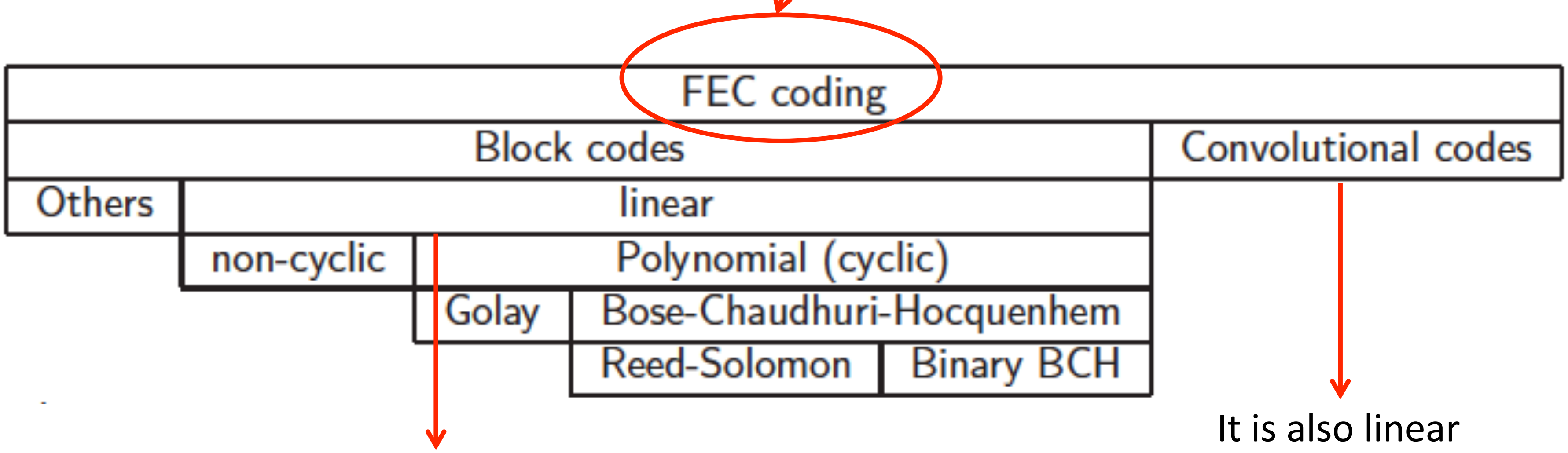
1. Códigos *Bloque* (la decodifica de un bloque de bits se hace de modo independiente de las otras secuencias enviadas).

With memory

2. Códigos *Convolucionales* (sistema con *memoria*).

OVERVIEW: classification

When re-transmission is not an option: forward error correction coding, which introduces extra information (redundancy) into transmitted data for receiver to detect and correct errors



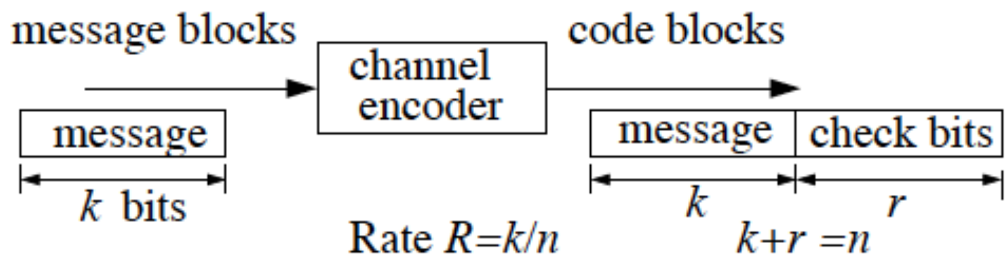
It is also linear

Also the Hamming code is a special case of the linear block codes

LINEAR BLOCK CODES

Systematic vs Non-Systematic

- (n, k) systematic block code



  Systematic: k information bits must be explicitly transmitted (more strict definition also requires they are transmitted together as a block)

- Systematic **linear** block code: first k bits of a codeword are message bits, and last $n - k$ check bits are linear combinations of the k message bits

There are systematic and non-systematic codes. For block codes, systematic ones are more powerful

Linear Block coding

Las palabras códigos en un código bloque lineal se generan utilizando una **matriz generadora G**:

Generating matrix

$$c = bG$$

$1 \times n$ $1 \times k$ $k \times n$

- Cada palabra código se puede expresar como una combinación lineal con coeficientes 0 y 1 de unas palabras de base:

$$\vec{c} = a_1 \cdot \vec{c}_1 + a_2 \cdot \vec{c}_2 + a_3 \cdot \vec{c}_3 + a_3 \cdot \vec{c}_3 + \ldots + a_k \cdot \vec{c}_k$$

$$\vec{c} = \vec{a} \cdot G$$

Recall: we are indirectly building a table

Input b	Output c
00	010
10	101
01	000
11	110

Linear Block coding

- Let $\mathbf{c}$ be n-bit codeword and b be k-bit message, written in row-vector form
- An (n, k) linear block code is defined by its $k \times n$ **generating matrix** G

$$G = [I_k \mid P]$$

  with $k \times (n-k)$ matrix P specifying the given (n, k) linear block code, and I_k being identity matrix of order k

- Encoding process can then be written as

$$c = bG$$

- All elements in P are binary valued, and binary (**modulo-2**) arithmetic operations are carried out

Binary field

Binary field :

- The set {0,1}, under modulo 2 binary addition and multiplication forms a field.

Addition	Multiplication
$0 \oplus 0 = 0$	$0 \cdot 0 = 0$
$0 \oplus 1 = 1$	$0 \cdot 1 = 0$
$1 \oplus 0 = 1$	$1 \cdot 0 = 0$
$1 \oplus 1 = 0$	$1 \cdot 1 = 1$

- Binary field is also called Galois field, GF(2).

Linear Block coding: example

$(6,3)$ linear block code with generating matrix and codebook

$$G = \begin{bmatrix} 1 & 0 & 0 & | & 0 & 1 & 1 \\ 0 & 1 & 0 & | & 1 & 0 & 1 \\ 0 & 0 & 1 & | & 1 & 1 & 0 \end{bmatrix}$$

massages	codewords
000	000 000
001	001 110
010	010 101
011	011 011
100	100 011
101	101 101
110	110 110
111	111 000
b	c

Always...

- For example, for message b=110, parity check bits are

$$c_4 = 1 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 = 0 + 1 + 0 = 1$$
$$c_5 = 1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 = 1 + 0 + 0 = 1$$
$$c_6 = 1 \cdot 1 + 1 \cdot 1 + 0 \cdot 0 = 1 + 1 + 0 = 0$$

Note the binary modulo-2 arithmetic operations involved

- $2^6 = 64$ but only $2^3 = 8$ legal codewords e.g. 111111 is not a legal codeword

- If receiver encounters 111111 it must be due to error, as 111111 will never be sent

Linear Block coding

THE CODEWORD WITH ALL ZEROS IS ALWAYS CONTAINED !!!

CÓDIGO SISTEMATICO:

** los *k* primeros o los *k* últimos bits de la palabra código se corresponden los bits informativos, la palabra de entrada al codificador.

$k \times (n-k) = k \times m$

$k \times n$ $\quad$ $k \times k$

redundancy

$$c = \begin{bmatrix} b & p \end{bmatrix} \qquad G = \begin{bmatrix} I_k & P \end{bmatrix}$$

$$c = \begin{bmatrix} p & b \end{bmatrix} \qquad G = \begin{bmatrix} P & I_k \end{bmatrix}$$

Hamming distance in linear block codes

- **Hamming distance** between two codewords $\mathbf{c}_1$ and $\mathbf{c}_2$ is the number of elements in which they differ

- **Minimum distance** of a codebook, $d_{\min}$, is the smallest Hamming distance between any pair of codewords in the codebook

In the linear block codes:

- **Weight** of a codeword $\mathbf{c}$ is the number of nonzero elements in $\mathbf{c}$

- The minimum distance $d_{\min}$ of a linear block code is equal to the minimum weight of any nonzero codeword in the code

Propiedades de un código bloque **lineal**:

1) Contiene la palabra código con todos ceros

2) Todas combinación lineal de cualquier conjunto de palabras código es a su vez una palabra código.

3) Todas las palabras código poseen al menos otra palabra código a distancia Hamming *dmin*.

4) La *dmin* de un código bloque lineal es igual al menor "peso" (menor número de 1) de una palabra código distinta de la todo ceros.

