

HERRAMIENTA PARA ANALIZAR EL SERVICIO DE RADIODIFUSIÓN DE DATOS DE LA TVD EN CUBA

Ing. Jorge Rodríguez Rodríguez¹

¹LACETEL, Cuba, Ave. Rancho Boyeros, #34515 (19200), Zona Postal: General Peraza, La Habana
¹e-mail: jorge@lacetel.cu

RESUMEN

Actualmente nuestro país no cuenta con una solución nacional que permita monitorear, registrar y analizar el Servicio de Radiodifusión de Datos transmitido en el actual flujo de TV Digital. El presente trabajo muestra el desarrollo de una herramienta software que permite monitorizar, examinar y estudiar la estructura y los datos del Servicio de Radiodifusión, describiéndose para ello cómo este Servicio se encuentra estructurado y registrando en forma de pseudocódigo los distintos bloques que conforman la herramienta. El software desarrollado fue implementado en la plataforma de desarrollo Microsoft Visual Studio y validada utilizando los software analizadores de tramas MPEG-2 "4T2 Content Analyzer", "TSR 110" y "MPEG TS Utils". La aplicación desarrollada sentará las bases para la futura implementación de esta solución dentro de una caja decodificadora.

PALABRAS CLAVE: TV Digital, Servicio de Radiodifusión de Datos, Trama de Transporte MPEG-2.

ABSTRACT

Currently, on our country a solution that allows to monitor, record and analyze the Data Broadcasting Service transmitted in the current Digital TV stream is not to our disposal. The present work shows the development of a software tool that allows to monitor, examine and study the structure and content of Data Broadcasting Service. The developed software is described by means of pseudocode, which permits to illustrate better the comprising blocks. The developed software was implemented on Microsoft Visual Studio development platform and validated using the MPEG-2 frame analyzer software's "4T2 Content Analyzer", "TSR 110" and "MPEG TS Utils". The application developed will establish the foundation to future implementation of this solution on a set-top box.

KEYWORDS: Digital TV, Data Broadcasting Service, MPEG-2 Transport Stream.

INTRODUCCIÓN

LACETEL[®], Instituto de Investigación y Desarrollo de Telecomunicaciones, es una de las entidades vinculadas al despliegue de la TV Digital en Cuba. Específicamente, es la entidad encargada de asimilar las nuevas tecnologías que este proceso comprende. En países como el nuestro, la asimilación tecnológica es realizada en forma de Transferencia de Tecnología, pues la adquisición de esta es más efectiva y rápida cuando es realizada mediante transferencia. Principalmente en el sector de las telecomunicaciones, donde el costo de la I+D es elevado y el obtener una solución requiere de un largo período de tiempo, así como por el corto plazo de obsolescencia que puede tener la misma. Además, **LACETEL**[®] es la entidad encargada de la evaluación técnica de todos los receptores digitales que entran al país para su comercialización.

Una de las bondades que la TV Digital brinda es la posibilidad de transmitir datos dentro el mismo flujo (Trama de Transporte MPEG) que transporta las señales de audio y video de los programas televisivos, permitiendo así enriquecer el contenido que los usuarios reciben en sus hogares, además de hacerlos partícipes en tiempo real en diversos programas cuando el sistema consta de un canal de retorno. Entre los datos que se transmiten podemos encontrar noticias nacionales e internacionales, el estado del tiempo para distintas regiones, así como ofertas laborales. También puede ser usado como canal de alerta temprana ante la ocurrencia de desastres naturales, o para brindar aquella información que precise de inmediatez en su difusión. Además, este servicio pudiera emplearse como soporte a la enseñanza en los distintos niveles educacionales, para la educación a distancia o el aprendizaje autodidacta, influyendo ventajosamente en el proceso de informatización de nuestra sociedad.

El estándar de TV Digital ATSC (**A**dvanced **T**elevision **S**ystems **C**ommittee), creado por el grupo de trabajo ATSC, desarrolló la plataforma DASE (**D**TV **A**pplication **S**oftware **E**nviroment) para el diseño de aplicaciones interactivas que funcionen en los receptores digitales. Según [1] la arquitectura de la plataforma DASE se basa en 8 partes:

- **Motor de Ejecución de Aplicación** o AEE (**A**pplication **E**xecution **E**ngine): permite la integración entre los diferentes módulos softwares, permitiendo la interacción entre ellos.
- **Administrador de Aplicación** o AM (**A**pplication **M**anagement): permite administrar las distintas aplicaciones del sistema.
- **Decodificador de Contenido** o CD (**C**ontent **D**ecoder): encargado de decodificar el contenido de la aplicación a un determinado formato.
- **Motor de Presentación** o PE (**P**resentation **E**ngine): es un tipo de CD específico para contenido HTML (**H**yper**T**ext **M**arkup **L**anguage), el cual puede mostrar resultados de otros CD en su propia pantalla de presentación.
- **Administrador de Presentación** o PM (**P**resentation **M**anagement): provee las distintas funciones de presentación del contenido de las aplicaciones.
- **Contenedor de Presentación** o PC (**P**resentation **C**ontainer): objeto que muestra elementos en pantalla y mantiene el orden de presentación de los mismos.
- **Administrador de Recursos** o RM (**R**esource **M**anagement): módulo que asigna o remueve los recursos del receptor entre las aplicaciones, por ejemplo, almacenamiento interno, sintonizador(es) de canal, etc.

