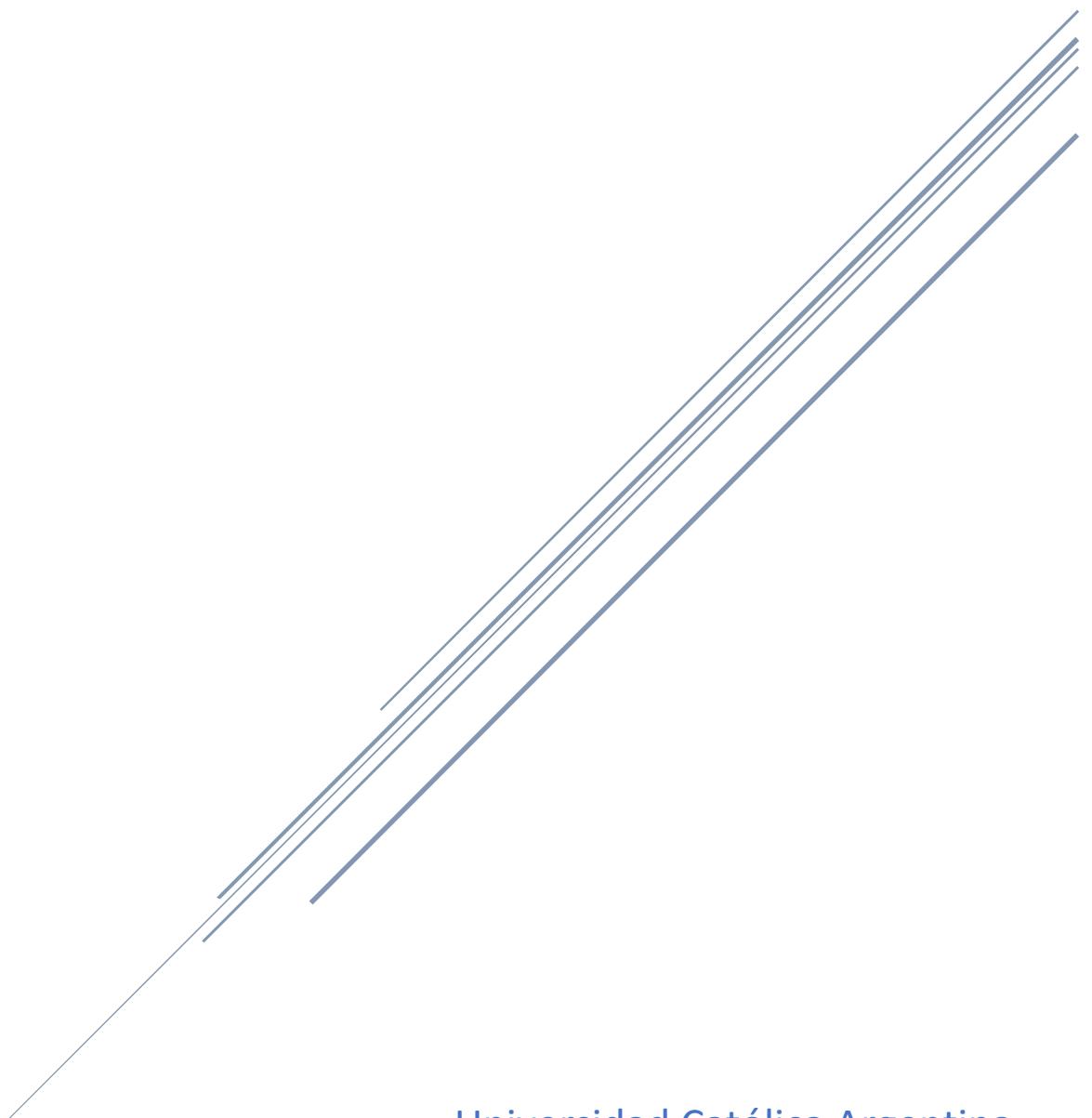


# DIGITAL VIDEO BROADCAST SATELLITE

Profesor: Pablo De Cesare

Alumno: Enrique Cametti



Universidad Católica Argentina  
Radiodifusión Sonora y Televisiva

## INTRODUCCIÓN

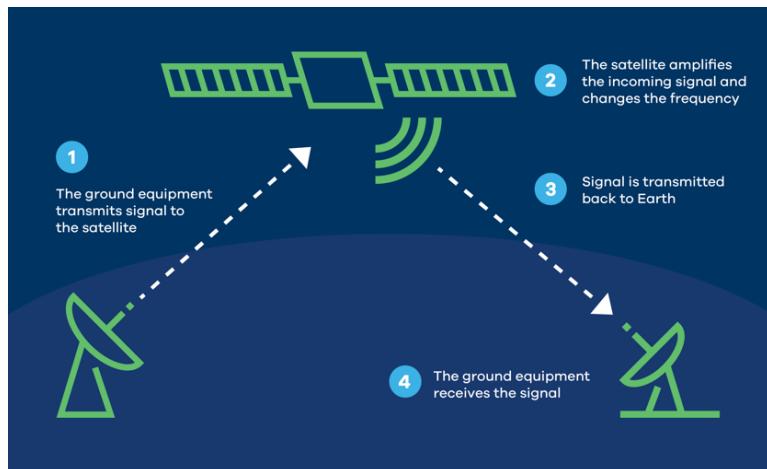
El informe busca explicar el funcionamiento de DVB-S y las mejoras que trajo DVB-S2, basándose en las normas EN 300 421 (1997-08) y ETSI TR 102 376 (2005-02). Después se mostrarán ejemplos prácticos utilizando GNU Radio para mostrar experimentalmente cómo funciona el sistema.

## SISTEMA SATELITAL

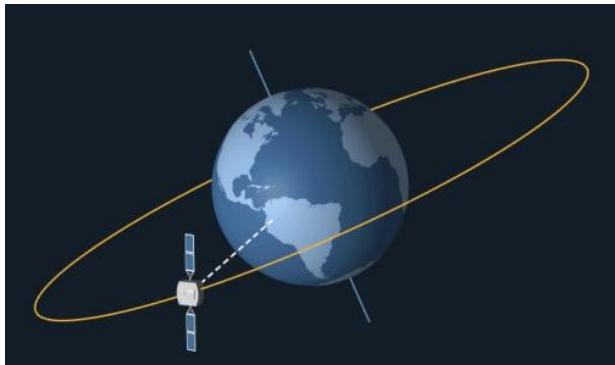
Digital Video Broadcast – Satellite (DVB-S) consiste en la difusión de video digital vía satélite que se utiliza comúnmente en la televisión. Es un estándar que estuvo en desarrollo entre 1993 y 1997. Para empezar a abarcar los temas, es conveniente empezar con las características del sistema de comunicación satelital.

La comunicación ocurre en cuatro pasos:

1. Una estación terrestre transmite la señal a una frecuencia determinada.
2. Un satélite amplifica o regenera la señal recibida y la cambia de frecuencia.
3. El satélite transmite la señal modificada hacia la Tierra.
4. El equipo terminal recibe la señal recibida.



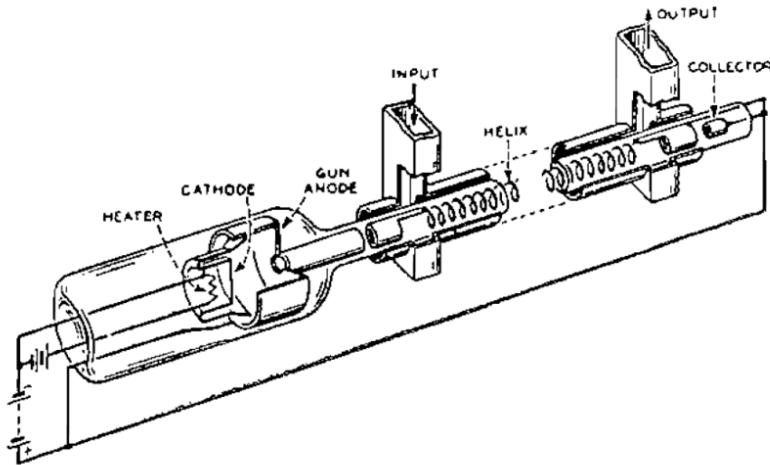
Para que el sistema funcione, es necesario que ambos extremos del radioenlace se encuentren dentro de la zona de visión uno del otro. Debido a esto se requiere que el movimiento del satélite siga el movimiento y rotación de la Tierra.



El mismo va a estar orbitando en la **órbita geoestacionaria (GEO)**<sup>[All]</sup> a 35.786 km respecto de la superficie de la Tierra, ya que ésta cumple con las particularidades de que su período es igual al período de rotación de la Tierra y su inclinación y excentricidad son nulas. Esto permite estar siempre sobre el mismo punto de la superficie terrestre. Esta órbita también es utilizada por los satélites meteorológicos.

Los satélites que se utilizan ofrecen transmisiones de mucha potencia para poder utilizar receptores más pequeños y con menores prestaciones. Esto se hace para poder reducir costos. Se debe entender que debido a la gran distancia que hay entre transmisor y receptor la señal suele sufrir una **atenuación de espacio libre** de unos 205 dB, haciendo que la señal recibida sea susceptible al ruido.

Para amplificar la señal, se utiliza un **Travelling Wave Tube Amplifier** (TWTA) activo que posee no linealidades en su respuesta característica que no se corrigen para no reducir la eficiencia energética del satélite. La energía que utiliza el satélite proviene de las celdas solares (en el día) o de las baterías de respaldo (durante la noche). Debido a la no linealidad del amplificador, se trabaja con técnicas de modulación que no dependan de la amplitud de la señal, MPSK.



## FUNCIONAMIENTO DEL SISTEMA DVB-S

Para poder transmitir los programas de televisión, suele utilizarse **MPEG-2**. Éste es un estándar que se encarga de realizar una compresión con pérdidas para codificar un flujo de audio y video con el tamaño y un ancho de banda menor al de los flujos originales. El nuevo flujo o stream puede ser un **Program Stream (PS)**, destinado al almacenamiento, o un **Transport Stream (TS)**, destinado a la transmisión o difusión.

Debido a la naturaleza del medio de transmisión, la señal suele deteriorarse antes de llegar al receptor. Debido a esto, es necesario generar una protección frente a errores, que va a reducir la velocidad de transmisión máxima que va a poder tener el TS. Los métodos que suelen utilizarse para este propósito son la codificación Reed-Solomon y codificación convolucional, posteriormente se entrará más en detalle sobre los mismos.

## VELOCIDAD DE TRANSMISIÓN DEL SISTEMA

Un canal de estos satélites suele tener un ancho de banda de 26 a 36 MHz. Debe elegirse una velocidad de símbolo con un espectro más estrecho. Se suele utilizar QPSK con una velocidad de modulación de 27,5 MSps, obteniéndose una **velocidad de transmisión bruta (GDR)** de 55 Mbps. El tiempo de símbolo es el doble al tiempo de bit debido a que estoy utilizando QPSK como esquema de modulación. Si aumento la cantidad de bits por símbolo, aumentaría la tasa de transmisión manteniendo el ancho de banda.