** Las prestaciones de un código dependen de la distancia minima de Hamming *dmin* entre las palabras código.

$$t \geq \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

Numero de errores corregibles

$$v \geq d_{\min} - 1$$

Numero de errores detectables

- Code with $d_{\min}$ can detect up to $d_{\min} - 1$ errors and correct up to $(d_{\min} - 1)/2$ errors in each codeword

Linear Block codes: DECODER

MATRIZ de CHEQUEO DE PARIDAD (parity check matrix):

$$GH^T = 0 \rightarrow k \times (n-k)$$

G: $k \times n$

H^T: $n \times (n-k)$

$$cH^T = bGH^T = 0$$

$$\boxed{cH^T = 0}$$

Como hallar *H* desde *G*:

$$G \Rightarrow G' \Rightarrow H$$

Sistemática

$$(n-k)\times n$$

$$G' = \begin{bmatrix} I_k & P \end{bmatrix} \qquad H = \begin{bmatrix} P^T & I_{n-k} \end{bmatrix}$$

$$G' = \begin{bmatrix} P & I_k \end{bmatrix} \qquad H = \begin{bmatrix} I_{n-k} & P^T \end{bmatrix}$$

SÍNDROME:

Numero de posibles síndromes

$$2^{n-k}$$

$$r = c + e$$

$$1 \times (n-k) \leftarrow s = rH^T$$

$$s = rH^T = (c+e)H^T = cH^T + eH^T$$

$$s = bGH^T + eH^T = 0 + eH^T$$

$$s = eH^T$$

Linear Block codes: DECODER

- Each $k \times n$ generating matrix $G = [I_k \mid P]$ is associated with a $(n-k) \times n$ **parity check matrix**
$$H = [P^T \mid I_{n-k}]$$
Basic **property of codeword**: $\mathbf{c}$ is a codeword in the (n, k) block code generated by G, if and only if $\mathbf{c}H^T = 0$

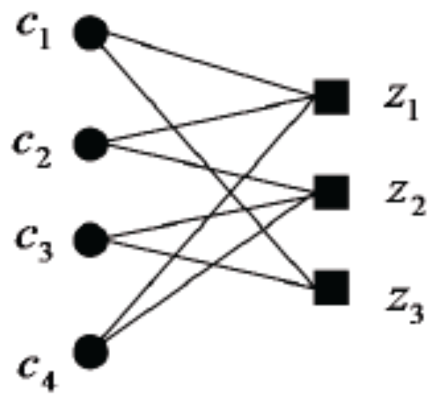
- Received row vector $\mathbf{r}$ can be written as
$$\mathbf{r} = \mathbf{c} + \mathbf{e}$$
All the elements are binary valued, e.g. if the transmitted $c_i = 1$ and is received in error: $r_i = 0$, then $e_i = 1$

- $(n-k)$ (row vector) **error syndrome**
$$\mathbf{s} = \mathbf{r}H^T = (\mathbf{c} + \mathbf{e})H^T = \mathbf{c}H^T + \mathbf{e}H^T = \mathbf{e}H^T$$
$\mathbf{s}$ is related to the error vector $\mathbf{e}$, and can be used to detect and correct errors

Tanner graph and check system

- A cada matriz de paridad *H* está asociado un gráfico compuesto por 2 conjuntos de nodos:

$$H_{3\times4} = \begin{bmatrix} 1\,1\,0\,1 \\ 0\,1\,1\,1 \\ 1\,0\,1\,0 \end{bmatrix}$$



$$\begin{cases} c_1 + c_2 + c_4 = 0 \\ c_2 + c_3 + c_4 = 0 \\ c_1 + c_3 = 0 \end{cases} \Longleftrightarrow cH^\top = 0$$

- se ve que c_1 interviene en el nodo z_1, z_3.

Procedimiento general de decodificación:

En general tenemos que hallar la $\hat{c}$
más cercana a $r = c + e$

En termino de distancia de Hamming.

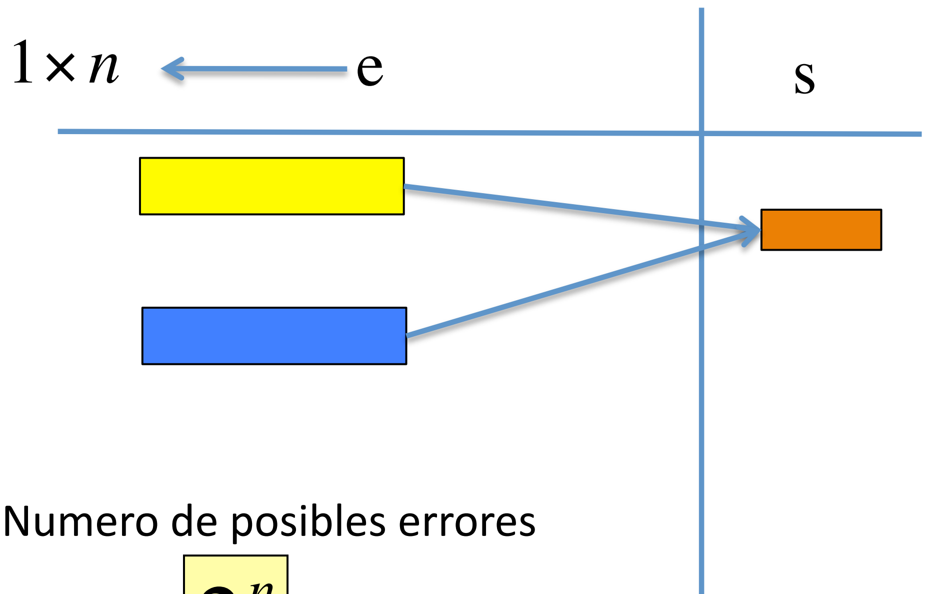
$$r \Rightarrow \hat{c} \Rightarrow \hat{b}$$

