

**Maestría en
Sistemas Embebidos**

Sistemas Digitales
para las
Comunicaciones

Sistemas de comunicación

El punto de vista de la teoría de la información

Parte 4

Cronograma

Parte 0

Parte 1

Parte 2

Parte 3

Parte 4

Parte 5

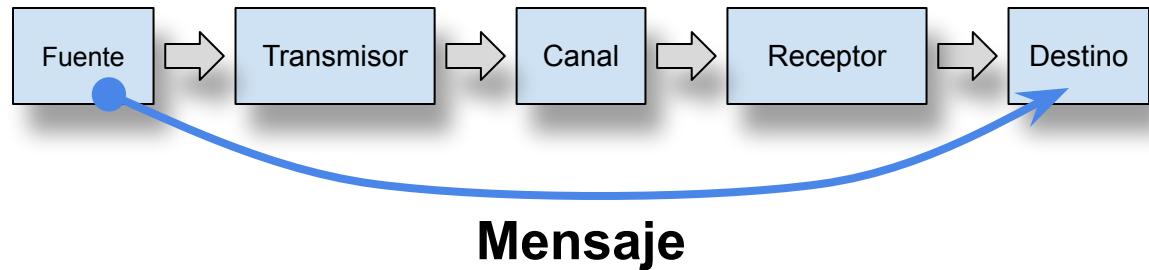
Parte 6

Parte 4: Sistema de comunicación - Teoría de información.

- Introducción a los sistemas de comunicación digitales.
 - Fuente, mensaje, transmisor, receptor y canal.
 - Codificación.
- Codificación de fuente.
 - Fuentes: Información y propiedades.
 - Codificación de Huffman y Lempel-Ziv.
- Codificación de canal.
 - Modelos y capacidad de canal.
 - Detección y corrección de errores.
 - Códigos por bloque lineales y convolucionales.
 - Entrelazado y codificación de línea.

Sistema de comunicación:

- **Fuente de mensaje**
- **Transmisor**
- **Canal**
- **Receptor**
- **Destino del mensaje**



- Existe un **mensaje** que se genera en la **fuente**.
- El mensaje es aleatorio.
- Se quiere enviar al **destino** ...
 - ... a través de un **canal**.
- Es **transmisor** se encarga de adaptar el mensaje para que atraviese el canal.
- El **receptor** se encarga de recuperar el mensaje y volverlo a su formato original.

Sistema de comunicación

Fuente de mensaje

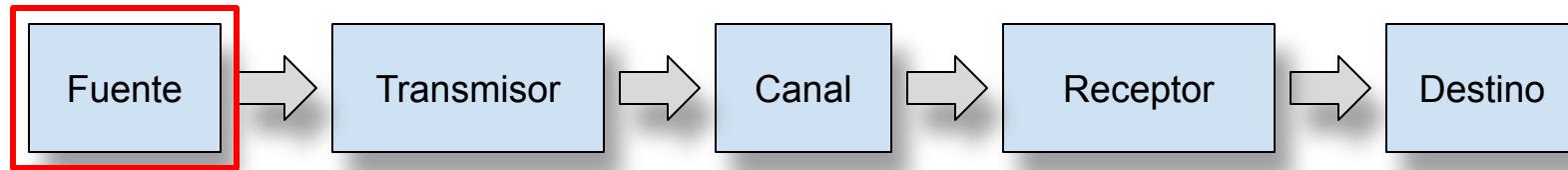


Emite un mensaje que es **aleatorio**.

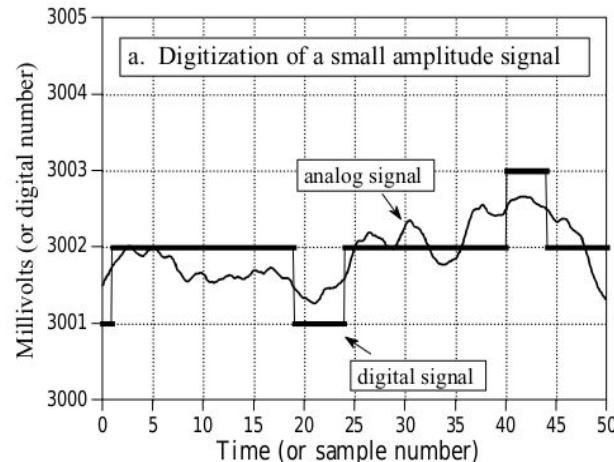
Si no fuese aleatorio no habría “nada para transmitir”.

Sistema de comunicación

Fuente de mensaje



- Fuentes **discretas** (en tiempo y amplitud):
 - Tener un alfabeto
 - Tener una tasa de símbolos R
- Fuente **continua**:
 - **Cuantización y muestreo**
 - En general con **pérdida** de información: Teoría de Rate-Distortion



Sistema de comunicación

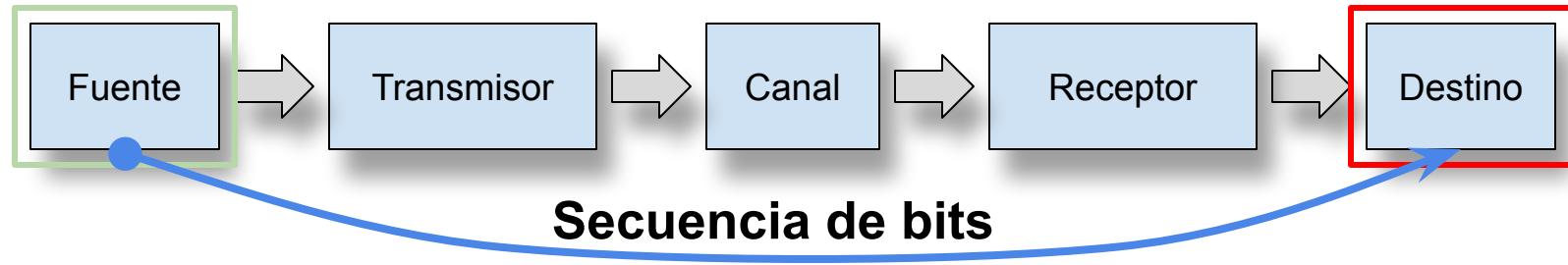
Destino



“Consumo” el mensaje en el formato original.

Sistema de comunicación

Sistema de comunicación digital



A partir de ahora consideramos que tenemos un mensaje discreto:

- Si tengo fuente analógica la cuantizo y muestreo
 - Me olvido de la distorsión y pérdida de información: Problema de otro área
- Alfabeto con $M = 2^b$ letras, equivalente a agrupar bits
 - Mensaje = Secuencia de bits
- Entonces tengo un sistema de **comunicación digital**

Sistema de comunicación

Canal



- El canal es un medio inherentemente **analógico**
- Introduce **distorsiones** en el mensaje que se quiere transmitir
 - Respuesta en frecuencia
 - Atenuación
 - Ruido
 - Otras fuentes de distorsión
lineales y no lineales



Sistema de comunicación

Canal



- El canal es un medio inherentemente **analógico**
- Introduce **distorsiones** en el mensaje que se quiere transmitir
 - Respuesta en frecuencia
 - Atenuación
 - Ruido
 - Otras fuentes de distorsión lineales y no lineales

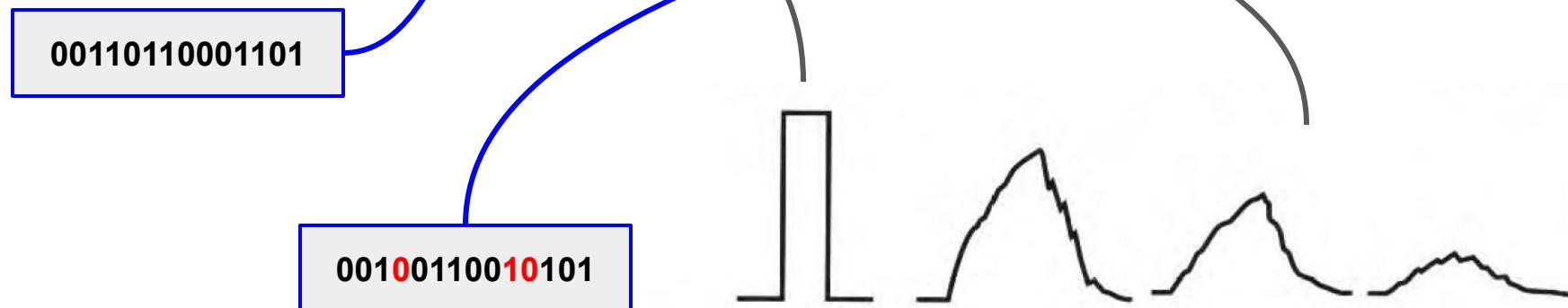


Sistema de comunicación

Canal



Desde un punto de vista más abstracto:



Sistema de comunicación

Capacidad del canal



¿Qué hacen el transmisor y el receptor?

Hacen posible el envío del mensaje

¿Cómo?

Sistema de comunicación

Capacidad del canal



Objetivo del transmisor y receptor en conjunto:

- Extraer la información de la fuente
- Revertir los errores introducidos por el canal

Sistema de comunicación

Transmisor y receptor



Objetivo del transmisor y receptor en conjunto:

- Extraer la información de la fuente
- Revertir los errores introducidos por el canal

El proceso para alcanzar ambos los objetivos se llama **codificación**

Se puede ver intuitivamente que es un proceso de dos etapas.

- Codificación de fuente
- Codificación de canal

Sistema de comunicación

Transmisor y receptor



Objetivo del transmisor y receptor en conjunto:

- Extraer la información de la fuente
- Revertir los errores introducidos por el canal

Se puede demostrar que las dos etapas son independientes. Es un resultado importante de la teoría de información: “*source-channel separation theorem*”

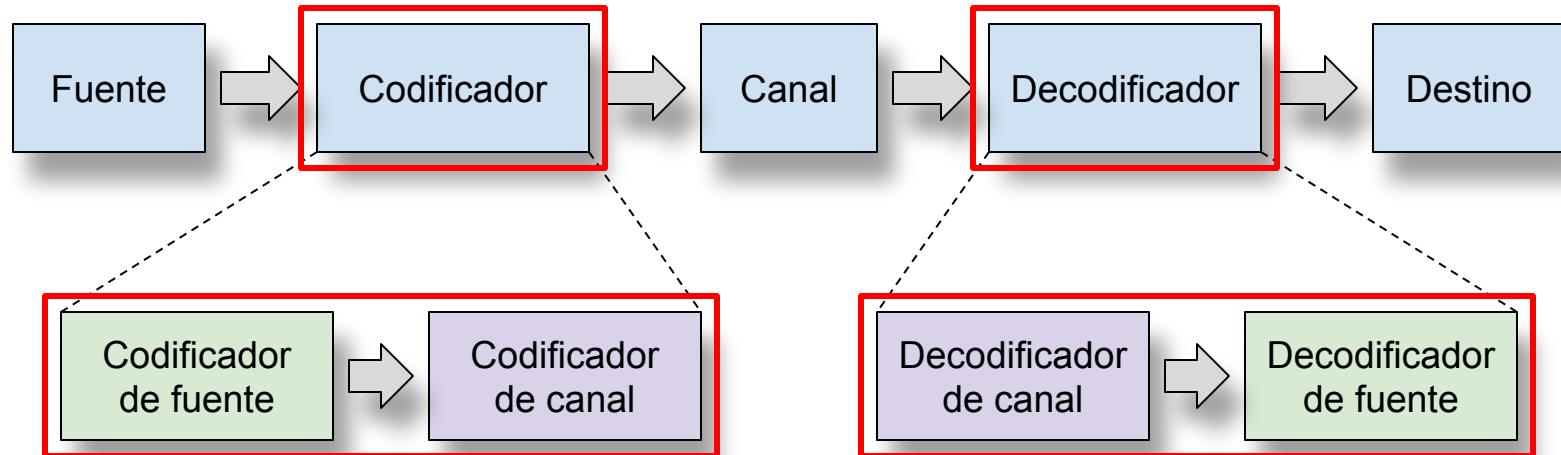
El proceso para alcanzar ambos los objetivos se llama **codificación**

Se puede ver intuitivamente que es un proceso de dos etapas.

- Codificación de fuente
- Codificación de canal

Sistema de comunicación

Transmisor y receptor

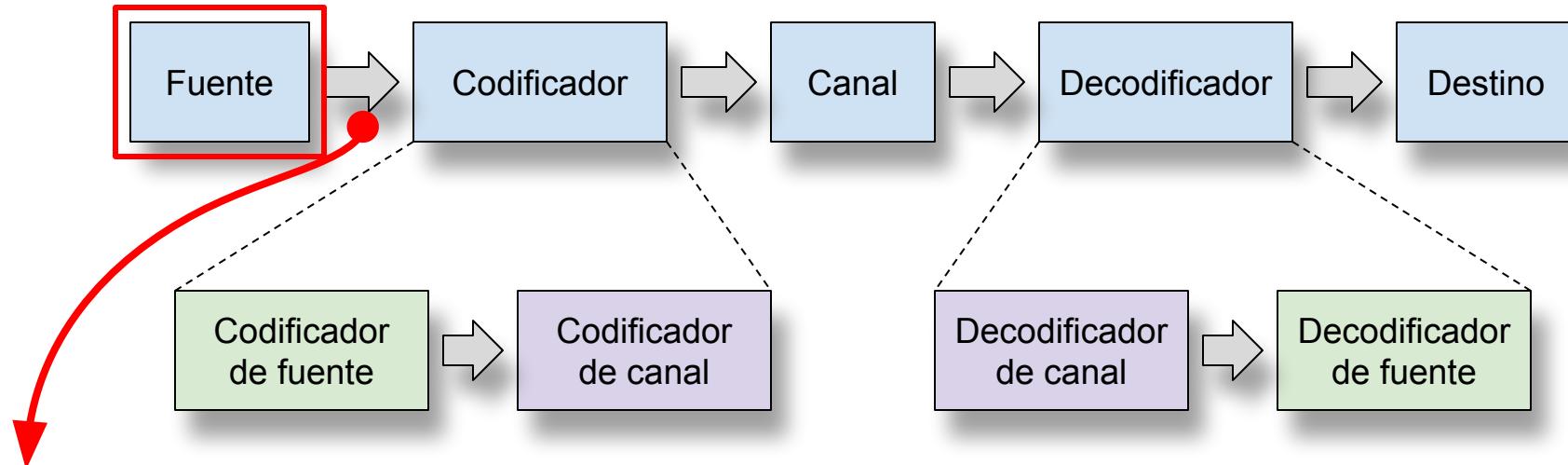


Se divide el problema en dos subproblemas pueden ser atacados de manera independiente:

[De]codificación de fuente y [De]codificación de canal

Sistema de comunicación

Resumen



Fuente y alfabeto de la fuente

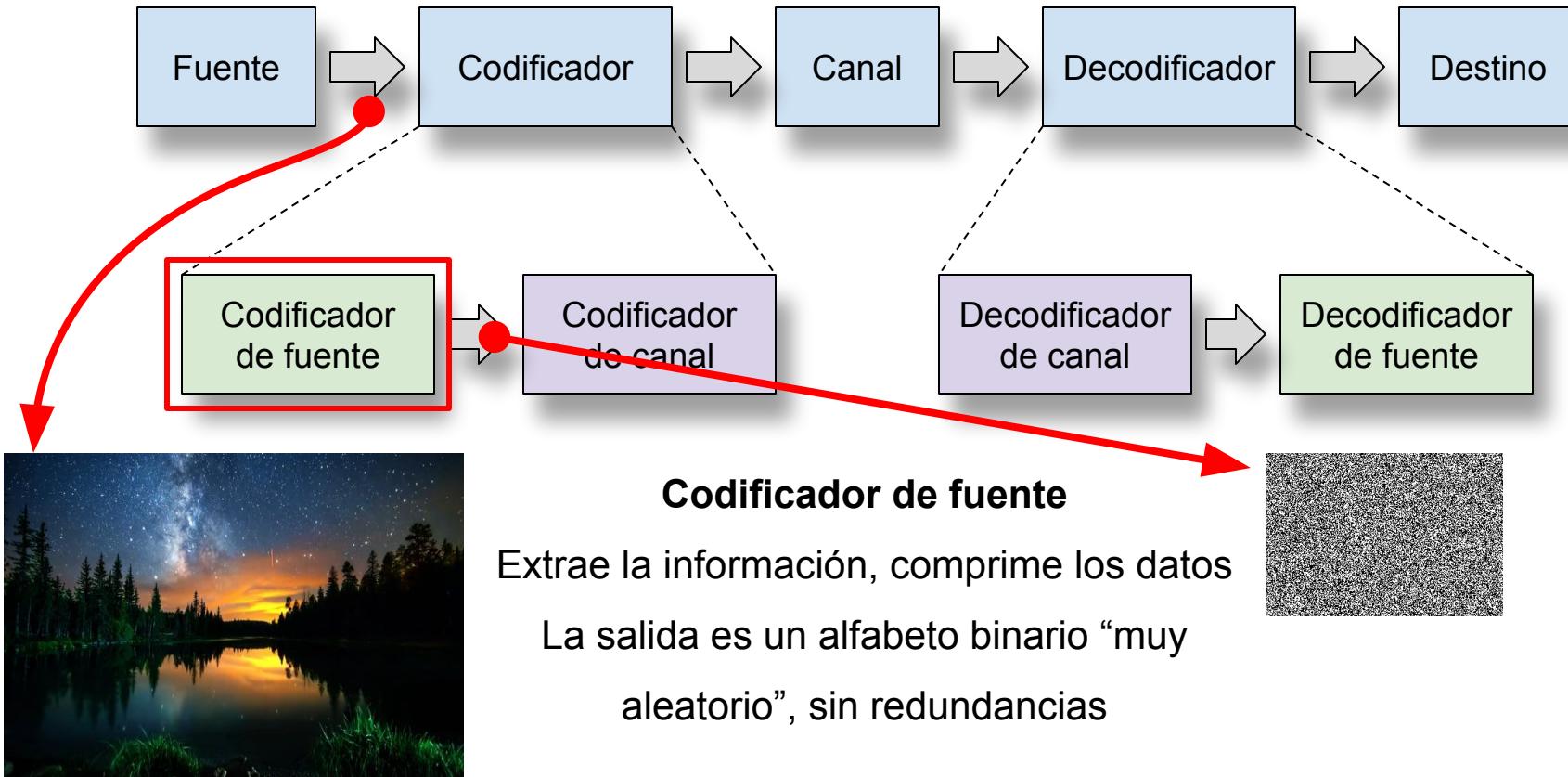
Puede tener muchos datos y poca información

Puede tener pocos datos y mucha información

Es necesaria una medida de información

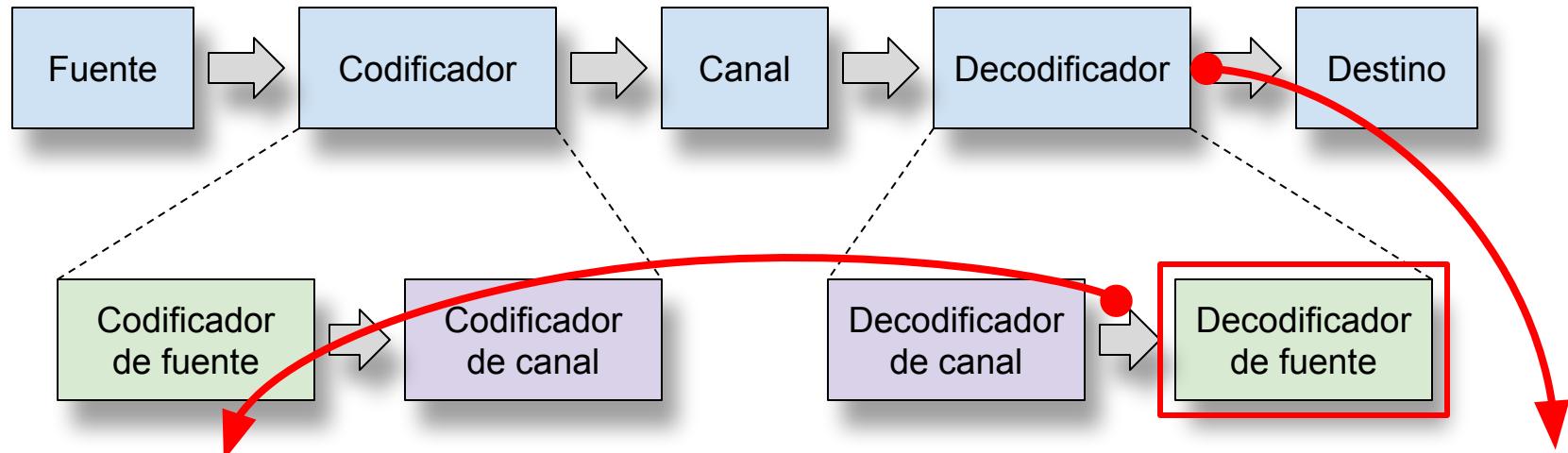
Sistema de comunicación

Resumen



Sistema de comunicación

Resumen



Decodificador de fuente

Descomprime los datos

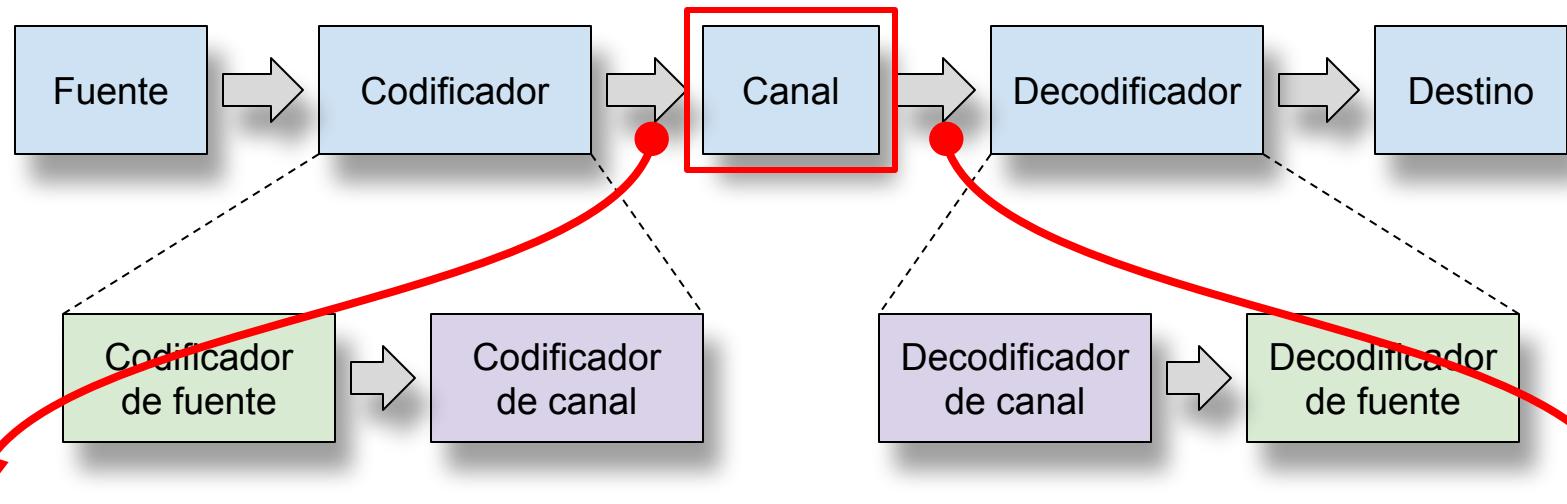
Revierte el proceso de codificación

Presenta los datos de manera conveniente



Sistema de comunicación

Resumen



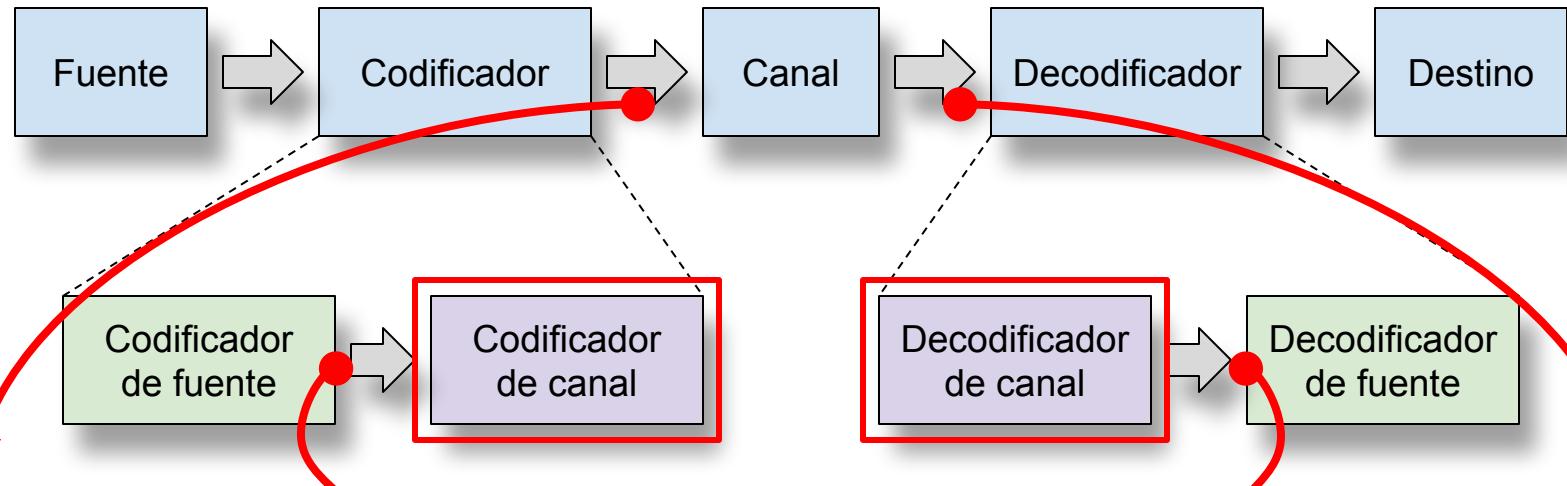
Canal

Introduce distorsiones: Ruido, atenuación y respuesta en frecuencia, lo cual genera errores en los datos.



