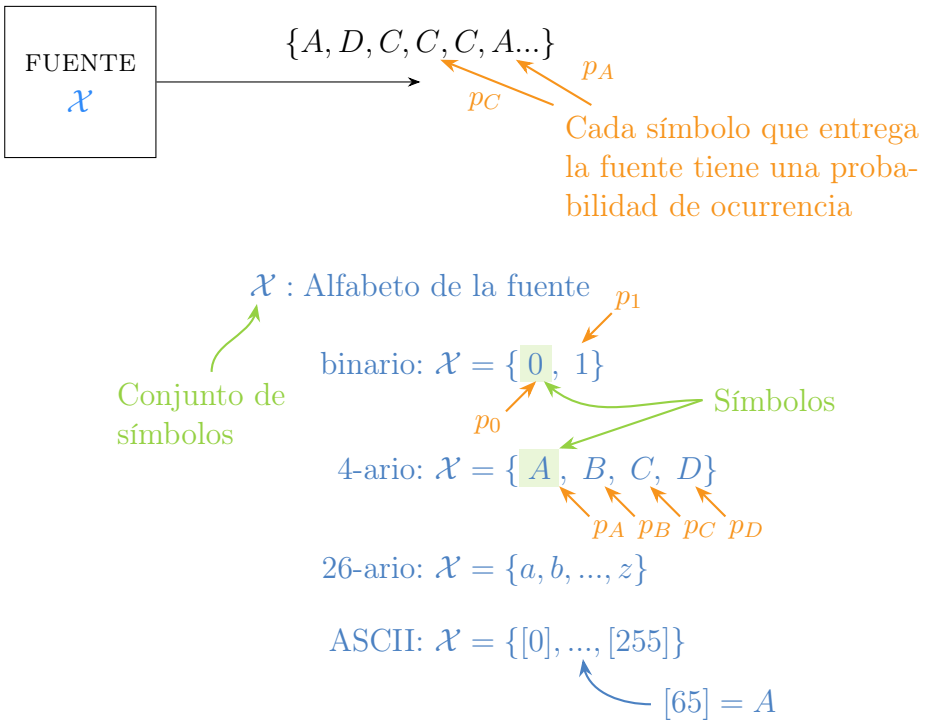
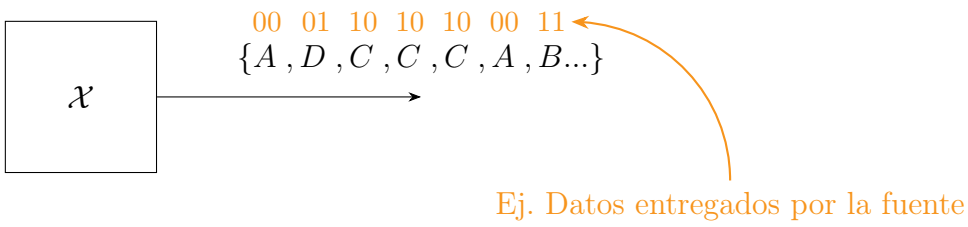
Módulo 16 — Teoría de información

Cápsula 16.1 — Fuentes de información



Cápsula 16.2 — Datos



Intuición: Símbolos más probables contienen menos INFORMACIÓN que símbolo menos probables. Pero cada símbolo representa una misma unidad de DATOS.

Ej: Supongamos que convertimos el alfabeto a binario:

A $\to$ 00 $\to$ 2 bits
B $\to$ 01 $\to$ 2 bits
C $\to$ 10 $\to$ 2 bits
D $\to$ 11 $\to$ 2 bits

Cada símbolo emitido por la fuente representa con esta codificación en promedio 2 bits de DATOS

El LARGO DESCRIPTIVO MEDIO de la representación anterior es:

$$L = \frac{1}{|\mathcal{X}|} \sum_{\mathcal{X}} l_x \tag{16.2.1}$$

l_x: Número de bits que representan al símbolo x

En el ejemplo: $L = 2$ bits de DATOS

Cápsula 16.3 — Información y entropía

La información contenida en cada símbolo x se mide en BITS y puede ser determinada como:

$$I(x) = -\log_2(p_x) \tag{16.3.1}$$

p_x: Probabilidad del símbolo x del alfabeto $\mathcal{X}$

Ej: $\mathcal{X} = \{A, B, C, D\}$ $P_x = \{p_A, p_b, p_c, p_d\} = \underbrace{\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right\}}_{\Sigma = 1}$