Procedimiento eficiente de decodificación para códigos bloque lineales:

Construir la tabla de síndromes utilizando la formula:

$$s = eH^T$$

$1 \times (n-k)$



Numero de posibles síndromes

$$2^m = 2^{n-k}$$

Numero de posibles errores

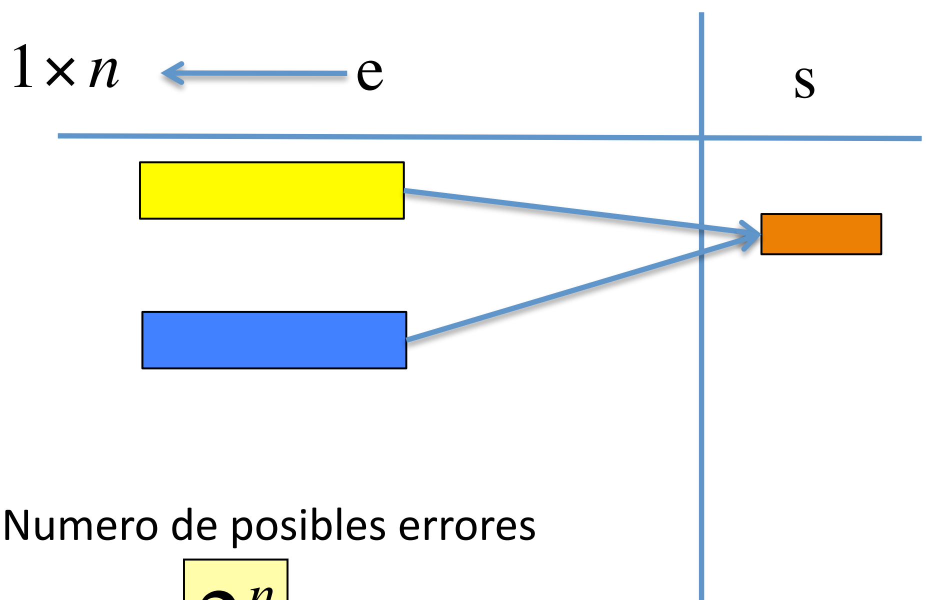
$$2^n$$

Procedimiento decodificación para códigos bloque lineales:

1) Construir la tabla de síndromes utilizando la formula:

$$s = eH^T$$

$1 \times (n - k)$



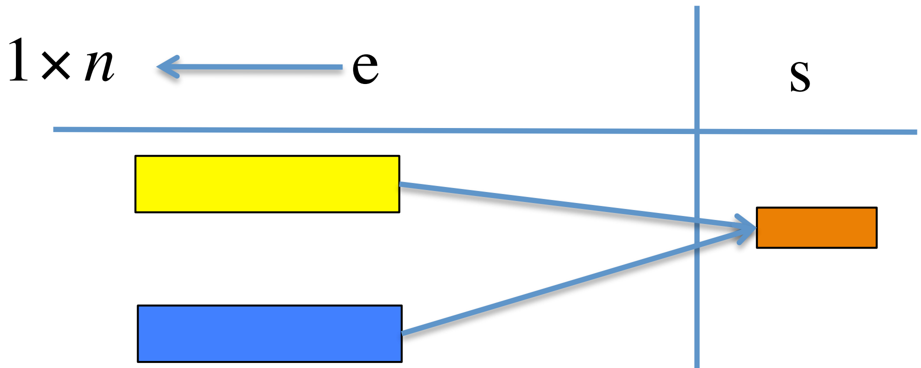
Numero de posibles errores

$$2^n$$

Numero de posibles síndromes

$$2^m = 2^{n-k}$$

Procedimiento decodificación para códigos bloque lineales:



Numero de posibles errores

$$2^n$$

Numero de posibles síndromes

$$2^m = 2^{n-k}$$

To each sequence of syndrome we have $2^{n-m} = 2^k$ patterns or error associated.