Sistema de comunicación

Resumen



Codificación y decodificación de canal

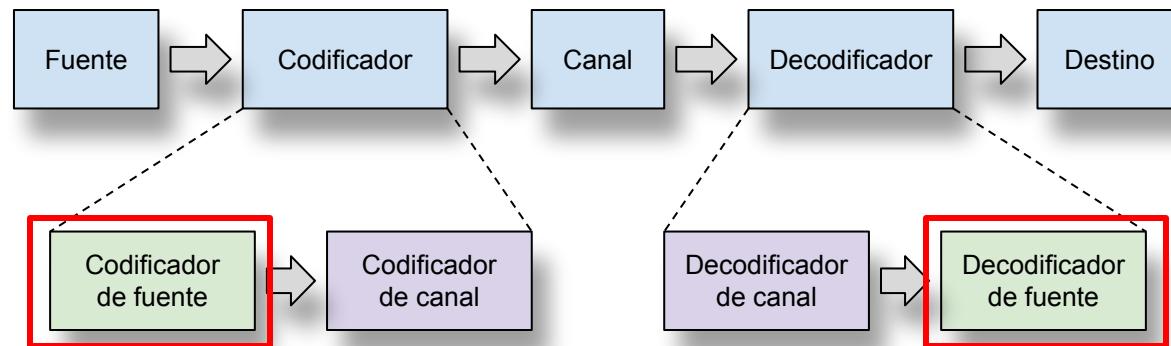
Agrega redundancia

Revierte los errores introducidos por el
canal

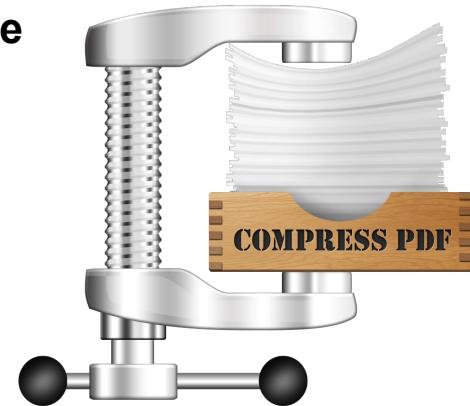


Codificación de fuente:

- Introducción a la teoría de información
 - Medida de información
 - Modelos de fuentes
- Códigos
 - De largo variable
 - Cod. de Huffman
 - Compresión Lempel-Ziv



El objetivo principal es representar los datos con la menor cantidad posible de bits. Esto se llama **compresión de datos**.



Codificación de fuente

Medida de información

Consideremos las siguientes titulos de una noticia:

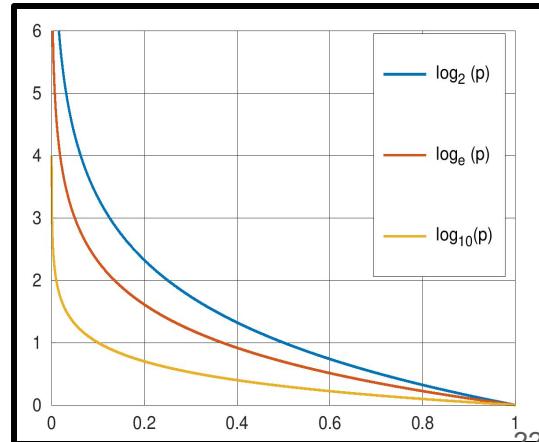
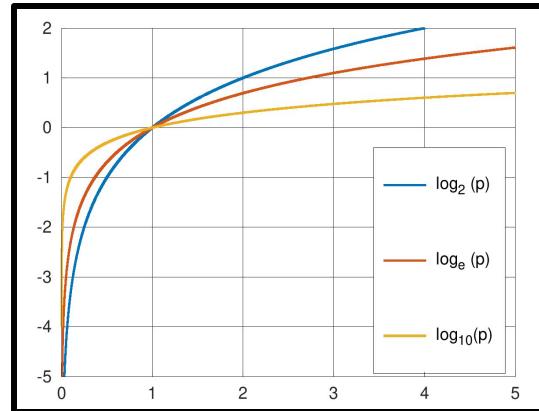
“El sol salió esta mañana por el este en nuestro país”

“El COVID-19 se extinguió por completo en el mundo”

“Se comprobó que el COVID-19 nunca existió”

Es una **secuencia de letras de un alfabeto**

- ¿Cuál de estas atrae más la atención del lector?
 - Evaluemos las probabilidades de cada uno de esos eventos
 - ¿Existe una relación?
- Si **Prob=1 \Rightarrow Info=0**, mientras que si **Prob=0 \Rightarrow Info= ∞**
- Matemáticamente una posible medida es: **Info = $\log_b(1/\text{Prob})$**



Codificación de fuente

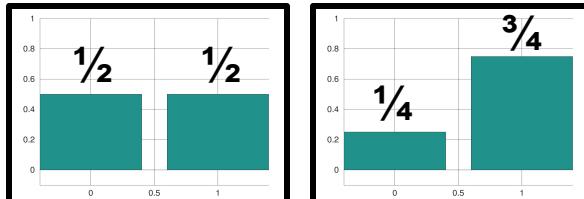
Modelos de fuentes discretas

Es una **secuencia de letras de un alfabeto**

- Alfabeto: Un conjunto de letras
- Fuente: Secuencia $\{X_1, X_2, X_3, \dots, X_L\}$ de letras
- Prob. de cada letra

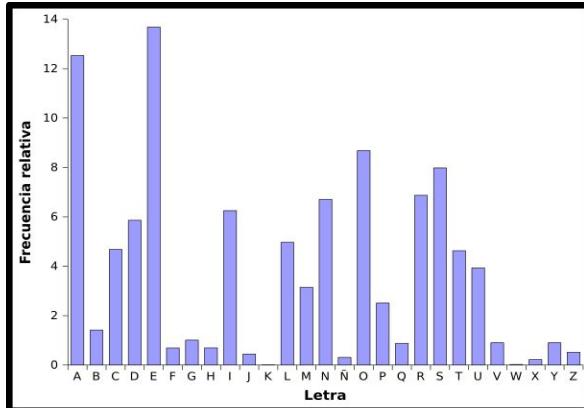
Ejemplo binario:

- $L = \{0,1\}$
- $P(X_i=0) = p, P(X_i=1) = 1-p$



Ejemplo del idioma español:

- $L = \{a, b, \dots, z\}$
- $P(X_i=a) = 0.1253\dots$
- $P(X_i=b) = 0.0142\dots$
- ...



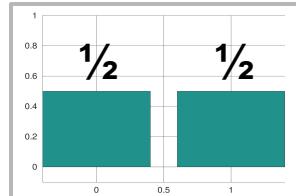
1110111011000110110101100010110111101100
10100110111001100110001011110101101101100
00110001000010101010111010011011011010101
10100101111101111111000110110101001000
00010001111011010001100011011111101000
0010111001111111011000101101011100101100
01001000101011110110001001111010100001
001111110001010101110000010011100111100
011001001111000111110100011101100001011100
11110011001010100011111000000101001111100
10100010110000110010101111101100101110011
1101011011011111111100000001100001111000
101001100101010101101011110100111010100101
0011101001010011111100000101011010101101
010111111101101000110110101101011000101101
01000011101001101111100110011000101111011
101101000011000100001010101011101001101101
001010001010010111111011111110001101101
001001010001000111101000110001100011011111

b z y g x g x c y b b z y g x g x c y b
e y c g a b c d y a e y c g a b c d y a
g c x x x x g x c y g c x x x x g x c y
x z b d b x z a g d x z b d b x z a g d
b d e z y e g z a e b d e z y e g z a e
e g x b a y x a y d e g x b a y x a y d
f z e b a f z z d x f z e b a f z z d x
x y c d f e c g z g x y c d f e c g z g
g e g x e z d y x e g e g x e z d y x e
e g x c e b c y x d e g x c e b c y x d

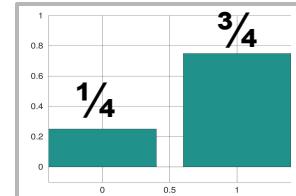
Codificación de fuente

Entropía de una fuente

Información de ...



$$M = \{0101\dots01010\}$$
$$M = \{X_1, X_2, \dots, X_L\}$$



$$M = \{0101\dots01010\}$$
$$M = \{X_1, X_2, \dots, X_L\}$$

... cada símbolo

$$I(X=0) = -\log_2(p) = 1$$

$$I(X=0) = -\log_2(1-p) = 1$$

$$I(X=0) = -\log_2(p) = 1$$

$$I(X=0) = -\log_2(1-p) = 1$$

... una letra: Prom.
información

$$H(X) = \text{Promedio}(I) =$$
$$= \frac{1}{2} (-\log_2(\frac{1}{2})) + \frac{1}{2} (-\log_2(\frac{1}{2})) =$$
$$= \frac{1}{2} 1 + \frac{1}{2} 1 = 1$$

$$H(X) = \text{Promedio}(I) =$$
$$= \frac{1}{4} (-\log_2(\frac{1}{4})) + \frac{3}{4} (-\log_2(\frac{3}{4})) =$$
$$= \frac{1}{4} 2 + \frac{3}{4} 0.415\dots = 0.811\dots$$

... un mensaje
(secuencia de
letras) es:

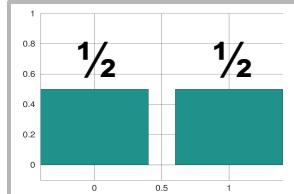
$$H(M) = H(\{X_1, X_2, \dots, X_L\}) =$$
$$= H(X_1) + H(X_2) + \dots + H(X_L) =$$
$$= L * H(X) = L$$

$$H(M) = H(\{X_1, X_2, \dots, X_L\}) =$$
$$= H(X_1) + H(X_2) + \dots + H(X_L) =$$
$$= L * H(X) = 0.81 L$$

Codificación de fuente

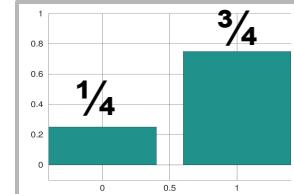
Entropía de una fuente

Información de ...



$$M = \{0101\dots01010\}$$

$$M = \{X_1, X_2, \dots, X_L\}$$



$$M = \{0101\dots01010\}$$

$$M = \{X_1, X_2, \dots, X_L\}$$

... cada símbolo

... una letra: Porm.
información

... un mensaje
(secuencia de
letras) es:

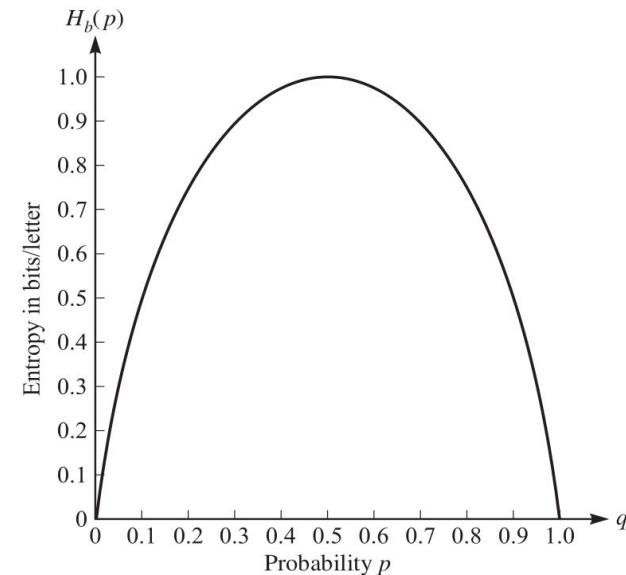
La entropía de una letra es el promedio de la información. Se mide en bits (cuando se usa \log_2)

Un mensaje es un conjunto de L letras y su entropía es.

Para una fuente **sin memoria, estacionaria y discreta** vale que:

$$H(X) : \text{(ver gráfico) [bits]}$$

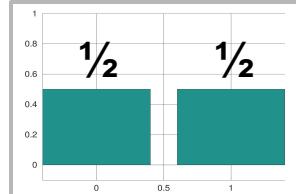
$$H(M) = L H(X) \text{ [bits]}$$



Codificación de fuente

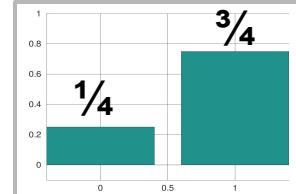
Entropía de una fuente

Información de ...



$$M = \{0101\dots01010\}$$

$$M = \{X_1, X_2, \dots, X_L\}$$



$$M = \{0101\dots01010\}$$

$$M = \{X_1, X_2, \dots, X_L\}$$

... cada símbolo

... una letra: Porm.
información

... un mensaje
(secuencia de
letras) es:

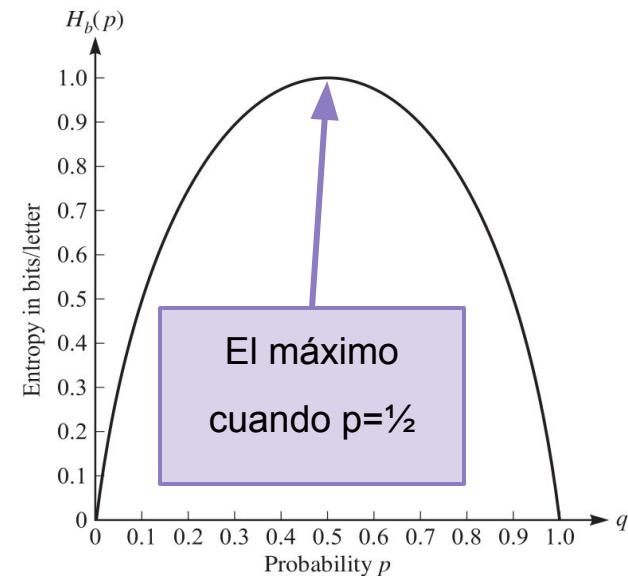
La entropía de una letra es el promedio de la información. Se mide en bits (cuando se usa \log_2)

Un mensaje es un conjunto de L letras y su entropía es.

Para una fuente **sin memoria, estacionaria y discreta** vale que:

$$H(X) : \text{(ver gráfico) [bits]}$$

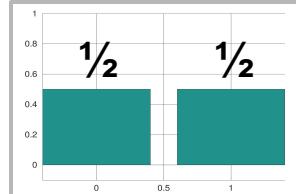
$$H(M) = L H(X) \text{ [bits]}$$



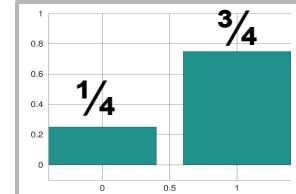
Codificación de fuente

Entropía de una fuente

Información de ...



$$M = \{0101\dots01010\}$$
$$M = \{X_1, X_2, \dots, X_L\}$$



$$M = \{0101\dots01010\}$$
$$M = \{X_1, X_2, \dots, X_L\}$$

... cada símbolo

... una letra: Porm.
información

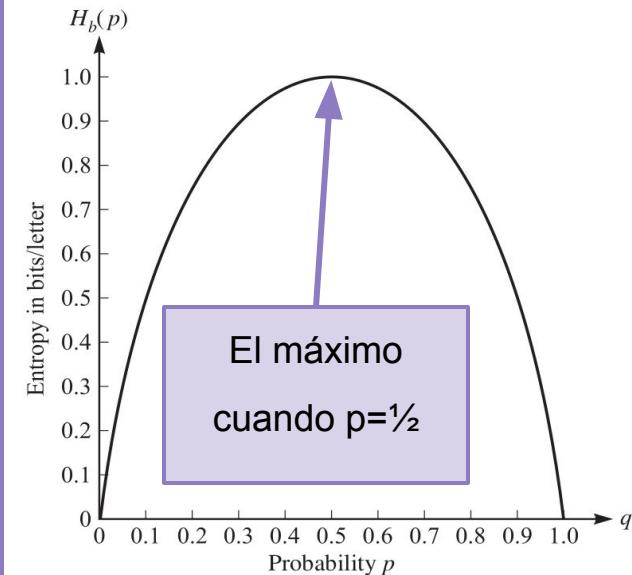
... un mensaje
(secuencia de
letras) es:

La entropía de una letra es el promedio de la información. Se mide en bits (cuando se usa \log_2)

Un mensaje es un conjunto de L letras y su entropía es.

Para una fuente **sin memoria, estacionaria y discreta** vale que:

$$H(X) : \text{(ver gráfico) [bits]}$$
$$H(M) = L H(X) \text{ [bits]}$$



Codificación de fuente

Modelos de fuentes discretas

Memoria de una fuente:

Si la probabilidad de las letras es independiente del pasado, se dice que el canal **no tiene memoria**

Ejemplo fuente binaria:

```
0100101010010101010110110011010101011000  
110101111010101001010100110010101101011
```

```
000000000111111111111111111100000000000  
00000000111111111111111111110000000111110000
```

Ejemplo idioma español:

```
b z y g x g x c y b  
e y c g a b c d y a  
g c x x x x g x c y  
x z b d b x z a g d  
b d e z y e g z a e  
e g x b a y x a y d  
f z e b a f z z d x  
x y c d f e c g z g  
g e g x e z d y x e  
e g x c e b c y x d
```

HOLA COMO ESTAS YO
ESTOY BIEN POR SUERTE

HOLA MUNDO

HOLA OLA VASO CAMA UVA
PURE CASA PLACA LAPIZ

Codificación de fuente

Modelos de fuentes discretas

Memoria de una fuente

Si la probabilidad de los símbolos es independiente del pasado, se dice que el canal **no tiene memoria**

¿Cuál tiene menos memoria?

Ejemplo fuente binaria:

```
0100101010010101010110110011010101011000  
1101011110101001010100110010101101011
```

```
000000000111111111111111000000000000  
000000001111111111111100000000111110000
```

Ejemplo idioma español:

```
b z y g x g x c y b  
e y c g a b c d y a  
g c x x x x g x c y  
x z b d b x z a g d  
b d e z y e g z a e  
e g x b a y x a y d  
f z e b a f z z d x  
x y c d f e c g z g  
g e g x e z d y x e  
e g x c e b c y x d
```

HOLA COMO ESTAS YO
ESTOY BIEN POR SUERTE

HOLA MUNDO

HOLA OLA VASO CAMA UVA
PURE CASA PLACA LAPIZ

Codificación de fuente

Modelos de fuentes discretas

Memoria de una fuente

Si la probabilidad de los símbolos es independiente del pasado, se dice que el canal **no tiene memoria**

¿Cuál tiene menos memoria?

Ejemplo fuente binaria:

01001 1010010101 10111011001101010 1000

110101111 10101001010 00110010101 01011

00000 000111111 111111111100000 0000

00000001 11111111111 10000000111 10000

Ejemplo idioma español:

b e g x g x c y b
e y c g a b c d y a
g c x x x x c y
x z b d b x z a g d
b d e z y e g z a e
e g x a y x a y d
f z e b a f z z d x
x y c d f e c g z g
e g x e z d x e
e g x c e b c y x d

HOL COMO ESTAS YO
ESTOY BIEN POR SUERTE

HOL MUNDO

OLA O A VASO CAM UVA
PURE ASA PLAC LAPIZ

Codificación de fuente

Modelos de fuentes discretas

Memoria de una fuente

Si la probabilidad de los símbolos es independiente del pasado, se dice que el canal **no tiene memoria**

¿Cuál tiene menos memoria?

Ejemplo fuente binaria:

01001 1010010101 10111011001101010 1000
1101011111 10101001010 000110010101 01011

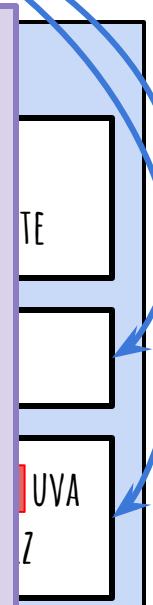
00000 0001111111 1111111111100000 0000
000000001 111111111111 10000000111 10000

En general la memoria de una fuente reduce su entropía (información):

$$H(M_1) = L \ H(X) > H(M_2)$$

M_1 : Fuente sin memoria

M_2 : Fuente con memoria



Codificación de fuente

Modelos de fuentes discretas

Estacionariedad:

Sus propiedades estadísticas no cambian en el tiempo.

```
0100101010011000110101110101001010100110010101101010111011000110101110  
10101001010100110010101101011100110101011000110101110101011000110101110101001  
0101001100101011010111001010100110010110001101011101010100101010011001011010111
```

```
01101000100011100101010100010001010010101000001000001001000010100000100000100100  
0100010000101000000001000000000100000000010010000000000000100000010000000000000000  
00000000000010000000000001000000000000000000000000000000000000000000000000001000000
```

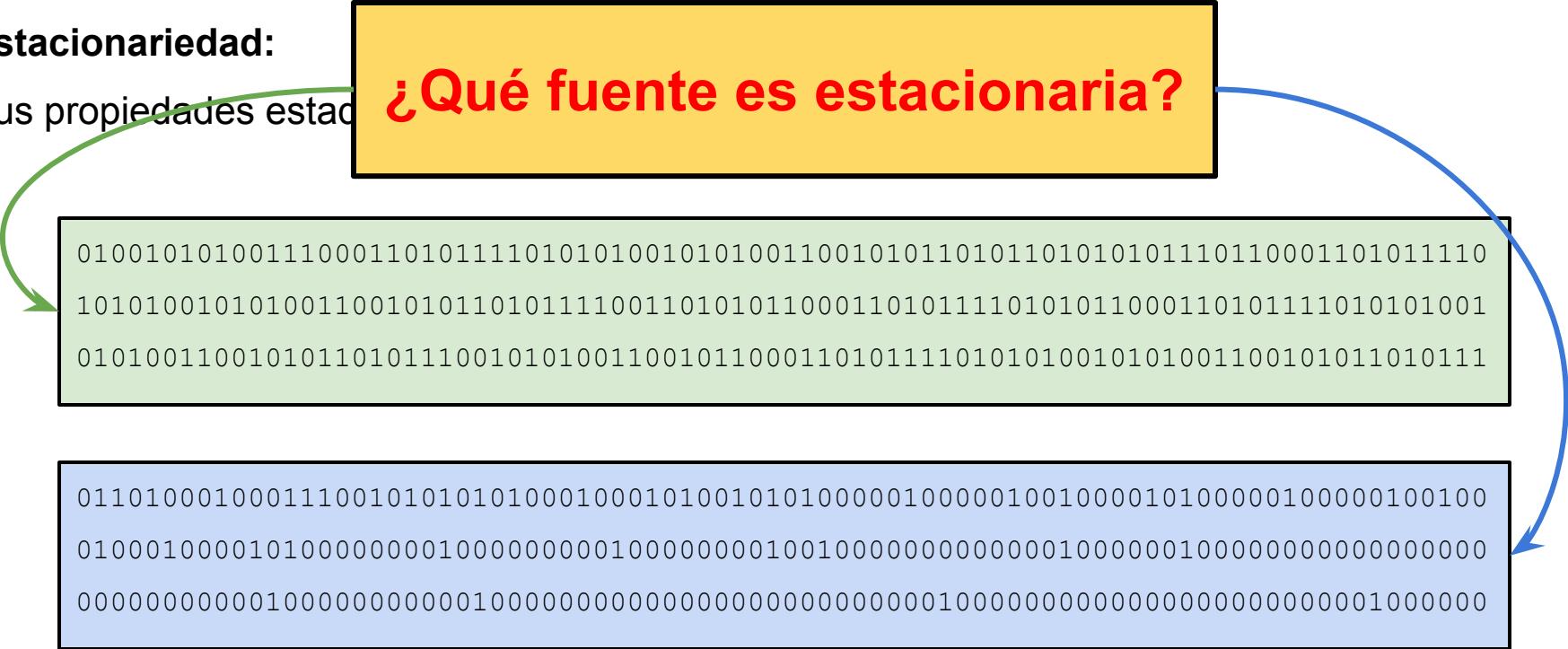
Codificación de fuente

Modelos de fuentes discretas

Estacionariedad:

Sus propiedades estadísticas no cambian con el tiempo.

¿Qué fuente es estacionaria?



```
0100101010011100011010111101010010101001100101011010101110110001101011110  
1010100101001100101011010111100110101100011010111101010110001101011110101001  
01010011001010110101110010101001100101100011010111101010100101010011001011010111
```

```
011010001000111001010101000100010100101010000100000100100001010000100000100100  
010001000010100000000100000000010000000001001000000000000100000010000000000000000  
00000000000100000000000100000000000000000000000100000000000000000000000001000000
```

Codificación de fuente

Modelos de fuentes discretas

Estacionariedad:

Sus propiedades estadísticas no cambian con el tiempo.

¿Qué fuente es estacionaria?

010010101001110001101011110101010010101001100101011010101110110001101011110
1010100101010011001010110101111001101011000110101111010101110001101011110101001
0101001100101011010111001010100110010110001101011110101010010101001100101011010111

011010001000111001010101010001000101001010100001000001001000010100000100000100100
0100010000101000000001000000000100000000010001000100000000000000000000000000000000000000
0000000000001000000000000100

Está relacionado con el concepto de **invarianza en el tiempo**.

Codificación de fuente

Códigos

RECORDEMOS:

El objetivo de la *codificación de fuente* es minimizar el número de bits necesarios para representar los mensajes (**compresión**).

IMPORTANTE:

Cuando hablamos de fuentes binarias el “**bit**” se utiliza como componente mínimo de una letra (letras compuestas por un conjunto de bits) y también como unidad para la entropía.

Codificación de fuente

Códigos

Códigos de largo variable:

Son útiles cuando las letras de la fuente no son equiprobables. La estrategia para reducir el número de bits es:

- Evaluar la probabilidad de cada letra.
- Asignar los códigos más cortos a las letras más probables y los códigos más largos a las letras menos probables.

Un ejemplo es el código Morse.

A	• -	U	• • -
B	- - -	V	• • • -
C	- - . .	W	• - -
D	- - . .	X	- - - .
E	•	Y	- - . -
F	• . - -	Z	- - - . .
G	- - - .		
H	• . . .		
I	• •		
J	• - - - -		
K	- . -	1	• - - - - -
L	• - . .	2	• - - - - -
M	- -	3	• - - - - -
N	- .	4	• - - - - -
O	- - - -	5	• - - - - -
P	• - - .	6	• - - - - -
Q	- - - - .	7	• - - - - -
R	• - . .	8	• - - - - -
S	• . .	9	• - - - - -
T	-	0	• - - - - -

Codificación de fuente

Códigos

Códigos de largo variable:

Otro ejemplo es:

$$a_1 = \{00\} \text{ con } p_1 = \frac{1}{2}$$

$$a_2 = \{01\} \text{ con } p_2 = \frac{1}{4}$$

$$a_3 = \{10\} \text{ con } p_3 = \frac{1}{8}$$

$$a_4 = \{11\} \text{ con } p_4 = \frac{1}{8}$$

Definiciones:

largo promedio : L = [bit]

tasa de bits : R = [bit/s]

L : Largo promedio en bits de un mensaje o letra, se calcula como su esperanza.

R : Depende de la cantidad de mensajes o letras que se quieran transmitir por segundo. Suele ser un valor restrictivo que depende del canal (se retoma más adelante).

