

# Codificación de la señal de televisión

Jorge Mata Díaz

PID\_00196651



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

# Índice

<b>Introducción</b> .....	5
<b>1. Estandar de codificación DVB MPEG-2</b> .....	9
<b>2. Codificación de fuente</b> .....	12
2.1. Entropía de la fuente .....	14
2.2. Codificación en secuencias de símbolos .....	15
<b>3. Codificación audiovisual MPEG</b> .....	18
3.1. Compresión de vídeo MPEG .....	18
3.2. Estructuración de la secuencia de vídeo MPEG .....	20
3.3. Codificación <i>intra-frame</i> (imágenes I) .....	21
3.3.1. Transformada coseno (DCT) .....	22
3.3.2. Ordenación en zig-zag .....	23
3.3.3. Cuantificación .....	24
3.3.4. Codificación a ráfagas RLC y codificación entrópica VLC .....	24
3.4. Codificación <i>inter-frame</i> .....	25
3.4.1. Codificación predictiva hacia delante (imágenes P) .....	25
3.4.2. Codificación bidireccional-predictiva (imágenes B) .....	27
3.5. Compresión de audio MPEG .....	28
<b>4. Multiplexación y <i>scrambling</i> en DVB</b> .....	32
4.1. Empaquetamiento y multiplexado MPEG .....	32
4.2. Información específica de un PROGRAMA (PSI) .....	35
4.3. <i>Scrambling</i> y acceso condicional .....	38
<b>5. Codificación de canal</b> .....	40
5.1. Aleatorización .....	41
5.2. Codificación de bloque Reed-Solomon .....	43
5.3. Entrelazado de Fourny .....	44
5.4. Codificación convolucional o continua y perforado .....	47



## Introducción

En este módulo se explican los aspectos necesarios para codificar la información audiovisual de forma que se facilite el aprovechamiento de los medios de transmisión y se contrarresten las degradaciones introducidas por dichos medios.

En primer lugar presentaremos los conceptos fundamentales de la codificación de fuente utilizando técnicas de propósito general y técnicas específicas para la información audiovisual. La teoría de codificación de fuente proporcionará la base para la reducción del volumen de datos a transmitir de un contenido audiovisual.

En segundo lugar presentaremos los mecanismos de empaquetamiento de la información y su multiplexación para un adecuado transporte de los datos en los medios de comunicación.

Finalmente, introduciremos los conceptos básicos de codificación de canal que protegerán los datos frente a las posibles distorsiones sufridas durante su transmisión.

En la figura 1 se indica el bloque de la cadena televisiva que se desarrolla en este módulo:

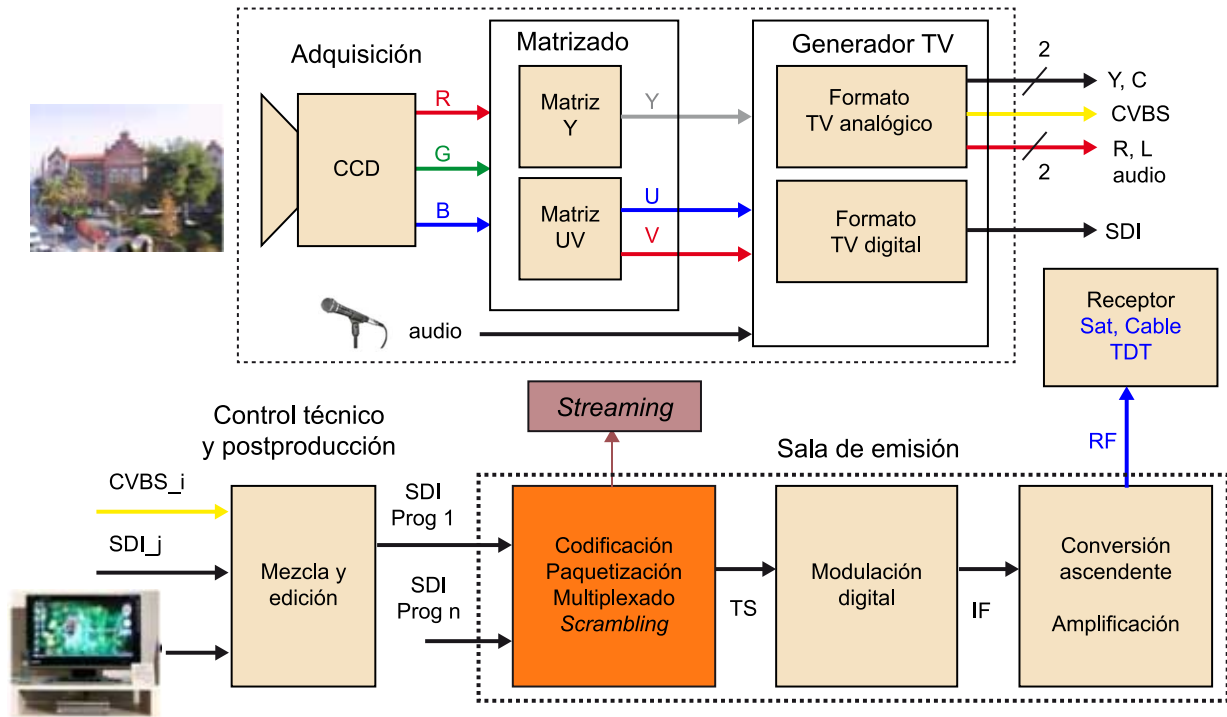


Figura 1. Cadena televisiva. En naranja se indica el bloque que se desarrolla en este módulo

El desarrollo previo de la televisión analógica predeterminó que los canales para la transmisión de la televisión tuvieran un ancho de banda convencional de 6 a 8 MHz en medios de cable o terrestre y de 27 a 36 MHz para satélite. Con estos valores, es posible plantearse transmitir señales digitales en esos mismos canales, cuyas velocidades binarias sean del orden de 30 a 40 Mbps, sin embargo, la señal digital de vídeo SDI requiere una velocidad de 270 Mbps.

Para poder convertir la señal SDI (ver el módulo "Digitalización de la señal de televisión") en una señal de televisión que pueda ser transmitida por los canales de broadcasting, se deben aplicar algoritmos de compresión. Estos se basan fundamentalmente en tres aspectos:

- La **redundancia espacial** (correlación entre píxeles vecinos) y las características de la percepción visual humana (baja sensibilidad a los detalles finos) de una imagen fija, que son la base de los métodos JPEG.
- La **redundancia temporal** entre imágenes consecutivas (base de los algoritmos MPEG).
- Las **características de la audición humana** que permite eliminar información no audible (códigos psicoacústicos).

La aplicación de todos estos algoritmos conforma lo que se denomina **codificación de fuente** específica para la señal audiovisual.

#### Ved también

La conversión de la señal SDI a señal de televisión se explica en el módulo "Digitalización de la señal de televisión".

La transmisión de la televisión digital se organiza en una multiplexación de paquetes que pertenecen a varios programas de televisión, de audio o de datos que denominaremos flujo de transporte (TS, *transport stream*). En dichos paquetes se incluye información sobre los programas que lo componen. La información de los programas de pago se encuentra cifrada mediante una técnica de *scrambling* (o de aleatorización impredecible) que imposibilita el acceso a estos programas si no se dispone de una clave para su descifrado.

El flujo de bits o *stream* que se origina en la transmisión requiere de eficientes algoritmos de detección y corrección de errores que configuran la denominada codificación de canal.

Todos estos aspectos, que se pueden resumir en la figura 2, van a ser explicados en este módulo.

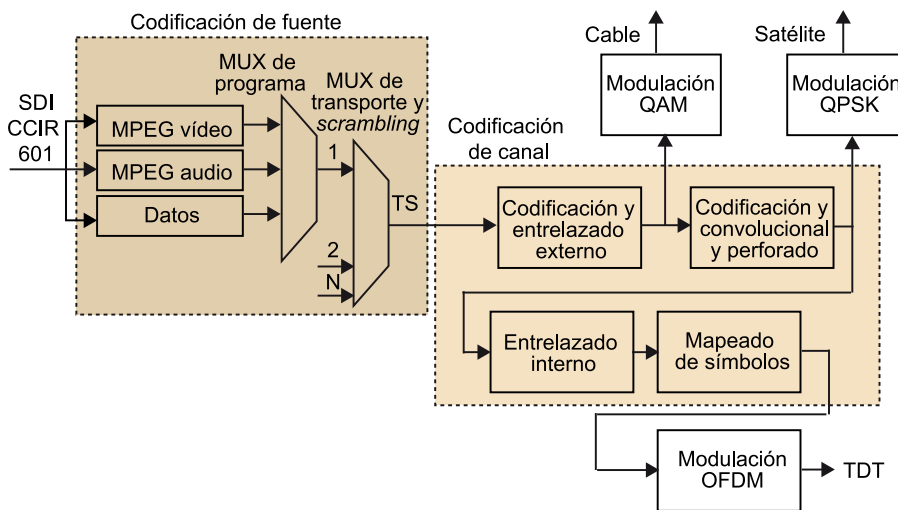


Figura 2. Diagrama de bloques de la codificación de la televisión digital





## 1. Estandar de codificación DVB MPEG-2

*Digital video broadcasting* (DVB) es un sistema de transmisión basado en el estándar MPEG-2. La recomendación MPEG-2 especifica los mecanismos de compresión de vídeo y audio, así como su multiplexación para ser almacenados o transmitidos. El DVB no es simplemente un sustituto de la difusión analógica de televisión, sino que mejora la calidad audiovisual con la posibilidad de ver imágenes en formato estándar o alta resolución, junto con sonido mono, dual, estéreo o envolvente y también permite una amplia gama de nuevas características y servicios que incluyen pistas de vídeo y audio múltiples, subtítulo, contenidos interactivos, etc.

DVB es una iniciativa europea que ha sido acogida en todos los continentes y que se ha convertido en estándar mundial para la televisión digital. Además de DVB, en EE. UU. se utiliza un sistema derivado del *advanced television systems committee* (ATSC) que es un poco diferente a DVB. Entre otras cosas ATSC adopta un estándar de codificación de audio diferente y banda lateral vestigial (*vestigial sideband modulation*, VSB) para la modulación. Sin embargo, hay mucho en común entre las normas estadounidenses y europeas.

La clave de diseño del DVB radica en que no se impone ninguna restricción sobre la naturaleza del contenido enviado, solo se centra en el transporte de la información y, por tanto, solo define la sintaxis del flujo; es decir, el orden y la diferenciación de los datos que se envían.

### Ejemplo

En el mismo flujo de datos se puede transportar una gran variedad de servicios de alta y baja velocidad de datos: desde la difusión de contenidos audiovisuales, con sus correspondientes guías de programación (*electronic program guides*, EPG) e información de tele-texto, hasta el transporte genérico de datos sobre el protocolo de Internet (IP).

El comité de estandarización Motion Pictures Experts Group (MPEG) propuso inicialmente MPEG-1 como estándar para el almacenamiento de contenidos audiovisuales en CD, que más tarde se normalizó como ISO/IEC 11172. El formato elegido para el codificador de vídeo seguía el formato CIF con submuestreo 4:2:0. El algoritmo de compresión de sonido que se especificó para el audio partía de una propuesta denominada MUSICAM.

El comité MPEG pasó posteriormente a la especificación de una nueva norma para la transmisión de contenidos audiovisuales con referencia ISO/IEC 13818, que sería adoptada como estándar de codificación para DVB. Trabajó en la definición de un estándar flexible que contemplaba la posibilidad de al-

canzar resoluciones de alta definición e imagen entrelazada, cuyo resultado fue el MPEG-2 que apareció en 1994. La normalización dio lugar a tres especificaciones diferenciadas:

- MPEG-2 vídeo (ISO / IEC 13818-2). Especifica la sintaxis de la codificación de un contenido elemental de vídeo y las operaciones necesarias para su decodificación.
- MPEG-2 de audio (ISO / IEC 13818-3). Especifica la sintaxis de la codificación de un contenido elemental de audio y las operaciones necesarias para su decodificación.
- MPEG-2 del sistema (ISO / IEC 13818-1). Define la transmisión de las unidades elementales codificadas y las relaciones temporales que deben cumplir para ser demultiplexadas y decodificadas.

El estándar MPEG-2 fue concebido para abordar el almacenamiento y distribución de contenidos audiovisuales para un amplio abanico de terminales conectados a diferentes redes. Teniendo en cuenta las características de los terminales, se especificaron distintos perfiles que se corresponderían con las capacidades de proceso y memoria del terminal al cual va destinada la codificación:

- **Perfil básico** (*simple*). Para terminales con poca capacidad de memoria y proceso, por ejemplo un teléfono móvil. El formato de vídeo digital empleado para la codificación sigue el patrón de submuestreo 4:2:0.
- **Perfil principal** (*main*). Para los terminales con mayores prestaciones, como televisores u ordenadores. También contempla como formato de entrada de vídeo digital el 4:2:0.
- **Perfil alto** (*high*). Para dispositivos de altas prestaciones como los televisores inteligentes o *set-top-boxes* con un hardware dedicado para la decodificación. Permite el uso de los formatos 4:2:0 y 4:2:2 como formatos de entrada de vídeo.

En la práctica, los perfiles que se utilizan son el perfil principal y el perfil alto.

Según las redes de transporte (satélite, cable, Internet, etc.) y la resolución de pantalla establecida para los dispositivos, se definen diferentes niveles de codificación de vídeo. Cuando el ancho de banda no sea una limitación, el factor determinante de la codificación a aplicar será el perfil del dispositivo y la presentación de vídeo requerida. En la práctica, muchas veces será el ancho de banda del medio de transmisión el condicionante principal de la codificación. Dependiendo del ancho de banda disponible, se podrá alcanzar una tasa máxima de transmisión que limitará la resolución de la imagen codificada a pesar de que el receptor pudiera llegar a presentar imágenes con mayor calidad.

