Tema 1. Entropía e información

1.1. Introducción

- Supongamos que queremos tener información del tiempo que hace en Málaga, para lo cual construimos un sistema de telecomunicaciones digital. Para nuestro propósito, nos basta con saber que en Málaga hace sol, está nublado, llueve u otra situación no descrita. Para ello codificamos de la siguiente manera:

Tiempo en Málaga	Codificación
Sol	00
Nublado	01
Lluvia	10
Otros	11

Pero como todos sabemos, en Málaga es muy probable que haga sol, mientras que una nevada en Málaga es bastante rara o improbable. Por tanto podríamos hacer una codificación en función de la probabilidad de los sucesos

Tiempo en Málaga	Probabilidad	Codificación
Sol	1/2	0
Nublado	1/4	10
Lluvia	1/5	11
Otros	1/20	111

Con esta codificación obtenemos una menor longitud media de código, reduciendo la cantidad de bits enviada.

- Como hemos visto en el ejemplo anterior, la cantidad de información es proporcional al inverso de la probabilidad de que ese suceso ocurra.

- Además debemos tener en cuenta dos casos particulares:

  * Si un suceso es conocido, entonces no hay información (todo el mundo lo conoce, es un hecho).

  * Si un suceso es imposible, entonces no hay información tampoco porque, a priori, sabemos que no ocurrirá.

2. Entropía

- La entropía es la función que nos da la información que tiene una fuente, y se define:

  * Sea una variable aleatoria discreta X que tiene una determinada distribución de probabilidades, $P(x)$.
  $H(X)$, entropía de X es:

$$H(X) = -\sum_{x \in X} P(x) \cdot \log_b P(x) = \sum_{x \in X} P(x) \log_b \cdot 1/P(x) = E\left[\log_b \frac{1}{P(x)}\right]$$

- Nota:

  - Unidades de la entropía $H(X)$:

BASE DEL LOGARITMO	UNIDAD
Decimal (10)	Dits
→ Binario (2)	Bits
Neperiano (e)	Nats

- Propiedades de la entropía

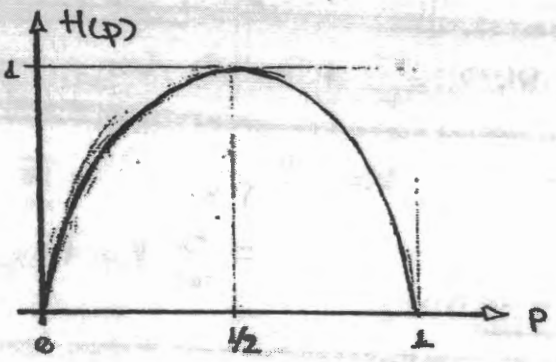
- $H(X) \geq 0$
- $H_b(X) = \sum_{x \in \mathcal{X}} p(x) \log_b(1/p(x)) = \log_b a \cdot H_a(X)$

* Caso particular:

Sea X v.a. discreta con $x \in \{0,1\}$ / $p(0) = p$

$\Rightarrow H(X) = -p \log_2 p - (1-p) \log_2 (1-p) \equiv H(p)$

que gráficamente:



1.3. Entropía de dos variables

- Entropía conjunta

- Sean X, Y v.a. discretas, se define $H(X,Y)$ entropía conjunta:

$$H(X,Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \cdot \log_2 \frac{1}{p(x,y)} = E_{xy}\left(\log_2 \frac{1}{p(x,y)}\right)$$

o

ISAL:

$$F_{XY}(x,y) = P(X \leq x, Y \leq y)$$

- ¿Cómo calcular las entropías "marginales" dada la conjunta?

- Sean X, Y v.a. discretas, con $H(X,Y) = \underset{xy}{E}\left(\log_2\left(\frac{1}{p(x,y)}\right)\right)$

¿$H(X)$? $\Rightarrow$ $p(x) = \sum_y p(x,y) \Rightarrow H(X) = \underset{x}{E}\left(\log_2\left(\frac{1}{p(x)}\right)\right)$

¿$H(Y)$? $\Rightarrow$ $p(y) = \sum_x p(x,y) \Rightarrow H(Y) = \underset{y}{E}\left(\log_2\left(\frac{1}{p(y)}\right)\right)$

- **Entropía condicionada**

- Sean X, Y v.a. discretas. Se define la entropía condicionada, $H(X/Y)$, como:

$$H(X/Y) = \sum_y p(y)\left(\sum_x p(x/y)\cdot\log_2\left(\frac{1}{p(x/y)}\right)\right) =$$

(esta es la buena; la que se usa)

$$= \sum_y \sum_x p(x,y)\cdot\log_2\left(\frac{1}{p(x/y)}\right) = \underset{xy}{E}\left(\log_2\left(\frac{1}{p(x/y)}\right)\right)$$

$$= \underset{\forall y}{E}\, p(y)\, H(X/Y=y)$$

Analogamente $H(Y/X)$:

$$H(Y/X) = \sum_x p(x)\left(\sum_y p(y/x)\cdot\log_2\left(\frac{1}{p(y/x)}\right)\right) = \ldots$$

$$\ldots = \underset{xy}{E}\left(\log_2\left(\frac{1}{p(y/x)}\right)\right)$$

Además se cumple:

$p(y) = \frac{p(x,y)}{p(x)}$ $\quad p(x,y) = p(y)\cdot p(x/y)$, tomando logaritmos y luego entropías tenemos:

$$H(X,Y) = H(X/Y) + H(Y)$$

Análogamente:

$$p(x,y) = p(x)\cdot p(y/x) \Rightarrow H(X,Y) = H(Y/X) + H(X)$$

** Ejemplo. Sea un descifrador de texto que tiene la siguiente cadena:

ENE

X Y

Si X, Y son variables aleatorias / $x \in \{a,b,c,d\}$
$y \in \{a,b,c,d\}$

y la distribución de probabilidades conjunta es:

Y \ X	a	b	c	d	
a	1/8	1/16	1/32	1/32	→ p(y=a)
b	1/16	1/8	1/32	1/32	→ p(y=b)
c	1/16	1/16	1/16	1/16	→ p(y=c)
d	1/4	0	0	0	→ p(y=d)
	↓ p(x=a)	↓ p(x=b)	↓ p(x=c)	↓ p(x=d)	

1º ¿ $H(X)$? ¿ $H(Y)$?

$$p(x=a) = \sum_{y \in Y} p(x=a, y) = 1/8 + 1/16 + 1/16 + 1/4 = 1/2$$

$$p(x=b) = \sum_{y \in Y} p(x=b, y) = 1/16 + 1/8 + 1/16 = 1/4$$

$$p(x=c) = \sum_{y \in Y} p(x=c, y) = 1/32 + 1/32 + 1/16 = 1/8$$

$$p(x=d) = \sum_{y \in Y} p(x=d, y) = 1/32 + 1/32 + 1/16 = 1/8$$

$$\Rightarrow p(x) = \{1/2, 1/4, 1/8, 1/8\} \Rightarrow H(X) = -\sum_i p_i \cdot \log_2 p_i = \frac{7}{4} \text{ bits}$$

Análogamente : $p(y) = \{1/4, 1/4, 1/4, 1/4\} \Rightarrow H(Y) = 2$ bits

2º ¿ $H(X/Y)$?

$$H(X/Y) = \underset{xy}{E}\left(-\log_2\left(p(x/y)\right)\right) \Rightarrow ¿p(x/y)?$$

$p(x/y=a) = p(x,a)/p(y=a) = 4\cdot p(x,a) = \{1/2, 1/4, 1/8, 1/8\}$

$p(x/y=b) = p(x,b)/p(y=b) = 4\cdot p(x,b) = \{1/4, 1/2, 1/8, 1/8\}$

$p(x/y=c) = p(x,c)/p(y=c) = 4\cdot p(x,c) = \{1/4, 1/4, 1/4, 1/4\}$

$p(x/y=d) = p(x,d)/p(y=d) = 4\cdot p(x,d) = \{1, 0, 0, 0\}$

$$\Rightarrow H(X/Y) = 1/8\cdot 1 + 1/16\cdot 2 + 1/16\cdot 2 + \cancel{1/4\cdot 0} + 1/16\cdot 2 + 1/8\cdot 1 +$$
$$+ 1/16\cdot 2 + 1/32\cdot 3 + 1/32\cdot 3 + 1/16\cdot 2 + 1/32\cdot 3 + 1/32\cdot 3$$
$$+ 1/16\cdot 2 = 1/4 + 3/4 + 3/8 = \frac{11}{8}$$

Análogamente $H(Y/X) = \ldots \frac{13}{8}$

Cumpliendose :

$$H(X,Y) = H(Y/X) + H(X) = 27/8$$
$$H(X,Y) = H(X/Y) + H(Y) = 27/8$$

Información mutua.

- Entropía relativa

- Sea X v.a. discreta con $x \in \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$

⇒ Podemos realizar una distribución de probabilidades $p(x)$, pero tambien podemos realizar dtn sobre esta misma v.a. de tal forma:

$$\{x_1, x_2, \ldots, x_i, \ldots, x_n\}$$

$$q(x) : q(x_1), q(x_2), \ldots, q(x_i), \ldots, q(x_n)$$

$$p(x) : p(x_1), p(x_2), \ldots, p(x_i), \ldots, p(x_n)$$

- Con todo esto definimos la entropía relativa, o distancia de Kullback-Leibler como:

$$D(p\|q) = \sum_{x\in\mathcal{X}} p(x)\cdot \log_2 \frac{p(x)}{q(x)} = E_{p(x)}\left(\log_2 \frac{p(x)}{q(x)}\right)$$

Nos da una medida de la distancia entre dos distribuciones de probabilidades de una misma v.a.

- Propiedades:
  - $D(p\|q) \geq 0$
  - $D(p\|q) = 0 \Leftrightarrow p(x) = q(x)$
  - $D(p\|q) \neq D(q\|p)$

- Información mutua entre $\mathcal{X}$ e $\mathcal{Y}$.

- Sean $\mathcal{X}, \mathcal{Y}$ v.a. discretas; definimos información mutua entre $\mathcal{X}$ e $\mathcal{Y}$, $I(\mathcal{X};\mathcal{Y})$, como:

$$I(\mathcal{X};\mathcal{Y}) = D(p(x,y)\|p(x)p(y)) =$$

$$= \sum_x \sum_y p(x,y)\cdot \log_2\left(\frac{p(x,y)}{p(x)p(y)}\right) = E_{x,y}\left(\log_2\left(\frac{p(x,y)}{p(x)p(y)}\right)\right)$$

* Además se cumple: $I(\mathcal{X};\mathcal{Y}) = I(\mathcal{Y};\mathcal{X})$

$I(\mathcal{X};\mathcal{Y})$ nos da una medida de la cantidad de información que una variable aleatoria contiene sobre la otra

* Analizando la expresión:

$$\frac{p(x,y)}{p(x)p(y)} = \frac{p(x/y)}{p(x)} = \frac{p(y/x)}{p(y)}$$

(1) (2)

(1): $\frac{p(x,y)}{p(x)p(y)} = \frac{p(x/y)}{p(x)}$ ⇒ Tomando logaritmos y luego medias,

llegamos a: $I(\mathcal{X};\mathcal{Y}) = H(\mathcal{X}) - H(\mathcal{X}/\mathcal{Y})$

De maneira análoga:

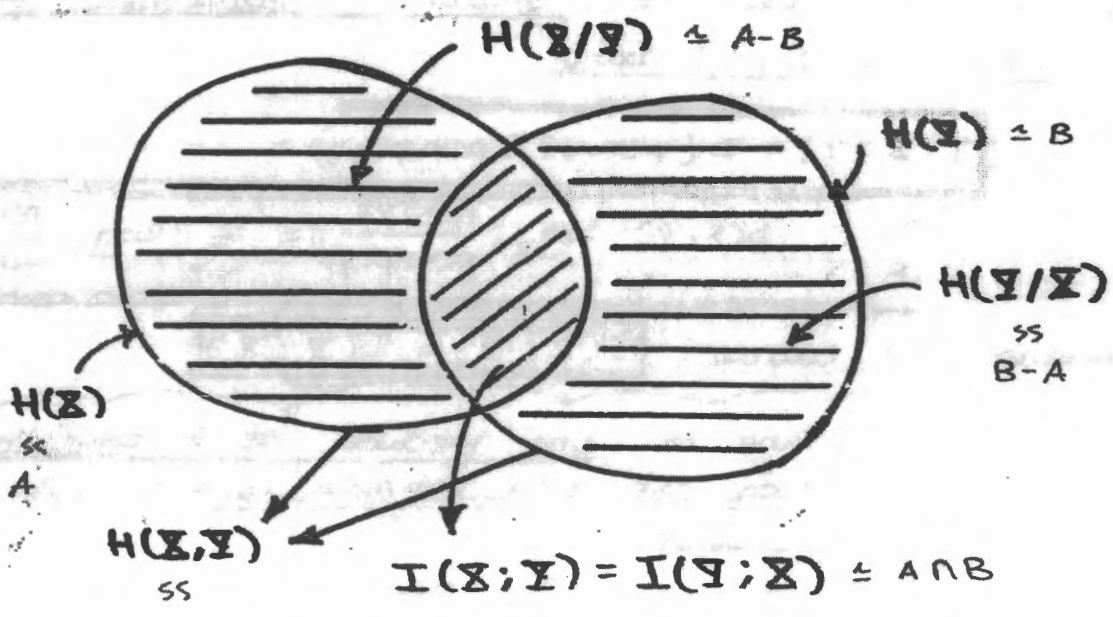
$$(2): \frac{p(x,y)}{p(x)p(y)} = \frac{p(y/x)}{p(y)} \Rightarrow \boxed{I(X;Y) = H(Y) - H(Y/X)}$$

$$\Rightarrow \boxed{I(X;Y) = I(Y;X) = H(X) - H(X/Y) = H(Y) - H(Y/X)}$$

* Ademais: $H(X,Y) = H(X) + H(Y/X) = H(Y) + H(X/Y)$

$$\Rightarrow \boxed{I(X;Y) = H(X) + H(Y) - H(X,Y)}$$

- Representando tudo em um diagrama de Venn:



$A \cup B = A + B - (A \cap B)$

.5. Entropía e información mutua multidimensional

- **Entropía conjunta**

- Sean $X_1, X_2, \dots, X_n$ v.a. discretas. Definimos entropía conjunta, $H(X_1, X_2, \dots, X_n)$, como:

$$H(X_1, X_2, \dots, X_n) = \underset{x_1, x_2 \dots x_n}{E}\left(\log_2 \frac{1}{p(x_1, x_2, \dots, x_n)}\right)$$

* esta es una definición rigurosa pero poco practica. Por ello utilizamos el teorema de la multiplicación:

$$p(x_1, x_2, \dots x_n) = p(x_1) \cdot p(x_2/x_1) \cdot p(x_3/x_1, x_2) \cdot \dots$$

* Tomando logaritmos y esperanzas:

$$H(X_1, X_2, \dots, X_n) = H(X_1) + H(X_2/X_1) + \dots + H(X_n/X_1, \dots, X_{n-1})$$

$$= \sum_{i=1}^{n} H(X_i/X_{i-1}, \dots, X_1)$$

- **Entropía relativa condicional**

- Sean X, Y, variables aleatorias discretas, con dos distribuciones p y q. Se define la entropía relativa condicional como:

$$D(p(y/x) \| q(y/x)) = \sum_x p(x) \sum_y p(y/x) \cdot \log_2 \frac{p(y/x)}{q(y/x)} =$$

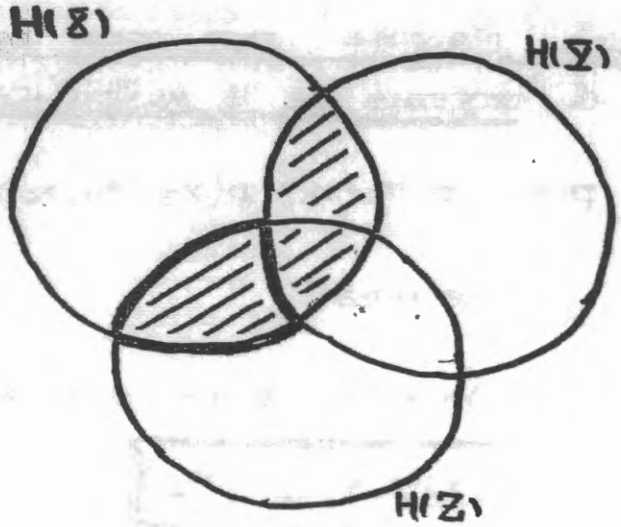
$$= \sum_x \sum_y p(x,y) \log_2 \frac{p(y/x)}{q(y/x)} = \underset{x,y\ p(x,y)}{E}\left(\log_2 \frac{p(y/x)}{q(y/x)}\right)$$

* Además se demuestra que:

$$D(p(x,y) \| q(x,y)) = D(p(x) \| q(x)) + D(p(y/x) \| q(y/x))$$

- Información mutua

- Consideramos el diagrama de Venn pero ahora vamos a considerar tres variables: X, Y, Z:



- Definimos el area rayada como $I(X; Y, Z)$ y vemos gráficamente que:

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

* Además se ve que:

$$I(X; Y, Z) \neq I(X; Y) + I(X; Z)$$

- Si consideramos el área sobresaltada, podemos definir $I(X;Z/Y)$, que por definición:

$$I(X;Z/Y) \overset{def}{=} D\left(p(x,z/y) \,\|\, p(x/y)\cdot p(z/y)\right) =$$

$$= \sum_x \sum_y \sum_z p(x,y,z) \log_2 \left(\underbrace{\frac{p(x,z/y)}{p(x/y)\,p(z/y)}}_{①} \right)$$

* Por otra parte:

$$I(X;Y,Z) \overset{def}{=} D\left(p(x,y,z) \,\|\, p(x)\,p(y,z)\right) =$$

$$= \sum_x \sum_y \sum_z p(x,y,z) \log_2 \left(\underbrace{\frac{p(x,y,z)}{p(x)\,p(y,z)}}_{②} \right)$$

* De ②: $\dfrac{p(x,y,z)}{p(x)p(y,z)} = \dfrac{p(x,z/y)\cdot p(y)}{p(x)\cdot p(y)\cdot p(z/y)} \cdot \dfrac{p(x/y)}{p(x/y)} =$

$$= \underbrace{\frac{p(x,z/y)}{p(x/y)\cdot p(z/y)}}_{①} \cdot \frac{p(x/y)\cdot p(y)}{p(x)\cdot p(y)} = \frac{p(x,z/y)}{p(x/y)\,p(z/y)} \cdot \frac{p(x,y)}{p(x)\,p(y)}$$

⇒ Tomando logaritmos y esperanzas:

$$\boxed{I(X;Z,Y) = I(X;Z/Y) + I(X;Y)}$$

* Además se ve gráficamente que:

$$\boxed{I(X;Y/Z) = H(X/Z) - H(X/Y,Z)}$$

- Si tenemos $X_1, X_2, \dots X_n$ v.a discretas, la información mutua $I(X_1, X_2, \dots, X_n ; Y)$ se define como: (Y: v.a. discreta.)

$$I(X_1, X_2, \dots, X_n ; Y) = H(X_1, X_2, \dots, X_n) - H(X_1, \dots, X_n / Y) =$$

$$= \sum_{i=1}^{n} H(X_i / X_{i-1}, \dots, X_1) - \sum_{i=1}^{n} H(X_i / X_{i-1}, \dots X_1, Y) =$$

$$= \sum_{i=1}^{n} I(X_i ; Y / X_{i-1}, \dots, X_1) = I(X_1 ; Y) + I(X_2 ; Y / X_1) + I(X_3 ; Y / X_1, X_2) + \dots + I(X_n ; Y / X_1, \dots, X_{n-1})$$

1.6. Desigualdad de Jensen y propiedades

- Función cóncava / convexa.

- Una función se dice que es cóncava en el intervalo (a,b) si se cumple que $\forall\, x_1, x_2 \in (a,b)$ y $\forall\, k \in [0,1]$:

$$f(kx_1 + (1-k)x_2) \leq k \cdot f(x_1) + (1-k) f(x_2)$$

- Una función se dice que es convexa en el intervalo (a,b) si se cumple que $\forall\, x_1, x_2 \in (a,b)$ y $\forall\, k \in [0,1]$:

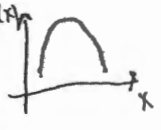
$$f(kx_1 + (1-k)x_2) \geq k \cdot f(x_1) + (1-k) f(x_2)$$

- Desigualdad de Jensen

- Sea f una función convexa y X una v.a. discreta con distribución de probabilidades $p_1, p_2, \dots, p_n$ (con $\sum_i p_i = 1$); entonces:

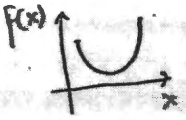
$$\sum_{i=1}^{n} p_i f(x_i) \leq f\left(\sum_{i=1}^{n} p_i \cdot x_i\right)$$

por lo que: $$E[f(X)] \leq f[E(X)]$$



* Si la función fuera cóncava:

$$E[f(X)] \geq f[E(X)]$$



- **Propiedades**

$D(p\|q) \geq 0$

$D(p\|q) = 0 \Leftrightarrow p(x) = q(x) \quad \forall x$

Demo:

$$-D(p\|q) = -\sum_x p(x) \cdot \log_2 \frac{p(x)}{q(x)} \leq \log_2 \sum_x p(x) \frac{q(x)}{p(x)} =$$

($\log_2 \equiv$ f. convexa.)

$$= \log_2 \sum_x q(x) = \log_2 1 = 0 \Rightarrow D(p\|q) \geq 0$$

* La única forma de que $D(p\|q) = 0$ es que:

$$\log_2 \frac{p(x)}{q(x)} = 0 \Leftrightarrow p(x) = q(x)$$

$I(X;Y) \geq 0$

$I(X;Y) = 0 \Leftrightarrow X, Y$ independientes

Demo: $I(X;Y) = D(p(x,y) \| p(x)p(y)) = \dots$ (igual que antes)

Sea X v.a. discreta con distribuciones $p(x)$ y $q(x)$, siendo $q(x)$ uniforme $\Rightarrow q_i(x) = 1/N$ siendo N el número de sucesos.

$$\Rightarrow D(p(x)\|q(x)) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)} = \sum_x p(x) \left[\log_2 N + \log_2 p(x)\right]$$

$$= \log_2 N - H(x) \geq 0$$

$$\Rightarrow \boxed{0 \leq H(x) \leq \log_2 N}$$

$H(X) \geq H(X/Y)$

$H(X) = H(X/Y) \iff X$ e Y son independientes

Demo:

$$H(X) - H(X/Y) = I(X;Y) \geq 0 \quad (\dots)$$

** Ejemplo. Tenemos un descifrador de texto y tenemos dos letras indeterminadas, X e Y modeladas por sendas v.a. discretas de distribución conjunta:

$Y \backslash X$	consonante	vocal	
consonante	0	3/4	$\to 3/4 \cdot p(y=cons)$
vocal	1/8	1/8	$\to 1/4\ p(y=voc.)$
	1/8 $p(x=cons)$	7/8 $p(x=voc.)$	1

¿$H(X)$?

$$H(X) = \sum_x p(x) \cdot \log_2 \frac{1}{p(x)} = 1/8 \cdot \log_2 8 + \frac{7}{8} \log_2 \frac{8}{7} = 0.544 \text{ bits}$$

$$p(x) = \sum_y p(x,y) \Rightarrow \{ 1/8 , (3/4 + 1/8) \} = \{ 1/8 , 7/8 \}$$

¿$H(X/Y = \text{consonante})$?

$$H(X/Y=\text{consonante}) = \sum_{xy} p(x,y) \cdot \log_2 \frac{1}{p(x/y)} = 1 \cdot \log_2 1 = 0 \text{ bits}$$

"consonante $\Rightarrow \{0, 1\}$

Si $Y = cons \to p(X=1) = 0$, $p(X=v) = 1$

¿$H(X/Y = \text{vocal})$?

$$H(X/Y = \text{vocal}) = \sum_{xy} p(x,y) \log_2 \frac{1}{p(x/y)} = \frac{3}{4} \cdot 1 + \frac{1}{8} \cdot 1 + \frac{1}{8} \cdot 1 = 1 \text{ bit}$$

$$p(x/y) = \frac{p(x,y)}{p(y)} = \frac{p(x,y)}{p(y)} \Rightarrow \{ 1/2 , 1/2 \}$$

y = vocal

Dado que si $Y = vocal$, la parte $p(x=voc) = p(x=cons)$, $\downarrow 1/2 \quad \downarrow 1/2$

¿ $H(X/Y)$?

$$H(X/Y) = \sum_{xy} p(x,y) \log_2\left(\frac{1}{p(x/y)}\right) = \frac{1}{8}\cdot 1 + \frac{1}{8}\cdot 1 = 1/4 = 0.25 \text{ bits}$$

$$p(x/y) = \frac{p(x,y)}{p(y)} \Rightarrow \{0, 1, 1/2, 1/2\}$$

$$H(X_1, X_2, \dots, X_n) \leq \sum_{i}^{n} H(X_i)$$

Demo:

$$H(X_1, X_2, \dots X_n) = H(X_1) + H(X_2/X_1) + \dots + H(X_n/X_1, \dots, X_{n-1})$$
$$\leq H(X_1) + H(X_2) + \dots + H(X_n)$$

Sea X v.a. discreta con $x \in \{x_1 \dots x_n\}$ de tal modo que:

$$p(x) \to p(x_1), p(x_2), \dots, p(x_n) \;,\; \sum_i p(x_i) = 1$$

$$q(x) \to q(x_1), q(x_2), \dots, q(x_n) \;,\; \sum_i q(x_i) = 1$$

definimos ahora otra distribución:

$$r_i = \lambda p_i + (1-\lambda) q_i \;,\; \lambda \in [0,1]$$

Se cumple que: $H_r(X) \geq H_p(X)$ y $H_r(X) \geq H_q(X)$

3) Agrupamiento

Sea X v.a. discreta con $x \in \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ con probabilidades $P_1, P_2, \dots, P_n$. Si además $\alpha_1, \dots, \alpha_n$ pueden estar distribuidos en m conjuntos A_m:

$$\boxed{H(X,A) = H(A) + H(X/A) = H(X) + H(A/X)}$$

Suponiendo que A es una función determinista:

$$H(X) = H(A) + H(X/A) = H(A) + p(A_1)H(X/A_1) + \dots + p(A_m)H(X/A_m)$$

⇒ La incertidumbre de X es la de saber primero la incertidumbre de los subgrupos y luego la incertidumbre de saber en qué caso nos encontramos.

7. Procesos de Markov

- Sean X, Y, Z v.a. discretas, de tal modo que $X \leftrightarrow Y \leftrightarrow Z$,

Se dice que $X \leftrightarrow Y \leftrightarrow Z$ es una relación markoviana $\Leftrightarrow$

$$\boxed{p(z/x,y) = p(z/y).}$$

$$\Rightarrow \boxed{p(x,y,z)} = p(x)\cdot p(y/x)\cdot p(z/y)\cdot \frac{p(y)}{p(y)} = p(x/y)\cdot p(y/z)\cdot p(z) =$$

$$= \boxed{p(x/y) = p(x/y,z)}$$

⇒ La cadena es en los dos sentidos.

También se cumple:

$$\boxed{p(x,z/y) = p(x/y)\cdot p(z/y,x) = p(x/y)\,p(z/y)}$$

⇒ Independencia condicionada ⇒ X, Z condicionadas por Y son independientes.

1.7. Teorema de procesado de información

- Sean X, Y, Z v.a. discretas, de modo que $X \to Y \to Z$, entonces:

$$\text{Si } X \to Y \to Z \Rightarrow \boxed{I(X;Y) \geq I(X;Z)}$$

Demo:

$$I(X;Y,Z) = I(X;Y) + I(X;Z/Y) = I(X;Z) + I(X;Y/Z)$$

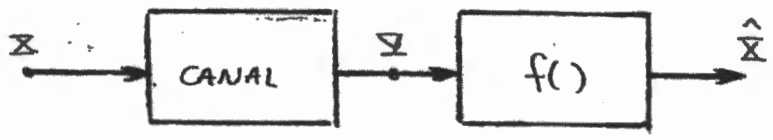
como $I(X;Z/Y) = 0$ y $I(X;Y/Z) \geq 0$

$$\Rightarrow I(X;Y) \geq I(X;Z)$$

- Este teorema viene a decir que nunca una manipulación más inteligente de los datos va a mejorar la información mutua entre las v.a. Es decir, dadas X e Y, podemos intentar mejorar $I(X;Y)$ mediante un procesado y obtener Z, pero siempre se cumplirá que $I(X;Y) \geq I(X;Z)$

1.8. Límite de Fano

- Supongamos el siguiente sistema:



Queremos estimar una v.a. X con una distribución $p(x)$. Observamos una v.a. Y relacionada con X a través de la distribución condicionada $p(y/x)$. A partir de Y, podemos calcular una función $f(Y) = \hat{X}$, que es una estimación de la v.a. X. ¿Cuál es la probabilidad de equivocarnos?

Definimos una v.a. de error E / $\begin{cases} E=0 \Leftrightarrow X=\hat{X} \\ E=1 \Leftrightarrow X\neq\hat{X} \end{cases}$

Hallando $H(E,X/Y) = H(E/Y) + H(X/Y,E) =$

$$= H(X/Y) + H(E/Y,X)$$

Con:

$H(E/Y,X)=0$

$H(E/Y) \leq H(E) = H(Pe) \leq 1$

$H(X/E,Y) = P(E=0)\cdot \underbrace{H(X/Y,E=0)}_{=0} + P(E=1)\cdot H(X/Y,E=1)$

$$\Rightarrow H(X/Y) = H(E/Y) + H(X/E,Y) \leq \underbrace{H(E)}_{\leq 1} + \underbrace{P(E=1)}_{Pe}\cdot \underbrace{H(X/Y,E=1)}_{\leq \log_2(|X|-1)}$$

$$\Rightarrow H(X/Y) \leq 1 + Pe\log_2(|X|-1)$$

$$\Rightarrow \boxed{Pe \geq \frac{H(X/Y)-1}{\log_2(|X|-1)} \geq \frac{H(X/Y)-1}{\log_2(|X|)}}$$

Tema 2. Propiedad Asintótica de Equipartición (A.E.P.)

2.1. Introducción

- La A.E.P. es una propiedad a la ley de los grandes números de la estadística. Esta propiedad establece que la probabilidad de una secuencia $S = X_1, X_2, \ldots, X_n$ procedente de una fuente F, es próxima a 2^{-nH}

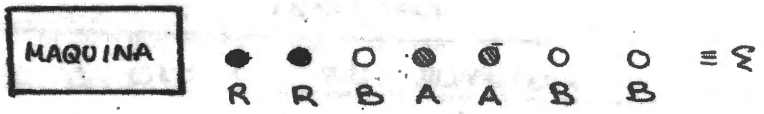
- Antes de definirla formalmente, veremos un ejemplo:

** Una máquina expulsa bolas de tres colores: rojo, blanco y azul, con distinta probabilidad y de forma independiente:

$p(\text{bola roja}) = r$

$p(\text{bola azul}) = a$

$p(\text{bola blanca}) = b$



- Si la máquina genera la secuencia ξ, la probabilidad de esa secuencia será:

$$p(\xi) = r^2 b^3 a^2$$

- En un caso genérico, la probabilidad de tener una secuencia cualquiera es:

$$P(S) = r^{N_r} \cdot b^{N_b} \cdot a^{N_a}$$

con:

$N_r \equiv$ número de bolas rojas

$N_b \equiv$ número de bolas blancas

$N_a \equiv$ número de bolas azules

· Tomando logaritmos y dividiendo por el número total de bolas, N:

$$\frac{1}{N}\log_2(p(S)) = \frac{Nr}{N}\log_2(r) + \frac{Nb}{N}\log_2(b) + \frac{Na}{N}\log_2(a)$$

con $\begin{cases} Nr/N = r \\ Na/N = a \\ Nb/N = b \end{cases}$, tenemos:

$$-\frac{1}{N}\log_2(p(S)) = +r\log_2(1/r) + b\log_2(1/b) + a\log_2(1/a)$$

$$\Rightarrow \log_2(p(S)) = -N \cdot H(\mathcal{X}) \Rightarrow \boxed{p(S) = 2^{-N \cdot H(\mathcal{X})}}$$

⇒ La probabilidad de que ocurra una secuencia S, típica, depende del número de símbolos y de la entropía de la fuente que la genera.

2.2. A.E.P. (Demostración)

- Sea F una fuente sin memoria, tal que todos los símbolos que salen de ella pertenecen a un conjunto de cardinal m. Si la fuente produce secuencias de n símbolos ⇒ podrá generar m^n secuencias diferentes.
- Si los símbolos son independientes e idénticamente distribuidos con probabilidad $p(x)$:

$$\boxed{p(S) = \prod_{i=1}^{n} p(x_i)}$$

⇒ tomando logaritmos: $\log_2 p(S) = \log_2 \prod_{i=1}^{n} p(x_i) = \sum_{i=1}^{n} \log_2 p(x_i)$

dividiendo por n:

$$-\frac{1}{n}\log_2 p(S) = -\frac{1}{n}\sum_{i=1}^{n}\log_2 p(x_i) \xrightarrow[(*)]{n\to\infty} E[\log_2 p(X)]$$

(*) Ley de los grandes números.

$$\Rightarrow -\frac{1}{n}\log_2 p(S) = E[\log_2 p(X)] = H(X)$$

$$\Rightarrow \boxed{p(S) = 2^{-nH(X)}}$$

- Se definen las secuencias típicas como las que verifican:

$$\boxed{2^{-n(H(X)+\xi)} \le p(x_1, x_2, \dots, x_n) \le 2^{-n(H(X)-\xi)}}$$

2.3. Propiedades de las secuencias típicas.

- Si $(x_1, \dots, x_n) \in A_\xi^n \Rightarrow H(X) - \xi \le -1/n \cdot \log_2(p(x_1, \dots, x_n)) \le H(X) + \xi$

  A_ξ^n: Conjunto de todas las secuencias típicas

- $\Pr\{A_\xi^n\} > 1 - \xi$ para un n suficientemente grande

- $|A_\xi^n| \le 2^{n(H(X)+\xi)}$

  $|A_\xi^n|$: Cardinal del conjunto de secuencias típicas.

