



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA

Memoria

*Diseño, desarrollo y validación de un
Gateway-T2 y un analizador T2-MI*

Autor: Carlos Barjau Estevan

Director: David Gómez Barquero

Tutor: Narcís Cardona Marcet

ÍNDICE

Introducción	3
1.1. Introducción	3
1.2. Motivación y objetivos	7
2. Tecnologías	10
2.1. DVB-T2	10
2.2. Gateway-T2	14
2.3. Interfaz T2-MI	26
3. Diseño del Gateway-T2 y del Analizador T2-MI	35
3.1. Gateway-T2	35
3.2. Analizador T2-MI	44
4. Desarrollo del Gateway-T2 y del Analizador T2-MI.....	47
4.1. Gateway-T2	47
4.2. Analizador T2-MI	55
5. Validación del Gateway-T2 y del Analizador T2-MI.....	63
5.1. Gateway-T2	63
1.2. Analizador T2-MI	66
6. Conclusiones y líneas nuevas.....	67
6.1. Conclusiones	67
6.2. Líneas nuevas.....	68
7. Aplicaciones futuras.....	76
7.1. Radioenlaces OFDM profesionales para servicios BAS/SAP	76
7.2. Canal de retorno terrestre inalámbrico de DVB-T2, DVB-RCT2	78
8. Bibliografía.....	80
ANEXO: Manual de uso del Gateway y del Analizador.	81

Introducción

1.1. Introducción

DVB-T2 [1] es un estándar desarrollado por la organización DVB (Digital Video Broadcasting), cuya motivación es sustituir, eventualmente, todos los sistemas DVB-T ya operativos en el planeta, con los últimos avances, tanto a nivel teórico como a avance tecnológico de circuitería. Los primeros estudios para la confección de la especificación comenzaron en 2006, y la primera versión fue aprobada por la ETSI (European Telecommunications Standard Institute) en 2009. Actualmente, la última versión de la especificación es la 1.3.1 y su fecha de publicación data de Abril de 2012.

La motivación principal de DVB-T2 es conseguir una mayor eficiencia en el uso del canal radio, aplicando los últimos avances en tratamiento de la señal y en velocidad de procesamiento de los receptores actuales. Esta necesidad surge por dos razones, principalmente; la escasez del espectro radioeléctrico destinado a difusión de vídeo y la mayor demanda de éste para los nuevos servicios de difusión (HDTV, 3DTV...) En concreto, los objetivos fijados por la DVB para este estándar son:

- Mayor tasa binaria y mayor robustez (menor C/N requerida) que en el anterior estándar, DVB-T, mediante códigos de protección frente a errores más eficientes y un entrelazado mayor y variable.
- Mayor flexibilidad a la hora de asignar recursos y parámetros para servicios (TV móvil, 3DTV, satélite...) en un mismo sistema.
- Reutilización de equipos receptores: es posible encapsular una señal de transporte MPEG-TS usando T2 para luego ser recibida con la misma antena utilizada en DVB-T, aunque precisaremos de un receptor DVB-T2 para demodular la señal.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Al igual que su predecesor, la modulación utilizada en DVB-T2 es una COFDM (Coded Orthogonal Frequency-Division Multiplexing), es decir, una OFDM junto una codificación de corrección/detección de errores, donde cada subportadora se encuentra modulada también con una QAM. Al símbolo se le añade un intervalo de guarda cuyo objetivo es paliar el efecto multicamino, especialmente crítico en redes de frecuencia única o SFN (Single Frequency Network).

La tabla siguiente compara las características principales de DVB-T [2] con las características de DVB-T2, donde se ha sombreado las nuevas funcionalidades de T2:

	DVB-T	DVB-T2
<i>Constelación</i>	QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM, 256QAM
<i>Forward Error Correction(FEC)</i>	Reed-Solomon + Convolutional. 1/2, 2/3, 3/5, 5/6, 7/8	LDPC + BCH. 1/2, 3/5 , 2/3, 3/4, 4/5 , 5/6
<i>Intervalo de Guarda</i>	1/4, 1/8, 1/16, 1/32	1/4, 19/256 , 1/8, 19/128 , 1/16, 1/32, 1/256
<i>Tamaño FFT</i>	2K, 8K	1K , 2K, 8K, 16K , 32K

Tabla 1: Comparativa entre DVB-T y DVB-T2.

De estos cambios, cabe resaltar que ya no usan los códigos Reed-Solomon + turbocódigos para proteger la información; en su lugar se usan códigos LDPC (Low Density Parity Coding) + BCH (Bose-Chaudhuri-Hocquenghem coding) de mejores prestaciones, y que no requieren de algoritmos privativos para su decodificación. Estos códigos ya se conocen desde 1960, pero el procesamiento necesario para codificar y decodificar los hicieron inviables computacionalmente hasta ahora.

La ganancia que experimentamos por el uso de estos códigos nos permite introducir una modulación de orden superior, 256QAM, cuyos símbolos transportan 8 bits cada uno. Por otro lado, también hemos ampliado el tamaño de la FFT, hasta 32K, lo que permite un uso más eficiente del espectro. Este cambio, junto a la reducción del intervalo de guarda de los símbolos OFDM nos otorga una ganancia en capacidad respecto a DVB-T.

DVB-T2 también profundiza en la gestión de las redes SFN, que ya fue considerado en DVB-T. Una red SFN es una red de transmisores que emiten, de forma sincronizada, la misma información sobre la misma frecuencia. Una red SFN tiene tres requisitos principales [3]:

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

- Que los transmisores emitan en la misma frecuencia.
- Que los transmisores radien la señal simultáneamente.
- Que los transmisores envíen la misma información (los mismos símbolos OFDM).

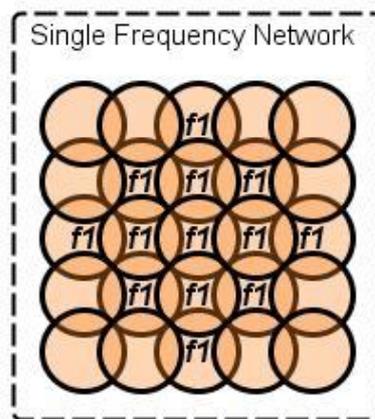


Ilustración 1: Plan de frecuencias de una red SFN

Que todos los transmisores de una red estén emitiendo la misma información, en el mismo instante de tiempo y en la misma frecuencia supone un caso extremo de multicamino en recepción.

El nuevo estándar recoge todos los métodos y funciones ya existentes en DVB-T que actúan sobre la SFN, y añade nuevos. Esto se detallará más en el apartado sobre la interfaz T2-MI.

No sólo se han ampliado las características existentes, sino que DVB-T2 introduce nuevas técnicas de procesamiento para solucionar problemas específicos:

- PLP (Physical Layer Pipe): Un canal lógico con una modulación y codificación determinada, que puede transportar uno o varios servicios. Se pueden tener distintos PLPs, cada uno con su parametrización, según el tipo de servicio que transporten.

- Constelación rotada: En DVB-T2, la constelación de símbolos posibles se rota un cierto ángulo, lo cual modifica el valor I/Q con el que se tiene que modular cada símbolo distinto de los 2^n posibles (siendo n el orden de la modulación).

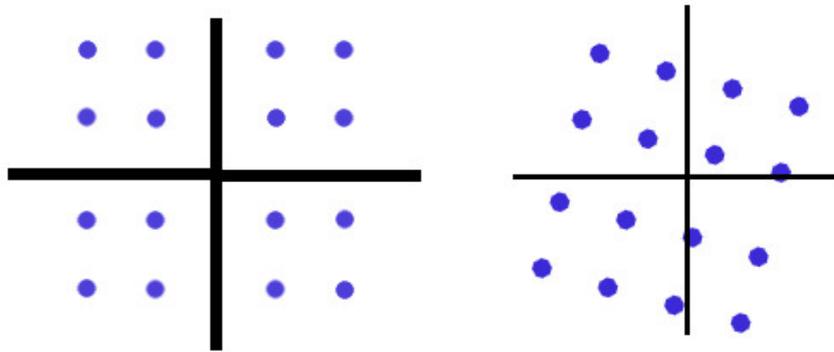


Ilustración 2 : Ejemplo de constelación rotada (16QAM).

- Técnicas de PAPR (Peak to Average Power Reduction). Uno de los problemas de la modulación OFDM son los picos que pueden surgir en la señal resultante, suma de todas las subportadoras. Es un problema para los amplificadores, porque corremos el riesgo de entrar en zona no-lineal (saturación), con la distorsión asociada. Para reducir las exigencias de los amplificadores, DVB-T2 introduce dos técnicas de reducción de potencia de pico: Tone Reservation y Active Constellation Extension.

El estándar está en uso o en proceso de ser implementado en 19 países distintos, en ocasiones compartiendo espectro con DVB-T. Un caso común es el uso de T2 para los multiplex con contenidos en HD, mientras que para aquellos que aún utilizan contenidos en SD se transmiten mediante T. Hay excepciones, como por ejemplo en Serbia que toda la difusión de vídeo se realiza mediante T2.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

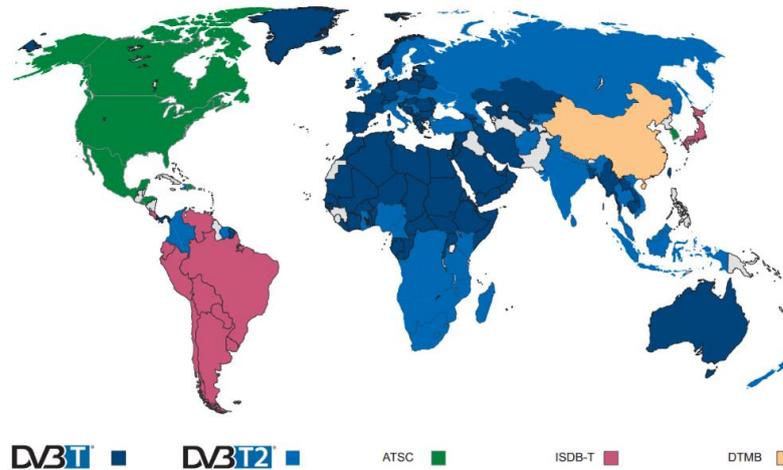


Ilustración 3: Despliegue actual de tecnologías de difusión de vídeo en el mundo [www.dvb.org]

1.2. Motivación y objetivos

a) Motivación

Para poder lograr toda la mejora en prestaciones, DVB-T2 introduce importantes cambios estructurales respecto a la arquitectura original DVB-T.

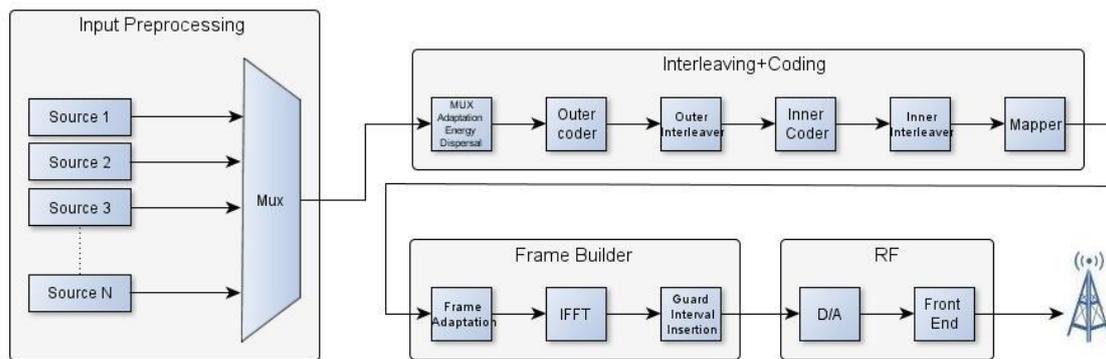


Ilustración 4: Arquitectura básica de un sistema DVB-T

El mayor cambio introducido es la adición de un Gateway entre la etapa de Input Preprocessing e Interleaving+Coding.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

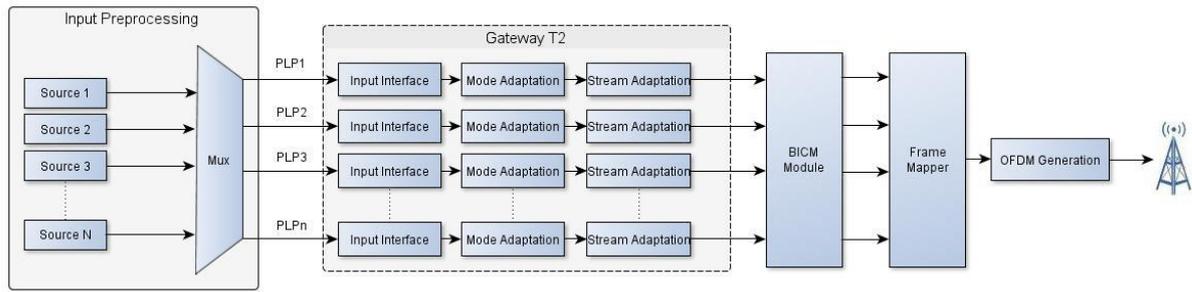


Ilustración 5: Arquitectura básica de un sistema DVB-T2

La función básica del Gateway es pasar de flujos TS entrantes a PLP de datos y a PLP comunes, y hacer un primer análisis de la información para convertirla a BBFrames, generar la señalización y sincronización correspondiente, toda la gestión de SFN. Todo lo anterior será posteriormente encapsulado en una secuencia de paquetes T2-MI que luego serán llevados a la red de distribución para alimentar a los moduladores T2. En la práctica, el Gateway generará un flujo TS a partir del T2-MI que luego podrá ser enviado mediante una interfaz ASI existente o mediante IP.

En el caso de tener múltiples PLP, el Gateway será el encargado de hacer el Scheduling y asignación de recursos dentro de las T2-Frame, es decir, el Gateway otorgará la inteligencia al sistema de transmisión, de ahí su importancia.

Aunque ya existen numerosos fabricantes que diseñan, manufacturan y venden Gateways-T2 implementados sobre FPGAs, estos sistemas no tienen la flexibilidad, transparencia, capacidad de extensión o interoperabilidad que un desarrollo propio en software ofrece para el ámbito investigador, o su coste económico es prohibitivo. Esta es la motivación principal para el diseño, desarrollo y validación de un Gateway-T2 a nivel software. Esta fue la filosofía seguida dentro del MCG (Mobile Communications Group), un grupo dentro del iTEAM (Instituto de Telecomunicaciones y Aplicaciones Multimedia). El grupo ya disponía de plataformas de encapsulamiento software de diversos estándares de difusión (DVB-T y DVB-H), y se comenzó una línea de investigación para el desarrollo de un Gateway-T2.

Como se ha dicho anteriormente, la salida del Gateway es un archivo T2-MI listo para ser distribuido. Para asegurar el buen funcionamiento del Gateway, existen varias posibilidades. Una de ellas consiste en alimentar un modulador T2 operativo y

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

posteriormente, con un receptor T2, asegurar la correcta demodulación de la señal. El principal problema de esta opción es la dificultad de implementar un demodulador en tiempo real a nivel software, o el elevado coste de los demoduladores profesionales. La otra opción posible a la hora de validar un Gateway-T2 es usar un Analizador T2-MI.

Un Analizador T2-MI interpreta los archivos T2-MI generados por un Gateway, pero no llega al proceso de codificar ni modular la señal, solo comprueba la coherencia del flujo, extrae las cabeceras y muestra la señalización asociada a la transmisión. Su implementación es más simple que la de un simulador de un sistema transmisor completo, por lo que a la hora de validar el Gateway, se ha escogido esta opción para ser desarrollada. Además, el Analizador abre una línea de investigación nueva sobre aplicaciones novedosas, tales como la implementación de un “Gateway Inverso”.

b) Objetivos

Los objetivos principales de este Proyecto Final de Carrera son:

- Diseño, desarrollo y validación software de un Gateway-T2 (plataforma de encapsulación de DVB-T2), que cumpla con lo mostrado en el estándar EN 302 755 v.1.3.1.
- Diseño, desarrollo y validación software de un Analizador T2-MI, que cumpla con lo expuesto en el documento ETSI TS 102 773.

2. Tecnologías

2.1.DVB-T2

DVB-T2 está pensado para adaptarse a la gran variedad de escenarios y demanda de servicios que han ido surgiendo en los últimos años. Para ello, introduce varios conceptos nuevos, cuyo conocimiento es fundamental para entender su funcionamiento:

El primer concepto a explicar son los PLP o Physical Layer Pipe. Un PLP es un canal virtual a nivel de capa física, con una modulación y un Code-Rate establecidos; configurables en tiempo real, en el que podemos agrupar servicios, no solo de contenido audiovisual (podemos tener un PLP dedicado al envío de las tablas de programación).

El funcionamiento de un sistema DVB-T2 cambia drásticamente de tener un solo PLP (Single PLP o modo A) a tener múltiples (Multiple PLP o modo B). Esto debe las necesidades de sincronización y dimensionamiento de las tramas T2.

El motivo de tener estas tuberías es que en T2 se pueden hasta 256 PLP distintos, cada uno de ellos con sus respectivas modulaciones y Code-Rates. A un grupo de servicios que no demande mucho ancho de espectro pero si robustez se le puede asignar un PLP de bajo orden de modulación y Code-Rate elevado para garantizar su recepción; y al contrario, varios servicios que demande mucho espectro, se le dará la modulación y Code-Rate más eficiente.

Como se ha mencionado anteriormente, los parámetros de cada PLP son configurables en tiempo real, e incluso es posible añadir y quitar PLP según lo consideremos oportuno. Es necesario, para poder soportar tanta flexibilidad, tener una fuerte carga de señalización. En el caso de DVB-T2, esta señalización se le denomina L1-Signalling, y será transportada al principio (o en el mismo PLP en caso de usar L1-Inband) de cada trama T2. Al igual que los PLP, la señalización está modulada en OFDM y cada subportadora en M-QAM.

Hay 3 tipos distintos de señalización L1:

- L1-Actual. Su uso es mandatorio. Tiene toda la información sobre el sistema, la configuración y los parámetros, tanto variables como constantes. Además contiene la información dinámica de cada PLP para la T2-Frame actual. Dentro de la señalización, hay 4 tipos distintos:
 - Preámbulo P1, siempre modulado en BPSK, Code Rate 1/2 y 2K de tamaño de FFT, con información sobre los parámetros de la modulación OFDM (Guard Interval y Tamaño) y el tipo de trama enviado.
 - L1-Presignalling, modulado en BSPK, Code Rate 1/2, con la FFT definida para el resto del sistema. Contiene información sobre los parámetros del L1-Configurable y Dynamic, así como información del sistema y parámetros de la trama. Estos valores permanecen constantes y no se pueden modificar.
 - L1-Configurable, con la modulación y el Code Rate indicados en el Presignalling. Contiene información sobre las portadoras y sobre los PLP, tales como la cantidad, el tipo, la codificación y la modulación, la cantidad máxima de bloques FEC por Interleaving, el uso de Inband... Estos parámetros pueden variar, pero solo entre Superframes.
 - L1-Dynamic, como se ha mencionado anteriormente, contiene información sobre el número de trama, la cantidad de bloques FEC de cada PLP en una T2-Frame, además de indicar el comienzo dentro de la trama de un PLP. Estos valores pueden cambiar de T2-Frame a T2-Frame, no se restringen los cambios a Superframes.
 - L1-Extension, extensiones futuras aun no especificadas por la DVB, pero que se pueden usar para enviar señalización de índole propietaria.

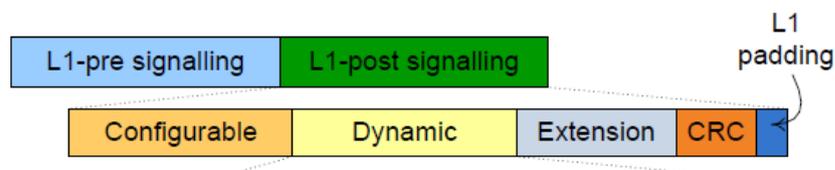


Ilustración 6: Esquema de la señalización L1-Actual [DVB especificación]

- L1-Future o Repetition. Su uso es opcional. Es la adición del L1-Dynamic de la T2-Frame $m+1$ a continuación del L1-Dynamic de la T2-Frame m . El objetivo es facilitar al receptor la tarea de adaptarse a posibles cambios en

las T2-Frames venideras. Implica tener un buffer con, al menos, dos T2-Frames, para poder analizar y sintetizar el L1-Future.

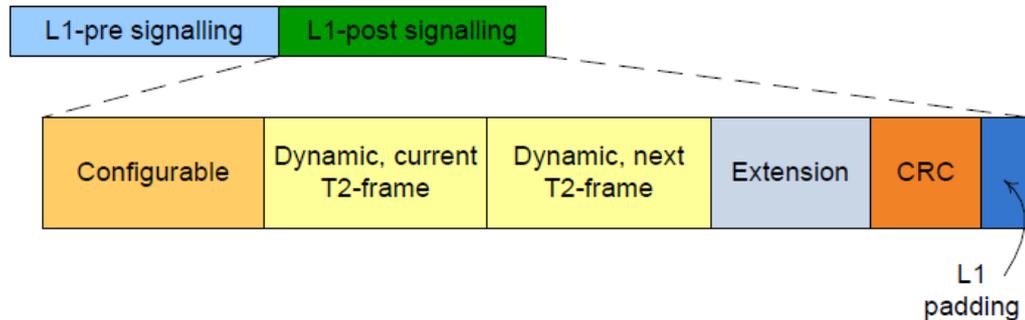


Ilustración 7: Esquema de la señalización L1 con L1-Future [DVB Especificación]

- L1-Inband. Su uso es opcional. Esta señalización no es transportada al principio de la trama como las anteriores, sino que va introducida al final de la primera BBFrame de una Interleaving Frame para un PLP, dentro del Padding. El objetivo es permitir una rápida sintonización entre PLPs, sin tener que esperar a la señalización L1 que viaja al principio de la trama. Existen dos tipos distintos de Inband:
 - Inband A. Lleva información sobre la configuración de los PLP, su comienzo dentro de la siguiente T2-Frame, si hay cambios en la configuración del PLP previstos o la cantidad de bloques transportados en la Interleaving Frame.
 - Inband B. Solo se puede usar si el flujo encapsulado es TS o GFPS. Lleva información de sincronización temporal, para facilitar una rápida adquisición inicial del PLP deseado. Es decir, lleva información del TTO, BUFS e ISCR, además de la tasa binaria del flujo.

Tanto la señalización como los datos de los PLPs van contenidos en las T2-Frames (Tramas T2), compartiendo el mismo recurso radio. La duración de estas T2-Frames es configurable, y depende, entre otras cosas, de la cantidad de símbolos OFDM transportados, el ancho de banda, el tamaño de la FFT y el intervalo de guarda.

Estas T2-Frames van agrupadas en un conjunto mayor llamado Superframes o Supertramas, el motivo es que un cambio en la configuración de un PLP o de la

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

señalización L1 solo puede ocurrir entre Superframes y no entre T2-Frames, para notificar al receptor con la correspondiente antelación. La especificación también permite el uso de FEF (Future Extension Frames), ubicadas al final de una trama T2, para usos privativos y aún no especificados en la norma.