En MPEG-2 se definen tres niveles de codificación de imagen asociados a la resolución de imagen y ancho de banda disponible que van desde el formato CIF hasta HD:

- **Nivel bajo** (*low*). Se corresponde con la capacidad del decodificador MPEG-2 para reproducir codificaciones en resolución CIF, 352 muestras/línea × 288 líneas/imagen.
- **Nivel principal** (*main*). Asociado a la televisión estándar, o calidad de disco DVD, 720 muestras/línea × 576 líneas/imagen.
- **Nivel alto** (*high*). Correspondiente a la televisión de alta definición. 1920 muestras/línea × 1152 líneas/imagen.

Así pues, un sistema combina un perfil con un nivel y se simboliza Perfil@Nivel. En este sentido, la combinación más empleada ha sido el perfil principal con el nivel principal, MP@ML, que corresponde a la codificación de la televisión estándar. La tasa binaria requerida en la difusión de un programa con estas características requiere de hasta 6 Mbps.

Debe tenerse en cuenta que el ancho de banda ocupado con la codificación MPEG-2 en un medio de transmisión es, para la misma resolución de imagen, la sexta parte de la usada en transmisión analógica. En otras palabras, en el ancho de banda donde antes se transmitía un programa de televisión analógica, ahora se transmiten seis programas, de equivalente calidad, en MPEG-2.

## 2. Codificación de fuente

Para reducir notablemente la velocidad requerida para transmitir un contenido audiovisual es necesario caracterizar la fuente de información. La caracterización de la fuente se basa fundamentalmente en la naturaleza de la información que genera y en su estadística. Podremos aplicar técnicas de compresión específicas y generales sobre los datos generados por la fuente que nos facilitarán la eliminación de gran parte de los elementos redundantes, reduciendo notablemente el espacio de almacenamiento o el ancho de banda de transmisión.

En términos generales, para cualquier fuente de información se pueden aplicar técnicas basadas en la entropía propia de la fuente. Para ello, inicialmente, se debe identificar el conjunto de símbolos que puede emitir una fuente de información. Este conjunto de símbolos definirá el **alfabeto de la fuente**.

### Ejemplo

Para el caso audiovisual podríamos considerar en vídeo los posibles valores que podrían tomar los píxeles de una imagen, o en audio, los valores de las muestras obtenidas a través de un micrófono. Estos valores formarían, respectivamente, el alfabeto de los símbolos de la fuente de vídeo y el alfabeto de los símbolos de la fuente de audio.

Una vez determinado el alfabeto de una fuente, se pueden aplicar técnicas de codificación. La codificación es propiamente la asignación de una palabra código a un símbolo de fuente o, de forma más general, a una sucesión de símbolos de fuente. En la práctica, las palabras código siempre son sucesiones de dígitos binarios.

### Ejemplo 1

Una fuente de vídeo cuyos píxeles toman valores dentro de una escala de cuatro grises tendría por alfabeto de fuente: {0, 1, 2, 3}. Se podría codificar asignando, respectivamente las palabras código de dos dígitos binarios {00, 01, 10, 11}. Esta codificación sería una codificación de longitud constante puesto que todos los símbolos de la fuente se han codificado con palabras código de longitud 2.

Ejemplo de codificación de longitud constante para un alfabeto de 4 símbolos

Símbolo fuente	Codificación binaria
0	00
1	01
2	10
3	11

Para valorar si una codificación es mejor o peor deberíamos tener en cuenta cuántos dígitos binarios han sido necesarios para codificar un determinado mensaje. En nuestro ejemplo, podríamos considerar una imagen de 1.000 píxeles como el mensaje a codificar. Aplicando a la imagen la codificación de longitud constante propuesta, se necesitarían 2.000 dígitos binarios para ser transmitida o almacenada.

La codificación de fuente nos permitirá caracterizar la fuente de manera que podamos encontrar una codificación que reduzca las necesidades de almacenamiento o transmisión de la fuente. La reducción obtenida se fundamenta en asignar palabras código más cortas a símbolos de fuente más probables y, viceversa, palabras código más largas a símbolos de fuente menos probables. El resultado obtenido es, habitualmente, una codificación de longitud variable.

La codificación de fuente trata de determinar la estadística subyacente en la generación de los símbolos del alfabeto de la fuente y, a partir de ese conocimiento, aplica una codificación.

### Ejemplo 2

Para la fuente del ejemplo anterior, supondremos que las frecuencias de aparición de los símbolos del alfabeto de fuente {0, 1, 2, 3} son  $\frac{1}{4}$ ,  $\frac{1}{8}$ ,  $\frac{1}{8}$  y  $\frac{1}{2}$ , respectivamente. La codificación debería premiar con menos dígitos binarios a los símbolos 0 y 3, mientras que debería penalizar a los símbolos 1 y 2. Una posible codificación de longitud variable asignaría, por ejemplo, las siguientes palabras código a cada uno de los símbolos de fuente:

Ejemplo de codificación de longitud variable para un alfabeto de 4 símbolos

Símbolo fuente	Codificación binaria
0	00
1	010
2	011
3	1

Como observamos, al símbolo más probable se le ha asignado una palabra de únicamente 1 dígito binario y palabras código de longitud 3 a los menos probables. Para este caso, la codificación binaria de una imagen de 1.000 píxeles requerirá de:

$$1000 \left( \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 + \frac{1}{2} \cdot 1 \right) = 1000 \cdot 1.75 = 1750 \text{ dígitos binarios} \quad (1)$$

Como vemos se han reducido las necesidades de almacenamiento o transmisión de 2.000 a 1.750 dígitos binarios. Es decir, se ha comprimido al 87,5%, por lo que se requerirá de un 12,5 % menos de espacio de almacenamiento o ancho de banda para el mismo contenido.

Para determinar la bondad de una codificación, en vez de realizar los cálculos sobre un mensaje compuesto por una cantidad de símbolos de fuente, se determina el valor medio de la longitud de las palabras código resultantes. Así, para el primer caso, de longitud constante, todas las palabras código son de 2 dígitos. En el segundo caso, el número medio de dígitos binarios de una palabra código será 1,75 dígitos. Obviamente, el segundo codificador es más eficiente porque requiere menos dígitos de media por símbolo de fuente.

El proceso de codificación extrae las redundancias de los datos hasta llegar a una representación de la información que requiere un cierto número de dígitos binarios que se aproxima, o incluso llega, a un valor mínimo. El valor mínimo marca una cota en la que se puede garantizar que se ha extraído toda la redundancia sin ninguna pérdida de información. Este tipo de codificación recibe el nombre de codificación sin pérdidas y se utiliza cuando se desea que la codificación no introduzca ningún tipo de distorsión sobre los datos originales.

En el caso de que la codificación empleara menos dígitos de los que impone el valor mínimo entonces, el proceso de descodificación no podría garantizar la recuperación de la información original; es decir, la información descodificada podría ser distinta de la codificada. En este caso, la codificación admite pérdidas. En el campo audiovisual es común aplicar técnicas de codificación con pérdidas que puedan introducir de forma ordenada un grado de distorsión entre la información original y la descodificada, debido a que la percepción humana admite ciertas degradaciones durante la presentación de la información.

## 2.1. Entropía de la fuente

La cota inferior que determina hasta cuánto se puede comprimir sin necesidad de introducir pérdidas en los contenidos viene determinada por la propia fuente. La estadística de la fuente define un grado o magnitud de desorden y variación de los valores de los símbolos generados que se denomina entropía. La fuente con mayor grado de entropía es aquella que emite sus símbolos de forma equiprobable e incorrelada. La emisión incorrelada hace que una fuente no tenga en cuenta las emisiones anteriores (no tiene memoria) para la siguiente generación. La entropía de una fuente ( $H$ ) sin memoria se define a través de la siguiente expresión:

$$H = \sum_{i=1}^K p_i \cdot \log_2(1/p_i) \quad (2)$$

### Lectura de la fórmula

Donde  $K$  es el número de símbolos que puede generar la fuente y  $p_i$  es la probabilidad de emisión de cada uno de ellos.

La unidad empleada para cuantificar la entropía es el bit de información por símbolo.

Si todos los símbolos son equiprobables tendremos que  $p_i=1/K$ , por lo que vemos que el mayor valor posible de entropía de una fuente es  $\log_2(K)$ .

*A priori*, si se desconoce el comportamiento de una fuente, la mejor hipótesis sobre su estadística es considerarla equiprobable. Como resultado, la codificación óptima asignará palabras código de una longitud muy similar, o igual, a todos los símbolos de fuente. Este es el proceso que hemos seguido para el primer ejemplo: teníamos una fuente que podía emitir 4 símbolos, por lo que la máxima entropía que podría tener esta fuente sería de  $\log_2(4)= 2$  bits por símbolo. Sin embargo, la estadística descrita por la fuente (segundo ejemplo) no muestra un comportamiento equiprobable, por lo que la entropía debería ser menor. Así, en este caso la entropía calculada es la siguiente:

$$H = \frac{1}{4} \log_2(4) + \frac{1}{8} \log_2(8) + \frac{1}{8} \log_2(8) + \frac{1}{2} \log_2(2) = 1.75 \text{ bit/símb} \quad (3)$$

El proceso de codificación sin pérdidas está limitado por la entropía, de forma que la longitud media de las palabras código binarias nunca podrá ser inferior a la entropía. En nuestro segundo ejemplo, hemos conseguido una codificación sin pérdidas óptima puesto que la longitud media obtenida es de 1,75 dígitos por símbolo que se corresponde exactamente con la entropía de la fuente. Por lo tanto, en este caso, 1 dígito binario lleva 1 bit de información.

El algoritmo de codificación debe intentar conseguir acercarse al máximo la longitud media de la codificación binaria a la entropía de la fuente. En general, se obtiene como resultado una codificación entrópica de longitud variable denominada (*variable length coding*, VLC).

## 2.2. Codificación en secuencias de símbolos

Además de estudiar la frecuencia de generación de los símbolos en una fuente se pueden tener en cuenta las relaciones en las secuencias de símbolos consecutivos. Muchas de las fuentes audiovisuales presentan un alto grado de correlación en los símbolos emitidos, de forma que, tras la emisión de un símbolo, es muy probable emitir otro igual o de valor muy semejante.

Cuando en una fuente la probabilidad de emitir un símbolo depende de las emisiones anteriores decimos que la fuente **tiene memoria**. Esta característica reduce la entropía de la fuente puesto que la emisión de los símbolos está más ordenada que si la fuente tuviera una generación incorrelada.

En fuentes con memoria sabemos más acerca de la emisión, ya que la generación actual condicionará las emisiones futuras. Este condicionamiento se puede aprovechar a la hora de codificar. Las fuentes con memoria se suelen codificar de forma que a una cadena de símbolos de fuente consecutivos se le hace corresponder una única palabra código.

### Ejemplo 3

Para la imagen con 4 niveles de grises estudiada anteriormente {0,1,2,3}, se podría realizar una codificación de longitud variable para cada par de píxeles. Así tendríamos la siguiente codificación de pares de píxeles:

Ejemplo de codificación de secuencias de 2 símbolos

Cadena	Codificación	Cadena	Codificación
00	1011	20	11110
01	10100	21	111110
02	10101	22	111111
03	100	23	1110
10	11010	30	010
11	110110	31	0110
12	110111	32	0111
13	1100	33	00

En situaciones de largas generaciones consecutivas del mismo símbolo se utilizan alternativas de codificación a los símbolos de fuente. Este es el caso de la codificación de imágenes por líneas, donde los valores de los píxeles en la misma línea de imagen se suelen repetir consecutivamente un número elevado de veces. Estas fuentes muy correladas se caracterizan por la emisión de símbolos iguales en forma de ráfagas. Una técnica eficiente de codificación se basa en codificar la longitud de la ráfaga y el valor del símbolo repetido.

### Ejemplo

Si tuviéramos la cadena BBBBBAACCC se podría codificar como 5B2A3C.

Para aumentar todavía más la eficiencia de la codificación de fuente, se puede caracterizar la frecuencia con que aparecen las ráfagas de una determinada longitud y la frecuencia del valor que toma la ráfaga. Una vez establecida la codificación para la longitud y la codificación para el valor del símbolo, el



codificador genera por cada ráfaga un par de palabras códigos asociadas a la longitud y valor. Este mecanismo de codificación se denomina *run length coding* (RLC).

Como se puede observar en las técnicas anteriores de codificación, el conocimiento de la naturaleza de la información generada por una fuente facilita el desarrollo de técnicas de codificación específicas para una fuente. En muchos casos, estas técnicas nos permiten aproximarnos a las cotas marcadas por la entropía sin introducir pérdidas. En particular, para la información audiovisual se pueden utilizar técnicas muy específicas, que junto a las de propósito general, nos permitirán reducir notablemente los tiempos de transmisión de la información. Dada la eficiencia de las técnicas de codificación aplicadas, consideraremos que los dígitos binarios resultantes de una codificación equivalen, prácticamente, a un bit de información. Por ello, consideraremos indistintamente el bit y el dígito binario a partir de este apartado.

### 3. Codificación audiovisual MPEG

La naturaleza de la información audiovisual hace posible el desarrollo de técnicas específicas para su codificación. Por lo que respecta a la distribución de contenidos audiovisuales, las técnicas empleadas aprovechan la caracterización de la percepción humana para desarrollar estrategias que maximicen la fidelidad de las señales reproducidas en aquellos aspectos donde es más sensible el ser humano.

#### **Ejemplo**

Si pensamos en imágenes, el sistema visual humano tiene mayor sensibilidad a la luminancia que a la crominancia. Algo parecido ocurre en el caso del audio, donde existen bandas frecuenciales que provocan mayor estimulación al sistema auditivo con la misma potencia que otras.