- **Administrador de Evento** o EM (**Event Management**): notifica a las distintas aplicaciones en ejecución sobre la ocurrencia de eventos asociados a estas.

En el caso del estándar europeo DVB-T (**Digital Video Broadcasting-Terrestrial**), los servicios de interactividad son implementados utilizando MHP (**Multimedia Home Platform**), cuya arquitectura según [2] se compone de 3 módulos fundamentales:

- **Recursos**: como su nombre indica, es el módulo encargado de interactuar con los recursos hardware y software del equipo.
- **Software de sistema**: encargado de administrar los recursos entre las distintas aplicaciones.
- **Aplicaciones**: implementan servicios interactivos como softwares en ejecución para una o varias entidades hardware.

El estándar brasileño ISDB-Tb (**Integrated Services of Digital Broadcasting – Terrestrial brazilian**) utiliza el middleware Ginga, que según expresa [3], su arquitectura se compone de dos capas fundamentales, el Motor de Presentación y el Núcleo Común. El primero interpreta los tiempos y eventos a presentar, mientras que el segundo realiza una traducción de los comandos de alto nivel a otros de bajo nivel que permitan manejar los recursos del sistema. Ginga definió NCL (**Nested Context Language**) [4] como lenguaje de programación para sus aplicaciones interactivas.

En el caso de nuestro país, que utiliza la norma China de TV Digital DTMB (**Digital Terrestrial Multimedia Broadcast**), este servicio es implementado según lo especificado en [5], donde los datos son estructurados y encapsulados en archivos XML (**eXtensible Language Markup**) [6], para su posterior transmisión. Sin embargo, las entidades encargadas de transmitir la señal digital y de generar, entre otros, los contenidos asociados al Servicio de Radiodifusión de Datos, no cuentan con una herramienta que les permita monitorizar y analizar el correcto funcionamiento del mismo.

Partiendo del resultado de estudios y pruebas que han sido realizadas a este servicio en nuestro instituto, se plantea como objetivo del presente trabajo el desarrollo de una solución software que permita analizar y detectar fallas asociadas al mismo. El desarrollo de esta herramienta será el punto de partida para su posterior implementación en una caja decodificadora, lo cual permitirá el análisis en tiempo real de este servicio.

SERVICIO DE RADIODIFUSIÓN DE DATOS

Como su nombre lo indica, el Servicio de Radiodifusión de Datos no es más que la transmisión de datos (texto, figuras, imágenes fijas, sonidos, etc.) sobre el mismo multiplex que transporta las señales de audio y video (Figura 1). Los datos de este servicio son almacenados temporalmente en el receptor digital y es el usuario quien selecciona a cuál información acceder. Desde hace unos años, el Servicio de Radiodifusión de Datos está siendo ampliamente fomentado en todo el mundo debido a la digitalización de las transmisiones terrestres y satelitales.

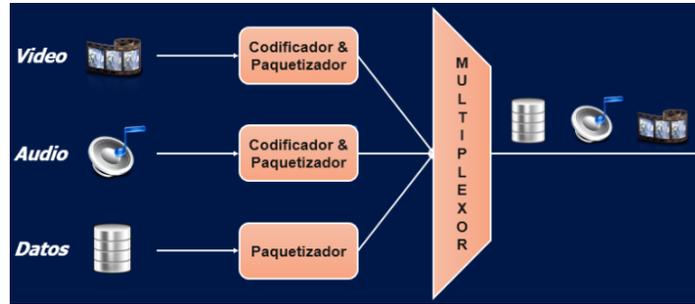


Figura 1. Multiplexación del video, audio y datos.

Este servicio puede ser clasificado, desde el punto de vista del tipo de programa, en Radiodifusión de Datos Independiente y Radiodifusión de Datos Dependiente. La Radiodifusión de Datos Independiente no es más que la transmisión exclusiva de datos como, por ejemplo, el estado del tiempo y el tráfico aeroportuario. Por otra parte, la Radiodifusión de Datos Dependiente es aquella en la cual los datos transmitidos sirven de complemento a la transmisión de determinado programa, como puede ser las estadísticas de un juego de béisbol o de futbol. La Figura 2 muestra ambas clasificaciones.

El Servicio de Radiodifusión de Datos puede ser clasificado en interactividad local o remota. Como interactividad local se entiende toda aquella que ocurre directamente entre la caja decodificadora y el usuario utilizando la información almacenada en ella, como, por ejemplo, la Guía Electrónica de Programa, mientras que la interactividad remota consiste en el intercambio continuo de información entre el usuario y los proveedores de servicios, utilizando para ello un canal de retorno, generalmente la red IP, como por ejemplo, la respuesta en tiempo real a preguntas hechas durante la transmisión de un programa, o participando en algún concurso de votación. En el caso nuestro, la interactividad ocurre de manera local, pues no existe en el país la infraestructura que permita conectar directamente a los proveedores de servicios con los usuarios de la TV Digital.



