

Generador de Flujos Únicos de Paquetes de Transporte TS en base a la Norma ISDB-Tb

Moncayo T.*; Pozo M.*; Mejía D.*; Bernal I.*

*Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica, Quito, Ecuador
e-mail: tatiana_23.moncayo@hotmail.com, pozocaicedom@gmail.com, {ivan.bernal; david.mejia}@epn.edu.ec

Resumen: Este artículo inicia con una breve síntesis acerca de la estructura de los flujos únicos de paquetes de transporte TS, así como su generación mediante OpenCaster. Luego se presenta una aplicación que sirve como interfaz entre el usuario y OpenCaster, cuyo objetivo es facilitar y automatizar el proceso de generación de los mencionados flujos. La aplicación presenta una interfaz gráfica y considera los casos en los que los flujos incluyen no solo el audio y el video de uno o varios programas, sino también el código y los medios necesarios para transmitir aplicaciones interactivas que deben difundirse en conjunto con dicha programación. Estos flujos también deben integrar las tablas que permiten al decodificador ensamblar y sincronizar el audio y el video de cada programa y recuperar el código y los medios de una aplicación. Se presentan también los resultados de emplear el Generador de Flujos con distintas configuraciones de número de programaciones, resolución y aplicaciones interactivas. Los flujos luego son probados en un sistema con receptores digitales reales.

Palabras clave: TS, BTS, ISDB-Tb, Tablas PSI/SI.

Abstract: This article starts by presenting a brief summary about the structure of single transport packet streams and the way these streams are generated by OpenCaster. Then an application that allows a user to visually interact with OpenCaster is presented. This application makes it extremely easy to automatically generate the streams. The application considers scenarios in which the streams must carry not only the audio and video of one or several TV programs but also the code and media for transmitting interactive applications that need to be broadcasted along with the regular programming. These streams must also include all of the tables that make it possible to reassemble and synchronize the audio and video of each program and collect the code and media of an application. The results of using this new application are presented considering a variety of configurations to generate the streams, with different number of programs, resolution and interactive applications. These streams are later tested on actual digital receptors.

Keywords: TS, BTS, ISDB-Tb, PSI/SI Tables.

1. INTRODUCCIÓN

El estándar para TV Digital Terrestre ISDB-Tb (*International System for Digital Broadcast, Terrestrial, Brazilian version*), adoptado por el Ecuador en marzo de 2010, fue desarrollado en Brasil y es el resultado de una modificación al estándar japonés ISDB-T y se diferencia básicamente en el esquema de compresión de audio y video empleado y en el middleware que se usa para la interactividad.

Entre las características más sobresalientes del estándar ISDB-Tb se pueden mencionar: soporte para movilidad y portabilidad; uso eficiente del espectro y multiprogramación; formato panorámico (16:9); y, la interactividad a través de su middleware abierto Ginga.

La interactividad es posible a través de la ejecución de aplicaciones interactivas, las cuales son programas informáticos que habilitan el contenido interactivo; y que son transmitidas, generalmente, por el aire junto con el audio y video de un programa televisivo. Las aplicaciones se ejecutan sobre Ginga que debe estar disponible en el receptor [2]. La Fig. 1 muestra un ejemplo de aplicación interactiva en ejecución.

La temática de TV Digital abarca dos aspectos fundamentales: el de transmisión y el desarrollo de aplicaciones interactivas. Estos aspectos se interrelacionan cuando las aplicaciones desarrolladas deben ser transmitidas.



Figura 1. Aplicación interactiva en ejecución [11]

Artículo recibido el 14 de diciembre de 2014; revisado el 25 de marzo de 2015. Este trabajo fue financiado por la Escuela Politécnica Nacional y con el soporte del MINTEL en el marco de las "Iniciativas sobre TV Digital" (<http://ginga.epn.edu.ec>) que se desarrollan en el DETRI bajo la dirección de David Mejía, MSc. e Iván Bernal, Ph.D. Información de contacto: Iván M. Bernal, e-mail: ivan.bernal@epn.edu.ec, Tel: 593-2-2507-144 ext.2338. Escuela Politécnica Nacional. Ladrón de Guevara E11 - 253. Quito, Ecuador.

Las señales de audio y video de una programación deben multiplexarse entre sí, luego de haber sido comprimidas de acuerdo a lo señalado por el estándar MPEG-4. Estas señales de audio y video multiplexadas de una programación deben a su vez multiplexarse junto con el audio y video de otras programaciones (multiprogramación), generalmente, de diferente resolución: resolución baja 1seg para los móviles, estándar (SD) y de alta definición (HD) para otros receptores. Estas multiprogramaciones deben también multiplexarse con información que permita en el receptor discriminar la información correspondiente a cada programa. Finalmente, todo lo anterior debe estar multiplexado con: información de *closedcaption*; guía electrónica de programación (EPG); y, el código de las aplicaciones interactivas y medios que ayudan a implementar su funcionalidad, como imágenes, páginas HTML, etc. Todo lo mencionado se resume en la Fig. 2.

En el estándar ISDB-Tb, audio, video y datos de una programación, deben ser multiplexados formando así, el *TransportStream* (TS). Luego, las múltiples salidas del multiplexor que contienen la información de las distintas programaciones deben ser re-multiplexadas para formar el denominado “**flujo único de paquetes de transporte TS**”, el cual después de realizar un proceso de codificación en la etapa de salida del re-multiplexor conforma la señal BTS (*BroadcastTransportStream*) [10]. Mediante la re-multiplexación, el estándar ISDB-T consiguió adaptar el TS de MPEG-2 de tal manera que funcione para transmisión jerárquica y recepción parcial.

Para que el desarrollo de las aplicaciones interactivas no se circunscriba solamente al uso de simuladores y emuladores sino que las aplicaciones puedan ser evaluadas en televisores y STB (*Set Top Boxes*) reales, y que las pruebas de las mismas puedan realizar con usuarios interactuando con las aplicaciones a través del control remoto, es necesario establecer un sistema básico y de bajo costo.

Una alternativa es generar en software el flujo único de paquetes de transporte TS, por cuanto la codificación a la salida del re-multiplexor se realiza normalmente en hardware para conforma el BTS. Luego de lo cual se debe emplear un transmisor de bajo costo que acepte el flujo generado.