La compresión MPEG audiovisual contempla la posibilidad de introducir pérdidas de información durante el proceso de codificación que darán lugar a una falta de fidelidad o discrepancia entre el contenido original y el contenido descodificado. Por lo tanto, la compresión MPEG es, en general, una codificación con pérdidas que permite rebasar el límite que marca la entropía. Sin embargo, la distorsión introducida al aumentar la compresión puede tener un mínimo impacto en la percepción humana si se codifican los elementos de la información de forma acorde a la sensibilidad del ser humano.

#### **Ejemplo**

El sistema visual humano presta mucha menos atención a los contornos que a la parte central de las imágenes. Lo mismo ocurre con el audio, donde el sistema auditivo del ser humano dispone de mucha menor sensibilidad en las frecuencias más altas del sonido.

#### 3.1. Compresión de vídeo MPEG

La compresión MPEG explota la naturaleza de la información de vídeo para definir una técnica eficiente para la codificación de la mayoría de las secuencias y se fundamenta en que los píxeles que están próximos espacialmente en una misma imagen, o en imágenes consecutivas, suelen ser en la mayoría de los casos de valores muy parecidos.

#### **Nota**

Debemos tener presente que los objetos que aparecen en una imagen disponen de una superficie que suele ocupar un número elevado de píxeles contiguos y que estos objetos aparecen repetidos en muchas imágenes consecutivas de una misma escena. El objetivo de la codificación será extraer toda la redundancia que hay dentro de una misma imagen y entre imágenes próximas temporalmente.

Cuando se aplican técnicas para la reducción de la redundancia espacial (parecido dentro de la imagen) se realiza una codificación que denominamos *intra-frame* (entendiendo *frame* como imagen). Cuando la redundancia extraída es temporal (parecido entre imágenes) se realiza una codificación que denominamos *inter-frame* o también, predictiva:

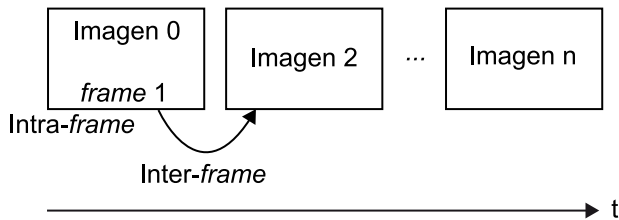


Figura 3. Tipos de extracción de la redundancia en los píxeles de la imagen

En la codificación *inter-frame* se lleva a cabo una codificación predictiva hacia delante (*P-frame*) que explota el parecido de los píxeles de la imagen a codificar con los de una imagen previamente codificada. Se busca la mejor predicción desde la imagen previa para reconstruir la actual, para ello lo que se codifica es la diferencia entre una zona de la imagen previa y una zona de la actual imagen. Las diferencias codificadas son un conjunto de residuos o errores de predicción que, en principio, deberían tomar valores pequeños y muy incoherentes cuando el parecido entre las imágenes sea muy alto.

Para mejorar la codificación de la redundancia temporal, aumentado la posibilidad de encontrar zonas de imagen parecidas entre imágenes, se incorporó una segunda forma de codificación *inter-frame* denominada codificación **bidireccional-predictiva** (*B-frame*), que permite que la codificación de un *frame* utilice como referencias porciones de imagen de un *frame* temporalmente anterior y otro temporalmente posterior al que se quiere codificar. Esta opción de codificación conlleva codificar *frames* en un orden distinto al orden temporal de presentación de las imágenes de una secuencia.

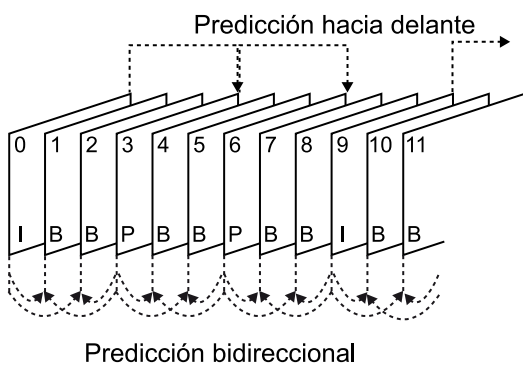


Figura 4. Ejemplo de codificación predictiva hacia delante y bidireccional

### 3.2. Estructuración de la secuencia de vídeo MPEG

Para proceder a la codificación de la información de una secuencia de vídeo es necesario estructurar la información que se va a codificar. Para ello se han definido un conjunto de unidades de información de vídeo que estructuran los elementos a codificar:

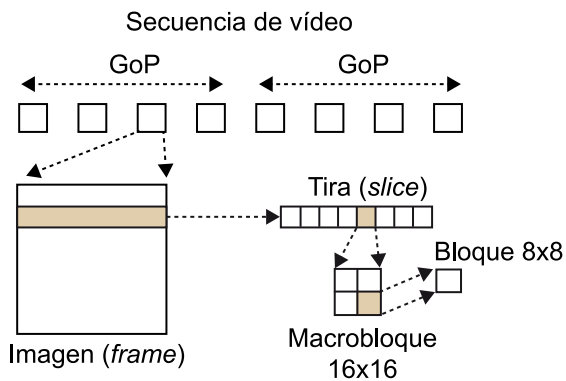


Figura 5. Estructuración de la secuencia de vídeo

En primera instancia se define propiamente la secuencia que estará compuesta por todas las imágenes del contenido audiovisual. La secuencia se organiza en grupos de imágenes llamados GoP (*group of pictures*), que contendrán un conjunto de *frames* que podrán estar relacionados en su codificación.

Los *frames* se estructuran en bloques de 8x8 píxeles, que se agrupan formando macrobloques. Los macrobloques están definidos por 4 bloques de 8x8 píxeles de luminancia. Así pues, el macrobloque constituye una zona cuadrada de la imagen de 16x16 píxeles de luminancia y los valores asociados de las componentes de croma. De forma que si la imagen está submuestreada con formato 4:2:0, el macrobloque estará compuesto por 4 bloques de luminancia y 1 bloque de croma por cada componente de color.

Los macrobloques en posiciones consecutivas, de izquierda a derecha y de arriba abajo dentro de una imagen, se suelen agrupar libremente para determinar un tira de imagen o *slice*. En la práctica, la mayoría de los codificadores utilizan como tira de imagen todos los macrobloques que están situados en la misma posición vertical de la imagen.

Para la codificación de las imágenes dentro de un GoP se aplica un patrón de codificación regularmente, en el cual se emplean las técnicas *intra-frame* o *inter-frame* en las que se basa estándar MPEG-2. En un GOP podremos encontrar:

- **Imágenes I o *intra-frames*.** No usan predicción, no dependen de imágenes anteriores. Sirven de puntos de resincronización en decodificación cuando se producen pérdidas o errores en el canal de transmisión. Pueden ser empleadas para facilitar una visualización rápida del contenido en la secuencia.

- **Imágenes P o *predicted-frames*.** Imágenes con codificación predictiva hacia adelante. Usan información de una imagen I o P anterior.
- **Imágenes B o *bidirectional-predicted-frames*.** Imágenes que usan información de una o varias imágenes I o P tanto anteriores como posteriores.

Puesto que el patrón de codificación se suele mantener constante durante toda la codificación, se definen dos magnitudes para determinar dicho patrón. A la distancia entre imágenes del tipo I, se le llama N, mientras que a la distancia entre imágenes P o entre I-P se le llama M. En la práctica un GoP se inicia cuando aparece una nueva codificación de una Imagen I, por lo que el tamaño N también indica el número de imágenes que contiene un GOP. El valor de N es un múltiplo de M, de esta forma se garantiza que el número de imágenes B consecutivas es siempre igual y de valor M-1.

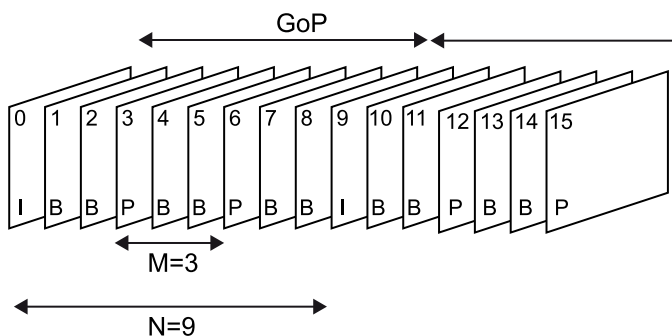


Figura 6. Ejemplo de codificación de imágenes en un GoP

### 3.3. Codificación *intra-frame* (imágenes I)

La codificación *intra-frame* realiza una codificación de todos los bloques de la imagen macrobloque a macrobloque. De manera que, para una imagen en formato 4:2:0, en cada macrobloque primero se codifican los cuatro bloques de luminancia y luego el bloque de croma azul y luego el de croma rojo.

La siguiente figura esquematiza el proceso seguido para cada macrobloque de la imagen que se codificará en modo *intra-frame*:

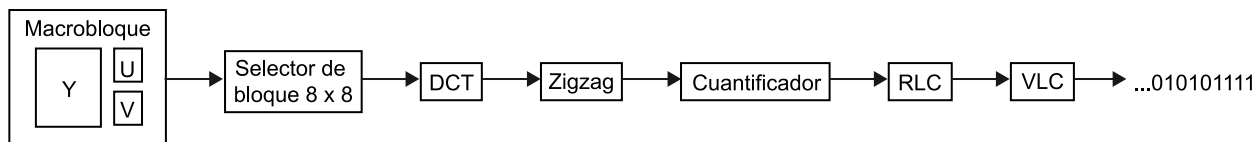


Figura 7. Esquema de codificación intra

Como se representa en la figura, los 6 bloques de cada macrobloque siguen el mismo proceso de codificación. El proceso se puede descomponer en una serie de operaciones básicas que se aplican a cada bloque hasta realizar su codificación. Estas operaciones están diseñadas con los siguientes propósitos:

- Ordenar la información del bloque (DCT y zig-zag) para facilitar su codificación.
- Realizar una codificación introduciendo pérdidas selectivas (cuantificador).
- Obtener un alto grado de compresión explotando un análisis previo del comportamiento estadístico de las fuentes de vídeo (RLC y VLC).

### 3.3.1. Transformada coseno (DCT)

Para desarrollar una codificación eficiente de una porción de una imagen tenemos que observar dos características fundamentales que aplican en la mayoría de los casos:

- Las imágenes suelen presentar un alto parecido de los píxeles en una misma zona.
- El parecido de los píxeles de una zona de imagen suele decrecer lentamente.

Estas dos características de los píxeles de la imagen pueden aprovecharse para desarrollar una codificación óptima que intente reducir al máximo el número de valores necesarios para representar la información contenida en la porción de imagen. Por lo tanto, el resultado es una representación equivalente de la información denominada transformada coseno discreta (DCT, *discrete cosine transform*). Esta técnica se ha utilizado desde el primero de los estándares MPEG-1 y se sigue aplicando en los actuales como MPEG-2 y MPEG-4.

La transformada DCT es un caso particular de la transformada discreta de Fourier, donde la información se puede representar como una serie de cosenos en fase. Para el caso de las imágenes será una transformación bidimensional donde porciones de imagen definidas por el valor de sus píxeles serán representadas por sus coeficientes transformados. La DCT es una función reversible por lo que no introduce ninguna distorsión. La DCT es simplemente una forma equivalente de representar la misma información y presenta como ventaja principal con respecto a la transformada de Fourier discreta que necesita la mitad de coeficientes para representar la misma información.

Las porciones de imagen que se utilizan para la aplicación de la transformada DCT son bloques rectangulares de píxeles que en los estándares de codificación MPEG suelen ser de 8x8 píxeles. En general, estos bloques dispondrán de valores de píxeles más parecidos cuanto mayor sea la resolución de la imagen a codificar. De forma equivalente, cuanto mayor sea el bloque empleado, mayor es la probabilidad de que los píxeles tomen valores más dispersos.

La transformación de un bloque de  $8 \times 8$  píxeles da lugar a un conjunto de  $8 \times 8$  coeficientes. En la práctica, los bloques y coeficientes se organizan matricialmente. El primer elemento de la matriz se indexa en la posición (0,0) y el último en la posición (7, 7).

Al aplicar la DCT, cada coeficiente indica la intensidad presente de una frecuencia dentro del bloque. El primer coeficiente (0,0), o coeficiente de continua, representa el valor medio de los píxeles del bloque. El resto de coeficientes, o coeficientes de alterna, tomarán un valor que dependerá de cómo sean las variaciones en el valor de los píxeles de la misma posición horizontal o vertical: si los valores de píxeles consecutivos en una dirección presentan variaciones lentas, el resultado será que los coeficientes próximos al de continua, que denominaremos coeficientes de baja frecuencia, dispondrán de un elevado grado de intensidad; si las variaciones de los píxeles son rápidas o bruscas, entonces serán los coeficientes más alejados del de continua, o coeficientes de alta frecuencia, los que presentarán mayores valores.

En la mayoría de casos, los bloques de  $8 \times 8$  píxeles de una imagen tendrán valores muy parecidos y correlados, por lo que los coeficientes de continua y de baja frecuencia serán aquellos que presentarán mayor intensidad. En los casos donde los píxeles de un bloque sean muy dispares e incorrelados, entonces todos los coeficientes presentarán un grado de intensidad parecido. Por lo tanto, para el primer caso, la energía del bloque está concentrada en las bajas frecuencias mientras que en el segundo la energía está distribuida en todas las frecuencias por igual.

Decimos que un bloque presenta mayor complejidad cuando su energía está más dispersa entre sus frecuencias porque los valores de los píxeles están más desordenados. Al estar más desordenados presentan mayor entropía y, por lo tanto, necesitaremos más bits para poderlos codificar.

### 3.3.2. Ordenación en zig-zag

En la mayoría de los bloques, la DCT consigue representar su información con un número reducido de coeficientes de baja frecuencia, manteniendo nulos o próximos a cero los de alta frecuencia. En esta situación, los coeficientes de intensidad notable serán pocos y aparecerán ordenados de mayor a menor intensidad conforme nos vayamos alejando de la componente de continua.