Figura 2. Radiodifusión de Datos Independiente (Izquierda) y Radiodifusión de Datos Dependiente (Derecha).

Desde el comienzo de despliegue de la TV Digital en nuestro país, el Servicio de Radiodifusión de Datos estuvo entre las prioridades a implementar por las entidades a cargo de introducir la tecnología digital. La Figura 3 muestra el sistema utilizado para transmitir este servicio, el cual consta de dos servidores, el

Servidor de EPG (**E**lectronic **P**rogram **G**uide) y el Servidor de DB (**D**ata **B**roadcast). Desde estos dos servidores se generan las tramas de transporte MPEG-2 que contienen los paquetes asociados al Servicio de Radiodifusión de Datos, el cual se divide en dos partes, la **Información** y el **Contenido**. Además de generar la **Información** del Servicio de Radiodifusión de Datos, el Servidor de EPG se encarga de crear las tablas asociadas a la Guía Electrónica de Programa, funcionalidad que posibilita a los usuarios conocer detalles de la programación transmitida. Para ello, este servidor conforma las tablas SDT (**S**ervice **D**escription **T**able) y EIT (**E**vent **I**nformation **T**able). Este servidor genera además las tablas TDT (**T**ime **D**ate **T**able) y TOT (**T**ime **O**ffset **T**able), tablas que transportan la información de fecha y hora, así como la tabla NIT (**N**etwork **I**nformation **T**able), la cual brinda información del proveedor de servicios.

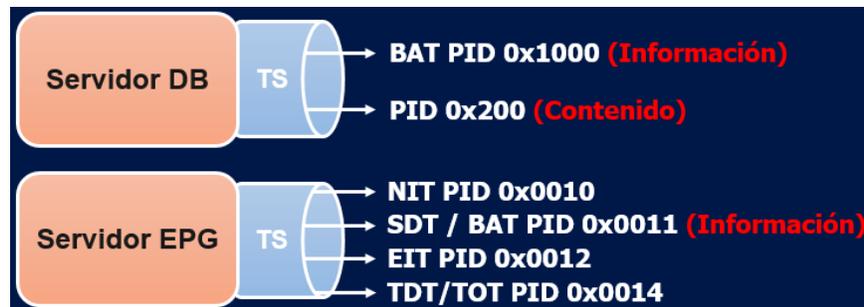


Figura 3. Sistema de transmisión del servicio.

La trama de transporte MPEG-2 o TS (**T**ransport **S**tream) utilizada para encapsular y transmitir toda la estructura de tablas anterior, así como los programas y datos asociados a estos, es definida en [7]. El paquete de transporte MPEG-2 está constituido por 188 bytes, de los cuales 4 componen su cabecera, la cual está formada por 8 campos (Figura 4). Debido a su importancia en el presente trabajo, solo los campos *sync_byte*, *payload_unit_start_indicator*, *PID* y *continuity_counter* serán explicados.

El primer byte (*sync_byte*) de cada paquete MPEG-2 tiene como valor 0x47. El campo *payload_unit_start_indicator* es "1" siempre que el paquete contenga el primer byte de una sección SI (**S**ervice **I**nformation), en caso contrario, su valor es "0". Los 13 bits menos significativos de los 2 bytes que siguen al *sync_byte* pertenecen al campo *PID*, campo que identifica los paquetes que contienen igual información de los restantes que conforman la trama de transporte. El campo *continuity_counter* permite conocer el orden de los paquetes que se reciben con igual *PID* y que no contenga campo de adaptación (*adaptation_field_control* igual a "00" o "10") [7].

La **Información** del servicio debe ser transmitida dentro de la tabla BAT (**B**ouquet **A**ssociation **T**able), específicamente dentro del descriptor *linkage_descriptor*. Según [8], esta tabla debe ser transmitida en paquetes con *PID* (**P**acket **I**dentifier) 0x0011 y con identificador 0x4A (campo *table_id*). Su función es brindar información detallada del Servicio de Radiodifusión como, por ejemplo, en qué paquetes es transportado, cuántos archivos lo componen, así como los nombres de estos. El **Contenido** del servicio es transmitido en los paquetes con el *PID* indicado dentro del *linkage_descriptor* y según [5], deben ser encapsulados siguiendo la estructura definida para la tabla FDT (**F**ile **D**ata **T**able). El **Contenido** del servicio está formado por archivos XML, imágenes BMP y frames I. Estos archivos estructuran los datos transmitidos, brindando además una interfaz visual de cara al usuario de este servicio.