Demo:

$$1 = \sum_{\forall \bar{x} \in \mathcal{X}^n} p(\bar{x}) \ge \sum_{\forall \bar{x} \in A_\xi^n} p(\bar{x}) \ge \sum_{\forall \bar{x} \in A_\xi^n} 2^{-n(H(X)+\xi)} =$$

$\mathcal{X}^n$: Conjunto de todas las secuencias

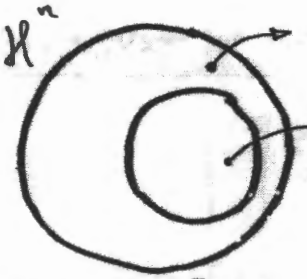
$$= |A_\xi^n| \, 2^{-n(H(X)+\xi)}$$

- $|A_\varepsilon^n| \geq (1-\varepsilon) \cdot 2^{n(H(X)-\varepsilon)}$ para n suficientemente grande

⇒ Deducimos que el conjunto de todas las secuencias típicas de una fuente tiene una probabilidad cercana a 1, entonces todas las secuencias típicas son casi equiprobables y el número de elementos del conjunto de dichas secuencias es cercano a $2^{nH(X)}$.

4. Compresión de datos.

- Supongamos la representación por diagramas de Venn del conjunto de todas las posibles secuencias de una fuente sin memoria:



Codificación de la fuente

a) Codificación de secuencias típicas

Asignando a todas las secuencias una longitud fija n:

$$nº \text{ bits} = \lg_2 2^{n[H(X)+\varepsilon]} = n[H+\varepsilon]$$

$$\Rightarrow \underbrace{\ell}_{\text{la longitud del código.}} \leq n(H+\varepsilon) + 1, \quad (\ell \in \mathbb{N})$$

Codificación de secuencias no típicas

Como $|[A_\xi^n]^c| \leq |\mathcal{H}^n| = |\mathcal{H}|^n$

$$\Rightarrow \text{nº bits} = \lg_2 |\mathcal{H}|^n + 1 = n \lg |\mathcal{H}| + 1$$

⇒ Consecuencia: necesitamos menos bits para codificar una secuencia típica que otra no típica.

- Si para poder realizar la decodificación, identificamos con un "0" a las secuencias típicas y con un "1" a las no típicas:

⇒ Necesitaremos un bit adicional para codificarlas.

- Calculando la longitud media:

$$\boxed{E[\ell(X^n)]} = \sum_{x^n \in A_\xi^n} p(x^n) \cdot l(x^n) + \sum_{x^n \in (A_\xi^n)^c} p(x^n) \cdot l(x^n) \leq$$

$$\leq \underbrace{\sum_{x^n \in A_\xi^n} p(x^n)}_{P_r\{A_\xi^n\}} \cdot \underbrace{[n(H(X)+\xi)+2]}_{\text{cte}} + \underbrace{\sum_{x^n \in (A_\xi^n)^c} p(x^n)}_{P_r\{(A_\xi^n)^c\}} \underbrace{[n \cdot \lg_2 |\mathcal{H}| + 2]}_{\text{cte}} =$$

$$= 2\underbrace{\left[P_r\{A_\xi^n\} + P_r\{(A_\xi^n)^c\}\right]}_{=1} + P_r\{A_\xi^n\}[n(H(X)+\xi)] +$$

$$+ \underbrace{P_r\{(A_n^n)^c\}}_{\to 1} \cdot n \lg_2 |\mathcal{H}| \leq$$

$$\leq n(H(X)+\xi) + 2 + n \cdot \lg_2 |\mathcal{H}| = \boxed{n(H(X)+\xi')}$$

con $\xi' = \xi + \xi \lg_2 |\mathcal{H}| + \frac{2}{n}$

⇒ La longitud media en bits de una cadena de bits está acotada por la entropía.

Tema 3. Cadenas de Markov.

3.1. Procesos estocásticos

- Definimos un proceso estocástico discreto en el tiempo como una secuencia indexada en el tiempo de variables aleatorias definido por una distribución de probabilidad conjunta

$$Pr\{X_1 = x_1, \dots, X_n = x_n\} = p(x_1, \dots, x_n) \quad \forall (x_1, \dots, x_n) \in \mathcal{H}^n$$

- Un proceso estocástico es estacionario, si sus estadísticas no varían con el tiempo:

$$Pr\{X_1 = x_1, \dots, X_n = x_n\} = Pr\{X_{1+l} = x_1, \dots, X_{n+l} = x_n\}$$

$$\forall (x_1, \dots, x_n) \in \mathcal{H}^n$$

3.2. Cadenas de Markov de orden 1.

- Un proceso estocástico es una cadena de Markov de orden 1

$$\Leftrightarrow Pr\{X_n = x_n / X_{n-1} = x_{n-1}, \dots, X_1 = x_1\} = Pr\{X_n = x_n / X_{n-1} = x_{n-1}\}$$

$\Leftrightarrow$ Solo depende del término anterior en la secuencia.

* Por tanto podremos expresar la probabilidad conjunta:

$$p(x_1, x_2, \dots, x_n) = p(x_1) \cdot p(x_2/x_1) \cdot p(x_3/x_2) \cdot \dots \cdot p(x_n/x_{n-1})$$

- Decimos que la cadena de Markov es homogénea $\Leftrightarrow$ las probabilidades condicionales no dependen del tiempo.

$$\Leftrightarrow Pr\{X_{n+1} = b / X_n = a\} = Pr\{X_2 = b / X_1 = a\}$$

- Un proceso de Markov se caracteriza por su estado inicial (estado anterior) y por la matriz de transiciones.

- El estado inicial se expresa por un vector fila que nos dice la probabilidad de estar en un estado para un tiempo t_0.

- La matriz de transiciones es una matriz $n \times n$ (caso homogéneo) donde n es el número de estados posibles, y nos dice la probabilidad de pasar al estado j estando en el estado i.

⇒ De modo que la probabilidad de un estado en el nivel temporal k se puede expresar:

$$\vec{\mu}(k) = \vec{\mu}(k-1) \cdot P$$

con $\vec{\mu}(k) = [x_{k,1}\ x_{k,2} \dots x_{k,n}]$ = vector de estado

$$P = \begin{bmatrix} P_{11} \dots & P_{1n} \\ \vdots & \vdots \\ P_{n1} \dots & P_{nn} \end{bmatrix} \equiv \text{matriz de transiciones}$$

Con $P_{ij} = P_r \{ X_{n+1} = j / X_n = i \} \ \forall i, j \in \mathbb{N}^n$

Además se cumple: $p(x_{n+1}) = \sum_{\forall x_n} p(x_n) \cdot P_{x_n, x_{n+1}}$

- Se define la cadena de Markov **irreducible** si desde cualquier estado, se puede volver a él después de un número finito de saltos.

-Una cadena de Markov es **periódica** si a partir de un cierto instante los estados se repiten con periodo p.

-Si una cadena de Markov es homogénea, irreducible y aperiódica, entonces existe una distribución $\vec{\mu}$ que cumple:

$$\vec{\mu} = \vec{\mu} \cdot P$$

⇒ Generamos una cadena estacionaria, ya que el estado no depende del tiempo.

$\vec{\mu_0} = \lim_{k \to \infty} \vec{\mu}(k)$ este límite, si existe converge a $\vec{\mu}$ que hace que la cadena sea estacionaria

⇒ Con las hipótesis anteriores, las probabilidades tienden a ser constantes con el tiempo.

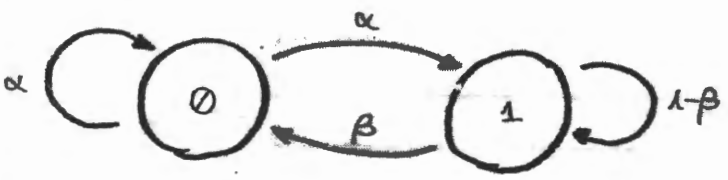
** Ejemplo

$$P = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix} \begin{matrix} 0 \\ 1 \end{matrix} \quad ; \quad \vec{\mu}(1) = (\mu_1, \mu_2)$$

(columnas: 0, 1)

de tal forma que $\vec{\mu}(2) = \vec{\mu}(1) \cdot P$

Gráficamente



* homogénea
* irreducible
* aperiódica

¿Cuál será la distribución estacionaria?

$$\vec{\mu} \cdot P = \vec{\mu} \Rightarrow \begin{cases} \mu_1(1-\alpha) + \beta\mu_2 = \mu_1 \Rightarrow \beta\mu_2 = \alpha\mu_1 \\ \mu_1\alpha + (1-\beta)\mu_2 = \mu_2 \Rightarrow \beta\mu_2 = \alpha\mu_1 \end{cases}$$

Añadimos la condición: $\mu_1 + \mu_2 = 1$

Obtenemos despejando:

$$\mu_1 = \frac{\beta}{\alpha+\beta} \quad ; \quad \mu_2 = \frac{\alpha}{\alpha+\beta}$$

3.3. Cadenas de Markov de orden n.

- Una cadena de Markov de orden n se define como un proceso estocástico que verifica:

$$Pr\{X_{n+5} / X_1, X_2, \dots X_n, \dots, X_{n+4}\} = Pr\{X_{n+5} / X_1, \dots, X_n\}$$

⇒ Depende de los n estados anteriores.

- Vamos a ver que cualquier cadena de orden n se puede expresar como una cadena de orden 1.

** Ejemplo (Vamos a particularizar para n=2)

X_n	X_{n-1}	X_{n-2}	S_k	S_{k-1}	
0	0	0	0	0	→ p
1	0	0	2	0	→ p'
0	0	1	0	1	→ q
1	0	1	2	1	→ q'
0	1	0	1	2	→ r
1	1	0	3	2	→ r'
0	1	1	1	3	→ s
1	1	1	3	3	→ s'

S_k S_{k-1}

⇒ Hemos conseguido una cadena de orden 1.

- La matriz P para el instante k:

$$P(k) = \begin{bmatrix} p & \emptyset & p' & \emptyset \\ q & \emptyset & q' & \emptyset \\ \emptyset & r & \emptyset & r' \\ \emptyset & s & \emptyset & s' \end{bmatrix}$$

* De manera que: $\vec{\mu}(k) = \vec{\mu}(k-1) \cdot P(k) = \ldots =$

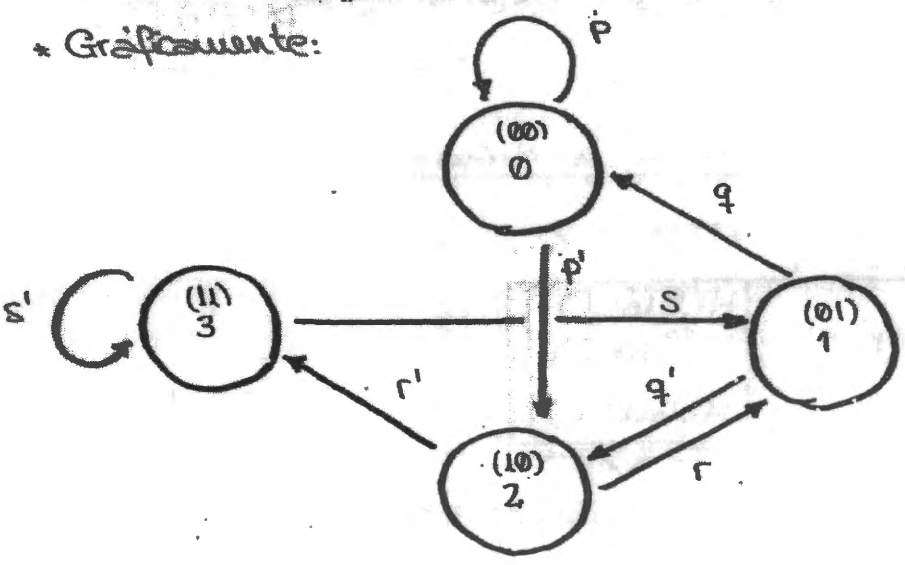
$$= \vec{\mu}(1) \cdot P(2) \cdot P(3) \cdot \ldots \cdot P(k-1) \cdot P(k)$$

$\Rightarrow$ No es una cadena homogénea (suponiendo esta relación).

* Si la cadena fuera homogénea:

$$\vec{\mu}(k) = \vec{\mu}(1) \cdot P^{k-1}$$

* Gráficamente:



* Al igual que antes, si el proceso es invariante (homogéneo), irreducible y aperiódico, existe $\vec{\mu}$ que hace estacionaria la cadena

$$\Rightarrow \vec{\mu}_\infty = \lim_{k \to \infty} \vec{\mu}(k) \Big|_{\exists} = \vec{\mu}_{ESTAC.}$$

3.4. Incertidumbre de un proceso markoviano.

- Supongamos un conjunto de v.a. discretas $X_1, X_2, \dots X_n$ que forman un proceso markoviano de orden 1.

Se establecen:

$$\vec{\mu}(1) = [P_a \; P_b \dots P_n] \equiv \text{vector del estado 1º}$$

$$TT(k) = [p_{ij}] \equiv \text{matriz de transiciones}$$

- La incertidumbre de llegar a $X(k)$ estando en $X(k-1)$:

$$H(X(k)/X(k-1)) = \vec{\mu}(k-1) \begin{bmatrix} H(\text{fila 1}) \\ H(\text{fila 2}) \\ \vdots \\ H(\text{fila n}) \end{bmatrix}$$

$$H(A/B) = \sum_b P_b \sum_a \left(p(a/b) \cdot \log_2 \frac{1}{p(a/b)} \right)$$

Por tanto, si el proceso es homogéneo:

$$H(X_1, \dots, X_n) = \sum_i H(X_i / X_1, \dots, X_{i-1}) =$$

$$= \sum_k \vec{\mu}(k-1) \cdot \begin{bmatrix} H(\text{fila 1}) \\ H(\text{fila 2}) \\ \vdots \\ H(\text{fila n}) \end{bmatrix}$$

independiente de k

$$\Rightarrow H(X_1, \dots, X_n) = \sum_i H(X_i / X_{i-1}) = \sum_k \vec{\mu}(k-1) \cdot \begin{bmatrix} H(\text{fila 1}) \\ H(\text{fila 2}) \\ \vdots \\ H(\text{fila n}) \end{bmatrix}$$

5. Tasa de incertidumbre (Tasa de entropía)

Se define la tasa de incertidumbre como:

$$H(X) = \lim_{n\to\infty} \frac{H(X_1, X_2, \dots X_n)}{n}$$ promediada

- También se puede definir mediante otra métrica:

$$H'(X) = \lim_{n\to\infty} H(X_n / X_1, \dots, X_{n-1})$$ incremental

Cumpliendose que si existen estos límites:

Si converge $H'(X) \Rightarrow$ converge $H(X)$

Demo:

Tenemos dos secuencias:

$a_1, a_2, \dots, a_n \to a$

$b_1, b_2, \dots, b_n \to a$?? / $b_i = \frac{a_1 + \dots + a_n}{i}$

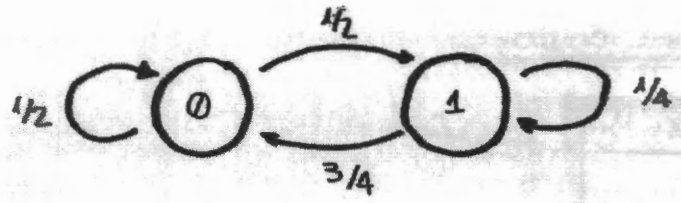
Por la definición de límite:

$\forall \varepsilon \; \exists N / \forall i > N \;,\; \varepsilon \geq |a_i - a|$

con $b_n = \frac{\sum_i^n a_i}{n}$

$|b_n - a| = |\frac{\sum_i^n a_i}{n} - a| \leq |a_n - a| \Rightarrow$ si se cumple que la secuencia "b" converge, entonces "a" también converge.

** Ejemplo:



proceso markoviano de orden 1.

$$\Rightarrow \Pi = \begin{bmatrix} 1/2 & 1/2 \\ 3/4 & 1/4 \end{bmatrix} \Rightarrow \text{proceso homogéneo.}$$

Si $\vec{\mu}(1) = (1, 0)$

a) Calcular $\vec{\mu}(2)$, $\vec{\mu}(3)$, $\vec{\mu}(4)$

$$\vec{\mu}(2) = \vec{\mu}(1)\,\Pi = [1, 0] \begin{bmatrix} 1/2 & 1/2 \\ 3/4 & 1/4 \end{bmatrix} = [1/2, 1/2]$$

$$\vec{\mu}(3) = \vec{\mu}(2)\cdot\Pi = [1/2, 1/2] \begin{bmatrix} 1/2 & 1/2 \\ 3/4 & 1/4 \end{bmatrix} = [1/4 + 3/8, 1/4 + 1/8] =$$

$$= [5/8, 3/8]$$

$$\vec{\mu}(4) = \vec{\mu}(3)\cdot\Pi = [5/8, 3/8] \cdot \begin{bmatrix} 1/2 & 1/2 \\ 3/4 & 1/4 \end{bmatrix} = [5/16 + 9/32, 5/16 + 3/32] =$$

$$= [19/32, 13/32]$$

b) Las entropías condicionadas $H(X_1)$, $H(X_2/X_1)$, $H(X_3/X_2)$, $H(X_4/X_3)$:

$$H(X_1) = \sum_i p_i \log_2 \frac{1}{p_i} = 0$$

$$H(X_2/X_1) = \sum_{x_1} p(x_1) \sum_{x_2} p(x_2/x_1) \cdot \log_2 \frac{1}{p(x_2/x_1)} = \vec{\mu}(x_1) \cdot \begin{bmatrix} H(1/2, 1/2) \\ H(3/4, 1/4) \end{bmatrix}$$

$$= (1, 0) \cdot \begin{bmatrix} 1 \\ 0.81 \end{bmatrix} = 1$$

$$H(X_3/X_2) = \vec{\mu}(X_2) \cdot \begin{bmatrix} H(1/2, 1/2) \\ H(3/4, 1/4) \end{bmatrix} = (1/2, 1/2) \cdot \begin{bmatrix} 1 \\ 0.81 \end{bmatrix} = 0.905$$

$$H(X_4/X_3) = \vec{\mu}(X_3) \cdot \begin{bmatrix} H(1/2, 1/2) \\ H(3/4, 1/4) \end{bmatrix} = (3/8, 3/8) \begin{bmatrix} 1 \\ 0.81 \end{bmatrix} = 0.92$$

c) La entropía conjunta $H(X_1, X_2, X_3, X_4)$

$$H(X_1, X_2, X_3, X_4) = H(X_1) + H(X_2/X_1) + H(X_3/X_2) + H(X_4/X_3) =$$

$$= 0 + 1 + 0.905 + 0.92 = 2.83$$

d) Calcular $\vec{\mu}_\infty$ como el $\lim_{k\to\infty} \vec{\mu}(k)$

$$\vec{\mu}_\infty / \vec{\mu} = \vec{\mu} \cdot \Pi = [\mu_{E_1}, \mu_{E_2}]$$

$$\Rightarrow [\mu_{E_1}, \mu_{E_2}] = [\mu_{E_1}, \mu_{E_2}] \begin{bmatrix} 1/2 & 1/2 \\ 3/4 & 1/4 \end{bmatrix} =$$

$$\Rightarrow \begin{cases} \mu_{E_1} = \mu_{E_1}/2 + \frac{3}{4}\mu_{E_2} \Rightarrow \frac{\mu_{E_1}}{2} = \frac{3}{4}\mu_{E_2} \\ \mu_{E_2} = \mu_{E_1}/2 + \frac{1}{4}\mu_{E_2} \Rightarrow \frac{3}{4}\mu_{E_2} = \mu_{E_1}/2 \end{cases}$$

Con $\mu_{E_1} + \mu_{E_2} = 1$

$$\Rightarrow \mu_{E_1} = 2\left(\frac{3}{4}(1 - \mu_{E_1})\right) = \frac{3}{2} - \frac{3}{2}\mu_{E_1}$$

$$\left.\begin{aligned} \mu_{E_1} &= \frac{2}{5} \cdot \frac{3}{2} = \frac{3}{5} \\ \mu_{E_2} &= \frac{2}{5} \end{aligned}\right\} \vec{\mu}_\infty = \left[\frac{3}{5}, \frac{2}{5}\right]$$

e) Calcular la tasa de incertidumbre de la secuencia:

$$H(X) = \lim_{n\to\infty} H(X_n/X_1 \dots X_{n-1}) = \lim_{n\to\infty} H(X_n/X_{n-1}) \cdot$$

$$= \left[\frac{3}{5}, \frac{2}{5}\right] \begin{bmatrix} 1 \\ 0.81 \end{bmatrix} = 0.924$$

Tema 4. Compresión de datos. Codificación de fuente.

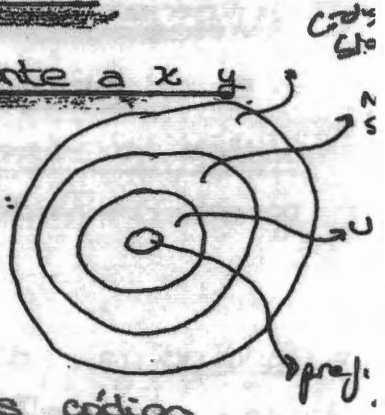
4.1. Definiciones

- Un código fuente C de una variable aleatoria X es una correspondencia del rango de X con el conjunto de cadenas de símbolos, de longitud finita, de un alfabeto D-ario.

- Se nota con $C(x)$ la palabra código correspondiente a x y $l(x)$ la longitud de $C(x)$.

- Se define la longitud media de código C como:

$$L(C) = \bar{L} = \sum_{x \in X} p(x) l(x)$$



- Un código es no singular si todas sus palabras código son distintas.

- Un código es unívocamente decodificable si dos secuencias de fuente distintas tienen dos secuencias de código distintas.

$$\forall\ x_1 x_2 \dots x_i \neq x_1' x_2' \dots x_i' \Rightarrow C(x_1)C(x_2)\dots C(x_i) \neq C(x_1')C(x_2')\dots C(x_i')$$

* Algoritmo de Sardinas-Patterson: Algoritmo para determinar si un código es unívocamente decodificable

C(X)	Sufijo 1	Sufijo 2	Sufijo 3	Sufijo 4
101	0	11	00	100
1100		01	10	110 → Palabra código
011			1	
001				
110				
1110				

⇒ No unívocamente decodificable.

⇒ El algoritmo de Sardinas-Patterson consiste en buscar palabras código que puedan ser prefijo de otras ya que esas serán las conflictivas.

⇒ A partir de ahí, se desarrollan los sufijos resultantes y se generan las palabras de las que son prefijos. Si durante el desarrollo encontramos alguna palabra código, entonces el código es no unívocamente decodificable

⇒ El algoritmo termina cuando encontremos una palabra código o cuando repetimos la secuencia de sufijos (no se generan sufijos nuevos)

- Un código se define como prefijo o instantáneo si ninguna palabra es prefijo de otra.

2. Desigualdad de Kraft

- Para un código instantáneo (prefijo), generado por un alfabeto D-ario, se verifica que las longitudes de las palabras $l_1, l_2, \dots l_m$ satisfacen:

$$\sum_i D^{-l_i} \leq 1$$

⇒ Consecuentemente, dadas las longitudes de un código que satisfagan esta inecuación, existirá un código instantáneo con esas longitudes.

4.3. Codigos óptimos

- Ahora queremos encontrar el código prefijo con la mínima longitud. Esto es, queremos un código cuya longitud media sea mínima y además cumpla la inecuación de Kraft.

$$\begin{cases} \overline{L} = \sum_i p_i l_i \\ \sum_i D^{-l_i} \leq 1 \end{cases}$$

- Aplicando multiplicadores de Lagrange obtenemos (...)

$$J = \sum_i p_i l_i + \lambda \sum_i D^{-l_i}$$

$$\Rightarrow \frac{\partial J}{\partial l_i} = p_i - \lambda D^{-l_i} \ln D = 0$$

$$\Rightarrow \boxed{D^{-l_i} = \frac{p_i}{\lambda \ln D}}$$

* Tomando sumatorios:

$$\sum_i D^{-l_i} = \frac{\sum p_i}{\lambda \ln D} = \boxed{1 = \frac{1}{\lambda \ln D}} \quad \lambda = \frac{1}{\ln D}$$

* Volviendo a la primera expresión:

$$\boxed{p_i = D^{-l_i}}$$

$$\Rightarrow \boxed{l_{i\,opt} = -\log_D p_i}$$

Si no da entero, buscamos el entero más próximo

$$\Rightarrow \boxed{\overline{L}_{opt} = \sum_i p_i l_{i\,opt} = +\sum_i p_i \log_D \frac{1}{p_i} = H_D(X)}$$

- Sin embargo, la realización de estos códigos sólo es posible si la fuente tiene una distribución de probabilidades D-ádica, ya que l_i deben ser números naturales.

$$\text{fuente D-ádica} \Leftrightarrow p_i = \frac{1}{2^n} \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16} \dots\right)$$

4. Códigos de Shannon-Fano

- Si la fuente no sigue una distribución de probabilidad D-ádica, la forma de generar un código más próximo al óptimo sin perder información será:

$$\boxed{l_i = \left\lceil \log_D \frac{1}{p_i} \right\rceil}$$

$$\Rightarrow D^{-\log_D\left(\frac{1}{p_i}\right)} \geq D^{\left\lceil -\log_D\left(\frac{1}{p_i}\right)\right\rceil}$$

Entonces:

$$1 = \sum_i D^{-\log_D\left(\frac{1}{p_i}\right)} \geq \sum_i D^{\left\lceil -\log_D\left(\frac{1}{p_i}\right)\right\rceil}$$

Con $\log_D\left(\frac{1}{p_i}\right) \leq l_i < \log_D\left(\frac{1}{p_i}\right) + 1$

$$\overset{(\dots)}{\Rightarrow} \boxed{H_D(X) \leq \overline{L} < H_D(X) + 1}$$

Teorema: Sean $l_1, l_2, \dots, l_m$ las longitudes optimas de un código de una fuente de distribución p en un alfabeto D-ario. Entonces, si $\overline{L}$ es la longitud media del código, se cumple:

$$\boxed{H_D(X) \leq \overline{L} < H_D(X) + 1}$$

Teorema: La longitud media de un código prefijo D-ario de una variable aleatoria X, $L(C) = \overline{L}$, es mayor o igual a la entropía $H_D(X)$:

$$\boxed{\overline{L} \geq H_D(X)} \rightsquigarrow \text{El límite de la } \overline{L} \text{ es la entropía...}$$

- Supongamos ahora un sistema en el cual enviamos una secuencia de n símbolos de $\mathcal{X}$. Si consideramos independientes e idénticamente distribuidos con probabilidad $p(x_i)$:

$$L_n = \frac{1}{n} \sum p(x_1, x_2, \ldots, x_n)\, \ell(x_1, x_2, \ldots, x_n) = \frac{1}{n} L(\mathcal{X}_1, \mathcal{X}_2, \ldots \mathcal{X}_n)$$

$$\Rightarrow H(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n) \leq L(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n) < H(\mathcal{X}_1, \mathcal{X}_2, \ldots \mathcal{X}_n) + 1$$

$$\text{Con } H(\mathcal{X}_1, \mathcal{X}_2, \ldots \mathcal{X}_n) \overset{(iid)}{=} \sum_i H(\mathcal{X}_i) = n H(\mathcal{X})$$

$$\Rightarrow \boxed{H(\mathcal{X}) \leq L_n < H(\mathcal{X}) + \frac{1}{n}}$$

- También podemos hacer un razonamiento similar para una secuencia de símbolos de un proceso estocástico que no es necesariamente independiente e idénticamente distribuido

$$H(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n) \leq L(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n) < H(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n) + 1$$

$$\Rightarrow \boxed{\frac{H(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n)}{n} \leq L_n < \frac{H(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n)}{n} + \frac{1}{n}}$$

- Si además el proceso es estacionario se cumple que:

$$\boxed{\lim_{n \to \infty} \frac{H(\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n)}{n} \to \mathcal{H}(\mathcal{X})} \quad \text{(tasa de entropía)}$$

$$\Rightarrow L_n \to \mathcal{H}(\mathcal{X})$$

4.5. Teorema de McMillan

- Las longitudes de palabras código de un código unívocamente decodificable deben satisfacer también la inecuación de Kraft

$$\sum_i D^{-l_i} \leq 1$$

⇒ Dadas una serie de longitudes que satisfagan esta inecuación, es posible construir un código unívocamente decodificable con esas longitudes de palabras.

4.6. Códigos Huffman

- Los códigos Huffman son códigos prefijos y óptimos, dada una distribución de probabilidades.

- Algoritmo de construcción

1º Se listan todos los sucesos posibles y se ordenan de mayor probabilidad a menor.

** Ejemplo

X	prob(x)				
1	0.6 —0—	0.6 —0—	0.6 —0—	0.6 —0—	1
2	0.2 —10—	0.2 —10—	0.2 —10—	0.4 —1—	
3	0.1 —110—	0.1 —110—	0.2 —11—		
4	0.05 —1110—	0.1 —111—			
5	0.05 —1111—				

2º Sumamos las dos probabilidades de menor peso y volvemos a ordenar. Continuamos con este sistema hasta tener solo dos sucesos

Al llegar a este punto, asignamos a cada suceso la palabra código correspondiente, siguiendo la estructura de árbol

$C(1) = 0$

$C(2) = 10$

$C(3) = 110$

$C(4) = 1110$

$C(5) = 1111$

- Hemos supuesto un alfabeto binario, en general, si quisiéramos codificar en un alfabeto D-ario, iríamos desde:

$$1 \to D \to (D-1)+D \to D+(D-1)\cdot 2 \to \dots$$

El número de hojas que tendríamos en el árbol, (número de palabras código):

$$\boxed{1+(D-1)K}$$

con K el número de niveles del árbol.

** Ejemplo: Codificar X en D-ario, D=3

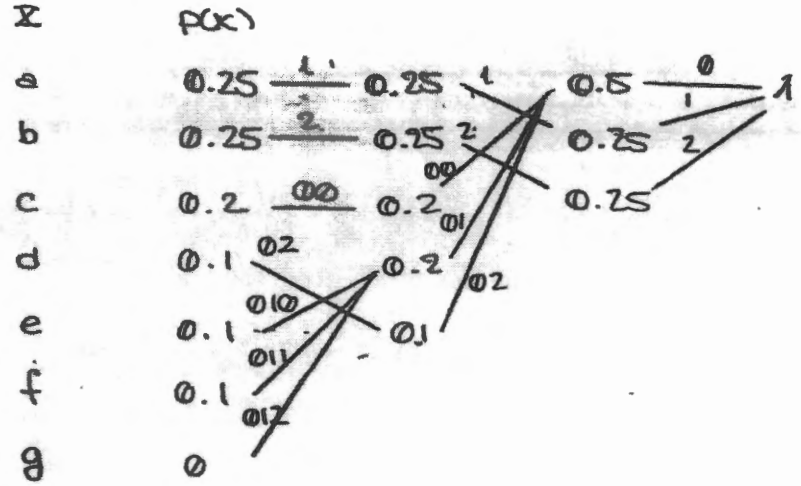
X	P(X)
a	0.25
b	0.25
c	0.2
d	0.1
e	0.1
f	0.1

No podríamos codificar ya que para D=3 tenemos:

1, 3, 5, 7, ... ⇒ No tenemos árboles de 6 hojas.

Para solucionar esto, agregamos otro suceso g con probabilidad 0, de modo que tengamos 7 palabras y sea posible la codificación

Ahora agruparemos probabilidades de 3 en 3.



$C(a) = 1$

$C(b) = 2$

$C(c) = 00$

$C(d) = 02$

$C(e) = 010$

$C(f) = 011$

$C(g) = 012$ ← Esta palabra no se utiliza ($p(g) = 0$)

- Propiedades de los códigos Hoffman

1) Si $p_i > p_j \Rightarrow l_i < l_j$

2) La palabra más larga tiene hermanos, es decir, la rama más larga no puede estar sola, porque quitando esa rama reduciríamos el código ($\Rightarrow$ código no óptimo)

3) Se pueden hacer codigos óptimos de manera que las palabras menos probables sean hermanas. No hace falta que tengan los mismos antecesores

4.7. Códigos Shannon - Elias - Fano → Alfabético !!!

- El código Shannon-Elias-Faro genera códigos prefijos pero no óptimos como los de Huffman.