Cada vez que:

$$\text{Ocurre A: } I(A) = -\log_2\left(\frac{1}{2}\right) = 1 \text{ bit}$$
$$\text{Ocurre A: } I(B) = -\log_2\left(\frac{1}{4}\right) = 2 \text{ bit}$$
$$\text{Ocurre C o D: } I(C) = I(D) = -\log_2\left(\frac{1}{8}\right) = 3 \text{ bit}$$

La cantidad de INFORMACIÓN MEDIA que contiene un símbolo emitido por la fuente es:

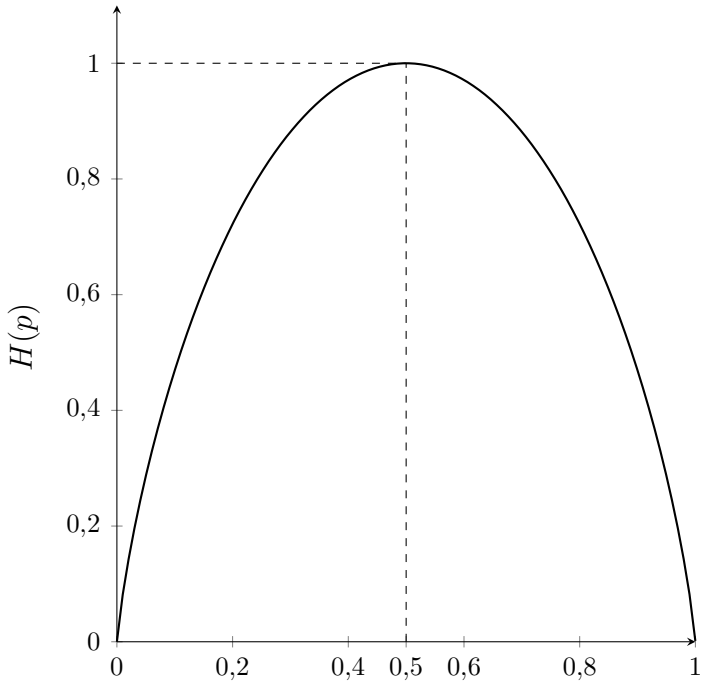
$$\mathbb{E}\{-\log_2(p_x)\} = -\sum_{x\in\mathcal{X}} p_x \log_2(p_x) \tag{16.3.2}$$
$$= \mathbb{E}\{I(x)\} \tag{16.3.3}$$
$$\triangleq H(x) \quad \left\{ \text{ENTROPÍA DE LA FUENTE} \right. \tag{16.3.4}$$

En el ejemplo:

$$H(x) = \frac{1}{4}\left(\frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{8}\cdot 3\right) = \frac{7}{4} \text{ bits de INFORMACIÓN} \tag{16.3.5}$$

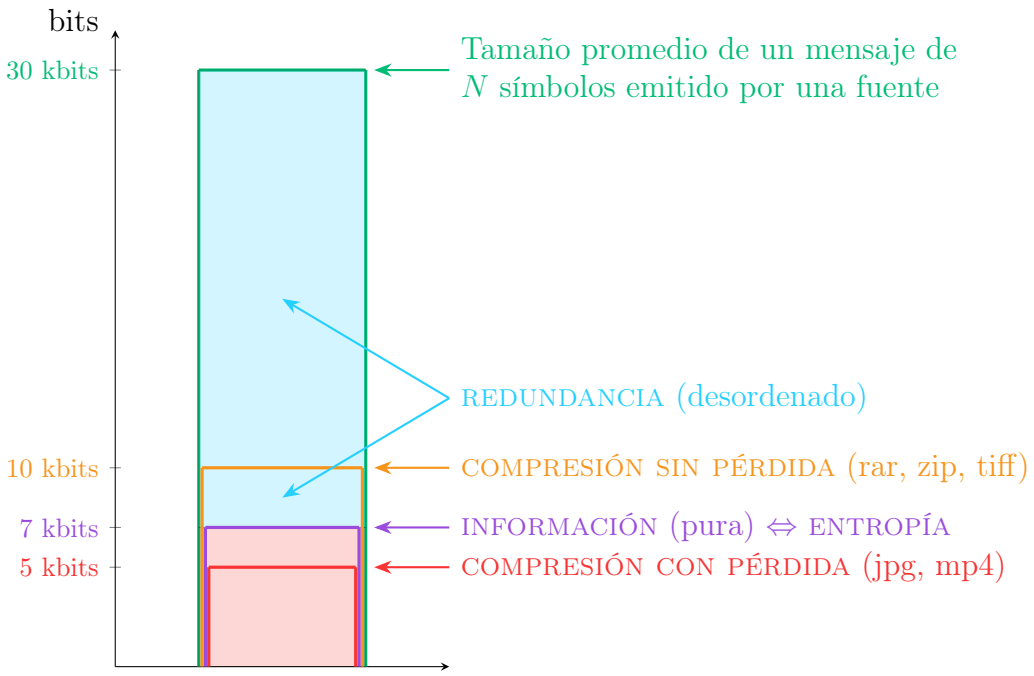
Si el alfabeto es binario: $\mathcal{X} = \{0, 1\}$ con $P(0) = p$ y $P(1) = 1 - p$, entonces:

$$H(x) = -p\log_2(p) - (1-p)\log_2(1-p) \tag{16.3.6}$$
$$\triangleq H(p) \tag{16.3.7}$$



Probabilidad de bit 0.

Cápsula 16.4 **Compresión (Codificación de fuente)**



Cápsula 16.5 Código Huffman (Compresión sin pérdida)

1.- Ordenar los símbolos de fuente de mayor a menor probabilidad.

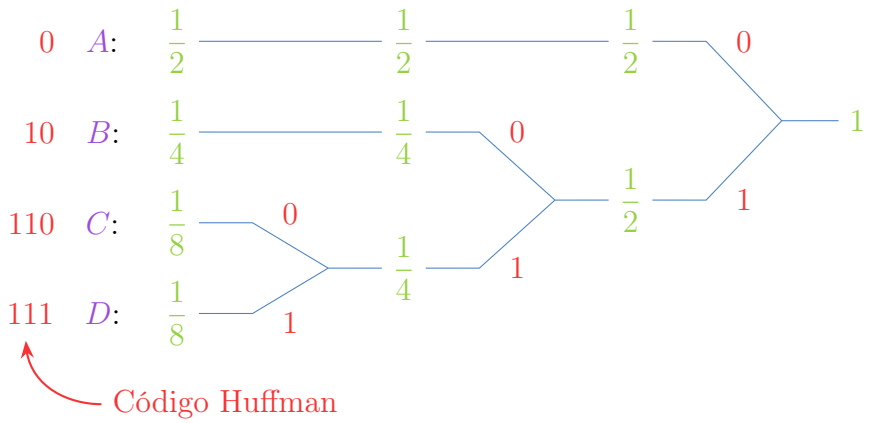
2.- Unir las dos probabilidades más pequeñas (las dos inferiores) y sumarlas.

3.- Extender las demás probabilidades.

4.- Volver al paso 1 hasta que quede una sola rama.

5.- Etiquetar las confluencias (bifurcaciones en sentido retrógrado) con 0 y 1 y leer el código de derecha a izquierda.

Ej. Fuente 4-aria A,B,C,D con probabilidad $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}$



Largo descriptivo medio del código:

$$L = 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} \cdot 2 = \frac{7}{4} \text{ bits!} \tag{16.5.1}$$

$\Rightarrow$ Este código logra comprimir a entropía!

Ej. Mensaje: BCABDB

Comprimido: 10 110 0 10 111 10

Decodificado: BCABDB

Código Huffman	
0	A:
10	B:
110	C:
111	D:

El código de Huffman permite una codificación/decodificación inambigua!

El tamaño medio de una código Huffman es:

$$H(x) \leq L \leq H(x) + 1 \tag{16.5.2}$$

Si p_x de todos los simbolos del alfabeto son de la forma $p_x = 2^{-k_x}$, con k_x entero para cada símbolo $x \in \mathcal{X}$, entonces $L = H(x)$. (Como en el ejemplo).