Una vez generado el TS, se le agrega Forward Error Correction (FEC) de **Reed-Solomon**. Normalmente suelen agregarse 16 bytes a cada paquete TS (188 bytes), se lo llama codificación RS(204,188). Esta codificación puede corregir hasta 8 errores. Si se excede este número, los errores serán detectados pero no podrán ser corregidos por RS. Como se mencionó anteriormente, esta protección reduce la velocidad de transmisión y depende de la cantidad de bits que se agregan como protección. En este caso está dado por la siguiente ecuación:

$$\text{velocidad neta}_{RS}(NDR_{RS}) = 55 \text{ Mbps} \cdot \frac{188}{204} = 50,69 \text{ Mbps}$$

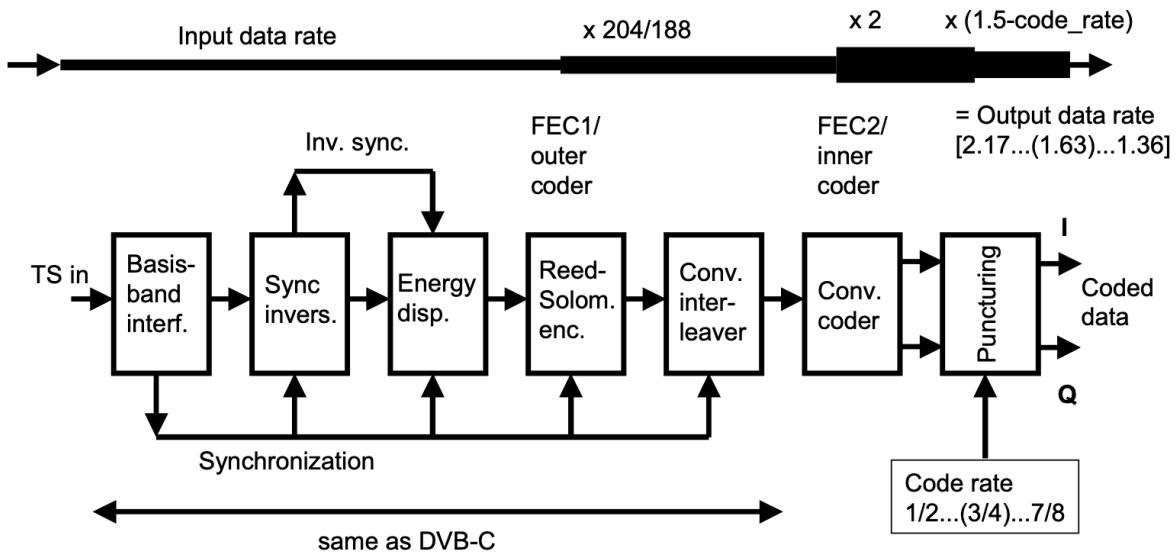
Como segunda medida de protección, se agrega **codificación convolucional** que agrega bits en función de su tasa de código (CR), que es la razón entre la tasa de datos a la entrada y la tasa de datos a la salida. Una menor tasa de código implica una mayor protección frente a errores pero sacrifica la velocidad de transmisión de los datos. Un buen punto de compromiso entre protección y velocidad suele ser  $\frac{3}{4}$ . Al agregar esta nueva capa, se vuelve a modificar la tasa de transmisión de datos.

$$\text{velocidad neta}_{RS+CC} = 55 \text{ Mbps} \cdot \frac{188}{204} \cdot \frac{3}{4} = 38,01 \text{ Mbps}$$

## MODULADOR

El modulador utilizado comúnmente en DVB-S consta de dos partes. En la primera se genera la señal en banda base con su protección frente a errores, mientras en la segunda se transporta la señal a las altas frecuencias para transmitirla por el medio.

### PRIMERA PARTE



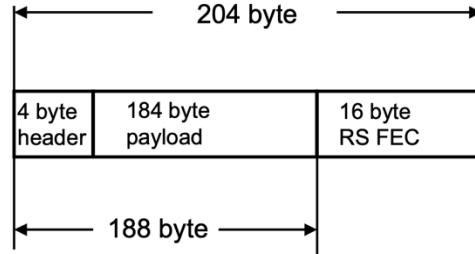
La primera parte del modulador consiste en siete bloques, que se muestran en la figura anterior. El sistema recibe un transport stream en paquetes de 188 bytes, de los cuales cuatro forman parte del encabezado. Cada encabezado comienza con el byte de sincronización, que tiene el valor de 0x47 (71 en decimal).

El primer bloque es la **interfaz de banda base**. Ésta se encarga de que la señal se sincronice con la estructura del sync byte cada cinco paquetes.

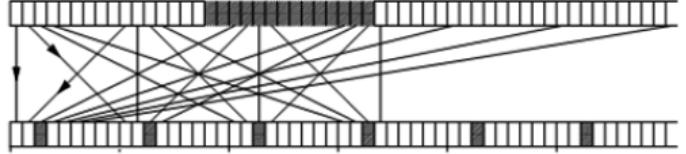
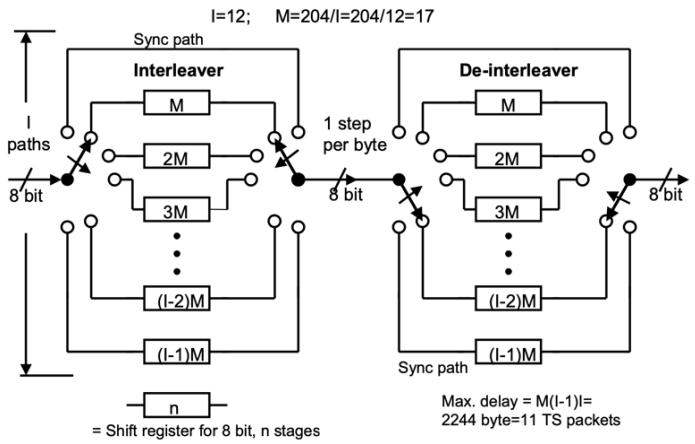
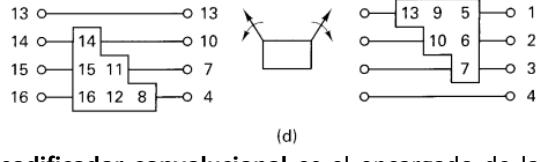
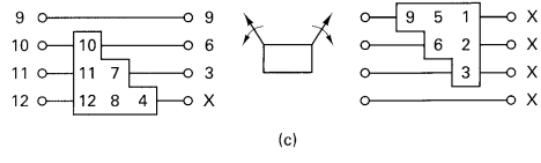
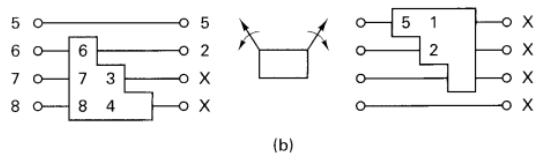
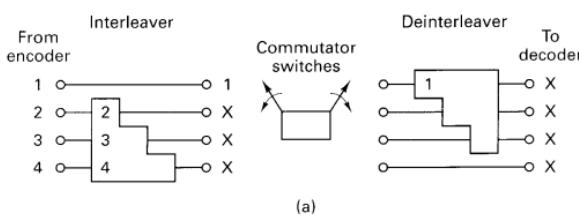
El bloque de **inversión de sync** se encarga de la inversión del sync byte (0x47 a 0xB8) cada ocho paquetes del transport stream. Esta inversión se utiliza para saber cuando hay que resetear el proceso del bloque siguiente.

El bloque de **dispersión de energía** se utiliza para poder mantener la sincronización más allá del transport stream. Se encarga de modificar las cadenas largas de unos y ceros para que el demodulador tenga menos problema en mantener la sincronización. Para cumplir con su objetivo, este bloque normalmente utiliza secuencias binarias pseudo aleatorias (PRBS)<sup>[AIII]</sup>, que permiten que los sync bytes, tanto normales como invertidos, pasen a través de la secuencia de manera transparente.

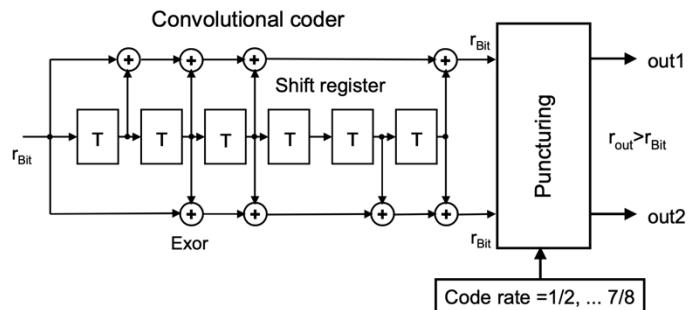
El **codificador Reed-Solomon** se encarga de aplicar Reed Solomon forward error correction (RS FEC)<sup>[AIII]</sup>. A cada paquete se le agregan 16 bytes, por lo que los paquetes que antes eran de bytes ahora son de 204 bytes. Esto permite que el receptor pueda corregir hasta ocho errores por paquete. Si se encuentra un número mayor de errores, el paquete es descartado.



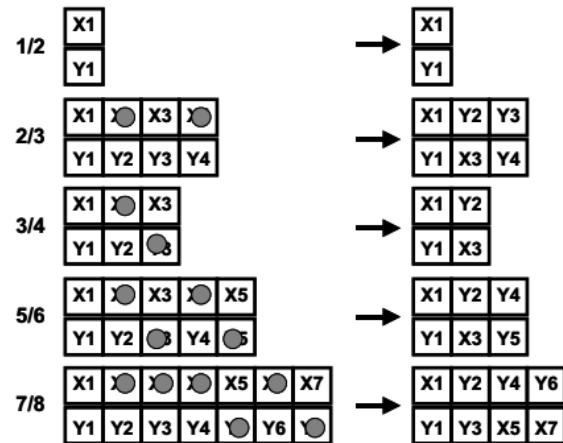
El **intercalador convolucional** se utiliza para proteger a la señal de errores de ráfaga. Se mezclan los paquetes alternándolos. De esta manera, los errores en ráfagas que recibe el demodulador se traducen a errores simples que se encuentran en los paquetes. Esto permite al decodificador Reed-Solomon corregir el paquete recibido y no tener que descartarlo. El intercalador tiene 12 ramas ( $N$ ) con una profundidad de 17 registros de desplazamiento ( $J$ ) para que los sync bytes no se desordenen. El retardo extremo a extremo es  $N \cdot J \cdot (N-1) = 2244$  bytes.



El **codificador convolucional** es el encargado de la protección a error interna. Se utiliza un shift register de seis niveles y se separa el flujo de datos en tres. El shift register influencia los dos flujos de la señal mediante compuertas XOR. La velocidad de transmisión se duplicó, como también la cantidad de bits que envío. Entonces, el code rate de este codificador es de  $\frac{1}{2}$ . En el receptor se utiliza el decodificador Viterbi, que reconstruye el camino en base a la mayor probabilidad.