Codificación de fuente

Códigos

Códigos de largo variable:

Otro ejemplo es:

$$L = \frac{1}{2} \text{ (2 bit)} + \frac{1}{4} \text{ (2 bit)} + \frac{1}{8} \text{ (2 bit)} + \frac{1}{8} \text{ (2 bit)} = 2 \text{ bit}$$

$$a_1 = \{00\} \text{ con } p_1 = \frac{1}{2}$$

$$a_2 = \{01\} \text{ con } p_2 = \frac{1}{4}$$

$$a_3 = \{10\} \text{ con } p_3 = \frac{1}{8}$$

$$a_4 = \{11\} \text{ con } p_4 = \frac{1}{8}$$

Necesito 2 bits por cada letra que quiero transmitir.

Si tengo un mensaje $M = \{X_1, X_2, X_3, \dots, X_L\}$

Voy a necesitar $2L$ bit para transmitirlo.

Codificación de fuente

Códigos

Códigos de largo variable:

Otro ejemplo es:

$$a_1 = \{00\} \text{ con } p_1 = \frac{1}{2}$$

$$a_2 = \{01\} \text{ con } p_2 = \frac{1}{4}$$

$$a_3 = \{10\} \text{ con } p_3 = \frac{1}{8}$$

$$a_4 = \{11\} \text{ con } p_4 = \frac{1}{8}$$

Letter	P [a _k]	Code I	Code II	Code III
a ₁	$\frac{1}{2}$	1	0	0
a ₂	$\frac{1}{4}$	00	10	01
a ₃	$\frac{1}{8}$	01	110	011
a ₄	$\frac{1}{8}$	10	111	111

"Digital Communication", John G. Proakis, Masoud Salehi, 5th edition.

Largo promedio:

$$L_0 = \frac{1}{2} \text{ (2 bit)} + \frac{1}{4} \text{ (2 bit)} + \frac{1}{8} \text{ (2 bit)} + \frac{1}{8} \text{ (2 bit)} = 2 \text{ bit}$$

$$L_I = \frac{1}{2} \text{ (1 bit)} + \frac{1}{4} \text{ (2 bit)} + \frac{1}{8} \text{ (2 bit)} + \frac{1}{8} \text{ (2 bit)} = 1,5 \text{ bit}$$

$$L_{II} = \frac{1}{2} \text{ (1 bit)} + \frac{1}{4} \text{ (2 bit)} + \frac{1}{8} \text{ (3 bit)} + \frac{1}{8} \text{ (3 bit)} = 1.75 \text{ bit}$$

$$L_{III} = \frac{1}{2} \text{ (1 bit)} + \frac{1}{4} \text{ (2 bit)} + \frac{1}{8} \text{ (3 bit)} + \frac{1}{8} \text{ (3 bit)} = 1.75 \text{ bit}$$

Codificación de fuente

Códigos

Códigos de largo variable:

Otro ejemplo es:

$$a_1 = \{00\} \text{ con } p_1 = \frac{1}{2}$$

$$a_2 = \{01\} \text{ con } p_2 = \frac{1}{4}$$

$$a_3 = \{10\} \text{ con } p_3 = \frac{1}{8}$$

$$a_4 = \{11\} \text{ con } p_4 = \frac{1}{8}$$

Letter	P [a _k]	Code I	Code II	Code III
a ₁	$\frac{1}{2}$	1	0	0
a ₂	$\frac{1}{4}$	00	10	01
a ₃	$\frac{1}{8}$	01	110	011
a ₄	$\frac{1}{8}$	10	111	111

"Digital Communication", John G. Proakis, Masoud Salehi, 5th edition.

Códigos de decodificación única e instantáneos: Supongamos S={001001...}

$$C_0 : \text{Si,Si} \rightarrow M=\{a_1, a_3, a_2\}$$

$$C_I : \text{No,No} \rightarrow M=\{a_2, a_1, a_2, a_1\} \text{ o } M=\{a_2, a_4, a_3\}$$

$$C_{II} : \text{Si,Si} \rightarrow M=\{a_1, a_1, a_2, a_1, \dots\}, \text{ el 0 indica el fin de la letra o cuando se llega a 3 bits}$$

$$C_{III} : \text{Si,No} \rightarrow M=\{a_1, a_2, a_1, a_2\}$$

Codificación de fuente

Códigos

Códigos de largo variable:

Otro ejemplo es:

$$a_1 = \{00\} \text{ con } p_1 = \frac{1}{2}$$

$$a_2 = \{01\} \text{ con } p_2 = \frac{1}{4}$$

$$a_3 = \{10\} \text{ con } p_3 = \frac{1}{8}$$

$$a_4 = \{11\} \text{ con } p_4 = \frac{1}{8}$$

Letter	P [a _k]	Code I	Code II	Code III
a ₁	$\frac{1}{2}$	1	0	0
a ₂	$\frac{1}{4}$	00	10	01
a ₃	$\frac{1}{8}$	01	110	011
a ₄	$\frac{1}{8}$	10	111	111

"Digital Communication", John G. Proakis, Masoud Salehi, 5th edition.

Códigos de decodificación única e instantáneos: Supongamos S={001001...}

$$C_0 : \text{Si,Si} \rightarrow M=\{a_1, a_3, a_2\}$$

$$C_I : \text{No,No} \rightarrow M=\{a_2, a_1, a_2, a_1\} \text{ o } M=\{a_2, a_4, a_3\}$$

$$C_{II} : \text{Si,Si} \rightarrow M=\{a_1, a_1, a_2, a_1, \dots\}, \text{ el } 0 \text{ indica el}$$

$$C_{III} : \text{Si,No} \rightarrow M=\{a_1, a_2, a_1, a_2\}$$

Condición suficiente para ser de
decodificación única e instantáneo:
Codewords libres de prefijo

Codificación de fuente

Códigos

Códigos de largo variable:

Resumen de propiedades

Propiedad	Código 0	Código I	Código II	Código III
largo promedio	2 bit	1,5 bit	1,75 bit	1,75 bit
decod. única	Si	No	Si	Si
instantáneo	Si	No	Si	No

Codificación de fuente

Códigos

Códigos de largo variable:

Resumen de propiedades

Propiedad	Código 0	Código I	Código II	Código III
largo promedio	2 bit	1,5 bit	1,75 bit	1,75 bit
decod. única	Si	No	Si	Si
instantáneo	Si	No	Si	No

Solo queremos códigos de **decodificación única e instantáneos** y lo más **cortos** posible.

¿Hay algo mejor que el código II?

Codificación de fuente

Códigos

Teorema de la codificación de fuente:

SHANNON'S FIRST THEOREM (LOSSLESS SOURCE CODING THEOREM) Let X denote a DMS with entropy $H(X)$. There exists a lossless source code for this source at any rate R if $R > H(X)$. There exists no lossless code for this source at rates less than $H(X)$.

Codificación de fuente

Códigos

Teorema de la codificación de fuente:

Para una fuente discreta sin memoria, es posible construir un código que cumpla con:

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

En el ejemplo anterior:

$$H(X) = -(\frac{1}{2} \log(\frac{1}{2}) + \frac{1}{4} \log(\frac{1}{4}) + \frac{1}{8} \log(\frac{1}{8}) + \frac{1}{8} \log(\frac{1}{8})) = 1,75 \text{ bit}$$

Entonces el código 2 alcanza el mejor valor posible para el largo promedio.

Definiciones:

Eficiencia : $\eta = H(X)/L$ = 1 (para el ejemplo anterior)

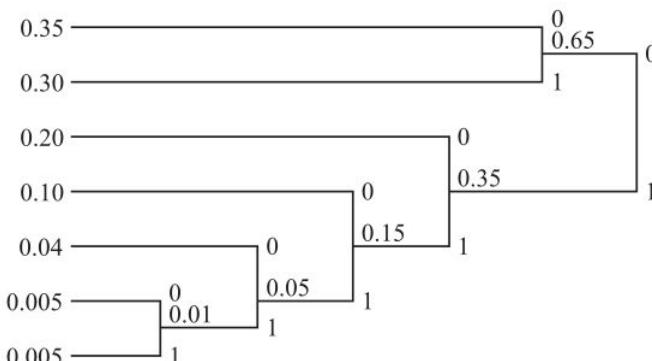
Redundancia : $\gamma = 1 - \eta$ = 0 (para el ejemplo anterior)

Codificación de fuente

Códigos

Codificación de Huffman:

Procedimiento: Supongamos que tenemos los símbolos x_1 a x_7 , se construye el código agrupando de a 2 los 2 símbolos menos probables:



Letter	Probability	Self-information	Code
x_1	0.35	1.5146	00
x_2	0.30	1.7370	01
x_3	0.20	2.3219	10
x_4	0.10	3.3219	110
x_5	0.04	4.6439	1110
x_6	0.005	7.6439	11110
x_7	0.005	7.6439	11111

$$H(X) = 2.11$$

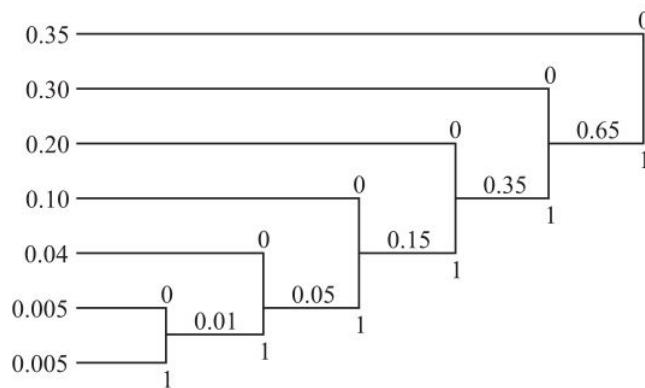
$$\bar{R} = 2.21$$

Codificación de fuente

Códigos

Codificación de Huffman:

La codificación no es única:



Letter	Code
x_1	0
x_2	10
x_3	110
x_4	1110
x_5	11110
x_6	111110
x_7	111111

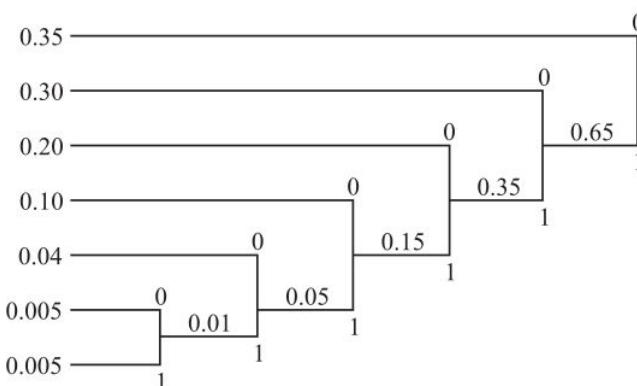
$$\bar{R} = 2.21$$

Codificación de fuente

Códigos

Codificación de Huffman:

La codificación no es única:



Letter	Code	Code
x_1	0	00
x_2	10	01
x_3	110	10
x_4	1110	110
x_5	11110	1110
x_6	111110	11110
x_7	111111	11111

$$\bar{R} = 2.21$$

$$\bar{R} = 2.21$$

$$H(X) = 2,11$$

$$L = 2,21$$

$$\eta = H(X)/L = 0.95475$$

Codificación de fuente

Códigos

Codificación de Huffman:

Alcanza la eficiencia de manera asintótica, es decir codificando por bloques de símbolos o letras.

$$\boxed{\begin{aligned} N &\rightarrow \infty \\ L &\rightarrow H(X) \\ \eta &= H(X)/L \rightarrow 1 \\ \gamma &\rightarrow 0 \end{aligned}}$$

Letter	Probability	Self-information	Code
x_1	0.45	1.156	1
x_2	0.35	1.520	00
x_3	0.20	2.330	01
$H(X) = 1.513 \text{ bits/letter}$			
$\bar{R}_1 = 1.55 \text{ bits/letter}$			
Efficiency = 97.6%			

Letter pair	Probability	Self-information	Code
x_1x_1	0.2025	2.312	10
x_1x_2	0.1575	2.676	001
x_2x_1	0.1575	2.676	010
x_2x_2	0.1225	3.039	011
x_1x_3	0.09	3.486	111
x_3x_1	0.09	3.486	0000
x_2x_3	0.07	3.850	0001
x_3x_2	0.07	3.850	1100
x_3x_3	0.04	4.660	1101
$2H(X) = 3.026 \text{ bits/letter pair}$			
$\bar{R}_2 = 3.0675 \text{ bits/letter pair}$			
$\frac{1}{2}\bar{R}_2 = 1.534 \text{ bits/letter}$			
Efficiency = 98.6%			

Codificación de fuente

Códigos

Codificación Lempel-Ziv:

Empecemos viendo una mala propiedad de la codificación de Huffman:

Para hallar el código es necesario conocer a la perfección las probabilidades de cada símbolo y de cada conjunto de símbolos.

Esto es impráctico en la mayoría de las ocasiones.

La codificación de Lempel-Ziv es una **codificación de fuentes universales**, es decir que sirve para todo tipo de fuente.

Es una codificación basada en **diccionario adaptativo**.

Hay dos grupos: "**ventana deslizante**" y "**estructurado en forma de árbol**".

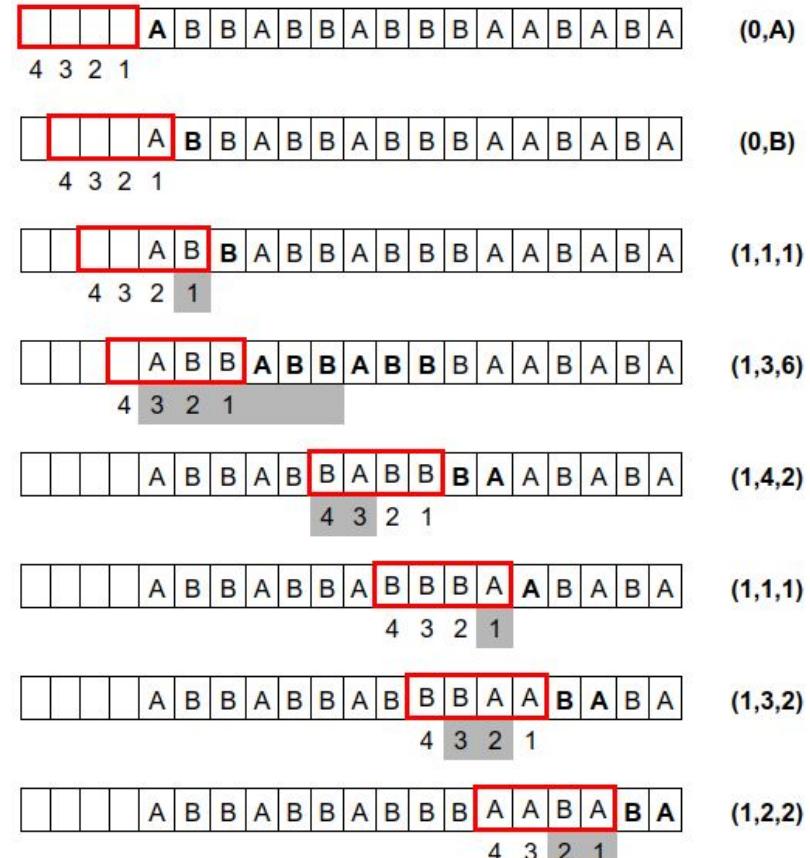
Codificación de fuente

Códigos

Codificación Lempel-Ziv:

Ejemplo de codificación Lempel-Ziv del tipo “ventana deslizante”.

- Se buscan cadenas repetidas en una ventana.
- Se codifica con (F,C) o (F,P,L)
 - F : Flag, indicando ocurrencia
 - C : Carácter
 - P : Posición (contada desde atrás)
 - L : Largo
- Se utiliza en gzip y pkzip.
- Una variación de éste es óptimo asintóticamente.



Codificación de fuente

Códigos

10101101001001110101000011001110101100011011

Codificación Lempel-Ziv:

1, 0, 10, 11, 01, 00, 100, 111, 010, 1000, 011, 001, 110, 101, 10001, 1011

Ejemplo de codificación Lempel-Ziv del tipo

“estructurada en forma de árbol”:

- Códigos de largo fijo
- Se arma un diccionario dinámicamente
- [posición] + [último bit]
- La tabla puede crecer indefinidamente
 - Se limita el largo
 - Se utiliza una política de reemplazo
- Se utiliza en el formato gif
- Es asintóticamente óptimo

	Dictionary location	Dictionary contents	Code word
1	0001	1	00001
2	0010	0	00000
3	0011	10	00010
4	0100	11	00011
5	0101	01	00101
6	0110	00	00100
7	0111	100	00110
8	1000	111	01001
9	1001	010	01010
10	1010	1000	01110
11	1011	011	01011
12	1100	001	01101
13	1101	110	01000
14	1110	101	00111
15	1111	10001	10101
16		1011	11101

Codificación de fuente

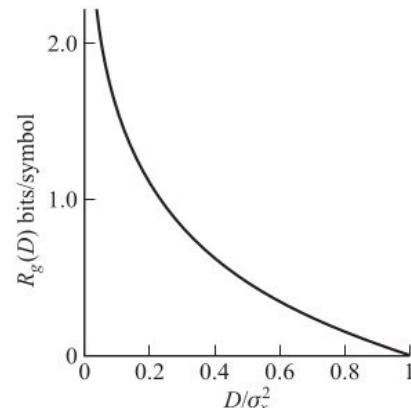
Códigos

Codificación con pérdida:

Los ejemplos anteriores son capaces de recuperar el mensaje original.

Hay otros códigos que reducen más los datos, pero no son capaces de recuperar el mensaje original, aunque sí sus partes más “importantes”.

La performance de estos algoritmos se mide con la curva de “*Rate-Distortion*”.



Codificación de fuente

Resumen

Características deseables de los códigos de fuente:

- Decodificable de manera **única, instantánea** (o que cumpla con la **condición de prefijo**)
- Obtener **secuencia de símbolos sin redundancia**:
 - Una secuencia de símbolos de largo mínimo
 - Una “nueva fuente” con máxima entropía
 - Una fuente que no se puede comprimir
 - Eficiencia $\eta=1$
- Que codifique **fuentes universales**:
 - Que no sea necesario conocer previamente las estadísticas de las fuentes

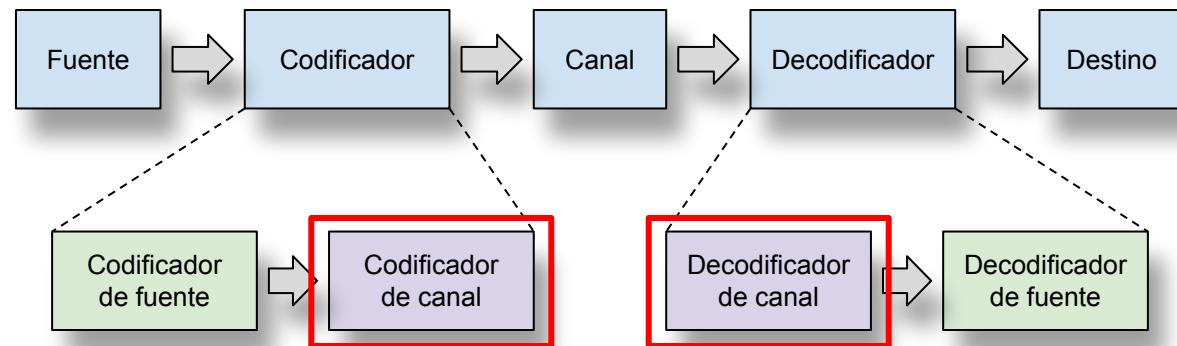
Codificación de canal:

- **Canal**

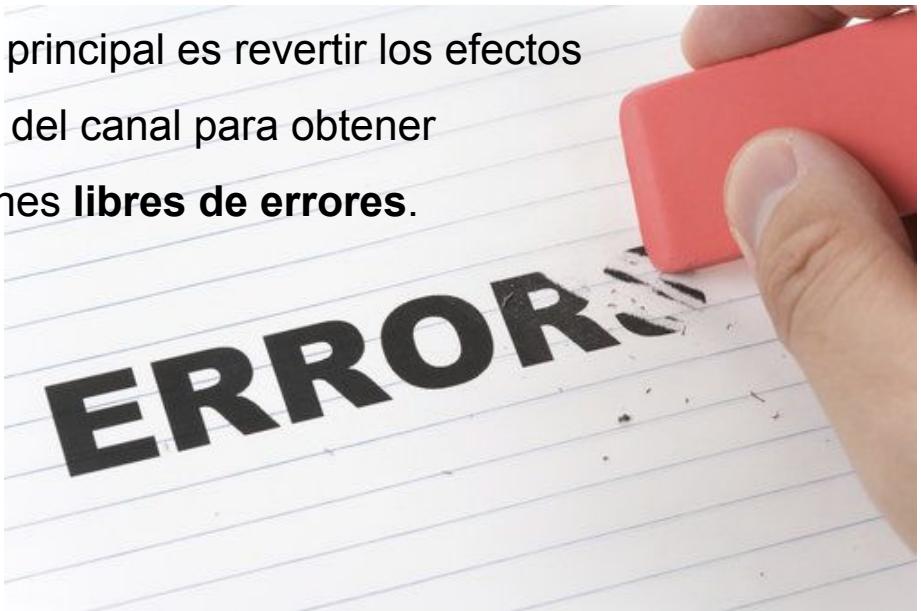
- Modelos
- Capacidad

- **Codificación**

- Detección de errores
- Corrección de errores
- Cod. por bloque
- Cod. convolucional
- Algoritmo Viterbi
- Entrelazado
- Cod. de línea

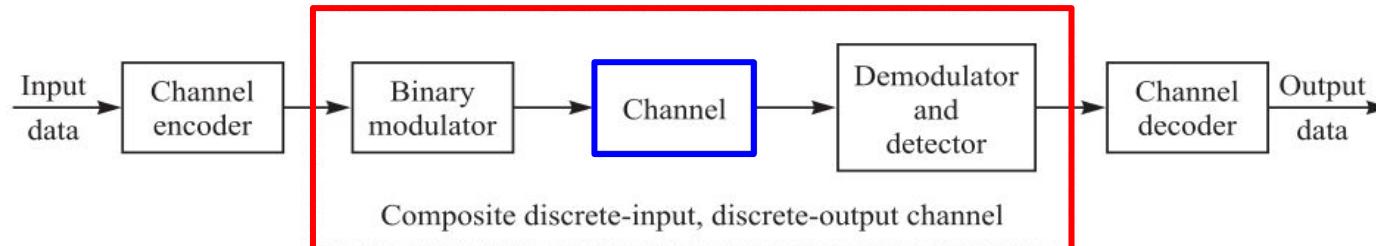


El objetivo principal es revertir los efectos distorsivos del canal para obtener transmisiones **libres de errores**.



Codificación de canal

Modelos de canal

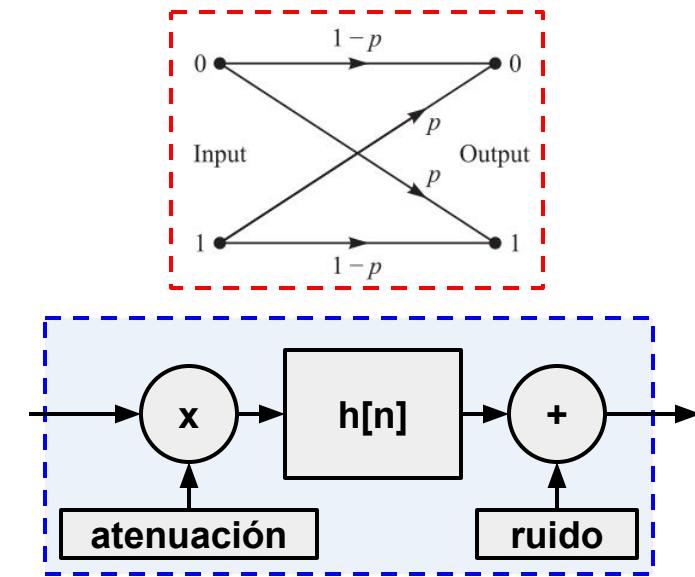


Fuente: "Digital Communication", John G. Proakis, Masoud Salehi, 5th edition.

Desde el punto de vista de la **teoría de información**:

- Canal de **entradas y salidas discretas**
- Recibe y devuelve **símbolos**
- Se caracterizan a través de $P(Y | X)$.

Nota: Es distinto al canal y los modelos utilizados en la **capa física** de los sistemas de comunicaciones, donde el canal es **continuo**.



