

# Teoría de la Información

Enrique Alexandre

Dpto. de Teoría de la Señal y Comunicaciones

Curso 2022/2023

# Índice

1. Introducción
  - Ejemplo
2. Teoría de la información
  - Medida de la información
  - Características de los códigos fuente
3. Codificación Huffman
  - Introducción
  - Ejemplo
4. Codificación LZW
  - Definición
  - Pseudocódigo
  - Ejemplo
  - Decodificación
  - Ejemplo



# Introducción

## Ejemplo

TM

**IEEE Standard for  
Information technology—  
Telecommunications and information  
exchange between systems—  
Local and metropolitan area networks—  
Specific requirements**

**Part 11: Wireless LAN Medium Access Control (MAC)  
and Physical Layer (PHY) Specifications**

---

**IEEE Computer Society**

Sponsored by the  
LAN/MAN Standards Committee

# Introducción

## Ejemplo

### PART 11: WIRELESS LAN MAC AND PHY SPECIFICATIONS

IEEE  
Std 802.11-2007

A short OFDM training symbol consists of 12 subcarriers, which are modulated by the elements of the sequence S, given by

$$S_{-26:26} = \sqrt{(13/6)} \times \{0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 0, 0, 0, 0, -1-j, 0, 0, 0, -1-j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0, 0, 1+j, 0, 0\} \quad (17-6)$$

The multiplication by a factor of  $\sqrt{(13/6)}$  is in order to normalize the average power of the resulting OFDM symbol, which utilizes 12 out of 52 subcarriers.

The signal shall be generated according to the following equation:

$$r_{SHORT}(t) = w_{TSHORT}(t) \sum_{k=-N_{ST}/2}^{N_{ST}/2} S_k \exp(j2\pi k \Delta_F t) \quad (17-7)$$

The fact that only spectral lines of  $S_{-26:26}$  with indices that are a multiple of 4 have nonzero amplitude results in a periodicity of  $T_{FFT}/4 = 0.8 \mu s$ . The interval  $T_{SHORT}$  is equal to ten  $0.8 \mu s$  periods (i.e.,  $8 \mu s$ ).

Generation of the short training sequence is illustrated in Table G.2.

# Introducción

## Ejemplo

**Table 17-3—Modulation-dependent parameters**

Modulation	Coding rate ( $R$ )	Coded bits per subcarrier ( $N_{BPSK}$ )	Coded bits per OFDM symbol ( $N_{CBPS}$ )	Data bits per OFDM symbol ( $N_{DBPS}$ )	Data rate (Mb/s) (20 MHz channel spacing)	Data rate (Mb/s) (10 MHz channel spacing)	Data rate (Mb/s) (5 MHz channel spacing)
BPSK	1/2	1	48	24	6	3	1.5
BPSK	3/4	1	48	36	9	4.5	2.25
QPSK	1/2	2	96	48	12	6	3
QPSK	3/4	2	96	72	18	9	4.5
16-QAM	1/2	4	192	96	24	12	6
16-QAM	3/4	4	192	144	36	18	9
64-QAM	2/3	6	288	192	48	24	12
64-QAM	3/4	6	288	216	54	27	13.5

# Introducción

## Ejemplo

PART 11: WIRELESS LAN MAC AND PHY SPECIFICATIONS

IEEE  
Std 802.11-2007

### 18.4.6.6 DSSS/PBCC data modulation and modulation rate (optional)

This optional coding scheme uses a binary convolutional coding with a 64-state binary convolutional code (BCC) and a cover sequence. The output of the BCC is encoded jointly onto the I and Q channels, as described in this subclause.

The encoder for this scheme is shown in Figure 18-12. Incoming data are first encoded with a binary convolutional code. A cover code is applied to the encoded data prior to transmission through the channel.