- El algoritmo de generación del código es el siguiente:

** Ejemplo

$X = \{a, e, i, o, u, s\}$, con $p(X) = \left\{\frac{4}{21}, \frac{5}{21}, \frac{2}{21}, \frac{1}{21}, \frac{3}{21}, \frac{6}{21}\right\}$

En forma de lista (sin ordenar):

X	p(x)
a	4/21
e	5/21
i	2/21
o	1/21
u	3/21
s	6/21

Ahora calculamos F(x) (función distribución) como la suma de p(x) acumuladas

Seguidamente calculamos $\overline{F}(x)$ como:

$$\overline{F}(x) = \frac{1}{2}\left[F(x) + F(x-1)\right]$$

y lo pasamos a binario

$l(x) = \lceil \log_2 \frac{1}{p(x)} \rceil + 1$

X	p(x)	F(x)	$\overline{F}(x)$	$\overline{F}(x)$ en binario	l(x)
a	4/21	4/21	4/42	000110	4
e	5/21	9/21	13/42	010011	4
i	2/21	11/21	20/42	011110	5
o	1/21	12/21	23/42	100011	6
u	3/21	15/21	27/42	101001	4
s	6/21	21/21	36/42	110110	3

Por último calculamos la longitud de cada palabra de la forma:

$$l(x) = \left[\log_2 \frac{1}{p(x)}\right] + 1$$

y nos quedamos con tantos dígitos binarios como indique l(x)

* De forma que:

C(a) = 0001
C(e) = 0100
C(i) = 0111
C(o) = 10001
C(u) = 1010
C(s) = 110

→ Recortando adecuadamente →

C(a) = 00
C(e) = 010
C(i) = 011
C(o) = 100
C(u) = 101
C(s) = 11

NOTA: Si se recorta adecuadamente:

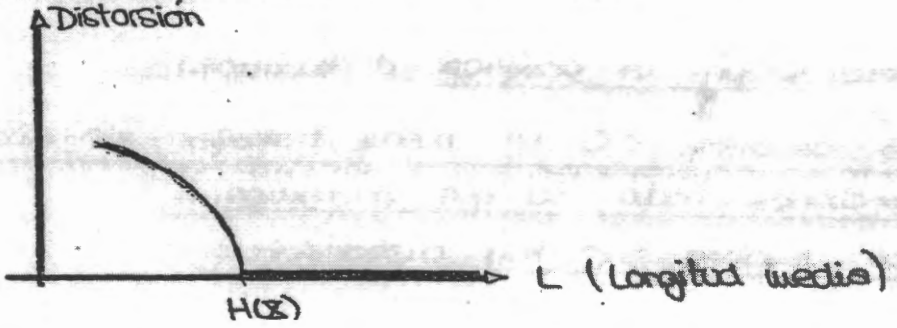
• $H(x) \leq L(c) \leq H(x)+2$.

8. Código Lempel-Ziv

Tema 5 Capacidad de canal

5.1. Introducción

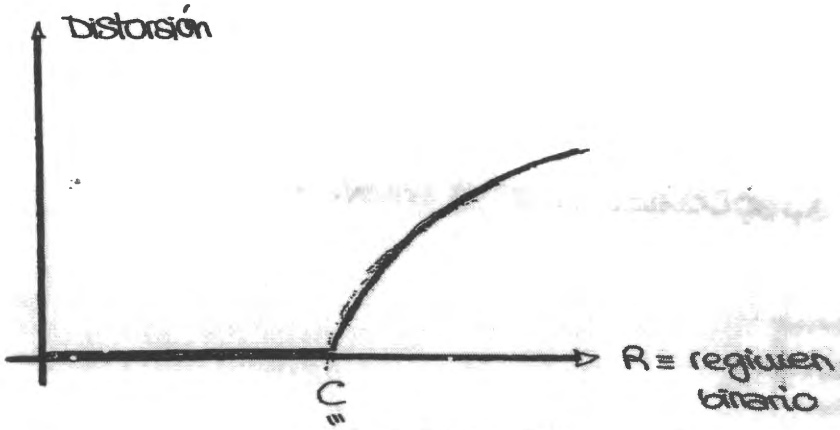
- En el tema anterior se vió que teníamos un límite para la codificación de fuente sin distorsión que venía impuesto por la entropía de tal modo:



- Haciendo que la codificación óptima fuera:

$$H(X) \leq L \leq H(X) + 1$$

- En este tema vamos a analizar otro aspecto de la teoría de la información: el canal.
- Se puede definir un canal como una entrada y una salida, que puede ser distinta debido a la distorsión.
- El objetivo de este tema es estudiar y determinar el mayor rendimiento que podemos hacer de un canal.
- Análogamente tenemos una gráfica que determina la distorsión en función del regimen binario (nº bits por s).
- Nuestro objetivo, por tanto, será trabajar sin distorsión pero aprovechando el canal al máximo.



- Por tanto, según la gráfica deducimos:

  - Para valores $< C$ se puede trabajar sin distorsión, acercandonos todo lo que queramos.
  - Para valores $> C$ hay distorsión.

5.2. Capacidad del canal

- Un canal se modela como la relación de dos variables aleatorias

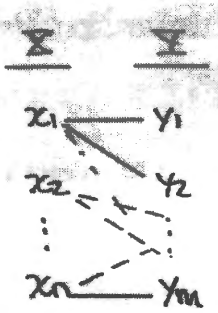
$$X \xrightarrow{p(y/x)} Y$$

- Esta relación que hay entre ambas será la información mutua por lo que la capacidad del canal será, por definición, el valor máximo de esa información mutua entre X e Y

$$I(X;Y) = f(y,x) \Rightarrow \text{nos da una medida de la información de } X \text{ que llega a } Y$$

$$\Rightarrow \boxed{C = \max_{p(x)} I(X;Y)}$$

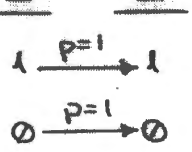
- Si un canal sin memoria está definido por dos v.a. X, Y:



⇒ Supongamos también todas sus relaciones $P(y_i/x_j)$, entonces este canal queda totalmente caracterizado por una matriz $P_{n\times m}$ / $P_{ij} = P(y_j/x_i)$

5.3. Modelos triviales de canales

- Canal binario sin ruido



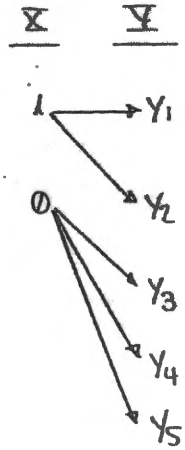
$$\left.\begin{array}{l} \\ \\ \end{array}\right\} I(X;Y) = H(X) - \cancel{H(X/Y)}^{\,0} = H(X)$$

$$\Rightarrow \boxed{C = \max H(X) = \lg_2 |X| = 1 \text{ bit}}$$

⇒ $I(X;Y)$ nos da una medida del flujo que hay

⇒ C = flujo máximo.

- Canal ruidoso con salidas no solapadas



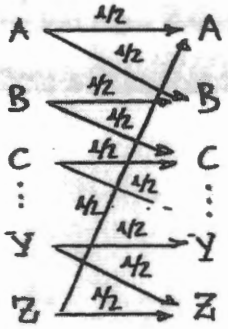
$$\Rightarrow I(X;Y) = H(X) - \cancel{H(X/Y)}^{\,0} = H(X)$$

$$\Rightarrow \boxed{C = \max H(X) = \lg_2 |X| = 1 \text{ bit}}$$

- Máquina de escribir ruidosa

* Teclado que con probabilidad 0.5 acierta y con probabilidad 0.5 pulsa la tecla siguiente.

X Y Supuesto 26 símbolos.



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

$$H(Y/X) = \sum_x p(x) \sum_y p(y/x) \log_2 \frac{1}{p(y/x)}$$

Para cada caso particular:

$$H(Y/X = x_i) = H(1/2, 1/2) = 1$$

$$\Rightarrow H(Y/X) = \sum_x p(x) = 1 \equiv \text{bit q perdemos por incertidumbre.}$$

$$\Rightarrow I(X;Y) = H(Y) - 1 \Rightarrow \boxed{C = \max (H(Y) - 1) =}$$

$$= \log_2 26 - 1 = \boxed{\log_2 13}$$

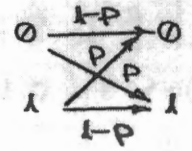
* Viendo los resultados, podemos determinar que no puedo distinguir 26 símbolos, pero sí 13 con total fiabilidad

$\Rightarrow$ Puedo distinguir 13 subconjuntos de entradas, que podemos asignar a 13 símbolos diferentes, pero no puedo enviar 26.

$\Rightarrow$ Limitación de la entrada.

- **Canal binario simétrico (BSC)**

$$P(Y/X) = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} \begin{matrix} \to x=0 \\ \to x=1 \end{matrix}$$
$$\begin{matrix} \downarrow & \downarrow \\ y=0 & y=1 \end{matrix}$$



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

Son iguales

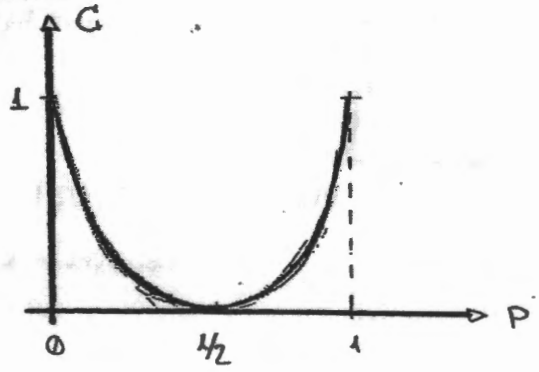
$$H(Y/X) = p(0)\cdot H(Y/X=0) + p(1)\cdot H(Y/X=1) =$$

$$= (\underbrace{p(0)+p(1)}_{1})\,\underbrace{H(Y/X=0)}_{H(1-p,p)} = H(1-p,p)$$

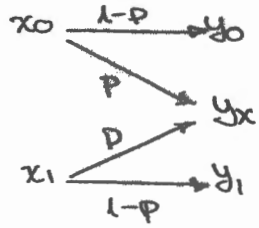
$$\Rightarrow I(X;Y) = H(Y) - H(1-p,p) \Rightarrow \boxed{C = \lg_2|Y| - H(1-p,p) = 1 - H(1-p,p)}$$

$H(p)$ (para abrev)

$\Rightarrow$ Gráficamente:



- **Canal binario con borrado**



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

$y_x \to$ se puede suponer como una señal borrosa.

$$H(Y/X) = p(x_0)\cdot H(Y/X=x_0) + p(x_1)\cdot H(Y/X=x_1) = \ldots = H(1-p,p) = H(p)$$

$$\Longrightarrow \left[P(y_0), P(y_x), P(y_1)\right] = \left[P_1, P_2\right] \begin{bmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{bmatrix} = \left[P_1(1-p), p, P_2(1-p)\right]$$

Si asignamos a x_0 la probabilidad P_1 y a x_1, P_2, entonces:

$P(y) = P(x) \cdot \underbrace{P}_{\text{matriz del canal}}$: $P = \begin{bmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{bmatrix}$

$$\Rightarrow H(Y) = H(P_1(1-p), p, P_2(1-p)) = H(p) + H(P_1(1-p), P_2(1-p))$$

$$= H(p) + (1-p) P_1 \log_2 \frac{1}{P_1(1-p)} + (1-p) P_2 \log_2 \frac{1}{P_2(1-p)} =$$

$$= H(p) + (1-p)\left[P_1(-\log_2(1-p) - \log_2 P_1) + P_2(-\log_2(1-p) - \log_2 P_2)\right] =$$

$$= H(p) + (1-p)\left[-\log_2(1-p)\underbrace{\left[(P_1+P_2)\right]}_{1} + H(P_1, P_2)\right] =$$

$$= H(p, 1-p) + (1-p) H(P_1, P_2)$$

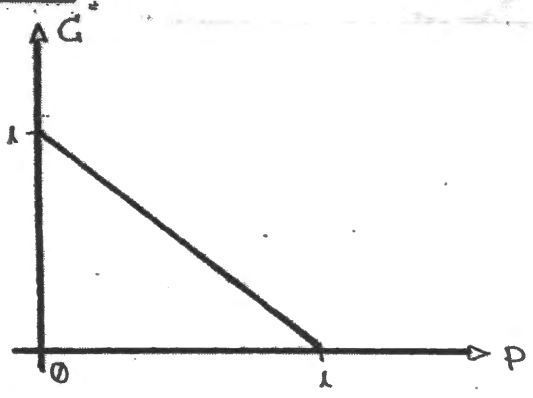
$$\Rightarrow I(X;Y) = \cancel{H(p,1-p)} + (1-p) H(P_1, P_2) - \cancel{H(p,1-p)} = (1-p) H(P_1, P_2)$$

$$\boxed{C = \max I(X;Y) = (1-p)}$$

$\Rightarrow H(P_1, P_2) = H(1/2, 1/2) = 1$

$\overset{\max}{\Rightarrow}$ distrib. uniforme de prob.

$\Rightarrow$ Gráficamente:



- Canal simétrico

* Un canal es simétrico $\Leftrightarrow$
  - $|\mathcal{X}| = |\mathcal{Y}| \Rightarrow$ Matriz cuadrada.
  - Todas las filas y las columnas de P son permutaciones.

Ejemplo:

$$P = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.2 & 0.5 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{bmatrix} \Rightarrow \text{Canal simétrico.}$$

$$I(\mathcal{X};\mathcal{Y}) = H(\mathcal{Y}) - H(\mathcal{Y}/\mathcal{X})$$

$$H(Y/\mathcal{X}) = \sum_x p(x) \sum_y p(y/x) \cdot \log_2 \frac{1}{p(y/x)} = H(\text{fila})$$

$$\Rightarrow C = \max I(\mathcal{X};\mathcal{Y}) = \max\left(H(\mathcal{Y}) - H(\text{fila})\right)$$

- Si el canal es simetrico, entonces:

  Con $p(x) = \frac{1}{|\mathcal{X}|}$ (uniforme) si $\mathcal{X}$ uniforme $\Rightarrow \mathcal{Y}$ uniforme

  $p(y) = p(x) \cdot P \Rightarrow$ distribución de $\mathcal{Y}$ también uniforme.
  $= p(x)$

$$\Rightarrow \boxed{C = \max I(\mathcal{X};\mathcal{Y}) = \log_2 |\mathcal{X}| - H(\text{fila})}$$

$$\overset{|\mathcal{X}| = |\mathcal{Y}|}{=} \log_2 |\mathcal{Y}| - H(\text{fila}).$$

↳ Num. de columnas...

- Canal semisimétrico

* Un canal es semisimetrico $\Leftrightarrow$
  - $|\mathcal{X}| \neq |\mathcal{Y}|$.
  - Las columnas suman lo mismo.

Ejemplo:

$$P = \begin{bmatrix} 1/6 & 1/3 & 1/2 \\ 1/2 & 1/3 & 1/6 \end{bmatrix}$$

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

$$H(Y/X) = \sum_x p(x) \sum_y p(y/x) \cdot \lg_2 \frac{1}{p(y/x)} = H(\text{fila})$$

$$\Rightarrow \boxed{C = \max I(X;Y) = \lg_2 |Y| - H(\text{fila})}$$

$\S$ la distr. es uniforme xo diferente razonamiento igual que el anterior

5.4. Propiedades de la capacidad de canal

- $C \geq 0$ (es el mínimo de $I(X;Y) \geq 0$)
- $C \leq \lg_2 |X|$ ($I(X;Y) \leq H(X) \Rightarrow \max(\ ) \leq \max(\ )$)
- $C \leq \lg_2 |Y|$ ($I(X;Y) \leq H(Y) \Rightarrow \max(\ ) \leq \max(\ )$)

5.5. Teorema de codificación del canal

- Canal con memoria es aquel que cumple:

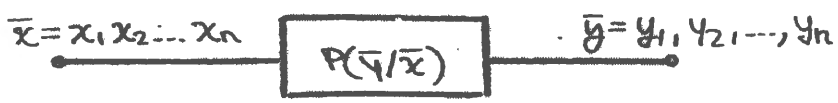
$$\boxed{p(\bar{y}/\bar{x}) = \prod_i p(y_i / x_1, \dots, x_i)}$$

- Canal sin memoria es aquel que cumple:

$$\boxed{p(\bar{y}/\bar{x}) = \prod_i p(y_i/x_i)}$$

- Nosotros vamos a suponer canales sin memoria en el que introducimos una secuencia de entrada y obtenemos otra secuencia a la salida: No tienen realimentación

↳ las entradas no dependen de las salidas



- Si tenemos una secuencia de entrada conocida, la salida puede ser aleatoria. La salida se está generando como una fuente de símbolos independientes:

$$P(\bar{y}/\bar{x}) = P(y_1/x_1)\,P(y_2/x_2)\cdots P(y_n/x_n)$$

- Si es suficientemente larga (n grande), las salidas posibles se pueden clasificar como típicas o atípicas.

⇒ Dado $\bar{x}$ = sec de n términos (n grande)

$$|Y|^n = 2^{n \log_2 |Y|} \equiv \text{nº de secuencias posibles}$$

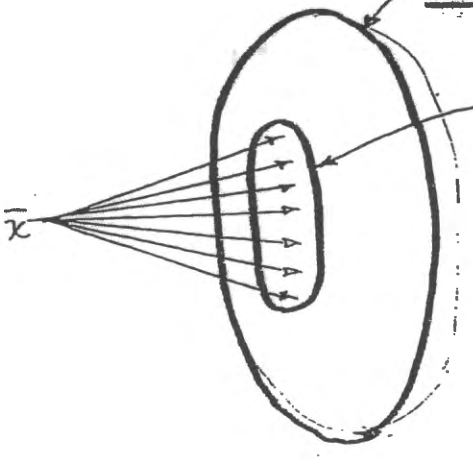
$$|\text{Sec. típicas}| = 2^{n H(Y/X)}$$

- Para n grande, las secuencias típicas sucederán con probabilidad cercana a 1.

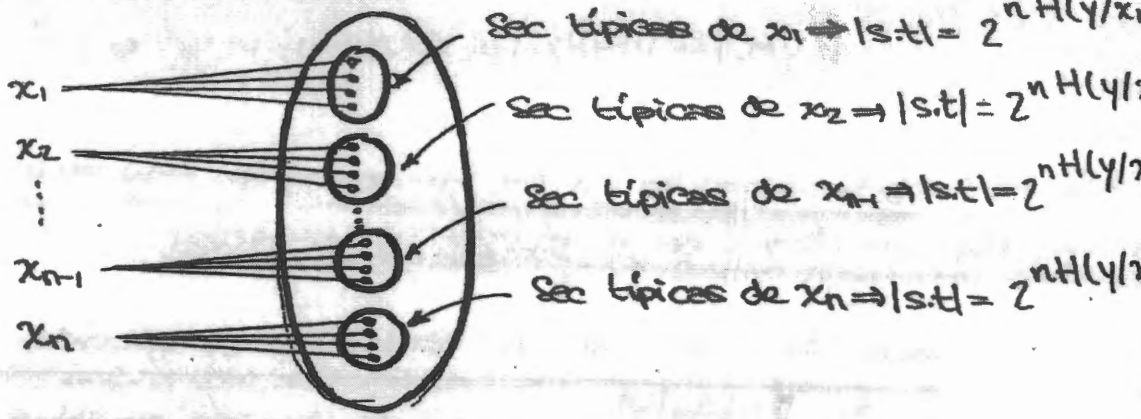
- Se puede suponer que las entradas son aleatorias con distribución de probabilidad $p(x)$ ⇒ las salidas son fruto de dos sucesos aleatorios $\begin{cases} p(x) \\ p(y/x) \end{cases}$

⇒ Tenemos a la salida una cierta distribución $p(y)$ independientes entre sí.

- Gráficamente:



- Si reprimimos la capacidad de expresión de la entrada y generamos menos entradas, de tal manera que se dispersen las salidas en grupos disjuntos (sin solapamiento)



⇒ En este caso, para codificar W símbolos tenemos que encontrar W grupos disjuntos de secuencias conjuntamente típicas, de modo que:

$$W \leq \frac{2^{nH(Y)}}{2^{nH(Y/X)}} = 2^{n(H(Y)-H(Y/X))} = \boxed{2^{n \cdot I(X;Y)}}$$

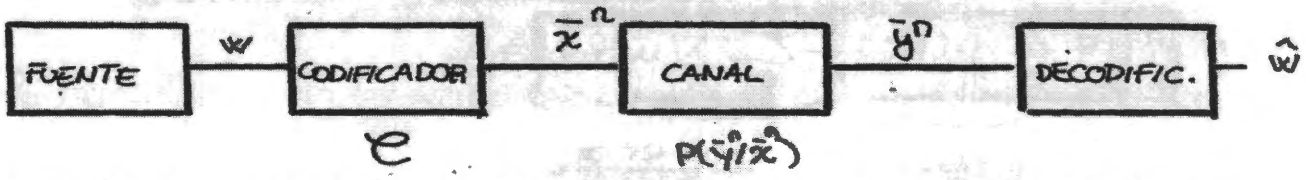
⇒ Sólo podremos enviar $2^{n \cdot I(X;Y)}$ $\frac{\text{símbolos}}{\text{secuencias}}$ de longitud n, para que en la salida puedan ser distinguidos con una probabilidad de error tan pequeña como deseemos.

⇒ podemos enviar $K = n \cdot I(X;Y)$ $\frac{\text{bits}}{\text{secuencia}}$ por $\frac{\text{símbolos}}{\text{secuencia}}$.

* Además con $I(X;Y) \leq \log_2 |X| = 1 \Rightarrow \boxed{K \leq n}$

5.6. Probabilidad de error.

- Suponemos un canal no fiable con el modelo siguiente:



- La fuente genera un mensaje $w \in \{1, 2, \ldots, M\} \Rightarrow K = \log_2 M$
- El codificador de fuente genera una cadena de longitud n para cada mensaje:

$$\mathcal{C} = \begin{bmatrix} x_1(1)\, x_2(1) \ldots x_n(1) & \longrightarrow \text{Codificación del mensaje 1} \\ x_1(2)\, x_2(2) \ldots x_n(2) & \longrightarrow \text{Codificación del mensaje 2} \\ \vdots & \\ x_1(w)\, x_2(w) \ldots x_n(w) & \longrightarrow \text{Codificación del mensaje } w \\ \vdots & \\ x_1(M)\, x_2(M) \ldots x_n(M) & \longrightarrow \text{Codificación del mensaje } M \end{bmatrix}$$

- En el canal se genera una secuencia de salida $\bar{y}^n$ a partir de la secuencia de entrada $\bar{x}^n$.
- El decodificador mostrará un mensaje $\hat{w}$ a partir de $\bar{y}^n$ que puede ser o no el transmitido.
- Definimos el criterio de selección en el decodificador como:

$\hat{w} = w\,(\bar{x}^n(w), \bar{y}^n \in A_\varepsilon^n)$, es decir, $\hat{w}$ será aquel w que haga que la salida del canal y la entrada de ese mensaje sean conjuntamente típicas.

$$\Rightarrow \boxed{P_e^n = \text{Prob}\{\hat{w} \neq w\} \leq \varepsilon + M \cdot 2^{-nI(\bar{X};\bar{Y})}}$$

* Demostración.

• Suponemos que la generación del código es aleatoria

⇒ P_e es distinta para cada código

⇒ Tenemos $\lambda(w,\ell)$ ≡ p. error para cada símbolo y código

⇒ $$P_e = \sum_{\ell} p(\ell) \cdot \frac{1}{M} \sum_{w} \lambda(w,\ell) =$$

prob. código — prob error para cada símbolo y código

$$= \frac{1}{M} \sum_{w} \sum_{\ell} p(\ell) \cdot \lambda(w,\ell)$$

• Si nos fijamos en un mensaje genérico:

ℓ : $x_1(w)\ x_2(w) \dots x_i(w) \dots x_n(w)$

Como el código se genera al azar, nos da igual estudiar todos los mensajes que uno de ellos.

• Suponemos $w=1$ (p. ejemplo)

$$P_e = \text{Prob}\left\{ \hat{w} \neq 1 \ / \ \bar{x}^n = \bar{x}^n(w=1) \right\}$$

con $\bar{x}^n(1) = x_1(1)\ x_2(1) \dots x_n(1)$

• Definimos evento i ≡ $E_i = \left\{ \bar{x}^n(i), \bar{y}^n \in A_\varepsilon^n \right\}$, es decir, que la secuencia de entrada y la de salida sean conjuntamente típicas.

⇒ $$P_e(w=1) = \text{Prob}\left\{ E_1 \cup E_2 \cup \dots \cup E_M \right\} \leq$$

$$\leq \text{Prob}(\bar{E}_1) \cup \text{Prob}(E_2) \cup \dots \cup \text{Prob}(E_M) \leq$$

$$\leq \underbrace{\varepsilon}_{\text{prob. de no ...}} + (M-1)\, 2^{-n(I(x;y) - 3\varepsilon)}$$

$$\Rightarrow \boxed{P_e(w=1)} \leq \varepsilon + (M-1)\,2^{-n(I(X;Y)-3\varepsilon)}$$

$$\leq \varepsilon + M\,2^{-n(I(X;Y)-3\varepsilon)} \leq \boxed{\varepsilon + 2^{-n3\varepsilon}\; M\,2^{-nI(X;Y)}}$$

- Con $\boxed{R = \text{tasa o velocidad binaria} = \frac{\lg_2 M}{n}}$

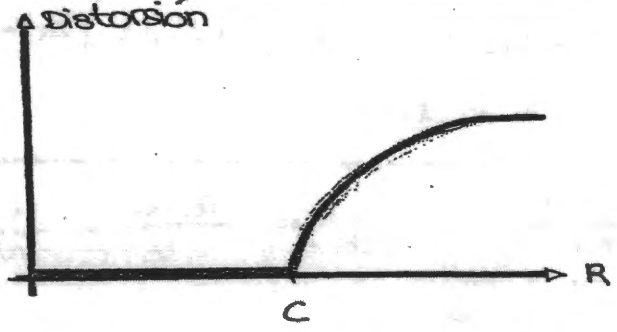
$$\boxed{P_e(w=1)} \leq \varepsilon + 2^{nR}\,2^{-nI(X;Y)} = \boxed{\varepsilon + 2^{-n(I(X;Y)-R)}}$$

Deducimos que si $R < I(X;Y) \Rightarrow$ Puedo hacer tan pequeña como quiera la P_e.

- Para hacer el mejor uso del canal posible, hacemos máxima $I(X;Y) \Rightarrow C = \max I(X;Y)$

$$\Rightarrow \boxed{P_e(w=1) \leq \varepsilon + 2^{-n(C-R)}}$$

De nuevo deducimos que para $R<C \Rightarrow$ hago el error tan pequeño como quiera.



- Si definimos ahora la v.a. E / $\begin{cases} E=1 & \hat{w} \neq w \\ E=0 & \hat{w} = w \end{cases}$

$$H(E, w/\bar{y}^n) = H(w/\bar{y}^n) + H(E/w, \bar{y}^n)^{\nearrow 0} = H(E/\bar{y}^n) + H(w/E, \bar{y}^n)$$

$$H(E/\bar{y}^n) \leq 1$$

$$H(w/E, \bar{y}^n) = P(E=0)\cdot H(w/E=0, y^n)^{\nearrow 0} + \underbrace{P(E=1)}_{P_e}\,\underbrace{H(w/E=1, \bar{y}^n)}_{H_{max}=\lg_2 M} \leq$$

$$\leq P_e \lg_2 M$$

$$\Rightarrow H(w/\bar{y}^n) \leq 1 + P_e \lg_2 M = 1 + P_e nR$$

$$\overset{\circ}{0} \left\{ w = 2^{nR} \;\; 2^{nI(X;Y)} \right.$$

* Por el teorema de procesado de la información.

$$I(w;\bar{y}^n) \leq I(\bar{x}^n;\bar{y}^n)$$

$$\Rightarrow I(\bar{x}^n;\bar{y}^n) \leq n\,I(\bar{x}^n;\bar{y}^n) \leq nC$$

* Por otra parte:

$$H(w)_{max} = nR \quad \circ O$$

M símbolos equipro. $\equiv Z^{nR}$

$\Rightarrow H_{max}(w) = \log_2 |M| = nR$

* Nos queda finalmente:

$$H(w/\bar{y}^n) \leq 1 + Pe\,nR$$

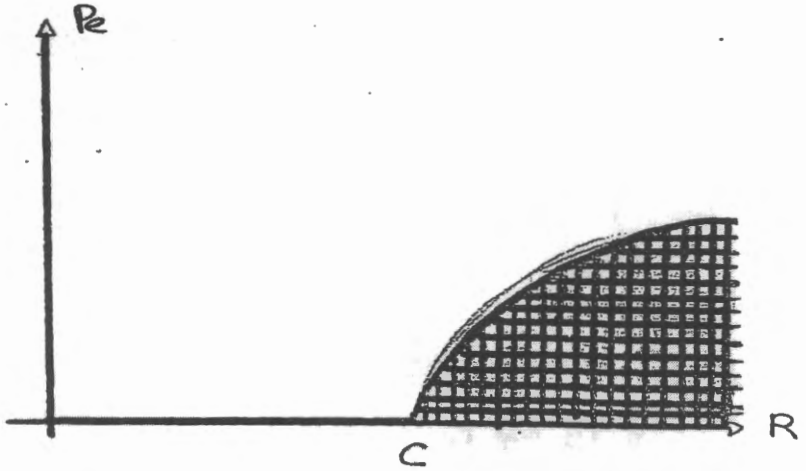
$$H(w/\bar{y}^n) \geq nR - nC = n(R-C)$$

$$\Rightarrow \quad n(R-C) \leq H(w/\bar{y}^n) \leq 1 + Pe\,nR$$

$$\Rightarrow \quad n(R-C) \leq 1 + Pe\,nR$$

$$\Rightarrow \boxed{Pe \geq \frac{n(R-C)}{nR} - \frac{1}{nR} = \frac{R-C}{R} - \frac{1}{nR}}$$

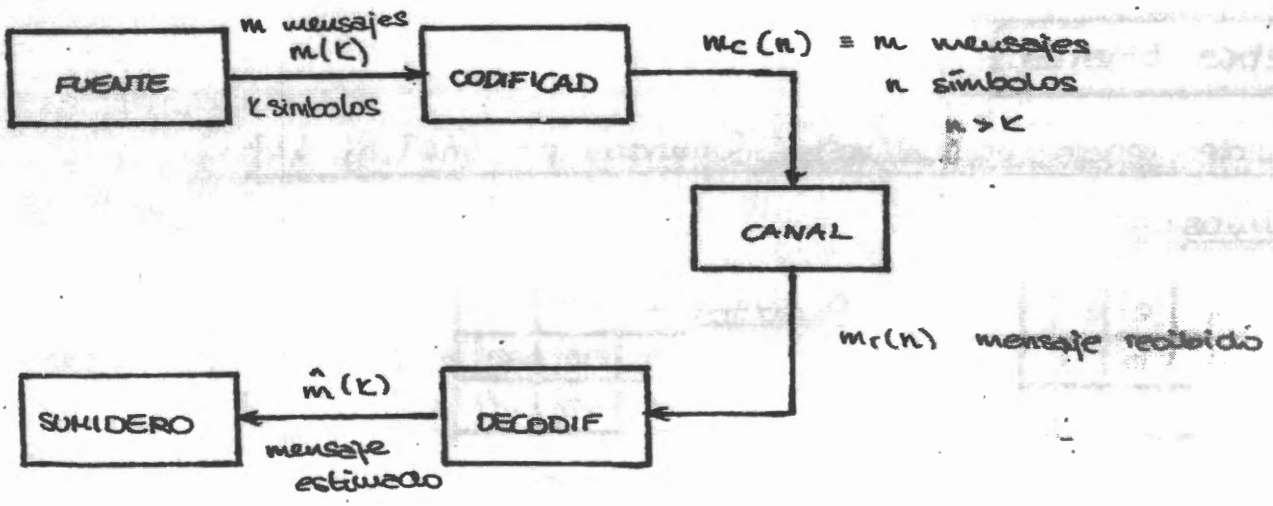
Si $R > C \Rightarrow Pe > 0$



Tema 6. Códigos lineales

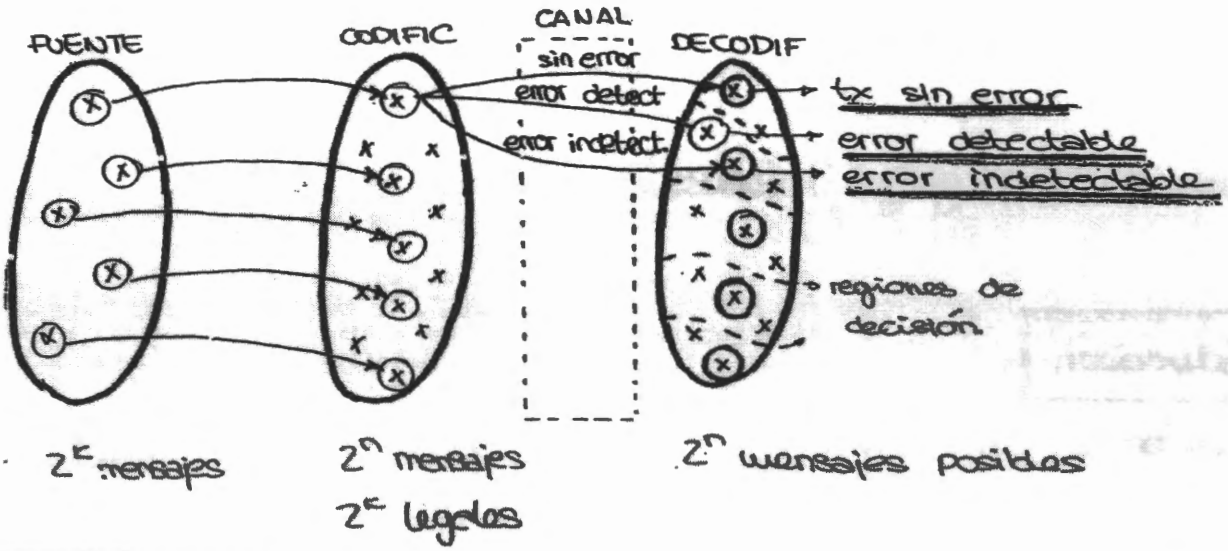
6-1. Introducción

- Partimos de:



- El teorema de codificación de canal nos dice que siempre que $R = \frac{K}{n} < C$ se puede tener un error que tiende asintóticamente a cero.

- La fuente genera un mensaje formado por K bits $\equiv 2^K$ mensajes
- Para cada mensaje fuente elegimos un mensaje código $\Rightarrow$ Sólo tenemos un subconjunto de mensajes código "legales".
- Estos mensajes código los mandamos por el canal. El canal introducirá ruido y generaremos errores.
- Para facilitar la tarea de decisión definiremos zonas de inferencia para cada palabra código.