Como se observa en la Figura 3, actualmente la **Información** del servicio es transmitida en paquetes con dos *PID* distintos, 0x0011 y 0x1000. La transmisión en el *PID* 0x1000 se debe a que las primeras cajas

decodificadoras introducidas en el país fueron donadas por el gobierno chino y las mismas fueron diseñadas para recibir la **Información** del servicio en este *PID*, incumpliendo así con lo establecido en el estándar EN 300 468. No obstante, y partiendo de la base de continuar brindándole el servicio a estas cajas, se decidió continuar con el envío de la **Información** por ambos *PID*.

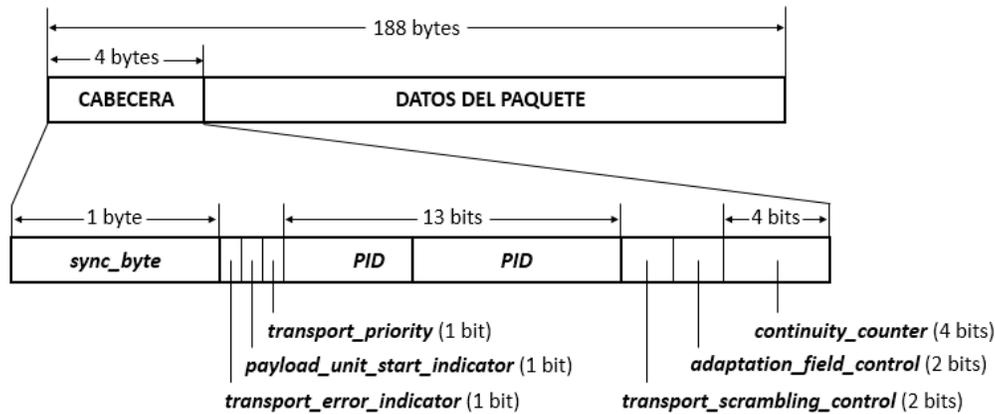


Figura 4. Paquete de la trama de transporte MPEG-2.

ESTRUCTURA DE LAS TABLAS BAT Y FDT

La **¡Error! No se encuentra el origen de la referencia.** y Tabla 3 muestran la estructura de las tablas BAT y FDT definidas en [8] y [5] respectivamente. Tanto las tablas SI como la BAT, como las privadas, la FDT en nuestro caso, están constituidas por sub-tablas, las cuales a su vez se componen por secciones de hasta un máximo de 1024 bytes. La tabla BAT brinda información relacionada a los bouquets, entendiéndose por bouquet, una colección de servicios (por ejemplo, grupo de programas de deportes o de documentales). La misma debe ser transmitida en paquetes con *PID* 0x0011 y valor 0x4A en el campo *table_id*, diferenciándola de esta forma de la tabla SDT (**S**ervice **D**escription **T**able), la cual es transmitida en paquetes con igual *PID*, pero con valor 0x42 en el campo *table_id* [8]. No obstante, en nuestro país es utilizada para transmitir la **Información** del Servicio de Radiodifusión de Datos, no para brindar información relacionada a un grupo de servicios.

Semánticamente, el campo *table_id* permite identificar a qué tabla pertenece la sección transmitida y el campo *section_syntax_indicator* debe tener siempre valor "1". El campo *section_length* especifica el número de bytes que pertenecen a la sección, los cuales comienzan inmediatamente después de este campo e incluyen al campo *CRC_32*, campo que permite realizar un chequeo de redundancia cíclica una vez que la sección es recibida completamente. El *bouquet_id* permite identificar cada uno de los bouquets presentes en la trama de transporte. El campo *version_number* indica el número de versión de la sub-tabla recibida y debe incrementarse en "1" con cada nueva sub-tabla (igual *table_id* y distinto *bouquet_id*). Cuando el campo *current_next_indicator* presenta valor "1", la sub-tabla recibida es la actualmente aplicable, si este campo es recibido con valor "0", la sub-tabla contenida en él será la próxima en ser válida [8].

El campo *section_number* identifica el número de la sección que está siendo recibida en el paquete y debe ser incrementado en "1" con cada nueva sección que se procese de la misma sub-tabla (igual *table_id* y *bouquet_id*). El *last_section_number* permite conocer el número de la última sección que debe ser

procesada para completar la sub-tabla. El campo *bouquet_descriptors_length* brinda el tamaño en bytes de todos los descriptores presentes en la sección, los cuales comienzan inmediatamente después de este campo [8].

El campo *transport_stream_loop_length* especifica el tamaño en bytes del lazo de información de la trama de transporte que le sigue inmediatamente: *transport_stream_id*, campo que identifica este TS (Transport Stream) de cualquier otro dentro del sistema de transmisión; *original_network_id*, campo que identifica el *network_id* del sistema de transmisión original; *transport_descriptors_length*, campo que especifica el tamaño en bytes de los descriptores que siguen a continuación [8].

Tabla 1. Sintaxis de la tabla BAT.