Conviene aclarar que la norma detalla otro término muy relacionado con las Tramas T2, las Interleaving Frame o Tramas de Entrelazado. En la configuración de un PLP, podemos elegir entre dos tipos distintos de entrelazado, el primero de ellos consiste en entrelazar temporalmente los datos de un PLP en una sola T2-Frame, entonces la Interleaving Frame y la T2-Frame coinciden, sin embargo, en el segundo método, el PLP se encuentra entrelazado en n T2-Frames, por lo que la Interleaving Frame contiene n T2-Frames.

Siguiendo en la línea del encapsulado, y como ya se ha mencionado anteriormente, una T2-Frame está formada por la señalización L1, los flujos auxiliares, en el caso que los necesitemos y los BBFrame (Tramas Banda Base) de los PLP.

Un BBFrame es una trama, de longitud fija (K_{BCH}) bits (64800 o 16400), necesarios para el bloque BCH. Esta trama contiene ya datos de un flujo elemental (o varios) para un PLP. Se añade al principio una cabecera (BBHeader) que detalla, entre otras cosas, el tipo de información que contiene el PLP (Transport Stream, GSE...), si se usa codificación constante o variable, la ID del PLP al que pertenece el BBFrame...

Este sistema jerárquico de encapsulamiento es bastante intuitivo a la hora de ser programado: un BBFrame es una agrupación de bits + la cabecera; un T2-Frame es una agrupación de BBFrames + la señalización L1; una Interleaving Frame es una agrupación (o una T2-Frame solo) de T2-Frames + el L1-Inband (si se utiliza).

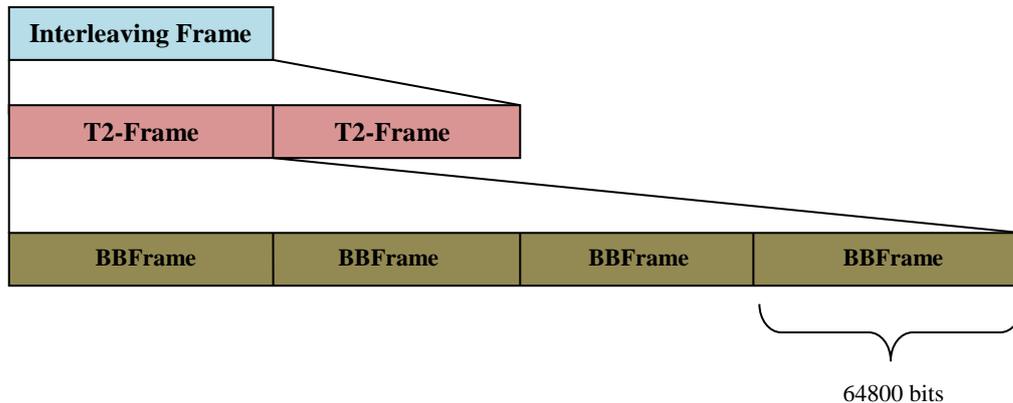


Ilustración 8: Ejemplo de un esquema general de la jerarquía de encapsulación en DVB-T2 para un PLP.

2.2. Gateway-T2

Centrándonos ya en la parte técnica de un Gateway-T2, la finalidad de este subsistema es la generación de la señal común de DVB-T2 y la sincronización que luego será alimentada a la red de transmisores o al transmisor individual. Esto incluye todas las operaciones a nivel de capa física, tales como la adecuación de los flujos de entrada, el scheduling, la asignación de los PLP a las tramas T2, la generación de la señalización L1...

La especificación de DVB-T2 indica dos modos distintos para la salida de un Gateway: Una salida acorde a la interfaz T2-MI [4], diseñada para alimentar a una red de transmisores remotamente, o una salida acorde a la interfaz BICM (Bit-Interleaved Coded Modulation), diseñada como entrada directamente a un modulador. Se muestra primero el sistema con el uso de la interfaz BICM y a continuación el diagrama de bloques simplificado cuando se usa el interfaz T2-MI.

Hoy en día, a nivel comercial la interfaz BICM no se usa a no ser que se tenga un transmisor DVB-T2 integral (Gateway+Codificador+Frame builder). Lo más común es tener un único Gateway que haga de cabecera y varios moduladores ubicados remotamente, gestionados mediante la interfaz T2-MI.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

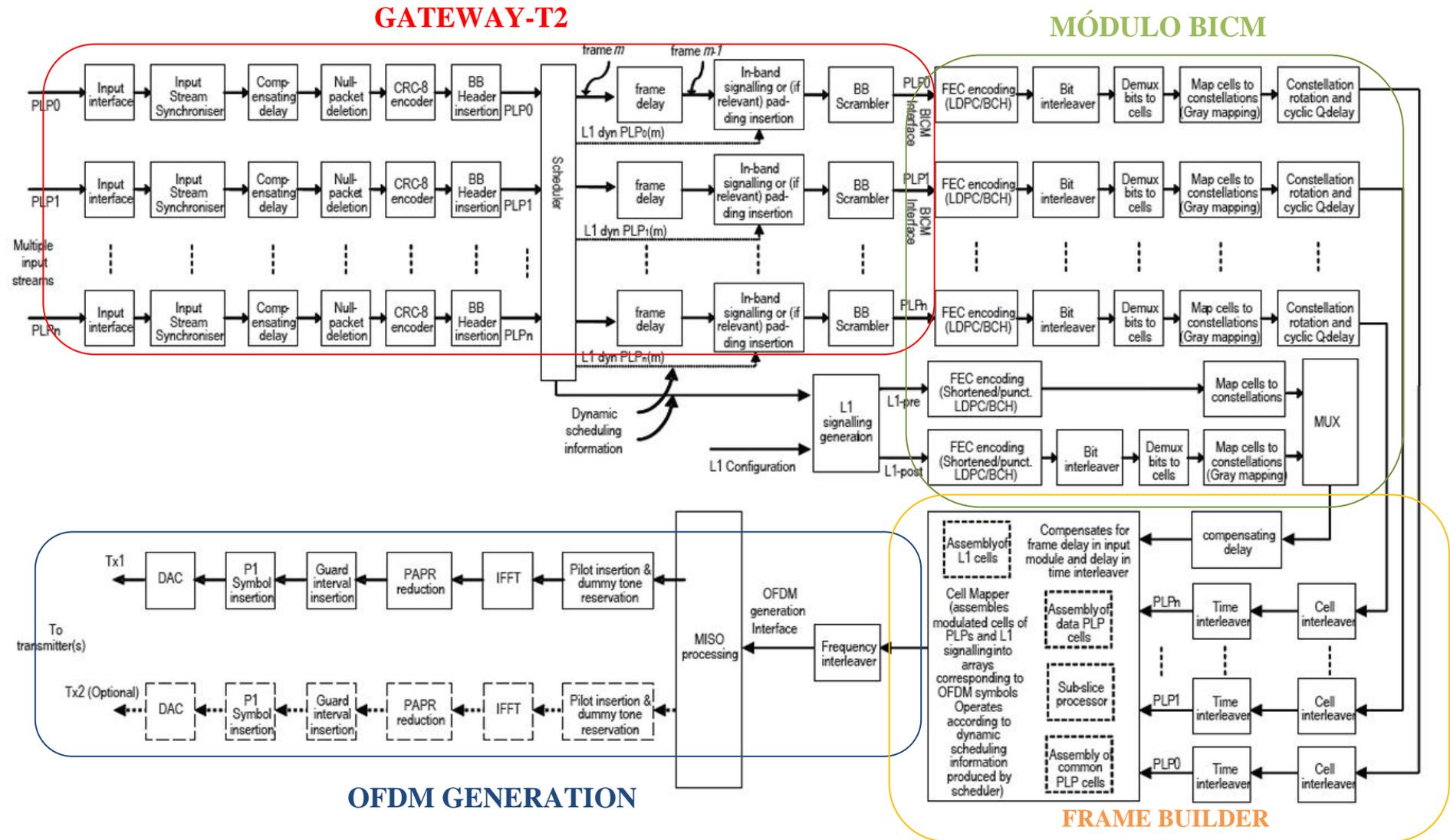


Ilustración 9: Esquema general de un Transmisor DVB-T2 con múltiples PLP [www.rohde-schwarz.es]

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

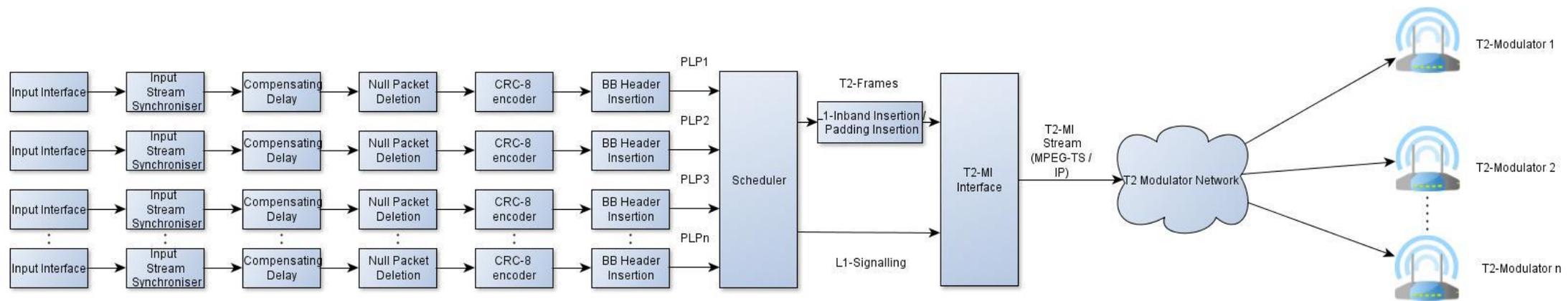


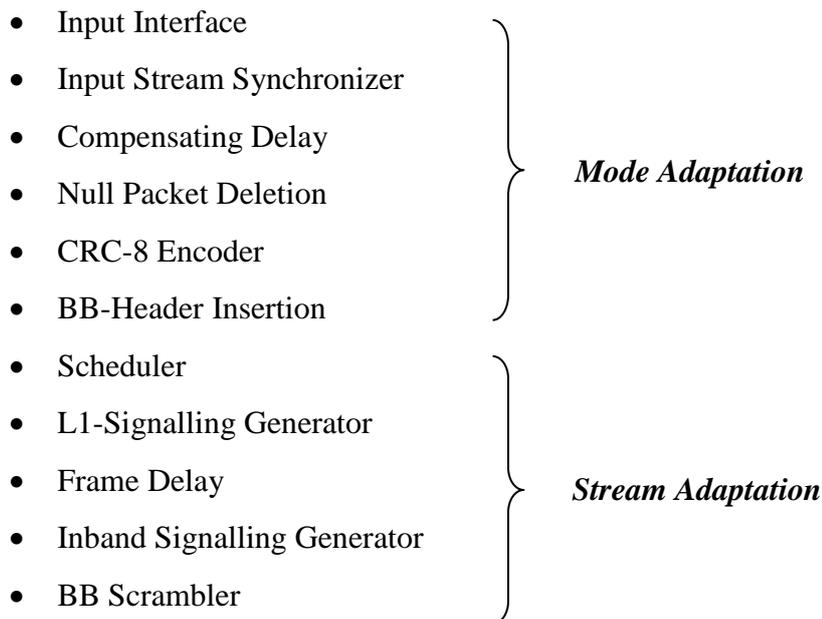
Ilustración 10: Esquema simplificado de un Gateway-T2 con salida T2-MI

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

La entrada del Gateway son uno o varios flujos MPEG-TS, cada uno transportando contenido audiovisual. No solo está limitado a este formato, sino que también permite otros distintos, tales como GSE (Generic Stream Encapsulation) o GFPS (Generic Fixed-length Packetized Stream).

En concreto, un Gateway tiene las siguientes etapas si se usa la interfaz BICM:



Sin embargo, en el uso de una interfaz T2-MI, obviamos el BB Scrambler, el Scheduler funciona de otro modo y añadimos un módulo “T2MI-Interface” que adecue la salida de éste para la interfaz. Se puede observar en el esquema anterior que, cada PLP tendrá su cadena de módulos, es decir, un PLP tendrá **Input Interface, Input Stream Synchroniser, Compensating Delay, Null Packet Deletion, CRC-8 Encoder, BB-Header Insertion y Inband Signalling Generator.**

Hay dos partes diferenciadas del Gateway, antes del Scheduling y después. Las etapas anteriores al Scheduler son parte del **Mode Adaptation**, mientras que la parte posterior y el propio Scheduler forman parte del **Stream Adaptation**.

Podemos escoger entre dos modos de funcionamiento distinto para un PLP: NM (Normal) y HEM (Alta Eficiencia), donde se transporta más información en la cabecera de las BBFrame, lo cual repercute positivamente en el Throughput. Elegir un modo u otro definirá el comportamiento de las etapas del **Mode Adaptation**.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

El **Stream Adaptation** asigna los PLPs a las T2-Frames (Scheduling), genera la señalización necesaria (L1-Signalling) y añade el relleno necesario a las BBFrame (padding), que luego se reemplazará con una parte de señalización (L1-Inband).

Dentro de los módulos del **Mode Adaptation**, el primero de la lista, **Input Interface**, hace una primera conversión entre un flujo de entrada y el formato lógico de bit del subsistema, para poder componer las BBFrame. De un flujo constante de bits, esta etapa lo fragmenta en bloques de DFL (Data Field Length) bits. Este valor, como máximo, es de $K_{BCH} - 80$ bits, siendo $K_{BCH} = 64800$ bits, el tamaño necesario para la codificación LDPC/BCH, además se sustrae 80 bits por la cabecera de la BBFrame. Todas las BBFrame deben de tener K_{BCH} bits, y en el caso de no tener bits suficientes, habrá que añadir un Padding o relleno, para completar la BBFrame.

En el caso de que utilicemos L1-Inband, tendremos menos capacidad en la BBFrame para datos útiles, puesto que el Inband va ubicado al final de una BBFrame. Es bastante probable que en esta primera paquetización, el flujo original quede fragmentado, sobre todo si el flujo original era un MPEG-TS (paquetes de 188 Bytes), esto lo indicaremos más adelante, en un campo del BBHeader.

El siguiente bloque es el **Input Stream Synchroniser**. Esta etapa se encarga de corregir las variaciones temporales que pueden ocurrir en el procesamiento, codificación del vídeo... para garantizar una salida de tasa binaria y retardo extremo a extremo constante, y además, ayudar en la gestión de la memoria del receptor, para deshacer el entrelazado y corregir el jitter. El uso de este bloque es obligado si la entrada es un flujo TS, pero bajo ciertas circunstancias muy determinadas se puede omitir:

- Solo hay un PLP.
- Data Field Length = $K_{BCH} - 80$ en todas las BBFrame.
- En todas las Interleaving Frame, tenemos el número máximo de bloques FEC (PLP_NUMBLOCKS=PLP_NUMBLOCKS_MAX).
- No hay Null Packet Deletion.
- No hay FEFs.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Si se cumplen todas las condiciones, no es necesario añadir la sincronización temporal para un PLP. En el caso de cualquiera no se cumpla, es necesario el Input Stream Synchronizer para garantizar que el PLP se radia en el instante adecuado. Este módulo añade un campo denominado ISSY (Input Stream SYNchroniser). Este atributo puede tener 2 o 3 bytes de tamaño, y se añade al final de cada User Packet (paquetes originales, después de ser extraídos por el Input Interface), con 3 valores posibles:

- ISCR (Input Stream Clock Reference), que transporta los bits menos significativos de un contador de módulo 2^{22} , y período T, que depende del ancho de banda utilizado:

Ancho de banda	1.7 MHz	5 MHz	6 MHz	7 MHz	8 MHz	10 MHz
Período T	71/131 μ s	7/40 μ s	7/48 μ s	1/8 μ s	7/64 μ s	7/80 μ s

Tabla 2: Valores del período elemental T, en función del ancho de banda.

Este campo se añade en el tercer ISSYs, y tiene de longitud 15 o 22 bits, en función de si el ISSY es largo (3 bytes) o corto (2 bytes).

- BUFS indica el tamaño máximo del buffer del receptor para corregir variaciones en el retraso. Su tamaño es de 2+10 bits, donde los dos primeros bits indican las unidades de la memoria y los 10 siguientes el valor. Esta variable se añade en el segundo ISSYs.
- TTO (Time To Output), tiene 7 o 15 bits de mantisa y 5 de exponente. Su finalidad es aportar un mecanismo para corregir el Jitter que pueda experimentar la BBFrame. TTO define el tiempo en unidades de T desde que el primer símbolo P1 de la primera T2-Frame de una Interleaving Frame hasta que el primer User Packet del correspondiente PLP se encuentra ubicado. Se envía en el primer ISSYs.

El siguiente bloque de la cadena es el **Compensating Delay**. La finalidad que tiene es añadir un Delay a los PLP de un grupo (PLPs que comparten un PLP común para la información de PSI/SI) pero que tengan distintas configuraciones (saltos entre tramas, distintos entrelazados temporales...), para que cuando sean recibidos, no haya

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

una diferencia mayor de 1ms entre los ISCR de cada PLP de un grupo al ser decodificados, es decir, lleguen conjuntamente. Si solo tenemos un PLP, no es necesario añadir un Delay, pero conviene hacerlo para evitar el underflow (buffers vacíos).

A continuación, viene un módulo opcional llamado **Null Packet Deletion**. Este bloque solo se utiliza si la entrada es un flujo MPEG-TS. Un Transport Stream necesita una tasa binaria de salida del multiplexor del transmisor y del demultiplexor del receptor constante. Para lograr esto, es posible que introduzcamos una gran cantidad de paquetes TS Null, con $PID = 8191_D$. El objetivo de esta etapa es quitar los paquetes Null de un flujo TS, para evitar transmisiones innecesarias, mejorando así el throughput.

El método que se sigue es el siguiente: Después de cada paquete TS Null, se borra y se incrementa un contador; en cuanto un paquete con información útil llega, se añade al final de un UP un campo llamado NPD de tamaño 1 byte, con el valor del contador, y luego se reinicializa. El receptor puede luego añadir, a partir de este campo, todos los paquetes Null borrados, asegurando la tasa binaria.

El siguiente bloque del Mode Adaptation es el **CRC-8 Encoder**. Para GFPS y TS, y solo en modo normal, los bits del User Packet, sin el byte de sincronización, son procesados por el Encoder y el CRC-8 calculado se añade al final del User Packet. El objetivo es asegurar que los datos de un User Packet no han sido dañados por el canal o por el procesado. En el modo HEM este bloque no se implementa, para mejorar el throughput.

El último módulo del **Mode Adaptation** es el **BBHeader Insertion**. En esta etapa, añadimos 80 bits, para describir el contenido de la BBFrame. La sintaxis de la cabecera cambia si el PLP usa el modo normal (NM) o el modo de alta eficiencia (HEM). En las ilustraciones siguientes se muestran los campos, según el modo:

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Matype	UPL	DFL	SYNC	SYNCD	CRC-8 MODE
Matype	ISSY 2MSB	DFL	ISSY 1LSB	SYNCD	CRC-8 MODE

Tabla 3: Campos del BBHeader, para Normal Mode (arriba) y HEM, para MPEG-TS.

- Matype, contiene información sobre el flujo encapsulado en la BBFrame. En concreto, Matype tiene información sobre el tipo de flujo, la codificación, si usamos Null Packet Deletion y/o Input Stream Synchronization y si tenemos varios PLP, a cuál de ellos pertenece esta BBFrame.
- UPL (User Packet Length) define la longitud de los User Packet encapsulados. Si nuestra entrada es MPEG-TS, UPL valdrá $188 * 8_D$ bits, el tamaño de un paquete TS.
- DFL (Data Field Length) define la longitud del Data Field, la cantidad de datos útiles de una BBFrame antes de ser codificada por el LDPC.
- SYNC indica cual es el byte de sincronización, que indica el comienzo de un User Packet, para TS, el valor es 47_{16} .
- SYNCD es un puntero al comienzo de un User Packet fragmentado entre BBFrames.
- CRC-8 MODE es una XOR entre el CRC-8 de los primeros 9 bytes del BBHeader con MODE, y MODE vale 0_D para Normal Mode (NM) y 1_D para HEM.

En el caso de usar Alta Eficiencia, incluimos el ISSY dentro del BBHeader, sustituyendo el UPL y el SYNC, que ambos son valores conocidos para MPEG-TS.

Por otra parte, el primer bloque del **Stream Adaptation** es donde reside toda la inteligencia y complejidad del Gateway: el **Scheduler**. La misión del Scheduler es asignar la posición de las BBFrames de cada PLP dentro de sus respectivas Interleaving Frames. La especificación no detalla la implementación de este subsistema, pero hay que cumplir ciertos requerimientos:

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

- El número de celdas OFDM asignadas a PLP no debe exceder la capacidad máxima de la trama.
- El receptor debe de ser capaz de localizar y extraer los PLPs utilizando la señalización L1.
- El buffer corrector de Jitter no debe sufrir de underflow, y el buffer del desentrelazado temporal no debe sufrir de overflow.

El problema es trivial para un solo PLP puesto que la cantidad de bloques FEC (bloques de 64800 bits) es siempre constante y entero. Sin embargo, para múltiples PLP, surgen complicaciones, puesto que cada PLP puede tener su modulación y su Interleaving Frame, y es posible que no llenemos enteramente una trama T2.

Incluso con varios PLP con la misma configuración, puede ocurrir que la cantidad de bloques FEC media de una Interleaving no sea un entero, mientras que la cantidad de bloques FEC estrictamente sí que tiene que serlo. Esto implica que, T2-Frame a T2-Frame, la cantidad de bloques FEC para cada PLP es variable. En función del algoritmo de asignación de celdas OFDM, obtendremos una mayor eficiencia del sistema, a costa de la complejidad del mismo.

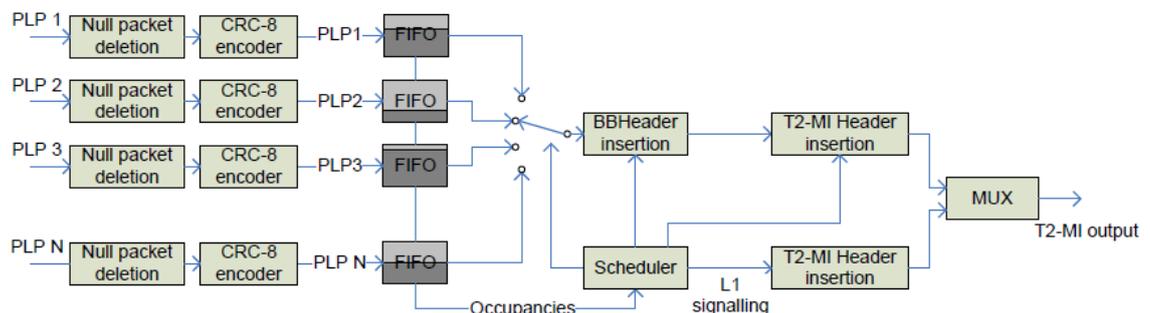


Ilustración 11: Posible implementación del Scheduler [5]

Para lograr la asignación de celdas OFDM a los PLP, el Scheduler debe ser el encargado de generar la señalización L1 dinámica, analizando la cantidad de bloques

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

FEC de cada PLP, para después decidir su ubicación dentro de la T2-Frame. Tanto la cantidad de bloques como el comienzo de un PLP vienen indicados dentro del L1-Dynamic.