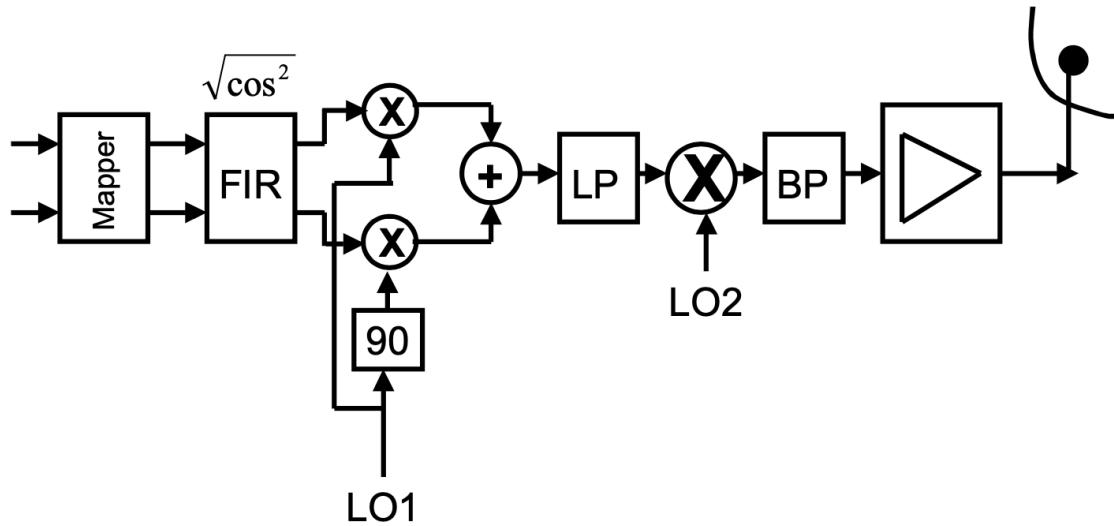


El **puncturing block** permite aumentar la tasa de código, sacrificando la protección. Lo que hace básicamente es omitir bits de forma selectiva. En el receptor, los punctured bits son llenados con “don’t care bits” que se considerarán como errores y serán corregidos por el decodificador Viterbi.



| Code Rates r | Puncturing pattern                   | Transmitted sequence (after parallel-to-serial conversion)  |
|--------------|--------------------------------------|---|
| 1/2          | X: 1<br>Y: 1                         | X <sub>1</sub> Y <sub>1</sub>   |
| 2/3          | X: 1 0<br>Y: 1 1                     | X <sub>1</sub> Y <sub>1</sub> Y <sub>2</sub>  |
| 3/4          | X: 1 0 1<br>Y: 1 1 0                 | X <sub>1</sub> Y <sub>1</sub> Y <sub>2</sub> X <sub>3</sub>   |
| 5/6          | X: 1 0 1 0 1<br>Y: 1 1 0 1 0         | X <sub>1</sub> Y <sub>1</sub> Y <sub>2</sub> X <sub>3</sub> Y <sub>4</sub> X <sub>5</sub>                               |
| 7/8          | X: 1 0 0 0 1 0 1<br>Y: 1 1 1 1 0 1 0 | X <sub>1</sub> Y <sub>1</sub> Y <sub>2</sub> Y <sub>3</sub> Y <sub>4</sub> X <sub>5</sub> Y <sub>6</sub> X <sub>7</sub> |

## SEGUNDA PARTE



La segunda parte del modulador se encarga de preparar la señal para ser transmitida a través del medio.

El bloque **mapeador** se encarga de traducir la cadena de bits al símbolo correspondiente en QPSK. Normalmente para dicho mapeo se considera el código gray ya que para un error de símbolo, normalmente obtengo un menor error de bit. El esquema de modulación utilizado es QPSK.

Posteriormente la señal pasa por un **filtro de respuesta impulsiva finita**. Se utiliza el filtro root raised cosine ya que este es un filtro adaptado y minimiza la probabilidad de error. El mismo tiene un factor de rolloff de 0,35.

El paso siguiente es formar la señal en QPSK que se hace con un **mixer**. En el cual las señales IQ se modulan y la señal Q se desfaza 90º.

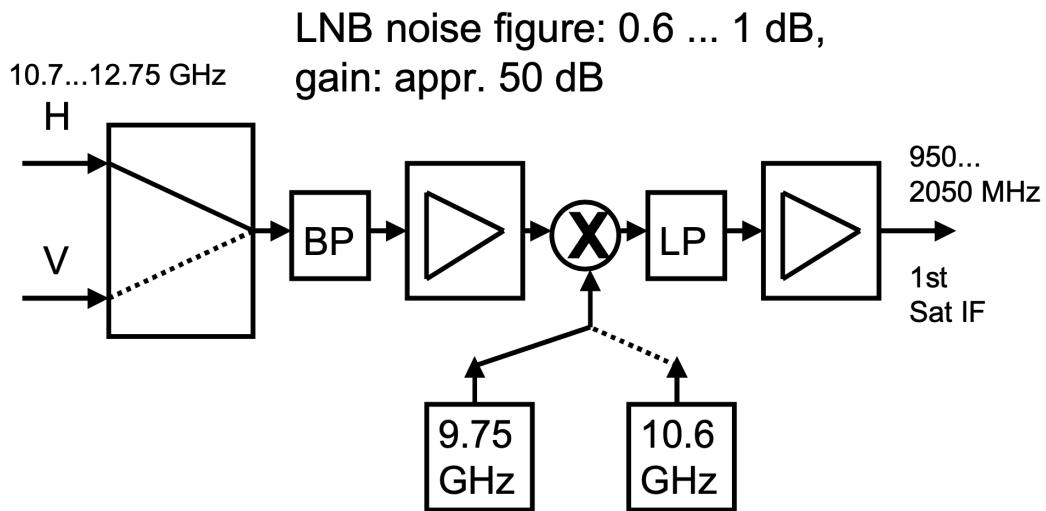
Una vez generada, se la traslada a RF y se la amplifica para luego ser transmitida por la antena. Los filtros simplemente se encargan de no generar interferencia con otras señales.

El satélite recibe la señal atenuada debido al medio, se la cambia de frecuencia y se la amplifica. Luego se filtran las componentes que no se encuentran en la banda a transmitir y se envía la señal a la Tierra.

## RECEPTOR

Como en el caso transmisor, el receptor cuenta con dos partes. En la primera, se recibe la señal en RF, se la traslada a una frecuencia intermedia (IF) y la amplifican. La segunda parte se encarga de decodificar la señal, obviamente, en el orden inverso en el que fue codificada.

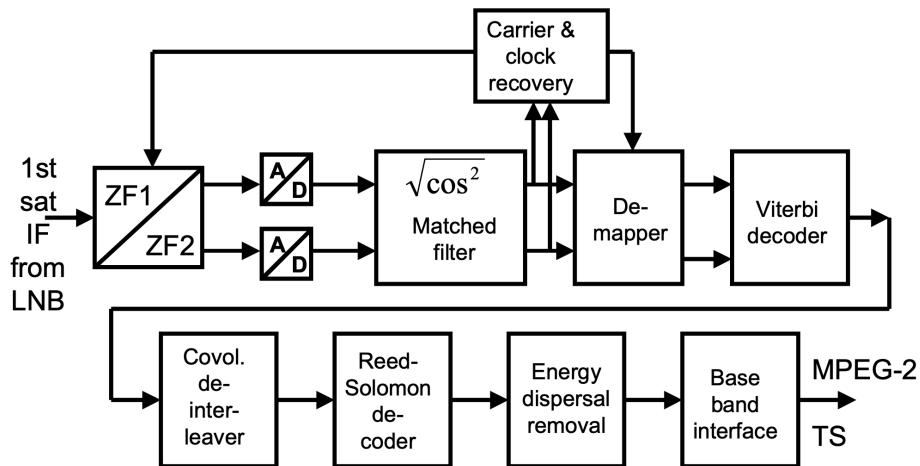
### PRIMERA PARTE



La primera parte también es conocida como **Low Noise Block (LNB)**. Se encuentra dentro de la antena y contiene una guía de onda con un detector para el plano de polarización vertical y otro para el horizontal, alternables. Los filtros se utilizan para filtrar las componentes indeseadas de la señal. Los amplificadores son necesarios ya que la señal se atenuó por lo menos 200 dB respecto al transmisor del satélite, normalmente son de arsienuro de galio.

Para el traslado de frecuencia, normalmente se cuenta con dos osciladores, uno de 9,75 GHz y otro en 10,6 GHz. Se decide cual se utiliza dependiendo de si la señal recibida se encuentra en la banda superior (comúnmente utilizada) o en la banda inferior.

### SEGUNDA PARTE



El sistema comienza trasladando la señal a una frecuencia intermedia más baja con un **mezclador IQ** para recuperar las señales IQ. Éstas pasan por un **conversor A/D** y se pasan a un **filtro adaptado RRC** con un roll-off de 0,35.

Posteriormente se debe recuperar la portadora y el reloj de la señal para demapear la señal. Una vez pasado este bloque, se puede pasar a decodificar la señal.

El primer bloque de la decodificación es el **decodificador Viterbi**, que se encarga de deshacer los cambios producidos por el codificador convolucional. Necesita saber el code rate que se utilizó en la codificación para poder funcionar correctamente.

El bloque siguiente es el **desentrelazador convolucional** que se encarga de deshacer el desorden que generó el entrelazador para proteger el sistema frente a una ráfaga de errores.

Ahora se recuperaron los datos en bloques de 204 bytes. Se utiliza el **decodificador Reed-Solomon** para quitar los 16 bytes de protección utilizados. Si el paquete tiene más de 8 errores, se marca el paquete como erróneo para que luego se descarte. A la salida de este bloque los paquetes son de 188 bytes

Por último, los paquetes pasan por el **removedor de dispersión de energía** que restaura las cadenas largas de unos y ceros y vuelve a invertir los bytes de sincronización. Una vez pasados por este bloque, los paquetes están listos para pasar por el decodificador MPEG-2.

Un dato a destacar es que a partir del filtro adaptado, todos los bloques suelen estar sobre un mismo chip.

## PROBLEMAS QUE ENFRENTA DVB-S

### MODULADOR IQ

El modulador puede presentar gran cantidad de problemas en casos de mal diseño o fallas. Entre ellos se encuentran la diferencia de amplificación entre las señales I y Q, un error en el desfasador de 90º y la no supresión de la portadora.

También pueden aparecer ruido o jitter, pero para que éstos afecten la señal, deben tener un orden de magnitud bastante grande.

### SATÉLITE

El satélite utiliza un Travelling Wave Tube Amplifier (TWTA) que posee varias no linealidades. Las mismas no pueden encontrarse en el ancho de banda de trabajo del sistema DVB-S.

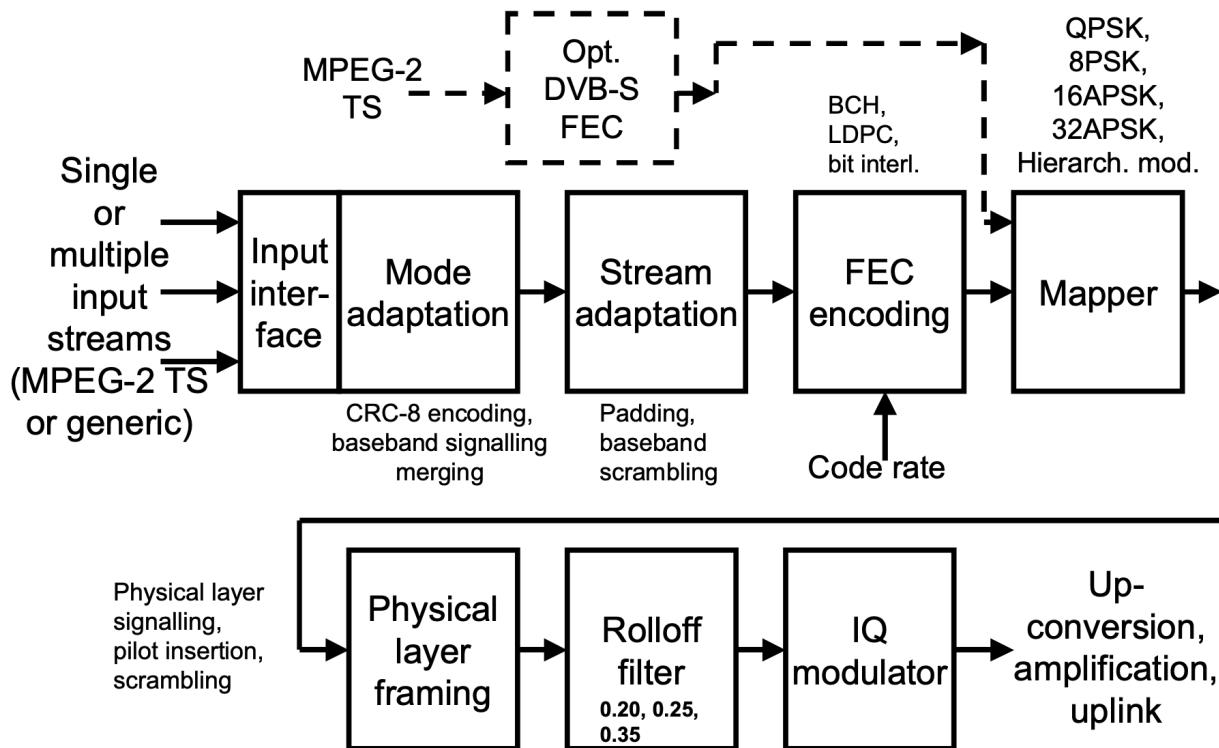
### MEDIO

Al transmitirse la señal, se va atenuando y se le va agregando AWGN. Este mismo ruido puede generar grandes problemas en el momento de recuperar la señal. Es necesario que el transmisor envíe la señal con una potencia tal que la señal no solo pueda ser detectada por el receptor, sino también entendida.