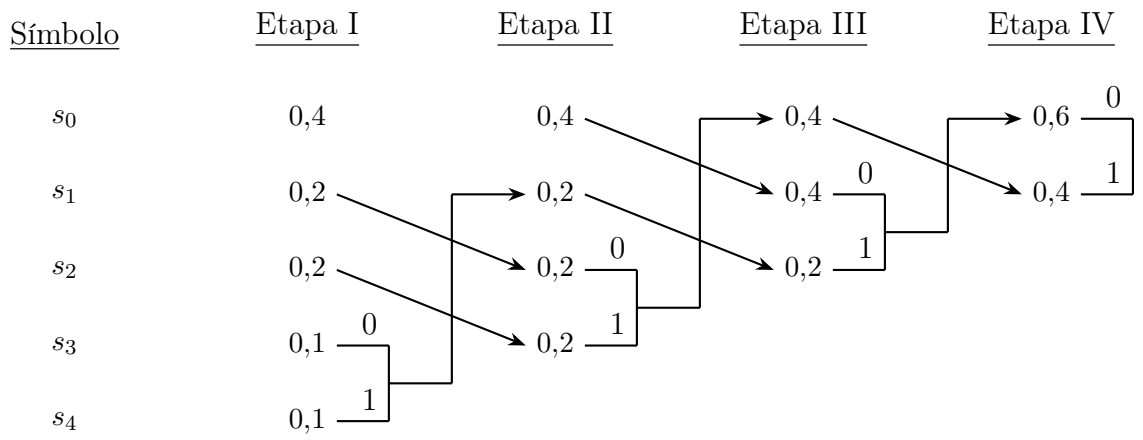
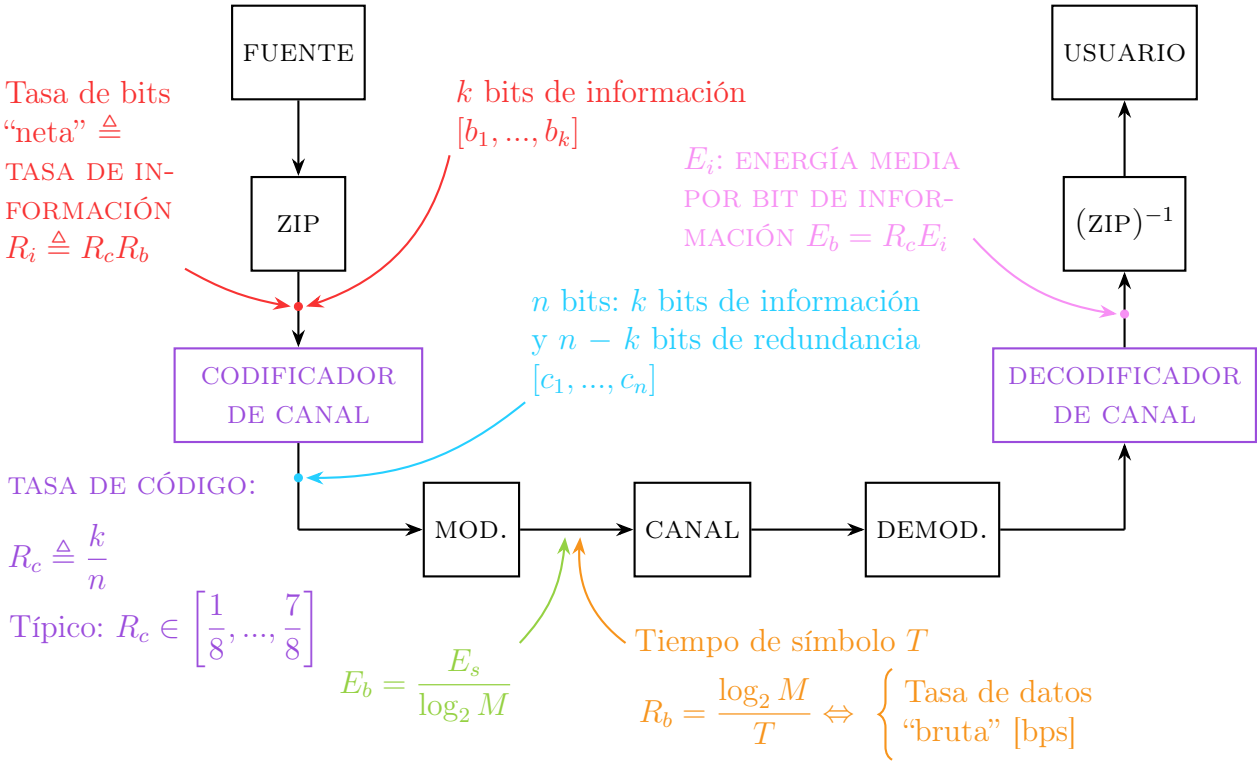


Figura 16.5.1: Ejemplo recreado de [Haykin].

Símbolo	Probabilidad	Codificación
s_0	0,4	1
s_1	0,2	01
s_2	0,2	000
s_3	0,1	0010
s_4	0,1	0011

Figura 16.5.2: Ejemplo recreado de [Haykin].

© 2010–2025 Christian Oberli



Cápsula 16.7 — Codificación de canal: comentarios

1.- Todos los métodos de codificación de canal comparten el mismo principio: agregar redundancia en forma controlada y estructurada a un mensaje de información tal que dicha estructura puede ser aprovechada en el receptor para detectar y posiblemente corregir errores.