En las guías de implementación de DVB-T2, vienen dos posibles implementaciones distintas del Scheduler para la asignación de bloques FEC, una basada en ventanas de colección y asignación solapadas entre ellas y otra también basadas en ventanas pero sin solape.

El primer caso es el más complejo, y para su correcto funcionamiento es necesario imponer una tasa binaria mínima a los PLP; pero es el más eficiente. En el segundo caso, no tenemos ninguna imposición, puesto que al acabar el período de una ventana, si no hemos llenado una trama, lo haremos con Padding; el algoritmo es más simple, a costa de la complejidad.

Otro bloque fundamental dentro del **Stream Adaptation** es el **L1-Signalling**. Este bloque se encarga de generar toda la información necesaria para que el receptor pueda demodular y decodificar correctamente la señal T2 radiada. La información dinámica la obtendrá del Scheduler, mientras que el L1-Configurable solo puede variar entre Superframes, por lo que el propio módulo puede guardar las variables en su memoria, junto al L1-Presignalling y al preámbulo P1. La señalización L1 no va codificada con los mismos bloques FEC que los datos de los PLP; en su lugar se utilizan bloques cortos de 16200 bits.

Por otro lado, para poder generar el L1-Future, necesitamos un buffer de T2-Frames para que el Scheduler pueda analizar la trama siguiente y crear la señalización L1 futura. El bloque que se ocupa de ello es el **Frame Delay**. No es más que un bloque opcional con una memoria. Este bloque solo es necesario si tenemos activado el L1Repetition.

El siguiente paso de la cadena es añadir el Padding o el Inband, si lo hemos activado. El módulo encargado de ello es el **Inband-Signalling/Padding Insertion**. Es el encargado de generar la señalización Inband es el Scheduler, este módulo solo introducirá al final de la primera BBFrame de una Interleaving Frame el Inband

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

seleccionado. Si no tenemos la opción de Inband activada, este bloque rellenará con '0' las BBFrame hasta llegar a 64800 bits para el código LDPC.

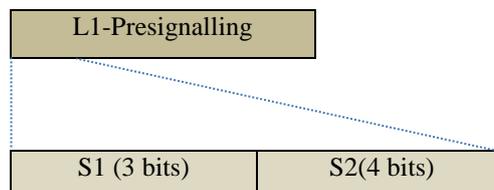
El último bloque del **Stream Adaptation** es el **BB Scrambler**. Tiene como finalidad aleatorizar y dispersar la energía de un BBFrame. Se usa un registro de desplazamiento con el polinomio:

$$1 + x^{14} + x^{15}$$

También tenemos la opción de usar Scrambling en el L1-Configurable y L1-Dynamic, con el mismo objetivo, dispersar la energía.

Después de pasar por el **Mode Adaptation** y el **Stream Adaptation**, las BBFrame y la señalización L1 ya se encuentran preparadas para ser codificadas, moduladas y radiadas por el resto de los subsistemas del transmisor DVB-T2.

La especificación también permite el uso de FEF (Future Extension Frames), ubicadas al final de una trama T2, para usos privativos o no especificados en la norma. Los encargados de señalar el uso o no de las Future Extension Frames son los preámbulos S1 y S2, dentro del L1-Presignalling, necesarios los dos para poder decodificar el resto de la transmisión.



S1	Formato del preámbulo S2	Descripción
000	T2_SISO	El preámbulo corresponde a un sistema T2 SISO
001	T2_MISO	El preámbulo corresponde a un sistema T2 MISO
010	No es T2	El preámbulo no corresponde a un sistema T2, si no a un

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

		sistema que solamente utilice FEFs
011	T2_LITE_SISO	El preámbulo corresponde a un sistema T2-LITE SISO
100	T2_LITE_MISO	El preámbulo corresponde a un sistema T2-LITE MISO
101	Reservado para futuro uso	El preámbulo corresponde a sistemas no especificados aún y de uso futuro
110	Reservado para futuro uso	
111	Reservado para futuro uso	

S1	S2 1er campo	S2 2do campo	Significado	Descripción
XXX	XXX	0	No mezclado	Todos los preámbulos de este sistema son del mismo tipo
XXX	XXX	1	Mezclado	Se emiten preámbulos de distintos tipo en este sistema

Tabla 4: Posición dentro del L1-Presignalling y sintaxis de los campos S1 y S2.

Para señalar el uso conjunto de FEFs junto a una transmisión de T2-Frames, basta con poner el cuarto bit del S2 a “1”, pero sin ignorar que el resto del preámbulo debe de indicar correctamente que tipo de trama estamos radiando.

Dentro de la estructura de la señal T2, los FEFs van incluidos dentro de una Superframe junto a T2Frame. El parámetro *FEF_interval* nos indica cada cuantas T2Frames va a ir una Future Extension Frame, mientras que *FEF_LENGTH* indica la duración en milisegundos. Ambos parámetros son señalizados en el L1-Postsignalling.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Por ejemplo, con un FEF_Interval de 2, y una Superframe compuesta por 4 T2Frame, la señal resultante tiene la siguiente forma:

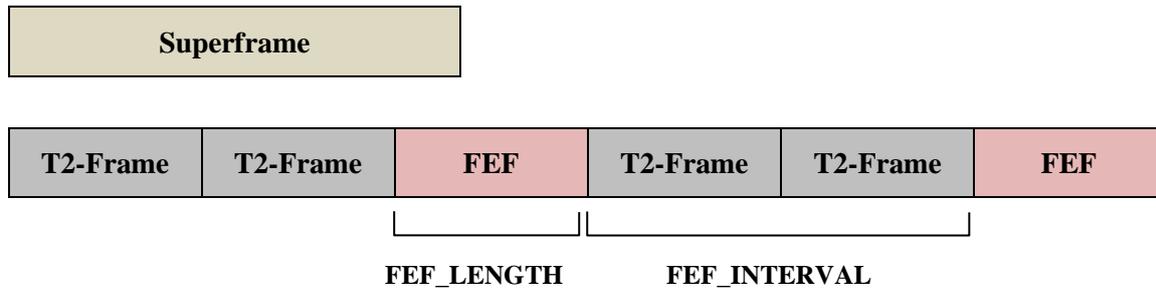


Ilustración 12: Ejemplo de una señal T2 usando FEF y un FEF_INTERVAL de 2.

La especificación de DVB-T2 exige, en el caso de usar FEF, que lo último enviado en una SuperFrame antes de empezar la siguiente sea un FEF. Esto implica que *FEF_Interval* tiene que ser un divisor del número de T2-Frames por SuperFrame.

Dentro de un Future Extension Frame, solo estamos obligados por la norma a incluir un preámbulo P1, es decir, un símbolo OFDM generado por una FFT de 2k subportadoras, modulado en BSPK y codificado con Code Rate 1/2. Con este preámbulo indicamos al receptor DVB-T2 convencional que los datos que vienen a continuación deben de ser ignorados.

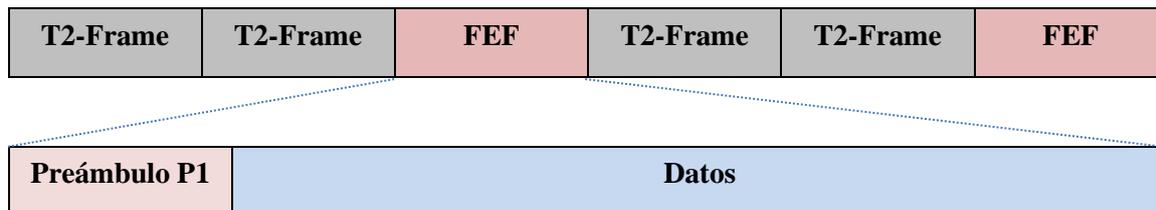


Ilustración 13: Estructura de un FEF

2.3. Interfaz T2-MI

La **interfaz T2-MI** [4] es la otra posible interfaz de salida que permite el Gateway-T2. Mientras que la interfaz BICM está pensada para ser la alimentación del

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

codificador de DVB-T2, la interfaz T2-MI cumple otras funcionalidades. En concreto, la interfaz T2-MI supe las necesidades requeridas por una red **SFN (Single Frequency Network)**.

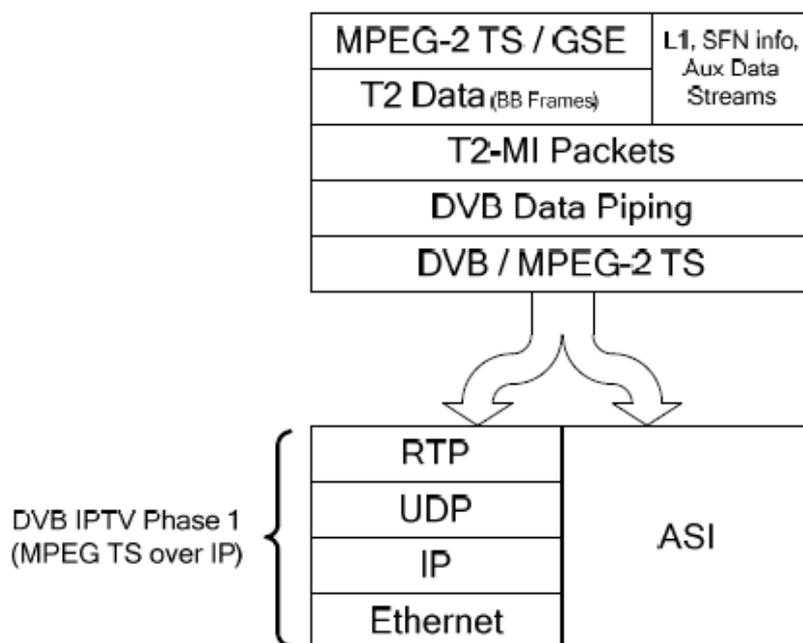


Ilustración 14: Pila de protocolos de la interfaz T2-MI [ETSI TS 102 773]

Para las redes orientadas a difusión que utilicen DVB-T2, el uso de la interfaz T2-MI otorga todas las herramientas necesarias para la correcta gestión de la SFN. Por la interfaz T2-MI viajan flujos T2-MI generados por el Gateway-T2. Estos flujos pueden llevar información sobre:

- El contenido audiovisual, dentro de las BBFrame, es decir, la información para broadcast.
- Señalización L1 para que un modulador sea capaz de construir las tramas T2 correctamente.
- Los Timestamp para que todos los moduladores de una red puedan emitir las Superframes en el instante adecuado.
- Vectores I/Q de flujos auxiliares o complementarios.
- Future Extension Frames.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

- Funciones específicas para moduladores individuales, para habilitar, deshabilitar o variar algún parámetro de transmisión.

El flujo T2-MI es transportado en paquetes T2-MI. Existen distintos tipos de paquetes T2-MI, en función de la carga útil que lleven (Señalización, flujo de vídeo, funciones específicas...).

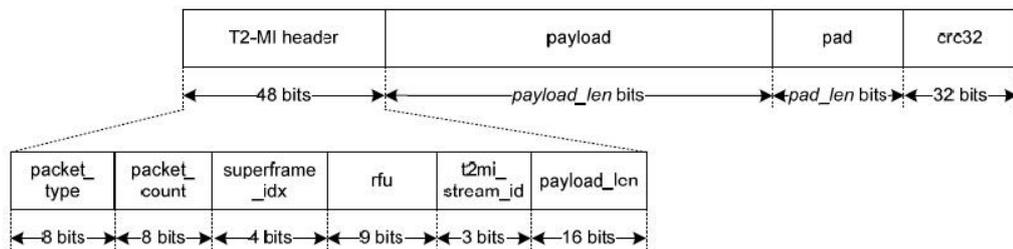


Ilustración 15: Sintaxis general de un paquete T2-MI (ETSI TS 102 773)

La cabecera consta de 6 bytes. El primero de ellos, packet_type, lleva información sobre el tipo de paquete T2-MI. Este campo puede tomar los siguientes valores:

T2-MI packet_type	Description
00 ₁₆	Baseband Frame
01 ₁₆	Auxiliary stream I/Q data
02 ₁₆	Arbitrary cell insertion
10 ₁₆	L1-current
11 ₁₆	L1-future
12 ₁₆	P2 bias balancing cells
20 ₁₆	DVB-T2 timestamp
21 ₁₆	Individual addressing
30 ₁₆	FEF part: Null
31 ₁₆	FEF part: I/Q data
32 ₁₆	FEF part: composite
33 ₁₆	FEF sub-part
all other values	Reserved for future use

Ilustración 16: Posibles valores del packet_type de un paquete T2-MI (ETSI TS 102 773)

Los tipos de paquete T2-MI son:

- Baseband Frame, este paquete transporta las BBFrame de los PLP desde el Gateway al modulador.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

- Auxiliary Stream I/Q Data, en el caso de usar flujos auxiliares de información, al transmisor le aportamos la información sobre el símbolo en fase y cuadratura para que pueda modularlo.
- Arbitrary Cell Insertion, sirve para fijar “manualmente” el valor de ciertas celdas OFDM, informando al transmisor del valor I/Q de las celdas. Útil para llevar información propietaria.
- L1-Current, transporta toda la señalización L1, tanto los preámbulos P1, como el L1-Presignalling, el L1-Configurable y el L1-Dynamic de la T2-Frame actual.
- L1-Future, transporta la señalización L1-Dynamic de la T2-Frame siguiente si usamos L1 Repetition, y/o el L1-Inband si hemos activado la opción.
- P2 bias balancing cells, este paquete indica cuantas celdas de los símbolos P2 de una T2-Frame son usadas para compensar la polarización de la señal, es decir, equilibrar la cantidad de ‘1’ y ‘0’s.
- DVB-T2 Timestamp, indica al modulador el tiempo de emisión de una Superframe. Puede funcionar en modo absoluto, relativo y “null”. Es mandatorio enviarlo porque lleva información del ancho de banda del sistema, aunque no precisemos de sincronización temporal.
- Individual Addressing, un conjunto de funciones destinadas a la gestión de SFN, que son enviadas a transmisores individuales. DVB-T ya introdujo funciones para ello, y DVB-T2 añade nuevas, principalmente relacionadas con técnicas de PAPR (Peak to Average Power Reduction).
- FEF part: null, este paquete indica al modulador que no tiene que enviar información dentro la Future Extension Frame. Solo llevamos información del preámbulo P1, mandatorio en los FEFs.
- FEF part: I/Q data, informa al transmisor de los valores I/Q que se tienen que modular en una Future Extension Frame, junto al preámbulo P1.
- FEF part: composite, alertamos al modulador del uso de sub parts en una Future Extension Frame y de la longitud de cada subparte.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

- FEF sub part, en una Future Extension Frame podemos tener distintos tipos de datos, tales como: Valores útiles I/Q, una subparte vacía o “null”, o una secuencia pseudo-aleatoria para pruebas. Todo esto lo indicamos en este paquete.

En función del tipo de paquete, el payload tendrá un formato, sintaxis y longitud distintos.

Por otro lado, todos los paquetes T2-MI son broadcast, es decir llegan a todos los moduladores de una red por igual. El protocolo T2-MI también especifica el orden en el que estos paquetes deben llegar al modulador:

- Primero, todos los paquetes con información de las Baseband Frame deben ser enviados y recibidos en el orden original con el que deben ser radiados, hasta haber enviado todos los datos de la T2-Frame.
- Al tener todos los datos de una trama T2, hay que enviar un paquete de Timestamp. Este paquete es obligatorio, queramos o no una sincronización temporal entre moduladores, puesto que porta la información del ancho de banda utilizado.
- Después de haber enviado un Timestamp, hay que transmitir la señalización L1 correspondiente a la trama, es decir, un paquete L1-Current, para que el modulador sea capaz de componer correctamente la trama con la información contenida en los paquetes Baseband. Si tenemos habilitado la repetición del L1 o Inband para un PLP, lo enviaremos a continuación del L1-Current, en un paquete L1-Future.
- Los paquetes de funciones específicas (Individual Addressing) se pueden enviar en cualquier momento.
- Con la información de una T2-Frame ya completa, incluida la señalización, ya podemos pasar a enviar la información relativa a una nueva trama.

Es el Gateway-T2 quién se tiene que asegurar que se respete el orden de los paquetes T2-MI en un flujo T2-MI.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Un flujo T2-MI puede viajar encapsulado en MPEG-TS, para ser enviado por un enlace ASI (Asynchronous Serial Interface). El hecho de tener un Transport Stream nos permite reutilizar toda la infraestructura existente, basada en ASI, para los antiguos sistemas de DVB-T. La especificación nos permite una alternativa, el envío de los flujos T2-MI mediante el protocolo IPv4, a fecha actual, IPv6 no se encuentra soportado.

Se ha mencionado anteriormente que el interfaz T2-MI suplente las necesidades de una red SFN que use la tecnología DVB-T2. Recordando, los requisitos eran los siguientes: Los transmisores tienen que emitir en la misma frecuencia y radiar la misma señal simultáneamente.

Para satisfacer la primera condición, que los transmisores emitan en la misma frecuencia, el paquete T2-MI de señalización L1 contiene toda la información de configuración del sistema (Ancho de banda, frecuencia central, tamaño de la FFT...) Estos valores viajan en el L1-Presignalling.

De forma similar, para que todos los transmisores puedan componer los mismos símbolos OFDM, toda la información a transmitir se encuentra en los paquetes T2-MI que contienen las BBFrame. Junto a los datos aportados por el L1-PostSignalling, todos los moduladores son capaces de generar las T2-Frames acorde con lo calculado por el Scheduler. Esto satisface la segunda condición de una red SFN.

La última condición, la simultaneidad temporal de la información radiada por todos los transmisores, es la más compleja de cumplir. Si este requisito no fuera cumplido, el receptor podría experimentar una interferencia entre símbolos, al superar el delay sufrido por el multicamino el intervalo de guarda de los símbolos OFDM. En DVB-T, el método utilizado para la corrección del delay era el envío de un paquete TS especial, el Mega-Frame Initialization Packet o MIP.

Este paquete contiene, además de la información sobre los parámetros de transmisión, dos campos que habilitan la sincronización: **synchronization_time_stamp** y **maximum_network_delay**. Synchronization_time_stamp contiene el valor de un contador de período de 100 ns, desde el último segundo de una referencia común (una

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

señal GPS o de la información PSI/SI); y `maximum_network_delay` es el delay máximo permitido.

En una red SFN, cada transmisor experimentará un delay distinto. Cuando un transmisor reciba un paquete MIP, revisará la marca de tiempo (`synchronization_time_stamp`), calculará entonces la diferencia con su referencia temporal para saber el delay que experimenta. Entonces retrasará aún más la señal para llegar al `maximum_network_delay` y radiar la señal.

En DVB-T2, el paquete T2-MI Timestamp es quien asegura la sincronización temporal de los transmisores. Este paquete indica el instante de tiempo en la que la primera trama T2 dentro de una supertrama debe de enviarse al aire.

T2-MI Timestamp tiene **dos modos de funcionamiento**, el absoluto y el relativo. En el caso de que no queramos usar sincronización, hay que enviar un “ Null Timestamp”. La sintaxis del paquete es la siguiente:

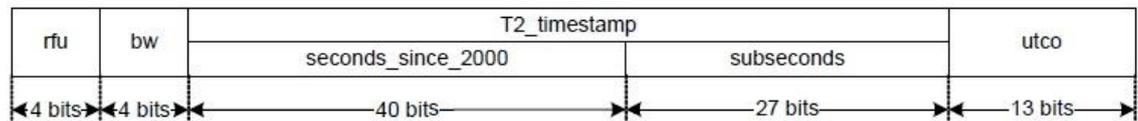


Ilustración 17: Sintaxis del T2-MI Timestamp (ETSI TS 102 733)

En el modo absoluto, el paquete transporta dos campos, uno de ellos se denomina “seconds_since_2000”, es el valor que ha transcurrido en segundos desde el 01-01-2000 00:00:00 GMT; y el otro campo necesario se llama “subseconds”, un contador del número de períodos de la señal FFT. “seconds_since_2000” es el ajuste grueso de la sincronización y “subseconds” es el ajuste fino. Para obtener el tiempo de emisión, se calcula de la siguiente forma:

$$T_{emission} = seconds_since_2000 + subseconds * T_{sub}$$

En el modo relativo, no tenemos el valor absoluto de los segundos. El ajuste grueso lo obtenemos a partir del SI(Service Information), mientras que el ajuste fino lo

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

sigue proporcionando el campo “subseconds”. En este caso, el tiempo de emisión se calcularía de la siguiente forma:

$$T_{emission} = subseconds * T_{sub}$$

Para los dos modos de funcionamiento, el valor de $T_{emission}$ se actualiza Superframe a Superframe, añadiendo al tiempo de emisión la duración temporal de una Superframe. Un modulador T2 puede calcular, en el caso de perder la sincronización y a partir de un valor de $T_{emission}$ previo, el $T_{emission}$ necesario para la Superframe siguiente.

Se puede usar, en el caso de transmisores o relays accedidos por radio, una versión del MIP de DVB-T, llamado T2-MIP; el paquete T2-MIP es el equivalente en DVB-T2 para el paquete MIP de DVB-T.

La función es la misma, asegurar que todos los transmisores de la red SFN emiten a la misma frecuencia, simultáneamente, la misma información, en el caso especial de que tengamos un repetidor o regenerador al cual accedemos por el acceso radio. Sin embargo, no tenemos información sobre los parámetros de transmisión, puesto que son transportados en la señalización L1.

Para lograr la sincronización temporal entre transmisores, hay un campo dentro del paquete T2-MI Timestamp ya mencionado anteriormente, transportado dentro del T2-MIP, llamado `seconds_since_2000`, que junto al otro campo llamado `subseconds`, permiten hallar el tiempo de emisión de forma absoluta de una superframe. Con este valor, el transmisor puede esperar el tiempo suficiente para asegurar la sincronización entre transmisores.

La sintaxis del paquete T2-MIP es la siguiente:

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Syntax	Number of bits	Identifier
t2_modulator_information_packet() {		
transport_packet_header	32	bslbf
synchronization_id	8	uimsbf
section_length	8	uimsbf
t2_timestamp_mip_length	8	uimsbf
t2_timestamp_mip	88	bslbf
rfu_length	8	uimsbf
for i = 1..rfu_length {		
rfu_byte	8	uimsbf
}		
individual_addressing_length	8	uimsbf
for j = 1..individual_addressing_length {		
individual_addressing_byte	8	uimsbf
}		
crc_32	32	rpchof
for k = 1..stuffing_length {		
stuffing_byte	8	uimsbf
}		
}		
NOTE 1: Optional parameters are shown in <i>italic</i> .		
NOTE 2: The total length of a T2-MIP shall always be 188 bytes.		