OpenCaster [5] es un software que permite la generación de flujos únicos de transporte de paquetes para el estándar europeo DVB-T. El laboratorio LIFIA [13] de Argentina generó un parche para OpenCaster [14] para permitir la generación de flujos únicos de transporte de paquetes para el estándar ISDB-Tb. En [6], [12] se han presentado diferentes alternativas para la generación de estos flujos; sin embargo, en la mayoría de los casos el proceso no es claro, y depende de que el usuario del software conozca todas las particularidades de los diferentes comandos requeridos.

El objetivo del artículo es presentar una aplicación desarrollada para automatizar el proceso de generación de los flujos únicos de paquetes de transporte con OpenCaster.

Este artículo está organizado de la siguiente forma: en la sección 2 se presenta una breve discusión de la estructura de los TS y BTS, lo cual es fundamental para entender la forma en que se generan en OpenCaster y la automatización de este proceso. Una revisión del procedimiento de generación de los

flujos con OpenCaster se presenta en la sección 3; en la sección 4 se introduce la aplicación “Generador de Flujos”, se describe la idea detrás de la automatización del proceso de generación de flujos y funcionalidad de la aplicación. La sección 5 presenta los resultados obtenidos al emplear el generador de flujos y usar los flujos generados en un sistema con receptores reales. Finalmente, en la sección 6 se presentan las conclusiones del trabajo.

2. TS y BTS

ISDB-Tb utiliza para la compresión del audio y video MPEG-4 y para la multiplexación de las varias programaciones sigue empleando parte del estándar MPEG-2 [16], el denominado Flujo de Transporte o *TransportStream* (TS).

Para obtener un paquete TS, primero deben pasar, tanto el audio como el video, por un proceso de codificación y compresión independiente, formando cada uno de ellos un flujo elemental de información denominado ES (*ElementaryStream*), como se indica en la Fig.3. Un ES es la unidad básica de transporte de paquetes de audio y video que se encuentra a la salida de los codificadores. Luego, cada ES de audio o video, pasa por un paquetizador, que lo convierte en un PES (*PacketizedElementaryStream*). Un PES es un paquete de tamaño variable, de máximo 65.536 bytes.

En la Fig. 3 se puede observar que a la salida del multiplexor MPEG-2, se obtienen paquetes de 188 bytes, de los cuales 4 bytes son de cabecera y el resto corresponde a la información.

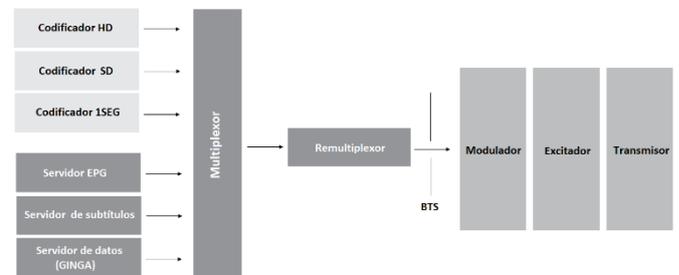


Figura 2. Generación de una señal con multiprogramación y aplicación interactiva incluida [8]

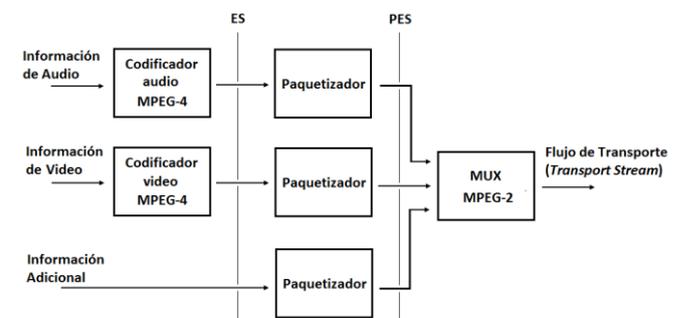


Figura 3. Conformación del TS para una programación [18]

Como se mencionó anteriormente, los TS de varias fuentes de video deben ser re-multiplexados para formar el flujo único de paquetes de transporte TS, el cual sometido a una codificación de Reed-Solomon conforma el BTS. El BTS está conformado por unidades básicas denominadas TSP (*TransportStreamPacket*) que tienen un tamaño de 204 bytes, como se indica en la Fig. 4. Los flujos únicos de paquetes de transporte TS pasan por un proceso de codificación, en el cual se agregan 16 bytes; los ocho primeros sirven para identificar la información de la capa jerárquica, que permite tener programaciones transmitidas a distinta resolución, y pueden transportar información auxiliar, y los 8 bytes restantes identifican la paridad que se obtiene del codificador Reed-Solomon.

2.1 Cabecera del paquete TS

La cabecera del paquete TS se presenta en la Fig.5. De la cabecera es relevante mencionar los 13 bits correspondientes al PID (*PacketIdentifier*), que especifica el tipo de información que se encuentra en la carga útil. De forma análoga a los PES, se tienen las denominadas secciones que son paquetizadas y transportadas en los paquetes TS de 188 bytes. Una sección encapsula las tablas PSI/SI y carrusel de datos para organizar la transmisión de las aplicaciones interactivas y sus medios.

2.2 Tablas PSI/SI

Las Tablas PSI/SI (*ProgramSpecificInformation/Service Information*), contienen datos con información del TS para que sea demultiplexado correctamente en el decodificador, de tal forma que el decodificador pueda localizar tanto el video como el audio de cada programa; además, permiten la sintonización automática de los terminales, creación de la guía electrónica, entre otros.

Las tablas PSI proveen información específica de uno o varios programas que fueron multiplexados de acuerdo al estándar MPEG-2, y su función principal es ayudar en la demultiplexación y la presentación de la información en el decodificador. Las tablas que contribuyen con información específica del programa son: *ProgramAssociationTable* (PAT) – proporciona la correspondencia entre el número de programa y los paquetes de audio, video y datos que lo componen, *ProgramMapTable* (PMT) – define los PID que identifican los flujos individuales que constituyen un programa y *Conditional Access Table* (CAT) – empleada para definir los accesos en servicios del tipo *pay-per-view*. Estas tablas están definidas en la norma MPEG-2.