Con el fin de facilitar el proceso de codificación, se realiza una ordenación de las muestras siguiendo un recorrido en zig-zag que se va alejando paulatinamente del coeficiente de continua (0,0) hasta el (7,7). Esta ordenación nos permite pasar de una representación bidimensional a una unidimensional de 64 elementos. Una vez ordenados los coeficientes, y teniendo en cuenta que

la mayoría de ellos estarán próximos a 0, se podrán aplicar fácilmente técnicas clásicas RLC de codificación en ráfagas para las secuencias de ceros consecutivos.

### 3.3.3. Cuantificación

Puesto que MPEG considera la posibilidad de introducir degradaciones en la información de vídeo, después de la operación de ordenamiento zig-zag, se aplica la cuantificación de los coeficientes de la DCT, que implica un redondeo de los coeficientes a un valor más o menos próximo. Si se aplica un paso de cuantificación pequeño, el redondeo que puede llegar a sufrir un coeficiente también será pequeño y, por el contrario, si se utiliza un paso de cuantificación grande, el error entre el valor original y el redondeado puede ser notable. Este error lo denominaremos error de cuantificación.

El paso de cuantificación,  $Q$ , puede tomar valores que suelen ir de 1 a 31. Así, para  $Q = 1$  podemos decir que los coeficientes en general tienen un error de cuantificación que afecta fundamentalmente al bit de menor peso, mientras que para  $Q = 31$ , el error puede llegar a afectar a los 5 bits de menor peso del coeficiente. Por lo tanto, cuando se emplea un paso de cuantificación relativamente grande, la mayoría de los coeficientes de la DCT se redondean al valor 0, lo que nos lleva a aumentar las secuencias de ceros seguidos que debemos codificar después de la cuantificación.

### 3.3.4. Codificación a ráfagas RLC y codificación entrópica VLC

La ordenación frecuencial de menor a mayor frecuencia realizada con el zig-zag y la posterior cuantificación dan lugar a que la información a codificar de un bloque esté representada por secuencias de 64 coeficientes cuantificados. Estas secuencias disponen de largas sucesiones de valores nulos que facilitan una eficiente codificación. Para este tipo de datos lo más apropiado es utilizar la codificación por ráfagas RLC estudiada anteriormente. El codificador determina para cada sucesión la longitud de la ráfaga de ceros y el valor que la sucede. Este par de elementos (longitud, valor) disponen de una codificación predeterminada que se halla en la tabla de códigos VLC (*variable length coding*).

#### Recordad

La tabla de códigos VLC se habrá construido previamente según la probabilidad de los elementos a codificar.

En este caso, los valores de codificación VLC se obtuvieron analizando la estadística de los bloques transformados de muchas secuencias distintas. A partir de este estudio se determinan las frecuencias de aparición de las ráfagas de ceros y los valores que las suceden y definen la codificación de los elementos (longitud, valor).

El coeficiente de continua se codifica de una manera diferente puesto que depende del valor medio de los píxeles del bloque. La codificación aplicada es diferencial respecto al coeficiente de continua codificado para el bloque anterior. En este caso, se considera que bloques adyacentes disponen de valores



semejantes en sus píxeles, por lo que la diferencia entre la componente de continua de un bloque y de otro próximo debe ser relativamente pequeña. Para iniciar la codificación del primer coeficiente de continua se parte del valor central que pueda ser codificado.

La cuantificación del valor de continua se suele realizar con muy poco redondeo, de forma que no se comentan errores notables entre un bloque y cualquiera de sus colindantes. La estadística de los valores diferenciales cuantificados determina las palabras generadas en el proceso de codificación VLC.

El error introducido en las muestras de continua provoca la percepción del denominado efecto bloque en las imágenes presentadas. Así, se puede observar sobre la imagen zonas rectangulares con distintos valores de los píxeles. Los efectos de la cuantificación en los coeficientes de alterna provocan fundamentalmente la difuminación de los contornos de los objetos dentro de la imagen.

### **3.4. Codificación *inter-frame***

La codificación *inter-frame* intenta extraer de forma eficiente la redundancia temporal entre imágenes consecutivas de la secuencia. Para llevar a cabo el proceso de extracción de la redundancia, se determinan inicialmente la imagen o imágenes que se van a emplear como referencia para realizar la codificación de la imagen actual.

#### **3.4.1. Codificación predictiva hacia delante (imágenes P)**

Cuando se aplica una codificación predictiva, el *frame* P (*predicted*) a codificar utiliza el *frame* I o P inmediatamente anterior, que ya habrá sido codificado y almacenado en memoria para ser utilizado como referencia.

La porción de imagen que se emplea para realizar la predicción es el macrobloque. En el momento de realizar la codificación predictiva de una imagen se procede macrobloque a macrobloque en orden secuencial (de izquierda a derecha y de arriba a abajo). Para cada macrobloque se busca la porción de imagen del mismo tamaño en la imagen de referencia cuyos píxeles tengan menos diferencia con los píxeles del macrobloque a codificar. Esta búsqueda de la porción de imagen más parecida en la imagen de referencia no se realiza sobre toda la superficie de la imagen. Se considera que la búsqueda debe estar acotada a un determinado rango de desplazamiento relativo a la posición del macrobloque a codificar que vendrá determinado por la resolución de la imagen, de forma que para imágenes más grandes se debería utilizar un rango mayor de búsqueda. Para realizar la búsqueda se aplican algoritmos expertos que intentan realizar el mínimo número de cálculos. Una vez seleccionada la porción de la imagen de referencia, se indicará cuál es mediante sus coordenadas, expresadas mediante el denominado vector de compensación de movimiento.

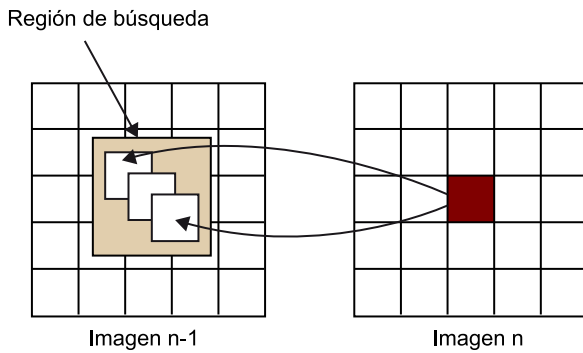


Figura 8. Rango de búsqueda para la codificación predictiva

En la codificación de un macrobloque predictivo, además del vector de movimiento, se puede codificar la diferencia del macrobloque respecto a su referencia. Para ello se obtiene un macrobloque residual resultado de la diferencia entre cada píxel del macrobloque a codificar y de la porción de imagen seleccionada como referencia. Si existe un alto grado de parecido entre ambos conjuntos de píxeles, el macrobloque diferencial dispondrá de un conjunto de valores con comportamiento altamente aleatorio y próximo a 0. Si no es así, el codificador puede decidir en cada macrobloque de la imagen predictiva si renuncia a la codificación diferencial. En ese caso, el codificador puede realizar la codificación del macrobloque de forma autónoma aplicando una codificación como se realizaría si fuera un macrobloque de una imagen donde se aplicara la codificación *intra-frame* explicada anteriormente. Por ello, el codificador señalará para cada macrobloque de una imagen predictiva el tipo de codificación aplicado en cada uno de ellos, sea predictiva o intra.

En los macrobloques diferenciales se aplica la misma metodología que para la codificación *intra-frame*: DCT, zig-zag, cuantificación y RLC con VLC para cada bloque. La principal diferencia entre ambos casos radica en que se considera que la energía de la información diferencial está distribuida uniformemente entre todas las frecuencias, lo que lleva a cuantificar igual todos los coeficientes frecuenciales, incluso el coeficiente de continua. Por lo tanto, la codificación de cada bloque después del zig-zag y la cuantificación dará lugar a ráfagas de ceros y valores cuya estadística será diferente a la de los *frames* I. Por ello, las tablas de codificación VLC serán distintas para la codificación predictiva de un macrobloque respecto a la codificación intra.

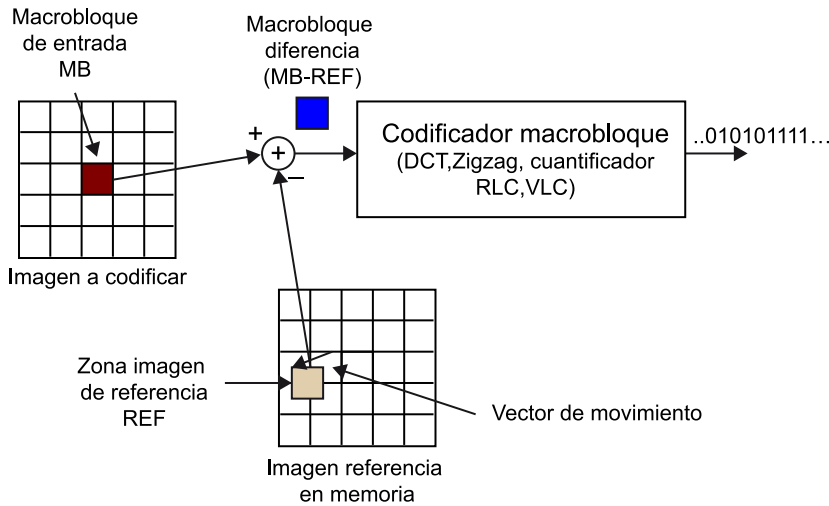


Figura 9. Esquema de codificación predictiva

### 3.4.2. Codificación bidireccional-predictiva (imágenes B)

Para facilitar la extracción de la redundancia temporal, los estándares de codificación MPEG incluyen la posibilidad de utilizar dos imágenes de referencia en el momento de la codificación. Estas imágenes de referencia se corresponden, respectivamente, con la imagen I o P anterior y con la I o P posterior. Para poder ser empleadas en la codificación de la imagen actual, es necesario que ambas hayan sido previamente codificadas, lo que implica que el orden de codificación y transmisión no es el mismo que el orden de captura y presentación.

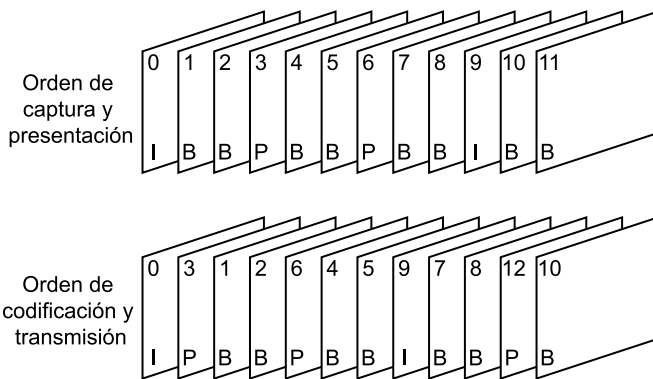


Figura 10. Modificación del orden transmisión-codificación respecto al de captura-presentación

La codificación de las imágenes bidireccional-predictivas, o imágenes B, se realiza de forma muy semejante a la explicada para las predictivas. La generalización, en este caso, se basa en la selección de la referencia que empleará un macrobloque; para ello, se realiza una búsqueda de una porción de imagen semejante en la imagen de referencia anterior, y también en la posterior. Cada porción quedará definida por un vector de compensación de movimiento. Una vez halladas las porciones de imagen, el codificador deberá decidir si utiliza la referencia hacia atrás, hacia delante o un valor promedio de ambas. Para ello, comparará el valor de los píxeles del macrobloque a codificar con los de la referencia anterior, posterior o una semisuma de ambas, y seleccionará el

que más se parezca. De esta manera, la codificación del macrobloque podrá tener un solo vector de compensación de movimiento, para los dos primeros casos, o los dos vectores si se decide el último caso.

Una vez fijada la mejor referencia, en última instancia se debe analizar la aleatoriedad del macrobloque diferencial. Como en el caso predictivo, el codificador puede decidir utilizar una codificación intra para cada macrobloque si lo considera conveniente. Si el grado de aleatoriedad es alto, se realizará la codificación del macrobloque diferencial aplicando exactamente el mismo proceso que en el caso predictivo: DCT, zig-zag, cuantificador y RLC con VLC por bloque del macrobloque.

### 3.5. Compresión de audio MPEG

La compresión de audio empleada principalmente para la transmisión se deriva de MUSICAM (*masking pattern adapted universal subband integrated coding and multiplexing*) desarrollado para la transmisión de radio digital. El estándar MPEG-2 define además otras técnicas de codificación que no se utilizan habitualmente en la transmisión de audio en DVB. La recomendación que fue adoptada inicialmente para la televisión fue la codificación de audio conocida como MPEG 2 audio capa 2.

MPEG definió una codificación más simple que no ha sido empleada extensivamente, MPEG-2 de capa 1 que realiza las mismas operaciones que la versión de capa 2, pero requiere menos memoria que la utilizada en DVB. Por otro lado, MPEG definió también la exitosa codificación MPEG-2 de capa 3, conocida por los archivos mp3 que se fundamenta en la codificación de capa 2, pero añade una transformación DCT adicional que mejora la compresión. Sin embargo, el uso de la transformación frecuencial dificulta la interpolación de las muestras cuando se producen errores en la transmisión, por lo que se optó por una decisión conservadora utilizando el MPEG-2 de capa 2.

Para la codificación MPEG-2 de audio de capa 2 se especifican un conjunto de frecuencias de muestreo ( $f_m$ ) de valores: 16, 22,05, 24, 32, 44,1 y 48 kHz que determinan el máximo ancho de banda de la señal de audio que se va a reproducir ( $f_m/2$ ). La codificación de audio intentará mantener una velocidad constante que en el estándar se especifica con valores que van desde 32 hasta los 384 kbps.