2- Algebra binaria.

- La fuente genera un alfabeto formado por {0} y {1}

- Definimos:

Suma:

+	0	1
0	0	1
1	1	0

(XOR)

Producto:

·	0	1
0	0	0
1	0	1

(AND)

→ Galois cuerpo de Galois

- Sea G un conjunto : {0,1}

G es grupo ⇔
a) Hay definida una operación (*) binaria
b) * es asociativa
c) $\exists\, e \in G / \forall\, a \in G ;\ a*e = e*a = a$ (el. neutro)
d) $\forall\, a \in G ; \exists\, a' \in G / a*a' = e$ (el. inverso)

⇒ Nuestro alfabeto junto con "suma" es grupo

G es grupo conmutativo ⇔ es grupo y $\forall\, a,b \in G$:

$$a*b = b*a$$

⇒ Nuestro alfabeto junto con "suma" es grupo conmutativo

Sea G un conjunto de elementos con dos operaciones: +, ·

G es cuerpo ⇔
- *(G,+) es grupo conmutativo
- *El conjunto de elementos ≠ ∅ de G junto con (·) es grupo conmutativo.
- *(G,+,·) es distributivo ⇔ $a\cdot(b+c) = a\cdot b + a\cdot c$

G es cuerpo de Galois si la dimensión del alfabeto es finita

⇒ {0,1} junto con (+),(·) forman un cuerpo de Galois: G(2)

3- Propiedades

1- El conjunto V_n formado por todas las n-tuplas:

$$\bar{v} = (v_1\ v_2 \dots v_n),\ v \in G(2)$$

junto con las operaciones $(+): \bar{v} + \bar{u} = \bar{w} / w_i = v_i + u_i$

$(\cdot): \lambda\cdot\bar{u} = \bar{w} / w_i = \lambda\cdot u_i$ $(\lambda \in G(2))$

⇒ tiene estructura de espacio vectorial sobre G(2)

Nº elementos: 2^n

Dimensión: n

2- Sea $S \subseteq V_n$, "S es subespacio vectorial ⇔

* $\bar{0} \in S$
* $\forall\ \bar{u}, \bar{v} \in S \Rightarrow (\bar{u}+\bar{v}) \in S$
* $(\lambda\cdot\bar{u}) \in S$, $(\lambda \in G(2))$

4- Dado un conjunto de vectores $\bar{v}_1, \bar{v}_2, \dots \bar{v}_k$ se dice que son linealmente independientes (li) $\Leftrightarrow$

$$\forall c_i \Rightarrow c_1\bar{v}_1 + c_2\bar{v}_2 + \dots + c_k\bar{v}_k = \emptyset \Leftrightarrow c_i = \emptyset$$

4- El conjunto formado por los vectores

$\{\bar{g}_1, \bar{g}_2, \dots \bar{g}_n\}$ generan el espacio V_n si:

$$\forall\ \bar{v} \in V_n \Rightarrow \bar{v} = \sum_{i=1}^{n} c_i\bar{g}_i$$

5- En todo V_n, $\exists\ \{\bar{g}_1, \bar{g}_2, \dots, \bar{g}_n\} \subseteq V_n$ l.i. que generan el espacio vectorial V_n y con $\text{Card}(\bar{g}_n) = \dim(V_n)$

6- Si $k<n$ y $\{\bar{v}_1, \dots, \bar{v}_k\} \subseteq V_n$ y son li $\Rightarrow$ $S = \{\bar{u} / \bar{u} = \sum_{i=1}^{k} c_i\bar{v}_i\}$ $\subseteq V_n$, es un subespacio vectorial de dimensión k.

7- Se define la operación producto interior (escalar):

$$\bar{u}\cdot\bar{v} = \sum_{i=1}^{n} u_i v_i = u_1v_1 + u_2v_2 + \dots + u_nv_n$$

8- Dos vectores son ortogonales $\Leftrightarrow$ $\bar{u}\cdot\bar{v} = \emptyset$

9- Sea S un subespacio vectorial de dimensión $k \subseteq V_n$.
Sea S_d el conjunto de vectores $\subseteq V_n$ /

$$\forall\ \bar{u} \in S\ /\ \forall\ \bar{v} \in S_d \Rightarrow \bar{u}\cdot\bar{v} = \emptyset$$

$\Rightarrow$ S_d = Espacio dual de S → (o los vectores ortogonales a S)

$$\dim(S_d) = n - k$$

4- Códigos lineales $n>k$

- Un código se define como una transformación entre una palabra de k bits y otra de n bits.

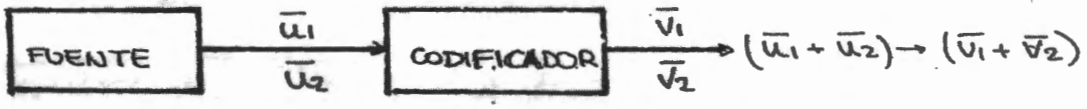
$\Leftrightarrow$
* Subconjunto de palabras de n bits.
* Asignación de: k bits → n bits.

- Clasificación

Códigos de CANAL
- Lineal
  - De bloque
    - cíclico
    - sistemáticos
    - no sistemáticos
  - Convolucionales
- No lineal

* Lineal = se pueden formalizar matemáticamente.
* Bloque = una palabra código sólo depende de su palabra fuente (los n bits transmitidos sólo dependen de los k bits emitidos)

- Para nosotros un código lineal será el que genere un subespacio vectorial.



- Definición: Un código de longitud n y 2^k palabras de código es lineal $C(n,k) \Leftrightarrow$ sus 2^k palabras código forman un subespacio vectorial de dimensión $k \leq V_n$ sobre $G(2)$.

** Ejemplo

$C(7,4)$ $\left\{\begin{matrix} n=7 \\ k=4 \end{matrix}\right\} \Rightarrow 2^k = 16$ palabras.

Código sistemático: la palabra de entrada está incluida en la palabra código

Mensajes	Palabras código		
0000	→	0000000	
0001	→	1010001	
0010	→	1110010	
0011	→	0100011	
0100	→	0110100	
0101	→	1100101	
0110	→	1000110	
0111	→	0010111	
1000	→	1101000	
1001	→	0111001	
1010	→	0011010	
1011	→	1001011	
1100	→	1011100	
1101	→	0001101	
1110	→	0101110	
1111	→	1111111	

6.5- Matriz generadora. Estructuración sistemática. Codificador

- Sea $C(n,k)$ ⇒ subespacio vectorial de dimensión k

⇒ es posible encontrar k palabras código linealmente independientes $\{\bar{g}_0, \bar{g}_1, \bar{g}_2, \dots, \bar{g}_{k-1}\}$ / $\bar{g}_i \in C(n,k)$ de forma que:

$$\forall\ \bar{v} \in C\ /\ \bar{v} = u_0\bar{g}_0 + u_1\bar{g}_1 + \dots + u_{k-1}\bar{g}_{k-1}$$

⇒ $\bar{v} = \bar{u} \cdot G$; $G \equiv$ matriz generadora.

$$G = \begin{bmatrix} g_{0,0} & g_{0,1} & \dots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \dots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \dots & g_{k-1,n-1} \end{bmatrix}_{k \times n} = \begin{pmatrix} \bar{g}_0 \\ \bar{g}_1 \\ \vdots \\ \bar{g}_{k-1} \end{pmatrix}$$

con $\begin{cases} \bar{v} = (v_1, \dots, v_n) \\ \bar{u} = (u_1, \dots, u_k) \end{cases}$

** Ejemplo:

Del código anteriormente generado, se puede demostrar que:

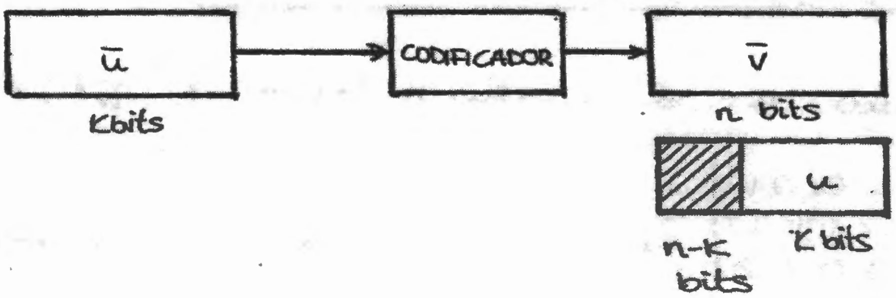
$(1101000, 0110100, 1110010, 1010001)$ forman base del subespacio vectorial generado.

$\bar{g}_0 \quad \bar{g}_1 \quad \bar{g}_2 \quad \bar{g}_3$

$$\Rightarrow G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \left[\begin{array}{ccc|cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

Si cambio en G las bases tengo otra asignación pero tengo el mismo código

- Los códigos que nosotros buscamos son los que contienen a la palabra mensaje ya que su reconocimiento y posterior decodificación será más fácil



- Los bits añadidos se denominan redundancia o bits de protección
- Por tanto para generar el código:

⇒ Debo escoger la matriz G determinada.

- Se puede demostrar que $\exists$ G de la forma $G = (P \mid I_k)$ que genera el código deseado ≡ matriz sistemática.

$$G = \left[\begin{array}{ccc|ccc} P_{0,0} & P_{0,1} \dots & P_{0,n-k-1} & 1 & 0 \dots & 0 \\ P_{1,0} & P_{1,1} \dots & P_{1,n-k-1} & 0 & 1 \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ P_{k-1,0} & P_{k-1,1} \dots & P_{k-1,n-k-1} & 0 & 0 \dots & 1 \end{array}\right] \quad k \times n$$

$P_{k \times n-k}$; I_k → B. de información

v_0 v_1 ... v_{n-k-1} v_{n-k} ... v_{n-1}

B. de redundancia

$$\Rightarrow \bar{v} = \bar{u} G \Rightarrow (v_0\ v_1 \dots v_{n-1}) = (u_0\ u_1 \dots u_{k-1})\, G$$

$$\Rightarrow \begin{cases} v_j = u_0 P_{0j} + u_1 P_{1j} + \dots + u_{k-1} P_{k-1,j} \;,\; 0 \le j < n-k \\ v_{n-k+i} = u_i \;,\; 0 \le i < k \end{cases}$$

≡ Ecuaciones de comprobación de paridad.

** Ejemplo: Seguimos con el ejemplo anterior $G(7,4)$

$$G = \left[\begin{array}{ccc|cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \Rightarrow (v_0\ v_1\ v_2\ v_3\ v_4\ v_5\ v_6) = (u_0\ u_1\ u_2\ u_3)\, G$$

$$\Rightarrow \begin{cases} v_0 = u_0 + u_2 + u_3 \\ v_1 = u_0 + u_1 + u_2 \\ v_2 = u_1 + u_2 + u_3 \\ v_i = u_{i-3} \quad \forall\, i \in \{3,4,5,6\} \end{cases}$$

⇒ Para encontrar una estructura sistemática hay que diagonalizar la submatriz de la derecha.

** Ejemplo: Código de bit de paridad simple

⇒ Se añade 1 bit $\begin{cases} 1 \Rightarrow \text{Nº impar de 1's} \\ 0 \Rightarrow \text{Nº par de 0's} \end{cases}$

Con un código C(6,5):

$$\bar{u} = (u_0\ u_1\ u_2\ u_3\ u_4)$$

$$\bar{v} = (v_0\ v_1\ v_2\ v_3\ v_4\ v_5)$$

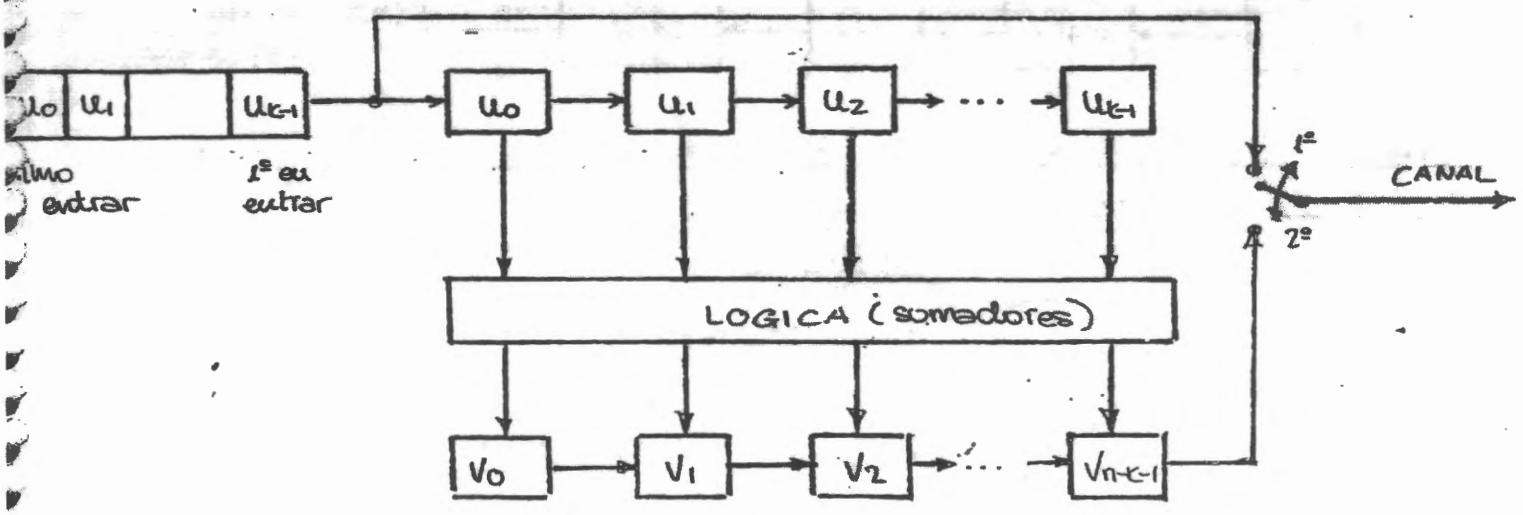
" " u_0 " u_1 " u_2 " u_3 " u_4

$$\Rightarrow \begin{cases} v_0 = u_0 + u_1 + u_2 + u_3 + u_4 \\ v_1 = u_0 \\ v_2 = u_1 \\ \vdots \\ v_5 = u_4 \end{cases}$$

⇒ La matriz sistemática será:

$$G = \left[\begin{array}{c|ccccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

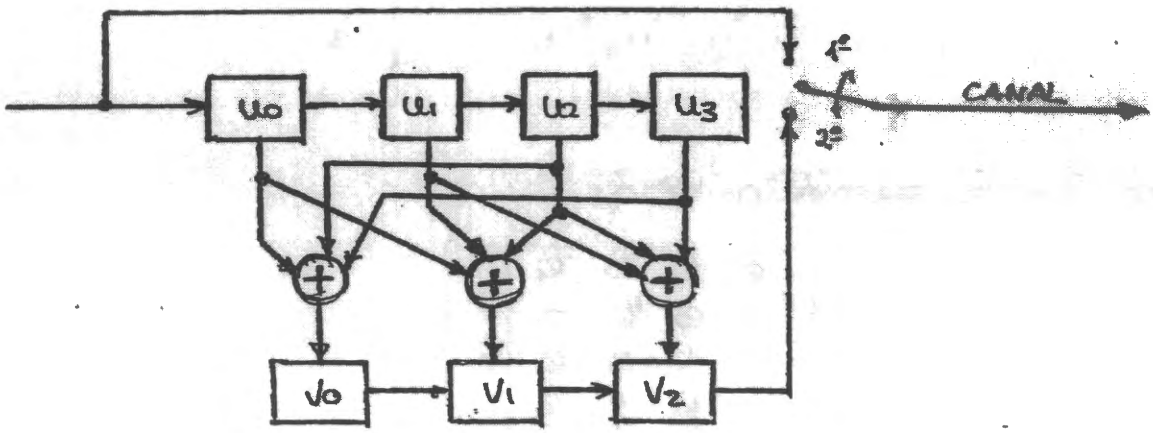
El diseño del codificador para este código se basa en un registro de desplazamiento que recibe el mensaje $\bar{u}$ y que además generará la redundancia:



** Ejemplo: $C(7,4)$

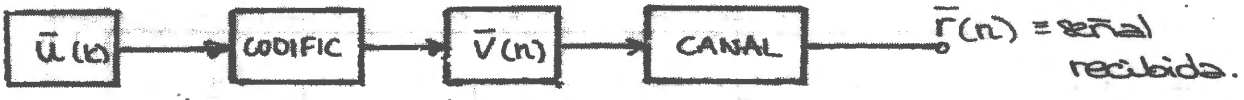
$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \rightarrow u_0 \\ \rightarrow u_1 \\ \rightarrow u_2 \\ \rightarrow u_3 \end{matrix} \Rightarrow \begin{cases} V_0 = u_0 + u_2 + u_3 \\ V_1 = u_0 + u_1 + u_2 \\ V_2 = u_1 + u_2 + u_3 \end{cases}$$

$v_0\ v_1\ v_2\ v_3\ v_4\ v_5\ v_6$



6- Matriz H. Introducción al control de errores.

- Sea el siguiente sistema de comunicaciones digital:



- Cuando a la salida del canal tenemos una señal $\bar{r}(n)$ hay que ver si se han producido errores:

¿$\bar{r}(n) \in C(n,k)$?
- Sí → No hay error
- Sí → Error ≡ indetectable
- No → Hay error. (detectable)

- Se define H, matriz de comprobación de paridad, que se utiliza para ver si una palabra es una palabra código.

- T^a: $\forall\ G_{k\times n}$ con k filas l.i., $\exists\ H_{n-k\times n}$ con $n-k$ filas l.i. que cumple:

  * $\forall$ vector generado por G es ortogonal a las filas de H
  * $\forall$ vector ortogonal a las filas de H es generado por las filas de G.

$$\Rightarrow \boxed{\bar{v} \in C(n,k) \Leftrightarrow \bar{v}\cdot H^t = \emptyset}$$

- Cálculo de la matriz H: Partimos de G sistemática

$$\boxed{G = (P_{n-k} \mid I_k)_{k\times n} \overset{\text{def}}{\Rightarrow} H = (I_{n-k} \mid P^t)_{n-k\times n}}$$

$$G = \underbrace{\begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,n-k-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}}_{k\times n-k} \; \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{k\times k}_{k\times n}$$

$$\Downarrow \text{def}$$

$$H = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}}_{n-k\times n-k} \; \underbrace{\begin{bmatrix} p_{0,0} & p_{1,0} & \cdots & p_{k-1,0} \\ p_{0,1} & p_{1,1} & \cdots & p_{k-1,1} \\ \vdots & & & \\ p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}}_{n-k\times k}_{n-k\times n}$$

$$\Rightarrow \forall\ \bar{g}_i \in G\ ;\ \forall\ \bar{h}_i \in H \Rightarrow \bar{g}_i \cdot \bar{h}_j = p_{ij} + p'_{ij} = \emptyset$$

* Ejemplo: $C(7,4)$

$$G = \left[\begin{array}{ccc|cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right]_{4\times 7} \Rightarrow H = \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right]_{3\times 7}$$

De forma que por ejemplo: $(1001011) \in C$

$$(1001011)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}_{7\times 3} = (0,0,0) \Rightarrow \bar{v}\cdot H^t = \bar{0}$$

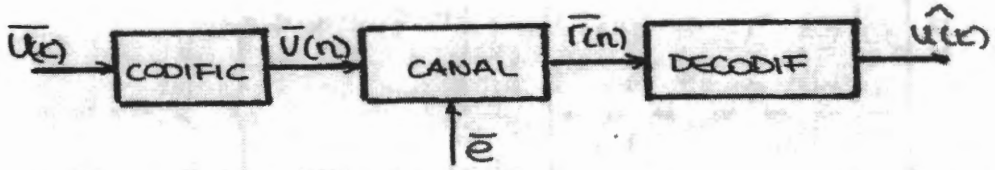
- Dada $H_{n-k\times n}$ $\exists$ $n-k$ filas l.i $\Rightarrow$ generan otro código = código dual de C

* $\boxed{C \text{ dual} \equiv C_d(n-k, n)}$

$$\Rightarrow \left\{\begin{array}{l} \forall \bar{v} \in C(n,k) \\ \forall \bar{w} \in C_d(n-k, n) \end{array}\right\} \Rightarrow \boxed{\bar{v}\cdot\bar{w} = \bar{0}}$$

$\Rightarrow$ H es la generadora del código dual de C

7. Síndrome



- Dado un sistema de comunicaciones como el del dibujo, se pretende determinar la señal recibida $\hat{u}(k)$, distorsionada por la acción del canal.
- Para modelar el canal suponemos que a la señal código $\bar{v}(n)$ se le añade otra señal $\bar{e}(n) \equiv$ vector error.

$$\bar{e} = (e_0\ e_1 \dots e_{n-1}) = \begin{cases} 1 \to \text{el ruido afecta a la posición} \\ 0 \to \text{no afecta a la posición} \end{cases}$$

- Definimos el vector síndrome $\boxed{\bar{s} = \bar{r} \cdot H^t}$. Puede suceder que

$$\bar{s} \begin{cases} \bar{s} \neq 0 \Rightarrow \bar{r} \notin G(n,k) \supset \text{error (detectable)} \\ \bar{s} = 0 \Rightarrow \bar{r} \in G(n,k) \begin{cases} \text{no hay error} \\ \text{Error indetectable} \end{cases} \end{cases}$$

$$\Rightarrow \bar{s} = \bar{r} \cdot H^t = (\bar{v} + \bar{e}) \cdot H^t = \underbrace{\bar{v} \cdot H^t}_{\in G(n,k)} + \bar{e} \cdot H^t = \bar{e} \cdot H^t$$

$$\text{si } \bar{s} = 0 \Rightarrow \bar{e} \cdot H^t = 0 \begin{cases} \bar{e} = 0 \Rightarrow \text{no hay error} \\ \bar{e} \neq 0 \Rightarrow \bar{e} \in G(n,k) - \{\bar{0}\} \equiv \boxed{2^k - 1 \text{ posi}} \end{cases}$$

$(\Rightarrow$ Error indetectable)

- Nuestro objetivo ahora será también reducir la probabilidad de que se produzcan este tipo de errores.

- Diseño del generador del síndrome

$$\bar{s} = \bar{r} \cdot H^t \quad ; \quad H^t = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \\ \hline P_{00} & P_{01} & \cdots & P_{0,n-k-1} \\ \vdots & \vdots & & \vdots \\ P_{k-1,0} & P_{k-1,1} & \cdots & P_{k-1,n-k-1} \end{bmatrix} \begin{matrix} \} \; n-k \times n-k \\ \\ \\ \\ \} \; k \times n-k \end{matrix} \Big\} \; n \times n-k$$

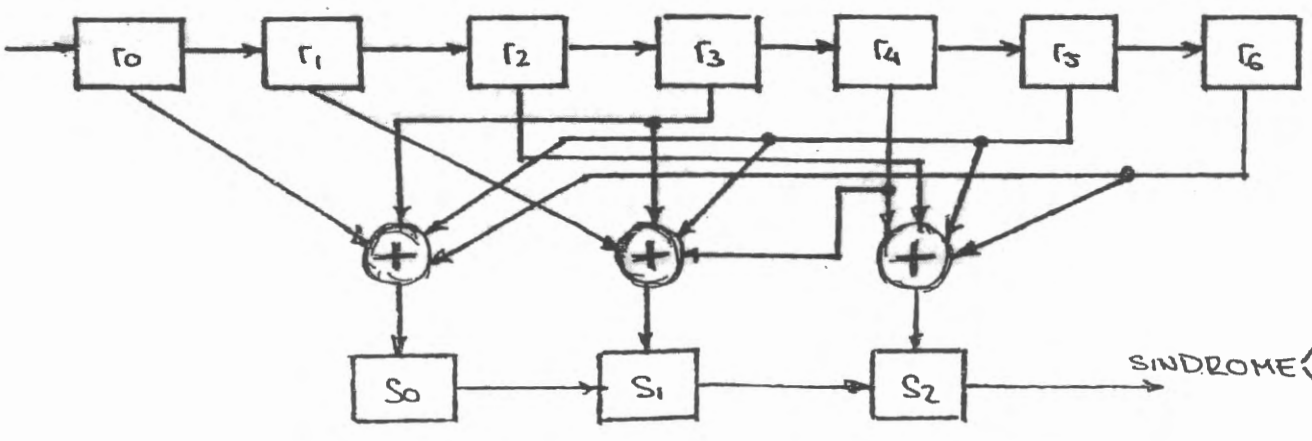
$$\Rightarrow \begin{cases} s_0 = r_0 + r_{n-k} P_{00} + r_{n-k+1} P_{1,0} + \cdots + r_{n-1} P_{k-1,0} \\ s_1 = r_1 + r_{n-k} P_{01} + r_{n-k+1} P_{1,1} + \cdots + r_{n-1} P_{k-1,1} \\ \vdots \\ s_{n-k-1} = r_{n-k-1} + r_{n-k} P_{0,n-k-1} + \cdots + r_{n-1} P_{k-1,n-k-1} \end{cases}$$

⇒ El circuito es muy similar al del codificador.

** Ejemplo: $C(7,4)$

$$H = \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right] \quad ; \quad \bar{s} = \bar{r} \cdot H^t$$

$$\bar{s} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{cases} s_0 = r_0 + r_3 + r_5 + r_6 \\ s_1 = r_1 + r_3 + r_4 + r_5 \\ s_2 = r_2 + r_4 + r_5 + r_6 \end{cases}$$



¿ Se pueden corregir errores?

$$\Rightarrow \boxed{\bar{S} = \bar{r}\cdot H^t = (\bar{v}+\bar{e})\cdot H^t = \bar{e}\cdot H^t}$$

De forma que:

$$\begin{cases} s_0 = e_0 + e_{n-k}p_{00} + \dots + e_{n-1}p_{k-1,0} \\ s_1 = e_1 + e_{n-k}p_{01} + \dots + e_{n-1}p_{k-1,1} \\ \vdots \\ s_{n-k-1} = e_{n-k-1} + e_{n-k}p_{0,n-k-1} + \dots + e_{n-1}p_{k-1,n-k-1} \end{cases}$$

De estas ecuaciones no puedo determinar el vector error ya que hay más incognitas que ecuaciones $\Rightarrow \exists\ 2^k$ posibles soluciones.

- Por tanto, corregir será escoger la más probable.

* Ejemplo $C_i(7,4)$

$$\bar{v} = (1001011) \longrightarrow \boxed{\text{CANAL}} \longrightarrow \bar{r} = (1001001)$$

$$\bar{S} = \bar{r}\cdot H^t = (1001001)\begin{bmatrix} 100 \\ 010 \\ 001 \\ 110 \\ 011 \\ 111 \\ 101 \end{bmatrix} = (1,1,1) \neq \bar{0} \Rightarrow \text{Hay error}$$

$$\Rightarrow \begin{cases} s_0 = e_0 + e_3 + e_5 + e_6 = 1 \\ s_1 = e_1 + e_3 + e_4 + e_5 = 1 \\ s_2 = e_2 + e_4 + e_5 + e_6 = 1 \end{cases} \Rightarrow$$

Generando los 2^k posibles vectores de error elegiremos el que menos cambios realice:

$$(0000\textcircled{1}0) \equiv \hat{e}$$

(e5)

* menor nº de 1's

8. Distancia de Hamming

- Sea $\bar{v} = (v_0\ v_1 \dots v_{n-1})$

- Se define **peso de Hamming** de $\bar{v} \equiv \omega(\bar{v}) \equiv$ nº de componentes distintas de 0 en $\bar{v}$ ≡ nº de 1's de $\bar{v}$

- Se define la **distancia de Hamming** entre $\bar{u}, \bar{v} \equiv d(\bar{u}, \bar{v})$ como el nº de dígitos en los que difieren $\bar{u}$ y $\bar{v}$

  * Medida de cercanía/lejanía entre vectores
  * Cumple la desigualdad triangular

- Al estar definida la operación suma como:

$$a + b \begin{cases} = 1 & a \neq b \\ = 0 & a = b \end{cases}$$

Se cumple que $\boxed{d(\bar{v}, \bar{w}) = \omega(\bar{v} + \bar{w})}$

- Dado un código $C(n,k)$ se define la distancia mínima del código como:

$$\boxed{d_{min} = \min \{ d(\bar{v}, \bar{w});\ \bar{v}, \bar{w} \in C;\ \bar{v} \neq \bar{w} \}}$$

* Como $d(\bar{v}, \bar{w}) = \omega(\bar{v} + \bar{w}) = \omega(\bar{x})$, $\bar{x} \in C$

$$\Rightarrow \boxed{d_{min} = \min \{ \omega(\bar{x}),\ \bar{x} \in C;\ \bar{x} \neq 0 \}}$$

- **T^a**: Dado un código $C(n,k)$ con H, $\forall \bar{v} \in C / \omega(\bar{v}) = \ell$

$\Rightarrow \exists\ \ell$ columnas de H que sumen $\emptyset$ //

DEMO: $H = (\bar{h}_0, \bar{h}_1, \ldots, \bar{h}_{n-1})$

$\bar{v} \in C$, $\bar{v} = (v_0, v_1, \ldots, v_{n-1})$

$$\bar{v} \cdot H^t = v_0 \bar{h}_0 + v_1 \bar{h}_1 + \ldots + v_{n-1} \bar{h}_{n-1} = \emptyset$$

- **T^a**: Si $\exists\ \ell$ columnas de H que sumen $\emptyset \Rightarrow \exists \bar{v} \in C / \omega(\bar{v}) = \ell$

DEMO: $H = (\bar{h}_0, \bar{h}_1, \ldots, \bar{h}_{n-1}) \Rightarrow \sum_{i=0}^{\ell} \bar{h}_i = \emptyset$

Formo $\bar{x}$ / $\text{comp}_i(\bar{x}) \neq \emptyset$, $\emptyset < i < \ell$

$\Rightarrow \bar{x} \cdot H^t = \emptyset \Rightarrow \bar{x} \in C \Rightarrow \omega(\bar{x}) = \ell$

- **Corolario:** Si $C(n,k)$ con H matriz de paridad, y si $\nexists$ $(d-1)$ o menos columnas de H que sumen $\emptyset \Rightarrow d_{min} \geq d$

- **Corolario:** $d_{min}(C)$ = menor número de columnas de H que sumen $\emptyset$.

** Ejemplo: $C(7,4)$

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

$\Rightarrow \nexists$ columna nula $\Rightarrow d_{min} \neq 1$

$\Rightarrow \nexists\ \bar{h}_1, \bar{h}_2$ / $\bar{h}_1 + \bar{h}_2 = \emptyset \Rightarrow d_{min} \neq 2$

La suma de columnas iguales es diferente de $\emptyset$

$\exists\ \bar{h}_1, \bar{h}_2, \bar{h}_3$ / $\bar{h}_1 + \bar{h}_2 + \bar{h}_3 = \bar{\emptyset} \Rightarrow d_{min} = 3$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

h_0 h_1 h_3

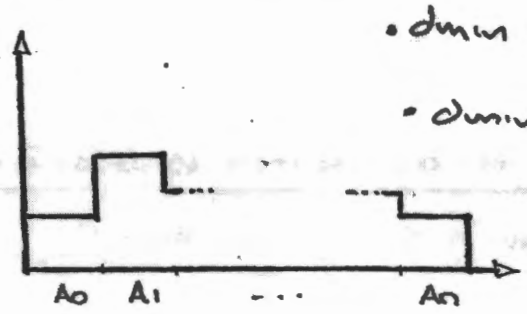
6.9. Capacidad de detección.

$\forall \bar{e} / w(\bar{e}) < d_{min} \rightarrow$ Detecto el error

- Sea $C(n,k)$ y sea A_i el número de palabras código con peso i:

$$A_i = \text{card}\{\bar{v}_k / w(\bar{v}_k) = i,\ \forall \bar{v}_k \in C(n,k)\}$$

- Se define el conjunto $\{A_0, A_1, \ldots, A_n\}$ ≡ distribución de pesos del código.



- d_{min} impar ⇒ La mitad peso par la mitad peso impar
- d_{min} par ⇒ Todos peso par.

** En nuestro ejemplo:

$A_0 = 1$ (solo existe una vez el 0); $A_1 = 0$; $A_2 = 0$; $A_3 = 7$; $A_4 = 7$; $A_5 = 0$; $A_6 = 0$; $A_7 = 1$

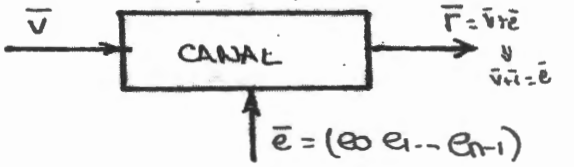
Si tenemos la palabra 111...11 (todos 1s) en el código, la distribución de pesos es simétrica.