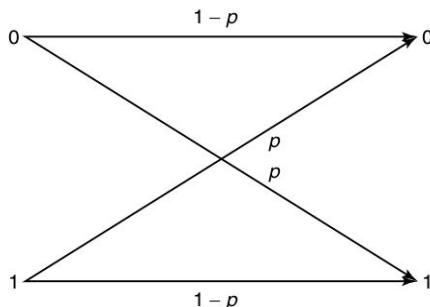
Codificación de canal

Modelos de canal

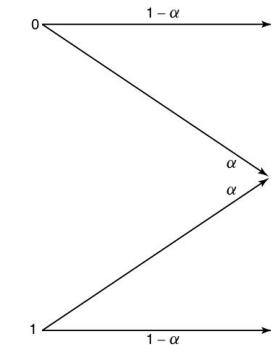
Canal binario sin ruido



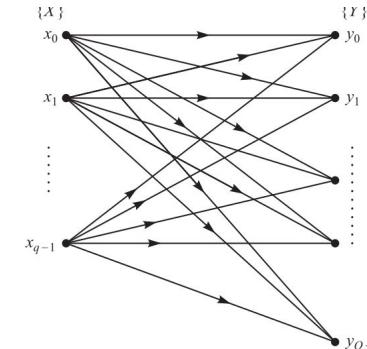
Canal binario simétrico (BSC)



Canal binario + detección de error



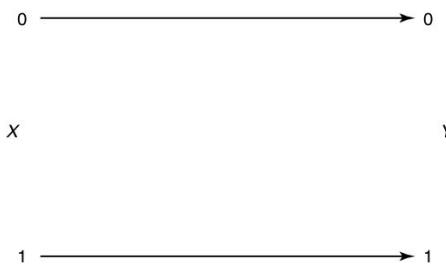
Canal genérico sin memoria



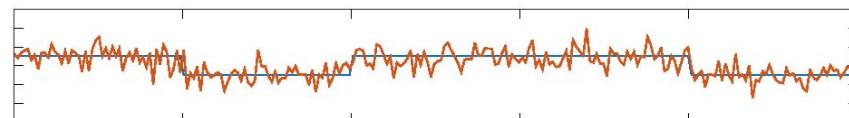
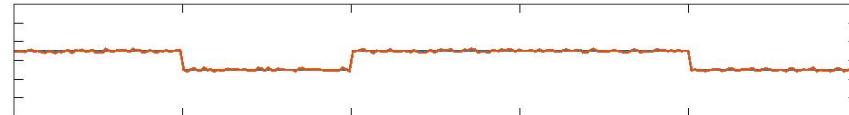
Codificación de canal

Modelos de canal

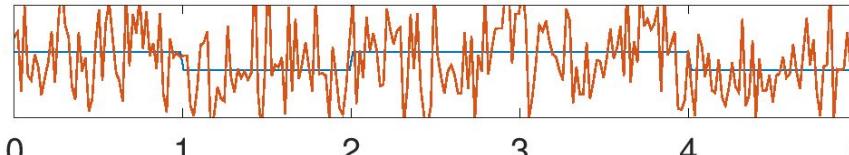
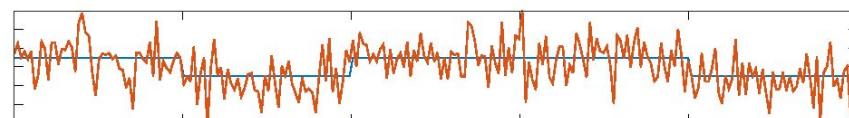
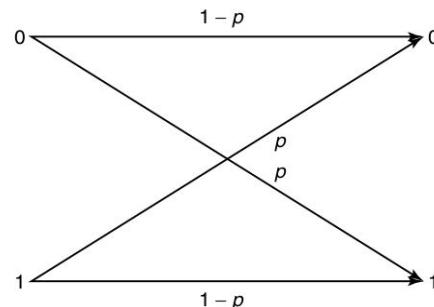
Canal binario sin ruido



Canal continuo en la capa física



Canal binario simétrico (BSC)



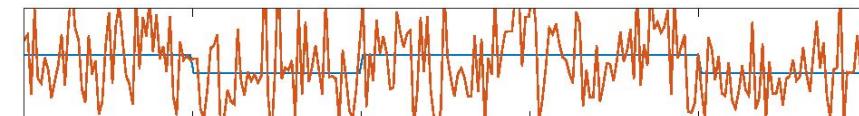
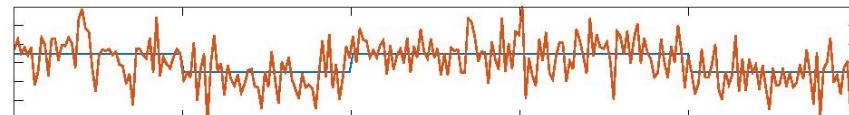
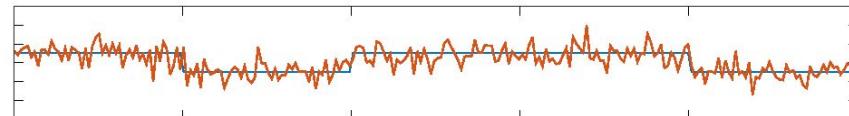
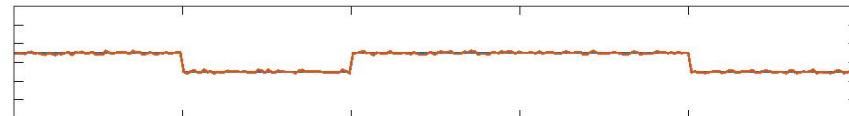
Codificación de canal

Modelos de canal

¿Es posible enviar mensajes **sin errores** a través de un **canal con ruido**?

Durante mucho tiempo, se creyó que esto no era posible, que solo era posible reducir la probabilidad de error aumentando la relación que hay entre la señal y el ruido.

Canal continuo en la capa física



0 1 2 3 4 5

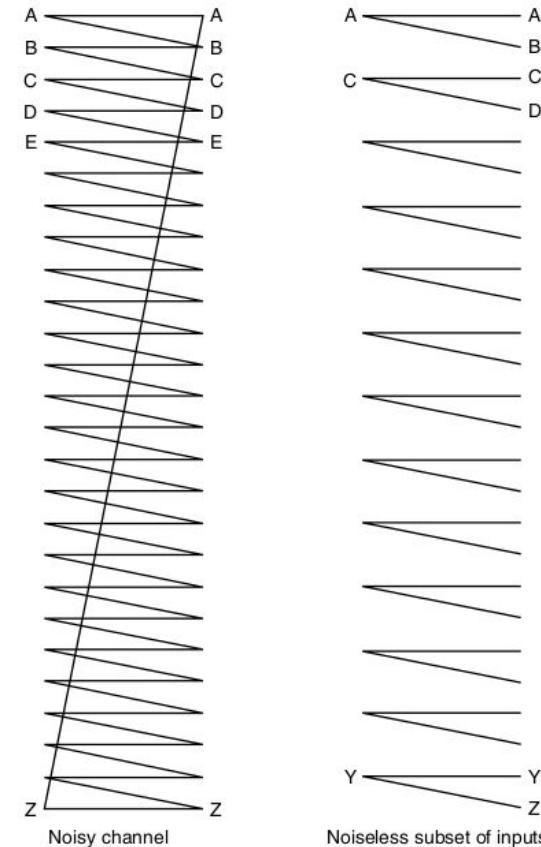
Codificación de canal

Modelos de canal

¿Es posible enviar mensajes **sin errores** a través de un **canal con ruido**?

Durante mucho tiempo, se creyó que esto no era posible, que solo era posible reducir la probabilidad de error aumentando la relación que hay entre la señal y el ruido.

Se demuestra que sí es posible.



Codificación de canal

Capacidad del canal

Capacidad del canal: Concepto introducido por Shannon:

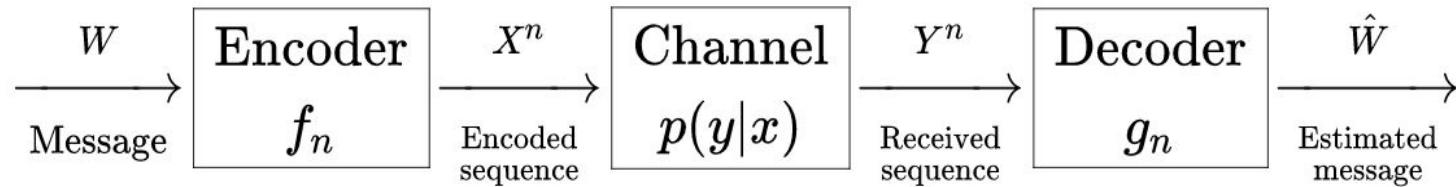
SHANNON'S SECOND THEOREM—THE NOISY CHANNEL CODING THEOREM (SHANNON 1948)

Reliable communication over a discrete memoryless channel is possible if the communication rate R satisfies $R < C$, where C is the channel capacity. At rates higher than capacity, reliable communication is impossible.

Codificación de canal

Capacidad del canal

Capacidad del canal: Concepto introducido por Shannon:



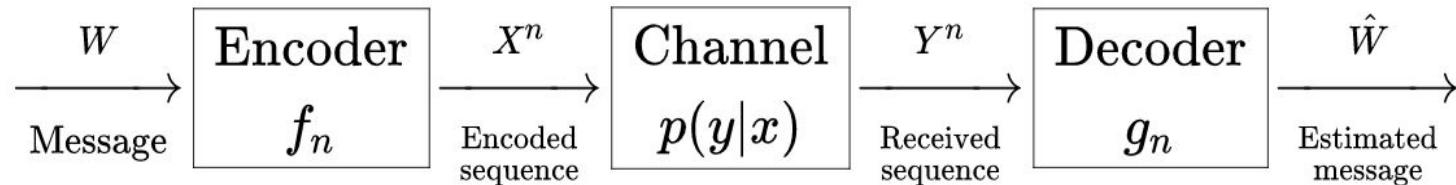
$$C = \sup_{p_X(x)} I(X; Y)$$

where the **supremum** is taken over all possible choices of $p_X(x)$.

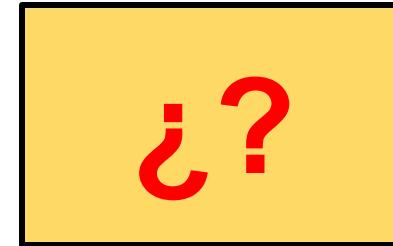
Codificación de canal

Capacidad del canal

Capacidad del canal: Concepto introducido por Shannon:



$$C = \sup_{p_X(x)} I(X; Y)$$

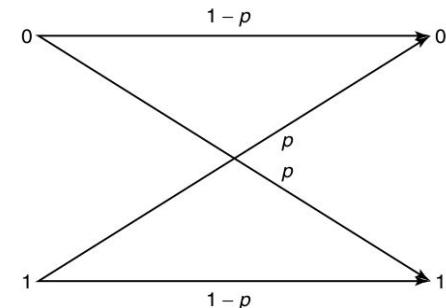
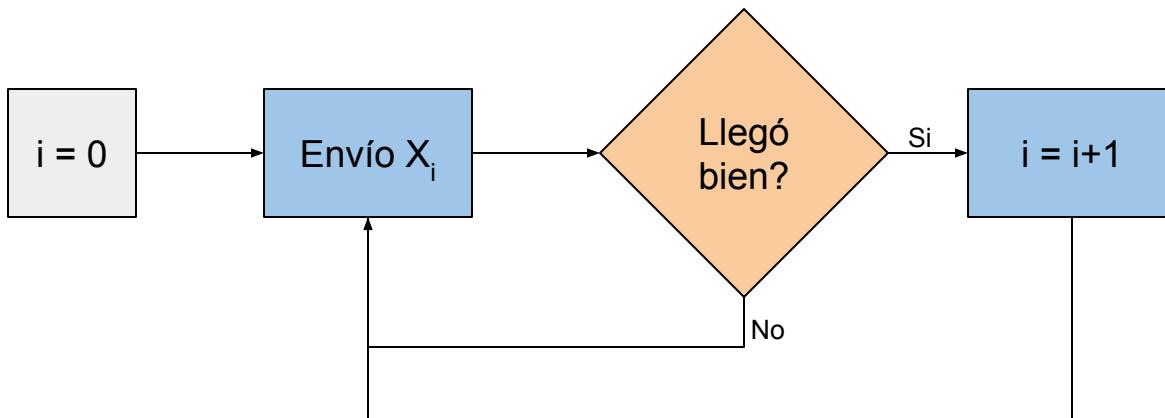


where the **supremum** is taken over all possible choices of $p_X(x)$.

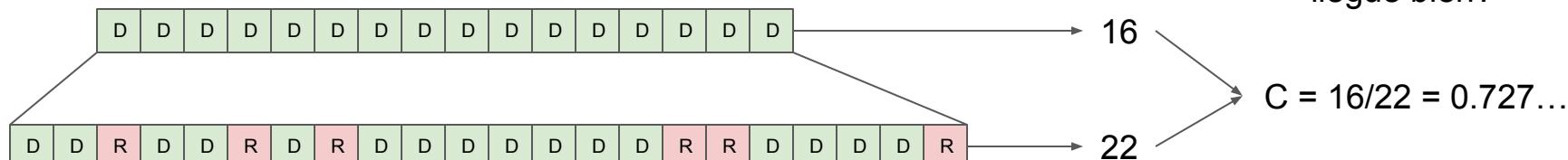
Codificación de canal

Capacidad del canal

Capacidad del canal: Un ejemplo intuitivo.



¿Cuántas veces tengo que mandar un dato en promedio hasta que llegue bien?



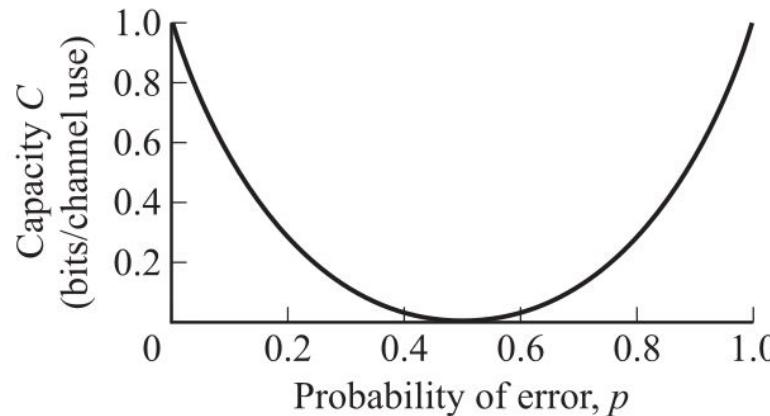
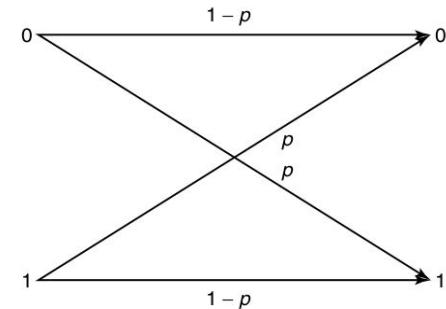
Codificación de canal

Capacidad del canal

Capacidad del canal: Ejemplo concreto.

Capacidad del canal para el canal BSC (discreto):

$$C = 1 + p \log 2p + (1 - p) \log 2(1 - p) = 1 - H(p)$$



Unidad de medida:
[bits/uso del canal]

Codificación de canal

Capacidad del canal

Capacidad del canal

Capacidad del canal para el canal AWGN (contínuo):

Unidad de medida:
[bits/s]

$$C = W \log_2 \left(1 + \frac{P}{N_0 W} \right) \text{ bits/s}$$

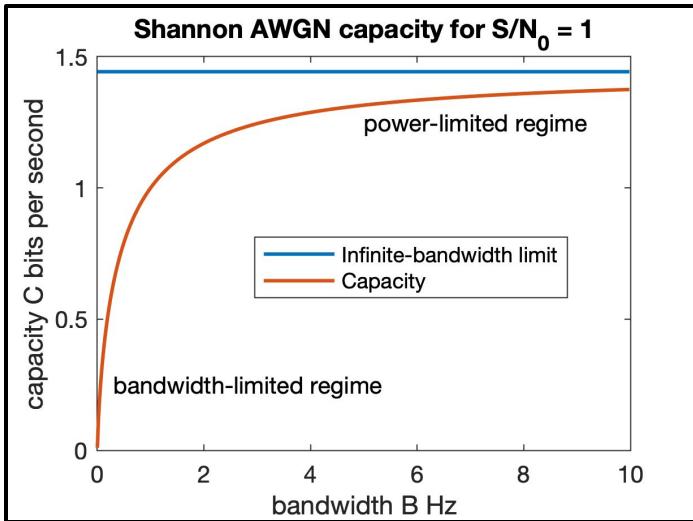
$$\begin{aligned} P &\rightarrow \infty \\ C &\rightarrow \infty \end{aligned}$$

$$C_\infty = \lim_{W \rightarrow \infty} W \log_2 \left(1 + \frac{P}{N_0 W} \right) = (\log_2 e) \frac{P}{N_0} \approx 1.44 \frac{P}{N_0} \text{ bits/s}$$

Codificación de canal

Capacidad del canal

Capacidad del canal: Algunas relaciones



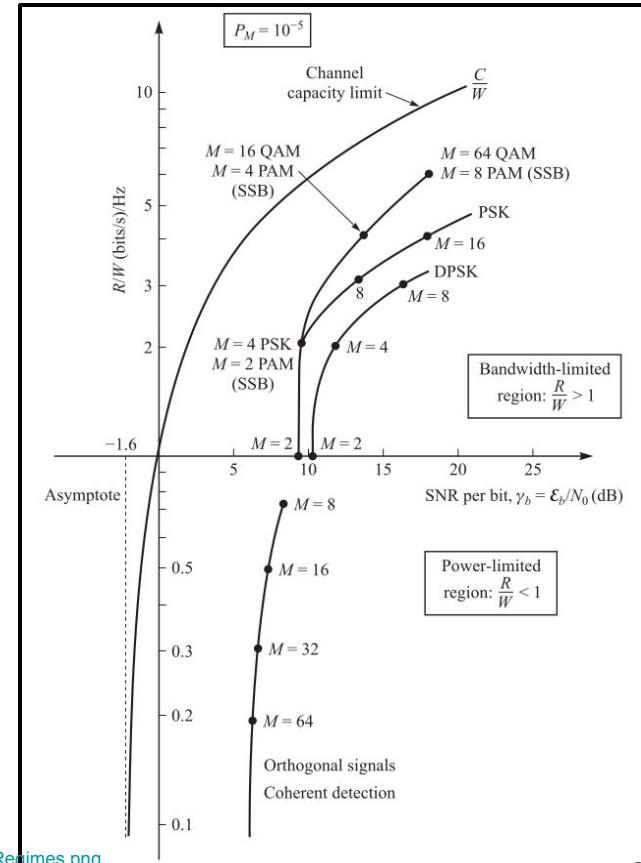
$$R < C$$

$$C_{\infty} \approx 1.44 \frac{P}{N_0} \text{ bits/s}$$

$$\begin{aligned} P &\rightarrow \infty \\ C &\rightarrow \infty \end{aligned}$$

$$C = W \log_2 \left(1 + \frac{P}{N_0 W} \right) \text{ bits/s}$$

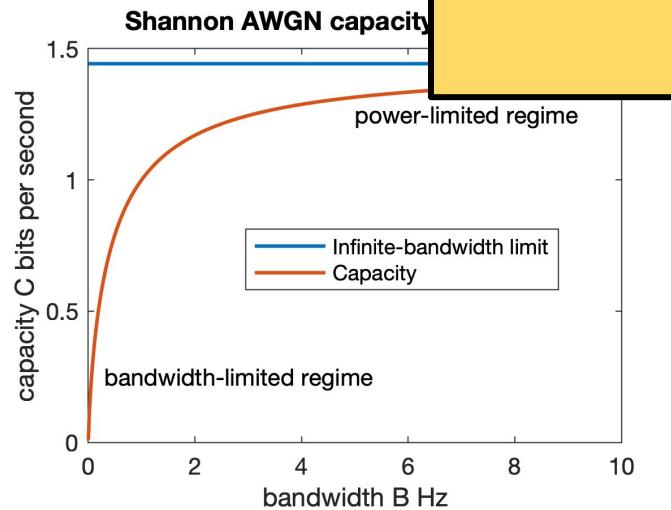
$$\frac{\mathcal{E}_b}{N_0} > \ln 2 \approx 0.693 \sim -1.6 \text{ dB}$$



Codificación de canal

Capacidad del canal

Capacidad del canal:



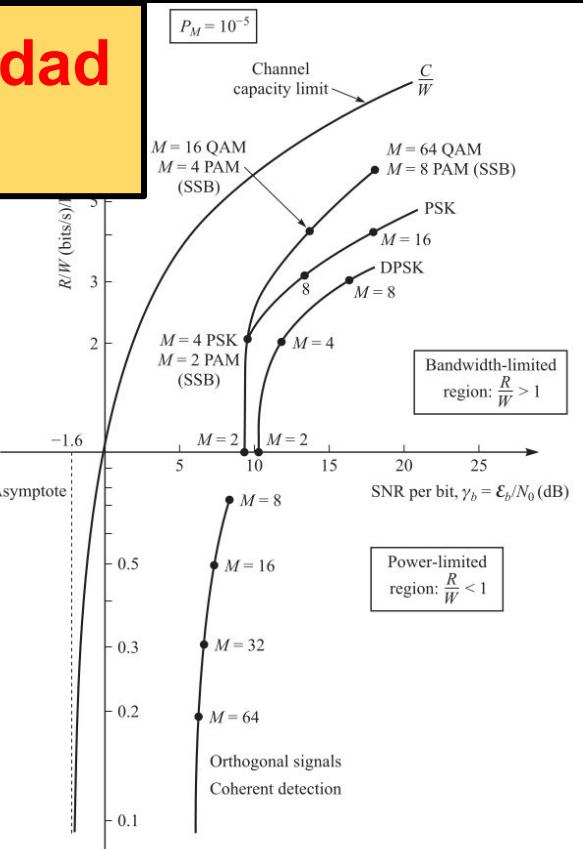
¿Cómo alcanzar la capacidad del canal?

$$C_{\infty} \approx 1.44 \frac{P}{N_0} \text{ bits/s}$$

$$\begin{matrix} P \rightarrow \infty \\ C \rightarrow \infty \end{matrix}$$

$$C = W \log_2 \left(1 + \frac{P}{N_0 W} \right) \text{ bits/s}$$

$$\frac{\mathcal{E}_b}{N_0} > \ln 2 \approx 0.693 \sim -1.6 \text{ dB}$$



Codificación de canal

Códigos

Códigos de detección de error:

Paridad:

- Detecta errores impares
- Simple

Cyclic Redundancy Check (CRC):

- Detecta distintos tipos de errores
- Simple

Agrenan **redundancia**, existe una **trade-off** entre el **overhead**, la capacidad de **detección y penalización**. Necesitan **realimentación** entre receptor y transmisor para corregir errores.

7 bits of data	(count of 1-bits)	8 bits including parity	
		even	odd
0000000	0	00000000	10000000
1010001	3	11010001	01010001
1101001	4	01101001	11101001
1111111	7	11111111	01111111

Fuente: https://en.wikipedia.org/wiki/Parity_bit

101101110000

1101

011001110000

1101

000011110000

1101

000000100000

1101

000000010100

1101

000000001110

1101

000000000011

1101

Codificación de canal

Códigos

Códigos de corrección de error:

No solo detectan, sino que son capaces de corregir algunos de los errores que aparecen sin la necesidad de una retransmisión de datos (realimentación entre receptor y transmisor).

Al igual que antes:

- El código agrega **redundancia**
- Existe una **trade-off** entre el **overhead**, la capacidad de **corrección**

Hay dos tipos de códigos que se utilizan:

- Códigos por **bloques lineales**
- Códigos **convolucionales** (lineales)

Caracterizamos un código según:

- Su capacidad de corrección
- Su eficiencia o tasa $\eta = k/n$
- La complejidad computacional/espacial para la codificación y decodificación

Codificación de canal

Códigos

Códigos por bloques lineales:

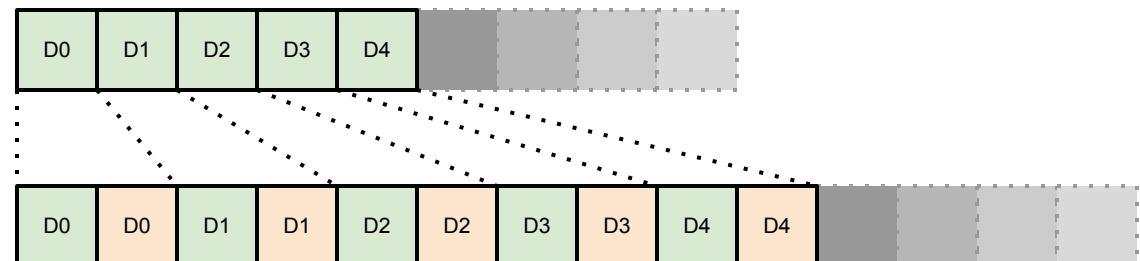
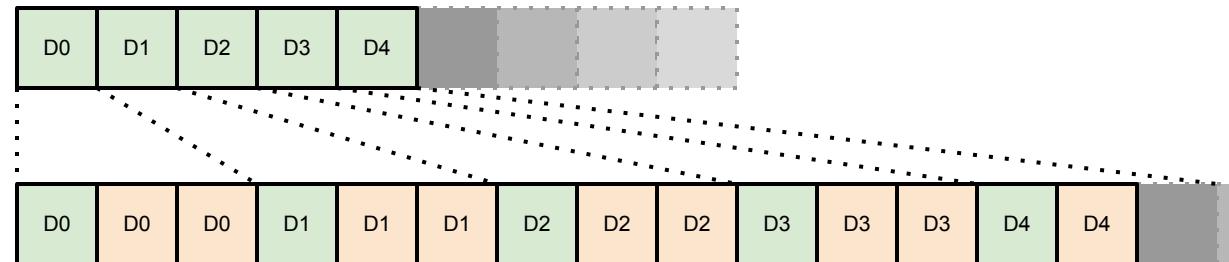
Códigos de repetición:

Se repite el dato n veces.

Su tasa es de 1/n.

Si n es impar puede corregir hasta $(n-1)/2$ errores.

Si n es par puede corregir hasta $(n-2)/2$ errores.