## CAMBIOS DVB-S2

Con el desarrollo y la experiencia del sistema DVB-S se fueron desarrollando diversas técnicas y esquemas de codificación que culminaron en el estándar DVB-S2, que trajo mejoras de rendimiento de 30% al mismo ancho de banda del transponder. Permitió transportar un canal HDTV en el mismo ancho de banda que un canal SDTV utilizando DVB-S.

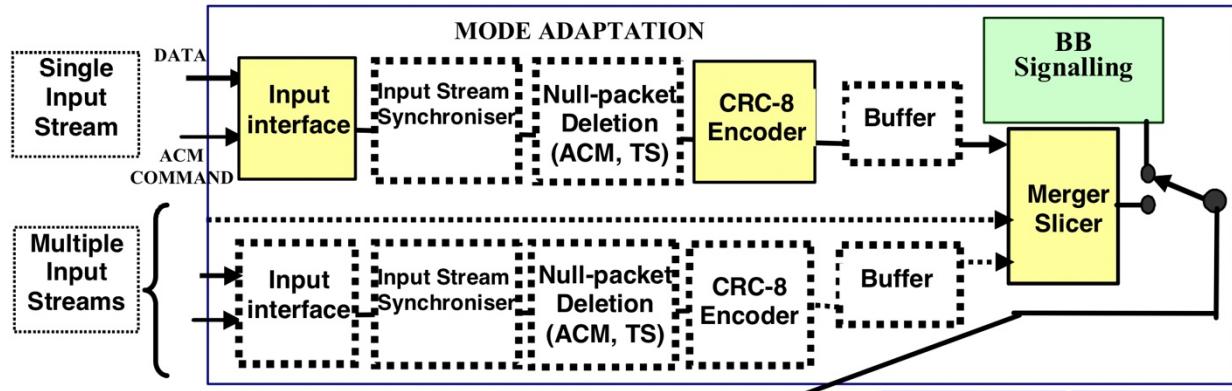
### ESQUEMA DEL SISTEMA



### MODE ADAPTATION

Este subsistema se encarga:

- Interfaz de entrada.
- Sincronización de el flujo de datos de entra (opcional).
- Supresión de paquetes nulos (flujos TS y ACM).
- Codificación de Redundancia Cíclica 8 (CRC-8) para detección de errores (para flujos paquetizados).
- Unión de varios flujos en uno solo (múltiples entradas).
- Cortar el flujo de datos en campos de datos.
- Señalización de banda base para informar el modo de adaptación utilizado.



### INPUT INTERFACE

La interfaz de entrada se encarga de mapear las entradas a un formato de bit lógico. Los flujos de entrada pueden ser de dos tipos:

- **Transport stream:** se componen de paquetes de usuario (UP) de 188 bytes (1316 bits), con el primer byte siendo de sincronización (47<sub>HEX</sub> o 71<sub>DEC</sub>).
- **Generic stream:** pueden ser un flujo continuo de bits o un flujo de paquetes de usuario (UP) de longitud constante. Los flujos con paquetes de longitud variable o una longitud de paquete mayor a 64 kbit son considerados flujos continuos. Los flujos continuos son marcados con una longitud de paquete de usuario (UPL) igual a cero. Si los paquetes no cuentan con un byte de sincronización, debe agregarse uno al principio con el valor 0<sub>D</sub> y aumentar el UPL en 8 bits.

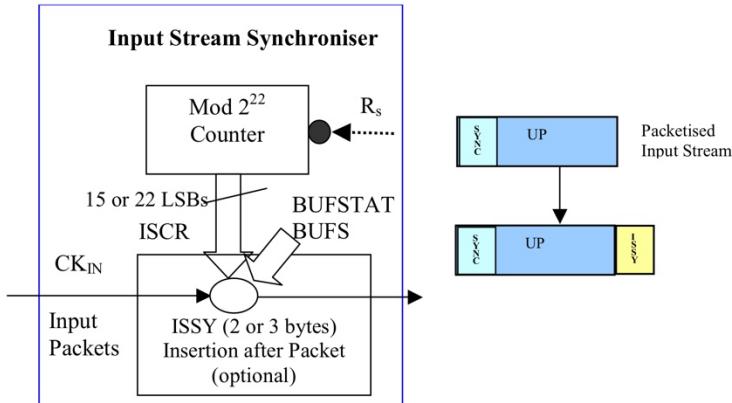
| Location         | Interface          | Interface type                                | Connection                 | Multiplicity       |
|------------------|--------------------|---|----------------------------|--------------------|
| Transmit station | Input              | MPEG [1, 4] Transport Stream (see note 1)     | from MPEG multiplexer      | Single or multiple |
| Transmit station | Input (see note 2) | Generic Stream                                | From data sources          | Single or multiple |
| Transmit station | Input (see note 3) | ACM command                                   | From rate control unit     | Single             |
| Transmit station | Output             | 70 MHz/140 MHz IF, L-band IF, RF (see note 4) | to RF devices              | Single or multiple |
| Transmit station | Input              | Mode Adaptation                               | from Mode Adaptation block | Single             |

NOTE 1: For interoperability reasons, the Asynchronous Serial Interface (ASI) with 188 bytes format, data burst mode (bytes regularly spread over time) is recommended.  
 NOTE 2: For data services.  
 NOTE 3: For ACM only. Allows external setting of the ACM transmission mode.  
 NOTE 4: IF shall be higher than twice the symbol rate.

### INPUT STREAM SYNCHRONIZER

Se encarga de garantizar una tasa de bits constante (CBR) y un retardo constante para los flujos paquetizados. Agrega un campo llamado Input Stream Synchronizer (ISSY). El mismo contiene información sobre tres variables:

- **Input Stream Time Reference (ISCR):** contiene los bits menos significativos del contador cuando el paquete es procesado y el instante en que los bits más significativos llegan al modulador.
- **BUFS:** tamaño máximo del buffer del receptor para compensar variaciones de retardo.. Se asume que el buffer del receptor opera como FIFO de un stream. Si se utiliza, debe enviarse cinco veces por segundo. El tamaño máximo del buffer es 20 Mbits.
- **BUFSTAT:** número de bits de relleno. También debe mandarse cinco veces por segundo si es utilizado. Se utiliza para arrancar el procedimiento de sincronización y verificar el funcionamiento en estado normal.

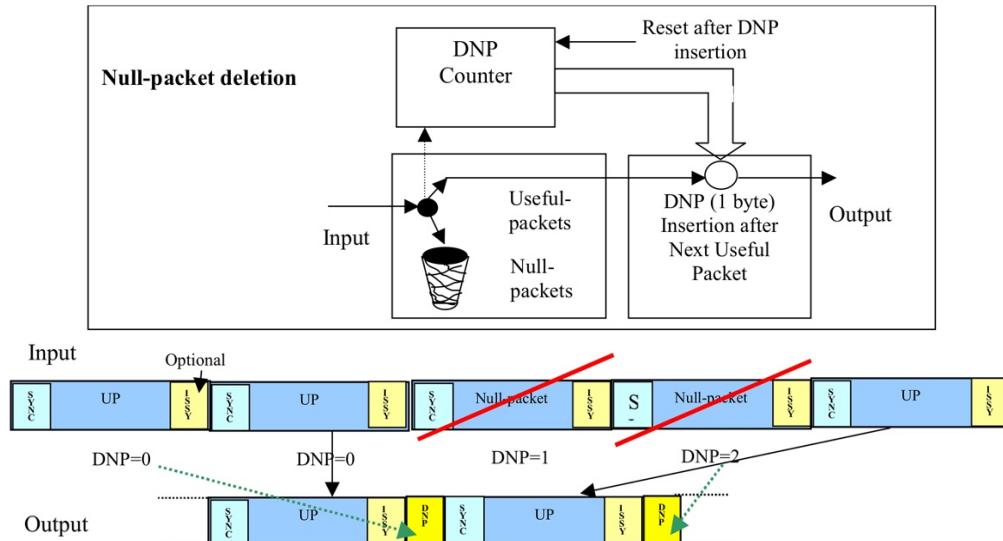


| First Byte                |                              |                                      |  |                   | Second Byte            | Third Byte  |
|---------------------------|------------------------------|--------------------------------------|--|-------------------|------------------------|---|
| bit-7 (MSB)               | bit-6                        | bit-5 and bit-4                      | bit-3 and bit-2  | bit-1 and bit-0   | bit-7 to bit-0         | bit-7 bit-0   |
| 0 = ISCR <sub>short</sub> | MSB of ISCR <sub>short</sub> | next 6 bits of ISCR <sub>short</sub> |  |                   |                        |   |
| 1                         | 0 = ISCR <sub>long</sub>     | 6 MSBs of ISCR <sub>long</sub>       |  |                   |                        |   |
| 1                         | 1                            | 00 = BUFS                            | BUFS unit<br>00 = bits<br>01 = Kbits<br>10 = Mbits<br>11 = reserved    | 2 MSBs of BUFS    | next 8 bits of BUFS    | not present when ISCR <sub>short</sub> is used; else reserved |
| 1                         | 1                            | 10 = BUFSTAT                         | BUFSTAT unit<br>00 = bits<br>01 = Kbits<br>10 = Mbits<br>11 = reserved | 2 MSBs of BUFSTAT | next 8 bits of BUFSTAT | not present when ISCR <sub>short</sub> is used; else reserved |
| 1                         | 1                            | others = reserved                    | reserved   | reserved          | reserved               | not present when ISCR <sub>short</sub> is used; else reserved |

NOTE: For Generic Packetized Streams optional ISCR shall be limited to the "short" format.

### NULL-PACKET DELETION

Se utiliza exclusivamente en transport streams, con PID 8191<sub>DEC</sub> o 1FFF<sub>HEX</sub>, y en ACM. Esto ayuda a reducir la tasa de información y aumentar la protección a errores. En el receptor los paquetes son reinsertados. Se utiliza un contador para saber la cantidad de paquetes nulos que deben reinsertarse posteriormente.

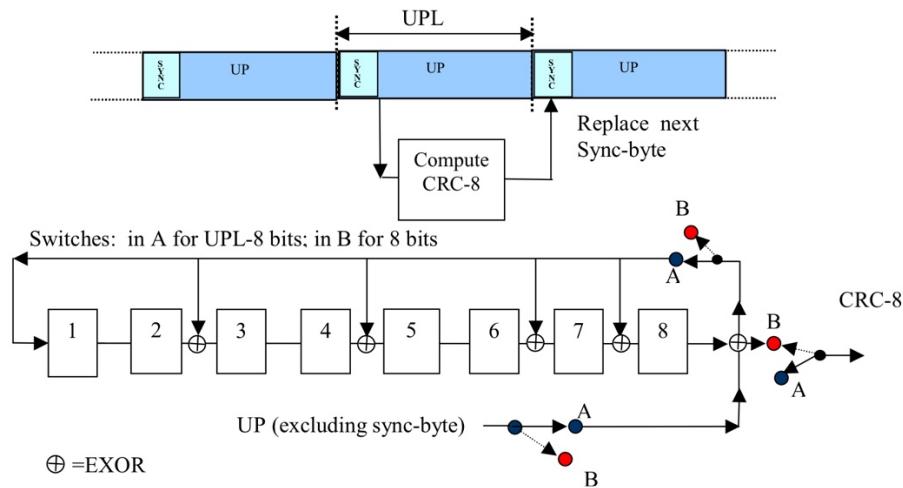


## CRC-8 ENCODER

Solo se utiliza para flujos empaquetados únicamente. Se revisa el campo de longitud de paquete de usuario sea distinto de cero y se procede a hacer codificación de redundancia cíclica de ocho bits sobre el paquete menos el sync byte. El polinomio generador utiliza el siguiente polinomio generador:

$$g(x) = x^8 + x^7 + x^6 + x^4 + x^2 + 1$$

El registro se inicializa con todos ceros antes del primer bit de cada secuencia. El sync byte del siguiente paquete es reemplazado por el código. Esto no genera problemas porque el sync byte es copiado en el BBHEADER.

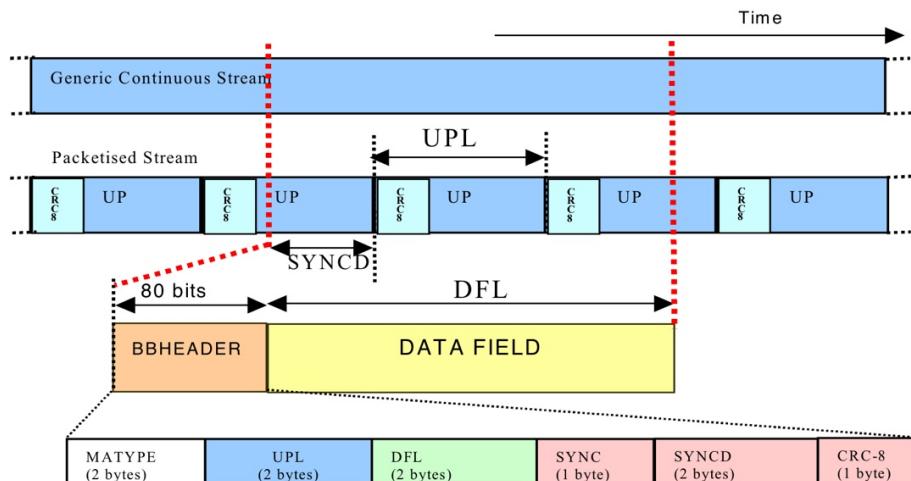


## MERGER/Slicer

Se encarga de cortar el flujo a la entrada en campos de datos. Si hay varios flujos de datos, se concatenan los campos de datos de cada flujo para ser enviados. La longitud de los campos de datos está limitada por la cantidad de bits en un bloque BCH no codificado y por el encabezado de banda base. Dependiendo del modo de uso, se mandan los paquetes de usuarios de forma finita (DFL variable) o se cortan los paquetes dependiendo de la capacidad máxima de campo de datos (DFL constante).

$$DFL \leq k_{bch} - 80 \text{ bits}$$

Este bloque también debe detectar la distancia entre el comienzo del campo de datos y el fin del paquete de usuario y agregar al campo SYNC'D del encabezado de banda base.



## BBHEADER INSERTION

El BBHeader es un encabezado de 10 bytes que se inserta antes del campo de datos. El mismo cuenta con los siguientes campos:

- **MATYPE-1 (1 byte):**
  - *TS/GS (2 bit)*: indica la entrada es un transport stream o generic stream y se es continuo o empaquetado.
  - *SIS/MIS (1 bit)*: indica si es un stream de una entrada o múltiples entradas.
  - *CCM/ACM (1 bit)*: indica si se utiliza codificación y modulación constante o adaptativa.
  - *ISSYI (1 bit)*: indica si el campo ISSY es insertado después del paquete de usuario.
  - *NPD (1 bit)*: indica si la supresión de paquetes nulos está activada.
  - *RO (2 bits)*: indica el factor de roll-off.

| TS/GS                   | SIS/MIS      | CCM/ACM | ISSYI          | NPD            | RO            |
|-------------------------|--------------|---------|----------------|----------------|---------------|
| 11 = Transport          | 1 = single   | 1 = CCM | 1 = active     | 1 = active     | 00 = 0,35     |
| 00 = Generic Packetized | 0 = multiple | 0 = ACM | 0 = not-active | 0 = not-active | 01 = 0,25     |
| 01 = Generic continuous |              |         |                |                | 10 = 0,20     |
| 10 = reserved           |              |         |                |                | 11 = reserved |

- **MATYPE-2 (1 byte):** se utiliza como identificador del flujo de entrada en caso de múltiples entradas en un flujo (MIS).
- **UPL (2 bytes):** indica la longitud del paquete de usuario en bits. Puede tomar un valor entre 0 y 65535.
- **DFL (2 bytes):** indica la longitud del campo de dato sen bits. Puede tomar un valor entre 0 y 58112.
- **SYNC (1 byte):** es una copia del sync byte del paquete de usuario. Si tiene el valor 00<sub>HEX</sub> el decodificador CRC-8 debe quitar los bits en vez de reemplazarlos.
- **SYNCD (2 bytes):** se utiliza en flujo de paquetes como la distancia en bits desde el inicio del campo de datos y el primer paquetes de usuario de la trama. Si el valor es 65535<sub>DEC</sub>, no comienza ningún paquete de usuario en la trama.
- **CRC-8 (1 byte):** codificación de detección de error sobre los primeros nueve bytes.

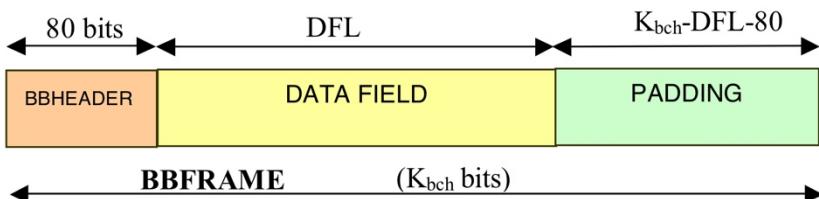
| Application area/configuration         | MATYPE-1     | MATYPE-2 | UPL                 | DFL                               | SYNC              | SYNCD | CRC-8 | Slicing policy                                      |
|--|--------------|----------|---------------------|-----------------------------------|-------------------|-------|-------|---|
| Broadcasting services / CCM, single TS | 11-1-1-0-0-Y | XXXXXXXX | 188 <sub>D</sub> X8 | K <sub>bch</sub> -80 <sub>D</sub> | 47 <sub>HEX</sub> | Y     | Y     | Break<br>No timeout<br>No Padding<br>No Dummy frame |

X= not defined; Y = according to configuration/computation.  
Break = break packets in subsequent DATAFIELDS; Timeout: maximum delay in merger/slicer buffer.

## STREAM ADAPTATION

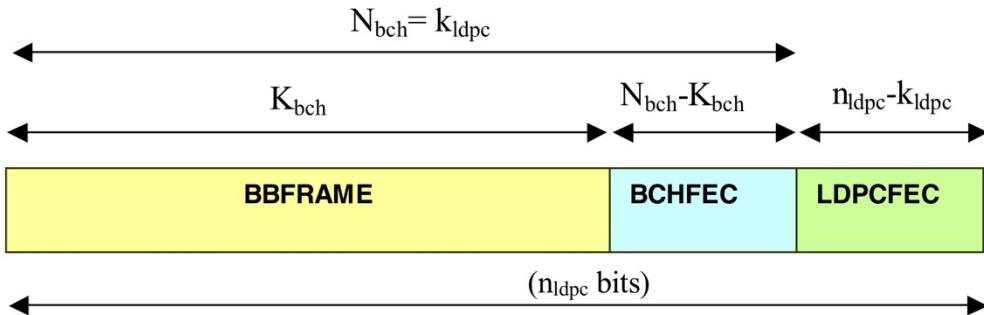
Esta parte del sistema se encarga de llenar la trama de banda base (BBFrame) al valor de bloque BCH sin codificar ( $k_{BCH}$ ) que depende del FEC rate. También se encarga de aleatorizar la trama con una PRBS sincronizada con la misma. Cuando empieza una BBFrame, vuelve a empezar la secuencia. El polinomio generador de la misma es:

$$g(x) = x^{15} + x^{14} + 1$$



## FEC ENCODING

Este subsistema se encarga de codificación externa (BCH), la codificación interna (LDPC) y del entrelazamiento de bit. Antes del entrelazamiento, la trama puede ser de 16200 bits o de 64800 bits y tendrá la siguiente forma:



El **código Bose–Chaudhuri–Hocquenghem (BCH)** es una generalización de los códigos de Hamming. Es altamente configurable<sup>[AV]</sup>, permitiendo seleccionar la longitud de los bloques, la tasa de código, el tamaño del alfabeto y la capacidad de corrección. El rendimiento de BCH cuando se trabaja con longitud de bloque de unas centenas es considerablemente mayor frente a otros códigos con la misma longitud de bloque y tasa de código. Su funcionamiento es el siguiente:

1. Se multiplica el mensaje a transmitir por  $x$  elevado a la diferencia entre la longitud de la trama de salida y la de entrada.

$$m(x) \cdot x^{n_{BCH} - k_{BCH}}$$

2. Dividir el resultado por el polinomio generador  $g(x)$ .

$$d(x) = \frac{m(x) \cdot x^{n_{BCH} - k_{BCH}}}{g(x)}$$

3. Generar el polinomio de la palabra de código.

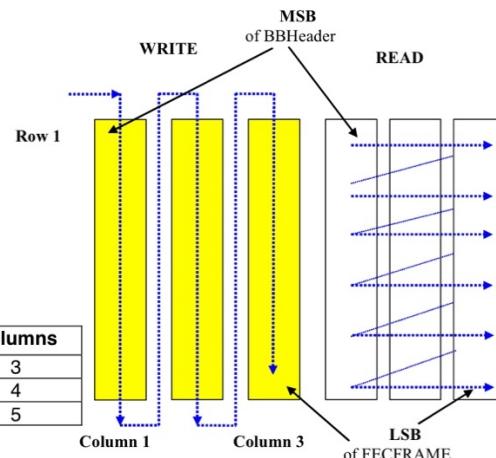
$$c(x) = m(x) \cdot x^{n_{BCH} - k_{BCH}} + d(x)$$

La trama luego pasa por la codificación interna. Para esta, se utiliza **low density parity check (LDPC)**. Es un sistema de codificación de corrección de errores lineal que toman la capacidad en consideración, lo que permite trabajar cerca del límite de Shannon. Agrega a cada bloque una cantidad de bits de paridad igual a  $n_{LDPC} - k_{LPDC}$ .

Las ventajas que proporciona este código son posibles porque los subsets de bits de paridad son de menor tamaño, lo que reduce el procesamiento considerablemente. Aun así, los subsets toman en consideración los propios bits de paridad y por lo que permite tener un BER muy bajo.

Una vez codificada la señal, se utiliza un **entrelazador de bit**. Se cargan los datos de manera serial en las columnas de las matrices. Después, se leen los datos leyendo cada fila en vez de la columna. El MSB del encabezado de banda base es el primero que se lee, salvo en 8PSK 3/5 donde se lee tercero. Este procedimiento solo se hace cuando el esquema de modulación utilizado es 8PSK, 16APSK o 32 APSK.

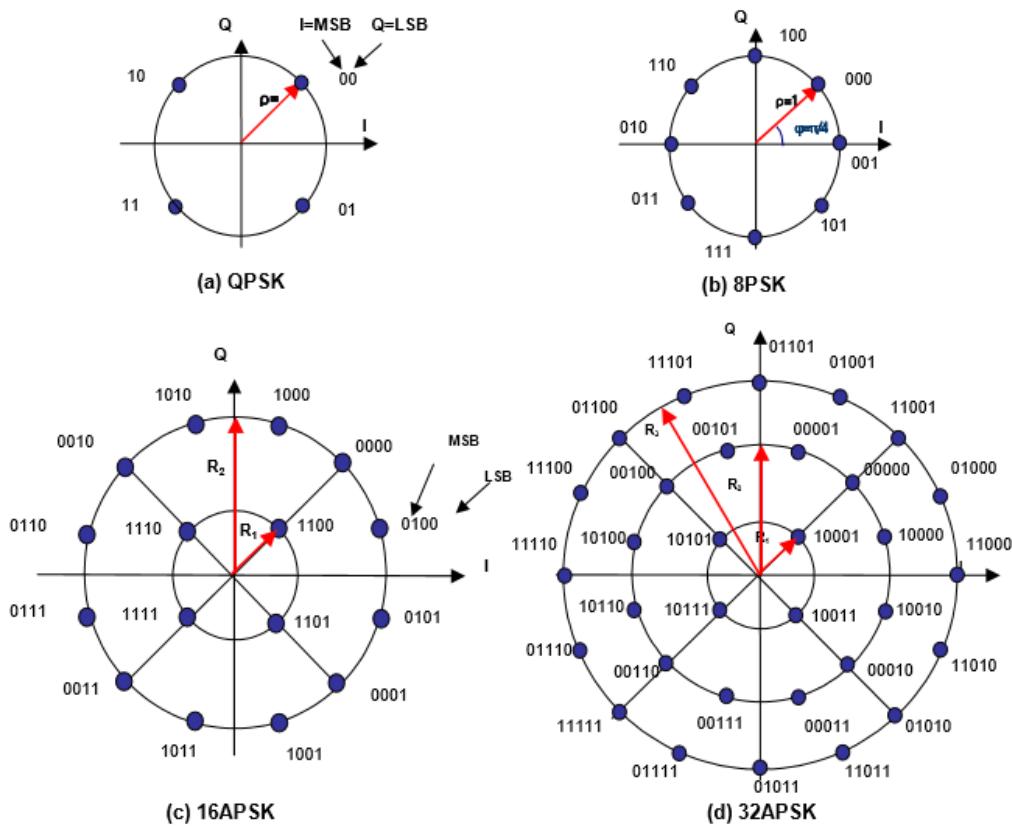
| Modulation | Rows (for $n_{ldpc} = 64\ 800$ ) | Rows (for $n_{ldpc} = 16\ 200$ ) | Columns |
|------------|----------------------------------|----------------------------------|---------|
| 8PSK       | 21 600                           | 5 400                            | 3       |
| 16APSK     | 16 200                           | 4 050                            | 4       |
| 32APSK     | 12 960                           | 3 240                            | 5       |



## MAPPING SYSTEM

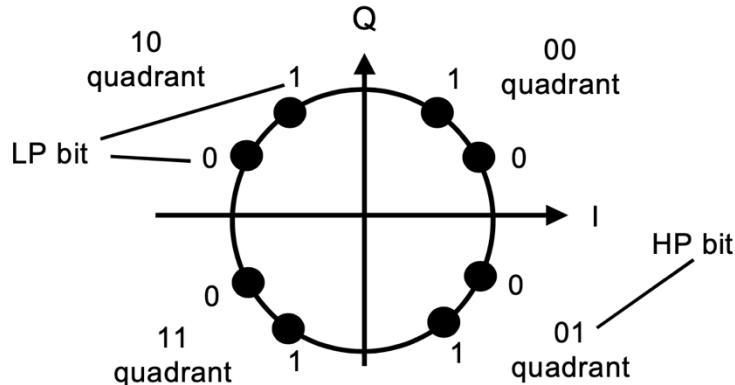
Además del esquema **QPSK**, utilizado en DVB-S, se pueden utilizar **8PSK**, **16APSK** y **32APSK** (utilizado en Digital Satellite News Gathering). Los mismos siguen utilizando **código Gray**, ya que suele minimizar el error de bit por error de símbolo, y no utilizan modulación diferencial.

| Mod.   | CR<br>=1/4 | CR<br>=1/3 | CR<br>=2/5 | CR<br>=1/2 | CR<br>=3/5 | CR<br>=2/3 | CR<br>=3/4 | CR<br>=4/5 | CR<br>=5/6 | CR<br>=8/9 | CR<br>=9/10 |
|--------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|-------------|
| QPSK   | -2.4<br>dB | -1.2<br>dB | 0<br>dB    | 1<br>dB    | 2.2<br>dB  | 3.1<br>dB  | 4<br>dB    | 4.6<br>dB  | 5.2<br>dB  | 6.2<br>dB  | 6.5<br>dB   |
| 8PSK   | -          | -          | -          | -          | 5.5<br>dB  | 6.6<br>dB  | 7.9<br>dB  | -          | 9.4<br>dB  | 10.6<br>dB | 11<br>dB    |
| 16APSK | -          | -          | -          | -          | -          | 9<br>dB    | 10.2<br>dB | 11<br>dB   | 11.6<br>dB | 12.9<br>dB | 13.1<br>dB  |
| 32APSK | -          | -          | -          | -          | -          | -          | 12.6<br>dB | 13.6<br>dB | 14.3<br>dB | 15.7<br>dB | 16.1<br>dB  |



Otro esquema utilizado es el esquema de **modulación jerárquica QPSK**, que tiene la ventaja de tener compatibilidad con DVB-S. La misma se puede interpretar de dos maneras:

- Se define a cada cuadrante como un punto de la constelación, ganando dos bits para transmisión de alta prioridad.
- Discerniendo entre dos puntos por cuadrante puedo agregar un bit de baja prioridad para esta modulación. Tengo un total de tres bits por símbolo.



## PHYSICAL LAYER FRAMING

Este sistema debe cumplir las siguientes funciones:

- Enviar un Dummy PLFrame cuando la trama a la entrada no está lista para ser procesada
- Rebanar la XFECFrame en slots de 90 símbolos.
- Generar el encabezado de capa física (PLH) de 90 símbolos e insertarlo antes del XFECF.
- Insertar bloques de piloto de 36 símbolos cada 16 slots.
- Aleatorización de las señales IQ.

El **Dummy PLFrame** esta compuesto por el PLHeader y por 36 slots de la portadora sin modular. El **PLHeader** es de 90 símbolos y deber ser modulado con  $\pi/2$  BPSK. Cuenta con los siguientes campos:

- **SOF (26 símbolos):** indica el principio de la trama. Tiene el valor  $18D2E82_{HEX}$ .
- **PLS code (64 símbolos):** el código de señalización de capa física tiene una distancia mínima de Hamming de 32. Se utiliza para transmitir 7 bits de señalización que se reparten en dos campos:
  - MODCOD (5 símbolos): indica la modulación y el FEC rate.
  - TYPE (2 símbolos): indica la longitud del FECF y la presencia de los pilotos.

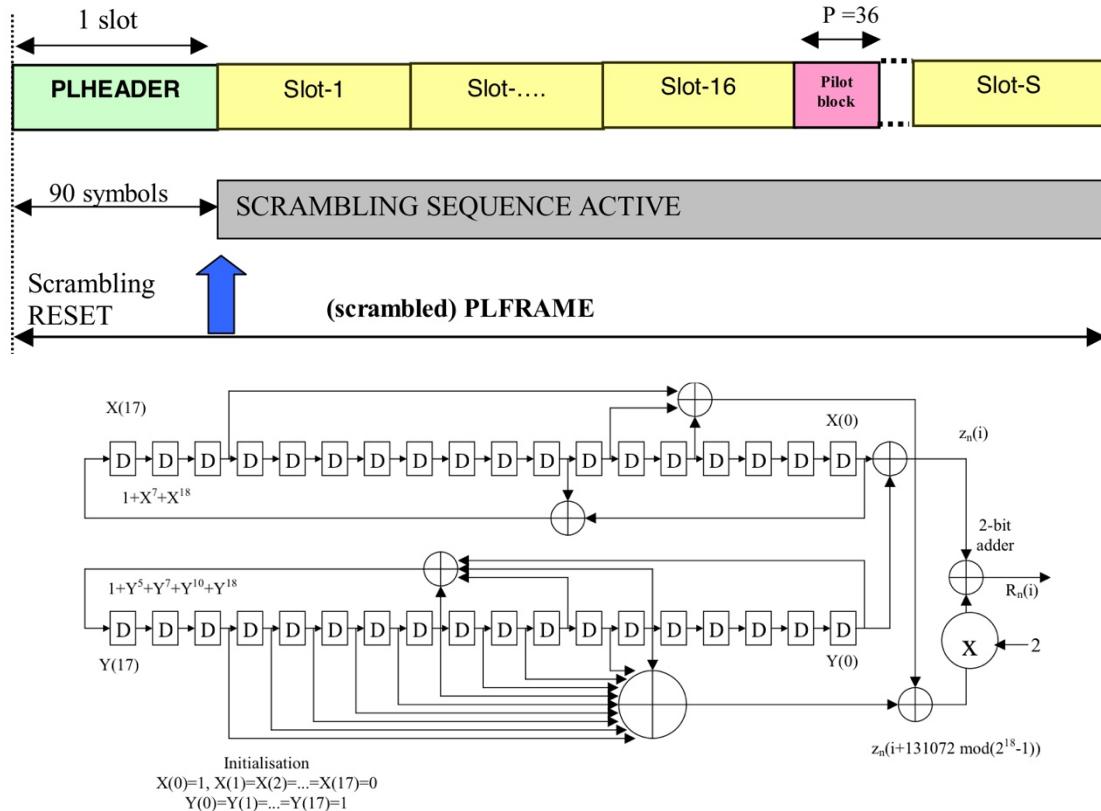
| Mode     | MOD COD | Mode      | MOD COD | Mode        | MOD COD | Mode          | MOD COD |
|----------|---------|-----------|---------|-------------|---------|---------------|---------|
| QPSK 1/4 | $1_D$   | QPSK 5/6  | $9_D$   | 8PSK 9/10   | $17_D$  | 32APSK 4/5    | $25_D$  |
| QPSK 1/3 | $2_D$   | QPSK 8/9  | $10_D$  | 16APSK 2/3  | $18_D$  | 32APSK 5/6    | $26_D$  |
| QPSK 2/5 | $3_D$   | QPSK 9/10 | $11_D$  | 16APSK 3/4  | $19_D$  | 32APSK 8/9    | $27_D$  |
| QPSK 1/2 | $4_D$   | 8PSK 3/5  | $12_D$  | 16APSK 4/5  | $20_D$  | 32APSK 9/10   | $28_D$  |
| QPSK 3/5 | $5_D$   | 8PSK 2/3  | $13_D$  | 16APSK 5/6  | $21_D$  | Reserved      | $29_D$  |
| QPSK 2/3 | $6_D$   | 8PSK 3/4  | $14_D$  | 16APSK 8/9  | $22_D$  | Reserved      | $30_D$  |
| QPSK 3/4 | $7_D$   | 8PSK 5/6  | $15_D$  | 16APSK 9/10 | $23_D$  | Reserved      | $31_D$  |
| QPSK 4/5 | $8_D$   | 8PSK 8/9  | $16_D$  | 32APSK 3/4  | $24_D$  | DUMMY PLFRAME | $0_D$   |

Los dos campos son codificados biortogonalmente con un código (64,7). Esto permite que todos los bits impares sean o iguales al anterior o opuestos, lo que facilita el funcionamiento de detección diferencial coherente en el receptor. Luego se multiplica los 64 bits de salida por otra secuencia binaria. La matriz generadora del código es la siguiente:

$$G = \begin{bmatrix} 01010101010101010101010101 \\ 0011001100110011001100110011 \\ 00001111000011110000111100001111 \\ 0000000011111110000000011111111 \\ 00000000000000011111111111111111 \\ 11111111111111111111111111111111 \end{bmatrix} \quad 0111000110011101100001110010010101001101000010001011011111010$$

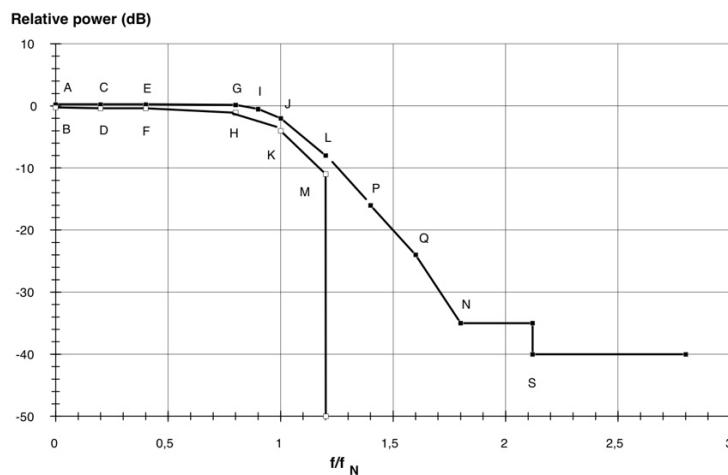
Puede configurarse si el sistema realiza la **inserción de pilotos**. Las mismas se insertan cada 16 slots de la trama de banda base. Si su posición coincide con el Start of Frame (SOF), no se inserta. En sistemas de codificación y modulación variable o adaptada (VCM o ACM) se puede configurar la inserción de pilotos de trama en trama.

Como paso final en esta etapa, se realiza una **dispersión de energía** sobre los datos de PLFrame. La secuencia que se utiliza se reinicia al final de cada PLHeader. La misma no tiene impacto en el ancho de banda que utiliza el sistema.



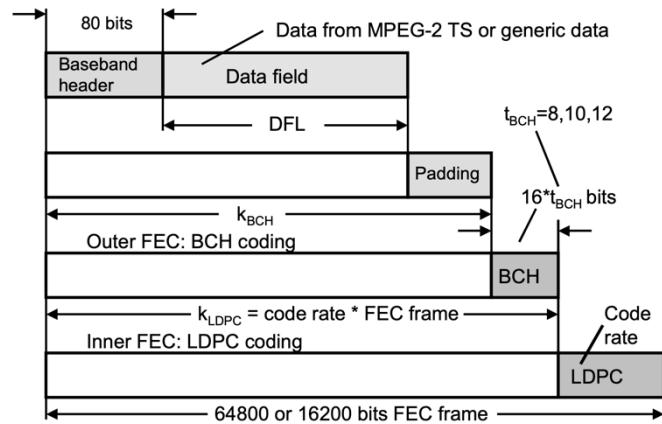
## MODULATION

La etapa final del sistema. Recibe las PLFrames, utiliza un **filtro RRC** con factor de roll-off 0.2, 0.25 o 0.35. Una vez filtradas, se las **multiplica por la portadora** con frecuencia  $f_0$  y se las suma. El espectro de la señal de salida debe mantenerse dentro de las dos figuras que se muestran a continuación.

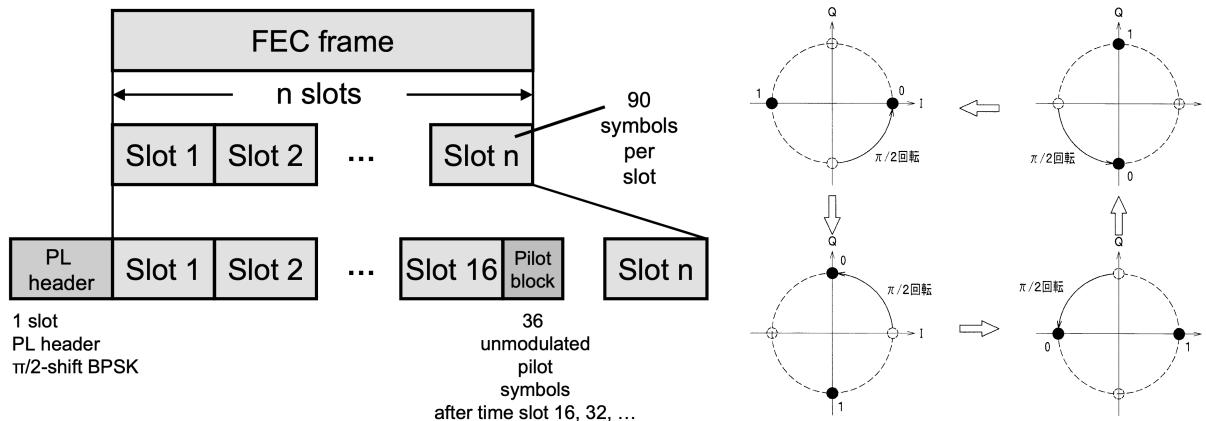


## ESTRUCTURA DE TRAMA

El nuevo estándar funciona con tramas. Primero se codifica la información con **Forward Error Correction (FEC)**. Cada trama FEC tiene la información a ser transmitida. A esa información se le agrega un header de 80 bits. Una vez colocado el header, se coloca el padding que depende del code rate seleccionado para esta codificación. La trama puede ser de 64800 bits o de 16200 bits, dependiendo del modo utilizado. Luego la trama es codificada utilizando codificación **Bose-Chaudhuri-Hocquenghem (BCH)** y **Low Density Parity Check (LDPC)**.



Luego de mapear la trama al esquema de modulación utilizado, se divide la trama en una trama de capa física de N puestos. El primer puesto es el header de la trama, que es modulada con  $\pi/2$  shift BPSK para garantizar sincronización. Luego se mandan 16 slots de datos con un slot para el bloque piloto (17). Se transmite la información repitiendo el procedimiento de 16 bloques de datos y uno piloto. Cada slot tiene 90 símbolos mientras que los bloques de la piloto tienen 36 símbolos.



La **velocidad de transmisión neta** está dada por la siguiente ecuación:

$$\text{velocidad neta}_{FEC+PLF}(NDR_{FEC+PLF}) = \frac{R_s \cdot [L_{FEC} \cdot R_c - (16 \cdot t_{BCH}) - 80]}{\frac{L_{FEC}}{Q} + 90 + \text{ceil}\left(\frac{L_{FEC}}{Q \cdot 90 \cdot 16} - 1\right) \cdot 36}$$

Donde,

$R_s \rightarrow$  velocidad de modulación (símbolos/segundos o baudios)

$L_{FEC} \rightarrow$  longitud de trama FEC (64800 o 16200)

$R_c \rightarrow$  tasa de código de LPDC

$t_{BCH} \rightarrow$  cantidad de polinomios BCH

$Q \rightarrow$  bits/símbolo, depende del esquema de modulación usado

90  $\rightarrow$  cantidad de símbolos del PL Header

80  $\rightarrow$  cantidad de bits del Baseband Header

36 → cantidad de símbolos de piloto

16 → Slots entre pilot blocks

## ANEXO I - OBTENCIÓN DEL RADIO DE LA ÓRBITA GEOESTACIONARIA

Para poder mantener un satélite en órbita, debo garantizar que la fuerza gravitacional que produce la Tierra sobre el satélite sea igual en módulo pero en sentido opuesto a la fuerza centrífuga sobre el mismo debido a su movimiento.

Fuerza centrípeta

$$F_G = \frac{G \cdot m_{TIERRA} \cdot m_{SAT}}{r^2}$$

$G = 6,67 \cdot 10^{-11} \frac{m^3}{kg \cdot s^2}$ , constante gravitacional.

$m_{TIERRA} = 5,972 \cdot 10^{24} kg$ , masa de la Tierra.

$m_{SAT}$ , masa del satélite.

$r$ , distancia entre los centros de los cuerpos.

Fuerza centrífuga

$$F_C = m_{SAT} \cdot \omega^2 \cdot r$$

$\omega = \frac{2\pi}{T} = \frac{2\pi}{86400} \frac{rad}{s}$ , velocidad angular del satélite.

$m_{SAT} = 5,972 \cdot 10^{24} kg$ , masa del satélite.

$r$ , distancia entre los centros de los cuerpos.

$$\frac{G \cdot m_{TIERRA} \cdot m_{SAT}}{r^2} = m_{SAT} \cdot \omega^2 \cdot r$$

$$r = \sqrt[3]{\frac{G \cdot m_{TIERRA}}{\omega^2}} = \sqrt[3]{\frac{6,67 \cdot 10^{-11} \cdot 5,972 \cdot 10^{24} \cdot 86400^2}{4\pi^2}} m = 42.231 km$$

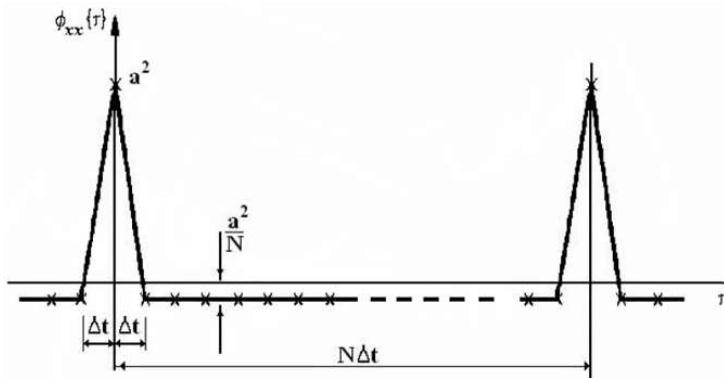
Entonces,

$$d_{SUP} = r - r_{SUP} = 42.231 km - 6.371 km = 35.860 km$$

## ANEXO II – PSEUDO RANDOM BINARY SEQUENCE

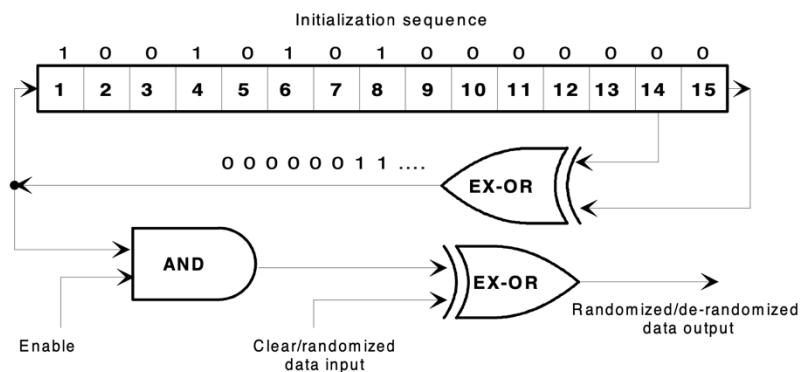
Las secuencias pseudo aleatorias son muy utilizadas comúnmente en la transmisión ya que cuenta con las siguientes características:

- **Balance:** la cantidad de unos es igual a la cantidad de ceros. La componente continua de la señal es muy cercana a 0.
- No hay **largas secuencias** con el mismo número. Esto facilita la capacidad de sincronización.
- La **correlación cruzada de la secuencia** con un shifteo cíclico es pequeña.



Para generar estas secuencias se utiliza realimentación lineal con shifteo de registro, específicamente PRBS15 que esta dado por el polinomio  $x^{15} + x^{14} + 1$ .

La señal enable se debe a que quiero mantener el sync byte tanto normal como negado. Al estar en cero, permite que la señal de entrada no sea modificada por la PRBS.



La secuencia utilizada tiene la característica de ser una secuencia de longitud máxima  $m=15$ , que tiene las siguientes propiedades:

- La cantidad de estados posibles es  $2^{15} = 32768$
- El estado de todos ceros no está permitido porque no se puede salir de ese estado.
- La longitud máxima de la secuencia es igual  $2^{15} - 1 = 32767$ .
- La proporción de las corridas está dada por  $\frac{1}{2^i}$  donde  $i$  es la longitud de la corrida.
  - La corrida más larga de unos está dada por la cantidad de registros, entonces tengo una corrida de longitud 15 (m).
  - Hay dos corridas de ceros y dos corridas de unos de longitud 14 (m-1).
  - Hay 15 secuencias de longitud 1, 8 de 1 y 7 de 0.