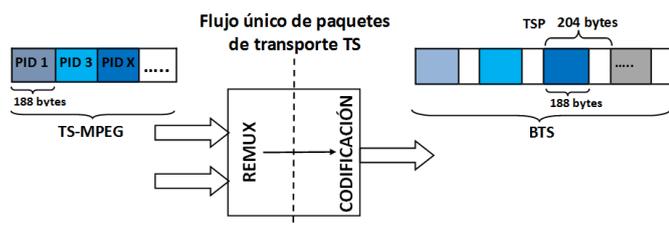


Figura 4. Re-multiplexación del TransportStream [10]

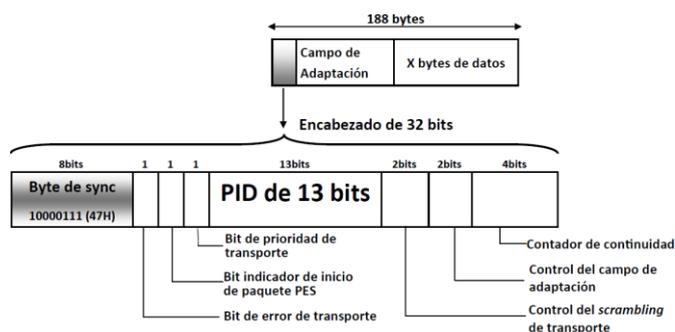


Figura 5. Estructura de un paquete TS [17]

Las tablas SI funcionan como complemento de las tablas PSI, permiten el control del receptor y el envío de la EPG, aplicaciones interactivas, entre otros servicios. También se las conoce como tablas de servicios de información. Existen 15 tablas SI según la norma ABNT NBR 15603-1 [1], [2], de estas las que más información brindan son: *Network InformationTable* (NIT) – ayuda en la sintonización de un nuevo canal, *Service DescriptionTable* (SDT) – describe los servicios contenidos en el TS, los nombres de proveedores, los nombres de los canales y *ApplicationInformationTable* (AIT) – contiene información sobre la aplicación interactiva.

3. GENERACIÓN DE FLUJOS ÚNICOS DE PAQUETES DE TRANSPORTE CON OPENCASTER

OpenCaster es un software de código abierto desarrollado principalmente por AVALPA Digital Engineering SRL como parte de AVALPA Broadcaster Server. El Laboratorio de Investigación y Formación en Informática Avanzada (LIFIA) de la Universidad Nacional de la Plata realizó modificaciones a OpenCaster, para soportar el estándar ISDB-Tb. Estas modificaciones están disponibles como un parche que deben ser aplicadas a OpenCaster. OpenCaster puede ser descargado de la página oficial de AVALPA [5] y la versión modificada por LIFIA se encuentra disponible en [14].

Para la instalación de la versión 2.4 de OpenCaster es necesario cumplir con los siguientes requerimientos: sistema operativo GNU/Linux (Ubuntu 12.10), lenguaje de programación Python (Python 2.7.3), y el compilador del lenguaje C (GCC 4.6.3). Un resumen del proceso de instalación se presenta a continuación:

1. Ubicar en un mismo directorio los archivos: OpenCaster2.4.tgz y OpenCaster2.4-lifia-rev362.patch.
2. Ingresar en el directorio creado y descomprimir el archivo mediante:

```
$ tarzvf OpenCaster2.4.tgz
```
3. Ejecutar el parche desarrollado por LIFIA:

```
$ cd OpenCaster2.4/
$ patch -p1< ../OpenCaster2.4-lifia-rev362.patch
```
4. Instalar OpenCaster con los privilegios del usuario root:

```
$ sudo make
$ sudo makeinstall
```

3.1 Comandos de OpenCaster

OpenCaster emplea una serie de comandos que deben ser ejecutados en el terminal de Linux, puesto que no dispone de una interfaz gráfica. Algunos de los comandos más importantes con su respectiva función se mencionan a continuación:

- esaudio2pes: encapsula los ES de audio en PES.
- esaudioinfo: entrega información sobre los ES de audio.
- esvideo2pes: encapsula los ES de video en PES.
- oc-update.sh: genera un TS a partir de una carpeta que contiene el código y los objetos media de las aplicaciones interactivas de las que se generará un carrusel de objetos.
- pesaudio2ts: encapsula un PES de audio en un TS.
- pesinfo: analiza la información contenida en un PES y la muestra en pantalla.
- pesvideo2ts: encapsula un PES de video en un TS.
- tschrmuxer: multiplexa los flujos de audio y video de todos los servicios, las tablas PSI/SI y los flujos que contienen la información de la aplicación interactiva, en un flujo único de transporte TS.
- tspcrmeasure: presenta la tasa de bits de un TS.
- tsstamp: corrige los PCR de un TS para que la llegada de los paquetes al STB sea precisa. Un PCR (*Program Clock Reference*) permite al decodificador presentar el contenido sincronizado.

3.2 Generación de un flujo único de paquetes de transporte TS para un servicio sin aplicación interactiva

En la Fig.6 se muestra el proceso que debe seguirse para obtener a partir de un video un archivo .ts que contenga el flujo único de paquetes de transporte. Inicialmente se separa del archivo original su audio y video. Este proceso de separación da como resultado un ES de audio y un ES de video, luego de lo cual cada ES se procesa por separado. A cada ES se le añade una cabecera transformándolo así en un PES. Un PES es un archivo de gran tamaño, por lo que es necesario dividirlo en paquetes de 184 bytes y añadir a cada uno de estos paquetes una cabecera de 4 bytes, obteniendo así 188 bytes que conforman un paquete TS. Este TS se almacena en un archivo .ts.

Una vez que se tiene un archivo .ts de audio, y uno de video, se deben generar las tablas PSI/SI y convertirlas en un archivo .ts. OpenCaster incluye un script en Python el cual debe ser modificado de acuerdo a las necesidades del usuario para generar estas tablas. También es necesario un archivo null.ts, el cual contiene un paquete nulo, y que ayudará a mantener la tasa de datos constante (29,9582 Mbps) en caso de que el audio, video y tablas PSI/SI no lo consigan.

Una vez se dispone de todos estos archivos, se procede a multiplexarlos para obtener finalmente el flujo único de paquetes de transporte TS; luego es necesario sincronizar las estampillas de tiempo para que los paquetes lleguen en el orden correcto al receptor; al final del proceso se obtiene un archivo que contiene el flujo único de paquetes de transporte TS, el cual puede ser entregado al transmisor para su posterior difusión.