- La probabilidad de no detectar un error ≡ P_{ND}

$$P_{ND} = P(\bar{S} = \bar{0} \wedge \bar{e} \in C') \equiv P(\bar{e} \in C')$$

No detecto error si el error pertenece al código

$d(\bar{v},\bar{r})$ = nº de bits diferentes = $w(\bar{v} \oplus \bar{r}) = w(\bar{e})$



Si p ≡ probabilidad de error de bit del canal y $\bar{e} = (1101000)$ (p-e), entonces

$$P(\bar{e}) = p^3 (1-p)^4$$

* Deducimos que: $\bar{e} / w(\bar{e}) = i \Rightarrow$

$$P(\bar{e}) = p^{w(\bar{e})} (1-p)^{n-w(\bar{e})}$$

$$\Rightarrow P_{ND} = A_1 p (1-p)^{n-1} + A_2 p^2 (1-p)^{n-2} + \ldots + A_{n-1} p^{n-1} (1-p) + A_n p^n$$

$$= \sum_{i=1}^{n} A_i p^i (1-p)^{n-i}$$

A_0 no lo incluimos porque con $\bar{e} = \bar{0}$ no hay error

$\sum$ nº palabras · $p^{peso} (1-p)^{n-peso}$

- Muchas veces A_i no son conocidos por lo que podemos aproximar la expresión por una cota superior.

$$A_i \leq \binom{n}{i} \Rightarrow P_{ND} \leq \sum_{i=1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

- Otra forma de aproximación:

$$P_{ND} = \sum_{i=1}^{n} A_i \, p^i (1-p)^{n-i} \quad , \text{ con } C(n,k) \text{ „ } d_{min}$$

$\binom{n}{s+1} p^{s+1} (1-p)^{n-(s+1)}$

Si $i < d_{min} \Rightarrow A_i = \emptyset$, por lo que:

prob error residual (P_{ND}, ARQ) $\sim \binom{n}{d_{min}} p^{d_{min}} (1-p)^{n-d_{min}}$

$$P_{ND} = \sum_{i=d_{min}}^{n} A_i \, p^i (1-p)^{n-i} \leq \sum_{i=d_{min}}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

$\emptyset$ (es la buena) — porque si es lineal $\exists$ vector $\vec{\emptyset}$ y no puede haber vectores más cercanos de d_{min}

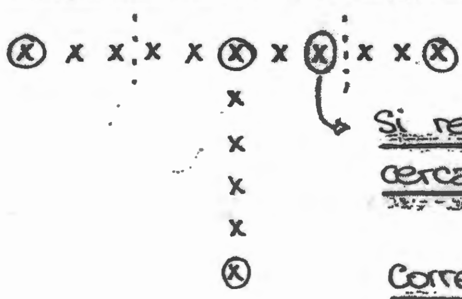
- Por tanto, si $w(\bar{e}) < d_{min} \Rightarrow P_{ND} = \emptyset \equiv$ Un código $C(n,k)$ detecta todos los errores / $w(\bar{e}) < d_{min}$

Se define la capacidad de detección de un código $C(n,k)$ como:

$$s_{max} = d_{min} - 1$$

Garantizamos que todos los errores tal que sus pesos $w(\bar{e}) < d_{min}$ los detectemos.

6.10. Capacidad de corrección



Si recibo esta palabra, corrijo a la más cercana $\Rightarrow$ regiones de decisión.

Corregir = tomar una decisión

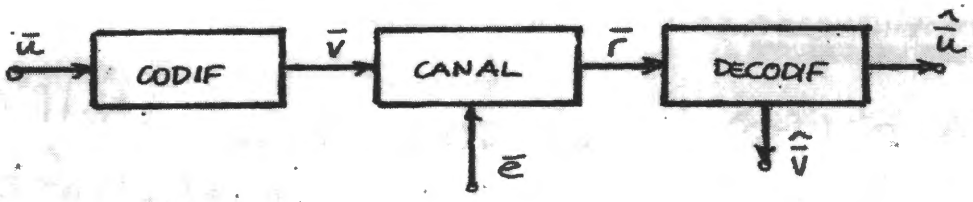
$\Rightarrow$ recibo $\bar{r} \rightarrow \hat{v}$

- Algoritmo de decisión (DECODIF. DE MÁXIMA SIMILITUD)

* Si $\bar{e} = (e_0\ e_1 \dots e_{n-1})$

$$p(\bar{e}) = p^{w(\bar{e})}(1-p)^{n-w(\bar{e})} = (1-p)^n \left(\frac{p}{1-p}\right)^{w(\bar{e})} = K\alpha^{w(\bar{e})}$$

α: exp. decreciente



* MLD = escoger $\hat{v}$ que minimice $p(E/\bar{r}) = p(\hat{v} \neq \bar{v}/\bar{r}) =$

$= \dots \Rightarrow$ Escoger $\hat{v} = \bar{v}$ / se maximice $p(\bar{e}_i)$ (caso peor)

CANAL $\xrightarrow{\bar{r}}$ $\begin{cases} \hat{v} = \bar{v}_0 \Rightarrow \bar{e}_0 \,/\, p_0 \\ \hat{v} = \bar{v}_1 \Rightarrow \bar{e}_1 \,/\, p_1 \\ \vdots \end{cases}$

escogemos el que maximice este p_i.

$$\Rightarrow \max_{v_i}\left(K\alpha^{w(e_i)}\right) = \min_{v_i}\left(w(e_i)\right) = \min_{v_i}\left[d(\bar{v}_i, \bar{r})\right]$$

≡ escoger la palabra código más próxima a la recibida.

- Propiedades

1- Sea un código $C(n,k)$ / d_{min} } $\Rightarrow$ $C(n,k)$ es capaz de

Sea t / $2t+1 \leq d_{min} \leq 2t+2$ } corregir todos los errores de peso t o menor:

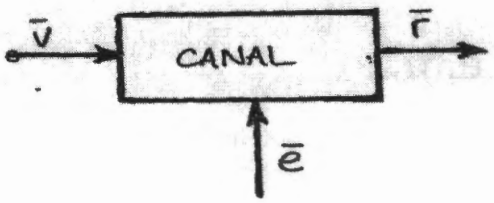
def: $t \equiv$ capacidad de corrección del código

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

Garantiza la corrección de todos los errores de peso t o menor. La corrección [illegible]

$\forall e_i / w(e_i) \leq t$

DEMO:



Sean $\bar{v}, \bar{w} \in C(n, K)$

$\Rightarrow d(\bar{v}, \bar{r}) + d(\bar{w}, \bar{r}) \geq d(\bar{v}, \bar{w})$

Supongo que $\exists\ t'$ errores $\Rightarrow w(\bar{e}) = t' \Rightarrow d(\bar{v}, \bar{r}) = t'$

Como: $d(\bar{v}, \bar{w}) \geq d_{min} \geq 2t+1$

$t' + d(\bar{w}, \bar{r}) \geq 2t+1$

$d(\bar{w}, \bar{r}) \geq 2t+1-t'$

Si $t' \leq t \Rightarrow d(\bar{w}, \bar{r}) \geq t+1 \Rightarrow \begin{cases} d(\bar{v}, \bar{r}) = t' \\ d(\bar{w}, \bar{r}) > t' \end{cases}$

Si $t' < t \Rightarrow d(\bar{w}, \bar{r}) > t$

$\Rightarrow$ El vector recibido es el más próximo al enviado.

2- El código w es capaz de corregir errores de peso mayor que t.

DEMO:

Sean $\bar{v}, \bar{w}$ palabras del código y $d(\bar{v}, \bar{w}) = d_{min}$

Sean $\bar{e}_1$ y $\bar{e}_2$ / $\bar{e}_1 + \bar{e}_2 = \bar{v} + \bar{w}$

con $\bar{e}_1$ y $\bar{e}_2$ no tienen componentes $= \emptyset$ en lugares comunes

$\Rightarrow w(\bar{e}_1 + \bar{e}_2) = w(\bar{e}_1) + w(\bar{e}_2) = w(\bar{v} + \bar{w}) = d_{min}$

$\bar{r} = \bar{v} + \bar{e}_1 \Rightarrow d(\bar{v}, \bar{r}) = w(\bar{v} + \bar{r}) = w(\bar{e}_1)$

$d(\bar{w}, \bar{r}) = w(\bar{w} + \bar{r}) = w(\bar{w} + \bar{v} + \bar{e}_1) = w(\bar{e}_2)$

Supongo que e_1 tiene más de t errores $\Rightarrow w(\bar{e}_1) > t$

Como: $2t+1 \leq d_{min} \leq 2t+2$

$$w(\bar{e}_2) = d_{min} - w(\bar{e}_1) \leq 2t+2 - w(\bar{e}_1)$$

$$w(\bar{e}_1) > t \geq t+1$$

$$w(\bar{e}_2) \leq 2t+2-(t+1) = t+1$$

Considerando $\begin{cases} d(\bar{v},\bar{r}) = w(\bar{e}_1) > t \\ d(\bar{w},\bar{r}) = w(\bar{e}_2) \leq t+1 \end{cases}$

$\Rightarrow d(\bar{w},\bar{r}) \leq d(\bar{v},\bar{r}) \Rightarrow$ Cometemos error al corregir.

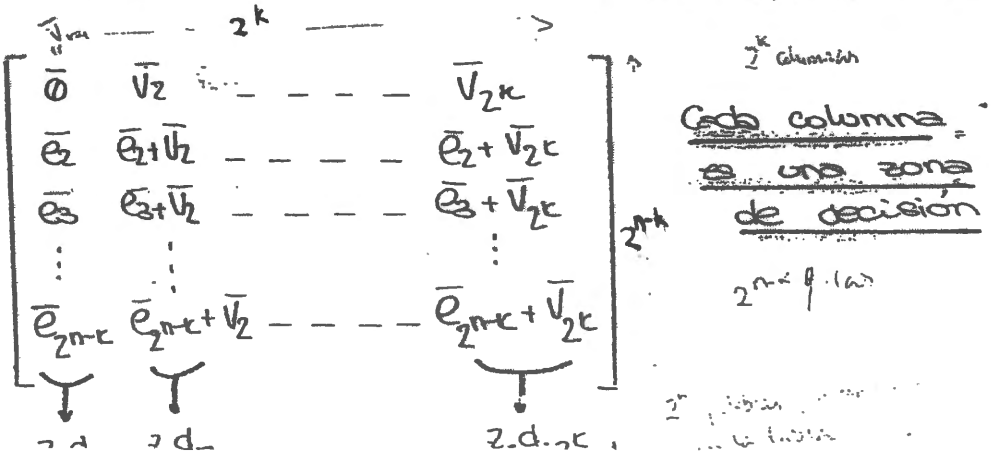
10. Tabla estander

- Método para decidir la palabra código recibida.
- Intenta particionar el espacio V_n en 2^k subconjuntos $\Rightarrow$ en cada subconjunto ∃! palabra código legal
- Método de construcción

a) 1ª fila: Vector nulo y todas las palabras código legales

b) La siguiente n-tupla escojo un vector de error $\bar{e}_2$, $\bar{e}_2 \notin C$, lo pongo debajo del nulo y relleno las demás celdas con $\bar{e}_2 + \bar{v}_i$ en la columna i

c) Repito b) pero con un $\bar{e}_i$ que no esté ya dentro de la tabla



- Teorema: En una fila todas las tuplas son distintas.

DEMO: Supongamos dos tuplas iguales.

$$\Rightarrow \bar{e}_\ell + \bar{v}_i = \bar{e}_\ell + \bar{v}_j \Rightarrow \bar{v}_i = \bar{v}_j \quad \text{IMPOSIBLE}$$
$$i \neq j \qquad\qquad i \neq j$$

- Teorema: Cada tupla aparece en una sola fila.

DEMO: Supongamos la misma tupla en dos filas distintas

$$\Rightarrow \bar{e}_\ell + \bar{v}_i = \bar{e}_m + \bar{v}_j \Rightarrow \bar{e}_\ell = \bar{e}_m + \bar{v}_i + \bar{v}_j$$
$$= \bar{e}_m + \bar{v}_s \quad \text{IMPOSIBLE}$$

- Por todo esto, queda demostrado que se generan 2^n palabras que son todos los posibles vectores de V^n

- Existen 2^{n-k} filas distintas y 2^k columnas

  * Fila $\overset{\text{def}}{=}$ Cogrupo (coset) del código

  * Primera palabra de fila $\overset{\text{def}}{=}$ líder de cogrupo (coset leaders)

- ¿Cuales de estos errores corrijo bien?

  a) $\bar{e}$ es un líder de cogrupo $\Rightarrow$ $\bar{r} = \bar{e} + \bar{v}_j \Rightarrow \bar{v}_j = \bar{r} + \bar{e} \in D_j$

  $\Rightarrow$ Siempre corrijo bien.

  b) $\bar{e}$ no es líder de cogrupo $\Rightarrow$ $\bar{r} = \bar{v}_j + \bar{x} = \bar{v}_j + (\bar{e}_\ell + \bar{v}_i) =$

  $= \bar{e}_\ell + \bar{v}_j + \bar{v}_i = e_\ell + v_s \in D_s$ $\Rightarrow$ Corrijo mal

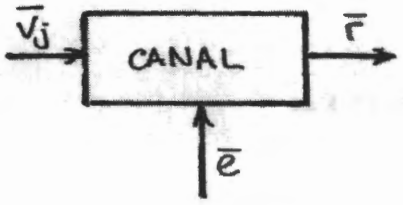
* Teorema: Un código $C(n,k)$ es capaz de corregir 2^{n-k} errores $\Rightarrow$ líderes de cogrupo.

* Por tanto, para minimizar la probabilidad de error en corrección, los líderes de cogrupo deben ser los errores más probables. = errores de menos peso. $\rightarrow w(\bar{e}) \leq t$

El [illegible] corrige todas las tuplas de la fila

$$P_{NC} \leq \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

- ¿Cual es la probabilidad de error de corrección?



Sea $\alpha_i \overset{def}{\equiv} \text{Card}\{\bar{e}_j / w(e_j) = i\}$

$\bar{e}_j$ = líder de cogrupo

{Distribución de pesos de los líderes de cogrupo.

$$P_E = P_{NC} = 1 - P_C = 1 - \sum_{i=0}^{n} \alpha_i p^i (1-p)^{n-i}$$

P_C: Prob. líder cogrupo

* Se define código perfecto como el código en el que en la distribución están todos los errores de peso ínfimo y ninguno de peso superior.

II. Métodos mixtos de detección y corrección

(x) x x | x x (x)

$$\begin{cases} d_{min} = 5 \\ s_{max} = 4 \\ t_{max} = 2 \end{cases}$$

- Dado un código, podemos hacer que funcione de formas distintas, dependiendo de nuestros intereses:

$d_{min} = 5$

① $\begin{cases} s = 4 \\ t = 0 \end{cases}$ (solo detección) (ARQ)

② $\begin{cases} s = 2 \\ t = 2 \end{cases}$ (detección y corrección) (FEC) → todo lo errores lo corrijo bien

③ $\begin{cases} s = 3 \\ t = 1 \end{cases}$ (mixto) → no todo lo que detecto lo corrijo bien.

(HÍBRIDO (FEC + ARQ))

$w(\bar{e}) = 1$ detecto y corrijo

$w(\bar{e}) = 2$ detecto

$w(\bar{e}) = 3$ detecto

$w(\bar{e}) = 4$ detecto pero corrijo mal ⇓ P_E residual

Siempre podremos hacerlo si: $d_{min} = s + t + 1$

12. Diseño de decodificadores

- **Teorema**: Todas las 2^k tuples de un cogrupo tienen el mismo síndrome.

  DEMO: Cogrupo con líder $\bar{e}_l$

  si $\bar{x} \in$ cogrupo de $\bar{e}_l \Rightarrow \bar{x} = \bar{e}_l + \bar{v}_i$

  $$\Rightarrow \bar{s} = \bar{x} \cdot H^t = (\bar{v}_i + \bar{e}_l) \cdot H^t = \cancel{\bar{v}_i H^t}^{\,0} + \bar{e}_l H^t = \bar{e}_l H^t$$

- **Teorema**: Los síndromes de distintos cogrupos son distintos

  DEMO: Supongo dos filas distintas con síndromes iguales.

  $$\Rightarrow \bar{e}_j \cdot H^t = \bar{e}_l \cdot H^t \Rightarrow (\bar{e}_j + \bar{e}_l) H^t = 0 \in C \equiv \bar{v}_s$$

  $$\Rightarrow \bar{e}_l = \bar{e}_j + \bar{v}_s \quad \text{IMPOSIBLE}$$

- El síndrome de una tupla tiene $n-k$ dígitos

  $\Rightarrow \exists\ 2^{n-k}$ síndromes posibles $\equiv$ nº cogrupos

  $\Rightarrow$ Generamos una tabla de decodificación entre síndromes y líderes de cogrupo.

- Pasos a seguir en la decodificación:

  a) $\boxed{\bar{r} \Rightarrow \bar{s} = \bar{r} \cdot H^t}$

  b) Miro a ver el cogrupo asociado y el líder.

  c) Decodifico: $\boxed{\hat{\bar{v}} = \bar{e}_i + \bar{r}}$

** Ejemplo : $G(7,4)$

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

Dado $\bar{r} \Rightarrow \bar{c} = \bar{r} \cdot H^t = (S_0, S_1, S_2) \equiv \bar{s}$

$\bar{e}_i$ $\bar{e}_i$

Tabla : Líderes de cegrupo - Síndrome asociado

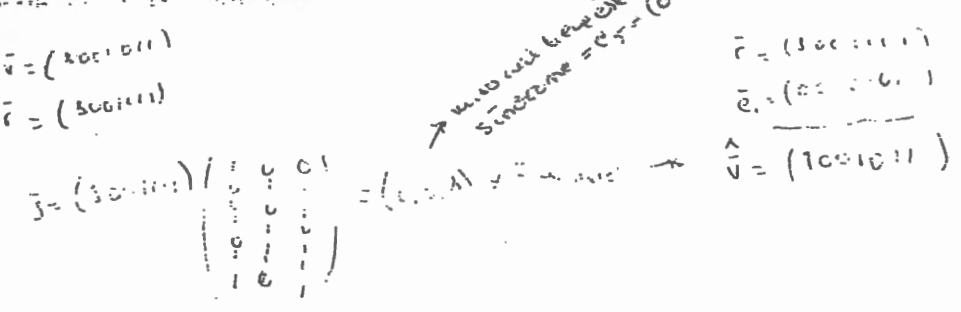
	e_i	$s_i = e_i H^t$
0	(0 0 0 0 0 0 0)	(0 0 0)
1	(1 0 0 0 0 0 0)	(1 0 0)
2	(0 1 0 0 0 0 0)	(0 1 0)
3	(0 0 1 0 0 0 0)	(0 0 1)
4	(0 0 0 1 0 0 0)	(1 1 0)
5	(0 0 0 0 1 0 0)	(0 1 1)
6	(0 0 0 0 0 1 0)	(1 1 1)
7	(0 0 0 0 0 0 1)	(1 0 1)

¿Cómo elijo los líderes de cegrupo?

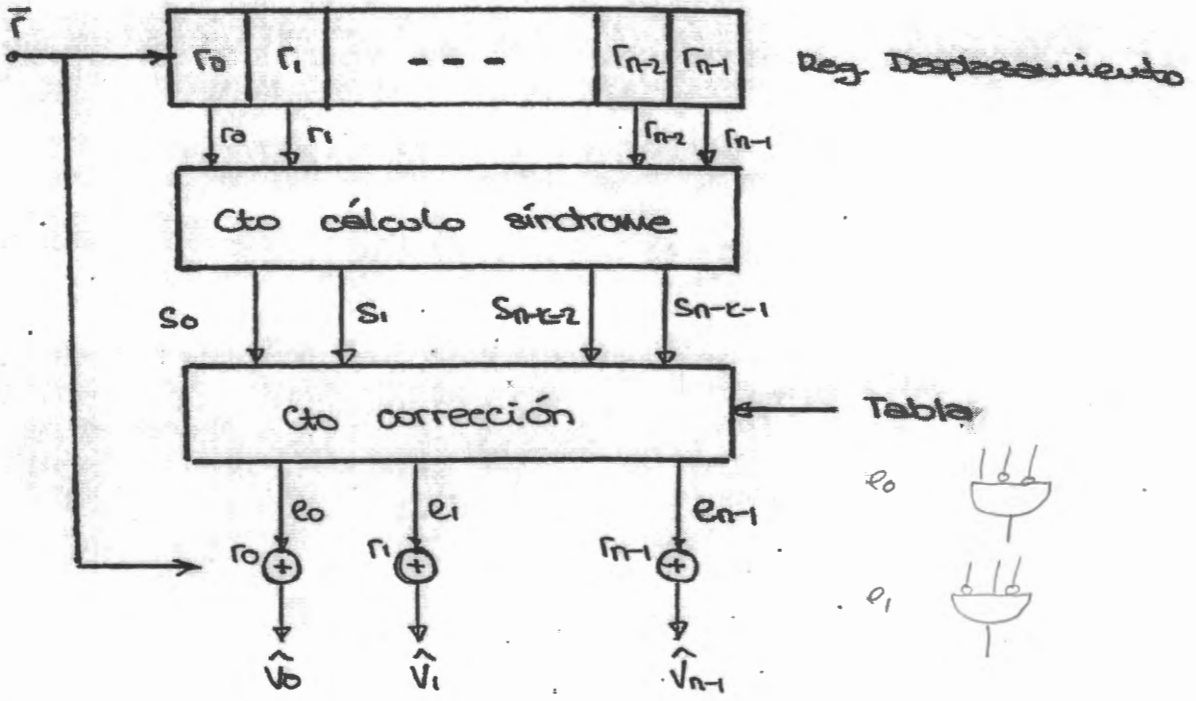
1º De menor peso (más probables)

2º Asegurarse de que $t \geq$ peso líderes

(En este caso, $t = 1 \Rightarrow$ corrijo bien todos los errores de peso 1)



- El decodificador será de la forma:



3. Diseño de códigos. Códigos de Hamming

- Son difíciles de implementar en general

Hamming desarrolló un algoritmo para la generación de códigos lineales:

* $\forall\ m \geq 3\ ,\ \exists$ código Hamming $\Rightarrow$ $\begin{cases} n = 2^m - 1 \\ k = 2^m - m - 1 \\ d_{min} = 3 \text{ (siempre)} \end{cases}$ $n - k = m$

* La matriz H tiene todas las n-tuplas como columnas, excepto el vector nulo. $H = (I_m : Q)$

* Los líderes de cogrupo son $2^m - 1$ de peso 1 y el $\bar{0}$

* Los códigos duales de los códigos Hamming también cumplen las mismas propiedades.

(Ej)

$m=3 \Rightarrow n = 2^3 - 1 = 7$

$k = 2^3 - 3 - 1 = 4$

$$H = \left(\begin{array}{ccc:cccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right) \qquad G = \left(Q^t : I_k\right)$$

Tema 7. Codigos cíclicos

V 1

.1- Introducción. Notación polinomial

- Definición: Sea $\bar{v} = (v_0\ v_1 \dots v_{n-1})$

* Definimos rotación cíclica $\bar{v}^{(1)} \overset{def}{=} (v_{n-1}\ v_0 \dots v_{n-2})$

en general $\bar{v}^{(i)} = (v_{n-i}\ v_{n-i+1} \dots v_{n-1}\ v_0\ v_1 \dots v_{n-i-1})$

- Un código cíclico se define como el código lineal en el que se cumple $\forall\, \bar{v} \in C \Rightarrow \boxed{\bar{v}^{(i)} \in C}$ → las rotaciones de cualquier vector pertenecen al cód. x.

- Podemos expresar cualquier vector $\bar{v}$ como un polinomio de grado n-1:

$$\bar{v} = (v_0\ v_1 \dots v_{n-1}) \Rightarrow V(x) = v_0 + v_1 x + v_2 x^2 + \dots + v_{n-1} x^{n-1}$$

con grado $V(x) \leq n-1$

* Es la forma más sencilla de relacionar la rotación cíclica:

$$\bar{v} = (v_0\ v_1 \dots v_{n-1}) \Rightarrow V(x) = v_0 + v_1 x + \dots + v_{n-1} x^{n-1}$$

$$\bar{v}^{(i)} = (v_{n-i}\ v_{n-i+1} \dots v_{n-1}\ v_0\ v_1 \dots v_{n-i-1})$$

$$\Rightarrow V^{(i)}(x) = v_{n-i} + v_{n-i+1} x + \dots + v_{n-1} x^{i-1} + v_0 x^i + \dots + v_{n-i-1} x^{n-1}$$

* Multiplicando: $\boxed{V(x) \cdot x^i}$, obtenemos:

$$V(x) x^i = v_0 x^i + v_1 x^{i+1} + \dots + v_{n-1} x^{n+i-1} =$$

$$= v_{n-i} + v_{n-i+1} x + \dots + v_{n-1} x^{i-1} + v_0 x^i + \dots + v_{n-i-1} x^{n-1} + \text{(also)}$$

$$\text{(also)} = v_{n-i}(x^n + 1) + v_{n-i+1} x (x^n + 1) + \dots + v_{n-1} x^{i-1} (x^n + 1)$$

* Resumiendo:

$$V(x)\cdot x^i = \underbrace{q(x)}_{\text{cociente}}\underbrace{(x^n+1)}_{\text{divisor}} + \underbrace{V^{(i)}(x)}_{\text{resto}}$$

dividendo

$$\Rightarrow \boxed{V^{(i)}(x) = \text{resto}\left[\frac{V(x)\,x^i}{(x^n+1)}\right]}$$

7.2- Polinomio de grado mínimo

- Sea $C(n,k)$ con:

$\bar{V}_0 \rightarrow V_0(x)$

$\bar{V}_1 \rightarrow V_1(x)$

$\vdots \qquad \vdots$

$\bar{V}_{2^n} \rightarrow V_{2^n}(x)$

- Se define la palabra $\bar{g} \rightarrow g(x)$, $\bar{g} \neq \bar{0}$ / $\bar{g}$ es de grado mínimo.

- Teorema.

El polinomio código distinto del $\bar{0}$ de grado mínimo es único.

$g_0 = g_r = 1$ obligatorio!!!

DEMO:

Sea $g(x) = g_0 + g_1 x + g_2 x^2 + \dots + g_{r-1}x^{r-1} + x^r$, grado r (mínim.)

Supongo que $\exists\ g'(x) = g_0' + g_1' x + g_2' x^2 + \dots + g_{r-1}' x^{r-1} + x^r$

$\Rightarrow g(x) + g'(x) \in C$ (cod. lineal)

$$\Rightarrow g(x)+g'(x) = (g_0+g_0') + (g_1+g_1')x + \dots + (g_{r-1}+g_{r-1}')x^{r-1} + 0$$

$x^r + x^r = x^r(1+1) = x^r \cdot 0 = 0$

grado $(g(x)+g'(x)) = r-1 < r$!! IMPOSIBLE

- **Teorema**

Si $g(x)$ es de grado mínimo $\Rightarrow$ $g_0 = 1$

DEMO:

Supongo $g_0 = 0 \Rightarrow g(x) = g_1 x + g_2 x^2 + \dots + g_{r-1} x^{r-1} + x^r$

Desplazando $r-1$ posiciones:

$$g'(x) = g_1 + g_2 x + g_3 x^2 + \dots + g_{r-1} x^{r-2} + x^{r-1} \in C$$

$$\text{grado } g'(x) = r-1 < r \quad !! \text{ IMPOSIBLE}$$

$\Rightarrow$ **El polinomio de grado mínimo será de la forma:**

$$g(x) = 1 + g_1 x + g_2 x^2 + \dots + g_{r-1} x^{r-1} + x^r$$

¿**Por qué es tan importante?**

$g(x) \rightarrow$ grado r

$x g(x) \rightarrow$ grado $r+1$

$\vdots \qquad \vdots$

$x^{n-r-1} g(x) \rightarrow$ grado $n-1$

} Todas estas palabras pertenecen a un mismo código.

* Como $x^i V(x) = q(x)(x^n+1) + V^{(i)}(x)$

$\Rightarrow$ Cuando grado $(x^i V(x)) < n$, entonces $x^i V(x) = V^{(i)}(x)$

Suponemos:

$$V(x) = u_0 g(x) + u_1 x g(x) + \dots + u_{n-r-1} x^{n-r-1} g(x) = u(x) g(x) \in C$$

$\Rightarrow$ **todos los multiplos de $g(x)$ que tengan grado $n-1 \in C$**

Teorema

Sea $g(x) = 1 + g_1 x + g_2 x^2 + \dots + g_{r-1} x^{r-1} + x^r$ polinomio de grado mínimo de C, entonces un polinomio de grado menor o igual a $n-1 \in C \Leftrightarrow$ es múltiplo de $g(x)$

DEMO:

$\Leftarrow$) Si es múltiplo $\Rightarrow \in C$

$$V(x) = (a_0 + a_1 x + \dots + a_{n-r-1} x^{n-r-1})\, g(x) =$$

$$= \underbrace{a_0 g(x)}_{\in C} + \underbrace{a_1 x g(x)}_{\in C} + \dots + \underbrace{a_{n-r-1} x^{n-r-1} g(x)}_{\in C} \Rightarrow V(x) \in C$$

$\Rightarrow$) Si $\in C \Rightarrow$ múltiplo de $g(x)$

Supongo que no es multiplo:

$$V(x) = a(x) g(x) + b(x), \quad b(x) \neq 0$$

$$\Rightarrow b(x) = \underbrace{V(x)}_{\in C} + \underbrace{a(x) g(x)}_{\in C} \Rightarrow b(x) \in C$$

Grado: $V(x) = \underbrace{a(x) g(x)}_{\text{grado } r} + \underbrace{b(x)}_{\text{grado} < r}$

$\Rightarrow b(x)$ mínimo grado!! IMPOSIBLE

$\Rightarrow$ Los polinomios código son múltiplos de $g(x)$, grado r

entonces:

$$V(x) = a(x)\, g(x) = (a_0 + a_1 x + a_2 x^2 + \dots + a_{n-r-1} x^{n-r-1})\, g(x)$$

Obtenemos 2^{n-r} múltiplos $\equiv$ nº palabras código

$$\Rightarrow \boxed{2^k = 2^{n-r} \Rightarrow r = n-k}$$

EL GRADO DEL POLINOMIO [illegible]

- Por tanto, $g(x) = 1 + g_1 x + g_2 x^2 + \dots + g_{n-k-1} x^{n-k-1} + x^{n-k}$ se define también como el **polinomio generador**, de modo que:

$$\underbrace{V(x)}_{n-1} = \underbrace{U(x)}_{k-1}\ \underbrace{g(x)}_{n-k}$$

$V(x) \equiv$ palabra código

$U(x) \equiv$ palabra fuente

$\omega(\bar{g}) = 3$

ojo! $\omega(\bar{g}) =$ puede ser cualquiera no tiene por qué ser el peso mínimo ($\geq n-k$)

** Ejemplo: $C(7,4)$, $g(x) = 1 + x + x^3$

$\bar{g} \to (1101000)$
$\bar{g}^{(1)} \to (0110100)$
$\bar{g}^{(2)} \to (0011010)$
$\bar{g}^{(3)} \to (0001101)$
$\bar{g}^{(4)} \to (1000110)$
$\bar{g}^{(5)} \to (0100011)$
$\bar{g}^{(6)} \to (1010001)$
$\bar{0} \to (0000000)$

$\Big\} \otimes$

$(1011100) \to \bar{v}$
$(0101110) \to \bar{v}^{(1)}$
$(0010111) \to \bar{v}^{(2)}$
$(1001011) \to \bar{v}^{(3)}$
$(1100101) \to \bar{v}^{(4)}$
$(1110010) \to \bar{v}^{(5)}$
$(0111001) \to \bar{v}^{(6)}$
$(1111111) \to \bar{g} + \bar{v}^{(2)}$

pero: $x^7 + x^5 + 1$ $\omega(\bar{g}) = 3$ $g^{\dots} = 7$

$\Rightarrow 16 = 2^C$ vectores que forman el código (p.ej)

* Cogiendo K vectores l.i. consigo una base del subespacio vectorial

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

(no sistemática)

- Teorema

El polinomio $g(x)$ de $G(n,k)$ es factor de (x^n+1)

DEMO:

$g(x) \Rightarrow$ grado $g(x) = n-k$

$x^k g(x) \Rightarrow$ grado $x^k g(x) = n$

$$x^k g(x) = \underbrace{a(x)}_{\parallel 1}(x^n+1) + \underbrace{g^{(k)}(x)}_{\in G \equiv b(x)g(x)} \quad \text{rotación de orden } k \text{ de } g(x)$$

$$\Rightarrow x^k g(x) = (x^n+1) + b(x)g(x)$$

$$\Rightarrow (x^n+1) = (x^k + b(x))\, g(x)$$

** Ejemplo: $G(7,4) \Rightarrow (x^7+1) = a(x)\,b(x)\,c(x)$

el polinomio generador será el de grado 3.

Pero, ¿ $\forall n,k, \exists\ G(n,k)$ cíclico?

- Teorema

Si $g(x)$ es polinomio de grado $n-k$ y es factor de $(x^n+1) \Rightarrow$ genera un código cíclico $G(n,k)$

DEMO: Sea $V(x) = v_0 + v_1 x + \dots + v_{n-1} x^{n-1}$ palabra código

$$\underbrace{xV(x)}_{\text{divisible por } g(x) \Rightarrow \in G} = v_{n-1}\underbrace{(x^n+1)}_{\text{divisible por } g(x)} + \underbrace{V^{(1)}(x)}_{\in G \Rightarrow \text{ divisible x } g(x) \Rightarrow \in G \Rightarrow \boxed{G \text{ es cíclico.}}}$$

Ejemplo: $g(x) = 1+x+x^2+x^4+x^6 = (1+x^3+x^4)(1+x+x^2+x^3+x^4)$

factor de $(1+x^{2^4-1})$ | factor de $(1+x^{2^4-1})$

* Ejemplo: $n=7$

$$\Rightarrow (x^7+1) = \dots = (1+x)(1+x+x^3)(1+x^2+x^3)$$

* El número de codigos posibles será: $2^l - 2$

siendo l el número de factores.

$r=3 : n-k \Rightarrow k=4$

$g_1(x) = 1+x \Rightarrow C_1(7,6)$

$g_2(x) = 1+x+x^3 \Rightarrow C_2(7,4)$

$g_3(x) = 1+x^2+x^3 \Rightarrow C_3(7,4)$

$g_4(x) = (1+x)(1+x+x^3) \Rightarrow C_4(7,3)$

$g_5(x) = (1+x)(1+x^2+x^3) \Rightarrow C_5(7,3)$

$g_6(x) = (1+x+x^3)(1+x^2+x^3) \Rightarrow C_6(7,1)$

* Si escogemos $g_2(x) = 1+x+x^3 \Rightarrow \bar{u} = (1010)$ (1er factor)

$$\Rightarrow v(x) = u(x)g(x) = (1+x^2)(1+x+x^3) =$$

$$= 1+x+x^3+x^2+x^3+x^5 = 1+x+x^2+x^5$$

$$\Rightarrow \bar{v} = (1110010)$$ no sistemático

1.3- Codificación sistemática.