Codificación de canal

Códigos

Códigos por bloques lineales:

Otros códigos conocidos

- Códigos Reed-Muller
- Códigos Hardamard
- Códigos Golay
- Códigos Reed-Solomon
- Códigos Low-Density Parity-Check (LDPC)
- Códigos Cíclicos

Codificación de canal

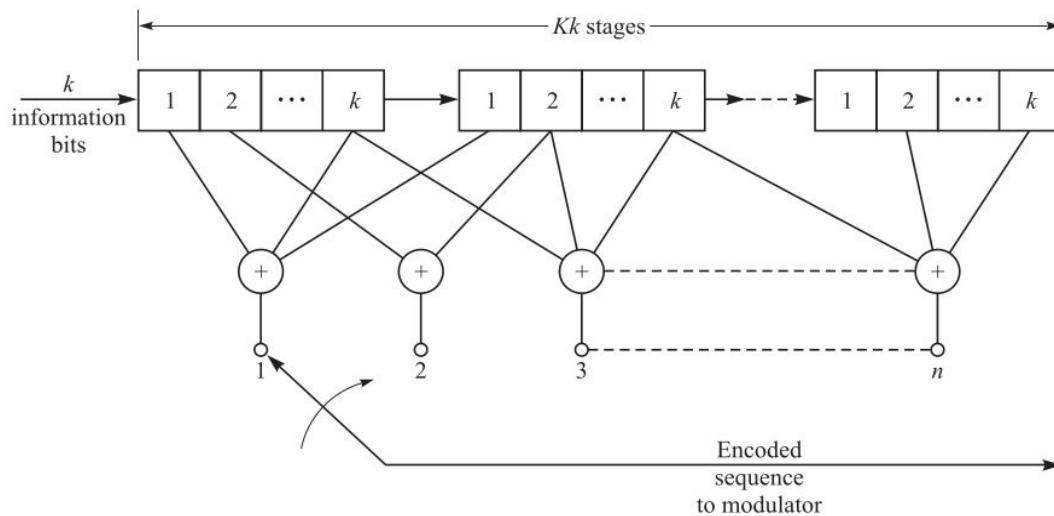
Códigos

Códigos convolucionales:

K : etapas

k : bits de datos

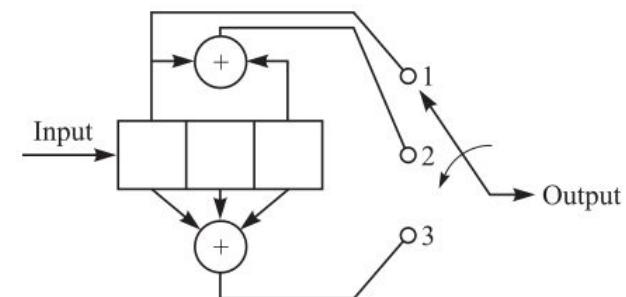
n : bits de código



$K = 3$

$k = 1$

$n = 3$



Codificación de canal

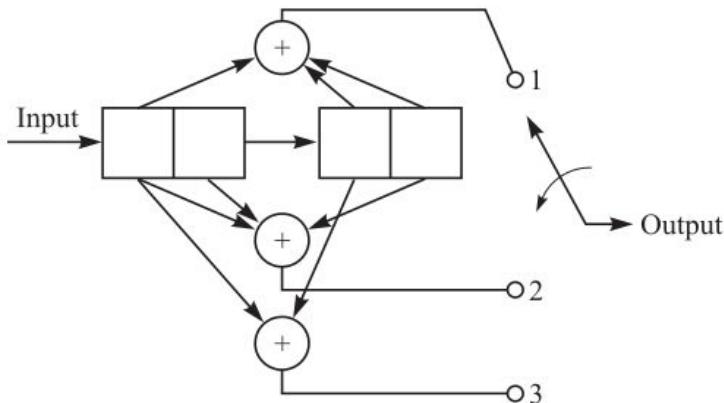
Códigos

Códigos convolucionales:

$$K = 2$$

$$k = 2$$

$$n = 3$$

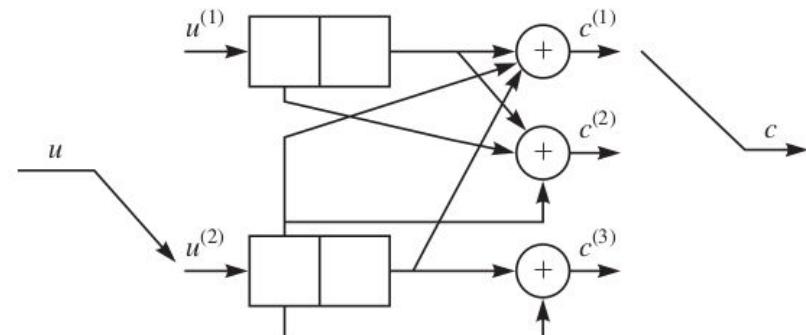


Implementación 1

$$K = 2$$

$$k = 2$$

$$n = 3$$



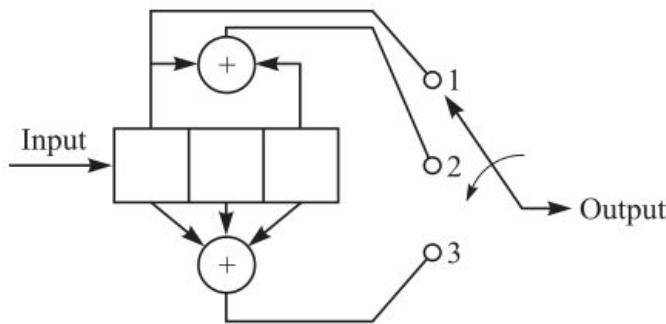
Implementación 2

Codificación de canal

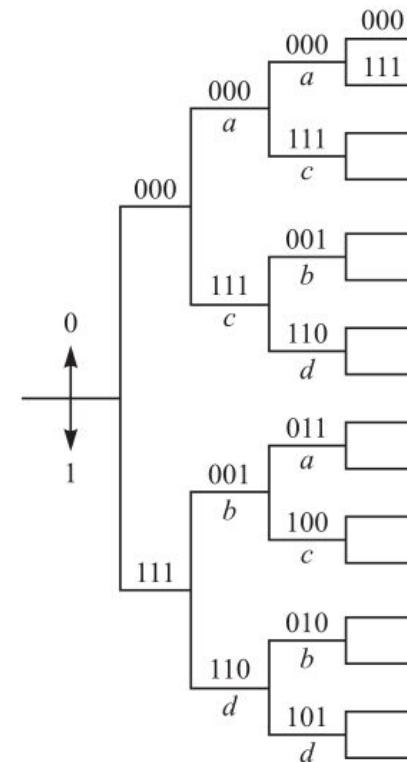
Códigos

Códigos convolucionales:

$$K = 3, k = 1, n = 3$$



Representación en forma de árbol

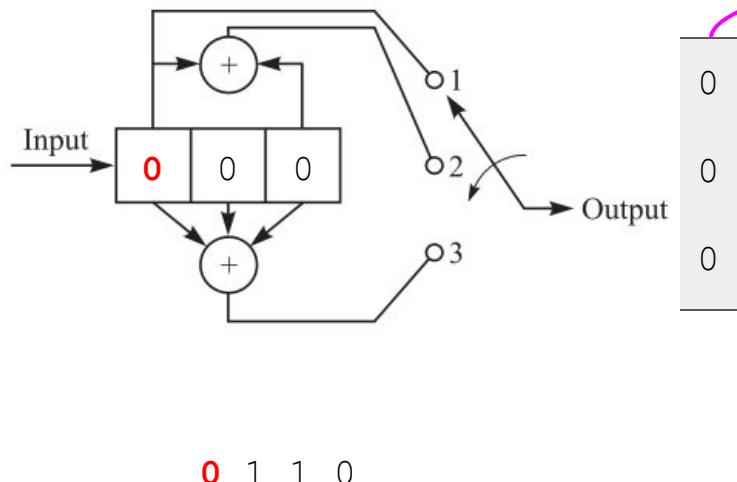


Codificación de canal

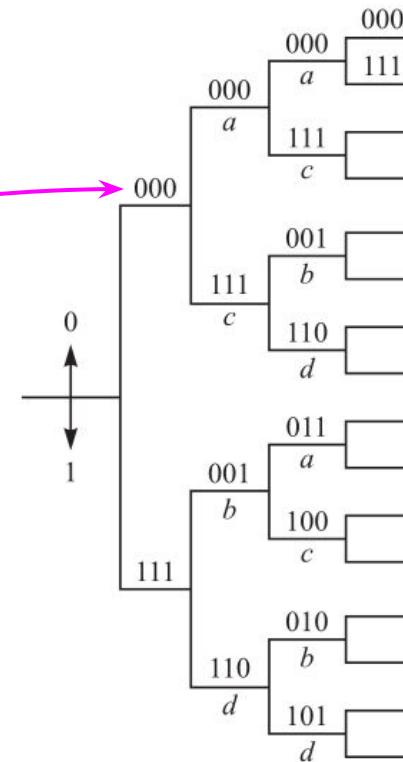
Códigos

Códigos convolucionales:

$K = 3, k = 1, n = 3$



Representación en forma de árbol

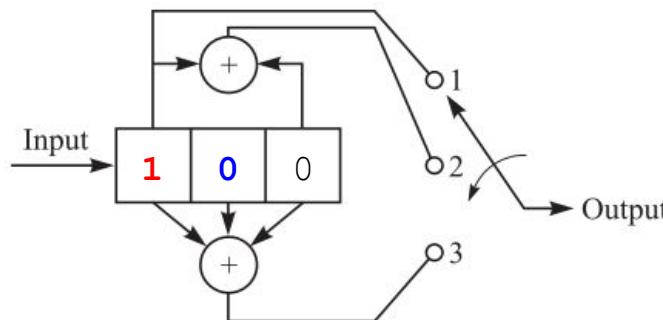


Codificación de canal

Códigos

Códigos convolucionales:

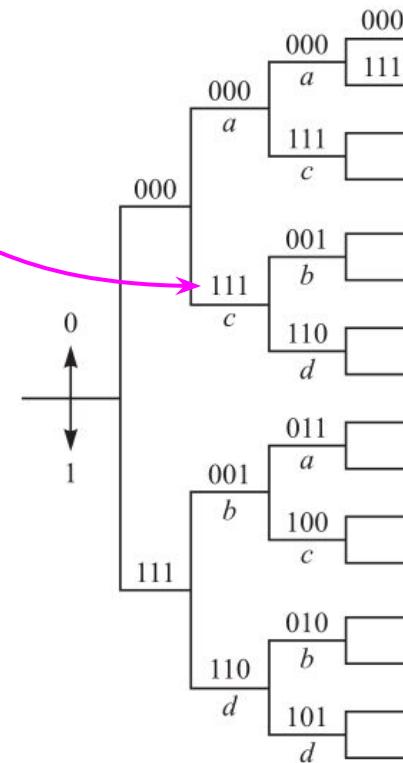
$K = 3, k = 1, n = 3$



$0 \ 1 \ 1 \ 0$

0
1
1
0

Representación en forma de árbol

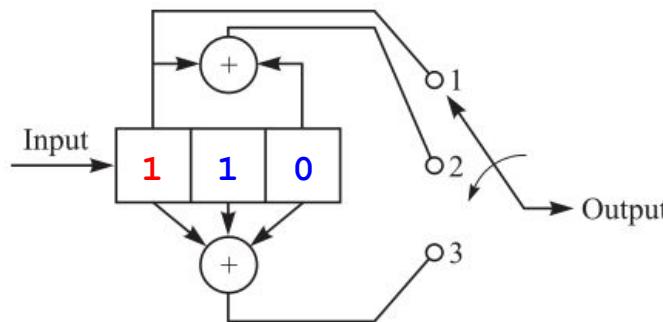


Codificación de canal

Códigos

Códigos convolucionales:

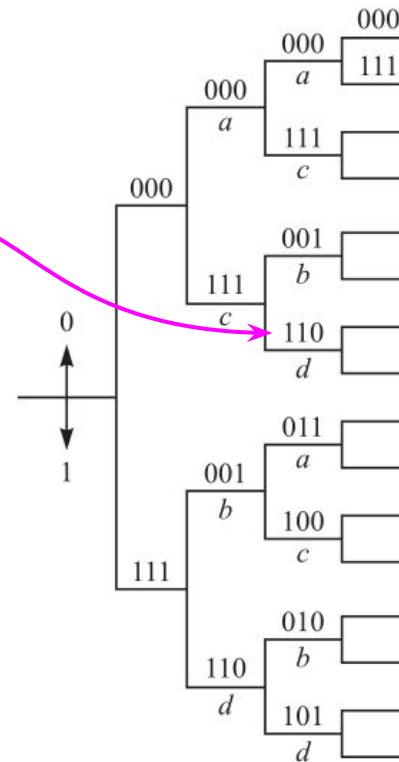
$K = 3, k = 1, n = 3$



$0 \ 1 \ 1 \ 0$

0 1 1
0 1 1
0 1 0

Representación en forma de árbol

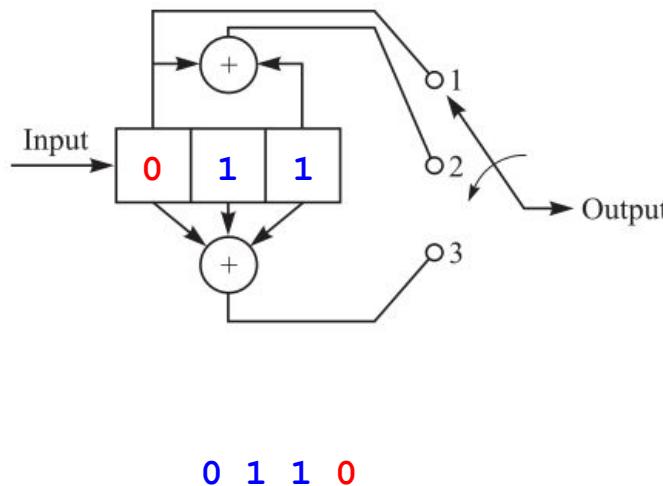


Codificación de canal

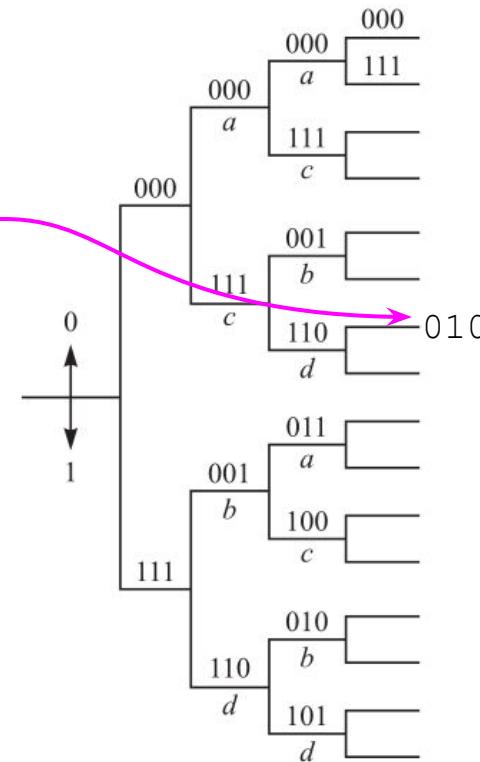
Códigos

Códigos convolucionales:

$K = 3, k = 1, n = 3$



Representación en forma de árbol

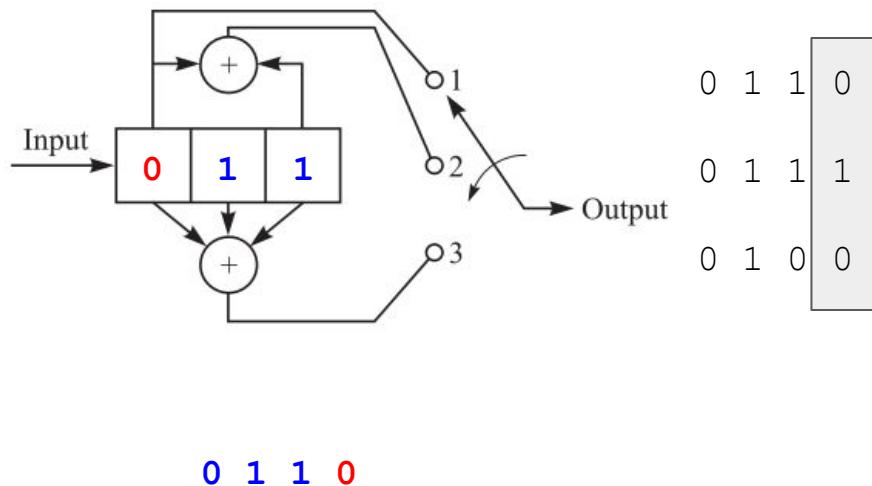


Codificación de canal

Códigos

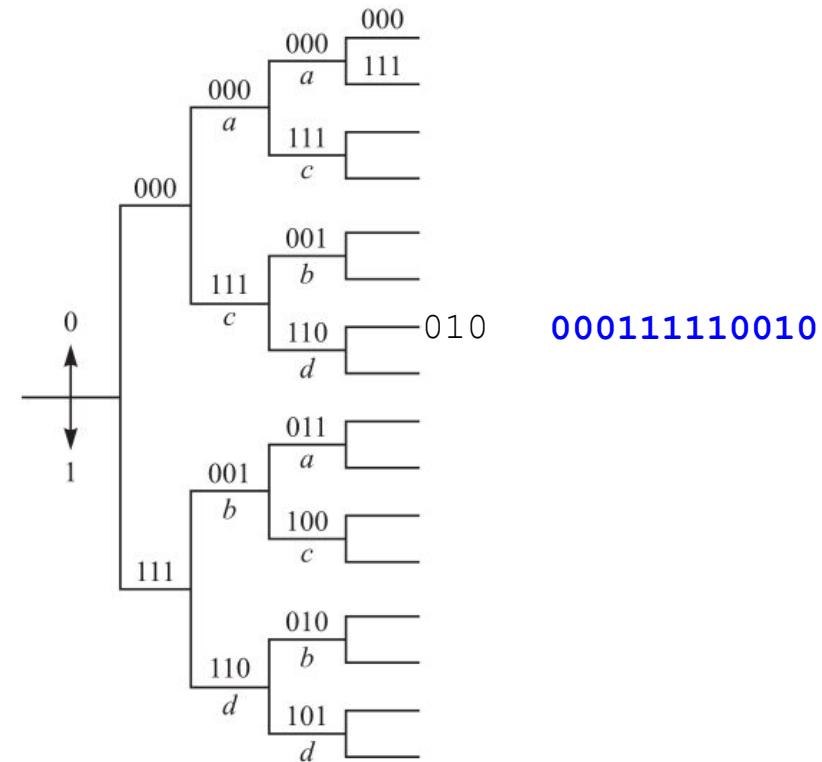
Códigos convolucionales:

$K = 3, k = 1, n = 3$



0 1 1 0
0 1 1 1
0 1 0 0

Representación en forma de árbol



000111110010

Codificación de canal

Códigos

Códigos convolucionales:

$K = 3, k = 1, n = 3$

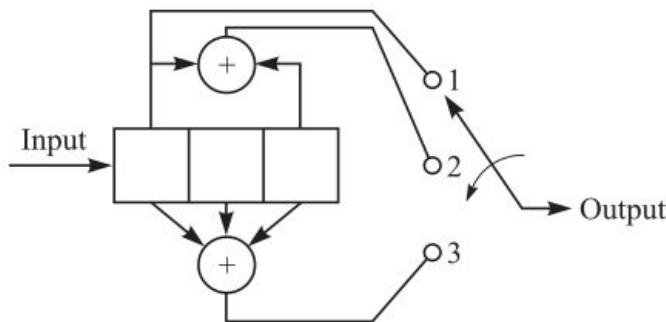
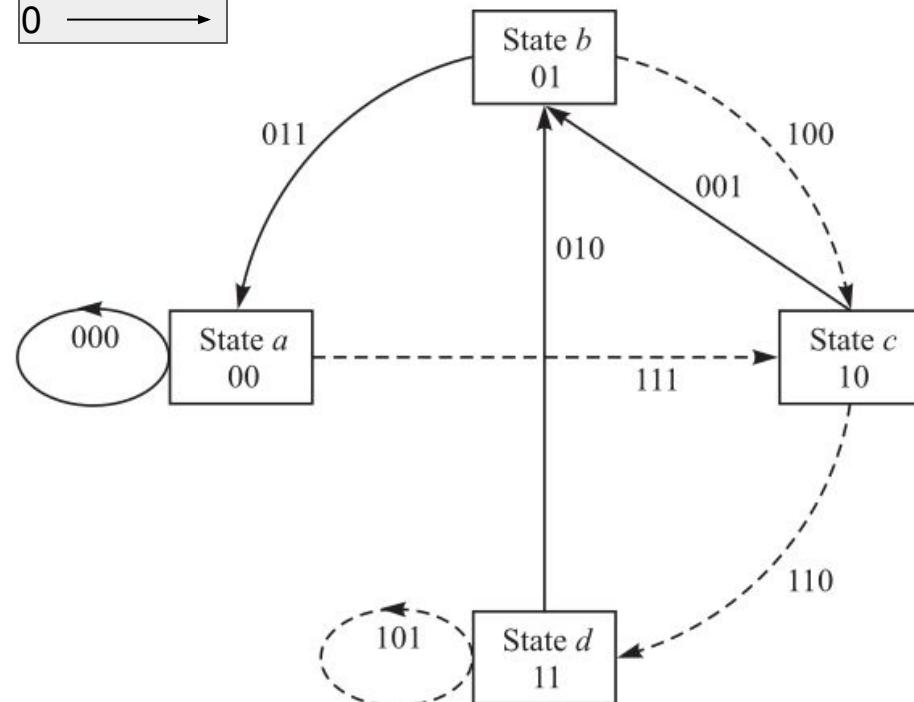


Diagrama de estados



Codificación de canal

Códigos

Códigos convolucionales:

$K = 3, k = 1, n = 3$

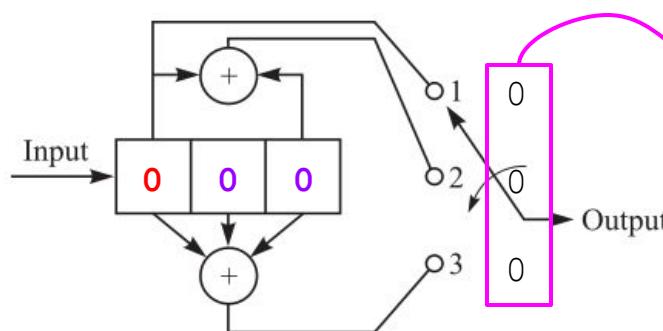
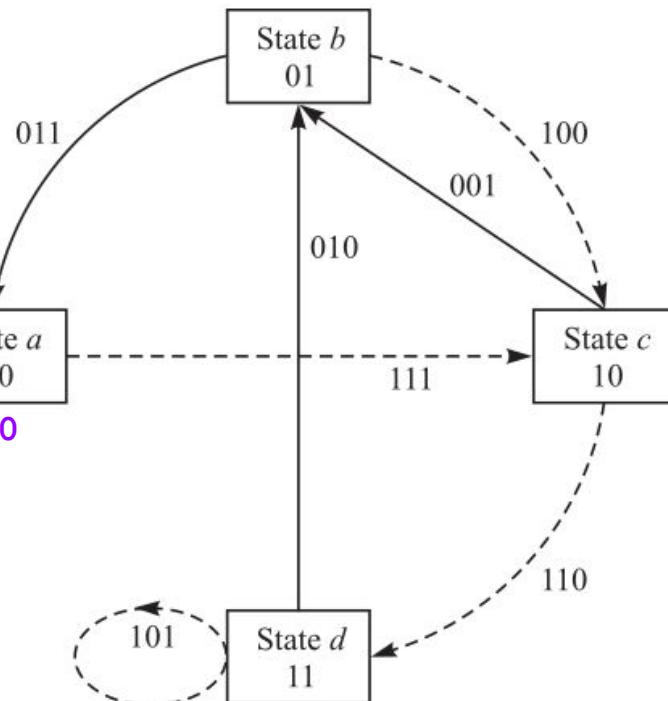


Diagrama de estados



Codificación de canal

Códigos

Códigos convolucionales:

$K = 3, k = 1, n = 3$

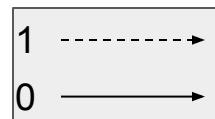
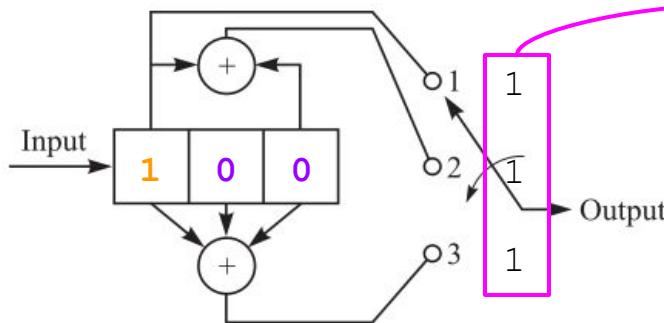
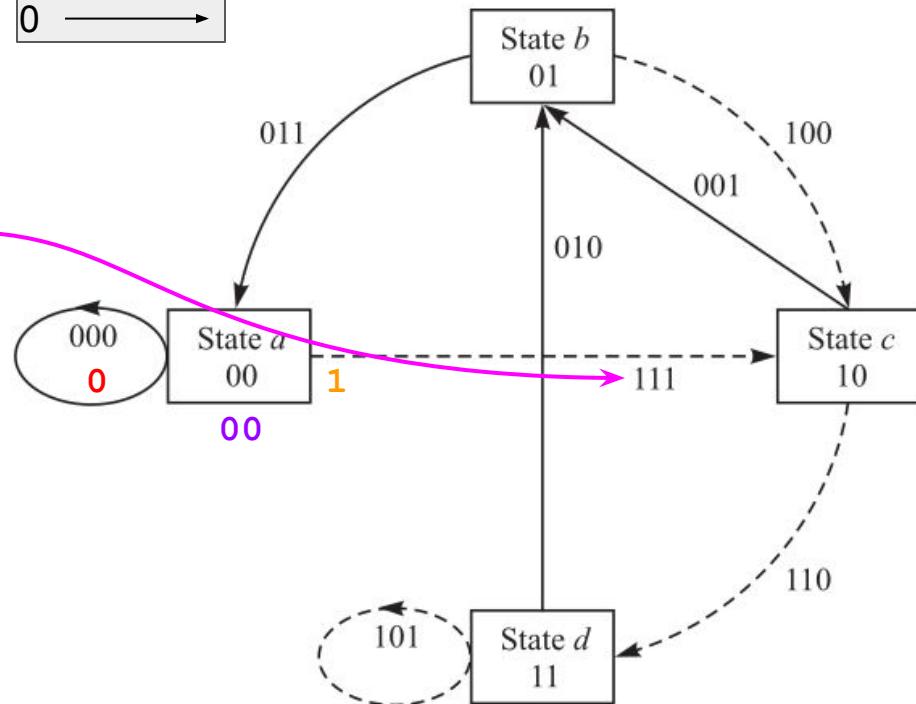


Diagrama de estados



Codificación de canal

Códigos

Códigos convolucionales:

$K = 3, k = 1, n = 3$

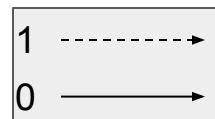
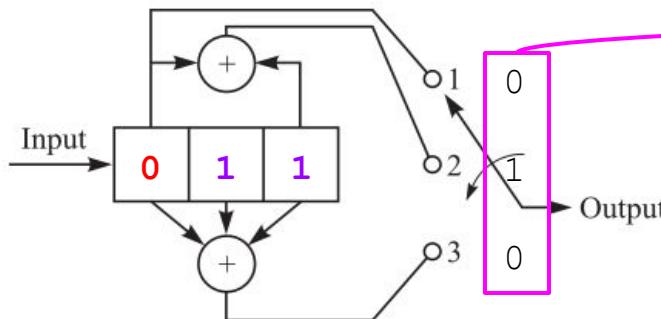
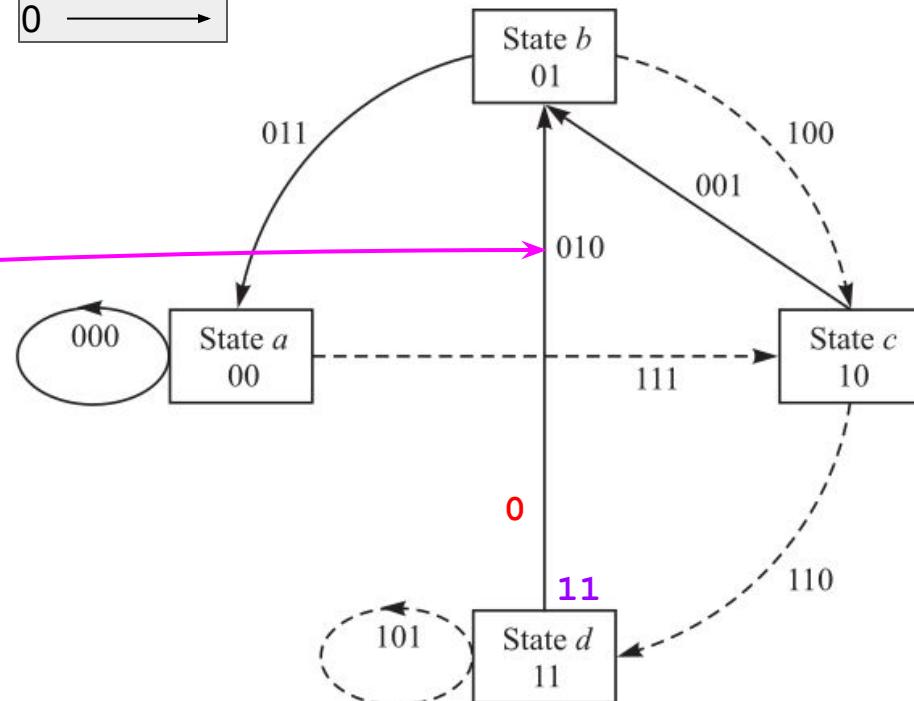