Los codificadores MPEG-2 mencionados operan con bloques de muestras obtenidos del filtrado y muestreo de la señal. Estos bloques son de tamaño fijo y de valor 1.152 muestras para los codificadores de capa 2 y capa 3. Estos bloques de muestras, denominados tramas, constituyen la entidad básica dentro de la estructuración de la codificación de audio. Debemos tener en cuenta que la duración temporal de una trama de audio dependerá de la frecuencia de muestreo.

### Ejemplo

Empleando la máxima frecuencia de muestreo ( $f_m = 48\text{KHz}$ ) la duración de la trama será de 24ms y utilizando la frecuencia de muestreo más lenta ( $f_m = 16\text{KHz}$ ) la duración de la trama será de 72 ms.

La base fundamental de la codificación es explotar el conocimiento de la percepción auditiva humana para determinar qué información es más relevante y cuál menos. Este conocimiento determinará cuándo se debe garantizar un alto grado de fidelidad entre la información original y la descodificada o cuándo se pueden introducir elevados grados de pérdidas en la información sin que se aprecie una distorsión elevada durante la reproducción del sonido.

Los tres fenómenos que se tienen en cuenta con respecto a la percepción del audio para definir la codificación MPEG son:

- La **sensibilidad** es dependiente de la frecuencia. De forma que en el ancho de banda vocal de 300 a 3.400 Hz el oído es más sensible. Esta característica va asociada a la generación aparato fonador humano.
- El **enmascaramiento frecuencial**. Una señal de gran energía en una frecuencia reduce la sensibilidad del sistema auditivo humano en las frecuencias próximas.
- El **enmascaramiento temporal**. Una estimulación elevada reduce la sensibilidad del sistema auditivo durante un corto período de tiempo. En este caso, una gran energía durante un período de tiempo, debido a una gran amplitud del sonido, enmascara temporalmente sonidos de reducida amplitud que la suceden.

En principio, el proceso de codificación consiste en dividir la banda de audio en 32 subbandas de igual tamaño mediante un banco de filtros. La entrada al banco de filtros se realiza dividiendo las 1.152 muestras en grupos de 32 muestras que se entregan a todos los filtros simultáneamente. Cada grupo de 32 muestras genera una única salida por cada filtro. De esta forma se obtienen, para cada bloque de entrada de 1.152 muestras, 36 salidas para cada filtro que se cuantificarán según el modelo psicoacústico.

Como se muestra en la figura 11, además de repartir la muestras de cada trama entre los 32 filtros, las tramas son empleadas para explotar el modelo psicoacústico de percepción humano. En función de los valores temporales que toman las muestras, el conformador psicoacústico determina los umbrales de enmascaramiento temporal y define en qué escala de valores se deben encontrar las muestras para ser percibidas. El conformador también determina los efectos del enmascaramiento frecuencial aplicando una transformada FFT de 1.024 puntos sobre la trama de audio.

Como resultado de estas operaciones se escoge el conjunto de cuantificadores que se debe aplicar en el banco de filtros. La elección del cuantificador dependerá del análisis psicoacústico y de la frecuencia en la que está centrado el filtro. De esta manera, las regiones de frecuencia donde el oído es más sensible se cuantifican con mayor precisión que las de menor sensibilidad auditiva, en otras palabras, el paso de cuantificación es más pequeño para las frecuencias donde el oído es más sensible.

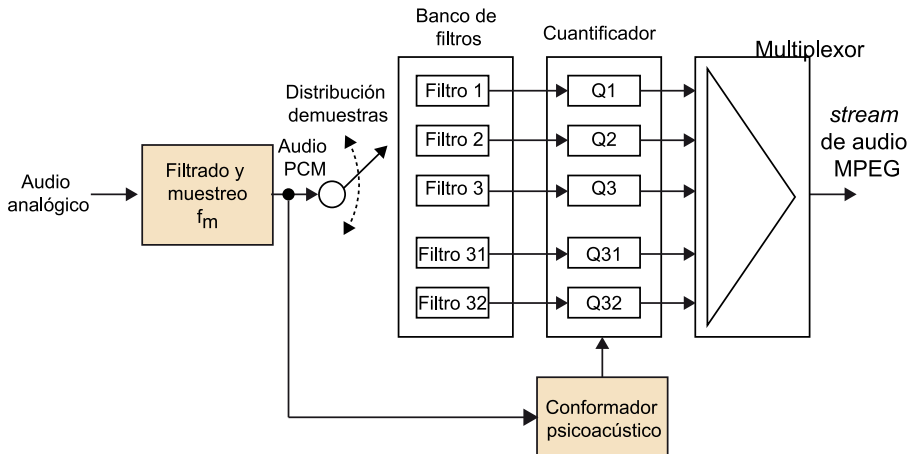


Figura 11. Esquema del codificador de audio MPEG 2

Las 36 muestras de salida de cada cuantificador son recogidas por un multiplexor que las codifica para formar un *stream* elemental de audio.

Cuando se codifican varios canales, como en el caso del sonido estéreo, para cada canal se realiza el mismo conjunto de operaciones. En algunos casos, se explota la redundancia de la información entre canales y se desarrolla una codificación psicoacústica conjunta.

Cuando se codifican varios canales el multiplexor estructura la información en un único *stream* elemental donde las codificaciones de los canales pueden ser independientes o no.

Para DVB, se han propuesto inicialmente cuatro formatos de codificación MPEG-2 posibles:

- Formato **mono**, donde se codifica un único canal.
- Formato **estéreo** con canales codificados independientemente que se decodifican y reproducen simultáneamente.
- Formato **estéreo de codificación conjunta** donde se explota la redundancia entre los canales derecho e izquierdo. Esta codificación requiere una decodificación simultánea para su reproducción en paralelo.

- Modo **dual**, donde hay dos canales que se codifican independientemente y solo se decodifica y reproduce uno de ellos según la selección del usuario.

## 4. Multiplexación y *scrambling* en DVB

Las especificaciones para el transporte de la codificación MPEG para audio y vídeo dan lugar a la recomendación conocida como capa de sistema. En la capa de sistema se indica cómo se deben agrupar los flujos elementales (audio o vídeo comprimidos) y otros flujos de datos de propósito general para formar el multiplex de transporte MPEG. La multiplexación de los flujos audiovisuales debe cumplir una serie de requerimientos temporales para garantizar que el receptor dispondrá de la información necesaria para poder realizar su descodificación y presentación de forma síncrona.

### 4.1. Empaquetamiento y multiplexado MPEG

Para realizar la multiplexación y transmisión de los flujos elementales (*elementary streams*, ES) el comité MPEG especificó el formato de transporte denominado *transport stream* (TS), que facilita al usuario el conocimiento de los programas contenidos en el flujo de información y las facilidades para seleccionar uno de ellos. Por lo tanto, en un TS se podrán distribuir simultáneamente la codificación audiovisual de varios programas (por ejemplo, la 1, Telecinco, Antena3, etc.). Cada programa del *stream* podrá disponer de varias pistas o codificaciones elementales de audio, correspondientes, por ejemplo, a distintas lenguas, y varias pistas o codificaciones de vídeo, correspondientes, por ejemplo, a distintos ángulos de la misma toma.

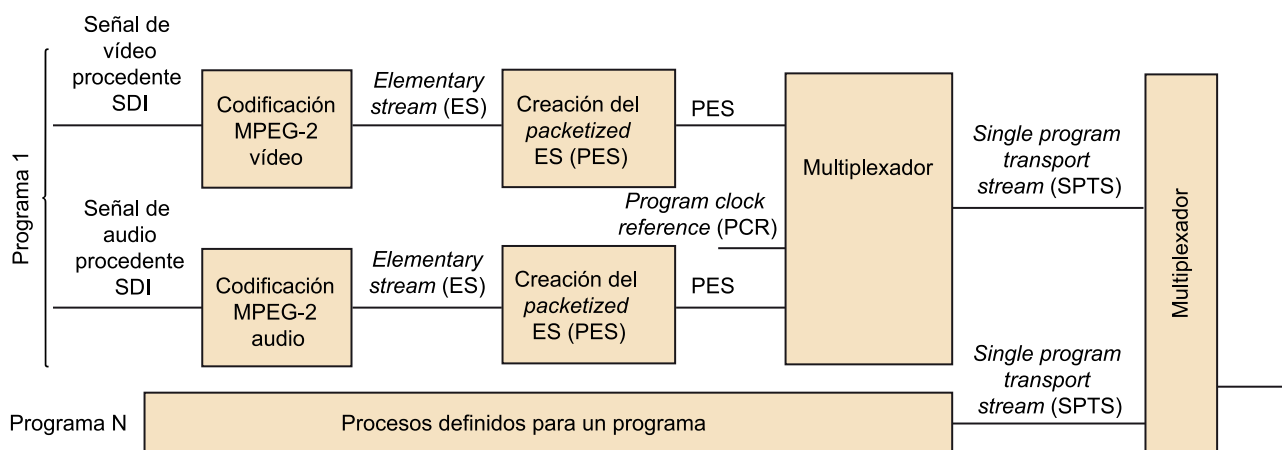


Figura 12. Esquema de la capa de sistema MPEG 2

En primer lugar, los flujos binarios de vídeo y audio (procedentes de la señal SDI) se comprimen independientemente siguiendo las codificaciones de vídeo y audio definidas por MPEG-2 y cada uno de ellos forma un *elementary stream* (ES), que también puede ser una señal de datos (por ejemplo, para aplicaciones). Cada ES contiene *access units* (AU), que suelen ser un *frame* para la imagen (de tipo I, P o B) y una trama de muestras codificadas para el sonido.



A continuación, dado que un ES es una corriente de bits, cada uno de los ES del programa, es decir, vídeo y audio, se estructura en paquetes, y para ello, cada cierto número de bits del ES (este número no está definido y depende del diseñador del sistema) se añade una cabecera. Este número de bits suele venir determinado por el tamaño de los AU. El resultado de unir una cabecera a un conjunto de bits del ES se denomina *packetized elementary stream* (PES).

Dado que los AU son de longitud variable (no ocupa lo mismo un *frame* I que uno P o B), la cabecera incorpora un campo donde se indica el tamaño del paquete PES. De esta manera el receptor sabe cuándo finaliza el PES y se inicia la cabecera del siguiente paquete PES. Aunque el tamaño del PES puede ser el que desee el diseñador del sistema, se ha establecido un máximo de 64KB.

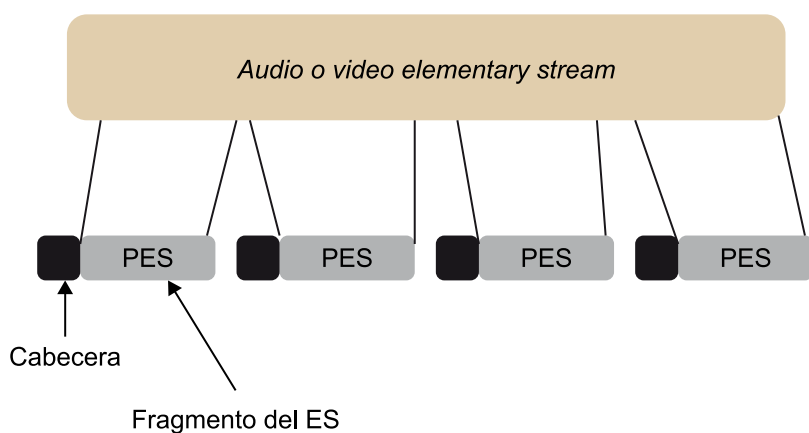


Figura 13. Paquetización de ES en PES

En la cabecera del paquete PES se han definido distintos campos para cubrir las necesidades de la transmisión multiplexada:

- **Identificación.** Para que el receptor seleccione las AU de un ES se ha propuesto un campo de identificación cuyo valor es el mismo en todos los paquetes PES del mismo ES.
- **Referencias temporales.** Son necesarias para garantizar la decodificación y presentación sincronizada de los ES de un programa en el decodificador. Así, los flujos audiovisuales de un programa dispondrán de unas marcas temporales que facilitarán el sincronismo durante la reproducción del programa. Esta marcación garantiza, por ejemplo, la sincronización de voz e imagen durante la reproducción del contenido.
- **Número de secuencia.** Facilita al decodificador la detección de pérdidas de información que puedan haberse producido durante la transmisión.

Finalmente, los paquetes PES de los diferentes ES de un programa se fragmentan originando paquetes más pequeños de tamaño fijo, exactamente de 188 bytes (4 de cabecera + 184 de datos), llamados *transport stream packets* (TSP), que reducen el impacto de los errores de transmisión: si hay errores en la transmisión, el sistema deberá proteger unidades de transmisión pequeñas. La ca-

becera de los TSP incluye un identificador PID (*transport packet identifier*) que indica de qué ES proviene la información. Un TSP solo puede llevar partes de un único paquete PES y puede incorporar un relleno adicional (campo llamado *adaptation field*), si es necesario, hasta completar los 184 bytes de datos que debe contener.

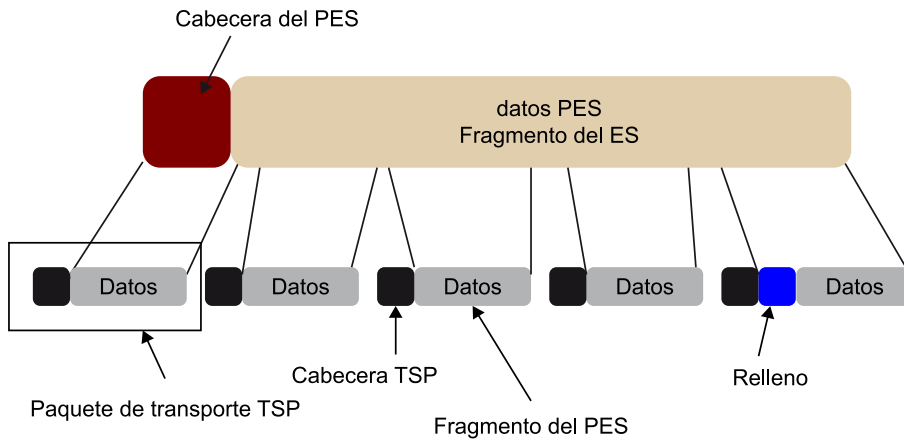


Figura 14. Fragmentación de los paquetes PES en paquetes de transporte TSP

El conjunto de paquetes TSP de un programa se conoce como SPTS (*single program transport stream*). Si por un mismo canal se envían (como suele ser habitual) varios programas, es decir, varios SPTS, los diferentes SPTS se multiplexan originando un único flujo binario que se denomina MPTS (*multi program transport system*). En este caso hay que tener en cuenta que cada SPTS (programa) tiene su propia sincronización y para poder coordinar las temporizaciones de los ES de un mismo programa, se añade información de sincronismo de forma independiente para cada programa multiplexado (cada SPTS). Esta información temporal, denominada *program clock reference* (PCR), se incorpora, cuando aparece, en el campo *adaptation field* de los paquetes de transporte (TSP).