Sintaxis	Número de bits
bouquet_association_section () {	
table_id	8
section_syntax_indicator	1
reserved_future_use	1
reserved	2
section_length	12
bouquet_id	16
reserved	2
version_number	5
current_next_indicator	1
section_number	8
last_section_number	8
reserved_future_use	4
bouquet_descriptors_length	12
for (i=0; i<N; i++) {	
descriptor ()	
}	
reserved_future_use	4
transport_stream_loop_length	12
for (i=0; j<N; j++) {	
transport_stream_id	16
original_network_id	16
reserved_future_use	4
transport_descriptors_length	12
for (j=0; j<N; j++) {	
descriptor ()	
}	
}	
CRC_32	32
}	

Los descriptores permitidos en la tabla BAT son: *service_list_descriptor*, que especifica el tipo de servicio que es transmitido en la trama de transporte; *stuffing_descriptor*, que permite invalidar el descriptor al cual hace referencia o ser utilizado como relleno en la sub-tabla; *bouquet_name_descriptor*, para nombrar al bouquet; *country_availability_descriptor*, que puede aparecer dos veces por cada servicio listado, proporcionando una lista de países para los cuales el servicio está disponible y otra para los que no está

disponible; *multilingual_bouquet_name_descriptor*, que provee el nombre del descriptor en varios idiomas; *CA_identifier_descriptor*, que indica si un bouquet en particular está asociado con un sistema de acceso condicional; *private_data_specifier_descriptor*, es usado para identificar el especificador de algún descriptor o campo privado dentro de los descriptores; *FTA_content_management_descriptor*, que provee un significado de definición de la política de administración de contenido para un elemento del contenido transmitido, como parte de un servicio FTA (Free To Air); *extension_descriptor*, que es usado para ampliar el nombre de 8bits del campo *descriptor_tag* [8].

Específicamente, el *linkage_descriptor*, descriptor que transporta la **Información** del servicio, tiene la estructura mostrada en la

Tabla 2. Este descriptor está presente en la tabla BAT cuando el valor del campo *descriptor_tag* tiene valor 0x4A. El tamaño de este descriptor es especificado con el valor del campo *descriptor_length*. El campo *transport_stream_id* identifica la trama de transporte que contiene el **Contenido** del servicio, mientras que el *network_original_id* identifica la red donde se origina el servicio. El *service_id* permite conocer el servicio que transporta tanto la **Información** como el **Contenido** [8].

Tabla 2. Sintaxis del linkage_descriptor.

Sintaxis	Número de bits
<code>linkage_descriptor () {</code>	
<i>descriptor_tag</i>	8
<i>descriptor_length</i>	8
<i>transport_stream_id</i>	16
<i>original_network_id</i>	16
<i>service_id</i>	16
<i>linkage_type</i>	8
if (<i>linkage_type</i> == 0x80) {	
<i>segment_number</i>	3
<i>PID</i>	13
<i>table_id</i>	8
<i>last_segment_number</i>	3
<i>ver_num</i>	5
for (<i>i</i> = 0; <i>i</i> < <i>N</i> ; <i>i</i> ++) {	
<i>table_ext_id</i>	16
<i>last_section_number</i>	8
}	
}	
}	

Cuando el campo *linkage_type* tiene valor 0x80, su estructura informa acerca de los datos del servicio. Los campos *segment_number* y *last_segment_number* permiten conocer el primer y último segmento del *linkage_type*. El campo *PID* nos permite conocer los paquetes en los cuales el **Contenido** es transportado, mientras que el *table_id* identifica la tabla que lo contiene. El campo *ver_num* identifica la versión de la tabla transmitida. Los campos *table_ext_id* y *last_section_number* también están presentes en la FDT. El primero de ellos enumera cada uno de los archivos transmitidos, mientras que el segundo indica la última sección de la FDT de cada archivo transportado [5].

Por su parte, la tabla FDT transporta todo el **Contenido** asociado al servicio. Los archivos XML, imágenes BMP y frames I que conforman el mismo, son encapsulados utilizando la sintaxis de esta tabla, la cual tiene como valor de *table_id* 0x90. El campo *section_syntax_indicator* debe tener siempre valor "1". El campo *start_indicator* indica que el paquete transporta la página XML principal cuando su valor "1", para cualquier otro archivo transportado su valor es "0". El campo *section_length* permite conocer la cantidad de bytes que siguen a este campo, es decir, el tamaño de la sección FDT y su valor no debe exceder de 4093. El campo *table_ext_id* enumera los distintos archivos que conforman el **Contenido**, comenzando desde "0" (archivo XML principal) e incrementándose en "1" con cada nuevo archivo transmitido [5].

El campo *version* indica el número de la versión y su valor debe ser incrementa en "1" con cada nueva sección transmitida. Cuando el campo *current_next_indicator* presenta valor "1", la sub-tabla recibida es la actualmente aplicable, si este campo es recibido con valor "0", la sub-tabla contenida en él será la próxima en ser válida. Los campos *section_number* y *last_section_number* permiten conocer el número de secciones que deben procesarse para cada archivo transmitido. El campo *file_name_length* brinda el tamaño en bytes que tiene el campo *name_byte*, el cual indica el nombre del archivo transmitido. Asimismo, el campo *file_data_length* permite conocer el tamaño total en bytes del archivo transmitido dentro del campo *data_byte*. Por último, el campo *CRC_32*, permite realizar un chequeo de redundancia cíclica a la sección transmitida [5].