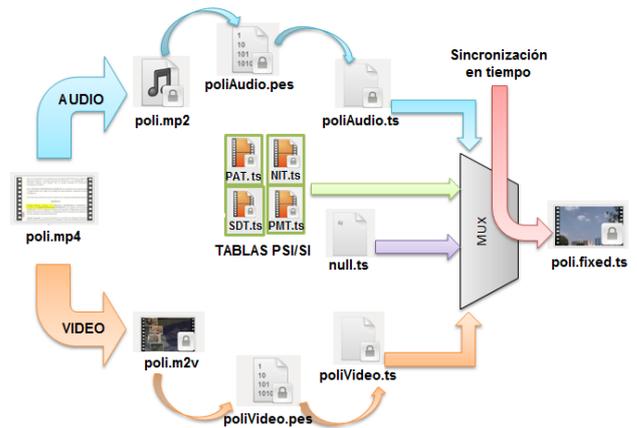


Figura6. Procedimiento para la generación del flujo

Este proceso se presenta mediante un ejemplo práctico para clarificarlo, una explicación detallada de los diferentes valores empleados en los comandos presentados se puede encontrar en [6], [12], [19]:

1. Es necesario colocar en una sola carpeta todos los archivos necesarios, por ejemplo: el script para generación de tablas gtables.py, el archivo con el paquete nulo null.ts y el video que será procesado poli.mp4.
2. Editar el script de generación de tablas para especificar, por ejemplo, un ID de red, los ID de los servicios que se incluirán, los ID de las tablas, entre otros. Una vez editado el archivo se lo ejecuta:


```
$ python gtables.py
```
3. Convertir el video al formato AVI (*Audio Video Interleave*). Este paso es necesario debido a que luego de realizar varias pruebas se comprobó que este formato de video permite que OpenCaster realice su trabajo correctamente. Para esto se debe ejecutar el comando:


```
$ ffmpeg -i poli.mp4 -s 720x576 -r 25 -aspect 16:9 -b 5000k convertido.avi
```
4. Obtener el ES de audio:


```
$ ffmpeg -i convertido.avi -vn -ac 2 -acodec mp2 -f mp2 -ab 128000 -ar 48000 poli.mp2
```
5. Obtener el ES de video:


```
$ ffmpeg -i convertido.avi -an -vcodec mpeg2video -f mpeg2video -s 720x576 -r 25 -aspect 16:9 -deinterlace -b 5000k -minrate 5000k -maxrate 5000k -bf 2 -bufsize 1835008 poli.m2v
```
6. Encapsular el ES de audio en un PES de audio:


```
$ esaudio2pes poli.mp2 1152 48000 384 3600 >poliAudio.pes
```
7. Encapsular el ES de video en un PES de video:


```
$ esvideo2pes poli.m2v >poliVideo.pes
```
8. Obtener el archivo .ts de audio:


```
$ pesaudio2ts 2075 1152 48000 384 0 poliAudio.pes >poliAudio.ts
```
9. Obtener el archivo .ts de video:

```

$ pesvideo2ts 2065 25 112 5750000 0
poliVideo.pes>poliVideo.ts
10. Multiplexar los archivos .ts de audio, video y tablas:
$tsctbrmuxer 4220000 b:15040pat.ts
b:15040 pmt_sd.ts b:3008 sdt.ts
b:3008 nit.ts b:5750000 poliVideo.ts
b:188000 poliAudio.ts
b:23984198null.ts>poli.ts
11. Sincronizar audio y video:
$ tsstamppoli.ts 29958294
>poli.fixed.ts
    
```

3.3 Generación de un flujo único de paquetes de transporte TS para un servicio con aplicación interactiva

En la Fig.7 se presenta un diagrama de la generación del flujo único de transporte TS en caso de que exista una aplicación interactiva. Tanto audio como video serán procesados de la misma manera que en el caso anterior, la diferencia es que cuando se incluye una aplicación interactiva se necesitan los archivos de la aplicación en formato .ts para el multiplexado final, para lo cual se debe generar un carrusel de objetos del directorio que contiene el código de la aplicación y los medios. También es necesario incluir la tabla AIT en el multiplexado. Finalmente, todos los archivos obtenidos ingresan al multiplexor, teniendo a la salida, el flujo único de paquetes de transporte TS final, que incluye audio, video, tablas y la aplicación interactiva y que podrá ser entregado al transmisor.

El proceso de generación de un flujo único de paquetes de transporte TS para un servicio con aplicación interactiva se presenta mediante un ejemplo práctico para clarificarlo:

1. Es necesario colocar en una sola carpeta todos los archivos necesarios, por ejemplo: el script para generación de tablas (gtables.py), el archivo con el paquete nulo (null.ts) y el video que será procesado (poli.mp4), así como la carpeta que contiene el código y los medios de la aplicación interactiva (Ginga).
2. Generar el carrusel de objetos:


```
$ oc-update.shGinga 0x0C 1 2004 2
```

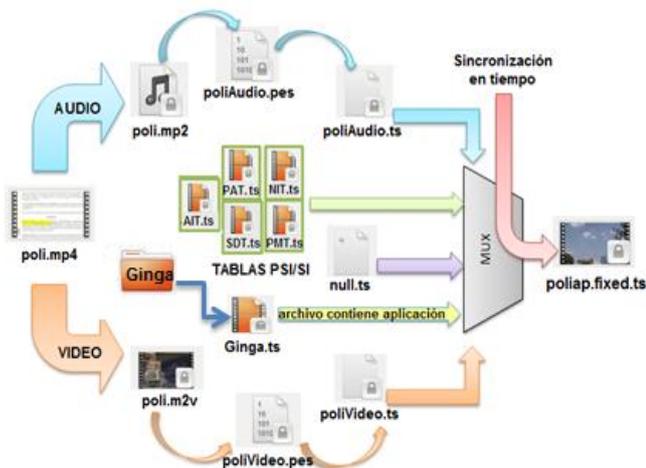


Figura 7. Procedimiento para la generación del flujo con interactividad

3. Editar el script de generación de tablas para incluir la tabla AIT y ejecutarlo:


```
$ python gtables.py
```
4. Obtener los archivos .ts de audio y video siguiendo el proceso descrito en el caso anterior. Multiplexar todos los archivos .ts mediante:


```
$tsctbrmuxer 4220000 b:15040pat.ts
b:15040 pmt_sd.ts b:3008 sdt.ts
b:3008 nit.tsb:3008
ait.tsb:2500000Ginga.ts b:5750000
poliVideo.ts b:188000 poliAudio.ts
b:21481190null.ts>poliap.ts
```
5. Sincronizar la llegada de audio y video al STB, incluyendo las estampillas de tiempo sincronizadas:


```
$ tsstamppoliap.ts 29958294
>poliap.fixed.ts
```