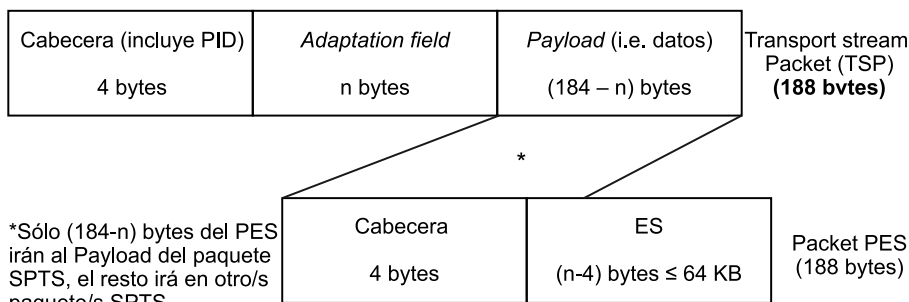
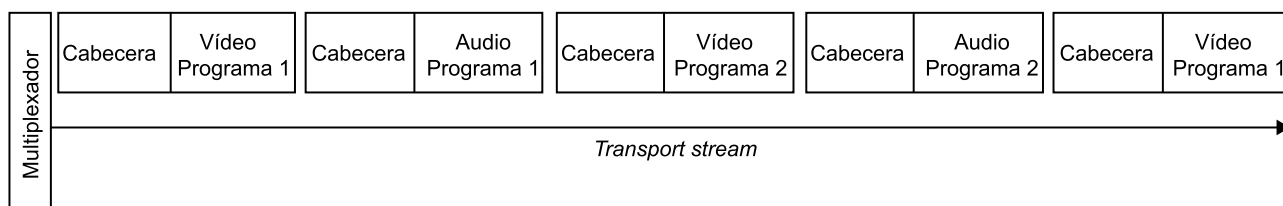


Figura 15. Formato de los paquetes de transporte TSP

El resultado final de multiplexar diferentes SPTS es que se entremezclan los paquetes de los diferentes programas (SPTS). A continuación mostramos un posible resultado de una multiplexación:

Figura 16. Ejemplo de *stream* de transporte MPTS

Además de los *transport stream packets* (TSP) con información de vídeo/audio/datos de un programa, el *stream* de transporte (SPTS o MPTS) lleva otros paquetes de transporte (TSP) que contienen metainformación de cada programa que la capa de sistema MPEG y el estándar DVB estructuran en tablas. La codificación de estas tablas se envía mediante paquetes de transporte (TSP) cuyos identificadores están definidos *a priori* o dentro de las mismas tablas. A continuación veremos qué tipos de tablas (o metainformaciones) han sido definidos. Estas tablas se conocen con el nombre de *program-specific information* (PSI).

#### 4.2. Información específica de un PROGRAMA (PSI)

El decodificador MPEG-2, además de decodificar cada uno de los ES de audio y vídeo (y, a veces, datos) que conforman un programa, ha de ser capaz de encontrarlos dentro de un *stream* de transporte de tipo MPTS. La información específica de un programa (*program specific information*, PSI) es lo que permite que el decodificador realice esa tarea. Las principales tablas en las que se organiza el transporte de información en DVB son las siguientes:

- **Tabla de asignación de programas** (*program allocation table*, PAT). Esta tabla es el primer elemento que necesita un decodificador para poder determinar el conjunto de programas que se están transmitiendo en el multiplex; es decir, en el MPTS. Por lo tanto, es una tabla básica y obligatoria cuyo identificador asociado en los paquetes de transporte es PID = 0. Su propósito es indicar para cada programa transportado cuál es el identificador de los paquetes de transporte que contienen la tabla PMT (ver siguiente elemento) que lo describe. Conceptualmente, esta tabla está compuesta por un conjunto de punteros que indican dónde encontrar la descripción de cada programa transmitido, como se muestra en la figura 17.
- **Tabla del mapa de programa** (*program map table*, PMT). Esta tabla detalla el conjunto de ES que componen el programa y su descripción. Se compone de un conjunto de identificadores de paquetes de transmisión asociados a cada ES y el tipo de flujo audiovisual, mediante los cuales, junto a las descripciones de tipo, el decodificador podrá ofrecer al usuario distintas posibilidades de presentación del programa (por ejemplo, se puede seleccionar el ángulo de la toma de vídeo y el idioma del audio). Una vez realizada la selección, el receptor realizará la demultiplexación de los paquetes de transporte y su reensamblado en paquetes PES antes de ser entregados a los decodificadores de audio y vídeo. Para poder sincronizar la presenta-

ción de los ES audiovisuales, en la PMT se indica también cuál es el PID de los paquetes de transporte que llevarán periódicamente las referencias temporales del PCR.

- **Tabla de acceso condicional** (*conditional access table*, CAT). Esta tabla controla la posibilidad de descodificación de algunos programas del *stream* de transporte (TS). En otras palabras, permite el cifrado de la señal. Por tanto, se deberá disponer de una clave de acceso para poder acceder a los contenidos de los programas referenciados en esta tabla. Esta tabla dispone de un identificador para sus paquetes de transporte de valor PID = 1.
- **Tabla de información de la red de comunicaciones** (*network information table*, NIT). Esta tabla transmite la información relativa a la ubicación física en la red de comunicaciones del multiplex de transporte. Facilita información específica de la red como frecuencias y número de canales utilizados. La información se envía en paquetes de transporte con identificador PID = 16.
- **Tabla de fecha y hora** (*time and date table*, TDT). Esta tabla se utiliza para sincronizar el reloj del sistema del decodificador. Se transporta en paquetes de transporte con identificador PID = 20.

Estas cinco tablas se envían en forma de paquetes de transporte (TSP) multiplexándose con los paquetes de transporte que llevan la codificación de audio y vídeo de los programas. En la figura 17 se muestra un ejemplo de multiplexación de varios programas en un *stream* de transporte. En este caso la tabla de asignación de programas PAT especifica que existen más de 28 programas multiplexados simultáneamente y para cada uno de ellos se define una referencia o PID que indica dónde se detallarán los contenidos (o *elementary streams*) que componen dicho programa. Esta referencia será utilizada por el decodificador para acceder a esta información analizando los paquetes de transporte TSP cuyo identificador tenga por valor el PID asociado al programa. Por ejemplo, en la figura se detalla que para el programa 1 se utilizarán paquetes señalizados con el PID 124 y para el 2 con el PID 130.

Los TSP identificados con el PID 124 y PID 130 contendrán la información detallada de qué contenidos conforman el programa 1 y 2 respectivamente. Como se muestra en la figura, esta información está conformada según una tabla de mapeo de programas PMT que se envía en un TSP para cada uno de ellos.

Los paquetes de transporte que contienen la información de una PMT también indican en qué paquetes TSP se incluirán las marcas temporales del PCR que facilitan la decodificación y presentación sincronizada de los componentes del programa. Como se ha mostrado en la figura 15, los paquetes TSP que llevan los datos del reloj PCR realizan una extensión de la cabecera con un campo de adaptación (*adaptation field*) que transporta esta señalización. En la

PMT se indica el PID de estos paquetes para que el receptor los utilice como fuente de sincronismo para la decodificación y presentación de los *streams* elementales (ES) del programa. Habitualmente, el PID que transporta el PCR de un programa coincide con el *stream* de vídeo principal del programa.

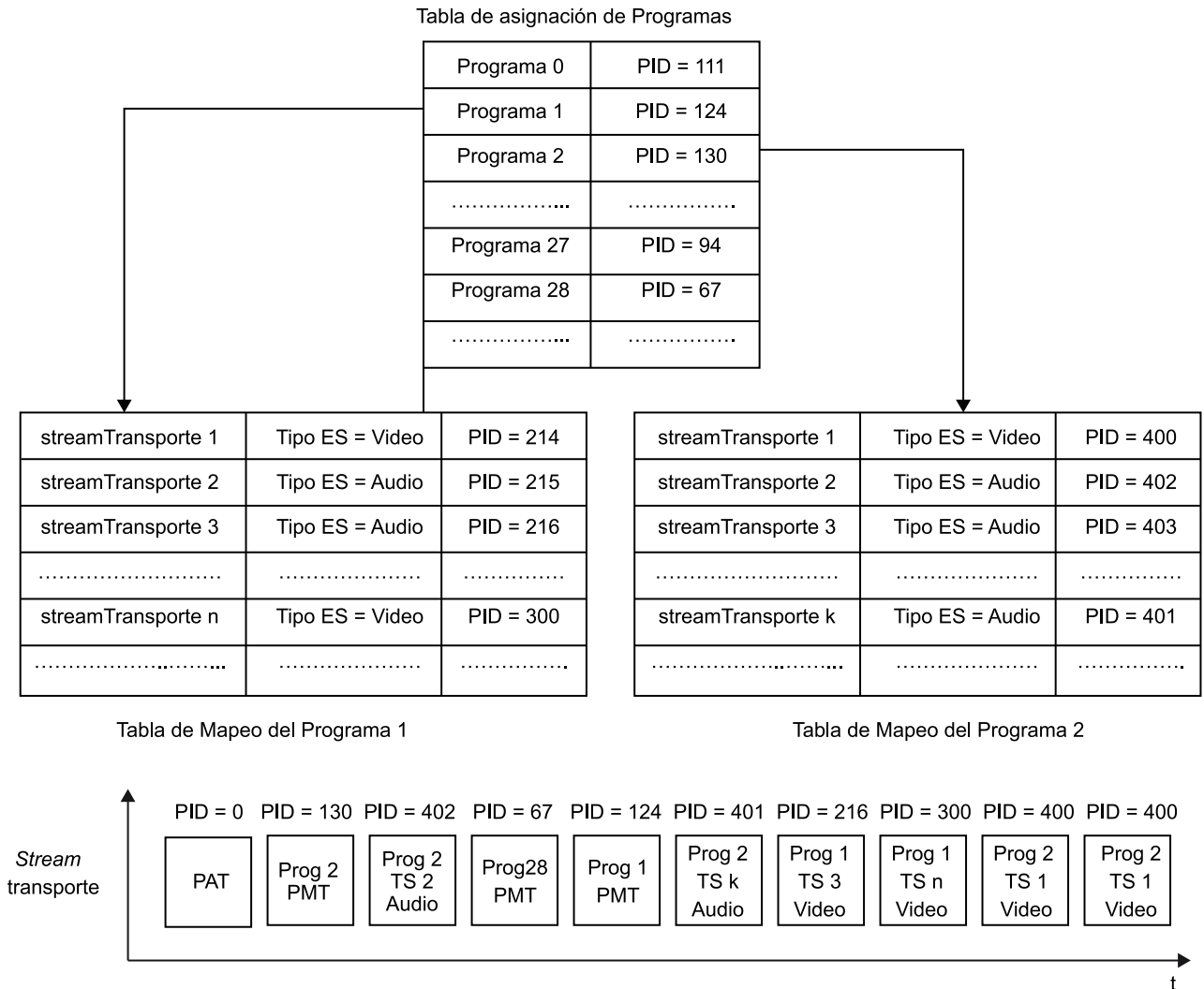


Figura 17. Ejemplo de tablas y multiplexación en un sistema con multiples programas

Para acceder a un conjunto de *streams* elementales que forman parte de un mismo programa, el decodificador selecciona los paquetes TSP del *stream* de transporte que contienen los PID seleccionados de un programa. La selección dependerá de la configuración especificada por el usuario.

**Ejemplo**

Un usuario escogerá del programa 1 un determinado ángulo de vista para un plano en el vídeo (por ejemplo, PID = 300) y un idioma para el audio (por ejemplo, el TSP con PID = 215 podría ser audio en español y el TSP PID = 216 podría ser audio en inglés).

Así pues, a grandes rasgos, los pasos del decodificador a partir de una selección hecha por el usuario serían lo siguientes:

- Buscar PID = 0 que corresponde con la tabla PAT.
- Analizar PAT para seleccionar el programa

- Buscar PID de la PMT.
- Analizar PMT.
- Filtrar los paquetes TSP del programa asociados a los streams elementales (ES) de vídeo y audio seleccionados.
- Reensamblar los paquetes de transporte TSP en paquetes PES para entregarlos al decodificador de vídeo o audio.
- Decodificar cada uno de los flujos elementales (ES).
- Reproducir sincronizadamente la información audiovisual utilizando el PCR.

Además de las tablas de PSI, MPEG-2 define una estructura extensible que permite la transmisión de datos privados. Esta transmisión privada se emplea para distribuir información adicional para la navegación sobre los programas, subtítulos, datos adicionales para el acceso condicional relacionado con la distribución de claves de cifrado, actualizaciones de software para los receptores, flujos de datos encapsulados en IP, etc.

### 4.3. *Scrambling* y acceso condicional

El DVB admite que cualquier programa pueda disponer de acceso condicional, por ejemplo, se requiere del pago por su visión (*pay per view*, PPV).