2.- La transformación de los k bits de mensaje (información) $[b_1, ..., b_k]$ a los n bits codificados $[c_1, ..., c_n]$ puede ocurrir de varias formas:

$$[c_1, ..., c_n] = \begin{cases} [b_1, ..., b_k, c_{k+1}, ..., c_n] \\ \uparrow \quad\quad \uparrow \\ c_1 \quad\quad c_k \\ [c_1, ..., c_k] \end{cases}$$

PALABRA DE CÓDIGO $\vec{c}$

* Códigos de bloque.
* Códigos Turbo.

* Código convolucionales.
* Códigos LDPC.
* Códigos Trellis.

3.- Los códigos se caracterizan principalmente por:

- Capacidad para detectar errores.
- Capacidad para corregir errores.
- Complejidad de implementación.

4.- Acerca de la potencia de la señal que sale al aire:

$$P_s = \frac{E_s}{T} = \frac{\log_2(M)}{T} \cdot \frac{E_s}{\log_2(M)} \tag{16.7.1}$$

$$= R_b \cdot E_b \tag{16.7.2}$$

$$= R_c R_b \cdot \frac{1}{R_c} E_b \tag{16.7.3}$$

$$= R_i \cdot E_i \tag{16.7.4}$$

Cápsula 16.8 Codificación de canal: noción conceptual

Condiciones esenciales para que un código de canal funcione:

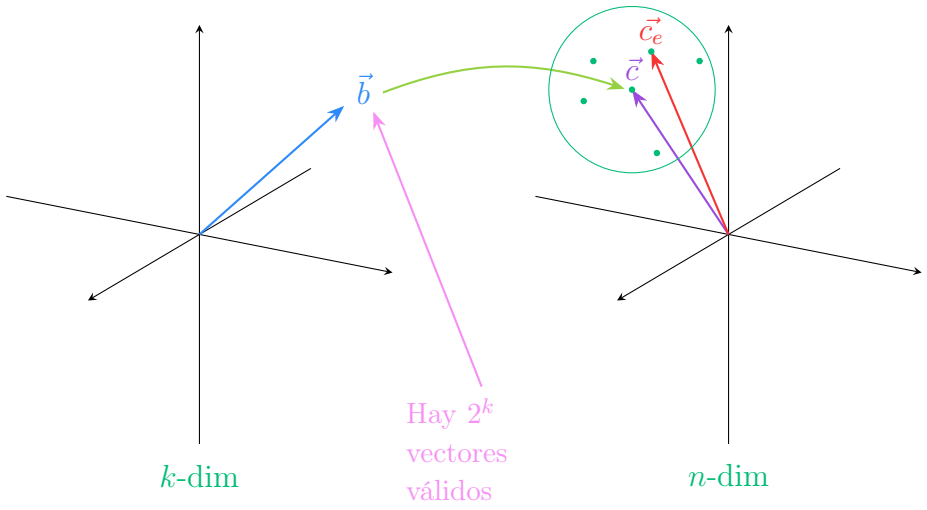
1.- Dada una palabra de código $\vec{c} = [c_1, ..., c_n]$, debe ser posible recuperar el mensaje original $\vec{b} = [b_1, ..., b_k]$ en forma inambigua.

$\Leftrightarrow$ La transformación $\vec{b} \to \vec{c}$ ("codificación") debe tener inversa.

2.- La transformación inversa debe funcionar incluso cuando $\vec{c}$ tiene errores:

$$[\checkmark c_1, \checkmark c_2, \times c_3, \checkmark c_{n-3}, \ldots, \checkmark c_{n-3}, \times c_{n-2}, \checkmark c_{n-1}, \checkmark c_n] \xrightarrow{()^{-1}} \vec{b}$$

Conceptualmente esto se puede lograr resolviendo un problema de EMPAQUETAMIENTO DE ESFERAS.



* Hay 2^n vectores $\vec{c}$ posibles, pero solo 2^k de ellos corresponden a un vector $\vec{b}$ de origen. Para cada vector válido, hay:

$$\frac{2^n - 2^k}{2^k} = 2^k - 1 \tag{16.8.1}$$

vectores $\vec{c}$ "no válidos" $\triangleq \vec{c}_e$.

* Empaquetamiento de esferas: ¿Cómo escoger 2^k vectores $\vec{c}$ válidos que entre sí estén a la mayor distancia posible y rodeado cada uno de $2^k - 1$ vectores no válidos ($\vec{c}_e$) que le sean similares?