Tabla 3. Sintaxis de la tabla FDT.

Sintaxis	Número de bits
<code>file_data_section () {</code>	
<code>table_id</code>	8
<code>section_syntax_indicator</code>	1
<code>start_indicator</code>	1
<code>reserved</code>	2
<code>section_length</code>	12
<code>table_id_ext</code>	12
<code>reserved_future_use</code>	6
<code>version</code>	5
<code>current_next_indicator</code>	1
<code>section_number</code>	8
<code>last_section_number</code>	8
<code>file_name_length</code>	8
for (i=0; i< file_name_length; i++) {	
<code>name_byte</code>	8
}	
<code>file_data_length</code>	32
for (i=0; j< file_data_length; i++) {	
<code>data_byte</code>	8
}	
<code>CRC_32</code>	32
}	

CONCEPCIÓN DE LA HERRAMIENTA PROPUESTA

La solución propuesta se concibió para ser desarrollada utilizando el lenguaje de programación C, pues en un futuro, se pretende que la misma sea implementada en un STB (**Set-Top Box**) comercial. El uso de un

lenguaje de programación de bajo nivel se hace necesario para lograr una correcta interacción con el hardware del equipo. La misma debe ser capaz de almacenar de manera correcta tanto la **Información**, tabla BAT, como el **Contenido** del servicio, tabla FDT. Una vez almacenados ambos, debe comprobar que los datos sean consecuentes con lo establecido en los estándares EN 300 468 y GD/J018. Además, debe chequear que la **Información** del servicio enviada en los paquetes con *PID* 0x0011 sea igual a la enviada en los paquetes con *PID* 0x1000. Asimismo, debe realizar el análisis del **Contenido** transmitido y emitir y registrar todos los errores que sean encontrados durante todo el análisis realizado.

El software consta de 6 módulos principales como puede observarse en la Figura 5. El módulo **SINCRONIZACIÓN**, es el encargado de cargar y sincronizar el software con la trama de transporte almacenada en la computadora. Una vez lograda la sincronización con la trama, el módulo debe velar por la permanencia de la misma y en caso de perderla, re-sincronizarse. El módulo **EXTRACCIÓN DE SECCIONES BAT** debe realizar un filtrado de paquetes buscando aquellos con *PID* 0x0011 y 0x1000. Específicamente en el caso de los paquetes 0x0011, debe filtrar aquellos con un valor 0x42 en el campo *table_id*. Una vez filtrados, debe extraer de ellos ambas tablas BAT, comprobarlas mediante el código CRC (**Cyclic Redundancy Code**) y almacenarlas para su posterior análisis en el módulo **ANALIZAR DATOS BAT**. Este módulo es el encargado de validar todos los valores de los campos de ambas tablas, es decir, comprobar que los mismos estén acorde a lo establecido en el estándar que los define. Además, debe chequear que ambas estructuras (la de los paquetes 0x0011 y 0x1000) sean iguales y encontrar dónde es transportado el Contenido del servicio mediante el análisis de sus descriptores. Por su parte, el módulo **EXTRACCIÓN DE SECCIONES FDT** debe filtrar los paquetes de secciones FDT en los cuales se encuentra encapsulado y es transportado el **Contenido** del servicio. Una vez filtrados estos paquetes, debe extraer y almacenar todas las secciones que componen a la tabla FDT, comprobar su integridad mediante el chequeo CRC y finalmente almacenarla para su posterior análisis. El módulo **ANALIZAR DATOS FDT** es el encargado de verificar que lo transportado en estas secciones esté conforme a lo declarado en el estándar que lo define. Además, debe comprobar la correcta sintaxis e integridad de todos los archivos XML almacenados. Por último, el módulo **EMITIR LOGS** es el encargado de generar los distintos archivos de texto conteniendo toda la información del análisis realizado, haciendo énfasis en los errores encontrados durante el procesamiento de los datos transmitidos.



Figura 5. Módulos del software.

RESULTADOS OBTENIDOS Y VALIDACIÓN

Hasta la fecha, ha sido concluida la programación de los módulos **SINCRONIZACIÓN** y **EXTRACCIÓN DE SECCIONES BAT**, por lo cual la herramienta realiza correctamente el proceso de sincronización con la trama de transporte a analizar y, además, filtra y almacena los datos asociados a la **Información** del servicio. Asimismo, la herramienta compara los datos de ambas secciones BAT almacenadas y chequea, en cierta medida, que los valores de los datos recibidos sean consecuentes con lo especificado en el estándar, pues actualmente se encuentra en proceso de realización el módulo **ANALIZAR DATOS BAT**. Además, la herramienta registra en archivos de texto las secciones BAT de los paquetes 0x1000 y 0x0011, así como los errores encontrados en las mismas. Como resultado directo de la programación de los bloques anteriores, se obtuvo el seudocódigo de los mismos, el cual podrá ser utilizado en el futuro por empresas desarrolladoras de softwares para desarrollar una solución profesional.