El conjunto de programas que se transmiten cifrados dentro del *stream* de transporte aparecen en la tabla de acceso condicional CAT del sistema de información. La tabla CAT contiene una lista de identificadores PID, iguales a los de la tabla de asignación PAT, que indican los programas cuya distribución está cifrada. Las técnicas específicas de cifrado y otros detalles pueden ser especificados en la CAT.

El acceso condicional a los programas y contenidos genéricos del *stream* de transporte se basa en la aleatorización de los paquetes de información a través de un cifrado en bloque. Esta funcionalidad se especifica para las aplicaciones comunes de DVB de forma que se minimice la probabilidad de pirateo. Los detalles técnicos del algoritmo de cifrado aplicado están solo accesibles a desarrolladores previa firma de cláusulas de confidencialidad.

El algoritmo de cifrado se basa en dos cifrados de bloque y utiliza una clave de 128 bits:

- Advanced Encryption Standard (AES128), de seguridad probada y de uso extensivo.
- eXtended emulation Resistant Cipher (XRC), desarrollado explícitamente por la organización DVB.

El proceso de cifrado se puede aplicar sobre los paquetes de transporte TSP o sobre los paquetes PES, pero no en ambos niveles de paquetización simultáneamente. Para ello, las cabeceras de los paquetes PES y de los paquetes de transporte TSP indican si los paquetes van cifrados y, en ese caso, con qué clave. En la práctica se manejan dos claves simultáneamente para un programa con el fin de dificultar su pirateo.

El acceso condicional (CA) en sí mismo no se define por la norma ya que la mayoría de los operadores han preferido utilizar su propia tecnología. El estándar DVB permite definir el control de acceso utilizado en un programa a través de la tabla de acceso (CAT). A fin de evitar que el abonado requiera de varios decodificadores, se facilitan tarjetas para el acceso condicional que se insertan en las diferentes ranuras del decodificador. Los estándares seguidos por las operadoras en DVB son el Simulcrypt y el Multicrypt.

## 5. Codificación de canal

Una vez ya se tiene el *transport stream* (ya sea SPTS o MPTS), debe enviarse por un canal satélite, terrestre o por cable. Para ello define unas características comunes para los tres canales, algunas de ellas son las siguientes:

- La codificación de vídeo es MPEG-2 (MP@ML).
- Multiplexado de *streams*, es decir, *transport stream*.
- Códigos protectores de errores basado en Reed-Solomon
- Ampliación de las tablas PSI. Esto da lugar a DVB-SI (*service information*).

Para cada canal variará:

- La mayor o menor protección contra los errores.
- Modulación.

Ítem	Satélite (DVB-S)	Cable (DVB-C)	Terrestre (DVB-T)
Ancho de banda	30MHz	8MHz	8MHz
Potencia/ruido	Poca potencia, canal ruidoso	Mucha potencia, canal poco ruidoso	Canal ruidoso con trayectoria multicamino
Modulación	QPSK → muy robusta, poco eficiente	QAM → poco robusta, muy eficiente	COFDM → muy robusta, muy eficiente

Comparativa entre medios de transmisión según estándar DVB

La codificación de canal aplicada sobre el *transport stream* (SPTS o MPTS), dependerá del medio de transmisión que se vaya a usar. La codificación de canal consiste en añadir un conjunto de dígitos binarios al *transport stream* que faciliten la detección y corrección de errores en el receptor. El conjunto de dígitos binarios añadidos constituye un valor calculado redundante que consume capacidad del canal de comunicaciones. Por ello, la codificación de canal determina de forma eficiente la cantidad de redundancia que se debe introducir en función de los errores de transmisión que se generen en el canal. La codificación de canal también indica cómo se debe introducir esta redundancia para conseguir su mayor capacidad correctora.

Hay tres bloques de corrección comunes en DVB para la transmisión sobre los canales satélite, terrestre y cable (aleatorización, Reed-Solomon y entrelazado de Fourny). En el estándar DVB además se incluyen otros dos bloques específicos (convolucional y perforado) para los canales con más errores que son el satélite y el terrestre.



En la figura 18 se representa el conjunto de bloques que desarrollan las tareas de codificación de canal que se van enlazando desde la entrada del *transport stream* hasta la entrega del flujo codificado a las etapas de modulación. Como podemos observar, cada medio tiene sus propios bloques de codificación y modulación.

#### Nota

En este módulo explicamos los tres bloques comunes de codificación de canal y los bloques específicos de codificación de canal compartidos por los canales satélite y terrestre.

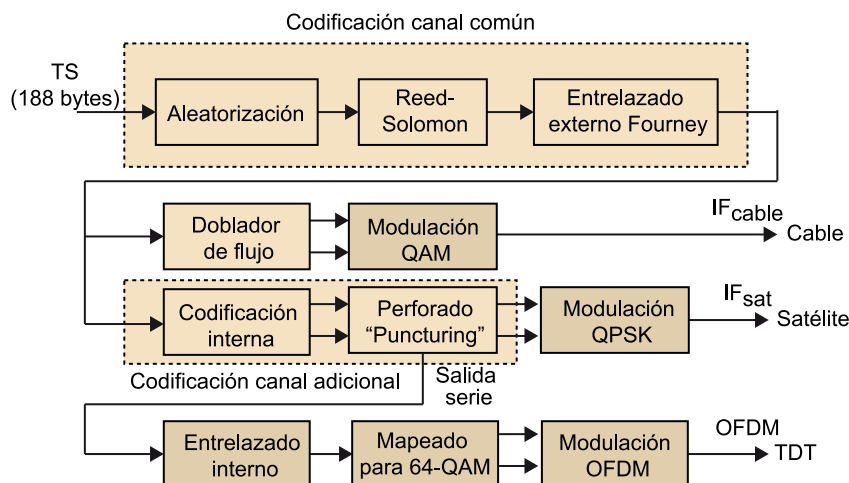


Figura 18. Diagrama de bloques de la codificación y formación de los paquetes de transporte en DVB-S, DVB-T y DVB-C

## 5.1. Aleatorización

El flujo de bits generado por el *stream* de transporte (SPTS o MPTS) puede presentar un comportamiento poco aleatorio, lo que se traduce en que las probabilidades de aparición de bits con valor 0 y 1 pueden ser significativamente distintas o que secuencias de bits con determinados valores de 0 y 1 no aparezcan con la frecuencia esperada. Estas características de aleatoriedad son una de las condiciones de diseño especificadas para las etapas de modulación del nivel físico.

En el nivel físico se parte de la hipótesis de que todos los puntos de una constelación se van a visitar con la misma probabilidad y de forma incorrelada, lo que comporta que la señal transmitida no tendrá componente de continua y ocupará el máximo ancho de banda posible. De esta forma, la energía de la señal estará repartida entre todas las frecuencias menos la de continua.

Para garantizar que esta hipótesis de diseño es siempre cierta, todos los sistemas de difusión DVB incorporan un aleatorizador. Las condiciones de aleatoriedad darán como resultado una permanente distribución de la energía en toda la banda de frecuencias ocupada por la transmisión del multiplex de transporte.

La implementación de un aleatorizador se basa en la generación de una secuencia pseudoaleatoria mediante un registro de desplazamiento de bits linealmente realimentado, *linear feedback shift register* (LFSR).

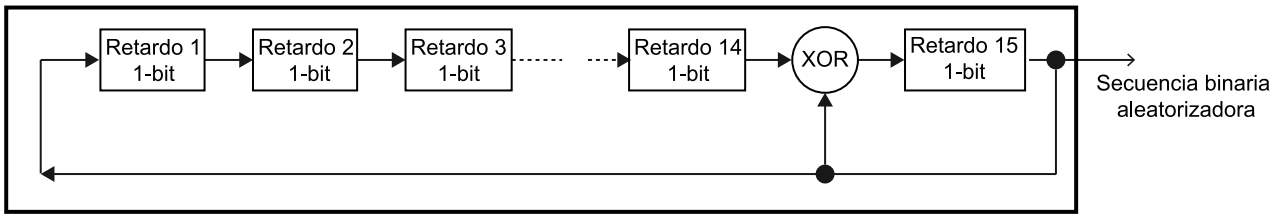


Figura 19. Registro LFSR de 15 elementos

El LFSR se diseña para que el periodo de la señal pseudoaleatoria sea máximo, eligiendo una realimentación que lo garantice. El periodo máximo de la secuencia binaria queda determinado por todas las combinaciones distintas de los estados de los registros de desplazamiento que no sean nulas. Así, un LFSR de 15 registros binarios con la adecuada realimentación lineal da lugar a una secuencia pseudoaleatoria con periodo 32.767 bits.

En el emisor se genera una secuencia binaria a través de un LFSR y se realiza una combinación bit a bit con el *stream* de transporte obtenido a la salida del multiplexor. Esta combinación se desarrolla a través de una simple operación OR exclusiva como se muestra en la siguiente figura:

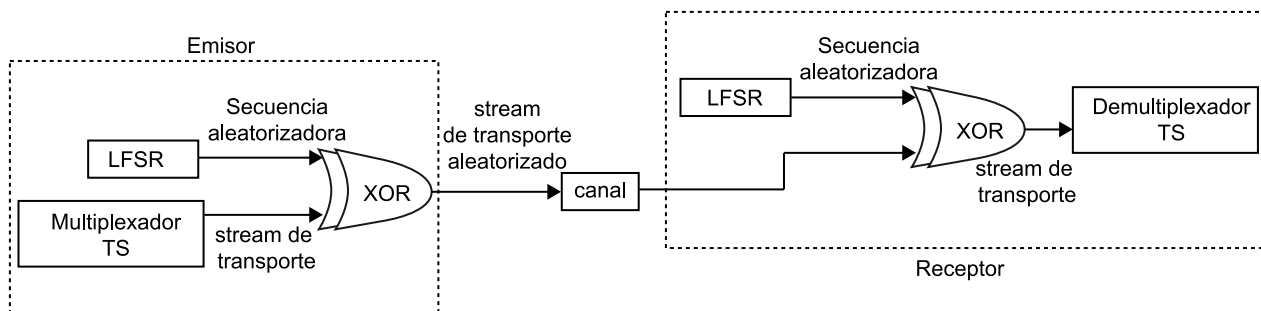


Figura 20. Esquema de operación del aleatorizador y desaleatorizador para la transmisión aleatorizada del *stream* de transporte

Para proceder a la desaleatorización en el receptor se requiere de un LFSR que aplique sincronizadamente la misma secuencia aleatoria sobre el flujo binario recibido. La operación realizada en el receptor vuelve a ser una OR exclusiva bit a bit entre el flujo binario recibido y el LFSR local sincronizado. Debe tenerse en cuenta que la operación XOR es asociativa y que en binario se verifica que:

$$(A \text{ XOR } B) \text{ XOR } B = A \quad (4)$$

El DVB especifica un aleatorizador de 15 registros cuya implementación es fácil y con un reducido coste computacional en aritmética entera de 16 bits. Este dimensionado del LFSR permitiría aleatorizar completamente hasta 21 paquetes de transporte de 188 octetos. Sin embargo, la aleatorización se realiza tan solo en 8 paquetes consecutivos debido a que se pretende facilitar una rápida resincronización del aleatorizador ante desvanecimientos de la señal en el receptor. Para ello, la secuencia aleatorizada incorpora puntos de sincronismo en la cabecera de los paquetes del *stream* de transporte de forma regular.

## 5.2. Codificación de bloque Reed-Solomon

La codificación de fuente realizada sobre los contenidos audiovisuales extrae gran parte de la redundancia de los contenidos generando un flujo binario con una alta eficiencia de codificación. El elevado grado de compresión obtenido permite transferir los contenidos audiovisuales con un mínimo de ancho de banda. No obstante, si los canales de transmisión introducen errores o pérdidas en el flujo transmitido, la descodificación realizada da lugar a notables distorsiones en la reproducción de los contenidos audiovisuales. En la práctica, la codificación de fuente realizada requiere que el receptor disponga de un stream de transporte libre de errores.

Para poder corregir los errores y pérdidas que se introducen durante la transmisión del flujo binario se debe incluir una redundancia calculada mediante técnicas de codificación de canal. Esta redundancia son dígitos binarios adicionales que se añaden al *stream* de transporte aleatorizado en bloques de información o intercalados con los bits de datos. En el primer caso hablamos de **codificación de canal en bloque** y en el segundo, de **codificación de canal convolucional o continua**.

En la cadena de transmisión se aplica primero la codificación en bloque, dado que en el diseño de la codificación de canal se tiene en cuenta la propia estructuración de los datos generados. En este caso, la estructura de los paquetes de transporte aleatorizados definidos en los apartados anteriores. Posteriormente veremos que en DVB de televisión terrestre y por satélite también se aplica la codificación de canal continua.

La codificación de canal en bloque calcula un conjunto de dígitos binarios, o redundancia, que se añade al final de cada paquete de transporte. Esta redundancia constituye un tráiler para el paquete. En DVB la redundancia calculada para cada paquete de transporte es de 16 octetos, por lo que el paquete de transporte con su redundancia alcanza un tamaño de 204 octetos. La técnica de codificación en bloque empleada en DVB se basa en la generación de códigos de canal propuesta por Reed y Solomon.

La codificación de Reed-Solomon opera con agrupaciones de  $M$  bits que definen un símbolo de codificación. Para el caso particular de una codificación binaria  $M$  sería 1. En la codificación Reed-Solomon del DVB se utiliza  $M = 8$ , por lo tanto, el símbolo empleado en DVB para la codificación bloque es de un octeto, lo que simplifica la implementación en aritmética entera de las operaciones a realizar. En general, la codificación Reed-Solomon debe cumplir que la longitud  $N$  de los bloques de símbolos codificados debe seguir la relación  $N < 2^M$ , lo que implica que los bloques de datos con los que se debe realizar la detección y corrección de errores en DVB serán como máximo de 255 octetos, incluyendo la redundancia.