4. AUTOMATIZACIÓN DEL PROCESO DE GENERACIÓN DE FLUJOS ÚNICOS DE PAQUETES DE TRANSPORTE TS MEDIANTE UNA APLICACIÓN

Como se presentó en la sección anterior, para generar los flujos, es necesario que el usuario conozca una serie de parámetros, configuraciones y comandos que deben emplearse para la generación de los flujos únicos de paquetes de transporte TS mediante OpenCaster. Si el usuario comete un error en uno de los pasos, o emplea un valor equivocado en una opción de los comandos u olvida ejecutar uno de los procesos posible que el archivo con el flujo único de paquetes de transporte tenga errores o no se genere.

Para automatizar este proceso y reducir la complejidad del mismo, y debido a que OpenCaster no dispone de una interfaz gráfica, se desarrolló una aplicación que sirva como interfaz entre el usuario y OpenCaster y simplifique el proceso de generación del flujo único.

4.1 Requerimientos previos a la generación

Existen dos herramientas importantes que ayudan con información y procesamiento antes de realizar la multiplexación: *ffmpeg*: es un programa que permite la conversión rápida de formato de audio y video con gran calidad, haciéndola incluso en tiempo real, además incluye libavcodec, que es la librería de codificación de audio y video más importante actualmente, soporta una gran variedad de formatos tanto para audio como para video. Es necesario para la generación, ya que ayuda a obtener los ES, tanto de audio como de video; *tovid*: es una herramienta de software libre que reúne una colección de herramientas para Linux, y permite crear DVD, es importante para la generación ya que brinda información sobre las propiedades del video fuente. Para lograr la generación de los flujos únicos de paquetes de transporte TS, se utilizan las tablas PSI/SI, las cuales típicamente se generan mediante un script en Python, que contiene el código de las distintas tablas; este script debe ser modificado manualmente dependiendo del número de servicios que se vayan a multiplexar, así como también si se incluye o no una aplicación interactiva. Lo mencionado

implica tener conocimiento de la estructura de las tablas PSI/SI y sus descriptores, para que al momento de ejecutar el script en el terminal de Linux, no se generen errores. El generador de flujos se encarga de la manipulación y modificación de los scripts para así evitar al usuario complicaciones que se puedan presentar.

Los scripts que internamente emplea el Generador de Flujos permiten multiplexar hasta tres programaciones, que es el número que se está exigiendo en el Ecuador en las concesiones temporales que actualmente tienen los canales de TV para realizar sus pruebas con señales de TV digital.

Los scripts incluyen los comandos que determinan las acciones a realizarse en el shell de Linux, es decir los ejecutables que serán invocados con las opciones que sean adecuadas y en el orden indicado anteriormente, y dependiendo de si se incluye o no una aplicación interactiva. En base a lo ingresado por el usuario en la ventana principal y si desea en la ventana de opciones avanzadas, la aplicación modifica estos scripts para que generen adecuadamente las tablas PSI/SI que correspondan.

La aplicación emplea dos hilos de ejecución (*threads*), uno para manejar la interfaz de usuario y otro para realizar el procesamiento contemplado en el script. Cuando el hilo concluya el procesamiento se le informará al usuario que el proceso ha concluido. El Generador de Flujos fue desarrollado usando Qt que a su vez se basa en C++.

4.2 Implementación del Generador de Flujos

Para el desarrollo de esta aplicación se plantearon los siguientes objetivos: elaborar una interfaz gráfica amigable para el usuario, y que sea fácil de usar; permitir al usuario modificar información importante para la generación de los flujos únicos de paquetes de transporte TS; automatizar el proceso de generación y simplificarlo; permitir que el usuario modifique características propias del video como: resolución, relación de aspecto, calidad de video, entre otros.

Estos objetivos fueron modelados mediante un diagrama UML (*Unified Modeling Language*) [4] de casos de uso, el mismo que se presenta en la Fig. 8.

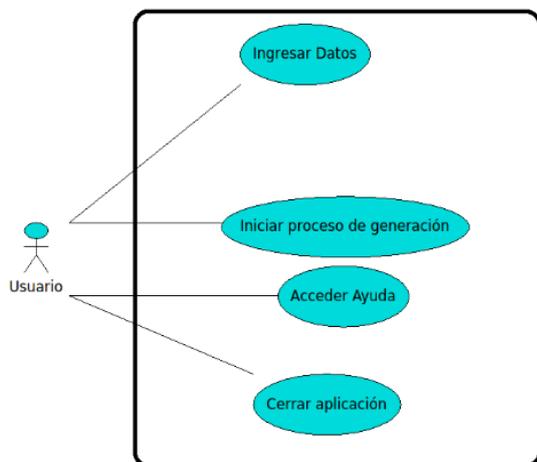


Figura 8. Diagrama de casos de uso del Generador de Flujos

El usuario de la aplicación no requiere conocimientos avanzados sobre OpenCaster, pero si es necesario que conozca algunas particularidades del procesamiento de audio y video, sobre todo si desea modificar ciertas características en el proceso de generación de flujos únicos de paquetes de transporte TS, para ello la aplicación ofrece un modo avanzado de operación. En la Tabla 1 se detalla cada caso de uso y las actividades que debe realizar la aplicación.

4.3 Ventanas del Generador de Flujos

Al iniciar el Generador de Flujos se presenta la ventana principal que se muestra en la Fig.9. Esta ventana permite seleccionar el archivo de video para su procesamiento y establecer el número de programaciones, el tipo de servicio (LD, SD, HD), la aplicación que se incluirá o si se desea establecer opciones avanzadas. En la ventana principal del Generador de Flujos se encuentran los siguientes elementos:

1. Botón buscar: despliega una ventana que permite seleccionar el directorio donde se guardará el resultado.
2. Directorio: espacio que muestra la ubicación completa del directorio seleccionado.
3. Nombre: en este espacio se ingresa el nombre que se desea dar al archivo que contendrá el flujo único de paquetes de transporte TS.
4. Programaciones: permite establecer el número de programaciones que serán multiplexadas.
5. Tipo de servicio: permite escoger el tipo de servicio que se quiere tener en cada una de las programaciones.
6. Directorios: muestran los directorios completos de los videos escogidos por el usuario.
7. Botones buscar: abren ventanas que permiten la selección de los archivos de video que van a ser multiplexados, estos archivos de preferencia deben tener formato .avi o .mp4.
8. Resolución: permite seleccionar resoluciones contempladas en la norma de acuerdo al tipo de servicio.
9. Aplicación: al seleccionar un servicio, se tiene la opción de incluir una aplicación interactiva al flujo único de paquetes de transporte TS.