Ilustración 18: Sintaxis del paquete T2-MIP (ETSI TS 102 773)

Destacar que el formato de los campos `t2_timestamp_mip` y las funciones relacionadas con `individual_adressing` son las mismas que las definidas para los tipos de paquetes T2-MI Timestamp e Individual Adressing Functions, respectivamente.

Tiene que haber solo un paquete T2-MIP en cada Superframe, incluso en el caso de utilizar múltiples PLP, aunque no se encuentre al principio de ella. El PLP Común es un buen candidato para transportar este paquete.

3. Diseño del Gateway-T2 y del Analizador T2-MI

Este apartado trata sobre el diseño inicial del Gateway y del Analizador, tanto a nivel esquemático como en diagrama de flujo. Mientras que el Analizador sí que fue un programa comenzado desde cero, el Gateway-T2 está basado en plataformas de encapsulación ya existentes (DVB-H y DVB-T) dentro del iTEAM.

La primera etapa del Gateway-T2, adaptación de los archivos de entrada a un flujo constante de paquetes TS, generación de Components y Providers de cada servicio, junto a la generación de las tablas PSI/SI también ha sido heredada de plataformas anteriores, puesto que DVB-T2 solo cambia la capa física y la encapsulación, pero todas las capas superiores las mantiene.

3.1. Gateway-T2

El objetivo es implementar un Gateway-T2 que soporte “Mode A”, es decir, un solo PLP, con las funcionalidades básicas e imprescindibles para su funcionamiento, y a partir del esquema básico, ir añadiendo funcionalidades extra.

Aunque muchos bloques para un solo PLP no son necesarios o redundantes, también han sido introducidos para la futura actualización de la plataforma, con “Mode B” o múltiples PLP. El funcionamiento será siempre en “offline”, con una posible extensión en el futuro a “online” o en directo.

El esquema inicial del Gateway, a nivel de implementación, con todos los objetos necesarios, sería el siguiente:

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

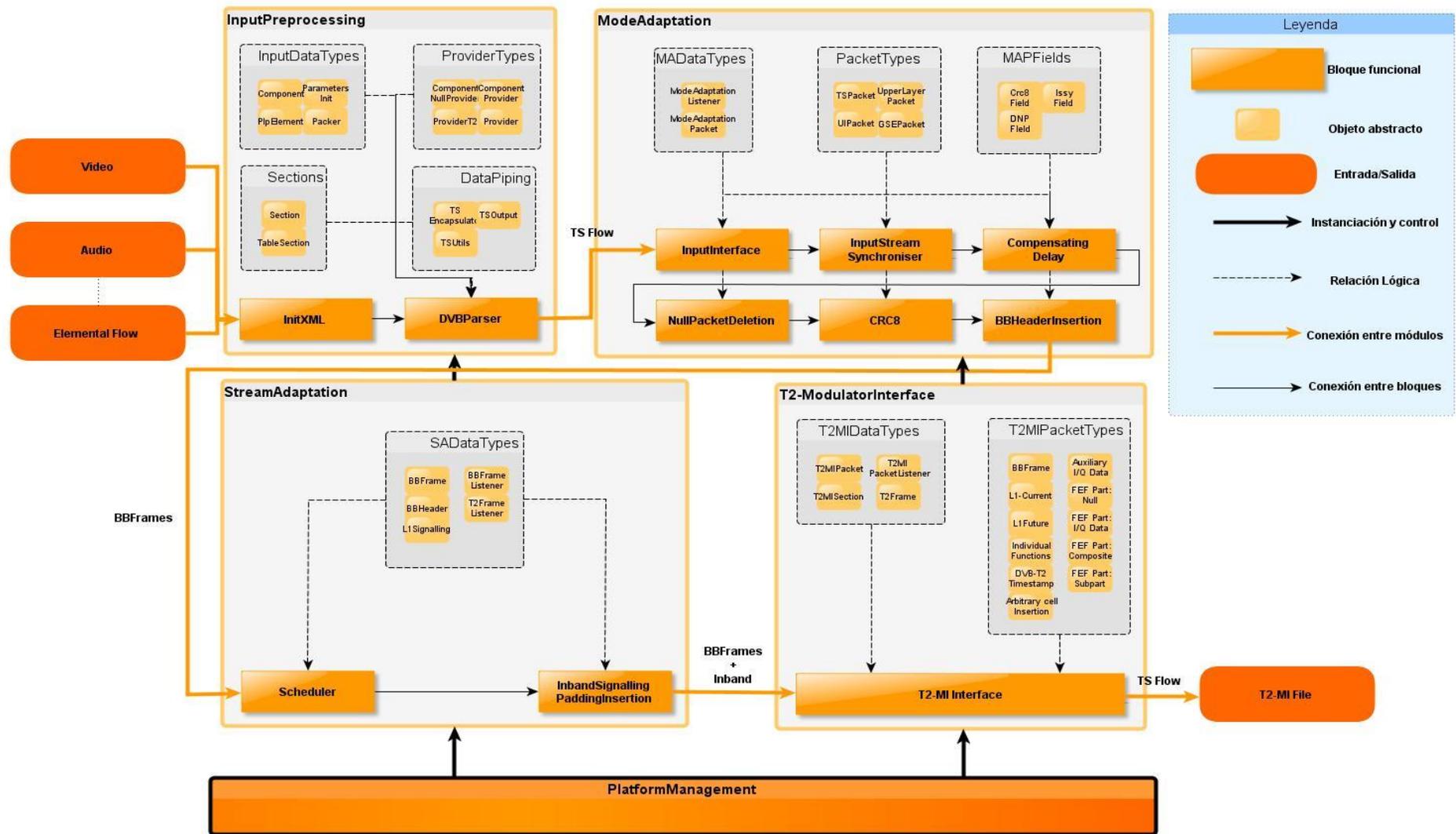


Ilustración 19: Esquema inicial del Gateway-T2

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

El módulo PlatformManagement es el que controla todo el sistema. Dentro de él, instanciamos el resto de módulos del Gateway (InputPreprocessing, ModeAdaptation, StreamAdaptation y T2MIInterface). También hacemos una labor de adaptación de las entradas al Gateway para los PLP: Pasamos de varios flujos continuos de información a un Transport Stream por PLP formado por paquetes de 188 bytes. El paquetizado de archivos se realiza mediante OpenCaster, un conjunto de funciones de uso abierto diseñadas especialmente para el Broadcast Digital. Por último, la información PSI/SI también es generada e introducida desde este módulo.

Comenzando por la entrada, el Gateway acepta cualquier tipo de archivo para ser encapsulado en T2, aunque en principio solo vayamos a usarlo para contenido audiovisual (vídeos, audios, subtítulos, información privada...).

Todos los archivos fuente tienen que ser indicados en el primer módulo, dentro del bloque inicial del sistema, "InitXML". Como su nombre indica, este bloque es un archivo XML donde definimos los parámetros del sistema (ParametersInit), la cantidad de PLPs (PlpElement), los servicios y los flujos elementales que los forman (Component)... La razón de usar este formato de archivo es principalmente reutilizar código y mantener el formato con las anteriores plataformas de encapsulación del iTEAM.

A continuación, DVBPaser, el otro bloque funcional del módulo InputPreprocessing es el encargado de leer los datos introducidos dentro del XML. DVBPaser instancia los objetos necesarios para el sistema, es decir, a partir de los datos crea un objeto ParametersInit, todos los PLP necesarios (PLPelement), y dentro de cada PLP, los servicios (Component) que transporten. Con la información ya lista, PlatformManagement ya puede procesar los archivos fuente para adaptarlos a la cadena de transmisión del Gateway, como se ha explicado anteriormente.

Para cada PLP, replicamos los módulos de ModeAdaptation y StreamAdaptation, puesto que cada PLP puede tener una configuración distinta (Por ejemplo, podemos tener activo el modo de alta eficiencia (HEM) para el PLP1 y en otro PLP distinto no).

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Todos los bloques del ModeAdaptation y del StreamAdaptation contienen un campo formado por un tipo de objeto ya definido en Java llamado LinkedBlockingQueue.

Este objeto [6] se comporta como una FIFO, el primer objeto de entrada, aquel que lleva más tiempo en la cola es el primero en ser sacado y atendido; “Linked” nos indica que se comporta como una lista enlazada, es decir, cada objeto de la cola apunta al siguiente y al anterior, esto acelera el acceso al primer elemento o al último, “Blocking” indica que las operaciones de introducir un objeto o extraerlo son bloqueantes, es decir, si queremos extraer un paquete de una LinkedBlockingQueue, el programa parará su ejecución y esperará hasta que tengamos un paquete disponible para seguir con su rutina habitual.

Para la cadena de bloques del Gateway, esta clase de objeto se adecúa bastante. Gracias a él, podemos independizar el funcionamiento de cada bloque del resto, implementando cada bloque en un “Thread” o hilo de ejecución distinto, con su cola de entrada, y al acabar el procesamiento de un paquete, pasárselo al siguiente bloque.

En el esquema adjuntado, el módulo a continuación del InputPreprocessing es el ModeAdaptation. Como pasará más adelante con el StreamAdaptation, todos los bloques del ModeAdaptation implementan la interfaz “ModeAdaptationListener”, que les permite estar interconectados entre ellos, indistintamente del orden o el nº de elementos. Todos los elementos que implementen la interfaz ModeAdaptationListener están obligados a tener una cola de entrada y un método para acceder a ella.

Esto nos permite una flexibilidad a la hora de definir los elementos de la cadena, puesto que todos los elementos manejarán objetos del tipo “ModeAdaptationPacket” en sus colas de entrada, independientemente del procesamiento que se realice en cada uno. ModeAdaptationPacket es un objeto cuyo principal campo es el Payload, un vector de Bytes con los datos del flujo de entrada. También tenemos una copia del paquete original, con la información de la cabecera, esto es útil si queremos saber el PID de los extraídos del flujo.

PlatformManagement es el encargado de instanciar los objetos “ModeAdaptation” y “StreamAdaptation” para cada PLP, y dentro de éstos, se instanciará el resto de

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

elementos que forman cada módulo, con los parámetros que necesiten. La misión del ModeAdaptation es pasar de un flujo paquetizado a BBFrames de 64800 bits, y transferir estas BBFrames al StreamAdaptation.

El primer bloque funcional del MA es el “InputInterface”. En el apartado de tecnologías, se ha mencionado que el objetivo de este bloque es extraer K_{BCH} bits de un flujo constante, pero en este Gateway en particular, el comportamiento es bastante distinto. Los flujos de entrada, aunque la especificación indica varios tipos de paquetes posibles, en la práctica solo se ha contemplado el uso de paquetes TS de 188 bytes. Lo que hacemos es una conversión de los paquetes genéricos de capa superior “UpperLayerPacket” al tipo de paquete soportado por las LinkedBlockingQueue del resto de bloques, “ModeAdaptationPacket”, y pasamos el paquete al siguiente elemento de la cadena.

A continuación, el siguiente eslabón de la cadena es el “InputStreamSynchroniser”. Se ha detallado en apartados anteriores cual es la función de este bloque: asegurar una tasa binaria y un delay constante extremo a extremo. Dentro del Gateway, este bloque se comporta de distinta forma en el caso de tener el Normal Mode o el High Efficiency Mode activado. En el caso de Normal Mode, a los ModeAdaptationPacket extraídos de la entrada se les añade un campo llamado ISSY al final del payload de 188 bytes. En cambio, si usamos HEM, el ISSY se introduce en la cabecera de las BBFrame, y el valor cambia del ISSY en lugar de variar de ModeAdaptationPacket a ModeAdaptationPacket, varía de BBFrame a BBFrame.

ISSY alterna entre 3 posibles valores distintos: ISCR, BUFS o TTO; para controlar que valor es introducido se usa un contador de módulo 3, siguiendo la especificación de DVB-T2:

Contador	ISSY
0	TTO
1	BUFS
2	ISCR

Tabla 5: Valores de ISSY en función del contador

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Una vez finalizado el procesamiento del paquete, se coloca en la cola del siguiente bloque, “CompensatingDelay”.

Para un funcionamiento en “Mode A” o un solo PLP, el bloque de CompensatingDelay tiene poco sentido. La finalidad es ajustar temporalmente la entrada al Stream Adaptation de los PLP de datos con el PLP común. Si solo tenemos un PLP, necesariamente será de datos, y no tendremos el requisito de la simultaneidad temporal. Por lo tanto, y hasta que no se implementen múltiples PLP, este bloque funciona de forma transparente, extrae ModeAdaptationPackets y sin realizar ninguna acción adicional, los coloca en la cola del siguiente.

“NullPacketDeletion” es la etapa a continuación del CompensatingDelay. Este bloque solo puede ser implementado si el flujo de entrada del Gateway es un Transport Stream. En caso contrario, no es necesario incluirlo. Lo que hacemos es sacar un ModeAdaptationPacket de la cola, mirar la PID del Transport Stream contenido, y si coincide con la PID reservada para los Null Packet (8191), borramos el paquete e incrementamos el valor del contador NPD (NullPacketDeletion). Por cada paquete Null consecutivo, lo borramos e incrementamos el contador. Cuando el paquete que viene a continuación sea un paquete de datos útiles, le añadimos al principio el valor del contador, reiniciamos el contador y enviamos el paquete ya listo a la siguiente cola del módulo.

El penúltimo bloque del ModeAdaptation es el “CRC8”. La implementación elegida para el cálculo del CRC8 es la generación de una tabla de consulta con los 256 valores posibles. Una vez calculada, para generar el CRC8, vamos pasando los bytes de un ModeAdaptationPacket extraído de la cola, sin contar el ISSY o el NPD, y calculamos el índice de la tabla que le corresponda, consultamos la tabla, obtenemos el CRC8, y lo añadimos al final del ModeAdaptationPacket, a continuación del ISSY si está siendo usado. Una vez acabado el procesado, pasamos el paquete al siguiente elemento del módulo.

Hasta este momento, lo único que han realizado los bloques es procesar los ModeAdaptationPacket, pero no hemos formado en ningún momento las BBFrame o las

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

BBHeader necesarias para el StreamAdaptation. El último bloque del ModeAdaptation se encargará de generar tanto las BBFrame como las correspondientes BBHeader.

“BBHeaderInsertion” tiene un papel muy importante, puesto que define la cantidad de bits útiles dentro una BBFrame, en función del Code Rate del PLP y si el L1-Inband está activado o no. El objeto BBFrame está formado, principalmente, por un vector de ModeAdaptationPacket, un vector de Bytes llamado “payload” con la información volcada del vector de ModeAdaptationPacket, un campo de L1Inband donde llevaremos información del PLP y todas las funciones para acceder a estos o modificarlos.

Muy probablemente tengamos que fragmentar algún ModeAdaptationPacket entre BBFrames, el que controlará el puntero para saber la posición en el paquete fragmentado será el BBHeaderInsertion. Por otro lado, no solo hay que llenar la BBFrame, sino también generar el BBHeader correspondiente. Esa es la funcionalidad principal de este bloque, calcular correctamente todos los campos de la cabecera de una BBFrame, en función del modo escogido (Normal o HEM).

Finalmente, antes de pasar al StreamAdaptation, BBHeaderInsertion realiza una conversión implícita, pasa de un conjunto de ModeAdaptationPacket a un objeto de tipo de BBFrame. Las colas del siguiente módulo, StreamAdaptation, ya no son de ModeAdaptationPacket, sino de BBFrame. Todos los bloques del StreamAdaptation ahora implementarán la interfaz “BBFrameListener”, muy similar al ModeAdaptationListener, pero ahora será obligado tener una cola de entrada de BBFrame y un método para acceder a ella.

En el primer bloque del ModeAdaptation, “Scheduler” reside la complejidad del sistema, aunque para el caso de Single PLP, el caso es trivial. Aunque el Scheduler debería ser el bloque que generara la señalización L1, a partir del análisis; en el primer diseño se delegó esta función a la interfaz T2-MI, junto a la generación del L1-Future si el parámetro L1-Repetition estaba activado. El Scheduler solo transfería las BBFrame al “InbandPaddingInsertion” de forma transparente. Será luego en el desarrollo donde se le otorgue al Scheduler la importancia que merece.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

InbandPaddingInsertion es el bloque encargado de insertar el Padding dentro de las BBFrame en el caso de que no estén completas del todo con datos útiles. El relleno de bits se realiza con “0”s, hasta alcanzar los K_{BCH} bits necesarios para la codificación BCH. En el caso de tengamos activado la señalización Inband en lugar del Padding, esta etapa es la que genera dicha señalización, pero no es insertada en la carga útil sino que viaja en un campo interno de la BBFrame, al tener un paquete T2-MI específico donde la información Inband es llevada.

Una vez acabado el procesado realizado por el StreamAdaptation, es el momento de pasar las BBFrame a la interfaz T2-MI para la generación del archivo T2-MI de salida. El módulo “T2-ModulatorInterface” está formado por un único bloque funcional, el “T2-MI Interface”. Dentro de este programa, vamos a asegurarnos de generar el archivo T2-MI encapsulado dentro de un flujo TS, listo para ser enviado por un interfaz ASI.

T2-MI Interface recoge las BBFrame de su cola de entrada, y con la información de ParametersInit, dimensiona el tamaño de una T2Frame, calculando a cuantos Data Symbols equivale una BBFrame, que dependerá del orden de la modulación utilizada. Una vez hemos llenado lo mayor posible una T2Frame, enviamos al TSOOutput estos paquetes, que se encargarán de paquetizar en bloques de 188 bytes la salida.

A continuación, y respetando el orden especificado por la DVB, creamos un paquete T2-MI “Timestamp”, con no solo la marca temporal de SuperFrame, si no también información con el ancho de banda del sistema T2. Después generamos la señalización L1 de la T2Frame actual, y la metemos en un paquete T2-MI “L1-Current”, a partir de la información de entrada al sistema. Por último, si tenemos el campo “L1-Repetition” activado o usamos Inband, hay que enviar un paquete T2-MI “L1-Future”.

Si no definimos una condición de parada dentro de la interfaz T2-MI, la plataforma nunca acabará su ejecución, incluso al finalizar los archivos de origen. Esto se debe a que sigue añadiendo las tablas PSI/SI y la señalización L1 cada cierto tiempo.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Todo lo anteriormente explicado forma el diseño inicial del Gateway. Esta implementación permite la funcionalidad básica de un solo PLP, en Normal Mode, con señalización L1.

Las funcionalidades de este Gateway inicial son:

BW soportados	5,6,7,8 y 10 MHz	PLP Mode	Single PLP (Tipo 1)
Tamaño FFT soportados	1,2,4,8,16 y 32K	Modo Transmisión	Normal
GI soportados	1/4, 1/8, 1/16, 1/32, 1/128, 19/128 y 19/256	Extensión BW	Permitido
Modulaciones PLPs datos	QPSK, 16QAM, 64QAM y 256QAM	Modulaciones L1	BPSK, QPSK, 16QAM y 64QAM
Datos CRs	1/2, 3/5, 2/3, 3/4, 4/5 y 5/6	L1 CRs	1/2
LDPC Size	16 y 64k	L1 extension	No permitido
PAPR	No permitido	L1 repetition	No permitido
FEF	No permitido	Tipo TI	0
TFS	No Permitido	Num RF Ch	1
Miso_Siso	Siso	T2MI	Permitido
Formatos de Entrada	Video MP2 → (MPEG2-TS)	Formatos de Salida	T2-MI → (MPEG2-TS)

Tabla 6: Prestaciones del diseño original del Gateway-T2

3.2. Analizador T2-MI

Un Analizador T2-MI, como su propio nombre indica, monitoriza la interfaz T2-MI. Tiene como entrada un archivo T2-MI, encapsulado dentro de un flujo TS, y como salida un archivo de texto donde volcar todos los valores, cabeceras, contadores... extraídos de éste. La función principal es hacer humanamente legible la información codificada de las cabeceras y señalización y comprobar la validez de los archivos T2-MI generados. El diagrama de flujo del funcionamiento es el siguiente:

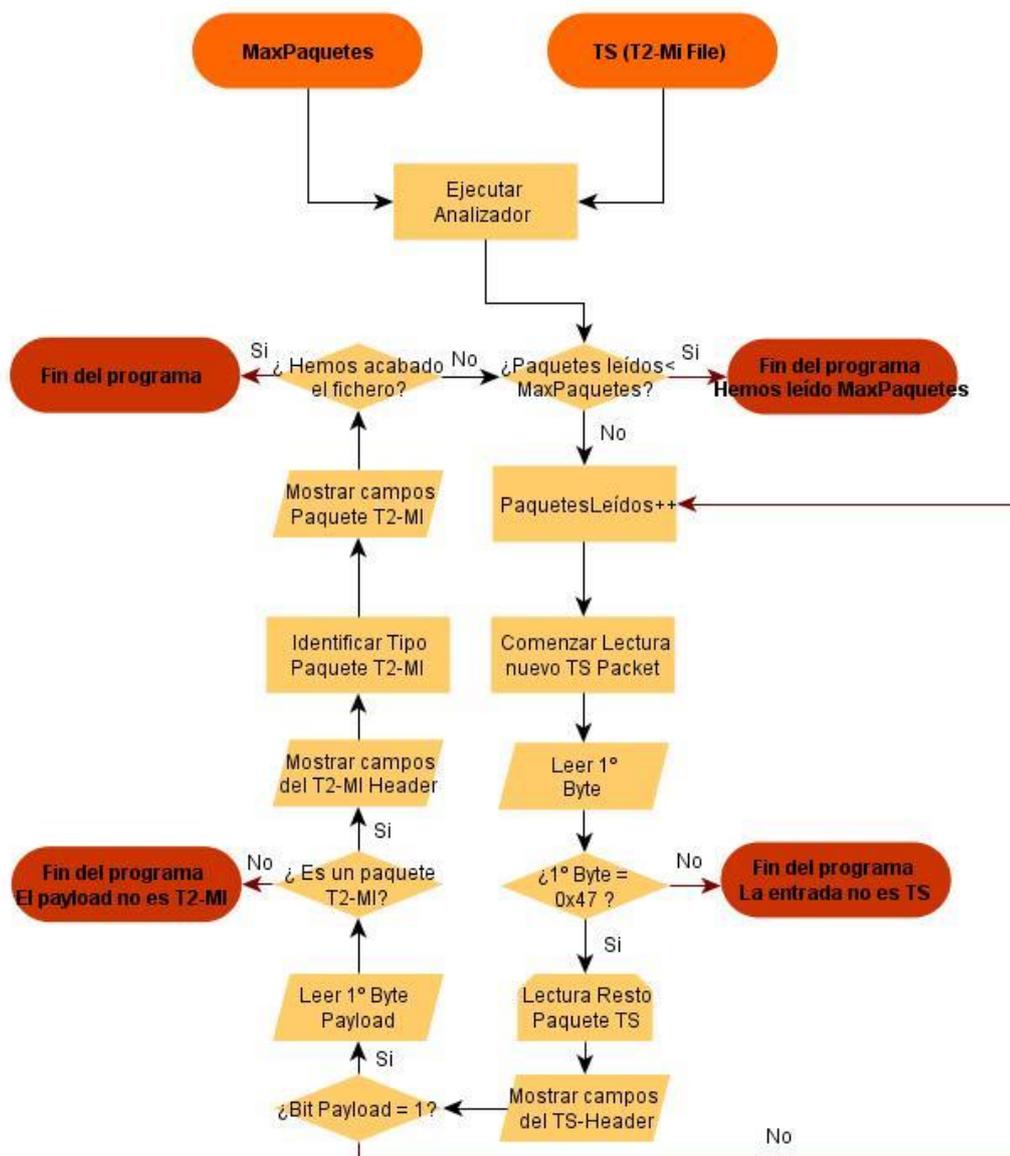


Ilustración 20: Diagrama de flujo del Analizador T2-MI.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