- Sea $\bar{u} = (u_0\ u_1 \dots u_{k-1}) \Rightarrow u(x) = u_0 + u_1 x + \dots + u_{k-1} x^{k-1}$

$\Rightarrow$ Queremos generar una palabra codigo de la forma:

$$\boxed{\bar{v} = (- - - - \dots - u_0\ u_1 \dots u_{k-1})}$$

De modo que habrá que desplazar la palabra $\bar{u}$ $n-k$ posiciones de forma que:

$$\boxed{v(x) = \underbrace{\quad}_{k-1} + \underbrace{u(x)}_{n-k} x^{n-k}}$$

$$\Rightarrow \underbrace{x^{n-k}u(x)}_{n-1} = \underbrace{a(x)}_{k-1}\underbrace{g(x)}_{n-k} + \underbrace{b(x)}_{\leq n-k-1}$$

$n-1$

- Por tanto, $b(x) \Rightarrow$ grado $\leq n-k-1$.

$$\Rightarrow \boxed{b(x) = b_0 + b_1 x + b_2 x^2 + \dots + b_{n-k-1} x^{n-k-1}}$$

- Reagrupando:

$$\boxed{b(x) + x^{n-k} u(x) = \underbrace{a(x) g(x)}_{\in C}}$$

$$\Rightarrow a(x) g(x) = b_0 + b_1 x + b_2 x^2 + \dots + b_{n-k-1} x^{n-k-1} + u_0 x^{n-k} + \dots + u_{k-1} x^{n}$$

* es la palabra código que busco, por tanto para generar un código de manera sistemática tengo que:

1. Hallar $x^{n-k} u(x)$
2. Hallar $\boxed{b(x) = \text{resto} \dfrac{x^{n-k} u(x)}{g(x)}}$
3. $\boxed{v(x) = b(x) + x^{n-k} u(x)}$

** Ejemplo: $C(7,4)$, $g(x) = 1 + x + x^3$

$\bar{u} = (1001) \Rightarrow u(x) = 1 + x^3$

1. $x^3 u(x) = x^3 + x^6$
2. $b(x) = \text{resto} \dfrac{x^6 + x^3}{1 + x + x^3}$

$$\begin{array}{r|l} x^6 + x^3 & x^3 + x + 1 \\ \hline x^6 + x^4 + x^3 & x^3 + x \\ \hline x^4 & \\ x^4 + x^2 + x & \\ \hline x^2 + x = b(x) & \end{array}$$

3. $v(x) = x + x^2 + x^3 + x^6$

$\Rightarrow \bar{v} = (0\,1\,1\,1\,0\,0\,1)$

$b \quad u$

$b_0 = 0$
$b_1 = 1$
$b_2 = 1$

- Matriz generadora.

- Para formar la matriz generadora de un código lineal cíclico, hay que coger K vectores del código linealmente independientes:

$$\bar{v} = \bar{u} \cdot G \quad , \quad G = \begin{bmatrix} \bar{g}_0 \\ \bar{g}_1 \\ \vdots \\ \bar{g}_{k-1} \end{bmatrix} \Rightarrow \text{Generan el subespacio vectorial}$$

- Una forma fácil de encontrar estos K vectores l.i. es coger los primeros K rotaciones del vector generador $\bar{g}$.

- Por tanto G será de la forma:

$$G = \begin{bmatrix} g_0 & g_1 \dots g_{n-k} & \emptyset & \dots & \emptyset \\ \emptyset & g_0 \dots g_{n-k-1} & g_{n-k} & \dots & \emptyset \\ \vdots & \vdots & & & \\ \emptyset & \emptyset \dots g_0 & g_1 & \dots & g_{n-k} \end{bmatrix}_{k \times n} = \begin{pmatrix} g(x) \\ x\,g(x) \\ x^2 g(x) \\ \vdots \\ x^{k-1} g(x) \end{pmatrix}$$

* Ejemplo: $C(7,4)$, $g(x) = 1 + x + x^3$

Diagonal

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \quad \text{No sistemática.}$$

* Para hacer G sistemática diagonalizamos a la derecha:

$$G = \left[\begin{array}{ccc|cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \begin{matrix} \\ \\ = f_1 + f_3 \\ = f_1 + f_2 + f_4 \end{matrix} \quad \text{Sistemática.}$$

Diagonalizar a partir de columna g_{n-k} = 3 ej. Contando a partir de 0 ... 3

- Si $v(x) = b(x) + u(x)\,x^{n-k}$, $b(x) = \text{resto}\ \dfrac{x^{n-k}\,u(x)}{g(x)}$

MÉTODO DE LOS RESTOS (Alternativa a Gauss)

⇓

⇒ Haciendo $\bar{u}_0 = (1000)$; $\bar{u}_1 = (0100)$; $\bar{u}_2 = (0010)$; $\bar{u}_3 = (0001)$

(Filas de la submatriz diagonal)

Obtenemos: caso general $b_i = \text{resto}\ \dfrac{x^{n-k+i}}{g(x)}$

$$V_0(x) = b_0(x) + x^{n-k} \quad , \quad b_0(x) = \text{resto}\ \frac{x^{n-k}}{g(x)}$$

$$V_1(x) = b_1(x) + x^{n-k+1} \quad , \quad b_1(x) = \text{resto}\ \frac{x^{n-k+1}}{g(x)}$$

$$V_2(x) = b_2(x) + x^{n-k+2} \quad , \quad b_2(x) = \text{resto}\ \frac{x^{n-k+2}}{g(x)}$$

$$V_3(x) = b_3(x) + x^{n-k+3} \quad , \quad b_3(x) = \text{resto}\ \frac{x^{n-k+3}}{g(x)}$$

Que son las filas de la submatriz P.

* En general: $b_i(x) = b_{i0} + b_{i1}x + b_{i2}x^2 + \dots + b_{in-k-1}x^{n-k-1}$

$$\Rightarrow G = \begin{bmatrix} b_{00} & b_{01} & \dots & b_{0\,n-k-1} & 1 & 0 & \dots & 0 \\ b_{10} & b_{11} & \dots & b_{1\,n-k-1} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ b_{k-1\,0} & b_{k-1\,1} & \dots & b_{k-1\,n-k-1} & 0 & 0 & \dots & 1 \end{bmatrix}$$

** Ejemplo: $C(7,4)$; $g(x) = 1 + x + x^3$

$$\bar{u}_0 = (1000) \Rightarrow u_0(x) = 1 \Rightarrow b_0 = \text{resto}\ \frac{x^3}{1+x+x^3} = 1+x$$

$$\bar{u}_1 = (0100) \Rightarrow u_1(x) = x \Rightarrow b_1 = \text{resto}\ \frac{x^4}{1+x+x^3} = x+x^2$$

$$\bar{u}_2 = (0010) \Rightarrow u_2(x) = x^2 \Rightarrow b_2 = \text{resto}\ \frac{x^5}{1+x+x^3} = 1+x+x^2$$

$$\bar{u}_3 = (0001) \Rightarrow u_3(x) = x^3 \Rightarrow b_3 = \text{resto}\ \frac{x^6}{1+x+x^3} = 1+x^2$$

$$\Rightarrow G = \left[\begin{array}{cccc|cccc} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

C
1
0
0
0
1
0
0

5. Matriz H

- Partimos de que $g(x)$ es factor de (x^n+1).

$$\Rightarrow \boxed{(x^n+1) = g(x)\cdot h(x)}$$

$$h(x) = h_0 + h_1 x + \dots + h_k x^k$$

Cumpliendose además que $h_0 = h_k = 1$

- Se demuestra que H se puede generar a partir de este polinomio $h(x)$:

Sea $\bar{v} \in C / \bar{v} = (v_0\ v_1 \dots v_{n-1})$

$$\Rightarrow v(x) = a(x)\,g(x) \;;\; h(x)\,v(x) = a(x)\cdot g(x)\cdot h(x) = a(x)(x^n+1)$$

$$\Rightarrow v(x)\,h(x) = a(x) + x^n a(x)$$

grado: $0 \dots k-1$ grado: $n \dots n+k-1$

$\Rightarrow$ Hay un "hecio" de grados desde $k-1$ hasta n $\Rightarrow$ los coeficientes de grado k hasta $n-1 = 0$

$$v(x)\cdot h(x) = (v_0 + v_1 x + \dots + v_{n-1} x^{n-1})(h_0 + h_1 x + \dots + h_k x^k)$$

$$= (v_0 h_0) + (v_0 h_1 + v_1 h_0)x + \dots$$

$\Rightarrow$ Quedan $n-k$ ecuaciones de las condiciones anteriores:

$$\boxed{\sum_{i=0}^{k} h_i\, v_{n-i-j} = 0 \quad ; \quad 1 \le j \le n-k}$$

$h(x)$ = polinomio de paridad

$x^k h(x^{-1})$ ≡ polinomio generador del espacio vectorial dual de $g(x)$ ∈

- Defino el polinomio recíproco de $h(x) \equiv x^k h(x^{-1})$

$= h_k + h_{k-1}x + \dots + h_0 x^k$ ⇒ Factor de (x^n+1).

⇒ Genera un código cíclico $C(n, n-k)$

$$H = \begin{bmatrix} h_k & h_{k-1} & \dots & h_1 & h_0 & 0 & \dots & 0 \\ 0 & h_k & \dots & h_2 & h_1 & h_0 & \dots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \dots & h_k & h_{k-1} & h_{k-2} & \dots & h_0 \end{bmatrix}_{n-k \times n}$$

matriz de paridad de C

* Se cumple que $\bar{w}_i \cdot \bar{v}_j = 0$ (por las ecuaciones anteriores)

⇒ Filas ortogonales $\forall\, \bar{v} \in C$

⇒ Genera el código dual de C

* $h(x)$ = polinomio de paridad de C

- Sea $C(n,k)$ cíclico con $g(x)$ generador ⇒ Su código dual es cíclico y se genera mediante $x^k h(x^{-1})$ donde

$$h(x) = \frac{x^n+1}{g(x)}$$

** Ejemplo: $C(7,4)$, $g(x) = 1 + x + x^3$

$$h(x) = \frac{x^7+1}{1+x+x^3} = x^4 + x^2 + x + 1$$

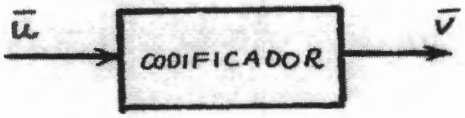
Recíproco : $x^k h(x^{-1}) = x^4(1 + x^{-1} + x^{-2} + x^{-4}) = 1 + x^2 + x^3 + x^4$

$$H = \left[\begin{array}{ccc|cccc} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right] \Rightarrow H = \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}\right]$$

(No sistemática) (Sistemática)

$$G \equiv (P \mid I_k)\ ;\ H \equiv (I_{n-k} \mid P^t)$$

6- Codificación de códigos cíclicos

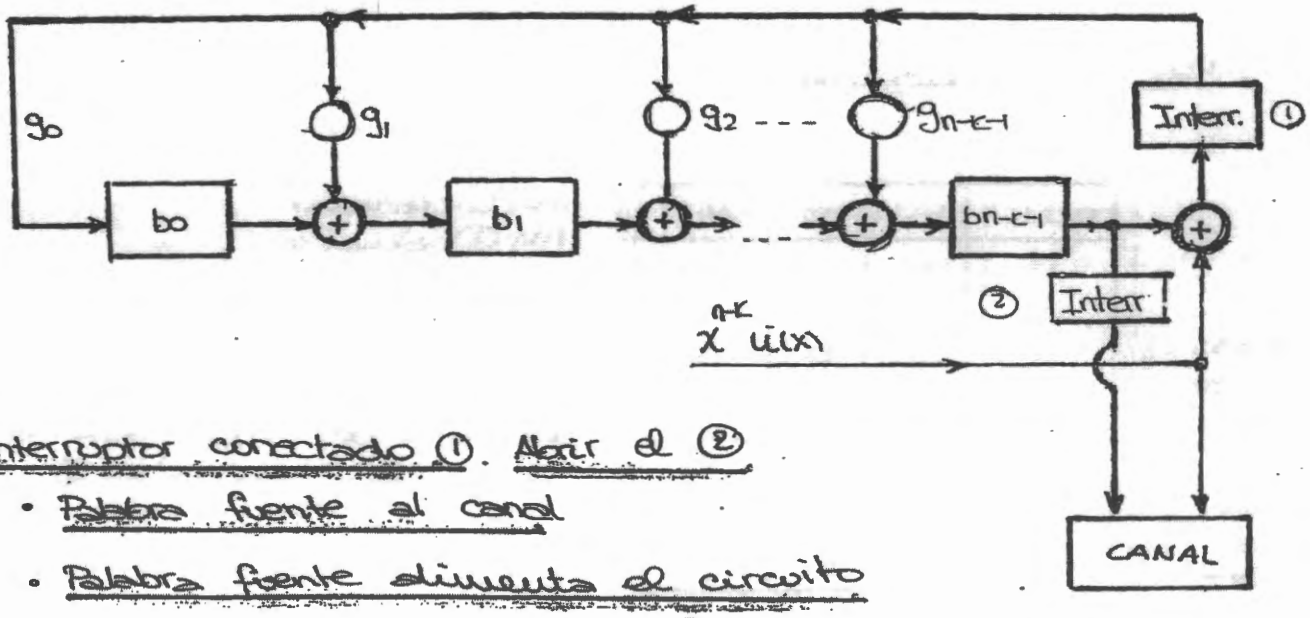


- Determinación de $\bar{v}$.

1. $x^{n-k}\,u(x)$
2. $b(x) = \text{resto}\,\dfrac{x^{n-k}\,u(x)}{g(x)}$
3. $v(x) = b(x) + x^{n-k}\,u(x)$

- Implementación del circuito lógico del codificador

⇒ Registro de desplazamiento con realimentación.



a) Interruptor conectado ①. Abrir el ②

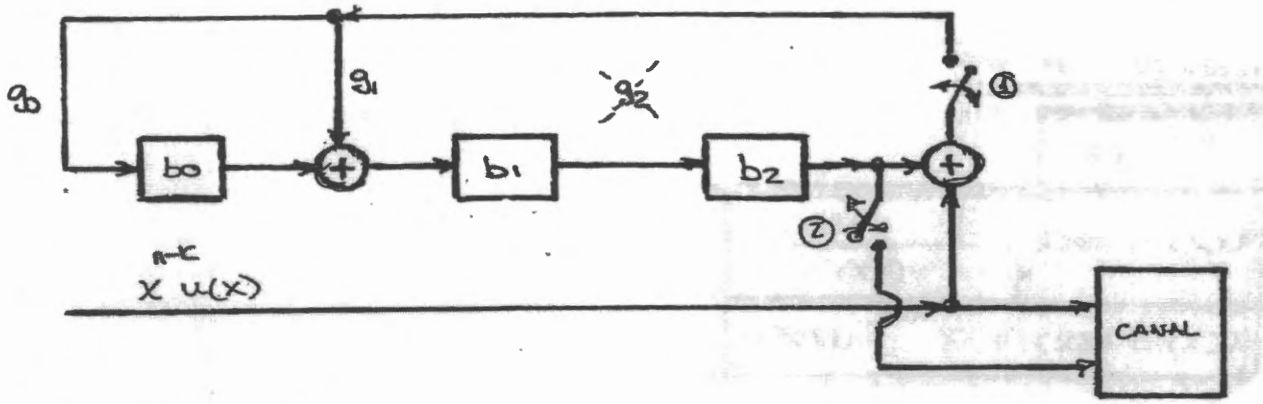
- Palabra fuente al canal
- Palabra fuente alimenta el circuito

b) Abrir interruptor ①. Cerrar el ②

- Contenido de los registros al canal (sin realimentación)

** Ejemplo: $G(7,4)$, $g(x) = 1 + x + x^3$

* Circuito:



Si $\bar{u} = (1011)$

$\bar{v} = (v_0\ v_1\ v_2\ u_0\ u_1\ u_2\ u_3)$

$x^3 u(x) = x^3 + x^5 + x^6 \equiv (0001011)$ $\quad s(x) = 1$.

* Tabla de funcionamiento:

	ENTRADA	b0 b1 b2	CANAL
Entrada el vector fuente	1	1 1 0	1
	1	1 0 1	1
	0	1 0 0	0
	1	1 0 0	1
Salida de los registros	-	0 1 0	0
	-	0 0 1	0
	-	0 0 0	1

int ① cerrado
int ② abierto

Apertura del interruptor ①
cierre del interruptor ②

$\Rightarrow \quad v(x) = 1 + x^3 + x^5 + x^6$

$\Rightarrow \bar{v} = (1001011)$

Codificación basada en h(x)

$$h(x) = h_0 + h_1 x + \dots + h_K x^K \quad , \quad h_K = h_0 = 1$$

$$\text{con} \sum_{i=0}^{K} h_i V_{n-j-i} = 0 \quad , \quad 1 \le j \le n-K$$

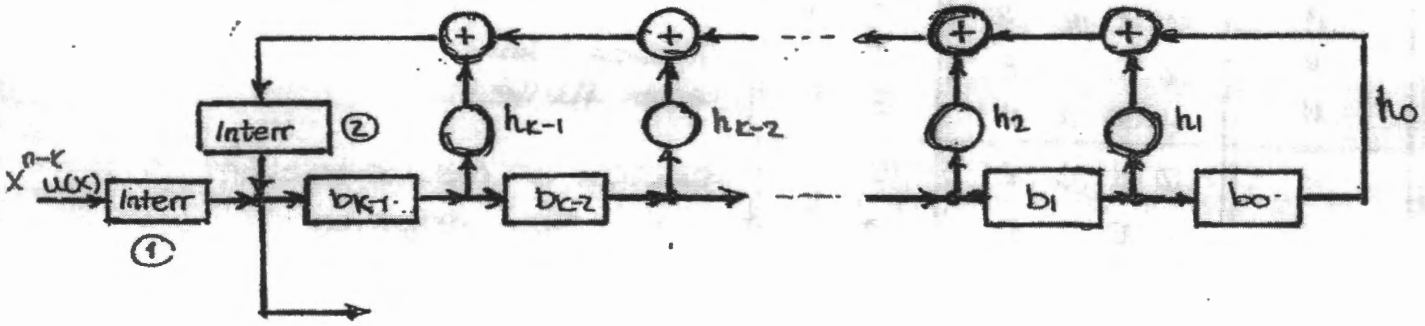
$$\text{Con } h_K = 1 \Rightarrow -V_{n-K-j} = \sum_{i=0}^{K-1} h_i V_{n-i-j} \quad , \quad 1 \le j \le n-K$$

$$\Rightarrow \bar{v} = (v_0 \; v_1 \dots v_{n-K-1} \; \underbrace{v_{n-K}}_{u_0} \dots \underbrace{v_{n-1}}_{u_{K-1}})$$

$$\text{con } j=1 \Rightarrow \boxed{V_{n-K-1} = \sum_{i=0}^{K-1} h_i V_{n-i-1}}$$

⇒ Puedo calcular la redundancia en función de los últimos términos.

* El circuito que implemente el codificador:



a) Interruptor 1 conectado. Interruptor 2 abierto

- Palabra fuente al canal
- Palabra fuente al cto.

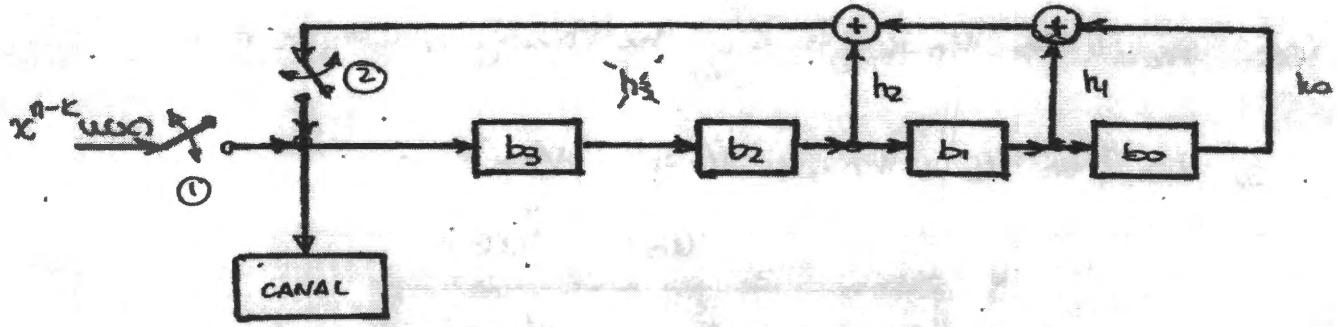
b) Interruptor 1 abierto. Interruptor 2 cerrado

- Contenido de los registros al canal y realimentamos

** Ejemplo: $C(7,4)$, $g(x) = 1 + x + x^3$

$$h(x) = \frac{(x^7+1)}{1+x+x^3} = 1 + x + x^2 + x^4$$

* Circuito:



$\bar{u} = (1011) \Rightarrow x^3 u(x) = x^3 + x^5 + x^6 \equiv (0001011)$

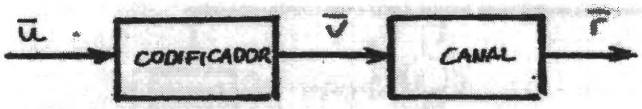
ENTRADA	b_3 b_2 b_1 b_0	CANAL
1	1 0 0 0	1
1	1 1 0 0	1
0	0 1 1 0	0
1	1 0 1 1	1
–	0 1 0 1	0
–	0 0 1 0	0
–	1 0 0 1	1

Entrada del vector fuente

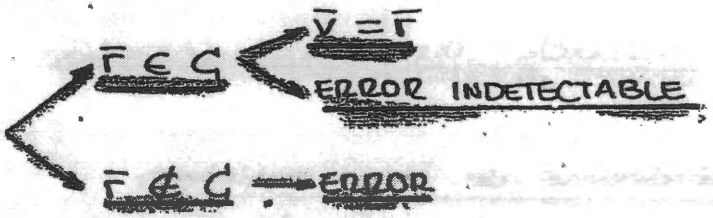
Salida y realimentación de los registros

$\Rightarrow V(x) = 1 + x^3 + x^5 + x^6 \equiv (1001011)$

- Síndrome.



- Al transmitir el código por un canal nos pueden pasar dos cosas:



- El vector recibido $\bar{r}$ tiene un polinomio asociado de la forma:

$$r(x) = r_0 + r_1 x + r_2 x^2 + \dots + r_{n-1} x^{n-1}$$

- ¿Cómo puedo saber si tengo un error?

Si dividimos entre $g(x)$:

$$r(x) = a(x)g(x) + S(x)$$

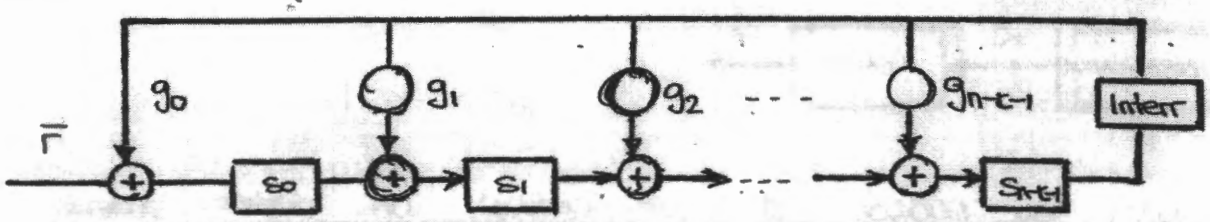
Existen dos posibilidades:

- $\bar{r} \in C \Rightarrow r(x)$ múltiplo de $g(x) \Rightarrow r(x) = a(x)g(x)$
- $\bar{r} \notin C \Rightarrow r(x)$ no es múltiplo de $g(x) \Rightarrow r(x) = a(x)g(x) + S(x)$
  $\Rightarrow S(x) \neq 0 \Rightarrow S(x) =$ Polinomio síndrome.

$\Rightarrow$ El polinomio síndrome determina si $r(x) \in C$ o no.

$$S(x) = \text{resto } \frac{r(x)}{g(x)}$$

- Circuito de calculo de síndrome & calcular el resto $\frac{r(x)}{g(x)}$



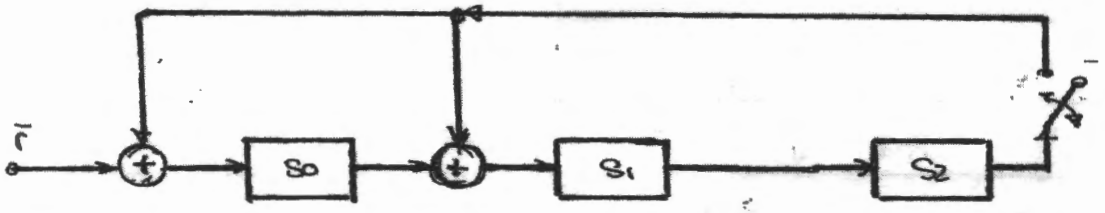
a) Interruptor cerrado

- Introducimos el vector recibido en realimentación

b) Interruptor abierto

- Obtenemos el vector síndrome de los registros

* Ejemplo: $C(7,4)$, $g(x) = 1 + x + x^3$



Si $\bar{r} = (0010110)$

Entrada	s_0	s_1	s_2
0	0	0	0
1	1	0	0
1	1	1	0
0	0	1	1
1	0	1	1
0	1	1	1
0	1	0	1

$S(x) = 1 + x^2 = (101) \neq \bar{0}$

$\Rightarrow \notin C$

- Si el error recibido se considera aditivo:

$$\bar{r} = \bar{v} + \bar{e} \Leftrightarrow \boxed{r(x) = v(x) + e(x) = b(x)g(x) + e(x)}$$

Además: $\boxed{r(x) = a(x)g(x) + s(x)}$

* Igualando: $b(x)g(x) + e(x) = a(x)g(x) + s(x)$

$$\Rightarrow \boxed{e(x) = (b(x) + a(x))\, g(x) + s(x)}$$

$\Rightarrow$ el síndrome se puede calcular también como:

$$\boxed{s(x) = \text{resto}\ \frac{e(x)}{g(x)}}$$

7.8 Capacidad de detección

- Sea el código $C_1(n,k) \Rightarrow d_{min} \begin{cases} s_{max} = d_{min} - 1 \\ t_{max} = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor \end{cases}$

- Propiedades

* Si $g(x) \neq 1 \Rightarrow$ El código detecta errores simples

DEMO: Error simple $\neq$ peso = 1 $\Rightarrow e(x) = x^j$

Si el error fuera indetectable $\Rightarrow e(x) = \dot{g}(x)$

$\Rightarrow$ Si $g(x) = 1 + \ldots + x^{n-k}$

Entonces $x^j \neq \dot{g}(x)$!!

* Si $g(x) = (1+x)^{\cdot}$ (múltiplo de) $\Rightarrow$ Este código detecta todos los errores impares.

DEMO: Error impar $\Rightarrow e(1) = 1 \Rightarrow e(x) \neq (1+x)^{\cdot}$

$\Rightarrow e(x)$ nunca puede ser multiplo de $g(x)$

$$e(x) \neq \dot{g}(x)$$

Definimos polinomio primitivo ⇔

a) Es indivisible.

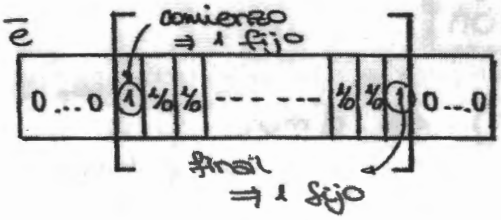
b) Es factor de $(x^{2^r-1}+1)$

c) No es factor de (x^m+1) $m < 2^r-1$.

Si $g(x)$ es múltiplo de un polinomio primitivo y

$n \leq 2^r-1 \Rightarrow C(n,k)$ detecta errores dobles ($d_{min}=3$ → $s_{MAX}=2$)

Errores tipo ráfaga son los que se producen en una parte de la trama modificando un número de bits continuo de longitud l.



Ráfaga de error ≡ B(x)

* Si la longitud de la ráfaga es de longitud n-k o menor

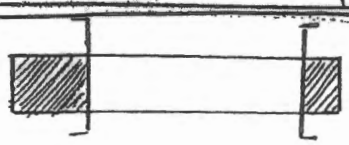
$e(x) = x^j B(x) \equiv B(x)$ con un cierto desplazamiento

Suponemos error indetectable ⇒ $e(x) \in C \Rightarrow e^{(-j)}(x) \in C$

$\Rightarrow B(x) \in C$

grado $B(x) = n-k-1 \Rightarrow \notin C$ por ser de menor grado que $g(x)$ (caso peor)

teor ⇒ Un código cíclico es capaz de detectar ráfagas de longitud n-k o menor, incluyendo las ráfagas finales ∎



- Si la longitud es $\ell = n-k+1$

⇒ Solo hay una ráfaga indetectable ⇒ $B(x) = g(x)$

La fracción de ráfagas indetectables:

$$f.r.i = \frac{1}{2^{(n-k-1)}} = 2^{-(n-k-1)}$$

Teor La fracción de ráfagas indetectables de longitud $(n-k+1)$ es $2^{-(n-k-1)}$

- Si $\ell > n-k+1$

nº total de ráfagas: $2^{\ell-2}$ (hay dos valores fijos)

nº ráfagas indetectables ⇒ son de la forma:

$$e(x) = a(x)\,g(x)$$

$\ell-1$ $\quad$ $\ell-n+k-1$ $\quad$ $n-k$

⇒ habrá tantos como polinomios de grado: $\ell-(n-k)-1$.

$$a(x) = a_0 + a_1 x + \dots + a_{\ell-n+k-1} x^{\ell-(n-k)-1}$$

$$a_0 = a_{\ell-(n-k)-1} = 1$$

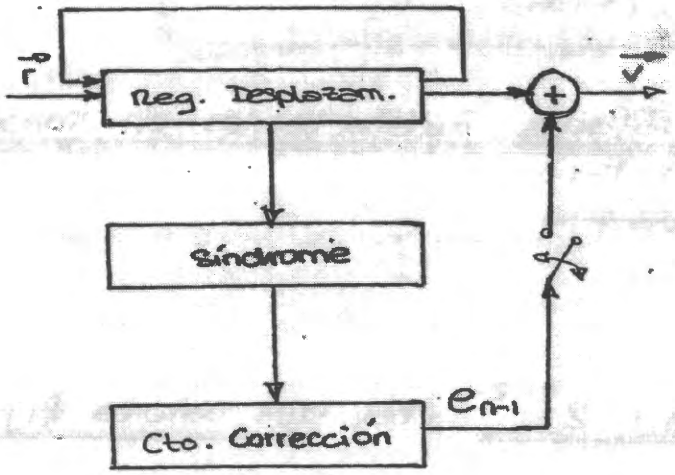
⇒ Número de polinomios: $2^{\ell-(n-k)-2}$

Por tanto:

$$f.r.i = \frac{2^{\ell-(n-k)-2}}{2^{\ell-2}} = 2^{-(n-k)}$$

9. Diseño del decodificador

- La estrategia para el diseño del decodificador de códigos cíclicos será la de obtener el error del último bit del vector recibido e ir rotándolo para corregir el vector bit a bit:



- Para realizar todas estas operaciones habrá que conseguir en cada momento el síndrome del vector de recepción que corresponda.

- Teorema:

Sea $s(x)$ el síndrome de $r(x) = r_0 + r_1 x + \dots + r_{n-1} x^{n-1}$. Si divido $x s(x)$ entre $g(x)$, el resto es el síndrome de $r^{(1)}(x)$

DEMO:

$$x r(x) = r_{n-1}(x^n + 1) + r^{(1)}(x)$$

$$\Rightarrow r^{(1)}(x) = r_{n-1}(x^n + 1) + x r(x) = c(x) g(x) + s^{(1)}(x)$$

↳ Síndrome de $r^{(1)}(x)$; luego es la rotación del síndrome

"El síndrome de rotación cíclica de un vector recibido es la rotación cíclica del síndrome del vector original módulo $g(x)+1$"

Siguientes...

$$\Rightarrow c(x)g(x) + s^{(1)}(x) = r_{n-1}g(x)h(x) + x\left[a(x)g(x) + s(x)\right]$$

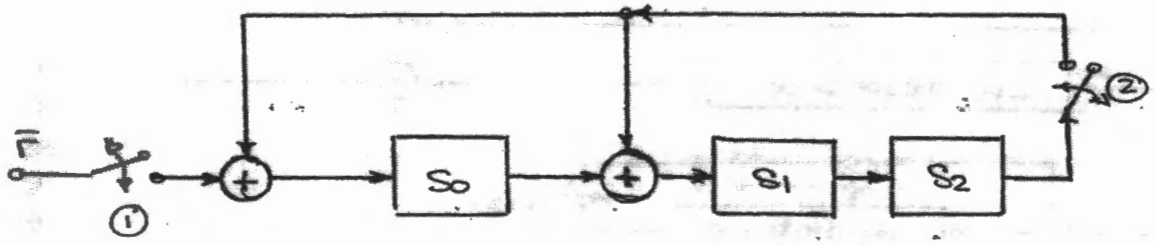
Llegamos a:

$$xs(x) = \left[c(x) + r_{n-1}h(x) + xa(x)\right]g(x) + s^{(1)}(x)$$

$$\Rightarrow \boxed{s^{(1)}(x) = \text{resto}\ \frac{xs(x)}{g(x)}}$$

** Ejemplo: $C(7,4)$ // $g(x) = 1 + x + x^3$

Para calcular el resto de $xs(x)/g(x)$ hay que abrir el interruptor ① y seguir realimentando en el cto. de calculo de síndrome.