Para validar los resultados obtenidos fueron utilizadas otras herramientas software que permiten verificar, en cierta medida, el Servicio de Radiodifusión de Datos empleado en nuestro país, pues las mismas solo son capaces de analizar la información de las secciones BAT, que es donde es transportada la **Información** del servicio. Las herramientas software utilizadas fueron: 4T2 Content Analyser, MPEG TS Utils y TS Reader. Además, se utilizó una trama de transporte obtenida en nuestro laboratorio y de la cual se conocía completamente su estructura, posibilitando conocer si la herramienta hacía o no un correcto procesamiento de la misma. El TS contenía la **Información** del Servicio de Radiodifusión de Datos en los *PID* 0x0011 y 0x1000, pero solo la transmitida en los paquetes 0x1000 estaba completa y no contenía errores. El **Contenido** del servicio era transmitido en el *PID* 0x200 y su estructura era correcta.

En la Figura 6 se observa como la herramienta desarrollada es capaz de mostrar los campos de la estructura de la tabla BAT obtenidos de ambos paquetes. Obsérvese, que en el caso de los paquetes 0x0011, la **Información** del servicio no es transmitida completamente (los valores de los campos *section_length* y la cantidad de descriptores en los paquetes 0x0011 no son iguales a los 0x1000), lo que posteriormente será mostrado en el registro de errores. La Figura 7 muestra los descriptores que son recibidos en los paquetes 0x0011 y 0x1000. Como puede apreciarse, el software 4T2 Content Analyser recibe 4 descriptores, los mismos que la solución implementada obtiene de los paquetes 0x1000. Sin embargo, en los paquetes 0x0011 se recibe un descriptor diferente (*private_data_descriptor*) a los recibidos en los paquetes 0x1000. Además, solo es recibido el primer segmento de los cinco (enumerados desde 0x0 a 0x4) que conforman al descriptor *linkage_descriptor*.

La Figura 8 muestra una lista de los errores encontrados durante el análisis y comparación de las secciones BAT transportadas en los paquetes con *PID* 0x0011 y 0x1000. Vale destacar que el registro de errores que la solución desarrollada muestra no es posible obtenerlo en ninguna de las herramientas utilizadas para validar la solución.

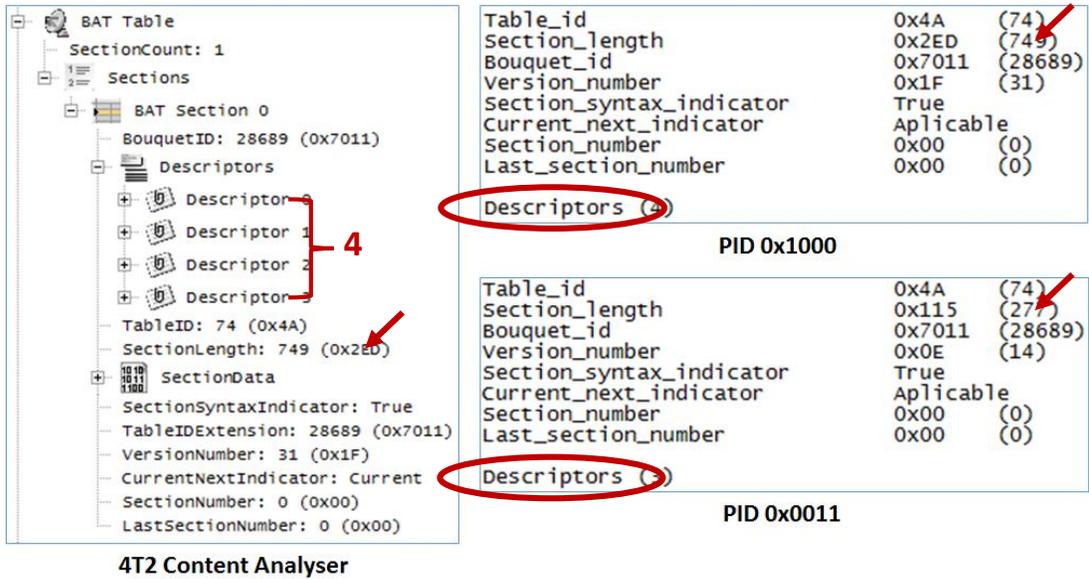


Figura 6. Contenidos de la Tabla BAT: a la izquierda, 4T2 Content Analyser, a la derecha, herramienta desarrollada.