* El desafío de diseño de un código corrector de errores está en encontrar una transformación que además de ser invertible también mapee correctamente los $2^k - 1$ vectores no válidos en la esfera de un vector $\vec{c}$ a su vector $\vec{b}$ de origen.
$\Leftrightarrow$ "Todos los vectores en la esfera de $\vec{c}$ son mapeados de vuelta a $\vec{b}$".

* El error de decodificación de canal ocurre cuando debido al ruido los errores de decisión en $\vec{c}$ generan un $\vec{c}_e$ fuera de la esfera correcta.
$\Rightarrow$ La transformación inversa aterriza en un $\vec{b}$ incorrecto.

* La capacidad de corrección de un código depende fuertemente de la razón $\frac{k}{n} = R_c$
$\Leftrightarrow$ Mayor redundancia $\Leftrightarrow \frac{k}{n}$ menor $\Leftrightarrow R_e \downarrow$
$\frac{k}{n} \downarrow \Rightarrow 2^{n-k} \uparrow \Rightarrow \uparrow$ tamaño de la esfera.

Cápsula 16.9 **Códigos de repetición**

© 2010–2025 Christian Oberli

$k = 1 \quad \Rightarrow \quad \vec{b} = [b_1]$

Cod. rep.

$n = 3 \quad \Rightarrow \quad \vec{c} = [c_1\ c_2\ c_3] = [b_1\ b_1\ b_1] = \begin{cases} [0,0,0] \\ [1,1,1] \end{cases}$

$R_c = \dfrac{1}{3}$

Sup. se transmite $\vec{c}$ y se recibe:

$$\vec{c}_e = [b_1\ \bar{b}_1\ b_1]$$

$$\text{Ej: } \vec{c}_e = \begin{cases} [0,1,0] \text{ si se transmitió } 0 \\ [1,0,1] \text{ si se transmitió } 1 \end{cases}$$

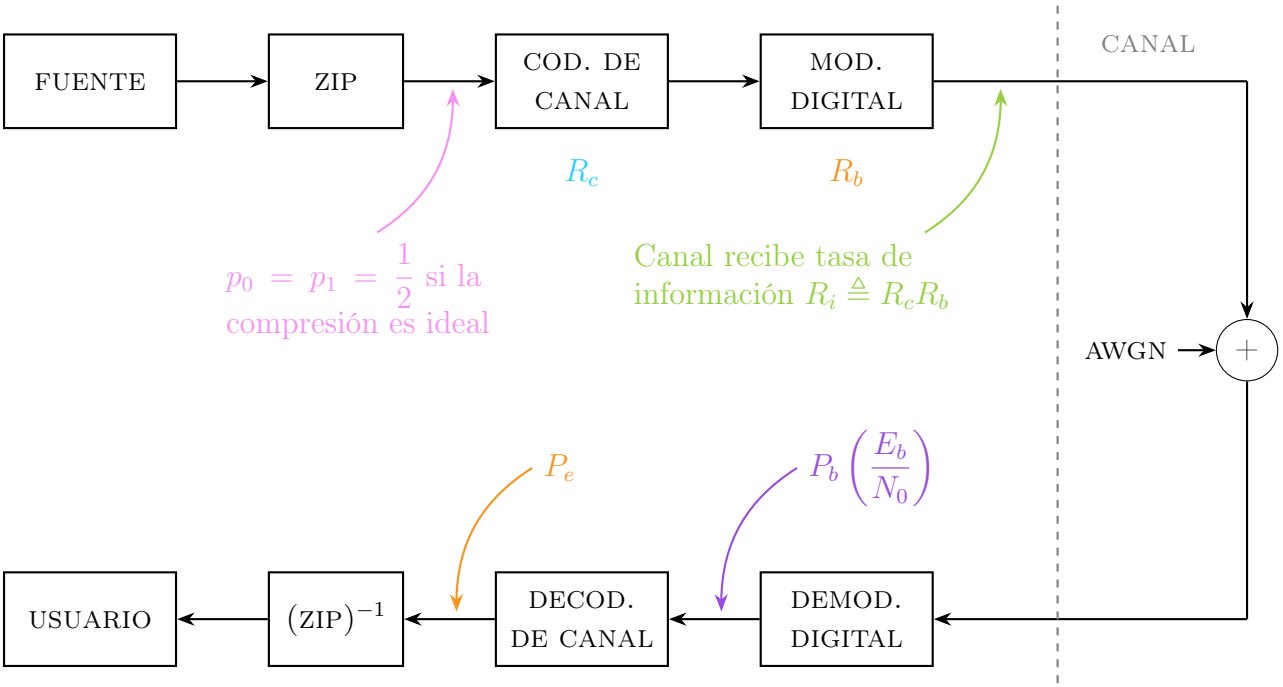
Es claro que el $\vec{c}$ recibido es un vector no válido.
La capacidad de DETECCIÓN de errores de este código es 2.
La capacidad de CORRECCIÓN de errores de este código es 1. (Error de 1 bit es más probable que error de 2 bits $\Rightarrow$ se opta por decretar que la palabra original $\vec{b}$ es la que corresponde al vector $\vec{c}$ más cercano a $\vec{c}_e$.