El proceso que sigue el analizador es el siguiente:

- 1) Se le pasan dos parámetros de entrada. Uno es la ruta del archivo y el otro la cantidad de paquetes que tiene que analizar: Debido a que un archivo T2-MI puede generar un archivo muy grande de salida, analizar una traza entera puede suponer una salida de varios Gigabytes. Si el objetivo solo es validar un archivo T2-MI, con 3000-4000 paquetes, que incluyan 2-3 T2Frames distintas es suficiente.
- 2) Extraer el primer byte, comprobar que es 0x47. Si lo es, el archivo de entrada es un candidato a ser un flujo TS. Si no lo es, salimos del programa porque el archivo de entrada no es un Transport Stream o está mal generado.
- 3) Incrementamos en 1 el contador de Paquetes Leídos y comenzamos a leer los bytes del Transport Stream. Extraemos un paquete TS y comprobamos si el bit nº10 de la cabecera del paquete TS, que indica si empieza un nuevo Payload en el resto del paquete o es la continuación de uno fraccionado.
- 4) Si el bit de Payload=0, esto indica de que no comienza un nuevo paquete T2-MI en este paquete, y la información transportada es porque no cabía en uno solo. Si esto es así, no analizamos más este paquete y volvemos a 3). Lo más usual es que corresponda a una BBFrame fragmentada.
- 5) Por otro lado, si este bit vale 1, en este TS empieza un paquete T2-MI nuevo del que habrá que leer el primer byte de su cabecera para saber qué tipo de información T2-MI porta. Si no conseguimos identificar el tipo de paquete T2-MI, es posible que el archivo TS no lleve Payload de T2-MI; si esto fuera así, salimos del programa. En cuanto identificamos el tipo de paquete T2-MI; el analizador se comportará de una forma u otra para mostrar los campos de cada paquete correctamente. Los paquetes implementados para analizar son aquellos que pueden generar el Gateway:

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

- 5.a) BBFrame. El Packet Type toma el valor 0x00. Dentro de este paquete vamos a decodificar los campos del BBHeader, en el caso de Normal Mode, son: Matype, DFL, UPL, SYNC, SYNCN y CRC8-MODE; sin embargo para HEM los campos que extraemos son: Matype, ISSY, DFL, SYNCN y CRC8-MODE.
- 5.b) L1-Current, con Packet Type = 0x10. Lleva toda la señalización L1 correspondiente al sistema T2 y a la T2Frame actual. Se compone del L1-Presignalling, L1-Configurable, L1-Dynamic y si lo hubiera, L1-Extension. El analizador debe extraer todas variables contenidas en este paquete.
- 5.c) L1-Future. Su Packet Type relacionado es 0x11 y su uso es opcional. Lleva el L1-Dynamic de la siguiente T2Frame, y el L1-Inband si hemos activado la opción. De forma similar al L1-Current, extraemos toda la información de la señalización.
- 5.d) Timestamp. Packet Type = 0x20. Lleva una marca temporal que indica el $T_{emission}$, el instante en el que una Superframe debe de ser radiado al aire, y la información del ancho de banda del sistema. El analizador tiene que ser capaz de calcular el $T_{emission}$ y mostrar el ancho de banda.
- 5.e) IndividualAddressing con Packet Type=0x21. Este paquete engloba una serie de funciones, dirigidas a moduladores individuales dentro de una SFN, para su gestión, y para definir parámetros que deben de ser comunes a toda la red (principalmente, el uso de técnicas de PAPR). El analizador decodifica este paquete enteramente, independientemente de la cantidad de funciones que transporte.
- 6) Una vez hemos acabado de analizar el paquete T2-MI, si no hemos acabado el fichero y el contador de Paquetes leídos es menos que MaxPaquetes, volvemos a 3) y vuelve a empezar el proceso.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

La salida de este programa será un archivo de texto con los paquetes T2-MI decodificados.

4. Desarrollo del Gateway-T2 y del Analizador T2-MI

4.1. Gateway-T2

El lenguaje de programación escogido para la implementación del Gateway es Java. El motivo principal es reutilizar código ya realizado dentro del grupo para plataformas anteriores, como ya se ha mencionado anteriormente. Otra razón es la simplicidad y el nivel de abstracción que permiten un lenguaje de programación orientado a objetos, que se adecúa bastante a los elementos lógicos (BBFrames, T2Frames, Interleaving...) en los que se basa el Gateway T2.

El entorno de desarrollo del proyecto es Netbeans IDE 6.9.1, comúnmente utilizado para implementaciones de gran calibre y software profesional, junto a Subversion, un sistema de control de revisiones y backup. Aunque los programas escritos en Java son multiplataforma, el Gateway solo funciona en distribuciones UNIX (el motivo se especificará más adelante), en concreto, se ha trabajado sobre Ubuntu 12.04, pero cualquier distribución que soporte Netbeans 6.9.1 serviría.

El desarrollo se ha basado en una línea de investigación ya comenzada en el iTEAM, cuyo estado inicial era el mostrado en el diseño. Durante la implementación de nuevas funcionalidades y depuración de las existentes, han ido surgiendo problemas y desafíos cuya resolución se detallará más adelante. El esquema final del Gateway, una vez finalizado el desarrollo, es el siguiente:

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

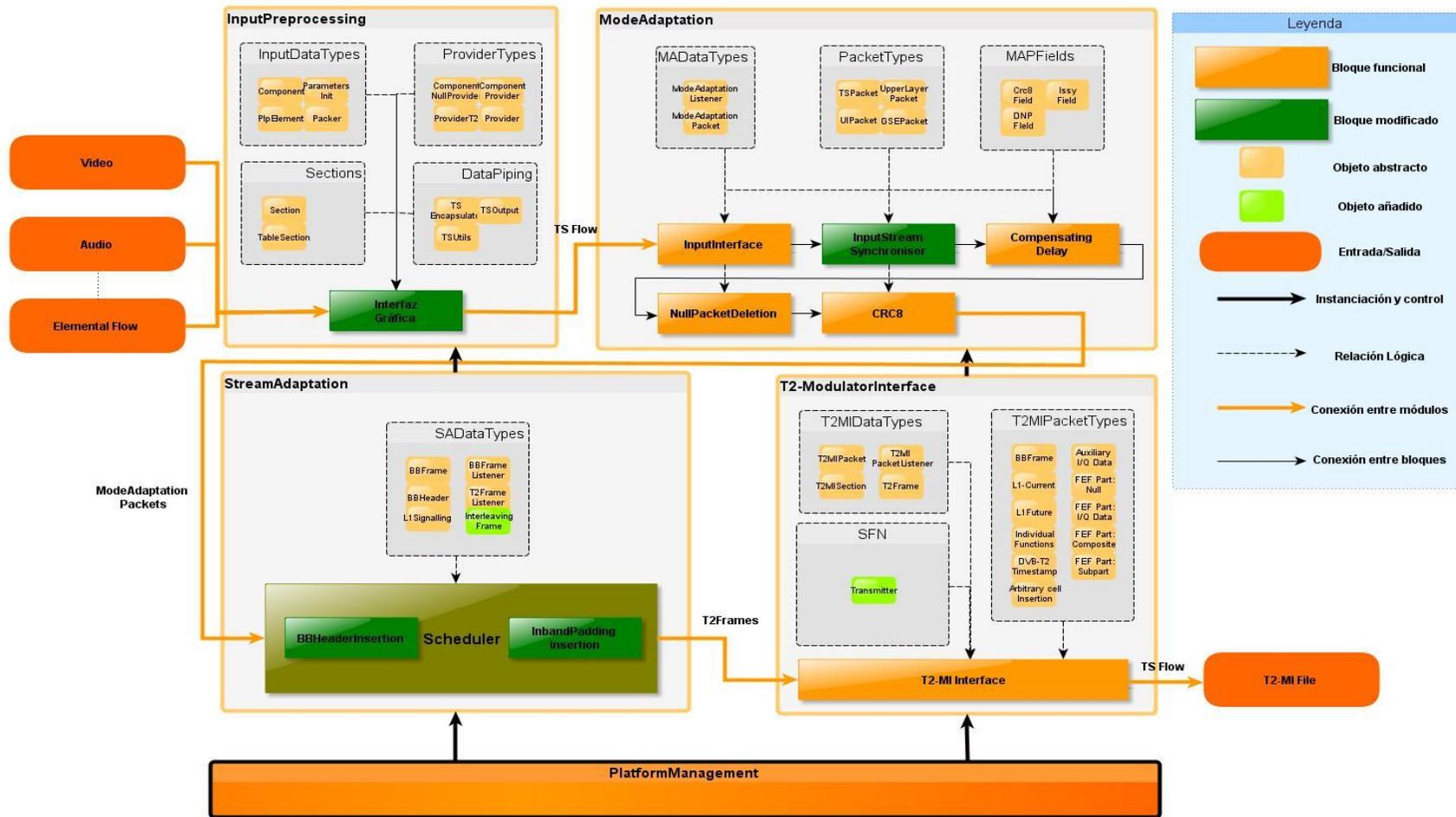


Ilustración 21: Esquema final del Gateway-T2

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Del diseño inicial, las mayores modificaciones se han realizado sobre la entrada, sobre el `InputStreamSynchroniser` y sobre el `Scheduler`. También se han añadido soporte para `Future Extension Frames` para posibles aplicaciones y líneas de investigación futuras.

a) Entrada

Para configurar los parámetros de todas las anteriores plataformas del iTEAM, se tenía que modificar un archivo XML el cual era interpretado por la plataforma, creaba los objetos necesarios y arrancaba su ejecución. Este método, aunque rápido computacionalmente, era poco intuitivo y no permitía ningún tipo de control sobre la coherencia de parámetros introducidos.

Es por esto que se diseñó una interfaz gráfica para configurar el Gateway-T2. Por un lado, se quería hacer más visual la configuración elegida. Otra de las ventajas es que es la interfaz es interactiva; no solo comprueba que la configuración escogida es válida, sino que además muestra las prestaciones del sistema (Capacidad, duración de trama...) con lo escogido.

Pasando a un apartado más técnico, la interfaz se puede dividir en cuatro bloques principales, según qué aspecto del sistema configuren: Parámetros comunes, PLPs, SFN y Prestaciones.

Parámetros comunes controla la configuración global del sistema. De todos los valores introducidos, son especialmente relevantes el tamaño de FFT utilizado, el ancho de banda, el número de Data Symbols de una T2Frame, el número de frecuencias y la frecuencia central de emisión.

En el bloque de PLPs se añaden o quitan PLPs al sistema, y es posible definir los grupos de PLP, con un PLP común. De cada PLP, también se configuran todos sus parámetros, tales como modulación, Code Rate, el uso de Inband, la cantidad máxima de FECBlocks por Interleaving Frame... Además, aquí definimos cada servicio (Component) que van a llevar los PLP por separado, con el bitrate asociado.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Respecto al bloque de SFN, como su nombre indica, controla todo lo relacionado con la gestión y configuración de la red SFN asociada. En ella añadimos o quitamos transmisores con sus respectivos valores de Offset, tanto frecuenciales como temporales, la potencia de transmisión... y definimos los parámetros globales, principalmente relacionado con el PAPR.

Una vez elegida la configuración general, el número de PLPs y su parametrización y los datos sobre una posible SFN, la interfaz transfiere toda la información al resto del programa que comenzará su ejecución. En concreto, se le pasa un objeto “ParaIni”, que contiene toda la configuración general del sistema, incluyendo el número de PLPs; además de un objeto “FEF” con la posible información de extensión y un objeto “SFN” con todo aquello relacionado con los transmisores de la red y sus parámetros de configuración (Time o Frequency Offset, Transmission Power...)

b) Scheduler

Otro gran punto importante del desarrollo recae sobre el Scheduler. Sobre este bloque se han realizado numerosos cambios y mejoras, con diversos objetivos:

- Preparar el código para soportar múltiples PLPs y a tamaños de BBFrame variables
- Llenar las T2-Frames sin exceder el número máximo de Data Symbols o el número máximo de FECBLOCKS definidos para un PLP, asegurando siempre un mínimo de bits útiles en una BBFrame. Con esta asignación de Data Symbols a PLPs, el Scheduler controla la tasa binaria que va a ofrecer cada PLP.
- Comportarse de forma similar para las Interleaving Frame, que será un vector de T2-Frames de longitud configurable. Dar soporte a Señalización L1-Future. . Dar soporte a la señalización L1-Inband.
- Si tenemos el campo de sincronización activo (ISSY), el Scheduler será el encargado de, una vez definida la ubicación y asignación de los PLP,

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

calcular e incrustar el campo de TTO (Time to Output) dentro de la BBFrame.

- Cálculos de parámetros indirectos a partir de la configuración, para facilitar la ejecución, en un objeto aparte para aprovechar el paralelismo.

El primer punto, adaptar el código para soportar múltiples PLP, aunque no solo atañe al Scheduler sino a todo el Gateway, pero hay que prestar especial atención en este bloque puesto que pasamos de N cadenas de Mode Adaptation (donde N es el número de PLP de datos) a un solo flujo de T2-Frames que el módulo de T2-MI sea capaz de interpretar para generar el flujo T2-MI. Para ello, el Scheduler chequea cada P_i T2-Frames (el tiempo que dura una Interleaving Frame) la cola de entrada de cada PLP, y si supera un umbral de ModeAdaptationPackets, se llenará una BBFrame con ellos, y cuando tengamos un conjunto que luego serán asignados a T2-Frames. La cantidad de paquetes será variable cada vez que el Scheduler compruebe la cola, por lo que el tamaño de las BBFrame de datos útiles también será variable. Se puede dar el caso de que no superemos el umbral, por lo que el Scheduler esperará un tiempo y comprobará de nuevo la cola de un PLP.

Esto enlaza con el siguiente punto realizado en el Scheduler: controlar que no excedemos ni la capacidad de la T2-Frame ni asignamos más bloques FEC a un PLP de los definidos inicialmente por MAXFEQBLOCKS. Para ello, mientras vamos llenando una T2-Frame, tenemos un contador decreciente que controla la cantidad de BBFrames que un PLP tiene ya dentro de una trama, y si este contador llega a 0 antes de llenar la trama T2, o no nos quedan Data Symbols disponibles, nos salimos del bucle de llenado.

La siguiente función implementada es la generación de las Interleaving Frame y la asignación de T2-Frames a estas. Las Interleaving Frames están gobernadas por dos parámetros: TIME_IL_LENGTH y TIME_IL_TYPE. Si TIME_IL_TYPE es igual a 1, TIME_IL_LENGTH define la cantidad de T2-Frames que van a componer una Interleaving Frame. Cabe destacar que el parámetro MAXFEQBLOCKS de un PLP define la cantidad de BBFrames que una Interleaving Frame puede contener. Se ha definido un objeto llamado “Interleaving Frame”, constituido por un vector de T2-

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Frames, que el Scheduler se encargará de llenar cada vez que tengamos una T2-Frame lista.

Esta implementación nos permite calcular de una forma muy sencilla el L1-Repetition (si lo usamos): la primera T2-Frame n que añadimos a la Interleaving es la actual, calculamos la señalización L1 correspondiente, que será el L1-Actual; y de la siguiente trama $n+1$ que transporta la Interleaving obtendremos su señalización, que pasará a ser el L1-Future de la T2-Frame n . En la siguiente iteración, el L1-Future de la trama $n+1$ pasará a ser el L1-Actual, y la señalización de la trama $n+2$ pasará a ser el L1-Future de la trama anterior. La generación y la asignación de la señalización Inband también corren a cargo del Scheduler, puesto que necesitamos tener una Interleaving Frame completa para poder elaborar la señalización.

Por otro lado, si utilizamos sincronización (ISSY), hay 3 campos distintos que introducir cíclicamente: TTO, BUFS y ISCR, como se ha comentado anteriormente. Mientras que ISCR y BUFS pueden ser generados sin problemas por el InputStreamSynchronizer, TTO (Time to Output) es el número de períodos elementales desde que termina la señalización hasta que comienza determinado PLP de datos. No es posible conocerlo a priori hasta que los PLP han sido asignados por el Scheduler en las T2-Frames, por lo que el Scheduler será el encargado de generar el TTO e introducirlo cuando corresponda dentro de las BBFrame para Normal Mode o en la BBHeader para High Efficiency Mode.

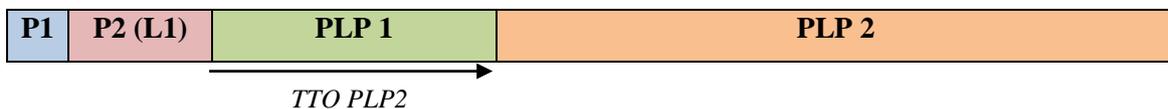


Ilustración 22: Ejemplo de Time to Output para una T2-Frame

Por último, para reducir el procesamiento del Scheduler, aprovechar la potencia de los hilos de ejecución y mantener el código limpio, se ha decidido crear un objeto aparte llamado SchedulerUtils, que realiza todos los cálculos relacionados con las T2-Frames, tales como calcular la duración de una T2-Frame, de una Interleaving, la conversión de

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

celdas OFDM (Data symbols) a FEC BLOCKS y viceversa, la generación de la señalización L1 actual y futura...

Para realizar todo lo mencionado anteriormente, se ha tomado la decisión de incluir dentro del Scheduler el bloque de BBHeaderInsertion del Mode Adaptation, y del Stream Adaptation, el módulo de L1InbandGeneration:

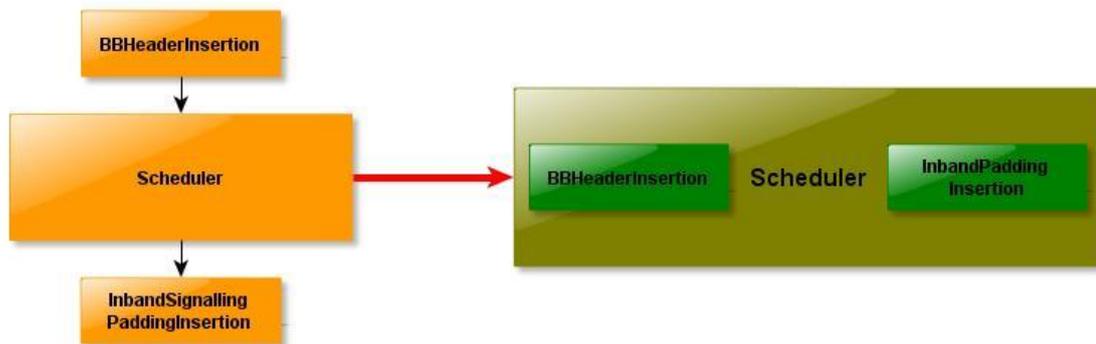


Ilustración 23: Cambio realizado en los bloques del Scheduler

c) InputStreamSynchronizer

Otro bloque que ha sufrido grandes cambios desde el diseño ha sido el módulo dedicado a sincronizar distintos PLP: InputStreamSynchronizer. El objetivo de este bloque es asegurar una tasa binaria constante (CBR) y asegurar un retardo extremo a extremo constante.

Como se ha explicado anteriormente, InputStreamSynchronizer genera un campo llamado ISSY, que tiene 3 atributos distintos: BUFS, TTO e ISCR. De estos, el más problemático a la hora de ser implementado ha sido el ISCR (Input Stream Clock Reference). Cada dos ModeAdaptationPacket (cada dos BBFrames en el caso de usar HEM) que pasen por el módulo de InputStreamSynchronizer, el siguiente deberá llevar en su campo ISSY el ISCR.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

ISCR coge su valor de un contador de períodos elementales, como se puede ver en la [tabla 2](#). Por ejemplo, para un ancho de banda de 10 MHz, el período elemental es de $7/80 \mu s$, que pasándolo a decimal, da un tiempo de 87.5 nanosegundos. Desgraciadamente, Java no permite “Timers” de nanosegundos por cuestiones técnicas del lenguaje. Para poder conseguir un contador con tanta resolución, se ha escogido por implementar un programa en C, con instrucciones en ensamblador, capaz de llevar la cuenta de los ciclos del procesador (obtiene el Time Stamp Counter del procesador), que se ha integrado en Java mediante el uso de JNI (Java Native Interface), que permite la inclusión de scripts en C en programas Java.

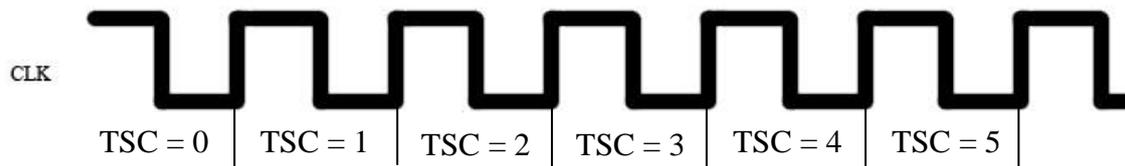


Ilustración 24: Ejemplo de señal de reloj y Time Stamp Counter

Aun así, es posible que haya incoherencia en estos números de un ordenador multinúcleo, el planificador de tareas de un Sistema Operativo, si un núcleo no se está utilizando, se pondrá en hibernación, reseteando su valor de TSC en algunos casos. Para asegurar la coherencia del TSC, se ha tomado la decisión de ejecutar el Gateway en un solo procesador, en lugar de repartir la ejecución entre varios núcleos.

Conociendo la frecuencia de reloj de un procesador (en los sistemas UNIX, `/sys/devices/system/cpu/cpu0/cpufreq/cpuinfo_max_freq` contiene la información de la máxima frecuencia que permite una CPU) y su valor de TSC, podemos realizar una medida de tiempo haciendo dos llamadas a la función: una inicial para obtener un referencia y otra cada vez que llegue un `ModeAdaptationPacket`. Restando la muestra actual con la de referencia, sabemos los ciclos de reloj que llevamos; esto dividido entre la frecuencia del procesador, nos da el tiempo absoluto. Por último, hay que transformar este tiempo en múltiplos de T_{elem} , simplemente realizando una división más, obtenemos el ISCR.

$$ISCR = \text{mod}_{2^{22}} \left(\frac{n_{actual} - n_{referencia}}{f_{cpu}} * \frac{1}{T_{elem}} \right)$$

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

d) Otros

Además de todo lo mencionado anteriormente, se han realizado diversas tareas en toda la plataforma de encapsulación. Todo el Mode Adaptation ha sido modificado para dar soporte al modo de Alta Eficiencia o HEM, bloque a bloque.

Entre la interfaz T2-MI y el Stream Adaptation, se ha creado un nuevo tipo de objeto llamado T2-Frame, que contiene un vector de BBFrames de datos útiles, además de la señalización L1-Actual y L1-Future si tenemos activado el Repetition. El T2-MI Interface también implementa un listener de T2-Frames para poder recibir correctamente en su cola de entrada lo que le envíe el Scheduler.

Además, todos los paquetes T2-MI han sido implementados. De estos, destacar todos los relacionados con la gestión de una SFN, que son enviados al principio de un archivo T2-MI para definir una posible red de transmisores.

4.2. Analizador T2-MI

Se ha escogido el uso del lenguaje C para la implementación del Analizador. El principal motivo es la facilidad del lenguaje para realizar operaciones a nivel de bit, imprescindibles para decodificar y dar sentido a la información transportada en el T2-MI, y la rapidez que brinda a la hora de ejecutar.

Este programa es mucho menos complejo que el Gateway, por lo que no es necesario el uso de un lenguaje orientado a objetos, como C++. De la misma manera que la plataforma de encapsulación, se ha trabajado sobre Ubuntu 12.04. Se ha utilizado el editor de textos por defecto de Gnome, “gedit” para la confección de código, y para compilar se ha usado GCC 4.7.1.

El procedimiento utilizado ha sido secuencial: una a una, los estados descritos en la [ilustración 18](#) se han ido implementado uno detrás de otro, puesto que para llegar a un estado era necesario tener implementado todos los anteriores.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Se ha seguido fielmente lo explicado en el apartado de diseño para la decodificación. Desde esta perspectiva, el Analizador se puede desglosar en tres partes diferenciadas: Inicio del programa, lectura y decodificación del paquete TS y lectura y decodificación del paquete T2-MI y salida.

a) Inicio del programa

Cuando se ejecuta el programa se le pasan dos parámetros, el primero de ellos es el archivo .ts que tiene que analizar, y el segundo es la cantidad máxima de paquetes a decodificar, antes de acabar su ejecución.

Lo primero que hace el Analizador al comenzar es inicializar todos los contadores que vamos a necesitar para comprobar la coherencia del flujo de paquetes TS y T2-MI. En concreto, usamos cuatro contadores: ContinuityCounter, que asegura la continuidad de los paquetes TS; T2MIPacketCounter que lleva la cuenta de los paquetes T2-MI y lo comparará con un campo de la cabecera del paquete T2MI; FrameIDXCounter que controla el número de T2-Frames que llevamos analizado dentro de una Superframe; y por último, SuperFrameCounter que es un contador de Superframes.

A continuación, se comprueba que, en efecto, el archivo a analizar existe y puede ser abierto. Si no es así, el programa indica por pantalla de que el fichero no existe o no se encuentra en el mismo directorio y detiene su ejecución. Una vez abierto, el puntero que controlará que byte estamos leyendo es dirigido al comienzo del fichero y leemos el primer byte, que al ser un Transport Stream, debería ser 0x47. Si no lo es, el fichero de entrada o tiene algún error o no es un TS, por lo que el programa se interrumpe.

Si el primer byte es el de sincronización, comienza el bucle de lectura de los paquetes TS. Volcamos en un vector de 188 bytes el paquete TS, y el siguiente bloque analizará sus contenidos.

b) Lectura y decodificación del paquete TS

Para la lectura de los bits del archivo, se han implementado tres funciones muy similares, llamadas `GetBits`, `GetBits2` y `GetBitsX`. El motivo de implementar estas funciones se debe a que el lenguaje C sólo permite como unidad mínima de lectura 1 byte (8 bits), pero muchos de los campos que van a componer las cabeceras a analizar no son múltiplos de 8 bits (van a estar fragmentados en distintos bytes). Para facilitar la extracción de un número aleatorio de bits sin importar la cantidad de bytes en los que estén fragmentados, a todas las funciones anteriores se les pasa como parámetro el número de bits y el vector de bytes del cual extraer, y devuelve el entero que formarían esos bits.

Respecto a la función `GetBits`, está gobernada por dos punteros: `Offset`, que indica la posición del byte dentro del fichero y se incrementa automáticamente al extraer un determinado número de bits; y `Marker`, que indica, dentro de un byte, cual estamos leyendo. En el caso de que no nos interese variar este puntero para no perder la posición de lectura, la función auxiliar `GetBits2` utiliza un puntero distinto, llamado `Offset2` y `Marker2`, que al leer bits no altera las variables de `GetBits`. Hay un caso especial, de algunos valores que pueden tomar valores positivos y negativos, para este caso, utilizamos `GetBitsX`, gobernada por `Offset`, que para calcular el entero, hace la conversión de binario a decimal en complemento a 2.

Puede ocurrir que una cabecera está distribuida en dos paquetes TS distintos. `GetBits` está preparada para esta adversidad y salta de un paquete a otro para poder decodificar correctamente las cabeceras, utilizando un vector de 188 bytes auxiliar.

Respecto al hilo de ejecución del Analizador, se comienza inicialmente leyendo las cabeceras de un paquete TS, decodificando los valores de estos 4 bytes. De estos 32 bits, el más importante corresponde al número 10, Payload Indicator. Este bit indica cuando vale 1 si en este paquete va encapsulado el comienzo de otro (en nuestro caso, será un paquete T2-MI); en el caso contrario, será la continuación de otro anterior, cuyo tamaño habrá sido mayor de 183 bytes.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

En el caso de que si empiece un nuevo paquete T2-MI, sí que tendremos que decodificar el contenido que venga a continuación de la cabecera TS. En el caso contrario y sea continuación de uno fragmentado, pasaremos directamente a decodificar el siguiente paquete TS. Por último, es necesario incrementar el Continuity Counter al finalizar una iteración del bucle.

Antes de pasar ya a la cabecera del T2-MI, mostramos por pantalla el resto de la cabecera TS y su significado. En nuestro caso, la mayoría de los campos siempre tendrán un valor constante durante todo el flujo, que nos servirá para identificar algún problema en la generación del Transport Stream.

A continuación, empezamos a decodificar la primera cabecera del paquete T2-MI, para identificar qué tipo de paquete tenemos a continuación. El Analizador puede identificar estos paquetes:

- Baseband Frame, si el valor de Packet type es de 0x00.
- L1-Current, en el caso de que Packet Type sea igual a 0x10.
- L1-Future, para las situaciones en las que Packet Type valga 0x11.
- DVB-T2 Timestamp, para los T2-MI con 0x20 de Packet Type.
- Individual addressing, para Packet Type = 0x21.
- FEF part: Null, con Packet Type igual a 0x30.
- FEF part: I/Q Data, para un Packet Type de 0x31.

En función del tipo de paquete T2-MI, utilizando una sentencia Switch-Case, se saltará a la función adecuada que decodifique el paquete. Cualquier otro valor de Packet Type será considerado erróneo y detendrá la ejecución del Analizador.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

c) Lectura y decodificación del paquete T2-MI

Se ha implementado una función para cada tipo de paquete T2-MI, para mantener el código limpio y validar por separado cada parte de la decodificación.

La primera función implementada fue AnalizaBBFrames. Como su nombre indica, el Analizador extrae la información de la cabecera del T2-MI Payload específico para las BBFrame, junto a los contenidos del BBHeader.

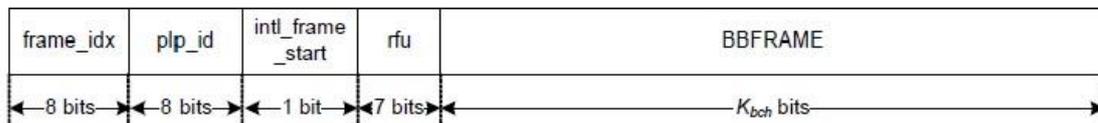


Ilustración 25: Formato del paquete T2-MI BBFrame [DVB Spec]

El Analizador comienza decodificando *frame_idx* y lo compara con FrameIDXCOUNTER para comprobar la continuidad de los T2-Frame. Una vez corroborada la continuidad del flujo, sacamos el resto de valores de la cabecera y los mostramos por pantalla. A continuación pasamos a decodificar los primeros contenidos de la BBFrame: el BBHeader.

A la hora de analizar los campos que forman el BBHeader surge un problema: Según el modo de funcionamiento, Normal Mode o HEM, la información que compone la cabecera variará. Para poder saber qué modo estamos usando, hay que calcular el CRC8 de los primeros 9 bytes que componen la cabecera y hacer un XOR con 0x00 en para Normal Mode y 0x01 para HEM. Se ha escogido, para calcular el CRC, usar un vector con los 256 (2^8) valores posibles que puede tomar esta redundancia. Para poder extraer la cabecera, nos podemos encontrar el problema de que esté fragmentada entre distintos paquetes TS. Para ello, se ha creado una función auxiliar, llamada fillBBHeader, que utilizando GetBits2, llena un vector de 9 bytes con el BBHeader, sin alterar los punteros principales, para después calcular el CRC.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Cuando tenemos la certeza del modo que cierto PLP está usando, a continuación extraemos el resto de la información del BBHeader y la mostramos por pantalla. Después del BBHeader ya viene la información útil del BBFrame, principalmente contenido audiovisual. El Analizador no es capaz de interpretar la carga útil de un PLP, por lo que al finalizar el procesado de la cabecera, volvemos al bucle de lectura de paquetes TS.

Las siguientes dos funciones desarrolladas para en el Analizador son AnalizaL1Current y AnalizaL1Future, ambas muy similares. La finalidad es extraer toda la información de señalización L1 que portan estos paquetes. Estas funciones recorren toda la señalización y muestran al usuario el significado de los bits. Dentro de un flujo T2-MI, la señalización viaja justo cuando acaban todos los BBFrame de una T2-Frame, por lo que cuando el Analizador detecta un paquete tipo L1-Current, sabe que la T2-Frame ha terminado y actualiza el contador de FrameIDXCounter.

Lo más complejo a la hora de implementar estas funciones ha sido dar soporte a tamaños de campo variables, puesto que el tamaño del L1-Configurable y L1-Dynamic variará en función, del número de PLPs, portadoras de RF, Auxiliary Streams o FEFs que el Gateway haya introducido. La manera escogida para este análisis es extraer y asignar los valores que van a definir los bucles de lectura de los campos en variables. En concreto, las variables hacen referencia a: número de PLPs, número de portadoras de RF, número de Streams auxiliares y cantidad de FEFs.

Los paquetes L1-Current llevan información sobre el P1-Signalling, el L1-Presignalling, L1-Configurable y L1-Dynamic. Sin embargo, los paquetes L1-Future transportan la señalización sobre el L1-Dynamic de la siguiente trama T2 y el L1-Inband respecto a la siguiente Interleaving Frame. El uso de L1-Future no es mandatorio y su uso depende del Gateway, por lo que es posible que en algunos flujos T2-MI no lo encontremos.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

El siguiente método implementado es `AnalizaTimestamp`. Su objetivo es obtener la $T_{emission}$ que transporta el paquete DVB-T2 Timestamp. La función identifica cual es el modo de Timestamp (Que puede ser absoluto, relativo o nulo) y muestra por pantalla su valor decodificado, pasado a un formato de año, mes, día, hora, minuto y milisegundos para modo absoluto. En modo relativo, solo se muestran los milisegundos y en modo nulo, no se muestra nada. DVB-T2 Timestamp también informa del ancho de banda del sistema, y el Analizador de aquí extrae el $T_{elemental}$ para el cálculo e interpretación del ISSY.

A continuación, el Analizador implementa la función `AnalizaIndividual`, cuya motivación es extraer todos los paquetes de `IndividualAdressing`, referidos a la gestión de una SFN. Esta función es capaz de decodificar los comandos de gestión introducidos en DVB-T (Time Offset, Freq Offset, Tx Power, Private Data y Cell ID) como aquellos incluidos en el estándar DVB-T2 (ACE-PAPR, TR-PAPR, L1ACE-PAPR, MISO Group, TX-SIG FEF y TX-SIG ID).

Este paquete también tiene un tamaño variable, en función de la cantidad de transmisores que tengamos en una red. Para la decodificación, se han programado dos bucles, el primero que recorre todos los transmisores, y dentro de estos, otro bucle que recorre las funciones dirigidas a ese transmisor.

El estándar no define expresamente donde van ubicados los paquetes de `Individual Adressing` dentro de un T2-MI (pueden llegar en cualquier momento). En nuestro caso, se ha escogido que se envíe al principio del flujo, una sola vez, la información sobre SFN.

Por último, el Analizador puede decodificar las cabeceras de dos de los paquetes relacionados con las Future Extension Frames: `AnalizaFEFNull` interpreta los paquetes FEF Part:Null y `AnalizaFEFIQ` decodifica los paquetes FEF Part:I/Q. Ambos paquetes tienen información, en los campos S1 y S2, sobre la modulación, el tamaño de la FFT y el intervalo de guarda, necesarios para que un receptor pueda demodularlos.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

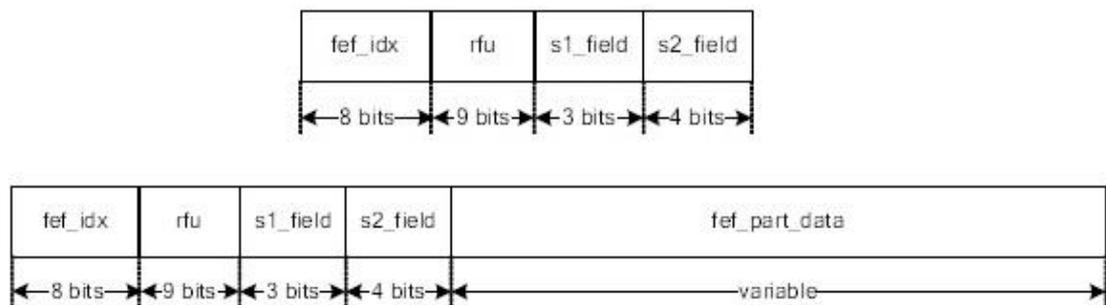


Ilustración 26: Anatomía del paquete FEF Part:Null (Arriba) y FEF Part:I/Q (abajo) [T2-MI Spec]

Como se muestra en la ilustración, para controlar la continuidad de los FEF es necesaria comprobar el *fef_idx* y compararlo con un contador nuevo, llamado en el programa FEFCounter. En el caso de FEF I/Q, la carga variable es información útil, y no es función del Analizador interpretar estos datos.

d) Salida

El Analizador tiene varias condiciones de salida, como se muestra en el diagrama de flujo del Diseño. Si el archivo a analizar no es un Transport Stream, o los contenidos no son un flujo T2-MI, el Analizador avisa por pantalla y termina su ejecución. Otra forma de salir del programa es haber superado el número máximo de paquetes TS a analizar. Debido a que una simple traza de 2 Superframes puede contener centenares de miles de paquetes TS, no era práctico un volcado en un archivo que fácilmente puede ocupar Gigabytes de memoria. Por este motivo se limita la cantidad de información a procesar.

Por otro lado, el Analizador se apoya en los comandos de consola de UNIX para especificar donde mostrar la información decodificada. Si no redirigimos la salida estándar, por defecto la información se mostrará en el terminal que tengamos abierto. Si por el contrario volcamos la traza interpretada en un fichero, abriendo este archivo con cualquier editor de textos podemos fácilmente acceder a la información que nos interese, realizando búsquedas de palabras clave.

5. Validación del Gateway-T2 y del Analizador T2-MI

5.1. Gateway-T2

Para la validación del Gateway, se han utilizado dos mecanismos principales: El Analizador de flujos T2-MI y los Streams de referencia de V&V [7] (Verification and Validation) proporcionados por DVB en su página web. Secundariamente, para la depuración del código se ha hecho uso de mensajes mostrados por pantalla en la misma pasarela, para controlar el valor que tomaban las variables en distintos momentos de la ejecución.

El procedimiento utilizado para validar mediante el Analizador se ha basado en probar una a una todas las funcionalidades que proporciona el Gateway y comprobar que realmente, que la tarea final del Gateway, generar el T2-MI, se realizaba correctamente. Al hacer las pruebas, la plataforma volcaba la configuración inicial introducida en la interfaz gráfica, para luego ser comparada con la información extraída mediante el Analizador. En concreto, se han validado las siguientes funcionalidades:

- Número de BBFrames por T2-Frame.
- Normal Mode y High Efficiency Mode para la etapa de Mode Adaptation.
- Inserción de ISSY, DNP y CRC8 para el Mode Adaptation.
- Señalización L1, tanto los símbolos P1, el L1-Presignalling, Configurable y Dynamic.
- L1 Repetition, señalización L1 de la T2-Frame siguiente.
- L1 Inband, señalización especial que es transportada al final del primer BBFrame de una Interleaving Frame para cierto PLP.
- Generación de Interleaving Frames.
- Generación del DVB-T2 Timestamp, en modo Null, Relative y Absolute.
- Generación de FEF Null.

Debido a que el Analizador no puede interpretar la carga útil de las BBFrame, no ha sido posible validar si el Gateway es totalmente funcional. Para poder comprobar la carga útil, se ha utilizado los Test Streams de V&V.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Los Streams de referencia son un conjunto de trazas, en distintos puntos de un modulador T2, llamados Test Points, para poder validar no solo una plataforma de encapsulación sino también otros dispositivos de la cadena de transmisión.

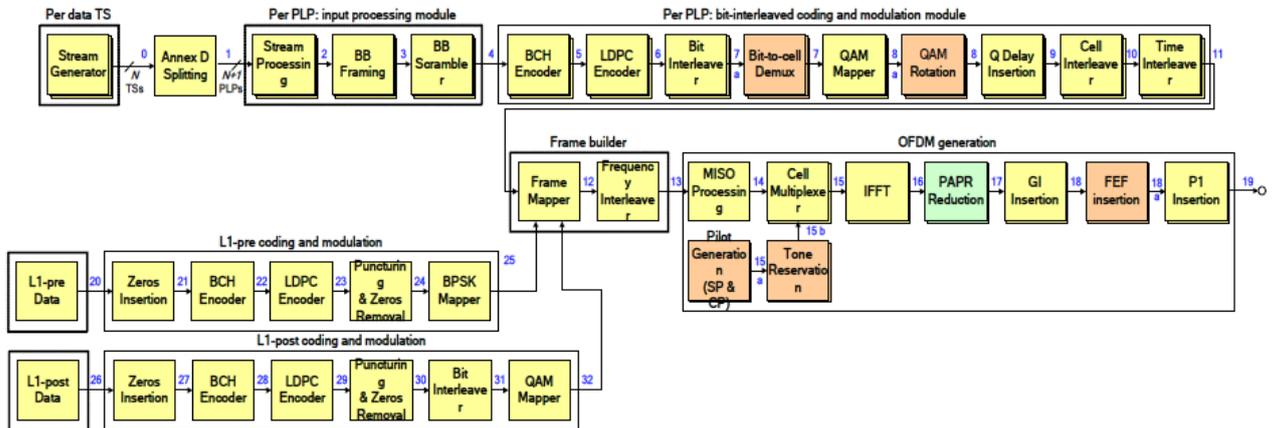


Ilustración 27: Bloque del modulador T2 y ubicación de los Test Points [Reference Stream]

Los Test Points relevantes a la hora de validar el Gateway son el 3, que corresponde a las BBFrame; y el 20 junto al 26, que son las trazas de la Señalización L1. La carga útil que llevan los PLP es una secuencia pseudoaleatoria, generada por un registro de desplazamiento cuyo polinomio generador es:

$$x^{23} + x^{18} + 1$$

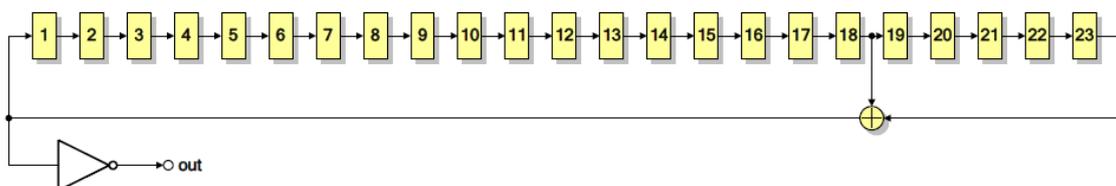


Ilustración 28: Registro de desplazamiento y polinomio generador del Payload. [Reference Stream]

Se han hecho pruebas con cinco Test Streams distintos, el procedimiento que se ha seguido es: se generaba un archivo T2-MI desde el Gateway con los parámetros del Test Stream en cuestión y como entrada el archivo binario generado a partir del registro de desplazamiento (implementado en Matlab). Posteriormente, se le pasaba el fichero por el Analizador, donde se compara los resultados obtenidos con las trazas de los Test Points en cuestión (el 3 para las BBFrame, el 20 y el 26 para la señalización L1).

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Los Streams probados y su parametrización son los siguientes:

V&V	VV001	VV006	VV015	VV019	VV035
T2-Frame per Superframe	2	2	2	2	2
FFTSIZE	32K	16K	8K	32K	16K
Guard Interval	1/128	¼	1/32	1/128	1/8
Data Symbols	59	22	238	59	100
PAPR	No	No	No	No	TR
Null packet deletion	No	No	No	No	No
Bandwidth	8 MHz	8 MHz	8 MHz	8 MHz	8 MHz
Extended Carrier	Si	No	Si	Si	No
Pilot Pattern	PP7	PP1	PP7	PP7	PP3
L1 Modulation	64QAM	64QAM	64QAM	64QAM	64QAM
L1 Repetition	No	No	No	No	No
Nº RFs	1	1	1	1	1
PLP_ID	0	0	0	0	0
Modo	HEM	HEM	HEM	HEM	Normal
Modulation	256QAM	64QAM	256QAM	256QAM	64QAM
Rate	3/5	5/6	3/5	3/5	¾
FEC Type	64800	64800	64800	64800	64800
Constelación rotada	Si	Si	Si	No	Si
Max FEC Blocks	202	50	200	202	119
Frame Interval	1	1	1	1	1
Time Interleaving Type	0	1	0	0	0
Time Interleaving Length	3	2	3	3	3

Tabla 7: Parámetros de los Test Streams usados en la validación del GW [Excel Reference]

Gracias a estos Streams, se ha podido comprobar que el Gateway no modifica la carga útil contenida en los BBFrame, y que la señalización se genera adecuadamente.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

1.2. Analizador T2-MI

A la hora de validar el Analizador (comprobar que cumple la especificación de DVB para T2-MI), se han utilizado distintos Streams de diversas fuentes. En concreto, se han utilizado los Streams de V&V de DVB ya explicados anteriormente, flujos T2-MI generados por un Gateway profesional de DEKTEK y el propio Gateway T2 desarrollado en esta memoria.