Diagrama de estados



Codificación de canal

Códigos

Códigos convolucionales:

$K = 3, k = 1, n = 3$

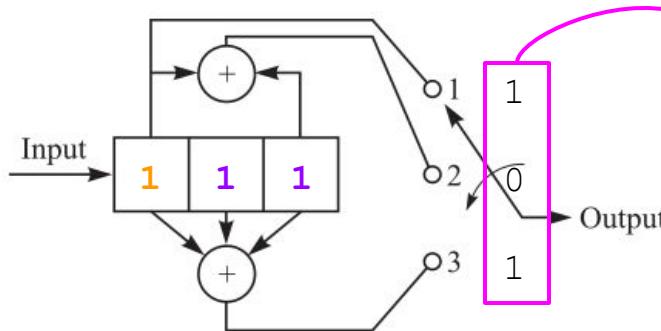
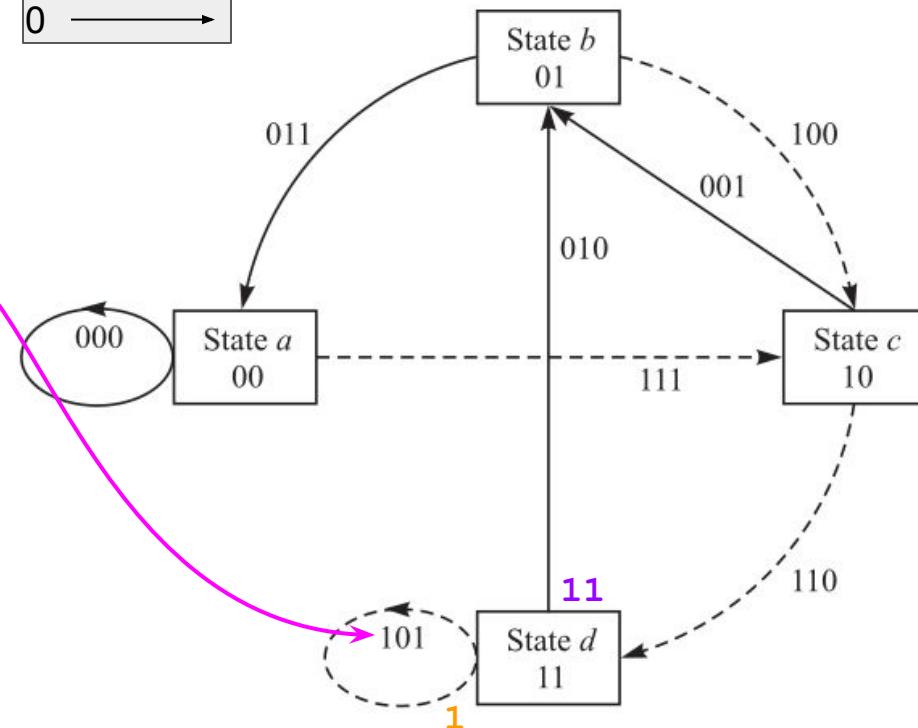


Diagrama de estados



Codificación de canal

Códigos

Códigos convolucionales:

$K = 3, k = 1, n = 3$

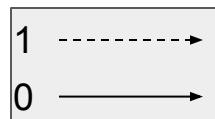
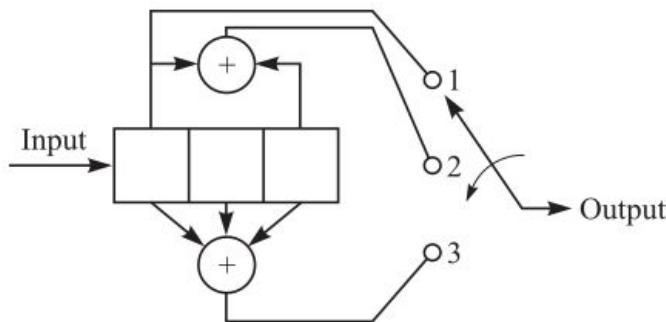
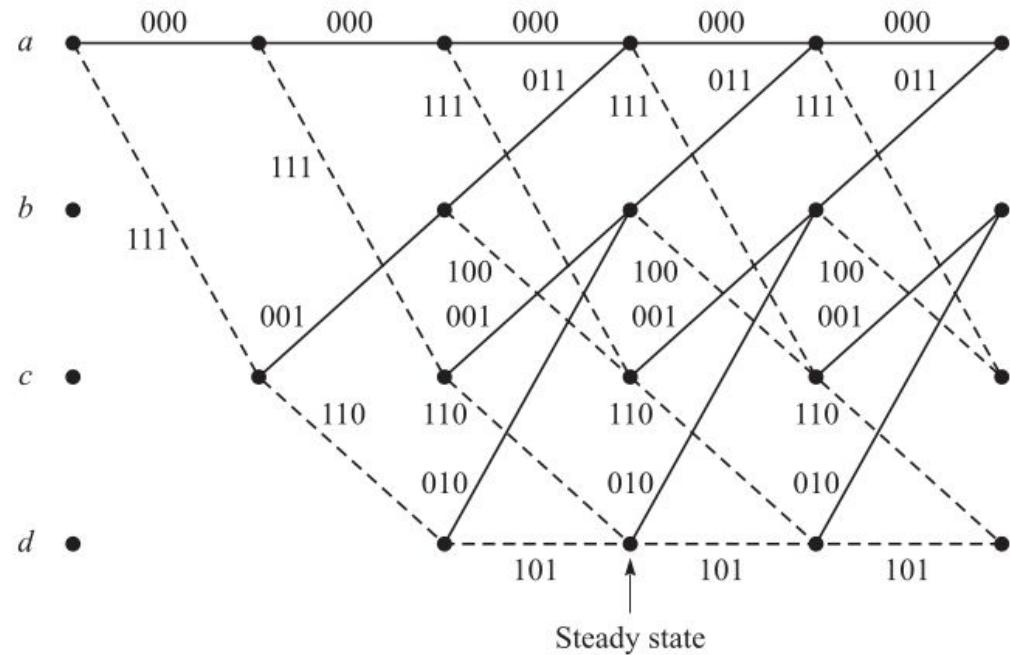


Diagrama de Trellis



Codificación de canal

Códigos

Códigos convolucionales:

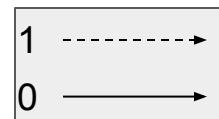
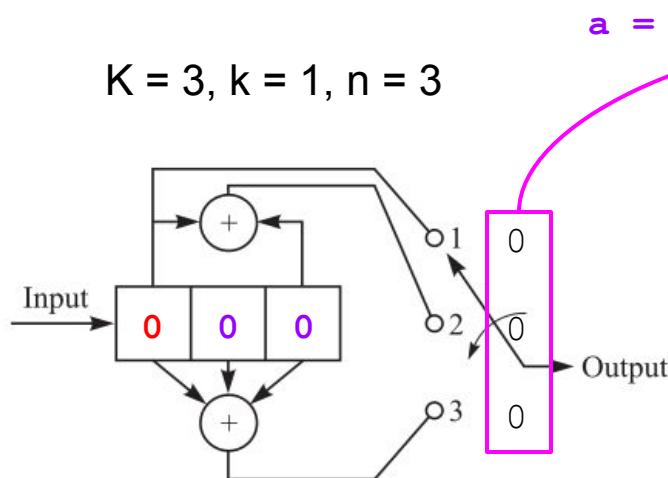
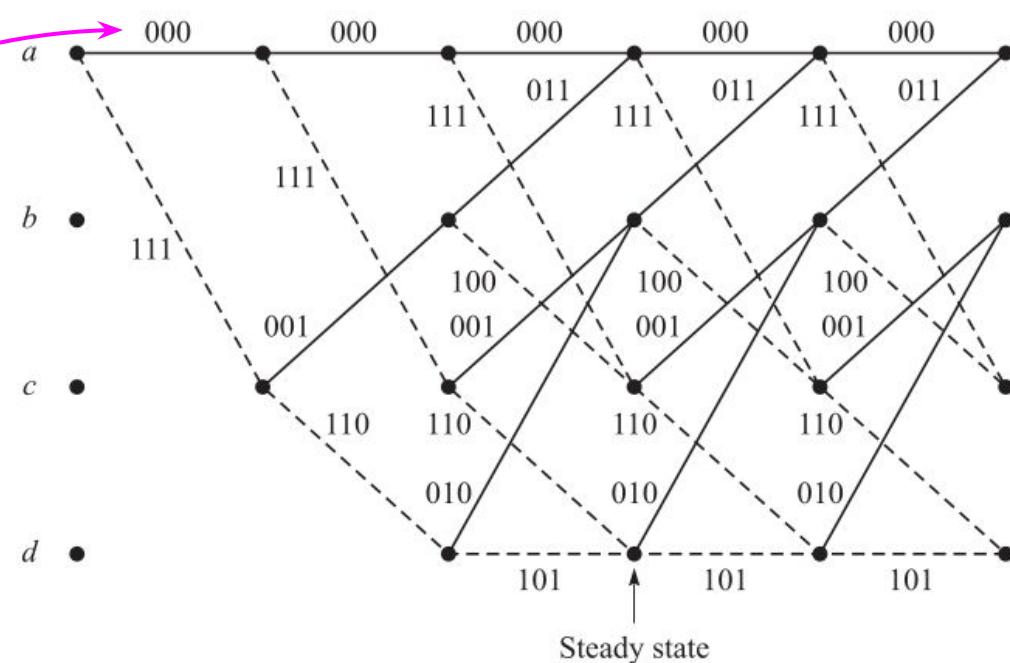


Diagrama de Trellis



Codificación de canal

Códigos

Códigos convolucionales:

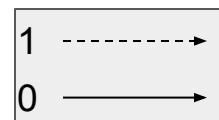
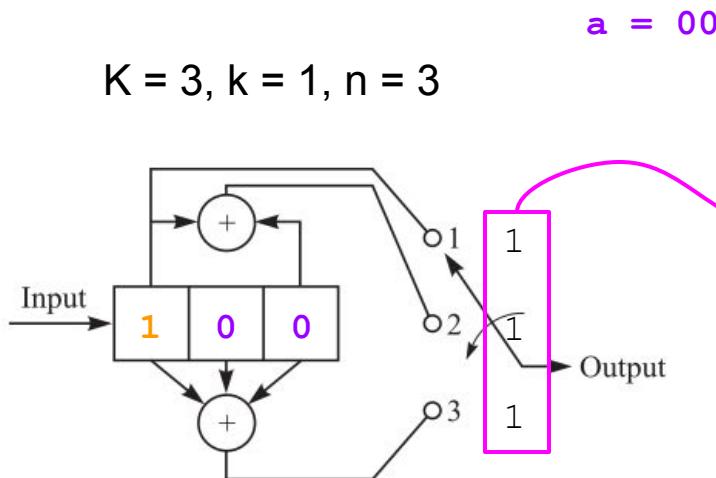
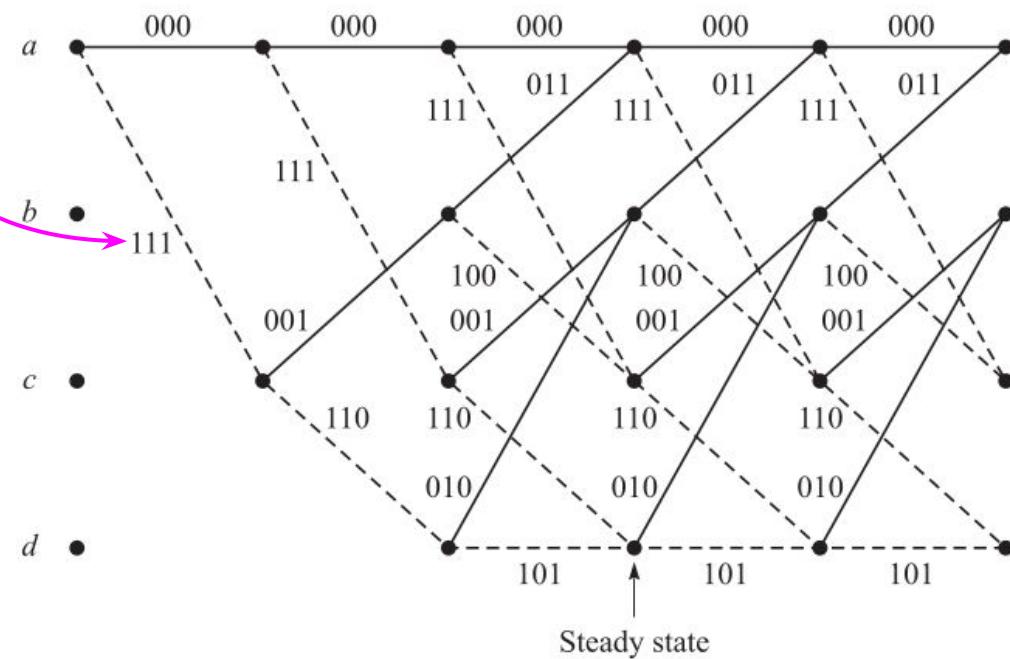


Diagrama de Trellis



Codificación de canal

Códigos

Códigos convolucionales:

$K = 3, k = 1, n = 3$

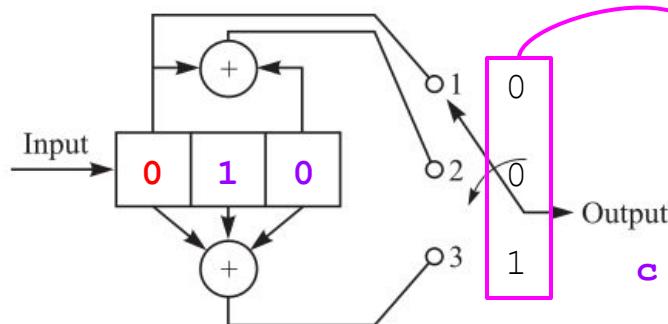
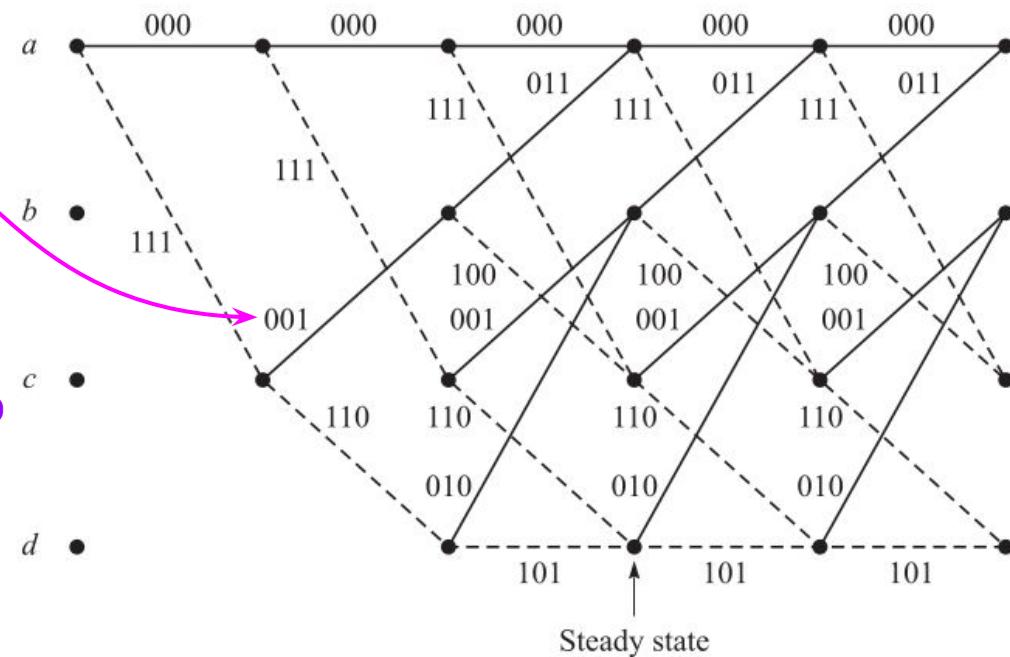


Diagrama de Trellis



Codificación de canal

Códigos

Códigos convolucionales:

$K = 3, k = 1, n = 3$

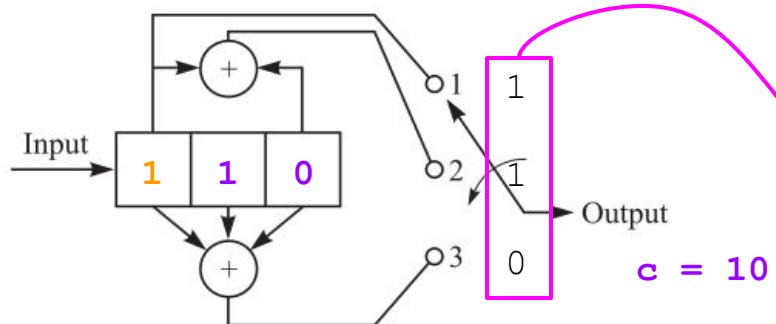
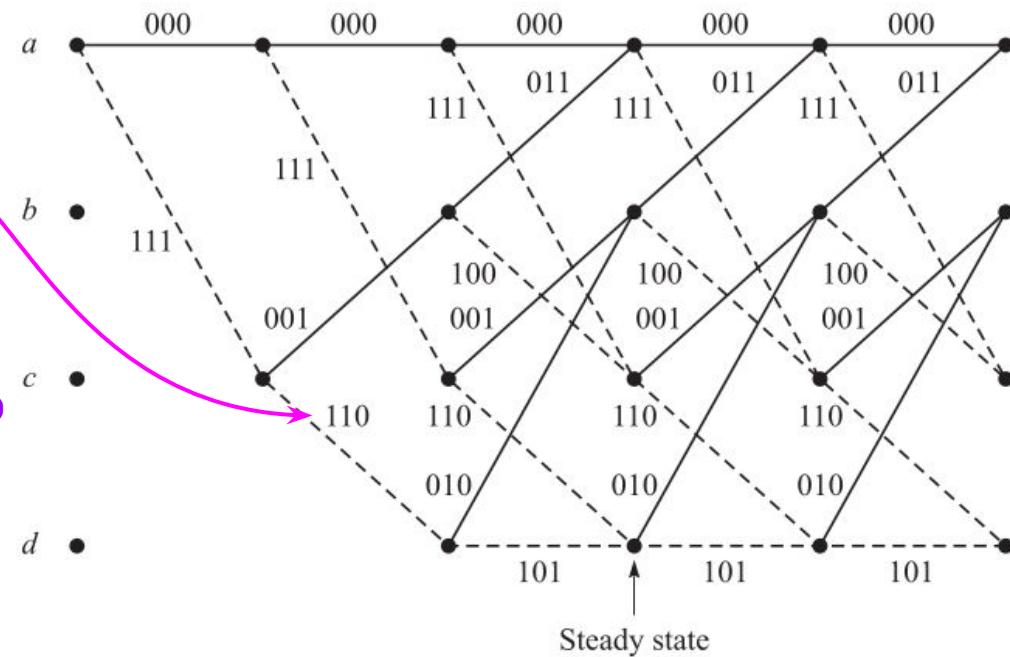


Diagrama de Trellis



Codificación de canal

Códigos

Algoritmo de Viterbi:

Datos		0		1		0		1		1	
TX		000		111		001		100		110	
Estado		00		10		01		10		11	
		a		c		b		c		d	
RX		010		110		011		100		100	

¿Cómo se realiza la decodificación?

Con el algoritmo de Viterbi.

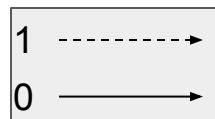
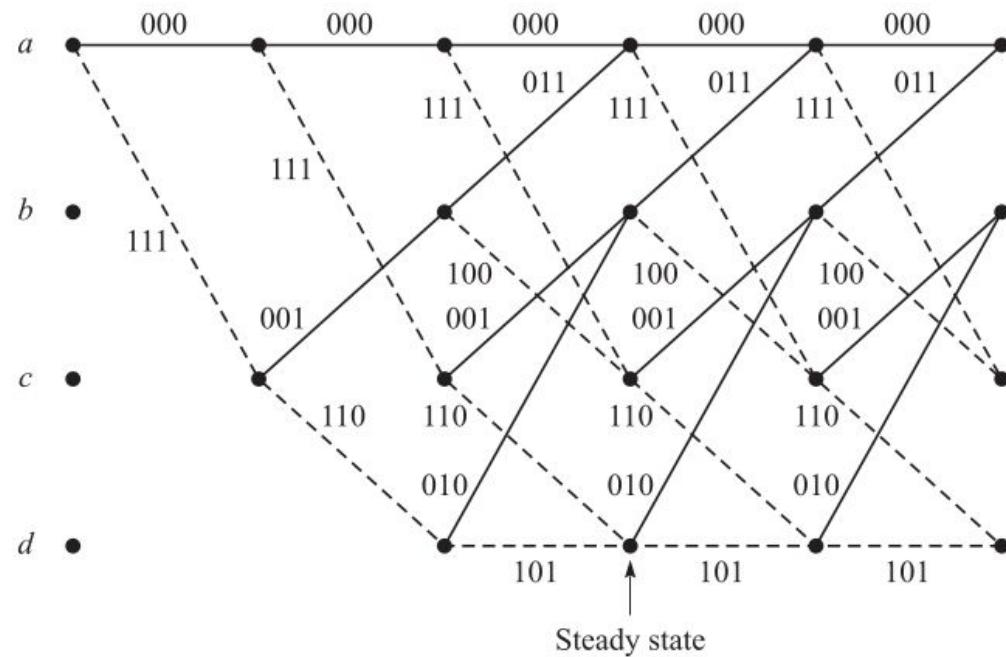


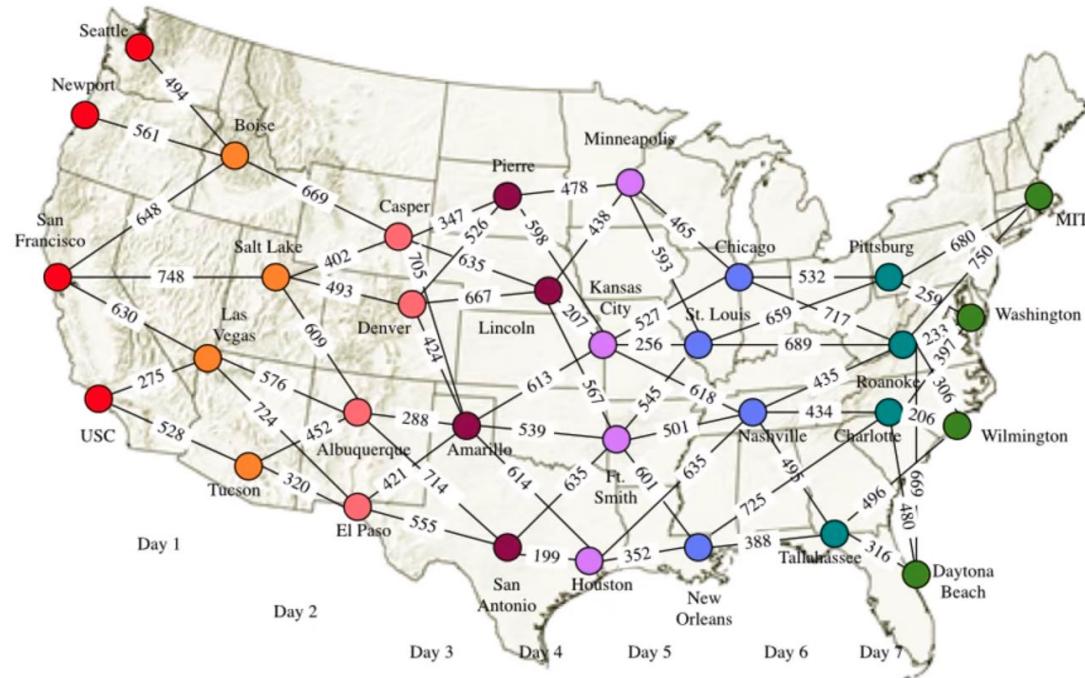
Diagrama de Trellis



Codificación de canal

Códigos

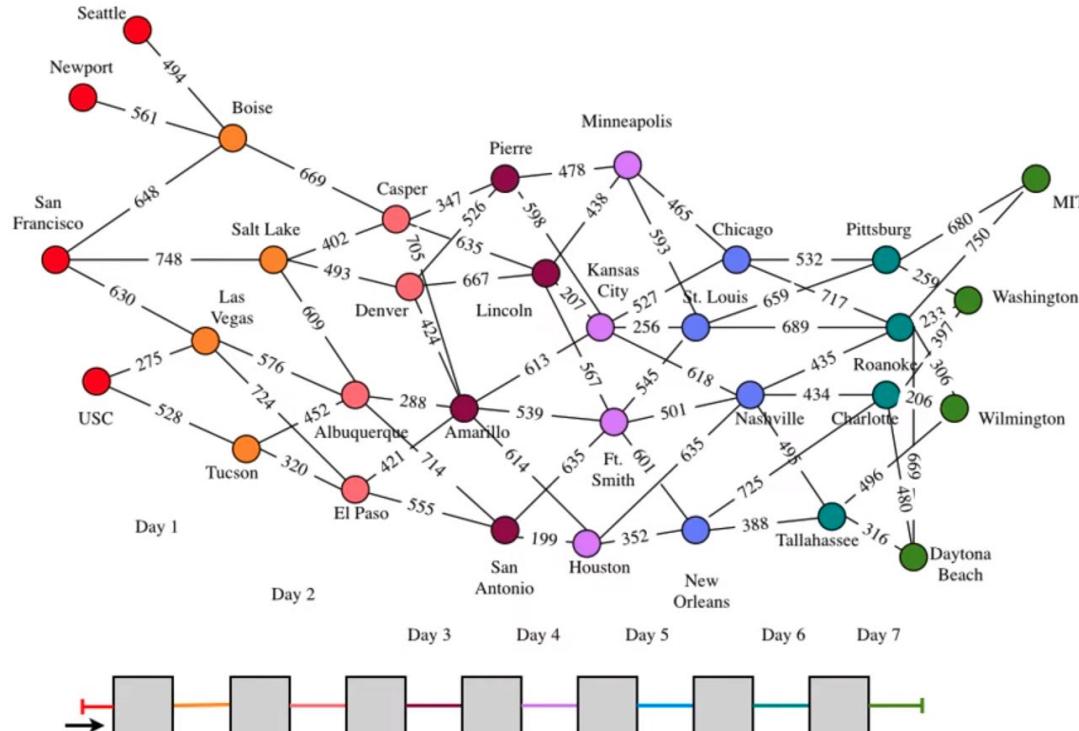
Algoritmo de Viterbi:



Codificación de canal

Códigos

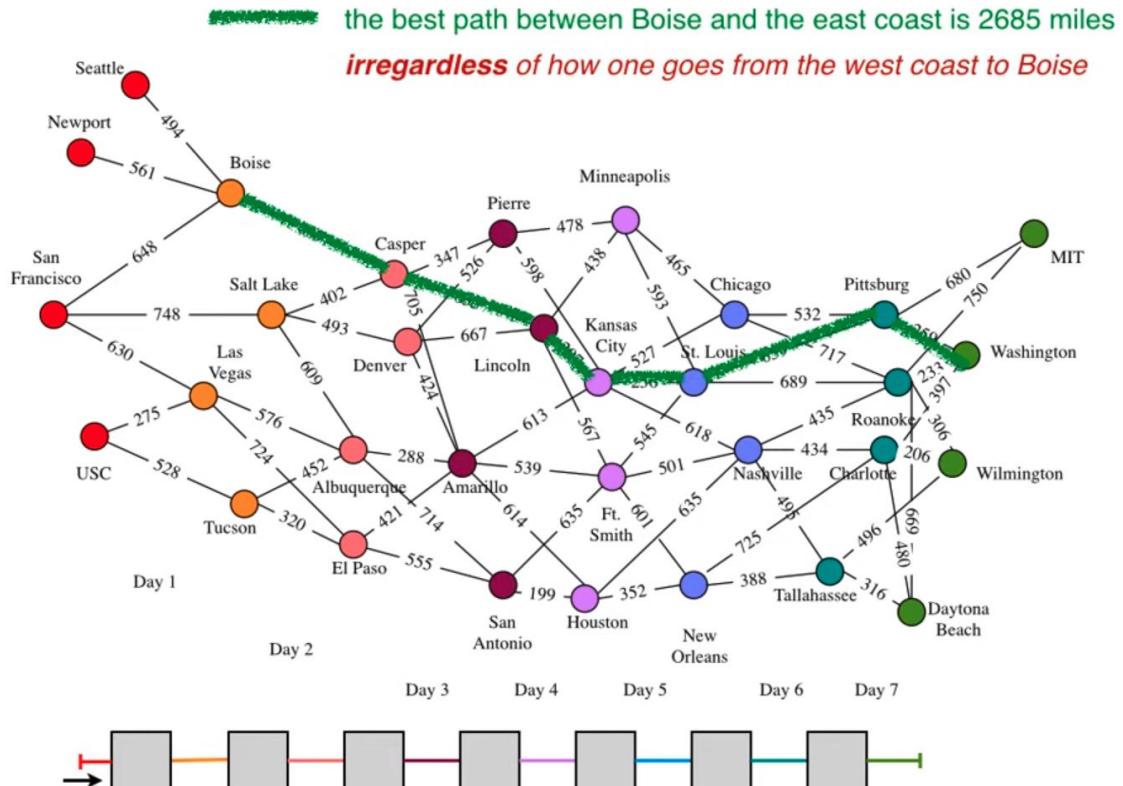
Algoritmo de Viterbi:



Codificación de canal

Códigos

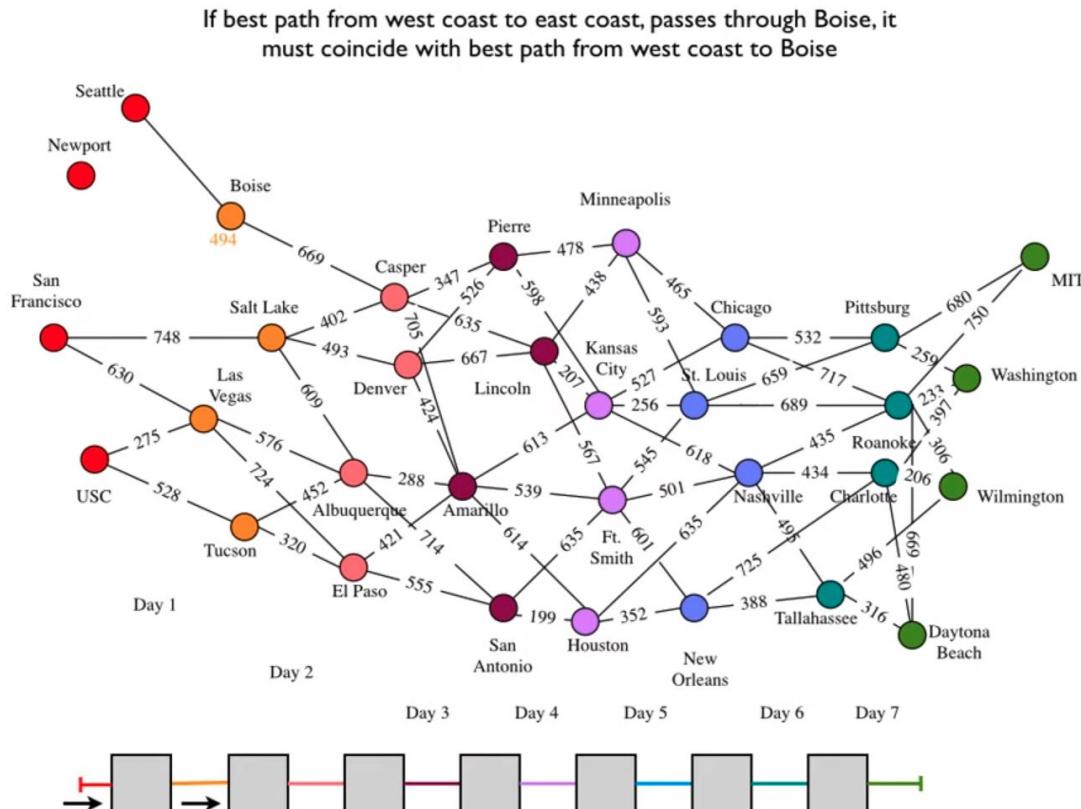
Algoritmo de Viterbi:



Codificación de canal

Códigos

Algoritmo de Viterbi:

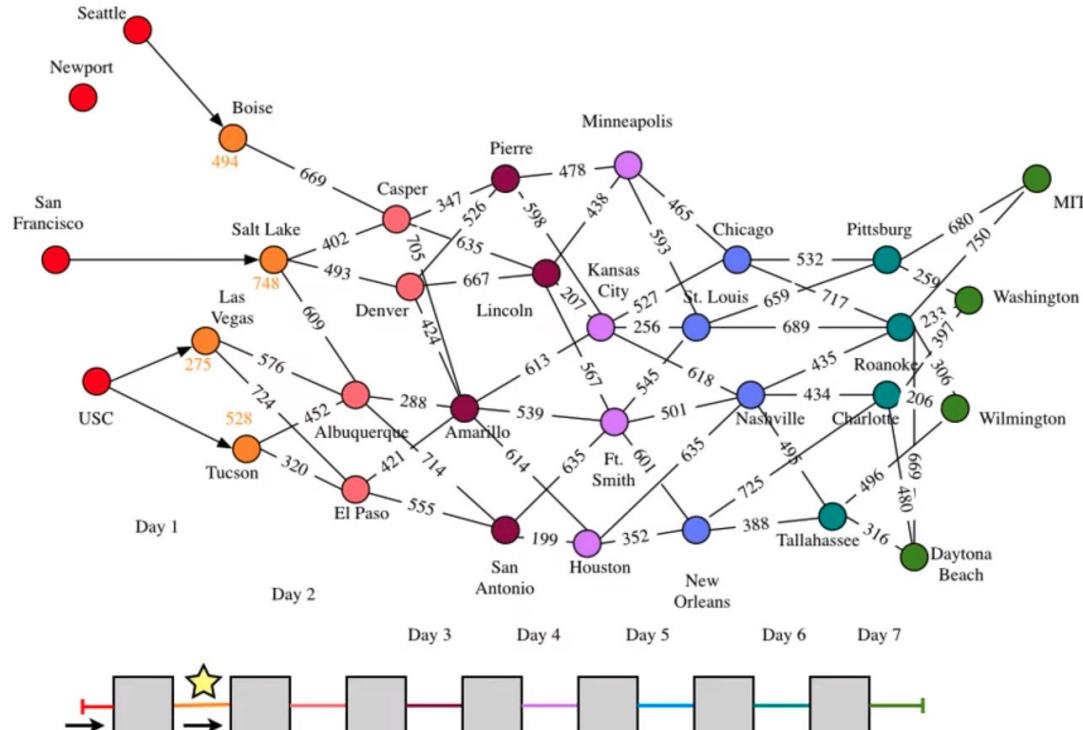


Codificación de canal

Códigos

Algoritmo de Viterbi:

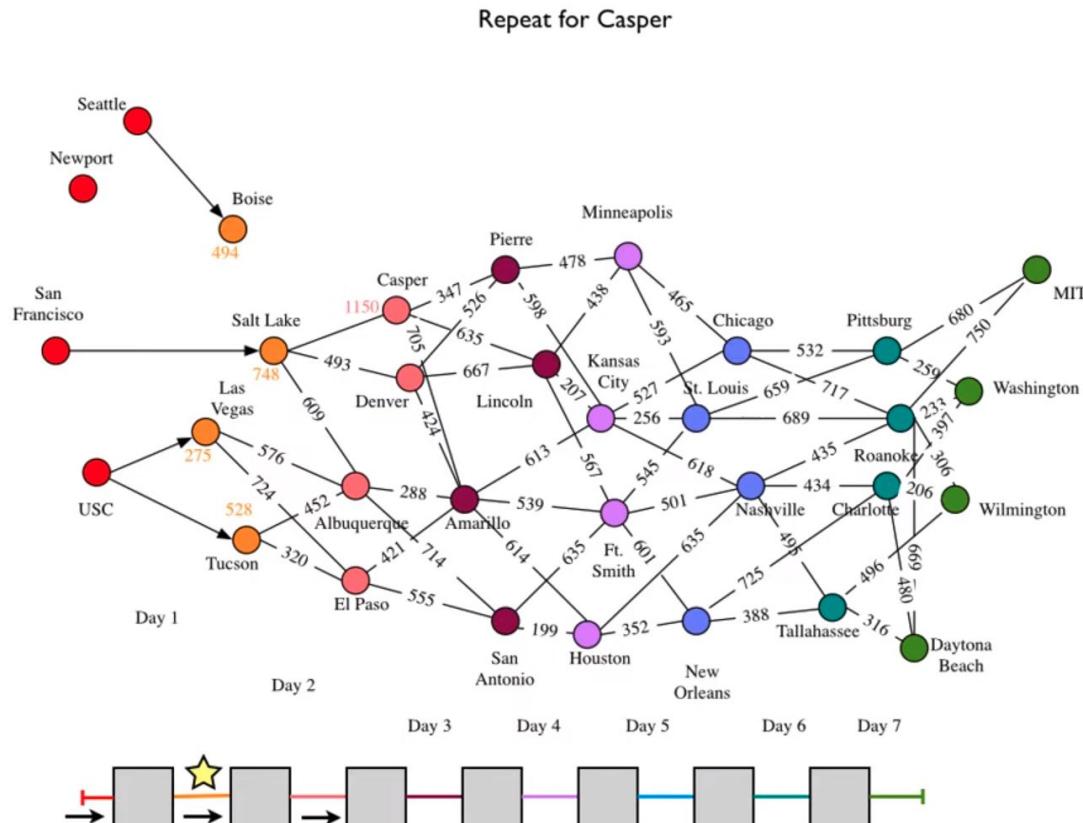
Repeat Boise-argument for Salt Lake, Las Vegas, Tucson



Codificación de canal

Códigos

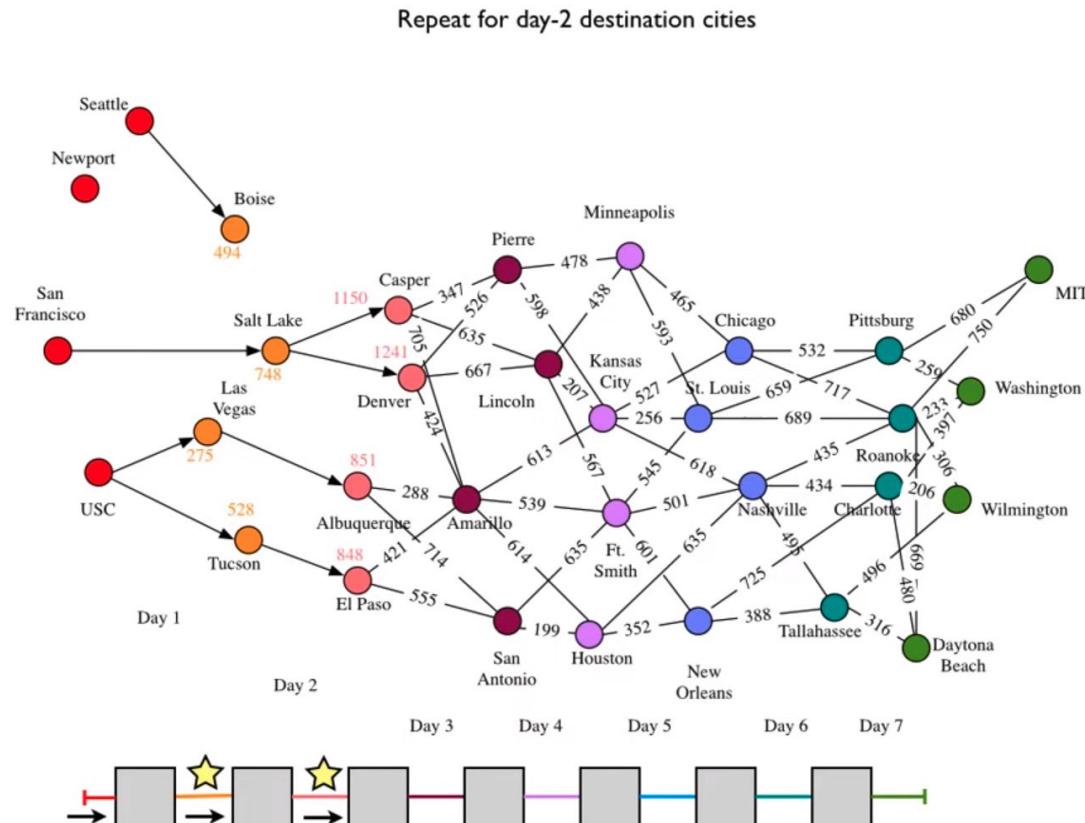
Algoritmo de Viterbi:



Codificación de canal

Códigos

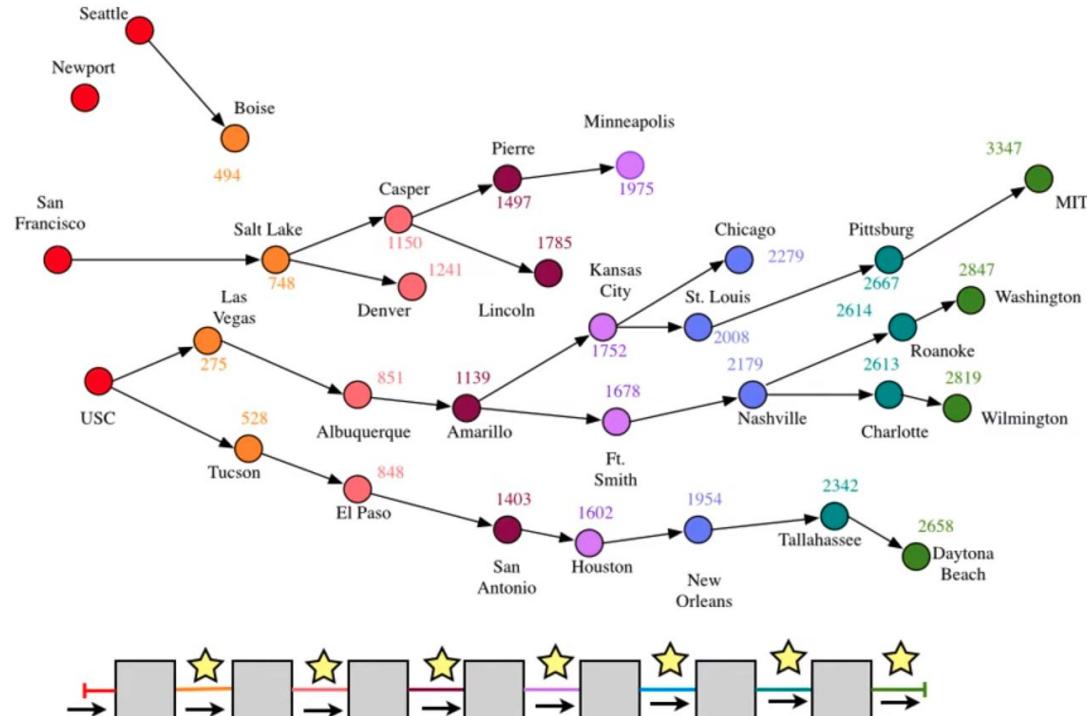
Algoritmo de Viterbi:



Codificación de canal

Códigos

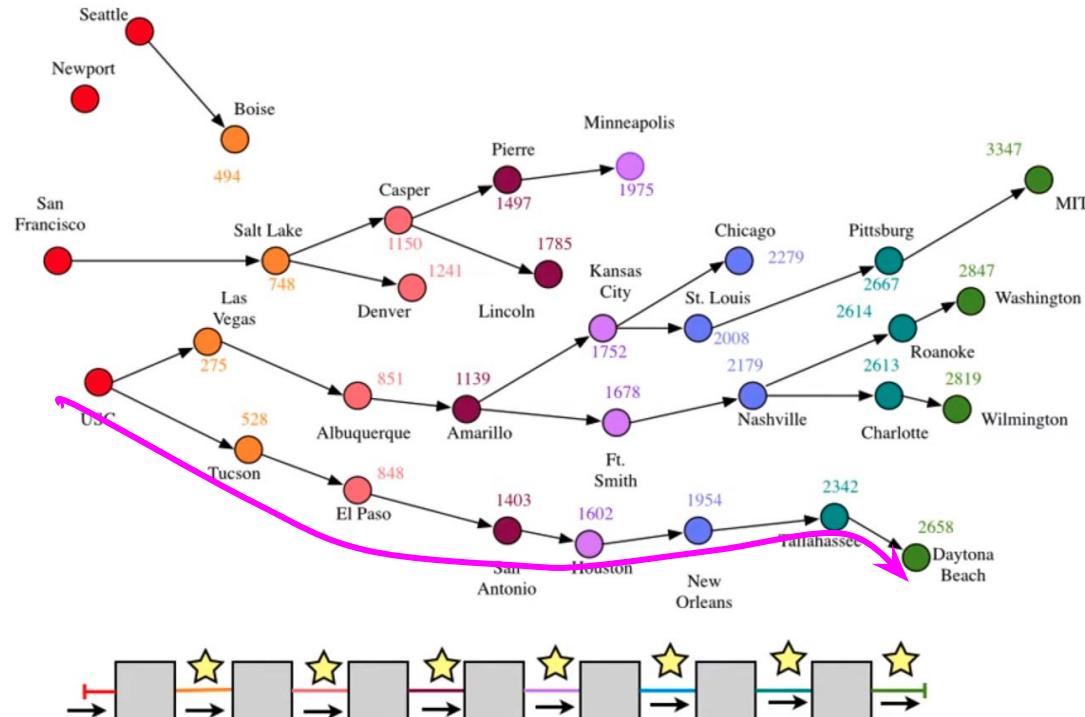
Algoritmo de Viterbi:



Codificación de canal

Códigos

Algoritmo de Viterbi:



Codificación de canal

Códigos

Algoritmo de Viterbi:

Datos		0		1		0		1		1	
TX		000		111		001		100		110	
Estado		00		10		01		10		11	
		a		c		b		c		d	
RX		000		111		001		100		110	

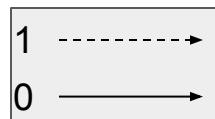
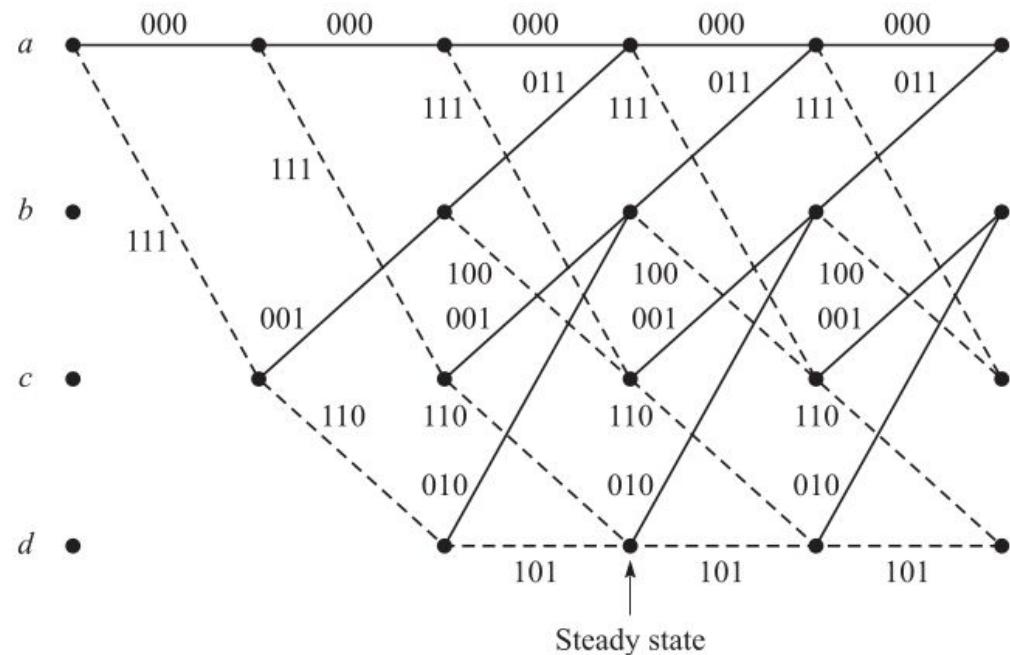


Diagrama de Trellis



¿Cómo se realiza la decodificación?

Con el algoritmo de Viterbi.