$\bar{F} = (0010110)$

ENTRADA	REGISTROS	
0	0 0 0	
1	1 0 0	
1	1 1 0	
0	0 1 1	
1	0 1 1	
0	1 1 1	
0	1 0 1	→ Síndrome de $\bar{F}$
–	1 0 0	→ Síndrome de $\bar{F}^{(1)} \equiv S^{(1)}$
–	0 1 0	→ Síndrome de $\bar{F}^{(2)} \equiv \bar{c}^{(2)}$

① cerrado
② cerrado

① abto
② cerrado

- Otro aspecto importante:

Al rotar el vector $\bar{r}$, ya lo voy corregido por lo que el síndrome será diferente:

$r(x)$ genera $e_{n-1} = 1 \Rightarrow$ corrijo y obtengo $r_1(x)$

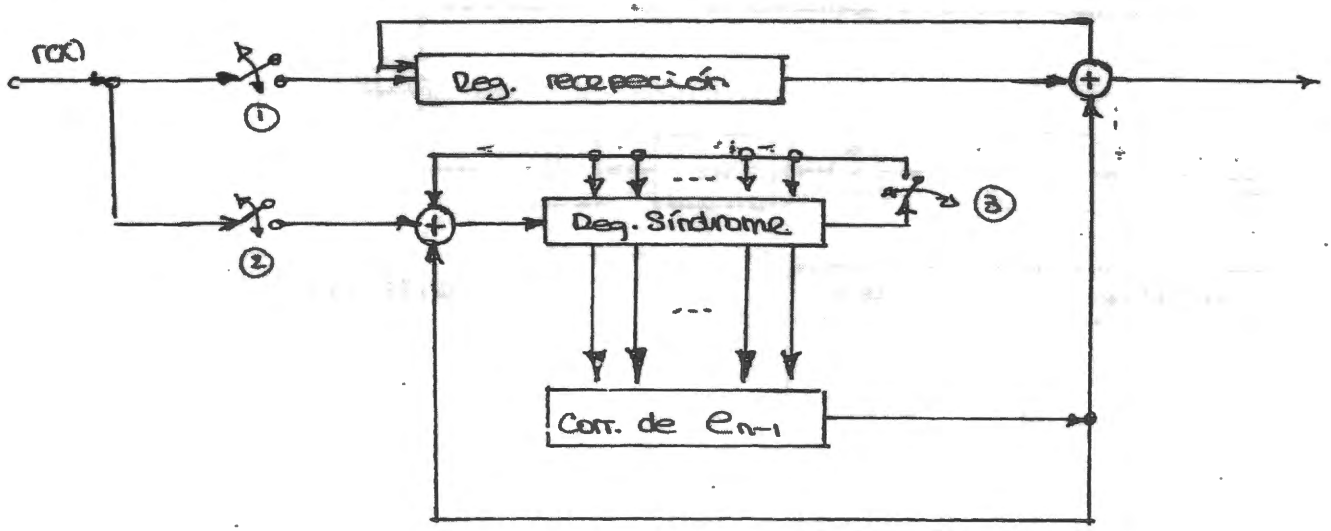
$\Rightarrow r_1(x) = r_0(x) + x^{n-1}$ $\xrightarrow{\text{rotar}}$ $r_1^{(1)}(x) = r^{(1)}(x) + 1$

Calculando los síndromes

$$\text{sindr}\left(r_1(x)\right) = s(x) + \text{sindr}\left(x^{n-1}\right)$$

$$\text{sindr}\left(r_1^{(1)}(x)\right) = \text{sindr}\left(r^{(1)}(x)\right) + \text{sindr}(1) = x\,s(x) + \text{sindr}(1)$$

$\Rightarrow$ Tengo que sumar el valor del error e_{n-1} para que el síndrome se calcule sin problemas:



* Este decodificador es conocido como el decodificador de Meggit

- ¿Cómo se implementa el circuito de corrección?

* Para el circuito de corrección debemos utilizar la fórmula de cálculo del síndrome:

$$s(x) = \text{resto}\ \frac{x^i}{g(x)} \quad \text{(peso 1)}$$

* Podemos obtener una tabla de síndromes en función de la posición del error y a partir de ahí, determinar la función lógica deseada.

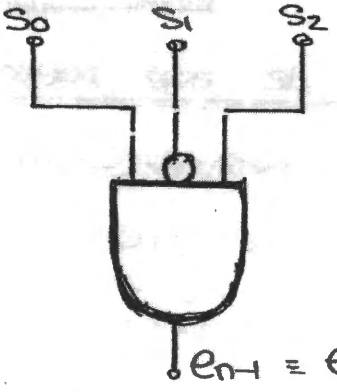
** Ejemplo: $C(7,4)$, $g(x) = 1 + x + x^3$

$d_{min} = 3 \Rightarrow t_{max} = 1 \Rightarrow$ errores de peso 1

$$s(x) = \text{resto}\ \frac{x^i}{g(x)}$$

$e(x)$	$s(x)$
x^6	$1+x^2$
x^5	$1+x+x^2$
x^4	$x+x^2$
x^3	$1+x$
x^2	x^2
x	x
1	1

→ Este es el síndrome que buscamos:



* Será "1" cuando se detecte un error en e_6.

10 Códigos de Hamming cíclicos

-Teorema: Un código de Hamming cíclico de longitud $n = 2^m - 1$ con $m \geq 3$ es generado por un polinomio primitivo $p(x)$ de grado m.

$p(x)$ { irreductible
factor de $(x^{2^m-1}+1) = (x^n+1)$
no factor de (x^t+1) $t < 2^m-1$ $(t<n)$

- Se sigue cumpliendo que: $d_{min} = 3 \Rightarrow s_{max} = 2 \Rightarrow t_{max} = 1$

- Si tengo $g(x) = p(x)$ con $d_{min} = 3$, para aumentar en 1 la distancia mínima:

{ $g(x) = p(x) \Rightarrow$ Detecta dobles $(1,2)$
$g(x) = (1+x) \Rightarrow$ Detecta impares $(1,3,5,\ldots)$

* Si hacemos $g(x) = p(x)(1+x) \Rightarrow$ Detecta $(1,2,3,5,\ldots)$
$\Rightarrow \boxed{d_{min} = 4}$

* Dimensiones de este nuevo código:

$g(x) = p(x) \Rightarrow G(n,k)$

$g'(x) = \underbrace{(1+x)\,p(x)}_{n-k+1} \Rightarrow$ { $G(n+1,k) \rightarrow$ factor de $(x^{n+1}+1)$
$G(n,k-1) \rightarrow$ factor de (x^n+1)

$\Rightarrow$ El código generado será: $\boxed{G(n,k-1)}$

(el otro no se verifica casi nunca)

1. Códigos recortados

- Supongamos que queremos codificar palabras de 20 bits. El código Hamming que más se aproxima es un $C_1(31, 26)$ pero no me vale porque serían 26 bits por palabra y no 20 como queríamos.
- Los códigos recortados consisten en eliminar bits que no aportan información en la codificación que queremos

↳ Ejemplo $C_1(10,7)$

* Las palabras fuente serán de la forma:

0000000	0000000 b_0 b_1 b_2
0000001	0000001 b_0 b_1 b_2
⋮	⋮ Redundancia
2^7 palabras	2^{10} palabras

* Si queremos generar un código de 4 bits de fuente, necesitaremos 2^4 palabras, lo que se consigue solo cogiendo las palabras de la forma: 000xxxx y eliminado los 3 primeros bits. Igualmente, se eliminan los 3 primeros bits de las palabras código, obteniendo un código $C_1(7,4)$

- En general, si tenemos un código $C_1(n,k)$ podemos pasar a un código $C_1(n-l, k-l)$, donde l es el número de bits eliminado.

Propiedades

- Si $C(n,k)$ es cíclico $\Rightarrow$ el código recortado $C(n-1, k-1)$ es lineal pero no es cíclico.
- $C(n-l, k-l)$ tiene la misma distancia mínima, capacidad de detección y corrección que el código original.

$$d_{min}\big|_C = d_{min}\big|_{C_n}$$

$$\Downarrow$$

$$S_{max}\big|_C = S_{max}\big|_{C_n}$$

$$\Downarrow$$

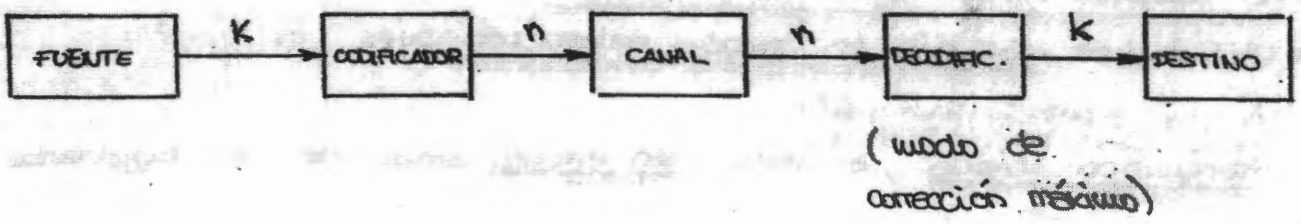
$$t_{max}\big|_C = t_{max}\big|_{C_n}.$$

Tema 8. Técnicas ARQ

1. Introducción

- Existen dos técnicas principales para controlar los errores de transmisión. La primera es la FEC (Forward Error Correction) en la que el receptor trata de corregir el error recibido, siendo la probabilidad de error la probabilidad de no corregir.

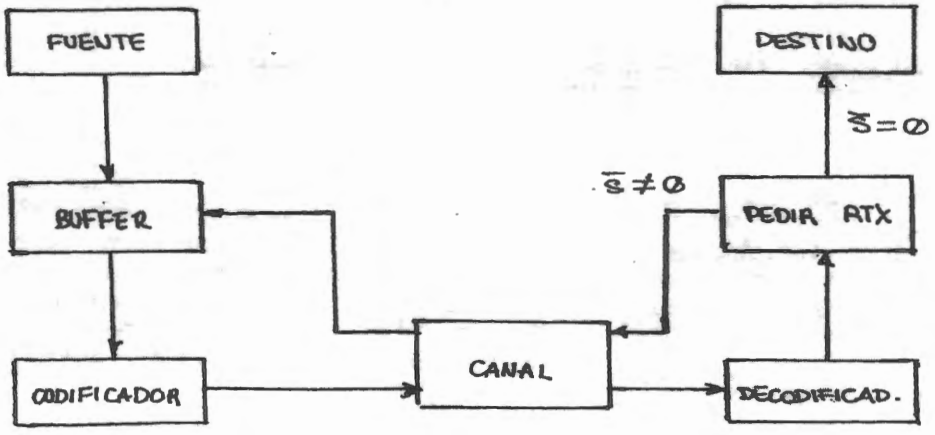
(Código en modo de corrección pura)



(modo de corrección máximo)

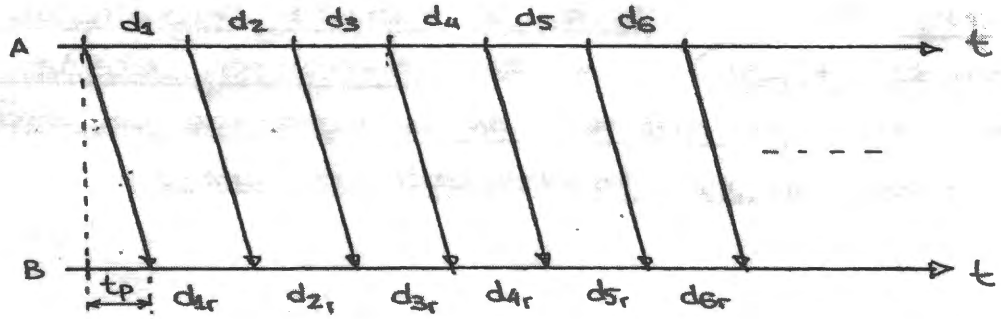
- La otra alternativa es el conjunto de técnicas denominadas ARQ (Automatic Repeat Request) en la que necesitamos detectar el máximo número de errores posibles. Cuando recibamos un error pediremos al transmisor una retransmisión del mismo.

(Código en modo de detección pura)



2- Situación de partida. Tipos de ARQ

- Supongamos dos elementos, A y B, entre los que se quiere establecer una comunicación.
- Para ello suponemos además que:
  - * A quiere mandar muchos datos a B
  - * A siempre tiene algo que enviar
  - * A va a segmentar los datos en tramas y les añade la redundancia (cabecera o pie de trama).
  - * B siempre está listo para recibir.
  - * Utilizamos códigos en modo detección para. [illegible]
  - Ni hay errores en NAK y ACK
  - Retransmitimos todos los errores.
- La transmisión básica, es decir, sin errores sería de la siguiente forma:

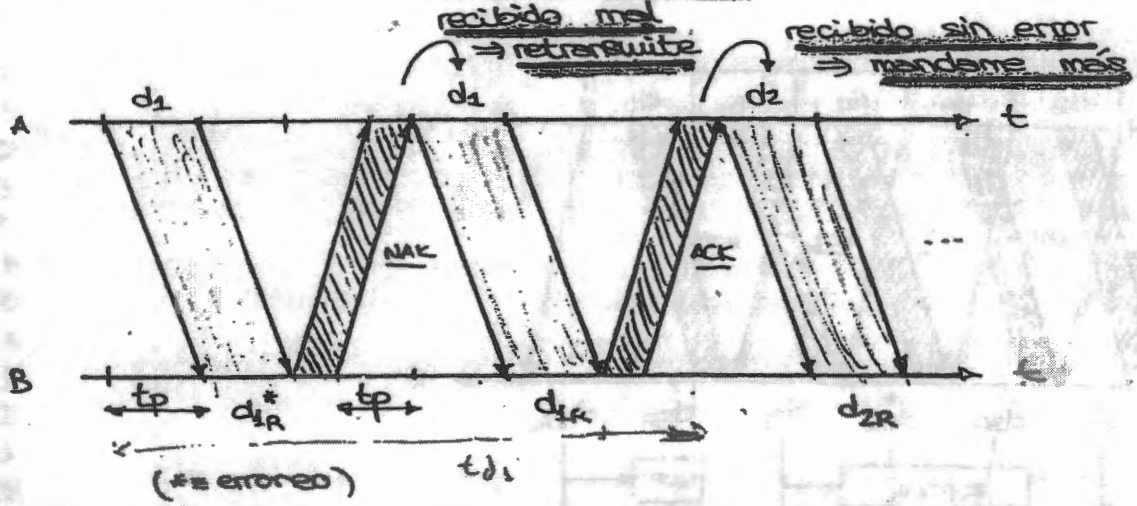


Hay un retardo de propagación tp constante

- Si se producen errores en el canal el receptor debe informar al transmisor de ese incidente y pedir la retransmisión del mismo. Así mismo si el mensaje ha llegado correctamente se debe informar también al transmisor. Para ello existen tramas de control que el receptor envía al transmisor:

+ { ACK = Acknowledge
  { NAK = Not acknowledge

- Por tanto, para un canal semidúplex:



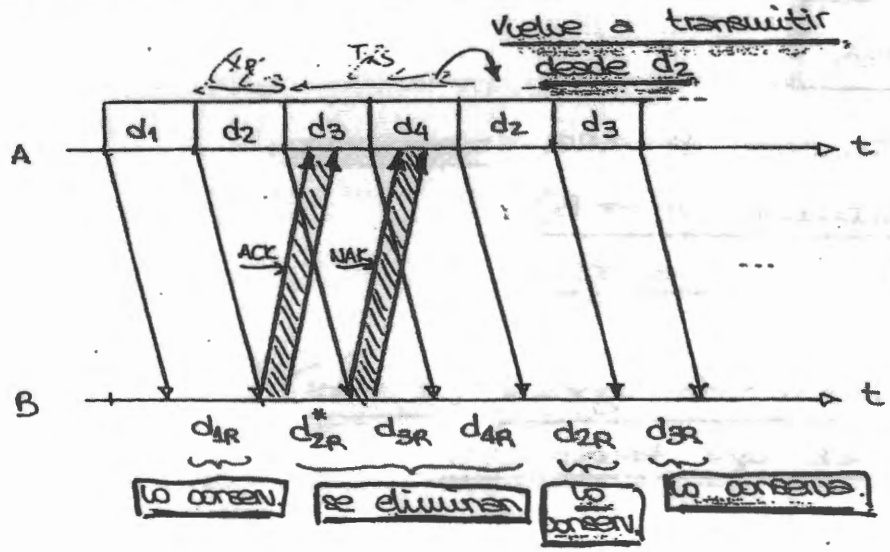
* Este método es el conocido como **ARQ de parada y espera.**

- Si el canal es full-dúplex podemos seguir enviando datos aun no habiendo recibido la confirmación del receptor ⇒ **Técnicas de ARQ con envío continuo.** Distinguimos dos casos:

* **ARQ de rechazo retroactivo.** (simple)

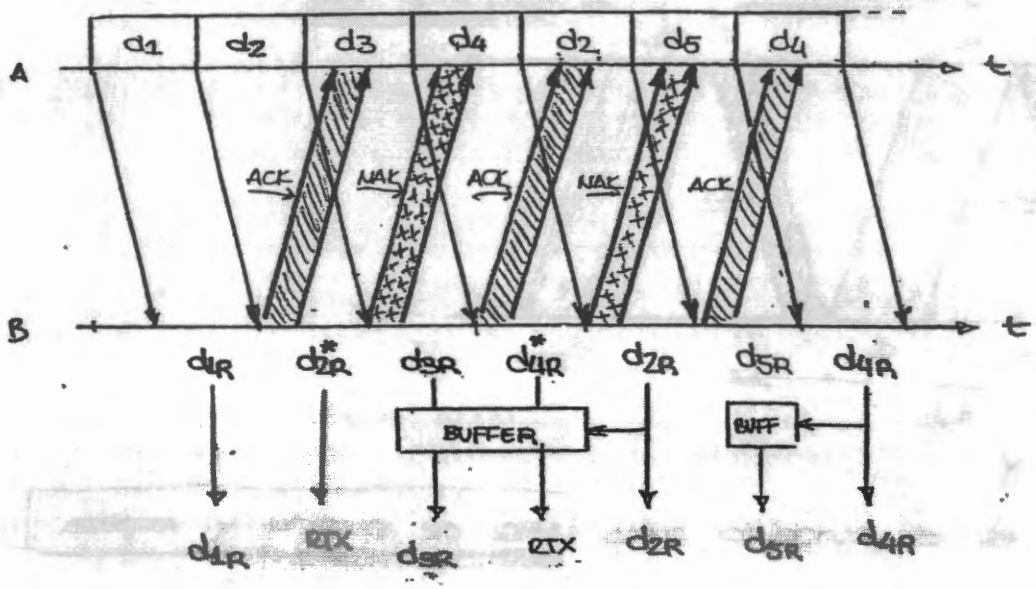
$$x_p = \frac{N}{R} = \frac{K+m}{R}$$

= tiempo de ocupación



⇒ Al llegar una petición NAK se vuelve a transmitir a partir de ahí eliminando los datos recibidos después del error (sean válidos o no)

* ARQ de rechazo selectivo.



En este caso sólo se pide la retransmisión de un solo paquete y se guardan los siguientes en un buffer hasta que se reciba bien el paquete anterior. Así aseguramos que los paquetes se reciben en orden.

3- Analisis de prestaciones

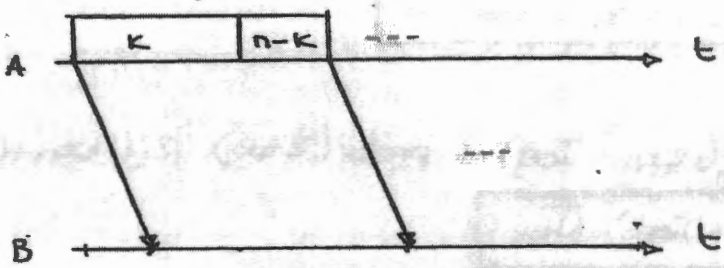
- Para analizar las prestaciones de ARQ suponderemos:
  - * Tráfico unidireccional (A → B)
  - * Tramas de control sin errores
  - * $P_{ND} = 0$
  - * Bloques de información iguales (k bits)
  - * Siempre hay datos que transmitir.

- Definimos la **cadencia eficaz** como la relación entre el número de bits de información por trama y el (tiempo medio de ocupación del canal).

$$C_{ef} = \frac{k}{T_{oc}}$$

* **Técnicas FEC** (todo lo que llega se contesta)

Corrección pura



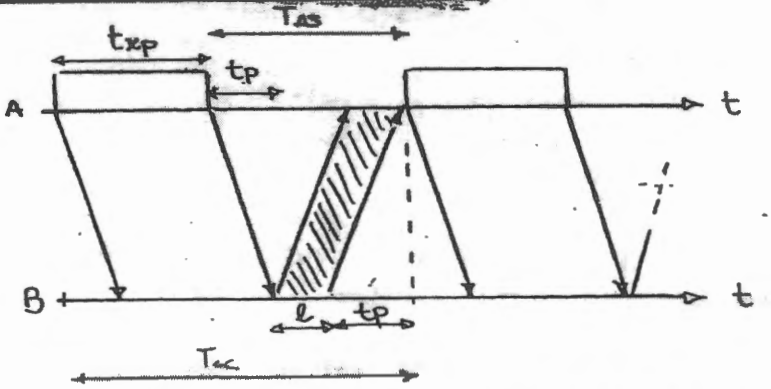
$$T_{oc} = \frac{n}{R}\left(\frac{bit}{bit/s}\right) = [s]$$

$$C_{ef} = \frac{k}{n/R} = \frac{k}{n} \cdot R$$

R ≡ Régimen binario del canal.

- Rendimiento: $\rho = \frac{C_{ef}}{R} = \frac{k}{n}$

* **ARQ de parada y espera**



$T_{ocupacion}$ = Variable aleatoria ⇒ Busco su media.

$$C_{ef} = \frac{k}{T_{oc}}$$

con T_{AS} = tiempo de asentimiento $= 2t_p + \frac{\ell}{R}$

- Toc es una variable aleatoria que dependerá del número de transmisiones necesarias

Nº transmisiones	Toc
1	$t_{xp} + T_{as}$
2	$2(t_{xp} + T_{as})$
3	$3(t_{xp} + T_{as})$
⋮	⋮
i	$i(t_{xp} + T_{as})$

$$\Rightarrow \boxed{\overline{T_{oc}} = \text{prob}(1\,tx)\cdot(t_{xp}+T_{as}) + \text{prob}(2\,tx)\cdot 2(t_{xp}+T_{as}) + \ldots = \boxed{(t_{xp}+T_{as})\,\overline{N_{tx}}}}$$

con $\overline{N_{tx}}$ = número medio de transmisiones.

- La probabilidad de retransmisión (prob. de error de bloque) será:

$$\boxed{P_{Eb} \equiv P_{RTX}} = 1 - P_{NO\ ERR.} - P_{ERR.\ INDETEC.} =$$

$$= 1 - (1-p)^n - \sum_{i=d_{min}}^{n} A_i\, p^i (1-p)^{n-i} \simeq$$

$$\simeq 1 - (1-p)^n - \sum_{i=d_{min}}^{n} \binom{n}{i} p^i (1-p)^{n-i} \simeq$$

$$\simeq \sum_{i=1}^{d_{min}-1} \boxed{\binom{n}{i} p^i (1-p)^{n-i}} = \sum_{i=t+1}^{s} \binom{n}{i} p^i (1-p)^{n-i}$$

(Si suponemos que siempre detectamos errores, entonces: $P_{eb} = P_{RTX} = 1-(1-p)^n \simeq pn$)

• Por tanto:

Ntx	prob(Ntx)
1	$1-P_{Eb}$
2	$P_{Eb}(1-P_{Eb})$
3	$P_{Eb}^2(1-P_{Eb})$
⋮	⋮

$$\Rightarrow \overline{N_{tx}} = \sum_{i=1}^{\infty} i\, P_{Eb}^{\,i-1}(1-P_{Eb}) =$$

$$= (1-P_{Eb}) \sum_{i=1}^{\infty} i\, P_{Eb}^{\,i-1} = (1-P_{Eb}) \frac{1}{(1-P_{Eb})^2} = \frac{1}{1-P_{Eb}}$$

n.º m. medio de retransmisiones

$$\Rightarrow C_{ef} = \frac{K}{T_{cc}} = \frac{K}{\left(\frac{K+m}{R} + T_{as}\right)\left(\frac{1}{1-P_{Eb}}\right)} =$$

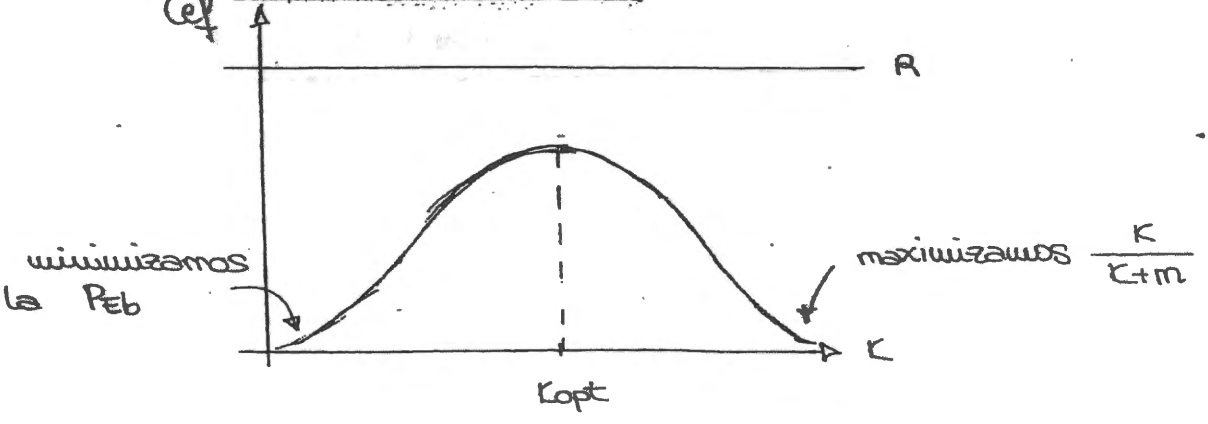
$$= (1-P_{Eb}) \frac{K}{K+m+T_{as}R} R$$

Con m = bits de redundancia

⇒ Viendo la expresión de C_{ef} deducimos que existe una K_{optima} para maximizar C_{ef}

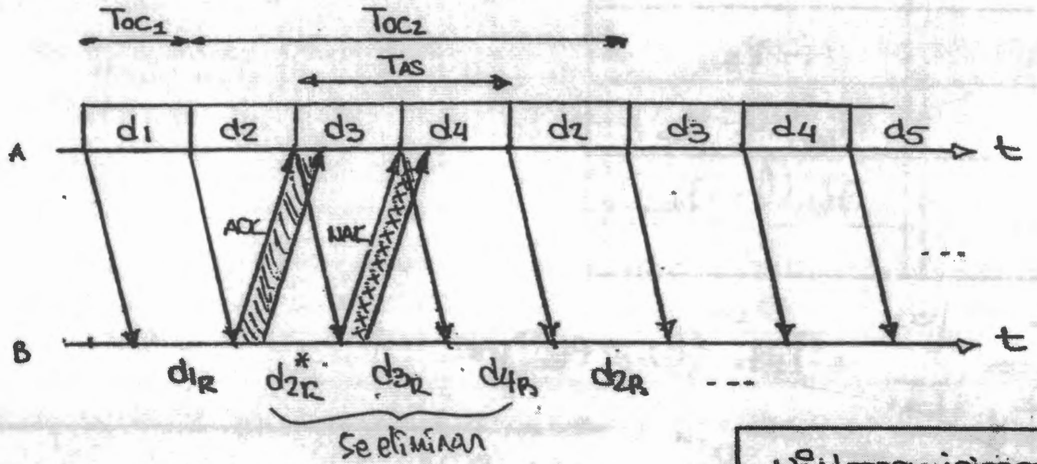
$$\Rightarrow K_{opt} \cong \sqrt{\frac{m+R\,T_{as}}{P}}$$

P: Prob. error en bit.



4.- Analisis de prestaciones : ARQ de envio continuo

- ARQ de rechazo simple (Rechazo [illegible])



$$* \; C_{ef} = \frac{k}{\overline{T_{oc}}}$$

Nº transmisiones	Toc
1	t_{xp}
2	$2t_{xp} + T_{as}$
3	$3t_{xp} + 2T_{as}$
⋮	⋮

$$\boxed{\overline{T_{oc}} = \overline{N_{tx}}\, t_{xp} + (\overline{N_{tx}} - 1)\, T_{as}}$$

con $\boxed{\overline{N_{tx}} = \frac{1}{(1-P_{Eb})}}$

$$\Rightarrow \boxed{C_{ef} = \frac{k}{\overline{T_{oc}}}} = \frac{k}{\left[\frac{1}{1-P_{Eb}} \cdot \frac{k+m}{R} + \left[\frac{1}{1-P_{Eb}} - 1\right] T_{as}\right]} =$$

$$= \boxed{(1-P_{Eb}) \frac{kR}{[k+m+T_{as} R \cdot P_{Eb}]}}$$

Problemas:

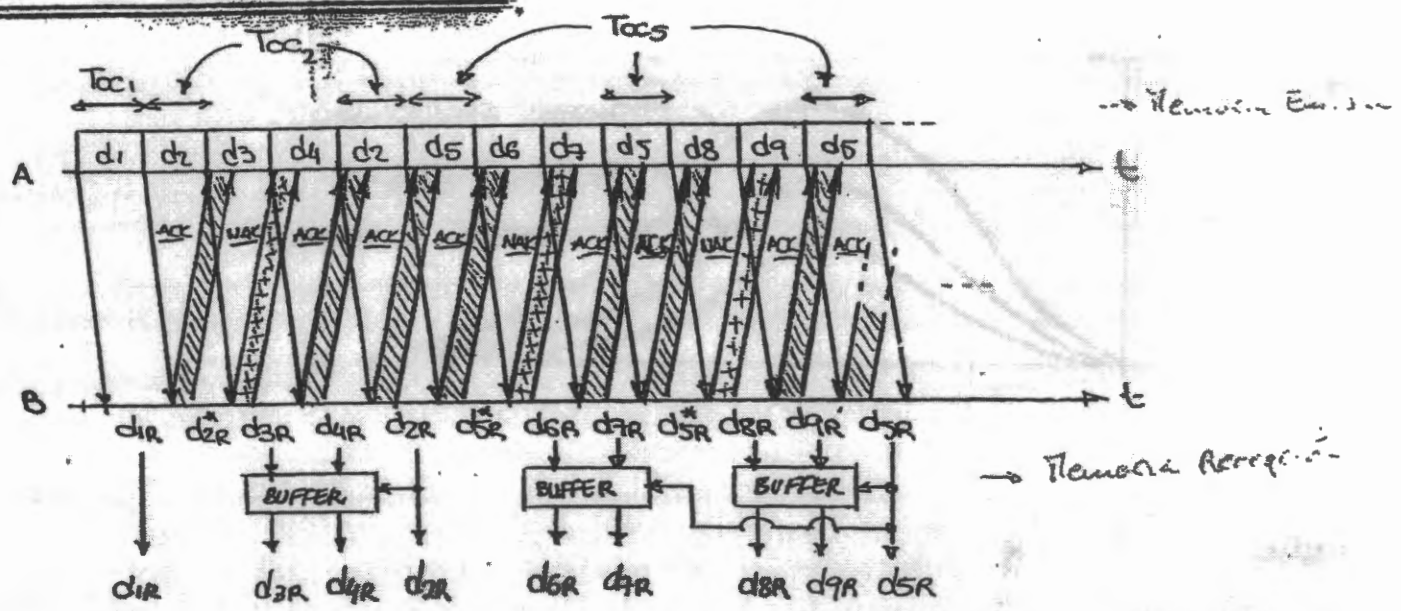
-> Memoria en emisor [illegible]

-> [illegible] tramas y respuestas

· [illegible] protocolo de nivel inferior

· [illegible] k bits para numeración tramas

- ARQ de rechazo selectivo



* $Cef = \frac{K}{\overline{Toc}}$

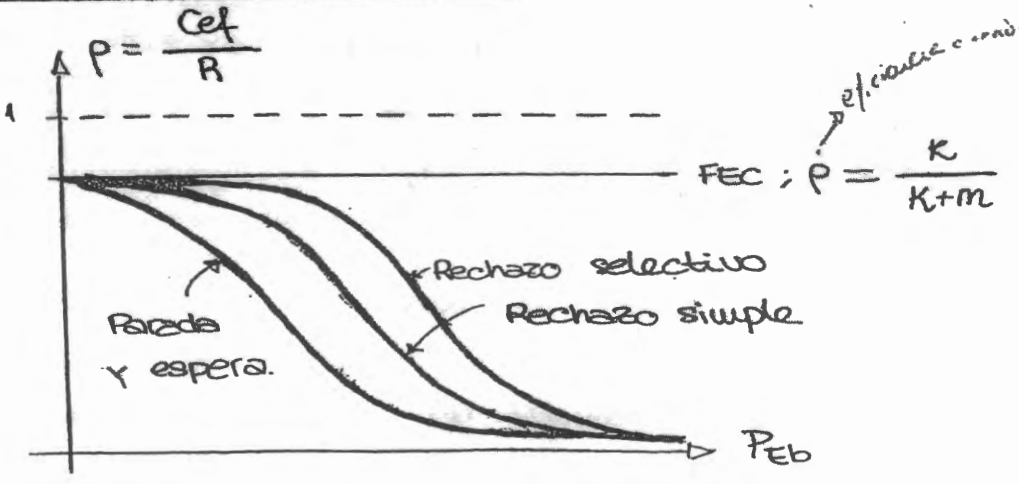
Nº transmisiones	Toc
1	txp
2	2txp
3	3txp
⋮	⋮

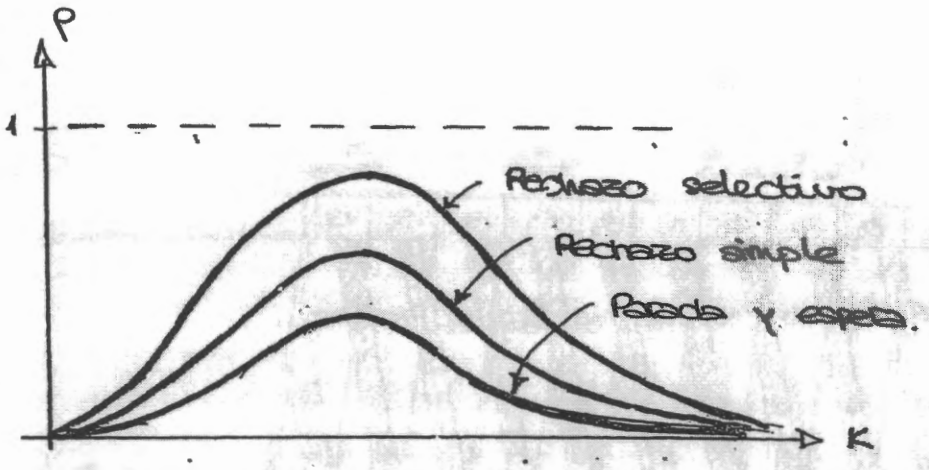
$\Rightarrow \boxed{\overline{Toc} = \overline{Ntx} \cdot txp}$

$\boxed{\overline{Ntx} = \frac{1}{(1-P_{Eb})}}$

$\Rightarrow \boxed{Cef = \frac{K}{\overline{Ntx} \cdot txp} = (1-P_{Eb})\frac{K}{K+m} R}$

5- Comparación de técnicas ARQ





** Ejemplo

K= 1000 bits

G(1048, 1000) ⇒ m= 48

R= 1200 bps

$p = 5 \cdot 10^{-6}$

tp = 100 ms

Las tramas de control son de m bits ≡ 48 bits

$$\Rightarrow Tes = 2\,tp + \frac{m}{R} = 2 \cdot 0.1 + \frac{48}{1200} = 0.24\ s$$

Haciendo Cef para todas las técnicas ARQ:

	Parada y espera	Rechazo simple	Rechazo selectivo
Cef	893 bps	1137 bps	1139 bps
$\rho = \frac{Cef}{R}$	74%	94.7%	94.9%

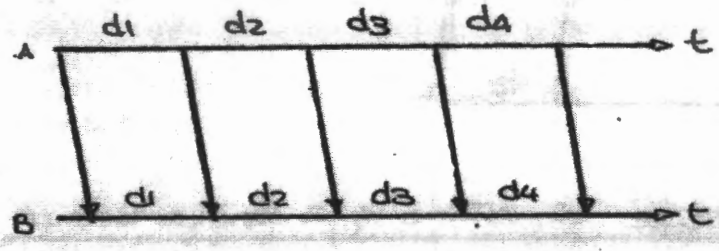
FEC: 1145 bps, 95,48 %

⇒ No merece la pena el rechazo selectivo ya que implica una complicación excesiva en el hardware del receptor; por tanto la mejor opción será rechazo simple.