La longitud y valor de la redundancia dependen de la codificación Reed-Solomon disponible. Para un tamaño de bloque de 255 octetos la redundancia necesaria de una codificación podría ser 16 o 32 octetos. La capacidad correctora de estos códigos es siempre la mitad de la redundancia introducida por lo que dependiendo de la máxima cantidad de errores que podamos esperar decidiremos un caso u otro. En DVB, como ya hemos comentado, se añade una redundancia de 16 octetos a cada paquete de transporte. De esta manera, cualquier se puede corregir cualquier combinación de 8 o menos octetos erróneos que se produzca durante la transmisión del paquete y su tráiler.

16 octetos sobre 204 es una sobrecarga de codificación muy reducida para los resultados que se pueden obtener en la corrección de errores de transmisión.

### 5.3. Entrelazado de Fourny

El entrelazado es una técnica que varía el orden de una secuencia de bits. El uso clásico del entrelazador es separar los errores introducidos durante la transmisión de la señal en el canal de comunicaciones. Esto se debe a que la transmisión en los canales de comunicaciones suele introducir errores en ráfaga, o consecutivos, debido a los desvanecimientos o interferencias momentáneas que sufre la señal al propagarse por el medio de transmisión.

Para protegerse de múltiples errores consecutivos en los bits transmitidos se debe emplear una codificación de canal muy poco eficiente puesto que requiere de una gran redundancia. Como solución, la técnica de entrelazado lo que hace es desordenar los bits, de forma que, al ser vueltos a ordenar en el receptor, los errores queden suficientemente repartidos como para considerarlos independientes entre sí.

El entrelazador desordena de una forma predeterminada los símbolos de canal, definidos como  $M$  bits consecutivos en la codificación Reed-Solomon de *stream* de transporte. El desentrelazador realiza la operación contraria para que los bloques lleguen al decodificador de canal con el orden original. Esta operación hace que la modificación del valor de un conjunto consecutivo de bits transmitido durante la transmisión no afecte a un conjunto consecutivo de símbolos de canal. En la práctica lo que se consigue con el entrelazado es distribuir los errores consecutivos que podrían afectar a un único bloque entre dos bloques de codificación y, por lo tanto, se reduce la cantidad de errores que se deben corregir en un único bloque de codificación, lo que permite introducir menos redundancia en la codificación bloque porque el número de símbolos erróneos que podemos esperar en cada bloque es menor.

Para determinar hasta cuánto hay que desordenar los símbolos de codificación de  $M$  bits de las palabras código antes de ser transmitidos, se debe analizar la estadística de la longitud de las ráfagas de errores. Teniendo en cuenta la distribución de las longitudes, determinaremos cuál es la profundidad del entrelazado, magnitud que denotaremos  $D$ . En la práctica, el valor de  $D$  será superior o del mismo orden de magnitud que tenga el valor medio de la longitud de la ráfaga.

El entrelazado de Fourny se basa en ir distribuyendo los símbolos de canal entre distintos registros de desplazamiento. Cada registro de desplazamiento almacena símbolos de canal, que en el caso de DVB son de 8 bits. El número de registros de desplazamiento que se utilizan se corresponde con la profundidad del entrelazado que se vaya a emplear. El desentrelazado realiza la operación inversa, empleando también el mismo número de registros de desplazamiento que se van alimentando con los símbolos de canal recibidos.

El estado inicial de los registros de desplazamiento es distinto para cada registro empleado en el entrelazador y en el desentrelazador. En el entrelazador, el primer registro de desplazamiento está inicialmente vacío; el segundo dispone de un solo elemento de inicialización; el tercero, de dos elementos de inicialización, y de esta forma se sigue hasta el registro  $D$  que dispondrá de  $D-1$  elementos de inicialización. La diferente precarga de los registros de desplazamiento dará lugar a la separación de los símbolos que habían sido generados de forma consecutiva a la salida del codificador de canal.

El desentrelazador dispone de un estado inicial inverso al entrelazador. El primer registro dispone de un conjunto de  $D-1$  símbolos inicialmente; el segundo, de  $D-2$  y se sigue reduciendo el número de símbolos en una unidad hasta llegar al registro  $D$  el cual no tiene ningún elemento.

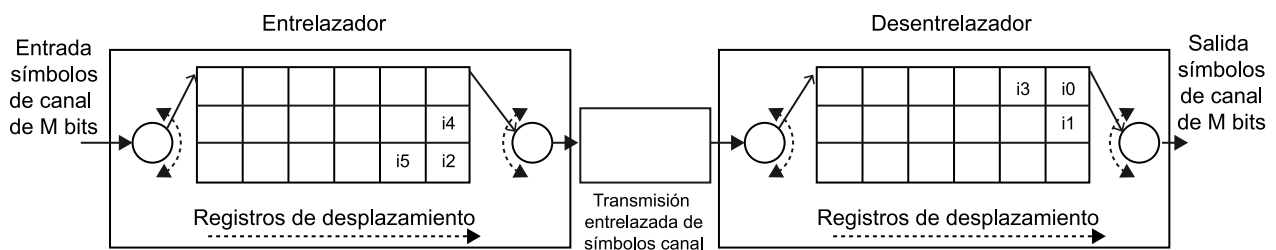


Figura 21. Ejemplo estado inicial de entrelazador y desentrelazador de Fourny para  $D = 3$

A partir del estado de precarga representado en la figura anterior, la operación realizada en cada grupo de registros es escribir un símbolo en un registro distinto cada vez y, simultáneamente, ir extrayendo un símbolo en el mismo orden de escritura. De esta manera, el símbolo que se escribe en el primer registro del entrelazador es inmediatamente retransmitido hacia el canal de comunicaciones.

En el desentrelazador se realiza la misma operación. El primer símbolo que llega se escribe en el primer registro, pero la primera extracción en este caso es de un valor de inicialización. De esta manera se compensan las esperas de los símbolos en el entrelazador y desentrelazador, consiguiendo la reordenación a la salida del desentrelazador.

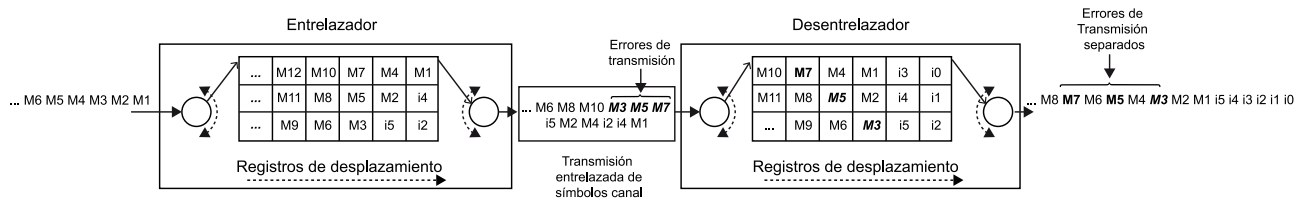


Figura 22. Ejemplo operación de entrelazador y desentrelazador de Fourny para  $D = 3$

La figura anterior muestra el proceso llevado a cabo durante la transmisión de los símbolos  $M_i$ . Cada símbolo se va escribiendo cada vez en un registro diferente y cuando se llega al último registro, se vuelve a alimentar al primero, por lo que se realiza una operación circular de escritura sobre los registros. Conforme se va escribiendo, se van leyendo los primeros elementos que están en los registros de desplazamiento y se van enviando al canal.

En el canal se pueden producir errores en los bits transmitidos. En el ejemplo de la figura consideramos que se produce una ráfaga de 15 bits erróneos que afectan a los bits transmitidos de los símbolos de canal  $M_7$ ,  $M_5$  y  $M_3$ .

En desentrelazador realiza la operación de escritura en sus registros de la forma circular mencionada anteriormente. De esta manera, se separa en registros distintos los símbolos erróneos que llegan consecutivamente al receptor. La lectura del desentrelazador devuelve el orden inicial a los símbolos para que puedan ser corregidos por el decodificador de canal.

La operación del entrelazador y desentrelazador de Fourny da lugar a un retardo desde que llega un símbolo  $M_i$  al entrelazador hasta que sale del desentrelazador. El retardo se corresponde con el número de símbolos de precarga que tienen el entrelazador y desentrelazador, cuyo valor es de  $D \cdot (D-1)$  símbolos. En el ejemplo observamos que la operación del entrelazado tiene una salida encabezada por los seis símbolos de inicialización que determinan el retardo introducido por esta operación y que son descartados por el receptor antes de proceder con la decodificación de canal.

El entrelazado de Fourny especificado para DVB dispone de una profundidad de entrelazador  $D = 12$ , lo que comporta que las ráfagas de errores de bits previstas serán del orden de las decenas de bits.

#### 5.4. Codificación convolucional o continua y perforado

El estándar DVB añade más redundancia específicamente para los canales con mayor tasa de errores: para los medios de transmisión por satélite y terrestres se incorpora una codificación continua con un elevado grado de protección que resulta muy eficaz para hacer frente a los errores aleatorios introducidos por el canal durante la transmisión de los datos. Sin embargo, esta codificación no es muy efectiva frente a grandes ráfagas consecutivas de errores debidas a interferencias co-canal, interferencias destructivas multicamino o ruido impulsivo. Por ello, como ya hemos comentado, la codificación de canal en DVB realiza previamente una codificación bloque Reed-Solomon y un entrelazado de Fourney que facilitan la corrección de las ráfagas de errores.

A diferencia de los códigos de bloque, los códigos continuos operan directamente con el flujo de bits que, en este caso, se obtiene de la salida del entrelazador de Fourney. El codificador continuo trabaja en paralelo con un conjunto reducido de bits, realizando operaciones elementales basadas en retardos y OR exclusivas. En el caso de DVB el codificador genera dos bits de salida por cada bit de entrada.

La codificación continua diseñada depende de los valores que hayan ido llegando al codificador. El codificador incorpora un conjunto de registros de desplazamiento para ir almacenando dichos valores entrantes de forma que la combinación de la entrada en un instante, y sus valores en instantes pasados, determinará los valores de la salida en ese instante. Las combinaciones aplicadas sobre los valores entrantes se diseñan heurísticamente.

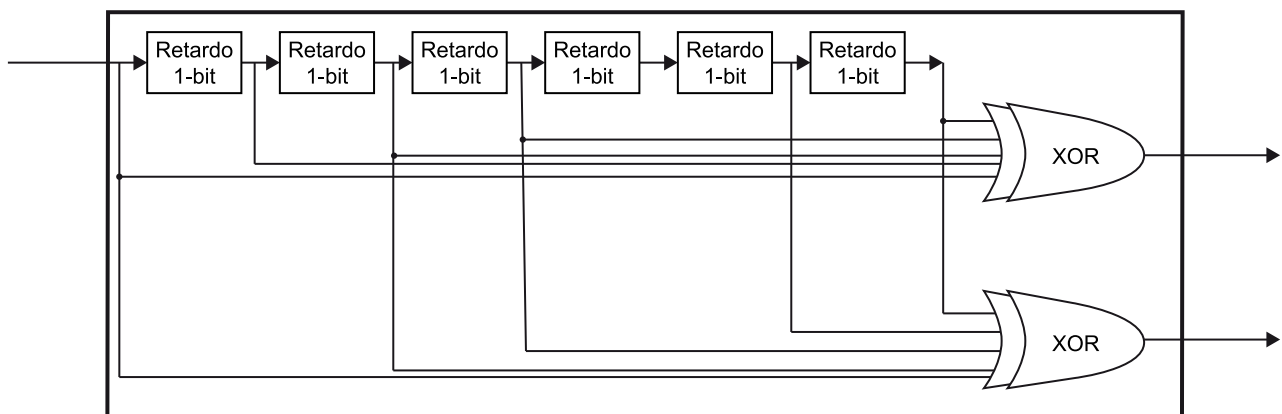


Figura 23. Codificador continuo de DVB-T y DVB-S

La decodificación se realiza utilizando el algoritmo de Viterbi cuyo propósito es reducir el número de operaciones para determinar cuál es la secuencia más verosímil respecto a la recibida en el receptor. Si se realizara de forma exhaustiva, se debería hallar la secuencia válida, o generable por el codificador, que discrepa en menos bits con la recibida. El algoritmo de Viterbi evita realizar comparaciones innecesarias acelerando el proceso de decisión.

Dado que la redundancia introducida por el codificador es muy elevada  $2\frac{1}{2}$ , 2 bits de salida por cada 1 de entrada, DVB propuso de forma opcional realizar una perforación del codificador continuo. La perforación es un proceso de diezmado controlado de los bits a la salida del codificador continuo. La perforación implica, no transmitir todos los bits de la salida del codificador continuo.

### **Ejemplo**

Para 4 bits de entrada el codificador continuo genera 8 bits a su salida. Si de estos 8 bits se transmiten solo 6, el perforador habrá diezmado en 2 bits la salida del codificador continuo.

En este ejemplo, la relación entre bits de salida del perforador respecto a los bits de entrada en el codificador continuo es  $6\frac{1}{4}$ .

Las posibilidades de perforación ofrecidas a las operadoras de DVB son:  $2\frac{1}{1}$  (no perforada),  $6\frac{1}{4}$ ,  $4\frac{1}{3}$ ,  $6\frac{1}{5}$  y  $8\frac{1}{7}$ . Para la codificación perforada elegida deberemos considerar que la tasa de codificación resultante se derivará inversamente de la relación entre bits de salida del perforador y bits de entrada en el codificador continuo.

### **Ejemplo**

Para una relación de bits de salida de 6 por 4 de entrada la tasa del codificador será  $\frac{2}{3}$ , lo cual implica 1 bit de redundancia por cada 3 bits transmitidos (33% de redundancia).

La elección de una tasa de codificación dependerá, en última instancia, de la potencia que reciba un receptor en la zona de cobertura: a menor potencia más redundancia.

La perforación mejorará el aprovechamiento del ancho de banda disponible a costa de reducir la capacidad de corrección de errores.