Por otro lado, era un requisito que el Analizador estuviera ya validado antes de realizar la validación del Gateway, por lo que la validación de ambos desarrollos se ha llevado conjuntamente: A medida que se añadían funcionalidades al Analizador ya existentes en la plataforma, se comprobaban antes de pasar a la implementación de otras nuevas. Y en el momento que al Gateway se le han añadido funciones, el paso siguiente ha sido implementarlas en el Analizador para poder validar ambos dispositivos.

Se han validado una a una todas las funciones mencionadas en el apartado de Desarrollo, por cada una de las fuentes nombradas. En concreto, el Analizador tiene validado:

- La lectura de cabeceras de un flujo TS y comprobación de la continuidad.
- La lectura de los BBHeader correspondientes a los BBFrame que forman una T2-Frame, tanto para Normal Mode como para HEM.
- La lectura de la señalización L1 del T2Frame actual, del T2Frame siguiente y de la señalización Inband.
- La lectura de los DVB-T2 Timestamp, en todos sus modos posibles.
- La lectura de las cabeceras de los paquetes T2-MI FEFPart:Null y FEFPart:I/Q.

6. Conclusiones y líneas nuevas

6.1. Conclusiones

El objetivo de este Proyecto, el diseño, desarrollo y validación de un Gateway-T2, que implemente el Mode A (Single PLP), acorde con lo especificado en el documento EN 302 755, ha sido completado, además de incluir funcionalidades añadidas.

Para el diseño, se ha apoyado en las arquitecturas de las plataformas de encapsulación que el MCG poseía anteriormente. En base a esto se diseñó un primer prototipo funcional pero muy básico. Sobre diseño inicial se trabajó para desarrollar el resto de funcionalidades.

Respecto al desarrollo, modificamos lo necesario para incluir todas lo especificado de Mode A, junto a una interfaz gráfica más amigable hacia el usuario, dar más importancia al Scheduler, prepararlo para futuras extensiones y resolver todos los problemas que han surgiendo durante la implementación.

Por último, para validar la plataforma, se utilizó el programa expuesto en esta memoria, el Analizador de flujos T2-MI, junto a flujos de prueba distribuidos por el propio DVB, y las herramientas proporcionadas por el entorno de desarrollo.

La plataforma es funcional y podría ser el elemento de cabecera de una red de distribución profesional. Además se encuentra lista para desarrollar nuevas extensiones y funcionalidades, según la aplicación de interés.

La segunda meta, el diseño, desarrollo y validación de un Analizador de archivos T2-MI también ha sido superada. Este programa serviría para atestar el buen funcionamiento de cualquier tipo de Gateway-T2, analizando los flujos T2-MI generados, independientemente del fabricante.

Para realizar el diseño, se ha apoyado en un diagrama de flujo, del comportamiento que tiene que mostrar el Analizador, con las condiciones de salida de la ejecución.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Una vez claro el diseño, y antes de pasar al desarrollo principal, se ha tenido que implementar funciones auxiliares que nos permitieran analizar correctamente los ficheros binarios. A continuación, se ha desarrollado uno a uno todos los estados, con las funciones asociadas.

Posteriormente, a la hora de validar que el Analizador cumplía con lo mostrado en

6.2. Líneas nuevas

Aunque este Proyecto haya finalizado, no implica que la línea de investigación sobre el Gateway o el Analizador este cerrada. Al contrario, se han abierto distintas líneas para los dos dispositivos para ser desarrolladas en el futuro.

Respecto al Gateway, las posibles extensiones en el futuro son: Múltiples PLP, Multiplexación estadística a nivel de PLP, ejecución en tiempo real, bidireccionalidad e interoperabilidad entre tecnologías. Cabe resaltar que no todas podrían ser comenzadas, puesto que existe una dependencia entre ellas (Algoritmos de Scheduling requiere tener múltiples PLP, bidireccionalidad sin tiempo real no tiene sentido).

a) Implementación de Mode B (M-PLP).

El objetivo de este de Proyecto es implementar un Gateway-T2 que soporte Mode A (Single PLP). Sin embargo, existe otro modo posible de funcionamiento del Gateway, aquel que otorga la flexibilidad y la capacidad de agregar distintos servicios de difusión en el mismo sistema: Mode B, o múltiples PLP.

Un Gateway que soporte múltiples PLP, puede agregar en el mismo sistema distintos servicios (Alta definición o baja definición) o escenarios (Recepción fija o recepción móvil), en función de las necesidades de cada uno.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI



Ilustración 29: Ejemplo de múltiples PLP con su servicio asociado [TS 102 831]

No todos los PLP que estén presentes en un sistema tienen que ser de datos, existe un tipo de PLP, llamado PLP común (Common PLP) que no llevará información útil, sino señalización de capa 2 (L2 o PSI/SI). Si nuestra entrada es una aglomeración de Transport Streams multiplexados, para usar un PLP común tendremos que realizar un procesamiento previo, donde extraeremos las tablas de señalización que coincidan simultáneamente en todos ellos, lo sustituimos por un Null Packet e insertamos las tablas en un TS auxiliar de control, que viajará dentro del PLP común. Dentro de una T2-Frame, los PLP comunes viajan justo a continuación de la señalización L1.

Es decir, en lugar de repetir N veces (N es el número de Transport Streams empleado) todas las tablas, solo la enviamos una vez por el PLP común. Si junto al uso de un Common PLP, activamos el bloque de NullPacketDeletion, donde se extraen los Null Packet, experimentaremos una ganancia en la capacidad del sistema, al suprimir los paquetes vacíos que antes ocupaba la señalización de nivel 2.

Respecto a los PLP de datos, también pueden ser distintos, según estén fragmentados y entrelazados temporalmente en la T2-Frame o no: Type 1 o Type 2.

P1	L1 Signalling	Common PLP	Type 1 PLP s	Type 2 PLP 1	Type 2 PLP 2	Type 2 PLP 3	PLP 1	PLP 2	PLP 3
----	---------------	------------	--------------	--------------	--------------	--------------	-------	-------	-------

Ilustración 30: Ejemplo de una T2-Frame con PLP Tipo 1, PLP común y tres PLP Tipo 2.

Los PLP Type 1 van a continuación de los Common PLP, y todas sus BBFrame se encuentran ubicadas consecutivamente. Esta opción resulta interesante para un receptor que quiera ahorrar energía, porque al estar sintonizado a un PLP de tipo 1, solo tiene que recibir una pequeña fracción de la trama total, mientras se apaga para el resto.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

En el caso de tener solo un PLP en el sistema (Mode A), el PLP tendrá que ser obligatoriamente de Tipo 1.

Por otro lado, los PLP de tipo 2 van segmentados en “Sub-Slices”. Esto es equivalente a añadir un entrelazado temporal a nivel de celda OFDM, lo cual vuelve la señal más robusta ante los desvanecimientos rápidos que pueda sufrir la señal. El primero de los 2 parámetros fundamentales que define el entrelazado es el SUB_SLICE_INTERVAL, que indica el intervalo de transmisión de información de un mismo PLP. El otro parámetro es el SUB_SLICES_PER_FRAME que indica cuantos Sub-Slices hay en una T2-Frame. Estas variables están contenidas en el L1-Configurable.

Para el Gateway, soportar múltiples PLP implica elevar la complejidad del Scheduler, puesto que es el encargado de asignar las celdas OFDM que componen una T2-Frame a cada PLP. Uno de los problemas que surgen es la asignación de Data Symbols (Símbolos OFDM que transportan datos) a PLPs con distinta configuración, puesto que la equivalencia entre BBFrames y las celdas OFDM cambia, es decir, ocurren problemas de fragmentación entre símbolos. Otra adversidad es el control de las colas de entrada de cada PLP, en el caso más crítico: cuando cada PLP tiene una Interleaving Frame distinta, lo que implica tener un temporizador distinto para cada PLP.

b) Multiplexación estadística a nivel de PLP

Requiere M-PLP. Una línea interesante para desarrollar en la plataforma es la implementación de distintos algoritmos de Scheduling, una comparación entre ellos y observar si se consigue algún tipo de ganancia en capacidad por Multiplexación estadística.

Aunque para el caso de múltiples PLP, el Scheduler ya tiene que realizar una asignación de celdas OFDM, esta se hace de forma estática y no cambia de trama a trama. Estos algoritmos, sin embargo, harían una asignación dinámica, las celdas asignadas a cada PLP variarían entre T2-Frames, en función de un criterio o parámetro

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

(PLPs con prioridad, según la tasa binaria que necesiten, minimizar la fragmentación dentro de una trama...).

Como ejemplo, se propone un algoritmo que asigne una cantidad variable de celdas en función de la ocupación de las colas de entrada de Scheduler. Esta variación en los buffers de entrada se deben a que la codificación de vídeo puede tener VBR (Variable Bit Rate). A los PLP con las colas más llenas, se les dará más recursos en la T2-Frame para vaciar las colas y a PLPs con menos exigencias, mientras se respete en media la tasa exigida, se les puede privar de espacio.

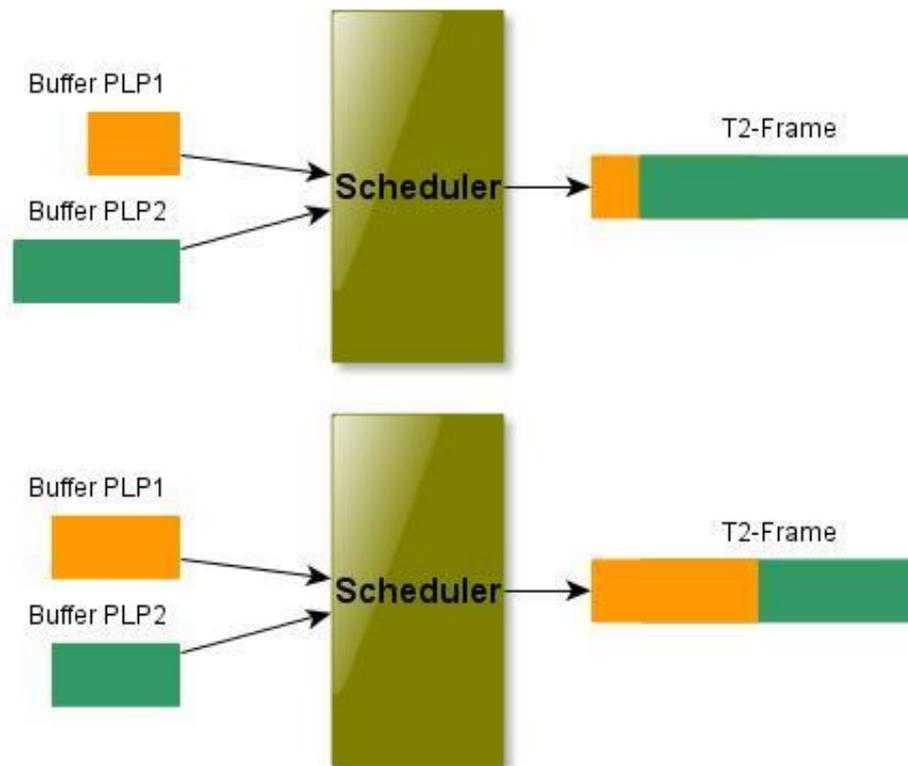


Ilustración 31: Ejemplo de un algoritmo de Scheduling

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

c) Ejecución en tiempo real.

Como se ha mencionado anteriormente, la plataforma de encapsulación se ha diseñado para trabajar con un fichero de entrada y su salida es otro fichero binario.

Una de las grandes mejoras que le se podría proporcionar es soportar la ejecución en tiempo real. En concreto, implementar la ejecución en tiempo real consta de tres partes principales: Entrada en tiempo real, salida en tiempo real y reconfiguración durante la ejecución.

El primero de ellos, entrada en directo, implica capturar un flujo de vídeo (satélite, Internet, una cámara...) y procesarlo en el Gateway. La entrada principalmente vendrá por un interfaz ASI o IP, y es posible que ya venga encapsulado en un Transport Stream o no. El Gateway deberá de ser capaz de reconocer esto y realizar un preprocesamiento del flujo adecuado.

De igual manera, salida en tiempo real, trata sobre el envío por un interfaz ASI o IP de la información audiovisual ya procesada de la entrada, hacia un modulador o una red de moduladores DVB-T2.

Lo último es explotar la capacidad de versatilidad que permite DVB-T2: reconfiguraciones en tiempo real, es decir, añadir o quitar PLPs durante la ejecución del Gateway, o reconfigurar la parametrización de los existentes. También incluimos la capacidad de extraer o añadir más Transport Streams de otras fuentes en este bloque, y la gestión de una SFN en tiempo real.

d) Bidireccionalidad.

Requiere ejecución en tiempo real. El Gateway, como elemento de cabecera, también sea capaz de recibir información de los moduladores de la red o cualquier otro tipo de elemento en la red de distribución. Este concepto engloba tanto la gestión y mantenimiento como recibir flujos de vídeo y encaminarlos en sentido contrario.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Tener un Gateway bidireccional permitiría a DVB-T2 ser un sistema de transmisión dúplex. Es posible implementar un sistema TDM (Time Division Multiplexing), donde la información audiovisual en Downlink se envía en T2-Frames convencionales, y la información en Uplink se envía en los períodos de FEF, donde los moduladores no van a radiar nada.

Otra posibilidad sería tener un sistema FDM (Frequency Division Multiplexing), donde la información de Downlink y Uplink viaja en distintas frecuencias. A un nivel abstracto, se reduce a tener dos Gateway (en el mismo lugar) operativos trabajando a distintas frecuencias, pero que encaminan la información en sentidos contrarios.

e) Interoperabilidad entre tecnologías.

Utilizando Future Extension Frames, implementar un sistema de transmisión híbrido, donde se transmite DVB-T2 en T2-Frames convencionales y otras tecnologías de difusión de contenido (eMBMS de LTE) en el intervalo temporal de los FEFs.

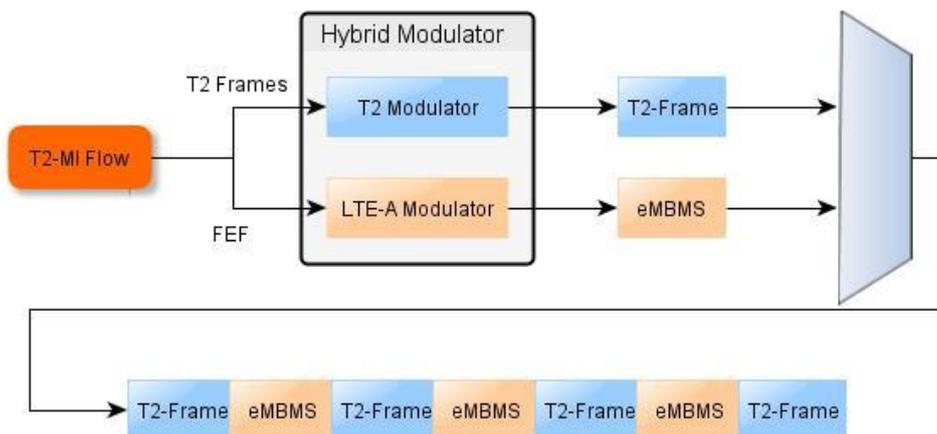


Ilustración 32: Ejemplo de Interoperabilidad entre servicios eMBMS de LTE y DVB-T2

Aunque esta línea va enfocada principalmente a la cooperación entre LTE (4G) y DVB-T2, no se descarta el estudio de interoperabilidad con otras tecnologías (WiMAX...). La suposición principal es la reutilización de emplazamientos ya existentes de TDT o DVB-T2 para enviar información de broadcast de otras tecnologías.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

En el caso de LTE, además de los elementos que forman la red estándar; para enviar eMBMS [8], es necesario añadir varios bloques adicionales, tales como Multicast Coordination Entity (MCE), eMBMS Gateway o Broadcast/Multicast Service Center (BM-SC). En función de cuantos de estos bloques decidamos integrar dentro del Gateway-T2, la plataforma tendrá más control sobre el contenido y la configuración del servicio eMBMS.

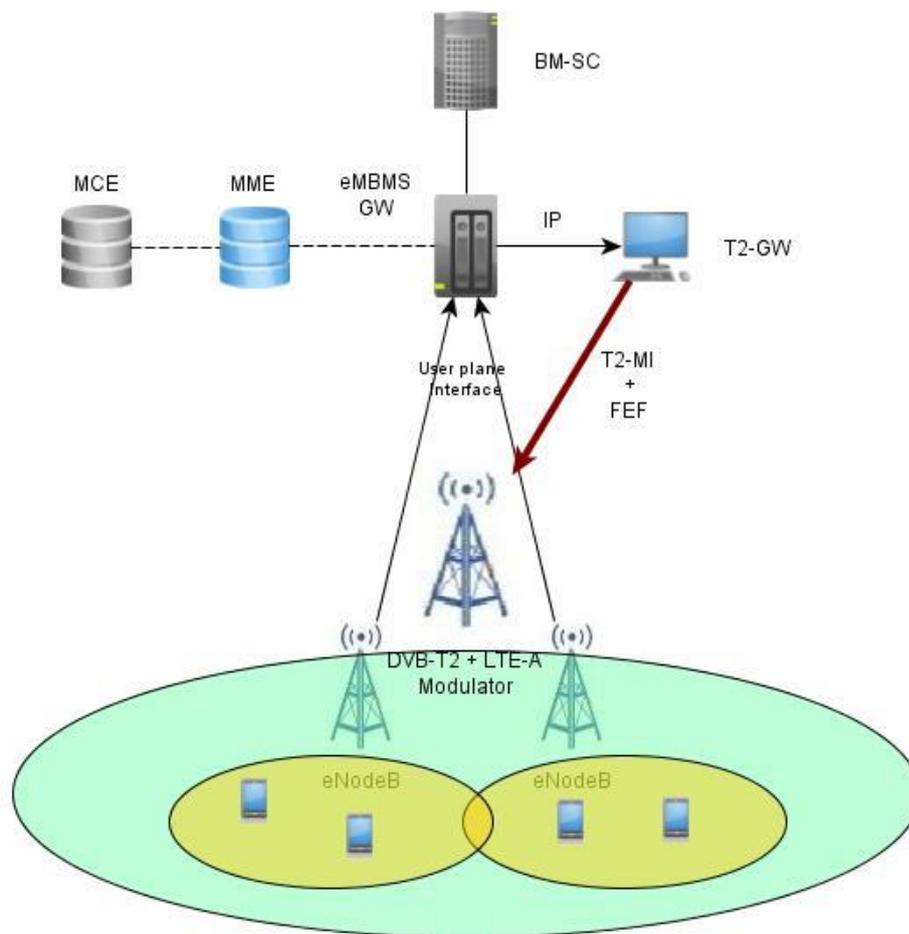


Ilustración 33: Posible solución para la interoperabilidad entre eMBMS y DVB-T2

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Las líneas futuras sobre las cuales mejorar **el Analizador** son: Extracción de PLPs y Gateway inverso.

a) Extracción de PLPs.

El Analizador T2-MI aquí mostrado solamente lee las cabeceras de los BBFrame de cada PLP. Un desarrollo adicional que se podría tomar es volcar la carga útil de cada PLP, quitando todos los campos que introduce el Mode Adaption. Al final obtendremos el contenido de origen, con el que podemos verificar la integridad de la información, pasando este archivo por un visualizador, como por ejemplo el VLC.

Llevando esto un paso más allá, se le podría dar al Analizador funcionalidades de un analizador de Transport Streams, en concreto, separar las tablas de señalización de capa 2 (PSI/SI) de la carga útil, para visualizar más información o separar los flujos elementales que forman un Transport Stream.

b) Gateway Inverso.

Requiere poder extraer PLPs. Mientras que un Gateway convencional pasa de un flujo TS a un archivo T2-MI, un Gateway Inverso, como su nombre indica, pasa de un archivo T2-MI a un archivo TS.

La principal diferencia es que la ejecución debe hacerse en tiempo real, se debe pasar de un flujo T2-MI continuo a un Transport Stream que se encaminará a otro dispositivo de la red (un visualizador, en principio). El primer paso es extraer los PLPs que formen el T2-MI y hacer un gran TS con toda la carga de los PLP. A continuación llevamos este TS a un visualizador donde podamos escoger el canal que deseamos.

La motivación de esta línea es buscar una manera de dar cobertura en zonas donde el despliegue terrestre es incapaz, y la única manera de llegar es mediante un segmento satelital. El satélite es el utilizado en la misma red de distribución utilizado para llevar el T2-MI a los moduladores terrestres, y con una instalación de una parabólica orientada a éste, recibir el T2-MI, pasarlo a TS y ser capaces de visualizarlo.

7. Aplicaciones futuras

En este apartado se detallan dos aplicaciones donde el Gateway y el Analizador, con ciertas funcionalidades que habría que añadir, desempeñará un importante rol.

7.1. Radioenlaces OFDM profesionales para servicios BAS/SAP

Una aplicación que ha utilizado ampliamente DVB-T para transmitir el contenido audiovisual han sido los enlaces de contribución de contenido (desde la fuente hasta el estudio de televisión), conocidos como servicios BAS/SAP.

Formalmente, los servicios BAS/SAP (Broadcast auxiliary to service/ Service Ancillary to Programme) son aquellos que involucran una cobertura de eventos alejados del centro o estudio de producción, lo que implica una transmisión remota de vídeo/audio desde el lugar de interés hasta el estudio. Un buen ejemplo es la cobertura de eventos deportivos en directo, como la vuelta ciclista.

Al no tener un acceso a una infraestructura que permita la conexión cableada entre el lugar donde transcurre el evento a retransmitir, esta transmisión del contenido audiovisual se realiza, habitualmente, utilizando el acceso radio (DVB-T) hasta el nodo repetidor, y de ahí, usar un segmento satelital (DVB-S2) para llegar al centro de producción.

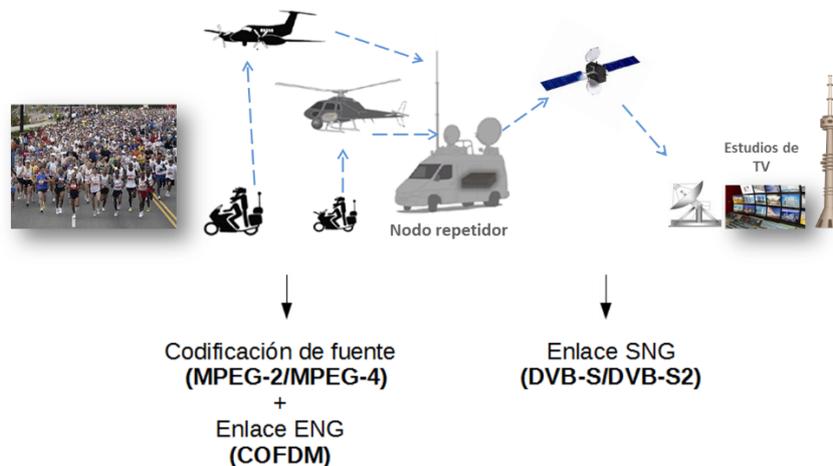


Ilustración 34: Ejemplo de la topología de un servicio BAS/SAP.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

Este escenario es distinto a un despliegue de una red para difusión de vídeo terrestre, por lo que sus requerimientos también son distintos. Los servicios BAS/SAP requieren:

- La robustez y la capacidad están relacionadas con la calidad del contenido recibido. En general, interesa que el contenido que proporcionen las fuentes de audio y vídeo sea de alta calidad con el fin de que sea en el propio centro de producción donde se decida con qué calidad finalmente difundir los contenidos.
- La movilidad es interesante, sobretodo, para la cobertura de eventos que abarquen un área extensa.
- El alcance es la distancia máxima entre el nodo repetidor y las cámaras inalámbricas. Principalmente, viene definida por la potencia de salida de las cámaras, la topología de las antenas utilizadas, la tecnología de modulación empleada y las características del canal radio.
- La cota superior de retardo extremo a extremo, debe de estar acotada.