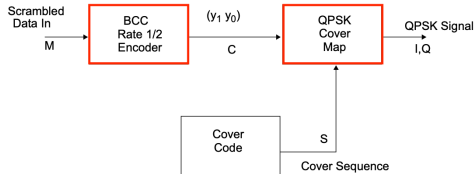


Figure 18-12—PBCC modulator scheme

# Introducción

## Ejemplo

The BCC that is used is a 64-state, rate  $\frac{1}{2}$  code. The generator matrix for the code is given as

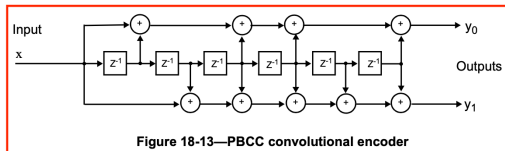
$$G = [D^6 + D^4 + D^3 + D + 1, \quad D^6 + D^5 + D^4 + D^3 + D^2 + 1]$$

or in octal notation, it is given by

$$G = [133, \quad 175]$$

Because the system is frame (PPDU) based, the encoder shall be in state zero (i.e., all memory elements contain zero at the beginning of each PPDU). The encoder must also be placed in a known state at the end of each PPDU to prevent the data bits near the end of the PPDU from being substantially less reliable than those early on in the PPDU. To place the encoder in a known state at the end of a PPDU, at least six deterministic bits must be input immediately following the last data bit input to the convolutional encoder. This is achieved by appending one octet containing all zeros to the end of the PPDU prior to transmission, and discarding the final octet of each received PPDU. In this manner, the decoding process can be completed reliably on the last data bits.

An encoder block diagram is shown in Figure 18-13. It consists of six memory elements. For every data bit input, two output bits are generated.





# Teoría de la información

## Medida de la información

- Se define la cantidad de información como:

$$I = -\log_2(p_k)$$

siendo  $p_k$  la probabilidad de aparición de cada uno de los posibles símbolos de la fuente.

- Se define la **entropía** de una fuente como:

$$H = - \sum_{k=0}^{N-1} p_k \cdot \log_2(p_k)$$

La entropía se mide en bits, e indica cuántos bits por símbolo como mínimo son necesarios para codificar la fuente.

# Teoría de la información

## Codificación de fuente

- **Objetivo:** Reducir el número de bits necesarios para representar la salida de una fuente.
- **Teorema de Shannon:** El número mínimo de bits por símbolo nunca va a ser inferior al valor de la entropía de la fuente.
- Códigos fuente:
  - Codificación Huffman: creada por David Huffman.
  - Codificación LZW: creada por Abraham Lempel, Jacob Ziv y Terry Welch.

# Teoría de la información

## Características de los códigos fuente

- **Longitud media:** Se define como la longitud promedio de las palabras del código

$$\bar{L} = \sum_{i=1}^M p_i n_i$$

- **Tasa de compresión:** Se define como la relación frente a un código de longitud fija:

$$\Gamma = \frac{\lceil \log_2 M \rceil}{\bar{L}}$$

- **Eficiencia:** Mide lo cerca que estamos del límite teórico de la entropía:

$$\eta = \frac{H(x)}{\bar{L}} \leq 1$$

# Codificación Huffman

## Introducción

- Símbolos menos probables -> Se codifican con más bits
- Símbolos más probables -> Se codifican con menos bits
- El código se construye de forma iterativa en forma de árbol

# Codificación Huffman

## Ejemplo

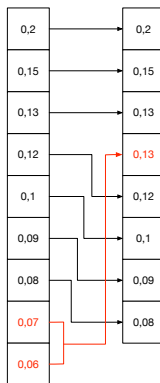
- Supongamos una fuente con nueve símbolos con las siguientes probabilidades:

$$\mathbf{p} = (0,2, 0,15, 0,13, 0,12, 0,1, 0,09, 0,08, 0,07, 0,06)$$

# Codificación Huffman

## Ejemplo

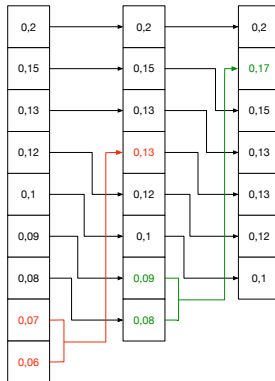
- Agrupo los dos símbolos menos probables en uno cuya probabilidad será la suma y reordeno:



# Codificación Huffman

## Ejemplo

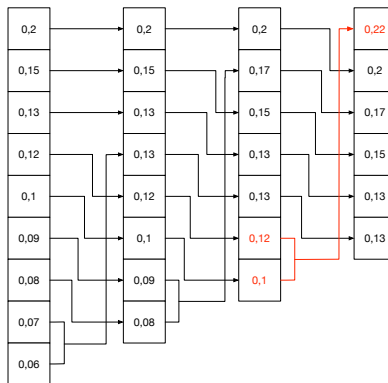
- Repito el proceso hasta que sólo queden dos símbolos:



# Codificación Huffman

## Ejemplo

- Repito el proceso hasta que sólo queden dos símbolos:

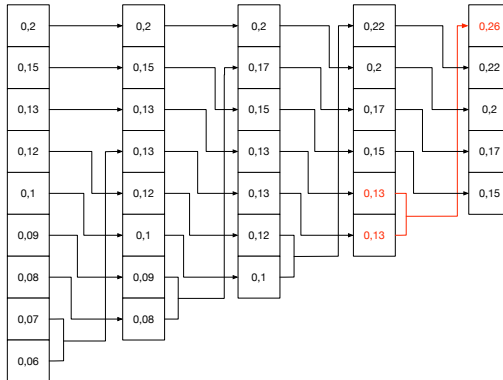




# Codificación Huffman

## Ejemplo

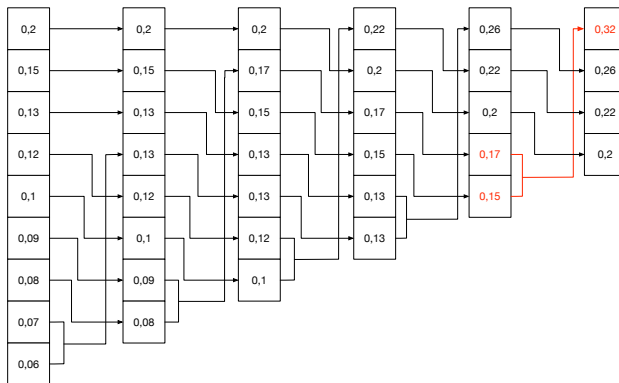
- Repito el proceso hasta que sólo queden dos símbolos:



# Codificación Huffman

## Ejemplo

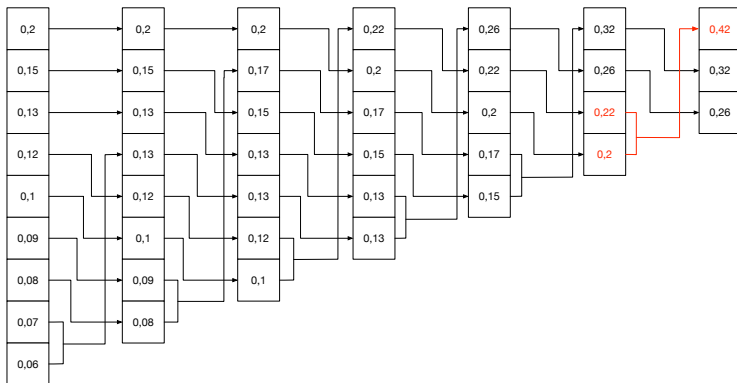
- Repito el proceso hasta que sólo queden dos símbolos:



# Codificación Huffman

## Ejemplo

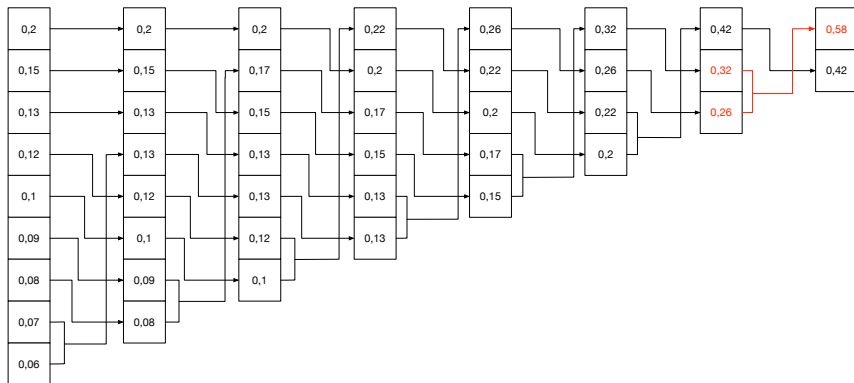
- Repito el proceso hasta que sólo queden dos símbolos:



# Codificación Huffman

## Ejemplo

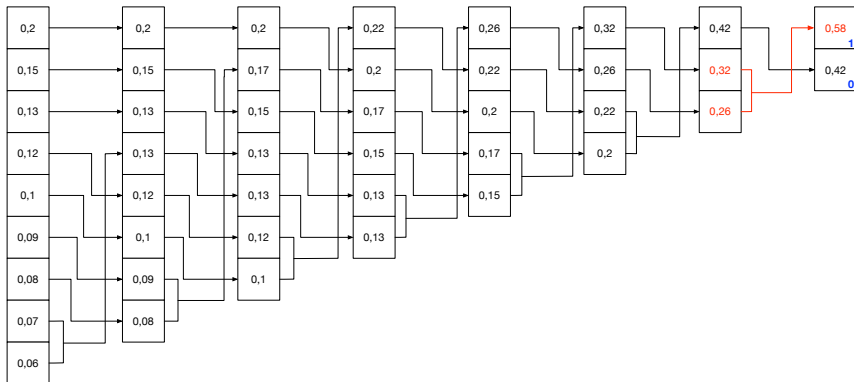
- Repito el proceso hasta que sólo queden dos símbolos:



# Codificación Huffman

## Ejemplo

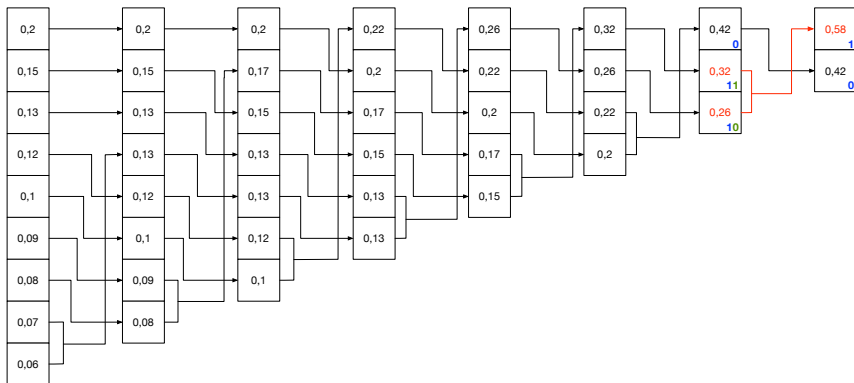
- Asigno 'o' a un símbolo y 'r' al otro:



# Codificación Huffman

## Ejemplo

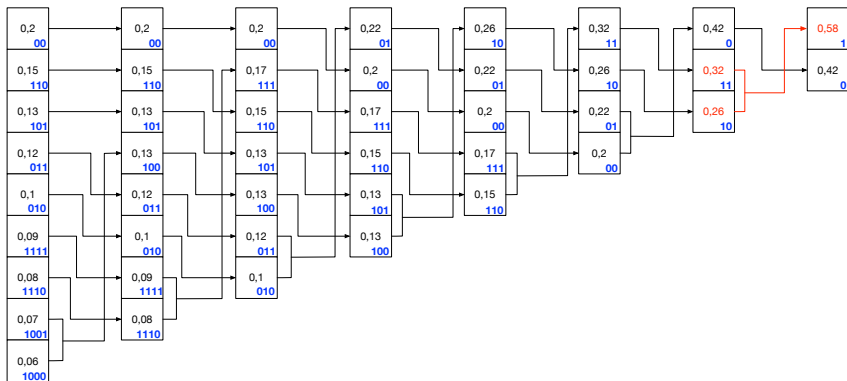
- Voy hacia atrás, añadiendo 'o' ó 'r' según corresponda:



# Codificación Huffman

## Ejemplo

- Completo el proceso:



# Codificación Huffman

## Ejemplo

- Así, finalmente:

0, 2	00
0, 15	110
0, 13	101
0, 12	011
0, 1	010
0, 09	1111
0, 08	1110
0, 07	1001
0, 06	1000



# Codificación Huffman

## Ejemplo

- Longitud media:

$$\bar{L} = 2 \cdot 0,2 + 3 \cdot (0,15 + 0,13 + 0,12 + 0,1) + 4 \cdot (0,09 + 0,08 + 0,07 + 0,06) = 3,1$$

- Se puede demostrar que se cumple que:

$$H \leq \bar{L} \leq H + 1$$

# Codificación LZW

## Definición

- Tienen la ventaja de que no necesitamos conocer las probabilidades a priori de los símbolos.
- Se utiliza, por ejemplo, en los formatos de imagen GIF y TIFF.

# Codificación LZW

## Pseudocódigo

```
CADENA = cadena vacía
WHILE quedan caracteres por codificar DO
    CHARACTER = coger el siguiente carácter
    IF CADENA+CHARACTER está en el diccionario
        CADENA = CADENA+CHARACTER
    ELSE
        código correspondiente a CADENA -> SALIDA
        Añadir CADENA+CHARACTER al diccionario
        CADENA = CHARACTER
    END
END
código para CADENA -> SALIDA
```

# Codificación LZW

## Ejemplo

- Supongamos que partimos del siguiente diccionario:

Entrada	Código
a	0
b	1
n	2

- El mensaje de entrada al codificador es: “banana”

# Codificación LZW

## Ejemplo

- Siguiendo el proceso del pseudocódigo, vamos construyendo el diccionario y codificando el mensaje:

CADENA	CARACTER	¿En el Diccionario?	Al Diccionario	Salida
	b	Si		
b	a	No	3 - ba	1
a	n	No	4 - an	0
n	a	No	5 - na	2
a	n	Sí		
an	a	No	6 - ana	4
a				0

- Al final, el mensaje codificado es: 1 0 2 4 0.

# Codificación LZW

## Decodificación

```
CODIGO_1 = Leer primer código del mensaje
Traducción de CODIGO_1 -> SALIDA
WHILE queden caracteres por decodificar
    CODIGO_2 = Leer siguiente código
    CADENA = traducción de CODIGO_2
    CADENA -> SALIDA
    CARACTER = Primer carácter de CADENA
    Añadir (Traducción de CODIGO_1)+(CARACTER) al diccionario
    CODIGO_1 = CODIGO_2
END
```

# Codificación LZW

## Ejemplo

- Siguiendo el proceso del pseudocódigo, vamos reconstruyendo el diccionario y decodificando el mensaje:

CODIGO <sub>1</sub>	CODIGO <sub>2</sub>	CADENA	CARACTER	Salida	Dicc.
1				b	
1	0	a	a	a	3 - ba
0	2	n	n	n	4 - an
2	4	an	a	an	5 - na
4	0	a	a	a	6 - ana