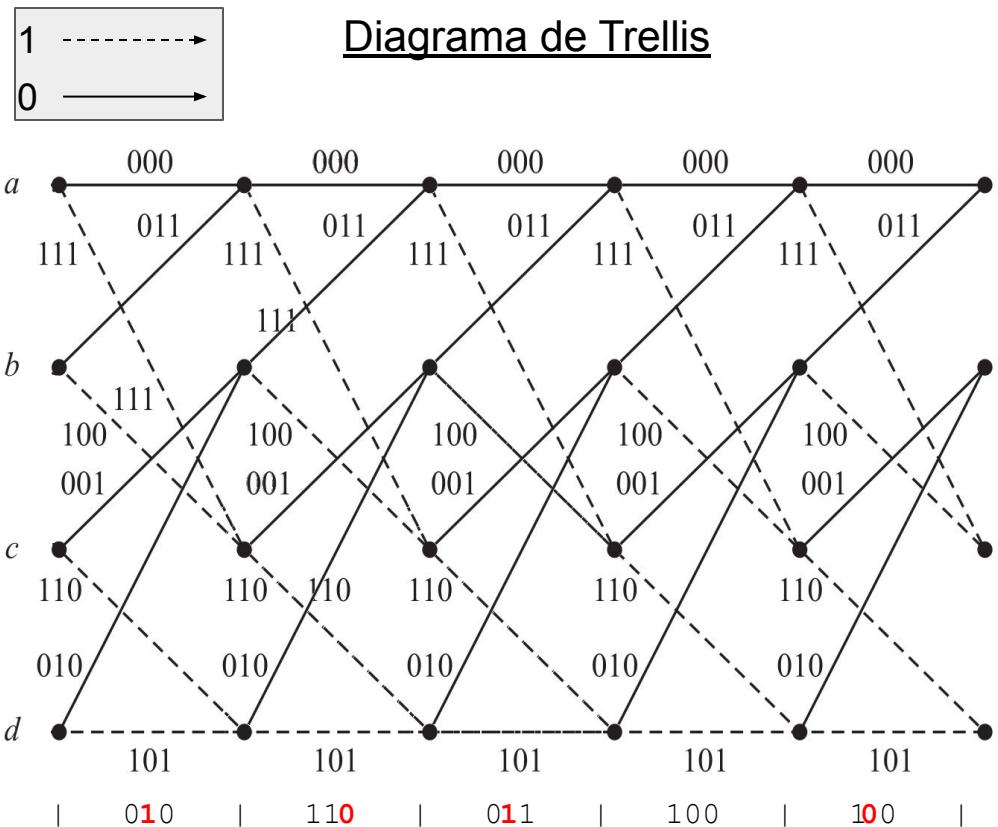
Codificación de canal

Códigos

Algoritmo de Viterbi:

Distancias:

RX	0	1	0	1	1	0	0	1
000	1	2	1	2	1	1	1	
001	2	3	1	1	2	2		
010	0	1	1	1	2	2		
011	1	2	0	0	3	3		
100	2	1	3	0	0	0		
101	3	2	2	1	1	1		
110	1	0	2	1	1	1		
111	2	1	1	1	2	2		



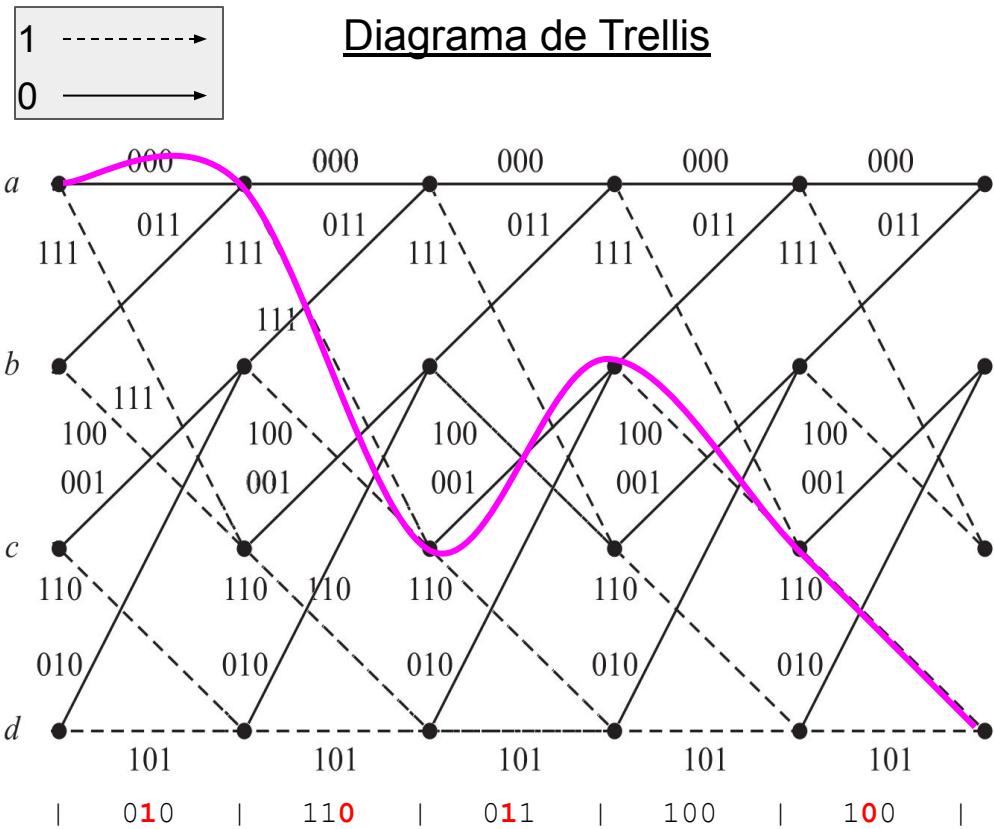
Codificación de canal

Códigos

Algoritmo de Viterbi:

Distancias:

RX	0	1	0	1	1	0	0	1
000	1	2	1	2	1	1	1	
001	2	3	1	1	2	2		
010	0	1	1	1	2	2		
011	1	2	0	0	3	3		
100	2	1	3	0	0	0		
101	3	2	2	1	1	1		
110	1	0	2	1	1	1		
111	2	1	1	1	2	2		



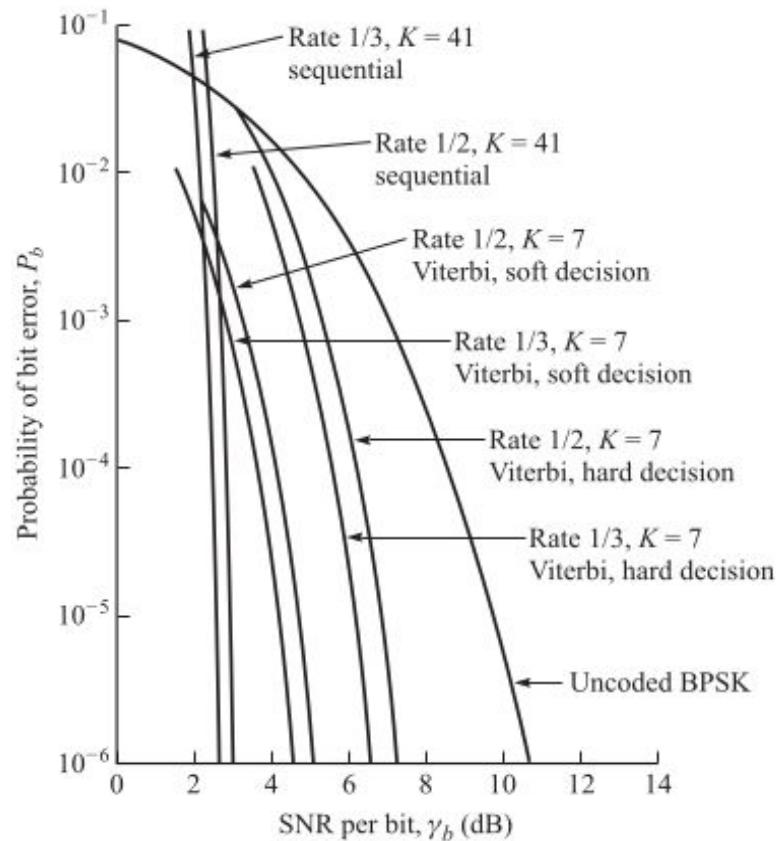
Codificación de canal

Códigos

Códigos convolucionales:

Algunos decodificadores:

- Decodificador Viterbi
 - Decisión Soft
 - Decisión Hard
- Decodificador Secuencial
- Otros decodificadores



Codificación de canal

Códigos

Entrelazado:

Hace más robusta la detección y corrección de errores en presencia de ruido tipo ráfaga.

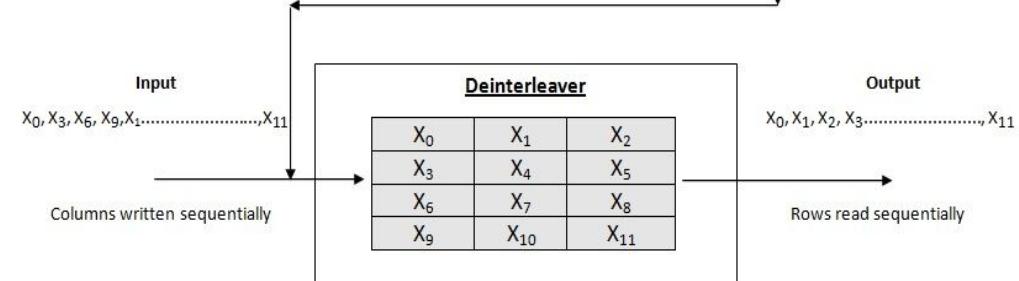
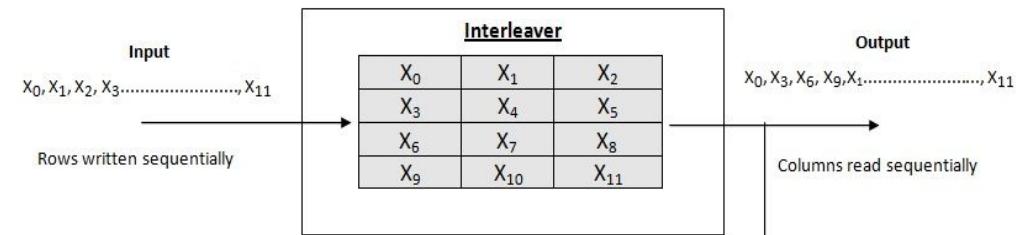
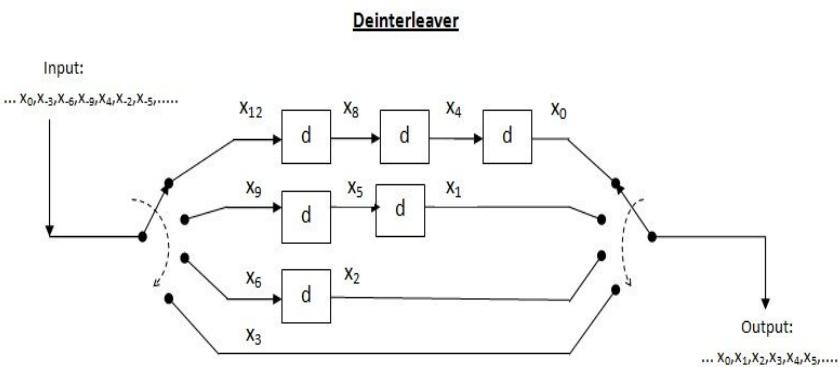
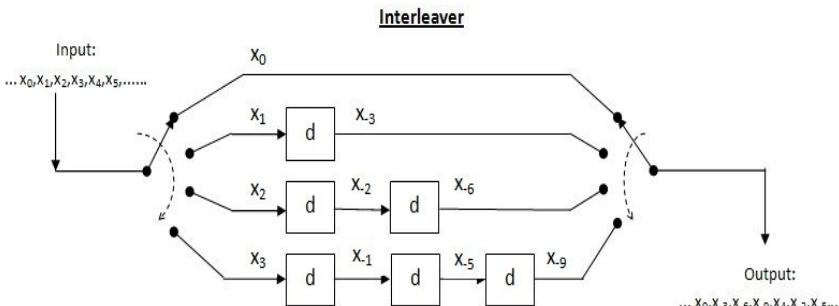


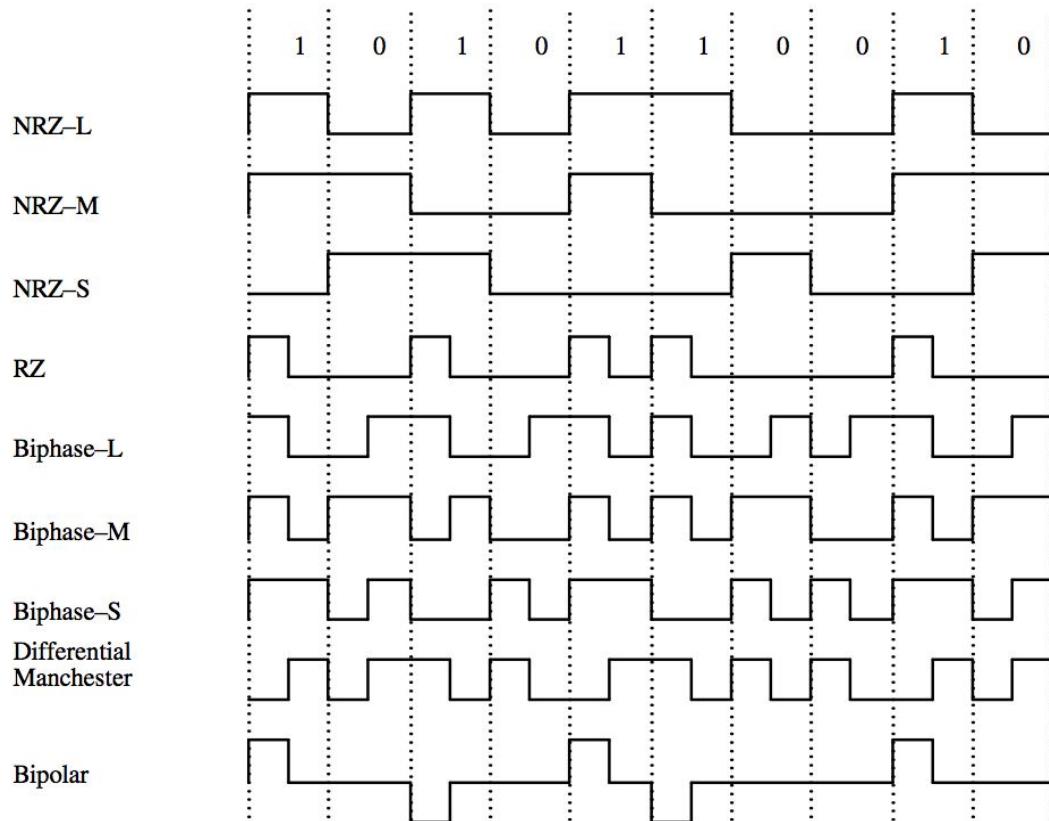
Fig. A.4 X 3 Interleaver and Deinterleaver

Codificación de canal

Códigos

Codificación de línea:

- Minimizar el Hardware
- Facilitar la sincronización
- Facilitar la detección y corrección de errores
- Modificar la densidad espectral
- Eliminar la componente de continua



Codificación de canal

Códigos

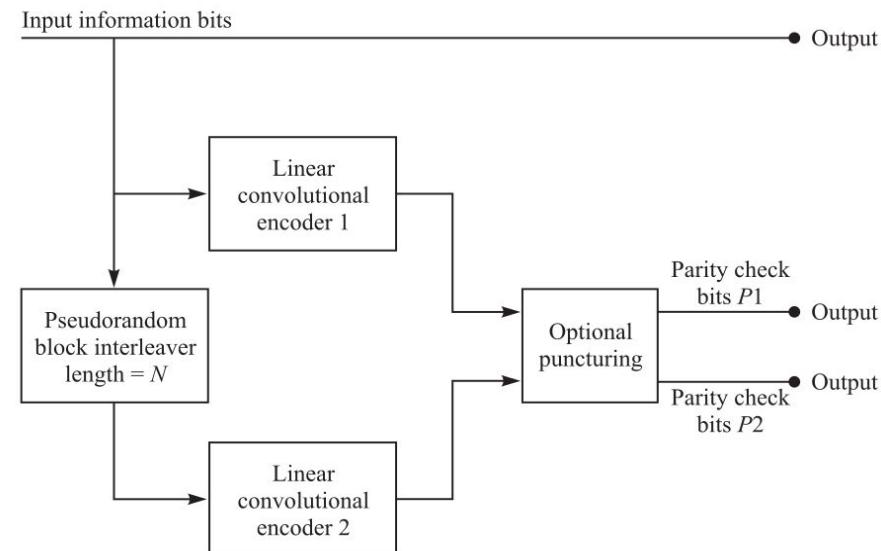
Más sobre códigos:

Los distintos códigos que se presentaron se pueden combinar para obtener nuevos códigos con mejores características.

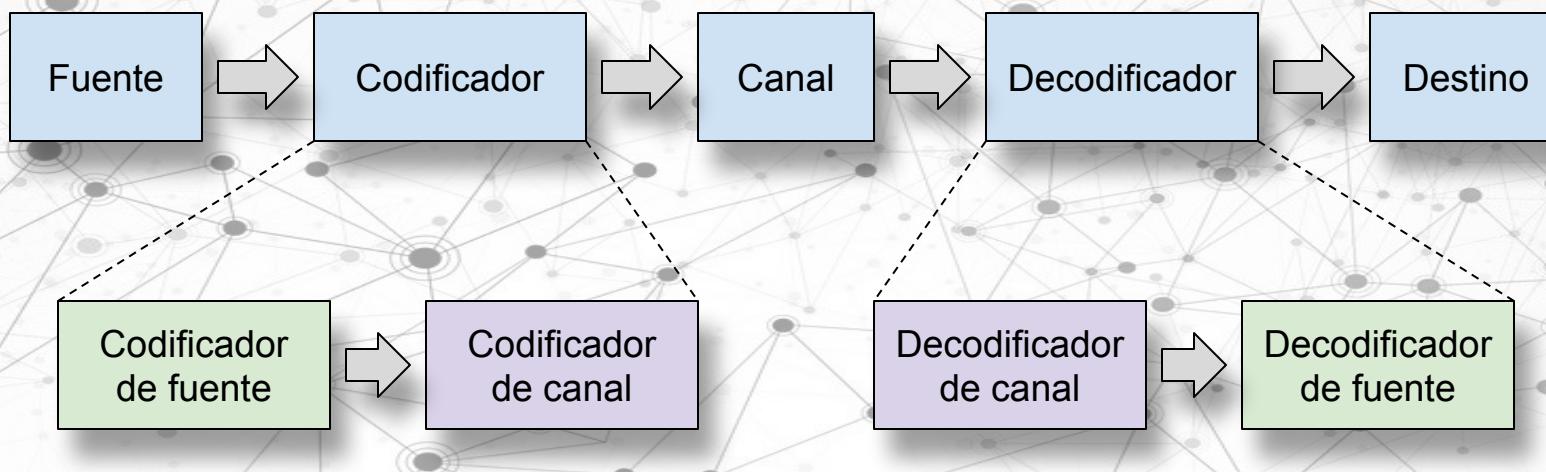
En general el gran problema para alcanzar los límites es complejidad de los decodificadores particularmente para códigos largos.

Turbo Codes (para investigar)

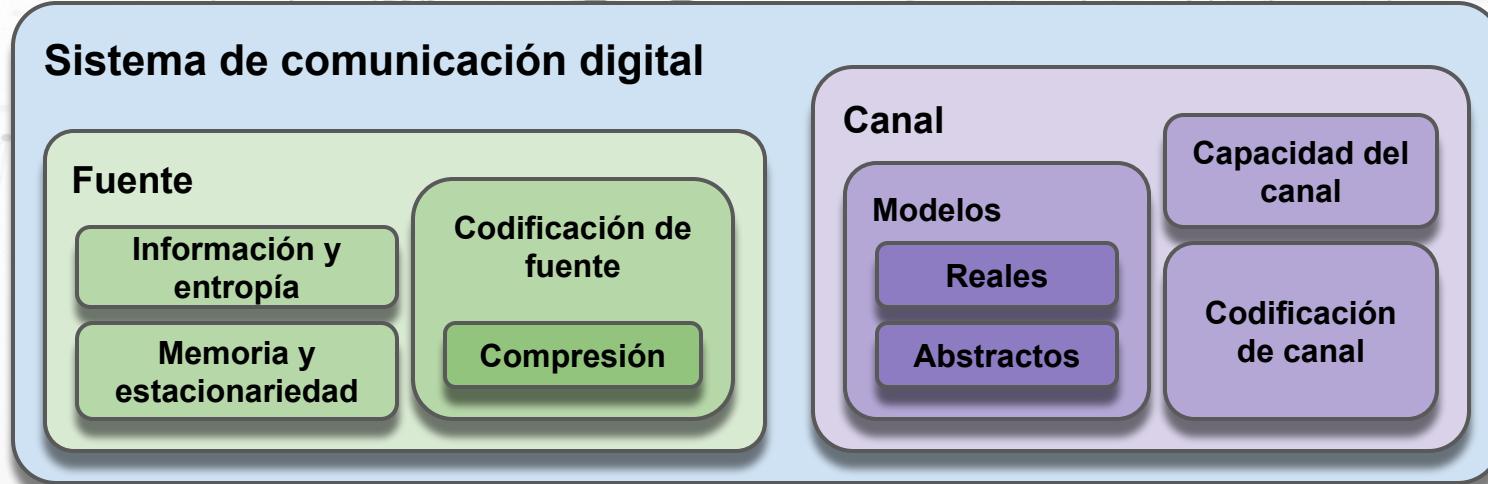
El mecanismo de decodificación de los códigos turbo también impulsaron a los LDPC.



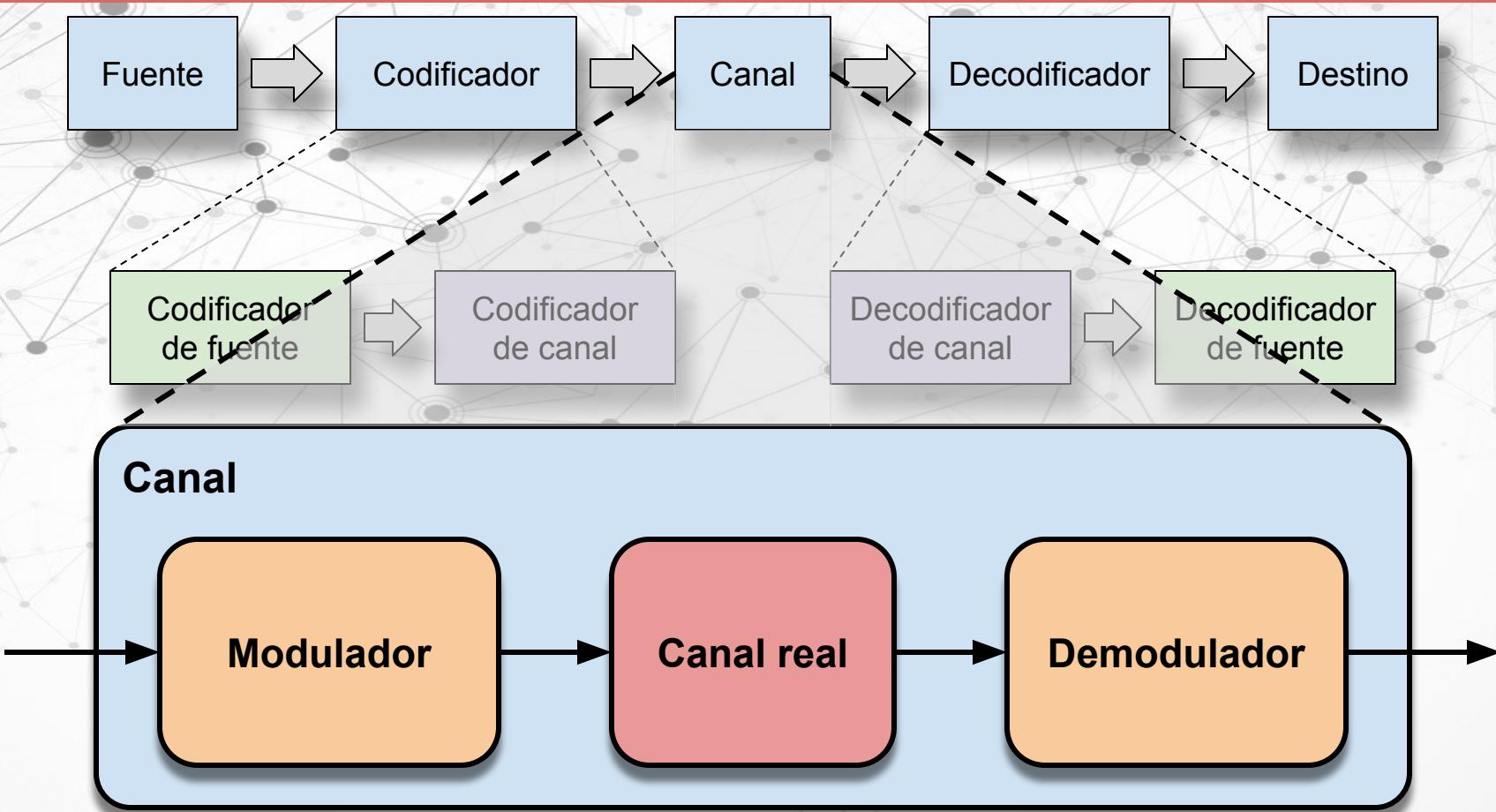
Resumen y siguientes pasos



Sistema de comunicación digital



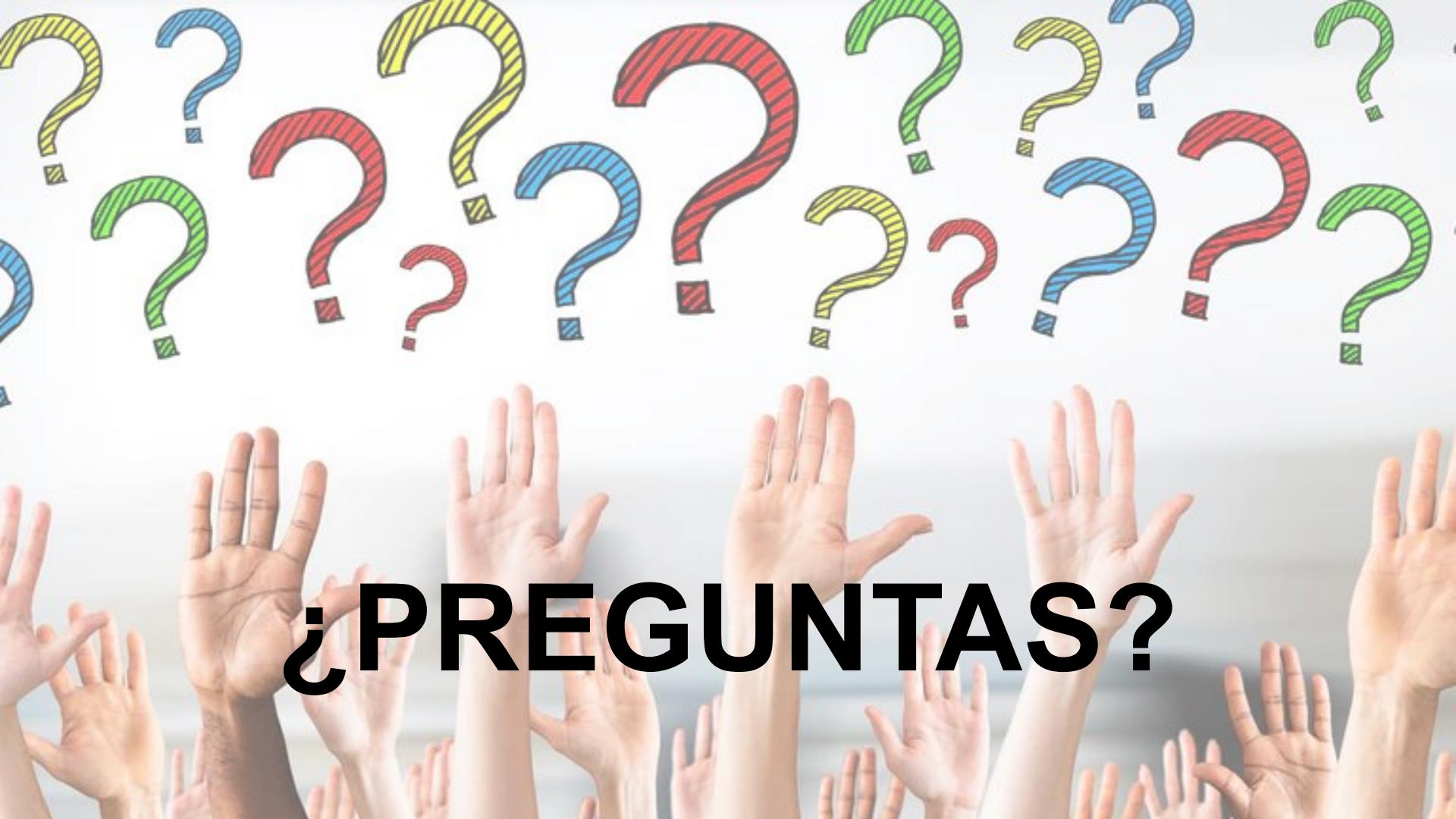
Resumen y siguientes pasos



Bibliografía y fuentes

Bibliografía:

- “*Digital Communication*”, John G. Proakis, Masoud Salehi, 5th edition.
- Andrew J. Viterbi, “*Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm*,” IEEE Trans. Information Theory, vol. 13, pp. 259-260, April 1967.
- Thomas M. Cover and Joy A. Thomas. 2006. “*Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*”. Wiley-Interscience, USA.
- B. P. Lathi. 1998. “*Modern Digital and Analog Communication Systems 3e Osece*” (3rd. ed.). Oxford University Press, Inc., USA.
- Steven W. Smith. 1997. “*The scientist and engineer's guide to digital signal processing*”. California Technical Publishing, USA.
- Bernard Sklar. 1988. “*Digital communications: fundamentals and applications*”. Prentice-Hall, Inc., USA.
- Ejemplo Viterbi: <https://www.youtube.com/watch?v=6JVqutwtzmo>



¿PREGUNTAS?



¡Gracias!