El mayor inconveniente de esta arquitectura de transmisión son las implicaciones que conlleva el uso del segmento satelital. Por una parte aumentan los costes logísticos y de operación. La necesidad de coordinación técnica y administrativa con los operadores de red para el uso temporal de los segmentos satelitales, la escasez de espectro en eventos de gran interés social unido a los retardos de transmisión extremo a extremo que hacen que disminuya la calidad de experiencia del usuario, hacen necesaria la aparición de nuevas propuestas que solucionen los inconvenientes del uso del satélite.

Con la aparición del nuevo estándar de difusión de vídeo terrestre, DVB-T2, ya es posible prescindir del segmento satelital para llevar la señal hasta el centro de producción, puesto que T2 aporta la capacidad suficiente y robustez suficiente hasta un primer nodo repetidor (Unidad Móvil) y mediante radioenlaces, alcanzar el centro de producción.

Es en el nodo más cercano al centro de producción donde iría ubicado el Gateway-T2 bidireccional. En un sentido, para recibir y hacer un primer procesado del

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

contenido de contribución, recuperar los distintos flujos, agregarlos y enviarlo al centro de producción. En el otro sentido, serviría para reconfigurar el Gateway remotamente desde una estación de control o enviar información sensitiva a otras Unidades Móviles mediante direccionamiento individual u otras extensiones (interesante para cubrir eventos de catástrofes naturales y emergencias).

7.2. Canal de retorno terrestre inalámbrico de DVB-T2, DVB-RCT2

DVB-RCT (Return Channel Terrestrial) [9] es un estándar, publicado en Abril 2001, destinado a proveer un canal de retorno inalámbrico, en las bandas de VHF/UHF, para que usuarios fijos y móviles puedan interactuar.

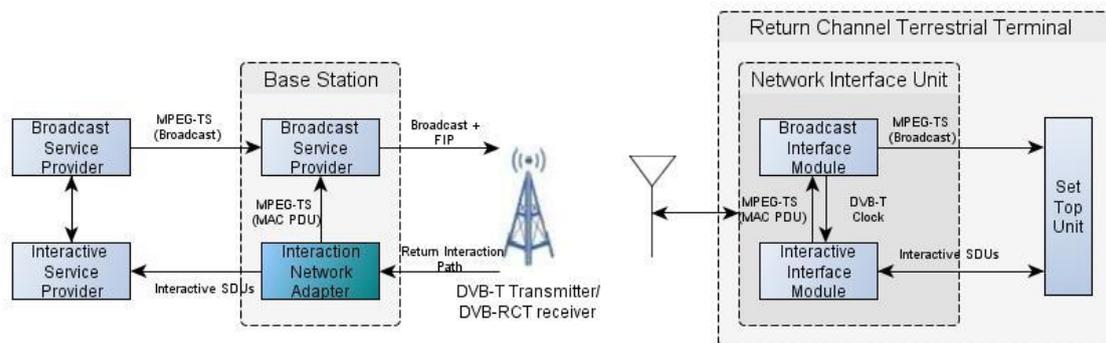


Ilustración 35: Esquema de un sistema DVB-RCT [Spec RCT]

En Downlink, la información destinada a los usuarios va incluida en el propio canal de Broadcast de una arquitectura de DVB-T convencional, encapsulada en paquetes MPEG-2 TS especiales, a este canal se le llama **Forward Interaction Path**.

Sin embargo, en Uplink, se destina una portadora RF específica para el canal de retorno, donde varios usuarios accederán simultáneamente al canal. El acceso al medio utilizado es TDMA/OFDMA, y a este canal se le llama **Return Interaction Path**. Los datos de vuelta no viajan encapsulados en TS sino en celdas ATM. La asignación de recursos a usuarios se realiza en la estación base.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

A pesar de estar desarrollado, DVB-RCT no tuvo un despliegue comercial, aunque se realizaron pruebas piloto; y se abandonó. Hoy en día, con el nuevo estándar de difusión terrestre, DVB-T2, no hay un canal de retorno que asegure la interacción de los usuarios con el contenido. El objetivo de esta aplicación sería desarrollar un sistema de difusión, que incorpore todas las prestaciones de t2, con canal de retorno, mediante un Gateway mejorado que controle ambos sistemas, el de interacción y el de difusión. Al sistema se le denomina DVB-RCT2.

El GW debe, además de todas las funciones ya existentes en T2, controlar toda la capa MAC (Media Access Control) de RCT, es decir, asignar recursos a los usuarios utilizando algún algoritmo de Scheduling.

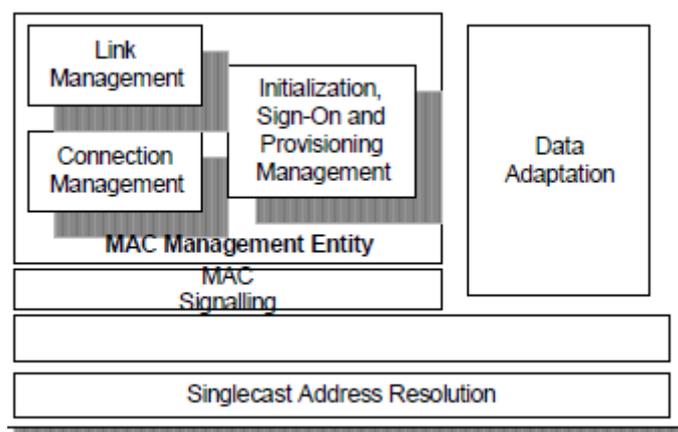


Ilustración 36: Capa MAC de DVB-RCT. Estas funciones las tendrá que implementar el Gateway

8. Bibliografía y referencias

[1].ETSI EN 302 755 V1.3.1: “*Frame structure, channel coding and modulation for a second generation digital terrestrial television broadcasting system, DVB-T2*”. Francia, Abril de 2012.

[2].ETSI TS 101 191 V1.4.1: “*Framing estructura, channel coding and modulation for digital terrestrial television, DVB-T*” Francia, Junio de 2004.

[3] Enensys White Paper: “*Technical overview of Single Frequency Network*” Francia.

[4].ETSI TS 102 773 V1.3.1: “*Modulator Interface (T2-MI) for a second generation digital terrestrial television broadcasting system, DVB-T2*”, Francia, Enero de 2012.

[5].ETSI TS 102 831 V1.2.1: “*Implementation guidelines for a second generation digital terrestrial television broadcasting system, DVB-T2*”, Francia, Agosto de 2012.

[6].Sun Microsystems: “*LinkedBlockingQueue Javadoc documentation*”, <<http://docs.oracle.com/javase/6/docs/api/java/util/concurrent/LinkedBlockingQueue.html>> (Accedido el 1 de Julio de 2013).

[7]. DVB-T2 V&V group: “*DVB-T2 Reference Streams*”, Francia, Agosto de 2011.

[8]. Rohde & Schwarz webinar: “*Evolved MBMS- Broadcast and Multicast in LTE*”, <http://www.rohde-schwarz.com/en/solutions/wireless-communications/lte/lte-webinars/video-evolved-mbms_109118.html> (Accedido el 1 de Julio de 2013).

[9] ETSI EN 301 958 V1.1.1: “*Interaction channel for Digital Terrestrial Television (RCT) incorporating Multiple Access OFDM*”, Francia, Marzo de 2002.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

ANEXO: Manual de uso del Gateway y del Analizador.

Este apartado detalla lo necesario para que cualquier usuario, con unos conocimientos básicos de UNIX y DVB-T2, pueda configurar y utilizar tanto la plataforma como el Analizador.

Manual del Gateway.

El Gateway se configura mediante una interfaz gráfica. Una vez nos hemos asegurado de que la plataforma se encuentra bien compilada y lista para funcionar, el paso siguiente es elegir los parámetros de entrada que deseamos. La interfaz se divide en 4 partes principales:

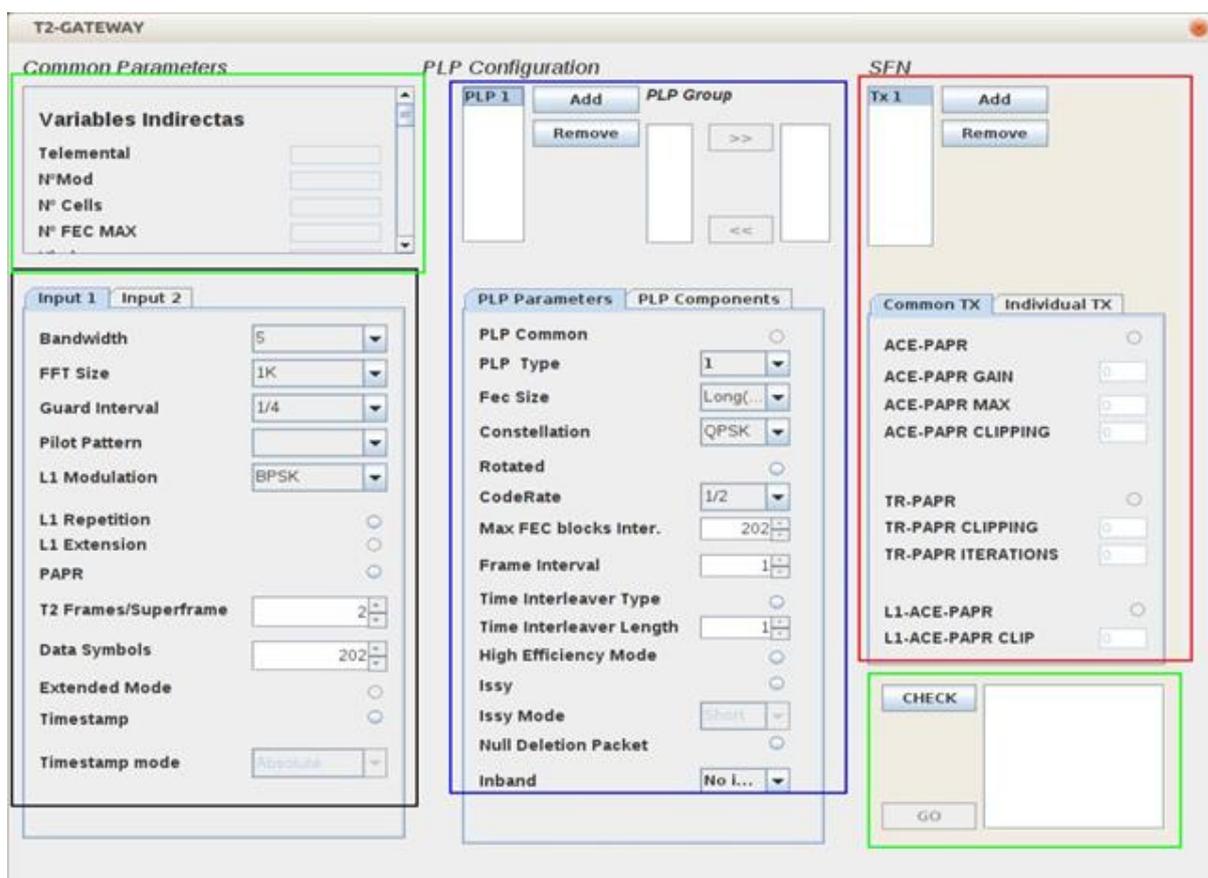


Ilustración 37: Interfaz gráfica del Gateway, con sus partes diferenciadas.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

El rectángulo negro son los parámetros generales del sistema, los cuales definirán el resto de variables y prestaciones del sistema. Aquí es donde elegimos el modo de transmisión que deseamos, y en la pestaña de Input 2 hay más opciones, tales como el uso de FEF.

El siguiente bloque importante es la configuración de los PLP y los componentes (servicios) que encapsularán estos PLPs. Aunque la interfaz soporta múltiples PLP, y agruparlos en grupos, con un PLP común, el resto de la plataforma no tiene implementado M-PLPs, y daría error de ejecución. En la pestaña de “Plp Components” es donde vamos a elegir que programas van a ir en cada PLP.

A continuación, tenemos el bloque de SFN. Esto es opcional, y en principio puede ser ignorado para una transmisión normal. Desde este bloque vamos a configurar la hipotética parametrización de una red remota de transmisores. Por ahora, como la plataforma no soporta tiempo real, este bloque solo ha servido para validar el Analizador T2-MI.

Por último, y antes de pasar a ejecutar el resto del programa, hay que clicar al botón de “Check”, que hará un análisis de los parámetros introducidos y comprobará la coherencia, alertando al usuario de que si hay parámetros con valores inadecuados. Además, la interfaz calcula unas métricas del sistema, tales como Capacidad, duración de la trama... que se mostrarán en la ventana incrustada arriba a la izquierda, en el rectángulo verde. Si todo está correcto, el botón de OK se habilitará, y clickeando sobre él, la interfaz se cierra, pasándole todos los parámetros al resto de la plataforma.

En este momento, la plataforma vuelca en un fichero de texto la configuración introducida, en la misma carpeta donde esté ubicado la plataforma. Esto resulta útil si se quiere validar la señalización L1 generada en el Gateway con el Analizador.

Una vez comienza la ejecución, la plataforma empieza a volcar la información en un flujo T2-MI, encapsulado en un TS, en la misma carpet. El archivo se llama “iteam_t2gateway_dvbtplatform_...” y un número aleatorio, correspondiente a la instancia de objeto de la interfaz, asignado por Java.

Proyecto Final de Carrera

Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI

En principio, el Gateway no tiene una condición de salida, pero para realizar pruebas, se ha limitado la cantidad de Superframes generados a 2, en lugar de estar siempre funcionando o tener que forzar el proceso a detenerse.

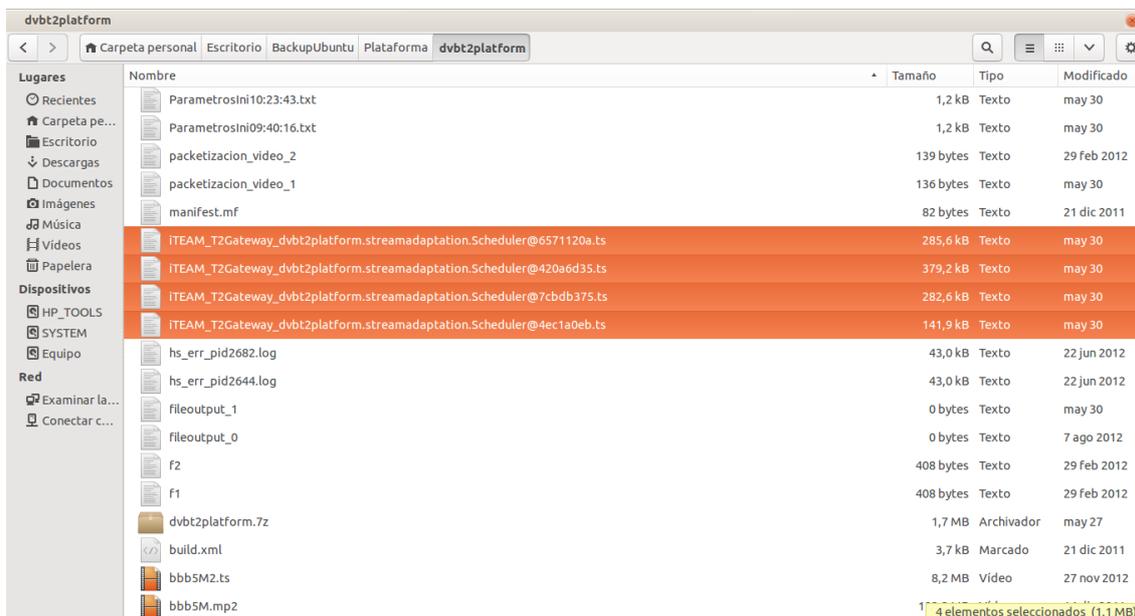


Ilustración 38: Ventana con cuatro archivos T2-MI generados por la Plataforma.

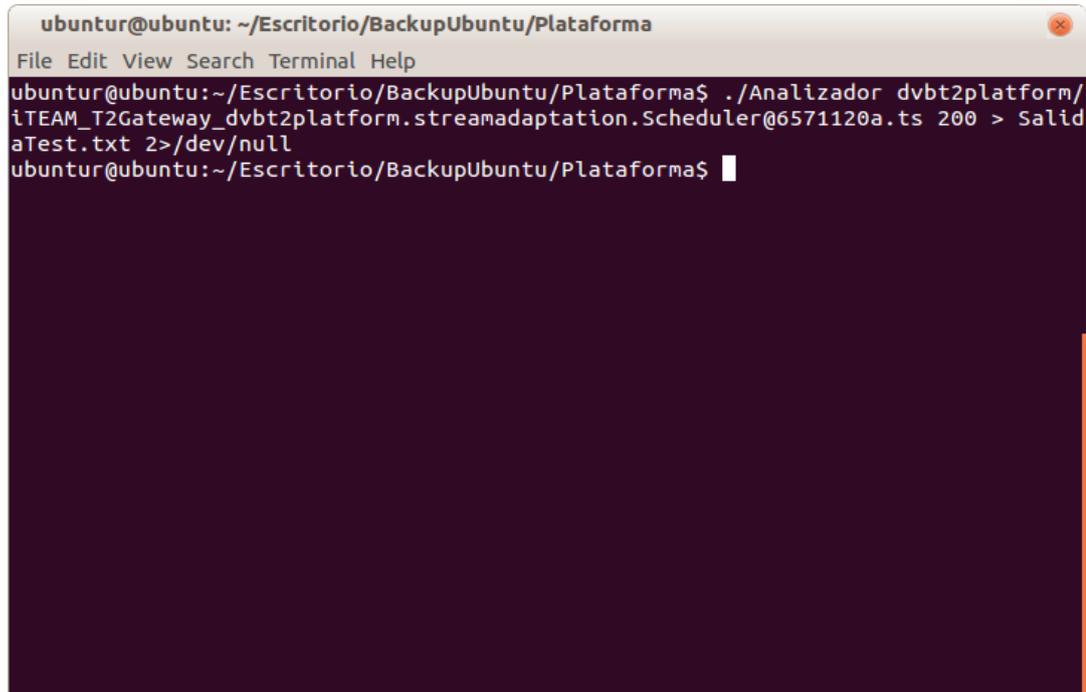
Manual del Analizador T2-MI

Para utilizar el Analizador, primero hay que trabajar en un entorno Linux, ya sea nativo o mediante máquina virtual. Al Analizador hay que pasarle dos parámetros: el fichero a analizar y el número de paquetes TS a leer. La salida tiene que ser redirigida si se quiere volcar en un archivo de texto, por defecto aparece en la consola que tengamos abierta.

A la hora de ejecutarlo, abrimos una consola, navegamos hasta la carpeta donde esté el Analizador e introducimos la siguiente línea: `./Analizador "Ruta del Archivo a leer" "Nº paquetes TS a leer" > "Fichero de salida.txt"`. Como ejemplo:

Proyecto Final de Carrera

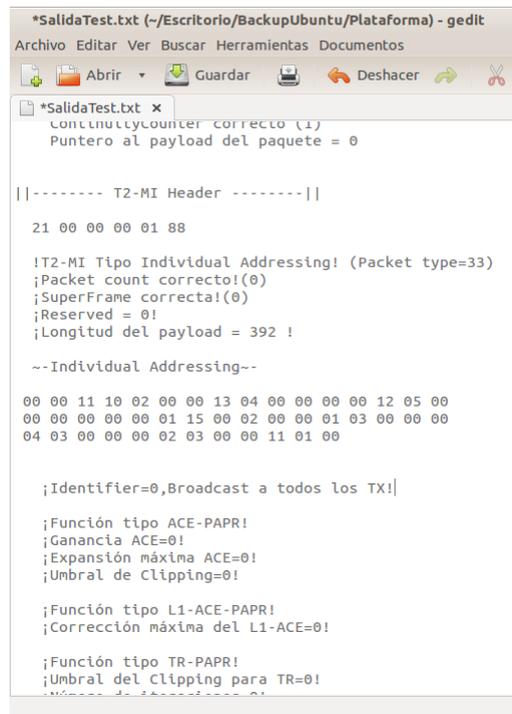
Diseño, desarrollo y validación de un Gateway-T2 y un Analizador T2-MI



```
ubuntur@ubuntu: ~/Escritorio/BackupUbuntu/Plataforma
File Edit View Search Terminal Help
ubuntur@ubuntu:~/Escritorio/BackupUbuntu/Plataforma$ ./Analizador dvbt2platform/
iTEAM_T2Gateway_dvbt2platform.streamadaptation.Scheduler@6571120a.ts 200 > Salida
aTest.txt 2>/dev/null
ubuntur@ubuntu:~/Escritorio/BackupUbuntu/Plataforma$
```

Ilustración 39: Ejemplo de ejecución del Analizador.

Si ahora abrimos el archivo generado con un editor de textos, tal como gedit, podremos leer la información de la traza obtenida.



```
*SalidaTest.txt (~/Escritorio/BackupUbuntu/Plataforma) - gedit
Archivo Editar Ver Buscar Herramientas Documentos
Abrir Guardar Deshacer
*SalidaTest.txt x
continuitycounter correcto (1)
Puntero al payload del paquete = 0

||----- T2-MI Header -----||
21 00 00 00 01 88

!T2-MI Tipo Individual Addressing! (Packet type=33)
;Packet count correcto!(0)
;SuperFrame correcta!(0)
;Reserved = 0!
;Longitud del payload = 392 !

--Individual Addressing--

00 00 11 10 02 00 00 13 04 00 00 00 00 12 05 00
00 00 00 00 00 01 15 00 02 00 00 01 03 00 00 00
04 03 00 00 00 02 03 00 00 11 01 00

;Identifier=0,Broadcast a todos los TX!

;Función tipo ACE-PAPR!
;Ganancia ACE=0!
;Expansión máxima ACE=0!
;Umbral de Clipping=0!

;Función tipo L1-ACE-PAPR!
;Corrección máxima del L1-ACE=0!

;Función tipo TR-PAPR!
;Umbral del Clipping para TR=0!
```

Ilustración 40: Salida del Analizador, abierta con “gedit”