Tabla 1. Casos de uso del Generador de Flujos

Caso de uso	Aplicación
Ingresar Datos	Habilitar funcionalidades
Iniciar proceso de generación	Guardar datos y realizar los procesos correspondientes. Comprobar los datos ingresados. Modificar los scripts en función de la información ingresada por el usuario. Ejecutar los scripts en un proceso. Informar al usuario sobre el proceso.
Acceder a la Ayuda	Presentar un manual de ayuda.
Cerrar aplicación	Limpiar, finalizar proceso y cerrar aplicación.

10. Buscar aplicación: abre una ventana que permite la selección del archivo principal .ncl de la aplicación a ser incluida. Se debe aclarar que para la generación del carrusel de objetos se empearán todos los archivos que estén en el mismo directorio que el archivo principal .ncl.
 11. Directorio aplicación: contiene la ubicación completa del archivo .ncl.
 12. Botón opciones avanzadas: abre la ventana secundaria que permite ingresar información técnica adicional sobre el flujo único de paquetes de transporte TS.
 13. Botón generar: al pulsar el botón generar empieza a procesar toda la información adquirida anteriormente, el usuario deberá esperar unos minutos para obtener su flujo único de paquetes de transporte TS final.
 14. Botón limpiar: permite limpiar todos los campos mostrados en la ventana principal en caso de alguna equivocación al ingresar los datos.
 15. Botón cerrar: permite salir de la aplicación.
- La aplicación Generador de Flujos contiene una ventana secundaria, la misma que permite a usuarios con más conocimientos, ingresar información adicional para la generación. Esta ventana se presenta en la Fig. 10. En la ventana secundaria del Generador de Flujos se encuentran los siguientes elementos:
1. Nombre del canal: en este campo el usuario puede ingresar el nombre que quiere darle al canal.
 2. Frecuencia central: en este campo el usuario debe ingresar la frecuencia central en MHz, correspondiente al canal en el que se quiere transmitir el flujo.
 3. Código de área: permite ingresar el código de área correspondiente al lugar de transmisión.
 4. Intervalo de guarda: permite seleccionar el intervalo de guarda que se utilizará para transmisión, este parámetro es importante para que no exista interferencia entre símbolos.
 5. Modo de transmisión: permite seleccionar el modo que va utilizar el flujo único de paquetes de transporte TS, este valor está relacionado con el número de portadoras OFDM.
 6. Nombre del proveedor: permite ingresar el nombre del proveedor del servicio, la fuente de donde se obtuvo el video.
 7. Nombre del servicio: permite establecer el nombre que el usuario le asignará a su servicio, de preferencia debería tener relación con el tipo de servicio.
 8. Número identificador de programa: este campo permite al usuario ingresar el número que identifica al programa que se está transmitiendo.
 9. PID de audio: el usuario debe ingresar el número identificador de los paquetes de audio, debe tomar en cuenta que este valor debe estar entre 16 y 8190.
 10. PID de video: el usuario debe ingresar el número identificador de los paquetes de video, debe tomar en cuenta que este valor debe estar entre 16 y 8190.
 11. Relación de aspecto: el usuario puede escoger de acuerdo al tipo de servicio, la relación de aspecto que requiera cada programación.