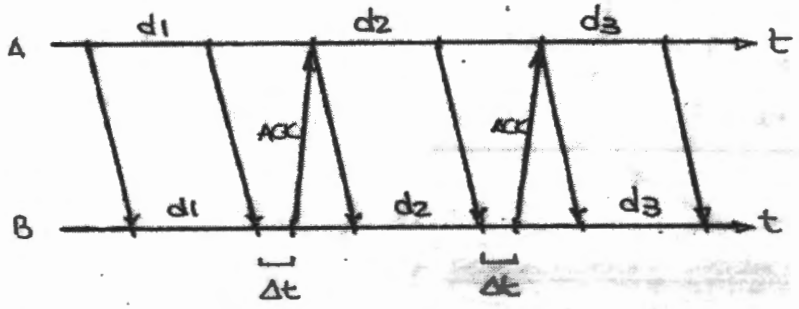
5. Mínimo protocolo corrector de errores

- Caso ideal

  * No existen errores ; capacidad de recepción ilimitada



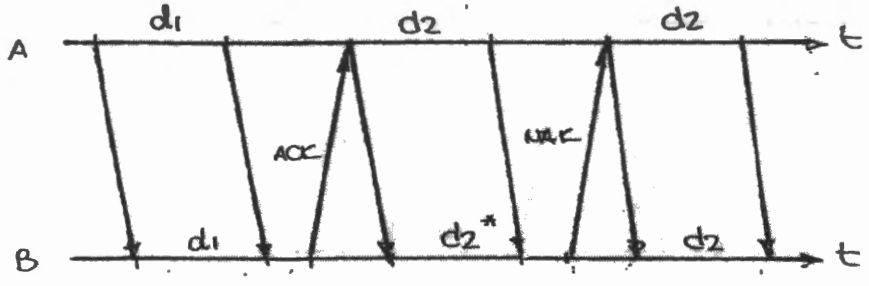
- Ahora suponemos que el receptor necesita un Δt para procesar la información ⇒ El emisor necesita saber cuando el receptor está listo ⇒ Técnicas de control de flujo.



⇒ ACK de control (no transporta datos) ⇒ Retardo pequeño

- Si ahora el canal mete errores

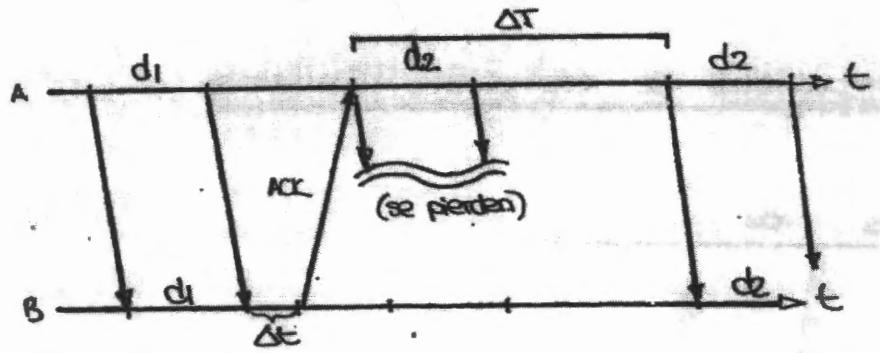
⇒ Trama de control { ACK ⇒ recepción positiva
NAK ⇒ recepción negativa ⇒ solicitud de retransmisión



ARQ parada y espera.

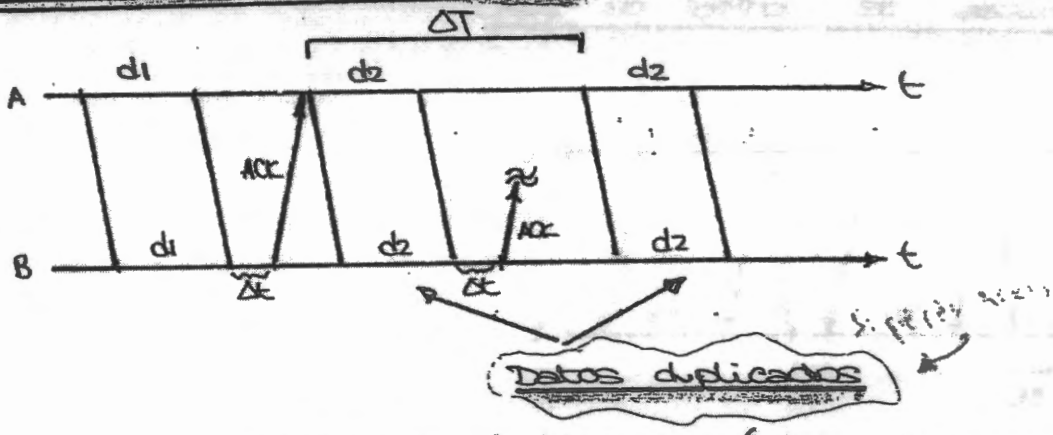
Puede haber pérdida de datos

⇒ Temporizador en emisor



Si el emisor no recibe el ACK o NAK en un tiempo ΔT, entonces reenvía el mensaje.

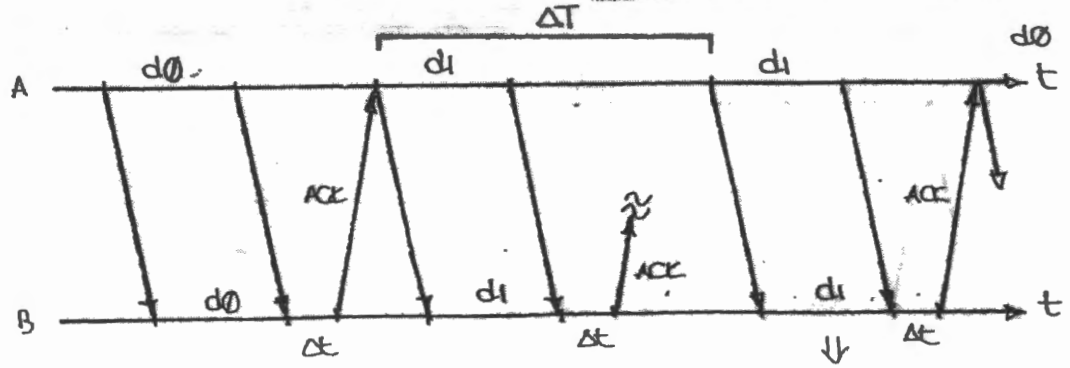
Puede perderse el ACK / NAK



⇒ Necesitamos introducir números de secuencia en los datos. Los números de secuencia:

* Viajan en la cabecera de la trama
* Lo más pequeño posible ⇒ BIT DE PARIDAD

(Sólo queremos saber si es distinto del anterior)

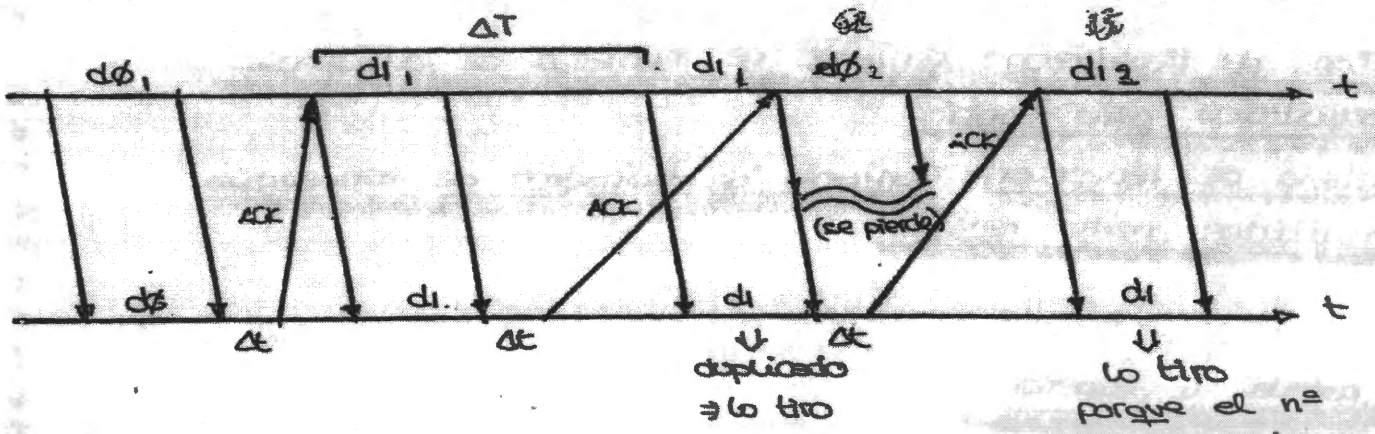


⇓

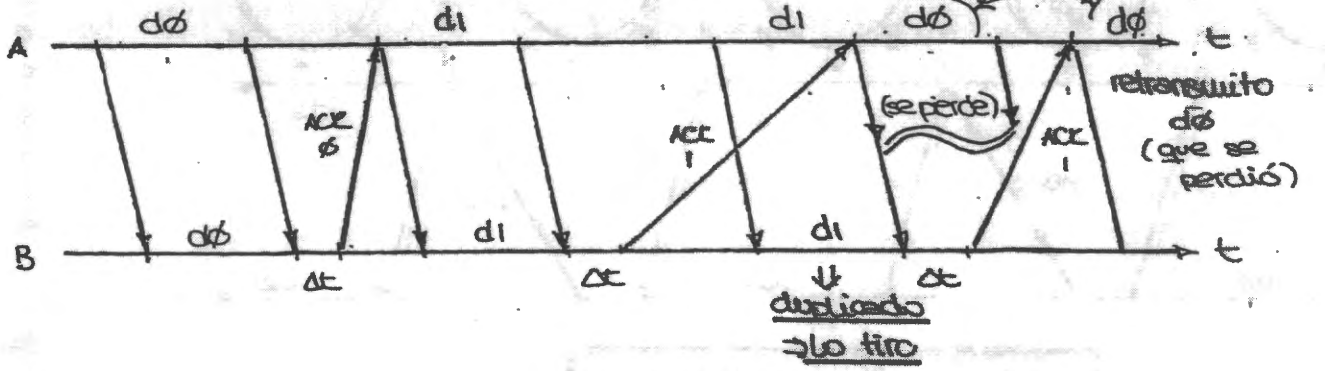
Lo tiro porque ya lo tengo

d_{ij} → i = bit de paridad
d_{ij} → j = núm. mensaje

- Retardos de propagación variable



⇒ Hemos perdido datos | por confundir los ACK's | ⇒ Hay que introducir números de secuencia en los ACK's.



⇒ Mínimo protocolo corrector de errores : BIT ALTERNANTE.

5. Técnicas de ventana deslizante

- Se utiliza para técnicas ARQ con envío continuo ⇒ Tenemos memoria en emisor y receptor. Necesitamos números de secuencia:

$$nº\ secuencia > 1\ bit$$

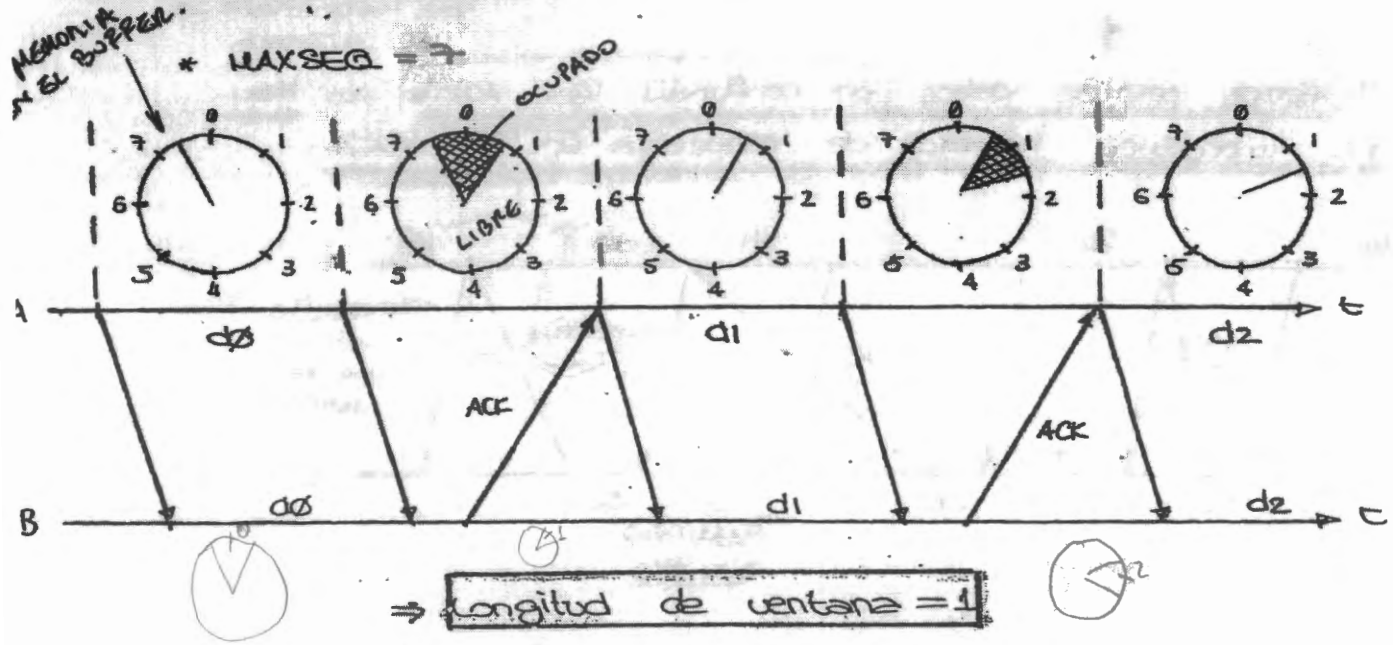
- Número de secuencias máximo para un número de secuencia de n bits :

$$MAXSEQ = 2^n - 1$$

- Se define una ventana como un buffer de memoria en el emisor y/o en el receptor.

  - Ventana de transmisión: Conjunto de números de secuencia permitidos para enviar.
  - Ventana de recepción: Conjunto de números de secuencia permitidos para recibir.

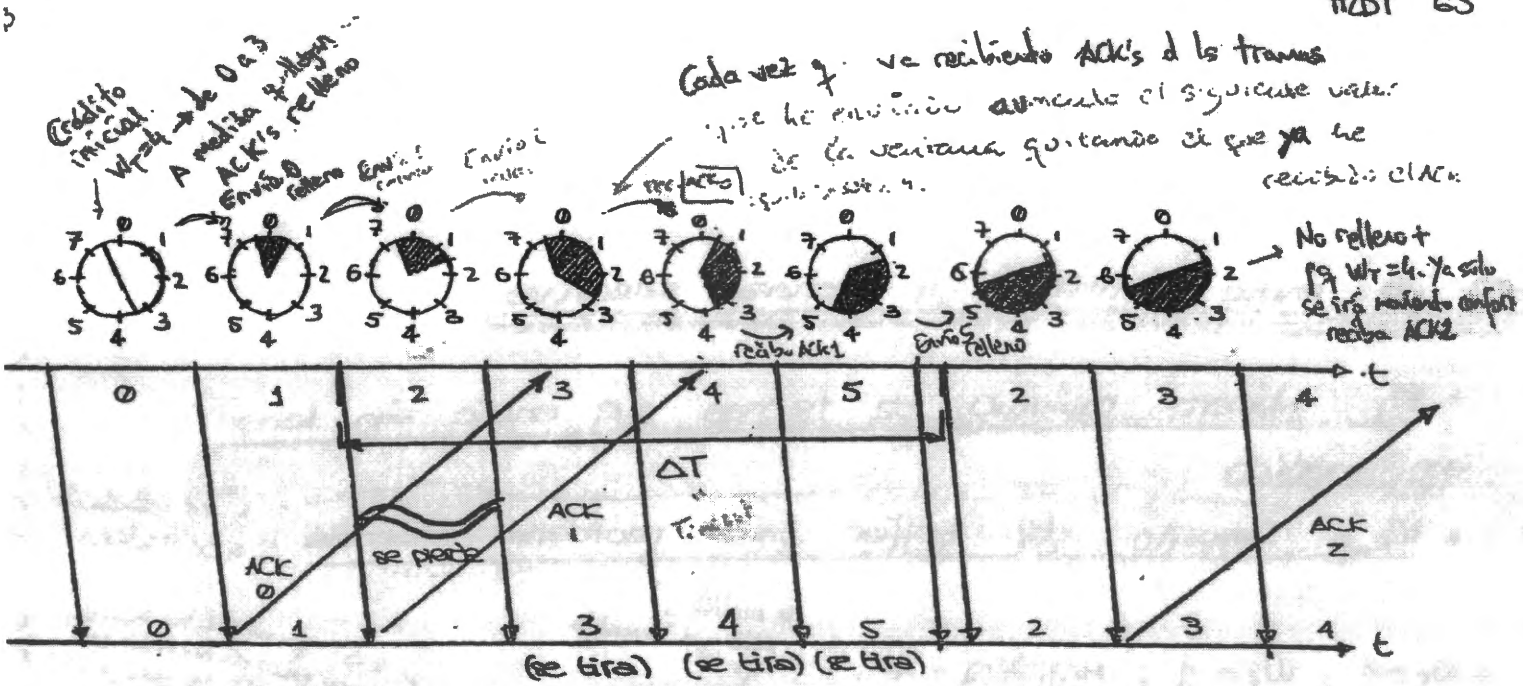
- ARQ parada y espera



⇒ Longitud de ventana = 1

- ARQ de envío continuo y rechazo simple

  * W_T = Tamaño máximo de la ventana de transmisión
  * W_R = Tamaño máximo de la ventana de recepción



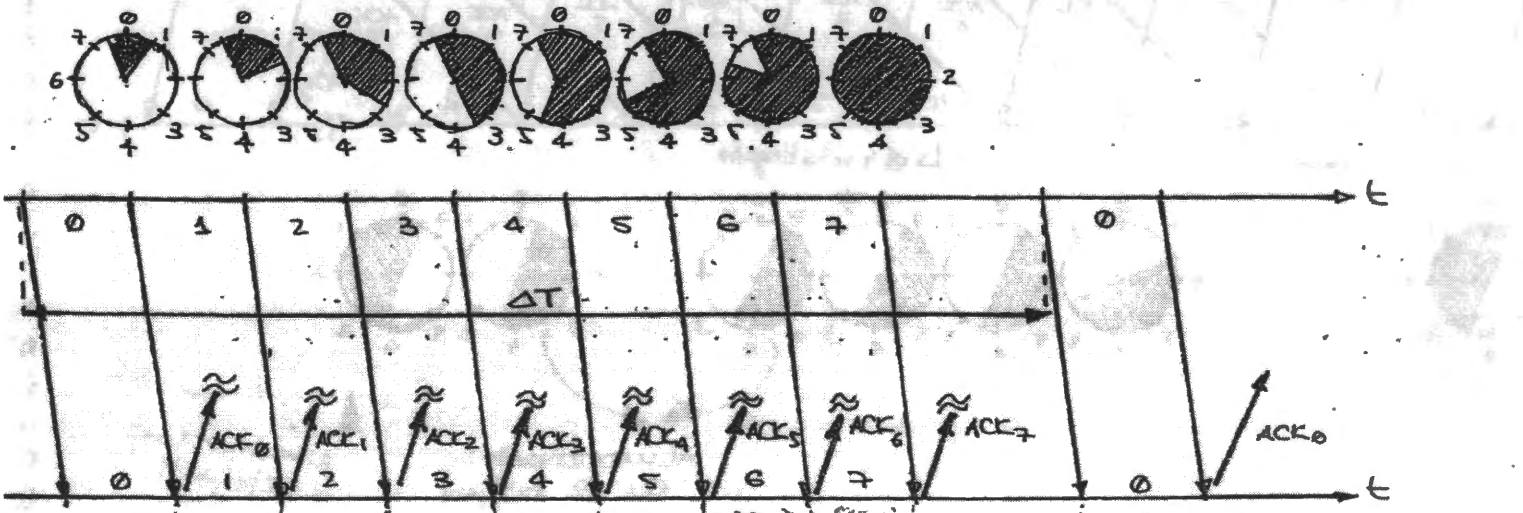
$\Rightarrow$ MAXSEQ = 7 ; $W_T = 4$.

* Hasta que no reciba el ACK_2 el sistema permanecerá bloqueado $\Rightarrow$ Cuando reciba ACK_2 envío el dato 6

Rechazo simple!!

* Si utilizamos ahora: MAXSEQ = 7 ; $W_T = 8$:

W_T = MAXSEQ + 1



espero Φ $\Rightarrow$ Recibiría datos duplicados pero no se tiran porque a la aplicación no se lo parecen $\Rightarrow$ Error a la aplicación

$\Rightarrow$ El tamaño de la ventana debe ser del modo:

$$W_T \leq MAXSEQ$$

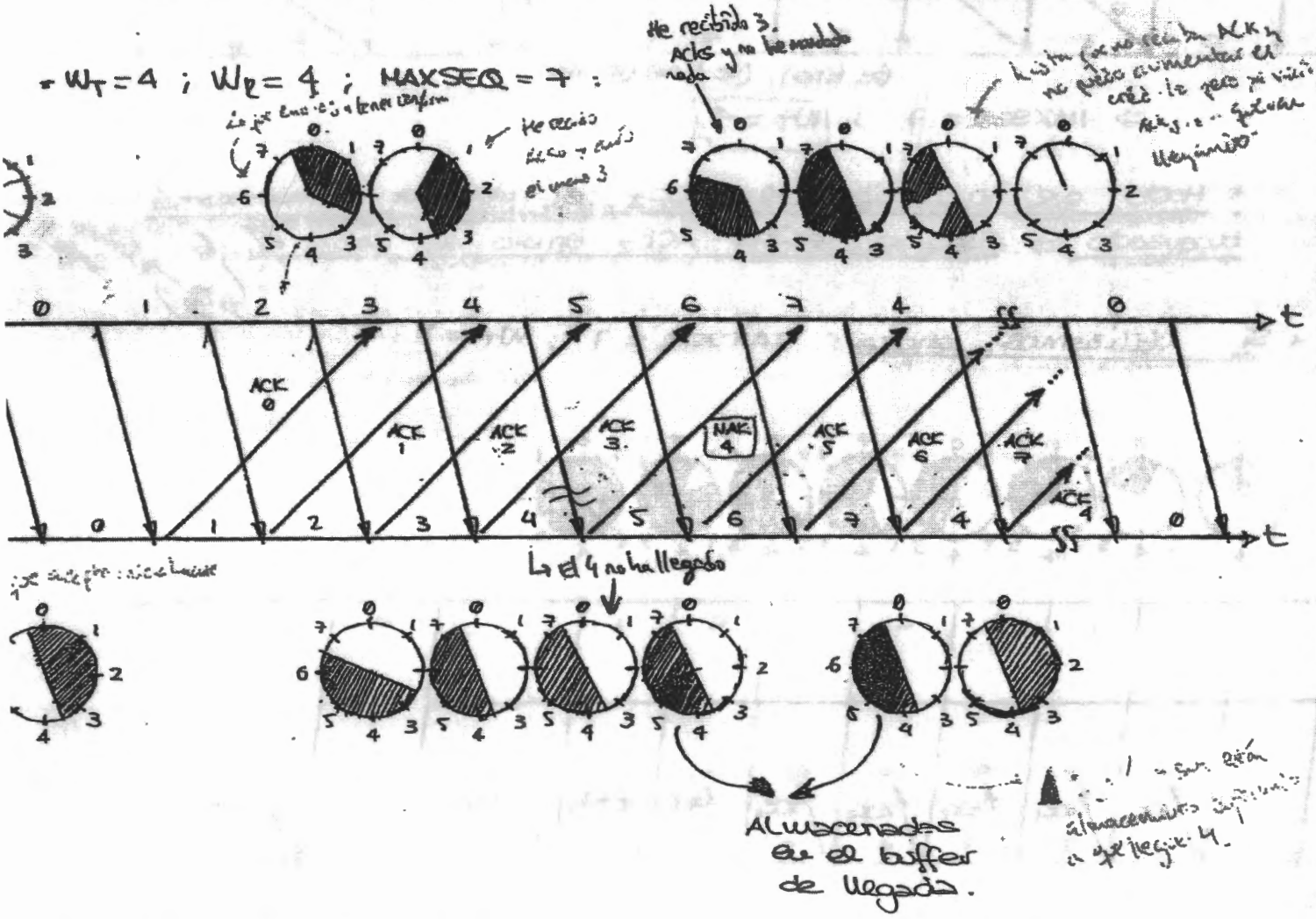
En stop-and-wait $W_T = 1$

Solo espero 1 paquete

ARQ de envío continuo y rechazo selectivo

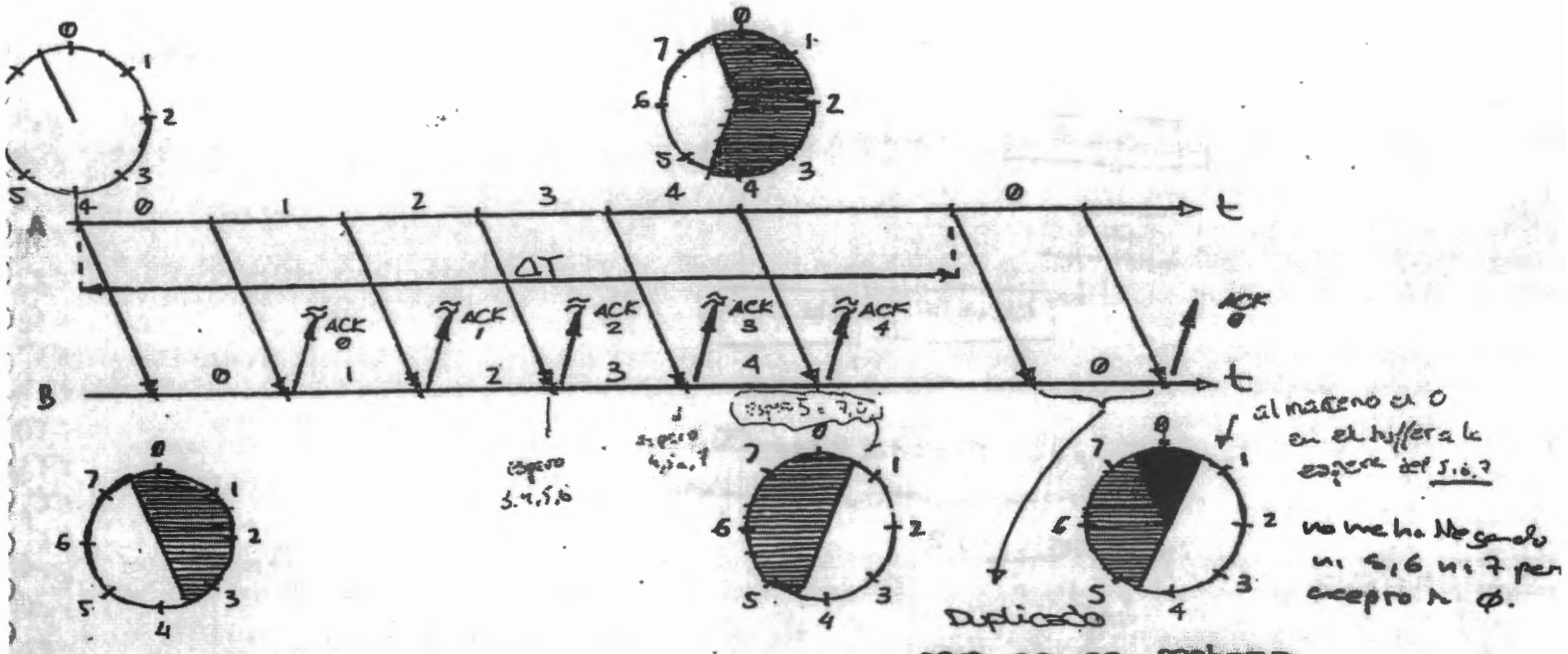
* W_T = Número máximo de tramas que envío sin tener confirmación
* W_R = Tamaño del buffer para reordenar tramas | aceptables

* $W_T = 4$; $W_R = 4$; MAXSEQ = 7 :



* Si suponemos que $W_T = 5$ y $W_R = 4$, con MAXSEQ = 7, vemos que suponiendo que se pierden todos los ACK's (caso peor) cuando el temporizador del emisor se activa y retransmitimos el mensaje 0 el receptor no lo desecha ⇒ Mensaje duplicado.

ΔT ≡ Timeout.



Por tanto, se debe cumplir que:

$$W_T + W_R \leq MAXSEQ + 1$$

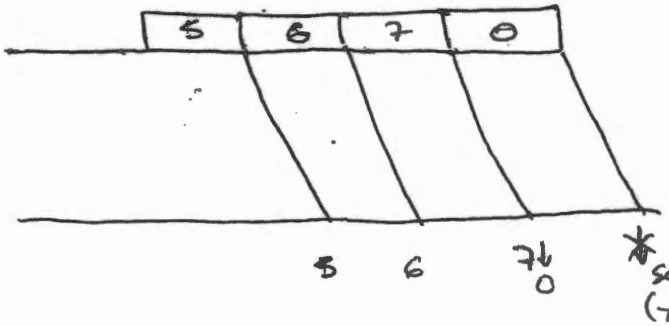
Normalmente

$W_T = W_R$

$MAXSEQ = 2^n - 1$

⇒ nº frames numerados = MAXSEQ + 1

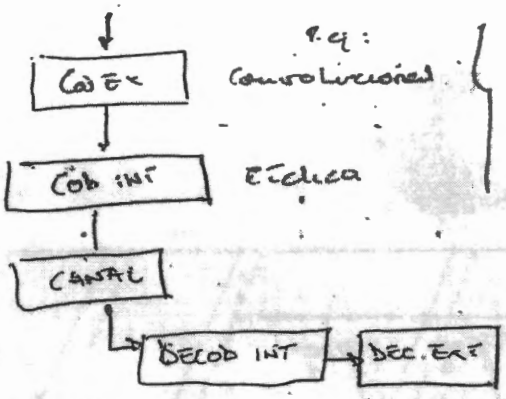
Si se ha perdido el rechazo simple en el que $W_R = 3$.



= 1000
= 48
= 1200
= $5 \cdot 10^{-6}$
= 300 ms

TÉCNICAS HÍBRIDAS

Concatenación de códigos



p.ej: convolucionales

Eclica

¿Las prestaciones de la unión son mayores que por separado?

- FEC-ARQ rígido (tipo I)

Código $\mathcal{C}_{unión} = ?$

$t = 5$

$s = 15$

- FEC-ARQ adaptativo

1) Envío con código de alta FEC

2) RTx en FEC puro { mismo código / otro distinto