A continuación se muestra el código que genera una secuencia pseudo aleatoria máxima de longitud máxima 15:

```
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>

int main(void) {
    uint16_t start = 0x02;
    uint16_t a = start;
    int i;
    for (i = 1;; i++) {
        int newbit = (((a >> 14) ^ (a >> 13)) & 1);
        a = ((a << 1) | newbit) & 0x7fff;
        printf("%x\n", a);
        if (a == start) {
            printf("repetition period is %d\n", i);
            break;
        }
    }
    return 0;
}
```

### ANEXO III – REED-SOLOMON FORWARD ERROR CORRECTION

Esta protección aprovecha el concepto matemático de campos de Galois de aritmética finita. Deja de considerarse la cadena de bits y se trabaja con una cadena de símbolos donde cada símbolo es un conjunto de bits. Esto implica que no se considera los errores de bits en el símbolo, sino errores del propio símbolo. La codificación Reed-Solomon es un tipo especial de código cíclico. Los códigos cílicos forman parte de los códigos lineales y tienen la característica que son fáciles de implementar con técnicas de shifteo y realimentaciones. La diferencia con los códigos CRC, es que al utilizar los campos de Galois, se está trabajando con las potencias de los números primos, pudiendo trabajar solo en modulo de ese numero primo. Esto los hace perfectos para trabajar con la electrónica ya que si utilizamos el número 2, estaríamos trabajando con módulo 2. Esto es equivalente a utilizar compuertas XOR, que pueden realizarse fácilmente por hardware.

### CÓDIGOS LINEALES

Se empieza teniendo una **matriz generadora ( $\mathbf{G}$ )** que tiene todos los vectores no lineales del subespacio. El mensaje que quiero transmitir se multiplica por esta matriz generadora para generar la palabra de código que voy a transmitir. La misma está formada por el código de protección y la matriz identidad. Las filas ( $k$ ) son la cantidad entradas que tengo en un bloque mientras que las columnas ( $n$ ) son la cantidad de salidas que tengo del mismo, con esto puedo sacar el code rate.

$$\mathbf{G} = \left[ \begin{array}{c|c} \mathbf{P} & \mathbf{I}_k \end{array} \right] = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1,(n-k)} & 1 & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & p_{2,(n-k)} & 0 & 1 & \cdots & 0 \\ \vdots & & & & & & \ddots & \\ p_{k1} & p_{k2} & \cdots & p_{k,(n-k)} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

En el lado del receptor, necesito una **matriz de chequeo de paridad ( $\mathbf{H}$ )** que es la inversa de la matriz generadora de manera que pueda traducir la palabra de código al mensaje que se envió.

$$\mathbf{H}^T = \left[ \begin{array}{c|c} \mathbf{I}_{n-k} & \mathbf{P} \end{array} \right] = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 1 \\ p_{11} & p_{12} & \cdots & p_{1,(n-k)} \\ p_{21} & p_{22} & \cdots & p_{2,(n-k)} \\ \vdots & & & \\ p_{k1} & p_{k2} & \cdots & p_{k,(n-k)} \end{bmatrix}$$

Se puede considerar la palabra de código recibida ( $r = U + e$ ) como la suma de la transmitida más (módulo 2) un error. Esto permite detectar errores a través del **síndrome ( $\mathbf{S}$ )**. Para que no haya error, el síndrome debe ser 0. La cantidad de palabras de códigos erroneos ( $e$ ) que puedo recibir son  $2^{n-k}$ .

$$S = r \cdot H^T = (U + e) \cdot H^T = U \cdot H^T + e \cdot H^T = 0 + e \cdot H^T$$

Definiendo la **distancia de Hamming ( $d$ )** como la cantidad de bits diferentes entre palabras de código. La misma me sirve para saber la cantidad de errores que puedo corregir ( $Q_{ec}$ ) o detectar ( $Q_{ed}$ ) por palabra de código. En caso de tener una cantidad de errores mayor a los detectables, es probable que ese error esté enmascarado en una palabra de código válida.

$$Q_{ed} = d_{min} - 1, \quad Q_{ec} = \frac{d_{min} - 1}{2}$$

## COMPONENTES DEL CÓDIGO RS

En los códigos cíclicos (como el caso de RS), se trabaja con polinomios generadores para obtener la matriz generadora. En DVB-S se utiliza, por norma, el siguiente polinomio generador:

$$g(x) = \sum_{i=0}^{15} (x + 0x02^i)$$

También existe el polinomio de campo de Galois, donde se mapea el mensaje. Por norma es:

$$p(x) = x^8 + x^4 + x^3 + x^2 + 1 \text{ ó } 11d_{HEX}$$

Con  $p(x)$  genero el mensaje y multiplico a  $g(x)$  con  $p(x)$  para generar el código.

$$c(x) = p(x) \cdot g(x)$$

## NOMENCLATURA CÓDIGO RS

La codificación RS de símbolos hechos de secuencias de  $m$ -bits ( $m > 2$ ) se expresa como:

$$RS(n, k), 0 < k < n < 2^m + 2$$

Donde **k** es el número de bytes por símbolo y **n** es la cantidad de palabras de código.

También puede escribirse como

$$RS(n, k) = RS(2^m - 1, 2^m - 1 - 2t)$$

Donde **t** es la cantidad de bytes que puedo corregir.

La distancia mínima de Hamming máxima es

$$d_{min} = n - k + 1$$

Para el caso de DVB-S, se utiliza un código acortado  $RS(255, 239) \rightarrow RS(204, 188)$

$$RS(n, k) = RS(255, 239)$$

$$m = \log_2(255 + 1) = 8$$

$$t = \frac{256 - 1 - 239}{2} = 8$$

$$d_{min} = 255 - 239 + 1 = 16$$

Como se puede ver, se respeta la propiedad de errores corregibles por un código lineal.

#### ANEXO IV – BOSE-CHAUDHURY-HOCQUENGHEM

A continuación se muestran las posibles configuraciones que puede tener esta configuración para DVB-S2 y su polinomio generador.

| LDPC code | BCH Uncoded Block $K_{bch}$ | BCH coded block $N_{bch}$<br>LDPC Uncoded Block $k_{ldpc}$ | BCH t-error correction | LDPC Coded Block $n_{ldpc}$ |
|-----------|-----------------------------|--|------------------------|-----------------------------|
| 1/4       | 16 008                      | 16 200   | 12                     | 64 800                      |
| 1/3       | 21 408                      | 21 600   | 12                     | 64 800                      |
| 2/5       | 25 728                      | 25 920   | 12                     | 64 800                      |
| 1/2       | 32 208                      | 32 400   | 12                     | 64 800                      |
| 3/5       | 38 688                      | 38 880   | 12                     | 64 800                      |
| 2/3       | 43 040                      | 43 200   | 10                     | 64 800                      |
| 3/4       | 48 408                      | 48 600   | 12                     | 64 800                      |
| 4/5       | 51 648                      | 51 840   | 12                     | 64 800                      |
| 5/6       | 53 840                      | 54 000   | 10                     | 64 800                      |
| 8/9       | 57 472                      | 57 600   | 8                      | 64 800                      |
| 9/10      | 58 192                      | 58 320   | 8                      | 64 800                      |

|             |   |
|-------------|---|
| $g_1(x)$    | $1+x^2+x^3+x^5+x^{16}$  |
| $g_2(x)$    | $1+x+x^4+x^5+x^6+x^8+x^{16}$  |
| $g_3(x)$    | $1+x^2+x^3+x^4+x^5+x^7+x^8+x^9+x^{10}+x^{11}+x^{16}$                  |
| $g_4(x)$    | $1+x^2+x^4+x^6+x^9+x^{11}+x^{12}+x^{14}+x^{16}$                       |
| $g_5(x)$    | $1+x+x^2+x^3+x^5+x^8+x^9+x^{10}+x^{11}+x^{12}+x^{16}$                 |
| $g_6(x)$    | $1+x^2+x^4+x^5+x^7+x^8+x^9+x^{10}+x^{12}+x^{13}+x^{14}+x^{15}+x^{16}$ |
| $g_7(x)$    | $1+x^2+x^5+x^6+x^8+x^9+x^{10}+x^{11}+x^{13}+x^{15}+x^{16}$            |
| $g_8(x)$    | $1+x+x^2+x^5+x^6+x^8+x^9+x^{12}+x^{13}+x^{14}+x^{16}$                 |
| $g_9(x)$    | $1+x^5+x^7+x^9+x^{10}+x^{11}+x^{16}$                                  |
| $g_{10}(x)$ | $1+x+x^2+x^5+x^7+x^8+x^{10}+x^{12}+x^{13}+x^{14}+x^{16}$              |
| $g_{11}(x)$ | $1+x^2+x^3+x^5+x^9+x^{11}+x^{12}+x^{13}+x^{16}$                       |
| $g_{12}(x)$ | $1+x+x^5+x^6+x^7+x^9+x^{11}+x^{12}+x^{16}$                            |

| LDPC<br>Code<br>identifier | BCH Uncoded<br>Block $K_{bch}$ | BCH coded block $N_{bch}$<br>LDPC Uncoded Block $k_{ldpc}$ | BCH<br>t-error<br>correction | Effective<br>LDPC Rate<br>$k_{ldpc}/16\ 200$ | LDPC Coded<br>Block<br>$n_{ldpc}$ |
|----------------------------|--------------------------------|--|------------------------------|--|-----------------------------------|
| 1/4                        | 3 072                          | 3 240  | 12                           | 1/5  | 16 200                            |
| 1/3                        | 5 232                          | 5 400  | 12                           | 1/3  | 16 200                            |
| 2/5                        | 6 312                          | 6 480  | 12                           | 2/5  | 16 200                            |
| 1/2                        | 7 032                          | 7 200  | 12                           | 4/9  | 16 200                            |
| 3/5                        | 9 552                          | 9 720  | 12                           | 3/5  | 16 200                            |
| 2/3                        | 10 632                         | 10 800   | 12                           | 2/3  | 16 200                            |
| 3/4                        | 11 712                         | 11 880   | 12                           | 11/15  | 16 200                            |
| 4/5                        | 12 432                         | 12 600   | 12                           | 7/9  | 16 200                            |
| 5/6                        | 13 152                         | 13 320   | 12                           | 37/45  | 16 200                            |
| 8/9                        | 14 232                         | 14 400   | 12                           | 8/9  | 16 200                            |
| 9/10                       | NA                             | NA   | NA                           | NA   | NA                                |

|             |  |
|-------------|--|
| $g_1(x)$    | $1+x+x^3+x^5+x^{14}$                               |
| $g_2(x)$    | $1+x^6+x^8+x^{11}+x^{14}$                          |
| $g_3(x)$    | $1+x+x^2+x^6+x^9+x^{10}+x^{14}$                    |
| $g_4(x)$    | $1+x^4+x^7+x^8+x^{10}+x^{12}+x^{14}$               |
| $g_5(x)$    | $1+x^2+x^4+x^6+x^8+x^9+x^{11}+x^{13}+x^{14}$       |
| $g_6(x)$    | $1+x^3+x^7+x^8+x^9+x^{13}+x^{14}$                  |
| $g_7(x)$    | $1+x^2+x^5+x^6+x^7+x^{10}+x^{11}+x^{13}+x^{14}$    |
| $g_8(x)$    | $1+x^5+x^8+x^9+x^{10}+x^{11}+x^{14}$               |
| $g_9(x)$    | $1+x+x^2+x^3+x^9+x^{10}+x^{14}$                    |
| $g_{10}(x)$ | $1+x^3+x^6+x^9+x^{11}+x^{12}+x^{14}$               |
| $g_{11}(x)$ | $1+x^4+x^{11}+x^{12}+x^{14}$                       |
| $g_{12}(x)$ | $1+x+x^2+x^3+x^5+x^6+x^7+x^8+x^{10}+x^{13}+x^{14}$ |

## BIBLIOGRAFÍA

- [Satellite 101 - INTELSAT](#)
- [Modulation Schemes for Satellite Communications – Keysight Technologies](#)
- [YouTube – Leslie Rusch](#)
- [YouTube - Computerphile](#)