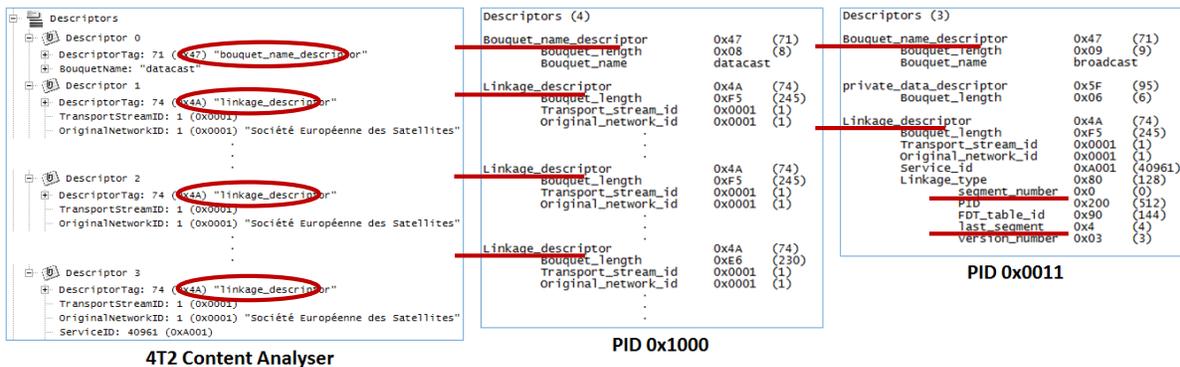


Figura 7. Descriptores mostrados por el software 4T2 Content Analyser (izquierda), por la herramienta desarrollada en el PID 0x1000 (centro) y en el PID 0x0011 (derecha).

Errores:

- 1 - La sección BAT no tiene la longitud en bytes requerida (contiene 277 de 749 bytes).
- 2 - No son recibidos en la sección los 4 descriptores correspondientes (1 bouquet_descriptor y 3 linkage_descriptor).
- 3 - Envío de descriptor erróneo (private_data_descriptor).
- 4 - Solo es recibido el primer segmento del linkage_descriptor (1 de 4).

Figura 8. Logs de errores mostrados por la herramienta desarrollada.

FUTURAS LÍNEAS DE TRABAJO

Las futuras líneas de trabajo del presente proyecto de investigación tienen como principal objetivo el desarrollo de los restantes módulos de la herramienta propuesta. En el caso del módulo **ANALIZAR DATOS FDT**, debe realizarse un estudio de las estrategias y herramientas existentes actualmente para el análisis de archivos XML, para escoger e implementar una de ellas en nuestra solución. Una vez programados y validados todos los módulos de la herramienta, se procederá a su implementación en el STB. Para ello deberá realizarse un estudio del sistema operativo eCos, sistema utilizado por el STB, y de cómo el mismo interactúa con el hardware del equipo. Además, familiarizarse con las herramientas provistas por el fabricante para el diseño de la interfaz gráfica, así como para la programación, compilación y grabación de la solución final en el equipo. Todas estas herramientas son libres de pago (licencias GNU), lo que posibilita la independencia tecnológica de la solución desarrollada.

CONCLUSIONES

La aplicación desarrollada, aunque todavía en su etapa inicial, logra cumplir con los objetivos propuestos, pues en su primera versión consigue establecer una correcta sincronización con la trama de transporte MPEG-2 (módulo **SINCRONIZACIÓN**), permitiendo la extracción de las secciones BAT (módulo **EXTRACCIÓN DE SECCIONES BAT**) para su posterior análisis (módulo **ANALIZAR DATOS BAT**). Además, permite detectar y registrar errores que están ocurriendo actualmente en la transmisión nacional de la señal digital y están asociados a la **Información** del servicio que es transmitida en los paquetes con *PID* 0x0011 (módulo **EMITIR LOGS**). Con su implementación en el STB (etapa final del proyecto), dotaríamos al proveedor del servicio digital de una herramienta que posibilitaría la detección en tiempo real de los errores asociados al Servicio de Radiodifusión de Datos y su rápida corrección por los operadores de la cabeza de línea.

REFERENCIAS

- [1] ATSC, "A/100-1 - DTV Application Software Environment Level 1 (DASE-1) Part 1: Introduction, Architecture, and Common Facilities," p. 66, 2003.
- [2] ETSI, "TS 102 812 v1.3.1 - Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.3," p. 1243, 2012.
- [3] G. Baum and L. F. G. Soares, "Ginga middleware and digital TV in Latin America," *IT Professional*, vol. 14, pp. 59-61, 2012.
- [4] ITU, "H.761 - Nested context language (NCL) and Ginga-NCL," p. 154, 2013.
- [5] SARFT, "GD/J018 - Technical requirements and measurement methods for integrated receiver decoder (basic) of satellite broadcast system (provisional)," p. 126, 2008.
- [6] T. Bray, J. Paoli, C. Sperberg-McQueen, Y. Mailer, and F. Yergeau, "Extensible Markup Language (XML) 1.0 5th Edition, W3C recommendation, November 2008," ed.
- [7] ISO/IEC, "13818-1 Information technology – Generic coding of moving pictures and associated audio information: Systems," 2007.
- [8] ETSI, "EN 300 468 v.1.13.1 - Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems," p. 144, 2012.