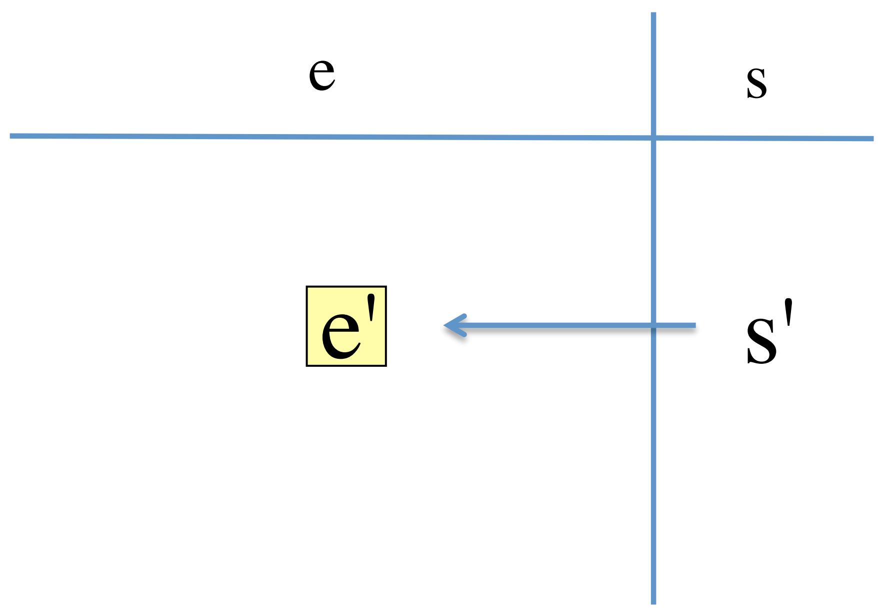
Received stream of bits

$$r' = c + e$$

Obtain the syndrome of r' with the formula:

$$s' = r' H^T$$

Find the corresponding pattern of error
(the MOST LIKELY ONE in term of probability)



Hallar la palabra código estimada (la más cercana a r’), corrigiendo r’ utilizando e’, es decir:

$$\hat{c} = r' - e' = r' + e'$$

Operaciones en binario, restar=sumar.

Obtain the information bits using the table:

b	c
$\hat{b}$ ←	$\hat{c}$

- Example: Block code (6,3)

$$\mathbf{G} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{G} = [\mathbf{P} \mid \mathbf{I}_k]$$

$$\mathbf{H} = [\mathbf{I}_{n-k} \mid \mathbf{P}^T]$$

Message vector	Codeword
000	000000
100	110100
010	011010
110	101110
001	101001
101	011101
011	110011
111	000111
b	c

$$\mathbf{c} = \mathbf{b}\mathbf{G}$$

THIS TABLE IS NOT COMPLETE!!
IT IS PARTIAL

Columns of H

Error pattern	Syndrome
000000	000
000001	101
000010	011
000100	110
001000	001
010000	010
100000	100
010001	111
e	s

$\mathbf{U} = (101110)$ transmitted.

$\mathbf{r} = (001110)$ is received.

- The syndrome of $\mathbf{r}$ is computed:

$$\mathbf{S} = \mathbf{r}\mathbf{H}^T = (001110)\mathbf{H}^T = (100)$$

- Error pattern corresponding to this syndrome is

$$\hat{\mathbf{e}} = (100000)$$

- The corrected vector is estimated

$$\hat{\mathbf{U}} = \mathbf{r} + \hat{\mathbf{e}} = (001110) + (100000) = (101110)$$

....JUST ZERO ERRORS, ONE ERROR, AND AN EXAMPLE OF TWO ERRORS.... But there are not all the possible ERROR PATTERNS (error vectors)

Hamming codes: special case

Hamming codes

- Hamming codes are a subclass of linear block codes and belong to the category of *perfect codes*.
- Hamming codes are expressed as a function of a single integer .

$$m \geq 2$$

Code length :	$n = 2^m - 1$
Number of information bits :	$k = 2^m - m - 1$
Number of parity bits :	$n\text{-}k = m$
Error correction capability :	$t = 1$

- The columns of the parity-check matrix, **H**, consist of all non-zero binary m-tuples.

Hamming codes: special case

- Example: Systematic Hamming code (7,4)

$$\mathbf{H} = \left[\begin{array}{ccc:cccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right] = [\mathbf{I}_{3\times3} \;\vdots\; \mathbf{P}^T]$$

$$\mathbf{G} = \left[\begin{array}{ccc:cccc} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] = [\mathbf{P} \;\vdots\; \mathbf{I}_{4\times4}]$$