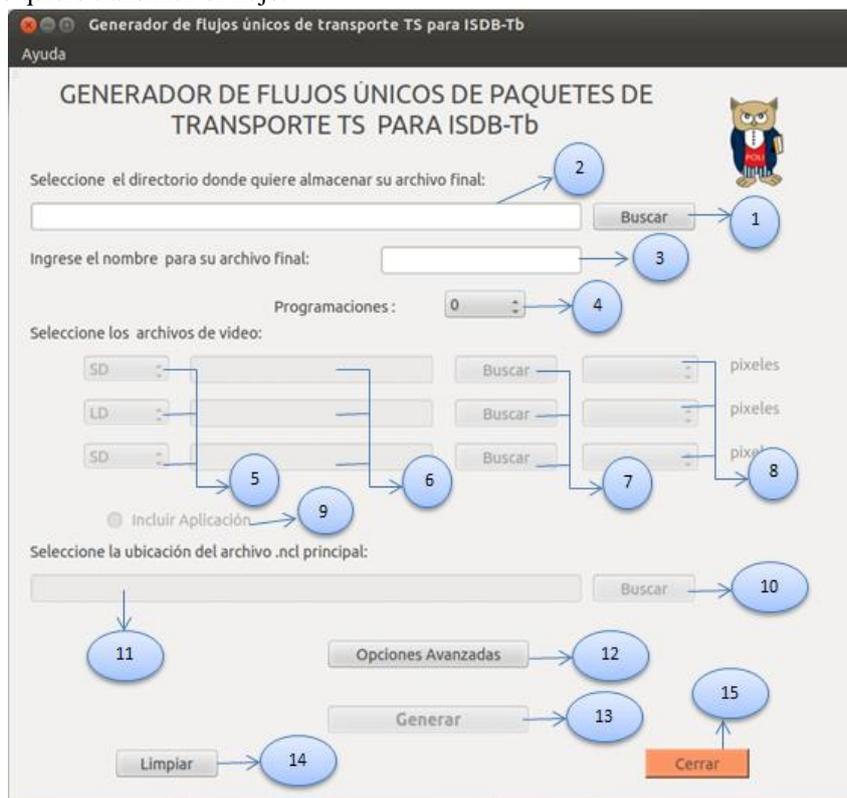


Figura 9. Ventana principal de la aplicación

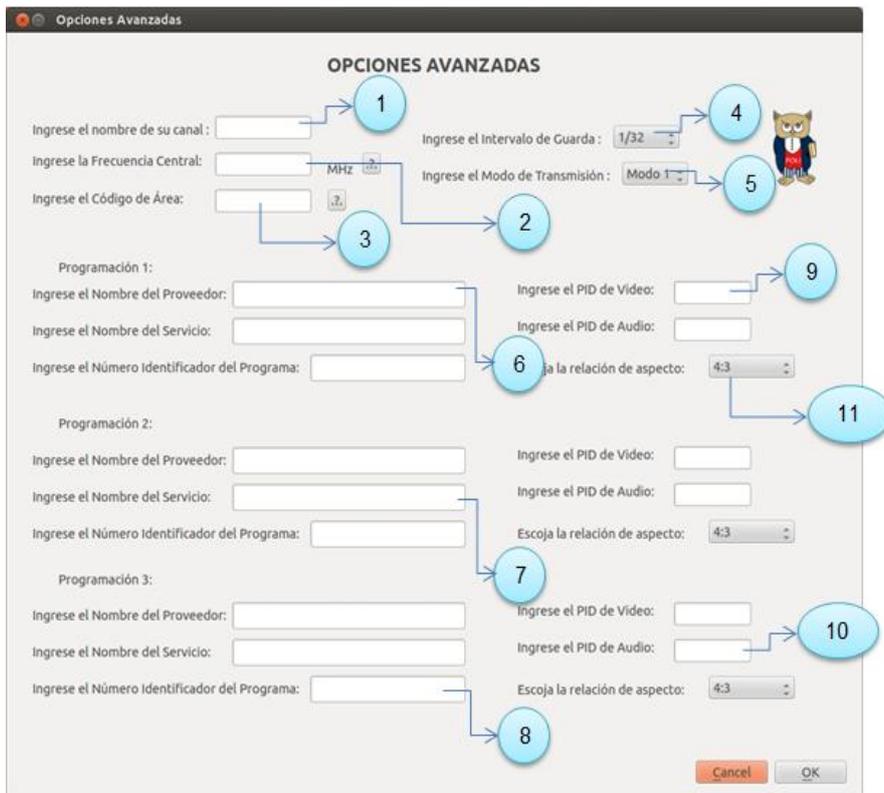


Figura 10. Ventana secundaria de opciones avanzadas

5. RESULTADOS

5.1 Escenario de pruebas

En la Fig. 11 se presenta los elementos empleados para realizar las pruebas con los flujos generados. Entre estos elementos están: PC, dispone del software StreamXpress del transmisor, y del archivo del flujo único de paquetes de transporte TS generado, sus características se resumen en la Tabla 2; transmisor, DEKTEC DTU-215-I-SP, que realiza la respectiva modulación y transmisión del flujo único de paquetes de transporte TS generado, una imagen del dispositivo real se presenta en la Fig. 12; STB: uno marca EiTVDDeveloper Box y uno marca CORADIR CDR1000D, una imagen de estos dispositivos se presenta en la Fig. 13.

Tabla 2. Características del PC utilizado

Tipo	Características
Sistema Operativo	Windows 7
Procesador	AMD Athlon (tm) II Dual-Core M300, 2GHz
Memoria RAM	3 GB
Espacio en Disco	60 GB
Pantalla	Resolución: 1024 x 768, colores de 32 bits

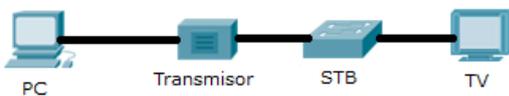


Figura 11. Escenario de pruebas

5.2 Resultados de las pruebas realizadas

Para comprobar el correcto funcionamiento de las aplicaciones, se realizaron muchas pruebas que están reportadas en [19]. Las pruebas incluyeron casos en los que se incluían una o varias programaciones sin utilizar las configuraciones avanzadas y otros casos en los que sí, además de integrar aplicaciones interactivas.



Figura 12. Transmisor DEKTEC DTU-215-I-SP [7]



Figura 13. Set-Top Box Coradir e EiTV[9]

La Fig. 14 indica las capturas de pantalla del software StreamXpress, con los parámetros para la transmisión de dos servicios y la visualización de lo recibido en el STB Coradir y presentado en un televisor. La Fig. 15 presenta los resultados de la recepción de la señal BTS en el STB y luego en el televisor, en este caso se visualizan tres imágenes correspondientes a cada uno de los servicios y una imagen que muestra la lista de programaciones. La Tabla 3 contiene información general del flujo generado e indica: el nombre del archivo, los tipos de servicios y el número de programaciones. La señal BTS generada y luego transmitida, cuenta con tres programaciones, cada una con audio y video sincronizados, la calidad de las tres programaciones es HD, respectivamente; no se visualizó problemas de imágenes pixeladas, ni imágenes congeladas, por lo que se comprobó la correcta generación del flujo con tres servicios.

Es importante mencionar, que debido a que OpenCaster está orientado a generar flujos para el estándar europeo DVB, a pesar de la modificación realizada por LIFIA, los flujos generados emplean la compresión MPEG-2 para el video y MPEG-1 para el audio, lo que no se ajusta a lo señalado en el estándar ISDB-Tb. Por tal motivo, se requiere realizar una transcodificación del video para obtener un flujo transmitido que emplee MPEG-4. El transmisor DEKTEC DTU-215-I-SP ofrece esta funcionalidad por lo que el sistema de pruebas básico presentado permite hacer las pruebas de los flujos incluidas las aplicaciones interactivas.

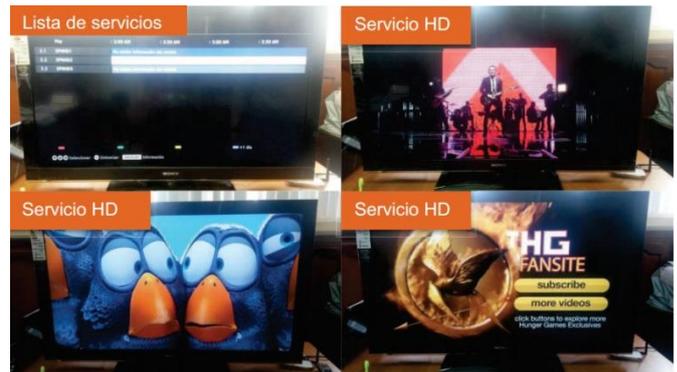


Figura 15. Servicios recibidos en el STB EITV y presentados en un televisor

Tabla 3. Información general del flujo generado

Nombre del archivo final	3serviciosHDHDHD.ts
Tipo de Servicios	HD-HD-HD
Número de programaciones	3

6. CONCLUSIONES

Para lograr la generación de los flujos únicos de paquetes de transporte TS mediante software, una opción es utilizar OpenCaster, que usa líneas de comando con varios argumentos, numerosos en varios casos, y cada uno con valores permisibles sobre los cuales no existe información suficiente, lo que constituye un problema al momento de la generación de estos flujos; problema que se solucionó mediante la consulta de documentos referentes al tema y, principalmente, mediante la realización de un amplio grupo de combinaciones, usando el método de ensayo y error, con lo que se consiguió los parámetros para la correcta generación. Para el desarrollo de la aplicación Generador de Flujos Únicos de Paquetes de Transporte TS, se utilizó Qt con C++, y un conjunto de librerías que facilita la implementación de interfaces gráficas en Linux.

A través del Generador de Flujos, el usuario ya no tiene la necesidad de emplear la línea de comandos, pues la aplicación es la que esconde esos detalles y realiza la comunicación con el terminal de Linux para dar las órdenes a OpenCaster y así realizar la generación.

En base a un gran número de pruebas con diferentes flujos conteniendo más de una aplicación interactiva, se comprobó que estas pruebas no fueron exitosas, por lo que se evidenció que OpenCaster no multiplexa de forma correcta más de una aplicación interactiva. La aplicación desarrollada considera esta limitación, permitiendo que solamente se pueda colocar una aplicación interactiva en un servicio.

El trabajo futuro incluye la tarea de una revisión más profunda para tratar de multiplexar en un mismo flujo varias aplicaciones interactivas asociadas a diferentes servicios, de ser posible.

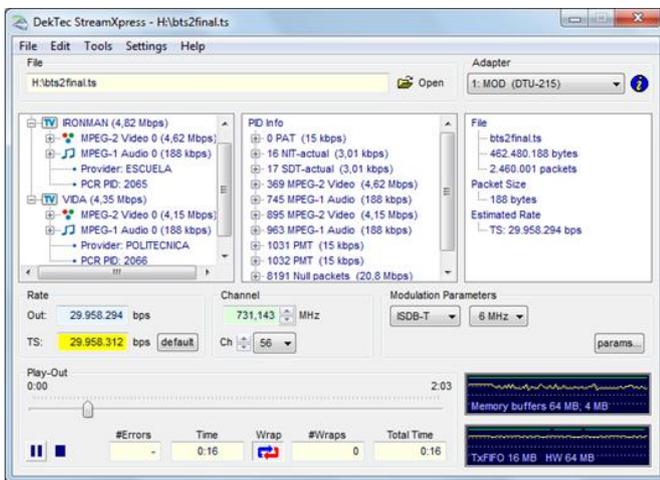


Figura 14. Parámetros de Transmisión empleados en StreamXpress2 para dos servicios y resultados visualizados con el STB Coradir

REFERENCIAS

- [1] ABN NBR 15603-1 Televisión Digital Terrestre - Multiplexación y servicios de información (SI) Parte 1: SI del sistema de radiodifusión, Asociación Brasileira de Normas Técnicas, 2007. Disponible: http://www.abnt.org.br/imagens/Normalizacao_TV_Digital/ABNTNB R15603-1_2007Vc_2008.pdf.
- [2] ABN NBR 15603-2 Televisión Digital Terrestre - Multiplexación y servicios de información (SI) Parte 2: Estructura de datos y definiciones de la información básica de SI, Asociación Brasileira de Normas Técnicas, 2007. Disponible: http://www.abnt.org.br/imagens/Normalizacao_TV_Digital/ABNTNB R15603-2_2007Vc_2008.pdf.
- [3] Anónimo, "MPEG transport stream Pictures, MPEG transport stream Image, Computer Photo Gallery." Disponible: <http://withfriendship.com/user/mithunss/mpeg-transport-stream.php>.
- [4] Anónimo, "UML diagram software - create sequence diagrams, use case diagrams, and more with UML Diagrammer." Disponible: <http://www.pacestar.com/uml/>.
- [5] AVALPA; OpenCaster, Disponible: <http://www.avalpa.com/the-key-values/15-free-software/33-opencaster>.
- [6] D. Vargas, "Selección de contenidos OneSeg a través del Canal de Retorno", Perú, 2013.
- [7] "DekTec - DTU-215 - USB-2 VHF / UHF del modulador." [Online]. Disponible: <http://www.dektec.com/Products/USB/DTU-215/>.
- [8] EITV, <http://www.eitv.com.br/en/produtos/eitv-playout-professional>.
- [9] EITV, "EiTVDeveloper Box - Set-top box ISDB-T y IPTV de desarrollo de aplicativos Ginga (DTVi)." [Online]. Disponible: http://www.eitv.com.br/devbox_es.php.
- [10] INTI, Instituto Nacional De Tecnología Industrial, "INTI Introducción a Televisión Digital. Estándar ISDB-T," Octubre 2011. Disponible: <http://www.inti.gob.ar/electronicaeinformatica/utc/pdf/ISDBT-2011.pdf>.
- [11] J. Valencia e I. Bernal, "Desarrollo de Aplicaciones Interactivas para TV Digital orientadas a formar a la Población en Desastres Naturales". Quito-Ecuador. Revista Politécnica, 2013.
- [12] L. Venegas, "Generación de una trama BroadcastTransportStream (BTS) usando el software libre OpenCaster", Perú, 2012.
- [13] LIFIA, Página oficial, Disponible: <http://www.avalpa.com/>
- [14] LIFIA, Parche para OpenCaster, Disponible: <ftp://tvd.lifia.info.unlp.edu.ar/OpenCaster2.4/>.
- [15] LIFIA, "OpenCaster para SATVD-T", Mayo de 2011.
- [16] M. Sebastini y B. How, "MPEG 101 - Transport Stream Demystification"; Puerto Rico; 2011.
- [17] R. Brice, "Newnes Guide to Digital TV". Elseiver, 2da edición, 2003.
- [18] R. de Bruin and J. Smits, Digital video broadcasting: technology, standards, and regulations. ArtechHouse, 1999.
- [19] T. Moncayo y M. Pozo; "Generación de un Flujo Único de Paquetes de Transporte de acuerdo a la norma ISDB-Tb y Desarrollo de una aplicación para su análisis"; EPN, Quito-Ecuador, 2014.