Ej.

$$\vec{c} = [0,0,0] \Rightarrow \vec{c}_e = \begin{cases} [1,0,0] \\ [0,1,0] \\ [0,0,1] \end{cases}$$

Hay $2^{n-k} - 1 = 2^{3-1} - 1 = 3$ vectores no válidos $\vec{c}_e$ que corresponden a cada vector $\vec{c}$.

Cápsula 16.10 **Teorema de capacidad de canal**

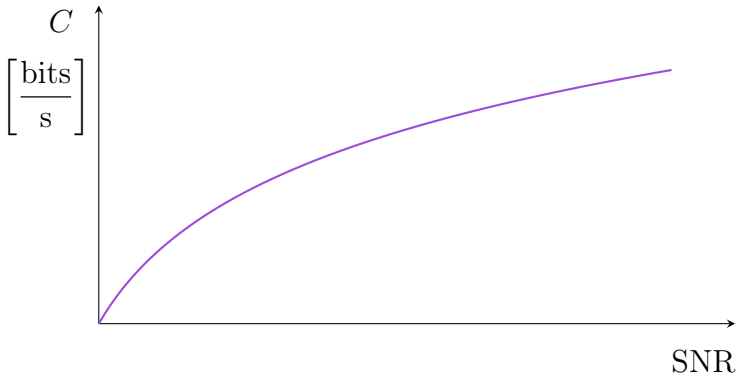


TEOREMA (Claude Shannon, 1948):

"El límite teórico para la tasa de información máxima posible que puede de transferida SIN ERRORES a través de una canal AWGN de ancho de banda W está dado por:

$$C = W \log_2 (1 + \text{SNR}) \ \frac{\text{bits}}{\text{s}} \tag{16.10.1}$$

(W: (Hz); SNR: $\frac{P_s}{P_n}$ en unidades absolutas)



Es común expresar C como eficiencia espectral:

$$\eta_W = \frac{C}{W} = \log_2(1+\text{SNR}) \ \frac{\frac{\text{bits}}{\text{s}}}{\text{Hz}} \tag{16.10.2}$$

Algunos comentarios:

1.- Conocer C es bueno: "rayado de cancha".

2.- Conocer C es "frustrante": El teorema de capacidad no da luces sobre cómo implementar sistemas que logren C en la práctica.

3.- C es un límite basado en leyes de la física y matemática.

4.- SNR$\uparrow \Rightarrow C \uparrow$ no linealmente (rendimiento decreciente).
$W \uparrow \Rightarrow C \uparrow$ linealmente.

5.- C es una herramienta para valoriazr el espectro radioeléctrico.

Cápsula 16.11 — Frontera de capacidad

Consideramos un sistema de comunicación IDEAL:

* Transmite sin errores a tasa C bps
* Utiliza E_b por bit. $\Rightarrow P_s = E_b \cdot C \quad [W]$
* $P_n = N_0 W$

Entonces:

$$\overset{\eta_W}{\frac{C}{W}} = \log_2\left(1+\frac{P_s}{P_n}\right) = \log_2\left(1+\frac{E_b}{N_0}\overset{\eta_W}{\frac{C}{W}}\right) \tag{16.11.1}$$

$$\eta_W = \log_2\left(1+\frac{E_b}{N_0}\eta_W\right) \ \Rightarrow \text{Solver} \tag{16.11.2}$$

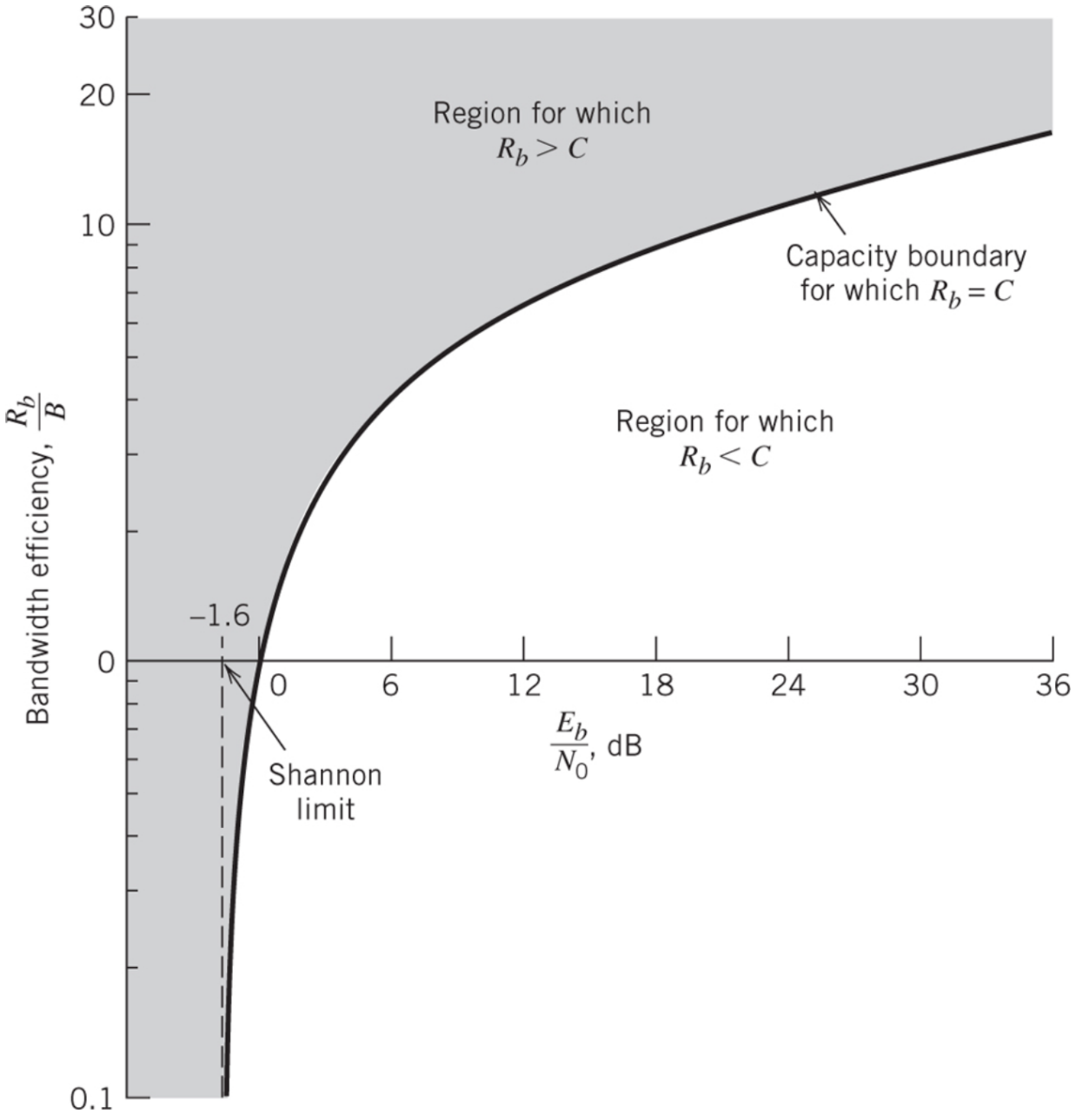


Figura 16.11.1: Fuente: Haykin

La FRONTERA DE CAPACIDAD es la relación entre η_W y $\frac{E_b}{N_0}$ de sistemas ideales que operan a capacidad.

Usando el resultado anterior:

$$\frac{C}{W} = \log_2\left(1 + \frac{E_b}{N_0}\frac{C}{W}\right) \tag{16.12.1}$$

Despejando $\frac{E_b}{N_0}$:

$$\Rightarrow \frac{E_b}{N_0} = \frac{2^{\frac{C}{W}} - 1}{\frac{C}{W}} \tag{16.12.2}$$

Hacemos $\underbrace{W \to \infty}_{\frac{C}{W} = \eta_W \to 0} \Rightarrow \frac{E_b}{N_0} \to \ln(2) \approx 0{,}693 \Leftrightarrow -1{,}59 \text{ dB}$

Visto como curvas de P_e:

